JOURNAL OF COMPUTATIONAL PHYSICS **129,** 383–405 (1996)
ARTICLE NO. 0257

Solving Diffusion Equations with Rough Coefficients in Rough Grids

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

Received March 27, 1996; revised August 8, 1996

A finite-difference algorithm for the numerical solution of diffusion problems in strongly heterogeneous and nonisotropic media is constructed for logically rectangular grids. The performance of this algorithm is comparable to other algorithms for problems with smooth coefficients and regular grids, and it is superior for problems with rough coefficients and/or skewed grids. The algorithm is derived using the *support-operators method,* which constructs discrete analogs of the divergence and flux operator that satisfy discrete analogs of the important integral identities relating the continuum operators. This paper gives the first application of this method to the solution of diffusion problems in heterogeneous an nonisotropic media.

The support-operators method forces the discrete analog of the flux operator to be the negative adjoint of the discrete divergence in an inner product weighted by the conductivity, as in the differential case. Once this is accomplished, many other important properties follow; for example, the scheme is conservative and the discrete analog of the variable material Laplacian is symmetric and negative definite. In addition, on any grid, the discrete divergence is zero on constant vectors and the discrete flux operator is exact for linear functions in case when **K** is piecewise constant. Moreover, the discrete gradient's null space is the constant functions, just as in the continuum. Because the algorithm is flux based, it has twice as many unknowns as more standard algorithms. However, the matrices that need to be inverted are symmetric and positive definite, so the most powerful linear solvers can be applied. Also, the scheme is second-order accurate so, all things considered, it is efficient. For rectangular grids, the discrete operators reduce to well-known discrete operators and the treatment of discontinuous conductivity coefficients in the case of isotropic media is equivalent to the well-known harmonic-averaging procedure. Comparison with standard schemes is presented. Numerical examples validate advantage of new method. © 1996 Academic Press, Inc.

1. INTRODUCTION

The main goal of this paper is the description and investigation of a new finite-difference algorithm for solving diffusion equations with rough coefficients on general logically rectangular grids. The algorithm is derived using the method of *support operators* [10, 30, 31], which requires that the diffusion equation be written in terms of invariant operators, in this case, the divergence $\nabla\cdot$ and the gradient ∇:

$$a\frac{\partial u}{\partial t} = \nabla\cdot\mathbf{K}\,\nabla u + f. \tag{1.1}$$

The diffusion equation (1.1) can be used to describe heat processes, general diffusion processes [22], and flows through porous media [36, 7, 5], where problems of the generality considered here commonly occur. In the case of heat diffusion, u is the temperature, $\mathbf{K}$ is the conductivity matrix or tensor, $a = c\rho > 0$, where c is the heat capacity, ρ is the density, and f is the density of the sources or sinks. The operator $\nabla\cdot\,\mathbf{K}\,\nabla$ is called the *diffusion operator* and is the Laplacian when $\mathbf{K}$ is the identity.

Equation (1.1) holds in some planar region V and in this region it is assumed that a is uniformly bounded above and below by some positive constants. It is *not* assumed that $\mathbf{K}$ is continuous; $\mathbf{K}$ may have a finite number of jump discontinuities. The main results in this paper are for the case of diagonal conductivity $\mathbf{K} = k\mathbf{I}$, where it is assumed that k has uniform upper and lower bounds in V, but much of the preliminary discussion is carried out for general symmetric $\mathbf{K}$ whose eigenvalues have uniform upper and lower bounds in V (see [13] for an extensive discussion of the nondiagonal case).

The flux vector $\mathbf{w}$ plays an important role in the numerical method to be derived, so it is introduced and used to write the diffusion Equation (1.1) as a first-order system:

$$a\frac{\partial u}{\partial t} = -\nabla\cdot\mathbf{w} + f, \quad \mathbf{w} = -\mathbf{K}\,\nabla u. \tag{1.2}$$

This is the *flux* form of diffusion equation that is commonly used in the *mixed* finite-element method [7, 38] and is a natural form to use in the case of discontinuous $\mathbf{K}$. In this paper, the support-operators method is extended so that the *flux operator* $\mathbf{K}\,\nabla$, rather than just the gradient ∇, is used as a main invariant operator. This is particularly im-

0021-9991/96 $18.00

portant for the case of discontinuous coefficients and is closely related to the well-known [24, 36] harmonic averaging of coefficients commonly used for such problems.

The boundary conditions on ∂V are chosen to be of general mixed or Robin type,

$$\beta(\mathbf{K}\,\nabla u, \mathbf{n}) + \alpha u = \psi, \tag{1.3}$$

where α, β, and ψ are given smooth functions on ∂V, $\mathbf{n}$ is the unit outward normal to ∂V, and $(\cdot,\cdot)$ is the standard scalar product of vectors. Only problems with solutions that are bounded in time are considered, so it is assumed that $\alpha\beta \leq 0$. Using the flux vector, the boundary condition can be written

$$\beta(\mathbf{w}, \mathbf{n}) + \alpha u = \psi. \tag{1.4}$$

Also, the standard initial condition, $u = u_0$ at $t = 0$, is used.

We are particularly interested in solving problems on grids of the type that appear in Lagrangian fluid dynamics codes; that is, the grids are logically rectangular, which means that, in two dimensions, the grids points can be written as $x(i, j)$, $y(i, j)$, $1 \leq i \leq M$, $1 \leq j \leq N$, for some positive M and N. Lagrangian grids are typically not smooth because nodes of the grid are moving with the fluid. In 1992, Morel, Dendy, Hall, and White in [22] give a scheme that is very robust and which they claim is better than all other known schemes compatible with Lagrangian hydrodynamics codes. One *main result* of this paper is that the performance of the algorithm presented here is as good as the Morel algorithm on smooth grids, and it is superior on rough grids. In addition, the new algorithm performs exceptionally well on problems with discontinuous coefficients. The new algorithm has some additional advantages. For example, the linear equations that appear in the algorithm always have a symmetric and positive definite coefficient matrix.

1.1. Discussion and Background

The finite-difference schemes are constructed using the *support-operators* method (see [10, 30, 31]), which is a method for constructing discrete analogs of invariant first-order differential operators like the divergence and gradient. The main ideas behind this method is to require that the operators satisfy some discrete analogs of the well-known integral identities that relate the differential operators to their adjoints. The integral identities are easily derived from the divergence theorem and are closely related to conservation laws for the initial boundary-value problem given by (1.1) and (1.3). We note that the terminology *support-operators method* is not a particularly good translation of the original Russian; perhaps, *basic* or *reference* operators would be better.

It is important to note that all of the discrete operators are invariantly defined; that is, the definitions contain only coordinate invariant quantities like volumes, areas, and angles. This means that the discrete operators can be used in any coordinate system by simply changing the formulas for these geometric quantities.

Once the basic setup for the support-operators method has been established, it is possible to use discrete analogs of continuum operator identities to show that the discretized problem shares many important properties of the original continuum problem and, because of this, the difference scheme is called *mimetic* [12, 30]. For example, the difference scheme is *conservative,* and the discrete analog of $-\nabla\cdot\mathbf{K}\,\nabla$ is symmetric and positive definite. Another important property is that the discrete divergence is equal to zero on constant vectors and the discrete flux operator is zero on a function if and only if the function is constant. Moreover, the gradient operator applied to a linear function is a constant vector, which implies that the discrete Laplacian is exact on linear functions. The fact that only constants are in the null space of the flux operator is particularly important because, for schemes that do not satisfy this property, the highest-frequency mode on the grid is typically in the null space of the discrete flux operator, and then a noise filter is required, as in Margolin [21].

To consider general boundary conditions, the divergence operator is extended to the boundary as the operator which gives the normal component of a vector. Now, if the definition of the inner product for functions is extended to include a boundary integral of the functions, then the approximation of the boundary conditions is made consistent with the approximation in the interior. This leads to the finite-difference scheme, where the discrete boundary conditions do not destroy the symmetry and positive definiteness of the discrete analog of the operator $-\nabla\cdot\mathbf{K}\,\nabla$.

The basic discretization uses cell-centered values for the solution u and the parameters in the differential equation: f, a, and $\mathbf{K}$, which is consistent with standard Lagrangian codes. The fluxes are discretized using components which are projections onto the normals to the cell sides and are located at the centers of the cell sides. These components are chosen because they are continuous on the interface between two different materials, and as usual in Lagrangian gas dynamics, it is assumed that the internal boundary between different materials consists of cell sides.

The discretization of the fluxes leads to a so-called *compact* approximation of the equation for the flux (1.2) (see, for example, [17]). Usually, compact finite-difference schemes are used to obtain high-order approximations to differential operators. Here the compact differencing is critical for obtaining the mimetic properties. As is usual for compact differencing, the discrete analog of the gradient will not be *local*. That is, the matrix corresponding to the discrete gradient is not banded. However, this matrix is the inverse of a banded matrix times a banded matrix, just like for standard compact differences. In the case of an orthogonal grid, the discrete gradient becomes a local operator.

There are two good solution strategies for the discrete equations. A discrete analog of the flux form of the diffusion equation (1.2) will be a system of two equations for two unknowns, the discrete temperature and the discrete fluxes. Now eliminate one of the unknowns between the two equations. If the fluxes are eliminated, then a discretized diffusion equation for the temperature is obtained, while if the temperature is eliminated, a discrete equation for the fluxes is obtained. For both approaches, the support-operators method produces a system of linear equations determined by a symmetric positive-definite operator.

For the temperature-based algorithm, the operator for the linear system is nonlocal, but can be represented as a product of a local operator and the inverse of a local operator, so it is possible to efficiently compute the product of a nonlocal operator times a vector. This can be used as the basis of an iterative method of solving for the discretized temperature. There are many iterative methods for solving the discrete equations that only use the product of the matrix times a vector. For example, two-level gradient methods and three-level conjugate-direction methods [27]. We have used the incomplete Cholesky conjugate gradient method from the package NSPCG [23] (also see [14, 22]). The effectiveness of this approach depends strongly on the choice of a preconditioner, and there are some natural choices, such as an operator that corresponds to a finite-difference scheme with a local gradient. We do not have enough data and space to present a comparison of different iterative methods, so this is left to a future paper.

For the flux-based algorithm, the equations can be reformulated so that the operator that must be inverted is symmetric, positive definite, and banded, so there are many effective methods for this form of the algorithm. Once the fluxes are computed, there is an explicit formula for the temperature in terms of the fluxes. There is only one obvious disadvantage to this procedure; in 2D there are approximately twice as many unknowns than there are for more standard methods. However, the nice properties of the matrices seem to easily compensate for this disadvantage. The structure of our operator is close to one considered in [22], and one can use modification of the multigrid method described in [22] to invert our matrix. We use the flux-based method with a block Gauss–Seidel iteration method to compute the examples in this paper.

In many applications, the flux, and not the temperature, is needed. For example, to trace contaminants in porous media flow, the Darcy velocities, which are just the fluxes, are what is needed. Moreover, the transport of the contaminants is sensitive to the accuracy of the fluxes, so it is important to compute them accurately. In the case of smooth grids and conductivity, there does not seem to be any overall advantage to the flux-based method. However, in the case of rough grids or rough conductivity, the flux-based methods produce more accurate fluxes [6].

The nicest grids used in this study are generated using smooth mappings, so all of the cells are convex and look like mildly distorted rectangles. The nicest rough grids used are generated from a simple uniform grid by randomly moving the node by a small amount, but so that all the cells are still convex. The next worse grids contain a few too many nonconvex cells. No grids with self-intersecting cells are used, but if the *signed* area of the cells are all positive, the method will still work. The method is very robust for grids with convex cells, no matter how rough the grids. For grids with nonconvex cells, the only difficulty is that the iterative matrix solver may not converge. A modified algorithm that is less accurate but more robust is introduced to compensate for this problem. In fact, the modified algorithm should be robust even for grids with slightly self-intersecting cells. The coefficient matrix in the modified algorithm is always symmetric and positive-definite, so that the convergence of iterative methods is ensured. For convex cells, the modified scheme coincides with the original scheme, so there is no need for special procedures to detect nonconvex cells during a computation.

The discontinuities in the conductivity k, which are also called interfaces, that are considered are simple jumps along straight lines. The theory for interfaces only implies that the normal flux at such interfaces is continuous. It is to be expected that the tangential flux is not continuous. All of the examples with an analytical solution that we know of in the literature have both the normal and tangential flux continuous at any discontinuity and, thus, do not rigorously test algorithms for problems with discontinuous conductivity. Consequently, we introduce a simple but important example due to J. Morel that has a solution with discontinuous tangential flux. Any algorithm that is not accurate for this example is inadequate for our applications.

The truncation errors for the divergence and flux operator depend strongly on the smoothness of the grid. So, for the study of truncation errors, minimal smoothness conditions similar to the conditions used in the theory of finite elements, are placed on the grids. In addition, the approximate solution may be compared to either the point values or the cell-averaged values of the exact solution, that is, the *integral average* and *point* projections of the exact solution. For the integral-average projection, the divergence is exact. For the point projection, we show that for grids with minimal smoothness the truncation error for the divergence is first order in the max norm and the truncation error for the flux operator is also first order in a discrete vector L_2 norm. On a smooth grid, both operators have second-order truncation error in the interior of the region in the max norm. In addition, the discrete gradient is *exact* for linear functions for *any* grid, and discrete divergence is exact for constant vectors. All of the theoretical results were checked numerically. In fact, the theoretical results in this paper hold for general symmetric conductivity matrices [13].

An important property of the continuum problem is that its solutions satisfy a maximum principal.We do not know of an algorithm which simultaneously preserves the symmetry of the Laplacian, is exact for linear functions, and satisfies the maximum principle on general grids, even for smooth k. As expected, the support-operators method does not have a maximum principal, but the violation of the maximum principal is small and localized (see [32]).

Recall that when the conductivity matrix is the identity, the flux operator is the gradient and the diffusion operator is the Laplacian. Here is a summary of the main properties of the flux-based support-operators finite-difference scheme:

- The scheme is conservative.
- The finite-difference scheme is second-order accurate.
- The matrices that must be inverted to solve the discrete problem are symmetric and positive definite for Dirichlet, Neumann, and Robin boundary conditions for any grid.
- Material discontinuities are treated rigorously.
- All discrete operators are linear.
- The discrete flux operator is the negative adjoint of the discrete divergence.
- The discrete diffusion operator is the composition of the discrete divergence and flux operators.
- The discrete diffusion operator is symmetric and positive definite.
- The discrete divergence of a constant vector is exactly zero on any grid.
- The discrete flux operator is exact on linear functions for any grid.
- The null space of the discrete flux operator is the constant functions.
- On rectangular grids, all discrete operators reduce to a standard differencing.
- On rectangular grids, the treatment of discontinuous coefficients is equivalent to the usual harmonic averaging.
- The construction can be used for the case of nondiagonal symmetric positive-definite matrices with discontinuous elements.

To our knowledge, no other scheme has been proposed with all of these properties.

The scheme which gives numerical results closest to the support-operators scheme was proposed by Morel, Dendy, Hall, and White in [22], where the authors claim that their scheme is the best of all known schemes for the problem class they are considering. The main idea used in their scheme is to introduce additional discrete values for u on the cell edges and then to use these values to write down additional equations which express the continuity of the fluxes on the edges. The authors of [22] mentioned some main disadvantages of their scheme:

- There are cell-edge unknowns in addition to the cell-centered unknowns.
- The matrix to be inverted is not symmetric.
- The matrix to be inverted can become ill-conditioned if the mesh is sufficiently distorted.

The last disadvantage can be fixed using positive weights as in this paper, but then their scheme is not exact for linear functions. It is important to note that they may need this modification even when the cells are convex. Distorted cells must be detected and treated specially. In our opinion, only the last two items in this list are real disadvantages, and they are not present in the support-operators method.

There are many other methods of constructing approximations for diffusion equations, a few of which we mention here. Many schemes are based on the mapping method, where the original equations are transformed to a general curvilinear coordinate system, and then the resulting equations are approximated on a rectangular grid in curvilinear coordinates. For example, see Kershaw [14], Pert [25], Robertson [26], or Shashkov and Steinberg [30]. Some of these ideas were also used by Morel *et al.* [22]. Usually such schemes are only satisfactory for smooth grids.

In a recently published paper [1] by van Beek, van Nooyen, and Wesseling, the authors describe an algorithm for discretizing the gradient on nonsmooth grids and mentioned that their aim is to provide a theoretical derivation for the best scheme given in Bernard and Kapitza [2]. The main assumption made in [1] is that flux vector $\mathbf{K}\,\nabla u$ is continuous; that is, both the tangential and normal components of flux are continuous at interfaces. Unfortunately, for the problems which interest us, this continuity assumption is not valid. In Section 6, we present an example with discontinuous tangential flux, where the method from [1] does not converge. We note that for many applications the method from [1] works well (see [37, 39]). In particular, for all of the test problems presented in Section 6, where the flux vector is continuous, the method from [1] works very well, even for very distorted grids. Note that the expression for the flux operator obtained in [1] is local and is exact for linear functions. However, the discrete divergence is not the negative adjoint of the discrete gradient, and consequently the discrete Laplacian is not symmetric and in principle may not be positive-definite for very distorted grids.

Another interesting approach, which is not related to the mapping idea, is presented by Margolin and Adams [20] and Margolin and Tarwater [21], where the authors try to construct *local* discrete operators analogous to the divergence and gradient, which are negative adjoints of each other and the gradient is exact on linear functions. They show that, in general, this is not possible. The authors of [21] use an original idea involving the use of some nonlinear conditions related to the direction of the gradient, which makes it possible to find a discrete gradient with the required properties. However, this means that the discrete gradient depends weakly on the scalar field as well as the geometry of the mesh. Also, they compute the gradient at the nodes, which leads to spurious modes in the null space of the discrete gradient and, consequently, the need to use a special filtering procedure to remove these modes from the solution. Also, it is not clear how well this algorithm will work for the case of discontinuous coefficients.

A finite-element scheme using a nodal discretization of the scalar field is given by Shestakov, Harte, and Kershaw [33]. As mentioned in [22], finite-element methods do not require a smooth grid, but the nodal discretization can cause considerable difficulty in coupling such schemes with standard Lagrangian hydrodynamic codes. For a general discussion of using the finite-element method in the framework of Lagrangian methods, we refer the reader to [22].

This paper is arranged as follows. In Section 2, the diffusion equation is discretized in time but not in space, and then this problem is written in terms of abstract operators corresponding to the divergence combined with the normal derivative boundary operator and the flux operator. The integral and symmetry properties of these operators illuminate the properties that must be preserved when the finite-difference scheme is constructed. The grid, the discretization of scalar and vector functions, and the inner products and related spaces of discrete functions are introduced in Section 3. In Section 4, these structures and the support-operators method are used to construct the discrete divergence and a discrete flux operator. Both scalar and flux-based algorithms for solving the diffusing equations are described. The operators are written out in detail for a rectangular grid.

The theoretical properties of the support-operators algorithm are discussed in Section 5. The truncation error is investigated for the discrete divergence and flux operators, for both smooth and general grids. In addition, the null space of the discrete flux operator is investigated and a relationship between the null space of the flux operator and conservation is presented. Finally, we show how to make the algorithm robust when there are nonconvex cells in the grid. In Section 6, the flux-based algorithm is used to compute some examples. The beginning examples study the properties of the algorithm on rough grids, while the latter look at the algorithm for discontinuous conductivity. In Section 7 possible extensions are described.

2. PROPERTIES OF THE CONTINUUM PROBLEM

The support-operators method requires that the fully discretized problem mimic the semidiscretized problem, the problem where time, but not space, has been discretized. The analysis will be performed for the fully implicit time discretization, but the ideas extend easily to other time discretizations. The case of Dirichlet boundary conditions must be analyzed separately, but is easy, so the following discussion is restricted to the case where β in (1.3) is not zero at any point on the boundary, in which case we choose $\beta = -1$ and then it is assumed that $\alpha \geq 0$. The first task is to write the diffusion equation and the flux form of the diffusion equation in terms of abstract operators on inner product spaces and then to enumerate the important properties of the abstract operators. As long as it does not cause any additional work, the case of general symmetric conductivity matrix $\mathbf{K}$ will be considered. However, the main applications are to the case of a diagonal matrix $\mathbf{K} = k\mathbf{I}$.

2.1. The Abstract Operators

The fully implicit semi-discretization of the diffusion equation (1.1) and the boundary conditions (1.3) gives

$$\begin{aligned} a\frac{u^{n+1}-u^n}{\Delta t} - \nabla\cdot\mathbf{K}\,\nabla u^{n+1} &= f, \quad \text{on } V, \\ (\mathbf{K}\,\nabla u^{n+1}, \mathbf{n}) + \alpha u^{n+1} &= \psi, \quad \text{on } \partial V, \end{aligned} \tag{2.1}$$

where the index n corresponds to time level $t_n = n\,\Delta t$ and $u^n = u(t^n, x, y)$. It is assumed that a, $\mathbf{K}$, f, α, and ψ are functions of coordinates and time, but because the main goal is to study the spatial discretization, the time index on these functions is suppressed. Equation (2.1) can be written in the operator form

$$\mathbf{A}u^{n+1} = F^{n+1}, \tag{2.2}$$

where the operator $\mathbf{A}$ is given by

$$\mathbf{A}u = \begin{cases} au/\Delta t - \nabla\cdot\mathbf{K}\,\nabla u, & \text{on } V, \\ (\mathbf{K}\,\nabla u, \mathbf{n}) + \alpha u, & \text{on } \partial V, \end{cases} \tag{2.3}$$

and where the right-hand side of the previous has the form

$$F^{n+1} = \begin{cases} f + au^n/\Delta t, & \text{on } V, \\ \psi, & \text{on } \partial V. \end{cases} \tag{2.4}$$

The flux form of the diffusion equation (1.2) and the boundary conditions (1.4) can be equivalently discretized in time as

$$\begin{aligned} \mathbf{w}^{n+1} = &\quad -\mathbf{K}\,\nabla u^{n+1}, && \text{on } V, \\ a(u^{n+1} - u^n)/\Delta t + \nabla\cdot\mathbf{w}^{n+1} = &\quad f^{n+1}, && \text{on } V, \\ -(\mathbf{w}^{n+1}, \mathbf{n}) + \alpha u^{n+1} = &\quad \psi, && \text{on } \partial V. \end{aligned} \tag{2.5}$$

This can be written in the operator form

$$\mathbf{w}^{n+1} - \mathbf{G}u^{n+1} = 0, \quad \mathbf{D}\mathbf{w}^{n+1} + \mathbf{\Omega}u^{n+1} = F^{n+1}, \tag{2.6}$$

where the operators $\mathbf{G}$, $\mathbf{D}$, and $\mathbf{\Omega}$ are defined by

$$\mathbf{G}u = -\mathbf{K}\,\nabla u, \qquad \text{on } V, \tag{2.7}$$

$$\mathbf{D}\mathbf{w} = \begin{cases} +\nabla\cdot\mathbf{w}, & \text{on } V, \\ -(\mathbf{w}, \mathbf{n}), & \text{on } \partial V, \end{cases} \tag{2.8}$$

$$\mathbf{\Omega}u = \begin{cases} (a/\Delta t)u, & \text{on } V, \\ \alpha u, & \text{on } \partial V. \end{cases} \tag{2.9}$$

Eliminating the flux from (2.6) shows that

$$\mathbf{A} = \mathbf{\Omega} + \mathbf{D}\mathbf{G}. \tag{2.10}$$

The inner-product space (Hilbert space) H is the closure of scalar functions u that are smooth on the closure of V using the inner product

$$(u, v)_H = \int_V uv\, dV + \oint_{\partial V} uv\, dS, \quad u, v \in H, \tag{2.11}$$

while the inner-product space $\mathbf{H}$ is the closure of the vector functions $\mathbf{w}$ that are smooth on the closure of V using the inner product

$$(\mathbf{A}, \mathbf{B})_{\mathbf{H}} = \int_V (\mathbf{K}^{-1}\mathbf{A}, \mathbf{B})\, dV, \quad \mathbf{A}, \mathbf{B} \in \mathbf{H}. \tag{2.12}$$

Because matrix $\mathbf{K}$ is symmetric and uniformly bounded above and below, then so is $\mathbf{K}^{-1}$. This implies that $(\cdot,\cdot)$ is, in fact, an inner product, even if $\mathbf{K}$ is not continuous. Such weighted inner products also naturally appear in mixed finite-element formulations (see, for example, [7, 38]). Now the domains and ranges of the abstract operators can be given precisely:

$$\mathbf{G}: H \to \mathbf{H}, \quad \mathbf{D}: \mathbf{H} \to H, \quad \mathbf{\Omega}: H \to H, \quad \mathbf{A}: H \to H. \tag{2.13}$$

2.2. Properties of the Abstract Operators

The support-operators method requires that we choose a prime operator, a derived operator, an integral identity connecting the prime and derived operator, a discretization of the prime operator, and discrete analogs of the above inner products. The divergence $\nabla\cdot$ is chosen as the prime operator and it will be discretized using an analog of the divergence theorem

$$\int_V \nabla\cdot\mathbf{w}\, dV = \oint_{\partial V} (\mathbf{w}, \mathbf{n})\, dS. \tag{2.14}$$

Previously, the gradient was always chosen as the derived operator, but we now choose it to be the flux operator $\mathbf{K}\,\nabla$. The required integral identity is Gauss' theorem written in the form

$$\int_V u\nabla\cdot\mathbf{w}\, dV + \int_V (\mathbf{w}, \mathbf{K}^{-1}\mathbf{K}\,\nabla u)\, dV = \oint_S u(\mathbf{w}, \mathbf{n})\, dS. \tag{2.15}$$

The definition of the operator $\mathbf{D}$, the formulas (2.11) and (2.12) for the inner product, and integral identity (2.15), which connects $\nabla\cdot$ and ∇, give $(\mathbf{D}\mathbf{w}, u)_H = (\mathbf{w}, \mathbf{G}u)_{\mathbf{H}}$ which implies that $\mathbf{G} = \mathbf{D}^*$. Also, the divergence property (2.14) is equivalent to $(\mathbf{D}\mathbf{w}, 1)_H = 0$, where 1 is the constant function with value 1. Because it is assumed that $a > 0$ and $\alpha \geq 0$, it follows that $\mathbf{\Omega} = \mathbf{\Omega}^* \geq 0$. If it is assumed that $\alpha > 0$, then $\mathbf{\Omega} > 0$. In summary,

$$(\mathbf{D}\mathbf{w}, 1)_H = 0, \tag{2.16}$$

and if $\alpha > 0$, then

$$\mathbf{G} = \mathbf{D}^*, \quad \mathbf{\Omega} = \mathbf{\Omega}^* > 0, \tag{2.17}$$

and because

$$\mathbf{A} = \mathbf{\Omega} + \mathbf{D}\mathbf{G} = \mathbf{\Omega} + \mathbf{G}^*\mathbf{G}, \tag{2.18}$$

it follows that

$$\mathbf{A} = \mathbf{A}^* > 0. \tag{2.19}$$

These are the important properties of the continuum operators that the fully discretized operators should mimic. Because the properties of $\mathbf{A}$ follow from the properties of $\mathbf{\Omega}$, $\mathbf{D}$, and $\mathbf{G}$ and the definitions of the inner products, the goal of the support-operators method is to build discrete analogs of the operators $\mathbf{\Omega}$, $\mathbf{D}$, and $\mathbf{G}$ and the inner products that satisfy an analog of (2.17) and (2.16).

2.3. Forms of the Semidiscretized Diffusion Equation

The semidiscretized diffusion equation can be written in three useful forms: as a system of first-order equations, as a second-order equation for the temperature; or as a second-order system for the fluxes. The system for the fluxes is very useful for computations. The first-order system (2.6) is

$$\mathbf{w}^{n+1} - \mathbf{G}u^{n+1} = 0, \quad \mathbf{D}\mathbf{w}^{n+1} + \mathbf{\Omega}u^{n+1} = F^{n+1}. \qquad (2.20)$$

If the flux $\mathbf{w}$ is eliminated from these two equations, then the abstract form of the diffusion equation (1.1) is obtained:

$$\mathbf{A}u^{n+1} = (\mathbf{\Omega} + \mathbf{DG})u^{n+1} = F^{n+1}. \qquad (2.21)$$

On the other hand, if u is eliminated, then an equation for the flux at the new time level is obtained:

$$\mathbf{B}\mathbf{w}^{n+1} = (\mathbf{I} + \mathbf{G\Omega}^{-1}\mathbf{D})\mathbf{w}^{n+1} = \mathbf{G\Omega}^{-1}F^{n+1}. \qquad (2.22)$$

The results in the previous section imply that $\mathbf{A} = \mathbf{A}^* > 0$ and $\mathbf{B} = \mathbf{B}^* > 0$ so both the equation for u and the one for $\mathbf{w}$ have nice properties. It is important to note that the boundary conditions are included in the definitions of the operators and the spaces of functions in a natural way. In addition, the conservation law for u follows from the second equation in (2.20), because

$$(\mathbf{D}\mathbf{w}^{n+1}, 1)_H + (\mathbf{\Omega}u^{n+1}, 1)_H = (F^{n+1}, 1)_H, \qquad (2.23)$$

but then the divergence property (2.16) gives

$$(\mathbf{\Omega}u^{n+1}, 1)_H = (F^{n+1}, 1)_H, \qquad (2.24)$$

and then the definition of F implies that

$$\begin{aligned}(\mathbf{\Omega}u^{n+1}, 1)_H - (\mathbf{\Omega}u^{n}, 1)_H = \int_V f\,dV& \\ + \int_{\partial V} (\psi - \alpha u^n)\,dS,&\end{aligned} \qquad (2.25)$$

which is the equivalent to the standard conservation law. In particular, if there are no sources in the problem, that is, $f = 0$, $\psi = 0$, and $\alpha = 0$; then $(\mathbf{\Omega}u^{n+1}, 1)_H = (\mathbf{\Omega}u^{n}, 1)_H$, which is the standard conservation law.

3. THE GRID AND SPACES OF DISCRETE FUNCTIONS

This section describes the grids used in this paper, along with the discretization of scalar and vector functions. Scalars are discretized at cell centers, while the components of the discretized vectors are located at the center of the cell sides and are the components in the direction normal to the cell side. Also the spaces of discrete scalar and vector functions and their inner-products are introduced. The important explicit formulas for the inner-products are given in a form useful for characterizing the matrices generated by the discretization.

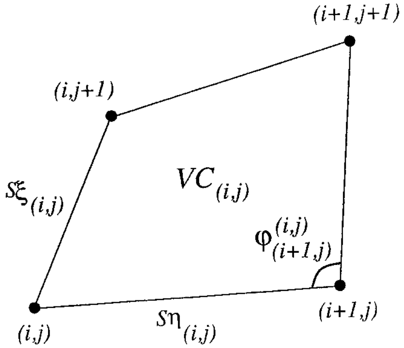


FIG. 1. Typical cell of a logically rectangular grid.

It is straight forward, but requires a lengthy exposition, to extend the ideas in this paper to three-dimensional grids. However, because the two-dimensional geometry is a simplified model of the three-dimensional geometry, most of the terminology used in the discretization come from the three-dimensional setting. For example, the area of a 2D cell corresponds to the volume of 3D cell, the length of a side corresponds to the area of a face, and the center of an edge of a cell corresponds to the center of a cell face, which explains some of the notation in the cell in Fig. 1. A logically rectangular grid can be interpreted as a grid which is formed by the intersections of the coordinate lines of some curvilinear coordinate system $x = x(\xi, \eta)$, $y = y(\xi, \eta)$, where ξ and η are the curvilinear coordinates. Then the coordinate lines are given by ξ varying while η is constant, and η varying while ξ is fixed. In three dimensions, with curvilinear coordinates ξ, η, and ζ, one coordinate surface is given by ξ constant while η and ζ vary, so this surface is best labeled by ξ. When this setup is projected into two dimensions, this surface becomes the curve with ξ constant and η varying, so this curve is labeled with ξ, which explains more of the notation in Fig. 1. An extensive discussion of these points can be found in Chapter 2 of [15], by Knupp and Steinberg.

3.1. The Grid and the Discretization of Scalars and Vectors

By definition, a logically rectangular grid can be indexed in exactly the same way as a rectangular grid. That is, if M and N are positive integers, then the (i, j) node of the grid is $(x(i, j), y(i, j))$, $1 \le i \le M$, $1 \le j \le N$. The quadrilateral 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. 1). The area of this cell is denoted by $VC_{(i,j)}$ (V is for volume in 3D).

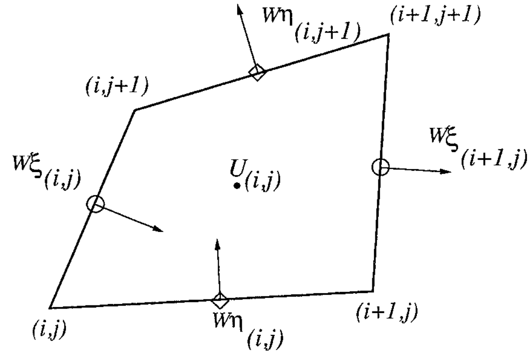


FIG. 2. Discretization of scalars and vectors.

The length of the side of the (i, j) cell that connects the vertices (i, j) and $(i, j + 1)$ is denoted $S\xi_{(i,j)}$ (this is the curve with ξ constant), 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 $\varphi^{(i,j)}_{k,l}$ (the angle $\varphi^{(i,j)}_{(i+1,j)}$ is displayed in Fig. 1).

The methods described in this paper have provably good properties and work well for general convex cells. In Section 6, grids with nonconvex cells and even cells with self-intersecting sides are considered. Naturally, as the types of cells allowed become more general, the algorithm ceases to have some important properties, but it can still produce good results.

To study truncation errors or convergence rates, some mild smoothness assumptions are placed on the grid. If

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

then h is used as a small parameter which characterizes the density of the grid. It is assumed that there exists constants $C^{(i)}_{\max}$ and $C^{(i)}_{\min}$, $i = 1, 2$, and a constant $\delta > 0$, which do not depend on h, so that

$$\begin{gathered} C^1_{\min} h^2 \le VC_{(i,j)} \le C^1_{\max} h^2, \\ C^2_{\min} h \le S\xi_{(i,j)}, S\eta_{(i,j)} \le C^2_{\max} h, \\ \sin\left(\varphi^{(i,j)}_{(k,l)}\right) > \delta. \end{gathered} \tag{3.2}$$

The discrete analog of scalar function u is a cell-centered discrete scalar function $U_{(i,j)}$ (see Fig. 2), where the indices vary in the same range as for the cells $VC_{(i,j)}$. The treatment of the boundary conditions requires the introduction of the values of the scalar function on the centers of the boundary segments, that is, $U_{(0,j)}$, $U_{(M,j)}$ for $j = 1, ..., N - 1$ and $U_{(i,0)}$, $U_{(i,N)}$ for $i = 1, ..., M - 1$. Also, the scalar parameter a and the matrix parameter $\mathbf{K}$ appearing in the diffusion equation are discretized in the same way as u, except their values on the boundary are not needed. The scalar functions α and ψ that appear in the boundary condition use the same discretization as is used for u on the boundary.

Vector functions are discretized using components which are the orthogonal projections on the directions perpendicular to the sides of the cells (see Fig. 2),

$$\begin{aligned} &W\xi_{(i,j)}: i = 1, ..., M; j = 1, ..., N - 1, \\ &W\eta_{(i,j)}: i = 1, ..., M - 1; j = 1, ..., N, \end{aligned} \tag{3.3}$$

where $W\xi_{(i,j)}$ is the component at the center of side with vertices (i, j), $(i + 1, j)$ and $W\eta_{(i,j)}$ is the component at the side with vertices (i, j), $(i, j + 1)$. Now the 3D analog for cells with planar faces helps clarify what is going on; vectors in 3D are discretized using components that are normal to the faces of the 3D cells and located at the center of the face. Therefore, if a face is given by ξ constant, then the normal to this face is labeled with $W\xi$.

3.2. The Natural Inner Products

Now, to proceed with the support-operators method, the discrete scalar and vector functions must have an inner product which is called the *natural* inner product. The space of discrete scalar functions is labeled HC, and a natural analog for the continuum inner product (2.11) is

$$\begin{aligned} (U, V)_{HC} = &\sum_{i=1}^{M-1}\sum_{j=1}^{N-1} U_{(i,j)} V_{(i,j)} VC_{(i,j)} \\ &+ \sum_{i=1}^{M-1} U_{(i,0)} V_{(i,0)} S\eta_{(i,1)} \\ &+ \sum_{j=1}^{N-1} U_{(N,j)} V_{(N,j)} S\xi_{(N,j)} \\ &+ \sum_{i=1}^{M-1} U_{(i,M)} V_{(i,M)} S\eta_{(i,M)} \\ &+ \sum_{j=1}^{N-1} U_{(0,j)} V_{(0,j)} S\xi_{(1,j)}. \end{aligned} \tag{3.4}$$

It is clear that the inner product is symmetric, $(U, V)_{HC} = (V, U)_{HC}$, and that if all volumes $VC_{(i,j)}$ and lengths $S\xi_{(i,j)}$, $S\eta_{(i,j)}$ are positive, then $(U, U)_{HC} \ge 0$ and $(U, U)_{HC} = 0$ if and only if U ⌂ 0, so the inner product is properly defined.

The space of vector functions is labeled $\mathcal{HS}$, and a *natural* inner product, which is the analog of the continuum inner product (2.12) is

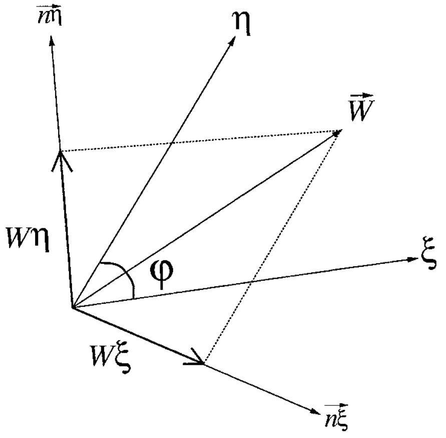


FIG. 3. Components of a vector in the local basis.

$$(\mathbf{A},\mathbf{B})_{\mathcal{HS}} = \sum_{i=1}^{M-1}\sum_{j=1}^{N-1} \frac{(\mathbf{A},\mathbf{B})_{(i,j)}}{k_{(i,j)}} VC_{(i,j)}, \tag{3.5}$$

where $(\cdot,\cdot)_{(i,j)}$ is an inner product associated with a cell and $k_{(i,j)}$ is the value of $k(x, y)$ at the center of the cell (i, j). This inner product will be proper, provided that $k_{(i,j)} > 0$ and $VC_{(i,j)} > 0$, which are assumed, and if the cell inner product is properly defined. Note that we have used, in a critical way, the assumption that $\mathbf{K}$ is diagonal, if it is not, then the expression for the inner product is much more complicated.

As the components of vectors are not located at the same point (not co-located), the definition of the inner product is a bit involved. To define the inner product of two vectors in a cell, first a formula for the inner product of two vectors with co-located components is given in terms of their components perpendicular to the cell sides (see Fig. 3). Thus, suppose that the ξ and η axes form a non-orthogonal basis system and that φ is the angle between these axes. If the unit normals to the axes are $\mathbf{n}\boldsymbol{\xi}$ and $\mathbf{n}\boldsymbol{\eta}$, then the components of the vector $\mathbf{W}$ in this basis are the orthogonal projections $W\xi$ and $W\eta$ of $\mathbf{W}$ onto these normal vectors. Now, some simple vector algebra shows that if $\mathbf{A} = (A\xi, A\eta)$ and $\mathbf{B} = (B\xi, B\eta)$, then the expression for their inner product is

$$(\mathbf{A},\mathbf{B}) = \frac{A\xi\, B\xi + A\eta\, B\eta + (A\xi\, B\eta + A\eta\, B\xi)\cos(\varphi)}{\sin^2(\varphi)} \tag{3.6}$$

Next define $(\mathbf{A},\mathbf{B})_{(i,j)}^{(k,l)}$, where each index (k, l) corresponds to one of the vertices of cell (i, j), by making the following replacements in (3.6): $A\xi \to A\xi_{(i+k,j)}$; $B\xi \to B\xi_{(i+k,j)}$; $A\eta \to A\eta_{(i,j+l)}$; $B\eta \to B\eta_{(i,j+l)}$; $\varphi \to \varphi_{(i+k,j+l)}^{(i,j)}$. The *natural* cell inner product is a symmetric analog of (3.6),

$$(\mathbf{A},\mathbf{B})_{(i,j)} = \sum_{k,l=0}^{1} V_{(i+k,j+l)}^{(i,j)} (\mathbf{A},\mathbf{B})_{(i,j)}^{(k,l)}, \tag{3.7}$$

where the $V_{(i+k,j+l)}^{(i,j)}$ are some weights for which $\sum_{k,l=0}^{1} V_{(i+k,j+l)}^{(i,j)} = 1$ where the notation for weights is the same as for the angles of a cell. It is shown in [32] that to obtain a first-order approximation for the gradient operator it is necessary that the weights $V_{(i+k,j+l)}^{(i,j)}$ are one half of the area of the triangle in cell (i, j) which contains the angle at node $(i + k, j + l)$, divided by the volume of the cell. Using the cosine theorem, it is easy to prove that expression (3.7) satisfies the properties of an inner product and, in particular, $(\mathbf{A}, \mathbf{A})_{(i,j)} > 0$ if $\mathbf{A} \neq \mathbf{0}$.

3.3. The Formal Inner Products

For the computation of adjoint relationships and the entries of the matrices corresponding to the discretized operators, the *formal* inner products, $[\cdot,\cdot]$ in the spaces of scalar and vector functions are introduced. In space HC,

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

while in space $\mathcal{HS}$,

$$\begin{aligned}[\mathbf{A},\mathbf{B}]_{\mathcal{HS}} = {} & \sum_{i=1}^{M-1}\sum_{j=1}^{N-1} A\xi_{(i,j)}B\xi_{(i,j)} \\ & + \sum_{i=1}^{M-1}\sum_{j=1}^{N} A\eta_{(i,j)}B\eta_{(i,j)}.\end{aligned} \tag{3.9}$$

Then the relationships between natural inner products and the formal inner products have the form

$$(U,V)_{HC} = [\mathcal{M}U, V]_{HC}, \quad (\mathbf{A},\mathbf{B})_{\mathcal{HS}} = [\mathcal{S}\mathbf{A},\mathbf{B}]_{\mathcal{HS}}. \tag{3.10}$$

A comparison of the natural and formal inner products gives

$$\begin{aligned}(\mathcal{M}U)_{(i,j)} &= VC_{(i,j)}U_{(i,j)}, \\ i &= 1, ..., M-1; \\ j &= 1, ..., N-1,\end{aligned} \tag{3.11}$$

$$\begin{aligned}(\mathcal{M}U)_{(i,j)} &= S\xi_{(i,j)}U_{(i,j)}, \\ i &= 0; i = M; \\ j &= 1, ..., N-1,\end{aligned} \tag{3.12}$$

$$\begin{aligned}(\mathcal{M}U)_{(i,j)} &= S\eta_{(i,j)}U_{(i,j)}, \\ i &= 1, ..., M-1; \\ j &= 0; j = N.\end{aligned} \tag{3.13}$$

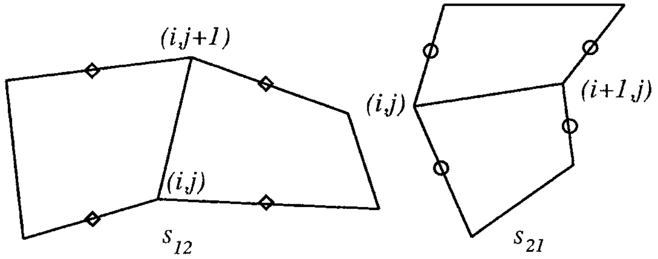


FIG. 4. Footprint of the operators S_{12} and S_{21}.

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

$$\mathcal{S} = \begin{pmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{pmatrix}. \tag{3.14}$$

Then a comparison of formal and natural inner products gives

$$\begin{aligned}
(S_{11}A\xi)_{(i,j)} &= \left(\sum_{k,l=0}^{1} \frac{1}{k_{(i-k,j)}} \cdot \frac{V^{(i-k,j)}_{(i,j+l)}}{\sin^2(\varphi^{(i-k,j)}_{(i,j+l)})} \right) A\xi_{(i,j)}, \\
(S_{12}A\eta)_{(i,j)} &= \sum_{k,l=0}^{1} \left(\frac{(-1)^{k+l}}{k_{(i-k,j)}} \cdot \frac{V^{(i-k,j)}_{(i,j+l)}}{\sin^2(\varphi^{(i-k,j)}_{(i,j+l)})} \cos(\varphi^{(i-k,j)}_{(i,j+l)}) \right) \\
&\quad \times A\eta_{(i-k,j+l)}, \qquad\qquad (3.15) \\
(S_{21}A\xi)_{(i,j)} &= \sum_{k,l=0}^{1} \left(\frac{(-1)^{k+l}}{k_{(i,j-l)}} \cdot \frac{V^{(i,j-l)}_{(i+k,j)}}{\sin^2(\varphi^{(i,j-l)}_{(i+k,j)})} \cos(\varphi^{(i,j-l)}_{(i+k,j)}) \right) \\
&\quad \times A\xi(i+k, j-l), \\
(S_{22}A\eta)_{(i,j)} &= \left(\sum_{k,l=0}^{1} \frac{1}{k_{(i,j-l)}} \cdot \frac{V^{(i,j-l)}_{(i+k,j)}}{\sin^2(\varphi^{(i,j-l)}_{(i+k,j)})} \right) A\eta_{(i,j)}.
\end{aligned}$$

Actually, these formulas are valid only for the interior nodes: $i = 2, ..., M - 2; j = 2; ..., N - 2$; but it is possible to prove that if fictitious nodes are introduced for $i = 0$; $i = M + 1; j = 0; j = N + 1$, then these formulas are valid for $i = 1, ..., M - 1; j = 1, ..., N - 1$. The operators S_{11} and S_{22} are diagonal and the footprint (points where the coefficients of the stencil of the finite-difference scheme are not zero) for the operators S_{12} and S_{21} are shown in Fig. 4.

It is critical for the support-operators method that the operators $\mathcal{M}$ and $\mathcal{S}$ are symmetric and positive definite. For the algorithms introduced in this paper, this is always true for convex cells. Using standard arguments for estimating inner products, it can be shown that there exists constants Δ_i and δ_i, $i = 1, 2$, such that the formal inner products satisfy

$$\begin{aligned}
\Delta_1[U, V]_{HC} \geq [\mathcal{M}U, V]_{HC} \geq \delta_1[U, V]_{HC}, \\
\Delta_2[\mathbf{A}, \mathbf{B}]_{\mathcal{HS}} \geq [\mathcal{S}\mathbf{A}, \mathbf{B}]_{\mathcal{HS}} \geq \delta_2[\mathbf{A}, \mathbf{B}]_{\mathcal{HS}},
\end{aligned} \tag{3.16}$$

where the constants depend on the upper and lower bounds for k and the geometric properties of the cell. The formulas for the constants confirm that if all cells are convex, then the inner products are positive definite. For highly distorted nonconvex cells, it is easy to construct discrete scalar or vector fields for which the inner products will have negative values. Thus a modification of the definition of the inner product which ensure that these operators are positive in the case of nonconvex grids is presented in Section 5.

4. FINITE-DIFFERENCE SCHEME

Following the method of support-operators, the divergence is chosen as the prime operator and then discretized using an analog of the divergence property (2.14). Next, the discretization of the flux operator is derived. These discrete operators are then used to discretize the various forms of the diffusion equation, and then some properties of these discretizations are derived. All of the objects studied in this section are particularly simple on rectangular grids, so in the final subsection they are all written out in this context.

4.1. The Prime Operator or the Divergence

The analog of the abstract operator, **D**, which is defined in (2.8) and is the divergence in the interior of the region, and the normal component of a vector on the boundary, is labeled $\mathcal{D}$, and is the *prime* discrete operator. On the interior of the region, it is

$$\begin{aligned}
(\mathcal{D}\mathbf{W})_{(i,j)} &= (\mathbf{DIV}\ \mathbf{W})_{(i,j)} \\
&= \frac{1}{VC_{(i,j)}} \{(W\xi_{(i+1,j)}S\xi_{(i+1,j)} - W\xi_{(i,j)}S\xi_{(i,j)}) \\
&\quad + (W\eta_{(i,j+1)}S\eta_{(i,j+1)} - W\eta_{(i,j)}S\eta_{(i,j)})\},
\end{aligned} \tag{4.1}$$

while on the boundary of the region, it is

$$\begin{aligned}
(\mathcal{D}\mathbf{W})_{(i,0)} &= -W\eta_{(i,1)}, \quad i = 1, ..., M-1, \\
(\mathcal{D}\mathbf{W})_{(i,N)} &= +W\eta_{(i,N)}, \quad i = 1, ..., M-1, \\
(\mathcal{D}\mathbf{W})_{(0,j)} &= -W\xi_{(1,j)}, \quad j = 1, ..., N-1, \\
(\mathcal{D}\mathbf{W})_{(M,j)} &= +W\xi_{(M,j)}, \quad j = 1, ..., N-1.
\end{aligned} \tag{4.2}$$

Note that telescoping the sums gives

$$(\mathcal{D}\mathbf{W}, 1)_{HC} = 0 \tag{4.3}$$

which is called the divergence property of the discrete divergence $\mathscr{D}$ and is the direct analog of the divergence property (2.16) for the continuum operator.

4.2. The Derived or Flux Operator

The derived operator $\mathscr{G}$ is called the discrete flux operator and is the analog of the continuum flux operator $\mathbf{G} = -\mathbf{K}\,\nabla$ and is defined by $\mathscr{G} = \mathscr{D}^*$, where the adjoint is taken in the natural inner products. As will be seen, for nonorthogonal logically rectangular grids, this operator is not banded. However, this operator can be expressed as a product of a banded and an inverse of a banded matrix, a fact that is important for the numerical algorithms. Recall that $\mathscr{D}: \mathscr{HS} \to HC$ and that the definition of the adjoint gives

$$(\mathscr{D}\,\mathbf{W}, U)_{HC} = (\mathbf{W}, \mathscr{D}^* U)_{\mathscr{HS}}, \tag{4.4}$$

which can be translated to the formal inner products as

$$[\mathscr{D}\,\mathbf{W}, \mathscr{M} U]_{HC} = [\mathbf{W}, \mathscr{S}\mathscr{D}^* U]_{\mathscr{HS}}. \tag{4.5}$$

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

$$[\mathbf{W}, \mathscr{D}\dagger \mathscr{M} U]_{\mathscr{HS}} = [\mathbf{W}, \mathscr{S}\mathscr{D}^* U]_{\mathscr{HS}}. \tag{4.6}$$

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

$$\mathscr{D}\dagger\mathscr{M} = \mathscr{S}\mathscr{D}^*, \tag{4.7}$$

and then the discrete analog of operator $\mathbf{G}$ is given by

$$\mathscr{G} = \mathscr{D}^* = \mathscr{S}^{-1}\mathscr{D}\dagger\mathscr{M}. \tag{4.8}$$

Note that, in general, $\mathscr{S}$ is banded and consequently $\mathscr{S}^{-1}$ is not usually banded so that $\mathscr{G}$ is not usually banded. That is, $\mathscr{G}$ has a *nonlocal* stencil. Also, when the diffusion coefficient k is not constant, then it is not possible to write $\mathscr{G}$ as the product of a discrete diffusion coefficient times a discrete gradient.

Summation by parts applied to the left-hand side of (4.6) gives

$$-(\mathscr{D}\dagger\mathscr{M} U)_{(i,j)} = \begin{pmatrix} S\xi_{(i,j)}(U_{(i,j)} - U_{(i-1,j)}) \\ S\eta_{(i,j)}(U_{(i,j)} - U_{(i,j-1)}) \end{pmatrix}, \tag{4.9}$$

where $\mathscr{M}$ is defined in (3.10).

4.3. The Ω Operator

The discrtete form for $\mathbf{\Omega}$ (2.9) is

$$(\Omega U)_{(i,j)} = \begin{cases} (a_{(i,j)}/\Delta t)\, U_{(i,j)}, & \text{in } V, \\ \alpha_{(i,j)} U_{(i,j)}, & \text{on } \partial V, \end{cases}. \tag{4.10}$$

The operator Ω is symmetric and because Ω and $\mathscr{M}$ are diagonal, $\Omega\mathscr{M} = \mathscr{M}\Omega$. It is assumed that $a > 0$ and $\alpha \geq 0$, so $\Omega \geq 0$. In addition, if both a and α are bounded below by c and above by C, then Ω is bounded below and above:

$$C(U, V)_{HC} \geq (\Omega U, V)_{HC} \geq c(U, V)_{HC}. \tag{4.11}$$

4.4. Forms of the Fully Discretized Diffusion Equation

Now the tools are in place to look at the fully discretized forms of the semidiscretized equations presented in Section 2.3. The first-order system (2.20) is discretized as

$$\Omega U^{n+1} + \mathscr{D}\,\mathbf{W}^{n+1} = F^n, \quad \mathscr{G} U^{n+1} - \mathbf{W}^{n+1} = 0. \tag{4.12}$$

If the flux is eliminated from (4.12), then the fully discretized form of the diffusion equation (2.21) is obtained:

$$\mathscr{A} U^{n+1} = (\Omega + \mathscr{D}\mathscr{G})U^{n+1} = F^{n+1}. \tag{4.13}$$

On the other hand, if U is eliminated from (4.12) then a single equation for the flux analogous to (2.22), is obtained:

$$\mathscr{B}\,\mathbf{W}^{n+1} = (\mathscr{I} + \mathscr{G}\,\Omega^{-1}\mathscr{D}\,)\mathbf{W}^{n+1} = \mathscr{G}\,\Omega^{-1}F^{n+1}. \tag{4.14}$$

Here it is assumed that Ω is positive definite.

The operators $\mathscr{A}$ and $\mathscr{B}$ have nice symmetry and positivity properties. In the natural inner product

$$(\mathscr{A} U, V)_{HC} = (\Omega U, V)_{HC} + (\mathscr{G} U, \mathscr{G} V)_{\mathscr{HS}}, \tag{4.15}$$

which makes it clear that $\mathscr{A}$ is symmetric and positive, and positive definite if either Ω or $\mathscr{G}$ is positive definite. Next,

$$(\mathscr{B}\,\mathbf{W}, \mathbf{V})_{\mathscr{HS}} = (\mathbf{W}, \mathbf{V})_{\mathscr{HS}} + (\Omega^{-1}\mathscr{D}\,\mathbf{W}, \mathscr{D}\,\mathbf{V})_{HC}, \tag{4.16}$$

and then it is clear that $\mathscr{B}$ is symmetric and positive definite if Ω is positive definite.

Unfortunately, both $\mathscr{A}$ and $\mathscr{B}$ are nonlocal whenever $\mathscr{G}$ is nonlocal. There is a way of rearranging the computation of $\mathscr{A}$, as shown later, so that $\mathscr{A}U$ can be computed efficiently. This allows the use of a large class of iterative solvers in the numerical algorithms. In the case of $\mathscr{B}$, note that

$$\mathscr{S}\mathscr{B} = \mathscr{S} + \mathscr{D}\dagger\mathscr{M}\,\Omega^{-1}\mathscr{D} \tag{4.17}$$

is banded, so $\mathscr{B}^{-1} = (\mathscr{S}\mathscr{B})^{-1}\mathscr{S}$ is the product of an inverse of a banded matrix and a banded matrix.

The structure of $\mathscr{B}$ is related to the notion of *compact finite-difference schemes* that are widely used on 1D and 2D on rectangular grids (see, for example, [3, 17, 16]). To see this, note that the discrete fluxes can be written as

$$\mathbf{W} = \mathscr{G}U = \mathscr{S}^{-1}\mathscr{D}\dagger\mathscr{M}U$$

and, then, applying the operator $\mathscr{S}$ to both sides of this equation gives

$$\mathscr{S}\mathbf{W} = \mathscr{D}\dagger\mathscr{M}U. \tag{4.18}$$

As above, $\mathscr{S}$ and $\mathscr{D}\dagger\mathscr{M}$ are banded, so this is a standard form for a compact representation of the fluxes.

4.5. The Computational Sequence

Two approaches are considered here: solve Eq. (4.13) for U; or solve Eq. (4.14) for $\mathbf{W}$. In the first case, it has already been observed that the operator $\mathscr{A} = \Omega + \mathscr{D}\mathscr{G}$ is nonlocal because $\mathscr{G}$ is nonlocal, and thus it is not practical to use direct methods on this problem for large systems. The main difficulty in arranging this equation for an efficient solution is that $\mathscr{G}$ is to the right of $\mathscr{D}$ in the formula for $\mathscr{A}$. However, many iterative algorithms only require an efficient way of multiplying by the operator $\mathscr{A}$, and this can be implemented in a reasonable way. To compute $\mathscr{A}U$, reintroduce the flux $\mathbf{W} = \mathscr{G}U$ and note that $\mathbf{W}$ satisfies Eq. (4.18), $\mathscr{S}\mathbf{W} = \mathscr{D}\dagger\mathscr{M}U$, which can be solved for $\mathbf{W}$ using standard methods because $\mathscr{S}$ is banded, symmetric, and positive definite. Then $\mathscr{A}U = \Omega U + \mathbf{W}$. Note that this approach requires the computation of the fluxes as an intermediate step so we will turn to a flux-based algorithm and not pursue this approach any further.

In the second case, as observed before, $\mathscr{B}$ is nonlocal, but since $\mathscr{S}\mathscr{B}$ is local, combine Eq. (4.14) multiplied by $\mathscr{S}$, and (4.18) to get

$$(\mathscr{S} + \mathscr{D}\dagger\mathscr{M}\Omega^{-1}\mathscr{D})\mathbf{W}^{n+1} = \mathscr{D}\dagger\mathscr{M}\Omega^{-1}F^{n+1}, \tag{4.19}$$

which is convenient for computation; in particular, (4.19) has a local stencil. To see the properties of the coefficient matrix of (4.19), note that

$$\begin{aligned}
&[(\mathscr{S} + \mathscr{D}\dagger\mathscr{M}\Omega^{-1}\mathscr{D})\mathbf{W}, \mathbf{W}]_{\mathscr{H}\mathscr{S}} \\
&\quad = [\mathscr{S}\mathbf{W}, \mathbf{W}]_{\mathscr{H}\mathscr{S}} + [\mathscr{M}\Omega^{-1}\mathscr{D}\mathbf{W}, \mathscr{D}\mathbf{W}] \\
&\quad \geq \delta_2\|\mathbf{W}\|_{\mathscr{H}\mathscr{S}} + \frac{\delta_1}{C}\|\mathscr{D}\mathbf{W}\|_{HC},
\end{aligned} \tag{4.20}$$

where the constants are defined in (3.16) and (4.11). Thus the coefficient matrix is symmetric and positive definite.

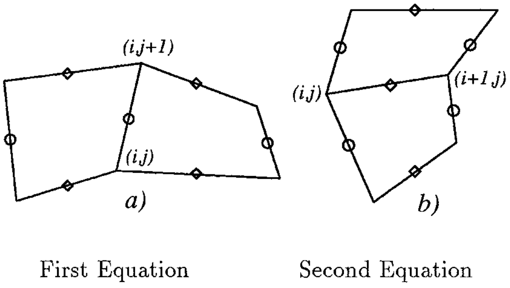


FIG. 5. Footprint for the equations for the fluxes.

Equation (4.19) can be written in components as

$$\mathscr{C}_{1,1}W\xi^{n+1} + \mathscr{C}_{1,2}W\eta^{n+1} = \mathscr{F}_1^{n+1}, \tag{4.21}$$

$$\mathscr{C}_{2,1}W\xi^{n+1} + \mathscr{C}_{2,2}W\eta^{n+1} = \mathscr{F}_2^{n+1}. \tag{4.22}$$

The definition $\mathscr{C} = \mathscr{S} + \mathscr{D}\dagger\mathscr{M}\Omega^{-1}\mathscr{D}$ implies that Eqs. (4.21) and (4.22) have the stencil footprints shown in Fig. 5 and the operators $\mathscr{C}_{1,1}$ and $\mathscr{C}_{2,2}$ are tridiagonal. The block or line Gauss–Seidel method is used to solve (4.22) and (4.23):

$$\mathscr{C}_{1,1}W\xi^{(s+1)} + \mathscr{C}_{1,2}W\eta^{(s)} = \mathscr{F}_1^{n+1}, \tag{4.23}$$

$$\mathscr{C}_{2,1}W\xi^{(s+1)} + \mathscr{C}_{2,2}W\eta^{(s+1)} = \mathscr{F}_2^{n+1}. \tag{4.24}$$

To compute $W\xi^{(s+1)}$ from 4.23 or $W\eta^{(s+1)}$ from 4.24, a tridiagonal system of equations must be solved, which can be done using direct methods. After the fluxes $\mathbf{W}^{n+1}$ are computed, the temperature is computed using the explicit formula

$$U^{n+1} = \Omega^{-1}(F^{n+1} - \mathscr{D}\mathbf{W}^{n+1}). \tag{4.25}$$

Also, note that by multiplying Eq. (4.25) by Ω and using the divergence property (4.3) gives

$$(\Omega U^{n+1}, 1)_{HC} = (F^{n+1}, 1)_{HC}, \tag{4.26}$$

which is a discrete conservation law analogous to the continuum conservation law (2.24). It is important to note that this conservation law does not depend on Eq. (4.19). Thus if (4.25) is computed exactly, then the conservation law will hold exactly. It is common to solve (4.19) using an iterative method, so the solution for $\mathbf{W}$ may contain significant errors, but this will not effect the conservation law. In fact, for many algorithms used for solving the diffusion equations, the conservation property is violated because an iterative method is used to solve for U, and the U obtained does not exactly satisfy the discrete equations (see [28]).

4.6. Discrete Operators on a Rectangular Grid

One can see more about the relationship between the support-operators discretizations and other discretizations by looking at rectangular grids: $S\xi_{(i,j)} = \Delta y$; $S\eta_{(i,j)} = \Delta x$; $VC_{(i,j)} = \Delta x\, \Delta y$; $\sin(\varphi^{(i,j)}_{(k,l)}) = 1$; and $\cos(\varphi^{(i,j)}_{(k,l)}) = 0$. The formula for the divergence, $\mathbf{DIV} = \mathscr{D}$, on the interior of the grid, is then

$$(\mathbf{DIV}\,\mathbf{W})_{(i,j)} = \frac{W\xi_{(i+1,j)} - W\xi_{(i,j)}}{\Delta x} + \frac{W\eta_{(i,j+1)} - W\eta_{(i,j)}}{\Delta y}, \tag{4.27}$$

which is a natural discretization for rectangular grids.

In this simple situation, Eqs. (3.15) and (3.7) imply that the flux $\mathbf{W} = \mathscr{G}U$ in the interior can be written as

$$W\xi_{(i,j)} = \frac{2k_{(i-1,j)}k_{(i,j)}}{k_{(i-1,j)} + k_{(i,j)}}\frac{U_{(i,j)} - U_{(i-1,j)}}{\Delta x}, \tag{4.28}$$

$$W\eta_{(i,j)} = \frac{2k_{(i,j-1)}k_{(i,j)}}{k_{(i,j-1)} + k_{(i,j)}} \cdot \frac{U_{(i,j)} - U_{(i,j-1)}}{\Delta y}. \tag{4.29}$$

Thus, this form of the support-operators approach leads to the well-known harmonic average for the coefficient k (see [24]). On the boundary, the fluxes are given by one-sided differencing. For example, on the left boundary,

$$W\xi_{(1,j)} = k_{(1,j)} \cdot \frac{U_{(1,j)} - U_{(0,j)}}{\Delta x/2}. \tag{4.30}$$

The formulas on the remaining parts of the boundary are similar.

To find the discrete analog of $\nabla \cdot k\, \nabla$, introduce the notation

$$k\xi_{(i,j)} = \frac{2k_{(i-1,j)}k_{(i,j)}}{k_{(i-1,j)} + k_{(i,j)}}, \quad k\eta_{(i,j)} = \frac{2k_{(i,j-1)}k_{(i,j)}}{k_{(i,j-1)} + k_{(i,j)}}, \tag{4.31}$$

for the harmonic averages of k, and then in the internal cells, the discrete analog of $\nabla \cdot k\, \nabla u$ is

$$\frac{k\xi_{(i+1,j)}\dfrac{U_{(i+1,j)} - U_{(i,j)}}{\Delta x} - k\xi_{(i,j)}\dfrac{U_{(i,j)} - U_{(i-1,j)}}{\Delta x}}{\Delta x} + \frac{k\eta_{(i,j+1)}\dfrac{U_{(i,j+1)} - U_{(i,j)}}{\Delta y} - k\eta_{(i,j)}\dfrac{U_{(i,j)} - U_{(i,j-1)}}{\Delta y}}{\Delta y} \tag{4.32}$$

which is a standard five-cell approximation on a rectangular grid.

5. PROPERTIES OF THE ALGORITHM

In the support-operators setting, the errors in approximating the divergence and gradient are checked first, and then these results are used to study other operators. There are two natural notions of truncation error: one for point values and one for averaged values. The truncation error is estimated both for *general* grids (those satisfying assumptions (3.2)) and *smooth* grids, for which there is a smooth transformation $(x(\xi, \eta), y(\xi, \eta))$ from the unit square in the logical space (ξ, η) to the region V, so that

$$x_{(i,j)} = x(\xi_i, \eta_j), \quad y_{(i,j)} = y(\xi_i, \eta_j), \tag{5.1}$$

where $\xi_i = (i - 1)/(M - 1)$, $\eta_i = (j - 1)/(N - 1)$, $1 \leq i \leq M$, and $1 \leq j \leq N$ (for details see [15]).

Because $\mathscr{G}$ is given by one-sided differences on the boundary for orthogonal grids, the local truncation error on the boundary cannot be more than first-order accurate. Thus the accuracy of $\mathscr{G}$ is only estimated in the interior of the region. The operator $\mathscr{D}$ is the normal component of flux on the boundary and hence is *exact*. When the conductivity k is variable, it is not possible to isolate a gradient in the discrete flux operator $\mathscr{G}$. However, when $k = 1$, then the gradient is defined by $\mathbf{GRAD} = \mathscr{G}$ for both the interior and boundary.

This section is completed by checking some other important properties. The null space of the discrete flux operator or gradient is particularly important; if the flux operator applied to a function or the gradient of a function is zero, then the function is a constant. Finally, the properties of the discrete operators are studied on very distorted grids.

In this section, we give the definitions of the various truncation errors and summarize the results for these errors. The proofs are lengthy Taylor series calculations and are given in detail in [32].

5.1. Truncation Error

Let $p_h : H \to HC$ be a projection operator from the space of continuous scalar functions to a space of discrete scalar functions, and let $P_h : \mathbf{H} \to \mathscr{HS}$ be a projection operator from the space of continuous vector functions to a space of discrete vector functions, where h estimates the size of the largest cell in the discrete grid. Then the truncation error ψ for the divergence, gradient, and Laplacian are given by

$$\psi_{\mathbf{DIV}}(\mathbf{w}) = p_h(\nabla \cdot \mathbf{w}) - \mathbf{DIV}(P_h\mathbf{w}), \tag{5.2}$$

$$\psi_{\mathbf{GRAD}}(u) = P_h(\nabla u) - \mathbf{GRAD}(p_h u), \tag{5.3}$$

$$\psi_{\mathbf{DIV\,GRAD}}(u) = p_h(\nabla \cdot \nabla u) - \mathbf{DIV\,GRAD}(p_h u). \tag{5.4}$$

5.1.1. *Projections for Scalars*

The *integral average* projection p^1 and the *point* projection p^2 for scalar functions are given by

$$p_h^1(u)_{(i,j)} = \frac{\int_{V_{(i,j)}} u dV}{VC_{(i,j)}}, \quad p_h^2(u)_{(i,j)} = u(x_{(i,j)}^c, y_{(i,j)}^c), \qquad (5.5)$$

where $V_{(i,j)}$ is cell (i, j) and (x_c, y_c) is the geometric center of a cell. Note that

$$(p_h^2 u)_{(i,j)} = (p_h^1 u)_{(i,j)} + O(h), \qquad (5.6)$$

because point values are first-order approximations of averages (see [32]).

5.1.2. *Projections for Vectors*

The *integral average* P_h^1 and the *point* projections for vectors are

$$P_h^1 = \begin{pmatrix} P^1 \xi_h \\ P^1 \eta_h \end{pmatrix}, \quad P_h^2 = \begin{pmatrix} P^2 \xi_h \\ P^2 \eta_h \end{pmatrix}, \qquad (5.7)$$

where the components of the integral average projection are the integral averages of the normal component of the vector over the cell sides:

$$\begin{aligned} (P^1 \xi_h(\mathbf{w}))_{(i,j)} &= \frac{\int_{S\xi_{(i,j)}} (\mathbf{w}, \mathbf{n})\, dS}{S\xi_{(i,j)}}, \\ (P^1 \eta_h(\mathbf{w}))_{(i,j)} &= \frac{\int_{S\eta_{(i,j)}} (\mathbf{w}, \mathbf{n})\, dS}{S\eta_{(i,j)}}, \end{aligned} \qquad (5.8)$$

and the components of point projection for a vector are given by the values of the normal component of the vector at the middle of the sides of the cell:

$$\begin{aligned} (P^2 \xi_h(\mathbf{w}))_{(i,j)} &= (\mathbf{w}, \mathbf{n})|_{(x\xi_{(i,j)}, y\xi_{(i,j)})}, \\ (P^2 \eta_h(\mathbf{w}))_{(i,j)} &= (\mathbf{w}, \mathbf{n})|_{(x\eta_{(i,j)}, y\eta_{(i,j)})}, \end{aligned} \qquad (5.9)$$

where

$$x\xi_{(i,j)} = \frac{(x_{(i,j)} + x_{(i,j+1)})}{2}, \quad y\xi_{(i,j)} = \frac{(y_{(i,j)} + y_{(i,j+1)})}{2}, \qquad (5.10)$$

$$x\eta_{(i,j)} = \frac{(x_{(i,j)} + x_{(i+1,j)})}{2}, \quad y\eta_{(i,j)} = \frac{(y_{(i,j)} + y_{(i+1,j)})}{2}. \qquad (5.11)$$

5.1.3. *The Truncation Error for the Divergence*

For general grids satisfying (3.2), the divergence is exact when the integral-average projection (5.2) is used:

$$\psi_{\mathbf{DIV}}^1(\mathbf{w}) = p_h^1(\nabla \cdot \mathbf{w})_{(i,j)} - \mathbf{DIV}(P^1\mathbf{w})_{(i,j)} = 0. \qquad (5.12)$$

For point projections, the truncation error is

$$\psi_{\mathbf{DIV}}^2(\mathbf{w}) = p_h^2(\nabla \cdot \mathbf{w})_{(i,j)} - \mathbf{DIV}(P^1\mathbf{w})_{(i,j)} = O(h), \qquad (5.13)$$

so the divergence has the first-order truncation error in the maximum norm.

It is also easy to check that

$$\mathbf{DIV}\, P_h^1(\mathbf{C}) = 0; \qquad (5.14)$$

that is, **DIV** of a constant vector is equal to zero.

For smooth grids (5.1) and point projections, the estimate of the truncation error for the divergence can be improved to second order in the maximum norm:

$$\psi_{\mathbf{DIV}}^2(\mathbf{w}) = p_h^2(\nabla \cdot \mathbf{w}) - \mathbf{DIV}(P_h^2\mathbf{w}) = O(h^2). \qquad (5.15)$$

This is done by writing the divergence in general coordinates, writing the discrete divergence in an analogous form, and then comparing the various parts of the two expressions (see [32] for the details).

5.1.4. *Truncation Error for the Flux Operator and the Gradient*

For general grids satisfying (3.2), the point-projection truncation error for the flux operator,

$$\psi_{\mathscr{G}}^2(u) = P_h^2(\mathbf{G}u) - \mathscr{G}(p_h^2 u), \qquad (5.16)$$

is first-order accurate in the mean-square norm,

$$\| \psi_{\mathscr{G}}^2 \|_{\mathscr{H}\mathscr{S}} = O(h). \qquad (5.17)$$

The proof of this result also shows that for u linear $\psi_{\mathscr{G}}^2(u) = 0$. For smooth grids, the previous estimate can be improved to

$$\psi_{\mathscr{G}}^2(u) = P_h^2(\mathbf{K}\nabla u) - \mathscr{G}(p_h^2 u) = O(h^2) \qquad (5.18)$$

on the interior of the grid. The proof of (5.17) requires a detailed analysis of the geometry of a cell and thus is rather lengthy. The proof of (5.18) is similar to the analogous result for the divergence. A proof of both results, when **K** is the identity, is given in [32]. The extension to the case of smooth variable diffusion coefficient k is straightforward, but lengthy.

Note that (5.17) can be used to show that the algorithm is first-order accurate in the integral norm by standard energy methods; see [34]. Also, note that Eq. (4.18), which gives the relationship between the fluxes and the temperature, is $\mathscr{S}\mathbf{W} = \mathscr{D}\dagger\mathscr{M}U$. A key point in the proof of (5.17) is to show that $\mathscr{S}\psi^2_{\mathbf{GRAD}}$ is third order, which then implies that

$$\mathscr{S}\,\mathbf{GRAD}\,U = \mathscr{D}\dagger\mathscr{M}U + O(h^3),$$

which is an analog of (4.18). The previous equation has a simple geometric meaning. The right-hand side of the equation is an expression which is proportional to the directional derivative in the direction from the center of one cell to the center of another. Also on the left-hand side, there is a combination of directional derivatives in directions normal to the edges of the cell, but all of these normal derivatives are at different locations.

5.2. The Conservation Law and the Gradient of Constants

The action of the approximate divergence and gradient on constant and linear fields has a strong impact on the quality of algorithms, especially for the support-operators algorithm. In fact, the flux operator applied to constants giving zero is closely related to the conservation law for the divergence. Equations (4.4) and (4.8) show that

$$(\mathscr{D}\mathbf{W}, 1)_{HC} = (\mathbf{W}, \mathscr{G}1)_{\mathscr{HS}} \tag{5.19}$$

for all $\mathbf{W}$, so that by the divergence property (4.3), $(\mathscr{D}\mathbf{W}, 1)_{HC} = 0$ is equivalent to $\mathscr{G}1 = 0$. When the diffusion coefficient k is constant, then this is equivalent to $\mathbf{GRAD}\ 1 = 0$. By linearity, this is equivalent to the discrete flux operator or the gradient killing constants. The equivalence of the divergence property of an operator to the adjoint operator killing constants is one of the nice features of this theory.

5.3. The Null Space of the Discrete Gradient

There is another important property of the gradient: $\nabla u = 0$ implies that u is a constant. Because $\mathbf{K} > 0$, this is the same as $\mathbf{G}u = \mathbf{K}\nabla u = 0$ implies that u is a constant. The same holds for the discrete operators. To see this, assume that $\mathscr{G}U = 0$ and recall that $\mathscr{G}U = \mathscr{S}^{-1}\mathscr{D}\dagger\mathscr{M}U$, so that

$$\mathscr{D}\dagger\mathscr{M}U = 0. \tag{5.20}$$

Then Eq. (4.9) gives

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

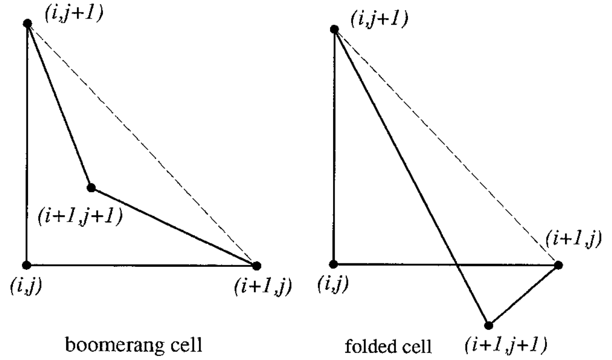


FIG. 6. Distorted grid cells.

The first formula above implies that U is constant in i, while the second implies that U is constant in j, so U is constant. When the diffusion coefficient is constant, this becomes $\mathbf{GRAD}\ U$ equals zero if and only if U is a constant.

The property that the null space of the discrete gradient $\mathbf{GRAD}$ contains only the constant functions, exactly as for the differential operator ∇, is an important property for numerical schemes. There are many finite-difference and support-operator schemes that do not have this property, particularly those where vectors are co-located at the corners of cells, as is the case in many Lagrangian codes. For such schemes, typically the highest-frequency mode on the grid is also in the null space of the discrete gradient, and then a special procedure for filtering noise from the solution is required, as in the method of Margolin [21].

5.4. Severe Grid Distortion

The algorithm presented in this paper is provably correct for grids with convex cells. The numerics show that the algorithm produces good results for grids with non-self-intersecting cells, including grids with nonconvex cells such as the boomerang cell shown in Fig. 6, provided the iterative solver converges. Nonconvex cells do cause trouble for the iterative solver.

The difficulty comes from the formula for the weights in the definition of the inner product of vectors (3.7) which also appear in the definition (3.15) of the operator $\mathscr{S}$ which must be iteratively inverted. As noted just after Eq. (3.7), the weights in this definition are one half of the area of the triangle which has its vertex at the point $(i + k, j + l)$ and belongs to cell (i, j), divided by the volume of the cell. For example, for both of the cells in Fig. 6, the weight $V^{(i,j)}_{(i+1,j+1)}$ is negative.

Sometimes it is desirable to use the algorithm presented here with grids that have some severely distorted cells. This can be done by introducing new weights

$$\tilde{V}^{(i,j)}_{(i+k,j+l)} = \frac{\left|V^{(i,j)}_{(i+k,j+l)}\right|}{\sum_{p,q=0}^{1}\left|V^{(i,j)}_{(i+p,j+q)}\right|}, \tag{5.21}$$

which are always positive, and consequently, no singularity exists in the formulas in the support-operators formulation. Formula (5.21) produces positive weights, even for grids with self-intersecting meshes, but where all cells still have positive total volume $VC_{(i,j)}$. For convex cells, the areas of all triangles are positive and the sum of their areas equals twice the area of $VC_{(i,j)}$, so these formulas give the same values as the original formulas. For grids with nonconvex cells, the new weights produce an $\mathscr{S}$ that is symmetric and positive-definite, and thus the iterative solver is stabilized. The numerical examples show that the results of the stabilized algorithm are reasonable. Clearly, if the weights are defined by (5.21), then the finite-difference scheme is no longer exact for linear functions if there are nonconvex cells in the grid.

It is important to note that for hourglassing, the mesh is still convex, so the algorithm presented here is not sensitive to this distortion as is the one presented in [22].

There is an alternative approach—subdivide the nonconvex cells into triangles with positive volumes and then introduce the values of U at the centers of each triangle and an additional flux on their common boundaries. Since this procedure requires special differencing for each nonconvex cell and, also, is not fully compatible with Lagrangian hydrodynamics, where there is just one U per cell, the details are not presented here.

6. NUMERICAL EXAMPLES

First, the algorithm was tested on smooth grids to confirm its second-order convergence rate (see [32] for details). In 1992, Morel *et al.* [22] developed a scheme that is applicable to the types of problems we are interested in and show that it is better than all other known schemes that are compatible with Lagrangian hydrodynamics codes. Thus, the first three examples compare the support-operators method to that of [22]. The first example shows that the support-operators method is better than Morel's method on the highly irregular Kershaw grid. The second example demonstrates a second-order convergence rate on random grids. The third example shows that the support-operator method is better than Morel's method on the highly distorted Shestakov grid which has some nonconvex cells. The fourth example demonstrates the performance of the positive weights on a strongly nonconvex grid.

The last three examples check the performance of the support-operators method on problems with discontinuous conductivity. The fifth and seventh examples have linear solutions, so the support-operators method is exact for these problems, while the sixth example demonstrates a second-order convergence rate. The last example is exceptionally important as it demonstrates the accuracy of the support-operators method on a problem with discontinuous tangential flux at an interface. In summary, for smooth grids many methods are comparable to the support-operators method, but for rough grids or discontinuous diffusion coefficients the support-operators method is superior.

The asymptotic error E_h on a grid of $M \times N$ nodes is estimated by

$$\|E_h\| = Ch^q + O(h^{q+1}), \tag{6.1}$$

where $h = \max\{1/(M-1), 1/(N-1)\}$, q is the order of the truncation error, the convergence-rate constant C is independent of h, and $\|\cdot\|$ is some norm. In the numerical examples, the asymptotic errors were evaluated on a sequence of grids with h, $h/2$, $h/4$, ... and then the order of convergence q is estimated by $q \approx \log_2\|E_h\|/\|E_{h/2}\|$. The exact solution is discretized using the point projection operator (5.5). The convergence rates can be 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,

$$\begin{aligned} E_{L_2}^2 &= \|U - p_h u\|_{L_2}^2 \\ &= \left(\sum_{i=1}^{M-1} \sum_{j=1}^{N-1} (U_{(i,j)} - (p_h u)_{(i,j)})^2 VC_{(i,j)} \right), \end{aligned} \tag{6.2}$$

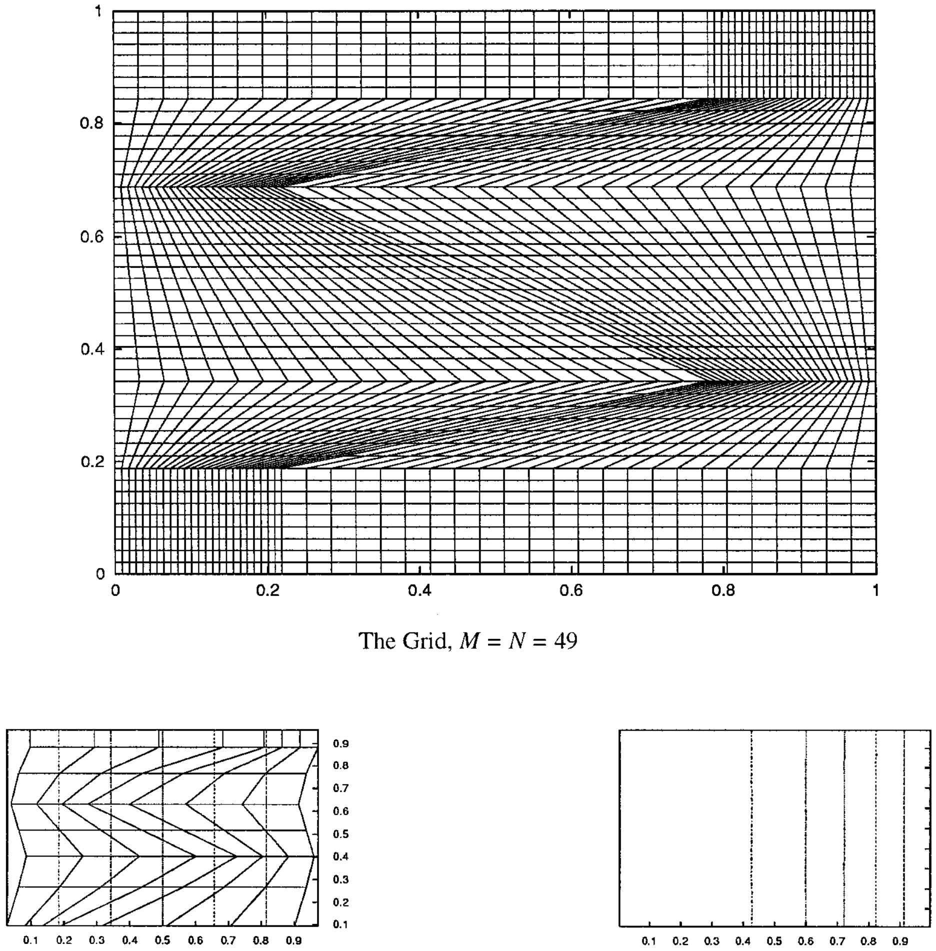
where $U_{(i,j)} = U^n_{(i,j)}$ is the solution of the finite-difference scheme and $u = u(x, y, t)$ is the exact solution of the given problem.

THE STANDARD TEST PROBLEM. The standard test problem is the diffusion equation in the unit square $0 \leq x, y \leq 1$. One-dimensional test problems are extended to two dimensions by changing the one-dimensional diffusion operator to a two-dimensional one. The symmetry of the heat equation and the boundary conditions will then imply that the solution is only one-dimensional. The equation to be solved is the diffusion equation

$$\frac{1}{v}\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}\left[D\frac{\partial u}{\partial x}\right] + \frac{\partial}{\partial y}\left[D\frac{\partial u}{\partial y}\right] + f, \tag{6.3}$$

where D is the diffusion coefficient, v is the capacity, and f is the source term. The notation is chosen to be consistent with Morel *et al.* [22], where v is the particle speed and u is the intensity.

The standard boundary conditions are that there is zero flux through the top and bottom boundaries and



The Grid, $M = N = 49$

Cell Centers & Isolines
$M = N = 9$

Solution at $t = 0.01$
$M = N = 9$

FIG. 7. Example 1: Kershaw Grid.

mixed or Robin boundary conditions on the left and right boundaries:

$$\begin{array}{llll} D\dfrac{\partial u}{\partial y} = 0 & \text{at } y = 0 & D\dfrac{\partial u}{\partial y} = 0 & \text{at } y = 1 \\ u - 2D\dfrac{\partial u}{\partial x} = 0 & \text{at } x = 0 & u + 2D\dfrac{\partial u}{\partial x} = 1 & \text{at } x = 1 \end{array}. \qquad (6.4)$$

The standard initial condition is $u(x, y, 0) = 0$. In some examples D is a function of the spatial variable, so it cannot be moved outside of the derivatives.

6.1. Comparison with Morel's Method

In [22], Morel *et al.* show that their algorithm gives better results than the algorithms proposed by Kershaw [14] and some other algorithms known from literature. Therefore, we will show that the support-operators algorithm developed here performs better than Morel's algorithm on the test problems from [22].

The first test problem is given by (6.3) and (6.4) with $v = 300$, $D = \frac{1}{30}$, and $f = 0$, which has 1D linear steady-state solution $u = (x + 2D)/(1 + 4D)$. As in [22], this problem was solved on the Kershaw *grid,* which is shown in Fig. 7. Because the support-operators scheme is exact for linear functions, it must, as it does, reproduce the steady-state solution exactly for any grid with convex cells. For $M = N = 9$, the lines connecting the cell centers and the isolines of the solution, which must be vertical straight lines, are shown in Fig. 7. The scheme proposed in [22] is also exact for linear functions on the Kershaw grid, but the other schemes mentioned in this paper, except the scheme from [1], are not exact for this problem.

Because of the symmetry of the problem, the isolines of the time-dependent solution must also be vertical straight lines. The isolines of the approximate solution at

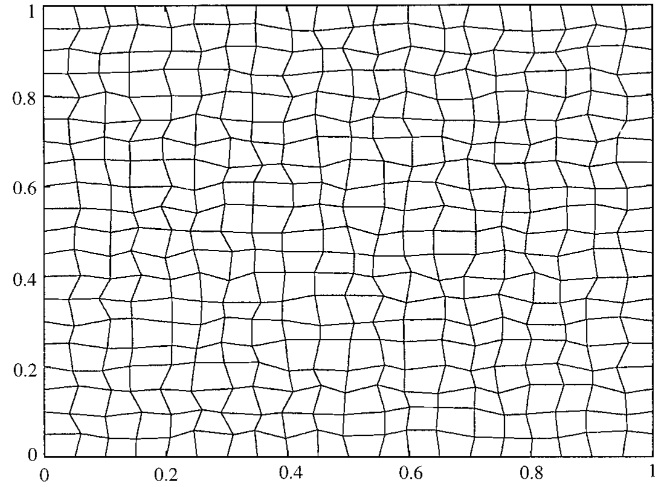


FIG. 8. Example 2: Random Grid, $M = N = 21$.

$t = 0.01$ are shown in Fig. 7. Because the exact solution for $t = 0.01$ is not linear, the support-operators method is not exact, and thus the isolines exhibit some small distortion. Comparison with Morel's method shows that the distortions cannot be distinguished visually. The Kershaw method gives very distorted solutions [22] for this problem.

6.2. Random Mesh Example

The second test problem is given by (6.3) and (6.4) with $v = 300$, $D = \frac{1}{30}$, and $f = Qx^2$ with $Q = 1$, which is just like the first test problem, except now there is a forcing term. The steady-state solution of this problem is $u = a + bx + cx^4$, where

$$a = \frac{Q}{6}\left[\frac{1+8D}{1+4D}\right], \quad b = \frac{Q}{12D}\left[\frac{1+8D}{1+4D}\right], \quad c = -\frac{Q}{12D}. \tag{6.5}$$

As in Morel *et al.* [22], this problem is used to demonstrate the accuracy of the support-operators algorithm as a function of the mesh size on "random" meshes. An example is shown in Fig. 8. This mesh was generated from a uniform orthogonal mesh by randomly displacing each interior node by 20% of the original cell width. Specifically, given an initial uniform cell width of w, each node is placed at a random position on a circle of radius $0.2w$, centered about the original position of the corner. Note that such grids satisfy the regularity condition for general grids formulated in Section 3.1. Because Morel *et al.* [22] use a *relative* mean-square norm, this norm of the error is also computed.

Results of the convergence tests on random grids are presented in Table I, where the first column gives the algorithm, the second gives the number of grid points with $N = M$, the next three columns give the maximum, mean-square, and the relative mean-square error, and the final three columns give the estimated orders of convergence. The results for the support-operators method are very close to the results for Morel's method. Both methods have a second-order convergence rate in all norms, and the error is very close to that obtained on the orthogonal grid to which the random perturbation was applied. It is shown in [22] that Kershaw's [14] and Pert's [25] methods do not converge on such grids.

TABLE I

Example 2: Errors on a Random Grid

Algorithm	M	Max norm	L_2 norm	Relative L_2	$q_{\max}$	q_2	$\tilde{q}_2$
Morel *et al.*	10	4.36E-2	1.79E-2	1.51E-2	2.06	1.95	1.96
	20	1.04E-2	4.61E-3	3.88E-3	1.60	1.70	2.01
	40	3.43E-3	1.41E-3	9.63E-4	—	—	—
Support operators	10	4.34E-2	1.87E-2	1.59E-2	2.04	2.20	1.96
	20	1.05E-2	4.06E-3	4.06E-3	1.72	1.79	2.02
	40	3.18E-3	1.17E-3	1.00E-3	—	—	—

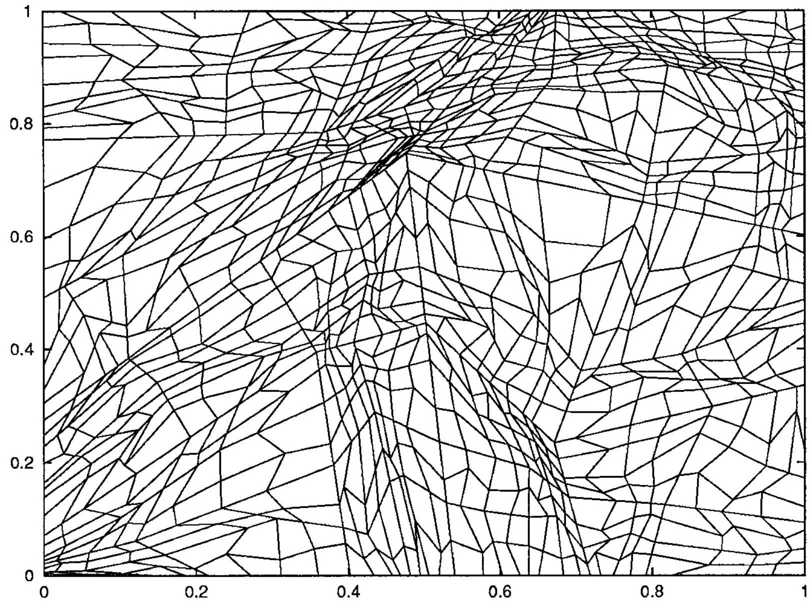
6.3. The Shestakov Grid

To complete the comparison with [22], the results of the computation of the stationary linear solution of Eq. (6.3) on the Shestakov grid, which is shown in Fig. 9, are given. This grid has three nonconvex cells. The isolines for the results obtained by Morel's method and results obtained by support-operators method, when the *modified* positive weights (5.21) are used, are also given in Fig. 9. Recall that, in this case, the support-operators scheme is no longer exact for linear functions. The isolines are significantly straighter for the support-operators method. The errors for both methods are shown in Table II. The errors show that the support-operators method is approximately 10 times more accurate in the max norm and approximately 20 times more accurate in the L_2 norm than Morel's method.

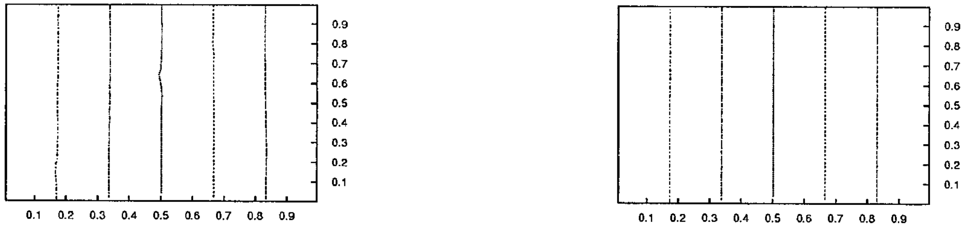
TABLE II

Example 3: Errors on the Shestakov Grid

Norm	Morel *et al.*	Support operators
Max-norm	1.01E-2	1.12E-3
L_2 norm	2.00E-3	1.07E-4
Morel *et al.* L_2 norm	2.98E-3	1.95E-4



Shestakov Grid, $M = N = 33$



Morel

Support Operators

FIG. 9. Example 3: Isolines on the Shestakov grid.

6.4. A Strongly Nonconvex Grid

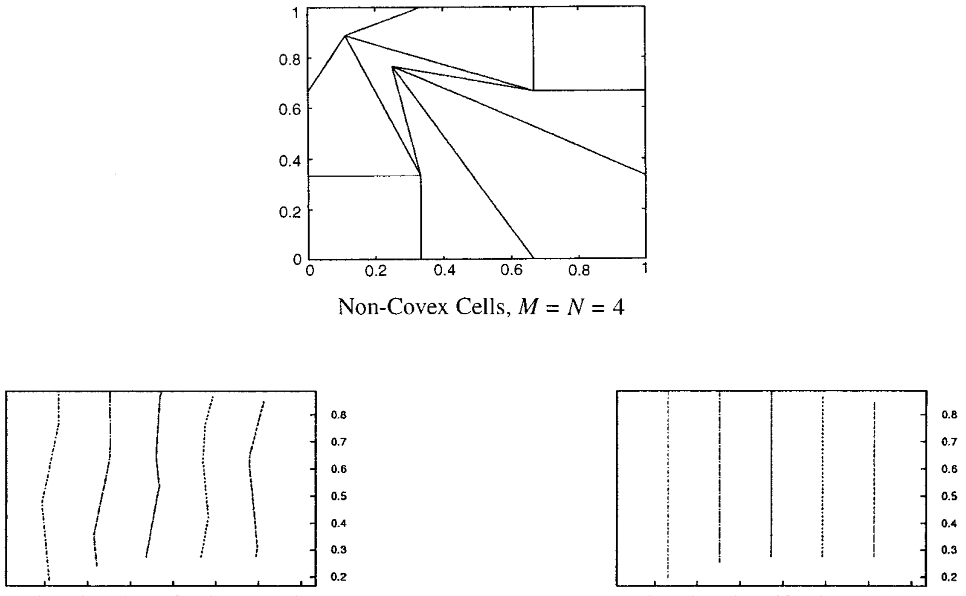
This test demonstrates the impact of using the *positive weights* (5.21) in the inner product on the accuracy of the support-operators method. The tests are made using the heat equation (6.3) and the boundary conditions (6.4) used in the first problem in this section and the grid displayed in Fig. 10. Most of the cells in this grid are nonconvex. The coordinates of the grid nodes are obtained by changing the coordinates of two nodes of the uniform 4 × 4 grid as follows: $x_{2,3} = \frac{1}{9}$; $y_{2,3} = \frac{8}{9}$; $x_{3,2} = \frac{1}{4}$; $y_{3,2} = \frac{23}{30}$. The isolines for the approximate solution for both the case of *positive* and the case of *negative* weights are shown in Fig. 10, which shows that using positive weights can lead to the loss of accuracy in the case when there are many nonconvex cells, and the use of the negative weights gives a much better solution.

This test also shows that the support-operators method is exact for linear solutions even when the grid cells are not convex, that is, when some of the weights in (3.7) are not positive. The main drawback of the negative weights is that the iterative method, used to solve the system of linear equations, may not converge. The drawback of using a modified positive weight is that the method is no longer exact on linear solutions, while the advantage is that the matrix is positive definite and then many iteration methods will always converge.

6.5. A Discontinuous Coefficient Problem

The diffusion coefficient D in the first problem, given by (6.3) and (6.4), is now changed to a discontinuous piecewise-constant function:

$$D = \begin{cases} D_1, & 0 < x < 0.5, \\ D_2, & 0.5 < x < 1. \end{cases} \tag{6.6}$$



Non-Covex Cells, $M = N = 4$

Positive Weights

Negative Weights

FIG. 10. Example 4: Isolines on the nonconvex grid.

In this case, the left and right boundary conditions are taken to be

$$u - 2D_1 \frac{\partial u}{\partial x} = 0, \quad u + 2D_2 \frac{\partial u}{\partial x} = 1, \tag{6.7}$$

and then the steady-state solution of this problem is

$$u = \begin{cases} \dfrac{D_2 x + 2D_1 D_2}{0.5(D_1 + D_2) + 4D_1 D_2}, & 0 < x < 0.5, \\ \dfrac{D_1 x + 2D_1 D_2 + 0.5(D_2 - D_1)}{0.5(D_1 + D_2) + 4D_1 D_2}, & 0.5 < x < 1, \end{cases} \tag{6.8}$$

which is a piecewise linear function. In the case when $D_1 = D_2 = D$, this solution is the same as for the first test problem.

This problem was solved on the 2D random grid shown in Fig. 11. The support-operators method is more accurate if the discontinuity coincides with a grid line, so the line with $x = \frac{1}{2}$ is fixed, but the y coordinates of points on this line are changed randomly. As expected, the support-operators scheme is exact for this problem; the isolines of the approximate solution are shown in Fig. 11. Note that Morel's method is also exact for this problem.

6.6. MacKinnon and Carey Example

The next test problem is from MacKinnon and Carey [19] and is the same as (6.3) and (6.4), except that the left and right boundary conditions are

$$\begin{aligned} u - 2D_1 \frac{\partial u}{\partial x} &= -2D_1 b_1, \\ u + 2D_2 \frac{\partial u}{\partial x} &= \frac{a_2}{2} + b_2 + c_2 + 2D_1(a_2 + b_2). \end{aligned} \tag{6.9}$$

The exact stationary 1D solution is

$$u(x) = \begin{cases} a_1 \dfrac{x^2}{2} + b_1 x, & 0 \le x \le \dfrac{1}{2}, \\ a_2 \dfrac{x^2}{2} + b_2 x + c_2, & \dfrac{1}{2} \le x \le 1, \end{cases} \tag{6.10}$$

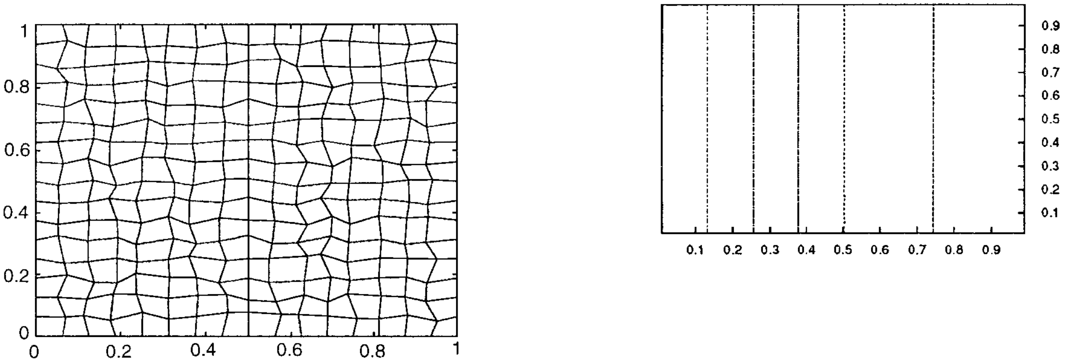
where

$$\begin{aligned} a_i &= \frac{-1}{D_i}, \quad b_1 = -\frac{3a_2 + a_1}{4} \frac{D_2}{D_1 + D_2}, \\ b_2 &= \frac{D_2}{D_1} b_1, \quad c_2 = -(b_2 + 0.5a_2). \end{aligned} \tag{6.11}$$

The convergence analysis is presented in Table III, where the data is arranged as in Table I. It is clear that the convergence rate is second order in both the max and L_2 norms.

6.7. Discontinuous Tangential Flux

All of the examples that we know of in the literature have both the normal and tangential components of flux



Random Grid

Isolines

FIG. 11. Example 5: Discontinuous coefficient, $M = N = 17$.

continuous at any discontinuity. However, the theory for discontinuous coefficients only implies that the normal component of flux is continuous. In fact, it is to be expected that the tangential component of flux is not continuous, as illustrated by the following simple (but important) example by Morel.

For this example the diffusion constant is chosen as

$$k(x, y) = \begin{cases} k_1, & x \leq \dfrac{1}{2}, \\ k_2, & x > \dfrac{1}{2}, \end{cases}.$$

A solution that has the discontinuity in the tangential flux at a discontinuity (interface) at $x = 0.5$ is

$$u(x, y) = \begin{cases} a + bx + cy, & x \leq \dfrac{1}{2}, \\ a + b\dfrac{k_1 - k_2}{2k_2} + b\dfrac{k_1}{k_2}x + cy, & x > \dfrac{1}{2}, \end{cases}$$

This solution and its normal component of flux are continuous at $x = \frac{1}{2}$, while tangential component of flux is $k_1 c$ on the left side of the interface and $k_2 c$ on the right side of the interface.

The numerical experiments use $a = b = c = 1$, $k_1 = \frac{1}{30}$, $k_2 = \frac{1}{300}$, and the smooth grid given by the transformation

$$x(\xi, \eta) = \xi + \varepsilon \sin(2\pi\xi)\sin(2\pi\eta),$$
$$y(\xi, \eta) = \eta + \varepsilon \sin(2\pi\xi)\sin(2\pi\eta),$$

with $\varepsilon = 0.1$. For this transformation, the grid line corresponding to $\xi = 0.5$ is a straight line and coincides with the interface. It is important that the transverse grid lines are not orthogonal to this interface.

The support-operators algorithm is exact for this problem because the solution is piecewise linear (see [32, 13]). Many other algorithms will have difficulties with this problem. For example, the errors for the method of Beek, Nooyen, and Wesseling [1] are presented in Table IV. The first column is the number of cells, the second and third are the max norm of the errors for the normal and tangential components of flux and the last column is the max error of the solution. It is clear from this data that the error in the solution is essentially independent of the number of nodes. Thus, this method cannot be used for problems which involve strongly discontinuous media and nonorthogonal grids.

This lack of convergence is due to an unremovable error which depends on the nonorthogonality of the grid near the interface. The derivation of the method in [1] assumes that the flux vector is continuous at the interface; that is,

TABLE III

Example 6: Errors for the MacKinnon Problem

M-1	Max norm	L_2-norm	$q_{\max}$	q_2
10	5.98E-2	3.40E-2	1.88	1.99
20	1.62E-2	8.54E-3	1.98	1.96
40	4.10E-3	2.18E-3	—	—

TABLE IV

Example 7: Errors for van Beek *et al.* Method

M	$E_{G\xi}$	$E_{G\eta}$	E_u
10	4.62E-4	1.98E-2	1.35E-2
20	2.77E-4	2.94E-2	1.28E-2
40	1.50E-4	3.24E-2	8.88E-3

any component of the flux is continuous, and this example seriously violates this assumption.

7. COMMENTS

It is clear that the results of this paper can be extended to 3D, but the algebra is quite complicated. This can be overcome by using a computer algebra system to write the code [18]. Because the support-operators method does not use any facts about the structure of the grid (just the structure of the cells) this method can be extended to unstructured grids (see [29]). Moreover, the support-operators method will work for the case of a general symmetric positive-definite matrix **K**, with discontinuous entries [13].

As we mentioned in the Introduction, the support-operators method is invariantly defined, so it can be used in any coordinate system. It is very easy to transform the formulas in this paper to any other coordinate system. In particular, one only needs to change the formulas for $S\xi$, $S\eta$, $V_{(i,j)}$, $V^{i,j}_{k,l}$, and then the formulas for the discrete operators in the new coordinates are the same as in the case of Cartesian coordinates. However, some care must be taken to preserve the symmetries of the solution. We just note here that this can be done for cylindrical coordinates.

The support-operators method, using natural discretizations for the magnetic and electric fields that involve the normal components of the magnetic field and the tangential component of an electrical field, can be used to solve Maxwell's equations and, in particular, solve the equations for the diffusion of magnetic fields, which is the natural generalization of the results of this paper to diffusion of vector fields. In this case, the derivation of the finite-difference scheme involves the construction of two different analogs of the curl operator [11].

ACKNOWLEDGMENTS

This work was performed under the auspices of the U.S. Department of Energy under Contract W-7405-ENG-36 and the DOE/BES Program in the Applied Mathematical Sciences Contract KC-07-01-01. The authors thank J. E. Morel for many helpful discussions, commenting on drafts of this paper and providing us with results of computation using his method. Thanks also to L. G. Margolin and J. M. Hyman for many fruitful discussions and comments on drafts of this paper, W. D. Joubert for his help with the NSPCG package, and J. E. Dendy and J. Otto for discussions related to using the multigrid method on problems in this paper. The authors thank P. Wesseling for discussions related to the method introduced in [1].

REFERENCES

1. P. van Beek, R. R. P. van Nooyen, and P. Wesseling, Accurate discretization of gradients on nonuniform curvilinear staggered grids, *J. Comput. Phys.* **117,** 364 (1995).
2. R. S. Bernard and H. Kapitza, How to discretize the pressure gradient for curvilinear MAC grids, *J. Comput. Phys.* **99,** 288 (1992).
3. M. H. Carpenter, 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).
4. J. E. Castillo, J. M. Hyman, M. J. Shashkov, and S. Steinberg, The sensitivity and accuracy of fourth order finite-difference schemes on non-uniform grids in one dimension, *Comput. Math. Appl.* **30**(8), 41 (1995).
5. B. Das, T. H. Robey, S. Steinberg, and D. Zhang, Comparisons of numerical solution methods for differential equations with discontinuous coefficients, *Math. Comput. Simul.* **36,** 57 (1994).
6. L. M. Degtyarev and A. P. Favorskii, A flow variant of the sweep method, *USSR Comput. Math. Math. Phys.* **8,** 252 (1968).
7. 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).
8. 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).
9. A. Favorskii, T. Korshiya, M. Shashkov, and V. Tishkin, A variational approach to the construction of difference schemes on curvilinear meshes for heat-conduction equation, *USSR Comput. Math. Math. Phys.* **20,** 135 (1980).
10. A. Favorskii, A. Samarskii, M. Shashkov, and V. Tishkin, Operational finite-difference schemes, *Differential Equations* **17,** 854 (1981).
11. A. Favorskii, T. Korshiya, M. Shashkov, and V. Tishkin, Variational approach to the construction of finite-difference schemes for the diffusion equations for magnetic field, *Differential Equations* **18**(7), 863 (1982).
12. J. M. Hyman, R. J. Knapp, and J. C. Scovel, High order finite volume approximations of differential operators on nonuniform grids, *Phys. D* **60,** 112 (1992).
13. J. Hyman, M. Shashkov, and S. Steinberg, Los Alamos National Laboratory Report LA-UR-96-532, 1996 (unpublished).
14. D. S. Kershaw, Differencing of the diffusion equation in the Lagrangian hydrodynamic codes, *J. Comput. Phys.* **39,** 375 (1981).
15. P. M. Knupp and S. Steinberg, *The Fundamentals of Grid Generation* (CRC Press, Boca Raton, FL, 1993).
16. S. H. Leventhal, An operator compact implicit method of exponential type, *J. Comput. Phys.* **46,** 138 (1982).
17. S. K. Lele, Compact finite difference schemes with spectral-like resolution, *J. Comput. Phys.* **103,** 16 (1992).
18. R. Liska, M. Shashkov, and A. Solovjov, Support-operators method for PDE discretisation: Symbolic algorithms and realization, *Math. Comput. Simul.* **35,** 173 (1993).
19. R. J. MacKinnon and G. F. Carey, Analysis of material interface discontinuities and superconvergent fluxes in finite difference theory, *J. Comput. Phys.* **75,** 151 (1988).
20. L. G. Margolin and T. F. Adams, Los Alamos National Laboratory Report LA-10249, 1985 (unpublished).
21. L. G. Margolin and A. E. Tarwater, ''A Diffusion Operator for Lagrangian Meshes,'' *Proceedings, Fifth International Conference on Numerical Methods in Thermal Problems, Montreal, Canada, June 1987,* edited by R. W. Lewis, K. Morgan, and W. G. Habashi, p. 1252; Lawrence Livermore National Laboratory Report UCRL-95652.
22. 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).
23. T. C. Oppe, W. D. Joubert, and D. R. Kincaid, *NSPCG User's Guide. Version 1.0, A Package for Solving Large Sparse Linear Systems by Various Iterative Methods,* CNA-216 (Center for Numerical Analysis, University of Texas at Austin, Austin, TX, 1988).

24. S. V. Patankar, *Numerical Heat Transfer and Fluid Flow* (Hemisphere, Washington, DC, 1980).

25. G. J. Pert, Physical constraints in numerical calculations of diffusion, *J. Comput. Phys.* **42,** 20 (1981).

26. S. R. Robertson, A finite-difference formulation of the equation of heat conduction in generalized coordinates, *Numer. Heat Transfer* **2,** 61 (1979).

27. A. A. Samarskii and E. S. Nikolaev, *Numerical Methods for Grid Equations, Vol. II, Iterative Methods* (Birkhauser, Basel, 1989).

28. M. Shashkov, Violation of conservation laws when solving difference equations by the iteration methods, *USSR Comput. Math. Math. Phys.* **22**(5), 131 (1982).

29. A. Solov'ov and M. Shashkov, Finite-difference schemes for solution of heat equation on Dirichlet grid, preprint 27 (All Union Center for Mathematical Modeling, U.S.S.R. Academy of Sciences, Moscow, 1991).

30. M. Shashkov and S. Steinberg, Support-operator finite-difference algorithms for general elliptic problems, *J. Comput. Phys.* **118,** 131 (1995).

31. M. Shashkov, *Conservative Finite-Difference Methods on General Grids* (CRC Press, Boca Raton, FL, 1995).

32. M. Shashkov, and S. Steinberg, Los Alamos National Laboratory Report LA-UR-95-3582, 1995 (unpublished).

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

34. J. C. Strikwerda, *Finite Difference Schemes and Partial Differential Equations* (Wadsworth, Belmont, CA, 1989).

35. J. F. Thompson, Z. U. A. Warsi, and C. W. Mastin, *Numerical Grid Generation: Foundations and Applications* (North-Holland/Elsevier, New York, 1985).

36. B. Das, Schaffer, S. Steinberg, and S. Weber, Finite difference methods for modeling porous media flows, *Transport Porous Media* **17,** 171 (1994).

37. P. Wesseling, P. van Beek, and R. R. P. van Nooyen, Aspects of non-smoothness in flow computation, in *Computational Methods in Water Resources X,* Vol. 2, edited by A. Peters, G. Wittum, B. Herrling, U. Meissner, C. Brebbia, W. G. Gray and G. F. Pinder (Kluwer Academic, New York, 1994), p. 1263.

38. M. F. Wheeler and R. Gonzalez, Mixed finite element methods for petroleum reservoir engineering problems, in *Computing Methods in Applied Science and Engineering, VI,* edited by R. Glowinski and J.-L. Lions, (Elsevier, Amsterdam, 1984), p. 639.

39. Z. Zijlema and P. Wesseling, On accurate discretization of turbulence transport equations in general coordinates, in *Numerical Methods in Laminar and Turbulent Flow,* Vol. 9, Part I, Proceedings, Ninth International Conference, Atlanta, July 10–14, 1995, edited by C. Taylor and P. Durbetaki (Pineridge Press, Swansea, UK, 1995), p. 34.