AIAA JOURNAL
Vol. 47, No. 9, September 2009

Obtaining and Verifying High-Order Unstructured Finite Volume Solutions to the Euler Equations

Carl Ollivier-Gooch,* Amir Nejat,† and Krzysztof Michalak‡
University of British Columbia, Vancouver, British Columbia V6T 1Z4, Canada

DOI: 10.2514/1.40585

Interest in high-order unstructured-mesh finite volume methods continues to grow, as researchers are attracted both to the possibilities of high-accuracy solutions on coarse meshes and to applications for which high-order accuracy is all but mandatory, including large-eddy simulations for complex geometries. A major obstacle for those interested in developing a high-order solver, however, is that there are many more details that must be exactly right for a nominally-high-order scheme to be genuinely-high-order than for a second-order scheme to be second-order. In many cases, these details are either omitted or glossed over in the literature. This paper attempts to close this gap by illuminating a number of obscure, or often overlooked, aspects of correctly implementing a high-order-accurate unstructured-mesh finite volume solver. We discuss reconstruction, flux integration, curved-boundary treatment, and demonstrating order of accuracy. In each case, we describe test cases that are appropriate for confirming the correct behavior of a high-order code, as well as demonstrating some of the common pitfalls in implementation and showing their effects. We hope that this paper will serve other researchers as a practical guide to creating their own high-order-accurate solvers.

I. Introduction

THE history of scientific computing in general and of computational fluid dynamics in particular is the story of a relentless pursuit of highly accurate solutions to increasingly complex problems with efficient use of computing resources. This desire for highly accurate solutions is the main general motivation for the development of high-order-accurate methods (by which we mean specifically, in this paper, third- and fourth-order-accurate methods). A more specific motivation that has increasing urgency in recent years is that practical application of large-eddy simulations for turbulent flows requires the use of high-order methods to minimize the effect of numerical error on the subgrid-scale model.

The first work on high-order finite volume methods in a production computational fluid dynamics code was, to the authors' knowledge, the high-order structured-mesh discretization included in INS3D, as described by Rogers et al. [1], which used third-order upwinding for the convective terms (though with single-point flux quadrature, which is strictly only third-order-accurate in one dimension) and a second-order centered discretization of the viscous terms. This work, as well as more recent studies ([2,3], for instance), have shown that high-order methods are superior to second-order methods, both in accuracy on a given mesh and in CPU time to compute a solution of given accuracy, in computing aerodynamic lift and drag coefficients on structured meshes. However, careful and detailed analysis of the true error in these computations is generally not attempted.

Since Barth and Frederickson's pioneering work [4], a number of researchers have studied high-order finite volume methods for computational aerodynamics using unstructured meshes. Much of this work has been based, as ours is, on the use of k-exact reconstruction to attain high-order accuracy ([5–8] for example), with Green–Gauss gradients sometimes used to compute viscous fluxes [7,8], even though this reduces the accuracy of viscous terms to second-order. Other researchers favor the essentially nonoscillatory (ENO) approach to reconstruction ([9–11] for instance). Regardless of the reconstruction approach, the other challenges in creating a genuinely-high-order solver remain the same: accurate flux integration, boundary treatment, and so on. For the most part, researchers working on high-order unstructured-mesh schemes are content, for practical cases, to show that high-order schemes are superior in accuracy to second-order schemes without verifying that their scheme attains its nominal order of accuracy. Some of the research described in the literature also explicitly fails to implement all terms to high-order accuracy, especially the boundary discretization and viscous terms. As a result, many existing research codes that are intended to be third- or fourth-order-accurate appear to be second-order in practice.

Despite the accuracy benefits on a particular mesh of properly implemented high-order schemes (which, it is fair to say, are relatively undisputed), these methods have not developed a wide following in the research community. We speculate that there are three reasons for this. First, high-order schemes are perceived as requiring prohibitive CPU time and/or memory, an issue we discuss in detail elsewhere [12–15]. Second, because high-order schemes have less dissipation, they tend to be less robust, especially for flows with shocks; this issue can be resolved for high-order finite volume methods, at least for inviscid and laminar flows, by careful attention to limiting and to design of a robust approach to advancing the solution to steady state [15–18]. Finally, developing a genuinely-high-order-accurate finite volume flow solver requires attention to a large number of details, which a careful reading of the literature suggests are often not adequately addressed and, in fact, regularly appear to be implemented incorrectly. This paper will address the last of these issues by relating our experience [6,19–23] in developing and demonstrating a high-order-accurate solver for the Euler equations.

This paper is intended as, and organized to be, a practical guide in the construction of a high-order-accurate finite volume flow solver. We discuss in turn important, if sometimes subtle, issues related to solution reconstruction (Sec. II), flux integration (Sec. III), and curved-boundary treatment (Sec. IV). In each section, we describe a key element for achieving high-order accuracy as well as the testing

---

Presented as Paper 4464 at the AIAA 18th Computational Fluid Dynamics Conference, Miami, FL, 25–28 June 2007; received 25 August 2008; revision received 22 January 2009; accepted for publication 15 February 2009.

*Associate Professor, Mechanical Engineering, Advanced Numerical Simulation Laboratory, 6250 Applied Science Lane; cfog@mech.ubc.ca. Senior Member AIAA.

†Ph.D. Graduate, Mechanical Engineering, Advanced Numerical Simulation Laboratory; currently Assistant Professor, Mechanical Engineering, University of Tehran, Tehran, Iran. Member AIAA.

‡Ph.D. Graduate, Mechanical Engineering, Advanced Numerical Simulation Laboratory. Student Member AIAA.

procedures we have developed in verifying our code and common pitfalls in algorithm development, including examples of the effects of these potential errors. We will focus our examples on two-dimensional problems, with discussion of numerical issues for both two and three dimensions. The question of designing a limiter that maintains solution monotonicity for high-order schemes without adversely affecting accuracy is sufficiently thorny that we give it separate consideration elsewhere [17,18]. Finally, Sec. V contains a summary and some concluding remarks.

II. Solution Reconstruction

The first and most obvious feature of a high-order-accurate unstructured-mesh finite volume solver is a high-order reconstruction. Although there are other alternatives for obtaining the gradients required by second-order schemes (notably, Green–Gauss reconstruction), obtaining higher than second-order accuracy requires an explicit polynomial reconstruction. In this work, we will focus on k-exact least-squares reconstruction (see [5,7,20], for instance); the use of ENO [9] or weighted ENO [10,11] reconstruction would alter the approach taken to compute reconstruction polynomials, though not the need to properly ensure conservation of the mean or approaches to verifying accurate reconstruction, as described subsequently.

The techniques of high-order-accurate k-exact reconstruction are well documented elsewhere, and here we will content ourselves with a brief summary, referring interested readers to Barth and Frederickson [4] or Ollivier-Gooch and Van Altena [20] for details.

In the finite volume method, the domain is tessellated into nonoverlapping control volumes. Each control volume V_i has a geometric reference point $\mathbf{x}_i$. Although, in principle, any point can be chosen as the reference point, the usual choices (which we recommend) are the cell centroid for cell-centered control volumes and the vertex for vertex-centered control volumes. The most natural interpretation of the solution values computed by the finite volume method is as control volume averages $\bar{u}_i$ of the solution; this is consistent with the control-volume integrated form of the partial differential equations that are the basis for the finite volume method. For vertex-centered schemes, the interpretation of the data is often as pointwise values at the vertices, especially near boundaries, for which this interpretation leads to finite-difference-like boundary-condition enforcement. However, this pointwise value interpretation of the solution data computed by the finite volume method is not consistent with high-order methods, because the value of a smooth function at the vertex and the control volume average typically differ by $\mathcal{O}(\Delta x)$; even the value of the function at a control volume centroid differs from the control volume average by $\mathcal{O}(\Delta x^2)$. Throughout this paper, we will consistently interpret the solution values from finite volume schemes as control volume averages, and enforce boundary conditions with appropriate boundary flux definitions. Although this results in a different treatment of boundary conditions compared with typical vertex-centered schemes, it is completely consistent with typical boundary-condition implementation for cell-centered schemes.

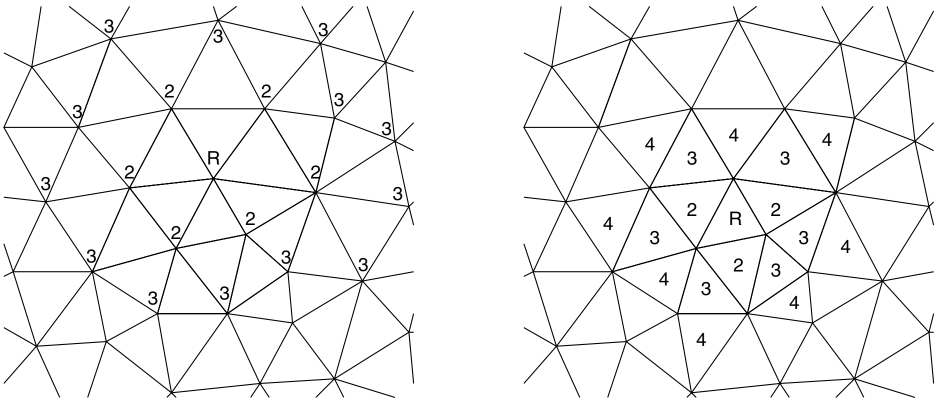
For any smooth function $u(\mathbf{x})$ and its control-volume-averaged values $\bar{u}_i$, the k-exact least-squares reconstruction will use a compact stencil in the neighborhood of control volume i to compute an expansion $R_i(\mathbf{x} - \mathbf{x}_i)$ that conserves the mean in control volume i and reconstructs exactly polynomials of degree $\leq k$ [equivalently, $R_i(\mathbf{x} - \mathbf{x}_i) - u(\mathbf{x}) = \mathcal{O}(\Delta x^{k+1})$]. The minimum number of neighboring control volumes in the reconstruction stencil is equal to the number of derivative terms to be approximated. In practice, we precompute stencils with approximately a 50% excess over this minimum, insisting in two dimensions on at least four neighbors for second-order accuracy, nine for third-order, and 15 for fourth-order. In three dimensions, we require at least 4, 15, and 25 neighbors for second-, third-, and fourth-order accuracy, respectively. Control volumes are added to the reconstruction stencil based on their 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 two-dimensional 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 this 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 are constructed using the same principles, although more layers of neighbors are typically required to get large enough stencils. Construction of stencils in three dimensions is completely analogous to these two-dimensional examples.

For brevity, we will write the reconstruction polynomial and describe how to compute its coefficients in two dimensions; in three dimensions, the approach is identical, though with additional derivatives included. Conservation of the mean requires that the average of the reconstructed function R_i and the original function u over control volume $i(\mathrm{CV}_i)$ be the same:

$$\frac{1}{A_i}\int_{\mathrm{CV}_i} R_i(\mathbf{x} - \mathbf{x}_i)\,\mathrm{d}A = \frac{1}{A_i}\int_{\mathrm{CV}_i} u(\mathbf{x})\,\mathrm{d}A \equiv \bar{u}_i \tag{1}$$

The expansion $R_i(\mathbf{x} - \mathbf{x}_i)$ can be written as

$$\begin{aligned} R_i(\mathbf{x} - \mathbf{x}_i) = u|_{\mathbf{x}_i} + \left.\frac{\partial u}{\partial x}\right|_{\mathbf{x}_i}(x - x_i) + \left.\frac{\partial u}{\partial y}\right|_{\mathbf{x}_i}(y - y_i) \\ + \left.\frac{\partial^2 u}{\partial x^2}\right|_{\mathbf{x}_i}\frac{(x - x_i)^2}{2} + \left.\frac{\partial^2 u}{\partial x \partial y}\right|_{\mathbf{x}_i}((x - x_i)(y - y_i)) \\ + \left.\frac{\partial^2 u}{\partial y^2}\right|\frac{(y - y_i)^2}{2} + \cdots \end{aligned} \tag{2}$$



a) Vertex-centered control volumes **b) Cell-centered control volumes**

Fig. 1 Sample reconstruction stencils.

Taking the control volume average of this expansion over control volume i and equating it to the mean value gives

$$\bar{u}_i = u|_{\mathbf{x}_i} + \left.\frac{\partial u}{\partial x}\right|_{\mathbf{x}_i} \bar{x}_i + \left.\frac{\partial u}{\partial y}\right|_{\mathbf{x}_i} \bar{y}_i + \left.\frac{\partial^2 u}{\partial x^2}\right|_{\mathbf{x}_i} \frac{\overline{x^2}_i}{2} + \left.\frac{\partial^2 u}{\partial x \partial y}\right|_{\mathbf{x}_i} \overline{xy}_i + \left.\frac{\partial^2 u}{\partial y^2}\right|_{\mathbf{x}_i} \frac{\overline{y^2}_i}{2} + \cdots \tag{3}$$

where

$$\overline{x^n y^m}_i \equiv \frac{1}{A_i} \int_{\mathrm{CV}_i} (x - x_i)^n (y - y_i)^m \,\mathrm{d}A \tag{4}$$

are control volume moments.

To achieve kth-order accuracy requires that we compute the kth derivatives by minimizing the error in predicting the mean value of the reconstructed function for control volumes in the stencil $\{\mathrm{CV}_j\}_i$: that is, by minimizing the difference between the actual control volume average $\bar{u}_j$ and the average of R_i over control volume j. The mean value for a single control volume V_j of the reconstructed function R_i is

$$\begin{aligned}\frac{1}{A_j}\int_{\mathrm{CV}_j} R_i(\mathbf{x} - \mathbf{x}_i)\,\mathrm{d}A &= u|_{\mathbf{x}_i} + \left.\frac{\partial u}{\partial x}\right|_{\mathbf{x}_i} \left(\frac{1}{A_j}\int_{\mathrm{CV}_j} (x - x_i)\,\mathrm{d}A\right) \\ &+ \left.\frac{\partial u}{\partial y}\right|_{\mathbf{x}_i} \left(\frac{1}{A_j}\int_{\mathrm{CV}_j} (y - y_i)\,\mathrm{d}A\right) \\ &+ \left.\frac{\partial^2 u}{\partial x^2}\right|_{\mathbf{x}_i} \left(\frac{1}{2A_j}\int_{\mathrm{CV}_j} (x - x_i)^2 \mathrm{d}A\right) \\ &+ \left.\frac{\partial^2 u}{\partial x \partial y}\right|_{\mathbf{x}_i} \left(\frac{1}{A_j}\int_{\mathrm{CV}_j} (x - x_i)(y - y_i)\,\mathrm{d}A\right) \\ &+ \left.\frac{\partial^2 u}{\partial y^2}\right|_{\mathbf{x}_i} \left(\frac{1}{2A_j}\int_{\mathrm{CV}_j} (y - y_i)^2\,\mathrm{d}A\right) + \cdots\end{aligned} \tag{5}$$

To avoid computing moments of each control volume in $\{\mathrm{CV}_j\}_i$ about $\mathbf{x}_i$, 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, we obtain

$$\begin{aligned}\frac{1}{A_j}\int_{\mathrm{CV}_j} R_i(\mathbf{x} - \mathbf{x}_i) &= u|_{\mathbf{x}_i} + \left.\frac{\partial u}{\partial x}\right|_{\mathbf{x}_i} (\bar{x}_j + (x_j - x_i)) \\ &+ \left.\frac{\partial u}{\partial y}\right|_{\mathbf{x}_i} (\bar{y}_j + (y_j - y_i)) \\ &+ \left.\frac{\partial^2 u}{\partial x^2}\right|_{\mathbf{x}_i} \frac{\bar{x}_j^2 + 2\bar{x}_j(x_j - x_i) + (x_j - x_i)^2}{2} \\ &+ \left.\frac{\partial^2 u}{\partial x \partial y}\right|_{\mathbf{x}_i} (\overline{xy}_j + \bar{x}_j(y_j - y_i) + (x_j - x_i)\bar{y}_j + (x_j - x_i)(y_j - y_i)) \\ &+ \left.\frac{\partial^2 u}{\partial y^2}\right|_{\mathbf{x}_i} \frac{\bar{y}_j^2 + 2\bar{y}_j(y_j - y_i) + (y_j - y_i)^2}{2} + \cdots\end{aligned} \tag{6}$$

This equation is written for every control volume in the stencil, of which there must be more than the number of derivatives to be computed, creating an overconstrained system. We satisfy the mean constraint exactly and minimize the sum of the squares of the errors in approximating the control volume averages in nearby control volumes. That is, after eliminating the constraint, we solve the other equations in a least-squares sense. If we write the mean constraint together with Eq. (6) for each control volume in the stencil, we have

$$\begin{bmatrix} 1 & \bar{x} & \bar{y} & \bar{x}^2 & \overline{xy} & \bar{y}^2 & \cdots \\ w_{i1} & w_{i1}\hat{x}_{i1} & w_{i1}\hat{y}_{i1} & w_{i1}\hat{x}^2_{i1} & w_{i1}\widehat{xy}_{i1} & w_{i1}\hat{y}^2_{i1} & \cdots \\ w_{i2} & w_{i2}\hat{x}_{i2} & w_{i2}\hat{y}_{i2} & w_{i2}\hat{x}^2_{i2} & w_{i2}\widehat{xy}_{i2} & w_{i2}\hat{y}^2_{i2} & \cdots \\ w_{i3} & w_{i3}\hat{x}_{i3} & w_{i3}\hat{y}_{i3} & w_{i3}\hat{x}^2_{i3} & w_{i3}\widehat{xy}_{i3} & w_{i3}\hat{y}^2_{i3} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \\ w_{iN} & w_{iN}\hat{x}_{iN} & w_{iN}\hat{y}_{iN} & w_{iN}\hat{x}^2_{iN} & w_{iN}\widehat{xy}_{iN} & w_{iN}\hat{y}^2_{iN} & \cdots \end{bmatrix} \times \begin{pmatrix} u \\ \frac{\partial u}{\partial x} \\ \frac{\partial u}{\partial y} \\ \frac{1}{2}\frac{\partial^2 u}{\partial x^2} \\ \frac{\partial^2 u}{\partial x \partial y} \\ \frac{1}{2}\frac{\partial^2 u}{\partial y^2} \\ \vdots \end{pmatrix}_i = \begin{pmatrix} \bar{u}_i \\ w_{i1}\bar{u}_1 \\ w_{i2}\bar{u}_2 \\ w_{i3}\bar{u}_3 \\ \vdots \\ w_{iN}\bar{u}_N \end{pmatrix} \tag{7}$$

where the first row is the mean constraint, the geometric terms have been abbreviated as

$$\begin{aligned}\widehat{x^n y^m}_{ij} &\equiv \frac{1}{A_j}\int_{\mathrm{CV}_j} ((x - x_j) + (x_j - x_i))^n ((y - y_j) + (y_j - y_i))^m \,\mathrm{d}A \\ &= \sum_{l=0}^{m} \sum_{k=0}^{n} \frac{m!}{l!(m-l)!} \frac{n!}{k!(n-k)!} (x_j - x_i)^k (y_j - y_i)^l \,\overline{x^{n-k} y^{m-l}}_j\end{aligned}$$

and the weights are set to emphasize geometrically nearby data:

$$w_{ij} = \frac{1}{|\mathbf{x}_j - \mathbf{x}_i|^p} \tag{8}$$

We use $p = 1$ in the geometric weight, as we have found that higher values slow convergence. We do not recommend $p = 0$ because of its known adverse effects when reconstructing data on anisotropic meshes [24].

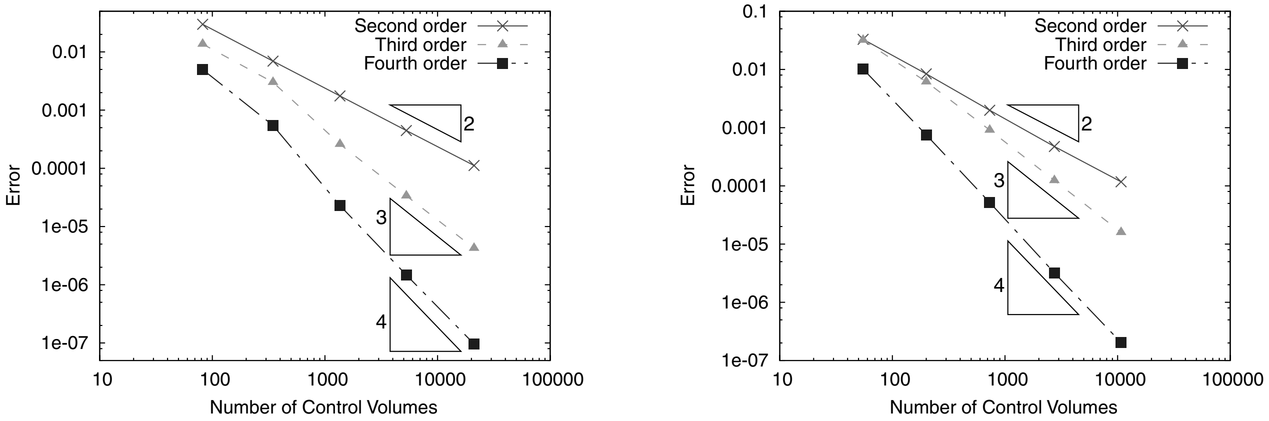
Typically, the least-squares problem of Eq. (7) is actually coded with the mean constraint eliminated analytically. After solving the remaining unconstrained least-squares problem, the mean constraint [Eq. (1)] must be used to compute the constant term in the expansion. For efficiency, we choose to compute and store a singular-value decomposition (see, for instance, [25] for details) of the matrix in Eq. (7). By computing and storing the singular-value decomposition as a preprocessing step, we can replace the least-squares problem with a matrix-vector multiplication at runtime, saving significant CPU time and accepting a reasonable cost in memory usage.

The correctness of reconstruction can be tested by confirming that all monomials of degree $\leq k$ are reconstructed exactly. A prerequisite for this test, obviously, is that control volume averages can be computed to at least the order of accuracy of the reconstruction being tested; see Sec. IV for details on how to do this for control volumes with curved boundaries. We recommend using this test, as it is both a complete test and an efficient one, in that errors in reconstructing a particular monomial are nearly always traceable to errors in the corresponding column of the least-squares system, either in moment calculation or matrix assembly.

In addition to testing monomials, we also recommend testing reconstruction of an arbitrary smooth function on a geometrically simple domain, to confirm that both maximum error (L_∞ norm) and average error (L_1 and/or L_2 norms) behave as expected. We show here the results of reconstructing the function

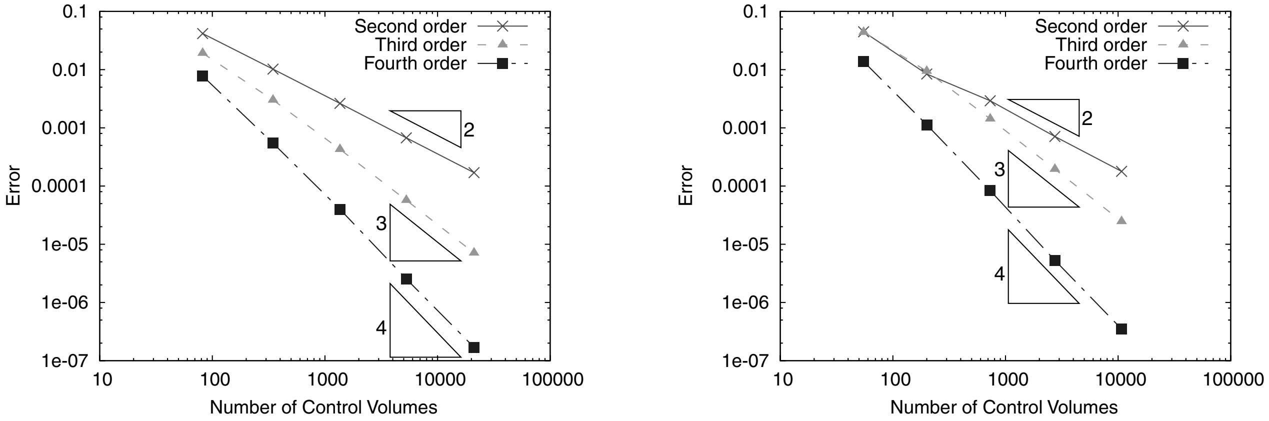
$$f(x, y) = \sin(\pi x)\cos(2\pi y)$$

over the square domain $[0, 1] \times [0, 1]$. A series of five quasi-uniform triangular meshes was generated within a unit square, ranging in size from 82 to 21,087 cells (55 to 10,746 vertices). The test function $f(x, y)$ was averaged over the control volumes using a six-order-accurate quadrature rule, and these control volume averages were reconstructed using linear, quadratic, and cubic reconstruction. Reconstruction error is assessed in the L_1 and L_2 norms by integration. For each control volume, we integrate the absolute value or square of the difference between the original function and the reconstruction computed for that control volume. The full-domain integral is simply the sum of the per-control-volume error. To avoid introducing integration error into the norms, we use a sixth-order quadrature rule in two dimensions (see Sec. III); for tetrahedra, we content ourselves with fourth-order integration (see Sec. III). For vertex-centered control volumes, we first decompose the control volume into triangles in two dimensions (respectively, tetrahedra in three dimensions); each triangle (respectively, tetrahedron) of the mesh is decomposed into six triangles (respectively, 24 tetrahedra), with two (respectively, six) of these being associated with each vertex. The L_∞ norm is approximated by finding the largest difference between the function and reconstruction at any of the quadrature points in the domain. The results are shown in Fig. 2. As expected, both cell- and vertex-centered schemes show well-behaved convergence of the reconstruction to the original function, in the



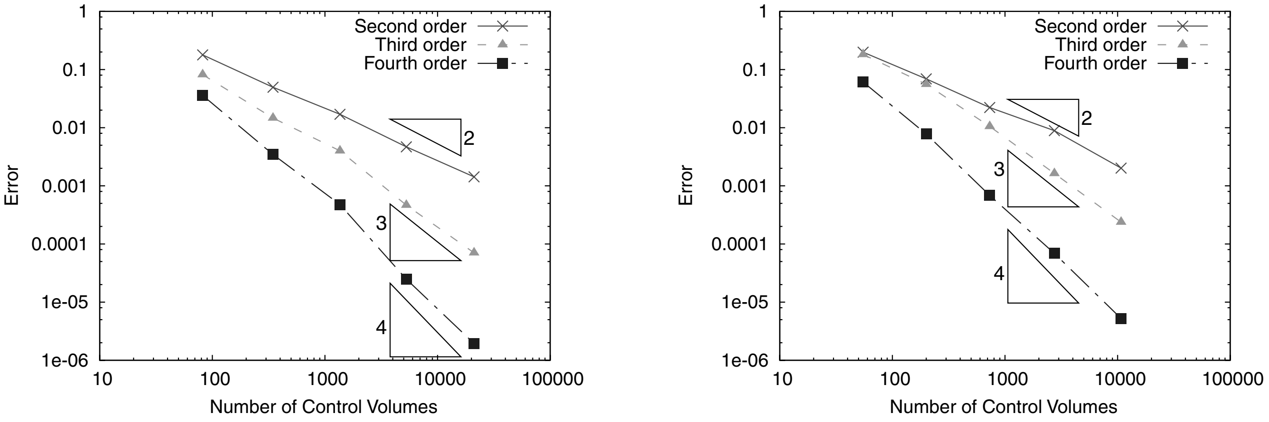
a) L_1 norm, cell-centered control volumes

b) L_1 norm, vertex-centered control volumes



c) L_2 norm, cell-centered control volumes

d) L_2 norm, vertex-centered control volumes



e) L_∞ norm, cell-centered control volumes

f) L_∞ norm, vertex-centered control volumes

Fig. 2 Errors in the reconstruction scheme should converge at the nominal order of accuracy in all norms, as shown here.

Table 1 Computed asymptotic order of accuracy for reconstruction based on the three finest meshes

	Measured asymptotic order		
Nominal order	L_1	L_2	L_∞
Cell-centered control volume			
2	2.008	1.996	1.805
3	3.000	2.990	2.944
4	3.994	3.983	3.996
Vertex-centered control volume			
2	2.107	2.079	1.786
3	3.013	3.021	2.819
4	4.123	4.075	3.628

sense that asymptotic convergence rates are achieved even for very coarse meshes. In addition, the reconstruction error is always decreased by increasing the order of accuracy for cell-centered schemes; for vertex-centered reconstruction on very coarse meshes, this case shows little difference between the linear and quadratic reconstructions. Finally, Table 1 shows the computed asymptotic order of accuracy, based on the three finest meshes; in all cases, the nominal order is achieved for the L_1 and L_2 norms; the data for L_∞ norms, although less perfectly behaved, still show convergence rates that are quite close to the nominal order of accuracy, especially for cell-centered control volumes.

Equation (3) clearly implies that the value at the control volume reference point must be computed explicitly rather than simply using the control volume average as the reference point value u_i in the expansion of R_i. Neglecting this key point introduces a second-order error in the solution reconstruction when the control volume centroid is used at the reference point, as shown in Fig. 3a for the L_2 norm. The high-order schemes degenerate to second-order accuracy in this case, with somewhat less error than the second-order scheme because solution curvature is captured by the reconstruction; also, note that the broken third- and fourth-order schemes show identical behavior for fine meshes, in which the differences between them are swamped by the large error in the mean value. For vertex-centered control volumes, the situation is even worse, with even a linear reconstruction producing errors if the offset is not taken into account, as shown in Fig. 3b, with the largest errors in practice falling near the domain boundary, in which the distance between vertices and control volume centroids is the largest.

As a practical matter, the control volume moments defined by Eq. (4) are easiest to compute for control volumes with curved sides by using Gauss's theorem to convert them to integrals around V_i:

$$\overline{x^n y^m} = \frac{1}{(n+1)A_i} \int_{\partial V_i} (x - x_i)^{n+1} (y - y_i)^m \hat{n}_x \, \mathrm{d}s \tag{9}$$

where $\hat{n}_x$ is the x component of the outward unit normal on the control volume boundary, and similarly in three dimensions. Converting the integral to a contour integral by using Green's theorem gives identical 2-D results in theory. In practice, for straight control volume boundaries, the numerical integration is identical as well; however, for curved control volume boundaries (see Sec. IV), Gauss's theorem gives an integral that is much easier to evaluate numerically, because the integration is along the arc length, not in y, as Green's theorem requires. The integrals of Eq. (9) may be evaluated exactly by using a Gaussian quadrature of appropriate order along each segment of the boundary of the control volume, as summarized in Table 2. Note that the values given in this table assume straight control volume boundaries (and therefore a constant normal vector); for curved control volume boundaries, three Gauss points are required for all moments to account for variations in the direction of the normal.

Table 2 Minimum number of quadrature points required for computing control volume moments by integration around the control volume, as in Eq. (9)

Moment	Required for order ≥	Number of Gauss points
Area	1	1
$\bar{x}$, $\bar{y}$	2	2
$\overline{x^2}$, $\overline{xy}$, $\overline{y^2}$	3	2
$\overline{x^3}$, $\overline{x^2y}$, $\overline{xy^2}$, $\overline{y^3}$	4	3

III. Flux Integration

Given a high-order reconstruction, as described in the previous section, high-order flux calculation is a straightforward matter of applying a flux function. To compute a high-order flux integral from this flux data requires a numerical quadrature scheme of commensurate accuracy. In two dimensions, we prefer Gauss quadrature because of its efficiency: Gauss quadrature with n points can exactly integrate polynomials of degree $2n - 1$. For flux integration in three dimensions, quadrature rules are more complex; we integrate over triangles using the rules recommended by Stroud and Secrest [26], found in Table 3. Point locations in the table are abbreviated using barycentric coordinates. That is, for a triangle with vertices located at $\mathbf{x}_a$, $\mathbf{x}_b$, and $\mathbf{x}_c$, a point location shown as (a, b, c) expands to $a\mathbf{x}_a + b\mathbf{x}_b + c\mathbf{x}_c$. Note that the negative weight for the centroid in the fourth-order quadrature rule could result in a computed average value that falls outside the range of function values in the triangle, especially for rapidly varying integrands. However, we have been unable to find a quadrature rule with positive coefficients, all quadrature points on or inside the triangle, and fewer than seven quadrature points.

Once flux integration code has been written, we recommend testing it by comparing the computed flux integral with the integral over the control volume of the flux divergence. That is, we recommend demonstrating that the discretization satisfies Gauss's theorem to within truncation error; that is, in two dimensions,

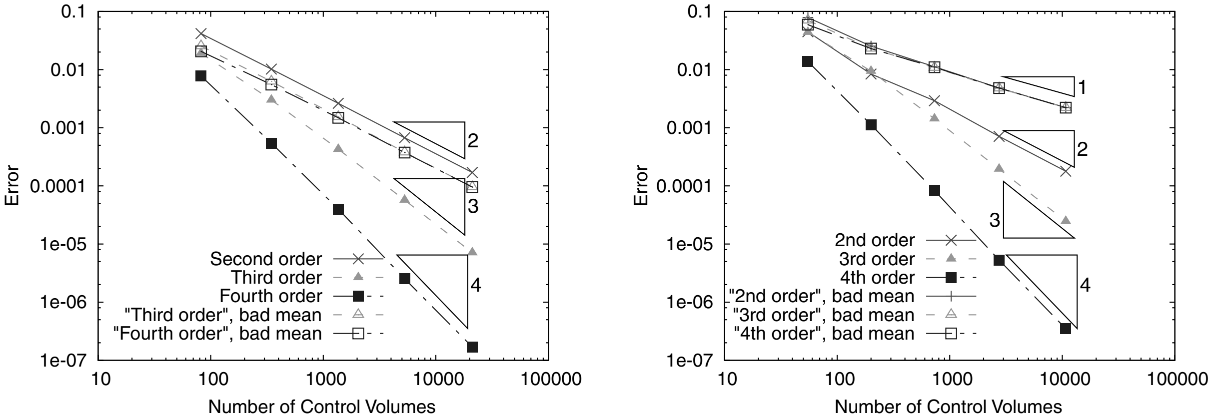


Fig. 3 The effects of failing to properly conserve the mean in computing a high-order reconstruction are severe, reducing all schemes to second-order accuracy for cell-centered control volumes and to *first*-order accuracy for vertex-centered control volumes.

Table 3 Quadrature points and weights for triangles, which we apply for flux integration in three dimensions and for integration over control volumes in two dimensions (weights are normalized by triangle area)

Point location	Weight
Second order (centroid)	
$(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$	1
Third order (edge midsides)	
$(\frac{1}{2}, \frac{1}{2}, 0)$	$\frac{1}{3}$
$(\frac{1}{2}, 0, \frac{1}{2})$	$\frac{1}{3}$
$(0, \frac{1}{2}, \frac{1}{2})$	$\frac{1}{3}$
Fourth order (centroid plus three interior points)	
$(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$	$-\frac{9}{16}$
$(\frac{1}{5}, \frac{1}{5}, \frac{3}{5})$	$\frac{25}{48}$
$(\frac{1}{5}, \frac{3}{5}, \frac{1}{5})$	$\frac{25}{48}$
$(\frac{3}{5}, \frac{1}{5}, \frac{1}{5})$	$\frac{25}{48}$
Sixth order (centroid plus six interior points)[a]	
$(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$	$\frac{9}{40}$
(r, r, s)	A
(r, s, r)	A
(s, r, r)	A
(t, t, u)	B
(t, u, t)	B
(u, t, t)	B

[a] $r = (6 - \sqrt{15})/21$, $s = (9 + 2\sqrt{15})/21$, $t = (6 + \sqrt{15})/21$, $u = (9 - 2\sqrt{15})/21$, $A = (155 - \sqrt{15})/1200$, and $B = (155 + \sqrt{15})/1200$.

$$\int_{\mathrm{CV}} \nabla \cdot \mathbf{F}\, \mathrm{d}A = \oint_{\partial\mathrm{CV}} \mathbf{F} \cdot \hat{n}\, \mathrm{d}s$$

or its equivalent in three dimensions. This test requires specifying a physically plausible analytic solution (for instance, density and pressure must be positive), along with coding the divergence of the flux for this solution explicitly. This approach is a simplified variant on the method of manufactured solutions [27], in which only the flux integral is verified, rather than the entire code. In particular, we typically choose to ignore boundary-condition issues, leaving verification of boundary conditions for a separate step.

Before examining results, however, we should determine what order of accuracy we expect to see in the computed flux integrals. Consider the triangular control volume illustrated in Fig. 4 and a second-order-accurate scheme (which implies a single Gauss point at the midside of each edge of the triangle). Now the flux integral around this triangle can be written as

$$\oint_{ABC} \mathcal{F} \cdot \hat{n}\, \mathrm{d}s = \sum \mathcal{F}_i \cdot \hat{n}_i l_i$$

where the sum is taken over all segments in the control volume boundary, $\mathcal{F} \equiv F\hat{i} + G\hat{j}$, $\mathcal{F}_i$ is evaluated at the midside of the ith face, and l_i is the length of that face. The fluxes F and G can be expanded in a Taylor series about the control volume centroid $\mathbf{x}_0$:

$$F_i = F_0 + \frac{\partial F}{\partial x}\Delta x_i + \frac{\partial F}{\partial y}\Delta y_i + \frac{\partial^2 F}{\partial x^2}\frac{\Delta x_i^2}{2} + \frac{\partial^2 F}{\partial x \partial y}\Delta x_i \Delta y_i + \frac{\partial^2 F}{\partial y^2}\frac{\Delta y_i^2}{2} + \cdots$$

$$G_i = G_0 + \frac{\partial G}{\partial x}\Delta x_i + \frac{\partial G}{\partial y}\Delta y_i + \frac{\partial^2 G}{\partial x^2}\frac{\Delta x_i^2}{2} + \frac{\partial^2 G}{\partial x \partial y}\Delta x_i \Delta y_i + \frac{\partial^2 G}{\partial y^2}\frac{\Delta y_i^2}{2} + \cdots$$

Inserting these in the sum and regrouping, we get

$$\begin{aligned}\sum \mathcal{F}_i \cdot \hat{n}_i l_i ={}& F_0\left(\sum n_{x,i} l_i\right) + G_0\left(\sum n_{y,i} l_i\right)\\ &+ \sum\left(\frac{\partial F}{\partial x}\Delta x_i + \frac{\partial F}{\partial y}\Delta y_i\right) n_{x,i} l_i\\ &+ \sum\left(\frac{\partial G}{\partial x}\Delta x_i + \frac{\partial G}{\partial y}\Delta y_i\right) n_{y,i} l_i\\ &+ \sum\left(\frac{\partial^2 F}{\partial x^2}\frac{\Delta x_i^2}{2} + \frac{\partial^2 F}{\partial x \partial y}\Delta x_i \Delta y_i + \frac{\partial^2 F}{\partial y^2}\frac{\Delta y_i^2}{2}\right) n_{x,i} l_i\\ &+ \sum\left(\frac{\partial^2 G}{\partial x^2}\frac{\Delta x_i^2}{2} + \frac{\partial^2 G}{\partial x \partial y}\Delta x_i \Delta y_i + \frac{\partial^2 G}{\partial y^2}\frac{\Delta y_i^2}{2}\right) n_{y,i} l_i + \cdots\end{aligned} \tag{10}$$

The first line in Eq. (10) is identically zero for a closed control volume. The second and third lines are the result we actually want to compute: the integral of the flux divergence over the control volume; note that it is a second-order quantity in the characteristic length for the cell, as Δx, Δy, and l_i are all proportional to edge length. The fourth and fifth lines are third-order terms in the characteristic length, and these are terms that are not calculated exactly in the second-order scheme. More precisely, the second derivative terms are explicitly missing from the solution reconstruction, which implies that any evaluation of the fluxes based on the (linear) reconstruction will not have a sufficiently accurate solution to compute the second derivative terms in the expansion of the fluxes.[§] Because we are comparing $\sum \mathcal{F}_i \cdot \hat{n}_i l_i$ with the analytic flux divergence $\int \nabla \cdot \mathcal{F} \mathrm{d}A$, we expect to measure a third-order error in flux integration with a second-order-accurate reconstruction. This is consistent with the results described by Diskin and Thomas [28], who analyzed the behavior of the flux integral divided by control volume size. For higher orders of accuracy, we can conclude from analogous arguments that we again expect flux integration error to be one order smaller than the reconstruction error.

Though the results of this analysis are perhaps counterintuitive, the analysis is consistent with the observed behavior of real solvers, which are capable of producing a k-order-accurate approximation to the exact solution of a problem from a kth-order-accurate reconstruction. One way to confirm this is by considering the effect, for a second-order scheme, of a second-order perturbation to the solution. Such a perturbation produces a second-order change in the mean fluxes F_0 and G_0, but a first-order perturbation in the gradients of the fluxes ($\partial F/\partial x$, etc.), because flux gradients depend directly on solution gradients, which in turn are only computed to first-order accuracy. This will cause a first-order change in the terms in lines 2 and 3 of Eq. (10) (before multiplication by Δx_i and l_i), which is the correct magnitude to cancel the errors in higher flux derivatives. Thus, at least on an order-of-magnitude basis, we can reasonably expect that solutions that differ by only $\mathcal{O}(h^k)$ can produce flux integrals that differ by $\mathcal{O}(h^{k+1})$ in the measure we use.

For our present purposes in testing the flux integral for the Euler equations in two dimensions, we specify an arbitrary solution in normalized primitive variables:

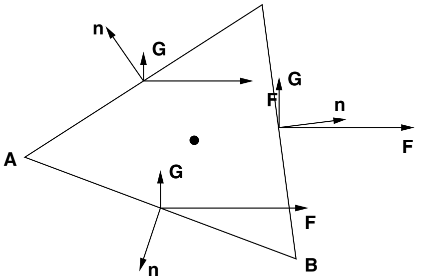
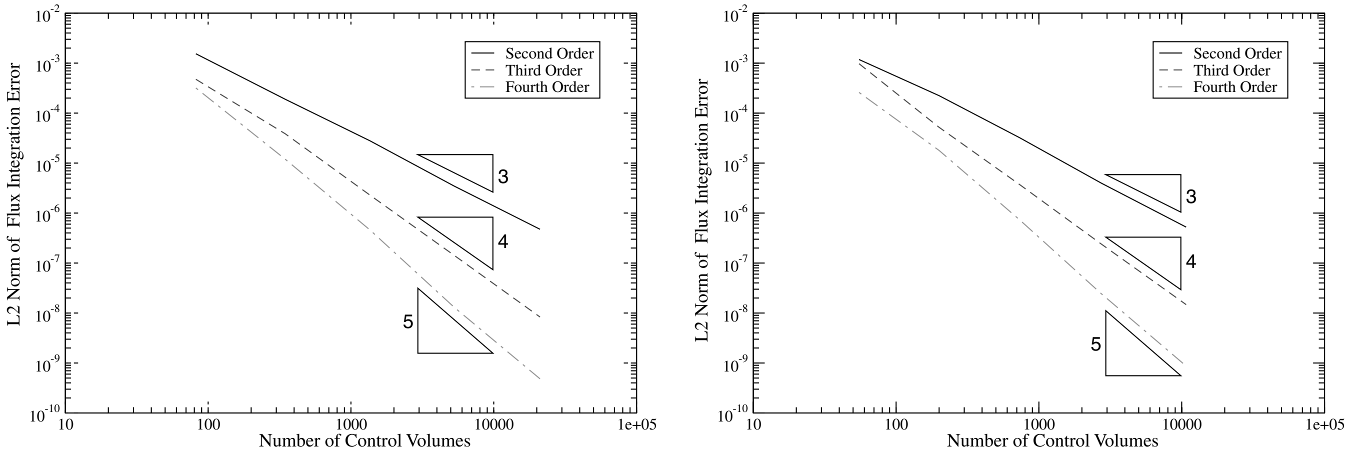


Fig. 4 Control volume used in analysis of the order of accuracy of flux integration, including fluxes and surface normals.

[§] It is worth noting here that structured-mesh schemes achieve a one-order-better flux integral because of cancellation in the error terms, at least for even orders of accuracy.



a) Cell-centered control volumes **b) Vertex-centered control volumes**

Fig. 5 Errors in the flux integral for the Euler equations should converge one order faster than the reconstruction; that is, the error should be $\mathcal{O}(h^{k+1})$ if the reconstruction error is $\mathcal{O}(h^k)$.

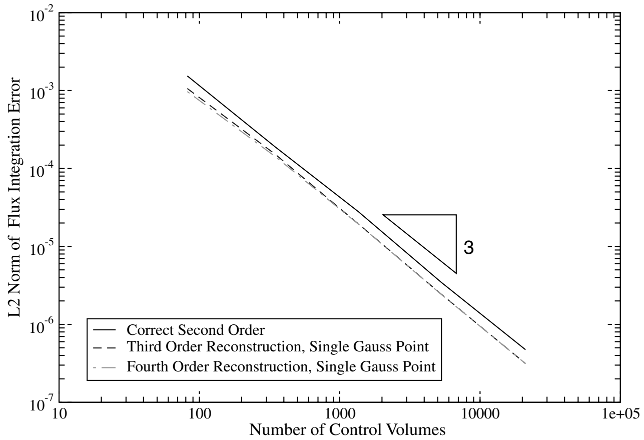


Fig. 6 Using the correct number of Gauss points when computing high-order flux integrals is essential; using only a single Gauss point reduces accuracy to second-order. This figure shows the cell-centered result.

$$\rho = 1 + \rho_0 \sin(\pi x)\sin(\pi y) \qquad u = u_0 \sin(\pi x)\cos(2\pi y)$$

$$v = v_0 \cos(2\pi x)\sin(\pi y) \qquad P = \frac{1}{\gamma} + P_0 \sin(2\pi x)\sin(2\pi y)$$

We use the same sequence of meshes as in the reconstruction test. Roe's flux function was used in the interior, but the results are not sensitive to the choice of flux function, because the solution is smooth. No physical boundary condition was applied; instead, the analytic flux was computed at the boundary based on reconstructed data. All the perturbation magnitudes (ρ_0, u_0, v_0, andP_0) were set to 0.1. We present results in Fig. 5 for the L_2 norms of the error in the energy flux integral; the energy flux integral produces the largest magnitude of error for this case, and the behavior of the L_2 norm is representative of the other cases.

The importance of having the correct number of Gauss points for high-order schemes can be seen in Fig. 6. Essentially, the high-order schemes are behaving like expensive second-order schemes, with little difference in the flux integrals. This result is expected, in that the single-point Gauss quadrature is known to be only second-order-accurate.

For many problems, accurate calculation of source terms is also required; we include a description here for completeness, even though these quadrature rules are unnecessary for the Euler equations. For second-order schemes, it is sufficient to use a single quadrature point located at the control volume centroid. For third- and fourth-order-accurate schemes, more quadrature points are required. In two dimensions, the quadrature rules described previously can be applied directly for cell-centered control volumes, whereas vertex-centered control volumes must be decomposed into triangles before they can be integrated. In three dimensions, multipoint quadrature rules for tetrahedra must be used, as shown in Table 4. Again, quadrature point locations are given in barycentric coordinates; for a tetrahedron with vertices located at $\mathbf{x}_a$, $\mathbf{x}_b$, $\mathbf{x}_c$, and $\mathbf{x}_d$, a point location shown as (a, b, c, d) expands to $a\mathbf{x}_a + b\mathbf{x}_b + c\mathbf{x}_c + d\mathbf{x}_d$. Just as in two dimensions, the negative weight for the centroid in the fourth-order quadrature rule in the table can cause problems, but we have been unable to find a quadrature rule with positive coefficients and all quadrature points on or inside the tetrahedron. We also note that recent work [29] suggests that source term integration can be done one order less accurately than flux integration, which simplifies matters greatly.

IV. Curved Boundaries

Perhaps the most challenging aspect of achieving high-order accuracy is proper treatment of curved boundaries. There are four distinct components to curved-boundary treatment: flux integration along boundaries, flux integration for interior sides of boundary control volumes, computation of control volume moments for boundary control volumes, and integration over control volumes with curved boundaries, both for source terms and for computing exact control volume averages for comparison when validating a

Table 4 Quadrature points and weights for tetrahedra, which we apply for source term integration in three dimensions (weights are normalized by tetrahedron volume)

Point location	Weight
Second order (centroid)	
$(\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4})$	1
Third order (four interior points)[a]	
(r, r, s, r)	$\frac{1}{4}$
(r, s, r, r)	$\frac{1}{4}$
(s, r, r, r)	$\frac{1}{4}$
Fourth order (centroid plus four interior points)	
$(\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4})$	$-\frac{4}{5}$
$(\frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{2})$	$\frac{9}{20}$
$(\frac{1}{6}, \frac{1}{6}, \frac{1}{2}, \frac{1}{6})$	$\frac{9}{20}$
$(\frac{1}{6}, \frac{1}{2}, \frac{1}{6}, \frac{1}{6})$	$\frac{9}{20}$
$(\frac{1}{2}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6})$	$\frac{9}{20}$

[a] Note that $r = \frac{1}{4} - \frac{\sqrt{5}}{20}$ and $s = \frac{1}{4} + \frac{3\sqrt{5}}{20}$.

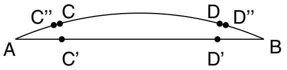


Fig. 7 Curved-boundary arc *AB* and its approximation by a straight-line segment. The arc has a radius of curvature *R*, and the angle subtended by the segment is $\theta \equiv 2\arcsin(l/2R)$.

code. In all of these cases, the key issue is that the curved boundary and its piecewise linear representation (as line segments between boundary vertices) are separated by a distance that is $\mathcal{O}(h^2)$ for edge length h. In general, the discrete approximation of the boundary must approach the true underlying boundary shape with the same order of accuracy as the discretization scheme.

A. Flux Integration Along Curved Boundaries

When integrating fluxes along curved boundaries, care must be taken that the solution values, boundary normals, and integration weights accurately reflect the shape of the boundary. The most straightforward way to do this in two dimensions is to use the boundary representation directly in setting up Gauss integration points. In two dimensions, Gauss point locations are determined as fractions of arc length. Normals at Gauss points are found directly by querying the boundary representation. Finally, Gauss weights are assigned based on arc length.

To illustrate the differences between exact and inexact approximations to Gauss point location, let us consider three possible choices in two dimensions, for a circular arc geometry, shown in Fig. 7.[¶] The arc has a radius of curvature R, and the angle subtended by the segment is $\theta \equiv 2\arcsin(l/2R)$. The three choices of Gauss point considered are as follows:

1) The exact Gauss point locations, based on distance along the arc **AB**, are labeled C and D in the figure.

2) The Gauss point locations for the segment **AB** are labeled C' and D'.

3) The normal projections of points C' and D' onto the arc are labeled C'' and D''.

Finding the locations, weights, and normals for these points is a straightforward exercise in trigonometry; the results are given in Table 5 for points C, C', and C''; only leading-order error terms are retained. As expected, the straight-line segment is a poor approximation, yielding a second-order error in the y component of Gauss point location; this error implies a second-order error in the value of the reconstruction polynomial R_i that will be computed at the Gauss points. The normal direction is only first-order in θ and therefore in l. Taking the segment Gauss point C' and projecting it normally onto the arc makes the dominant error a third-order error in the x component of the Gauss point location. Assuming that the Gauss weights are computed as half of the true arc length, using C'' is therefore a third-order-accurate choice, at least in the naive analysis; we shall return to this point later.

In three dimensions, the most straightforward approach to high-order flux integration with curved boundaries is to apply an isoparametric mapping to the curved surface triangle and transform the integral to the resulting flat triangle:

$$\iint_{T_{xyz}} \mathbf{F}(x,y,z)\cdot\hat{n}(x,y,z)\,\mathrm{d}A(x,y,z)$$
$$= \iint_{T_{uv}} \mathbf{F}(u(x,y,z), v(x,y,z))$$
$$\cdot\,\hat{n}(u(x,y,z), v(x,y,z))J(u,v)\,\mathrm{d}A(u,v)$$

where $J(u,v)$ is the Jacobian of the surface mapping at u and v; the left-hand side is the physical-space integral over a curved triangle T_{xyz} and the right-hand side is a parametric space integral over a flat, straight-sided triangle T_{uv}. The integral over T_{uv} can be evaluated using standard quadrature rules, with the flux dyad $\mathbf{F}$ and the unit normal $\hat{n}$ evaluated at the physical point corresponding to a particular point in parametric space (u, v). In the end, the combination of the Jacobian and area terms define a weight for each term in the numerical integration; this weight, combined with the location of the point (x, y, z) corresponding to each integration point provides precisely the same information required to compute any other flux integral. These values can be precomputed and stored for use at run time. Note that for vertex-centered control volumes, the footprint of a boundary control volume on a curved boundary is a union of multiple quadrilaterals (one for each surface triangle incident on the vertex), requiring more effort, though the same approach applies.

To demonstrate the effect of proper versus improper handling of curved boundaries, we consider the case of an inviscid supersonic vortex flow in a circular annulus [30,31]. Numerical solution of this supersonic isentropic flow requires that a solver correctly simulate weak expansion and compression waves; irregularities in the boundary, or in the boundary treatment, result in clearly visible pressure errors propagating across the domain. We choose an inner radius R_i of 2, an outer radius R_o of 3, a Mach number at the inner boundary M_i of 2, and a density at the inner boundary of $\rho_i = 1$. The exact solution for the supersonic vortex is given in nondimensional form by

$$\rho = \rho_i\left(1 + \frac{\gamma-1}{2}M_i^2\left(1 - \frac{R_i^2}{r^2}\right)\right)^{\frac{1}{\gamma-1}} \qquad U_i = M_i\rho_i^{\frac{\gamma-1}{2}}$$
$$U = \frac{U_i R_i}{r} \qquad u = \frac{yU}{r} \qquad v = \frac{-xU}{r} \qquad P = \frac{\rho^\gamma}{\gamma}$$

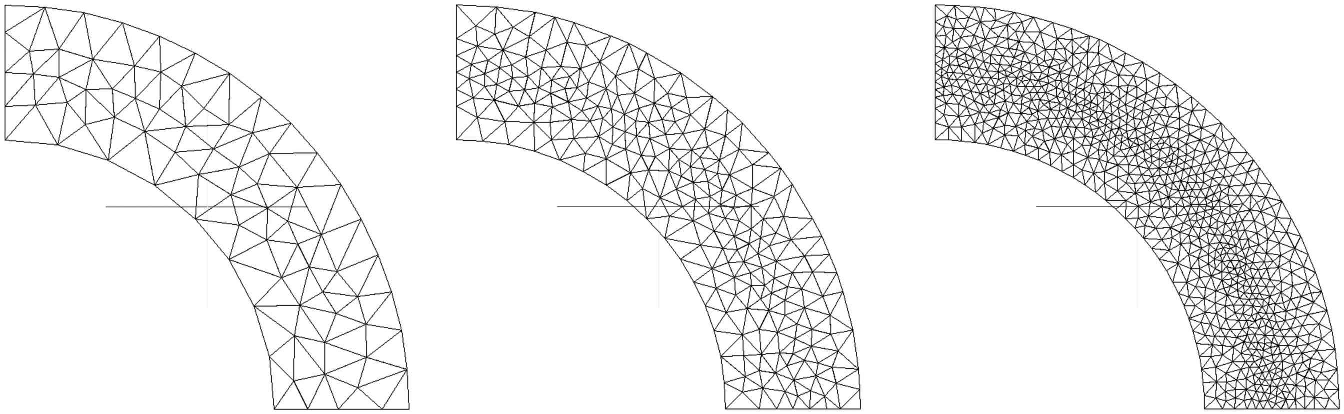
The unusual-looking form of U_i is actually a direct consequence of computing the sound speed with this nondimensionalization. We have solved this problem on a sequence of five meshes containing 128 to 22,058 control volumes, shown in Fig. 8. Error norms were computed by comparing the computed control volume averages with control volume averages of the exact solution, computed using the method described in Sec. III.

With proper boundary treatment, the solution error should have the same asymptotic behavior as the reconstruction, converging at the nominal order of accuracy. Figure 9 shows the error in density for several cases with cell-centered control volumes, for a mesh with 5686 cells (mesh 4, Fig. 8d). The top row of the figure (Figs. 9a, 9b, and 9d) shows results obtained when treating the boundary as a collection of straight-line segments. Although both the third- and fourth-order schemes have a lower error than the second-order scheme, the errors are still substantial for high-order schemes on a relatively fine mesh. Most of the error is concentrated along the inner boundary, where the flow acceleration is the strongest. When the boundaries are properly treated, as described previously, the third- and fourth-order schemes produce dramatically better results (Figs. 9c and 9e), with the third-order error reduced by about a factor of 2 and the fourth-order error reduced by an order of magnitude compared with straight boundaries. No curved-boundary result is presented for the second-order scheme, because the second-order

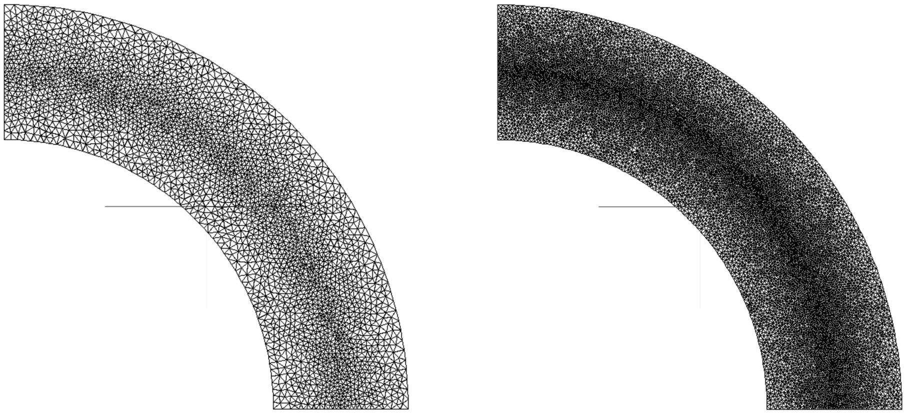
Table 5 Comparison of Gauss integration data for the exact Gauss point locations and two approximate locations.

Quantity	C	C'	C''
x	$R\sin\frac{\theta}{\sqrt{3}}$	$R\frac{\sin\theta}{\sqrt{3}}$	$R\frac{\sin\theta}{\sqrt{2\cos^2\theta+1}}$
y	$R\cos\frac{\theta}{\sqrt{3}}$	$R\cos\theta$	$R\cos\theta\sqrt{\frac{3}{2\cos^2\theta+1}}$
Weight	$R\theta$	$\frac{l}{2}$	$R\theta$
Normal direction	$\frac{\pi}{2}-\frac{\theta}{\sqrt{3}}$	$\frac{\pi}{2}$	$\frac{\pi}{2}-\arctan\frac{\sin\theta}{\sqrt{3}\cos\theta}$
x-location error	——	$\frac{\sqrt{3}}{216}\frac{l^3}{R^2}$	$-\frac{\sqrt{3}}{108}\frac{l^3}{R^2}$
y-location error	——	$\frac{l^2}{12R}$	$\frac{l^4}{216R^3}$
Weight error	——	$\frac{l^3}{48R^2}$	0
Normal angle error	——	$\frac{\theta}{\sqrt{3}}$	$\frac{2\sqrt{3}}{27}\theta^3$

[¶] This choice of geometry is not as restrictive as it sounds, in that any well-resolved curve (where $l \ll R$) asymptotically resembles a circular arc.



a) Mesh 1, 128 control volumes **b) Mesh 2, 430 control volumes** **c) Mesh 3, 1519 control volumes**



d) Mesh 4, 5686 control volumes **e) Mesh 5, 22058 control volumes**

Fig. 8 Meshes used in supersonic vortex calculations.

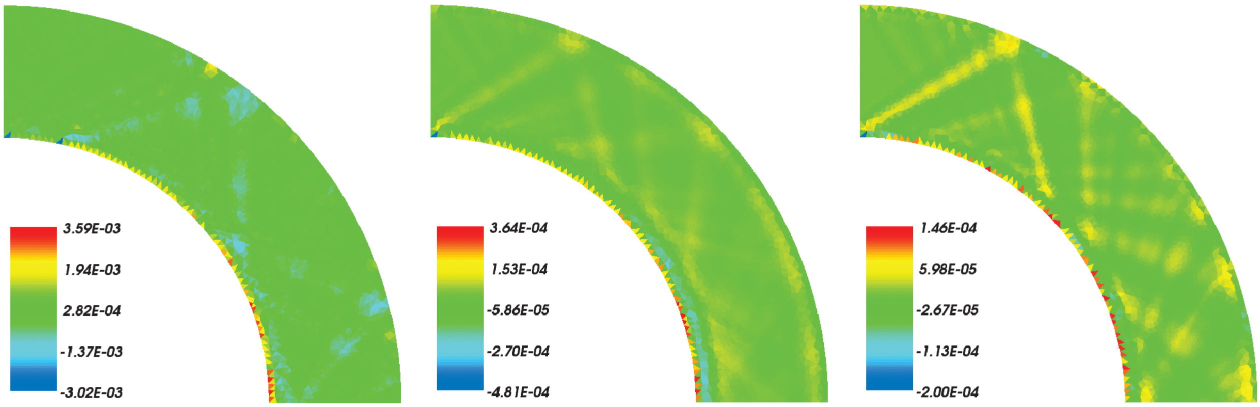
scheme ignores boundary curvature information and thus produces identical results for both cases. Error convergence for both straight and curved boundaries is shown in Figs. 9f and 9g, and asymptotic convergence rates, based on the two finest meshes, are given in Table 6. From both the table and the graphs, it is clear that the third-order scheme is, somewhat surprisingly, insensitive in its measured order of accuracy to whether the boundary is curved or not, with accuracy measured at between second- and third-order in each case; the somewhat lower asymptotic value for the L_∞ norm for curved boundaries appears to be an artifact. Despite this, proper curved-boundary implementation still makes a difference of approximately 20–30% in the magnitude of the error for the third-order scheme in both norms.

The fourth-order scheme shows the expected behavior for straight boundaries: its measured accuracy is second-order in the L_∞ norm and almost exactly one order better in the L_1 norm. This difference in convergence rates is expected, considering that a second-order error is made in $\sqrt{N}$ control volumes out of a total of N control volumes. When curved boundaries are implemented properly, the fourth-order scheme achieves a measured accuracy somewhat above fourth-order in the L_1 norm and somewhat above third-order in the L_∞ norm. We have been unable to determine the source of the degradation in convergence for the L_∞ norm for the fourth-order scheme, though it seems certain to be either an issue with boundary conditions or with interactions between boundary conditions. Given that the quantities of interest in aerodynamic flows are typically surface quantities or integrals, boundary errors are especially noxious, and these results demonstrate that proper treatment of boundaries is therefore especially important for the fourth-order scheme.

We now return our attention to the projected Gauss points (C'' and D'' in Fig. 7). The analysis of this section suggests that using these Gauss points will reduce the accuracy of flux integration to only third-order, because the difference in location of these projected points, compared with the exact Gauss points C and D, is third-order. Direct comparison of the locations of the exact and projected Gauss points indicates that the distance between them is in fact third-order in edge length and that the coefficient found in the analysis (see Table 5) is correct. Figure 10 shows results with both the exact and projected Gauss points for both third- and fourth-order accuracy. The figures, although generated by different runs with different boundary information, are indistinguishable. The error norms, even for the coarsest meshes, differ by no more than one part in several thousand; as such, the asymptotic convergence rates are identical to those given previously for curved boundaries. Two factors may be at work here. First, the difference in location of the Gauss points is very small: even on our coarsest mesh, the error in Gauss point location is never more than about 8×10^{-4}. Second, and more important to the asymptotic behavior, the errors in Gauss point location for the two Gauss points on the curved part of the boundary of a single control volume are of equal magnitude and in opposite directions, and so errors in reconstructed solution and normal direction will tend to cancel out. This is especially true for this test case, in which the boundary is circular and hence Gauss location errors are perfectly symmetric; for boundaries with varying curvature, cancellation will be increasingly exact with mesh refinement because the approximation of the boundary by a circular arc will be increasingly accurate.

B. Control Volume Moments Along Curved Boundaries

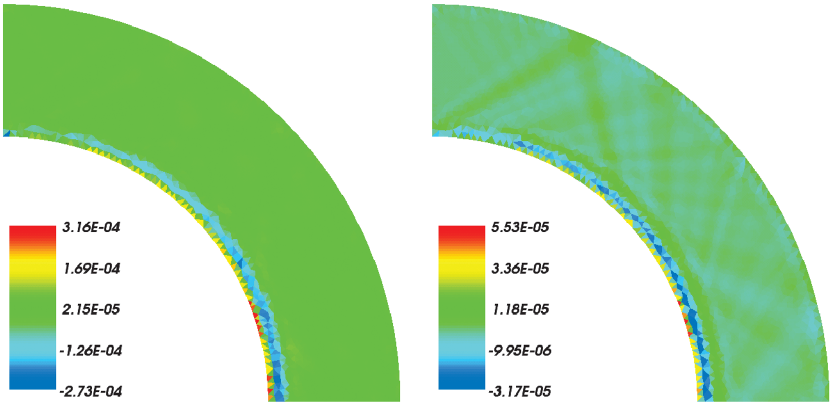
A related point is that control volume moments for control volumes adjacent to curved boundaries must be computed with care. Gauss integration for the moments must be performed along the



a) Second-order

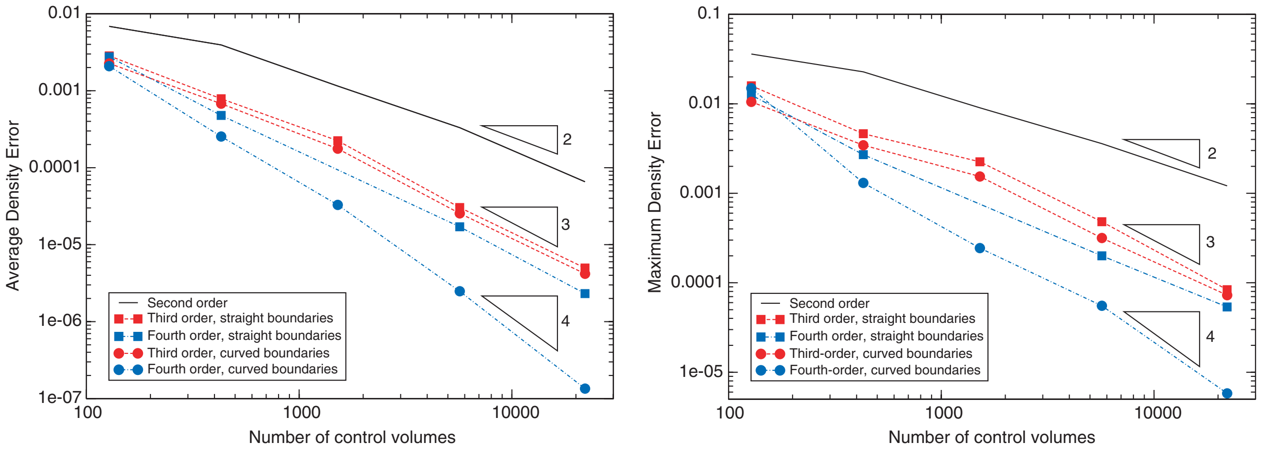
b) Third-order, straight boundaries

c) Third-order, curved boundaries, correct Gauss points



d) Fourth-order, straight boundaries

e) Fourth-order, curved boundaries, correct Gauss points



f) L_1-norm of density error

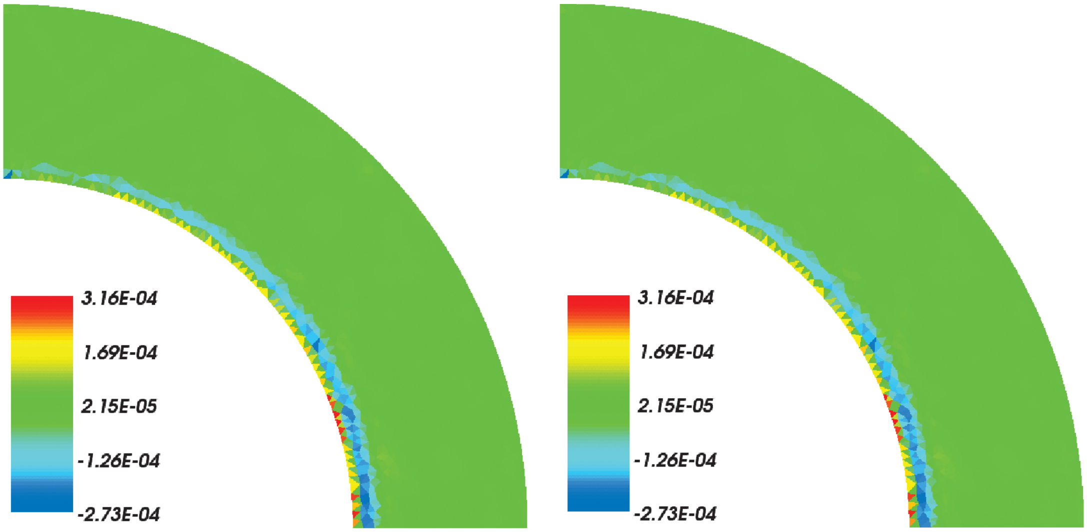
g) L_∞-norm of density error

Fig. 9 Improper treatment of curved boundaries leads to dramatically larger errors and poorer asymptotic convergence rates. This is especially harmful for fourth-order schemes, because there is more to lose by making second-order errors at straight boundaries.

Table 6 Asymptotic convergence rates for the supersonic vortex problem based on the two finest meshes

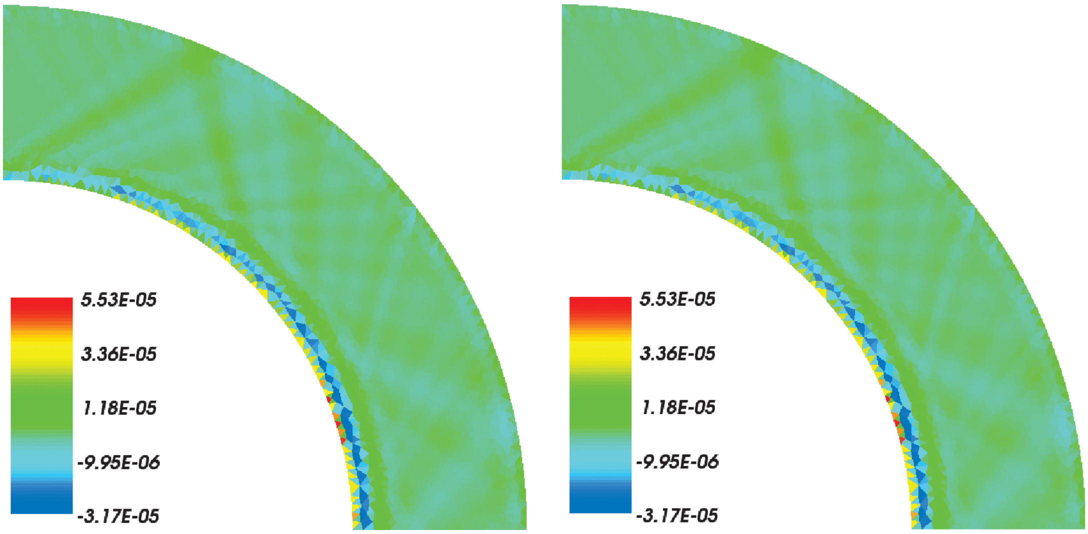
	Measured order			
	Straight boundary		Curved boundary	
Nominal order	L_1	L_∞	L_1	L_∞
2	2.35	1.60	——	——
3	2.67	2.58	2.66	2.17
4	2.95	1.94	4.29	3.32

curved boundaries, with Gauss point locations again determined based on arc length along the boundary. For interior edges, as described earlier, we use the minimum number of Gauss points for exact integration of the monomials in the moment definition. For curved boundaries, the number of Gauss points used in the interior is no longer sufficient, because the normal vector $\hat{n}$ varies along the curved edge. We choose to use three-point Gauss quadrature along curved-boundary edges when computing all the moments, which results in sixth-order-accurate moments for boundary control volumes. Although not as accurate as the exact interior moments, these sixth-order moments are more than accurate enough for practical purposes. In three dimensions, moment calculations should again be done to at least the order of accuracy of the reconstruction.



a) Third-order, correct boundary gauss points

b) Third-order, projected boundary gauss points



c) Fourth-order, correct boundary gauss points

d) Fourth-order, projected boundary gauss points

Fig. 10 The use of projected Gauss points on the curved boundary, although not recommended practice, can be surprisingly accurate, due in part to cancellation of the error when employing two of these Gauss points.

C. Integration over the Interior of Control Volumes Along Curved Boundaries

The final key point in accurately handling control volumes with curved boundaries is integration over the interior of these control volumes. For control volumes that have straight sides, we use Gauss integration over triangles or tetrahedra, as described in Sec. III. For control volumes with curved sides, we could extend that approach using ideas from isoparametric finite elements, but we choose instead to formulate an integration scheme that eliminates the need for decomposition of vertex-centered control volumes into triangles/tetrahedra; this technique is also applicable for simplifying integration over vertex-centered control volumes that are not adjacent to a curved boundary and is more efficient than decomposition into constituent parts. Our approach chooses integration points and weights to ensure that polynomials of degree $\leq k-1$ are integrated exactly, which in turn guarantees that the integration is k^{th}-order-accurate. Mathematically, this requires that for any monomial of degree $\leq k-1$,

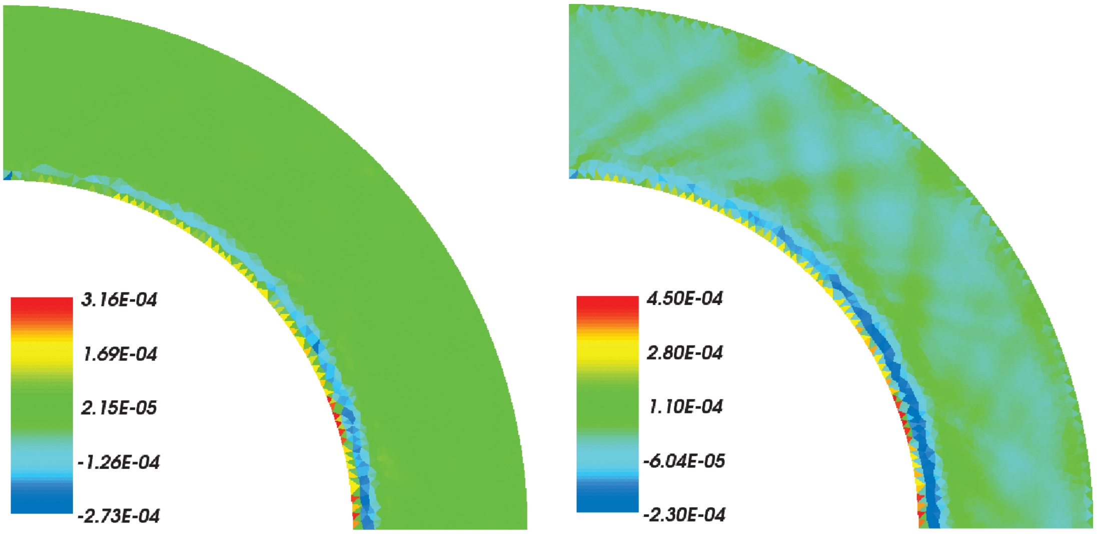
$$\sum_i w_i x_i^m y_i^n = \frac{1}{A}\int_{\mathrm{CV}} x^m y^n \, \mathrm{d}A$$

and similarly in three dimensions; note that this requirement can be restated in words by saying that the weighted moments of the integration points must match the moments of the region being integrated, up to moments of the order $k-1$. The moments themselves require no additional calculations, as they must be computed to high fidelity for reconstruction. In principle, we could use a very small number of integration points (potentially, as few as two for a third-order 2-D scheme, for instance, or as few as five for a fourth-order 3-D scheme), as there are $d+1$ deg of freedom per integration point in d dimensions. However, we have found the resulting nonlinear systems of equations to be intractable analytically. Instead, our approach has been to select enough integration points that we can specify $\mathbf{x}_i$ and solve for w_i. The expressions involved are lengthy and

Table 7 Comparison of function evaluations required for isoparametric integration and the proposed new quadrature rules

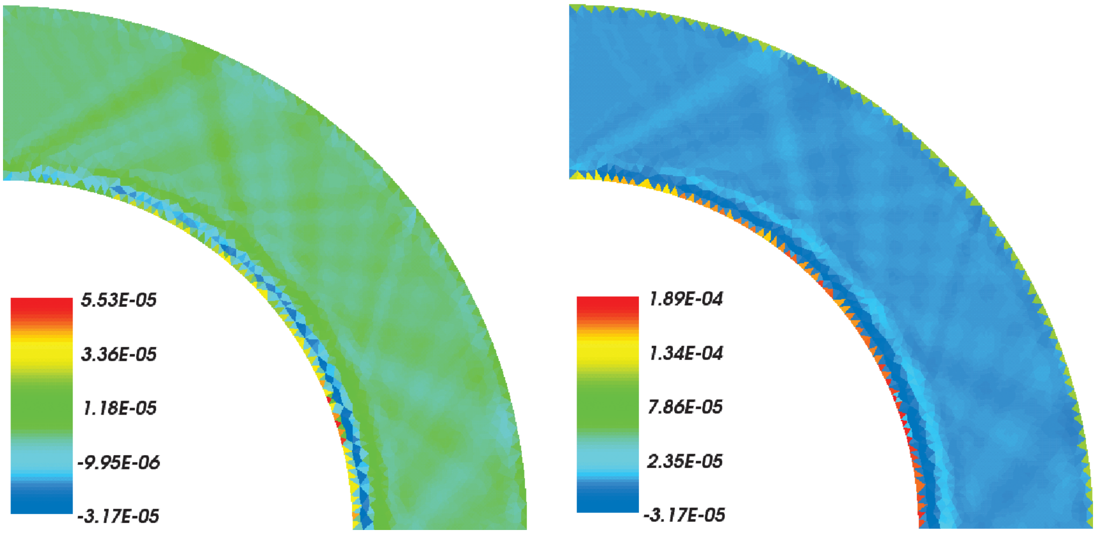
	Two dimensions				Three dimensions			
	Boundary		Interior		Boundary		Interior	
Order	3	4	3	4	3	4	3	4
Proposed quadrature	3	10	3	10	4	20	4	20
Isoparametric, for CC CV[a]	3	4	3	4	4	5	4	5
Quads/hexes per VC CV[b]	~3		~6		~10		~20	
Total points per VC CV[b]	~12		~24		~80		~160	

[a] CC CV is the cell-centered control volume. [b] VC CV is the vertex-centered control volume.



a) Third-order error, correct integration of exact solution

b) Third-order error, integration of exact solution corrupted by use of straight-sided triangles at curved boundaries



c) Fourth-order error, correct integration of exact solution

d) Fourth-order error, integration of exact solution corrupted by use of straight-sided triangles at curved boundaries

Fig. 11 Integrating over straight-sided rather than curved-sided triangles at domain boundaries can produce second-order errors, including spurious errors when comparing control volume averages of a correct high-order computed solution with averages of the exact solution over straight-sided triangles, as shown here.

are given in Appendix A. By design, these integration schemes are exact for low-degree monomials and therefore are kth-order-accurate when implemented to match the $k-1$ degree moments.

These new quadrature rules are more efficient for high-order integration over vertex-centered control volumes than applying standard quadrature rules in conjunction with isoparametric mapping when required, as Table 7 shows. The reason for this is that each vertex in the interior of a two-dimensional mesh has, on average, about six incident triangles; when using the standard median dual, each control volume contains a quadrilateral chunk from each triangle, requiring four points per quad for third- or fourth-order integration. In three dimensions, there are about 20 incident tetrahedra, each requiring eight-point integration in a hexahedron. For boundaries, these numbers are half as large on average. For cell-centered control volumes, standard quadrature rules are at least as efficient as the proposed new rules.

Accurate integration over control volumes is obviously a critical issue for problems with source terms, but it can also be a barrier to verifying high-order accuracy for problems without source terms. In particular, even if the solution is high-order-accurate for curved boundaries, comparing the control volume averages to averages of the exact solution over straight-sided triangles can show significant errors due not to the flow solver but to the verification methodology. Figure 11 and Table 8 show the effects of integration of the exact solution using straight-sided triangles at the boundaries. In the cases in which this approximation to the control volume average is used (Figs. 11b and 11d), the plots show significantly more error along the boundaries, even though the underlying computed solution is independent of the integration of the exact solution. For both third- and fourth-order schemes, the maximum error converges at a

Table 8 Measured error for correct high-order scheme with different control volume average calculations

	Measured order			
	Averaging over straight-sided triangles		Averaging over curved control volumes	
Nominal order	L_1	L_∞	L_1	L_∞
3	2.69	2.13	2.66	2.17
4	3.28	2.32	4.29	3.32

second-order rate when integration over straight-sided triangles rather than curved control volumes contaminates the comparison solution. In addition to its primary point about control volume integration, this example also highlights one of the challenges of verifying high-order accuracy: any calculations made to verify the accuracy of the solution must themselves be high-order-accurate.

V. Conclusions

This paper has described in detail a number of sometimes obscure aspects of verifying the correctness of a high-order-accurate unstructured-mesh finite volume solver. In addition to the reasonably well-known details of how to set up a least-squares problem to compute a *k*-exact reconstruction, we have touched on numerical issues in solving the least-squares problem, emphasized the importance of accounting for the difference between the control volume average and the pointwise value of the reconstruction at the control volume reference point, and provided detailed information about how to validate reconstruction code. We have also described how to compute high-order-accurate flux integrals in both two and three dimensions. Our discussion of curved-boundary treatment has explained why one must ensure that the control volume boundaries conform to the shape of the boundary to at least the truncation error of the scheme and has analytically showed the errors in boundary flux integration data that occur when this requirement is ignored. We have demonstrated the effect of using straight boundaries instead of properly curved boundaries by showing computational results for a smooth supersonic flow: the use of straight boundaries is incompatible with accuracy higher than (roughly) second-order. This effect is, not surprisingly, much more visible for a fourth-order scheme than for a third-order scheme. Finally, we have described how to integrate to high-order accuracy over control volumes with arbitrary shapes and demonstrated the effects of incorrect integration by showing the verification errors introduced by comparing a correct high-order-accurate solution with averages of the exact solution over straight-sided rather than curved control volumes. We hope that this paper will serve as a practical guide to others working to create their own high-order-accurate solvers.

Appendix A: Integration over Irregularly Shaped Control Volumes, Including Those with Curved Boundaries

We seek to match the low-degree moments of the control volume so that we can exactly integrate low-degree polynomials, because exact integration of a degree *k* polynomial implies $(k+1)$-order-accurate integration of an arbitrary smooth function. This requires that the integration points and weights satisfy

$$\sum_i w_i x_i^m y_i^n z_i^p = \frac{1}{V}\int_{\mathrm{CV}} x^m y^n z^p \,\mathrm{d}V$$

In practice, these integration points and their weights can be easily precomputed and stored. Our approach assumes the existence of the control volume moments, which is not a burden in this context, as they must be available for high-order reconstruction anyway. If the moments are not readily available, these integration rules may be less attractive in practice.

We use a coordinate system aligned with the principal axes of the control volume. That is, we use a set of coordinates (X, Y, Z) in which

$$\bar{X} = \bar{Y} = \bar{Z} = 0 \qquad \overline{XY} = \overline{XZ} = \overline{YZ} = 0$$

Without loss of generality, we can choose $\overline{X^2} \geq \overline{Y^2} \geq \overline{Z^2}$, and $\overline{X^3} > 0$ when the latter is applicable. Specification of integration rules is much simpler in this coordinate system, especially for third-order rules.

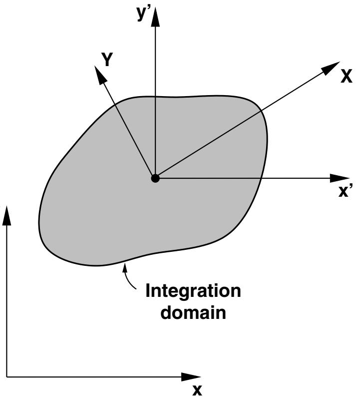


Fig. A1 Arbitrary (x, y), centroidal (x', y'), and principal (X, Y) coordinate systems for a general integration domain (shaded). The small circle is the location of the centroid of the integration domain.

I. Two Dimensions

A. *Finding the Coordinates* (X, Y)

In this case, we first create a local coordinate system (x', y') with its origin at the centroid of the integration domain $(\bar{x}, \bar{y})$, as shown in Fig. A1. Then the second moments of the domain in this translated coordinate system form a moment tensor:

$$M = \begin{bmatrix} \overline{x'^2} & \overline{x'y'} \\ \overline{x'y'} & \overline{y'^2} \end{bmatrix}$$

We can apply any of the standard approaches for finding the principal coordinates (X, Y), including using Mohr's circle and applying exact or numerical approaches to solving the 2 × 2 eigenvalue problem. As noted, we choose among the four possible orientations of the principal axes so that $\overline{X^2} \geq \overline{Y^2}$ and (for fourth order) $\overline{X^3} > 0$. The principal moments (analogous to principal stresses in solid mechanics) are found as a byproduct. The third moments can be found by replacing (x, y) with appropriate linear functions of (X, Y) (including the angle of rotation of the principal axes with respect to the (x', y') frame) and formally integrating; this process is tedious, though straightforward, and results in an expression for third moments in the (X, Y) space in terms of the original moments in the (x, y) space.

B. *Second Order*

In this case, a single point at the control volume centroid suffices, with a weight of 1.

C. *Third Order*

In the principal coordinate system, there are three nonzero moments:

$$\begin{array}{lll} \sum_i w_i = 1 & \sum_i w_i X_i^2 = \overline{X^2} & \sum_i w_i X_i = 0 \\ \sum_i w_i X_i Y_i = 0 & \sum_i w_i Y_i = 0 & \sum_i w_i Y_i^2 = \overline{Y^2} \end{array} \tag{A1}$$

We can exploit symmetry to find a particularly simple integration rule if we choose three points, as shown in Fig. A2, with $X_2 = X_3 = -X_1/2$, $Y_3 = -Y_2$, and $Y_1 = 0$. Also, we choose all weights equal to $\frac{1}{3}$. Then Eqs. (A1) reduce to

$$\tfrac{1}{2}X_1^2 = \overline{X^2} \rightarrow X_1 \equiv \tilde{X} = \sqrt{2\overline{X^2}}$$
$$\tfrac{2}{3}Y_2^2 = \overline{Y^2} \rightarrow Y_2 \equiv \tilde{Y} = \sqrt{\tfrac{3}{2}\overline{Y^2}}$$

D. *Fourth Order*

In this case, we again use the principal coordinates. The moment equations we must satisfy in this coordinate system are

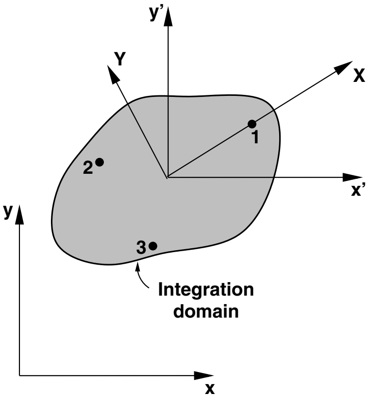


Fig. A2 Location of integration points for third-order integration in two dimensions.

$$\begin{gathered}\sum_i w_i = 1 \qquad \sum_i w_i X_i = 0 \qquad \sum_i w_i Y_i = 0 \\ \sum_i w_i X_i^2 = \overline{X^2} \qquad \sum_i w_i X_i Y_i = 0 \qquad \sum_i w_i Y_i^2 = \overline{Y^2} \\ \sum_i w_i X_i^3 = \overline{X^3} \qquad \sum_i w_i X_i^2 Y_i = \overline{X^2 Y} \qquad \sum_i w_i X_i Y_i^2 = \overline{XY^2} \\ \sum_i w_i Y_i^3 = \overline{Y^3}\end{gathered} \tag{A2}$$

Unfortunately, in this case, we cannot exploit symmetry because of the presence of the third moments. Instead, we place 10 points at the nodal locations for a cubic Lagrange triangle, as shown in Fig. A3 and Table A1. The dimensions $\tilde{X}$ and $\tilde{Y}$ are the same as in the third-order case. The integration weights are given by

$$w_i = \frac{1}{6}\begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{pmatrix} + \begin{bmatrix} 8 & 0 & 0 & 0 \\ -1 & 1 & -1 & 1 \\ -1 & -1 & -1 & -1 \\ -12 & 4 & 0 & 0 \\ 6 & -4 & 2 & 0 \\ -3 & 1 & 1 & -3 \\ -3 & -1 & 1 & 3 \\ 6 & 4 & 2 & 0 \\ -12 & -4 & 0 & 0 \\ 12 & 0 & -4 & 0 \end{bmatrix} \begin{pmatrix} \dfrac{\sqrt{3}\,\overline{X^3}}{54\,\tilde{X}^3} \\ \dfrac{\sqrt{3}\,\overline{X^2 Y}}{12\,\tilde{X}^2 \tilde{Y}} \\ \dfrac{\sqrt{3}\,\overline{XY^2}}{8\,\tilde{X}\tilde{Y}^2} \\ \dfrac{\sqrt{3}\,\overline{Y^3}}{16\,\tilde{Y}^3} \end{pmatrix}$$

Note that our choice of coordinate axes to require the $\overline{X^3} \geq 0$ ensures that point 1 will always have a positive weight; in our numerical experiments, this reduces the incidence of nonpositive integration weights so that fewer than 1% of control volumes have any negative weights in practice.

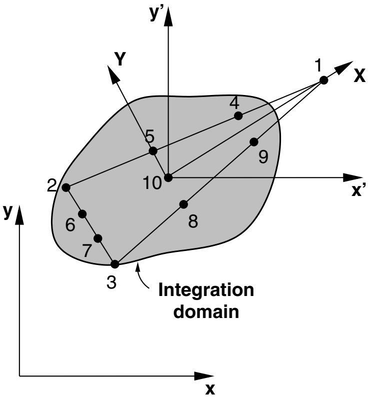


Fig. A3 Schematic of a ten-point triangle. Points 1–3 are vertices of the triangle, points 4–9 are all on edges, one-third of the way from one vertex to another, and point 10 is the triangle centroid. The dimensions $\tilde{X}$ and $\tilde{Y}$ are the same as in the third-order case.

Table A1 Location of integration points for two-dimensional fourth-order integration

i	$X_i/\tilde{X}$	$Y_i/\tilde{Y}$
1	$\sqrt{3}$	0
2	$-\frac{\sqrt{3}}{2}$	$\sqrt{3}$
3	$-\frac{\sqrt{3}}{2}$	$-\sqrt{3}$
4	$\frac{\sqrt{3}}{2}$	$\frac{\sqrt{3}}{3}$
5	0	$\frac{2\sqrt{3}}{3}$
6	$-\frac{\sqrt{3}}{2}$	$\frac{\sqrt{3}}{3}$
7	$-\frac{\sqrt{3}}{2}$	$-\frac{\sqrt{3}}{3}$
8	0	$\frac{2\sqrt{3}}{3}$
9	$\frac{\sqrt{3}}{2}$	$-\frac{\sqrt{3}}{3}$
10	0	0

Table A2 Location and weights of integration points for three-dimensional third-order integration

i	w_i	X_i	Y_i	Z_i
1	$\frac{1}{4}$	$\hat{X}$	0	0
2	$\frac{1}{4}$	$-\frac{\hat{X}}{3}$	$\hat{Y}$	0
3	$\frac{1}{4}$	$-\frac{\hat{X}}{3}$	$-\frac{\hat{Y}}{2}$	$\hat{Z}$
4	$\frac{1}{4}$	$-\frac{\hat{X}}{3}$	$-\frac{\hat{Y}}{2}$	$-\hat{Z}$

[a] Note that $\widehat{X} = \sqrt{3\overline{X^2}}$, $\hat{Y} = \sqrt{\frac{8}{3}\overline{Y^2}}$, and $\hat{Z} = \sqrt{2\overline{Z^2}}$. Despite these odd-looking definitions, if $\overline{X^2} = \overline{Y^2} = \overline{Z^2}$, the four points in this case form an equilateral tetrahedron.

II. Three Dimensions

A. *Finding the Coordinates* (X, Y, Z)

Similarly to the two-dimensional case, we first create a coordinate system (x', y', z') with its origin at the centroid of the integration domain $(\bar{x}, \bar{y}, \bar{z})$. Then the second moments of the domain in this translated coordinate system form a moment tensor:

$$M = \begin{bmatrix} \overline{x'^2} & \overline{x'y'} & \overline{x'z'} \\ \overline{x'y'} & \overline{y'^2} & \overline{y'z'} \\ \overline{x'z'} & \overline{y'z'} & \overline{z'^2} \end{bmatrix}$$

Again, we must find the principal coordinates (X, Y, Z); in this case, the simplest choice is solving the 3 × 3 eigenvalue problem numerically. As noted, we choose among the eight possible orientations of the principal axes so that $\overline{X^2} \geq \overline{Y^2} \geq \overline{Z^2}$ and (for fourth order) $\overline{X^3} > 0$. The principal moments (analogous to principal stresses in solid mechanics) are found as a byproduct. Just as in two dimensions, the third moments can be found by replacing (x, y, z) with appropriate linear functions of (X, Y, Z) and formally integrating.

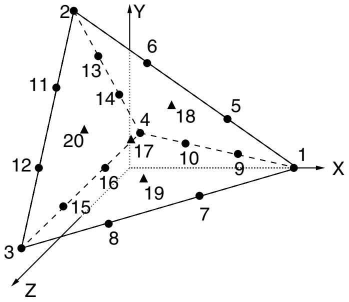


Fig. A4 Schematic of a 20-point tetrahedron. Points 1–4 are vertices of the tetrahedron, points 5–16 are all on edges, one-third of the way from one vertex to another, and points 17–20 (shown as triangles) are face centroids.

Table A3 Location of integration points for three-dimensional fourth-order integration

i	X_i	Y_i	Z_i
1	X_1	0	0
2	$-\frac{X_1}{3}$	Y_2	0
3	$-\frac{X_1}{3}$	$-\frac{Y_2}{2}$	Z_3
4	$-\frac{X_1}{3}$	$-\frac{Y_2}{2}$	$-Z_3$
5	$\frac{5X_1}{9}$	$\frac{Y_2}{3}$	0
6	$\frac{X_1}{9}$	$\frac{2Y_2}{3}$	0
7	$\frac{5X_1}{9}$	$-\frac{Y_2}{6}$	$\frac{Z_3}{3}$
8	$\frac{X_1}{9}$	$-\frac{Y_2}{3}$	$\frac{2Z_3}{3}$
9	$\frac{5X_1}{9}$	$-\frac{Y_2}{6}$	$-\frac{Z_3}{3}$
10	$\frac{X_1}{9}$	$-\frac{Y_2}{3}$	$-\frac{2Z_3}{3}$
11	$-\frac{X_1}{3}$	$\frac{Y_2}{2}$	$\frac{Z_3}{3}$
12	$-\frac{X_1}{3}$	0	$\frac{2Z_3}{3}$
13	$-\frac{X_1}{3}$	$\frac{Y_2}{2}$	$-\frac{Z_3}{3}$
14	$-\frac{X_1}{3}$	0	$-\frac{2Z_3}{3}$
15	$-\frac{X_1}{3}$	$-\frac{Y_2}{2}$	$\frac{Z_3}{3}$
16	$-\frac{X_1}{3}$	$-\frac{Y_2}{2}$	$-\frac{Z_3}{3}$
17	$\frac{X_1}{9}$	$\frac{Y_2}{6}$	$\frac{Z_3}{3}$
18	$\frac{X_1}{9}$	$\frac{Y_2}{6}$	$-\frac{Z_3}{3}$
19	$\frac{X_1}{9}$	$-\frac{Y_2}{3}$	0
20	$-\frac{X_1}{3}$	0	0

[a] Note that $X_1 \equiv 9\sqrt{\overline{X^2}/5}$, $Y_2 = 6\sqrt{2\overline{Y^2}/5}$, and $Z_3 = 3\sqrt{6\overline{Z^2}/5}$.

B. Second Order

In this case, a single point at the control volume centroid suffices, with a weight of 1.

C. Third Order

Analogously to the two-dimensional case, we can exploit the symmetry exposed by use of the principal coordinates (X, Y, Z); in this coordinate system, the moment equations reduce to

$$\begin{gathered}
\sum_i w_i = 1 \qquad \sum_i w_i X_i = 0 \qquad \sum_i w_i Y_i = 0 \qquad \sum_i w_i Z_i = 0 \\
\sum_i w_i X_i^2 = \overline{X^2} \qquad \sum_i w_i X_i Y_i = 0 \qquad \sum_i w_i Y_i^2 = \overline{Y^2} \\
\sum_i w_i Y_i Z_i = 0 \qquad \sum_i w_i Z_i^2 = \overline{Z^2} \qquad \sum_i w_i Z_i X_i = 0
\end{gathered} \tag{A3}$$

As in two dimensions, placing integration points at the corners of a symmetric coordinate-aligned tetrahedron makes it possible to satisfy Eqs. (A3) with four equally weighted points, shown in Table A2.

D. Fourth Order

In this case, we again use the principal coordinates. The moment equations we must satisfy in this coordinate system are

$$\begin{gathered}
\sum_i w_i = 1 \qquad \sum_i w_i X_i = 0 \qquad \sum_i w_i Y_i = 0 \qquad \sum_i w_i Z_i = 0 \\
\sum_i w_i X_i^2 = \overline{X^2} \qquad \sum_i w_i Y_i^2 = \overline{Y^2} \qquad \sum_i w_i Z_i^2 = \overline{Z^2} \\
\sum_i w_i X_i Y_i = 0 \qquad \sum_i w_i Y_i Z_i = 0 \qquad \sum_i w_i Z_i X_i = 0 \\
\sum_i w_i X_i^3 = \overline{X^3} \qquad \sum_i w_i X_i^2 Y_i = \overline{X^2 Y} \qquad \sum_i w_i X_i Y_i^2 = \overline{XY^2} \\
\sum_i w_i Y_i^3 = \overline{Y^3} \qquad \sum_i w_i Y_i^2 Z_i = \overline{Y^2 Z} \qquad \sum_i w_i Y_i Z_i^2 = \overline{YZ^2} \\
\sum_i w_i Z_i^3 = \overline{Z^3} \qquad \sum_i w_i Z_i^2 X_i = \overline{Z^2 X} \qquad \sum_i w_i X_i Z_i^2 = \overline{XZ^2} \\
\sum_i w_i X_i Y_i Z_i = \overline{XYZ}
\end{gathered} \tag{A4}$$

Unfortunately, in this case, we cannot exploit symmetry because of the presence of the third moments. Instead, we place 20 points at the nodal locations for a cubic Lagrange tetrahedron, as shown in Fig. A4 and Table A3. The weights are given by

$$w_i = \frac{1}{48}\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 9 \\ 9 \\ 9 \\ 9 \end{pmatrix} + \begin{bmatrix}
27 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 2 & -4 & 8 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & -1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 \\
-1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
-27 & 18 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
9 & -12 & 12 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-27 & -9 & 0 & 0 & 0 & 0 & 0 & 0 & 9 & 0 \\
9 & 6 & 3 & 0 & 0 & 0 & 0 & 3 & -6 & -3 \\
-27 & -9 & 0 & 0 & 0 & 0 & 0 & 0 & -9 & 0 \\
9 & 6 & 3 & 0 & 0 & 0 & 0 & 3 & 6 & 3 \\
-3 & 3 & 0 & -12 & 4 & 0 & 0 & 0 & 1 & -2 \\
-3 & 0 & 3 & 6 & -4 & 2 & 0 & -1 & 2 & -1 \\
-3 & 3 & 0 & -12 & -4 & 0 & 0 & 0 & -1 & 2 \\
-3 & 0 & 3 & 6 & 4 & 2 & 0 & -1 & -2 & 1 \\
-3 & -3 & -3 & -3 & 1 & 1 & -3 & 1 & 1 & 1 \\
-3 & -3 & -3 & -3 & -1 & 1 & 3 & 1 & -1 & -1 \\
18 & -6 & -12 & 0 & 0 & 0 & 0 & 0 & -6 & 6 \\
18 & -6 & -12 & 0 & 0 & 0 & 0 & 0 & 6 & -6 \\
18 & 12 & 6 & 0 & 0 & 0 & 0 & -6 & 0 & 0 \\
-6 & 0 & 6 & 12 & 0 & -4 & 0 & 2 & 0 & 0
\end{bmatrix}\begin{pmatrix}
\frac{9}{128}\frac{\overline{X^3}}{X_1^3} \\
\frac{9}{32}\frac{\overline{X^2 Y}}{X_1^2 Y_2} \\
\frac{3}{8}\frac{\overline{XY^2}}{X_1 Y_2^2} \\
\frac{1}{6}\frac{\overline{Y^3}}{Y_2^3} \\
\frac{3}{4}\frac{\overline{Y^2 Z}}{Y_2^2 Z_3} \\
\frac{9}{8}\frac{\overline{YZ^2}}{Y_2 Z_3^2} \\
\frac{9}{16}\frac{\overline{Z^3}}{Z_3^3} \\
\frac{27}{32}\frac{\overline{Z^2 X}}{X_1 Z_3^2} \\
\frac{27}{64}\frac{\overline{ZX^2}}{X_1^2 Z_3} \\
\frac{9}{8}\frac{\overline{XYZ}}{X_1 Y_2 Z_3}
\end{pmatrix} \tag{A5}$$

where again $X_1 \equiv 9\sqrt{\overline{X^2}/5}$, $Y_2 = 6\sqrt{2\overline{Y^2}/5}$, and $Z_3 = 3\sqrt{6\overline{Z^2}/5}$. Each of these quantities is larger than their counterparts for the third-order case by a factor of $(3\sqrt{3})/\sqrt{5}$, because this scaling makes the constant part of the weights nonnegative for all points.

References

[1] Rogers, S. E., Kwak, D., and Kiris, C., "Steady and Unsteady Solutions of the Incompressible Navier-Stokes Equations," *AIAA Journal*, Vol. 29, No. 4, Apr. 1991, pp. 603–610.
doi:10.2514/3.10627

[2] Zingg, D., De Rango, S., Nemec, M., and Pulliam, T., "Comparison of Several Spatial Discretizations for the Navier-Stokes Equations," *Journal of Computational Physics*, Vol. 160, No. 2, 2000, pp. 683–704.
doi:10.1006/jcph.2000.6482

[3] De Rango, S., and Zingg, D. W., "Higher-Order Spatial Discretization for Turbulent Aerodynamic Computations," *AIAA Journal*, Vol. 39, No. 7, July 2001, pp. 1296–1304.
doi:10.2514/2.1472

[4] Barth, T. J., and Frederickson, P. O., "Higher Order Solution of the Euler Equations on Unstructured Grids Using Quadratic Reconstruction," AIAA Paper 90-0013, Jan. 1990.

[5] Barth, T. J., "Recent Developments in High Order K-Exact Reconstruction on Unstructured Meshes," AIAA Paper 93-0668, Jan. 1993.

[6] Ollivier-Gooch, C. F., "High-Order ENO Schemes for Unstructured Meshes Based on Least-Squares Reconstruction," AIAA Paper 97-0540, Jan. 1997.

[7] Delanaye, M., and Essers, J. A., "Quadratic-Reconstruction Finite Volume Scheme for Compressible Flows on Unstructured Adaptive Grids," *AIAA Journal*, Vol. 35, No. 4, Apr. 1997, pp. 631–639.
doi:10.2514/2.183

[8] Geuzaine, P., Delanaye, M., and Essers, J.-A., "Computation of High Reynolds Number Flows with an Implicit Quadratic Reconstruction Scheme on Unstructured Grids," *Proceedings of the Thirteenth AIAA Computational Fluid Dynamics Conference*, AIAA, Reston, VA, 1997, pp. 610–619.

[9] Abgrall, R., "On Essentially Non-Oscillatory Schemes on Unstructured Meshes: Analysis and Implementation," *Journal of Computational Physics*, Vol. 114, No. 1, 1994, pp. 45–58.
doi:10.1006/jcph.1994.1148

[10] Friedrich, O., "Weighted Essentially Non-Oscillatory Schemes for the Interpolation of Mean Values on Unstructured Grids," *Journal of Computational Physics*, Vol. 144, No. 1, July 1998, pp. 194–212.
doi:10.1006/jcph.1998.5988

[11] Hu, C. Q., and Shu, C. W., "Weighted Essentially Nonoscillatory Schemes on Triangular Meshes," *Journal of Computational Physics*, Vol. 150, No. 1, Mar. 1999, pp. 97–127.
doi:10.1006/jcph.1998.6165

[12] Michalak, C., and Ollivier-Gooch, C., "Matrix-Explicit GMRES for a Higher-Order-Accurate Inviscid Compressible Flow Solver," 18th AIAA Computational Fluid Dynamics Conference, AIAA Paper 2007-3943, 2007.

[13] Nejat, A., and Ollivier-Gooch, C., "A High-Order-Accurate Unstructured Finite Volume Newton-Krylov Algorithm for Inviscid Compressible Flows," *Journal of Computational Physics*, Vol. 227, No. 4, 2008, pp. 2582–2609.
doi:10.1016/j.jcp.2007.11.011

[14] Nejat, A., and Ollivier-Gooch, C., "Effect of Discretization Order on Preconditioning and Convergence of a High-Order Unstructured Newton-GMRES Solver for the Euler Equations," *Journal of Computational Physics*, Vol. 227, No. 4, 2008, pp. 2366–2386.
doi:10.1016/j.jcp.2007.10.024

[15] Michalak, C., and Ollivier-Gooch, C., "Globalized Matrix-Explicit Newton-GMRES for the High-Order Accurate Solution of the Euler Equations," *Computers and Fluids* (submitted for publication).

[16] Michalak, C., and Ollivier-Gooch, C., "Differentiability of Slope Limiters on Unstructured Grids," *Proceedings of the Fourteenth Annual Conference of the Computational Fluid Dynamics Society of Canada*, CFD Society of Canada, 2006.

[17] Michalak, C., and Ollivier-Gooch, C., "Limiters for Unstructured Higher-Order-Accurate Solutions of the Euler Equations," 46th AIAA Aerospace Sciences Meeting, AIAA Paper 2008-0776, 2008.

[18] Michalak, C., and Ollivier-Gooch, C., "Limiters for Unstructured Higher-Order-Accurate Solutions of the Euler Equations," *Journal of Computational Physics* (submitted for publication).

[19] Ollivier-Gooch, C. F., "Quasi-ENO Schemes for Unstructured Meshes Based on Unlimited Data-Dependent Least-Squares Reconstruction," *Journal of Computational Physics*, Vol. 133, No. 1, 1997, pp. 6–17.
doi:10.1006/jcph.1996.5584

[20] Ollivier-Gooch, C. F., and Van Altena, M., "A High-Order-Accurate Unstructured Mesh Finite Volume Scheme for the Advection-Diffusion Equation," *Journal of Computational Physics*, Vol. 181, No. 2, 2002, pp. 729–752.
doi:10.1006/jcph.2002.7159

[21] Nejat, A., and Ollivier-Gooch, C., "A High-Order-Accurate Unstructured GMRES Solver for the Euler Equations," *Computational Fluid Dynamics 2004: Proceedings of the Third International Conference on CFD*, edited by C. Groth and D. W. Zingg, Springer–Verlag, Berlin, 2006, pp. 435–440.

[22] Michalak, C., and Ollivier-Gooch, C., "Parallelization of a High-Order-Accurate Unstructured Mesh Finite Volume Solver," *Proceedings of the Twelfth Annual Conference of the Computational Fluid Dynamics Society of Canada*, CFD Society of Canada, 2004.

[23] Nejat, A., and Ollivier-Gooch, C., "A High-order-accurate Unstructured GMRES Algorithm for Inviscid Compressible Flows," 17th AIAA Computational Fluid Dynamics Conference, AIAA Paper 2005-5341, June 2005.

[24] Mavriplis, D. J., "Revisiting the Least-Squares Procedure for Gradient Reconstruction on Unstructured Meshes," 16th AIAA Computational Fluid Dynamics Conference, AIAA Paper 2003-3986, 2003.

[25] Golub, G. H., and Loan, C. F. V., *Matrix Computations*, Johns Hopkins Univ. Press, Baltimore, MD, 1983.

[26] Stroud, A. H., and Secrest, D., *Gaussian Quadrature Formulas*, Prentice–Hall, Englewood Cliffs, NJ, 1966.

[27] Roache, P. J., "Code Verification by the Method of Manufactured Solutions," *Journal of Fluids Engineering*, Vol. 124, No. 1, Mar. 2002, pp. 4–10.
doi:10.1115/1.1436090

[28] Diskin, B., and Thomas, J., "Accuracy Analysis for Mixed-Element Finite Volume Discretization Schemes," National Inst. of Aerospace TR 2007-08, Hampton, VA, 2007.

[29] Thomas, J. L., Diskin, B., and Rumsey, C., "Towards Verification of Unstructured-Grid Solvers," 46th AIAA Aerospace Sciences Meeting, AIAA Paper 2008-666, 2008.

[30] Aftosmis, M. J., Gaitonde, D., and Tavares, T. S., "On the Accuracy, Stability, and Monotonicity of Various Reconstruction Algorithms for Unstructured Meshes," AIAA Paper 94-0415, Jan. 1994.

[31] Nejat, A., *A Higher-Order-Accurate Unstructured Finite Volume Newton-Krylov Algorithm for Inviscid Compressible Flows*, Ph.D. Thesis, Univ. of British Columbia, Dept. of Mechanical Engineering, Vancouver, BC, Canada, 2007.

Z. Wang
Associate Editor