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, NM 87545

Submitted to the Journal of Computational Physics, Summer 1999

Abstract

We derive a cell-centered 3-D diffusion differencing scheme for arbitrary 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 even on meshes that are not smooth, rigorously treats material discontinuities, and has a symmetric positive-definite coefficient matrix. 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.

1 Introduction

The purpose of this paper is to present a local support-operators diffusion discretization for arbitrary 3-D hexahedral meshes. We use the standard finite-element definition for hexahedra [2]. 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 [1]. The diffusion equation that we seek to solve can be expressed in the following general form:

$$\frac{\partial \phi}{\partial t} - \overrightarrow{\nabla} \cdot D \overrightarrow{\nabla} \phi = Q \quad , \tag{1}$$

where t denotes the time variable, 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, $\overrightarrow{F}$, that we refer to as the flux:

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

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 to essentially any type of diffusion problem.

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:

$$\int_V \frac{\partial \phi}{\partial t} \, dV + \oint_{\partial V} \overrightarrow{F} \cdot \overrightarrow{n} \, dA = \int_V Q \, dV \quad , \tag{3}$$

where V denotes the cell volume, ∂V denotes the cell surface, and $\overrightarrow{n}$ denotes the outward-directed unit surface normal. Note that we used the divergence theorem to convert the second 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 non-orthogonal 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 [2] 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 vertex-centered diffusion schemes [4]. Our new scheme has been developed with coupled radiation-diffusion/hydrodynamics applications in mind.

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 both structured and unstructured meshes consisting of combinations of arbitrary hexahedra and arbitrary degenerate hexahedra (i.e., wedges, pyramids, and tetrahedra). It yields second-order accurate solutions for the intensities on both smooth and non-smooth meshes even when material discontinuities are present, and it generates a sparse symmetric positive-definite coefficient matrix.

The literature relating to cell-centered diffusion discretization schemes for arbitrary hexahedra is not particularly extensive. One of the earliest relevant papers appeared about ten 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 everywhere unity. 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 non-smooth. 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 intensities, 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 like the mixed hybrid method but can sometimes require far fewer unknowns. Although they proved several convergence theorems for their hexahedral-mesh method, 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-difference technique since basis functions never appear in our formalism. The similarities between our method and mixed hybrid finite element methods suggest that there may be a connection between them, but we cannot presently demonstrate such a connection.

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 arbitrary hexahedral meshes.
- It gives second-order convergence of the intensity on both smooth and non-smooth 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 when the mesh is orthogonal.

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 an arbitrary 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 [7]. 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 $-D\overrightarrow{\nabla}$. The diffusion operator of interest is given by the product of the divergence operator and the flux operator: $-\overrightarrow{\nabla}\cdot D\overrightarrow{\nabla}$. 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 [8]. As explained in [8], the adjoint relationship between the flux and divergence operators is embodied in the following integral identity:

$$\oint_{\partial V} \phi \overrightarrow{H} \cdot \overrightarrow{n}\, dA - \int_V D^{-1} \overrightarrow{H} \cdot D \overrightarrow{\nabla} \phi \, dV = \int_V \phi \overrightarrow{\nabla} \cdot \overrightarrow{H}\, dV \quad , \tag{4}$$

where ϕ is an arbitrary scalar function, $\overrightarrow{H}$ is an arbitrary vector function, V denotes a volume, ∂V denotes its surface, and $\overrightarrow{n}$ denotes the outward-directed unit normal associated with that surface. 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 discrete scalar and vector spaces defined in Step 1.

4. Obtain the interior-mesh discretization of Eq. (4) by connecting 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 flux at the cell interfaces.

5. Change the flux operator at those cell faces on the exterior mesh boundary so as to satisfy the appropriate boundary conditions.

6. Combine the global divergence matrix and the global flux matrix to obtain the global diffusion matrix.

The actual method is somewhat more complicated because of the presence of both cell-center and cell-face intensities, but this description nonetheless conveys the central theme of the method.

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, 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.

The vector and matrix notation used from this point forward in this paper is as follows. Each vector is denoted by an upper-case symbol and the components of that vector are denoted by the corresponding lower-case symbol. An arrow is placed over the upper-case symbol if

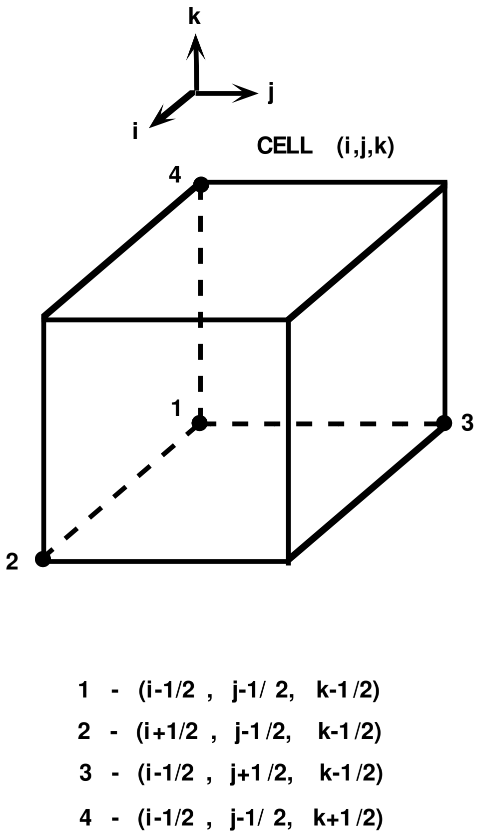

Figure 1: Global indices for four vertices associated with cell (i, j, k).

the vector is physical, while a chevron is placed above the upper-case symbol if the vector is algebraic. Each matrix is denoted by a bold-face upper-case symbol and the elements of that matrix are denoted by the corresponding lower-case symbol.

The intensities (scalars) are defined to exist at both cell center: $\phi^C_{i,j,k}$, and on the face centers: $\phi^L_{i,j,k}$, $\phi^R_{i,j,k}$, $\phi^B_{i,j,k}$, $\phi^T_{i,j,k}$, $\phi^D_{i,j,k}$, $\phi^U_{i,j,k}$. 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 $\phi^R_{i,j,k} \neq \phi^L_{i+1,j,k}$.

Vectors are defined in terms of face-area components located at the face centers: $f^L_{i,j,k}$, $f^R_{i,j,k}$, $f^B_{i,j,k}$, $f^T_{i,j,k}$,$f^D_{i,j,k}$, $f^U_{i,j,k}$, where $f^L_{i,j,k}$ denotes the dot product of $\overrightarrow{F}$ 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 outward-directed unit normal vector over the face, i.e.,

$$\overrightarrow{A} = \oint \overrightarrow{n}\, dA \quad , \tag{5}$$

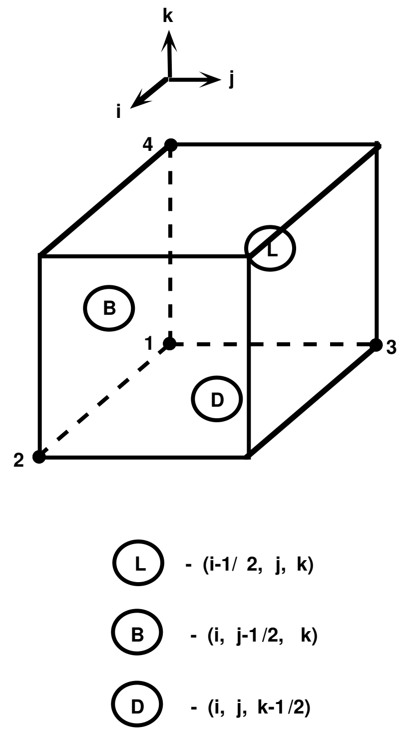

Figure 2: Local and global indices for three of six face centers associated with cell (i, j, k).

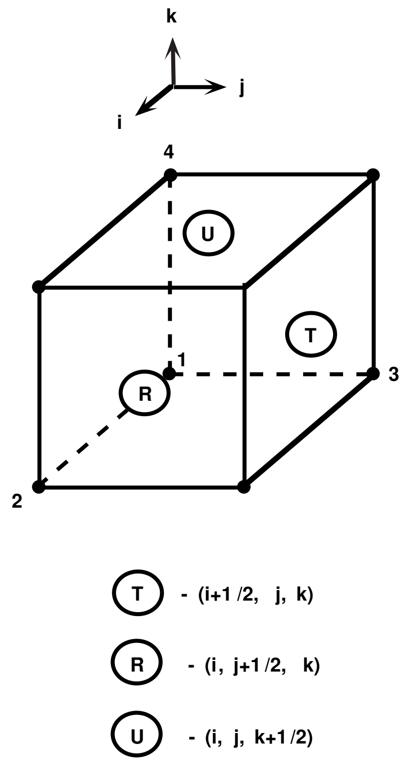

Figure 3: Local and global indices for three of six face centers associated with cell (i, j, k).

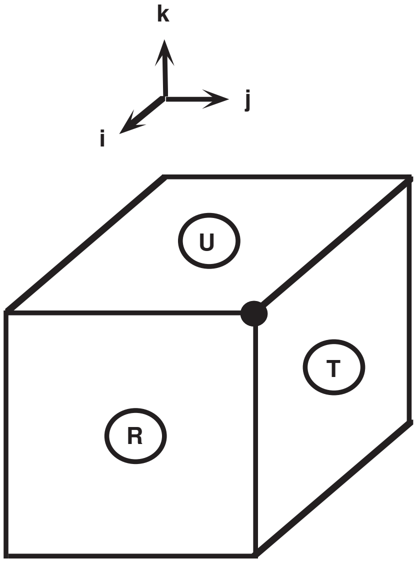

Figure 4: Vertex shared by the Right, Top, and Up faces having local index RTU.

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

$$\left\langle \overrightarrow{n} \right\rangle = \frac{\overrightarrow{A}}{\| \overrightarrow{A} \|} \quad , \tag{6}$$

where $\| \overrightarrow{A} \|$ denotes the magnitude (standard Euclidean norm) of $\overrightarrow{A}$. Equation (6) can be used to convert face-area flux components to face-normal components if desired, e.g.

$$\begin{aligned} \overrightarrow{F} \cdot \left\langle \overrightarrow{n} \right\rangle &= \overrightarrow{F} \cdot \frac{\overrightarrow{A}}{\| \overrightarrow{A} \|} \quad , \\ &= \frac{f}{\| \overrightarrow{A} \|} \quad . \end{aligned} \tag{7}$$

Note that $\| \overrightarrow{A} \|$ 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: $\overrightarrow{F}^{LBD}_{i,j,k}$, $\overrightarrow{F}^{RBD}_{i,j,k}$, $\overrightarrow{F}^{LTD}_{i,j,k}$, $\overrightarrow{F}^{RTD}_{i,j,k}$, $\overrightarrow{F}^{LBU}_{i,j,k}$, $\overrightarrow{F}^{RBU}_{i,j,k}$, $\overrightarrow{F}^{LTU}_{i,j,k}$, $\overrightarrow{F}^{RTU}_{i,j,k}$. Each vertex vector is constructed using the face-area components and area vectors associated with the three faces that share that vertex.

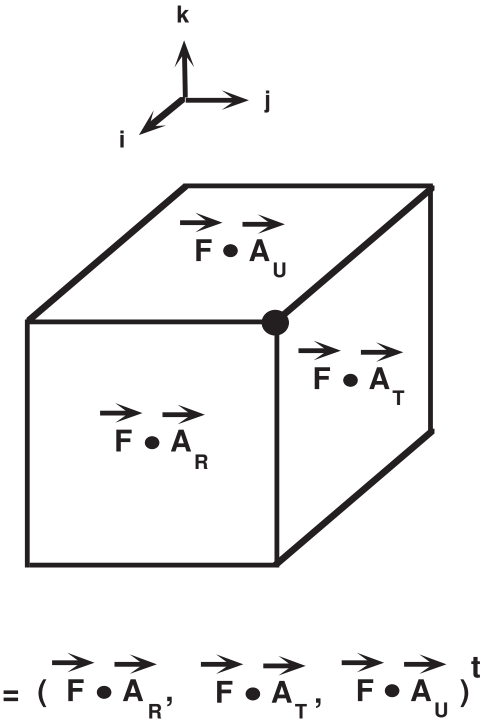

Figure 5: Three face-center face-area components defining the flux vector at vertex RTU.

For instance,

$$\overrightarrow{F}^{LBD}_{i,j,k} = \frac{f^L\left(\overrightarrow{A}^B \times \overrightarrow{A}^D\right)}{\overrightarrow{A}^L \cdot \left(\overrightarrow{A}^B \times \overrightarrow{A}^D\right)} + \frac{f^B\left(\overrightarrow{A}^D \times \overrightarrow{A}^L\right)}{\overrightarrow{A}^L \cdot \left(\overrightarrow{A}^D \times \overrightarrow{A}^L\right)} + \frac{f^D\left(\overrightarrow{A}^L \times \overrightarrow{A}^B\right)}{\overrightarrow{A}^D \cdot \left(\overrightarrow{A}^L \times \overrightarrow{A}^B\right)} \quad . \tag{8}$$

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, $\overrightarrow{F}$, e.g.,

$$\hat{F}_{LBD} = \left(f^L_{i,j,k},\, f^B_{i,j,k},\, f^D_{i,j,k}\right)^t \quad , \tag{9}$$

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. (8) and (9).

As explained in Reference [8], the adjoint relationship between the gradient and divergence operators is embodied in the following integral identity:

$$\oint_{\partial V} \phi \overrightarrow{H} \cdot \overrightarrow{n}\, dA - \int_V D^{-1} \overrightarrow{H} \cdot D \overrightarrow{\nabla} \phi\, dV = \int_V \phi \overrightarrow{\nabla} \cdot \overrightarrow{H}\, dV \quad , \tag{10}$$

where ϕ is an arbitrary scalar function, $\overrightarrow{H}$ is an arbitrary vector function, V denotes a volume, ∂V denotes its surface, and $\overrightarrow{n}$ denotes the outward-directed unit normal associated with that surface. The vector $\overrightarrow{H}$ has the same mesh locations as the flux vector $\overrightarrow{F}$, but is not necessarily equal to $-D\overrightarrow{\nabla}\phi$. We stress that the function ϕ at this point represents an arbitrary scalar function, and not necessarily the solution of the diffusion equation. The next step in our support-operators method is to discretize Eq. (10) over a single arbitrary cell in a special 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. (10) that follows. We first discretize the surface integral:

$$\oint_{\partial V} \phi \overrightarrow{H} \cdot \overrightarrow{n}\, dA \approx \phi^L h^L + \phi^R h^R + \phi^B h^B + \phi^T h^T + \phi^D h^D + \phi^U h^U. \tag{11}$$

Next we approximate the flux volumetric integral:

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

$$\begin{aligned}
&D^{-1}\left(\overrightarrow{H}^{LBD} \cdot \overrightarrow{F}^{LBD}\right)V^{LBD} + D^{-1}\left(\overrightarrow{H}^{RBD} \cdot \overrightarrow{F}^{RBD}\right)V^{RBD}\\
&D^{-1}\left(\overrightarrow{H}^{LTD} \cdot \overrightarrow{F}^{LTD}\right)V^{LTD} + D^{-1}\left(\overrightarrow{H}^{RTD} \cdot \overrightarrow{F}^{RTD}\right)V^{RTD}\\
&D^{-1}\left(\overrightarrow{H}^{LBU} \cdot \overrightarrow{F}^{LBU}\right)V^{LBU} + D^{-1}\left(\overrightarrow{H}^{RBU} \cdot \overrightarrow{F}^{RBU}\right)V^{RBU}\\
&D^{-1}\left(\overrightarrow{H}^{LTU} \cdot \overrightarrow{F}^{LTU}\right)V^{LTU} + D^{-1}\left(\overrightarrow{H}^{RTU} \cdot \overrightarrow{F}^{RTU}\right)V^{RTU} \quad ,
\end{aligned} \tag{12}$$

where $\overrightarrow{F}^{LBD}$ denotes $-D\overrightarrow{\nabla}\phi$ 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. We have investigated several different choices. Specifically:

1. Each vertex weight can be 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

$$V^{LBD} = \frac{1}{8}\overrightarrow{R}_{1,2} \times \overrightarrow{R}_{1,3} \cdot \overrightarrow{R}_{1,4} \quad , \tag{13}$$

   where $\overrightarrow{R}_{i,j}$ denotes the vector from vertex i to vertex j. Note that these vertex weights do not sum to the total volume of the hexahedron unless the hexahedron is a parallelepiped. We refer to these weights as the triple-product weights.

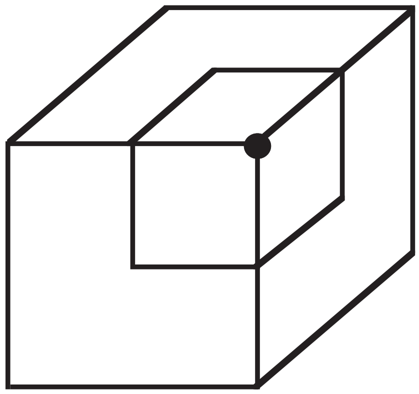

Figure 6: Sub-hexahedron associated with vertex.

2. The weights given in Eq. (13) can be normalized, i.e., multiplied by a single constant, so that they sum to the exact cell volume. We refer to these weights as the normalized triple product weights.

3. Each vertex weight can be set equal to 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 outer 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 is perhaps the most natural choice for the volumetric weights. We refer to these weights as the sub-hexahedron weights.

4. Each vertex weight can be set to one-eighth of the total hexahedron volume. We refer to these weights as the one-eighth weights.

Computational testing indicates that the sub-hexahedron 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-hexahedron 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. Thus we use the normalized triple-product weights.

One can evaluate the dot products in Eq. (12) using Eq. (8), but we find it better for our purposes to evaluate them with the algebraic face-area flux vectors defined by Eq. (9). This is achieved by first transforming the face-area vectors to Cartesian vectors and then taking the dot product. Rather than explicitly define 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:

$$\hat{H}^{LBD} = \left[\mathbf{A}^{LBD}\right]^{-1} \overrightarrow{H}^{LBD} \quad , \tag{14}$$

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

$$\hat{H} = \left(h^L, h^B, h^D\right)^t \quad , \tag{15}$$

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

$$\overrightarrow{H} = (h^x, h^y, h^z)^t \quad , \tag{16}$$

and

$$\left[\mathbf{A}^{LBD}\right]^{-1} = \begin{bmatrix} a_x^L & a_y^L & a_z^L \\ a_x^B & a_y^B & a_z^B \\ a_x^D & a_y^D & a_z^D \end{bmatrix} \quad , \tag{17}$$

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:

$$\begin{aligned} \overrightarrow{H}^{LBD} \cdot \overrightarrow{F}^{LBD} &= \mathbf{A}\hat{H}^{LBD} \cdot \mathbf{A}^{LBD}\hat{F}^{LBD} \quad , \\ &= \hat{H}^{LBD} \cdot \mathbf{S}^{LBD}\hat{F}^{LBD} \quad , \end{aligned} \tag{18}$$

where

$$\mathbf{S}^{LBD} = \left[\mathbf{A}^{LBD}\right]^t \mathbf{A}^{LBD} \quad . \tag{19}$$

Following Eq. (19), We now rewrite Eq. (12) in terms of face-area vectors as follows:

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

$$\begin{aligned} D^{-1}\left(\hat{H}^{LBD} \cdot \mathbf{S}^{LBD}\hat{F}^{LBD}\right)V^{LBD} &+ D^{-1}\left(\hat{H}^{RBD} \cdot \mathbf{S}^{RBD}\hat{F}^{RBD}\right)V^{RBD} \\ D^{-1}\left(\hat{H}^{LTD} \cdot \mathbf{S}^{LTD}\hat{F}^{LTD}\right)V^{LTD} &+ D^{-1}\left(\hat{H}^{RTD} \cdot \mathbf{S}^{RTD}\hat{F}^{RTD}\right)V^{RTD} \\ D^{-1}\left(\hat{H}^{LBU} \cdot \mathbf{S}^{LBU}\hat{F}^{LBU}\right)V^{LBU} &+ D^{-1}\left(\hat{H}^{RBU} \cdot \mathbf{S}^{RBU}\hat{F}^{RBU}\right)V^{RBU} \\ D^{-1}\left(\hat{H}^{LTU} \cdot \mathbf{S}^{LTU}\hat{F}^{LTU}\right)V^{LTU} &+ D^{-1}\left(\hat{H}^{RTU} \cdot \mathbf{S}^{RTU}\hat{F}^{RTU}\right)V^{RTU} \quad . \end{aligned} \tag{20}$$

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. (20) becomes

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

$$
\begin{array}{rcl}
D^{LBD^{-1}} \left(\hat{H}^{LBD} \cdot \mathbf{S}^{LBD} \hat{F}^{LBD} \right) V^{LBD} & + & D^{RBD^{-1}} \left(\hat{H}^{RBD} \cdot \mathbf{S}^{RBD} \hat{F}^{RBD} \right) V^{RBD} \\
D^{LTD^{-1}} \left(\hat{H}^{LTD} \cdot \mathbf{S}^{LTD} \hat{F}^{LTD} \right) V^{LTD} & + & D^{RTD^{-1}} \left(\hat{H}^{RTD} \cdot \mathbf{S}^{RTD} \hat{F}^{RTD} \right) V^{RTD} \\
D^{LBU^{-1}} \left(\hat{H}^{LBU} \cdot \mathbf{S}^{LBU} \hat{F}^{LBU} \right) V^{LBU} & + & D^{RBU^{-1}} \left(\hat{H}^{RBU} \cdot \mathbf{S}^{RBU} \hat{F}^{RBU} \right) V^{RBU} \\
D^{LTU^{-1}} \left(\hat{H}^{LTU} \cdot \mathbf{S}^{LTU} \hat{F}^{LTU} \right) V^{LTU} & + & D^{RTU^{-1}} \left(\hat{H}^{RTU} \cdot \mathbf{S}^{RTU} \hat{F}^{RTU} \right) V^{RTU} \ ,
\end{array} \tag{21}
$$

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

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

$$
\begin{array}{rcl}
\left(\hat{H}^{LBD} \cdot \mathbf{G}^{LBD} \hat{F}^{LBD} \right) V^{LBD} & + & \left(\hat{H}^{RBD} \cdot \mathbf{G}^{RBD} \hat{F}^{RBD} \right) V^{RBD} \\
\left(\hat{H}^{LTD} \cdot \mathbf{G}^{LTD} \hat{F}^{LTD} \right) V^{LTD} & + & \left(\hat{H}^{RTD} \cdot \mathbf{G}^{RTD} \hat{F}^{RTD} \right) V^{RTD} \\
\left(\hat{H}^{LBU} \cdot \mathbf{G}^{LBU} \hat{F}^{LBU} \right) V^{LBU} & + & \left(\hat{H}^{RBU} \cdot \mathbf{G}^{RBU} \hat{F}^{RBU} \right) V^{RBU} \\
\left(\hat{H}^{LTU} \cdot \mathbf{G}^{LTU} \hat{F}^{LTU} \right) V^{LTU} & + & \left(\hat{H}^{RTU} \cdot \mathbf{G}^{RTU} \hat{F}^{RTU} \right) V^{RTU} \quad ,
\end{array} \tag{22}
$$

where

$$
\mathbf{G}^{LBD} = \left[\mathbf{A}^{LBD} \right]^t \left[\mathbf{D}^{LBD} \right]^{-1} \mathbf{A}^{LBD} \quad , \tag{23}
$$

and $\mathbf{D}^{LBD}$ is the Left-Bottom-Down diffusion tensor in the Cartesian basis. The remaining $\mathbf{G}$-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:

$$
\int_V \phi \overrightarrow{\nabla} \cdot \overrightarrow{H} \, dV \approx \phi^C \left[h^L + h^R + h^B + h^T + h^D + h^U \right] \quad . \tag{24}
$$

Equations (11), (20), and (24) are certainly not unique, but they are fairly straightforward. For instance, Eq. (11) represents a face-centered second-order approximation to a surface integral. Equation (20) represents a vertex-based volumetric integral consisting of a dot-product contribution from each pair of vertex vectors. Equation (24) 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 $\overrightarrow{\nabla} \cdot \overrightarrow{H}$ that is analogous to the surface-integral used in Eq. (11).

Substituting from Eqs. (11), (20), and (24) into Eq. (10), we obtain the discrete version of Eq. (10):

$$
\begin{aligned}
&\phi^L h^L + \phi^R h^R + \phi^B h^B + \phi^T h^T + \phi^D h^D + \phi^U h^U + \\
&D^{-1}\left(\hat{H}^{LBD}\cdot \mathbf{S}^{LBD}\hat{F}^{LBD}\right)V^{LBD} + D^{-1}\left(\hat{H}^{RBD}\cdot \mathbf{S}^{RBD}\hat{F}^{RBD}\right)V^{RBD} + \\
&D^{-1}\left(\hat{H}^{LTD}\cdot \mathbf{S}^{LTD}\hat{F}^{LTD}\right)V^{LTD} + D^{-1}\left(\hat{H}^{RTD}\cdot \mathbf{S}^{RTD}\hat{F}^{RTD}\right)V^{RTD} + \\
&D^{-1}\left(\hat{H}^{LBU}\cdot \mathbf{S}^{LBU}\hat{F}^{LBU}\right)V^{LBU} + D^{-1}\left(\hat{H}^{RBU}\cdot \mathbf{S}^{RBU}\hat{F}^{RBU}\right)V^{RBU} + \\
&D^{-1}\left(\hat{H}^{LTU}\cdot \mathbf{S}^{LTU}\hat{F}^{LTU}\right)V^{LTU} + D^{-1}\left(\hat{H}^{RTU}\cdot \mathbf{S}^{LTU}\hat{F}^{RTU}\right)V^{RTU} = \\
&\phi^C\left[h^L + h^R + h^B + h^T + h^D + h^U\right] \quad .
\end{aligned}
\tag{25}
$$

Note that Eq. (25) defines the discrete inner products, discussed in Reference 8, 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 $\hat{H}$ and ϕ values. In particular, the equation for the face-area component of $\overrightarrow{F}$ on any given cell face is obtained from Eq. (25) simply by setting the same face-area component of $\overrightarrow{H}$ on that face to unity and setting the remaining face-area components of $\overrightarrow{H}$ on all other faces to zero. For instance, we obtain the equation for f^L from Eq. (25) by setting h^L to unity and all the other face-area components of $\overrightarrow{H}$, i.e., h^R, h^B, h^T, h^D, h^U, to zero:

$$
\begin{aligned}
\phi^L + \quad & D^{-1}\left(s^{LBD}_{L,L} f^L + s^{LBD}_{L,B} f^B + s^{LBD}_{L,D} f^D\right)V^{LBD} \\
+ \quad & D^{-1}\left(s^{LTD}_{L,L} f^L + s^{LTD}_{L,T} f^T + s^{LTD}_{L,D} f^D\right)V^{LTD} \\
+ \quad & D^{-1}\left(s^{LBU}_{L,L} f^L + s^{LBU}_{L,B} f^B + s^{LBU}_{L,U} f^U\right)V^{LBU} \\
+ \quad & D^{-1}\left(s^{LTU}_{L,L} f^L + s^{LTU}_{L,T} f^T + s^{LTU}_{L,U} f^U\right)V^{LTU} \quad = \phi^C \quad ,
\end{aligned}
\tag{26}
$$

where $s^{LBD}_{L,L}$ denotes the (L,L) element of the matrix $\mathcal{S}^{LBD}$ defined by Eq. (19). The remaining **S**-matrix elements are defined analogously. We obtain the equation for f^R from Eq. (25) by setting h^R to unity and all the other face components of $\overrightarrow{H}$ to zero:

$$
\begin{aligned}
\phi^R + \quad & D^{-1}\left(s^{RBD}_{R,R} f^R + s^{RBD}_{R,B} f^B + s^{RBD}_{R,D} f^D\right)V^{RBD} \\
+ \quad & D^{-1}\left(s^{RTD}_{R,R} f^R + s^{RTD}_{R,T} f^T + s^{RTD}_{R,D} f^D\right)V^{RTD} \\
+ \quad & D^{-1}\left(s^{RBU}_{R,R} f^R + s^{RBU}_{R,B} f^B + s^{RBU}_{R,U} f^U\right)V^{RBU} \\
+ \quad & D^{-1}\left(s^{RTU}_{R,R} f^R + s^{RTU}_{R,T} f^T + s^{RTU}_{R,U} f^U\right)V^{RTU} \quad = \phi^C \quad .
\end{aligned}
\tag{27}
$$

We obtain the equation for f^B from Eq. (25) by setting h^B to unity and all the other face components of $\overrightarrow{H}$ to zero:

$$\begin{aligned} \phi^B + \quad & D^{-1}\left(s^{LBD}_{B,L} f^L + s^{LBD}_{B,B} f^B + s^{LBD}_{B,D} f^D\right) V^{LBD} \\ + \quad & D^{-1}\left(s^{RBD}_{B,R} f^R + s^{RBD}_{B,B} f^B + s^{RBD}_{B,D} f^D\right) V^{RBD} \\ + \quad & D^{-1}\left(s^{LBU}_{B,L} f^L + s^{LBU}_{B,B} f^B + s^{LBU}_{B,U} f^U\right) V^{LBU} \\ + \quad & D^{-1}\left(s^{RBU}_{B,R} f^R + s^{RBU}_{B,B} f^B + s^{RBU}_{B,U} f^U\right) V^{RBU} \quad = \phi^C \quad . \end{aligned} \tag{28}$$

We obtain the equation for f^T from Eq. (25) by setting h^T to unity and all the other face components of $\overrightarrow{H}$ to zero:

$$\begin{aligned} \phi^T + \quad & D^{-1}\left(s^{LTD}_{T,L} f^L + s^{LTD}_{T,T} f^T + s^{LTD}_{T,D} f^D\right) V^{LTD} \\ + \quad & D^{-1}\left(s^{RTD}_{T,R} f^R + s^{RTD}_{T,T} f^T + s^{RTD}_{T,D} f^D\right) V^{RTD} \\ + \quad & D^{-1}\left(s^{LTU}_{T,L} f^L + s^{LTU}_{T,T} f^T + s^{LTU}_{T,U} f^U\right) V^{LTU} \\ + \quad & D^{-1}\left(s^{RTU}_{T,R} f^R + s^{RTU}_{T,T} f^T + s^{RTU}_{T,U} f^U\right) V^{RTU} \quad = \phi^C \quad . \end{aligned} \tag{29}$$

We obtain the equation for f^D from Eq. (25) by setting h^D to unity and all the other face components of $\overrightarrow{H}$ to zero:

$$\begin{aligned} \phi^D + \quad & D^{-1}\left(s^{LBD}_{D,L} f^L + s^{LBD}_{D,B} f^B + s^{LBD}_{D,D} f^D\right) V^{LBD} \\ + \quad & D^{-1}\left(s^{RBD}_{D,R} f^R + s^{RBD}_{D,B} f^B + s^{RBD}_{D,D} f^D\right) V^{RBD} \\ + \quad & D^{-1}\left(s^{LTD}_{D,L} f^L + s^{LTD}_{D,T} f^T + s^{LTD}_{D,D} f^D\right) V^{LTD} \\ + \quad & D^{-1}\left(s^{RTD}_{D,R} f^R + s^{RTD}_{D,T} f^T + s^{RTD}_{D,D} f^D\right) V^{RTD} \quad = \phi^C \quad . \end{aligned} \tag{30}$$

Finally, we obtain the equation for f^U from Eq. (25) by setting h^U to unity and all the other face components of $\overrightarrow{H}$ to zero:

$$\begin{aligned} \phi^U + \quad & D^{-1}\left(s^{LBU}_{U,L} f^L + s^{LBU}_{U,B} f^B + s^{LBU}_{U,U} f^U\right) V^{LBU} \\ + \quad & D^{-1}\left(s^{RBU}_{U,R} f^R + s^{RBU}_{U,B} f^B + s^{RBU}_{U,U} f^U\right) V^{RBU} \\ + \quad & D^{-1}\left(s^{LTU}_{U,L} f^L + s^{LTU}_{U,T} f^T + s^{LTU}_{U,U} f^U\right) V^{LTU} \\ + \quad & D^{-1}\left(s^{RTU}_{U,R} f^R + s^{RTU}_{U,T} f^T + s^{RTU}_{U,U} f^U\right) V^{RTU} \quad = \phi^C \quad . \end{aligned} \tag{31}$$

Equations (26) through (31) can be expressed in matrix form as follows:

$$\mathbf{W}^{-1}\hat{\mathcal{F}} = \Delta\hat{\Phi} \quad , \tag{32}$$

where

$$\hat{\mathcal{F}} = \left(f^L, f^R, f^B, f^T, f^D, f^U\right)^t \quad , \tag{33}$$

and

$$\Delta\hat{\Phi} = \left(\phi^C - \phi^L, \phi^C - \phi^R, \phi^C - \phi^B, \phi^C - \phi^T, \phi^C - \phi^D, \phi^C - \phi^U\right)^t \quad . \tag{34}$$

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. (32):

$$\hat{\mathcal{F}} = \mathbf{W}\Delta\hat{\Phi} \quad . \tag{35}$$

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 a rectangular mesh, $\mathbf{W}$ becomes diagonal and can be trivially inverted. For instance, under this assumption, Eq. (26) becomes:

$$\phi^L + D^{-1}\left(\Delta y \Delta z\right)^{-2} f^L \frac{\Delta x \Delta y \Delta z}{2} = \phi_C \quad , \tag{36}$$

where we have also assumed that the indices i, j, k, correspond to the spatial coordinates x, y, z, respectively. Solving Eq. (36) for f^L, we obtain

$$f^L = -\frac{2D}{\Delta x}\left(\phi^L - \phi^C\right)\Delta y \Delta z \quad , \tag{37}$$

which is exact for ϕ linearly-dependent upon x.

Having derived Eq. (35), 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. (25). 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:

$$\frac{\partial \phi^C}{\partial t} V + f^L + f^R + f^B + f^T + f^D + f^U = Q^C V \quad , \tag{38}$$

where V denotes the total volume of the cell, the face-area flux components are expressed in terms of the intensities via Eq. (35), and Q^C denotes the source or driving function evaluated at cell-center. We have chosen not to discretize the time derivative in Eq. (38) simply because essentially any standard discretization, e.g., the backward-Euler and Crank-Nicholson schemes [9], can be applied in conjunction with our spatial discretization. Equation (38) 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. (25), 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. (25) 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 the surface integral in Eq. (25) 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. (25) 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

$$\phi^R_{i,j,k} = \phi^L_{i+1,j,k} \equiv \phi_{i+\frac{1}{2},j,k} \quad , \tag{39}$$

$$\phi^T_{i,j,k} = \phi^B_{i,j+1,k} \equiv \phi_{i,j+\frac{1}{2},k} \quad , \tag{40}$$

$$\phi^U_{i,j,k} = \phi^D_{i,j,k+1} \equiv \phi_{i,j,k+\frac{1}{2}} \quad , \tag{41}$$

$$f^R_{i,j,k} + f^L_{i+1,j,k} = 0 \quad , \tag{42}$$

$$f^T_{i,j,k} + f^B_{i,j+1,k} = 0 \quad , \tag{43}$$

$$f^U_{i,j,k} + f^D_{i,j,k+1} = 0 \quad , \tag{44}$$

where the indices in Eqs. (39) through (44) take on all values associated with interior cell faces, and the flux components in Eqs. (42) through (44) are expressed in terms of intensities via Eq. (35). One would expect that the continuity of the face-area flux components expressed by Eqs. (42) through (44) 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 (39) through (41) establish that there is only one intensity unknown associated with each interior-mesh cell face. Thus, as shown in Eqs. (39) through (41), each such intensity can be uniquely referred to using a global mesh index. The equations for these intensities are given by Eqs. (42) through (44). For instance, Eq. (42) is the equation for $\phi_{i+\frac{1}{2},j,k}$. In general, Eq. (42) 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. (43) and (44) are completely analogous to those of Eq. (42).

If the mesh is orthogonal, Eqs. (42) through (44) 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. This is completely analogous to the 2-D case discussed in detail in [1]. However, if the mesh is non-orthogonal, the face-center intensities cannot be eliminated, and Eqs. (42) through (44) must be included in the diffusion matrix. In this case, these equations must be reversed in sign to obtain a symmetric diffusion matrix:

$$-f^R_{i,j,k} - f^L_{i+1,j,k} = 0 \quad , \tag{45}$$

$$-f^T_{i,j,k} - f^B_{i,j+1,k} = 0 \quad , \tag{46}$$

$$-f^U_{i,j,k} - f^D_{i,j,k+1} = 0 \quad . \tag{47}$$

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. (35). A ghost cell is a non-existent 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 follows:

$$\phi + d^e \overrightarrow{\nabla} \phi \cdot \overrightarrow{n} = \phi^e \quad , \tag{48}$$

where d^e is called the extrapolation distance, ϕ^e is called the extrapolated intensity (a specified function), and $\overrightarrow{n}$ denotes an outward-directed unit normal vector. Equation (48) 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 non-zero, 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. (48):

$$\phi - \frac{d^e}{D} \overrightarrow{F}^g \cdot \overrightarrow{n} = \phi^e \quad , \tag{49}$$

where $\overrightarrow{F}^g$ 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

$$\overrightarrow{n}^g = -\overrightarrow{n} \quad , \tag{50}$$

where $\overrightarrow{n}^g$ denotes a ghost-cell outward-directed unit normal vector. Substituting from Eq. (50) into Eq. (49), we obtain:

$$\phi + \frac{d^e}{D} \overrightarrow{F}^g \cdot \overrightarrow{n}^g = \phi^e \quad , \tag{51}$$

Next we solve Eq. (51) for the outward-directed flux component associated with a ghost cell:

$$\overrightarrow{F}^{g} \cdot \overrightarrow{n}^{g} = \frac{D}{d^e} \left(\phi^e - \phi\right) \qquad . \tag{52}$$

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. (52) 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:

$$f^R_{0,j,k} = -\frac{D_{1,j,k}}{d^e_{0,j,k}} \left(\phi_{\frac{1}{2},j,k} - \phi^e_{0,j,k}\right) \| \overrightarrow{A}^{L}_{1,j,k} \| \qquad , \tag{53}$$

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:

$$-f^R_{0,j,k} - f^L_{1,j,k} = 0 \qquad . \tag{54}$$

Note that Eq. (54) is identical to Eq. (45) with the latter equation evaluated at $i = 0$. Thus Eqs. (45) through (47) 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. (35). Note that Eq. (54) 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. (35).
- The discrete equation for each cell-centered intensity is given in Eq. (38).
- The equations for the interior-mesh face-centered intensities are given in Eqs. (45) through (47).
- The equation for a face-center intensity on the outer mesh boundary is given by Eqs. (53) and (54) when the boundary face is the Left face of a cell. Analogous equations for the other five cases are easily derived using Eqs. (45) through (47) and Eq. (53).

It is interesting to note that the equation for a cell-center intensity contains a time-derivative of that intensity, but the equations for the face-center equations do not contain any form of time derivative. Thus in time-dependent calculations, one must have initial values for the cell-center intensities, but initial values are not required for the face-center intensities. Thus only cell-center intensities must be saved from one time step to the next.

We have already shown that our diffusion matrix is sparse. It is also symmetric positive-definite. We demonstrate this latter property in the Appendix.

3 Solution of the Equations

We use a preconditioned conjugate-gradient method [7] to solve our discretized diffusion equations. The preconditioner is completely analogous to that used for the 2-D local support-operators scheme. [1] It is obtained simply by setting the off-diagonal elements of all the **S**-matrices, defined by Eq. (19), to zero. This makes the **W**-matrices diagonal, which ultimately results in a 7-point pure cell-center diffusion equation. That is to say that the balance equation for each cell in the mesh interior contains only the cell-center intensity in that cell together with the cell-center intensities in the six cells sharing a face with that cell. Each face-center intensity is expressed in terms of the cell-center intensities in the two cells that share that face-center intensity. For instance, if we set the off-diagonal elements of the **S**-matrices to zero, Eqs. (26) and (27) yield:

$$f^L_{i+1,j,k} = -\frac{2D_{i+1,j,k}}{\Delta^L_{i+1,j,k}}\left(\phi_{i+\frac{1}{2},j,k} - \phi_{i+1,j,k}\right) \quad , \tag{55}$$

and

$$f^R_{i,j,k} = -\frac{2D_{i,j,k}}{\Delta^R_{i,j,k}}\left(\phi_{i+\frac{1}{2},j,k} - \phi_{i,j,k}\right) \quad , \tag{56}$$

respectively, where

$$\Delta^L_{i+1,j,k} = 2\left[s^{LBD}_{L,L}V^{LBD} + s^{LTD}_{L,L}V^{LTD} + s^{LBU}_{L,L}V^{LBU} + s^{LTU}_{L,L}V^{LTU}\right]_{i+1,j,k} \quad , \tag{57}$$

$$\Delta^R_{i,j,k} = 2\left[s^{RBD}_{R,R}V^{RBD} + s^{RTD}_{R,R}V^{RTD} + s^{RBU}_{R,R}V^{RBU} + s^{RTU}_{R,R}V^{RTU}\right]_{i,j,k} \quad . \tag{58}$$

Substituting from Eqs. (55) and (56), into Eq. (45), we get the equation for $\phi_{i+\frac{1}{2},j,k}$:

$$\frac{2D_{i,j,k}\left(\phi_{i+\frac{1}{2},j,k} - \phi_{i,j,k}\right)}{\Delta^R_{i,j,k}} + \frac{2D_{i+1,j,k}\left(\phi_{i+\frac{1}{2},j,k} - \phi_{i+1,j,k}\right)}{\Delta^L_{i+1,j,k}} = 0 \quad , \tag{59}$$

Solving Eq. (59) for $\phi_{i+\frac{1}{2},j,k}$, we get:

$$\phi_{i+\frac{1}{2},j,k} = \left(\phi_{i,j,k}\frac{D_{i,j,k}}{\Delta^R_{i,j,k}} + \phi_{i+1,j,k}\frac{D_{i+1,j,k}}{\Delta^L_{i+1,j,k}}\right) \Bigg/ \left(\frac{D_{i,j,k}}{\Delta^R_{i,j,k}} + \frac{D_{i+1,j,k}}{\Delta^L_{i+1,j,k}}\right) \quad . \tag{60}$$

Thus we see from Eq. (60) 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. (60) into Eqs. (55) and (56) 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:

$$f^R_{i,j,k} = -f^L_{i+1,j,k} = -\frac{D_{i+\frac{1}{2},j,k}}{\Delta_{i+\frac{1}{2},j,k}} \left(\phi_{i+1,j,k} - \phi_{i,j,k}\right) \quad , \tag{61}$$

where

$$D_{i+\frac{1}{2},j,k} = \left[\left(\frac{\Delta^R_{i,j,k}}{D_{i,j,k}} + \frac{\Delta^L_{i+1,j,k}}{D_{i+1,j,k}} \right) \Big/ \left(\Delta^R_{i,j,k} + \Delta^L_{i+1,j,k} \right) \right]^{-1} \quad , \tag{62}$$

and

$$\Delta_{i+\frac{1}{2},j,k} = \frac{\Delta^R_{i,j,k} + \Delta^L_{i+1,j,k}}{2} \quad . \tag{63}$$

Thus we see that each interior-mesh face-area flux can be expressed in terms of a difference between the two adjacent cell-center intensities. Given Eq. (61), it is not difficult to see that the balance equation, Eq. (38), can be constructed from the cell-center intensities alone, resulting in 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

$$\begin{aligned} \frac{\partial \phi_{i,j,k}}{\partial t} V_{i,j,k} + \quad & -\frac{D_{i+\frac{1}{2},j,k}}{\Delta_{i+\frac{1}{2},j,k}} \left(\phi_{i+1,j,k} - \phi_{i,j,k}\right) + \frac{D_{i-\frac{1}{2},j,k}}{\Delta_{i-\frac{1}{2},j,k}} \left(\phi_{i,j,k} - \phi_{i-1,j,k}\right) \\ & -\frac{D_{i,j+\frac{1}{2},k}}{\Delta_{i,j+\frac{1}{2},k}} \left(\phi_{i,j+1,k} - \phi_{i,j,k}\right) + \frac{D_{i,j-\frac{1}{2},k}}{\Delta_{i,j-\frac{1}{2},k}} \left(\phi_{i,j,k} - \phi_{i,j-1,k}\right) \\ & -\frac{D_{i,j,k+\frac{1}{2}}}{\Delta_{i,j,k+\frac{1}{2}}} \left(\phi_{i,j,k+1} - \phi_{i,j,k}\right) + \frac{D_{i,j,k-\frac{1}{2}}}{\Delta_{i,j,k-\frac{1}{2}}} \left(\phi_{i,j,k} - \phi_{i,j,k-1}\right) \quad = Q_{i,j,k} V_{i,j,k} \quad . \end{aligned} \tag{64}$$

To obtain the analog of Eq. (61) 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. (53) and (55) into Eq. (54), we obtain the equation for $\phi_{\frac{1}{2},j,k}$:

$$\frac{2D_{1,j,k}}{\Delta^R_{0,j,k}} \left(\phi_{\frac{1}{2},j,k} - \phi^e_{\frac{1}{2},j,k} \right) + \frac{2D_{1,j,k}}{\Delta^L_{1,j,k}} \left(\phi_{\frac{1}{2},j,k} - \phi_{1,j,k} \right) = 0 \quad , \tag{65}$$

where

$$\Delta^R_{0,j,k} = \frac{2d^e_{0,j,k}}{\|\overrightarrow{A}^L_{1,j,k}\|} \quad . \tag{66}$$

Solving Eq. (65) for $\phi_{\frac{1}{2},j,k}$. we get

$$\phi_{\frac{1}{2},j,k} = \left(\phi_{0,j,k} \frac{D_{1,j,k}}{\Delta^R_{0,j,k}} + \phi_{1,j,k} \frac{D_{1,j,k}}{\Delta^L_{1,j,k}} \right) \Big/ \left(\frac{D_{1,j,k}}{\Delta^R_{0,j,k}} + \frac{D_{1,j,k}}{\Delta^L_{1,j,k}} \right) \quad . \tag{67}$$

Substituting from Eq. (67) into Eqs. (53) and (55), we obtain the desired expression for the face-area flux component on a boundary face:

$$f^R_{0,j,k} = -f^L_{1,j,k} = -\frac{D_{\frac{1}{2},j,k}}{\Delta_{\frac{1}{2},j,k}} \left(\phi_{\frac{1}{2},j,k} - \phi^e_{0,j,k} \right) \quad , \tag{68}$$

where

$$D_{\frac{1}{2},j,k} = \left[\left(\frac{\Delta^R_{0,j,k}}{D_{1,j,k}} + \frac{\Delta^L_{1,j,k}}{D_{1,j,k}} \right) \Big/ \left(\Delta^R_{0,j,k} + \Delta^L_{1,j,k} \right) \right]^{-1} \quad , \tag{69}$$

and $\Delta_{\frac{1}{2},j,k}$ is given by Eq. (63) evaluated with $i = 0$ and Eq. (66). This completes the derivation of the approximate cell-center diffusion scheme used to precondition the full cell-center/cell-face scheme.

Once the 7-point discretization equations have been solved for the cell-center intensities, the face-center intensities can be directly calculated. In particular, Eq. (60) and its analogs for the Bottom/Top and Down/Up faces are used to calculate the face-center intensities on the mesh interior, while Eq. (65) and its analogs for the Bottom/Top and Down/Up faces are used to calculate the face-center intensities on the outer mesh boundary. The 7-point cell-center discretization yields an SPD coefficient matrix. Most importantly, since the **S**-matrices are diagonal when the mesh is orthogonal, it follows that the 7-point cell-center discretization is *equivalent to the full cell-center/face-center discretization 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 preconditioning system has roughly one-fourth as many rows and one-sixth as many elements as the full-system coefficient matrix. Computational results presented in the next section confirm this expectation.

In closing this section we note that when the 7-point system is used for preconditioning purposes, an inhomogeneous source term 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 eliminated to obtain the 7-point cell-center system. This matter is extensively discussed for the 2-D case in [1].

4 Computational Results

In this section we perform two sets of calculations. The first set demonstrates that our support-operators method converges with second-order accuracy for a problem with a material discontinuity and a non-smooth mesh. The second set demonstrates the effectiveness of our preconditioner as a function of mesh skewness. There are three types of meshes used in all of the calculations: orthogonal, random, 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 orthogonal mesh is composed of uniform cubic cells having a characteristic length, l_c. The random meshes represent randomly distorted orthogonal grids. In particular, each vertex on the mesh interior is randomly relocated within a sphere of radius r_0, where $r_0 = 0.25 l_c$. These random meshes are both non-smooth and skewed, but these properties are approximately constant independent of the mesh size. The Kershaw-squared meshes are a 3-D variation on the 2-D Kershaw meshes that first appeared in [13]. An example of a $20 \times 20 \times 20$ Kershaw-squared mesh is shown in Fig. 7. This mesh becomes

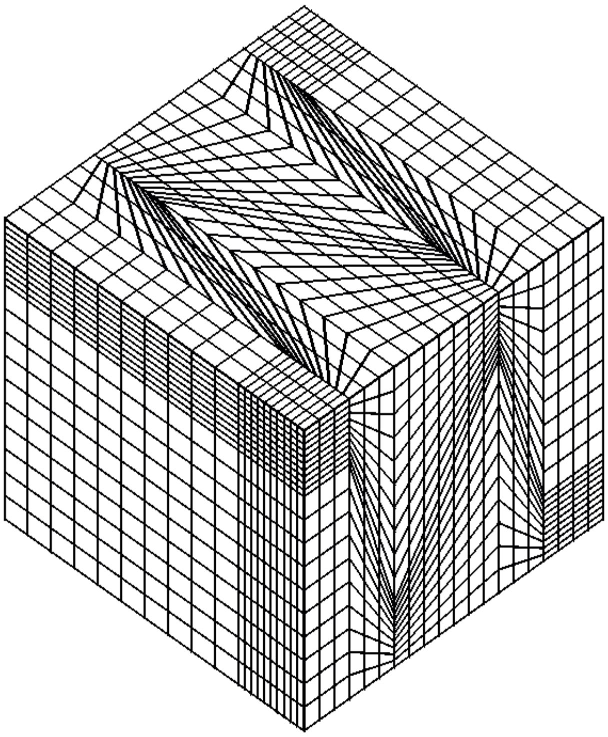

Figure 7: A $20 \times 20 \times 20$ Kershaw-squared mesh.

increasingly non-smooth and skewed as the mesh size is increased.

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

$$-D(z)\frac{\partial \phi}{\partial z} = Qz^2 \qquad , \tag{70}$$

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

$$D(z) = \begin{cases} D_1 & \text{, for } z \in [0, 0.5], \\ D_2 & \text{, for } z \in [0.5, 1], \end{cases} \tag{71}$$

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 problem. The exact solution to the two-material problem is:

$$\phi = \begin{cases} a + b + c_1 z^4 & \text{, for } z \in [0, 0.5], \\ a + c_2 z^4 & \text{, for } z \in [0.5, 1.0], \end{cases} \tag{72}$$

where

$$a = \frac{Q(1 + 8D_2)}{12D_2} \quad , b = \frac{Q\,(D_2 - D_1)}{192 D_1 D_2} \quad , c_1 = -\frac{Q}{12D_1} \quad , c_2 = -\frac{Q}{12D_2} \quad . \tag{73}$$

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

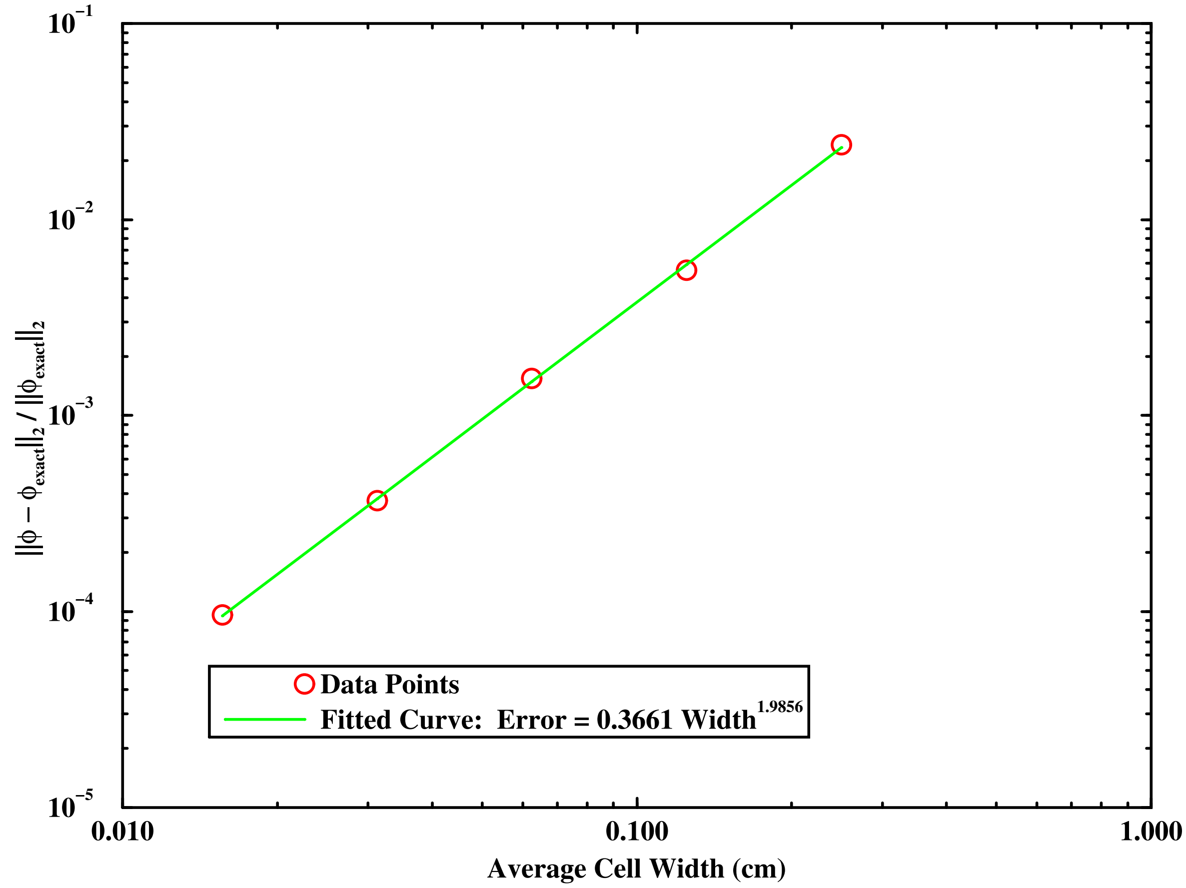

Figure 8: Plot of convergence data and least-squares fit to data.

performed several calculations for the two-material problem with meshes of various sizes. Each calculation uses a mesh with an average cell width that is half that of the preceding calculation. 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., $\|\hat{\phi}_{exact} - \hat{\phi}_{computed}\|_2 \big/ \|\hat{\phi}_{exact}\|_2$. The errors are plotted as a function of average cell length in Fig. 8 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 function 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 random meshes.

The problem associated with the second set of calculations can be described as follows:

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

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.

Table I: 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

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 solve the resulting system using the conjugate-gradient method in conjunction with symmetric successive over-relaxation (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 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 is given for each calculation in Table I. It can be seen from Table I 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. The reduction in iterations observed for the random mesh is indicative of the reduction seen in well-formed meshes. 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 time 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 [1]. 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 [14], 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.

Appendix

The purpose of this appendix is to demonstrate that the coefficient matrix for our support-operators method is SPD (symmetric positive-definite) . This is achieved in the following manner. First we demonstrate that the $\mathbf{W}$ matrix is SPD. Next we show that the the coefficient matrix for a single-cell problem with reflective boundary conditions is SPS (symmetric positive-semidefinite) 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 [1] 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 [1].

The following mathematical preliminaries are discussed in [7]. A matrix, $\mathbf{B}$ is symmetric if and only if

$$\mathbf{B} = \mathbf{B}^t \quad . \tag{75}$$

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

$$\hat{X}^t\mathbf{B}\hat{X} > 0 \quad , \text{for all vectors } \hat{X}. \tag{76}$$

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

$$\hat{X}^t\mathbf{B}\hat{X} \geq 0 \quad , \text{for all vectors } \hat{X}. \tag{77}$$

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 follows:

$$\mathbf{B} = \mathbf{K}^t\mathbf{K} \quad . \tag{78}$$

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. (35), $\mathbf{W}$, is SPD. It suffices to show that its inverse, explicitly given in Eqs. (26) through (31), is SPD. We begin the construction of $\mathbf{W}^{-1}$ by considering Eq. (25) and the $\mathbf{S}$-matrices that appear in it. Each of the $\mathbf{S}$-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:

$$\hat{F}^{LBD} = \left(f^L, f^B, f^D\right)^t \quad . \tag{79}$$

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

$$\hat{\mathcal{F}} = \left(f^L, f^R, f^B, f^T, f^D, f^U\right)^t \quad . \tag{80}$$

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

$$\mathbf{S}^{LBD}_{6\times 6} = \mathbf{P}^{LBD^t}\mathbf{S}^{LBD}\mathbf{P}^{LBD} \quad , \tag{81}$$

where

$$\mathbf{P}^{LBD}_{L,L} = \mathbf{P}^{LBD}_{B,B} = \mathbf{P}^{LBD}_{D,D} = 1 \quad , \tag{82}$$

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

$$\mathbf{P}^{LBD^t}\mathbf{S}^{LBD}\mathbf{P}^{LBD} = \begin{bmatrix} s_{L,L} & 0 & s_{L,B} & 0 & s_{L,D} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ s_{B,L} & 0 & s_{B,B} & 0 & s_{B,D} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ s_{D,L} & 0 & s_{D,B} & 0 & s_{D,D} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad . \tag{83}$$

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.,

$$\left(f^L, f^B, f^D\right)^t \rightarrow (f_1, f_2, f_3)^t \quad , \tag{84}$$

and

$$\left(f^L, f^R, f^B, f^T, f^D, f^U\right)^t \rightarrow (f_1, f_2, f_3, f_4, f_5, f_6)^t \quad . \tag{85}$$

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

$$p_{i,j} = 1 \quad , \tag{86}$$

otherwise

$$p_{i,j} = 0 \quad . \tag{87}$$

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. (25) as follows:

$$\hat{\mathcal{H}}^t\hat{\Phi} + D^{-1}\sum_{n=1}^{8} V_n\hat{\mathcal{H}}^t\mathbf{P}^t_n\mathbf{S}_n\mathbf{P}_n\hat{\mathcal{F}} = \hat{\mathcal{H}}^t\left(\phi^C\hat{1}\right) \quad , \tag{88}$$

where n is the numeric vertex index, and where

$$\hat{1} = (1, 1, 1, 1, 1, 1)^t \quad , \tag{89}$$

$$\hat{\Phi} = \left(\phi^L, \phi^R, \phi^B, \phi^T, \phi^D, \phi^U\right)^t \quad , \tag{90}$$

$$\hat{\mathcal{H}} = \left(h^L, h^R, h^B, h^T, h^D, h^U\right)^t \quad . \tag{91}$$

Since Eq. (88) must hold for all possible $\hat{\mathcal{H}}$, it follows that

$$\hat{\Phi} + D^{-1}\left[\sum_{n=1}^{8} V_n \mathbf{P}_n^t \mathbf{S}_n \mathbf{P}_n\right]\hat{\mathcal{F}} = \phi_C \hat{1} \quad . \tag{92}$$

Further manipulating Eq. (92), we obtain

$$D^{-1}\left[\sum_{n=1}^{8} V_n \mathbf{P}_n^t \mathbf{S}_n \mathbf{P}_n\right]\hat{\mathcal{F}} = \Delta\hat{\Phi} \quad , \tag{93}$$

where $\Delta\hat{\Phi}$ is defined by Eq. (34). Comparing Eqs. (32) and (93) it follows that

$$\mathbf{W}^{-1} = D^{-1}\left[\sum_{n=1}^{8} V_n \mathbf{P}_n^t \mathbf{S}_n \mathbf{P}_n\right] \quad . \tag{94}$$

From Eq. (19) it follows that each 3×3 $\mathbf{S}$-matrix is the product of a matrix $\mathbf{A}$ and its transpose. Substituting from Eq. (19) into Eq. (94), we get,

$$\begin{aligned} \mathbf{W}^{-1} &= D^{-1}\left[\sum_{n=1}^{8} V_n \mathbf{P}_n^t \mathbf{A}_n^t \mathbf{A}_n \mathbf{P}_n\right] \quad , \\ &= D^{-1}\left[\sum_{n=1}^{8} V_n \left(\mathbf{A}_n \mathbf{P}_n\right)^t \left(\mathbf{A}_n \mathbf{P}_n\right)\right] \quad , \end{aligned} \tag{95}$$

Since

- the matrix, $\left(\mathbf{A}_n \mathbf{P}_n\right)^t \left(\mathbf{A}_n \mathbf{P}_n\right)$, 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 $\mathbf{A}$-matrices will be invertible with any well-formed mesh,
- the $\mathbf{P}$-matrices are not invertible,

it follows from Eq. (95) that $\mathbf{M}_n$ must be SPS but not SPD for each value of n, where

$$\mathbf{M}_n = D^{-1} V_n \left(\mathbf{A}_n \mathbf{P}_n\right)^t \left(\mathbf{A}_n \mathbf{P}_n\right) \quad . \tag{96}$$

Substituting from Eq. (96) into Eq. (95) we find that $\mathbf{W}^{-1}$ is a sum of matrices with each constituent matrix, $\mathbf{M}_n$, being SPS:

$$\mathbf{W}^{-1} = \sum_{n=1}^{8} \mathbf{M}_n \quad . \tag{97}$$

It is shown in [1] 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 **A**-matrices and the **P**-matrices (see Eqs. (17), (86), and (87)), it follows that each **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

$$\hat{\mathcal{F}} = \left(0, f^R, 0, f^T, 0, f^U\right)^t \quad , \tag{98}$$

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 **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. We neglect the time-derivative term in Eq. (1) and consider only the diffusion operator. Let us assume a solution vector, $\hat{\Phi}$, of the form given in Eq. (90). 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.,

$$\left(\phi^L, \phi^R, \phi^B, \phi^T, \phi^D, \phi^U, \phi^C\right)^t \to (\phi_1, \phi_2, \phi_3, \phi_4, \phi_5, \phi_6, \phi_7)^t \quad . \tag{99}$$

The first 6 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. (45) through (47), we equivalently require that the *negative* of each component be zero. The **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. (35). Thus the first 6 equations can be expressed in terms of the matrix **W** as follows:

$$-\mathbf{W}\Delta\hat{\Phi} = 0 \quad , \tag{100}$$

where in accordance with Eqs. (34) and (99):

$$\Delta\hat{\Phi} = (\phi_7 - \phi_1, \phi_7 - \phi_2, \phi_7 - \phi_3, \phi_7 - \phi_4, \phi_7 - \phi_5, \phi_7 - \phi_6)^t \quad . \tag{101}$$

Using Eqs. (100), and (101), one can easily construct the first six rows of the single-cell coefficient matrix, **C**, as follows:

$$c_{i,j} = \mathbf{W}_{i,j} \quad , i = 1, 6,\ j = 1, 6, \tag{102}$$

$$c_{i,7} = -\sum_{j=1}^{6} \mathbf{W}_{i,j} \quad , i = 1, 6. \tag{103}$$

The seventh and last row of **C** corresponds to the steady-state balance equation, i.e., Eq. (38) with the time-derivative set to zero:

$$f^L + f^R + f^B + f^T + f^D + f^U = Q^C V \quad . \tag{104}$$

Using Eqs. (35) and (101) through (104), we define the last row of the coefficient matrix:

$$c_{7,j} = -\sum_{i=1}^{6} \mathbf{W}_{i,j} \quad , i = 1, 6 \tag{105}$$

$$c_{7,7} = \sum_{i=1}^{6}\sum_{j=1}^{6} \mathbf{W}_{i,j} \quad . \tag{106}$$

To summarize, the coefficient matrix takes the following block form:

$$\mathbf{C} = \begin{bmatrix} \mathbf{W} & \mathbf{W}_r \\ \mathbf{W}_c & \mathbf{W}_{rc} \end{bmatrix} \quad , \tag{107}$$

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. Towards this end we note that any vector $\hat{\Phi}$ can clearly be re-expressed as follows:

$$\hat{\Phi} = (\phi_1, \phi_2, \phi_3, \phi_4, \phi_5, \phi_6, \phi_7)^t = \hat{\Phi}_f + \hat{\Phi}_c \quad , \tag{108}$$

where

$$\hat{\Phi}_f = (\phi_1 - \phi_7, \phi_2 - \phi_7, \phi_3 - \phi_7, \phi_4 - \phi_7, \phi_5 - \phi_7, \phi_6 - \phi_7, 0)^t \quad , \tag{109}$$

and

$$\hat{\Phi}_c = (\phi_7, \phi_7, \phi_7, \phi_7, \phi_7, \phi_7, \phi_7)^t \quad . \tag{110}$$

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

$$\left(\hat{\Phi}_f + \hat{\Phi}_c\right)^t \mathbf{C} \left(\hat{\Phi}_f + \hat{\Phi}_c\right) =$$

$$\hat{\Phi}_f^t \mathbf{C}\,\hat{\Phi}_f + \hat{\Phi}_f^t \mathbf{C}\,\hat{\Phi}_c + \hat{\Phi}_c^t \mathbf{C}\,\hat{\Phi}_f + \hat{\Phi}_c^t \mathbf{C}\,\hat{\Phi}_c. \tag{111}$$

It is easily verified that

$$\mathbf{C}\hat{\Phi}_c = \hat{0} \quad , \text{for all } \hat{\Phi}_c. \tag{112}$$

Substituting from Eq. (112) into Eq. (111), we get

$$\left(\hat{\Phi}_f + \hat{\Phi}_c\right)^t \mathbf{C} \left(\hat{\Phi}_f + \hat{\Phi}_c\right) = \hat{\Phi}_f^t \mathbf{C}\,\hat{\Phi}_f + \hat{\Phi}^t{}_c \mathbf{C}\,\hat{\Phi}_f. \tag{113}$$

Since

$$\hat{\Phi}_c^t \mathbf{C}\,\hat{\Phi}_f = \hat{\Phi}_f^t \mathbf{C}^t\,\hat{\Phi}_c = 0 \quad , \tag{114}$$

Eq. (113) reduces to

$$\left(\hat{\Phi}_f + \hat{\Phi}_c\right)^t \mathbf{C} \left(\hat{\Phi}_f + \hat{\Phi}_c\right) = \hat{\Phi}_f^t \mathbf{C}\,\hat{\Phi}_f \quad . \tag{115}$$

Using Eq. (107), it is easily shown that

$$\hat{\Phi}_f^t \mathbf{C}\,\hat{\Phi}_f = \hat{\Phi}_{f6}^t \mathbf{W}\,\hat{\Phi}_{f6} \quad , \tag{116}$$

where

$$\hat{\Phi}_{f6} = (\phi_1 - \phi_7, \phi_2 - \phi_7, \phi_3 - \phi_7, \phi_4 - \phi_7, \phi_5 - \phi_7, \phi_6 - \phi_7)^t \quad , \tag{117}$$

Since $\mathbf{W}$ is SPD, it follows from Eqs. (114) and (117) that

$$\begin{aligned} \left(\hat{\Phi}_f + \hat{\Phi}_c\right)^t \mathbf{C} \left(\hat{\Phi}_f + \hat{\Phi}_c\right) &= 0 \quad , \text{if } \hat{\Phi}_f = \hat{0} \, , \\ &> 0 \quad , \text{otherwise.} \end{aligned} \tag{118}$$

Thus $\mathbf{C}$ is positive-semidefinite. Since it is also symmetric, $\mathbf{C}$ is SPS. Note from Eq. (118) that the null space of $\mathbf{C}$ is spanned by all vectors $\hat{\phi}^c$. Following Eq. (110), 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 [1]. The final steps can be briefly described as follows:

1. Given a multi-cell 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. (83). 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 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] 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).

[2] O. C. Zienkiewicz, *The Finite Element Method*, McGraw-Hill, London, 3rd Edition (1977).

[3] G. C. Pomraning, *Equations of Radiation Hydrodynamics*, Volume 54 of the International Series of Monographs in Natural Philosophy, Edited by D. ter Haar, Pergamon Press, New York, (1973).

[4] A. I. Shestakov, J. A. Harte, and D. S. Kershaw,"Solution of the Diffusion Equation by Finite Elements in Lagrangian Hydrodynamics Codes," *J. Comp. 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] Gene H. Golub and Charles F. Van Loan, *Matrix Computations*, second edition, The Johns Hopkins University Press, Baltimore, (1989).

[8] M. J. Shashkov and S. Steinberg, "Solving Diffusion Equations with Rough Coefficients in Rough Grids," *J. Comput. Phys.*, **129**, 383 (1996).

[9] Robert D. Richtmyer and K. W. Morton, *Difference Methods for Initial-Value Problems*, Interscience Publishers, New York (1967).

[10] A. Weiser and M. F. Wheeler, "On Convergence of Block-Centered Finite Differences for Elliptic Problems," *SIAM J. Numer. Anal.*, **25**, 351 (1988).

[11] J. E. Dendy, Jr., "Black Box Multigrid," *J. Comput. Phys.* **48**, 366 (1982).

[12] J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis*, Springer-Verlag, New York, (1980).

[13] D. S. Kershaw, "Differencing of the Diffusion Equation in Lagrangian Hydrodynamics Codes," *J. Comput. Phys.*, **39**, 375 (1981).

[14] J. E. Dendy, Jr., "Two Multigrid Methods for Three-Dimensional Problems with Discontinuous and Anisotropic Coefficients," *SIAM Sci. Stat. Comp.*, **8**, 673 (1987).