Journal of Computational Physics **181,** 729–752 (2002)
doi:10.1006/jcph.2002.7159

A High-Order-Accurate Unstructured Mesh Finite-Volume Scheme for the Advection–Diffusion Equation

Carl Ollivier-Gooch and Michael Van Altena

*University of British Columbia, Department of Mechanical Engineering, 2324 Main Mall,
Vancouver, BC V6T 1Z4, Canada*
E-mail: cfog@mech.ubc.ca and mva@alumni.uwaterloo.ca

Received February 7, 2002; revised June 27, 2002

High-order-accurate methods for viscous flow problems have the potential to reduce the computational effort required for a given level of solution accuracy. The state of the art in this area is more advanced for structured mesh methods and finite-element methods than for unstructured mesh finite-volume methods. In this paper, we present and analyze a new approach for high-order-accurate finite-volume discretization for diffusive fluxes that is based on the gradients computed during solution reconstruction. Our analysis results show that our schemes based on linear and cubic reconstruction can be reasonably expected to achieve second- and fourth-order accuracy in practice, respectively, while schemes based on quadratic reconstruction are expected to be only second-order accurate in practice. Numerical experiments show that in fact nominal accuracy is attained in all cases for two advection–diffusion problems, provided that curved boundaries are properly represented. To enforce boundary conditions on curved boundaries, we introduce a technique for constraining the least-squares reconstruction in boundary control volumes. Simply put, we require that the reconstructed solution satisfy the boundary condition exactly at all boundary flux integration points. Numerical experiments demonstrate the success of this approach, both in the reconstruction results and in simulation results. © 2002 Elsevier Science (USA)

Key Words: high-order accuracy; unstructured meshes; advection–diffusion equation; high-order reconstruction; boundary conditions; boundary constraints; Navier–Stokes equations.

1. INTRODUCTION

High-order-accurate methods for viscous flow problems—methods of higher than second-order accuracy—are being actively pursued in an effort to reduce computational effort for

0021-9991/02 $35.00
© 2002 Elsevier Science (USA)
All rights reserved.

a given quality of solution. For example, in a recent comparative study, Zingg *et al.* showed that high-order-accurate methods are indeed more efficient in obtaining accurate solutions on structured meshes for compressible turbulent aerodynamic flows than are second-order methods [18]. In a similar vein, Bassi and Rebay showed excellent accuracy for high-order-accurate finite-element methods for compressible inviscid and laminar viscous flows, although without detailed cost comparisons [5].

High-order-accurate method development for unstructured mesh finite-volume computations has historically focused on advective fluxes. This work dates back at least as far as Barth and Frederickson's work applying quadratic reconstruction techniques to the Euler equations ([3, 4], the former perhaps being more widely available). Since then, there have been advances in high-order-accurate unstructured ENO and weighted ENO schemes (e.g., [1, 8, 9, 14, 15]). Most recently, a high-order compact unstructured mesh scheme has been proposed for the Euler equations [2].

Development of high-order-accurate viscous flow methods has been slower. As an example, recent structured mesh work with the well-established INS3D code uses third- and fifth-order-accurate differencing for the convective terms, but only second-order-accurate discretization of the viscous terms [12]. Unstructured mesh schemes often use a mixed formulation, with a finite-volume scheme for the inviscid terms combined with a second-order-accurate Galerkin finite-element approximation for the viscous terms (see, for example, [3, 13]). Recently, Delanaye and co-workers used quadratic reconstruction to compute the inviscid terms in the Navier–Stokes equations, combined with a different, linear reconstruction for the gradients needed in the viscous terms [7, 10]. In neither of these references, however, do the authors provide accuracy analysis or careful convergence studies for the viscous terms in their scheme.

Our long-term goal is to develop a reliable high-order-accurate discretization scheme for viscous flows and other mathematically similar problems. The purpose of this paper is to describe a new unstructured mesh finite-volume scheme for the advection–diffusion equation and to present analysis and numerical experiments demonstrating that the new scheme attains high-order accuracy. This is an extension and enhancement of our previous work on the same topic [16, 17]. In particular, we present two significant results.

1. A new approach for using data from a single-solution reconstruction to compute not only advective fluxes but also diffusive fluxes (Section 2.3). While we focus on least-squares reconstruction, our technique could be applied in combination with other reconstruction schemes. Numerous previous authors have analyzed advective flux schemes, and so we focus our analysis efforts on diffusive fluxes. We present analysis of the truncation error in applying our approach (with least-squares reconstruction) to the Laplacian operator for both vertex-centered (Section 3.1) and cell-centered (Section 3.2) control volumes.
2. A new approach to boundary condition enforcement. We add constraints to the least-squares reconstruction in boundary control volumes so that the boundary conditions are satisfied exactly at flux integration points on the boundary (Section 2.2). This approach is flexible enough to be applied for vertex- or cell-centered control volumes, for all orders of accuracy, and for straight or curved boundaries.

In Section 4 we present the results of several numerical experiments which demonstrate the effectiveness of the innovations we propose.

2. HIGH-ORDER-ACCURATE SOLUTION RECONSTRUCTION AND FLUX INTEGRATION

In this section, we describe our approach to high-order-accurate solution reconstruction, boundary condition enforcement, and flux integration. We begin our discussion by describing how we perform least-squares reconstruction in the interior of the mesh. Next, we discuss how we enforce boundary conditions by imposing additional constraints on the reconstruction in boundary control volumes. We close this section with a summary of how we calculate both advective and diffusive fluxes, and how we integrate fluxes around control volume boundaries.

2.1. Least-Squares Reconstruction for Interior Control Volumes

Our goal, regardless of whether we use cell-centered or vertex-centered control volumes, is to replace the control volume averaged value of the solution $\bar{\phi}_i$ with a Taylor expansion,

$$\phi_i^R(x-x_i, y-y_i) = \phi|_i + \left.\frac{\partial\phi}{\partial x}\right|_i (x-x_i) + \left.\frac{\partial\phi}{\partial y}\right|_i (y-y_i) + \left.\frac{\partial^2\phi}{\partial x^2}\right|_i \frac{(x-x_i)^2}{2} + \left.\frac{\partial^2\phi}{\partial x\partial y}\right|_i (x-x_i)(y-y_i) + \left.\frac{\partial^2\phi}{\partial y^2}\right|_i \frac{(y-y_i)^2}{2} + \cdots,$$

where ϕ_i is the value of the reconstructed solution and $\frac{\partial^{k+l}\phi_i}{\partial x^k \partial y^l}$ are its derivatives at the reference point (x_i, y_i) of control volume i. We choose the coefficients of the expansion to conserve the mean value in the control volume and to minimize error in representing smooth solutions. Once the reconstruction is known, values of the reconstructed solution and its derivatives can be easily calculated anywhere within the control volume.

2.2.1. *Conservation of the Mean*

Conservation of the mean within a control volume requires that

$$\frac{1}{A_i}\int_{V_i} \phi_i^R \, dA = \bar{\phi}_i. \tag{1}$$

Expanding the left-hand side of this term by term, one can easily show (see, for example, [15]) that

$$\frac{1}{A_i}\int_{V_i} \phi_i^R \, dA = \phi|_i + \left.\frac{\partial\phi}{\partial x}\right|_i \bar{x}_i + \left.\frac{\partial\phi}{\partial y}\right|_i \bar{y}_i + \left.\frac{\partial^2\phi}{\partial x^2}\right|_i \frac{\overline{x^2}_i}{2} + \left.\frac{\partial^2\phi}{\partial x\partial y}\right|_i \overline{xy}_i + \left.\frac{\partial^2\phi}{\partial y^2}\right|_i \frac{\overline{y^2}_i}{2} + \cdots, \tag{2}$$

where

$$\overline{x^n y^m}_i \equiv \frac{1}{A_i}\int_{V_i} (x-x_i)^n (y-y_i)^m \, dA. \tag{3}$$

In contrast to most previous work in least-squares reconstruction (e.g., [3, 15]), we will include this mean constraint explicitly in the least-squares system rather than eliminating it analytically. For interior control volumes, analytic elimination of the mean constraint is straightforward, but for boundary control volumes, where the solution must also be constrained to enforce boundary conditions (see Section 2.2), analytic elimination of the mean constraint would be much more difficult. By including the mean constraint explicitly for interior control volumes, we accept a small performance penalty in favor of code simplification. This penalty is in fact small: the cost of Gauss elimination for a single constraint is approximately the same as applying a single Householder transform, so at most we pay a penalty equivalent to adding one additional column to the least-squares problem. In practice, this means that our scheme is comparable in efficiency to other least-squares reconstruction schemes, especially for high-order-accurate reconstruction. Comparisons between ENO schemes and least-squares schemes on unstructured meshes are absent from the literature, but we expect that the cost of our reconstruction scheme should be significantly less than the cost of computing several (non-least-squares) reconstructions, then computing the proper weights to use in combining them, as modern weighted ENO schemes do (see, for example, [9]).

2.1.2. *Accuracy for Smooth Functions*

Accuracy of the reconstruction for smooth functions can be stated in two equivalent ways. The reconstruction can be said to be k-exact, or $(k+1)$-order accurate, if, when reconstructing $P(\vec{x}) \in \{x^m y^n : m+n \leq k\}$,

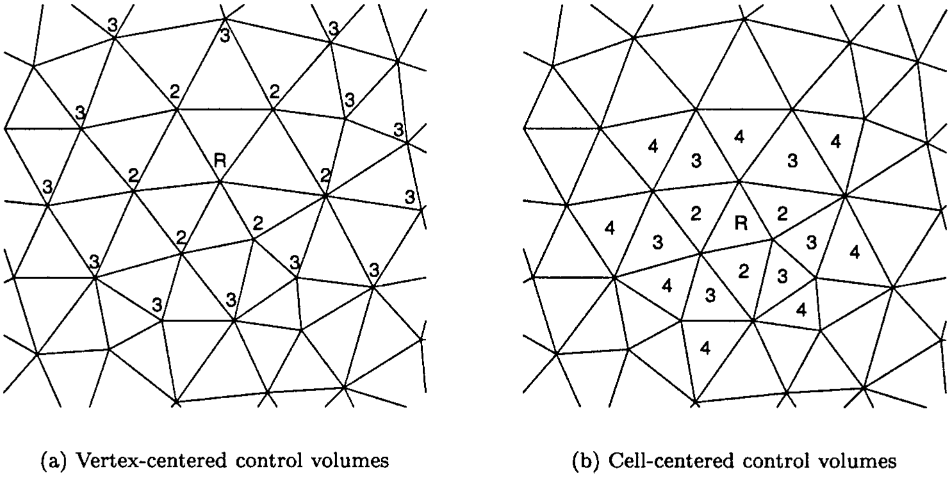
$$\phi_i^R(\vec{x}-\vec{x}_i) \equiv P(\vec{x}). \tag{4}$$

Equivalently, one can say that for any $\phi(\vec{x})$ that has been averaged over control volumes and reconstructed,

$$\phi_i^R(\vec{x}-\vec{x}_i) = \phi(\vec{x}) + \mathcal{O}(\Delta x^{k+1}). \tag{5}$$

In practice, this accuracy requirement means that the modified Taylor series expansion of ϕ_i^R given in Eq. (2) must be carried out through the kth derivatives.

To compute these derivatives, we seek to minimize the error in predicting the mean value of the function for control volumes in the stencil $\{V_j\}_i$. The minimum number of neighboring control volumes in the reconstruction stencil is equal to the number of derivative terms to be approximated. In accord with common practice, we choose to exceed that minimum to give some freedom for the least-squares reconstruction, improving robustness. Specifically, in practice we insist on three neighbors for second-order accuracy, nine for third order, and fourteen for fourth order. Control volumes are added to the reconstruction stencil based on thier topological proximity to the reconstruction control volume. All neighbors at a given level are added at once. Figure 1 gives examples of stencils for both vertex- and cell-centered control volumes in the interior of a mesh. Each figure shows the stencil for reconstruction in the control volume labeled R; the numeric labels indicate the order of accuracy at which a given control volume is added to the stencil. Note that the vertex-centered example does not require any additional control volumes in the stencil for fourth-order (cubic) reconstruction that are not already present for third-order (quadratic) reconstruction. At boundaries, stencils



(a) Vertex-centered control volumes

(b) Cell-centered control volumes

FIG. 1. Sample reconstruction stencils.

are constructed using the same principles, although more layers of neighbors are typically required to get large enough stencils.

The mean value, for a single control volume V_j, of the reconstructed function ϕ_i^R is

$$\begin{aligned}\frac{1}{A_j}\int_{V_j}&\phi_i^R(\vec{x}-\vec{x}_i)\,dA\\ &=\phi|_i+\left.\frac{\partial\phi}{\partial x}\right|_i\frac{1}{A_j}\int_{V_j}(x-x_i)\,dA+\left.\frac{\partial\phi}{\partial y}\right|_i\frac{1}{A_j}\int_{V_j}(y-y_i)\,dA\\ &\quad+\left.\frac{\partial^2\phi}{\partial x^2}\right|_i\frac{1}{2A_j}\int_{V_j}(x-x_i)^2\,dA+\left.\frac{\partial^2\phi}{\partial x\partial y}\right|_i\frac{1}{A_j}\int_{V_j}(x-x_i)\,(y-y_i)\,dA\\ &\quad+\left.\frac{\partial^2\phi}{\partial y^2}\right|_i\frac{1}{2A_j}\int_{V_j}(y-y_i)^2\,dA+\cdots.\end{aligned}$$

To avoid computing moments of each control volume in $\{V_j\}_i$ about v_i, we replace $x-x_i$ and $y-y_i$ with $(x-x_j)+(x_j-x_i)$ and $(y-y_j)+(y_j-y_i)$, respectively. Expanding and integrating,

$$\begin{aligned}\frac{1}{A_j}\int_{V_j}&\phi_i^R(\vec{x}-\vec{x}_i)\,dA\\ &=\phi_i+\left.\frac{\partial\phi}{\partial x}\right|_i(\bar{x}_j+(x_j-x_i))+\left.\frac{\partial\phi}{\partial y}\right|_i(\bar{y}_j+(y_j-y_i))\\ &\quad+\left.\frac{\partial^2\phi}{\partial x^2}\right|_i\frac{\overline{x^2}_j+2\bar{x}_j(x_j-x_i)+(x_j-x_i)^2}{2}+\left.\frac{\partial^2\phi}{\partial x\partial y}\right|_i(\overline{xy}_j+\bar{x}_j(y_j-y_i)+(x_j-x_i)\bar{y}_j\\ &\quad+(x_j-x_i)(y_j-y_i))+\left.\frac{\partial^2\phi}{\partial y^2}\right|_i\frac{\overline{y^2}_j+2\bar{y}_j(y_j-y_i)+(y_j-y_i)^2}{2}+\cdots.\end{aligned}$$

The geometric terms in this equation are of the general form

$$\begin{aligned}\widehat{x^n y^m}_{ij} &\equiv \frac{1}{A_j}\int_{V_j} ((x-x_j)+(x_j-x_i))^n \cdot ((y-y_j)+(y_j-y_i))^m \, dA \\ &= \sum_{l=0}^{m}\sum_{k=0}^{n} \binom{m}{l}\binom{n}{k}(x_j-x_i)^k \cdot (y_j-y_i)^l \cdot \overline{x^{n-k}y^{m-l}}_j .\end{aligned}$$

In these terms, we can write

$$\begin{aligned}\frac{1}{A_j}\int_{V_j} \phi_i^R(\vec{x}-\vec{x}_i)\, dA = \phi|_i + \left.\frac{\partial\phi}{\partial x}\right|_i \hat{x}_{ij} + \left.\frac{\partial\phi}{\partial y}\right|_i \hat{y}_{ij} \\ + \left.\frac{\partial^2\phi}{\partial x^2}\right|_i \frac{\widehat{x^2}_{ij}}{2} + \left.\frac{\partial^2\phi}{\partial x\partial y}\right|_i \widehat{xy}_{ij} + \left.\frac{\partial^2\phi}{\partial y^2}\right|_i \frac{\widehat{y^2}_{ij}}{2} + \cdots .\end{aligned} \tag{6}$$

Equation (6) evaluates the mean value of the reconstruction $\phi_i^R(\vec{x}-\vec{x}_i)$ for a control volume j, given the low-order derivatives of the solution at $\vec{x}_i$ and low-order moments of the control volumes. The difference between this prediction and the actual control volume average $\bar{\phi}_j$ is easy to assess. The derivatives at $\vec{x}_i$ are chosen to minimize this error over the stencil $\{V_j\}_i$ in a least-squares sense. Geometric weights w_{ij} are used to specify the relative importance of good prediction for various control volumes in the stencil, with the weights based on distance between control volume reference points. The resulting least-squares problem is

$$\left[\begin{array}{ccccccc} 1 & \bar{x}_i & \bar{y}_i & \overline{x^2}_i & \overline{xy}_i & \overline{y^2}_i & \cdots \\ \hline w_{i1} & w_{i1}\hat{x}_{i1} & w_{i1}\hat{y}_{i1} & w_{i1}\widehat{x^2}_{i1} & w_{i1}\widehat{xy}_{i1} & w_{i1}\widehat{y^2}_{i1} & \cdots \\ w_{i2} & w_{i2}\hat{x}_{i2} & w_{i2}\hat{y}_{i2} & w_{i2}\widehat{x^2}_{i2} & w_{i2}\widehat{xy}_{i2} & w_{i2}\widehat{y^2}_{i2} & \cdots \\ w_{i3} & w_{i3}\hat{x}_{i3} & w_{i3}\hat{y}_{i3} & w_{i3}\widehat{x^2}_{i3} & w_{i3}\widehat{xy}_{i3} & w_{i3}\widehat{y^2}_{i3} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \\ w_{iN} & w_{iN}\hat{x}_{iN} & w_{iN}\hat{y}_{iN} & w_{iN}\widehat{x^2}_{iN} & w_{iN}\widehat{xy}_{iN} & w_{iN}\widehat{y^2}_{iN} & \cdots \end{array}\right] \begin{pmatrix} \phi \\ \frac{\partial\phi}{\partial x} \\ \frac{\partial\phi}{\partial y} \\ \frac{1}{2}\frac{\partial^2\phi}{\partial x^2} \\ \frac{\partial^2\phi}{\partial x\partial y} \\ \frac{1}{2}\frac{\partial^2\phi}{\partial y^2} \\ \vdots \end{pmatrix}_i = \left(\begin{array}{c} \bar{\phi}_i \\ \hline w_{i1}\bar{\phi}_1 \\ w_{i2}\bar{\phi}_2 \\ w_{i3}\bar{\phi}_3 \\ \vdots \\ w_{iN}\bar{\phi}_N \end{array}\right), \tag{7}$$

where N is the number of nearby control volumes in the stencil and the line separates constraints from equations to be "solved" using least squares. The weights used are the inverse of distance between control volume reference locations squared:

$$w_{ij} = \frac{1}{|\vec{x}_j - \vec{x}_i|^2}. \tag{8}$$

This weighting was introduced for use in unstructured mesh computational fluid dynamics by Barth [3] and has the advantage of reducing the influence of data farther from the reconstruction control volume, where neglected higher order terms will have the greatest effect. Gauss elimination is applied for the constraint, replacing the first column of the left-hand side with zeroes. The remaining least-squares problem can be solved for the derivatives of ϕ by using any least-squares technique. We choose to use Householder transforms to reduce the left-hand side matrix to upper triangular, because conditioning problems are more severe with normal equation methods [11]. Finally, the constant term in the expansion, ϕ_i, is evaluated during the back-substitution phase of the least-squares solution, using the constraint equation and the derivatives.

2.2. Boundary Condition Enforcement Via Constrained Least-Squares Reconstruction

Our boundary condition enforcement scheme is parsimonious, in the sense that boundary conditions are enforced only at points where boundary data is actually used—the Gauss integration points along the boundary. We can enforce Dirichlet, Neumann, and mixed Dirichlet–Neumann boundary conditions by constraining the least-squares reconstruction in control volumes adjacent to the boundary. More-complex boundary conditions must be handled, at least in part, by careful definition of boundary fluxes.

2.2.1. *Boundary Constraints for Dirichlet Boundary Conditions*

Suppose that along part of the boundary $\partial\Omega_1$ the solution must satisfy a Dirichlet boundary condition $\phi(\vec{x}) = f_1(\vec{x})$. We enforce this at each Gauss integration point $\vec{x}_g$ on the boundary by requiring

$$\begin{aligned} f_1(\vec{x}_g) &= \phi_i^R(\vec{x}_g) \\ &= \phi|_i + \left.\frac{\partial\phi}{\partial x}\right|_i (x_g - x_i) + \left.\frac{\partial\phi}{\partial y}\right|_i (y_g - y_i) + \left.\frac{\partial^2\phi}{\partial x^2}\right|_i \frac{(x_g - x_i)^2}{2} \\ &\quad + \left.\frac{\partial^2\phi}{\partial x \partial y}\right|_i (x_g - x_i)(y_g - y_i) + \left.\frac{\partial^2\phi}{\partial y^2}\right|_i \frac{(y_g - y_i)^2}{2} + \cdots. \end{aligned} \tag{9}$$

For each control volume adjacent to the boundary, one constraint is added to the least-squares reconstruction problem for each boundary Gauss integration point. As an example, consider the cell-centered control volume shown in Fig. 2. The boundary constraint (Eq. (9)) must be enforced at points a and b and the resulting equalities included in the reconstruction

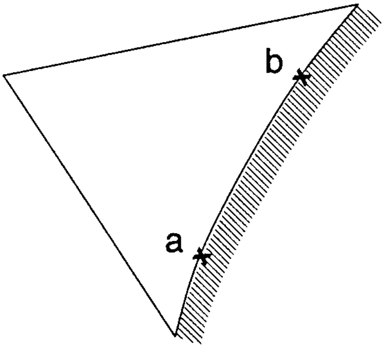


FIG. 2. Cell-centered control volume at the boundary, including Gauss integration points.

problem (Eq. (7)) to give

$$\left[\begin{array}{ccccccc} 1 & \bar{x}_i & \bar{y}_i & \overline{x^2}_i & \overline{xy}_i & \overline{y^2}_i & \cdots \\ 1 & x_a - x_i & y_a - y_i & (x_a - x_i)^2 & (x_a - x_i)(y_a - y_i) & (y_a - y_i)^2 & \cdots \\ 1 & x_b - x_i & y_b - y_i & (x_b - x_i)^2 & (x_b - x_i)(y_b - y_i) & (y_b - y_i)^2 & \cdots \\ \hline w_{i1} & w_{i1}\hat{x}_{i1} & w_{i1}\hat{y}_{i1} & w_{i1}\widehat{x^2}_{i1} & w_{i1}\widehat{xy}_{i1} & w_{i1}\widehat{y^2}_{i1} & \cdots \\ w_{i2} & w_{i2}\hat{x}_{i2} & w_{i2}\hat{y}_{i2} & w_{i2}\widehat{x^2}_{i2} & w_{i2}\widehat{xy}_{i2} & w_{i2}\widehat{y^2}_{i2} & \cdots \\ w_{i3} & w_{i3}\hat{x}_{i3} & w_{i3}\hat{y}_{i3} & w_{i3}\widehat{x^2}_{i3} & w_{i3}\widehat{xy}_{i3} & w_{i3}\widehat{y^2}_{i3} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \\ w_{iN} & w_{iN}\hat{x}_{iN} & w_{iN}\hat{y}_{iN} & w_{iN}\widehat{x^2}_{iN} & w_{iN}\widehat{xy}_{iN} & w_{iN}\widehat{y^2}_{iN} & \cdots \end{array}\right] \begin{pmatrix} \phi \\ \frac{\partial\phi}{\partial x} \\ \frac{\partial\phi}{\partial y} \\ \frac{1}{2}\frac{\partial^2\phi}{\partial x^2} \\ \frac{\partial^2\phi}{\partial x \partial y} \\ \frac{1}{2}\frac{\partial^2\phi}{\partial y^2} \\ \vdots \end{pmatrix}_i$$

$$= \left(\begin{array}{c} \bar{\phi}_i \\ f_1(\vec{x}_a) \\ f_1(\vec{x}_b) \\ \hline w_{i1}\bar{\phi}_1 \\ w_{i2}\bar{\phi}_2 \\ w_{i3}\bar{\phi}_3 \\ \vdots \\ w_{iN}\bar{\phi}_N \end{array}\right).$$

There are three equality constraints on the reconstruction for this control volume—conservation of the mean and two boundary constraints. Gauss elimination is used for all constraint rows, including elimination from the least-squares rows of the problem; with multiple constraints, pivoting is an essential adjunct to the Gauss elimination. The remaining least-squares problem is again solved using Householder transforms to upper triangularize the left-hand side, with back substitution including the constraint rows. This gives a least-squares reconstruction that both conserves the mean and satisfies the boundary conditions.

2.2.2. *Boundary Constraints for Neumann Boundary Conditions*

We apply Neumann boundary conditions in an analogous way. Along part of the boundary $\partial\Omega_2$ the solution must satisfy a Neumann boundary condition $\frac{\partial\phi(\vec{x})}{\partial n} = f_2(\vec{x})$. We enforce this at each Gauss integration point $\vec{x}_g$ on the boundary by requiring

$$\begin{aligned} f_2(\vec{x}_g) &= \frac{\partial\phi_i^R(\vec{x}_g)}{\partial n} = \nabla\phi_i^R(\vec{x}_g)\cdot\hat{n} \\ &= n_x\left(\left.\frac{\partial\phi}{\partial x}\right|_i + \left.\frac{\partial^2\phi}{\partial x^2}\right|_i (x_g - x_i) + \left.\frac{\partial^2\phi}{\partial x \partial y}\right|_i (y_g - y_i) + \cdots\right) \\ &\quad + n_y\left(\left.\frac{\partial\phi}{\partial y}\right|_i + \left.\frac{\partial^2\phi}{\partial x \partial y}\right|_i (x_g - x_i) + \left.\frac{\partial^2\phi}{\partial y^2}\right|_i (y_g - y_i) + \cdots\right), \end{aligned} \tag{10}$$

where $\hat{n}$ is the unit boundary normal. Implementation is completely analogous to the

Dirichlet case in its approach, with the constrained least squares system now given by

$$\left[\begin{array}{ccccccc}
1 & \bar{x}_i & \bar{y}_i & \overline{x^2}_i & \overline{xy}_i & \overline{y^2}_i & \cdots \\
0 & n_x & n_y & 2n_x(x_a - x_i) & \begin{array}{c}(n_x(y_a - y_i)\\ +n_y(x_a - x_i))\end{array} & 2n_y(y_a - y_i) & \cdots \\
0 & n_x & n_y & 2n_x(x_b - x_i) & \begin{array}{c}(n_x(y_b - y_i)\\ +n_y(x_b - x_i))\end{array} & 2n_y(y_b - y_i) & \cdots \\
\hline
w_{i1} & w_{i1}\hat{x}_{i1} & w_{i1}\hat{y}_{i1} & w_{i1}\widehat{x^2}_{i1} & w_{i1}\widehat{xy}_{i1} & w_{i1}\widehat{y^2}_{i1} & \cdots \\
w_{i2} & w_{i2}\hat{x}_{i2} & w_{i2}\hat{y}_{i2} & w_{i2}\widehat{x^2}_{i2} & w_{i2}\widehat{xy}_{i2} & w_{i2}\widehat{y^2}_{i2} & \cdots \\
w_{i3} & w_{i3}\hat{x}_{i3} & w_{i3}\hat{y}_{i3} & w_{i3}\widehat{x^2}_{i3} & w_{i3}\widehat{xy}_{i3} & w_{i3}\widehat{y^2}_{i3} & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \\
w_{iN} & w_{iN}\hat{x}_{iN} & w_{iN}\hat{y}_{iN} & w_{iN}\widehat{x^2}_{iN} & w_{iN}\widehat{xy}_{iN} & w_{iN}\widehat{y^2}_{iN} & \cdots
\end{array}\right]
\begin{pmatrix} \phi \\ \frac{\partial\phi}{\partial x} \\ \frac{\partial\phi}{\partial y} \\ \frac{1}{2}\frac{\partial^2\phi}{\partial x^2} \\ \frac{\partial^2\phi}{\partial x\partial y} \\ \frac{1}{2}\frac{\partial^2\phi}{\partial y^2} \\ \vdots \end{pmatrix}_i
= \left(\begin{array}{c} \bar{\phi}_i \\ f_2(\vec{x}_a) \\ f_2(\vec{x}_b) \\ \hline w_{i1}\bar{\phi}_1 \\ w_{i2}\bar{\phi}_2 \\ w_{i3}\bar{\phi}_3 \\ \vdots \\ w_{iN}\bar{\phi}_N \end{array}\right).$$

2.3. Flux Calculation and Integration

In the present context, we are solving the advection–diffusion equation in two dimensions,

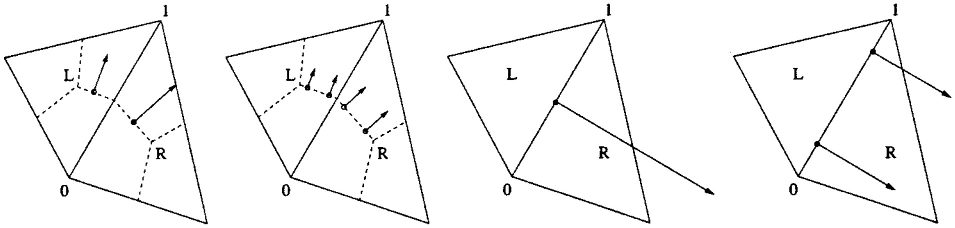
$$\frac{\partial\phi}{\partial t} + \frac{\partial u\phi}{\partial x} + \frac{\partial v\phi}{\partial y} = \alpha\left(\frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2}\right), \tag{11}$$

where the advection velocities u and v are known, possibly as functions of x and y. Using Gauss's theorem for fixed control volumes, we can write for control volume $i(CV_i)$

$$\frac{d\bar{\phi}_i}{dt} + \int_{\partial CV_i} \begin{pmatrix} u\phi - \alpha\frac{\partial\phi}{\partial x} \\ v\phi - \alpha\frac{\partial\phi}{\partial y} \end{pmatrix} \cdot \hat{n}\, dl = 0.$$

This problem contains both advective fluxes (which require the value of the solution) and diffusive fluxes (which require the gradient of the solution). At any point along the control volume boundary, the reconstructed solution and gradient are double valued, because the values from the two incident control volumes may be different. For advective fluxes, we use an upwind flux approximation. For diffusive fluxes, we average the gradients computed in each control volume at the boundary.

Having computed the fluxes, we integrate around each control volume by using Gauss quadrature. In keeping with standard practice, we use one quadrature point per segment with linear reconstruction, and two quadrature points per segment for quadratic and cubic



(a) Second-order accuracy, vertex-centered control volumes

(b) Third- and fourth-order accuracy, vertex-centered control volumes

(c) Second-order accuracy, cell-centered control volumes

(d) Third- and fourth-order accuracy, cell-centered control volumes

FIG. 3. Gauss quadrature for interior control volumes.

reconstruction. This is shown schematically for interior control volumes in Fig. 3, including surface normals scaled by integration weights.

Flux integration on the domain boundary is done similarly. Gauss quadrature points must be placed precisely on the curved boundary and spaced according to arclength rather than straight-line distance to achieve the nominal accuracy for schemes of order higher than two, as the distance between points on the curved boundary and on the straight-line segment between two boundary vertices is $O(\Delta s^2)$. In general, the boundary representation must be at least as accurate as the desired flux integral accuracy. For example, for a fourth-order-accurate solution, a curved boundary must be represented by at least a cubic.

3. ACCURACY ANALYSIS

We have described an approach for calculating and integrating diffusive fluxes with the goal of high-order accuracy. It is well known that a p-order-accurate reconstruction of a solution from control volume averages will produce p-order-accurate flux calculations for fluxes that require the solution but not the gradient. However, the same p-order-accurate reconstruction will produce only $(p-1)$-order-accurate gradients, which should in turn lead to $(p-1)$-order-accurate flux evaluation and flux integrals. To shed light on this issue, we now present analysis of flux integrals for the Laplace equation on uniform triangular meshes. In doing so, we proceed much as we would in analyzing a structured mesh scheme using uniform Cartesian meshes: we analyze the simplest case and perform numerical experiments to determine whether our analysis also predicts correctly results on more general meshes. The numerical experiments are necessary because we have no way to predict how our schemes will behave for nonuniform triangular meshes. In our analysis, we will examine both vertex- and cell-centered control volumes and linear, quadratic, and cubic reconstructions.

3.1. Vertex-Centered Control Volumes

For analysis, a coordinate and labeling system are created for a uniform, vertex-centered, equilateral triangular mesh, as shown in Fig. 4. Each triangle edge has length h. We will be computing analytically the flux integral for control volume A. The neighbors of A can be classified into *genuses* based on their position relative to A without regard to rotation.

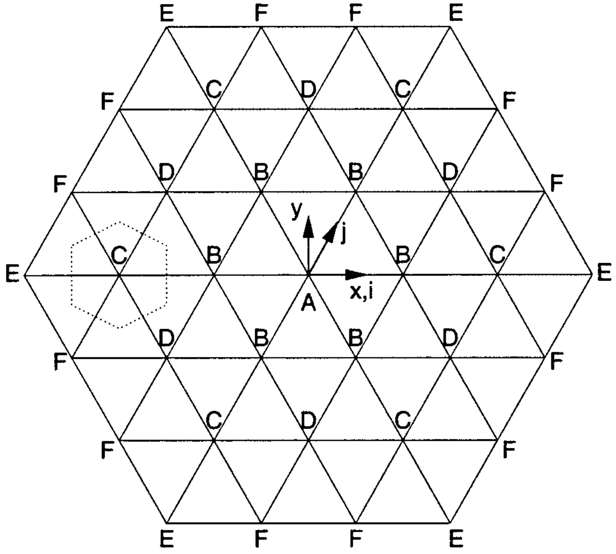


FIG. 4. Coordinates and vertex labels for vertex-centered analysis.

Because of symmetry, all control volumes of the same genus have the same coefficient in the flux integral. Note that the index coordinates (i, j) do not run parallel to the Cartesian coordinates (x, y); the $i = 0$ line runs from the bottom left to top right corners of the hexagonal mesh fragment shown. This choice of coordinates allows an easy mapping from (i, j) to (x, y) coordinates.

The reconstruction scheme requires control volume moments. For the vertex-centered scheme, the control volumes are all identical hexagons (shown as a dotted outline near the left side of Fig. 4), and moments may be found by analytic integration. The low-degree nonzero control volume moments are

$$
\begin{gathered}
\overline{x^2} = \overline{y^2} = \frac{5}{72}h^2, \\
\begin{aligned}
\overline{x^4} = \overline{y^4} &= \frac{7}{720}h^4, & \overline{x^2y^2} &= \frac{7}{2160}h^4, \\
\overline{x^6} &= \frac{3}{1792}h^6, & \overline{x^4y^2} &= \frac{31}{80640}h^6, \\
\overline{x^2y^4} &= \frac{73}{241920}h^6, & \overline{y^6} &= \frac{85}{48384}h^6.
\end{aligned}
\end{gathered}
\tag{12}
$$

With moments in hand, we proceed to reconstruct the solution in each control volume by solving the appropriate subset of the least-squares problem of Eq. (7) and compute fluxes and integrate them, as described in Section 2.3. The analysis technique is identical regardless of order of accuracy, with differences only in the size of the reconstruction stencil. In the next subsection, we give detailed analysis for the case of linear reconstruction to demonstrate the technique. For the remaining cases—quadratic reconstruction using only first neighbors; quadratic reconstruction with a broader, more realistic stencil; and cubic reconstruction—we will summarize and discuss results.

3.1.1. *Linear Reconstruction*

For linear reconstruction, we prefer at least three neighbors in the reconstruction stencil; for an equilateral mesh, first neighbors are more than sufficient. All geometric weights are equal to $1/h^2$. The solution of the least-squares problem in control volume (i, j) is

$$\begin{pmatrix} \phi \\ \phi_x \\ \phi_y \end{pmatrix} = \begin{pmatrix} \bar{\phi}_{i,j} \\ \frac{2(\bar{\phi}_{i+1,j} - \bar{\phi}_{i-1,j}) + \bar{\phi}_{i,j+1} + \bar{\phi}_{i+1,j-1} - \bar{\phi}_{i,j-1} - \bar{\phi}_{i-1,j+1}}{6h} \\ \frac{\sqrt{3}(\bar{\phi}_{i,j+1} + \bar{\phi}_{i-1,j+1} - \bar{\phi}_{i+1,j-1} - \bar{\phi}_{i,j-1})}{6h} \end{pmatrix}. \tag{13}$$

We compute the flux integral as described in Section 2.3; in this case, the fluxes are especially easy to calculate because there are no second derivatives in the reconstruction. One-point Gauss integration is sufficient in this case. After simplification, we obtain an expression for the computed Laplacian in control volume (0, 0) $\widetilde{\nabla^2}\phi_{0,0}$ in terms of control volume averages,

$$\widetilde{\nabla^2}\phi_{0,0} = \frac{-6\bar{\phi}_A - \sum \bar{\phi}_B + \sum \bar{\phi}_C + \sum \bar{\phi}_D}{9h^2}, \tag{14}$$

where, for example, $\sum \bar{\phi}_B$ is the sum of the average solution for all the control volumes labeled B. We can expand the control volume averages in terms of derivatives of the underlying smooth solution ϕ at the origin by using Eq. (6) and find the difference between this computed Laplacian $\widetilde{\nabla^2}\phi_{0,0}$ and the control volume average of the actual Laplacian $\overline{\nabla^2\phi}_{0,0}$[1]:

$$\widetilde{\nabla^2}\phi_{0,0} - \overline{\nabla^2\phi}_{0,0} = \frac{h^2}{4}\left(\frac{\partial^4\phi}{\partial x^4} + 2\frac{\partial^4\phi}{\partial x^2 \partial y^2} + \frac{\partial^4\phi}{\partial y^4}\right) + \mathcal{O}(h^4). \tag{15}$$

Clearly, this scheme for calculating the Laplacian is second-order accurate for equilateral meshes and vertex-centered control volumes, despite the fact that the first derivatives in the reconstruction can easily be shown to be only first-order accurate.

3.1.2. *Results for Other Cases*

For the linear reconstruction case, we required only first neighbors for the reconstruction and obtained, perhaps serendipitously, second-order accuracy. The quadratic reconstruction cases (see Table I) also give second-order-accurate flux integrals, which is consistent with the second-order accuracy of the computed fluxes. The compact (first neighbors only) reconstruction stencil is an unrealistic choice in practice; least-squares reconstruction schemes are typically implemented with more looseness in the reconstruction. Perhaps because of the greater degree of approximation in the reconstruction, the normal (first and second neighbors) version of the reconstruction stencil gives a less accurate calculation of the Laplacian than does the compact stencil.

With cubic reconstruction, we must use both first and second neighbors in the reconstruction. The resulting flux integral is fourth-order accurate, which is one order better than the computed fluxes.

In summary, reconstruction schemes that give even-order accuracy for advective-type fluxes (linear and cubic) provide the same order of accuracy with our Laplacian scheme.

[1] Note that it is critically important in this and all following cases to include the high-order terms in the average of the Laplacian, by applying Eq. 6, rather than using simply $\nabla^2\phi(0, 0)$.

TABLE I

Stencils and Truncation Error for Laplacian on a Vertex-Centered Mesh[a]

	Recon. type			
	Linear	Compact quadratic	Normal quadratic	Cubic
Recon. stencil (weight)	B$(1/h^2)$	B$(1/h^2)$	B$(1/h^2)$ C$(1/4h^2)$ D$(1/3h^2)$	B$(1/h^2)$ C$(1/4h^2)$ D$(1/3h^2)$
Leading T. E.	$\frac{h^2}{4}\nabla^2\nabla^2\phi$	$\frac{h^2}{16}\nabla^2\nabla^2\phi$	$\frac{53h^2}{304}\nabla^2\nabla^2\phi$	$-\frac{3137h^4}{76032}\nabla^2\nabla^2\nabla^2\phi$ $-\frac{439h^4}{380160}\mathcal{A}(\phi)$
Multiplier	$\frac{1}{9h^2}$	$\frac{1}{18h^2}$	$\frac{1}{12312h^2}$	$\frac{1}{35640h^2}$
Genus A (1)	−6	−66	−22284	−78894
Genus B (6)	−1	10	1799	4976
Genus C (6)	1	−1	674	4349
Genus D (6)	1	2	1212	6006
Genus E (6)	—	—	−63	−678
Genus F (12)	—	—	46	−752

[a] $\mathcal{A}(\phi) = \frac{\partial^6\phi}{\partial x^6} - 15\frac{\partial^6\phi}{\partial x^4\partial y^2} + 15\frac{\partial^6\phi}{\partial x^2\partial y^4} - \frac{\partial^6\phi}{\partial y^6}$.

Reconstructions that give odd-order accuracy for advective-type fluxes (quadratic) are one order less accurate in computing the Laplacian with our scheme. We postulate that similar behavior will continue for higher orders of accuracy as well.

3.2. Cell-Centered Control Volumes

For cell-centered meshes, the analysis proceeds in much the same way as for the vertex-centered case. A coordinate and labeling system are created for a uniform, equilateral triangular mesh, as shown in Fig. 5. Here, there is the additional complication that not all control volumes are identical; all moments for monomials of the form $x^{2n}y^{2m+1}$ have

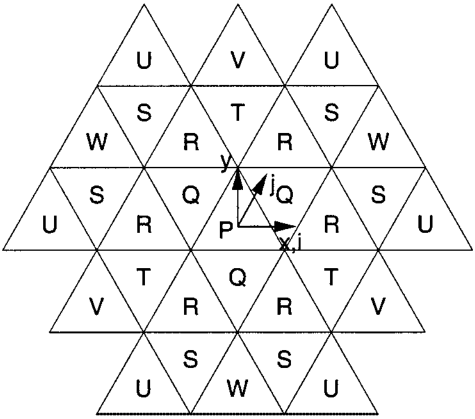


FIG. 5. Coordinates and cell labels for cell-centered analysis.

values that are equal in magnitude but opposite in sign for up-pointing versus down-pointing triangles. We label each parallelogram with sides parallel to the i and j axes with a single i, j pair and distinguish these triangles by their orientation (Δ for up pointing, ∇ for down pointing). Each triangle edge has length h, and again each control volume is marked in the figure with its genus. Once again, the index coordinates (i, j) do not run parallel to the Cartesian coordinates (x, y); the $i = 0$ line runs from the bottom left to top right corners of the mesh fragment shown.

The control volume moments are again found by analytic integration. For up-pointing triangles, the low-degree nonzero control volume moments are

$$\begin{array}{ccc} & \overline{x^2} = \overline{y^2} = \dfrac{1}{24}h^2, & \\ \overline{x^2y} = -\dfrac{\sqrt{3}}{360}h^3, & & \overline{y^3} = \dfrac{\sqrt{3}}{360}h^3, \\ \overline{x^4} = \overline{y^4} = \dfrac{1}{240}h^4, & & \overline{x^2y^2} = \dfrac{1}{720}h^4, \\ \overline{x^4y} = -\dfrac{\sqrt{3}}{2520}h^5, & \overline{x^2y^3} = -\dfrac{\sqrt{3}}{7560}h^5, & \overline{y^5} = \dfrac{\sqrt{3}}{1512}h^5, \\ \overline{x^6} = \dfrac{1}{1792}h^6, & & \overline{x^4y^2} = \dfrac{13}{80640}h^6, \\ \overline{x^2y^4} = \dfrac{19}{241920}h^6, & & \overline{y^6} = \dfrac{31}{48384}h^6. \end{array} \tag{16}$$

For down-pointing cells, the following moments have the opposite sign to those shown: $\overline{x^2y}$, $\overline{y^3}$, $\overline{x^4y}$, $\overline{x^2y^3}$, and $\overline{y^5}$. Given these moments, we proceed as before with reconstruction, flux computation, and flux integration. In the next subsection, we give detailed analysis for the case of liner reconstruction to demonstrate the technique. For the remaining cases—quadratic reconstruction; cubic reconstruction with a restricted stencil; and cubic reconstruction with a broader, more realistic stencil—we will summarize and discuss results.

3.2.1. *Linear Reconstruction*

For linear reconstruction, first neighbors are again sufficient for the reconstruction. All geometric weights are equal to $3/h^2$. The solution of the least-squares problem in control volume (i, j, Δ) is

$$\begin{pmatrix} \phi \\ \phi_x \\ \phi_y \end{pmatrix} = \begin{pmatrix} \bar{\phi}_{i,j,\Delta} \\ \frac{\bar{\phi}_{i,j,\nabla} - \bar{\phi}_{i-1,j,\nabla}}{h} \\ \frac{\sqrt{3}(\bar{\phi}_{i,j,\nabla} + \bar{\phi}_{i-1,j,\nabla} - 2\bar{\phi}_{i,j-1,\nabla})}{3h} \end{pmatrix}. \tag{17}$$

The result for down-pointing triangles is analogous. We compute the flux integral as described in Section 2.3. Again, one-point Gauss integration is sufficient in this case. After simplification, we obtain an expression for the computed Laplacian in control volume $(0, 0, \Delta)$ $\widetilde{\nabla^2\phi_{0,0}}$ in terms of control volume averages:

$$\widetilde{\nabla^2\phi_{0,0}} = \frac{-6\bar{\phi}_P + \sum \bar{\phi}_R}{\frac{3}{2}h^2}. \tag{18}$$

TABLE II

Stencils and Truncation Error for Laplacian on a Cell-Centered Mesh[a]

	Recon. type			
	Linear	Quadratic	Compact cubic	Normal cubic
Recon. stencil (weight)	$Q(3/h^2)$	$Q(3/h^2)$ $R(1/h^2)$	$Q(3/h^2)$ $R(1/h^2)$	$Q(3/h^2)$ $R(1/h^2)$ $S(3/4h^2)$ $T(3/7h^2)$
Leading T. E.	$\frac{h^2}{16}\nabla^2\nabla^2\phi$	$\frac{h\sqrt{3}}{45}\mathcal{B}(\phi)$ $+\frac{7h^2}{144}\nabla^2\nabla^2\phi$	$-\frac{h^3\sqrt{3}}{240}\nabla^2\mathcal{B}(\phi)$ $-\frac{h^2}{288}\nabla^2\nabla^2\nabla^2\phi$ $+\frac{h^4}{5760}\mathcal{A}(\phi)$	$-\frac{1.629h^3}{1000}\nabla^2\mathcal{B}(\phi)$ $-\frac{6.603h^4}{1000}\nabla^2\nabla^2\nabla^2\phi$ $-\frac{0.403h^4}{1000}\mathcal{A}(\phi)$
Equivalent Laplacian stencil				
Multiplier	$\frac{2}{3h^2}$	$\frac{4}{9h^2}$	$\frac{1}{6h^2}$	$\frac{1}{h^2}$
Genus P (1)	−6	−15	−42	−6.1150
Genus Q (3)	0	3	4	0.6985
Genus R (6)	1	1	7	0.6371
Genus S (6)	—	—	−1	0.05565
Genus T (3)	—	—	−2	0.1188
Genus U (6)	—	—	—	−0.04919
Genus V (3)	—	—	—	−0.01368
Genus W (3)	—	—	—	−0.05244

[a] $\mathcal{A}(\phi) \equiv \frac{\partial^6\phi}{\partial x^6} - 15\frac{\partial^6\phi}{\partial x^4\partial y^2} + 15\frac{\partial^6\phi}{\partial x^2\partial y^4} - \frac{\partial^6\phi}{\partial y^6}$ and $\mathcal{B}(\phi) \equiv 3\frac{\partial^3\phi}{\partial x^2\partial y} - \frac{\partial^3\phi}{\partial y^3}$.

Note that only up-pointing triangles are used in this computation, admitting the possibility of the analog to structured mesh odd-even solution decoupling for uniform meshes.[2]

As with the vertex-centered analysis, we replace control volume averages with their expansions (from Eq. (6)) and simplify to find the error in the computed Laplacian:

$$\widetilde{\nabla^2}\phi_{0,0} - \overline{\nabla^2\phi}_{0,0} = \frac{h^2}{16}\left(\frac{\partial^4\phi}{\partial x^4} + 2\frac{\partial^4\phi}{\partial x^2\partial y^2} + \frac{\partial^4\phi}{\partial y^4}\right) + \mathcal{O}(h^4). \tag{19}$$

Clearly, this scheme for calculating the Laplacian is second-order accurate for equilateral meshes and cell-centered control volumes, despite the fact that the first derivatives in the reconstruction can easily be shown to be only first-order accurate.

3.2.2. *Results for Other Cases*

For the quadratic reconstruction case, we require an additional level of neighbors for reconstruction. With inverse-distance-squared weighting, the computed Laplacian is especially simple, as seen in Table II. As we expect given the results for vertex-centered control volumes, the computed Laplacian is not third-order accurate. More surprising is the fact that the Laplacian is in fact only first-order accurate for this case! Note that, even without

[2] In practice, we have observed no difficulties with decoupling for advection–diffusion problems.

the troublesome first-order term, the truncation error for this approach is only slightly lower than for the linear reconstruction scheme.

The first-order error term is a combination of third derivatives that cannot be computed independently for any single genus of control volumes. Consider, for example, the control volumes of genus Q. Symmetry considerations require that each of these control volumes have the same weight in the Laplacian stencil. Unfortunately, the location of these control volumes is such that the x^2y and y^3 terms in the Taylor expansions of their control volume averages do not cancel. Making matters worse, the x^2y and y^3 moments of all the cell-centered control volumes ae nonzero. As numerical experiments in Section 4 will show, this is a case where actual performance exceeds analytic expectations, in part because realistic meshes do not have the fatal symmetry which causes this problem analytically.

With both first and second neighbors in the reconstruction stencil, we have precisely nine neighbors, which is the absolute minimum for cubic recontruction. For this case ("cubic compact" in Table II), the computed Laplacian is third-order accurate, which is all that we have any right to expect, given the accuracy of the first derivatives from the reconstruction. However, the leading-order term in the truncation error contains the Laplacian of the same term that was troublesome for the quadratic reconstruction case. Again, the numerical experiments will show that this term is not a factor in practice.

The stencil we would normally choose for cubic reconstruction in control volume P would include its first, second, and third cell neighbors (genus Q, R, S, and T), for a total of 18 neighbors; geometric weights for the reconstruction are shown in Table II ("normal cubic"). This case echoes the cubic compact case; the third-order error term has a slightly smaller constant here, although the fourth-order terms are somewhat larger.[3]

In summary, our analysis shown that the perfect, uniform mesh allows an ambiguity in calculating certain derivatives using clusters of cells of the same genus; this ambiguity appears in the analytic results as truncation error terms that do not meet our expectations for order of accuracy. Numerical experiments, presented in the next section, will show that for normal triangular meshes, the computed results exceed analytic expectations. Beyond this somewhat troubling quirk, we have shown that reconstruction schemes that would give even-order accuracy for advective-type fluxes (linear and cubic) provide the same order of accuracy with our Laplacian scheme. Reconstruction that gives odd-order accuracy for advective-type fluxes (quadratic) is one order less accurate in computing the Laplacian with our scheme. Again, we postulate that similar behavior will continue for higher orders of accuracy as well.

3.3. Discussion

For both vertex-centered and cell-centered control volumes, common practice (e.g., Galerkin finite-element or Green–Gauss finite-volume discretizations) for uniform triangular meshes results in simpler stencils than those given by our approach. In particular, for vertex-centered control volumes, the Galerkin finite-element discretization of the Laplacian can be written in our notation as

$$\widetilde{\nabla^2\phi}_{0,0} = \frac{2}{3}\frac{-6\bar{\phi}_A + \sum\bar{\phi}_B}{h^2}. \tag{20}$$

[3] For this case, the fractional representations of the stencil and errors, though computed analytically, are far too cumbersome to be of even illustrative use, so decimal values are used instead.

For this stencil, the difference between this computed Laplacian $\widetilde{\nabla^2\phi}_{0,0}$ and the control volume average of the actual Laplacian $\overline{\nabla^2\phi}_{0,0}$ is

$$\widetilde{\nabla^2\phi}_{0,0} - \overline{\nabla^2\phi}_{0,0} = \frac{h^2}{16}\left(\frac{\partial^4\phi}{\partial x^4} + 2\frac{\partial^4\phi}{\partial x^2\,\partial y^2} + \frac{\partial^4\phi}{\partial y^4}\right) + \mathcal{O}(h^4). \tag{21}$$

This truncation error is smaller by a factor of four than our result. The stencil is also positive (and remains so for nonuniform meshes) and more compact than ours. However, our approach requires less computation for problems with both advection and diffusion, because the gradient is available for free following reconstruction. On balance, either scheme is likely to work reasonably well in practice.

For cell-centered control volumes, gradients can be calculated by using Green–Gauss integration around a diamond connecting the end points of an edge and the centroids of the cells that share the edge. The solution at the end points of the edge can be estimated by averaging data in incident control volumes. For a uniform mesh, the resulting Laplacian stencil requiring only the first neighbor control volumes and can be written as

$$\widetilde{\nabla^2\phi}_{0,0} = 4\frac{-3\bar{\phi}_P + \sum\bar{\phi}_Q}{h^2}. \tag{22}$$

The error in the computed Laplacian for this case is

$$\widetilde{\nabla^2\phi}_{0,0} - \overline{\nabla^2\phi}_{0,0} = \frac{h\sqrt{3}}{15}\left(3\frac{\partial^3\phi}{\partial x^2\partial y} - \frac{\partial^3\phi}{\partial y^3}\right) + \frac{h^2}{48}\left(\frac{\partial^4\phi}{\partial x^4} + 2\frac{\partial^4\phi}{\partial x^2\partial y^2} + \frac{\partial^4\phi}{\partial y^4}\right) + \mathcal{O}(h^4). \tag{23}$$

This stencil suffers from the now-familiar mixed third-derivative problem in its simplest form. In addition, the three neighboring control volumes used in this stencil do not give enough degrees of freedom to compute even a zero-order-accurate Laplacian for general meshes. Consequently, our approach is superior, even though it allows the possibility of solution decoupling in some circumstances.

4. NUMERICAL EXPERIMENTS

In this section, we present several numerical experiments with the goals of confirming the analytical results of the previous section, of exploring whether the odd-derivative error terms present for equilateral meshes affect computational results in practice, and of demonstrating the importance of correct treatment of curved boundaries. All our results use the approach described in Section 2.1 to determine reconstruction stencils and, therefore, correspond more closely to the "normal" than the "compact" cases analyzed in Section 3.

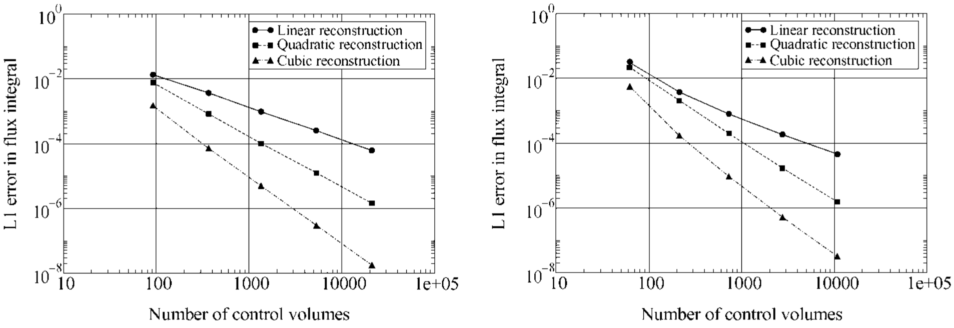
The meshes for these test cases were generated using a guaranteed-quality meshing scheme [6]; all triangles had angles between 30 and 120 degrees. While these are excellent meshes, they are far from being uniform and equilateral, providing a realistic test of our schemes for practical meshes. For the advection–diffusion cases, steady solutions for the unsteady problem were obtained by using Runge–Kutta time advance with local time stepping. Clearly this is not the most efficient approach because of the severe time-step limitations on explicit methods for the advection–diffusion equation. However, for our purposes—investigating the accuracy of the spatial discretization scheme—efficiency was not an issue, and convergence was not difficult to achieve. We computed error in each control volume by comparing the computed solution with the average of the exact solution; the average was calculated by using a sixth-order-accurate Gauss quadrature.

4.1. Flux Integral Tests for the Laplacian

We begin with an experiment designed to confirm that the accuracy of flux integrals computed using our scheme for the Laplacian is consistent with the analytic results of Section 3. For both vertex-centered and cell-centered control volumes, we computed flux integrals on a sequence of meshes in a unit square. The initial solution data was

$$\phi(x, y) = \sin(\pi x)\sin(\pi y),$$

implying homogeneous Dirichlet boundary conditions on the solution. In each case, the flux integrals were computed for all control volumes and compared with the average of the Laplacian of ϕ (where $\nabla^2\phi = -2\pi^2\phi$); the results are presented in Fig. 6. For both cell- and vertex-centered schemes, the accuracy of the flux integrals approximately matches the accuracy of the reconstruction, even in the L_∞ norm. This result exceeds the expectations of the analysis of Section 3. This result is perhaps not too surprising for the linear and cubic reconstruction cell-centered schemes, whose analytic accuracy was reduced by a symmetry



(a) Flux integration accuracy for cell-centered schemes

(b) Flux integration accuracy for vertex-centered schemes

Mesh	Cells	Verts
A	92	62
B	366	213
C	134	727
D	5308	2756
E	21091	10744

Reconstruction Order	Actual Order (Cell)			Actual Order (Vertex)		
	L_1	L_2	L_∞	L_1	L_2	L_∞
2	2.05	2.06	1.78	2.12	2.07	1.56
3	3.16	3.25	3.08	3.61	3.52	3.14
4	4.19	4.18	3.85	4.22	4.12	3.55

FIG. 6. Results for Laplacian flux integral.

in the uniform mesh used for the analysis; these cases appear to benefit from less than completely regular meshes.

4.2. Advection–Diffusion in a Rectangular Channel

This case solves the two-dimensional advection–diffusion equation, Eq. (11), in a rectangular channel of length $L = 3$ in the x-direction and unit height in the y-direction, with constant velocity of $(u, v) = (u_0, 0)$ and a diffusion coefficient of $\alpha = 0.01$.[4] The boundary conditions for this case are

$$\begin{aligned} T(x,0) = T(x,1) &= 0, \\ T(0,y) &= \sin(\pi y), \\ \frac{\partial T(L,y)}{\partial x} &= 0. \end{aligned}$$

The exact solution for this problem is

$$\phi(x,y) = \sin(\pi y)\left[\frac{r_2 \exp(r_1 x + r_2 L) - r_1 \exp(r_1 L + r_2 x)}{r_2 \exp(r_2 L) - r_1 \exp(r_1 L)}\right],$$

where

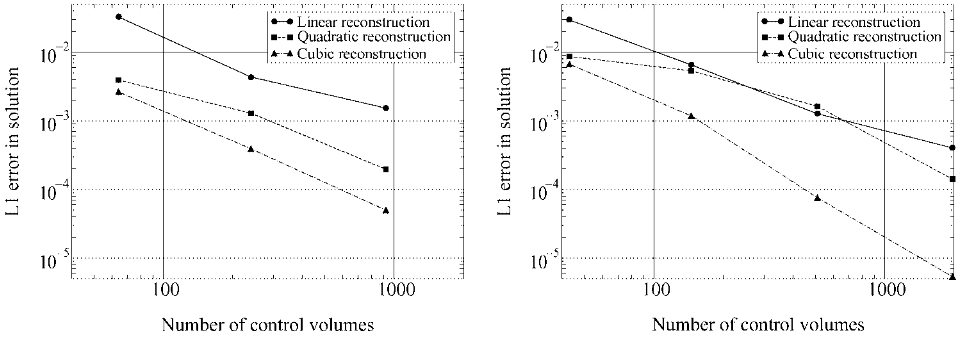
$$r_{1,2} = \frac{u_0}{2\alpha} \pm \sqrt{\frac{u_0^2}{4\alpha^2} + \pi^2}.$$

Solutions for this problem were computed on three meshes using linear, quadratic, and cubic reconstructions. Figure 7 summarizes the results for this test case; the tabulated values are norms based on the two finest meshes for each case. As the plots indicate, the convergence of the computed solutions to the exact solution with mesh refinement is not entirely smooth. Nevertheless, in the L_1 norm, five out of six combinations reach the order of accuracy of their reconstruction, an indication of the *de facto* behavior of the discretization scheme. This result is a pleasant surprise for both quadratic schemes, where analysis of the diffusive terms suggested that second-order accuracy was the best we could expect. While it may be that further mesh refinement would eventually reveal the presence of a second-order truncation error term with a very small coefficient for these two cases, on the practical level we are content, as the standard of accuracy achieved in these tests is tighter than typically needed in applications. The one exception to the otherwise excellent behavior of these schemes is the cell-centered cubic reconstruction scheme, which fails to achieve fourth-order convergence for this test case; a handful of cells with comparatively large error persist at the outflow boundary.

4.3. Advection–Diffusion in an Annulus

Now we turn our attention to a case with curved boundaries. In this case, the domain is an annular segment, as shown in Fig. 8. At the inflow, the boundary condition is given by $T(r, \theta = 0) = \sin\left(\frac{\pi \ln r}{\ln 2}\right)$, and the velocity is $(v_r, v_\theta) = (0, u_0/r)$; other boundary conditions

[4] This diffusion coefficient is the equivalent of $1/\mathrm{Re}$ for a viscous flow.



(a) Convergence for cell-centered schemes

(b) Convergence for vertex-centered schemes

Mesh	Cells	Verts
A	64	43
B	240	145
C	922	511
D	—	1965

Reconstruction Order	Actual Order (Cell)			Actual Order (Vertex)		
	L_1	L_2	L_∞	L_1	L_2	L_∞
2	1.65	1.67	1.92	1.70	1.65	1.66
3	2.80	2.77	1.91	3.62	3.57	3.08
4	3.07	2.79	1.45	3.93	3.09	2.13

FIG. 7. Results for rectangular channel test case.

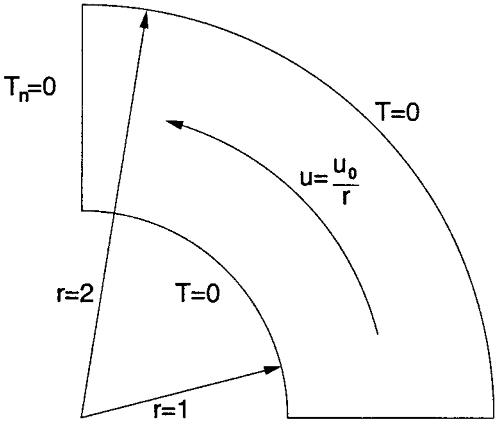


FIG. 8. Geometry and boundary conditions for annular test case. The inflow boundary condition is given in the text.

are shown in Fig. 8. With this data, the advection–diffusion equation in cylindrical coordinates is separable, and the solution is given by

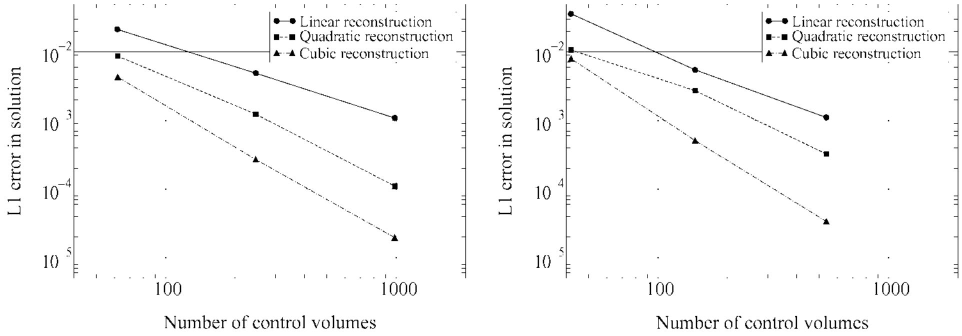
$$T(r, \theta) = \sin\left(\frac{\pi \ln r}{\ln 2}\right) \frac{s_1 \exp\left(s_1 \frac{\pi}{2} + s_2 \theta\right) - s_2 \exp\left(s_2 \frac{\pi}{2} + s_1 \theta\right)}{s_1 \exp\left(s_1 \frac{\pi}{2}\right) - s_2 \exp\left(s_2 \frac{\pi}{2}\right)},$$

where

$$s_{1,2} = \frac{u_0 \pm \sqrt{u_0^2 + \left(\frac{2\pi\alpha}{\ln 2}\right)^2}}{2\alpha}.$$

Note that although the solution is given in terms of r and θ, we solved the problem numerically by using the Cartesian advection–diffusion equation. For this problem, the boundary shape is treated exactly, but the use of cubic splines to describe the boundary would not affect the order of accuracy of the solution.

As discussed in Sections 2.2 and 2.3, care must be taken with boundary condition enforcement in the presence of curved boundaries. Figure 9 shows the results obtained for this case when boundaries are treated correctly. In all cases, despite some noise in the error norm data, the nominal order of accuracy of the scheme was achieved, except that the maximum error for the vertex-centered, cubic reconstruction scheme is somewhat poorly behaved. An examination of the error on the finest mesh for this scheme reveals that the only errors of consequence occur at the outflow boundary, although there is no obvious



(a) Convergence for cell-centered schemes

(b) Convergence for vertex-centered schemes

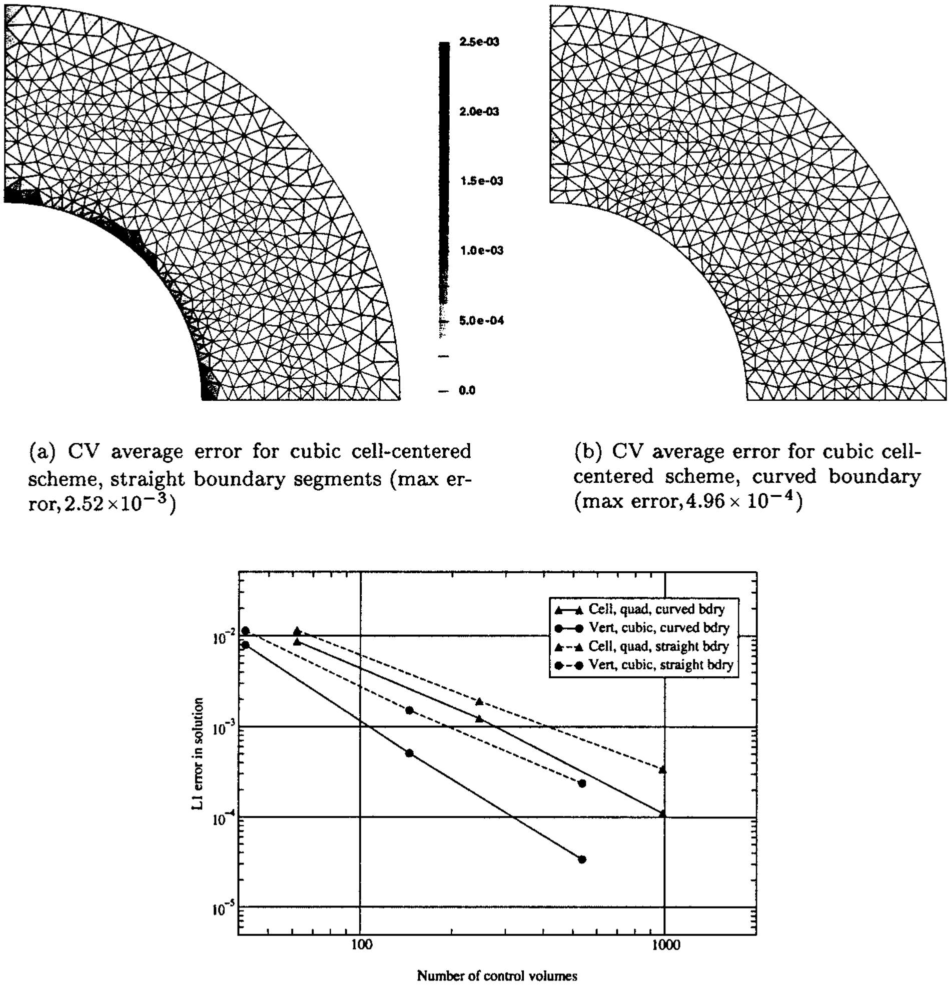
Mesh	Cells	Verts
A	62	42
B	246	145
C	985	537

Reconstruction Order	Actual Order (Cell)			Actual Order (Vertex)		
	L_1	L_2	L_∞	L_1	L_2	L_∞
2	2.17	2.18	2.39	1.98	2.08	2.10
3	3.48	3.31	3.42	3.25	3.27	3.06
4	3.78	3.74	3.46	4.14	4.17	2.63

FIG. 9. Results for annular test case.

mesh anomaly that might cause discretization problems. Note that the maximum error is much better behaved asymptotically than for the straight channel case, essentially ruling out a blanket problem with Neumann boundary conditions in our scheme. Consistent with this inference, a postcheck of solution values at the boundary confirms that the solution precisely satisfies the boundary conditions at the Gauss quadrature points and converges to zero at other points on the boundary at the same rate as the truncation error decreases.

We now examine the difficulties which arise when the curved boundaries of the computational domain are approximated by straight-line segments between vertices. Geometrically, we know that the difference in location between the curved boundary and the straight-line segments will be $O(\Delta^2)$, where Δ is a typical boundary edge length. This implies that boundary conditions will be enforced at points that differ by $O(\Delta^2)$ from the correct locations and that second-order errors in the solution should arise. This is precisely the effect observed in numerical experiments, as illustrated in Fig. 10, which clearly illustrates the



(a) CV average error for cubic cell-centered scheme, straight boundary segments (max error, 2.52×10^{-3})

(b) CV average error for cubic cell-centered scheme, curved boundary (max error, 4.96×10^{-4})

(c) Representative convergence comparison for straight versus curved boundaries

FIG. 10. Errors induced by treating curved boundaries as straight. (Error maps shaded with same scale.)

general result that none of the high-order schemes achieve their nominal order of accuracy with straight boundaries, even in the L_1 norm. Also, none of the high-order schemes achieve more than second-order accuracy in the L_2 norm, and for some schemes the L_∞ norm of the error behaves as first-order accurate or worse.

Simply reading a set of boundary points from a file is clearly insufficient to achieve high-order accuracy. The implication—which applies for high-order-accurate schemes generally—is that boundary information *must* be available to the solver for preprocessing control volume moments and boundary integration data. Whether that data is read or recovered from boundary points is immaterial, provided that the shape of the boundary used in preprocessing is at least as accurate as the desired discretization accuracy.

5. CONCLUSIONS

We have described a new scheme for high-order-accurate solution of the advection–diffusion equation using reconstruction. Our discussion focused on least-squares reconstruction; other reconstruction techniques, including ENO/WENO schemes, should work as well, although our analysis would not be directly applicable for those cases. We have also presented accuracy analysis of our scheme applied to the Laplacian operator on a uniform, equilateral triangular mesh, with the following conclusions.

1. Analytically, the full reconstruction order of accuracy was demonstrated for vertex- and cell-centered schemes based on linear reconstruction (second order) and for the vertex-centered cubic reconstruction scheme (fourth order).
2. For both vertex- and cell-centered schemes, the quadratic reconstruction variants show, under analysis, a second-order term in their truncation error, whereas the reconstructed solution is third-order accurate.
3. Finally, the quadratic and cubic reconstruction cell-centered schemes have leading-order-error terms (first and third order, respectively) that are associated with degeneracies in the reconstruction. This degeneracy occurs because some pairs of terms in a Taylor series expansion of the solution cannot be distinguished from one another by symmetric stencils on a uniform, triangular mesh.

We also presented a new approach for boundary condition enforcement. Boundary conditions are enforced at boundary flux integration points by constraining the reconstruction in boundary control volumes to precisely match those conditions. This approach was used for all examples in the paper and proven to work well.

Finally, we presented the results of several numerical experiments. These experiments show that our schemes appear to perform better in practice than the analysis suggests. In the case of the high-order cell-centered schemes, the degenerate terms in the Taylor series expansion are no longer degenerate with an irregular mesh, so that result is not unexpected. The quadratic schemes outperform analytic predictions to achieve third-order accuracy, at least to the error levels considered here. Finally, the numerical experiments illustrated the proper behavior of the boundary constraints we impose during reconstruction.

Our next step in this research is to optimize the convergence behavior of our schemes as applied to the Navier–Stokes equations to enable efficiency comparisons of schemes of different orders of accuracy at the same error level.

ACKNOWLEDGEMENTS

This work has been supported by the Canadian Natural Science and Engineering Research Council under Grant OPG-0194467 and by the University of British Columbia Research Development Fund.

REFERENCES

1. R. Abgrall, On essentially non-oscillatory schemes on unstructured meshes: Analysis and implementation, *J. Comput. Phys.* **114**, 45 (1994).
2. R. K. Agarwal and D. W. Halt, A compact high-order unstructured grids method for the solution of Euler equations, *Int. J. Numer. Methods Fluids* **31**(1), 121 (1999).
3. T. J. Barth, Aspects of unstructured grids and finite-volume solvers for the Euler and Navier–Stokes equations, in *Unstructured Grid Methods for Advection-Dominated Flows* (AGARD, 1992), Nevilly sur Seine, France pp. 6-1–6-61.
4. T. J. Barth and P. O. Frederickson, *Higher Order Solution of the Euler Equations on Unstructured Grids Using Quadratic Reconstruction*, AIAA paper 90-0013 (1990).
5. F. Bassi and S. Rebay, A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier–Stokes equations, *J. Comput. Phys.* **131**, 267 (1997).
6. C. Boivin and C. F. Ollivier-Gooch, Guaranteed-quality-triangular mesh generation for domains with curved boundaries, *Int. J. Numer. Methods Eng.*, in press.
7. M. Delanaye and J. A. Essers, Quadratic-reconstruction finite volume scheme for compressible flows on unstructured adaptive grids, *AIAA J.* **35**(4), 631 (1997).
8. L. J. Durlofsky, B. Enquist, and S. Osher, Triangle based adaptive stencils for the solution of hyperbolic conservation laws. *J. Comput. Phys.* **98**, 64 (1992).
9. O. Friedrich, Weighted essentially non-oscillatory schemes for the interpolation of mean values on unstructured grids, *J. Comput. Phys.* **144**, 194 (1998).
10. P. Geuzaine, M. Delanaye, and J.-A. Essers, Computation of high Reynolds number flows with an implicit quadratic reconstruction scheme on unstructured grids, in *Proceedings of the 13th AIAA Computational Fluid Dynamics Conference* (AIAA Press, Washington, DC, 1997), p. 610.
11. G. H. Golub and C. F. van Loan, *Matrix Computations* (Johns Hopkins Press, Baltimore, 1983).
12. C. Kiris and D. Kwak, Numerical solution of incompressible Navier–Stokes equations using a fractional-step approach, *Comput. Fluids* **30**(7–8), 829–851 (2001).
13. D. J. Mavriplis, A. Jameson, and L. Martinelli, *Multigrid Solution of the Navier–Stokes Equations on Triangular Meshes*, ICASE Report No. 89–11 (NASA Langley Research Center, 1989).
14. C. F. Ollivier-Gooch, *High-Order ENO Schemes for Unstructured Meshes Based on Least-Squares Reconstruction*, AIAA paper 97-0540 (1997).
15. C. F. Ollivier-Gooch, Quasi-ENO schemes for unstructured meshes based on unlimited data-dependent least-squares reconstruction, *J. Comput. Phys.* **133**, 6 (1997).
16. M. Van Altena, *High-Order Finite-Volume Discretisations for Solving a Modified Advection–Diffusion Problem on Unstructured Triangular Meshes*, Master's thesis (Dept. Mechanical Engineering, Univ. British Columbia, 1999).
17. M. Van Altena and C. F. Ollivier-Gooch, Finite-volume methods for solving the Laplace equation on unstructured triangular meshes, in *Proceedings of the Seventh Annual Conference of the Computational Fluid Dynamics, Society of Canada, May 1999*, p. 10–3.
18. D. Zingg, S. DeRango, M. Nemec, and T. Pulliam, Comparison of several spatial discretizations for the Navier–Stokes equations, *J. Comput. Phys.* **160**, 683 (2000).