JOURNAL OF COMPUTATIONAL PHYSICS **132,** 130–148 (1997)
ARTICLE NO. CP965633

The Numerical Solution of Diffusion Problems in Strongly Heterogeneous Non-isotropic Materials

James Hyman,* Mikhail Shashkov,* and Stanly Steinberg‡

*Los Alamos National Laboratory, T-7, MS-B284, Los Alamos, New Mexico 87545; and ‡Department of Mathematics and Statistics, University of New Mexico, Albuquerque, New Mexico 87131
E-mail: shashkov@goshawk.lanl.gov

Received February 20, 1996; revised September 9, 1996

A new second-order finite-difference algorithm for the numerical solution of diffusion problems in strongly heterogeneous and non-isotropic media is constructed. On problems with rough coefficients or highly nonuniform grids, the new algorithm is superior to all other algorithms we have compared it with. For problems with smooth coefficients on smooth grids, the method is comparable with other second-order methods. The new algorithm is formulated for logically rectangular grids and is derived using the *support-operators method.* A key idea in deriving the method was to replace the usual inner product of vector functions by an inner product weighted by the inverse of the material properties tensor and to use the flux operator, defined as the material properties tensor times the gradient, rather than the gradient, as one of the basic first-order operators in the support-operators method. The discrete analog of the flux operator must also be the negative adjoint of the discrete divergence, in an inner product that is a discrete analog of the continuum inner product. The resulting method is conservative and the discrete analog of the variable coefficient Laplacian is symmetric and negative definite on nonuniform grids. In addition, on any grid, the discrete divergence is zero on constant vectors, the null space for the gradient is the constant functions, and, when the material properties are piecewise constant, the discrete flux operator is exact for piecewise linear functions. We compare the methods on some of the most difficult examples to be found in the literature. © 1997 Academic Press

1. INTRODUCTION

The main goal of this paper is the description and investigation of new finite-difference algorithms for solving the elliptic partial differential equation (PDE) or stationary diffusion equation

$$-\mathbf{div}\,\mathbf{K}\,\mathbf{grad}\,u = f, \qquad (x, y) \in V. \tag{1.1}$$

The solution, $u = u(x, y)$, is the concentration to be solved for (temperature in heat diffusion problems, and pressure in flow problems). Here V is a two-dimensional region, **div** is the divergence, **grad** is the gradient, $\mathbf{K} = \mathbf{K}(x, y)$ is a symmetric positive-definite matrix, and $f = f(x, y)$ is a given right-hand side or forcing function. The boundary conditions are general Robin (or mixed),

$$(\mathbf{K}\,\mathbf{grad}\,u, \vec{n}) + \alpha\, u = \psi, \qquad (x, y) \in \partial V, \tag{1.2}$$

where $\vec{n}$ is the vector of unit outward normal to the boundary ∂V, and α and ψ are functions given on ∂V. The algorithm is constructed using a nontrivial generalization of the *support-operators method* for solving problems where the material properties tensor (or matrix) **K** may be *discontinuous* and *non-diagonal* and, moreover, the computational grid may not be *smooth.*

The support-operators method constructs discrete analogs of invariant differential operators **div** and **grad**, which satisfy discrete analogs of the integral identities responsible for the conservative properties of the continuum model. The method was initially developed in [1] by Samarskii, Tishkin, Favorskii, and Shashkov and is fully described in [2].

This paper is the third of a series on the support-operators method. In the first paper [3], the support-operators method was combined with the mapping method to produce an algorithm for equations with general boundary conditions. The resulting method was shown to be accurate when both **K** is *smooth* and the problem is solved on a *smooth* grid. In the second paper [4], the support-operators method was extended to define a new cell-centered finite-difference algorithm for solving time-dependent diffusion equations with discontinuous diagonal **K** on logically rectangular non-smooth grids, such as the grids associated with Lagrangian hydrodynamics calculations. This paper also contains an extensive review of the literature on constructing approximations of differential operators on nonuniform grids and motivation for using the support-operators method.

In this paper we extend the support-operators method to the non-diagonal non-smooth tensor **K** and non-smooth logically rectangular grids. A key to improving the accuracy for non-smooth **K** is to use the flux operator **K grad**, rather than the gradient operator **grad**, as one of the basic first-order operators. This requires that the usual inner product of vector functions be replaced by an inner product weighted by the inverse of the material properties tensor.

0021-9991/97 $25.00

The methods are linear, conservative and material discontinuities are assumed to occur at the surfaces of the grid cells. The methods use both the heat-flux and the temperature as primary variables.

In these methods, the temperature and the elements of the conductivity matrix **K** are defined at the cell centers. We will compare two different approaches in discretizing the heat-flux. In the *nodal discretization*, vector quantities, such as the heat-flux, are described by their Cartesian components defined at the nodes of the grid. When **K** is discontinuous, this approach is not as accurate as the *surface discretization* method where vector quantities are described by their orthogonal projections into the directions perpendicular to the faces of the cells. The surface discretization approach is more accurate when **K** is discontinuous because, even when **K** is discontinuous at the interface between the cells, the heat-flux perpendicular to the cell face is always continuous.

We now summarize the properties of the finite-difference method using the surface discretization for heat-flux. The discrete analog of the variable-coefficient Laplacian **div K grad** can be decomposed as a composition of two discrete operators: a divergence **DIV**, and a flux operator $\mathbf{G} \sim -\mathbf{K}\ \mathbf{grad}$, which are the adjoints of each other. This ensures the self-adjointness and negative definiteness of the discrete variable-coefficient Laplacian for general grids. Moreover, on any grid, the discrete analog of **div** is exactly equal to zero on constant vectors, the **GRAD** is equal to zero only for constant functions, and (when the material properties are piecewise constant) the discrete flux operator is exact for piecewise linear functions. On rectangular grids, all the discrete operators reduce to standard finite difference approximations and when the discontinuous heat conductivity is a scalar, the algorithm produces the appropriate harmonic average heat conductivity for the fluxes.

Because the fluxes are the primary unknowns for the method, when this approach is used to solve (1.1), the new method has twice as many unknowns as the more traditional algorithms. However, because the method is second-order accurate (as it is shown by numerical experiments) on general logically rectangular grids, fewer grid points are needed than the commonly used first order methods to give the same level of accuracy in the solution. In our computational comparisons with many of the existing methods, when the computational mesh is significantly skewed, we find that the new method is much more accurate than the traditional finite difference and finite volume methods. Also, because the discrete difference equations are symmetric and positive definite, only half the coefficients for the difference equations need be stored and the equations can be solved with some of the most powerful iterative methods for solving linear systems.

This paper is arranged as follows. In Section 2, the continuum problem is written in operator form to illuminate the properties of the operators that should have analogs in the discrete case. The main point in this section is to introduce an inner product on vectors that is weighted by $\mathbf{K}^{-1}$ and to introduce the operator **K grad** rather than just **grad**. In Section 3, the grid and the discretizations of scalar and vector functions are given. Both nodal and surface discretizations for fluxes are introduced, and the discrete inner product for general non-diagonal **K** is constructed.

In Section 4, following the support-operators method, approximations for **div** and **K grad** are derived using both the nodal and surface discretizations. Using these operators, the finite difference scheme for **div K grad** is constructed. The theoretical properties of the discrete operators are summarized and it is shown that, for the surface discretization, the null space of **grad** is the constant functions, while for the nodal discretization, the null space also contains the spurious highest-frequency mode on a square grid.

In Subsection 4.6, we describe strategies for solving the linear equations given by the discretizations of the variable-coefficient Laplacian. An important point here is that the discrete operator for the surface discretization is not local. However, it is the product of a local discrete operator with the inverse of a local operator, so the residual can still be computed as a local operation. Iterative methods that only require local operations, such as the preconditioned conjugate gradient method, can be used efficiently.

In Section 5, we test and compare our algorithms on five of the most difficult examples to be found in the literature. These examples verify that the surface discretization approach performs reliably on all of the examples, and the nodal discretization gives reasonable results. Additional numerical examples can be found in [4].

2. THE PROPERTIES OF THE CONTINUUM PROBLEM

In this section, we will develop the flux form of the elliptic PDE (1.1) as a system of first-order equations and analyze the system in terms of abstract operators on inner-product spaces of scalar and vector-valued functions. The analysis will be for Robin (mixed) boundary conditions (1.2). The case with Dirichlet boundary conditions is a straightforward extension of this analysis and will not be analyzed here.

We introduce the space of scalar functions H with the inner product

$$(u, v)_{\mathrm{H}} = \int_V u\, v\, dV + \oint_{\partial V} u\, v\, dS, \qquad u, v \in H, \tag{2.1}$$

and rewrite Eqs. (1.1), (1.2) as

$$\mathbf{A}\, u = \mathbf{F}. \tag{2.2}$$

The operator **A** is given by

$$\mathbf{A}: H \rightarrow H,$$

$$\mathbf{A}u = \begin{cases} -\mathbf{div}\,\mathbf{K}\,\mathbf{grad}\,u, & (x,y) \in V \\ (\mathbf{K}\,\mathbf{grad}\,u, \vec{n}) + \alpha\,u, & (x,y) \in \partial V \end{cases} \tag{2.3}$$

and has the properties

$$(\mathbf{A}u, v)_H = (u, \mathbf{A}v)_H, \qquad (\mathbf{A}u, u)_H > 0. \tag{2.4}$$

The right-hand side of (2.2) has the form

$$\mathbf{F} = \begin{cases} f, & (x,y) \in V \\ \psi, & (x,y) \in \partial V. \end{cases} \tag{2.5}$$

We investigate the properties of problem (1.1), (1.2) by writing it in terms of first-order operators in flux or mixed form:

$$\begin{aligned} \mathbf{div}\,\vec{w} = f, \qquad (x,y) \in V, \\ \vec{w} = -\mathbf{K}\,\mathbf{grad}\,u, \qquad (x,y) \in V, \\ -(\vec{w}, \vec{n}) + \alpha\,u = \psi, \qquad (x,y) \in \partial V. \end{aligned} \tag{2.6}$$

The flux $\vec{w} = \mathbf{G}\,u = -\mathbf{K}\,\mathbf{grad}\,u$ has physical meaning and is continuous across discontinuities in **K**, but **grad** u has neither. When the matrix **K** is non-diagonal and discontinuous, there are advantages (and it is natural) to analyzing the equations by considering the operator $\mathbf{G} = -\mathbf{K}\,\mathbf{grad}$ directly, rather than the operator **grad** and matrix **K** separately.

To investigate (2.6), we introduce the space of vector functions **H** with the inner product of two vector functions $\vec{A}, \vec{B} \in \mathbf{H}$, defined by

$$(\vec{A}, \vec{B})_{\mathbf{H}} = \int_V (\mathbf{K}^{-1}\vec{A}, \vec{B})\,dV. \tag{2.7}$$

Because the matrix **K** is symmetric and positive definite, so is $\mathbf{K}^{-1}$ and (2.7) satisfies all the axioms of an inner product. This "weighted" inner product is well defined for discontinuous **K** and naturally arises in mixed finite-element formulations (see, for example, [5, 6]).

From (2.6), it is clear that the operator **A** can be represented in the form

$$\mathbf{A} = \mathbf{\Omega} + \mathbf{D} \cdot \mathbf{G}, \tag{2.8}$$

where the operators **G**, **D**, and **Ω** have the definitions

$$\mathbf{G}u = -\mathbf{K}\,\mathbf{grad}\,u, \qquad (x,y) \in V, \tag{2.9}$$

$$\mathbf{D}\vec{w} = \begin{cases} +\mathbf{div}\,\vec{w}, & (x,y) \in V, \\ -(\vec{w}, \vec{n}), & (x,y) \in \partial V, \end{cases} \tag{2.10}$$

$$\mathbf{\Omega}u = \begin{cases} 0, & (x,y) \in V, \\ \alpha\,u, & (x,y) \in \partial V. \end{cases} \tag{2.11}$$

Here

$$\mathbf{G}: H \rightarrow \mathbf{H}; \qquad \mathbf{D}: \mathbf{H} \rightarrow H; \qquad \mathbf{\Omega}: H \rightarrow H. \tag{2.12}$$

Using the first-order operators, system (2.6) can be rewritten in the form

$$\mathbf{\Omega}\,u + \mathbf{D}\,\vec{w} = F, \qquad \vec{w} = \mathbf{G}\,u. \tag{2.13}$$

A crucial relation which we must retain in our discrete approximation is

$$\mathbf{D} = \mathbf{G}^*. \tag{2.14}$$

This is clear from the definition of operator **D**, the definition (2.1) for the inner product in the space H, and integral identity

$$\int_V \phi\,\mathbf{div}\,\vec{w}\,dV + \int_V (\vec{w}, \mathbf{grad}\,\phi)\,dV = \oint_S \phi(\vec{w}, \vec{n})\,dS, \tag{2.15}$$

which give

$$\begin{aligned} (\mathbf{D}\,\vec{w}, u)_H &= \int_V u\,\mathbf{div}\,\vec{w}\,dV - \oint_{\partial V} u(\vec{w}, \vec{n})\,dS \\ &= -\int_V (\vec{w}, \mathbf{grad}\,u)\,dV \\ &= -\int_V (\vec{w}, \mathbf{K}^{-1}(\mathbf{K}\,\mathbf{grad}\,u))\,dV \\ &= (\vec{w}, \mathbf{G}\,u)_{\mathbf{H}}. \end{aligned} \tag{2.16}$$

Also, it is evident that $\mathbf{\Omega} = \mathbf{\Omega}^* \geq 0$.

Because $\mathbf{A} = \mathbf{\Omega} + \mathbf{D} \cdot \mathbf{G}$, the properties (2.4) follow from the properties of operators **Ω**, **D**, and **G**. Note that boundary conditions are included in definitions of operators and spaces of functions in a natural way. The properties of first-order operators discussed in this section are preserved by the finite-difference methods derived using the support-operators method.

3. THE SPACES OF DISCRETE FUNCTIONS

In this section, we define our notation for a logically rectangular grid [7], a cell-centered discretization of scalar

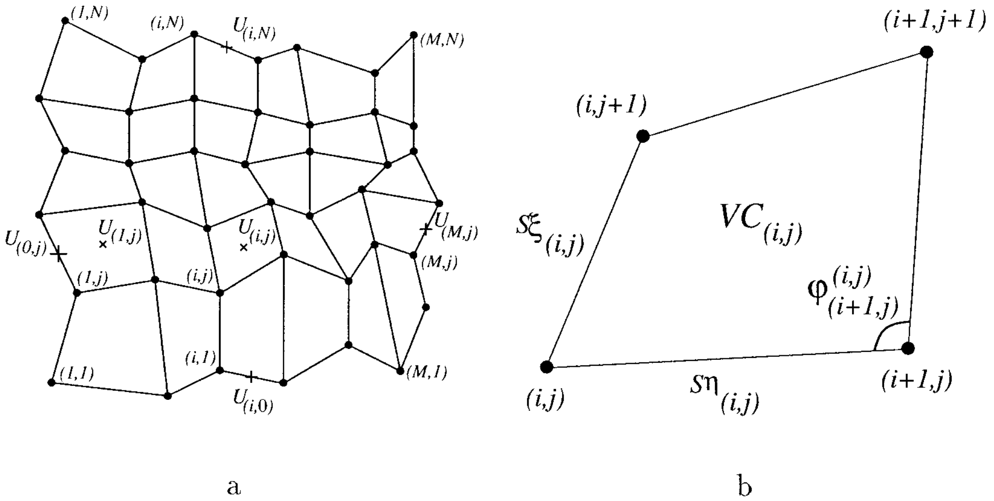


FIG. 1. (a) A logically rectangular grid and the discretization of a scalar, (b) a typical cell of a logically rectangular grid.

functions, and both nodal and face-centered discretizations of vector-valued functions.

3.1. The Discretization of Scalar and Vector Functions

The nodes of a logically rectangular grid can be indexed the same way as a rectangular grid with indices (i, j), $1 \leq i \leq M$, $1 \leq j \leq N$ (see Fig. 1a). The quadrangle defined by the nodes (i, j), $(i + 1, j)$, $(i + 1, j + 1)$, and $(i, j + 1)$ is called the (i, j) cell (see Fig. 1b). The area of this cell is denoted by $VC_{(i,j)}$. The length of the side of the (i, j) cell that connects the vertices (i, j) and $(i, j + 1)$ is denoted $S\xi_{(i,j)}$, while the length of the side that connects the vertices (i, j) and $(i + 1, j)$ is denoted $S\eta_{(i,j)}$. The angle between any two adjacent sides of cell (i, j) that meet at node (k, l) is denoted $\varphi_{k,l}^{(i,j)}$ (the angle $\varphi_{(i+1,j)}^{(i,j)}$ is displayed in Fig. 1b). We assume, unless otherwise stated, that the cells are convex. (Meshes with non-convex cells are considered in Subsection 3.2.2 and [4].)

To study convergence rates, we impose some standard mild smoothness assumptions on the family of grids. A small parameter which characterizes the density of the grid is

$$h = \max\left\{\frac{1}{M-1}, \frac{1}{N-1}\right\}. \tag{3.1}$$

We assume that there exist constants $C_{\max}^{(1)}$ and $C_{\min}^{(1)}$, which do not depend on h, such that

$$C_{\min}^{1} h^2 \leq VC_{(i,j)} \leq C_{\max}^{1} h^2, \tag{3.2}$$

and constants $C_{\max}^{(2)}$ and $C_{\min}^{(2)}$, which do not depend on h, such that

$$C_{\min}^{2} h \leq S\xi_{(i,j)}, S\eta_{(i,j)} \leq C_{\max}^{2} h, \tag{3.3}$$

and that there exists a constant $\varepsilon > 0$, which does not depend on h, such that

$$\sin(\varphi_{(k,l)}^{(i,j)}) \geq \varepsilon. \tag{3.4}$$

Our notation is motivated by considering the 2-D grid as a projection of a 3-D grid. This approach may seem awkward at first, but it becomes natural when put into a three-dimensional setting and it clarifies how the finite-difference methods generalize to 3-D.

In this paper we consider functions of the coordinates x and y and extend the grid into a third dimension, z, when convenient. The extended 3-D mesh is constructed by extending a grid line of unit length into the z direction to form a prism with unit height and with a 2-D quadrilateral cell as its base (see Fig. 2a).

It is also useful to interpret the grid as the discretization of a map from a curvilinear coordinate system $x = x(\xi, \eta, \zeta)$, $y = y(\xi, \eta, \zeta)$, $z = z(\xi, \eta, \zeta)$, where the nodes of the grid are given by $x_{(i,j,k)} = x(\xi_i, \eta_j, \zeta_k)$, with $\xi_i = i\,\Delta\xi$, and so forth, as shown in Fig. 2b. Thus increasing i corresponds to increasing ξ, and so on. Using this 3-D interpretation, the notation $S\xi_{(i,j)}$ refers to the area of the 3-D surface given by the points (i, j, k), $(i, j + 1, k)$, $(i, j, k + 1)$, $(i, j + 1, k + 1)$; that is, $S\xi$, gives the element of surface area for a surface where ξ is constant, because we took the height of the prism equal to one. Similar results hold for other sides of a 2-D cell.

3.1.1. *The Discrete Scalar Functions*

The discrete analog of the scalar function u is the cell-centered discrete scalar function $U_{(i,j)}$ (see Fig. 1a) whose

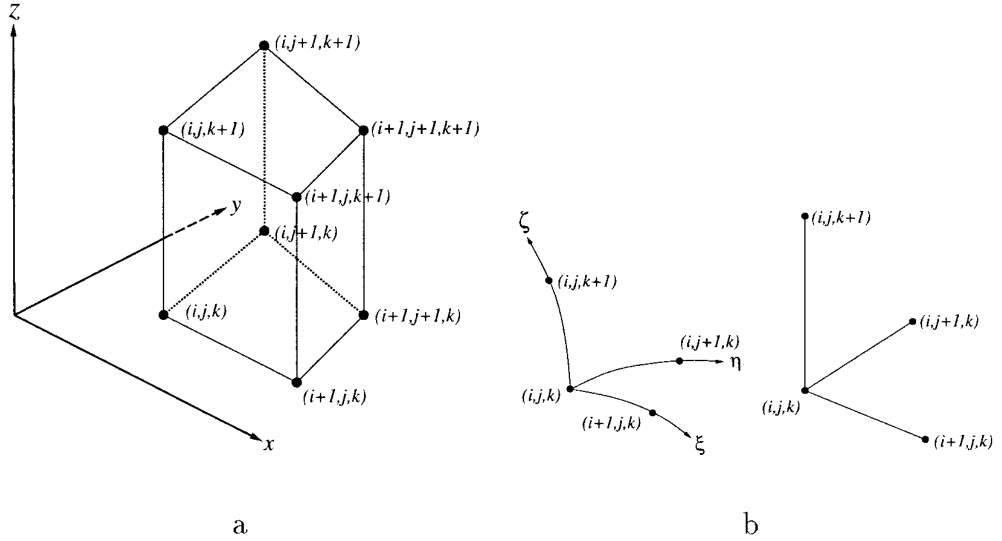


FIG. 2. (a) The 3-D mesh, (b) curvilinear coordinates and grid lines.

indices vary in the same range as the volume $VC_{(i,j)}$. The treatment of the boundary conditions requires the introduction of the values of the scalar function on the centers of the boundary segments (see Fig. 1a):

$$\begin{aligned} &U_{(0,j)}, U_{(M,j)}, j = 1, ..., N-1; \\ &U_{(i,0)}, U_{(i,N)}, i = 1, ..., M-1. \end{aligned} \tag{3.5}$$

In the 3-D interpretation, scalar functions are defined in the centers of the 3-D prisms and in the centers of the boundary surfaces. Again, because we only consider the 2-D case, these values can be projected to the 2-D cells, and the centers of the boundary sides.

The components of **K** are discretized in the same way as u. The scalar functions α and ψ from the boundary conditions are discretized in the same way as u is on the boundary.

3.1.2. *The Discrete Vector Functions*

Two possibilities are used for discretizing vector functions $\vec{W} = (WX, WY)$: the first uses the usual Cartesian components $WX_{(i,j)}$ and $WY_{(i,j)}$ of the vector at the nodes as shown in Fig. 3a; and the second uses the orthogonal projections of the vector on the direction which is perpendicular to the surfaces of 3-D cells at the centers of the surfaces. Because the 3-D cell is a right prism, we can interpret these components as the orthogonal projections on the directions which are perpendicular to the sides of the 2-D cell, as in Fig. 3b. The notation

$$WS\xi_{(i,j)} : i = 1, ..., M; j = 1, ..., N-1 \tag{3.6}$$

is used for the component at the center of side $S\xi_{(i,j)}$, and the notation

$$WS\eta_{(i,j)} : i = 1, ..., M-1; j = 1, ..., N \tag{3.7}$$

is used for the component at the center of side $S\eta_{(i,j)}$.

3.2. The Spaces of Discrete Functions

The spaces of discrete scalar and vector functions need inner products. For scalar functions this is straightforward, but for vector functions, there are two spaces: $\mathscr{HN}$ for the nodal discretization, and $\mathscr{HS}$ for the surface discretization. Neither inner product is simple because of the use of **K** in the inner product and, in both cases, the values **K** are not given at the same points as the vector components.

3.2.1. *The Space of Discrete Scalar Functions*

The space of discrete scalar functions is labeled HC and has the inner product

$$\begin{aligned} (U, V)_{HC} = &\sum_{i=1}^{M-1} \sum_{j=1}^{N-1} U_{(i,j)} V_{(i,j)} VC_{(i,j)} \\ &+ \sum_{i=1}^{M-1} U_{(i,0)} V_{(i,0)} S\eta_{(i,1)} + \sum_{j=1}^{N-1} U_{(N,j)} V_{(N,j)} S\xi_{(N,j)} \\ &+ \sum_{i=1}^{M-1} U_{(i,N)} V_{(i,N)} S\eta_{(i,N)} \\ &+ \sum_{j=1}^{N-1} U_{(0,j)} V_{(0,j)} S\xi_{(1,j)}. \end{aligned} \tag{3.8}$$

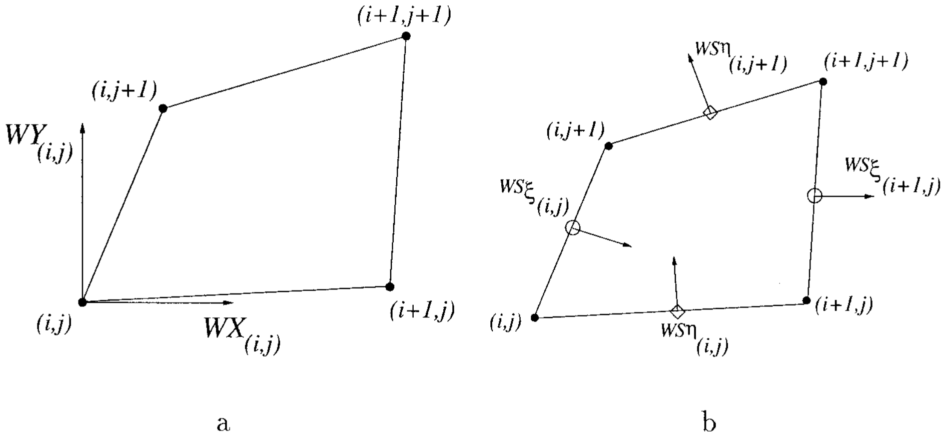


FIG. 3. (a) The nodal discretization of a vector, (b) the surface discretization of a vector.

3.2.2. *The Space of Discrete Nodal Vector Functions*

The space of discrete nodal vector functions is called $\mathcal{HN}$ and the inner product on this space is given by $(\cdot, \cdot)_{\mathcal{HN}}$.

$$(\vec{A}, \vec{B})_{\mathcal{HN}} = \sum_{i=1}^{M-1} \sum_{j=1}^{N-1} (\mathbf{K}^{-1}\vec{A}, \vec{B})_{(i,j)}\, VC_{(i,j)}, \tag{3.9}$$

where $(\mathbf{K}^{-1}\vec{A}, \vec{B})_{ij}$ is the approximation of the dot product of vectors $\mathbf{K}^{-1}\vec{A}$ and $\vec{B}$ in the cell. Note that $\mathbf{K}$ and consequently $\mathbf{K}^{-1}$ are defined at cell centers, but the components of the vectors are defined at the cell nodes, which complicates the definition of a dot product. For this type of discretization it is natural to assume that tensor $\mathbf{K}$ is given by its Cartesian components, K_{xx}, $K_{xy} = K_{yx}$, K_{yy}, and consequently

$$\begin{aligned}(\mathbf{K}^{-1}\vec{A}, \vec{B}) = (\mathbf{K}^{-1})_{xx}\, AX\, BX + (\mathbf{K}^{-1})_{xy}\, (AX\, BY \\ + AY\, BX) + (\mathbf{K}^{-1})_{yy}\, AY\, BY.\end{aligned} \tag{3.10}$$

For simplicity (where it is convenient), we will use the notation $\mathbf{KI} = \mathbf{K}^{-1}$ for the matrix inverse of $\mathbf{K}$.

We define $(\mathbf{K}^{-1}\vec{A}, \vec{B})_{(i,j)}$ as

$$\begin{aligned}(\mathbf{K}^{-1}\vec{A}, \vec{B})_{(i,j)} &= (\mathbf{KI}\,\vec{A}, \vec{B})_{(i,j)} \\ &= \sum_{k,l=0}^{1} V^{(i,j)}_{(i+k,j+l)} \{(\mathbf{KI}_{xx})_{(i,j)}\, AX_{(i+k,j+1)} \\ &\quad BX_{(i+k,j+l)} + (\mathbf{KI}_{xy})_{(i,j)}\, [AX_{(i+k,j+l)} \\ &\quad BY_{(i+k,j+l)} + AY_{(i+k,j+l)}\, BX_{(i+k,j+l)}] \\ &\quad + (\mathbf{KI}_{yy})_{(i,j)}\, AY_{(i+k,j+l)}\, BY_{(i+k,j+l)}\}\end{aligned} \tag{3.11}$$

and the $V^{(i,j)}_{(i+k,j+l)}$ are weights satisfying

$$\sum_{k,l=0}^{1} V^{(i,j)}_{(i+k,j+l)} = 1. \tag{3.12}$$

In this formula, each index (k, l) corresponds to one of the vertices of the (i, j) cell, and the notation for the weights is the same as for the angles of the cell. As was shown in [4], to obtain a first-order approximation for the gradient operator, it is necessary that the weights $V^{(i,j)}_{(i+k,j+l)}$ be one-half of the area of the triangle in the (i, j) cell, which contains the angle at the node $(i + k, j + l)$, divided by the volume of the cell $VC_{(i,j)}$.

Because the weights are positive if the cell is convex, we are guaranteed that the discrete analog of the variable-coefficient Laplacian is positive-definite (see [4] for details). When the cell is not convex, we modify the definition,

$$\tilde{V}^{(i,j)}_{(i+k,j+l)} = \frac{|V^{(i,j)}_{(i+k,j+l)}|}{\sum_{p,q=0}^{1} |V^{(i,j)}_{(i+p,j+q)}|}, \tag{3.13}$$

to give positive weights (see [4]). For convex cells, the two definitions coincide.

3.2.3. *The Space of Discrete Surface Vector Functions*

The space of discrete surface vector functions is called $\mathcal{HS}$ and the inner product on this space is $(\cdot, \cdot)_{\mathcal{HS}}$. Again, there is the problem that the components of the vectors are not defined at the cell centers where $\mathbf{K}$ is defined.

To obtain the formula for $(K^{-1}\vec{A}, \vec{B})_{i,j}$, let us first consider a non-orthogonal basis system with axes ξ and η and let φ be the angle between these axes. If the unit normals to the axes are $n\vec{S}\xi$ and $n\vec{S}\eta$, then the components of the vector $\vec{W}$ in this basis are the orthogonal projections $WS\xi$ and $WS\eta$ of $\vec{W}$ onto the normal vectors. (See the discussion in Chapter 2 of Knupp and Steinberg [7] for more details.) A simple vector algebra calculation shows that if $\vec{A} =$

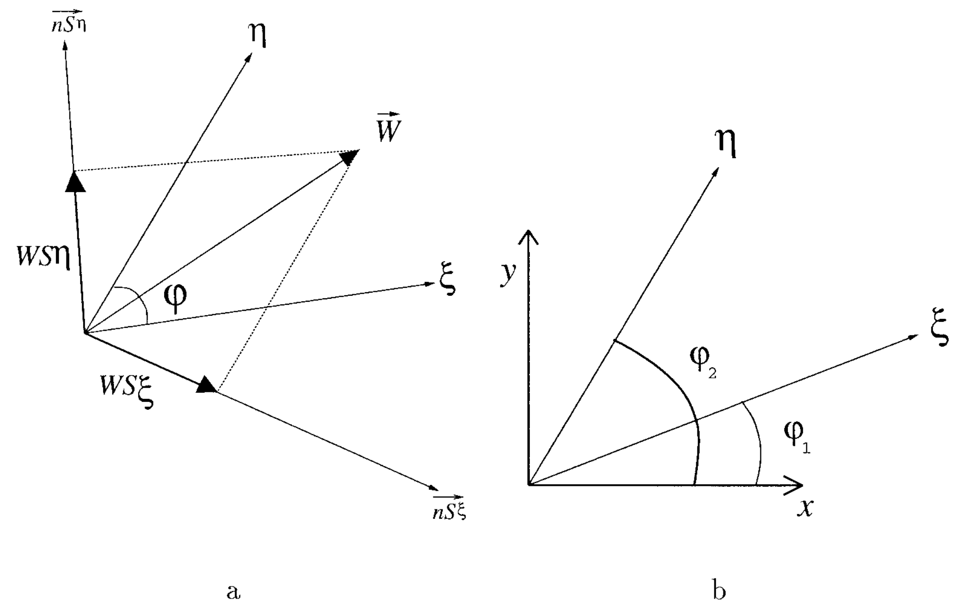


FIG. 4. (a) The components of a vector in a local basis, (b) the angles φ_1 and φ_2.

$(AS\xi, AS\eta)$ and $\vec{B} = (BS\xi, BS\eta)$, then the expression for the dot product is

$$(\vec{A}, \vec{B}) = [AS\xi\, BS\xi + AS\eta\, BS\eta + (AS\xi\, BS\eta + AS\eta\, BS\xi)\cos(\varphi)]/\sin^2(\varphi). \tag{3.14}$$

Denote by φ_1 and φ_2 the angles between the x-axis of the Cartesian coordinate system and the first and second axes of the local coordinate system, respectively (see Fig. 4b). In terms of the coordinates of vectors in the local basis system, the dot product $(\mathbf{KI}\,\vec{A}, \vec{B})$ is

$$(\mathbf{KI}\,\vec{A}, \vec{B}) = [T11\, AS\xi\, BS\xi + T22\, AS\eta\, BS\eta + \mathrm{T}12\,(AS\xi\, BS\eta + AS\eta\, BS\xi)]/\sin^2(\varphi), \tag{3.15}$$

where

$$T11 = \mathbf{KI}_{xx}\cos^2\varphi_1 + 2\,\mathbf{KI}_{xy}\cos\varphi_1\sin\varphi_1 + \mathbf{KI}_{yy}\sin^2\varphi_1,$$

$$T12 = \mathbf{KI}_{xx}\cos\varphi_1\cos\varphi_2 + 2\mathbf{KI}_{xy}(\cos\varphi_1\sin\varphi_2 + \sin\varphi_1\cos\varphi_2) + \mathbf{KI}_{yy}\sin\varphi_1\sin\varphi_2,$$

$$T22 = \mathbf{KI}_{xx}\cos^2\varphi_2 + 2\mathbf{KI}_{xy}\cos\varphi_2\sin\varphi_2 + \mathbf{KI}_{yy}\sin^2\varphi_2.$$

This formula is used to obtain the discrete inner product in the cell,

$$(\mathbf{KI}\,\vec{A}, \vec{B})_{(i,j)} = \sum_{k,l=0}^{1} \frac{V^{(i,j)}_{(i+k,j+l)}}{\sin^2(\varphi^{(i,j)}_{(i+k,j+l)})} [T11^{(i,j)}_{(i+k,j+l)}\, AS\xi_{(i+k,j)}\, BS\xi_{(i+k,j)} + T22^{(i,j)}_{(i+k,j+l)}\, AS\eta_{(i,j+l)}\, BS\eta_{(i,j+l)} + (-1)^{k+l}\, T12^{(i,j)}_{(i+k,j+l)}\,(AS\xi_{(i+k,j)}\, BS\eta_{(i,j+l)} + AS\eta_{(i,j+l)}\, BS\xi_{(i+k,j)})], \tag{3.16}$$

where, for example,

$$T11^{(i,j)}_{(i+k,j+l)} = (\mathbf{KI}_{xx})_{(i,j)}\cos^2((\varphi_1)^{(i,j)}_{(i+k,j+l)}) + 2\,(\mathbf{KI}_{xy})_{(i,j)}\cos((\varphi_1)^{(i,j)}_{(i+k,j+l)})\sin((\varphi_1)^{(i,j)}_{(i+k,j+l)}) + (\mathbf{KI}_{yy})_{(i,j)}\sin^2((\varphi_1)^{(i,j)}_{(i+k,j+l)}).$$

That is, the values of matrix elements are defined in the cell (i, j), and the angle φ_1 is related to the corresponding vertex of the cell. The formulas for $T12$ and $T22$ are similar. In general, the notation is the same as for any quantity related to the cell and vertex. Finally, as in (3.9), the inner product in $\mathscr{HS}$ is given by

$$(\vec{A}, \vec{B})_{\mathscr{HS}} = \sum_{i=1}^{M-1}\sum_{j=1}^{N-1} (\mathbf{KI}\,\vec{A}, \vec{B})_{(i,j)}\, VC_{(i,j)}. \tag{3.17}$$

3.2.4. *The Formal and Natural Inner Products*

To compute the adjoint relationships, it is helpful to introduce *formal* inner products, $[\cdot, \cdot]$, in the spaces of scalar and vector functions. In HC,

$$\begin{aligned}[U, V]_{HC} = {} & \sum_{i=1}^{M-1}\sum_{j=1}^{N-1} U_{(i,j)}\, V_{(i,j)} + \sum_{i=1}^{M-1} U_{(i,0)}\, V_{(i,0)} \\ & + \sum_{j=1}^{N-1} U_{(M,j)}\, V_{(M,j)} + \sum_{i=1}^{M-1} U_{(i,N)}\, V_{(i,N)} \\ & + \sum_{j=1}^{N-1} U_{(0,j)}\, V_{(0,j)},\end{aligned} \tag{3.18}$$

in $\mathscr{H}\mathscr{N}$,

$$\begin{aligned}[\vec{A}, \vec{B}]_{\mathscr{H}\mathscr{N}} = {} & \sum_{i=1}^{M}\sum_{j=1}^{N} AX_{(i,j)}\, BX_{(i,j)} \\ & + \sum_{i=1}^{M}\sum_{j=1}^{N} AY_{(i,j)}\, BY_{(i,j)},\end{aligned} \tag{3.19}$$

and in $\mathscr{H}\mathscr{S}$

$$\begin{aligned}[\vec{A}, \vec{B}]_{\mathscr{H}\mathscr{S}} = {} & \sum_{i=1}^{M}\sum_{j=1}^{N-1} AS\xi_{(i,j)}\, BS\xi_{(i,j)} \\ & + \sum_{i=1}^{M-1}\sum_{j=1}^{N} AS\eta_{(i,j)}\, BS\eta_{(i,j)}.\end{aligned} \tag{3.20}$$

Then the relationships between the natural inner products and the formal inner products are

$$\begin{aligned}(U, V)_{HC} &= [\mathscr{M} U, V]_{HC}, \\ (\vec{A}, \vec{B})_{\mathscr{H}\mathscr{N}} &= [\mathscr{N}\vec{A}, \vec{B}]_{\mathscr{H}\mathscr{N}}, \\ (\vec{A}, \vec{B})_{\mathscr{H}\mathscr{S}} &= [\mathscr{S}\vec{A}, \vec{B}]_{\mathscr{H}\mathscr{N}},\end{aligned} \tag{3.21}$$

where $\mathscr{M}$, $\mathscr{N}$, and $\mathscr{S}$ are matrices.

Formulas for these matrices can be found by direct comparison of the formal and natural inner products. The formula for $\mathscr{M}$ is

$$\begin{aligned}(\mathscr{M} U)_{(i,j)} &= VC_{(i,j)}\, U_{(i,j)}, \quad i = 1, \ldots, M-1; j = 1, \ldots, N-1. \\ (\mathscr{M} U)_{(i,j)} &= S\xi_{(i,j)}\, U_{(i,j)}, \quad i = 0 \text{ and } i = M; j = 1, \ldots, N-1. \\ (\mathscr{M} U)_{(i,j)} &= S\eta_{(i,j)}\, U_{(i,j)}, \quad i = 1, \ldots, M-1; j = 0 \text{ and } j = N.\end{aligned} \tag{3.22}$$

From these formulas, we see that the matrix $\mathscr{M}$ is a symmetric positive-definite operator in the formal inner product:

$$[\mathscr{M} U, V]_{HC} = [U, \mathscr{M} V]_{HC}, \qquad [\mathscr{M} U, U]_{HC} > 0. \tag{3.23}$$

The operator $\mathscr{N}$ can be written in block form:

$$\begin{aligned}\mathscr{N}\vec{A} &= \begin{pmatrix} N_{11} & N_{12} \\ N_{21} & N_{22} \end{pmatrix} \begin{pmatrix} AX \\ AY \end{pmatrix} \\ &= \begin{pmatrix} N_{11}\, AX + N_{12}\, AY \\ N_{21}\, AX + N_{22}\, AY \end{pmatrix}.\end{aligned} \tag{3.24}$$

A comparison of the formal inner product $(\vec{A}, \vec{B})_{\mathscr{H}\mathscr{N}}$ given in (3.9) and the natural inner product gives

$$\begin{aligned}[\mathscr{N}\vec{A}, \vec{B}]_{\mathscr{H}\mathscr{N}} = {} & \sum_{i=1}^{M}\sum_{j=1}^{N} \{[(N_{11}\, AX)_{(i,j)} \\ & + (N_{12}\, AY)_{(i,j)}]\, BX_{(i,j)} \\ & + [(N_{21}\, AX)_{(i,j)} + (N_{22}\, AY)_{(i,j)}]\, BY_{(i,j)}\}.\end{aligned} \tag{3.25}$$

Note that all components of the operator $\mathscr{N}$ are *diagonal* operators,

$$\begin{aligned}(N_{11}\, AX)_{(i,j)} &= n11_{(i,j),} AX_{(i,j)}, \\ (N_{12}\, AY)_{(i,j)} &= n12_{(i,j)}\, AY_{(i,j)}, \\ (N_{21}\, AX)_{(i,j)} &= n21_{(i,j)}, AX_{(i,j)}, \\ (N_{22}\, AY)_{(i,j)} &= n22_{(i,j)}\, AY_{(i,j)},\end{aligned} \tag{3.26}$$

where $n11_{(i,j)}$ is given by the formula

$$\begin{aligned}n11_{(i,j)} = {} & (\mathbf{KI}_{xx})_{(i,j)}\, V_{(i,j)}^{(i,j)} + (\mathbf{KI}_{xx})_{(i-1,j)}\, V_{(i,j)}^{(i-1,j)} \\ & + (\mathbf{KI}_{xx})_{(i-1,j-1)}\, V_{(i,j)}^{(i-1,j-1)} + (\mathbf{KI}_{xx})_{(i,j-1)}\, V_{(i,j)}^{(i,j-1)}.\end{aligned}$$

Formulas for $n12_{(i,j)}$, $n21_{(i,j)}$, $n22_{(i,j)}$ are similar.

This operator is symmetric and positive-definite in the formal inner product:

$$[\mathscr{N}\vec{A}, \vec{B}]_{\mathscr{H}\mathscr{N}} = [\vec{A}, \mathscr{N}\vec{B}]_{\mathscr{H}\mathscr{N}}, \qquad [\mathscr{N}\vec{A}, \vec{A}]_{\mathscr{H}\mathscr{N}} > 0. \tag{3.27}$$

The operator $\mathscr{S}$ can be written in block form:

$$\begin{aligned}\mathscr{S}\vec{A} &= \begin{pmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{pmatrix} \begin{pmatrix} AS\xi \\ AS\eta \end{pmatrix} \\ &= \begin{pmatrix} S_{11}\, AS\xi + S_{12}\, AS\eta \\ S_{21}\, AS\xi + S_{22}\, AS\eta \end{pmatrix}.\end{aligned} \tag{3.28}$$

The operators S_{11} and S_{22} are diagonal and the stencils for

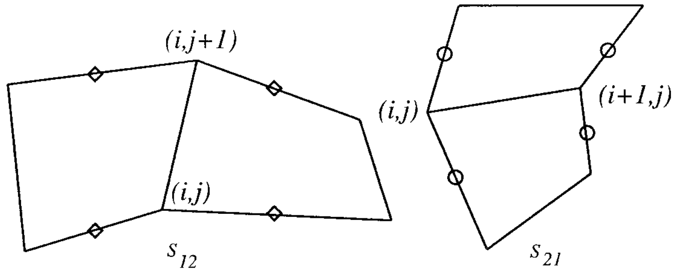


FIG. 5. The stencils for the operators S_{12} and S_{21}.

the operators S_{12} and S_{21} are shown in Fig. 5. A comparison of the natural inner product $(\vec{A}, \vec{B})_{\mathscr{HS}}$ and the formal inner product

$$\begin{aligned}[\mathscr{S}\vec{A}, \vec{B}]_{\mathscr{HS}} = & \sum_{i=1}^{M}\sum_{j=1}^{N-1} [(S_{11}\, AS\xi)_{(i,j)} + (S_{12}\, AS\eta)_{(i,j)}]\, BS\xi_{(i,j)} \\ & + \sum_{i=1}^{M-1}\sum_{j=1}^{N} [(S_{21}\, AS\xi)_{(i,j)} \\ & + (S_{22}\, AS\eta)_{(i,j)}]\, BS\eta_{(i,j)},\end{aligned} \tag{3.29}$$

gives

$$\begin{aligned}(S_{11}\, AS\xi)_{(i,j)} &= \left(\sum_{k,l=0}^{1} \frac{V_{(i,j+l)}^{(i-k,j)}\, T11_{(i,j+l)}^{(i-k,j)}}{\sin^2(\varphi_{(i,j+l)}^{(i-k,j)})}\right) AS\xi_{(i,j)}, \\ (S_{12}\, AS\eta)_{(i,j)} &= \sum_{k,l=0}^{1} (-1)^{k+l} \frac{V_{(i,j+l)}^{(i-k,j)}\, T12_{(i,j+l)}^{(i-k,j)}}{\sin^2(\varphi_{(i,j+l)}^{(i-k,j)})} AS\eta_{(i-k,j+l)}.\end{aligned} \tag{3.30}$$

Formulas for operators S_{21} and S_{22} are similar.

We remark that these formulas are valid only for $i = 2, ..., M - 2$; $j = 2, ..., N - 2$, but it is easy to show that if fictitious nodes are introduced for $i = 0$, $i = M + 1$, $j = 0$, and $j = N + 1$, whose coordinates are the same as for the corresponding real nodes, then the formulas are valid for all i and j. If all the weights $V_{(i,j)}^{p,q}$ are positive, then the operator $\mathscr{S}$ is symmetric and positive-definite in the formal inner product:

$$[\mathscr{S}\vec{A}, \vec{B}]_{\mathscr{HS}} = [\vec{A}, \mathscr{S}\vec{B}]_{\mathscr{HS}}, \qquad [\mathscr{S}\vec{A}, \vec{A}]_{\mathscr{HS}} > 0. \tag{3.31}$$

4. THE FINITE-DIFFERENCE METHOD

We now use the support-operators method to derive approximation to the divergence, flux operator, and variable coefficient Laplacian. We first derive a discrete approximation to the divergence, and then use this discrete divergence to derive the approximations to the flux operator and Laplacian using discrete analogs of the integral identities. Because of the principle role of the divergence operator, we call it the *prime operator*. Because the discrete approximations for the flux operator and Laplacian are derived from the prime operator, they are called *derived operators.*

4.1. The Prime Operator

A natural conservative invariant definition of the divergence operator is

$$\textbf{div}\, \vec{w} = \lim_{V\to 0} \frac{1}{V} \oint_V (\vec{w}, \vec{n})\, dV. \tag{4.1}$$

This identity is used in [2] to derive a discrete analog **DIV** of the divergence **div,** for both the nodal and surface discretizations.

4.1.1. *The Nodal Discretization for Vectors*

Here the cell discretization of scalar functions and the nodal discretization of vector functions are used. In the interior of the region, the prime operator $\mathscr{D}$ = **DIV**, which is the discrete analog of the divergence, is given by

$$\begin{aligned}(\mathscr{D}\vec{W})_{(i,j)} = & \frac{0.5}{VC_{(i,j)}} \\ & \{[(WX_{(i,j)} - WX_{(i+1,j+1)})\,(y_{(i+1,j)} - y_{(i,j+1)}) \\ & -(WX_{(i+1,j)} - WX_{(i,j+1)})\,(y_{(i,j)} - y_{(i+1,j+1)})] \\ & - [(WY_{(i,j)} - WY_{(i+1,j+1)})\,(x_{(i+1,j)} - x_{(i,j+1)}) \\ & - (WY_{(i+1,j)} - WY_{(i,j+1)})\,(x_{(i,j)} - x_{(i+1,j+1)})]\},\end{aligned} \tag{4.2}$$

while on the boundary, $\mathscr{D}$ gives an approximation of the normal component of the vector. For example, on the "bottom boundary" where $j = 1$ and $i = 1, ..., M - 1$, a unit normal vector is

$$\left(\frac{y_{(i+1,1)} - y_{(i,1)}}{l\xi_{(i,1)}}, -\frac{x_{(i+1,1)} - x_{(i,1)}}{l\xi_{(i,1)}}\right), \tag{4.3}$$

where, because the problem is two dimensional, $l\xi = S\eta$ and $l\eta = S\xi$ are the lengths of the edges of the cell, and then $\mathscr{D}$ is given by

$$\begin{aligned}(\mathscr{D}\vec{W})_{(i,0)} = -\Bigg(& \frac{W\,X_{(i,1)} + W\,X_{(i+1,1)}}{2} \frac{y_{(i+1,1)} - y_{(i,1)}}{l\xi_{(i,1)}} \\ & - \frac{WY_{(i,1)} + WY_{(i+1,1)}}{2} \frac{x_{(i+1,1)} - x_{(i,1)}}{l\xi_{(i,1)}}\Bigg).\end{aligned} \tag{4.4}$$

4.1.2. *The Surface Discretization for Vectors*

For a cell discretization for scalar functions and a surface discretization for vector functions, the prime operator (discrete divergence = $\mathscr{D}$ = **DIV**) is defined in the interior of the region by

$$(\mathscr{D}\vec{W})_{(i,j)} = \frac{1}{VC_{(i,j)}} \{(WS\xi_{(i+1,j)}\, S\xi_{(i+1,j)} - WS\xi_{(i,j)}\, S\xi_{(i,j)}) + (WS\eta_{(i,j+1)}\, S\eta_{(i,j+1)}) - WS\eta_{(i,j)}\, S\eta_{(i,j)})\}, \tag{4.5}$$

while on the boundary, $\mathscr{D}$ gives an approximation of the normal component of a vector,

$$\begin{aligned}(\mathscr{D}\vec{W})_{(i,0)} &= -WS\eta_{(i,1)}, \qquad i = 1, ..., M-1,\\ (\mathscr{D}\vec{W})_{(i,N)} &= +WS\eta_{(i,N)}, \qquad i = 1, ..., M-1,\end{aligned} \tag{4.6}$$

and similar formulas on other parts of the boundary.

4.2. The Derived Operator

The derived operator $\mathscr{G}$ is the discrete analog of the flux operator, and is defined by $\mathscr{G} = \mathscr{D}^*$. Here the adjoint is taken in the natural inner product. For the surface discretization on arbitrary grids, it is not possible to write a explicit formula for the components of the operator $\mathscr{G}$. However, it is possible to express $\mathscr{G}$ in terms of $\mathscr{M}$, $\mathscr{S}$, and $\mathscr{D}$. For the nodal discretization, $\mathscr{G}$ can be expressed explicitly in terms of $\mathscr{M}$, $\mathscr{N}$, and $\mathscr{D}$.

4.2.1. *The Surface Discretization for Vectors*

For the cell discretization of scalar functions and the surface discretization for vectors, $\mathscr{G}$ is computed by finding the adjoint of $\mathscr{D}: \mathscr{HS} \to HC$,

$$(\mathscr{D}\vec{W}, U)_{HC} = (\vec{W}, \mathscr{D}^*U)_{\mathscr{HS}}, \tag{4.7}$$

which can be rewritten in terms of the formal inner products as

$$[\mathscr{D}\vec{W}, \mathscr{M}U]_{HC} = [\vec{W}, \mathscr{S}\mathscr{D}^*U]_{\mathscr{HS}}. \tag{4.8}$$

The formal adjoint $\mathscr{D}\dagger$ of $\mathscr{D}$ is defined to be the adjoint in the formal inner product, so

$$[\vec{W}, \mathscr{D}\dagger\mathscr{M}U]_{\mathscr{HS}} = [\vec{W}, \mathscr{S}\mathscr{D}^*U]_{\mathscr{HS}}. \tag{4.9}$$

This relationship must be true for all $\vec{W}$ and U, so

$$\mathscr{D}\dagger\mathscr{M} = \mathscr{S}\mathscr{D}^*, \tag{4.10}$$

which gives

$$\mathscr{G} = \mathscr{D}^* = \mathscr{S}^{-1}\mathscr{D}\dagger\mathscr{M}. \tag{4.11}$$

Because $\mathscr{S}$ is banded, $\mathscr{S}^{-1}$ is likely to be full (unless $\mathscr{S}$ is diagonal). Hence $\mathscr{G}$ is full and has a *non-local* stencil. This is not a serious problem, because we do not need to explicitly form $\mathscr{G}$. The discrete fluxes are

$$\vec{W} = \mathscr{G}U = \mathscr{S}^{-1}\mathscr{D}\dagger\mathscr{M}U, \tag{4.12}$$

and if the operator $\mathscr{S}$ is applied to both sides of this equation, then

$$\mathscr{S}\vec{W} = \mathscr{D}\dagger\mathscr{M}U. \tag{4.13}$$

The operators on both sides of this equation have local stencils.

These equations are similar to the finite element and compact finite difference methods that can be expressed in the form (4.13) with local stencils (see, for example, [8–10]).

To find the fluxes for a given temperature, from (4.13) we must solve a system of linear equations. The discrete operator $\mathscr{S}$ is symmetric positive-definite and has five nonzero elements in each row (see (3.30) and Fig. 5). In Subsection 4.6, we discuss possible solution approaches.

The relationship $(\mathscr{D}\vec{W}, U)_{HC} = (\vec{W}, \mathscr{D}^*U)_{\mathscr{HS}}$ implies that

$$[\vec{W}, \mathscr{D}\dagger\mathscr{M}U]_{\mathscr{HS}} = [\mathscr{D}\vec{W}, \mathscr{M}U]_{HS}. \tag{4.14}$$

The right-hand side of this formula can be evaluated using (4.5) for $\mathscr{D}$ and summation by parts to give

$$-(\mathscr{D}\dagger\mathscr{M}U)_{(i,j)} = \begin{pmatrix} S\xi_{(i,j)}\,(U_{(i,j)} - U_{(i-1,j)}) \\ S\eta_{(i,j)}\,(U_{(i,j)} - U_{(i,j-1)}) \end{pmatrix}. \tag{4.15}$$

4.2.2. *The Nodal Discretization for Vectors*

In the case of the cell discretization of scalar functions and the nodal discretization of vectors, an argument similar to that given in the previous section gives

$$\mathscr{D}\dagger\mathscr{M} = \mathscr{N}\mathscr{D}^*, \tag{4.16}$$

which gives

$$\mathscr{G} = \mathscr{D}^* = \mathscr{N}^{-1}\mathscr{D}\dagger\mathscr{M}. \tag{4.17}$$

Note that the operator $\mathscr{D}$ here is not the same as in the previous section. As before, the fluxes are given by

$$\vec{W} = \mathscr{G}U = \mathscr{N}^{-1}\mathscr{D}\dagger\mathscr{M}U, \tag{4.18}$$

where $\vec{W} = (WX, WY)$. Applying the operator $\mathscr{N}$ to both sides gives

$$\mathcal{N}\vec{W} = \mathcal{D}\dagger\mathcal{M}U. \tag{4.19}$$

To simplify the notation, introduce $\vec{F}$,

$$\mathcal{N}\vec{W} = \vec{F} = \mathcal{D}\dagger\mathcal{M}U, \tag{4.20}$$

where $\vec{F} = (FX, FY)$. As before, the operator $\mathcal{N}$ is given by a two by two block of operators, but in this case, the blocks are diagonal, so $\mathcal{N}$ can be inverted explicitly by solving the left-hand equation in (4.20). The explicit form of this equations is

$$\begin{aligned} n11_{(i,j)}\, WX_{(i,j)} + n12_{(i,j)}\, WY_{(i,j)} &= FX_{(i,j)}\,, \\ n12_{(i,j)}\, WX_{(i,j)} + n22_{(i,j)}\, WY_{(i,j)} &= FY_{(i,j)}\,, \end{aligned} \tag{4.21}$$

and the solution is

$$\begin{aligned} WX_{(i,j)} &= \frac{FX_{(i,j)}\, n22_{(i,j)} - FY_{(i,j)}\, n12_{(i,j)}}{n11_{(i,j)}\, n22_{(i,j)} - n12^2_{(i,j)}}, \\ WY_{(i,j)} &= -\frac{FX_{(i,j)}\, n12_{(i,j)} - FY_{(i,j)}\, n11_{(i,j)}}{n11_{(i,j)}\, n22_{(i,j)} - n12^2_{(i,j)}}. \end{aligned} \tag{4.22}$$

The relationship $(\mathcal{D}\,\vec{W}, U)_{HC} = (\vec{W}, \mathcal{D}^*\, U)_{\mathcal{HN}}$ implies that

$$[\vec{W}, \mathcal{D}\dagger\mathcal{M}U]_{\mathcal{HN}} = [\mathcal{D}\vec{W}, \mathcal{M}U]_{HC}. \tag{4.23}$$

The right-hand side of the last formula can be evaluated using formula (4.2) for $\mathcal{D}$ and summation by parts to give an explicit formula for $\vec{F} = \mathcal{D}\dagger\mathcal{M}\,U$

$$\begin{aligned} -FX_{(i,j)} = 0.5\,\{&(y_{(i,j+1)} - y_{(i+1,j)})\, U_{(i,j)} + (y_{(i-1,j)} - y_{(i,j+1)}) \\ &U_{(i-1,j)} + (y_{(i,j-1)} - y_{(i-1,j)})\, U_{(i-1,j-1)} \\ &+ (y_{(i+1,j)} - y_{(i,j-1)})\, U_{(i,j-1)}\}. \end{aligned} \tag{4.24}$$

The formula for FY is given by changing y to $-x$ in the previous formula.

4.3. The Discrete Operator Equations

For both discretizations, the discrete analog of the continuum operator $\mathbf{\Omega}$ (see (2.11)) is defined by

$$(\Omega\, U)_{(i,j)} = \begin{cases} 0, & \text{in the interior,} \\ \alpha_{(i,j)} U_{(i,j)}, & \text{on the boundary.} \end{cases} \tag{4.25}$$

The finite difference method approximating the first-order system (2.6), written as an analog of the continuum-operator system (2.13), is

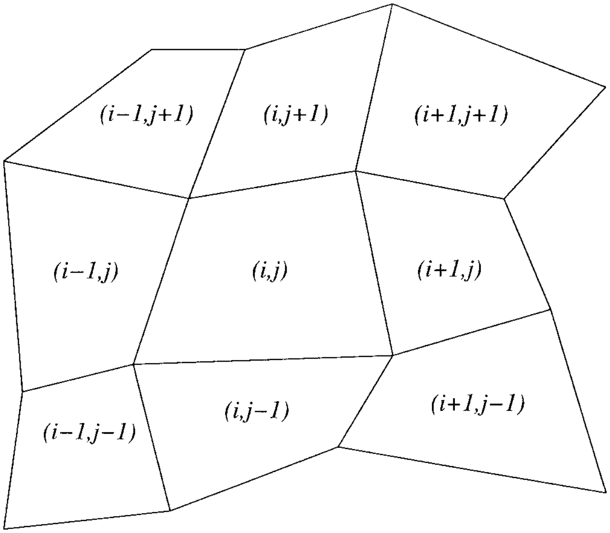


FIG. 6. The nine-cell stencil for the cell-node discretization for the operator $\mathcal{A}$.

$$\Omega U + \mathcal{D}\vec{W} = F, \qquad \vec{W} = \mathcal{G}U. \tag{4.26}$$

Then the discretization of the second-order equation (1.1), which is an analog of the operator equation (2.8), is

$$\mathcal{A}U = (\Omega + \mathcal{D}\mathcal{G})\, U = F. \tag{4.27}$$

For both discretizations, in the interior of the cells, Eq. (4.27) is

$$(\mathcal{D}\vec{W})_{(i,j)} = \mathbf{DIV}\,\vec{W}_{(i,j)} = f_{(i,j)}. \tag{4.28}$$

The approximation of the boundary conditions is

$$(\mathcal{D}\vec{W})_{(i,j)} + \alpha_{(i,j)}\; U_{(i,j)} = \psi_{(i,j)}\,, \tag{4.29}$$

where, on the boundary, the operator $\mathcal{D}$ is an approximation of the normal component of the vector.

Also, for both discretizations, the fluxes are determined from

$$\vec{W} = \mathcal{G}u. \tag{4.30}$$

For the cell-node discretization, the operator $\mathcal{G}$ can be constructed from (4.22) and (4.24) and there are local explicit formulas for the fluxes (see Fig. 6). For the cell-surface discretization, the operator $\mathcal{G}$ is non-local (see Subsection 4.2.1) and, consequently, there is no local explicit equation for the fluxes.

4.4. The Discrete Operators on a Rectangular Grid

On orthogonal grids and for diagonal **K**, the interior discretizations discussed in this paper reduce to well-known discretizations. In this section we consider the case where $\mathbf{K} = k\mathbf{I}$ on a rectangular grid with the cell sides $S\xi_{(i,j)} = hY$ and $S\eta_{(i,j)} = hX$ and cell volume $VC_{(i,j)} = hX\,hY$.

4.4.1. *The Cell-Node Discretization*

For the cell-node discretization, the **DIV** operator (4.2) is

$$\begin{aligned}(\mathbf{DIV}\,\vec{W})_{(i,j)} &= [0.5\,(WX_{(i+1,j)} + WX_{(i+1,j+1)})\\ &\quad - 0.5\,(WX_{(i,j)} + WX_{(i,j+1)})]/hX\\ &\quad + [0.5\,(WY_{(i+1,j+1)} + WY_{(i,j+1)})\\ &\quad - 0.5\,(WY_{(i+1,j)} + WY_{(i,j)})]/hY,\end{aligned} \tag{4.31}$$

which is a natural discretization for a rectangular grid.

Fluxes $WX_{(i,j)}$, $WY_{(i,j)}$ are given by

$$\begin{aligned}W\,X_{(i,j)} &= \tilde{k}_{(i,j)}\{[0.5\,(U_{(i,j)} + U_{(i,j-1)})\\ &\quad - 0.5\,(U_{(i-1,j)} + U_{(i-1,j-1)})]/hX\}.\\ W\,Y_{(i,j)} &= \tilde{k}_{(i,j)}\{[0.5\,(U_{(i,j)} + U_{(i-1,j)}\\ &\quad - 0.5\,(U_{(i,j-1)} + (U_{(i-1,j-1)})]/hX\}.\end{aligned} \tag{4.32}$$

where the expression

$$\tilde{k}_{(i,j)} = \left(\frac{1}{4}\sum_{k,l=0}^{1}\frac{1}{k_{(i-k,j-l)}}\right)^{-1} \tag{4.33}$$

is the two-dimensional cell-to-node harmonic average, and the expressions in curly braces are an approximation for the derivatives $\partial u/\partial x$ and $\partial u/\partial y$.

The stencil for the discrete analog of the Laplacian **div grad** is given by choosing **K** = **I:**

$\frac{1}{4\,hX^1} + \frac{1}{4\,hY^2}$	$\frac{-1}{2\,hX^2} + \frac{1}{2\,hY^2}$	$\frac{1}{4\,hX^2} + \frac{1}{4\,hY^2}$
$\frac{1}{2\,hX^2} + \frac{-1}{2\,hY^2}$	$\frac{-1}{hX^2} + \frac{-1}{hY^2}$	$\frac{1}{2\,hX^2} + \frac{-1}{2\,hY^2}$
$\frac{1}{4\,hX^2} + \frac{1}{4\,hY^2}$	$\frac{-1}{2\,hX^2} + \frac{1}{2\,hY^2}$	$\frac{1}{4\,hX^2} + \frac{1}{4\,hY^2}$

When the grid is square, the stencil becomes the five-point stencil shown in Fig. 7. On a square grid, the cell-node Laplacian has a nontrivial null space that includes the checker board mode.

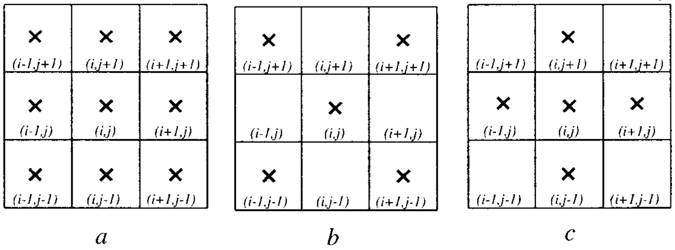


FIG. 7. The stencil of the Laplacian for (a) cell-node discretization, rectangular grid, (b) cell-node discretization, square grid, (c) cell-surface discretization, rectangular grid.

4.4.2. *The Cell-Surface Discretization*

For the surface discretization of vectors, formula (4.5) gives the operator **DIV** as

$$\begin{aligned}(\mathbf{DIV}\,\vec{W})_{(i,j)} &= \frac{WS\xi_{(i+1,j)} - WS\xi_{(i,j)}}{hX}\\ &\quad + \frac{WS\eta_{(i,j+1)} - WS\eta_{(i,j)}}{hY},\end{aligned} \tag{4.35}$$

which is also a natural discretization for the rectangular grid.

The operator $\mathscr{G}$ is computed from formula (4.13), and involves the operator $\mathscr{S}$ given by (3.30). For orthogonal grids, $\mathscr{S}$ is diagonal (the sines of all angles are one and the cosines of all angels are zero). Also, all $V^{(p,q)}_{(i,j)}$ are equal to $0.25\,hX\,hY$. Using (4.15), Eq. (4.13) for internal cells becomes

$$\left(0.5\,hX\,hY\sum_{k=0,1}\frac{1}{k_{(i-k,j)}}\right)WS\xi_{(i,j)} = hY\,(U_{(i,j)} - U_{(i-1,j)}) \tag{4.36}$$

or

$$WS\xi_{(i,j)} = \frac{2\,k_{(i-1,j)}\,k_{(i,j)}}{k_{(i-1,j)} + k_{(i,j)}}\,\frac{U_{(i,j)} - U_{(i-1,j)}}{hX}. \tag{4.37}$$

Thus, on rectangular grids, the cell-surface discretization leads to the well-known harmonic average for the coefficient k in the ξ direction. The formula for $WS\eta$ is similar,

$$WS\eta_{(i,j)} = \frac{2\,k_{(i,j-1)}\,k_{(i,j)}}{k_{(i,j-1)} + k_{(i,j)}}\,\frac{U_{(i,j)} - U_{(i,j-1)}}{hY}, \tag{4.38}$$

and contains the harmonic average for k in the η direction.

The fluxes on the boundary are one-sided differences. For example, on the left boundary

$$WS\xi_{(1,j)} = k_{(1,j)} \frac{U_{(1,j)} - U_{(0,j)}}{0.5\, hX}, \qquad WS\eta_{(i,1)} = k_{(i,1)} \frac{U_{(i,1)} - U_{(i,0)}}{0.5\, hY}. \tag{4.39}$$

The discrete analog of the Laplacian **div grad** is

$$\frac{U_{(i+1,j)} - 2\, U_{(i,j)} + U_{(i-1,j)}}{hX^2} + \frac{U_{(i,j+1)} - 2\, U_{(i,j)} + U_{(i,j-1)}}{hY^2} \tag{4.40}$$

which is the usual five point approximation on a rectangular grid with the stencil shown in Fig. 7.

4.5. Theoretical Properties of the Algorithms

For the cell-surface discretization, the properties for the operators **div** and **grad** were investigated in Shashkov and Steinberg [4], where it was shown that the divergence of a constant vector is zero (moreover in [11] we proved that **DIV** $\vec{A} = 0$ if and only if vector $\vec{A}$ can be presented as a discrete curl of another vector, $\vec{A} =$ **CURL** $\vec{B}$); that for smooth grids the point truncation errors for the divergence **DIV** and for the gradient **GRAD** are second order; and for general grids, **DIV** and **GRAD** are first-order accurate, and that the **DIV** is exact for the integral truncation error. In [12] we proved, using a rather lengthy geometric calculation, that for piecewise constant **K**, the discrete analog of **K grad** is exact on piecewise linear functions. In the following subsection, it is shown that, for the cell-surface discretization, the null space of $\mathscr{G}$ is exactly the constants, while for nodal discretization on square grids, the null space of $\mathscr{G}$ contains highly oscillatory checker board grid functions in addition to the constants.

In Section 5, the approximation properties of the variable-coefficient Laplacian **div K grad** are numerically shown to confirm these theoretical results.

4.5.1. *The Null Space of the Operator $\mathscr{G}$, Cell-Surface Discretization*

For the cell-surface discretization, we prove that $\mathscr{G}U$ is zero if and only if U is constant. Formula (4.11) gives

$$\mathscr{G}U = \mathscr{S}^{-1}\mathscr{D} \dagger \mathscr{M}U, \tag{4.41}$$

and then, if U is a constant, (4.15) shows that $\mathscr{D} \dagger \mathscr{M}U = 0$, so $\mathscr{G}U = 0$.

Conversely, assume that $\mathscr{G}U = 0$. Formula (4.41) and the fact that operator $\mathscr{S}$ is positive definite gives

$$\mathscr{D} \dagger \mathscr{M}U = 0. \tag{4.42}$$

This and formula (4.15) then give

$$U_{(i,j)} - U_{(i-1,j)} = 0; \qquad i = 1, ..., M; j = 1, ..., N-1;$$
$$U_{(i,j)} - U_{(i,j-1)} = 0; \qquad i = 1, ..., M-1; j = 1, ..., N;$$

which implies that U is a constant. Therefore the null space of the discrete operator $\mathscr{G}$ is the constant functions, exactly as for the different operator k **grad**.

4.5.2. *The Null Space of the Operator $\mathscr{G}$, Cell-Node Discretization*

For the cell-node discretization, the situation is quite different: in the case of a square grid, Formula (4.24) shows that both the constant function $U_{(i,j)} = 1$ and the "checkerboard" function $U_{(i,j)} = (-1)^{i+j}$ satisfy $\mathscr{G}U = 0$, as do any linear combinations of these functions. The checkerboard solution or mode is well known, especially in computational Lagrangian gas dynamics, where it leads to the so-called "hour-glassing" instability [13]. In case of elliptic equations, this mode leads to the presence of high-frequency noise in the solution, as illustrated by the numerical examples in Section 5.

4.6. Solving the System of Linear Equations

The discrete equations for both the cell-node and the cell-surface discretization have the form (4.26):

$$\Omega U + \mathscr{D}\vec{W} = F, \qquad \vec{W} = \mathscr{G}U, \tag{4.43}$$

The fluxes can be eliminated from this system to obtain an equation for U,

$$\mathscr{A}U = \Omega U + \mathscr{D}\mathscr{G}U = F, \tag{4.44}$$

where $\mathscr{A}$ is symmetric and positive definite.

For the cell-node discretization, the operator $\mathscr{A}$ is symmetric and positive definite, and has a local 9 cell stencil. In our numerical examples to compare the accuracy of the methods, we used a simple SOR iteration to solve this system. More efficient iteration methods, such as multigrid [14–17] or incomplete Cholesky conjugate gradient methods [18], could also have been used to solve these equations.

For the case of cell-surface discretization, the operators $\mathscr{G}$ and $\mathscr{A}$ are *non-local* and, therefore, algorithms that require explicit expressions for $\mathscr{A}$ are impractical for large problems. The equations can be formulated so that algorithms, such as preconditioned conjugate gradient methods, requiring only a multiplication of a vector by $\mathscr{A}$ can be used. Given U, $\mathscr{A}\,U$ can be computed efficiently by solving (4.13), $\mathscr{S}\,\vec{W} = \mathscr{D}^\dagger\,\mathscr{M}\,U$, for $\vec{W}$ and evaluating $\mathscr{A}\,U = \Omega U + \mathscr{D}\,\vec{W}$. All operators in this formulation are explicitly known and local. Moreover, because $\mathscr{S}$ is a positive definite symmetric local operator, the equation for $\vec{W}$ can be solved efficiently with iterative methods. On orthogonal grids $\mathscr{S}$ is a diagonal operator, and all steps of this procedure are local.

Other efficient algorithms to solve this system include the family of two-level gradient methods, including the minimal residual method, the minimal correction method, and the minimal error method. All these methods can be written as

$$B\,\frac{U^{(s+1)} - U^{(s)}}{\tau_s} + \mathscr{A}\,U^{(s)} = F, \tag{4.45}$$

where $U^{(s)}$ is the approximate solution to U^{n+1} on iteration number s, τ_s some iteration parameter, and operator B is the preconditioner. A family of three-level iteration methods, which require only the computation of $\mathscr{A}\,U$, includes the three-level conjugate-direction methods, like the conjugate gradient method. All these methods can be written as

$$\begin{aligned} BU^{(s+1)} &= \alpha_{s+1}\,(B - \tau_{s+1}\,\mathscr{A})\,U^{(s)} + (1 - \alpha_{s+1})\,BU^{(s-1)} \\ &\quad + \alpha_{s+1}\,\tau_{s+1}\,F, \\ BU^{(1)} &= (B - \tau_1\,\mathscr{A})\,U^{(0)} + \tau_1\,F. \end{aligned}$$

The effectiveness of these methods strongly depends on the choice of a preconditioner. The simplest Jacobi type preconditioner approximates $\mathscr{S}$ by its diagonal blocks. This is exact for orthogonal grids and produces a five-cell symmetric positive-definite operator corresponding to removing the mixed derivatives from the variable-coefficient Laplacian on non-orthogonal grids.

4.6.1. *Solving for the Fluxes*

Given U, the system (4.13) must be solved to obtain the flux. In our examples, we used the block Gauss–Seidel algorithm,

$$\mathscr{S}_{11}\,WS\xi^{(s+1)} + \mathscr{S}_{12}\,WS\eta^{(s)} = \mathscr{F}\mathscr{X}, \tag{4.46}$$

$$\mathscr{S}_{21}\,WS\xi^{(s+1)} + \mathscr{S}_{22}\,WS\eta^{(s+1)} = \mathscr{F}\mathscr{Y}, \tag{4.47}$$

where (s) and $(s + 1)$ are iteration indices. Equation (4.46) gives all $WS\xi^{(s+1)}$ fluxes for the new iteration $(s + 1)$, and then from Eq. (4.47), all fluxes $WS\eta^{(s+1)}$ can be found. Because the matrix of the operators $\mathscr{S}_{11}$ and $\mathscr{S}_{22}$ are diagonal, systems (4.46) and (4.47) can be solved explicitly. Because the operators for these equations are symmetric and positive definite, the block Gauss–Seidel method always converges.

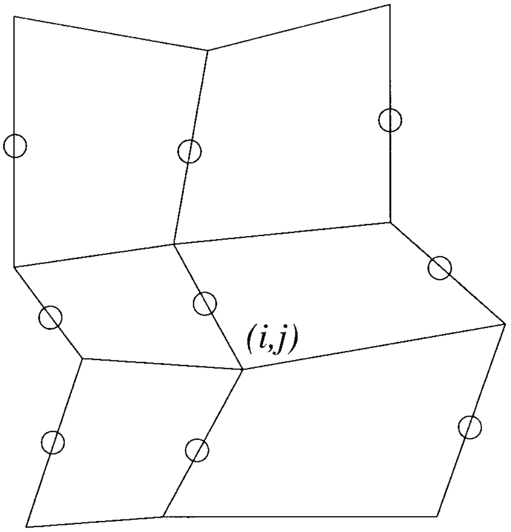


FIG. 8. The stencil for $WS\xi$.

Another approach to solving (4.13) is to use the fact that the operators $\mathscr{S}_{11}$ and $\mathscr{S}_{22}$ are diagonal, and then eliminate either $WS\xi$ or $WS\eta$ from (4.46). For example, the equation resulting from eliminating $WS\eta$ is

$$(\mathscr{S}_{11} + \mathscr{S}_{12}\mathscr{S}_{22}^{-1}\,\mathscr{S}_{21})\;WS\xi = \mathscr{F}\mathscr{X} - \mathscr{S}_{22}\,\mathscr{F}\mathscr{Y}. \tag{4.48}$$

The operator of this system,

$$(\mathscr{S}_{11} + \mathscr{S}_{12}\mathscr{S}_{22}^{-1}\,\mathscr{S}_{21}), \tag{4.49}$$

is symmetric and positive definite, and has the local stencil shown in Fig. 8.

5. NUMERICAL EXAMPLES

The examples in this section, summarized in Table I, solve the elliptic PDE (1.1) and were chosen to illustrate

TABLE I

Summary of Examples

No.	Name	**K** diag	**K** cont	Grid uniform
5.1	Crumption 1	no	yes	yes
5.2	Crumption 2	no	no	yes
5.3	Durlofsky 1	yes	no	yes
5.4	Durlofsky 2	no	no	no
5.5	Das	no	no	no

how the cell-node and cell-surface algorithms perform for discontinuous non-diagonal **K** and on nonuniform grids. Examples 1 and 2 are used to compare the convergence rates of the algorithms for various kinds of coefficients and grids. Example 3 shows that the algorithms can compute accurate total fluxes for very rough **K**. Example 4 confirms that the algorithms produce reasonable results for the case of non-diagonal and discontinuous **K** and nonuniform grids, while Example 5 verifies that for non-diagonal and discontinuous **K** and non-uniform grids, but where the solution is piecewise linear, that the cell-surface algorithm is exact and the cell-node algorithm is second-order accurate.

Some additional examples, and, in particular, comparison with method from [19] can be found in [4].

The asymptotic truncation error E_h on a grid of $M \times N$ nodes,

$$h = \max\left\{\frac{1}{M-1}, \frac{1}{N-1}\right\}, \tag{5.1}$$

is estimated by

$$\|E_h\| = C\,h^q + O(h^{q+1}), \tag{5.2}$$

where q is the order of the error, and the constant C, the convergence-rate constant, is independent of h, and $\|\cdot\|$ is some norm.

In the numerical examples the truncation errors were evaluated on a sequence of grids h, $h/2$, $h/4$, ..., and the convergence rate q estimated from the ratio between the norms of the errors $\|E_h\|$ and $\|E_{h/2}\|$ as

$$q \approx \log_2 \frac{\|E_h\|}{\|E_{h/2}\|}. \tag{5.3}$$

In the numerical experiments, continuum functions are discretized using the projection operator

$$(p_h\,u)_{i,j} = u(x^c_{i,j}, y^c_{i,j}),$$

where $x^c_{i,j}, y^c_{i,j}$ are the coordinates of the geometric center of the cell.

The convergence rates were estimated using both the maximum norm

$$E_{\max} = \|U - p_h\,u\|_{\max} = \max_{i,j} |U_{i,j} - (p_h\,u)_{i,j}|$$

and the mean-square norm

$$E_{L_2} = \|U - p_h\,u\|_{L_2} = \left(\sum_{i=1}^{M-1}\sum_{j=1}^{N-1} (U_{i,j} - (p_h\,u)_{i,j})^2\,VC_{i,j}\right)^{1/2}, \tag{5.4}$$

TABLE II

Example 5.1: Convergence Rates with Cell-Node and Cell-Surface Discretizations

Method	M	Max norm	L_2-norm	$q_{\max}$	q_2
Cell-node	17	3.74E-3	1.06E-3	1.84	2.03
	33	9.66E-4	2.58E-4	1.97	2.02
	65	2.45E-4	6.36E-5	—	—
Cell-surface	17	5.11E-3	1.68E-3	1.92	2.01
	33	1.35E-3	4.15E-4	1.95	2.09
	65	3.48E-4	9.73E-5	—	—

where U is the solution of the finite-difference method and u is the exact solution.

5.1. Non-diagonal Continuous K

Problem 1, from Crumpton, Shaw, and Ware [17], has a constant non-diagonal **K**, defined on the unit square with Dirichlet boundary conditions obtained from the exact solution. The permeability is

$$\mathbf{K} = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}, \tag{5.5}$$

where **K** is a positive definite matrix. The true solution is $u = e^{xy}$, which corresponds to this right-hand side

$$f(x, y) = -2(1 + x^2 + xy + y^2)e^{xy}. \tag{5.6}$$

Convergence-rate data for the cell-node and cell-surface discretizations are given in Table II, which indicates second-order convergence rates for both methods, both in max and L_2 norms.

5.2. Non-diagonal Discontinuous K

This test problem from Crumpton, Shaw, and Ware [17] is defined on the square $[-1, 1] \times [-1, 1]$, with Dirichlet boundary conditions. The diffusion coefficient is given by

$$\mathbf{K} = \begin{cases} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, & x < 0, \\ \alpha\begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}, & x > 0, \end{cases} \tag{5.7}$$

where the parameter α is used to vary the strength of the discontinuity at $x = 0$. The exact solution is

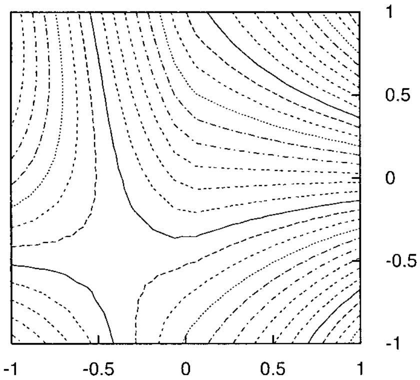


FIG. 9. Example 5.2. The isolines for the pressure, $M = N = 17$.

$$u(x, y) = \begin{cases} [2\sin(y) + \cos(y)]\,\alpha x + \sin(y), & x < 0, \\ \exp(x)\sin(y), & x > 0. \end{cases} \tag{5.8}$$

The right-hand side, which corresponds to this solution, is also discontinuous:

$$f(x, y) = \begin{cases} [-2\sin(y) - \cos(y)]\,\alpha x - \sin(y), & x < 0, \\ 2\alpha\exp(x)\cos(y), & x > 0. \end{cases} \tag{5.9}$$

No special discretization is needed for the right-hand side because our method uses a cell-centered discretization for u and the discrete analog of operator **div K grad** whose domain and range coincide with HC. The right-hand side is also assumed to be given in the cells and the material discontinuity coincides with a grid line.

Figure 9 displays the isolines for the approximate solution for $M = N = 17$, and $\alpha = 1$. Table III gives the convergence-rate data for both the cell-node and the cell-surface discretizations. These data verify the second-order convergence rate for the cell-surface algorithm and a first-order convergence rate for the cell-node algorithm.

TABLE III

Example 5.2: Convergence Rates with Cell-Node and Cell-Surface Discretization

Method	M	Max norm	L_2-norm	$q_{\max}$	q_2
Cell-node	17	1.00E-2	7.98E-3	0.82	1.17
	33	5.66E-3	3.53E-3	0.90	1.03
	65	3.02E-3	1.72E-3	—	—
Cell-surface	17	9.63E-3	7.05E-3	1.89	2.02
	33	2.59E-3	1.73E-3	1.96	2.12
	65	6.72E-4	3.96E-4	—	—

5.3. Flow through a Sand-Shale System

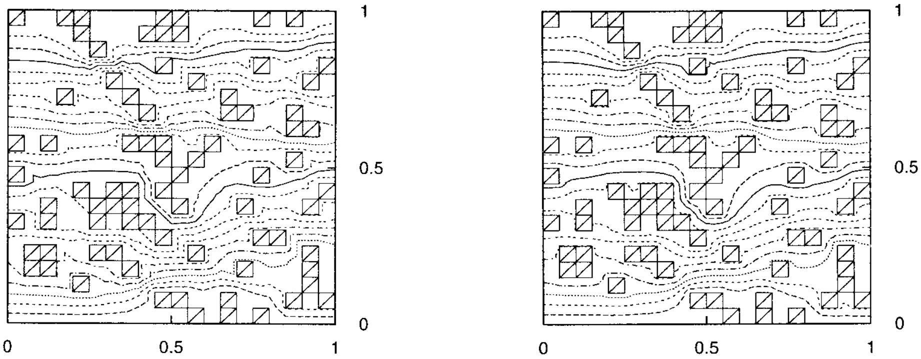
This example is from Durlofsky [20] and is defined on the unit square, with the boundary conditions $u = 0$ along $x = 0$, $u = 1$ along $x = 1$, and the flux is equal to zero along $y = 0$ and $y = 1$. The permeability field is generated by randomly placing shale blocks, of total area fraction 0.2, throughout the sand on a regular grid of dimension 20×20 (see Fig. 10). In the example, the permeabilities of both the sand (k_{sand}) and shale (k_{shale}) are taken to be uniform and isotropic, $\mathbf{K} = k\mathbf{I}$, with $k_{\text{sand}} = 1$ and $k_{\text{shale}} = 10^{-6}$. The ''exact'' flux through the system (which is flux obtained on a very fine grid) is 0.5202 (see [20]).

The streamlines for the case of the cell-surface discretization, which were obtained from the vector field by computing the stream function and then drawing its isolines, are shown in Fig. 10. Analysis of the isolines of the pressure also shows the more regular behavior of the pressure for the case of the cell-surface discretization. The convergence rates for the total flux are presented in Table IV. For $N = M = 21$, the cell-node discretization gives an absolutely unphysical result. This can be explained as follows: the fluxes are computed at nodes, and the corresponding elements of the matrix $\mathbf{K}$ are also computed at the nodes by a harmonic average from four neighboring cells, so for 20×20 cells, all nodes have very small permeability, which almost blocks the system.

It is interesting to compare the accuracy of our cell-surface discretization and the mixed finite-element method, described in Durlofsky [20], as a function of the number of unknowns. In [20], the total flux, which is obtained using 1240 unknowns, is 0.4508, while our method gives a flux of 0.451 for $M = N = 21$ (the correct result is 0.5205). For a uniform grid used in this example, and for a diagonal $\mathbf{K}$ matrix, as used in this example, the fluxes can be eliminated from the system and then the resulting system of linear equations contains only the pressure. That is, the number of unknowns, taking into account the Dirichlet boundary conditions on the left and right boundaries, is equal to $(M - 1) \times (N + 1) = 440$. If the fluxes are not eliminated, then the number of unknowns is equal to $(M - 1) \times (N + 1) + M \times (N - 1) + (N - 2) \times (M - 1) = 1240$, exactly as for the mixed finite-element method. Because of these differences, to compare the two methods, it is not appropriate to compare only the number of unknowns, but the structure of the matrix and the solution procedure for the system of linear equations must also be considered.

5.4. Flow through a System Containing an Impermeable Streak

This example, similar to one in Durlofsky [20], uses the logically rectangular grid on the unit square shown Fig.



N = M = 21

N = M = 41

FIG. 10. Example 5.3. Streamlines for the sand-shale problem, $M = N = 21$ and $M = N = 41$.

11a. The top curve is chosen to be an arc of a circle with the center at $(0.1, -0.4)$ and radius equal to 1.2. The bottom curve an arc of a circle with the same center and with radius equal to 1.1.

The permeability throughout the domain is uniform and isotropic ($\mathbf{K} = \mathbf{I}$), except in the low-permeability streak where the permeability is set such that the component parallel to the local streak orientation ($k_\parallel$) is equal to 0.1 and the component perpendicular to the streak orientation ($k_\perp$) is equal to 0.001. In the streak, the tensor $\mathbf{K}$ is a full tensor, in terms of its Cartesian components, which vary with (x, y) and are readily determined from the knowledge of $k_\parallel$ and $k_\perp$. For the Cartesian components $\mathbf{K}_{xx}$, $\mathbf{K}_{xy}$, $\mathbf{K}_{yy}$, which are used in cell-node discretization, the transformation formulas are

$$K_{xx} = k_\parallel \cos^2 \varphi + k_\perp \sin^2 \varphi, \tag{5.10}$$

$$K_{xy} = (k_\parallel - k_\perp) \cos \varphi \sin \varphi, \tag{5.11}$$

$$K_{yy} = k_\parallel \sin^2 \varphi + k_\perp \cos^2 \varphi, \tag{5.12}$$

TABLE IV

Example 5.3: Convergence Rates for Flux with Cell-Node and Cell-Surface Discretization

Method	M	Approx. flux	Exact flux	Error	q
Cell-node	21	0.022	0.5205	0.498	1.39
	41	0.33	0.5205	0.190	1.07
	81	0.43	0.5205	0.090	—
Cell-surface	21	0.45	0.5205	0.070	1.22
	41	0.49	0.5205	0.030	1.32
	81	0.508	0.5205	0.012	—

where $\varphi = \varphi(x, y)$ is the angle of rotation of the orthogonal coordinate system where the tensor $\mathbf{K}$ is diagonal and has components $k_\parallel$ and $k_\perp$. In our case

$$\sin \varphi = -\frac{x'}{\sqrt{((x')^2 + (y')^2)}}, \qquad \cos \varphi = \frac{y'}{\sqrt{((x')^2 + (y')^2)}}, \tag{5.13}$$

where $x' = x - 0.1$ and $y' = y + 0.4$.

Figure 11b displays the velocity field for the case of the cell-node discretization (the length of arrows is proportional to the module of the vectors). The results of the cell-surface discretization will be similar. As expected physically, no flow enters the streak, so these results are qualitatively similar to the best results in Durlofsky [20].

5.5. Non-diagonal, Piecewise Continuous K

In this example from Das, Schaffer, Steinberg, and Weber [21], the region is the unit square, the boundary conditions are the normal flux given by the exact solution on the top and the bottom boundaries, and the Dirichlet condition is given by the exact solution on the left and right boundaries. The non-diagonal permeability (or diffusion) matrix has a jump discontinuity of height λ along the line $r\,x + sy = \delta$, where $0 \leq r, s, \delta \leq 1$ and $r + s = 1$ (see Fig. 12a). The matrix $\mathbf{K}$ is

$$\mathbf{K} = k(x, y) \begin{pmatrix} 1 & 1/10 \\ 1/10 & 1 \end{pmatrix}, \tag{5.14}$$

where

$$k(x, y) = \begin{cases} 1, & \text{if } 0 \leq r\,x + sy < \delta, \\ \lambda, & \text{if } \delta \leq r\,x + sy < 1. \end{cases} \tag{5.15}$$

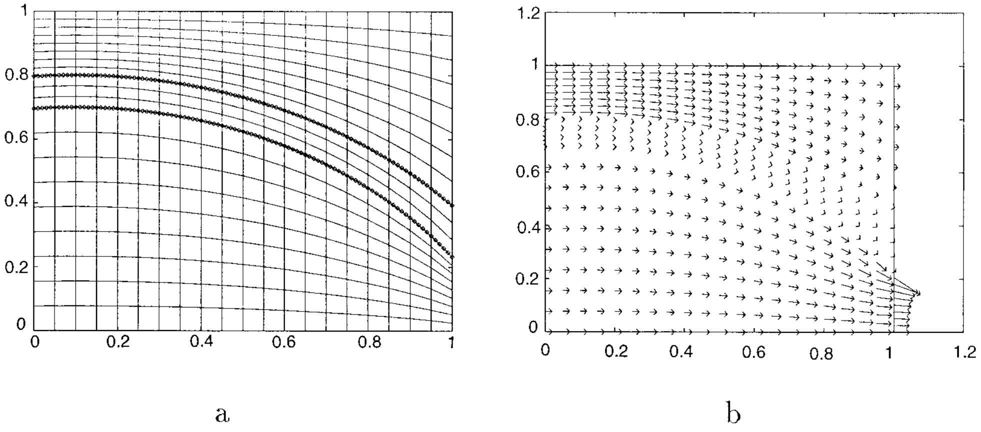


FIG. 11. Example 5.4. (a) The geometry and the grid for the streak, (b) velocity field for cell-node discretization.

The exact solution for the case when the right-hand side equals zero is

$$u(x, y) = \begin{cases} \dfrac{\lambda(rx + sy)}{1 + \delta(\lambda - 1)}, & \text{if } 0 \le rx + sy < \delta, \\ \dfrac{(rx + sy) + \delta(\lambda - 1)}{1 + \delta(\lambda - 1)}, & \text{if } \delta \le rx + sy < 1. \end{cases} \tag{5.16}$$

As expected, the method for the case of cell-surface discretization is exact for the piecewise linear solution.

Figure 12b displays the approximate solution for $r = 0.7$ and $\lambda = 10$, for the cell-node discretization. Table V verifies that the convergence rates are second-order.

ACKNOWLEDGMENTS

This work was performed under the auspices of the U.S. Department of Energy under Contract W-7405-ENG-36 and DOE/BES Program in the Applied Mathematical Sciences Contract KC-07-01-01.

The authors thank J. E. Dendy, L. G. Margolin, and J. E. Morel for many fruitful discussions and comments on various drafts of the paper, and B. Swartz for discussions on the form of the conductivity tensor.

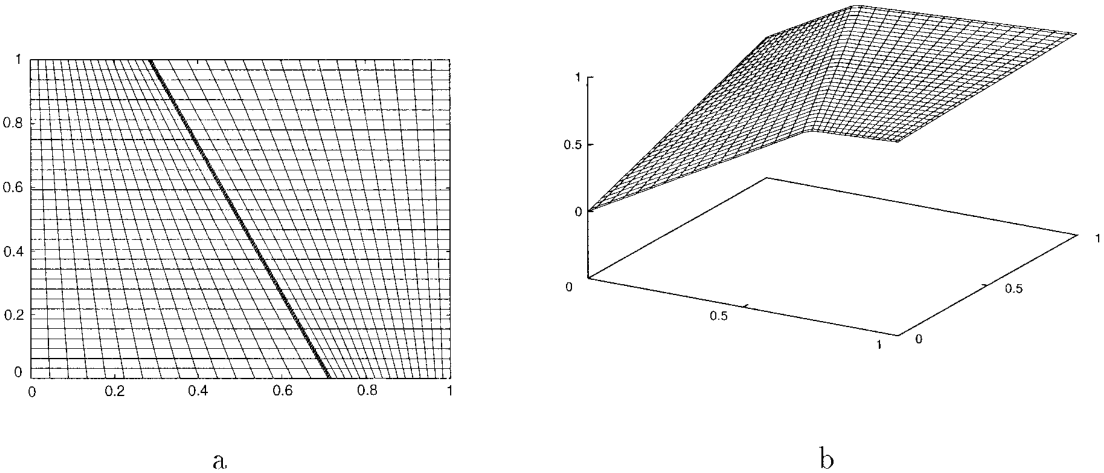


FIG. 12. Example 5.5. (a) Grid and discontinuity line, $M = N = 33$, (b) pressure, $M = N = 33$.

TABLE V

Example 5.5: Convergence Rates with Cell-Node Discretization

M	Max norm	L_2-norm	$q_{\max}$	q_2
17	5.98E-4	2.15E-4	1.74	1.99
33	1.78E-4	5.38E-5	1.83	1.98
65	4.98E-5	1.36E-5	—	—

REFERENCES

1. A. Favorskii, A. Samarskii, M. Shashkov, and V. Tishkin, Operational finite-difference schemes, *Differential Equations* **17,** 854 (1981).
2. M. Shashkov, *Conservative Finite-Difference Methods on General Grids* (CRC Press, Boca Raton, FL, 1995).
3. M. Shashkov and S. Steinberg, Support-operator finite-difference algorithms for general elliptic problems, *J. Comput. Phys.* **118,** 131 (1995).
4. M. Shashkov and S. Steinberg, Solving diffusion equations with rough coefficients in rough grids, *J. Comput. Phys.,* **129,** 383–405, (1996).
5. L. J. Durlofsky, A triangle based mixed finite element-finite volume technique for modeling two phase flow through porous media, *J. Comput. Phys.* **105,** 252 (1993).
6. M. F. Wheeler and R. Gonzalez, Mixed finite element methods for petroleum reservoir engineering problems, in *Computing Methods in Applied Science and Engineering,* Vol. VI, edited by R. Glowinski and J.-L. Lions, pp. 639–657.
7. P. M. Knupp and S. Steinberg, *The Fundamentals of Grid Generation* (CRC Press, Boca Raton, FL, 1993).
8. S. K. Lele, Compact finite difference schemes with spectral-like resolution, *J. Comput. Phys.* **103,** 16 (1992).
9. M. H. Carpetner, D. Gottlieb, and S. Abarbanel, Time stable boundary conditions for finite-difference schemes solving hyperbolic systems: Methodology and application to high-order compact schemes, *J. Comput. Phys.* **111,** 220 (1994).
10. S. H. Leventhal, An operator compact implicit method of exponential type, *J. Comput. Phys.* **46,** 138 (1982).
11. J. M. Hyman and M. Shashkov, Natural discretizations for the divergence, gradient and curl on logically rectangular grid, *Int. J. Comput. Math. Appl.* **33**(4), 81 (1997).
12. J. M. Hyman, M. Shashkov, and S. Steinberg, The numerical solution of diffusion problems in strongly heterogeneous non-isotropic materials, Technical Report (LA-UR-96-532, Los Alamos National Laboratory, Los Alamos, NM (unpublished).
13. L. G. Margolin and J. J. Pyun, A method for treating hourglass pattern, Technical Report (LA-UR-87-439, Los Alamos National Laboratory, Los Alamos, NM (unpublished).
14. R. E. Alcouffe, A. Brandt, J. E. Dendy, Jr., and J. W. Painter, The multi-grid method for the diffusion equation with strongly discontinuous coefficients, *SIAM J. Sci. Statist. Comp.* **2,** 430 (1981).
15. J. E. Dendy, Jr., Black box multigrid, *J. Comput. Phys.* **48,** 366 (1982).
16. P. M. de Zeeuw, Matrix-dependent prolongation and restriction in the blackbox multi-grid solver, *J. Comput. Appl. Math.* **3,** 1 (1990).
17. P. I. Crumpton, G. I. Shaw, and A. F. Ware, Discretization and multigrid solution of elliptic equations with mixed derivative terms and strongly discontinuous coefficients, *J. Comput. Phys.* **116,** 343 (1995).
18. D. S. Kershaw, The incomplete Choleski-conjugate gradient methods for the iterative solution of systems of linear equations, *J. Comput. Phys.* **26,** 43 (1978).
19. J. M. Morel, J. E. Dendy, Jr., M. L. Hall, and S. W. White, A cell-centered Lagrangian-mesh diffusion differencing scheme, *J. Comput. Phys.* **103,** 286 (1992).
20. L. J. Durlofsky, Accuracy of mixed and control volume finite element approximations to Darcy velocity and related quantities, *Water Resources Res.* **30**(4), 965 (1994).
21. B. Das, S. Schaffer, S. Steinberg, and S. Weber, Finite difference methods for modeling porous media flows, *Transport Porous Media* **17,** 171 (1994).