Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 224 (2007) 59–81
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Discrete calculus methods for diffusion

J.B. Perot *, V. Subramanian

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

Received 1 September 2006; received in revised form 2 November 2006; accepted 20 December 2006
Available online 10 January 2007

Abstract

A general methodology for the solution of partial differential equations is described in which the discretization of the calculus is exact and all approximation occurs as an interpolation problem on the material constitutive equations. The fact that the calculus is exact gives these methods the ability to capture the physics of PDE systems well. The construction of both node and cell based methods of first and second-order are described for the problem of unsteady heat conduction – though the method is applicable to any PDE system. The performance of these new methods are compared to classic solution methods on unstructured 2D and 3D meshes for a variety of simple and complex test cases.

Keywords: Discrete calculus; Staggered mesh; Face/edge elements; Unstructured; Finite volume; Finite element

1. Introduction

This paper is dedicated to Pieter Wesseling. It bears his hallmark at many levels. Philosophically, it is a paper about the intimate connection between physics and mathematics. Prof. Wesseling, an Aerospace Engineer turned Mathematician has always produced papers that are always keenly aware of the connection. Topically, it is a paper about staggered mesh methods – one of many areas in which Pieter and his coworkers are prolific (see the references in [1,2]). And in particular, the paper addresses fundamental questions about how to apply staggered mesh methods to compressible flow problems – an area Pieter is particularly interested in [3–6].

Staggered mesh methods have traditionally been applied to incompressible flows. The lack of pressure modes is particularly attractive in that application. There is therefore considerable literature addressing the issue of how to discretize the momentum equations with structured [7], curvilinear [8,9], and unstructured staggered mesh methods [10–14,30,31]. However, in the context of compressible flow there arises the additional issue of how to discretize the density and energy equations.

The discrete differential operators in incompressible staggered mesh methods have very unique and attractive mathematical properties that allow the discrete equations to physically mimic their continuous counterparts. This not only leads to a lack of pressure modes, but kinetic energy and vorticity conservation statements

* 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
doi:10.1016/j.jcp.2006.12.022

[15,16], maximum principles and many other attractive properties [17]. If we wish the compressible discretization to also have these sorts of attractive physical properties (like entropy increase), then presumably the scalar equations (density and energy) must also be discretized appropriately.

Until recently, it was not clear to the authors what criteria should be used to judge if a scalar transport equation was discretized 'appropriately'. We believe this dilemma is addressed by the Discrete Calculus approach presented in this paper. In order to carefully explain the Discrete Calculus approach, this paper actually only focuses on the unsteady diffusion equation (not the advection–diffusion equation). The diffusion term contains sufficient complexity to present the fundamental ideas of the Discrete Calculus approach. Due to space limitations, the issues concerning advection must be addressed in a subsequent paper.

The premise of this paper is that numerical methods that capture the physics of the equations well have an associated exact Discrete Calculus. The fact that PDE's can always be discretized exactly is demonstrated in the Section 2. To make the presentation clear and concrete the paper focuses on the diffusion (or heat) equation. However, we emphasize from the outset that the basic ideas presented are generally applicable to almost any PDE system. The paper is really an introduction to the Discrete Calculus method. The fact that the diffusion equation is simple and has analogs in many fields of application should make the paper, and hence this method, available to a broad audience.

Two different node-centered Discrete Calculus methods are derived in detail (Section 3). The paper then shows how these ideas can be applied to cell-based discretizations and how they differ from traditional finite volume and discrete Galerkin methods (Section 4). Section 5 then compares these four Discrete Calculus methods to some classic finite volume methods for the diffusion equation on a variety of test problems.

2. Exact discretization

Discretization takes a continuous PDE equation with essentially an infinite number of equations and unknowns (at least one for every point in space) and reduces it to a finite system of algebraic equations and unknowns. It is frequently assumed that the act of discretizing a PDE must involve approximation or the introduction of some sort of error. This is not the case [18]. *Solving* a PDE system numerically does indeed require approximation, but it is possible to separate the process of discretization and approximation and when this is done discretization can be performed exactly. One premise of this paper is that exact discretization is highly advantageous and leads to methods that have very interesting mathematical and physical properties.

Exact discretizations ultimately require approximation because the discretization is not closed. There are more discrete unknowns than algebraic equations. Closure of the system requires the coupling of some of the discrete unknowns. This coupling process is an interpolation problem where all the numerical approximation and errors are introduced. It is often characterized by a transfer of information from one mesh to a different (dual) mesh, and it invariably involves a material constitutive relation.

The profound benefits of separating the discretization process (where the continuous PDE system is made finite) from the approximation process (where the finite system becomes solvable) will become very clear as we proceed. Nevertheless, we describe the key ideas abstractly here to preview what will be seen in the paper. It will be seen that the closure (and therefore approximation) of the exact finite equation system always occurs in the material constitutive relations embedded in the PDE. These constitutive relations are actually physical approximations of bulk material behavior. They are not exact to begin with. This approach therefore places all numerical errors/approximation in the already physically approximate material relations. The physics of a PDE (such as conservation, and wave propagation) never depend on the details of the material. This approach will therefore always capture the physics of the PDE exactly by placing all numerical approximation or errors in the material properties.

To make the presentation of the Discrete Calculus method concrete we will use a simple equation that is common to many fields of engineering and science – the heat equation.

$$\frac{\mathrm{d}(\rho C T)}{\mathrm{d}t} = \nabla \cdot k \nabla T \tag{1}$$

In heat transfer, the temperature T is the fundamental unknown, and the material parameters are, k the conductivity, and ρC the heat capacity. However, this equation, or slight variants, finds application in many other fields with different physical interpretations for the variables.

It is convenient to consider the heat equation in an expanded form that clearly separates the physics/mathematics from the material constitutive approximations.

$$\frac{\mathrm{d}i}{\mathrm{d}t} + \nabla \cdot \mathbf{q} = 0 \qquad \text{Conservation of energy} \tag{2a}$$

$$\mathbf{g} = \nabla T \qquad \text{Definition of gradient} \tag{2b}$$

$$\mathbf{q} = -k\mathbf{g} \qquad \text{Fourier's Law} \tag{2c}$$

$$i = \rho C T \qquad \text{Perfectly Caloric Material} \tag{2d}$$

This formulation introduces two new physical variables, i the specific internal energy, and $\mathbf{q}$ the heat flux. The last two (algebraic) equations are physical approximations for the material. All numerical approximations will also be restricted to these last two equations. The first two equations, containing the physics and calculus, will be discretized exactly. The advantage of discretizing the physics and calculus exactly is that the resulting numerical methods and discrete solutions cannot violate any physical or mathematical principles.

3. Node based exact discretizations

One classic way to discretize Eq. (2a) exactly is using the idea of many small non-overlapping control volumes that completely cover the domain. However, the classic finite volume (FV) procedure of associating a control volume with each mesh cell has some difficulties – we return to it later in Section 4. It is easier to consider a set of control volumes in which each finite volume surrounds each node (vertex) of the mesh. The volumes surrounding each node are referred to as dual-mesh cells.

3.1. Exact discrete equations

For heat transfer, integrating over each dual cell gives the exact discrete equation,

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\tilde{c}} i\,\mathrm{d}V + \sum_{\tilde{f}} \int_{\tilde{f}} \mathbf{q}\cdot\mathbf{n}\,\mathrm{d}A = 0 \tag{3a}$$

There is one equation for each dual cell. The discrete unknowns in this equation are, $I_{\tilde{c}} = \int_{\tilde{c}} i\,\mathrm{d}V$ the total energy in the dual cell, and $Q_{\tilde{f}} = \int_{\tilde{f}} \mathbf{q}\cdot\mathbf{n}\,\mathrm{d}A$ the heat flux between dual cells or on the domain boundary. The notation convention is to label variables with their location on the primary mesh or the dual mesh. In addition, the dual mesh locations are distinguished by having a tilde. So far the method looks like a classic node-based finite volume method or discontinuous Galerkin method. The key difference therefore lies in the discretization of Eq. (2b).

In addition to discretizing (2a) exactly, it is critical that Eq. (2b) also be discretized exactly or most of the advantages of exact discretization are lost. Exact discretization of Eq. (2b) can be achieved by integrating along the line connecting the two nodes. This gives the exact discrete equation

$$\int_e \mathbf{g}\cdot\mathrm{d}\mathbf{l} = T_{n2} - T_{n1} \tag{3b}$$

on each edge, where T_n is the value of the temperature at the node position and the discrete unknown $g_e = \int_e \mathbf{g}\cdot\mathrm{d}\mathbf{l}$ is the average gradient along each edge. Actually any path connecting the two points is possible but the edge is the obvious initial choice.

We now have one equation at each node of the mesh (Eq. (3a)), and one primary unknown at each node, T_n. In linear algebraic terms the exact discretization is written,

$$\frac{\mathrm{d}I_{\tilde{c}}}{\mathrm{d}t} + \widetilde{D}Q_{\tilde{f}} = -Q_n^{\mathrm{BC}} \tag{4a}$$

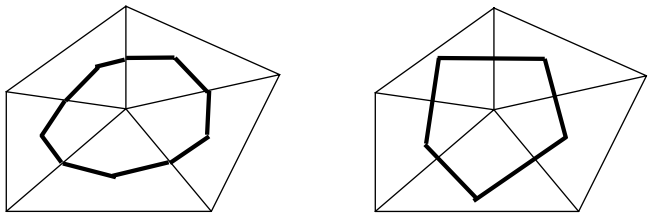

Fig. 1. Primary mesh (thin lines) and two commonly used dual meshes (thick lines) for a triangular primary mesh in two-dimensions. These dual meshes also exist in three-dimensions. The Median dual applies to arbitrary polygonal cells, the Voronoi dual mesh is only defined on simplices (triangles and tetrahedra), Cartesian meshes, and locally orthogonal meshes (such as cylindrical and spherical).

where Q_n^{BC} are the prescribed boundary condition fluxes on the dual cells associated with nodes that lie on the Neumann boundaries of the problem domain and $\widetilde{D}$ is the discrete divergence operator. On Dirichlet boundaries the temperature is known and the associated row of Eq. (4a) can be used to determine the flux on the Dirichlet boundary (if desired). Along with this equation we also have,

$$g_e = GT_n \tag{4b}$$

where G is the discrete gradient operator. There are no boundary condition issues associated with this equation since every edge of the mesh is always bounded by two nodes, but it should be noted that the vector of discrete nodal temperature values, T_n, also contains boundary values (even if they lie on Dirichlet boundaries and are known quantities).

At this point no assumptions about the primary mesh or the dual mesh have been made. The primary mesh can have curved edges and faces, and the dual mesh can be defined many different ways. Two of the more obvious choices for the dual mesh (the Voronoi dual and Median dual shown in 2D in Fig. 1) are discussed in more detail later. Since any dual mesh works, the choice of the dual mesh is one area of flexibility for Discrete Calculus methods. The other very significant area of flexibility in the method is how one relates $Q_{\tilde{f}}$ to g_e and $I_{\tilde{c}}$ to T_n in order to close the system (Section 3.3).

3.2. Discrete operators

The matrices $\widetilde{D}$ and G do not have subscripts associated with them because they are discrete operators that transfer information from one mesh location to another. The discrete divergence, $\widetilde{D}$ takes information from the dual faces and produces a result that resides on the dual cells. The discrete gradient, G, on the other hand, takes information from the nodes and produces a result on the mesh edges. These Discrete Calculus operators are sparse matrices consisting of nonzero entries with ± 1.

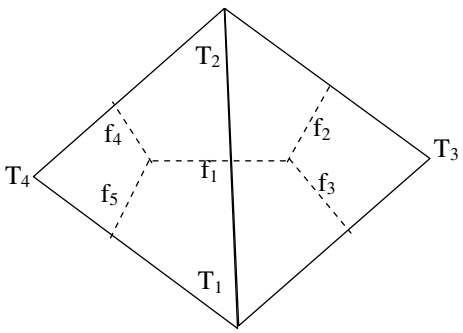

Fig. 2. 2D Mesh to illustrate DC operators.

For the simple 2D mesh shown in Fig. 2 the corresponding operators are,

$$\mathbf{G} = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \\ -1 & 0 & 0 & 1 \end{bmatrix}$$

$$\widetilde{\mathbf{D}} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ -1 & 1 & 0 & 1 & 0 \\ 0 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 \end{bmatrix}$$

Note that the gradient operator G is a 5×4 matrix converting node values (4 nodes) into edge values (5 edges). Similarly, the divergence operator $\widetilde{D}$ is a 4×5 matrix that converts dual face quantities into dual cell quantities. Many readers will notice that there is a symmetry between the discrete gradient and divergence operators, $G = -\widetilde{D}^{\mathrm{T}}$. This relation holds for many low-order Discrete Calculus methods. This type of symmetry also occurs in Galerkin Finite Element methods. It was originally thought by the authors to be an important property of Discrete Calculus methods. Interestingly, recent work on higher-order Discrete Calculus methods [19] indicates this type of symmetry is not actually necessary.

Cell based versions of these operators have been discussed previously [20–24]. In those works, the operators were first hypothesized and then later shown to have interesting and attractive mathematical properties. The current derivation makes it clear from the outset that these similar but node-based discrete operators will behave mathematically well. More importantly, the Discrete Calculus approach presents a general framework showing how to generate other well behaved discrete operators. For example, higher-order versions of the node-based discretization (Eqs. (4a) and (4b)) can be derived using the Discrete Calculus approach [19].

The operators $\widetilde{D}$ and G are discrete versions of the continuous divergence and gradient operators. They were derived using versions of the Gauss–Green theorem and no approximation was used in their derivation. The result is that these discrete operators mimic most of the mathematical properties of the corresponding continuous operators. For example, the only solution to $\nabla\phi = 0$ on an infinite or periodic domain is ϕ is constant. Similarly, the only solution to the discrete problem $G\varphi_n = 0$ on a periodic domain is that the vector of unknowns φ_n is constant. Phrased a different way, the null space of the gradient is a constant function and the null space of a properly derived discrete gradient operator is a constant vector. This is the fundamental reason that staggered mesh methods can not display spurious pressure modes. In the context of heat transfer, this implies that zero heat flux results in a constant temperature solution. There are many numerical methods where this is not the case. These methods use different forms of damping to remove the resulting spurious modes. These spurious modes are non-trivial solutions to $G\varphi_n = 0$ that result when the discrete gradient G is not derived using a Discrete Calculus approach.

Similarly, $\nabla \times \nabla\phi = 0$, the gradient operator is always in the null space of the curl operator. In the Discrete Calculus approach the sparse discrete operator C which is the (oriented) sum of edge values to faces (circulation on the face), has this same property $CG = 0$ [13].

The nested null spaces described above are described in full by the de Rham complex [25] which appears in the mathematical field Algebraic Topology. Algebraic topology is widely used to describe face/edge elements [26–29] which are the Finite Element variant of the Discrete Calculus approach. While highly expressive, algebraic topology is not widely accessible to scientists/engineers and is difficult to apply to nonlinear equation systems like the Navier–Stokes equations. We therefore leave it to the reader to explore this mathematical explanation more fully on their own if they are interested. For physicists and engineers the preceding explanation of the method may be sufficient.

3.3. Discrete equation closure

The system comprised of algebraic Eqs. (4a) and (4b) is discrete and exact, but closure and solution of this exact system requires relating the heat flux on the dual faces $Q_{\tilde{f}} = \int_{\tilde{f}} \mathbf{q} \cdot \mathbf{n}\,\mathrm{d}A$ to the temperature gradient along

the edges $g_e = \int_e \mathbf{g} \cdot \mathrm{d}\mathbf{l}$, and the temperature at the nodes T_n to the total energy in the dual cells surrounding each node $I_{\tilde{c}} = \int_{\tilde{c}} i \,\mathrm{d}V$. Mathematically the desired relations are written,

$$Q_{\tilde{f}} = -M_{(kA_{\tilde{f}}/L_e)} g_e \tag{4c}$$

$$I_{\tilde{c}} = M_{(\rho C V_{\tilde{c}})} T_n \tag{4d}$$

where $M_{(kA/L)}$ and $M_{(\rho CV)}$ are transfer matrices that connect quantities on the primary and dual meshes. These matrices contain information about the material properties and specific mesh geometries. Note that Eqs. (4c) and (4d) correspond directly to the material constitutive relations (2c) and (2d). These material relations can not be implemented exactly because the unknowns are averages over different geometric structures. Ultimately, it will be clear that $M_{(kA/L)}$ and $M_{(\rho CV)}$ are essentially interpolation operators transferring data from one mesh to another one (while also applying the material properties). Combining Eqs. (4a)–(4d) gives a single equation system for the discrete temperature at the nodes (equivalent to Eq. (1)),

$$\frac{\mathrm{d}}{\mathrm{d}t} M_{(\rho C V_{\tilde{c}})} T_n = \widetilde{D} M_{(kA_{\tilde{f}}/L_e)} G T_n - Q_n^{\mathrm{BC}} \tag{5}$$

In general, the Discrete Calculus approach does not require that the two meshes (primary and dual) have any relation to each other. In the general case the matrix $M_{(kA/L)}$ need not even be square or invertible. However, for any mesh there are an infinite number of 'closely associated' dual meshes where there is a one to one correspondence (in number) between the primary mesh edges and the dual mesh faces, the primary mesh nodes and dual mesh cells, and vice versa. For these 'closely associated' dual meshes the transfer matrices $M_{(kA/L)}$ and $M_{(\rho CV)}$ are square. Certain dual meshes (like the Voronoi dual mesh) even result in diagonal transfer matrices. When the primary mesh is Delaunay, the matrices $M_{(kA/L)}$ and $M_{(\rho CV)}$ for the Voronoi dual can in addition be shown to be positive definite. The simplest numerical methods have diagonal or sparse (usually local) transfer matrices (or sparse inverses).

In the algebraic topology construction the transfer matrices are referred to as discrete Hodge star operators that transfer data between the discrete de Rham complex associated with the primary mesh and the discrete de Rham complex associated with the dual mesh.

3.3.1. Voronoi dual interpolation

Any mesh which consists entirely of cells with a unique circumcenter has a closely associated Voronoi dual mesh that can be constructed from those circumcenters. All triangular (in 2D) and tetrahedral (in 3D) unstructured meshes have this property, as do Cartesian meshes, and cylindrical and spherical meshes, and many prismatic meshes. Arbitrary quadrilateral and hexahedral meshes are the most important class of meshes which do not have unique circumcenters or a Voronoi dual mesh.

The Voronoi dual mesh is of interest because it is everywhere locally orthogonal to the primary mesh. So for example, all the tetrahedra which share a common edge (no matter how many of them there are) have circumcenters that lie in a plane and that plane is orthogonal to the common edge. This has the attractive property that the dual face normal and edge tangential point in exactly the same direction. This means that $Q_{\tilde{f}} = \int_{\tilde{f}} \mathbf{q} \cdot \mathbf{n} \,\mathrm{d}A$ and $g_e = \int_e \mathbf{g} \cdot \mathrm{d}\mathbf{l}$ are referring to the same component of the vector. $Q_{\tilde{f}}/A_{\tilde{f}}$ is therefore a second-order approximation for that flux component at the center of the dual face, and $-kg_e/L_e$ is a second-order approximation for the flux at the midpoint of the edge. Because the midpoint of the edge and the center of the dual face lie close to each other (within a mesh spacing) the approximation

$$Q_{\tilde{f}} = -k \frac{A_{\tilde{f}}}{L_e} g_e \tag{6c}$$

is a first-order approximation. The matrix $M_{(kA_{\tilde{f}}/L_e)}$ is then diagonal with $k\frac{A_{\tilde{f}}}{L_e}$ as its entries. If the mesh is regular (such as equilateral triangles) the edge center and dual face center are identical and second-order accuracy is obtained by this approximation. In practice, second-order accuracy is also often observed for relatively smooth meshes. When the conductivity varies, it is assumed to be piecewise constant in each mesh cell then Eq. (6c) is more generally $Q_{\tilde{f}} = -\frac{(kA)_{\tilde{f}}}{L_e} g_e$ where $(kA)_{\tilde{f}} = \sum_{\text{edge cells}} k_c A_{\tilde{f}c}$ and $A_{\tilde{f}c}$ is the portion of the dual face residing in each cell.

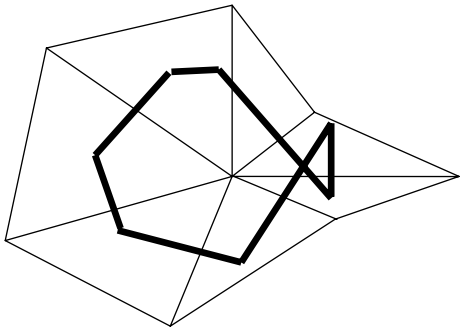

Fig. 3. Example of an inverted Voronoi cell.

Appling the same ideas to Eq. (4d) gives the first-order approximation,

$$I_{\tilde{c}} = (\rho C V_{\tilde{c}}) T_n \tag{6d}$$

so the matrix $M_{(\rho C V_{\tilde{c}})}$ is also diagonal with $\rho C V_{\tilde{c}}$ as its entries. When the mesh is uniform the centroid of the dual cell coincides with the node position and this approximation also becomes second-order accurate.

The placement of unknowns is very similar in this node based method to that of a linear finite element method. However, the resulting method is not the same. In 3D, classic linear finite elements on tetrahedral do not have a useful discrete maximum principle [23,24]. The requirement on the mesh is that all tetrahedra planar angles must be less than 90°, and in practice this requirement is impossible to satisfy. In contrast, the method just discussed has a maximum principle on any unstructured mesh that is Delaunay. This follows from the symmetry of the gradient and divergence operators and the positive definiteness of the interpolation matrix $M_{(kA/L)}$ for Delaunay meshes. This is a concrete example, for this particular problem, of how Discrete Calculus discretization can capture the physics/mathematics of the system well.

The Voronoi dual interpolation is of historic interest because it is the interpolation used in Cartesian staggered mesh methods [7]. It is also essentially the mesh that is used in many meshless or particle methods [32]. The lack of a Voronoi dual for arbitrary quadrilateral and hexahedral meshes explains why it is a non-trivial exercise to extend staggered mesh methods to those types of meshes. However, Wesseling [8] describes how this can be accomplished via a mapping of the problem to a Cartesian mesh.

Nicolaides [10] and Porshing [33,34] were the first to recognize the Voronoi dual as one of the logical generalizations of the Cartesian staggered mesh methods to unstructured meshes. But it should be noted that while every triangular or tetrahedral mesh has a Voronoi dual, that dual mesh is only well formed if the primary mesh is Delaunay. Non-Delaunay meshes produce Voronoi dual meshes where the Voronoi cells can be twisted over on themselves (see Fig. 3) resulting in logically negative volumes and lengths that can make the transfer matrices singular or indefinite. It is easy to find mesh generators that make Delaunay meshes, and algorithms that can convert almost Delaunay meshes into strictly Delaunay meshes are very fast. In practice, the disadvantage of the Voronoi dual is not the Delaunay requirement but the fact that for strongly distorted 2D meshes or even moderately distorted 3D meshes the Voronoi dual can be quite inaccurate [16]. An alternative and more accurate interpolation is the Median dual interpolation.

3.3.2. Median dual interpolation

The median dual mesh is formed from the primary mesh cell, face, and edge centroids. It is defined for any primary mesh and never results in logically negative volumes, areas, or lengths. However, the dual faces (on which the $Q_{\tilde{f}} = \int_{\tilde{f}} \mathbf{q} \cdot \mathbf{n} \mathrm{d}A$) are defined are no longer planar. In 2D each dual face now consists of two line segments (see Fig. 4) and in 3D it consists of two subtriangles from every tetrahedra touching that edge (only one of which is shown in Fig. 4 for reasons of clarity), each subtriangle being formed from the edge midpoint, the face centroid and the cell centroid. The orientation of the edge and the normals of the various subtriangles forming the dual face are no longer related. A more intricate interpolation is now required.

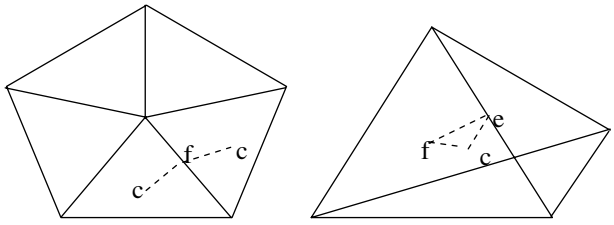

Fig. 4. Dual faces (shown in dotted lines) in 2D and 3D.

One possibility is to use the gradients along the cell edges, g_e, to determine the gradient within each cell. This can then be scaled by the conductivity and used to perform the integration for the flux on the dual faces. The idea is essentially: (1) assume the gradient, $\mathbf{g}$, has a locally polynomial form within each cell, (2) determine the coefficients in that polynomial from the known boundary data g_e, (3) determine the heat flux polynomial in each cell from $\mathbf{q}(\mathbf{x}) = -k\mathbf{g}(\mathbf{x})$ and then (4) integrate the heat flux polynomial over the dual face to obtain the flux $Q_{\tilde{f}}$.

On triangles and tetrahedra we assume the temperature is linear and therefore the gradient is constant. (see [19] for how to do higher-order interpolation). There appears to be more data than unknowns (three g_e for two unknowns on triangles and six g_e values for three unknowns on tetrahedra) but the extra data is redundant, and these problems have unique solutions. Determining the constant gradient can be done with a 2×2 matrix inversion in each cell in 2D or a 3×3 matrix inversion in 3D.

For triangles and tetrahedra, the inversion can be determined analytically [13] resulting in a direct formula for calculating the gradient in each cell,

$$\mathbf{g}_c = \frac{1}{V_c} \sum_{\text{edges}} g_e \mathbf{n}_{\tilde{f}e} \tag{7}$$

where $\mathbf{n}_{\tilde{f}e}$ is the normal (and area) associated with the dual face for that edge, and the summation is over all edges of a cell. This explicit formula is useful for several reasons. First it speeds up the computation. But perhaps more importantly it shows that the explicit polynomial reconstruction step can actually be eliminated. This formula can be applied to arbitrary cells (not just triangles and tetrahedra) without having to define the exact functional form (which may or may not be a polynomial) that it corresponds to. For example, it is easy to see that the formula also works on 2D and 3D Cartesian meshes though interpolation and integration on those meshes is much more complicated (temperature must be assumed to be bilinear or trilinear).

To obtain the flux on the dual faces we assume that the conductivity (and therefore the heat flux) is constant in each cell. The integration for the flux can then be performed exactly and,

$$Q_{\tilde{f}} = \sum_{\text{cells}} -k_c \mathbf{g}_c \cdot \mathbf{n}_{\tilde{f}e} \tag{8}$$

All together the transfer process involves using the edge gradient components to calculate the gradient within each cell and then the gradient in each cell to determine the flux through the dual face. In matrix terms we have $M_{(kA/L)} = N^{\mathrm{T}} K N$ where the rectangular matrix N has as entries the vectors $\mathbf{n}_{\tilde{f}e}$ and the matrix K is diagonal and positive definite with entries $\frac{k_c}{V_c}$. The transfer matrix is therefore positive definite. The geometric inputs to this transfer scheme are the dual face areas and the cell volumes. In contrast to the Voronoi transfer matrix, the edge lengths no longer enter directly.

The median dual mesh reconstruction is first-order accurate for the gradient of the temperature and therefore second-order accurate for the temperature field. It can capture piecewise linear temperature solutions with jumps in the gradient at the cell faces (usually due to jumps in the material properties) exactly. On triangles and tetrahedral this procedure results in the same ultimate diffusion matrix as the linear finite element method.

The transfer matrix $M_{(\rho CV)}$ for the unsteady term can be obtained in a similar manner to the diffusion term. Assuming a linear temperature profile in each triangle or tetrahedra and a constant heat capacity, ρC, in each cell means the internal energy varies linearly. Integrating over the dual mesh volume surrounding each node involves integrating over the portion of each cell contained in that dual volume. The result is

$$I_{\tilde{c}} = \sum_{\text{cells}} \rho C \frac{V_c}{(\mathrm{ND}+1)} \left(\frac{\mathrm{ND}^2 + (\mathrm{ND}-1)^2}{\mathrm{ND}^2(\mathrm{ND}+1)} T_n + \frac{\mathrm{ND}^3 - (\mathrm{ND}-1)^2}{\mathrm{ND}^2(\mathrm{ND}+1)} T_c \right) \tag{9a}$$

for tetrahedral, triangles, and 1D line segments. In this formula ND is the dimensionality of the problem (2 for 2D, 3 for 3D, etc.), T_c is the average of the cell nodal temperatures and the summation is over all cells that overlap the dual cell. The resulting mass matrix, $M_{(\rho CV)}$ has the same sparcity pattern as the linear FE one, but the values are different. This mass matrix can also be represented as a lumped mass plus a Laplacian term.

$$I_{\tilde{c}} = T_n(\rho CV)_n + \widetilde{D} \left\{ \frac{\mathrm{ND}^3 - (\mathrm{ND}-1)^2}{\mathrm{ND}^2(\mathrm{ND}+1)^2} \right\} (\rho CV)_e G T_n \tag{9b}$$

where $(\rho CV)_n = \sum_{\text{node cells}} \rho C \frac{V_c}{(\mathrm{ND}+1)}$ and $(\rho CV)_e = \sum_{\text{edge cells}} \rho C \frac{V_c}{(\mathrm{ND}+1)}$.

This form of the mass matrix makes it clear that the matrix is symmetric and invertible.

4. Cell based exact discretizations

The Discrete Calculus formalism is an approach, not a particular method. It does not require that unknowns be placed any particular location in the mesh, and in order to demonstrate that we consider a cell based method in this section.

4.1. Exact discrete equations

Integrating over each mesh cell gives an equation that is identical to most finite volume and discrete Galerkin methods.

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_c i \, \mathrm{d}V + \sum_f \int_f \mathbf{q} \cdot \mathbf{n} \, \mathrm{d}A = 0 \tag{10a}$$

This equation looks almost identical to Eq. (3a) but the key distinction is the lack of tildes on the locations meaning this equation applies on the primary mesh cells (not the dual cells like Eq. (3a)). The discrete unknowns in this equation are, $I_c = \int_c i \, \mathrm{d}V$ the total energy in the cell, and $Q_f = \int_f \mathbf{q} \cdot \mathbf{n} \, \mathrm{d}A$ the heat flux between cells or on the domain boundary. Similarly to Eq. (3b) we also have the exact discrete equation

$$\int_{\tilde{e}} \mathbf{g} \cdot \mathrm{d}\mathbf{l} = T_{\tilde{n}2} - T_{\tilde{n}1} \tag{10b}$$

along some line connecting the cell centers. Note that we do not use the notation T_c for the temperature at the cell centers. The cell subscript implies a cell averaged quantity and Eq. (10b) refers to point values at the cell centers (which are the nodes of the dual mesh).

In matrix terms these equations are expressed,

$$\frac{\mathrm{d}I_c}{\mathrm{d}t} + DQ_f = 0 \tag{11a}$$

$$g_{\tilde{e}} = \widetilde{G} T_{\tilde{n}} + T_f^{\mathrm{BC}} \tag{11b}$$

where T_f^{BC} is the vector of prescribed Dirichlet boundary condition values on the boundaries of the problem domain.

There is a great deal of duality between the cell and node based methods, but one place this symmetry is lost is at the domain boundaries. In this work, the roles of the primary mesh and the dual mesh are not entirely interchangeable. We assume the primary mesh is constructed so that the boundary faces of the primary mesh are aligned with the domain boundaries, and internal faces are aligned with material discontinuities (which are effectively internal boundaries). The result is that with the cell based method, the Neumann boundary condition (Q_f on the boundary) is easy to satisfy and the Dirichlet boundary condition requires extra effort in the gradient equation. In contrast, the node based method was trivial for Dirichlet boundary conditions (T_n) and the energy equation required modification for Neumann boundary conditions.

The closure approximations

$$Q_f = -M_{(kA_f/L_{\tilde{e}})} g_{\tilde{e}} \tag{11c}$$

$$I_c = M_{(\rho CV_c)} T_{\tilde{n}} \tag{11d}$$

are where the cell based Discrete Calculus approach differs from classic finite volume and discrete Galerkin methods. Note that these transfer matrices $M_{(kA_f/L_{\tilde{e}})}$ and $M_{(\rho CV_c)}$ are different from those found in the node based method (the subscripts have different tilde assignments).

4.2. Voronoi dual interpolation

If the Voronoi dual exists and is used then the local orthogonality property exists. In particular, the line between two cell circumcenters is always perpendicular to the face between those two cells, so we can use the first-order approximation

$$Q_f = -k \frac{A_f}{L_{\tilde{e}}} g_{\tilde{e}} \tag{12c}$$

This approximation becomes second-order accurate if the dual edge midpoint and the face centroid coincide (uniform meshes).

The variable conductivity case can be derived by assuming that the normal flux on each side of the face is equal $\left(Q_f/A_f = k_2 \frac{T_{\tilde{n}2} - T_f}{L_{\tilde{e}2}} = -k_1 \frac{T_{\tilde{n}1} - T_f}{L_{\tilde{e}1}}\right)$. Then solving for the face velocity and rearranging gives $Q_f = -\frac{A_f}{\left(\frac{L_{\tilde{e}1}}{k_1} + \frac{L_{\tilde{e}2}}{k_2}\right)} g_{\tilde{e}}$ where $L_{\tilde{e}i}$ is the portion of the dual edge in cell i. In other words, the length weighted harmonic average of the conductivity should be used (see also [37]).

For the other transfer matrix we have

$$I_c = \rho CV_c T_{\tilde{n}} \tag{12d}$$

which is first-order accurate unless the cell circumcenter and centroid coincide. Note that even if an unstructured mesh is Delaunay the cell circumcenter can lie outside the cell, and this usually leads to large errors in that region of the mesh. The median dual mesh does not have this problem.

4.3. Median dual interpolation

The use of the median dual can increase the accuracy, but as in the node centered versions, it also increases the complexity of the method. When the median dual is used, the dual edge consists of two straight line segments that connect the cell centroids to the face centroid. The data $g_{\tilde{e}}$ therefore is very difficult to reconstruct to determine the cell gradients. However it is possible to reverse the process and determine the gradients $g_{\tilde{e}}$ in terms of the fluxes. Applying Gauss' theorem to $\int (x_i q_j)_{,j} \, dV$ tells us that $\int (\mathbf{q}_i + \mathbf{x}\nabla \cdot \mathbf{q}) \, dV = \int \mathbf{x}(\mathbf{q} \cdot \mathbf{n}) \, dA$. If we assume that the flux is constant on faces and the divergence is constant in a polygonal cell (a first-order approximation for the heat flux), then

$$\mathbf{q}_c = \frac{1}{V_c} \sum_{\text{cell faces}} \mathbf{r}_{fc} \hat{Q}_f \tag{13}$$

where $\mathbf{r}_{fc} = \mathbf{x}_f - \mathbf{x}_c$ is the vector from the cell centroid to the face centroid and the circumflex (hat) on the flux indicates it is the flux out of the cell in question.

It is important to note that the flux is not assumed to be piecewise constant in each cell – this would violate the energy equation in the unsteady case. On triangles and tetrahedra the interpolation function is actually $\mathbf{q}(\mathbf{x}) = \mathbf{q}_c + \frac{\mathbf{x}}{\mathrm{ND}} (\nabla \cdot \mathbf{q})_c$. However, Eq. (13) applies to arbitrary polygons and removes the need to calculate the coefficients of some polynomial (which changes depending on the cell shape). Like the node based median dual interpolation this allows the method to remain independent of particular cell shapes (in the general philosophical train as finite volume methods).

Using first-order integration along the dual edge (using the end points) then gives,

$$g_{\tilde{e}} = -\{\mathbf{r}_{fc1} \cdot (\mathbf{q}_{c1}/k) - \mathbf{r}_{fc2} \cdot (\mathbf{q}_{c2}/k)\} \tag{14}$$

Using higher-order integration is possible but unnecessary. It would result in an unsymmetric transfer matrix that is still first-order (because Eq. (13) is still a first-order approximation). In matrix terms the transfer between meshes can be written, $g_{\tilde{e}} = -R^{\mathrm{T}} K_{1/Vk} R Q_f$ where the matrix R contains the vectors $\mathbf{r}_{fc}$ and the diagonal matrix $K_{1/Vk}$ has $\frac{1}{kV_c}$ as its entries. Note that $\left(M_{kA_f/L_{\tilde{e}}}\right)^{-1} = R^{\mathrm{T}} K_{1/Vk} R$ is symmetric and positive definite and sparse. The transfer matrix $M_{kA_f/L_{\tilde{e}}}$ is therefore a full matrix and writing a single equation system like Eq. (5) for the cell unknowns is not practical. Instead the symmetric coupled system,

$$\begin{bmatrix} \frac{1}{\Delta t} M_{\rho CV_c} & -D \\ G & \left(M_{kA_f/L_{\tilde{e}}}\right)^{-1} \end{bmatrix} \begin{pmatrix} T_{\tilde{n}}(t^{n+1}) \\ Q_f \end{pmatrix} = \begin{pmatrix} \frac{1}{\Delta t} M_{\rho CV_c} T_{\tilde{n}}(t^n) \\ T_f^{\mathrm{BC}} \end{pmatrix} \tag{15}$$

must be solved. The mass matrix $M_{\rho CV_c}$ remains diagonal and is the same as the Voronoi dual transfer matrix (Eq. (12d)). However, since the temperature is now located at the cell centroids, the accuracy of this approximation is now second-order. First-order reconstruction of the gradient via $\left(M_{kA_f/L_{\tilde{e}}}\right)^{-1}$ is sufficient to retain second-order accuracy for the temperature field.

Because the system given by Eq. (15) requires the simultaneous solution of both the temperature and the heat fluxes we refer to it as the Mixed Method. As far as the authors are aware it is completely new and represents an example of how the Discrete Calculus approach can lead to novel numerical methods that by their construction capture the physics of the system well.

The appearance of a transfer matrix in which the inverse is sparse contrasts with the node-based median dual mesh method. As with the boundary conditions, a symmetry between the two methods has been broken. The reason for the difference here is that while the role of the primary and dual meshes was switched in the two different methods, the reconstruction region (cells) was not changed. The reconstruction must occur on cells (the primary mesh) in both cases because material properties are assumed to be associated with the primary mesh (cells). It is certainly possible to change the regions over which material properties are defined (say to the dual cells), then the symmetry remains intact and the resulting methods are identical to what has already been described with the words dual and primal switched everywhere.

On simplices (triangles and tetrahedra), and Cartesian meshes, the assumptions used to derive Eq. (13) (constant $\mathbf{q}$ on the cell faces and constant divergence) are exactly equivalent to assuming the lowest-order face elements were used in the reconstruction of the heat flux. For steady heat conduction without sources, the Mixed method will therefore produce exactly the same result as face FE methods applied to the coupled PDE system (2a)–(2d). Remember, the node based median dual mesh was equivalent in this same limit to standard linear finite elements applied to Eq. (1). However with unsteady terms, source terms, or higher-order implementations, these Discrete Calculus methods differ from the low-order FE methods.

The mixed method becomes numerically difficult to solve in the limit as $k \to 0$. This might be partly ameliorated by non-dimensionalization of the unknowns $T_{\tilde{n}}$ and Q_f, but is ultimately a reflection of the fact that in this limit the inverse of the transfer matrix (which is used in this method) becomes singular. This method might therefore be ill-suited for a diffusionless transfer process (such as the mass equation in the Navier–Stokes equations).

5. Performance results

The four low-order Discrete Calculus methods derived in this paper might be considered more complex (at least conceptually) than classical methods for solving the heat equation. It is therefore of considerable interest to see if this added complexity and the exact treatment of the calculus has any actual tangible benefits in the solutions generated. This section will therefore compare the proposed methods with some classic finite volume approaches on a number of test cases. We have already shown that the methods are theoretically similar to some finite element methods so it is not necessary to directly compare with those finite element methods.

A direct comparison of accuracy is difficult when comparing node based and cell based methods. On 3D tetrahedral meshes there are roughly 5–6 more cells than nodes. Cell methods therefore have a significant

resolution advantage. On the other hand, node based methods are far less computationally intensive. As a compromise our ultimate metric of method performance in this work will be level of accuracy obtained per the computational cost.

This section first presents commonly used cell-based finite volume methods, which are then used for comparison against the Discrete Calculus methods. Numerical tests illustrating the spatial and temporal accuracy as well as the cost to obtain a desired accuracy are then presented. Finally, heat transfer in a complex geometry (a crank shaft) is considered and the computational cost of the Discrete Calculus methods are compared against the traditional finite volume methods in order to confirm the results established by the previous tests in a realistic problem configuration.

5.1. Cell based finite volume methods

Given the restrictions of space and time we will restrict our attention to cell based finite volume methods. These methods also use the conservation equation $\frac{\mathrm{d}(\rho C V T_{\hat{n}})}{\mathrm{d}t} + DQ_f = 0$, where $T_{\hat{n}}$ is typically located at the cell centroid. The key in these methods is to relate the face flux Q_f, to the cell temperature, $T_{\hat{n}}$. We will consider three alternatives for this relationship.

5.1.1. Basic

The basic interpolation is,

$$Q_f = -\frac{A_f}{\frac{L_1^\perp}{k_1} + \frac{L_2^\perp}{k_2}} G T_{\hat{n}} \tag{16}$$

This is very similar in form to the cell based Voronoi dual interpolation. However, the temperature is no longer located at circumcenters so this is not really an approximation for the normal gradient. As a result this formulation (while widely used) is zeroth-order accurate (nonconvergent) unless all cell centroids and circumcenters are identical (such as Cartesian meshes or uniform unstructured meshes). The distance $L^\perp$ is the perpendicular distance between the cell centers $|(\mathbf{x}_f - \mathbf{x}_c) \cdot \mathbf{n}_f / A_f|$.

5.1.2. Large stencil

The solution to the lack of convergence of the basic method is typically to determine the gradient in each cell and use a weighted average of the cell gradients to determine the normal heat flux,

$$Q_f = -(w_1 k_1 \nabla T_{c1} + w_2 k_2 \nabla T_{c2}) \cdot \mathbf{n}_f \tag{17}$$

The problem is then to determine the gradients and the weights. We determine the gradients using Gauss' theorem,

$$\nabla T_c \approx \frac{1}{V_c} \int (\nabla T)\, \mathrm{d}V = \frac{1}{V_c} \sum_{\text{cell faces}} T_f \mathbf{n}_f = \frac{1}{V_c} \sum_{\text{cell faces}} (T_f - T_{\hat{n}}) \mathbf{n}_f \tag{18a}$$

Assuming an interpolation for the face values $T_f = \hat{w}_2 T_{\hat{n}1} + \hat{w}_1 T_{\hat{n}2}$ gives,

$$\nabla T_c = \frac{1}{V_c} \sum_{\text{cell faces}} (\hat{w}_2 T_{\hat{n}1} + \hat{w}_1 T_{\hat{n}2} - (\hat{w}_1 + \hat{w}_2) T_{\hat{n}1}) \mathbf{n}_f = \frac{1}{V_c} \sum_{\text{cell faces}} \hat{w}_1 \mathbf{n}_f G T_{\hat{n}} \tag{18b}$$

from this we can see that the diffusion matrix is symmetric if $\hat{w}_2 = w_1$ and $\hat{w}_1 = w_2$. For higher-order methods symmetry is not required, but for low-order methods such as this it seems to be a good idea.

Choosing volume weights, $w_1 = V_1/(V_1 + V_2)$, results in a method where the face flux satisfies Gauss' theorem on the domain containing both cells touching that face, and the face value is linearly interpolated between the two cell values using the perpendicular distances (which is equivalent to assuming no variation in the cell temperatures tangential to the face).

Ultimately this calculation of the Laplacian term uses not only the cell neighbors, but the neighbors of those neighbors in the stencil, and this is how we choose the name. Calculating diffusion with this method is fairly expensive due to all the extra averaging involved (to get face temperatures and gradients). It also tends

to be less accurate because of all the smoothing that occurs in those averaging operators. Finally, the presence of averaging means that spurious modes are present. These are discrete solutions which satisfy the Laplace equation but are not linear functions. For example, on a Cartesian mesh, this method does not damp the 'checkerboard' mode.

5.1.3. Corrected

A compromise that is frequently used is to correct the basic method so that a normal flux is actually calculated. When the mesh is uniform the accuracy and small stencil of the basic method is recovered. When the mesh is strongly distorted the correction keeps the method convergent. In the corrected method the flux is calculated from,

$$Q_f = -\mathbf{q}_f \cdot \mathbf{n}_f - (kGT_{\tilde{n}} - \mathbf{q}_f \cdot \mathbf{d}) \frac{\mathbf{d} \cdot \mathbf{n}_f}{\mathbf{d} \cdot \mathbf{d}} \tag{19}$$

where $\mathbf{d} = \mathbf{x}_{c2} - \mathbf{x}_{c1}$ is the vector between the two cell positions, and $\mathbf{q}_f = -(w_1 k_1 \nabla T_{c1} + w_2 k_2 \nabla T_{c2})$ is calculated using the large stencil approach described previously. When $\mathbf{d}$ is aligned with the normal the first and last terms cancel and the basic method is recovered.

This approach does not produce a symmetric diffusion matrix. Often methods treat the basic part of the method implicitly (because it is symmetric) and the correction part explicitly. The stability of such a split method may be compromised on highly distorted meshes. This approach has roughly the same cost as the large stencil method, but often better accuracy due to only a small influence from the larger stencil, with the basic stencil dominating.

5.2. Tests of spatial accuracy

The first test compares the accuracy of the three traditional Finite Volume (FV) methods presented in the previous section, with the intent of choosing the best of these methods for comparison against the Discrete Calculus methods. The second test confirms that the Discrete Calculus methods are exact for linear functions even when the material properties are discontinuous across the domain. The third test proves the second-order convergence of the Discrete Calculus methods. The disadvantage of using the Voronoi dual mesh is illustrated in a fourth test problem.

For all the comparisons, the discrete L_2 error norm is adopted for verifying the order of convergence of the method,

$$L_2 = \left[\frac{1}{N} \sum_{n=1}^{N} (T_n - T_n^{\text{exact}})^2 \right]^{1/2} \tag{20}$$

where N refers to the number of unknowns, T_n refers to the numerical solution and T_n^{exact} refers to the analytical solution. In the cell-based methods T_n is $T_{\tilde{n}}$, the cell temperature value.

5.2.1. Comparison of FV methods

The steady state heat diffusion equation ($\nabla \cdot \mathbf{q} = 0$) on a unit square domain as shown in Fig. 5 is used as a test case with the following boundary conditions,

$$\begin{aligned} x &= 0 \quad T = 0 \\ x &= 1 \quad T = 1 \\ y &= 0 \quad \frac{\partial T}{\partial y} = 0 \\ y &= 1 \quad \frac{\partial T}{\partial y} = 0 \end{aligned} \tag{21}$$

The mesh size (defined as $\mathrm{d}x = (\mathrm{Vol}/\mathrm{NC})^{1/\mathrm{ND}}$ where Vol is the entire domain volume, NC is the total number of cells in the domain and ND is the dimensionality of the problem) is plotted against discrete L_2 error norm in Fig. 6 for the three FV methods presented in Section 5.1.

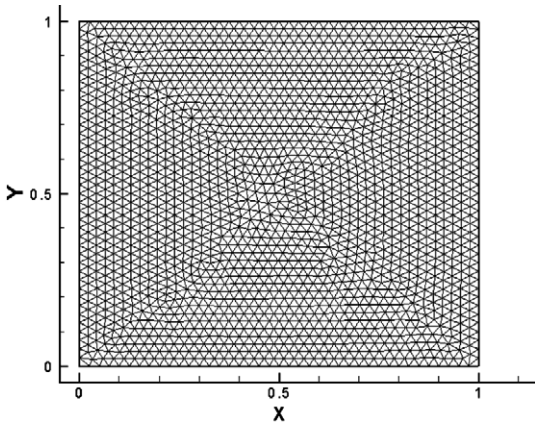

Fig. 5. Typical mesh used for convergence studies.

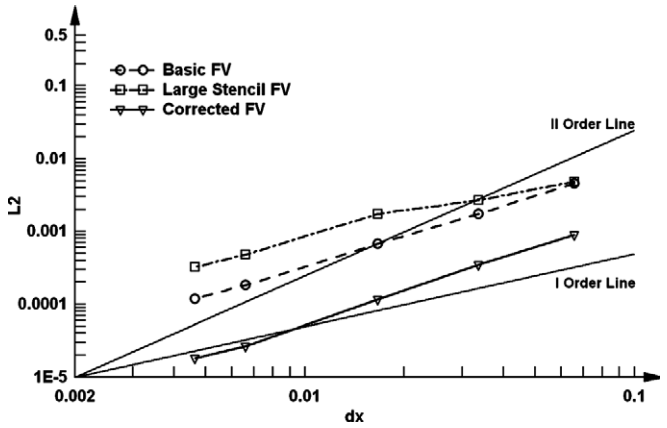

Fig. 6. Spatial accuracy of classical FV methods.

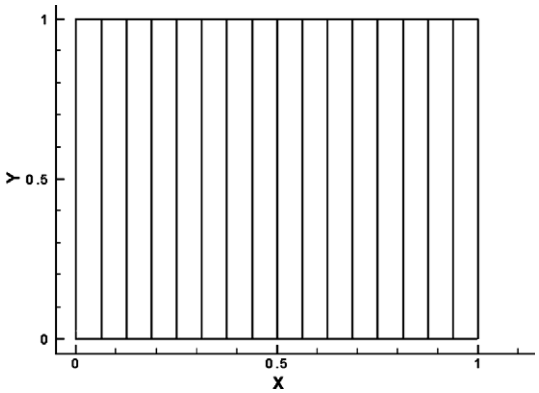

Fig. 7. Isolines of solution.

It is observed from Fig. 6 that all the FV methods tends to be first-order accurate, and the most accurate of these (the Corrected FV method) will be subsequently used for comparing against the Discrete Calculus methods. All the Discrete Calculus methods described in this paper were verified to achieve the exact answer to machine precision for this problem. Typical solution contours are presented in Fig. 7.

5.2.2. Discontinuous conductivity at an angle

This problem is taken from Shashkov [35] and Morel et al. [36]. 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 mesh (shown in Fig. 8) 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. The diffusion coefficients are defined as,

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

Dirichlet boundary conditions are enforced such that the exact steady state solution is

$$T = \begin{cases} a + bx + cy & 0 \leqslant x \leqslant 0.5 \\ a - b\frac{k_1 - k_2}{2k_2} + b\frac{k_1}{k_2}x + cy & 0.5 < x \leqslant 1 \end{cases} \tag{23}$$

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_{1c} on the left side and k_{2c} on the right side of the interface. The numerical experiments employ $a = b = c = 1$, $k_1 = 4$ and $k_2 = 1$. The boundary conditions are applied to the boundaries as shown below:

$$\begin{aligned} &x = 0 \quad T = 1 + y \\ &x = 1 \quad T = \frac{7}{2} + y \\ &y = 0, \quad 0 < x < 0.5 \quad T = 1 + x \\ &y = 1, \quad 0 < x < 0.5 \quad T = 2 + x \\ &y = 0, \quad 0.5 \leqslant x < 1 \quad T = 4x - 0.5 \\ &y = 1, \quad 0.5 \leqslant x < 1 \quad T = 4x + 0.5 \end{aligned} \tag{24}$$

The calculated temperature isolines for this problem are shown in Fig. 9. The solutions obtained by all the Discrete Calculus methods agree with the exact answer to machine precision.

5.2.3. Quadratic problem

In the third test problem, the spatial accuracy of the Discrete Calculus methods is compared with the corrected FV method in a steady-state heat diffusion problem with a uniform source term $S = 4$ and unit conductivity. The same mesh employed in Section 5.2.1 (Fig. 5) is used. Dirichlet boundary conditions are imposed on the left and right boundaries, and homogeneous Neumann boundary conditions are imposed on the top and

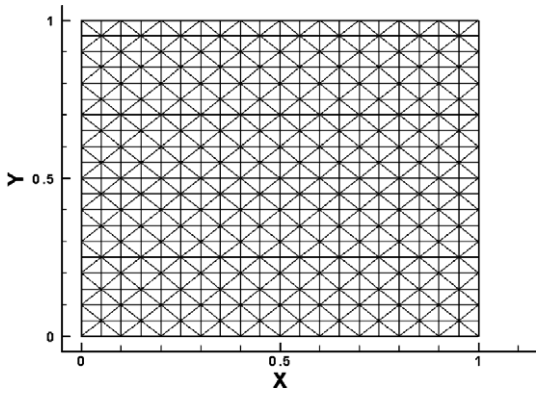

Fig. 8. Mesh with different diffusivities on either side of the interface ($x = 0.5$).

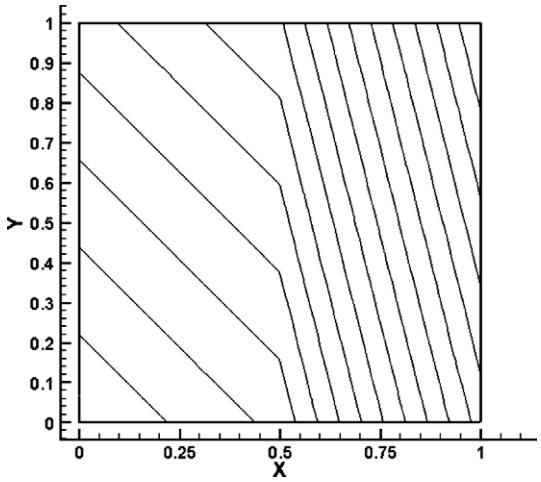

Fig. 9. Isolines of temperature contours obtained by the Discrete Calculus methods.

bottom boundaries. The exact solution $T(x) = 2x^2 - 2x$ is a quadratic function. The isolines are shown in Fig. 10 and the spatial accuracy is plotted in Fig. 11.

It is seen from Fig. 11 that all the Discrete Calculus methods exhibit close to second-order convergence (because the mesh is very close to uniform). It is to be noted that the Cell (Median Dual) method presented

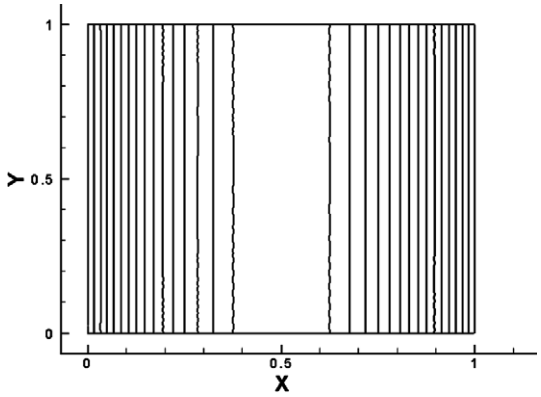

Fig. 10. Isolines of temperature contours for a non-linear problem.

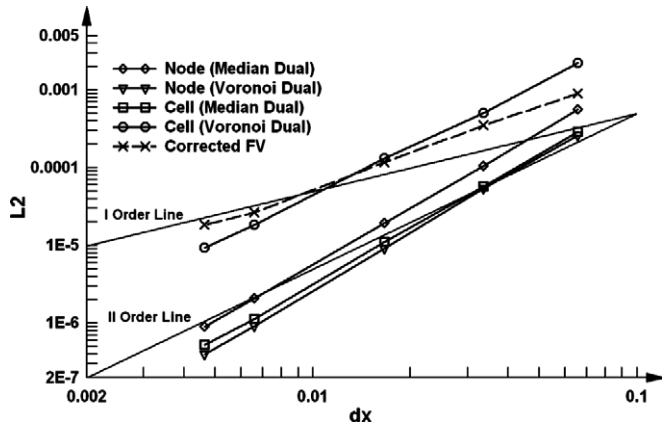

Fig. 11. Spatial accuracy of Discrete Calculus methods.

in Fig. 11 is the Mixed method described in Section 4.3. While both the median dual methods are more accurate than the classical FV methods, the mixed Voronoi dual method tends to be less accurate than the FV method on coarse meshes. As pointed out in Sections 3.3.1 and 4.2, the Voronoi dual mesh methods tend to be less accurate if the circumcenter and the centroid of the cells do not coincide. The fact that the Voronoi methods show second-order convergence in Fig. 11 is attributed to the good quality mesh that was employed for the test. However, when the mesh quality decreases, even for a Delaunay mesh, the circumcenter of the cell can sometimes lie outside of the cell causing large errors in the solution. In order to illustrate this behavior, the subsequent test compares the Median and Voronoi dual methods on a stretched mesh.

5.2.4. Median dual vs Voronoi dual methods on a stretched mesh

In this test, a series of stretched meshes (as shown in Fig. 12) is used to compare the accuracy of the Median and Voronoi Discrete Calculus methods employing the linear problem of Section 5.2.1. While the median-dual Discrete Calculus methods are still exact to machine precision on these meshes, the Voronoi dual methods show significant errors as depicted by Fig. 13 (note that this figure does not show the Medial Dual curves as they are exact).

It is seen from Fig. 13 that the errors are not only large even for a linear problem, but the errors seem to increase as the mesh becomes finer. This is due to the fact that as the mesh gets finer, there are more inverted Voronoi cells with a resulting increase in the overall error. It is for this reason that the added complexity and cost of the Median dual methods may be advantageous in complex geometries.

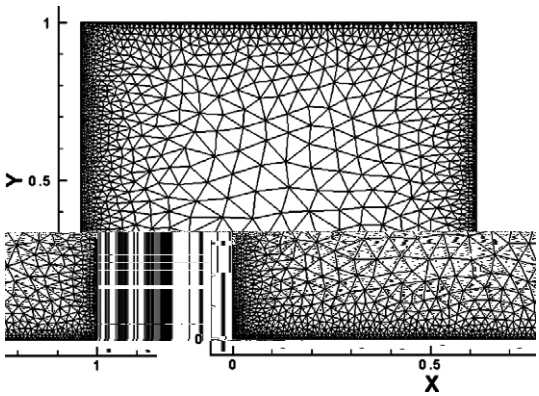

Fig. 12. Stretched mesh to compare median dual and Voronoi dual methods.

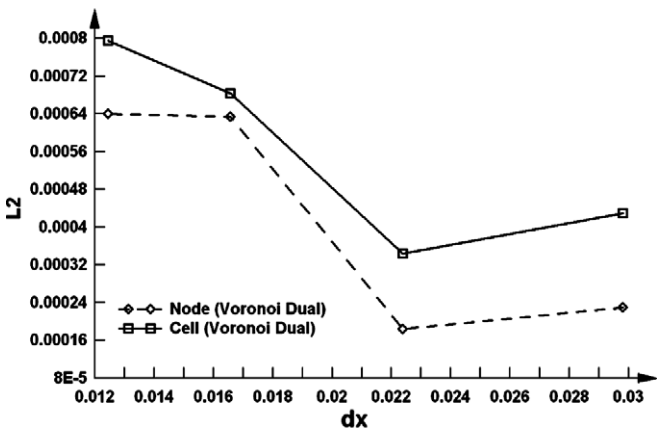

Fig. 13. Accuracy of Voronoi dual methods on stretched meshes.

5.3. Tests of temporal accuracy

While any time advancement scheme can be used with the Discrete Calculus methods, treating the diffusion term in a fully implicit manner would lead to a first-order temporal accuracy. In order to obtain a second-order temporal accuracy, it is preferable to employ the trapezoidal rule. As an example, Eq. (15) for the Mixed Method may be rewritten as,

$$\begin{bmatrix} \frac{1}{\Delta t} M_{\rho CV_c} & -\alpha D \\ G & (M_{kA_f/L_{\tilde{e}}})^{-1} \end{bmatrix} \begin{pmatrix} T_{\tilde{n}}(t^{n+1}) \\ Q_f(t^{n+1}) \end{pmatrix} = \begin{pmatrix} \frac{1}{\Delta t} M_{\rho CV_c} T_{\tilde{n}}(t^n) + (1-\alpha) D Q_f(t^n) \\ T_f^{\mathrm{BC}} \end{pmatrix} \tag{25}$$

where α is the "implicitness" of the diffusion term. With $\alpha = 1$, the time advancement scheme becomes fully implicit and with $\alpha = 0.5$, the scheme turns into a trapezoidal rule, which is employed for this test.

In this test, an unsteady diffusion equation is solved on the mesh considered in Section 5.2.1. The initial conditions and boundary conditions are specified as follows:

$$\begin{aligned} &\text{ICs} \quad T(x,t_0) = \frac{1}{\sqrt{t_0}} \exp\left\{-\frac{(x-0.5)^2}{4kt_0}\right\}; \quad Q_f(x,t_0) = \frac{(x-0.5)n_x}{2t_0\sqrt{t_0}} \exp\left\{-\frac{(x-0.5)^2}{4kt_0}\right\} \\ &\text{BCs} \quad x = 0 \quad T(t) = 0 \\ &\qquad\quad x = 1 \quad T(t) = 0 \\ &\qquad\quad y = 0 \quad \frac{\partial T(t)}{\partial y} = 0 \\ &\qquad\quad y = 1 \quad \frac{\partial T(t)}{\partial y} = 0 \end{aligned} \tag{26}$$

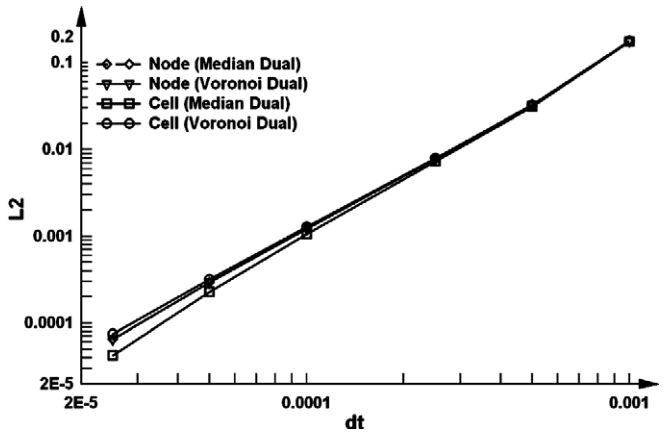

Fig. 14. Solution to the unsteady diffusion problem at $t = 0.002$.

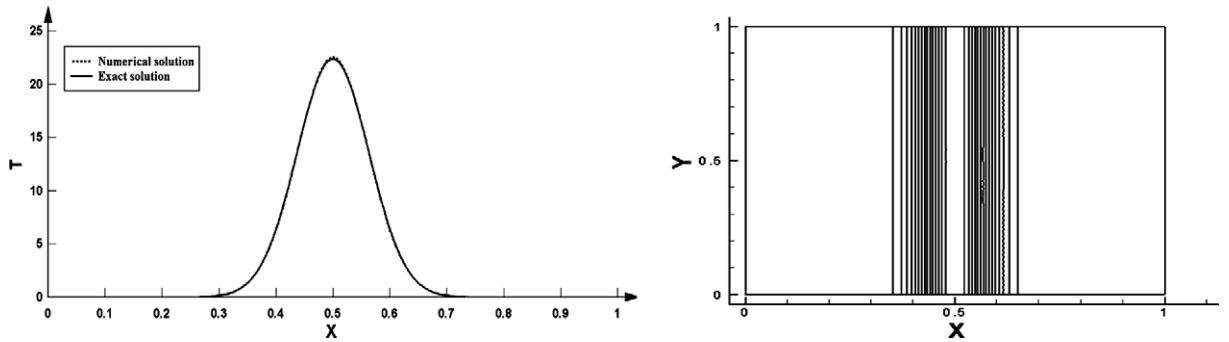

Fig. 15. Temporal accuracy of DC methods.

In Eq. (26), n_x refers to the x-component of the face normal vector at each face and k refers to the diffusivity (chosen as unity for this problem). The initial time t_0 is chosen as 0.001 and the simulation is run up to a final time of $t = 0.002$. The analytical solution to this problem is $T(x,t) = \frac{1}{\sqrt{t}} \exp\left\{-(x-0.5)^2/4kt\right\}$. The simulation is run with various values of the time step (dt) and the L2 norm of the error is computed at the same final time, $t = 0.002$. The result is plotted in Fig. 14, which confirms the second-order temporal accuracy of all of the four Discrete Calculus methods. The computed solution is compared with the exact answer at $t = 0.002$ in Fig. 15.

5.4. Computational cost for a desired accuracy

Although the Discrete Calculus methods were shown to be more accurate than the traditional lower-order methods in the previous tests, it might be more important to compare the cost-effectiveness of the Discrete Calculus methods against the classical methods. Hence, the computational cost (in terms of the CPU time) is plotted against the L2 error norm in Fig. 16 for the problem considered in Section 5.2.3, which really compares the cost incurred for a desired accuracy level.

Fig. 16 compares the median dual Discrete Calculus methods with the Corrected FV method. It is observed that for any given accuracy level the Discrete Calculus methods are always more cost-effective than the traditional methods. Also, it is clearly seen that the cost for the Corrected FV method tends to increase more rapidly than the Discrete Calculus methods as the need for accuracy increases. In Fig. 16, the other two FV methods are not used for comparison in order to enhance clarity. However, Fig. 17 compares the cost of the three FV methods verifying that the Corrected FV method is the most cost-effective method of the three FV methods considered.

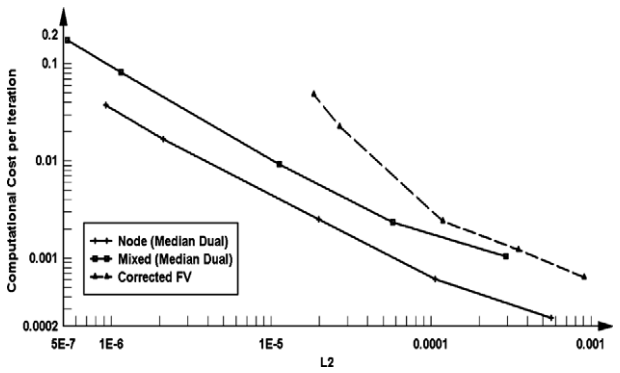

Fig. 16. Computational cost incurred for a desired accuracy level.

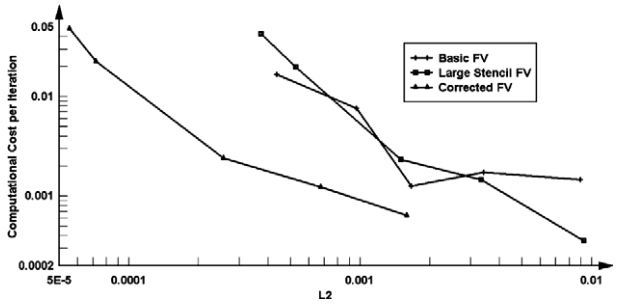

Fig. 17. Cost incurred by FV methods.

5.5. Heat transfer in a crank shaft

All the Discrete Calculus methods presented are applicable on any general 3D unstructured mesh, although the previous tests were run on 2D geometries. In order to illustrate this, a more realistic problem is considered in this section, which involves solving Eq. (2) on a complex 3D geometry. A typical mesh considered for the analysis is shown in Fig. 18. The coarsest mesh considered has 864 nodes and 2339 cells and the finest mesh contains 73,875 nodes and 360,512 cells. Fixed temperature (Dirichlet) boundary conditions are applied to the inlet and outlet faces (crankshaft ends) and the sides are insulated. Typical temperature contours are presented in Fig. 19.

The heat flux through the inlet and outlet faces (which were verified to be equal) are measured against the mesh size for the median-dual Discrete Calculus methods and extrapolated to obtain the "exact" heat flux.

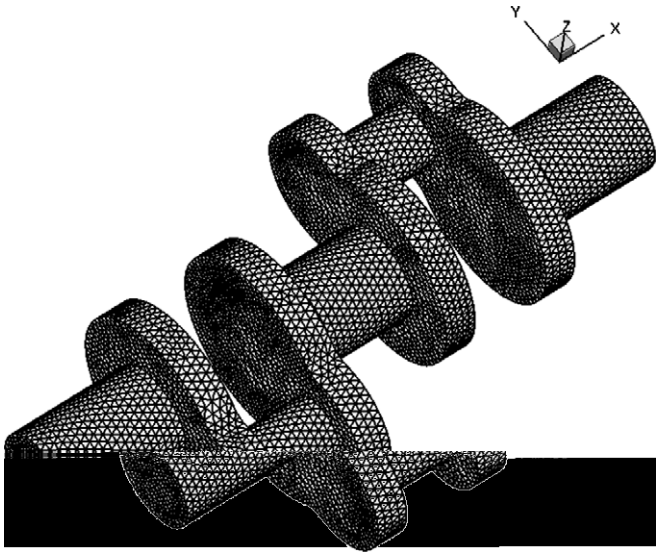

Fig. 18. Crank shaft mesh.

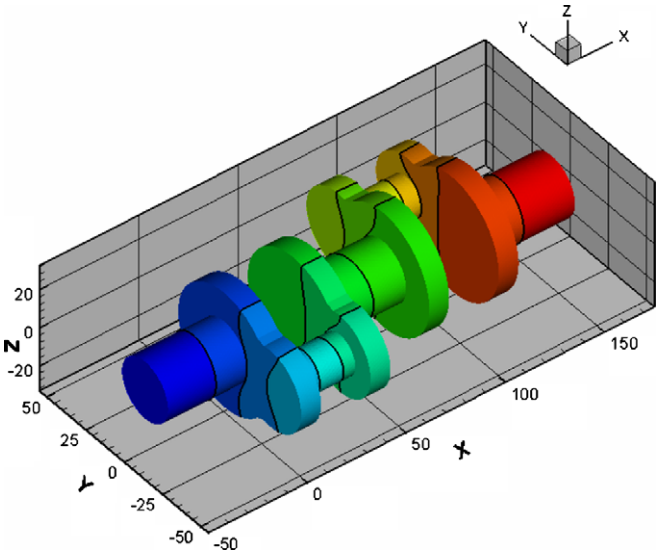

Fig. 19. Temperature contours along the crankshaft.

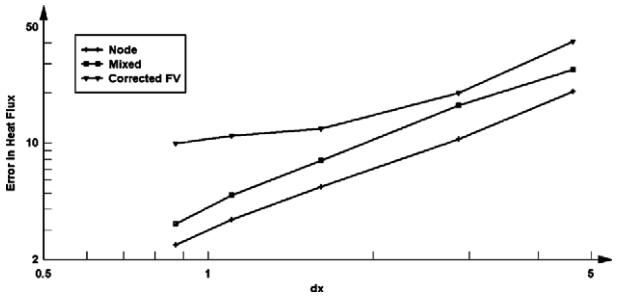

Fig. 20. Accuracy of Discrete Calculus and finite volume methods.

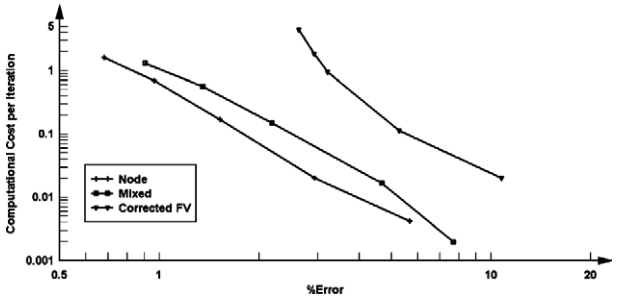

Fig. 21. Cost for a desired accuracy for the crank shaft test case.

This exact heat flux is then employed to compute the error in the finite volume and Discrete Calculus methods (Fig. 20). The mesh size (dx) is computed as the cube root of the average cell volume.

The computational time taken per solver iteration is then plotted against the percentage error, which gives the cost required to obtain a certain accuracy level (Fig. 21), which is in agreement with the results of the previous section. It is interesting to note that the cell based and node Discrete Calculus methods give very similar accuracy/cost performance despite the fact that the cell based method uses over fifteen more unknowns than the node based method for the same mesh.

6. Discussion

The Discrete Calculus methods, as developed in this work, have the philosophical flavor of finite volume methods. In particular, the algorithms are independent of cell shape and can be applied to arbitrary polygonal meshes. There is no need to explicitly specify the basis functions or interpolation functions being used. Finally, like finite volume methods there is a local energy conservation statement for each cell or dual cell. However, there is a close relationship to Finite Element methods. For the median dual mesh, both the nodal and cell based methods are closely related to certain finite element discretizations. The route to higher-order [19] also is similar to finite element methods, since more unknowns, not a larger stencil, is used. In this paper we have presented the finite volume looking versions of the method, but it should be noted that Galerkin Finite Element methods can also be constructed using the Discrete Calculus approach by exactly discretizing the weak form of the equations (see [18]).

The Discrete Calculus approach is an attempt to develop a methodology for developing numerical methods that capture physics well. The key is to exactly discretize the physics and calculus 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 equations where physical approximation has already been performed.

Tests of these Discrete Calculus methods demonstrate the theoretically predicted order of accuracy, the ability to accurately capture solutions with sharp discontinuities in the material properties, and a better accuracy/cost than classic low-order diffusion methods.

Acknowledgments

Partial financial support for this work was provided by the Office of Naval Research (Grant No. N00014-01-1-0267), the Air Force Office of Scientific Research (Grant No. FA9550-04-1-0023), and the National Science Foundation (Grant No. CTS-0522089).

References

[1] D. Vidovic, A. Segal, P. Wesseling, A superlinearly convergent finite volume method for the incompressible Navier–Stokes equations, J. Comput. Phys. 198 (2004) 159–177.
[2] I. Wenneker, A. Segal, P. Wesseling, Conservation properties of a new unstructured staggered scheme, Comput. Fluids 32 (2003) 139–147.
[3] I. Wenneker, A. Segal, P. Wesseling, A Mach-uniform unstructured staggered grid method, Int. J. Numer. Meth. Fluids 40 (2002) 1209–1235.
[4] D.R. van der Heul, C. Vuik, P. Wesseling, A staggered scheme for hyperbolic conservation laws applied to computation of flow with cavitation, in: E.F. Toro (Ed.), Godunov Methods: Theory and Applications, Kluwer Academic, New York, 2001, pp. 969–976, 1999.
[5] H. Bijl, P. Wesseling, Computation of unsteady flows at all speeds with a staggered scheme, in: E. Oñate, G. Bugeda, B. Suárez (Eds.), ECCOMAS 2000, Proceedings of the European Congress on Computational Methods in Applied Sciences and Engineering, Barcelona, September 2000.
[6] H. Bijl, Computation of flow at all speeds with a staggered scheme, Ph.D. Thesis, Delft University of Technology, 1999.
[7] F.H. Harlow, J.E. Welch, Numerical calculations of time dependent viscous incompressible flow of fluid with a free surface, Phys. Fluids 8 (12) (1965) 2182–2189.
[8] P. Wesseling, A. Segal, C.G.M. Kassels, H. Bijl, Computing flows on general two-dimensional nonsmooth staggered grids, J. Eng. Math. 34 (1998) 21–44.
[9] H. Bijl, P. Wesseling, A unified method for computing incompressible and compressible flows in boundary fitted coordinates, J. Comput. Phys. 141 (1998) 153–173.
[10] R.A. Nicolaides, The covolume approach to computing incompressible flow, in: M.D. Gunzburger, R.A. Nicolaides (Eds.), Incompressible Computational Fluid Dynamics, Cambridge University Press, 1993, pp. 234–295.
[11] J.M. Hyman, M. Shashkov, The orthogonal decomposition theorems for mimetic finite difference methods, SIAM J. Num. Anal. 36 (3) (1999) 788–818.
[12] J.B. Perot, X. Zhang, Reformulation of the unstructured staggered mesh method as a classic finite volume method, Finite Volumes for Complex Applications II, Hermes Science Publications, 1999, pp. 263–270.
[13] J.B. Perot, Conservation properties of unstructured staggered mesh schemes, J. Comput. Phys. 159 (2000) 58–89.
[14] 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.
[15] D.K. Lilly, On the computational stability of numerical solutions of time-dependent non-linear geophysical fluid dynamics problems, Mon. Weather Rev. 93 (1) (1965) 11–26.
[16] 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.
[17] W. Chang, F. Giraldo, J.B. Perot, Analysis of an exact fractional step method, J. Comput. Phys. 179 (2002) 1–17.
[18] C. Mattiussi, An analysis of finite volume, finite element, and finite difference methods using some concepts from algebraic topology, J. Comput. Phys. 133 (1997) 289–309.
[19] V. Subramanian, J.B. Perot, Higher order mimetic methods for unstructured meshes, J. Comput. Phys. 219 (2006) 68–85.
[20] R. Nicolaides, X. Wu, Covolume solutions of three-dimensional div–curl equations, SIAM J. Num. Anal. 34 (1997) 2195–2203.
[21] R.A. Nicolaides, Direct discretization of planar div–curl problems, ICASE Report 89-76, 1989.
[22] J.B. Perot, D. Vidovic, P. Wesseling, Mimetic reconstruction of vectors, in: D.N. Arnold, P.B. Bochev, R.B. Lehoucq, R.A. Nicolaides, M. Shashkov (Eds.), IMA Volumes in Mathematics and its Applications, Compatible Spatial Discretizations, vol. 142, Springer, New York, 2006.
[23] W. Chang, F. Giraldo, J.B. Perot, Analysis of an exact fractional step method, J. Comput. Phys. 179 (2002) 1–17.
[24] M. Desbrun, A. Hirani, M. Leok, J. Marsden, Discrete Exterior Calculus, www.citebase.org/abstract?id=oai:arXiv.org:math/0508341, 2005.
[25] G. de Rham, Sut l'analysis situs des varietes a n dimensions, J. Math. 10 (1931) 115–199.
[26] A. Bossavit, I. Mayergoyz, Edge elements for scattering problems, IEEE Trans. Mag. 25 (4) (1989) 2816–2821.
[27] D. White, Orthogonal vector basis functions for time domain finite element solution of the vector wave equation, in: Proceedings of the 8th Biennial IEEE Conference on Electromagnetic Field Computation, Tucson, AZ, 1998, UCRL-JC-129188.
[28] P. Castillo, J. Koning, R.Rieben, M. Stowell, D. White, Discrete Differential Forms: A Novel Methodology for Robust Computational Electromagnetics, LLNL Report UCRL-ID-151522, January 2003.
[29] G. Rodrigue, D. White, A vector finite element time domain method for solving Maxwells equations on unstructured hexahedral grids, SIAM J. Sci. Comput. 23 (3) (2001) 683–706.

[30] T.J. Barth, P.O. Frederickson, Higher Order Solution of the Euler Equations on Unstructured Grids Using Quadratic Reconstruction, AIAA 90-0013, 1990.
[31] T.J. Barth, Recent Improvements in High Order K-exact Reconstruction on Unstructured Meshes, AIAA 93-0668, 1993.
[32] N. Sukumar, Voronoi cell finite difference method for the diffusion operator on arbitrary unstructured grids, Int. J. Num. Meth. Eng. 57 (2003) 1–34.
[33] J.C. Cavendish, C.A. Hall, T.A. Porsching, A complementary volume approach for modeling three-dimensional Navier–Stokes equations using dual Delaunay/Voronoi tessellations, Int. J. Num. Meth. Heat Fluid Flow 4 (1994) 329–345.
[34] C.A. Hall, J.S. Peterson, T.A. Porsching, F.R. Sledge, The dual variable method for finite element discretizations of Navier/Stokes equations, Int. J. Num. Meth. 21 (1985) 883–898.
[35] M. Shashkov, S. Steinberg, Solving diffusion equations with rough coefficients in rough grids, J. Comput. Phys. 129 (1996) 383–405.
[36] 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.
[37] P. Wesseling, Principals of Computational Fluid Dynamics, Springer, Berlin, 2001, ISBN 3-540-67853-0.