Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 219 (2006) 68–85
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Higher-order mimetic methods for unstructured meshes

V. Subramanian, J.B. Perot *

University of Massachusetts, Amherst, Mechanical and Industrial Engineering, Amherst, MA 01003, United States

Received 27 October 2005; received in revised form 15 March 2006; accepted 16 March 2006
Available online 8 May 2006

Abstract

A higher-order mimetic method for the solution of partial differential equations on unstructured meshes is developed and demonstrated on the problem of conductive heat transfer. Mimetic discretization methods create discrete versions of the partial differential operators (such and the gradient and divergence) that are exact in some sense and therefore mimic the important mathematical properties of their continuous counterparts. The proposed numerical method is an interesting mixture of both finite volume and finite element ideas. While the ideas presented can be applied to arbitrarily high-order accuracy, we focus in this work on the details of creating a third-order accurate method. The proposed method is shown to be exact for piecewise quadratic solutions and shows third-order convergence on arbitrary triangular/tetrahedral meshes. The numerical accuracy of the method is confirmed on both two-dimensional and three-dimensional unstructured meshes. The computational cost required for a desired accuracy is analyzed against lower-order mimetic methods.
© 2006 Elsevier Inc. All rights reserved.

Keywords: High order; Unstructured; Staggered; Dual mesh; Mimetic; Diffusion

1. Introduction

The accuracy of a numerical method can be increased either by refining the mesh or by increasing the order of accuracy of the discretization scheme. A discussion of the use of mimetic methods with mesh refinement is found in Perot and Nallapati [1,2]. In contrast, this work focuses on increasing the discretization order of accuracy. Higher-order accuracy is useful for constructing multiscale turbulence models [3] and for minimizing the influence of discretization error on dynamic subgrid-scale models for large Eddy simulation (LES) [4]. Higher-order accuracy may also be attractive for problems involving moving meshes since the mesh motion overhead is proportionally smaller.

There are three fundamentally different approaches to increasing the order of accuracy of discrete operators. Finite volume methods tend to increase the effective stencil of the discrete operators (either explicitly or implicitly) but keep the number of unknowns used to solve the PDE fixed [5,6]. Finite element methods increase the number of unknowns (and equations) but keep the stencils highly local. Finally, Padé schemes

* Corresponding author. Tel.: +1 413 545 3925; fax: +1 413 545 1027.
E-mail address: perot@ecs.umass.edu (J.B. Perot).

0021-9991/$ - see front matter © 2006 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2006.03.028

keep both the stencil and the number of unknowns small, but use implicit unknowns (and therefore global coupling requiring matrix inversion) to increase the accuracy. Although Padé schemes are highly desirable on one-dimensional stencils (or Cartesian products of one-dimensional stencils), the use of Padé schemes on general stencils is unlikely since cross derivatives become far too numerous as the order increases. The method presented herein uses the finite element approach of more unknowns to obtain higher-order accuracy but otherwise has the basic traits of a finite volume (or discontinuous Galerkin) method. Since this is an entirely new approach to obtaining higher-order accuracy in finite volume methods, the focus in this paper is on the numerical method, and a straightforward demonstration problem (heat transfer) is used.

The use of large stencils in classic higher-order finite volume schemes leads to a number of difficult issues. Boundary conditions become difficult to implement. Either higher derivatives must be known or lopsided stencils must be used. The later are prone to instability. While the formal order may be higher when using a large stencil, the accuracy at practical resolutions is frequently not improved by using a larger stencil. Large stencils become problematic when material properties change rapidly. This is related to the boundary condition issues. When implementing these methods on parallel distributed memory computers, such as the PC clusters, large stencils require a great deal of domain overlap and repeated data communication. On unstructured meshes, large stencils can be expensive and unwieldy to program. They can be implemented implicitly through the repeated action of small local stencils, but the use of repeated small stencils tends to be very cache inefficient.

While there are both practical and performance issues associated with using large stencils, these are not our primary reason for exploring the use of small stencils (and more unknowns). The principal motivation of this work is to obtain high-order discrete operators that behave correctly and can be guaranteed not to cause spurious numerical phenomena. It will be demonstrated that by using more unknowns it is possible to create higher-order discrete operators which are, in some sense, exact.

Unstructured meshes can be automatically generated in arbitrarily complex domains. Using mesh motion, unstructured meshes are easy to adapt anisotropically while maintaining fixed solution cost. The adaptation tends to be smooth compared to Cartesian mesh refinement and unstructured meshes accurately capture complex domain surfaces. The focus is on unstructured meshes in this paper for these reasons and the fact that generalizing unstructured methods to the Cartesian case is fairly trivial whereas the converse is not true. Mimetic finite difference methods on polygonal meshes have been developed by Shashkov et al. [7] and Franco Brezzi et al. [8]. Methods for obtaining high-order mimetic operators on *Cartesian meshes* using the traditional finite volume approach of enlarging the stencil have been developed by Morinishi, Vassiliev, Verstappen and Veldman [9–13]. Our approach to producing mimetic methods combines ideas from both finite volume methods and finite element methods and is appropriate for unstructured meshes.

For simplicity, this paper focuses on the diffusion equation,

$$\frac{\partial(\rho C_v T)}{\partial t} = \nabla \cdot k \nabla T \tag{1}$$

This simple equation allows the emphasis to be placed on the numerical method and the procedure for obtaining higher-order rather than the intricacies of the equation being solved. While the ultimate intent is to use these numerical procedures to discretize the incompressible Navier–Stokes equations [14,15], there are many issues concerning discretization of the Navier–Stokes equations (such as how pressure and the incompressibility constraint are treated [16,17]) that we wish to avoid when outlining the fundamentals of the method.

The diffusion equation (Eq. (1)) occurs in many areas of science and engineering. We will discuss it here in the context of heat conduction since this is perhaps its most familiar physical context, but the actual physical interpretation is not central to this paper. In heat conduction, T is the unknown temperature. The material under investigation determines the conductivity k and the heat capacity ρC_v. In this paper, it is assumed that the mesh is always aligned with material discontinuities. Since the mesh can move this is easy to achieve.

The derivation of the higher-order mimetic scheme is presented in Section 2. This derivation first obtains an exact but finite system of equations and unknowns. The exact system is then closed via some interpolation assumptions which dictate the numerical accuracy but which have no impact on the discrete operators (which are exact). Numerical tests to confirm the accuracy and compare the cost to low-order methods are presented in Section 3. Finally, Section 4 presents a short discussion and some conclusions about the efficacy of this approach.

2. Dual mesh discretization

2.1. Background

The heat equation, like all partial differential equations, is essentially an infinite number of equations (one for every point in space) for an infinite number of unknowns (temperature at every point in space). Since a computer solution must deal with the finite, it is commonly assumed that some approximation (and associated loss of information) must be made in order to turn a partial differential equation (like the example heat equation) into a finite system of equations and unknowns. For this reason, it is usually assumed that discretization (making a PDE into a finite system) involves the introduction of errors. While discretization *usually* does involve the introduction of errors, it does not have to.

In the dual-mesh (or mimetic) method that is described herein the discretization process is exact. All numerical approximation is introduced only where physical approximations are made – in the constitutive equations (not in the calculus). The catch to this remarkable observation that exact discretization is quite possible is that the resulting exact finite system has too many unknowns. While there are a finite number of unknowns, they reside on different meshes and the system is therefore not closed. In dual-mesh methods all numerical approximation occurs in the coupling approximation between the unknowns on the two different meshes. The coupling approximation can either have a finite volume or a finite element character. In this work the focus is on the finite volume flavor of dual-mesh methods. Higher-order finite element dual-mesh methods (for electromagnetics) are discussed among other places in [18–22]. To our knowledge higher-order unstructured finite volume dual-mesh methods have never previously been discussed.

In dual-mesh methods it is important to separate the physics and mathematics from the material assumptions. The heat equation, as it is presented in the form given in Eq. (1), combines and therefore obfuscates these different aspects of the problem. Consider instead the alternative form,

$$\frac{\partial i}{\partial t} = -\nabla \cdot \mathbf{q} \tag{2a}$$

$$\mathbf{q} = -k\mathbf{g} \tag{2b}$$

$$i = \rho C_v T \tag{2c}$$

$$\mathbf{g} = \nabla T \tag{2d}$$

where $\mathbf{q}$ is the heat flux and i is the internal energy. Eq. (2a) contains the physics (energy is conserved). Eq. (2d) is simply mathematics (definition of the gradient). However, Eqs. (2b) and (2c) are constitutive relations. They are by no means 'true'. They are simply physical approximations that are commonly made and which close the system. They happen to be reasonably good assumptions for a wide variety of materials, but they are inventions of humans not properties of mathematics or physics. In the context of heat conduction (2b) is referred to as Fourier's Law, and (2c) is the assumption of a perfectly caloric material. In the dual mesh method, Eqs. (2a) and (2d) will be made finite using exact mathematics. All numerical approximation will then occur in Eqs. (2b) and (2c) – where physical approximation is also being made. The benefit of this approach is that the discrete divergence and gradient operators that result from making Eqs. (2a) and (2d) finite, are exact, and therefore behave in every way like their continuous counterparts. Similar ideas have been reported in the literature [23,24], but the key distinction in this approach is that the current formalism allows exact discrete operators to be derived a priori whereas previous approaches could only confirm such properties existed for a particular method after the method was already derived.

2.2. Lowest-order dual mesh method

We present first the lowest-order method. This will increase the familiarity with the dual-mesh approach before discussing the higher-order case.

2.2.1. Discretization

In the low-order approach Eq. (2a) is integrated over non-overlapping volumes that cover the domain (just like a finite volume method), and Eq. (2d) is integrated over line segments. In particular, the lines

connecting the neighboring control volume centers are used. This gives the following finite system of exact equations.

$$\frac{\partial}{\partial t}\int_{\text{cell}} \mathrm{id}V = -\sum_{\text{cell\_faces}} \int_{\text{face}} \mathbf{q}\cdot\mathbf{n}\,\mathrm{d}A \tag{3a}$$

for every cell, and

$$\int_{\text{n1}}^{\text{n2}} \mathbf{g}\cdot\,\mathrm{d}\mathbf{l} = T_{\text{n2}} - T_{\text{n1}} \tag{3b}$$

for every line segment. In the low-order node (or vertex) based method the volumes surround each vertex of the mesh and the line segments are the edges connecting the mesh vertices.

The system only becomes closed once we relate $\int_{\text{cell}} \mathrm{id}V$ to T_n and $\int_{\text{face}} \mathbf{q}\cdot\mathbf{n}\,\mathrm{d}A$ to $\int_{\text{n1}}^{\text{n2}} \mathbf{g}\cdot\,\mathrm{d}\mathbf{l}$. Note how the necessary numerical approximations mimic the necessary constitutive equations (Eqs. (2b) and (2c)). Also note that all numerical approximation is essentially an interpolation problem. No approximation of differential operators occurs. Fig. 1 illustrates the placement of the computational variables for the lower-order method.

2.2.2. Dual mesh specification

The choice of the dual mesh is one of the many options left to the method designer of dual-mesh methods. In alternative words, how exactly are the volumes surrounding each vertex to be defined? For triangular or tetrahedral meshes the Voronoi dual mesh can be an attractive choice since it is everywhere locally orthogonal to the primary mesh. However, this requires the primary mesh to be a Delaunay triangulation. In addition, the cell centers (circumcenters) using the Voronoi dual are not always within the cell which can cause large numerical errors. In this work, we present numerical results using the median dual mesh which connects cell centroids and face centroids to form the bounding volume around each vertex. However, the method is by no means restricted to this particular choice of the dual volume. It is formulated for any arbitrary polygonal dual mesh.

The choice of whether to use the primary or dual mesh cells is not arbitrary. We will assume that the primary mesh conforms to material boundaries. That is, each primary cell contains a single type of material. The same is not true of dual cells (the volumes surrounding a node). In this work the allocation of material, not which mesh is generated by a mesh generator, is what defines the primary mesh. While traditional control volume methods place the unknowns in the primary cells, this work will focus on methods in which the dual cells are used for the control volumes and the temperature unknown resides at the mesh nodes. The variable $\int_{\text{n1}}^{\text{n2}} \mathbf{g}\cdot\,\mathrm{d}\mathbf{l}$ is then defined along primary mesh edges and $\int_{\text{face}} \mathbf{q}\cdot\mathbf{n}\,\mathrm{d}A$ is defined on the dual mesh faces. Other arrangements, such as a more classic cell based approach are also possible, but are slightly more complex and are not discussed herein.

2.2.3. Interpolation via polynomial reconstruction

Note that there is a one-to-one relationship between primary mesh edges and dual mesh faces. In general, we can therefore write the interpolation approximation as

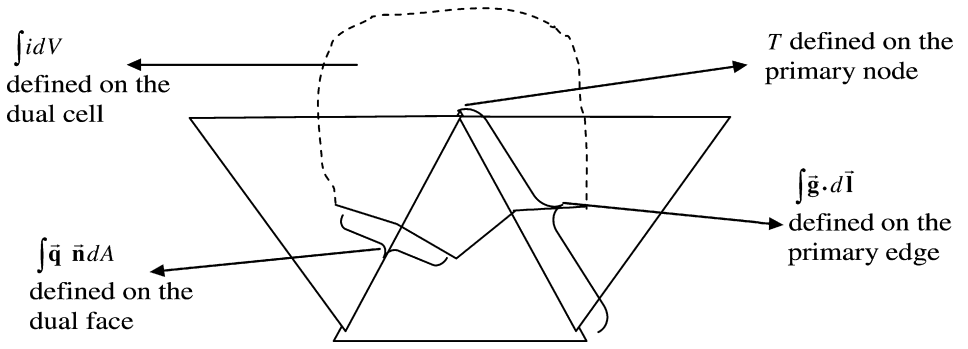

Fig. 1. Placement of variables for the lower order method.

$$\int_{\text{face}} \mathbf{q}\cdot\mathbf{n}\,\mathrm{d}A = -\mathbf{M}_k \int_{\text{n1}}^{\text{n2}} \mathbf{g}\cdot\mathrm{d}\mathbf{l} \tag{4}$$

where $\mathbf{M}_k$ is a square matrix with the same units as the conductivity k. For the Voronoi dual mesh, this matrix (sometimes referred to as a discrete Hodge star operator) has the attractive property of being diagonal. For a general dual mesh, the matrix is not diagonal but it is sparse (with a small stencil) and positive definite.

In this work, the matrix $\mathbf{M}_k$ is never explicitly derived or built. Instead, an explicit procedure for obtaining the dual face heat flux, $\int_{\text{face}} \mathbf{q}\cdot\mathbf{n}\,\mathrm{d}A$ from the line average temperature gradient, $\int_{\text{n1}}^{\text{n2}} \mathbf{g}\cdot\mathrm{d}\mathbf{l}$ is presented. The placement of the unknowns at the mesh nodes (rather than the more traditional cell centers) is akin to the unknowns in a low-order finite element method. The choice stems from the underlying continuity that temperature possesses across material boundaries. Placing the temperature on the nodes (which potentially lie on material interfaces) enforces this continuity on the numerical solution. It is possible to develop mimetic methods in which the unknowns are at the cell centers (publications are currently being prepared) but this approach is not discussed here.

Since the low-order method assumes temperature is given at the mesh nodes, it is natural to assume that the temperature varies linearly within the primary cells (if they are triangles or tetrahedra) or bilinearly (if they are quadrilaterals or hexahedra). As a result the temperature gradient, $\mathbf{g}$, and heat flux $\mathbf{q}$ should be constant within a triangle or tetrahedron (assuming a single material in each cell). Quadrilaterals and hexahedra have slightly more complex but still known functional behavior. This assumption about the functional form of the solution is where all numerical error enters the dual-mesh method.

Once the polynomial form of $\mathbf{g}$ is assumed, it is possible to determine the coefficients in the polynomial from the available data. The number of unknown polynomial coefficients is always chosen to equal the number of unique data values, so that this process is always well defined. On each triangle, there are three pieces of information about $\mathbf{g}$ (one on each of the three edges). However, one is redundant (since $\sum \int_{\text{n1}}^{\text{n2}} \mathbf{g}\cdot\mathrm{d}\mathbf{l} = 0$), leaving two independent pieces of information to determine the constant vector $\mathbf{g}$ in each cell. On a tetrahedron, there are six pieces of information (one for each edge), and three redundancies (four faces with one being redundant), leaving three independent pieces of information to determine the constant vector $\mathbf{g}$ in each cell. This results in a 3×3 matrix. Once $\mathbf{g}$ is determined, calculating the heat flux in each cell is simple, $\mathbf{q} = -k\mathbf{g}$, since we assume a single material exists in each cell. Integrating the constant heat flux over the dual mesh faces to determine $\int_{\text{face}} \mathbf{q}\cdot\mathbf{n}\,\mathrm{d}A$ is also relatively simple.

The determination of the polynomial coefficients of $\mathbf{g}$ based on certain data values requires a matrix inversion in each cell. In three-dimension, the inversion is a 3×3 matrix for a tetrahedra (as detailed in the previous paragraph). Similarly, a 7×7 inversion is necessary for hexahedra. However, for the next order up, this results in a 9×9 inversion for tetrahedra and 19×19 inversion for hexahedra [25]. Both the storage and inversion become very expensive. Another drawback of doing polynomial reconstruction is that a different formulation is necessary for each type of cell (triangle, tetrahedra, hexahedra, prism, etc). This approach can not be applied to arbitrary shaped polygons.

2.2.4. Direct interpolation

In this work, we describe a more direct way to perform the necessary interpolations between the different mesh quantities. This is equivalent to showing that the matrices described above (for the polynomial coefficients) can be inverted explicitly. This approach has the added benefit of being applicable to any polygonal cell type. The inversion of $\mathbf{g}$ starts with the exact relation

$$\int \mathbf{n}\times\mathbf{v}\,\mathrm{d}A = -\sum_{\text{edges}} \int \mathbf{x}\mathbf{v}\cdot\mathrm{d}\mathbf{l} \tag{5}$$

for any vector $\mathbf{v}$. If we assume that $\mathbf{g}$ is constant along edges (which is the case for the standard linear or bilinear polynomial interpolations), then

$$\mathbf{n}\times\int \mathbf{g}\,\mathrm{d}A = -\sum_{\text{edges}} \mathbf{x}_{\text{e}}^{\text{CG}} \int \mathbf{g}\cdot\mathrm{d}\mathbf{l} \tag{6}$$

The right hand side is an explicit function of the given data and the geometry (midpoint position of the edge). In two dimensions, we also assume that $\mathbf{g}$ is a constant plus some terms that are zero when averaged over the cell. In the case of a triangle these extra terms are exactly zero. In 2D we can therefore write

$$\mathbf{z} \times \mathbf{g}_c^{CG} = -\frac{1}{A_c} \sum_{\text{cell\_edges}} \mathbf{x}_e^{CG} \int \mathbf{g} \cdot \mathbf{dl} \tag{7}$$

where $\mathbf{z}$ is the vector pointing out of the 2D plane of interest and the summation assumes the edge orientations are counterclockwise (right hand rule).

If the polynomial function is expanded about the center of gravity, the value of the gradient at the cell center of gravity, $\mathbf{g}_c^{CG}$, is equal to the lowest order (constant) coefficients. Eq. (7) is essentially an explicit inversion formula. More importantly, this formula can be applied to arbitrary 2D polygonal cells. The only assumptions are that $\mathbf{g}$ is constant along the mesh edges and that the average value of $\mathbf{g}$ is equal to the center of gravity value, $\int \mathbf{g}\,dA = \mathbf{g}_c^{CG} A_c$. This formula recovers the standard triangle and quadrilateral interpolations.

If the integration of $\int_{\text{face}} \mathbf{q} \cdot \mathbf{n}\,dA$ is not required to be exact then it can be written as

$$\int_{\text{face}} \mathbf{q} \cdot \mathbf{n}\,dA \approx -\sum_{\text{edge\_cells}} \hat{\mathbf{n}}_f \hat{A}_f \cdot k \mathbf{g}_c^{CG} \tag{8}$$

where $\hat{\mathbf{n}}_f$ and $\hat{A}_f$ are the outward normal and area of the dual mesh faces. This integration assumes that the gradient is constant in each cell. It is therefore not exact for quadrilaterals or hexahedra but also does not introduce any errors that are larger than the original interpolation assumptions. It is therefore consistent with the interpolation error.

In two dimensions and using the median dual mesh it can be shown that the normal to the dual faces is directly related to the edge positions, $\hat{\mathbf{n}}_f \hat{A}_f = \mathbf{z} \times \mathbf{x}_e^{CG}$. In this case, the operation given by Eq. (8) is the transpose of the operation given by Eq. (7). The transformation matrix, $\mathbf{M}_k$, is therefore symmetric (and positive definite) and given by $\mathbf{M}_k = \mathbf{X}^T \frac{k}{A_c} \mathbf{X}$. Note that for the case of a Voronoi dual mesh the transformation matrix is diagonal and even simpler, $\mathbf{M}_k = \frac{k\hat{A}_f}{L_e}$.

In three dimensions we consider first the tetrahedral case. Using the identity,

$$\int \nabla T\,dV = -\frac{1}{ND-1} \sum_{\text{faces}} \int \mathbf{x} \times \mathbf{n} \times \nabla T\,dA \tag{9}$$

where ND is the number of dimensions. Then using Eq. (6) and the fact that $\mathbf{g}$ is constant in tetrahedra gives,

$$\mathbf{g}_c^{CG} V_c = \frac{1}{ND-1} \sum_{\text{faces}} \mathbf{x}_f^{CG} \times \sum_{\text{edges}} \mathbf{x}_e^{CG} \int \mathbf{g} \cdot \mathbf{dl} = \sum_{\text{cell\_edges}} \hat{\mathbf{n}}_f \hat{A}_f \int \mathbf{g} \cdot \mathbf{dl} \tag{10}$$

This equation is the 3D equivalent of Eq. (7). Eq. (10) is the more general formulation and can also be applied in 2D. With some algebra, it can be shown that this formula also applies for Cartesian mesh hexahedral cells (even though $\mathbf{g}$ is no longer constant). We will simply assume that some polynomial functions must exist such that it also holds for arbitrary polygons. As in the 2D case the resulting transformation matrix is symmetric (and positive definite).

The advantage of this approach is the significant savings in cost and storage that are achieved by performing the inversion explicitly, as well as the ability to easily generalize the formulas to arbitrary polygons.

2.2.5. Unsteady term

The transformation from temperature to internal energy (Eq. (2c)), that must be made in the unsteady term is similar though somewhat simpler. Again, an assumption about how the temperature varies within each cell must be made. It is not clear at this time, if this assumption must be consistent with the previous assumptions about the heat flux. If the temperature is assumed to be linear within triangles or tetrahedra, then the internal energy is also linear but discontinuous between cells (because the material properties can change between cells). The integral $\int_{\text{cell}} i\,dV$ can then be calculated in each dual mesh cell using appropriate order Gauss quadrature. It appears that exact Gauss quadrature is again not necessary. In the low-order case, only a formula sufficient to integrate linear functions is sufficient even though the 3D hexahedra have up to cubic terms. This

level of accuracy is still consistent with the interpolation error. The result of the integration is that the time derivative term will have a mass matrix involving nearest neighbors associated with it. This mass matrix is not the same as the finite element mass matrix, but is similar and has the same sparsity structure. The presence of a mass matrix is fundamentally appropriate for an unsteady diffusion equation since it forces the solution to be fully coupled (even if the diffusion term is computed explicitly). Physical solutions of the diffusion equation have the same coupled (parabolic) behavior. Unsteady solutions will not be tested in this paper (since the focus is on the higher-order spatial discretizations) but it is important to see that the basic method is by no means restricted to steady state.

2.3. Higher-order dual mesh method

Higher-order dual mesh methods can be constructed by increasing the number of unknowns and equations. Typically, the lower-order unknowns and equations are retained in the higher-order method. This is very useful. It allows the implementation of multiscale models, error extrapolation, and the possibility of local p-refinement of solutions (where the order of approximation is changed rather than the mesh size).

2.3.1. Discretization

For the node-based mimetic dual mesh scheme the fundamental unknowns at the next higher order are the nodal values of the temperature T_n (as in the low-order case) and the edge integral $\int T\,\mathrm{d}\ell$ (Fig. 2).

In the discussion that follows the letters f, c, and e refer to the face centroid, cell centroid and edge centroid (midpoint of the edge) respectively. Fig. 3 shows the situation in 2D and 3D. Note that in 2D, edges and faces can be identical structures. In any dimension, we always assume edges connect the nodes and faces bound the cells. The normal to the dual face, $\hat{\mathbf{n}}_f$, always points out of the dual cell away from node N. Only a small portion of the dual mesh is shown in 3D to keep the figure legible.

In this section, we consider the modification of the energy equation (Eq. (3a)) to include arbitrary source terms.

$$\frac{\partial}{\partial t}\int_{\text{cell}} i\,\mathrm{d}V = -\sum_{\text{cell\_faces}}\int_{\text{face}} \mathbf{q}\cdot\mathbf{n}\,\mathrm{d}A + \int_{\text{cell}} S\,\mathrm{d}V \tag{11}$$

As in the low-order method, this equation is applied on the dual cells surrounding each node. This provides one evolution equation for each node unknown.

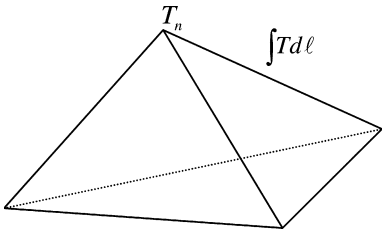

Fig. 2. Position of the node and edge unknowns for the third order dual-mesh method.

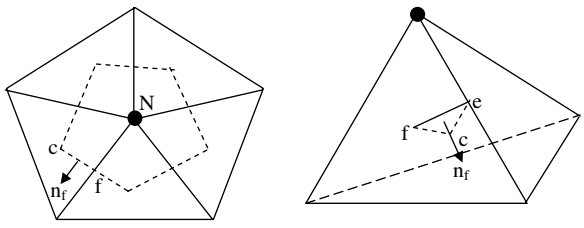

Fig. 3. Unified notational scheme for 2D and 3D meshes.

2.3.2. Edge evolution equation in 2D

For the higher-order method, the edge unknowns also require an evolution equation. Following the example of the low-order method we propose applying Eq. (11) on each dual face as well. This is essentially an infinitely thin cell which we compute by taking a finite thin cell surrounding the dual faces and then taking the limit as the cell width goes to zero. Fig. 4 shows a diagram of the situation in two dimensions. The important observation is that the perpendicular distance across the thin dual face cell is different in different cells and depends on the angle that the dual face makes with the edge.

For ease of presentation, the edge control volume is divided into three subvolumes (V'_1, V'_2 and V'_3) as shown in Fig. 4. Then, the diffusion term of Eq. (11) can be written as

$$\int_{\text{edge\_vol}} \nabla \cdot \mathbf{q}\, \mathrm{d}V = \int_{\text{edge\_cell1}} \nabla \cdot \mathbf{q}\, \mathrm{d}V'_1 + \int_{\text{edge\_cell2}} \nabla \cdot \mathbf{q}\, \mathrm{d}V'_2 + \int_{\text{edge\_face}} \nabla \cdot \mathbf{q}\, \mathrm{d}V'_3 \tag{12a}$$

With the assumption that the divergence of the heat flux is constant within each cell, the integral of $\nabla \cdot \mathbf{q}$ over the sub-volume V'_1 is given by

$$\int_{\text{edge\_cell1}} \nabla \cdot \mathbf{q}\, \mathrm{d}V'_1 = [\nabla \cdot \mathbf{q}]|^{\text{c1}} A'_1\, \mathrm{d}x'_1 = [\nabla \cdot \mathbf{q}]|^{\text{c1}} A'_1 |\hat{\mathbf{r}}_1 \cdot \hat{\mathbf{n}}_{\text{f}}|\, \mathrm{d}x = [\nabla \cdot \mathbf{q}]|^{\text{c1}} |\vec{\mathbf{r}}_1 \cdot \hat{\mathbf{n}}_{\text{f}}|\, \mathrm{d}x \tag{12b}$$

In Eq. (12b) $\vec{\mathbf{r}}_1 = \vec{\mathbf{x}}_{\text{f}} - \vec{\mathbf{x}}_{\text{c1}}$ and $\hat{\mathbf{n}}_{\text{f}}$ is the unit normal to the primary face (or edge, in 2D). A similar expression can be written for the edge cell 2. At the edge face (sub-volume V'_3) the flux may not be continuous across primary mesh faces, so the divergence there may be a delta function. In order to account for this, the integral of $\nabla \cdot \mathbf{q}$ over the edge face is given by Gauss' theorem,

$$\int_{\text{edge\_face}} \nabla \cdot \mathbf{q}\, \mathrm{d}V'_3 = \mathbf{q}_{\text{f}}^{\text{c2}} \cdot \hat{\mathbf{n}}_{\text{f}}\, \mathrm{d}x - \mathbf{q}_{\text{f}}^{\text{c1}} \cdot \hat{\mathbf{n}}_{\text{f}}\, \mathrm{d}x \tag{12c}$$

This term will tend to drive the solution to a state where the flux is continuous in this weak sense. In Eq. (12c), $\mathbf{q}_{\text{f}}^{\text{c1}}$ and $\mathbf{q}_{\text{f}}^{\text{c2}}$ are the reconstructions of the heat flux vectors evaluated at the primary face at cells c1 and c2. When Eq. (12) are substituted into Eq. (11), in the limit of $\mathrm{d}x \to 0$, the edge evolution equation in two dimensions becomes,

$$\sum_{\text{edge\_cells}} \left(|\vec{\mathbf{r}} \cdot \hat{\mathbf{n}}_{\text{f}}| \frac{\partial i}{\partial t}\right) = \sum_{\text{edge\_cells}} [|\vec{\mathbf{r}} \cdot \hat{\mathbf{n}}_{\text{f}}|(S - \nabla \cdot \mathbf{q})] + (\mathbf{q}_{\text{f}}^{\text{c2}} \cdot \hat{\mathbf{n}}_{\text{f}} - \mathbf{q}_{\text{f}}^{\text{c1}} \cdot \hat{\mathbf{n}}_{\text{f}}) \tag{13}$$

2.3.3. Edge evolution equation in 3D

Fig. 5 presents the edge control volume in three dimensions. Only a part of the edge control volume is presented for clarity.

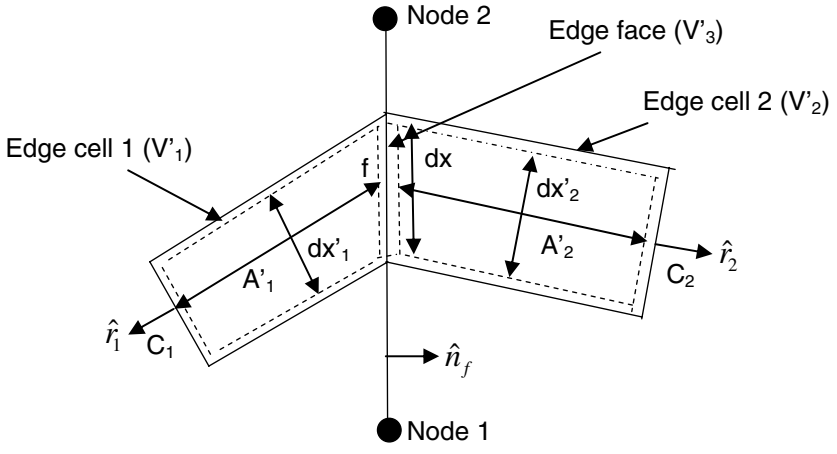

Fig. 4. Two-dimensional representation of the edge control volume.

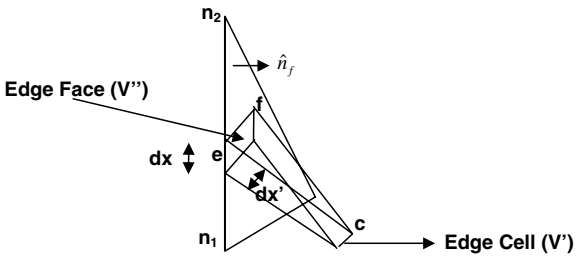

Fig. 5. Three-dimensional representation of the edge control volume.

Analogous expressions to Eqs. (12b) and (12c) are presented for the 3D case,

$$\int_{\text{edge\_cell}} \nabla \cdot \mathbf{q}\, \mathrm{d}V' = [\nabla \cdot \mathbf{q}]|^{\mathrm{c}} V_1' = [\nabla \cdot \mathbf{q}]|^{\mathrm{c}} \frac{\mathrm{d}x}{2} \vec{\mathbf{r}} \cdot (\vec{\mathbf{r}}_{\mathrm{fe}} \times \hat{\mathbf{t}}_{\mathrm{e}})| \tag{14a}$$

$$\int_{\text{edge\_face}} \nabla \cdot \mathbf{q}\, \mathrm{d}V'' = (\mathbf{q}_{\mathrm{f}}^{\mathrm{c2}} \cdot \hat{\mathbf{n}}_{\mathrm{f}} - \mathbf{q}_{\mathrm{f}}^{\mathrm{c1}} \cdot \hat{\mathbf{n}}_{\mathrm{f}}) |\vec{\mathbf{r}}_{\mathrm{fe}} \times \hat{\mathbf{t}}_{\mathrm{e}}|\, \mathrm{d}x \tag{14b}$$

In Eq. (14), $\vec{\mathbf{r}}_{\mathrm{fe}} = \vec{\mathbf{x}}_{\mathrm{f}} - \vec{\mathbf{x}}_{\mathrm{e}}$, $\hat{\mathbf{t}}_{\mathrm{e}} = (\vec{\mathbf{x}}_{\mathrm{n2}} - \vec{\mathbf{x}}_{\mathrm{n1}})/|\vec{\mathbf{x}}_{\mathrm{n2}} - \vec{\mathbf{x}}_{\mathrm{n1}}|$, $\vec{\mathbf{r}} = \vec{\mathbf{x}}_{\mathrm{f}} - \vec{\mathbf{x}}_{\mathrm{c}}$ and $\hat{\mathbf{n}}_{\mathrm{f}}$ is the unit normal to the primary face. When Eqs. (14) are substituted into Eq. (11), in the limit of $\mathrm{d}x \to 0$, the edge evolution equation in three dimensions becomes,

$$\sum_{\text{edge cells}} \left(\left| \frac{1}{2} \vec{\mathbf{r}} \cdot (\vec{\mathbf{r}}_{\mathrm{fe}} \times \hat{\mathbf{t}}_{\mathrm{e}}) \right| \frac{\partial i}{\partial t} \right) = \sum_{\text{edge cells}} \left[\left| \frac{1}{2} \vec{\mathbf{r}} \cdot (\vec{\mathbf{r}}_{\mathrm{fe}} \times \hat{\mathbf{t}}_{\mathrm{e}}) \right| (S - \nabla \cdot \mathbf{q}) \right] + \sum_{\text{edge faces}} (\mathbf{q}_{\mathrm{f}}^{\mathrm{c1}} \cdot \hat{\mathbf{n}}_{\mathrm{f}} - \mathbf{q}_{\mathrm{f}}^{\mathrm{c2}} \cdot \hat{\mathbf{n}}_{\mathrm{f}}) |\vec{\mathbf{r}}_{\mathrm{fe}} \times \hat{\mathbf{t}}_{\mathrm{e}}| \tag{15a}$$

For a Voronoi dual mesh the cross-products and dot products are trivial. However, in this work we derive the general form of the equations.

The additional Eq. (15a) requires additional unknowns compared to the low-order method. In particular, the right hand side now requires the divergence of the heat flux in each cell, and the heat flux normal to primary faces. In the low-order method, the heat flux was constant (in simplices) and therefore the divergence would be zero in cells. But now, the divergence exists and will be interpolated.

Before discussing the interpolation procedure, we must discuss the additional exact integral expressions corresponding to Eq. (3b) in the low-order method. With the additional unknown $\int T \mathrm{d}\ell$, we can also write the exact expressions,

$$\int_{\mathrm{n1}}^{\mathrm{n2}} \mathbf{x}\mathbf{g} \cdot \mathrm{d}\mathbf{l} = (\mathbf{x}_{\mathrm{n2}} T_{\mathrm{n2}} - \mathbf{x}_{\mathrm{n1}} T_{\mathrm{n1}}) - \mathbf{t}_{\mathrm{e}} \int_{\mathrm{n1}}^{\mathrm{n2}} T\, \mathrm{d}l \tag{15b}$$

$$\int_{\mathrm{face}} \mathbf{n} \times \mathbf{g}\, \mathrm{d}A = \sum_{\mathrm{edges}} \mathbf{t}_{\mathrm{e}} \int T\, \mathrm{d}l \tag{15c}$$

These state that the moment of the gradient along a primary edge (or the second derivative of the temperature along the edge) and the average gradient in the plane of a primary face can both be obtained exactly from the primary unknowns. It is this data (along with the low order $\int \mathbf{g} \cdot \mathrm{d}\mathbf{l}$ from Eq. (3b)) that is used to reconstruct the heat flux vector in each cell. Note that not all the information provided by these expressions is independent. Redundancies are given by the exact expressions,

$$\sum_{\mathrm{edges}} \int \mathbf{g} \cdot \mathrm{d}\mathbf{l} = 0 \tag{16a}$$

$$\sum_{\mathrm{edges}} \int \mathbf{x}\mathbf{g} \cdot \mathrm{d}\mathbf{l} = -\int \mathbf{n} \times \mathbf{g}\, \mathrm{d}A \tag{16b}$$

$$\sum_{\mathrm{faces}} \int \mathbf{n} \times \mathbf{g}\, \mathrm{d}A = 0 \tag{16c}$$

2.3.4. Interpolation

On tetrahedra we assume quadradic temperature variation in cells and a linear temperature gradient and linear heat flux. Direct solution for the polynomial coefficients would require an expensive matrix inversion. However, an explicit inversion process is again possible and makes the method highly efficient. Assuming **g** varies linearly along each edge, $\int \mathbf{g}\cdot d\mathbf{l}$ and $\int \mathbf{x}\mathbf{g}\cdot d\mathbf{l}$ can be used to determine $\mathbf{g}\cdot\mathbf{t}$ at the end of each edge. Where the three edges meet at the corner of a 3D polyhedra (or two edges meet at the corner of a 2D polygon) this is sufficient information to reconstruct the entire vector at that location,

$$\begin{bmatrix} t_{1x} & t_{1y} & t_{1z} \\ t_{2x} & t_{2y} & t_{2z} \\ t_{3x} & t_{3y} & t_{3z} \end{bmatrix} \begin{Bmatrix} g_{nx} \\ g_{ny} \\ g_{nz} \end{Bmatrix} = \begin{Bmatrix} g\cdot t_{e1} \\ g\cdot t_{e2} \\ g\cdot t_{e3} \end{Bmatrix} \tag{17}$$

where 1, 2, and 3 refers to the three edges, g_n refers to the gradient at the node and t_e refers to the tangent to edge vector at the three edges.

Once **g** at the cell corners is obtained, it is possible to average two corners to get the **g** value at the cell edges, and even possible to average edges to get face values, and faces to get cell values. The averaging assumes linearity in **g** and so this particular explicit inversion may only be applicable to simplices. The face values are sufficient to compute the heat flux divergence in each cell (using Gauss divergence theorem). Finally, these discrete values of the heat flux are sufficient to compute the integrals (using simple quadrature rules) that are found in the evolution equations (Eqs. (3a) and (15a)).

Specifying Dirichlet boundary conditions is straightforward. Values are specified at the boundary nodes and the boundary edges for the Dirichlet boundary condition. Neumann boundary conditions may be specified by specifying $\mathbf{q}\cdot\mathbf{n}$ on the boundary faces in the evolution equations (Eqs. (3a) and (15a)).

3. Numerical tests of accuracy and cost

Two example problems from Shashkov [26] are considered to illustrate that the method is exact for linear functions and when material properties are discontinuous. The third numerical test proves that the method is exact for quadratic functions. In order to test the accuracy of the method, the order of convergence is plotted in a fourth test problem which considers a cubic function. The computational cost, in terms of CPU time, required to obtain a desired accuracy is plotted as a function of the error norm. Finally, diffusion through a complex geometry (a crank shaft) is considered and the computational cost of the higher-order method is compared against the lower-order method in order to confirm the results established by the previous tests in a realistic problem configuration.

In this work the discrete L_2 error norm is sometimes adopted for verifying the order of convergence of the method where,

$$L_2 = \left[\frac{1}{NN}\sum_{n=1}^{NN}(T_n - T_n^{exact})^2\right]^{1/2} \tag{18}$$

In the above expressions, NN refers to the total number of nodes in the domain, T_n refers to the numerical solution and T_n^{exact} refers to the analytical solution at the nodes. This error norm is discrete in nature and compares the error only at the nodal points where the solution is obtained. However, since our method contains unknowns at nodes as well as edges and the edge unknowns are really an integral averaged quantity, a continuous error norm is also adopted (similar to finite element error norms), which measures the integral error over the whole domain. The continuous error norm is denoted as L_{2C}

$$L_{2C} = \left[\frac{1}{V}\int_V (T - T^{exact})^2\, dV\right]^{1/2} \tag{19}$$

where V refers to the volume of the entire domain. However, in order to evaluate Eq. (19), a quadrature rule needs to be employed that is more accurate than the numerical method, so that the error introduced by the numerical integration is insignificant. A third-order quadrature rule for tetrahedra is employed.

$$\int_V (T - T^{\text{exact}})^2 \, \mathrm{d}V = \sum_{\text{cells}} \left[\frac{1}{40} \sum_{\text{nodes}} (T_{\text{n}} - T_{\text{n}}^{\text{exact}})^2 + \frac{9}{40} \sum_{\text{faces}} (T_{\text{f}} - T_{\text{f}}^{\text{exact}})^2 \right] V_{\text{cell}} \tag{20}$$

$$T_{\text{f}} = \frac{2}{3} \sum_{\text{edges}} \left(\frac{1}{L_{\text{e}}} \int T \, \mathrm{d}l \right) - \frac{1}{3} \sum_{\text{nodes}} T_{\text{n}} \tag{21}$$

In Eq. (20), the square of the errors at the nodes and faces within each cell are summed and weighted by the volume of the cell, and the result summed for all the cells in the domain. The face value is obtained in Eq. (21) by summing over all the nodes and edges that belong to the face in question. Eqs. (19)–(21) together define the continuous error norm L_{2C}. Both the discrete error norm L_2 (Eq. (18)) and L_{2C} will be employed to present the results in the following sections.

3.1. Discontinuous conductivity

The first test problem involves steady diffusion in a square domain with a discontinuous diffusion coefficient, k

$$k = \begin{cases} k_1 & 0 < x < 0.5 \\ k_2 & 0.5 < x < 1 \end{cases} \tag{22}$$

The mesh employed is shown in Fig. 6. The mesh is divided into two different materials with different diffusivities along the interface $x = 0.5$. Note that the discontinuity in the material is captured by the mesh.

For this problem Dirichlet boundary conditions are applied on the left and right boundaries and homogeneous Neumann boundary conditions (symmetry) are applied at the top and bottom boundaries.

$$\begin{aligned} x &= 0 \quad T = \frac{8.0}{18.5} \\ x &= 1 \quad T = \frac{10.5}{18.5} \\ y &= 0 \quad \frac{\partial T}{\partial y} = 0 \\ y &= 1 \quad \frac{\partial T}{\partial y} = 0 \end{aligned} \tag{23}$$

There are no source terms and hence this problem has a piecewise linear solution, with a continuous temperature and heat flux at the interface $x = 0.5$. The exact steady state solution of this problem is

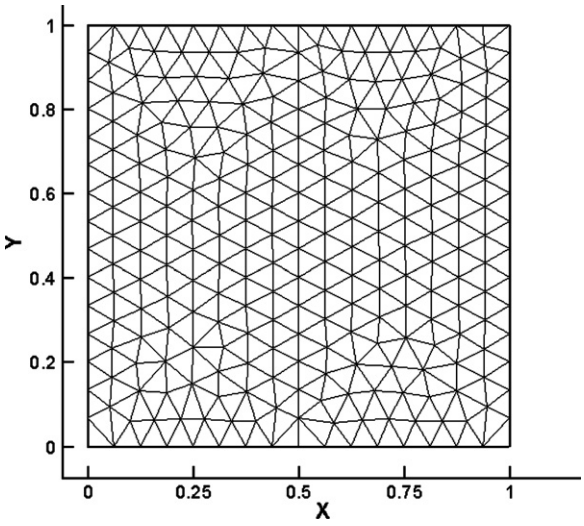

Fig. 6. Mesh with different diffusivities on either side of the interface (at $x = 0.5$).

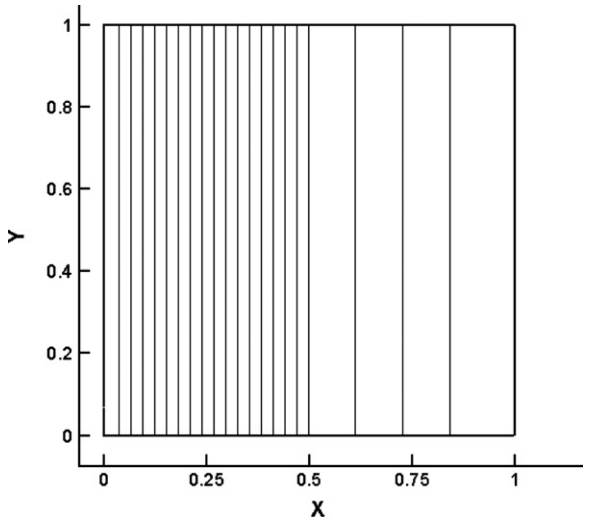

Fig. 7. Isolines of solution for the discontinuous coefficient problem.

$$T = \begin{cases} \frac{k_2 x + 2k_1 k_2}{0.5(k_1+k_2)+4k_1 k_2} & 0 < x < 0.5 \\ \frac{k_1 x + 2k_1 k_2 + 0.5(k_2-k_1)}{0.5(k_1+k_2)+4k_1 k_2} & 0.5 < x < 1 \end{cases} \tag{24}$$

The numerical experiments use $k_1 = 1$ and $k_2 = 4$. The isolines of the solution are presented in Fig. 7.

As expected, the isolines are perfect straight lines and the method achieves the exact answer to machine precision.

3.2. Discontinuous conductivity at an angle

The second problem is taken from Shashkov [26] and Morel et al. [27]. Although the theory for discontinuous coefficients only implies that the normal component of heat flux should be continuous, many numerical methods also assume that tangential flux components are continuous at a discontinuity. Such methods will have difficulties when solving for conduction that occurs at an angle to the discontinuity.

The same mesh (Fig. 6) as in the previous example is considered and the diffusion coefficients are defined as before. Dirichlet boundary conditions are enforced such that the exact steady state solution is

$$T = \begin{cases} a + bx + cy & 0 \leqslant x \leqslant 0.5 \\ a - b\frac{k_1-k_2}{2k_2} + b\frac{k_1}{k_2}x + cy & 0.5 < x \leqslant 1 \end{cases} \tag{25}$$

This problem has a discontinuity in the tangential flux at the material interface. The normal component of the flux (bk_1) is the same across the entire domain. However, the tangential flux component is $k_1 c$ on the left side and $k_2 c$ on the right side of the interface. The numerical experiments employ $a = b = c = 1$. The Dirichlet boundary conditions are applied to the boundaries as shown below.

$$\begin{aligned} &x = 0 \quad T = 1 + y \\ &x = 1 \quad T = \frac{7}{2} + y \\ &y = 0,\ 0 < x < 0.5 \quad T = 1 + x \\ &y = 1,\ 0 < x < 0.5 \quad T = 2 + x \\ &y = 0,\ 0.5 \leqslant x < 1 \quad T = 4x - 0.5 \\ &y = 1,\ 0.5 \leqslant x < 1 \quad T = 4x + 0.5 \end{aligned} \tag{26}$$

The calculated temperature isolines for this problem are shown in Fig. 8. The solution agrees with the exact answer to machine precision.

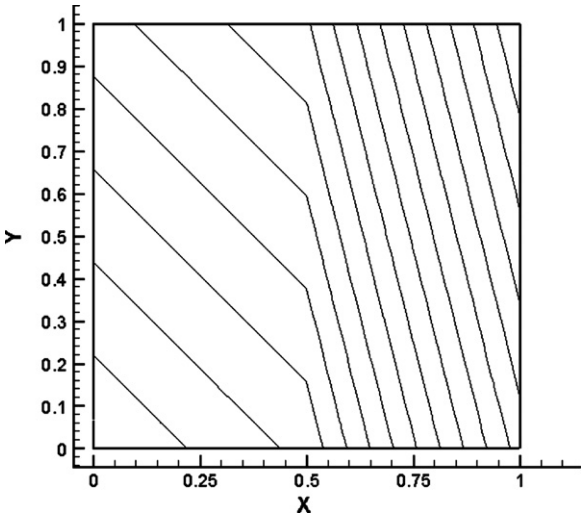

Fig. 8. Isolines of temperature for the discontinuous tangential flux problem.

3.3. Quadratic solution

In the third test problem a uniform source term $S = -4$ is imposed with unit conductivity. Homogeneous Dirichlet boundary conditions are imposed on the left and right boundaries, and homogeneous Neumann boundary conditions are imposed on the top and bottom boundaries (and the front and back boundaries in 3D). The exact solution $T(x) = 2x^2 - 2x$ is a quadratic. The mesh is shown for a 2D case and a 3D case in Fig. 9. For the 3D case, only a slice of the mesh is shown so that it can be clearly visualized. The higher order dual-mesh solution for the 2D and the 3D problems are shown in Fig. 10. The isolines are perfect straight lines and the results again match the analytical solution to machine precision.

3.4. Test of convergence

A cubic solution is now considered in order to demonstrate the order of convergence of the numerical method. The source term is now linear, $S = 4 - 24x$. When homogeneous Dirichlet boundary conditions are applied on the left and right and homogeneous Neumann boundary conditions are applied at the top

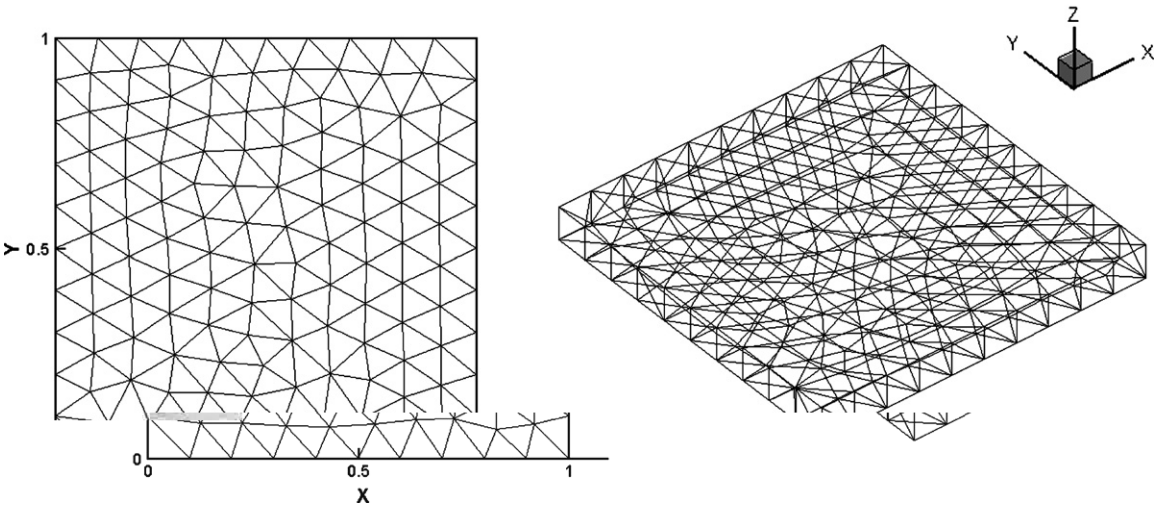

Fig. 9. 2D mesh and 3D mesh slice for testing of the uniform source problem.

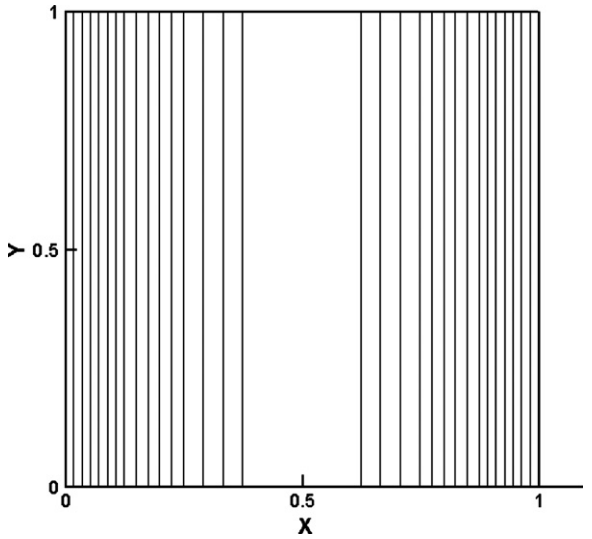

Fig. 10. The higher order dual-mesh solution for the uniform source problem.

and bottom, the exact solution is $T(x) = 4x^3 - 2x^2 + x + 1$. The integral source terms appearing in Eqs. (11) and (12a) are computed exactly (which is simple since the source is linear).

The mesh is shown in Fig. 11. The results for the L_2 and L_{2C} errors at various mesh resolutions are presented for both the lower order and the higher-order methods in Fig. 12. The mesh resolution is characterized by $dx = (\mathrm{Vol}/\mathrm{NC})^{1/3}$, where Vol is the entire domain volume and NC is the number of primary cell volumes (tetrahedra). Fig. 12 suggests the higher-order method is third-order accurate and also that more than an order of magnitude of improvement in accuracy is achieved when compared to the lower-order method even for a very coarse mesh.

Fig. 12 also compares the discrete (L_2) and the continuous (L_{2c}) error norms for the lower-order and higher-order methods. It is seen that while the discrete L_2 error is substantially lower than the continuous L_{2c} error for the lower-order method, they are of comparable magnitude for the higher-order method. This suggests that, for the lower-order method, the error is less at the nodes (where the unknowns are stored) and tends

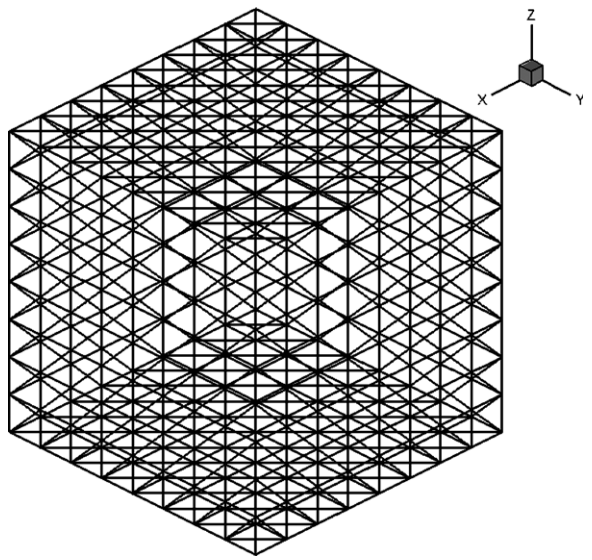

Fig. 11. 3D unit mesh employed for the convergence study of dual-mesh methods.

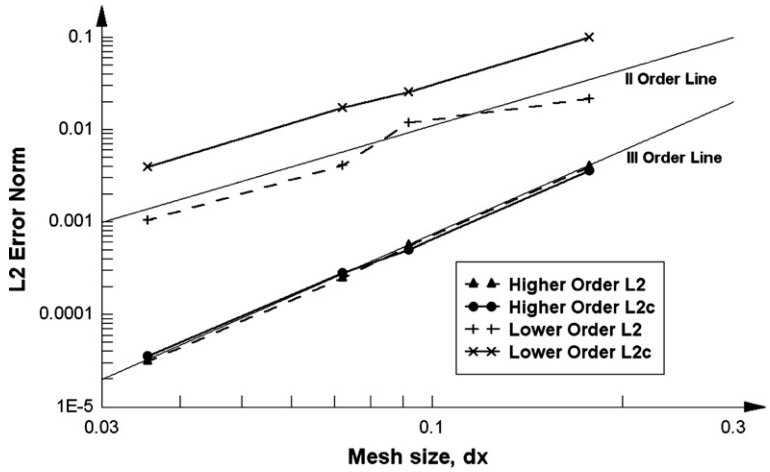

Fig. 12. Convergence of the dual-mesh methods for the linear source problem.

to be higher at all other parts of the domain. However, for the higher-order method, the presence of the edge unknown makes the solution more accurate within the domain, and the discrete error norm is an excellent proxy for the harder to compute continuous norm.

3.5. Computational cost

More important than the order of accuracy is the computational cost required to obtain a certain level of accuracy. This is studied by plotting the error against the CPU time taken per explicit time step (Fig. 13). The problem considered is the same as in the previous section. Fig. 13 again plots both the discrete (L_2) and continuous (L_{2c}) error norms. It can be seen that the higher-order method always proves to be more cost effective than the lower-order method for any desired accuracy level.

3.6. Diffusion through a crank shaft

In this section, a more realistic problem is considered, which involves solving Eq. (2) on a complex geometry. A typical mesh considered for the analysis is shown in Fig. 14. The coarsest mesh considered has 864 nodes and 2339 cells and the finest mesh contains 73875 nodes and 360512 cells. Fixed temperature (Dirichlet)

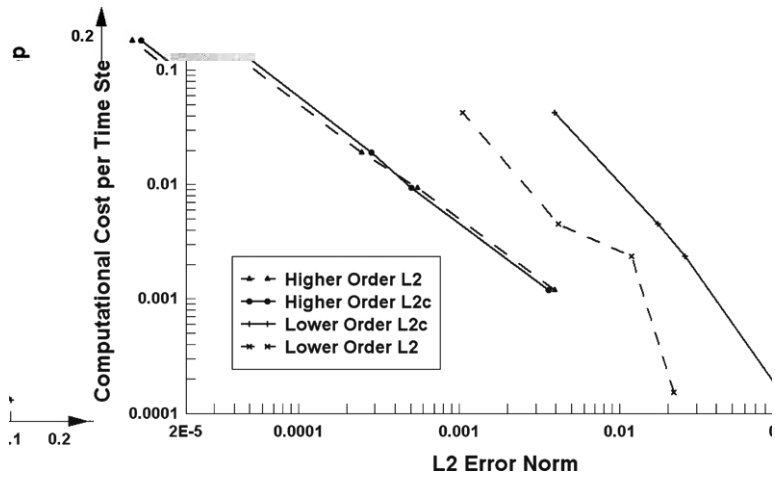

Fig. 13. Computational cost for a desired accuracy for the linear source problem.

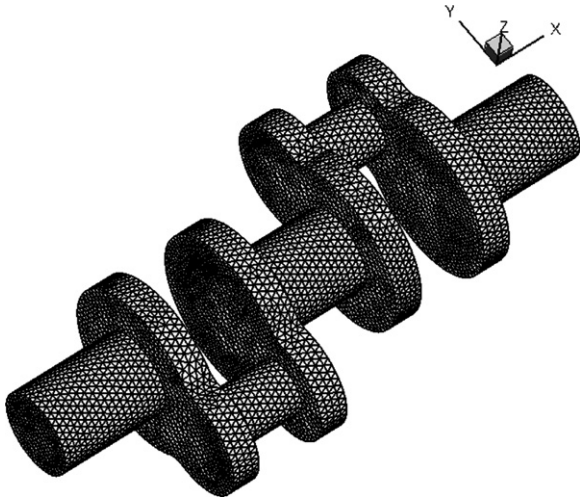

Fig. 14. Crank shaft mesh.

boundary conditions are applied to the inlet and outlet faces (crankshaft ends) and the sides are insulated. Typical temperature contours are presented in Fig. 15.

The heat flux through the inlet and outlet faces, which was verified to be equal, are measured and plotted against the mesh size for the lower-order and the higher-order methods (Fig. 16). The mesh size dx is computed as the cube root of the average cell volume.

The two curves in Fig. 16 are extrapolated in order to determine the exact heat flux, which is then employed to compute the percentage error in the lower-order and higher-order methods. The computational time taken per solver iteration is then plotted against this percentage error, which gives the cost required to obtain a certain accuracy level (Fig. 17). It is inferred from Fig. 17 that the higher-order method is an order of magnitude less expensive for any desired accuracy level, which is in agreement with the results of the previous section.

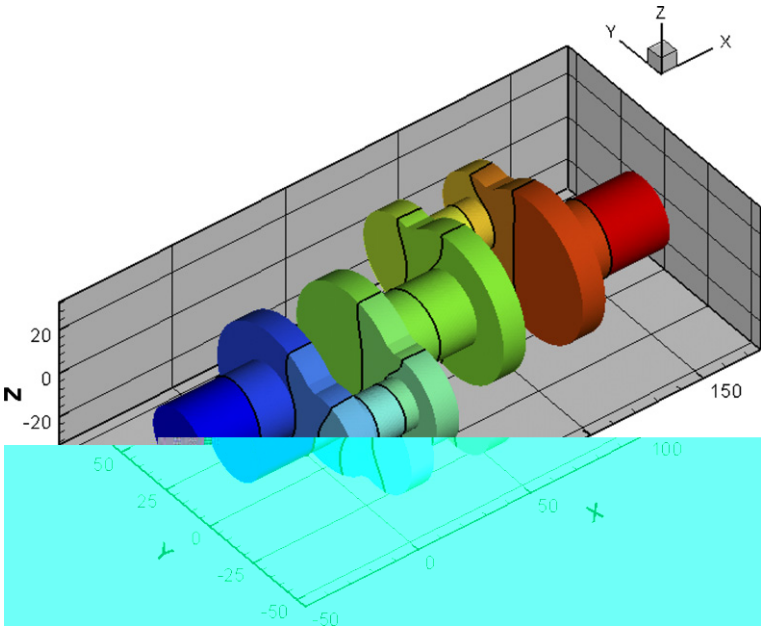

Fig. 15. Temperature contours along the crank shaft.

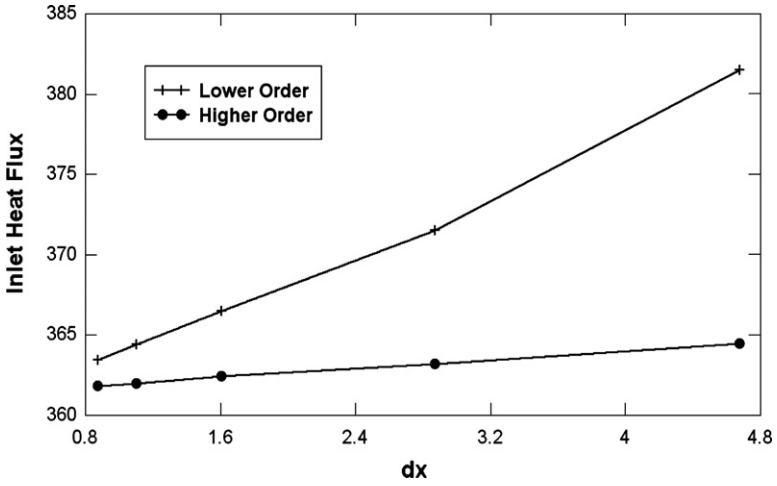

Fig. 16. Convergence of the second order and third order dual-mesh methods for the crank shaft test case.

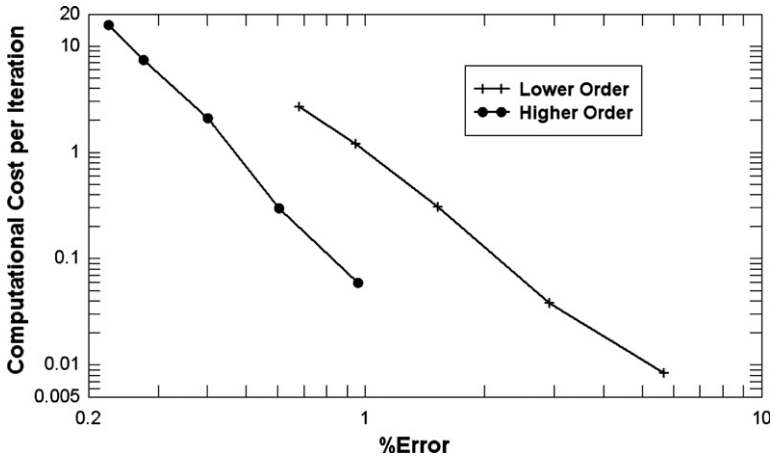

Fig. 17. Cost for a desired accuracy for the crank shaft test case.

4. Discussion

A general method for developing mimetic methods is presented. The key is to exactly discretize the equations before making any approximations. This means all the discrete differential operators are still exact and mimic the mathematical properties of the continuous differential operators. All approximation is then made in the algebraic constitutive material equation (Fourier's Law in the example problem) where physical approximation has already been performed.

Having developed this straightforward method for generating mimetic discretizations it is shown that this paradigm can be used to develop higher-order mimetic methods. The third-order case is discussed in detail within the paper but there are no restrictions to obtaining arbitrarily high order with this approach. The proposed approach to obtaining higher order uses more unknowns per mesh cell (like a finite element method) but the resulting discretization is like a finite volume (or discontinuous Galerkin) method in its ability to maintain a local conservation statement. Tests of the method demonstrate its order of accuracy and its ability to accurately capture solutions with sharp discontinuities in the material properties.

Acknowledgments

We gratefully acknowledge the partial financial support of this work by the Office of Naval Research (Grant Number N00014-01-1-0267), the Air Force Office of Scientific Research (Grant Number FA9550-04-1-0023), and the National Science Foundation (Grant Number CTS-0522089).

References

[1] J.B. Perot, R. Nallapati, A moving unstructured staggered mesh method for the simulation of incompressible free-surface flows, J. Comput. Phys. 184 (2003) 192–214.
[2] R. Nallapati, J.B. Perot, Numerical simulation of free surface flows using a moving mesh, in: Proceedings of the 2000 American Society of Mechanical Engineers, Fluids Engineering Summer Conference, 2000.
[3] T.J.R. Hughes, L. Mazzei, K.E. Janson, Large Eddy simulation and the variational multiscale method, Comput. Visual. Sci. 3 (2000) 47–59.
[4] S. Ghosal, An analysis of numerical errors in large Eddy simulations of turbulence, J. Comput. Phys. 125 (1996) 187.
[5] T.J. Barth, P.O. Frederickson, Higher Order Solution of the Euler Equations on Unstructured Grids Using Quadratic Reconstruction, AIAA 90-0013, 1990.
[6] T.J. Barth, Recent Improvements in High Order K-exact Reconstruction on Unstructured Meshes, AIAA 93-0668, 1993.
[7] Y. Kuznetsov, K. Lipnikov, M. Shashkov, Mimetic Finite Difference Method on Polygonal Meshes, LA-UR-03-7608, 2003.
[8] Franco Brezzi, Konstantin Lipnikov, Valeria Simoncini, A family of mimetic finite-difference methods on polygonal and polyhedral meshes, Math. Models Methods Appl. Sci. 15 (2005) 1533–1553.
[9] Y. Morinishi, T.S. Lund, O.V. Vasilyev, P. Moin, Fully conservative higher order finite difference schemes for incompressible flow, J. Comput. Phys. 143 (1998) 90–124.
[10] O.V. Vasilyev, High order finite difference schemes on non-uniform meshes with good conservation properties, J. Comput. Phys. 157 (2000) 746–761.
[11] R. Verstappen, A. Veldman, A fourth order finite volume method for direct numerical simulation of turbulence at higher Reynolds numbers, Computational Fluid Dynamics, J. Wiley & Sons, 1996, pp. 1073–1079.
[12] R. Verstappen, A. Veldman, Direct numerical simulation of turbulence at lower costs, J. Eng. Math. 32 (1997) 143–159.
[13] R.W.C.P. Verstappen, A.E.P. Veldman, Symmetry-preserving discretization of turbulent flow, J. Comput. Phys. 187 (2003) 343–368.
[14] J.B. Perot, Conservation properties of unstructured staggered mesh schemes, J. Comput. Phys. 159 (2000) 58–89.
[15] X. Zhang, D. Schmidt, J.B. Perot, Accuracy and conservation properties of a three-dimensional unstructured staggered mesh scheme for fluid dynamics, J. Comput. Phys. 175 (2002) 764–791.
[16] J.B. Perot, Comments on the fractional step method, J. Comput. Phys. 121 (1995) 190.
[17] W. Chang, F. Giraldo, J.B. Perot, Analysis of an exact fractional step method, J. Comput. Phys. 180 (2002) 183–199.
[18] R. Hiptmair, Discrete Hodge operators: an algebraic perspective, Prog. Electromag. Res. PIER 32 (2001) 247–269.
[19] A. Bossavit, I. Mayergoyz, Edge elements for scattering problems, IEEE Trans. Mag. 25 (4) (1989) 2816–2821.
[20] J.-C. Nedelec, Mixed finite elements in R^3, Numer. Math. 50 (1980) 315–341.
[21] G. Rodrigue, D. White, A vector finites element time domain method for solving Maxwells equations on unstructured hexahedral grids, SIAM J. Sci. Comput. 23 (3) (2001) 683–706.
[22] D. White, Orthogonal vector basis functions for time domain finite element solution of the vector wave equation, in: 8th Biennial IEEE Conference on Electromagnetic Field Computation, Tucson, AZ, UCRL-JC-129188, 1998.
[23] R. Nicolaides, X. Wu, Covolume solutions of three-dimensional div-curl equations, SIAM J. Num. Anal. 34 (1997) 2195–2203.
[24] R. Nicolaides, Da-Qing Wang, A higher order covolume method for planar div-curl problems, Int. J. Num. Methods Fluids 31 (1) (1999) 299–308.
[25] J.B. Perot, D. Vidovic, P. Wesseling, Mimetic Reconstruction of VectorsIMA Volumes in Mathematics and its Application, vol. 142, Springer, New York, 2006.
[26] M. Shashkov, S. Steinberg, Solving diffusion equations with rough coefficients in rough grids, J. Comput. Phys. 129 (1996) 383–405.
[27] J.M. Morel, J.E. Dendy Jr., M.L. Hall, S.W. White, A cell-centered Lagrangian-mesh diffusion differencing scheme, J. Comput. Phys. 103 (1992) 286.