Preprint accepted in Journal of Computational Physics. https://doi.org/10.1016/j.jcp.2017.04.075

Accuracy-Preserving Source Term Quadrature for Third-Order Edge-Based Discretization

Hiroaki Nishikawa*and Yi Liu†
National Institute of Aerospace, Hampton, VA 23666

May 12, 2017

Abstract

In this paper, we derive a family of source term quadrature formulas for preserving third-order accuracy of the node-centered edge-based discretization for conservation laws with source terms on arbitrary simplex grids. A three-parameter family of source term quadrature formulas is derived, and as a subset, a one-parameter family of economical formulas is identified that does not require second derivatives of the source term. Among the economical formulas, a unique formula is then derived that does not require gradients of the source term at neighbor nodes, thus leading to a significantly smaller discretization stencil for source terms. All the formulas derived in this paper do not require a boundary closure, and therefore can be directly applied at boundary nodes. Numerical results are presented to demonstrate third-order accuracy at interior and boundary nodes for one-dimensional grids and linear triangular/tetrahedral grids over straight and curved geometries.

1 Introduction

Node-centered edge-based discretization methods are widely used in practical Computational Fluid Dynamics (CFD) solvers [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]. The approach has recently been shown to achieve third-order accuracy for first-order systems of conservation laws on arbitrary triangular [13, 14, 15] and tetrahedral grids [16, 17, 18] with quadratic least-squares (LSQ) gradients and linearly-extrapolated fluxes. The third-order edge-based discretization scheme is extremely efficient in that the spatial residual can be computed over a single loop over edges with a single numerical flux evaluation per edge. Another striking feature of this scheme is the ability to deliver third-order accurate solutions on linear triangular/tetrahedral grids even for curved geometries [16, 19]. Therefore, the third-order edge-based method does not require generating high-order grids. Although high-order surface normal vectors are still needed at boundary nodes to provide formal third-order accuracy for some boundary conditions (e.g., slip wall), the scheme requires only normal vectors at nodes on solid bodies. Practical three-dimensional computations indicate that third-order accuracy is observed without high-order normals for some realistic geometries [16]. A high-order surface representation is required for evaluating integral quantities such as drag, but it is a matter of post-processing and thus does not affect the solution algorithm and the linear computational grid. Because of these attractive low-cost features, the third-order edge-based method has become a subject of great interest to CFD developers and practitioners; see Refs.[16, 17, 18, 20, 21, 22] for recent developments.

The third-order edge-based discretization scheme relies on a special error elimination mechanism. On a regular grid, the truncation error of this scheme has a leading second-order error term proportional to the derivatives of the target equation, which thus vanishes and leads to third-order accuracy. This is a well-known mechanism, often called the residual property, common to low-cost high-order methods such as the residual-compact method [23] and the residual-distribution method [24]. In these methods, any additional term added to the target equation must be carefully discretized in order to preserve the residual property and guarantee the design accuracy. This paper focuses on source terms, which arise, for example, in reaction equations, the method of manufactured solutions, and implicit time-stepping schemes [25]. General compatible discretization formulas for source terms are derived and demonstrated by numerical experiments.

*Associate Research Fellow (hiro@nianet.org), National Institute of Aerospace, 100 Exploration Way, Hampton, VA 23666 USA
†Senior Research Scientist (yi.liu@nianet.org), National Institute of Aerospace, 100 Exploration Way, Hampton, VA 23666 USA

In node-centered edge-based methods, source terms are typically evaluated directly at nodes. This placement of the source terms does not generate any truncation error in the Taylor expansion of the residual. To achieve third-order accuracy, the source term discretization must generate a second-order truncation error in the form compatible with that of other terms. Two techniques are currently available to meet the compatibility condition. One is an extended Galerkin source discretization formula [25, 26], and the other is a divergence formulation [27]. In the latter, a source term is cast in the form of a conservation law and thus allows a straightforward source term discretization by the third-order edge-based discretization. The compatibility condition is automatically met since all terms are discretized by the same algorithm. Both approaches have been successfully employed in the construction of third-order edge-based schemes. However, a major drawback of these approaches is the requirement for computations and storage of second derivatives of the source term. The additional cost associated with the source term discretizations can be substantial in three dimensions and even more so in unsteady computations where second derivatives need to be stored for all variables at all physical times required to approximate the physical time derivative. For example, in the three-dimensional Euler or Navier-Stokes equations, even with a symmetry assumption, six second derivatives are required for each of five variables at four time-levels with the third-order backward difference formula. In order to generate a truly efficient third-order edge-based scheme for practical three-dimensional unsteady computations, therefore, it is desirable to eliminate the second derivatives from the algorithm. This paper presents a new approach to deriving source term quadrature formulas that do not require second derivatives at all.

A detailed analysis on compatible source discretizations in two and three dimensions has not been reported in the literature. In Refs.[25, 26, 28], a one-dimensional truncation analysis is given, but it discusses mainly the elimination of first-order error terms and does not discuss the compatible discretization on regular grids in details for the third-order edge-based scheme considered here. In Ref.[27], the compatibility issue is discussed, but discretization details are not provided because the issue is resolved in the differential equation level to make the compatible discretization trivial. In this paper, we provide a detailed account for the compatible source term discretizations, and derive general formulas, demonstrating that previously reported formulas are only a small subset of a general three-parameter family of formulas. In particular, we derive a special subset that does not require second derivatives of source terms. These formulas completely eliminate the additional expense of computations and storage of second derivatives, and dramatically reduce the cost of the third-order edge-based scheme. Furthermore, a unique formula is identified, which does not require any derivative at the neighbor nodes, leading to a source term discretization with a significantly smaller stencil.

The paper is organized as follows. In Section 2, the third-order edge-based discretization is described for a general conservation law. In Section 3, the requirements for compatible source discretizations are discussed. In Section 4, previously reported formulas are reviewed. In Sections 5, new source quadrature formulas are derived. In Section 6, source discretizations at boundary nodes are discussed. In Section 7, numerical results are presented. Finally, the paper concludes with remarks.

2 Third-Order Edge-Based Discretization

Consider a scalar conservation law with a source term:

$$\mathrm{div}\mathbf{f} = s, \tag{2.1}$$

where $\mathbf{f}$ is a flux vector function of the solution u, and s is a source term. It is a steady conservation law, but directly relevant to unsteady equations, where physical time derivatives can be discretized in time and treated as a source term as is often done for implicit time-stepping schemes (see, e.g., Ref.[25]). Note also that all discussions are applicable to each component of a more general system of equations. For systems of equations, the flux $\mathbf{f}$ is a tensor and the source term is a vector $\mathbf{s}$.

The target conservation law is discretized on a triangular/tetrahedral grid by the node-centered edge-based method, where a dual control volume is defined around each node by connecting the edge midpoints and the element centroids (and the centroids of the element faces in three dimensions). The discretization at a node j is given by

$$0 = -\sum_{k\in\{k_j\}} \phi_{jk}(\mathbf{n}_{jk}) + \int_{V_j} s\, dV, \tag{2.2}$$

where V_j is the measure of the dual control volume around the node j, $\{k_j\}$ is a set of neighbor nodes of the node j, ϕ_{jk} is a numerical flux, and $\mathbf{n}_{jk}$ is the directed area vector, which is a sum of the directed-areas corresponding

to the dual faces associated with all elements sharing the edge $[j,k]$. In two dimensions, the directed area vector is defined as a sum of two face normal vectors $\mathbf{n}_{jk}^{\ell}$ and $\mathbf{n}_{jk}^{r}$ as illustrated in Figure 1. In three dimensions, it is defined as a sum of the surface normal vectors of the triangular dual faces as illustrated in Figure 2. The discretization of the source term, which is the main subject of the paper, will be discussed later. The numerical flux is computed at the edge midpoint as

$$\phi_{jk}(\mathbf{n}_{jk}) \quad = \quad \frac{1}{2}\left[\mathbf{f}_R + \mathbf{f}_L\right]\cdot\mathbf{n}_{jk} - \frac{1}{2}\left|a_n\right|\left(u_R - u_L\right)\left|\mathbf{n}_{jk}\right|, \quad a_n = \left(\frac{\partial \mathbf{f}}{\partial u}\right)\cdot\frac{\mathbf{n}_{jk}}{|\mathbf{n}_{jk}|}, \tag{2.3}$$

where the subscripts L and R indicate values at the "left" and "right" sides of the edge midpoint, which are linearly reconstructed from values at nodes j and k, respectively:

$$\begin{aligned} \mathbf{f}_L &= \mathbf{f}_j + \frac{1}{2}\hat{\nabla}\mathbf{f}_j\cdot\Delta\mathbf{x}_{jk} = \mathbf{f}_j + \frac{1}{2}\left(\frac{\partial \mathbf{f}}{\partial u}\right)_j\hat{\nabla}u_j\cdot\Delta\mathbf{x}_{jk}, \\ \mathbf{f}_R &= \mathbf{f}_k - \frac{1}{2}\hat{\nabla}\mathbf{f}_k\cdot\Delta\mathbf{x}_{jk} = \mathbf{f}_k - \frac{1}{2}\left(\frac{\partial \mathbf{f}}{\partial u}\right)_k\hat{\nabla}u_k\cdot\Delta\mathbf{x}_{jk}, \end{aligned} \tag{2.4}$$

$$u_L = u_j + \frac{1}{2}\hat{\nabla}u_j\cdot\Delta\mathbf{x}_{jk}, \qquad u_R = u_k - \frac{1}{2}\hat{\nabla}u_k\cdot\Delta\mathbf{x}_{jk}, \tag{2.5}$$

where the hat indicates that the gradients are evaluated numerically, and $\Delta\mathbf{x}_{jk}$ denotes the edge vector pointing from the node j to the node k. The nodal gradients are computed by a quadratic LSQ fit, which is implemented in two steps as described in Ref.[29]. The LSQ stencils extend to the neighbors of the neighbors, but each step involves only the edge neighbors and therefore it is simple to implement in a parallel code. On regular grids, on the other hand, we use only the edge neighbors (six neighbors in two dimensions, and fourteen neighbors in three dimensions), and thus the LSQ stencil is compact, at least at the interior nodes. In any case, the nodal gradients are computed in the form: e.g., in three dimensions,

$$\hat{\nabla}u_j = \begin{bmatrix} \hat{\partial}_x u_j \\ \hat{\partial}_y u_j \\ \hat{\partial}_z u_j \end{bmatrix} = \sum_{k\in\{k_j^{LSQ}\}} \begin{bmatrix} c_{jk}^x \\ c_{jk}^y \\ c_{jk}^z \end{bmatrix} (u_k - u_j), \tag{2.6}$$

where $(c_{jk}^x, c_{jk}^y, c_{jk}^z)$ are LSQ coefficients (i.e., coefficients of $u_k - u_j$ in the solution of a quadratic LSQ problem), and $\{k_j^{LSQ}\}$ is a set of neighbors involved in the LSQ fit. See Ref.[29] for details. Note that the edge-based discretization does not require computations nor storage for second derivatives of the fluxes and solutions, and therefore only the LSQ coefficients relevant to the gradient need to be stored [29].

The edge-based discretization described above is known to be third-order accurate when $s = 0$ on arbitrary triangular and tetrahedral grids [13, 14, 15]. Third-order accuracy can be achieved by other numerical fluxes with different dissipation coefficients because the dissipation term does not contribute to the leading truncation error: it cancels out over a regular grid [30] (see Appendix A), and vanishes for quadratic solutions [19] on irregular grids. Note that the second-order edge-based scheme, which uses linear LSQ gradients and no flux extrapolation, is second-order accurate only on such simplex-element grids: it is formally first-order accurate on other types of elements unless certain regularity conditions are satisfied (see Ref.[31] and Appendix B in Ref.[32]). The third-order edge-based scheme is highly economical because it requires only a single numerical flux per edge in both two and three dimensions; see discussions in Ref.[16]. To preserve third-order accuracy for $s \neq 0$, the source term integral needs to be discretized carefully in a compatible manner, which is discussed in the next section.

3 Compatible Source Discretization for Third-Order Accuracy

As discussed in Refs.[19, 29], in the case $s = 0$, the third-order edge-based scheme has the second order truncation error on arbitrary triangular/tetrahedral grids: substitute a smooth exact solution into the residual and expand it to yield

$$\frac{1}{V_j}\sum_{k\in\{k_j\}}\phi_{jk}(\mathbf{n}_{jk}) \quad = \quad \operatorname{div}\mathbf{f}_j + Ch^2, \tag{3.1}$$

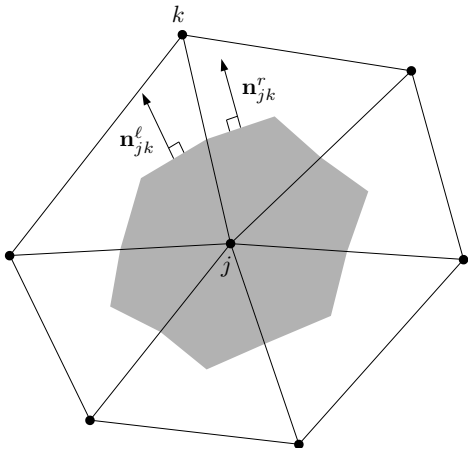


Figure 1: Dual control volume faces on a triangular grid.

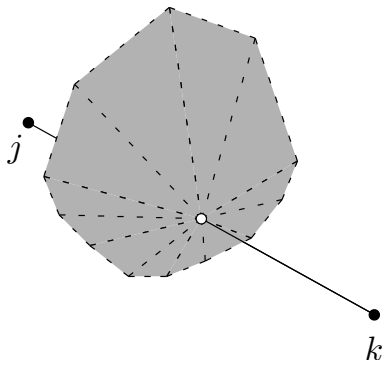


Figure 2: Dual control volume faces on a tetrahedral grid.

where $\operatorname{div}\mathbf{f}_j = 0$ for the exact solution, C is a constant depending on the geometry and third-derivatives of the flux, and h is a representative mesh spacing. For irregular grids, the truncation error is one order lower than the discretization error for a first-order system of conservation laws [31, 33], and therefore the second-order truncation error in Equation (3.1) indicates that the edge-based scheme can potentially achieve third-order discretization error. However, on regular grids, where the stencil composed of edge-connected neighbors is identical at all interior nodes, the truncation error order matches the discretization error order; therefore, the second-order truncation error must vanish. It vanishes by a special mechanism. In two dimensions, the truncation error can be factored on a regular triangular grid as (see Appendix A for the derivation)

$$\frac{1}{V_j}\sum_{k\in\{k_j\}}\phi_{jk}(\mathbf{n}_{jk}) \quad = \quad \operatorname{div}\mathbf{f}_j - \frac{1}{24V_j}\left[Q_{xx}+Q_{yy}+Q_{xy}\right](\operatorname{div}\mathbf{f})_j + O(h^3), \tag{3.2}$$

where Q_{xx}, Q_{yy}, and Q_{xy} are second-derivative operators defined by

$$Q_{xx} = \sum_{k\in\{k_j\}} n_x\Delta x^3\partial_{xx}, \quad Q_{yy} = \sum_{k\in\{k_j\}} n_y\Delta y^3\partial_{yy}, \quad Q_{xy} = \sum_{k\in\{k_j\}} 3n_x\Delta x^2\Delta y\,\partial_{xy}. \tag{3.3}$$

Note that we have dropped subscripts in the components: $\mathbf{n}_{jk} = (n_x, n_y)$ and $\Delta\mathbf{x}_{jk} = (\Delta x, \Delta y)$. For example, on a triangular grid composed of right isosceles triangles as in Figure 3, the truncation error reduces to

$$\frac{1}{V_j}\sum_{k\in\{k_j\}}\phi_{jk}(\mathbf{n}_{jk}) \quad = \quad \operatorname{div}\mathbf{f}_j - \frac{h^2}{12}\left[\partial_{xx}+\partial_{yy}+\partial_{xy}\right](\operatorname{div}\mathbf{f})_j + O(h^3), \tag{3.4}$$

where h corresponds to the length of the two equal sides of the right isosceles triangles (see Refs.[27, 29, 30]). As can be seen clearly, the second-order error terms are all proportional to the derivatives of the target equation. Therefore, they vanish for exact solutions that satisfy $\operatorname{div}\mathbf{f} = 0$. As a result, the leading truncation error becomes third-order:

$$\frac{1}{V_j}\sum_{k\in\{k_j\}}\phi_{jk}(\mathbf{n}_{jk}) \quad = \quad O(h^3). \tag{3.5}$$

This is the mechanism by which the edge-based scheme achieves third-order accuracy following an algorithm similar to a second-order algorithm. Similarly, in three dimensions, the second-order truncation error can be factored on a regular tetrahedral grid as (see Appendix A for the derivation)

$$\frac{1}{V_j}\sum_{k\in\{k_j\}}\phi_{jk}(\mathbf{n}_{jk}) = \operatorname{div}\mathbf{f}_j - \frac{1}{24V_j}\left[Q_{xx}+Q_{yy}+Q_{zz}+Q_{xy}+Q_{yz}+Q_{zx}\right](\operatorname{div}\mathbf{f})_j + O(h^3), \tag{3.6}$$

where

$$Q_{xx} = \sum_{k\in\{k_j\}} n_x \Delta x^3 \partial_{xx}, \quad Q_{yy} = \sum_{k\in\{k_j\}} n_y \Delta y^3 \partial_{yy}, \quad Q_{zz} = \sum_{k\in\{k_j\}} n_z \Delta z^3 \partial_{zz}, \tag{3.7}$$

$$Q_{xy} = \sum_{k\in\{k_j\}} 6 n_z \Delta x \Delta y \Delta z\, \partial_{xy}, \quad Q_{yz} = \sum_{k\in\{k_j\}} 6 n_x \Delta x \Delta y \Delta z\, \partial_{yz}, \quad Q_{zx} = \sum_{k\in\{k_j\}} 6 n_y \Delta x \Delta y \Delta z \partial_{zx}, \tag{3.8}$$

where $\mathbf{n}_{jk} = (n_x, n_y, n_z)$ and $\Delta\mathbf{x}_{jk} = (\Delta x, \Delta y, \Delta z)$. For example, on a regular tetrahedral grid constructed by dividing a hexahedral element into six tetrahedra [34] (see Figure 4), the truncation error reduces to

$$\frac{1}{V_j} \sum_{k\in\{k_j\}} \phi_{jk}(\mathbf{n}_{jk}) = \operatorname{div} \mathbf{f}_j - \frac{h^2}{12} \left[\partial_{xx} + \partial_{yy} + \partial_{zz} - \partial_{xy} - \partial_{yz} + \partial_{zx}\right] (\operatorname{div} \mathbf{f})_j + O(h^3). \tag{3.9}$$

Again, the second-order error terms vanish for exact solutions that satisfy $\operatorname{div} \mathbf{f} = 0$, thus leading to a third-order truncation error.

In the case $s \neq 0$, the source term discretization must have a leading second-order truncation error on arbitrary grids, and to achieve a similar error cancellation on regular grids, the second-order truncation term must have a specific form that ensures

$$\frac{1}{V_j} \sum_{k\in\{k_j\}} \phi_{jk}(\mathbf{n}_{jk}) - \frac{1}{V_j} \int_{V_j} s\, dV \quad = \quad \operatorname{div} \mathbf{f}_j - s_j - \frac{1}{24 V_j} \left[Q_{xx} + Q_{yy} + Q_{xy}\right] (\operatorname{div} \mathbf{f}_j - s_j) + O(h^3), \tag{3.10}$$

in two dimensions, and

$$\begin{aligned} &\frac{1}{V_j} \sum_{k\in\{k_j\}} \phi_{jk}(\mathbf{n}_{jk}) - \frac{1}{V_j} \int_{V_j} s\, dV \\ &= \quad \operatorname{div} \mathbf{f}_j - s_j - \frac{1}{24 V_j} \left[Q_{xx} + Q_{yy} + Q_{zz} + Q_{xy} + Q_{yz} + Q_{zx}\right] (\operatorname{div} \mathbf{f}_j - s_j) + O(h^3), \end{aligned} \tag{3.11}$$

in three dimensions, so that the second-order error terms vanish for exact solutions that satisfy $\operatorname{div} \mathbf{f}_j - s_j = 0$. This is the compatibility condition that has to be satisfied to achieve third-order accuracy for equations with source terms. It is known [27, 28] that the point quadrature, which is typically used in second-order edge-based schemes,

$$\frac{1}{V_j} \int_{V_j} s\, dV = s_j, \tag{3.12}$$

does not meet the compatibility condition. This is because it does not generate any error term, e.g., in two dimensions,

$$\frac{1}{V_j} \sum_{k\in\{k_j\}} \phi_{jk}(\mathbf{n}_{jk}) - \frac{1}{V_j} \int_{V_j} s\, dV \quad = \quad \operatorname{div} \mathbf{f}_j - s_j - \frac{1}{24 V_j} \left[Q_{xx} + Q_{yy} + Q_{xy}\right] (\operatorname{div} \mathbf{f}_j) + O(h^3), \tag{3.13}$$

and therefore the second-order error cannot vanish for $\operatorname{div} \mathbf{f} - s = 0$.

It follows that third-order accuracy requires two conditions: (1) second-order leading truncation error on irregular grids, and (2) the compatibility condition on regular grids. It is important to note that the elimination of the first-order truncation error is not sufficient to achieve third-order discretization errors.

4 Previously Reported Formulas

There exist two approaches to deriving compatible source discretizations. One approach is based on a truncation error analysis. In Refs.[25, 26, 28], a one-dimensional truncation error analysis is performed and a source discretization formula is derived by eliminating a first-order truncation error. The formula is applied to a two-dimensional scheme in the form of an extended Galerkin source discretization:

$$\int_{V_j} s\, dV = \sum_{k\in\{k_j\}} \frac{1}{2} (s_L + s_R) V_{jk}, \tag{4.1}$$

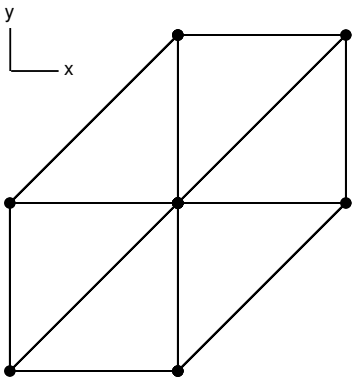


Figure 3: Regular triangular grid.

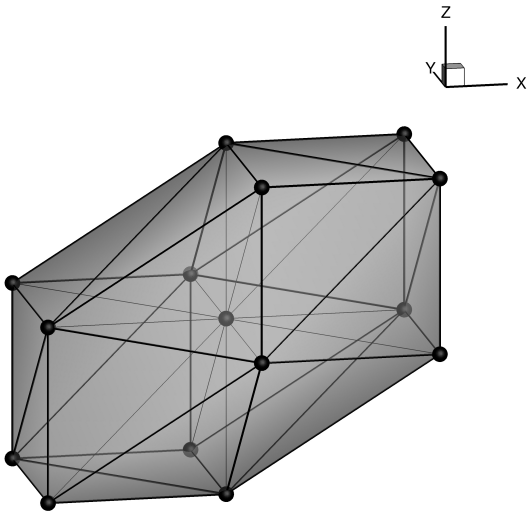


Figure 4: Regular tetrahedral grid.

where

$$V_{jk} = \frac{1}{4}(\Delta \mathbf{x}_{jk} \cdot \mathbf{n}_{jk}), \quad s_L = s_j - \frac{1}{2}\hat{\partial}_{jk} s_j - \frac{1}{8}\hat{\partial}^2_{jk} s_j, \quad s_R = s_k - \frac{1}{2}\hat{\partial}_{jk} s_k - \frac{1}{8}\hat{\partial}^2_{jk} s_k, \quad \hat{\partial}_{jk} \equiv \Delta \mathbf{x}_{jk} \cdot (\hat{\partial}_x, \hat{\partial}_y), \tag{4.2}$$

and the quantity V_{jk} may be called a partial volume since we have

$$V_j = \sum_{k \in \{k_j\}} V_{jk}. \tag{4.3}$$

Note that the formula requires not only first derivatives but also second derivatives of the source term. The derivatives can be obtained analytically (if possible) for a given source term function or by a quadratic LSQ method. The first derivatives $\hat{\partial}_x s_j$ and $\hat{\partial}_y s_j$ can be computed as in Equation (2.6) from discrete nodal values of the source term. The second derivatives $\hat{\partial}_{xx} s_j$, $\hat{\partial}_{xy} s_j$, and $\hat{\partial}_{yy} s_j$ can be computed by applying the quadratic LSQ method to the LSQ gradients, $\hat{\partial}_x s_j$ and $\hat{\partial}_y s_j$. This approach will greatly enlarge the stencil of the source discretization. Alternatively, one may store quadratic LSQ coefficients for the second derivatives in addition to c^x_{jk}, c^y_{jk}, and c^z_{jk} [29], and compute the second derivatives in a similar manner to Equation (2.6). This is a typical approach in quadratic LSQ methods; it requires a large amount of memory to store a total of six/nine LSQ coefficients in two/three dimensions. The above formula has been successfully used in two-dimensional computations [25]. However, it has never been applied to three-dimensional tetrahedral grids. We have found theoretically and numerically that the above formula is valid only in two dimensions, and does not provide third-order accuracy on tetrahedral grids (and also on one-dimensional grids) as we will discuss later.

The other approach is a divergence formulation of source terms [27]. In this approach, the source term is written in the divergence form in the neighborhood of a node j, where the residual is defined. The core idea is to find an approximate form of the source term resembling the conservation law, so that it can be discretized in the same way as the conservation law to yield the same form of truncation error and thus meet the compatibility condition. In two dimensions, the divergence formulation is given by

$$s = \partial_x f^s + \partial_y g^s, \tag{4.4}$$

where

$$f^s = \frac{1}{2}(x - x_j)s - \frac{1}{4}(x - x_j)^2 \partial_x s + \frac{1}{12}(x - x_j)^3 \partial_{xx} s, \tag{4.5}$$

$$g^s = \frac{1}{2}(y - y_j)s - \frac{1}{4}(y - y_j)^2 \partial_y s + \frac{1}{12}(y - y_j)^3 \partial_{yy} s. \tag{4.6}$$

In this paper, this formulation is referred to as the symmetric divergence formulation. It can be shown by differentiation that

$$\partial_x f^s + \partial_y g^s = s + \frac{1}{12}(x - x_j)^3 \partial_{xxx} s + \frac{1}{12}(y - y_j)^3 \partial_{yyy} s, \tag{4.7}$$

which shows that the symmetric divergence formulation is a third-order approximation to the source term, and therefore it will not degrade third-order accuracy. The source term written in the divergence form can be discretized straightforwardly by the same algorithm that is used to discretize the conservation law, and therefore the compatibility condition is guaranteed. Note that this approach also requires second derivatives of the source term in the flux extrapolation step, which uses the gradient of the fluxes f^s and g^s; but third-order derivative terms in the flux gradients can be ignored without degrading third-order accuracy [27]. A simple version, called the one-component divergence formulation [27], uses only the x-component of the source fluxes:

$$f^s = (x - x_j)s - \frac{1}{2}(x - x_j)^2 \partial_x s + \frac{1}{6}(x - x_j)^3 \partial_{xx} s, \tag{4.8}$$

$$g^s = 0, \tag{4.9}$$

which also gives a third-order approximation to the source term:

$$\partial_x f^s + \partial_y g^s = s + \frac{1}{6}(y - y_j)^3 \partial_{yyy} s. \tag{4.10}$$

This version requires only two second derivatives, $\partial_{xx} s$ and $\partial_{xy} s$, in the discretization. The divergence formulations can be extended easily to three dimensions:

$$s = \partial_x f^s + \partial_y g^s + \partial_z h^s, \tag{4.11}$$

where

$$f^s = \frac{1}{3}(x - x_j)s - \frac{1}{6}(x - x_j)^2 \partial_x s + \frac{1}{18}(x - x_j)^3 \partial_{xx} s, \tag{4.12}$$

$$g^s = \frac{1}{3}(y - y_j)s - \frac{1}{6}(y - y_j)^2 \partial_y s + \frac{1}{18}(y - y_j)^3 \partial_{yy} s, \tag{4.13}$$

$$h^s = \frac{1}{3}(z - z_j)s - \frac{1}{6}(z - z_j)^2 \partial_z s + \frac{1}{18}(z - z_j)^3 \partial_{zz} s, \tag{4.14}$$

or

$$f^s = (x - x_j)s - \frac{1}{2}(x - x_j)^2 \partial_x s + \frac{1}{6}(x - x_j)^3 \partial_{xx} s, \tag{4.15}$$

$$g^s = 0, \tag{4.16}$$

$$h^s = 0. \tag{4.17}$$

In three dimensions, the one-component divergence formulation uses only three second derivatives, $(\partial_{xx} s, \partial_{xy} s, \partial_{xz} s)$, while six second derivatives are required in the symmetric divergence formulation even with the symmetry assumption (e.g., $\partial_{xy} s = \partial_{yx} s$). These divergence formulations of source terms have been successfully applied to various applications, including the Euler and Navier-Stokes computations by the third-order edge-based scheme in both two and three dimensions [16, 19, 35].

A special divergence formulation exists for a vector of source terms $\mathbf{s} = (s_x, s_y, s_z)$ satisfying the curl-free condition:

$$\operatorname{curl} \mathbf{s} = 0. \tag{4.18}$$

If we define a source flux tensor in the neighborhood of a node j as

$$\mathbf{f}^s = \frac{\mathbf{s} \otimes \Delta\mathbf{x} - (\mathbf{s} \cdot \Delta\mathbf{x})\mathbf{I}}{2}$$

$$= \frac{1}{2} \begin{bmatrix} -s_y(y - y_j) - s_z(z - z_j) & s_x(y - y_j) & s_x(z - z_j) \\ s_y(x - x_j) & -s_x(x - x_j) - s_z(z - z_j) & s_y(z - z_j) \\ s_z(x - x_j) & s_z(y - y_j) & -s_y(y - y_j) - s_x(x - x_j) \end{bmatrix}, \tag{4.19}$$

where $\otimes$ denotes the dyadic product, $\mathbf{I}$ is the 3×3 identity matrix, and $\Delta\mathbf{x} = (x - x_j, y - y_j, z - z_j)$, then, we find

$$\begin{aligned}
\mathrm{div}\mathbf{f}^s &= \frac{1}{2}\left[\,\mathrm{div}(\mathbf{s}\otimes\Delta\mathbf{x}) - \mathrm{grad}(\mathbf{s}\cdot\Delta\mathbf{x})\,\right] \\
&= \frac{1}{2}\left[\,(\mathrm{grad}\,\mathbf{s})\Delta\mathbf{x} + \mathbf{s}\,\mathrm{div}\Delta\mathbf{x} - \{\mathrm{grad}\,\mathbf{s})\Delta\mathbf{x} + (\mathrm{grad}\Delta\mathbf{x})\mathbf{s} + \mathbf{s}\times\mathrm{curl}\,\Delta\mathbf{x} + \Delta\mathbf{x}\times\mathrm{curl}\,\mathbf{s}\}\,\right] \\
&= \frac{1}{2}\left[\,\mathbf{s}\,\mathrm{div}\Delta\mathbf{x} - (\mathrm{grad}\Delta\mathbf{x})\mathbf{s} - \mathbf{s}\times\mathrm{curl}\,\Delta\mathbf{x} - \Delta\mathbf{x}\times\mathrm{curl}\,\mathbf{s}\,\right] \\
&= \frac{1}{2}\left[\,3\mathbf{s} - \mathbf{s} - \Delta\mathbf{x}\times\mathrm{curl}\,\mathbf{s}\,\right] \\
&= \mathbf{s} - \frac{\Delta\mathbf{x}\times\mathrm{curl}\,\mathbf{s}}{2} \\
&= \mathbf{s}. \qquad (4.20)
\end{aligned}$$

Therefore, the curl-free source term vector $\mathbf{s}$ can be expressed exactly as

$$\mathbf{s} = \mathrm{div}\mathbf{f}^s. \qquad (4.21)$$

Then, we can discretize $\mathrm{div}\mathbf{f}^s$ by the third-order edge-based scheme to generate a compatible discretization of the source term $\mathbf{s}$. This approach does not require second derivatives of the source term because the flux $\mathbf{f}^s$ has no source-term derivatives. This approach has been demonstrated in two dimensions for third-order hyperbolic diffusion and advection-diffusion schemes [29, 36], and also for a third-order hyperbolic Navier-Stokes scheme [37], where source terms are equivalent to the gradients of solution variables and thus curl-free. The special source flux tensor, Equation (4.19), is a three-dimensional generalization of these two-dimensional formulations.

For general source terms, these reported approaches require computations and storage of second derivatives. Although the one-component divergence formulation requires fewer second derivatives, it still takes substantial resources to compute and store second derivatives, especially for three-dimensional unsteady applications. The special divergence formulation, Equation (4.21), does not require second derivatives, but its applicability is limited to a particular class of source terms and also it involves the gradients of the source term at the neighbor nodes, which extends the discretization stencil considerably. In the next section, we propose a new approach, and derive a family of formulas that do not require second derivatives at all. Then, a unique formula will be derived from the family that does not even require the gradients of the source terms at the neighbor nodes.

5 Accuracy-Preserving Source Term Quadrature

Generalizing the extended Galerkin formula [25, 26, 28], we seek a general accuracy-preserving source term quadrature in the form:

$$\int_{V_j} s\,dV = \sum_{k\in\{k_j\}} \frac{1}{2}(s_L + s_R)V_{jk}, \qquad (5.1)$$

where

$$V_{jk} = \frac{1}{2D}(\Delta\mathbf{x}_{jk}\cdot\mathbf{n}_{jk}), \qquad (5.2)$$

$$s_L = a_L s_j + b_L\hat{\partial}_{jk}s_j + c_L\hat{\partial}^2_{jk}s_j, \qquad (5.3)$$

$$s_R = a_R s_k + b_R\hat{\partial}_{jk}s_k + c_R\hat{\partial}^2_{jk}s_k, \qquad (5.4)$$

and aim to determine the coefficients $(a_L, b_L, c_L, a_R, b_R, c_R)$ to achieve third-order accuracy. Again, the hat indicates that the derivatives are evaluated numerically, e.g., by a quadratic LSQ method or analytically for

a given source term function. The constant D is defined as $D = 2$ in two dimensions and $D = 3$ in three dimensions, which allows us to unify the expression for the partial volume V_{jk}, satisfying

$$V_j = \sum_{k \in \{k_j\}} V_{jk}, \tag{5.5}$$

in both two and three dimensions. The derivative operator $\hat{\partial}_{jk}$ is defined as Equation (4.2) in two dimensions, and as

$$\hat{\partial}_{jk} = \Delta \mathbf{x}_{jk} \cdot (\hat{\partial}_x, \hat{\partial}_y, \hat{\partial}_z), \tag{5.6}$$

in three dimensions, and similarly for those without a hat that denote derivative operators in Taylor expansions. Note that the source quadrature (5.1) includes the point quadrature, which can be obtained with $a_L = 2, b_L = c_L = a_R = b_R = c_R = 0$. For third-order accuracy, it suffices to assume that the source term is a quadratic function. Then, the source term value at the neighbor k is expanded as

$$s_k = s_j + \partial_{jk} s_j + \frac{1}{2} \partial_{jk}^2 s_j, \tag{5.7}$$

the gradients at k and j are exact for quadratic functions and thus

$$\hat{\partial}_{jk} s_k = \partial_{jk} s_j + \partial_{jk}^2 s_j, \tag{5.8}$$

$$\hat{\partial}_{jk} s_j = \partial_{jk} s_j, \tag{5.9}$$

and the quadratic term is a global constant for quadratic functions,

$$\hat{\partial}_{jk}^2 s_k = \hat{\partial}_{jk}^2 s_j = \partial_{jk}^2 s_k = \partial_{jk}^2 s_j. \tag{5.10}$$

Substituting these expressions into Equation (5.1), we obtain the Taylor expansion up to the second-order error:

$$\begin{aligned}\sum_{k \in \{k_j\}} \frac{1}{2}(s_L + s_R) V_{jk} =& \sum_{k \in \{k_j\}} \left(\frac{a_L + a_R}{2} s_j + \frac{a_R + b_L + b_R}{2} \partial_{jk} s_j + \frac{2(b_R + c_R + c_L) + a_R}{4} \partial_{jk}^2 s_j \right) V_{jk} \\ =& \frac{a_L + a_R}{2} s_j V_j + \frac{a_R + b_L + b_R}{2} \sum_{k \in \{k_j\}} \partial_{jk} s_j V_{jk} + \frac{2(b_R + c_R + c_L) + a_R}{4} \sum_{k \in \{k_j\}} \partial_{jk}^2 s_j V_{jk}. \end{aligned} \tag{5.11}$$

$$\sum_{k \in \{k_j\}} \frac{1}{2}(s_L + s_R) V_{jk} = \frac{a_L + a_R}{2} s_j V_j + \frac{a_R + b_L + b_R}{2} \sum_{k \in \{k_j\}} \partial_{jk} s_j V_{jk} + \frac{2(b_R + c_R + c_L) + a_R}{4} \sum_{k \in \{k_j\}} \partial_{jk}^2 s_j V_{jk}$$

For consistency, we must have

$$a_L + a_R = 2. \tag{5.12}$$

For third-order accuracy, the truncation error is expected to be $O(h^2)$ on irregular grids, and therefore we require vanishing coefficient of the first-order error term:

$$a_R + b_L + b_R = 0. \tag{5.13}$$

These two conditions are not sufficient to guarantee third-order accuracy as numerically demonstrated later. Another condition is required on the second-order error term. This term must not vanish but needs to yield a compatible second-order error term on regular grids. Expanding the second-order term on a regular grid, we obtain

$$\sum_{k \in \{k_j\}} \frac{1}{2}(s_L + s_R) V_{jk} = s_j V_j + \frac{2(b_R + c_R + c_L) + a_R}{4} \left(\frac{D+2}{6D} \right) Q_s, \tag{5.14}$$

$$\frac{1}{V_j}\sum_{k\in\{k_j\}}\frac{1}{2}(s_L+s_R)V_{jk}=s_j+\frac{2(b_R+c_R+c_L)+a_R}{4}\left(\frac{1}{3V_j}\right)\left[Q_{xx}+Q_{yy}+Q_{xy}\right]s_j \tag{5.15}$$

where the conditions (5.12) and (5.13) have been assumed, and Q_s is a second-order error term that can be factored (see Appendix A for the derivation): in two dimensions,

$$Q_s=\left[Q_{xx}+Q_{xy}+Q_{yy}\right]s_j, \tag{5.16}$$

and in three dimensions,

$$Q_s=\left[Q_{xx}+Q_{yy}+Q_{zz}+Q_{xy}+Q_{yz}+Q_{zx}\right]s_j. \tag{5.17}$$

Remarkably, the terms in the square brackets are identical to those in Equations (3.2) and Equation (3.6), in two and three dimensions, respectively. Therefore, the compatibility condition (3.10) or (3.11) requires

$$\left(\frac{D+2}{24D}\right)(2c_L+2b_R+2c_R+a_R)=-\frac{1}{24}, \tag{5.18}$$

or

$$a_R=-2(b_R+c_R+c_L)-\frac{D}{D+2}. \tag{5.19}$$

The quadrature formula (5.1) achieves third-order accuracy on arbitrary simplex grids if the coefficients satisfy the three conditions:

$$a_L+a_R=2,\quad a_R+b_L+b_R=0,\quad a_R=-2(b_R+c_R+c_L)-\frac{D}{D+2}. \tag{5.20}$$

These conditions form an underdetermined system for the unknowns: a_L, b_L, c_L, a_R, b_R, and c_R. In effect, since there are three conditions for six unknowns, these conditions define a three-parameter family of accuracy preserving source term quadrature formulas. Various formulas can be generated by imposing three additional conditions. For example, the symmetry conditions:

$$a_L=a_R,\quad b_L=b_R,\quad c_L=c_R, \tag{5.21}$$

lead to

$$a_L=a_R=1,\quad b_L=b_R=-\frac{1}{2},\quad c_L=c_R=-\frac{D}{4(D+2)}. \tag{5.22}$$

In this paper, this formula is referred to as the symmetric formula. For $D=2$, the symmetric formula reproduces the formula (4.2) proposed by Katz in Refs.[25, 26, 28]. For this reason, in two dimensions only, it is referred to as Katz. Note that the above symmetric formula is unique, and no other symmetric formulas can provide third-order accuracy. This proves that the Katz formula (4.2) is valid only in two dimensions, and cannot provide third-order accuracy on tetrahedral grids (also on one-dimensional grids) as mentioned earlier. The incompatibility of the Katz formula (4.2) for tetrahedral and one-dimensional grids has been confirmed by numerical experiments.

It is also possible to derive a one-sided formula, which depends only on the quantities at the node j by setting

$$a_R=b_R=c_R=0, \tag{5.23}$$

resulting in

$$a_L=2,\quad b_L=0,\quad c_L=-\frac{D}{2(D+2)},\quad a_R=b_R=c_R=0. \tag{5.24}$$

This formula yields a source term discretization with a compact stencil if the second derivatives can be computed with a neighbor-only stencil. This is possible, but requires computing the LSQ fit with a compact stencil and storing the LSQ coefficients for the second derivatives. A more economical formula is obtained by requiring

$$a_L=1,\quad c_L=c_R=0, \tag{5.25}$$

	Grid type	a_L	b_L	c_L	a_R	b_R	c_R
Regular	Regular simplex	$\frac{3D+4}{D+2}$	0	0	$-\frac{D}{D+2}$	0	0
Compact	Arbitrary simplex	$\frac{3D+4}{D+2}$	$\frac{D}{D+2}$	0	$-\frac{D}{D+2}$	0	0
Economical(1)	Arbitrary simplex	1	$-\frac{1}{D+2}$	0	1	$-\frac{D+1}{D+2}$	0
One-sided	Arbitrary simplex	2	0	$-\frac{D}{2(D+2)}$	0	0	0
Symmetric	Arbitrary simplex	1	$-\frac{1}{2}$	$-\frac{D}{4(D+2)}$	1	$-\frac{1}{2}$	$-\frac{D}{4(D+2)}$

Table 1: Summary of accuracy-preserving source term quadrature formulas. $D = 2$ for triangular grids, and $D = 3$ for tetrahedral grids (and $D = 1$ for one-dimensional grids). Top one is for the regular grid, the next two do not require second derivatives of the source term; the bottom two require the second derivatives. The value in the parenthesis for Economical indicates the chosen value of a_L to generate the formula from the one-parameter family of formulas satisfying $c_L = c_R = 0$.

which leads to

$$a_L = a_R = 1, \quad b_L = -\frac{1}{D+2}, \quad b_R = -\frac{D+1}{D+2}, \quad c_L = c_R = 0. \tag{5.26}$$

This formula does not require second derivatives of the source term. It brings a substantial saving in computational cost, especially in three dimensions, since there is no need to compute and store nine (or six if symmetry is enforced) second derivatives. In fact, the above formula is just one example of a one-parameter family of economical formulas satisfying $c_L = c_R = 0$ (no second derivatives). A special formula that does not require any derivative at the neighbors can be derived from this family by imposing $b_R = 0$ instead of $a_L = 1$:

$$b_R = c_L = c_R = 0, \tag{5.27}$$

which yields

$$a_L = \frac{3D+4}{D+2}, \quad a_R = -\frac{D}{D+2}, \quad b_L = \frac{D}{D+2}, \quad b_R = c_L = c_R = 0. \tag{5.28}$$

This source term discretization depends only on the neighbors if the quadratic fit is performed at j with a compact stencil including neighbors only or if the solution gradient is stored as a part of the solution vector, e.g., in hyperbolic diffusion schemes [16, 18, 29]. Therefore, it allows the exact linearization, in the case s is a function of the solution, with a minimum bandwidth in the construction of implicit solvers or discrete adjoint equations. For this reason, this formula is referred to as the compact formula. The compact formula has a close connection with a formula on a regular grid. On a regular grid, the first order error term vanishes independently of $a_R + b_L + b_R$ (see Appendix A), and therefore the second condition in Equation (5.20) is redundant. Hence, we can seek a formula with no derivatives (see Equation (5.4)) by imposing

$$b_L = b_R = c_L = c_R = 0. \tag{5.29}$$

In this way, we obtain a formula for regular grids, which was previously known only in two dimensions [27],

$$a_L = \frac{3D+4}{D+2}, \quad a_R = -\frac{D}{D+2}, \quad b_L = b_R = c_L = c_R = 0. \tag{5.30}$$

No source derivatives are needed on regular grids. Comparing with Equation (5.28), we find that the compact formula (5.28) can be considered as a simple extension of this regular formula to irregular grids. All formulas discussed are summarized in Table 1.

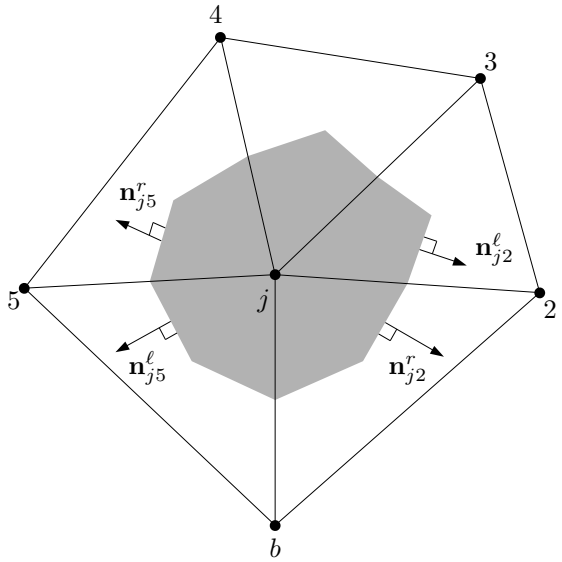


Figure 5: Interior stencil of the edge-based discretization.

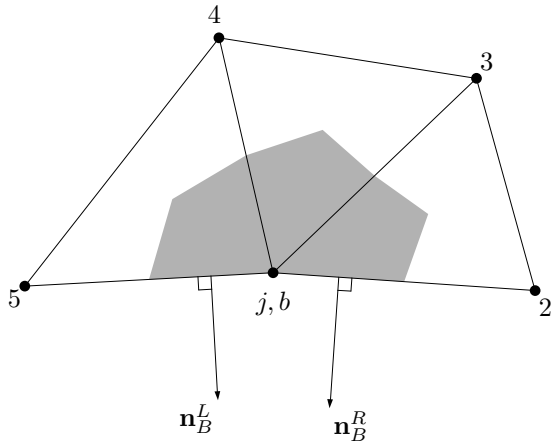


Figure 6: Interior stencil made a boundary stencil by moving the node b to the node j.

Remark: The formula for regular grids can yield third-order accuracy on special non-regular grids, where the first-order truncation error vanishes identically. See Appendix B.

Remark: Formulas in one dimension can be derived in a similar manner. For example, below is the compact formula in one dimension:

$$\int_{V_j} s\, dV = \sum_{k\in\{k_j\}} \frac{1}{2}(s_L + s_R)V_{jk}, \quad s_L = \frac{7}{3}s_j + \frac{1}{3}(x_k - x_j)\hat{\partial}_x s_j, \quad s_R = -\frac{1}{3}s_k. \tag{5.31}$$

where V_j is the dual control volume $V_j = (x_{j+1} - x_{j-1})/2$, $\{k_j\} = \{j-1, j+1\}$, $V_{jk} = |x_k - x_j|/2$, and the derivative $\hat{\partial}_x s_j$ is computed such that it is exact for quadratic functions. In fact, all the formulas presented in Table 1 are valid for one-dimensional grids with $D = 1$. In one dimension, the quadratic fit is possible with a compact stencil, and therefore the above formula yields a compact source discretization for one-dimensional third-order node-centered schemes. Our primary focus in this paper is on two and three dimensions, but numerical results will be shown later to demonstrate the new formulas in one dimension.

6 Source Term Discretization at Boundary Nodes

At a boundary node, the third-order edge-based discretization needs to be closed by the accuracy-preserving boundary flux quadrature formula [19] to guarantee third-order accuracy with a weak boundary condition. However, the family of source quadrature formulas derived in this paper do not require any boundary contribution. To show this, we use the edge-collapsing approach [19]. As demonstrated in Ref.[19], the accuracy-preserving boundary flux quadrature formula can be derived from the edge-based scheme at an interior node by collapsing an edge. The source term discretization derived at a boundary node in exactly the same manner preserves third-order accuracy for conservation laws with source terms. Consider a stencil shown in Figure 5. The source discretization (5.1) at a node j is given as a sum over the neighbors $\{b, 2, 3, 4, 5\}$:

$$\int_{V_j} s\, dV = \sum_{k=b,2}^{5} \frac{1}{2} s_{jk} V_{jk}, \tag{6.1}$$

where $s_{jk} = (s_L + s_R)/2$ across the node j and the neighbor k. By splitting V_{j2} and V_{j5} into two parts (see Figure 5):

$$V_{j2} = V_{j2}(\mathbf{n}^{\ell}_{j2}) + V_{j2}(\mathbf{n}^{r}_{j2}) = \frac{1}{4}(\Delta\mathbf{x}_{j2}\cdot\mathbf{n}^{\ell}_{j2}) + \frac{1}{4}(\Delta\mathbf{x}_{j2}\cdot\mathbf{n}^{r}_{j2}), \tag{6.2}$$

$$V_{j5} = V_{j5}(\mathbf{n}^{\ell}_{j5}) + V_{j5}(\mathbf{n}^{r}_{j5}) = \frac{1}{4}(\Delta\mathbf{x}_{j5}\cdot\mathbf{n}^{\ell}_{j5}) + \frac{1}{4}(\Delta\mathbf{x}_{j5}\cdot\mathbf{n}^{r}_{j5}), \tag{6.3}$$

we obtain

$$\int_{V_j} s\,dV = \sum_{k=3}^{4}\frac{1}{2}s_{jk}V_{jk} + \frac{1}{2}s_{j2}V_{j2}(\mathbf{n}^{\ell}_{j2}) + \frac{1}{2}s_{j2}V_{j2}(\mathbf{n}^{r}_{j2}) + \frac{1}{2}s_{j5}V_{j5}(\mathbf{n}^{\ell}_{j5}) + \frac{1}{2}s_{j5}V_{j5}(\mathbf{n}^{r}_{j5}) + \frac{1}{2}s_{jb}V_{jb}. \tag{6.4}$$

To derive a formula for a boundary node, we collapse the edge $[j, b]$ with the node j fixed and create a boundary stencil shown in Figure 6. In the limit, we have

$$\Delta\mathbf{x}_{j2}\cdot\mathbf{n}^{r}_{j2} \to \frac{1}{3}\Delta\mathbf{x}_{j2}\cdot\mathbf{n}^{R}_{B} \to 0, \quad \Delta\mathbf{x}_{j5}\cdot\mathbf{n}^{\ell}_{j5} \to \frac{1}{3}\Delta\mathbf{x}_{j5}\cdot\mathbf{n}^{L}_{B} \to 0, \quad \Delta\mathbf{x}_{jb} \to 0, \tag{6.5}$$

and thus,

$$V_{j2}(\mathbf{n}^{r}_{j2}) \to 0, \quad V_{j5}(\mathbf{n}^{\ell}_{j5}) \to 0, \quad V_{jb} \to 0. \tag{6.6}$$

Therefore, all contributions associated with the boundary vanish, and we are left with

$$\int_{V_j} s\,dV = \sum_{k=3}^{4}\frac{1}{2}s_{jk}V_{jk} + \frac{1}{2}s_{j2}V_{j2}(\mathbf{n}^{\ell}_{j2}) + \frac{1}{2}s_{j5}V_{j5}(\mathbf{n}^{r}_{j5}). \tag{6.7}$$

Hence, we only need to make a loop over edges, and do not need to close the source term residual by boundary contributions.

The vanishing boundary contribution is due to the vanishing partial volume V_{jk} in the collapsed stencil: $V_{j2}(\mathbf{n}^{r}_{j2}) = V_{j5}(\mathbf{n}^{\ell}_{j5}) = V_{jb} = 0$. The same is true in three dimensions, and thus the derivation is omitted. As a result, in three dimensions also, only the edge loop is needed, and no boundary closure is necessary for source quadrature formulas derived in this paper to preserve third-order accuracy at boundary nodes.

On the other hand, the formulas derived from the divergence formulations require the boundary closure, in principle, because boundary contributions do not vanish all together in the edge-collapsing process. To show this, consider the source quadrature formula obtained by discretizing the symmetric divergence formulation by the third-order edge-based scheme in the stencil in Figure 5 (see Ref.[27]):

$$\int_{V_j} s\,dV = \sum_{k=b,2}^{5}\frac{1}{2}(s_L + s_R), \tag{6.8}$$

where

$$s_L = s_j V_{jk}, \tag{6.9}$$

$$s_R = (s_k - \hat{\partial}_{jk}s_k + \frac{1}{2}\Delta x_{jk}\Delta y_{jk}\hat{\partial}_{xy}s_k)V_{jk} + \frac{1}{12}\left[\Delta x^2_{jk}\hat{\partial}_{xx}s_k(\Delta x_{jk}n_x) + \Delta y^2_{jk}\hat{\partial}_{yy}s_k(\Delta y_{jk}n_y)\right], \tag{6.10}$$

and n_x and n_y are the x- and y-components of the directed area vector $\mathbf{n}_{jk}$, respectively. The terms proportional to V_{jk} will vanish in the collapsed stencil, and therefore do not require the boundary closure. However, the last term in s_R does not vanish in general; it vanishes only if $\Delta x_{jk}n_x = 0$ and $\Delta y_{jk}n_y = 0$, e.g., at boundaries with x =constant or y =constant such as those on a rectangular domain. Also, the one-component divergence formulation gives a formula in the same form with

$$s_L = \frac{1}{2}s_j(\Delta x_{jk}n_x), \tag{6.11}$$

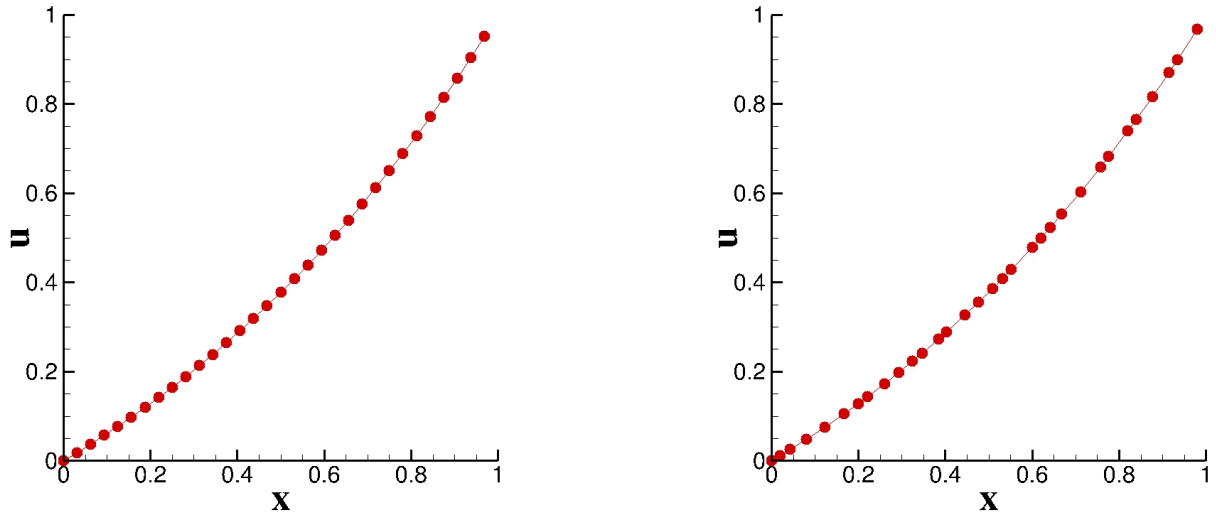
$$s_R = \frac{1}{2}\left[s_k - \hat{\partial}_{jk}s_k + \frac{1}{3}\Delta x^2_{jk}\hat{\partial}_{xx}s_k + \frac{1}{2}\Delta x_{jk}\Delta y_{jk}\hat{\partial}_{xy}s_k\right](\Delta x_{jk}n_x). \tag{6.12}$$

All terms remain in the collapsed stencil unless $\Delta x_{jk} n_x = 0$, and therefore it requires the boundary closure in general. If the boundary closure is ignored, there are $O(h^2)$ errors in the symmetric formula, and $O(1)$ errors in the one-component formula. Numerical experiments show that the former can be tolerated for third-order accuracy but the latter leads to severe accuracy deterioration and thus requires the boundary closure with the accuracy-preserving boundary flux quadrature formula [19]. In all previous works utilizing the one-component formula, the boundary closure has actually been implemented [19, 35, 37]. It is straightforward to show that in three dimensions, the divergence formulations also require the boundary flux closure; if the boundary contributions are ignored, the symmetric and one-component formulations yields $O(h^2)$ and $O(1)$ errors, respectively.

7 Numerical Results

Third-order accuracy of the edge-based scheme is demonstrated for source terms generated by the method of manufactured solutions as well as for equations with solution-dependent source terms. All problems are steady problems. Applications to unsteady problems will be discussed elsewhere. For all problems, the residual is fully converged by an implicit solver with ten orders of magnitude reduction from an initial residual. The focus here is the order of error convergence achieved by various source discretization formulas, and therefore iterative convergence histories are irrelevant and will not be discussed. Effects on iterative convergence are more relevant to unsteady schemes where the source term is a function of solutions, and will be addressed in the future study for unsteady problems.

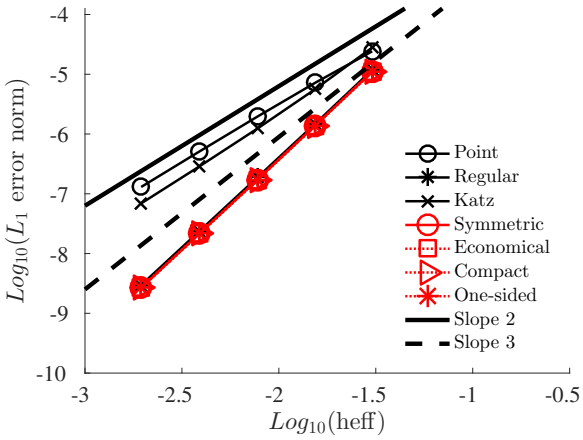
In the results presented here, the formula derived from the one-component divergence formulation with source fluxes (4.8) and (4.9) is referred to as Div(x), and the one derived from the symmetric divergence formulation with source fluxes (4.5) and (4.6) is referred to as Div(sym). The point quadrature (3.12) is referred to as Point. Other formulas are referred to as indicated in Table 1. Note that the symmetric formula with $D = 2$ is equivalent to the Katz formula (4.2). All solution gradients are computed by the unweighted quadratic LSQ fit. For two-dimensional regular grids, the quadratic LSQ fit is performed with neighbors only at interior nodes; at boundary nodes, an extra node is added along the direction normal to the boundary (two extra nodes at corner nodes) in order to avoid ill-conditioning of the LSQ matrix caused by having not enough nodes to fit a quadratic function. For irregular grids, the quadratic fit uses the neighbors of the neighbors also, but it is implemented in two steps, where each step involves edge neighbors only, as described in Ref.[29]. First and second derivatives of the source term arising from manufactured solutions are derived analytically and evaluated directly at each node.



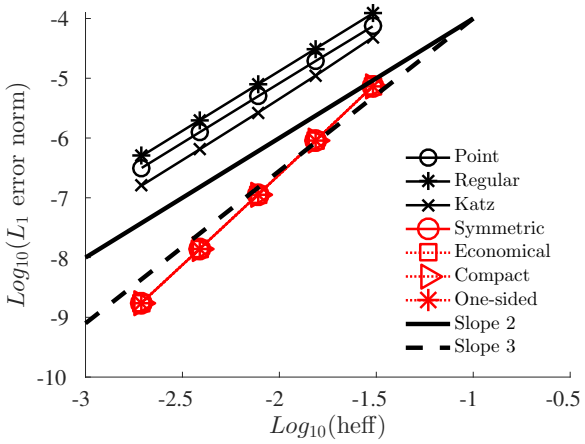
(a) Regular grid.

(b) Irregular grid.

Figure 7: Exact solution on regular and irregular one-dimensional grids with 32 nodes.

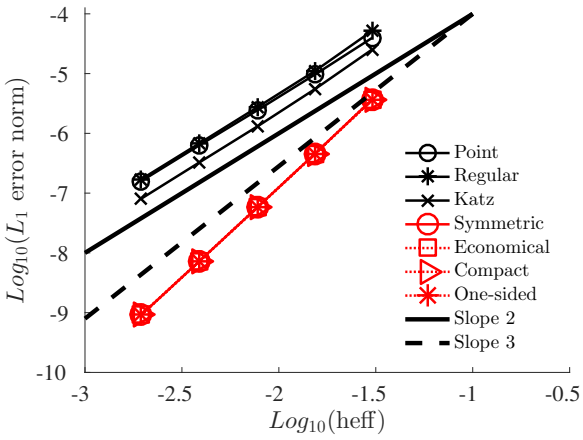


(a) Interior nodes.

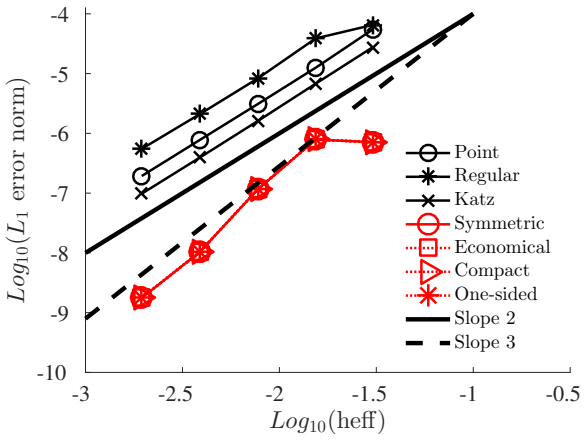


(b) Outflow boundary node ($x = 1$).

Figure 8: Error convergence results for the advection equation on regular one-dimensional grids.



(a) Interior nodes.



(b) Outflow boundary node ($x = 1$).

Figure 9: Error convergence results for the advection equation on irregular one-dimensional grids.

7.1 One Dimension

For the sake of completeness, one-dimensional results are presented for the advection equation:

$$a\partial_x u = s(x), \quad x \in (0, 1], \tag{7.1}$$

where $a = 1.0$, and the source term is defined as $s(x) = -a\exp(x)/(1 - \exp(1))$, so that the exact solution is given by

$$u(x) = \frac{1 - \exp(x)}{1 - \exp(1)}. \tag{7.2}$$

The advection equation is discretized by the edge-based scheme with the upwind flux (2.3) on a one-dimensional grid defined by the nodes $[x_0 = 0.0, x_1, x_2, \cdots, x_N = 1.0]$ with the nodal gradients are computed by a quadratic fit with two neighbor nodes. Both regular and irregular grids are used with $N = 31, 63, 127, 255, 511$. The irregular grids have been generated from the regular grids by applying random perturbations to the nodal coordinates except the two boundary nodes. Figure 7 shows the solution on the coarsest grids. The derivatives of the source terms are derived analytically and evaluated at each node. The solution is specified at the inflow boundary $x = x_0$, and a weak condition is applied at the outflow boundary $x = x_N$. The weak condition is implemented through the upwind numerical flux. The resulting discrete equations are solved by an implicit solver. See Ref.[22] for details on the weak boundary condition and the implicit solver implementations. The discretization errors are examined at interior nodes and at the outflow boundary node, separately, for all formulas in Table 1 with $D = 1$ and the point quadrature as well as the Katz formula (i.e., $a_L = a_R = 1, b_L = b_R = -1/2, c_L = c_R = -1/8$) for comparison. The discretization errors are measured in the L_1 norm separately for interior and boundary nodes, and the effective mesh spacing is computed as the L_1 norm of the dual control volumes.

Results obtained for regular grids are shown in Figure 8. As expected, all formulas except Point and Katz yield third-order accuracy at interior nodes (see Figure 8(a)). At the outflow boundary node, as shown in Figure 8(b), not only Point and Katz but also the regular formula fails to yield third-order accuracy because the stencil is not regular at the boundary node. All other formulas, which have been derived for irregular grids, successfully achieve third-order accuracy at the boundary node. Results for irregular grids are shown in Figure 9. As expected, all but Point, Katz, and Regular achieve third-order accuracy at both interior and boundary nodes. These numerical results confirm that the derived formulas in Table 1 are valid for one-dimensional grids with $D = 1$.

7.2 Two Dimensions

In two dimensions, we consider three examples of conservation laws in the form of Equation (2.1). The first example is a linear advection equation with $\mathbf{f} = (au, bu)$ and a source term,

$$s = ac\cos(cx)\sin(dy) + bd\sin(cx)\cos(dy) + (a + b)\exp(x + y), \tag{7.3}$$

where $(a, b) = (2.73, 1.31)$ and $c = 2.51\pi$ and $d = 3.48\pi$ for the exact solution given by

$$u(x, y) = \sin(cx)\sin(dy) + \exp(x + y). \tag{7.4}$$

As a nonlinear example, we consider Burgers' equation with $\mathbf{f} = (u^2/2, u)$ and a source term,

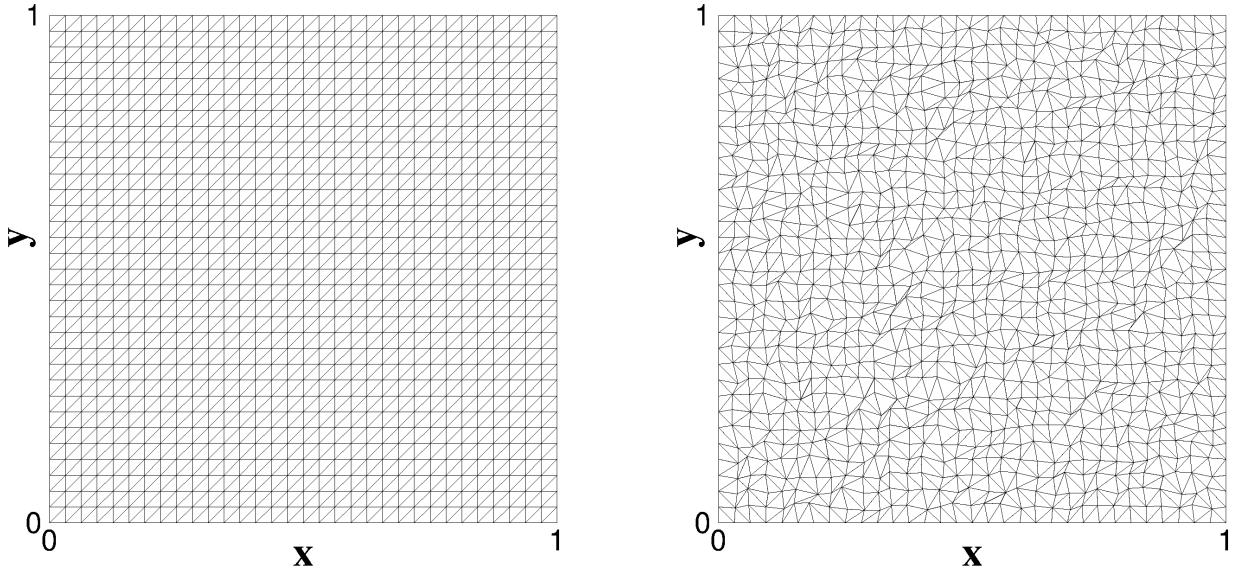
$$s = \cos(x - y)\sin(x - y) - \cos(x - y), \tag{7.5}$$

for the exact solution

$$u(x, y) = 2 + \sin(x - y). \tag{7.6}$$

Yet another example is a linear advection-diffusion equation written as a first-order conservation law with the flux tensor:

$$\mathbf{f} = \begin{bmatrix} a - \nu p & b - \nu q \\ -u & 0 \\ 0 & -u \end{bmatrix}, \tag{7.7}$$



(a) Regular grid.

(b) Irregular grid.

Figure 10: Regular and irregular triangular grids with 33×33 nodes.

and the source term vector:

$$\mathbf{s} = \begin{bmatrix} 0 \\ -p \\ -q \end{bmatrix}, \tag{7.8}$$

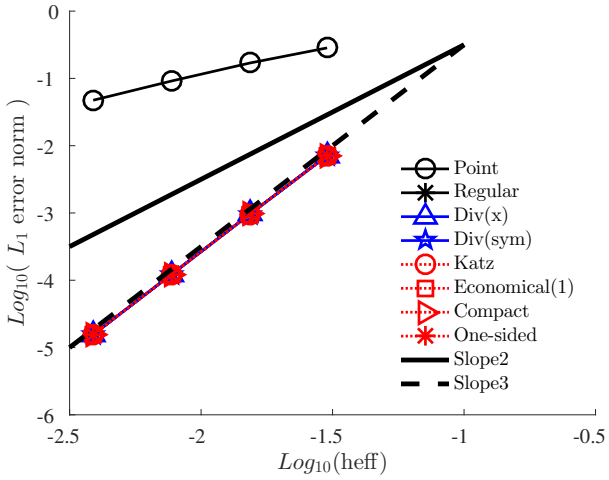
for the boundary-layer-type exact solution [38, 39]:

$$u(x,y) \quad = \quad \frac{\left[1-\exp\left((x-1)\frac{a}{\nu}\right)\right]\left[1-\exp\left((y-1)\frac{b}{\nu}\right)\right]}{\left[1-\exp\left(-\frac{a}{\nu}\right)\right]\left[1-\exp\left(-\frac{b}{\nu}\right)\right]}, \tag{7.9}$$

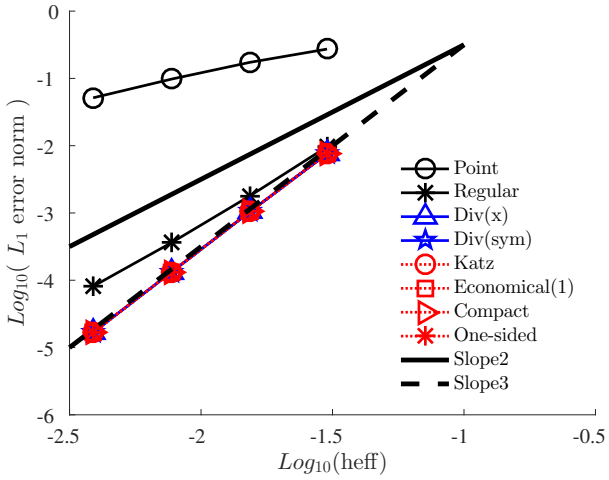
which serves as a non-manufactured-solution example. This system has a set of curl-free source terms, i.e., $\partial_x q - \partial_y p = 0$, and will be used to compare the new formulas with the special divergence formulation: Equation (4.21) with the source flux (4.19). In all cases, the numerical flux is computed by the upwind flux (2.3) and the corresponding vector version for the advection-diffusion system [29].

7.2.1 Square domain

We solve the linear advection and Burgers' equations in a square domain. The top and right boundaries correspond to outflow boundaries. Solutions at theses boundaries are determined by solving the residual equations. The residuals at boundary nodes are computed with the accuracy-preserving boundary flux quadrature formula [19], and the outflow boundary condition is imposed weakly with a ghost state set by the solution at the boundary node j: $u_b = u_j$, where the subscript b denotes the ghost state. See Refs.[16, 19] for more details on the implementation of weak boundary conditions. Note that the accuracy-preserving boundary flux quadrature is necessary for the conservation law since the third-order edge-based scheme does not necessarily tolerate lower-order accuracy at boundary nodes [19] unlike finite-difference schemes that allow lower-order accuracy at boundaries [40]. However, as mentioned in Section 6, the source discretization formulas derived from the divergence formulations do not require any boundary flux closure for the boundaries in the square domain.

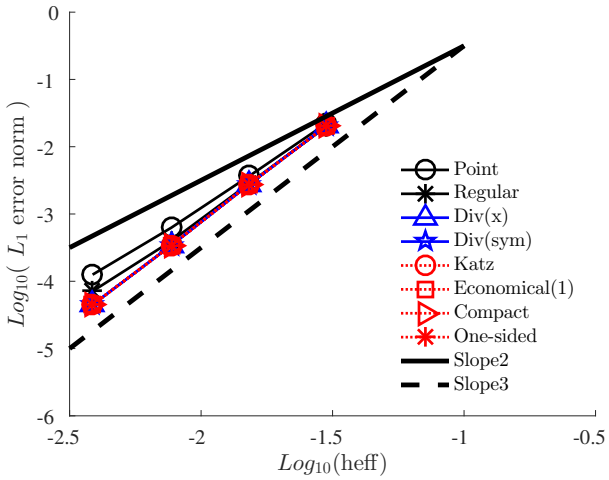


(a) Interior nodes.

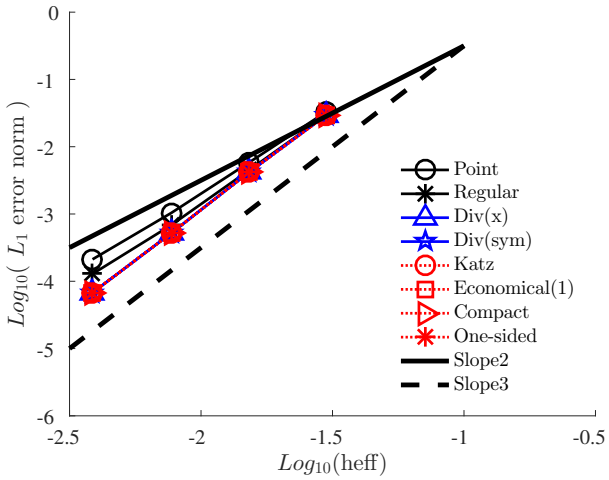


(b) Outflow boundary nodes.

Figure 11: Error convergence results for the advection equation on regular triangular grids.

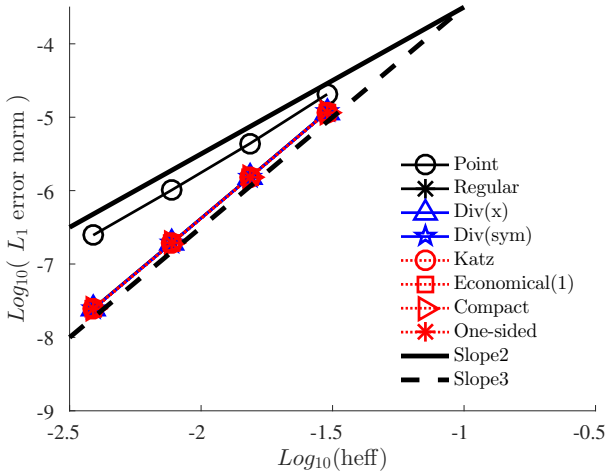


(a) Interior nodes.

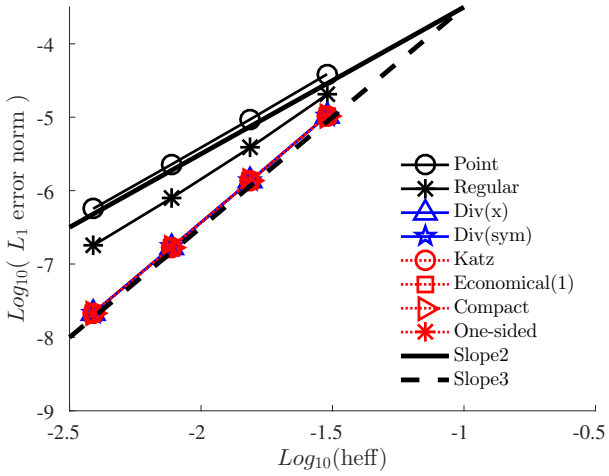


(b) Outflow boundary nodes.

Figure 12: Error convergence results for the advection equation on irregular triangular grids.

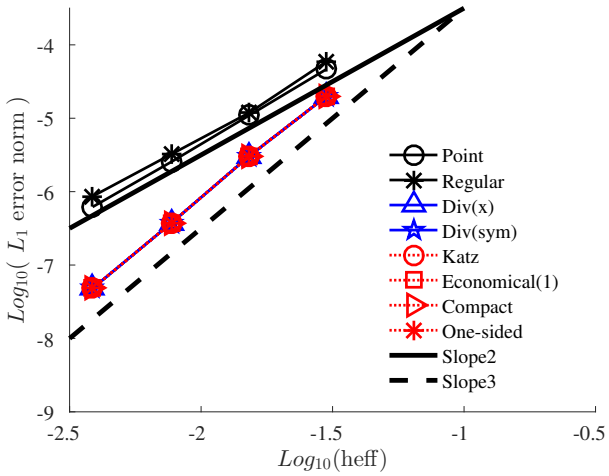


(a) Interior nodes.

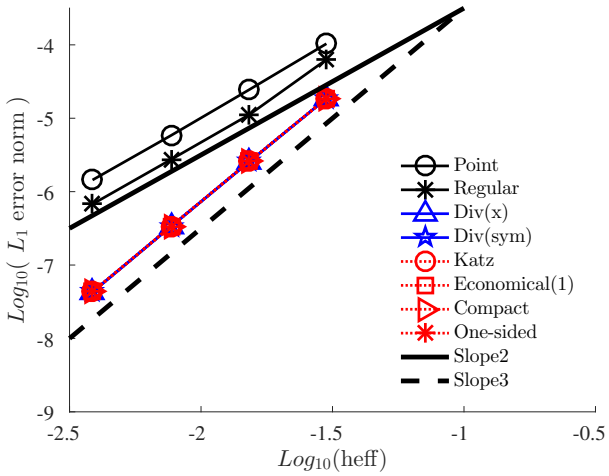


(b) Outflow boundary nodes.

Figure 13: Error convergence results for Burgers' equation on regular triangular grids.



(a) Interior nodes.



(b) Outflow boundary nodes.

Figure 14: Error convergence results for Burgers' equation on irregular triangular grids.

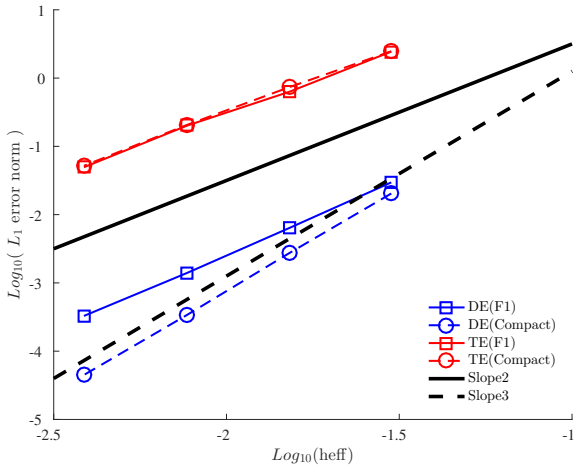


Figure 15: Truncation and discretization error convergence on irregular grids for an incompatible formula (F1) defined by $a_L = a_R = 1$, $b_L = -1$, $b_R = c_L = cR = 0$, which satisfies the consistency and zero first-order error conditions but does not satisfy the compatibility condition, and for a compatible formula ('Compact' in Table 1). DE and TE denote discretization and truncation errors, respectively.

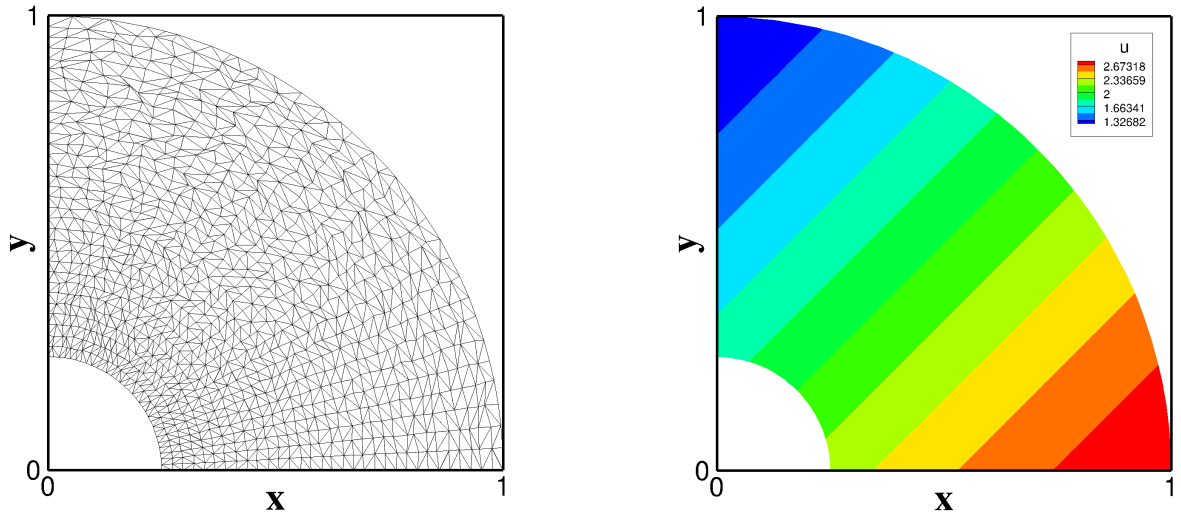
Therefore, the boundary flux closure is not applied in any of the formulas tested here; effects of the boundary closure for source terms will be discussed later in the next section. Computations are performed for regular and irregular grids of 33×33, 65×65, 129×129, and 257×257 nodes. The coarsest grids are shown in Figure 10. Irregular grids are generated from regular grids by random nodal perturbations at interior nodes and diagonal swappings, leading to random numbers of neighbors in contrast to regular grids where each stencil has exactly six neighbors everywhere inside the domain. Discretization errors are measured in the L_1 norm, and the effective mesh spacing is computed as the L_1 norm of the square root of the dual control volume V_j for all problems.

Results for the regular grids are shown in Figure 11. At interior nodes, all formulas except for the point quadrature (3.12) yield third-order accuracy as expected; see Figure 11(a). Note that the errors are less than second-order accurate with the point quadrature; such severe deterioration has been observed for the third-order edge-based scheme with incompatible boundary flux quadrature formulas in Ref.[19]. Discretization error convergence computed at the right outflow boundary is shown in Figure 11(b). Here, the regular formula fails to deliver third-order accuracy because the stencil is no longer regular at boundary nodes. The degraded accuracy at boundary nodes do not affect the errors in the interior nodes due to the upwind nature of the advection equation as discussed in details in Ref.[19].

Results for the irregular grids are shown in Figure 12. As expected, both the point quadrature and the regular formula fail to achieve third-order accuracy. All other formulas successfully delivers third-order accuracy. It is observed that error magnitudes obtained with the point quadrature and the regular formula are close to those obtained with other formulas, but the order of convergence deteriorates on fine grids. The large difference in the error level between the regular and irregular grids for the point quadrature seems to relate to the stencil size of the LSQ fit; the error level on the regular grids is much smaller if the LSQ fit includes neighbors of the neighbors, instead of the neighbors only.

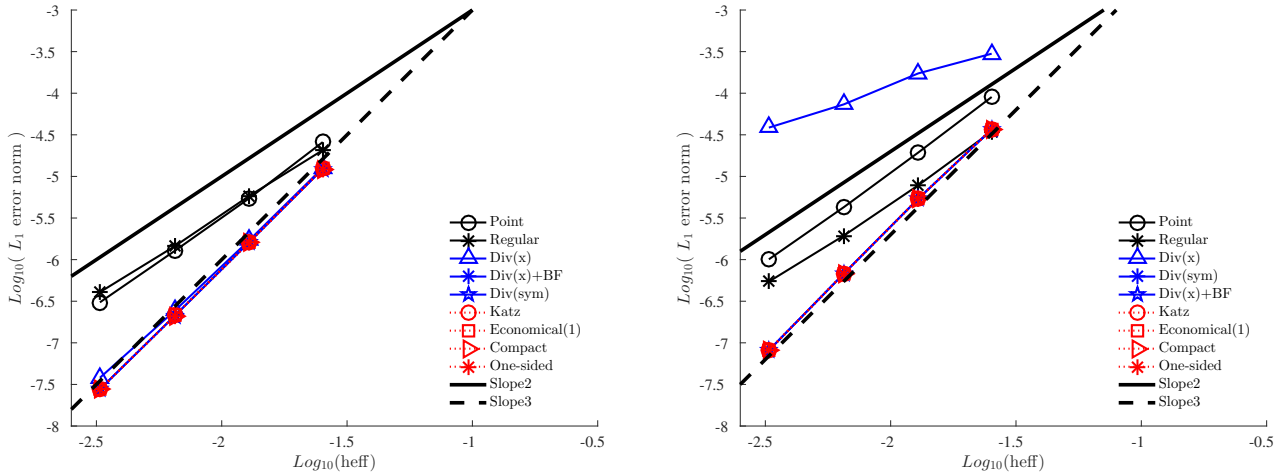
For the same regular and irregular grids, similar results have been obtained for Burgers' equation as shown in Figures 13 and 14. It is noted that the compatible source discretizations not only yield third-order accuracy, but also give significantly lower errors in comparison to the errors of solutions with the incompatible formulas, i.e., the point quadrature and the regular formula, even on irregular grids in contrast to the irregular-grid results in the linear case.

To demonstrate that the elimination of the first-order error is not sufficient to achieve third-order accuracy, we consider the linear advection problem, and computed the truncation and discretization errors with an incompatible formula: $a_L = a_R = 1$, $b_L = -1$, $b_R = c_L = c_R = 0$ on irregular grids. This formula, referred to as F1, satisfies the consistency condition (5.12) and the zero first-order truncation error condition (5.13), but does not satisfy the compatibility condition (5.19). The truncation errors are computed at interior nodes



(a) Irregular grid. (b) Solution contours.

Figure 16: Irregular grid with 33×33 nodes and exact solution contours.



(a) Interior nodes. (b) Outflow boundary nodes.

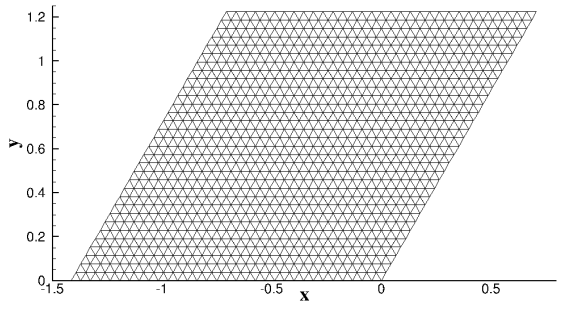
Figure 17: Error convergence results for Burgers' equation on irregular triangular grids with a curved outflow boundary. Div(x)+BF is the one-component divergence formulation with the boundary flux closure; no boundary flux contributions are added to others.

on each grid by computing the residuals with the exact solution values given at nodes. The L_1 norm of the residuals is used as a measure of the truncation error. The same is performed with the compatible formula 'Compact', and the results are shown in Figure 15. Observe that the truncation errors are second-order with both formulas, but the formula F1 yields second-order discretization errors while the compatible formula gives third-order discretization errors. The results show that the second-order truncation error does not guarantee third-order accuracy on irregular grids.

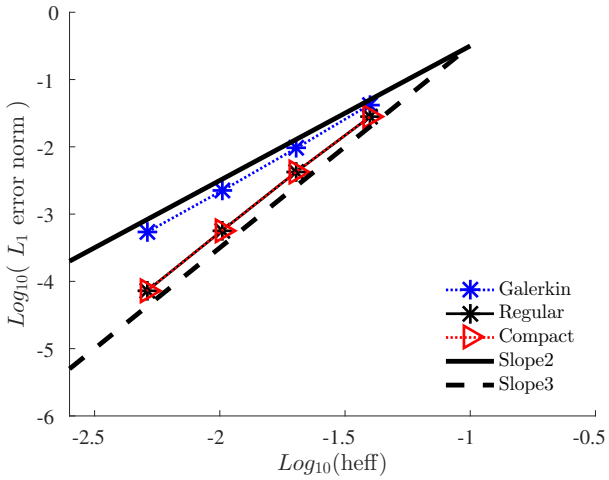
7.2.2 Curved boundary

To investigate the effect of curved boundaries, we consider a problem in a curved domain. The domain is bounded by two co-centric circles of radii 0.25 and 1.0, and x- and y-axes. We consider Burgers' equation with the same exact solution as before. Irregular grids are generated with 33×33, 65×65, 129×129, and 257×257 nodes. The outer circular boundary is taken as outflow, and a weak boundary condition is applied as described in the previous section. Note that the nodal distribution at the outflow boundary is perturbed, and thus the grids are fully irregular through the outflow boundary. A representative grid and solution contours are shown in Figure 16. It is emphasized that the third-order edge-based scheme is already known to achieve third-order accuracy at curved boundaries with straight edges. It still requires quadratically-accurate normal vectors at boundary nodes for boundary conditions involving boundary normals, but the outflow condition here does not require the boundary normals in the definition of the right state that goes into the numerical flux at a boundary node. Note also that a curved boundary representation is required for evaluating integral quantities to higher-order, but it is a matter of post-processing and only requires a high-order boundary representation (not a high-order grid over the entire domain). See Ref.[19] for details on the treatment of curved boundaries. Here, we focus on the effect of source term discretizations on the order of convergence at a curved boundary.

Numerical results are shown in Figure 17. For this problem, the one-component divergence formulation is applied also with the accuracy-preserving boundary flux quadrature formula [19], indicated Div(x)-BF. At interior nodes, all formulas except the point quadrature and the regular formula yield third-order accuracy as expected. See Figure 17(a). On the other hand, at boundary nodes, the one-component divergence formulation without the boundary closure loses third-order accuracy while it gives third-order accuracy with the boundary closure. It confirms the necessity of the boundary closure for the one-component form. Here again, we observe errors worse than first-order accuracy in the case of no boundary closure. Somewhat surprisingly, the symmetric divergence formula without the boundary closure yields third-order accuracy at boundary nodes. It indicates that the errors of $O(h^2)$ in the source discretization are tolerated for third-order accuracy. For all other formulas, the results confirm that the boundary closure is not needed to deliver third-order accuracy at boundary nodes.



(a) Regular equilateral triangular grid with 33×33 nodes.



(b) Error convergence results for interior nodes.

Figure 18: Grid and error convergence results for the equilateral-triangular-grid test.

7.2.3 Equilateral Triangular Grids

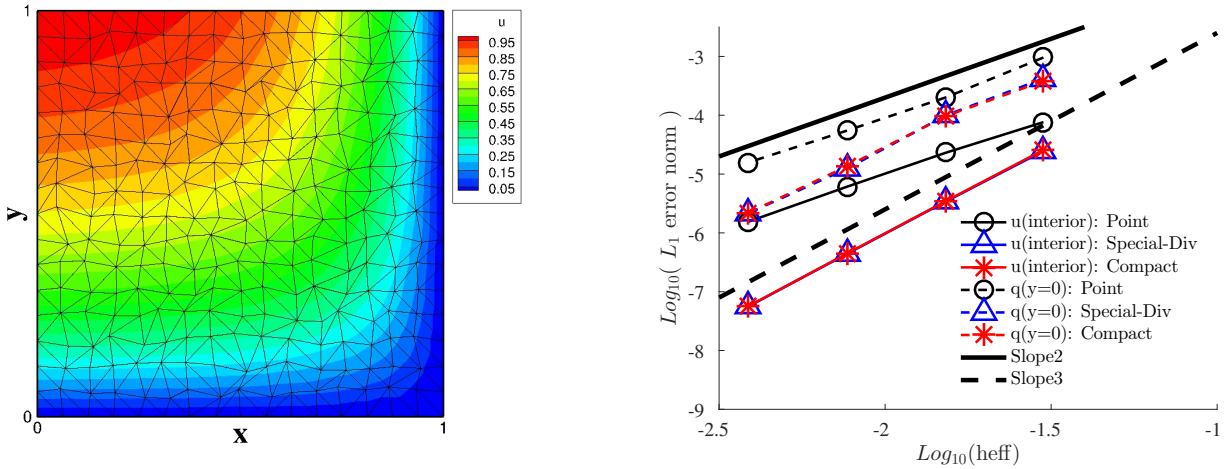
Some representative formulas have been tested for the linear advection problem on regular equilateral-triangular grids with 33×33, 65×65, 129×129, and 257×257 nodes. Figure 18(a) shows the coarsest grid. The top and right boundaries are treated as outflow as before. The regular and compact formulas are tested and compared with the Galerkin source discretization ($a_L = a_R = 1$ and $b_L = b_R = c_L = c_R = 0$). Error convergence results for interior nodes are shown in Figure 18(b). Both the regular and compact formulas give third-order accuracy while the Galerkin discretization does not. Very similar results have been obtained at outflow boundaries, and therefore not shown.

These results show that the Galerkin source discretization is not compatible with the third-order edge-based scheme. The first author stated in Ref.[27] that it is known that the Galerkin source discretization gives a compatible discretization on regular equilateral-triangular grids, citing Ref.[13], but he did not perform any numerical experiment to confirm the statement and these results force him to admit that the statement is false.

7.2.4 Advection-Diffusion

To compare the new source formulas and the special divergence formulation, Equation (4.21) with the source flux (4.19), we consider the advection-diffusion system defined by the flux (7.7) and the source term (7.8) in a square domain with the irregular triangular grids used in Section 7.2.1. The exact solution is a boundary-layer-type solution (7.9). For the parameters $(a, b) = (2.73, -1.31)$ and $\sqrt{a^2 + b^2}/\nu = 5$, the exact solution exhibits thick boundary layers towards $x = 1$ and $y = 0$ as shown in Figure 19(a) for a 17×17 grid. The advection-diffusion system is discretized by the edge-based scheme as described in details in Ref.[29]; one of the two schemes proposed in Ref.[29] called Scheme-II is used. Note that this scheme is known to yield the same order of accuracy in all variables $(u, p = \partial_x u, q = \partial_y u)$ on irregular grids [19, 29]. The solution and a gradient variable corresponding to the tangential gradient are specified at boundary nodes: (u, p) specified at $y = 0$ and $y = 1$, and (u, q) specified at $x = 0$ and $x = 1$. The tangential gradient can be taken as known since it can be obtained from the solution along the boundary given as a boundary condition. At corner nodes, both p and q are specified. The remaining variable, which represents the gradient normal to the boundary is computed by the numerical scheme with a weak boundary condition implemented with the accuracy-preserving boundary flux quadrature [19]. For the source term, $\mathbf{s} = (-p, -q)$, we consider the point quadrature, Compact, and the special divergence formulation (4.19). In the case of the special divergence formulation, the source terms are discretized by the edge-based scheme with an upwind flux (see Ref.[29]), and with the accuracy-preserving boundary flux quadrature. Further details can be found in Ref.[29].

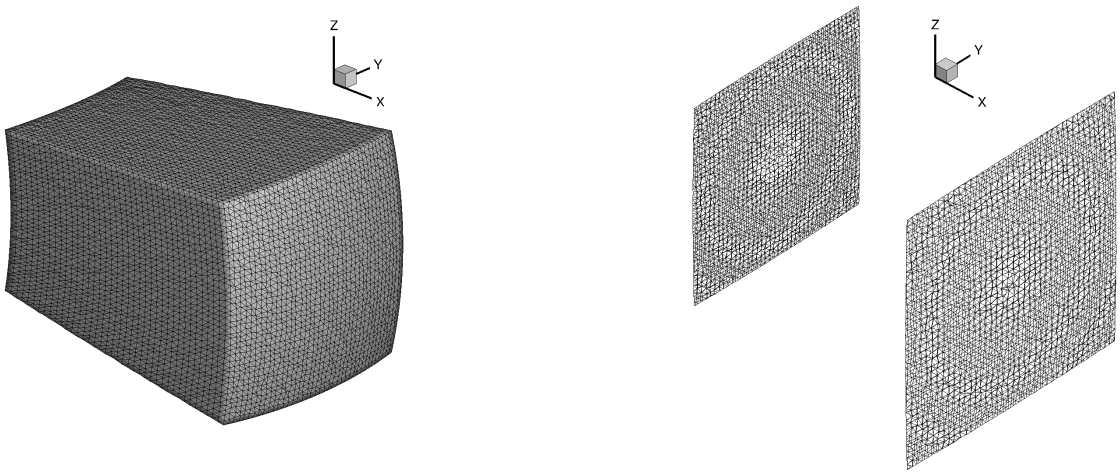
Error convergence results are shown in Figure 19(b), where the special divergence formulation is indicated as Special-Div. For the solution u, third-order accuracy is achieved by Special-Div and Compact, but second-order accuracy is observed for Point. Similar results are obtained for the variable q at the bottom boundary, which corresponds to the normal gradient. These results confirm that the source term discretization has a significant impact on the third-order edge-based scheme for equations with solution-dependent source terms. It is also noted that the newly derived formula, Compact, is an efficient alternative to the special divergence formulation because it does not require the boundary closure and also does not depend on the LSQ gradients at neighbor nodes.



(a) Solution contours on a coarse 17×17-node grid.

(b) Error convergence.

Figure 19: Error convergence results for the advection-diffusion problem.



(a) Irregular tetrahedral grid (40×40×40).

(b) Sections of the irregular tetrahedral grid.

Figure 20: Irregular grid and representative sections.

7.3 Three Dimensions

In three dimensions, the accuracy-preserving formulas are verified for the Euler equations defined by the flux tensor:

$$\mathbf{f} = \begin{bmatrix} \rho\mathbf{v} \\ \rho\mathbf{v}\otimes\mathbf{v} + p\mathbf{I} \\ \rho\mathbf{v}H \end{bmatrix}, \tag{7.10}$$

where $\otimes$ denotes the dyadic product, $\mathbf{I}$ is the 3×3 identity matrix, ρ is the density, $\mathbf{v} = (u, v, w)$ is the velocity vector, p is the pressure, and $H = (\gamma p/(\gamma - 1) + \rho\mathbf{v}^2/2)/\rho$ is the specific total enthalpy with $\gamma = 1.4$ (air). The system is nondimensionalized by free stream values as described in Ref.[2], and closed by the ideal gas law:

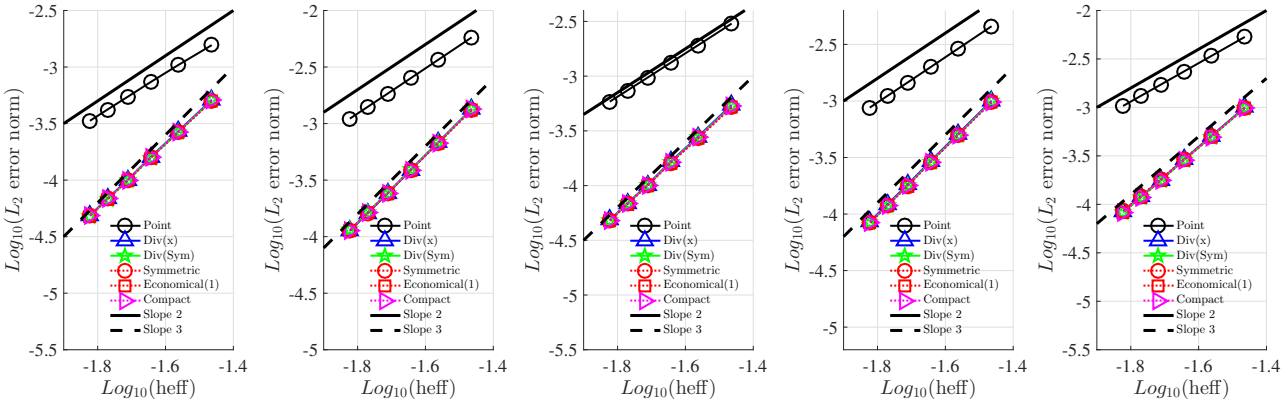
$$p = \rho T/\gamma, \tag{7.11}$$

where T is the temperature. The source term vector $\mathbf{s}$ is defined such that the following functions are the exact solutions:

$$u/M_\infty = 6.0 + \sin(\pi(x + 0.6y + 0.4z)), \tag{7.12}$$
$$v/M_\infty = 2.0 + \sin(\pi(0.6x + y + 0.4z)), \tag{7.13}$$
$$w/M_\infty = 2.0 + \sin(\pi(0.6x + 0.4y + z)), \tag{7.14}$$
$$p/M_\infty = 2.0 + \sin(\pi(0.8x + 0.6y + 0.4z)), \tag{7.15}$$
$$T/M_\infty = 2.0 + \sin(\pi(0.6x + 0.4y + 0.4z)), \tag{7.16}$$

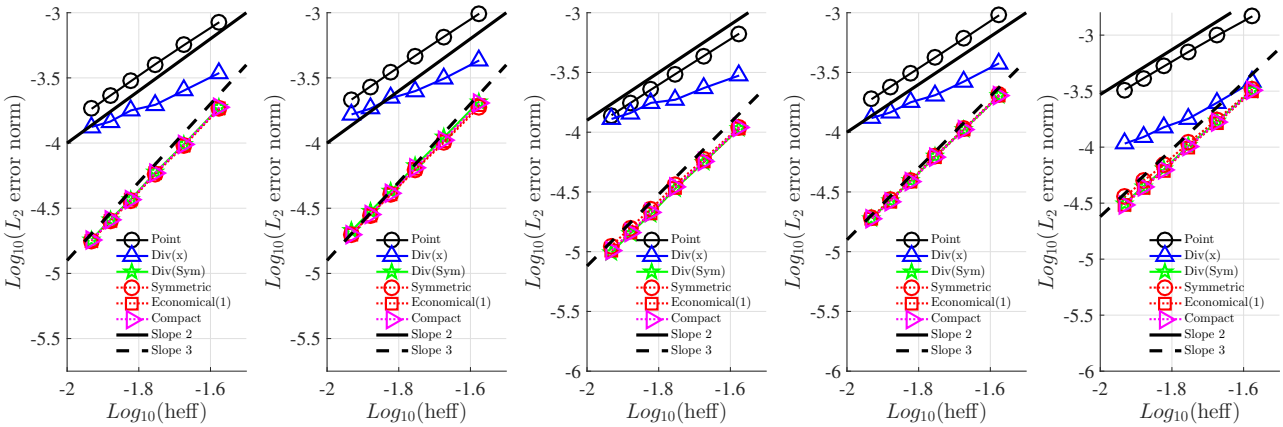
where $M_\infty = 1.2$. The parameters have been chosen such that the flow is rendered supersonic and a weak boundary condition can be implemented by a simple extrapolation.

The accuracy-preserving source discretization formulas have been implemented and verified in NASA's FUN3D code [2], which is a well-validated three-dimensional unstructured-grid solver developed by NASA Langley Research Center, and where the third-order edge-based scheme is already implemented for the Euler equations [16]. The numerical flux is the Roe flux [41] with linearly-extrapolated left and right fluxes as in Equation (2.4). All results have been obtained with the same third-order edge-based inviscid scheme, so that any difference in the results will be due to the effects of source term discretizations. In three dimensions, we consider the following formulas: Div(x), Div(sym), Symmetric, Economical(1), and Compact from Table 1. No boundary closure is implemented for these formulas. Results obtained by the Katz formula (i.e., $a_L = a_R = 1, b_L = b_R = -1/2, c_L = c_R = -1/8$) are omitted below, but it has been confirmed numerically that it does not yield third-order accuracy on tetrahedral grids as expected from the analysis in Section 5.



(a) Density, ρ. (b) Velocity, u. (c) Velocity, v. (d) Velocity, w. (e) Pressure, p.

Figure 21: Error convergence results for interior nodes in the three-dimensional Euler test case.



(a) Density, ρ. (b) Velocity, u. (c) Velocity, v. (d) Velocity, w. (e) Pressure, p.

Figure 22: Error convergence results for boundary nodes in the three-dimensional Euler test case.

The domain is a curved cube as shown in Figure 20(a). Irregular tetrahedral grids have been generated from regular tetrahedral grids of n^3 nodes, where $n = 40, 50, 60, 70, 80, 90$, in a cubic domain by nodal perturbations and a mapping onto the curved domain. The curved boundary facing the positive x-direction is taken as outflow, and a weak supersonic boundary condition is applied with a ghost state set by the solution at the boundary node j: $u_b = u_j$. At boundary nodes, the accuracy-preserving boundary flux quadrature is applied to the flux balance [19]. It is emphasized, again, that third-order accuracy on linear tetrahedral grids for curved geometries has already been demonstrated in Ref.[16]; here we focus on the effects of the source term discretization on third-order accuracy. See Ref.[16] for more details on the implementation of the weak boundary condition and third-order accuracy on curved geometries. The discretization errors in the primitive variables are measured in the L_2 norm separately for interior and boundary nodes. The effective mesh spacing is computed, separately for the interior and boundary nodes, as the L_2 norm of the cubic root of the dual control volumes.

Error convergence results for interior nodes are shown in Figure 21. As can be seen, third-order accuracy is completely lost with the point source evaluation. On the other hand, third-order accuracy is obtained with all other source discretization formulas: Div(x), Div(sym), Symmetric, Economical(1), and Compact, achieving nearly the same error levels. Figure 22 shows the error convergence results for boundary nodes. Again, the point source evaluation cannot achieve third-order accuracy. Also, the one-component divergence formulation, Div(x), leads to accuracy deterioration because no boundary closure is implemented. However, as in two dimensions, the symmetric divergence formulation, Div(sym), achieves third-order accuracy even with $O(h^2)$ committed by not implementing the boundary closure. It indicates that $O(h^2)$ is tolerated also in three dimensions. All other formulas, Symmetric, Economical(1), and Compact, successfully achieve third-order accuracy without any boundary contribution.

8 Concluding Remarks

A new approach was proposed to deriving accuracy-preserving source term discretizations for the third-order edge-based discretization in one, two, and three dimensions. Requirements for third-order accuracy have been identified and a family of source quadrature formulas has been derived. In all dimensions, a one-parameter family of formulas has been discovered that does not require second derivatives of source terms. An important finding is that there exists a unique formula in the one-parameter family, which does not require gradients at neighbor nodes also. In this case, the source term discretization stencil matches the LSQ stencil, and can be compact if the LSQ fit is performed within a compact stencil. It has also been shown that the new formulas do not require boundary closure in discretizations at boundary nodes, and therefore can be easily applied at boundary nodes. Numerical results have been presented to demonstrate third-order accuracy in one, two, and three dimensions, including domains with curved boundaries.

Acknowledgments

This work has been funded by NASA under Contract No. NNL09AA00A with Dr. Veer N. Vatsa as the technical monitor. The first author gratefully acknowledges support from Software CRADLE. The authors would like to thank Boris Diskin (National Institute of Aerospace) for valuable comments and discussions.

References

[1] T. J. Barth. Numerical aspects of computing viscous high Reynolds number flows on unstructured meshes. AIAA Paper 91-0721, 1991.

[2] R. T. Biedron, J.-R. Carlson, J. M. Derlaga, P. A. Gnoffo, D. P. Hammond, W. T. Jones, B. Kleb, E. M. Lee-Rausch, E. J. Nielsen, M. A. Park, C. L. Rumsey, J. L. Thomas, and W. A. Wood. FUN3D manual: 12.9. *NASA-TM–2016–219012*, February 2016.

[3] Y. Nakashima, N. Watanabe, and H. Nishikawa. Development of an effective implicit solver for general-purpose unstructured CFD software. In *The 28th Computational Fluid Dynamics Symposium*, C08-1, Tokyo, Japan, 2014.

[4] N. Kroll, M. Abu-Zurayk, D. Dimitrov, T. Franz, T. Führer, T. Gerhold, S. Görtz, R. Heinrich, C. Ilic, J. Jepsen, J. Jägersküpper, M. Kruse, A. Krumbein, S. Langer, D. Liu, R. Liepelt, L. Reimer, M. Ritter, A. Schwöppe, J. Scherer, F. Spiering, R. Thormann, V. Togiti, D. Vollmer, and J.-H. Wendisch. Dlr project digital-x: Towards virtual aircraft design and flight testing based on high-fidelity methods. *CEAS Aeronautical Journal*, 7(1):3–27, March 2016.

[5] D. J. Mavriplis and M. Long. NSU3D Results for the Fourth AIAA Drag Prediction Workshop. AIAA Paper 2010-4363, 2010.

[6] H. Luo, J. D. Baum, and R. Löhner. High-reynolds number viscous flow computations using an unstructured-grid method. In *Proc. of 42nd AIAA Aerospace Sciences Meeting*, AIAA Paper 2004-1103, Reno, Nevada, 2004.

[7] T. Kozubskaya, I. Abalakin, A. Dervieux, and H. Ouvrard. Accuracy improvement for finite-volume vertex-centered schemes solving aeroacoustics problems on unstructured meshes. In *Proc. of 16th AIAA/CEAS Aeroacoustics Conference*, AIAA Paper 2010-3933, 2010.

[8] A. Haselbacher and J. Blazek. Accurate and efficient discretization of Navier-Stokes equations on mixed grids. *AIAA J.*, 38(11):2094–2102, 2000.

[9] T. Smith, C. Ober, and A. Lorber. SIERRA/Premo-A new general purpose compressible flow simulation code. In *Proc. of 32nd AIAA Fluid Dynamics Conference and Exhibit*, AIAA Paper 2002-3292, St. Louis, MO, 2002.

[10] P. Eliasson. EDGE, a Navier-Stokes solver, for unstructured grids. Technical Report FOI-R-0298-SE, Swedish Defence Research Agency, December 2001.

[11] L. Fezoui and B. Stouffle. A class of implicit upwind schemes for Euler simulations with unstructured meshes. *J. Comput. Phys.*, 84:174206, 1989.

[12] Song Gao, Wagdi G. Habashi, Marco Fossati, Dario Isola, and Guido S. Baruzzi. Finite-element formulation of a Jacobian-free solver for supersonic viscous flows on hybrid grids. In *55th AIAA Aerospace Sciences Meeting*, AIAA Paper 2017-0085, Grapevine, Texas, 2017.

[13] A. Katz and V. Sankaran. Mesh quality effects on the accuracy of CFD solutions on unstructured meshes. *J. Comput. Phys.*, 230:7670–7686, 2011.

[14] A. Katz and V. Sankaran. An efficient correction method to obtain a formally third-order accurate flow solver for node-centered unstructured grids. *J. Sci. Comput.*, 51:375–393, 2012.

[15] B. Diskin and J. L. Thomas. Effects of mesh regularity on accuracy of finite-volume schemes. In *Proc. of 50th AIAA Aerospace Sciences Meeting*, AIAA Paper 2012-0609, Nashville, Tennessee, 2012.

[16] Yi Liu and Hiroaki Nishikawa. Third-order inviscid and second-order hyperbolic Navier-Stokes solvers for three-dimensional inviscid and viscous flows. In *46th AIAA Fluid Dynamics Conference*, AIAA Paper 2016-3969, Washington, D.C., 2016.

[17] Yi Liu and Hiroaki Nishikawa. Third-order inviscid and second-order hyperbolic Navier-Stokes solvers for three-dimensional unsteady inviscid and viscous flows. In *55th AIAA Aerospace Sciences Meeting*, AIAA Paper 2017-0081, Grapevine, Texas, 2017.

[18] Y. Nakashima, N. Watanabe, and H. Nishikawa. Hyperbolic Navier-Stokes solver for three-dimensional flows. In *54th AIAA Aerospace Sciences Meeting*, AIAA Paper 2016-1101, San Diego, CA, 2016.

[19] H. Nishikawa. Accuracy-preserving boundary flux quadrature for finite-volume discretization on unstructured grids. *J. Comput. Phys.*, 281:518–555, 2015.

[20] O. Tong, A. Katz, Y. Yanagita, A. Casey, and R. Schaap. High-order methods for turbulent flows on three-dimensional strand grids. *J. Sci. Comput.*, pages 1–19, 2015.

[21] A. Katz and Dalon Work. High-order flux correction/finite difference schemes for strand grids. *J. Comput. Phys.*, 282:360–380, 2015.

[22] H. Nishikawa and Yi Liu. Hyperbolic Navier-Stokes method for high-Reynolds-number boundary-layer flows. In *55th AIAA Aerospace Sciences Meeting*, AIAA Paper 2017-0738, Grapevine, Texas, 2017.

[23] C. Corre, G. Hanss, and A. Lerat. A residual-based compact schemes for the unsteady compressible Navier-Stokes equations. *Comput. Fluids*, 34:561–580, 2005.

[24] H. Nishikawa and P. L. Roe. On high-order fluctuation-splitting schemes for Navier-Stokes equations. In C. Groth and D. W. Zingg, editors, *Computational Fluid Dynamics 2004*, pages 799–804. Springer-Verlag, 2004.

[25] B. Pincock and A. Katz. High-order flux correction for viscous flows on arbitrary unstructured grids. *J. Sci. Comput.*, 61:454–476, 2014.

[26] A. Katz. Source term discretizations for the linear and corrected Galerkin schemes. Unpublished, 2012.

[27] H. Nishikawa. Divergence formulation of source term. *J. Comput. Phys.*, 231:6393–6400, 2012.

[28] J. Thorne and A. Katz. Source term discretization effects on the steady-state accuracy of finite volume schemes. *J. Sci. Comput.*, 69:146–169, 2016.

[29] H. Nishikawa. First, second, and third order finite-volume schemes for advection-diffusion. *J. Comput. Phys.*, 273:287–309, 2014.

[30] B. Diskin. Notes on a compact third order edge-based node-centered scheme. Unpublished, March 2011.

[31] B. Diskin and J. L. Thomas. Accuracy analysis for mixed-element finite-volume discretization schemes. *NIA Report No. 2007-08*, 2007.

[32] H. Nishikawa. Beyond interface gradient: A general principle for constructing diffusion schemes. In *Proc. of 40th AIAA Fluid Dynamics Conference and Exhibit*, AIAA Paper 2010-5093, Chicago, 2010.

[33] B. Diskin and J. L. Thomas. Notes on accuracy of finite-volume discretization schemes on irregular grids. *Appl. Numer. Math.*, 60:224–226, 2010.

[34] Marie-Gabrielle Vallet Julien Dompierre, Paul Labb and Ricardo Camarero. How to subdivide pyramids, prisms, and hexahedra into tetrahedra. In *Proc. of the 8th International Meshing Roundtable*, pages 195–204, 1999.

[35] H. Nishikawa. First, second, and third order finite-volume schemes for Navier-Stokes equations. In *Proc. of 7th AIAA Theoretical Fluid Mechanics Conference, AIAA Aviation and Aeronautics Forum and Exposition 2014*, AIAA Paper 2014-2091, Atlanta, GA, 2014.

[36] H. Nishikawa. First-, second-, and third-order finite-volume schemes for diffusion. *J. Comput. Phys.*, 256:791–805, 2014.

[37] H. Nishikawa. Alternative formulations for first-, second-, and third-order hyperbolic Navier-Stokes schemes. In *Proc. of 22nd AIAA Computational Fluid Dynamics Conference*, AIAA Paper 2015-2451, Dallas, TX, 2015.

[38] J. C. Tannehill, D. A. Anderson, and R. H. Pletcher. *Computational Fluid Mechanics and Heat Transfer*. Taylor & Francis, second edition, 1997.

[39] Katate Masatsuka. I do like CFD, VOL.1, Second Edition, version 2.3. `http://www.cfdbooks.com`, 2016.

[40] B. Gustafsson. The convergence rate for difference approximations to mixed initial boundary value problems. *Math. Comput.*, 29(130):396–406, 1975.

[41] P. L. Roe. Approximate Riemann solvers, parameter vectors, and difference schemes. *J. Comput. Phys.*, 43:357–372, 1981.

Appendix A: Truncation Errors on Regular Grids

We derive the truncation errors of the third-order edge-based scheme and the source term discretization on regular simplex (triangular and tetrahedral) grids. Regular grids are defined as those having an identical compact stencil (i.e., the stencil composed of edge-connected neighbors) at all interior nodes. The stencil is symmetric with respect to the central node and invariant under translation, i.e., for any edge-connected interior nodes j and k, the stencil at j translated along the edge $[jk]$ coincides with the stencil at k. Examples of regular simplex grids are shown in Figures 3, 4, 10(a), and 18(a). Note that regular simplex grids remain regular under any linear transformation (scaling, rotation, translation). Applying such transformations, one can obtain highly anisotropic and/or highly skewed regular grids.

A.1 Truncation error for the conservation law

Consider the Taylor expansions of the fluxes $\mathbf{f}$ satisfying div$\mathbf{f} = 0$ and LSQ gradients at edge-connected nodes j and k. The expansions are computed at node j:

$$\mathbf{f}_k = \mathbf{f}_j + \partial_{jk}\mathbf{f}_j + \frac{1}{2}\,\partial^2_{jk}\mathbf{f}_j + \frac{1}{6}\,\partial^3_{jk}\mathbf{f}_j + O(h^4), \tag{A.1}$$

$$\hat{\nabla}\mathbf{f}_k = \nabla\mathbf{f}_k + \mathcal{E}(\hat{\nabla}\mathbf{f}_k) = \nabla\mathbf{f}_j + \partial_{jk}(\nabla\mathbf{f}_j) + \frac{1}{2}\,\partial^2_{jk}(\nabla\mathbf{f}_j) + \mathcal{E}(\hat{\nabla}\mathbf{f}_j) + O(h^3), \tag{A.2}$$

$$\hat{\nabla}\mathbf{f}_j = \nabla\mathbf{f}_j + \mathcal{E}(\hat{\nabla}\mathbf{f}_j) + O(h^3). \tag{A.3}$$

where $\mathcal{E}(\hat{\nabla}\mathbf{f}_k)$ and $\mathcal{E}(\hat{\nabla}\mathbf{f}_j)$ denote the tensors of the local truncation errors of the nodal gradients at k and j, respectively. The truncation errors have magnitude of $O(h^2)$, as the third-order edge-based scheme requires nodal gradients to be exact for quadratic fluxes. Note that the truncation errors of the nodal gradients take the same form at all nodes on regular grids, and thus the leading term of the Taylor expansion of $\mathcal{E}(\hat{\nabla}\mathbf{f}_k)$ around j matches $\mathcal{E}(\hat{\nabla}\mathbf{f}_j)$ as implied in Equation (A.2). Substituting them into the left and right fluxes (2.4), we obtain

$$\mathbf{f}_L = \mathbf{f}_j + \frac{1}{2}\,\hat{\nabla}\mathbf{f}_j \cdot \Delta\mathbf{x}_{jk} = \mathbf{f}_j + \frac{1}{2}\,\partial_{jk}\mathbf{f}_j + \frac{1}{2}\,\mathcal{E}(\hat{\nabla}\mathbf{f}_j)\cdot\Delta\mathbf{x}_{jk} + O(h^4), \tag{A.4}$$

$$\mathbf{f}_R = \mathbf{f}_k - \frac{1}{2}\,\hat{\nabla}\mathbf{f}_k \cdot \Delta\mathbf{x}_{jk} = \mathbf{f}_j + \frac{1}{2}\,\partial_{jk}\mathbf{f}_j - \frac{1}{12}\,\partial^3_{jk}\mathbf{f}_j - \frac{1}{2}\,\mathcal{E}(\hat{\nabla}\mathbf{f}_j)\cdot\Delta\mathbf{x}_{jk} + O(h^4). \tag{A.5}$$

Therefore, the average flux in the numerical flux (2.3) becomes

$$\frac{1}{2}\,(\mathbf{f}_L + \mathbf{f}_R) = \mathbf{f}_j + \frac{1}{2}\,\partial_{jk}\mathbf{f}_j - \frac{1}{24}\,\partial^3_{jk}\mathbf{f}_j + O(h^4). \tag{A.6}$$

The dissipation term, $|a_n|(u_R - u_L)|\mathbf{n}_{jk}|$, will locally generate an $O(h^2)$ contribution to the overall truncation error of the edge-based scheme. However, it cancels out when summed over the edges by symmetry of regular stencils [30]. To see this, consider the Taylor expansion of $|a_n|(u_R - u_L)|\mathbf{n}_{jk}|$:

$$|a_n|(u_R - u_L)|\mathbf{n}_{jk}| = -|a_n|_j \left(\frac{1}{12}\,\partial^3_{jk}u_j + \mathcal{E}(\hat{\nabla}u_j)\cdot\Delta\mathbf{x}_{jk}\right)|\mathbf{n}_{jk}| + O(h^{3+D}), \tag{A.7}$$

where $D = 2$ in two dimensions and $D = 3$ in three dimensions. The terms in the parenthesis are of $O(h^3)$, and proportional to cubic products of the components of $\Delta\mathbf{x}_{jk}$. On regular grids, each edge in the stencil at node j has a symmetric edge with respect to the node j, having the same $|\mathbf{n}_{jk}|$. For such a pair of symmetric edges, these terms will have the same magnitude with the opposite signs. Therefore, the leading error in the dissipation term vanishes when summed over all edges in the stencil. As a result, the Taylor expansion of the edge-based scheme for the conservation law is given by

$$\sum_{k\in\{k_j\}} \phi_{jk}(\mathbf{n}_{jk}) = \sum_{k\in\{k_j\}} \mathbf{f}_j\cdot\mathbf{n}_{jk} + \sum_{k\in\{k_j\}} \frac{1}{2}\,\partial_{jk}\mathbf{f}_j\cdot\mathbf{n}_{jk} - \sum_{k\in\{k_j\}} \frac{1}{24}\,\partial^3_{jk}\mathbf{f}_j\cdot\mathbf{n}_{jk} + O(h^{3+D}). \tag{A.8}$$

The first term vanishes because $\sum_{k\in\{k_j\}} \mathbf{n}_{jk} = 0$. The second term simplifies to $(\mathrm{div}\mathbf{f}_j)V_j$ by the identity that holds on arbitrary simplex grids:

$$\frac{1}{2}\sum_{k\in\{k_j\}} \Delta\mathbf{x}_{jk}\otimes\mathbf{n}_{jk} = V_j\,\mathbf{I}, \tag{A.9}$$

where $\otimes$ denotes the dyadic product, and $\mathbf{I}$ is the $D\times D$ identity matrix. Therefore, we obtain

$$\frac{1}{V_j}\sum_{k\in\{k_j\}} \phi_{jk}(\mathbf{n}_{jk}) \quad = \quad \mathrm{div}\mathbf{f}_j - \sum_{k\in\{k_j\}} \frac{1}{24V_j}\,\partial^3_{jk}\mathbf{f}_j\cdot\mathbf{n}_{jk} + O(h^3), \tag{A.10}$$

for general simplex grids. The second term on the right hand side is the second-order truncation error, and it vanishes for exact solutions satisfying $\mathrm{div}\mathbf{f} = 0$. To see this, we first consider a regular triangular grid. The second-order truncation error term, denoted by $\mathcal{T}$, is given by

$$\mathcal{T} = -\frac{1}{24V_j}\sum_{k\in\{k_j\}} \left[\Delta x^3\partial_{xxx} + 3\Delta x^2\Delta y\partial_{xxy} + 3\Delta x\Delta y^2\partial_{xyy} + \Delta y^3\partial_{yyy}\right](\mathbf{f}_j\cdot\mathbf{n}_{jk}), \tag{A.11}$$

where $\Delta\mathbf{x}_{jk} = (\Delta x, \Delta y)$. On regular triangular grids, the following identities hold:

$$\sum_{k\in\{k_j\}} n_x\Delta y^3 = \sum_{k\in\{k_j\}} n_y\Delta x^3 = 0, \quad \sum_{k\in\{k_j\}} n_x\Delta x^2\Delta y = \sum_{k\in\{k_j\}} n_y\Delta x\Delta y^2 \tag{A.12}$$

$$\sum_{k\in\{k_j\}} n_y\Delta y^3 = 3\sum_{k\in\{k_j\}} n_x\Delta x\Delta y^2, \quad \sum_{k\in\{k_j\}} n_x\Delta x^3 = 3\sum_{k\in\{k_j\}} n_y\Delta x^2\Delta y, \tag{A.13}$$

where $\mathbf{n}_{jk} = (n_x, n_y)$, by which the second-order truncation error can be factored as

$$\mathcal{T} = -\frac{1}{24V_j}\sum_{k\in\{k_j\}} \left[n_x\Delta x^3\partial_{xx} + 3n_x\Delta x^2\Delta y\partial_{xy} + n_y\Delta y^3\partial_{yy}\right](\mathrm{div}\mathbf{f})_j. \tag{A.14}$$

It is expressed in Equation (3.2) as

$$\mathcal{T} = -\frac{1}{24V_j}\left[Q_{xx} + Q_{yy} + Q_{xy}\right](\mathrm{div}\,\mathbf{f})_j, \tag{A.15}$$

where Q_{xx}, Q_{yy}, and Q_{xy} are defined as

$$Q_{xx} = \sum_{k\in\{k_j\}} n_x\Delta x^3\partial_{xx}, \quad Q_{yy} = \sum_{k\in\{k_j\}} n_y\Delta y^3\partial_{yy}, \quad Q_{xy} = \sum_{k\in\{k_j\}} 3n_x\Delta x^2\Delta y\,\partial_{xy}. \tag{A.16}$$

Therefore, the second-order truncation error vanishes for exact solutions satisfying $\mathrm{div}\mathbf{f} = 0$, and the leading truncation error becomes $O(h^3)$. For a regular grid with right isosceles triangles as in Figure 3, the second-order truncation error $\mathcal{T}$ reduces to the one in Equation (3.4); and for a regular grid with equilateral triangles as in Figure 18(a), it reduces to

$$\mathcal{T} \quad = \quad -\frac{h^2}{16}(\partial_{xx} + \partial_{yy})(\mathrm{div}\,\mathbf{f})_j, \tag{A.17}$$

where h is the side length of the equilateral triangles.

In three dimensions, the second-order truncation error in Equation (A.10) is given by

$$\mathcal{T} = -\frac{1}{24V_j}\sum_{k\in\{k_j\}} \left[\Delta x\partial_x + \Delta y\partial_y + \Delta z\partial_z\right]^3(\mathbf{f}_j\cdot\mathbf{n}_{jk}), \tag{A.18}$$

where $\Delta\mathbf{x}_{jk} = (\Delta x, \Delta y, \Delta z)$. On regular tetrahedral grids, the following identities hold:

$$\sum_{k\in\{k_j\}} n_y\Delta x^3 = \sum_{k\in\{k_j\}} n_z\Delta x^3 = 0, \tag{A.19}$$

$$\sum_{k\in\{k_j\}} n_z\Delta y^3 = \sum_{k\in\{k_j\}} n_x\Delta y^3 = 0, \tag{A.20}$$

$$\sum_{k\in\{k_j\}} n_x\Delta z^3 = \sum_{k\in\{k_j\}} n_y\Delta z^3 = 0, \tag{A.21}$$

$$3\sum_{k\in\{k_j\}} n_x\Delta y^2\Delta z = 3\sum_{k\in\{k_j\}} \Delta y\Delta z^2 n_x = 0, \tag{A.22}$$

$$3\sum_{k\in\{k_j\}} n_y\Delta z^2\Delta x = 3\sum_{k\in\{k_j\}} n_y\Delta z\Delta x^2 = 0, \tag{A.23}$$

$$3\sum_{k\in\{k_j\}} n_z\Delta x^2\Delta y = 3\sum_{k\in\{k_j\}} n_z\Delta x\Delta y^2 = 0, \tag{A.24}$$

$$\sum_{k\in\{k_j\}} n_x\Delta x^3 = 3\sum_{k\in\{k_j\}} n_y\Delta x^2\Delta y = 3\sum_{k\in\{k_j\}} n_z\Delta x^2\Delta z, \tag{A.25}$$

$$\sum_{k\in\{k_j\}} n_y\Delta y^3 = 3\sum_{k\in\{k_j\}} n_z\Delta y^2\Delta z = 3\sum_{k\in\{k_j\}} n_x\Delta y^2\Delta x, \tag{A.26}$$

$$\sum_{k\in\{k_j\}} n_z\Delta z^3 = 3\sum_{k\in\{k_j\}} n_x\Delta z^2\Delta x = 3\sum_{k\in\{k_j\}} n_y\Delta z^2\Delta y, \tag{A.27}$$

$$3\sum_{k\in\{k_j\}} n_y\Delta y^2\Delta z = 3\sum_{k\in\{k_j\}} n_z\Delta y\Delta z^2 = 6\sum_{k\in\{k_j\}} n_x\Delta x\Delta y\Delta z, \tag{A.28}$$

$$3\sum_{k\in\{k_j\}} n_z\Delta z^2\Delta x = 3\sum_{k\in\{k_j\}} n_x\Delta z\Delta x^2 = 6\sum_{k\in\{k_j\}} n_y\Delta x\Delta y\Delta z, \tag{A.29}$$

$$3\sum_{k\in\{k_j\}} n_x\Delta x^2\Delta y = 3\sum_{k\in\{k_j\}} n_y\Delta x\Delta y^2 = 6\sum_{k\in\{k_j\}} n_z\Delta x\Delta y\Delta z, \tag{A.30}$$

where $\mathbf{n}_{jk} = (n_x, n_y, n_z)$, by which the truncation error can be factored as

$$\mathcal{T} = -\frac{1}{24V_j}\sum_{k\in\{k_j\}} \left[n_x\Delta x^3\partial_{xx} + n_y\Delta y^3\partial_{yy} + n_z\Delta z^3\partial_{zz} + 6(n_x\partial_{yz} + n_y\partial_{zx} + n_z\partial_{xy})\Delta x\Delta y\Delta z\right](\mathrm{div}\mathbf{f})_j. \tag{A.31}$$

It is expressed in Equation (3.6) as

$$\mathcal{T} = -\frac{1}{24V_j}\left[Q_{xx} + Q_{yy} + Q_{zz} + Q_{xy} + Q_{yz} + Q_{zx}\right](\mathrm{div}\,\mathbf{f})_j + O(h^3), \tag{A.32}$$

where

$$Q_{xx} = \sum_{k\in\{k_j\}} n_x\Delta x^3\partial_{xx}, \quad Q_{yy} = \sum_{k\in\{k_j\}} n_y\Delta y^3\partial_{yy}, \quad Q_{zz} = \sum_{k\in\{k_j\}} n_z\Delta z^3\partial_{zz}, \tag{A.33}$$

$$Q_{xy} = \sum_{k\in\{k_j\}} 6n_z\Delta x\Delta y\Delta z\,\partial_{xy}, \quad Q_{yz} = \sum_{k\in\{k_j\}} 6n_x\Delta x\Delta y\Delta z\,\partial_{yz}, \quad Q_{zx} = \sum_{k\in\{k_j\}} 6n_y\Delta x\Delta y\Delta z\partial_{zx}. \tag{A.34}$$

Therefore, the second-order truncation error vanishes for exact solutions satisfying $\mathrm{div}\mathbf{f} = 0$, and the leading truncation error becomes $O(h^3)$. For a regular tetrahedral grid in Figure 4, the second-order truncation error $\mathcal{T}$ reduces to the one in Equation (3.9).

A.2 Truncation error for the source term

Consider the Taylor expansion of the source term discretization (5.12) on arbitrary grids:

$$\sum_{k\in\{k_j\}} \frac{1}{2}(s_L + s_R)V_{jk} = \sum_{k\in\{k_j\}} \left(\frac{a_L + a_R}{2}s_j + \frac{a_R + b_L + b_R}{2}\partial_{jk}s_j + \frac{2(b_R + c_R + c_L) + a_R}{4}\partial_{jk}^2 s_j\right)V_{jk}. \tag{A.35}$$

On regular simplex grids, the second term in the parenthesis vanishes identically. In two dimensions, for example, we have

$$\begin{aligned}\sum_{k\in\{k_j\}} \partial_{jk} s_j V_{jk} &= \frac{1}{4}\sum_{k\in\{k_j\}} \left(\Delta\mathbf{x}_{jk}\cdot\mathbf{n}_{jk}\Delta x\partial_x + \Delta\mathbf{x}_{jk}\cdot\mathbf{n}_{jk}\Delta y\partial_y\right) s_j \\ &= \frac{1}{4}\sum_{k\in\{k_j\}} \left(n_x\Delta x^2\partial_x + n_y\Delta x\Delta y\partial_x + n_x\Delta x\Delta y\partial_x + n_y\Delta y^2\partial_y\right) s_j. \end{aligned} \tag{A.36}$$

The following identities hold on regular triangular grids:

$$\sum_{k\in\{k_j\}} n_x\Delta x^2 = \sum_{k\in\{k_j\}} n_x\Delta x\Delta y = \sum_{k\in\{k_j\}} n_y\Delta x\Delta y = \sum_{k\in\{k_j\}} n_y\Delta y^2 = 0, \tag{A.37}$$

and therefore

$$\frac{a_R + b_L + b_R}{2}\sum_{k\in\{k_j\}} \partial_{jk} s_j V_{jk} = 0, \tag{A.38}$$

This implies that the second condition in Equation (5.20) is redundant for regular grids.

Similarly, in three dimensions, we have

$$\begin{aligned}\sum_{k\in\{k_j\}} \partial_{jk} s_j V_{jk} &= \frac{1}{6}\sum_{k\in\{k_j\}} \left(\Delta\mathbf{x}_{jk}\cdot\mathbf{n}_{jk}\Delta x\partial_x + \Delta\mathbf{x}_{jk}\cdot\mathbf{n}_{jk}\Delta y\partial_y + \Delta\mathbf{x}_{jk}\cdot\mathbf{n}_{jk}\Delta z\partial_z\right) s_j \\ &= \frac{1}{6}\sum_{k\in\{k_j\}} \left(n_x\Delta x^2\partial_x + n_y\Delta y\Delta x\partial_x + n_z\Delta z\Delta x\partial_x + n_x\Delta x\Delta y\partial_y + n_y\Delta y^2\partial_y + n_z\Delta z\Delta y\partial_y\right. \\ &+ \left. n_x\Delta x\Delta z\partial_z + n_y\Delta y\Delta z\partial_z + n_z\Delta z^2\partial_z\right) s_j. \end{aligned} \tag{A.39}$$

On regular tetrahedral grids, the following identities hold:

$$\sum_{k\in\{k_j\}} n_x\Delta x^2 = \sum_{k\in\{k_j\}} n_y\Delta y^2 = \sum_{k\in\{k_j\}} n_z\Delta z^2 = 0, \tag{A.40}$$

$$\sum_{k\in\{k_j\}} n_x\Delta x\Delta y = \sum_{k\in\{k_j\}} n_y\Delta y\Delta x = 0, \tag{A.41}$$

$$\sum_{k\in\{k_j\}} n_y\Delta y\Delta z = \sum_{k\in\{k_j\}} n_z\Delta z\Delta y = 0, \tag{A.42}$$

$$\sum_{k\in\{k_j\}} n_z\Delta z\Delta x = \sum_{k\in\{k_j\}} n_x\Delta x\Delta z = 0, \tag{A.43}$$

and therefore

$$\frac{a_R + b_L + b_R}{2}\sum_{k\in\{k_j\}} \partial_{jk} s_j V_{jk} = 0. \tag{A.44}$$

The third term in the parenthesis of Equation (A.35), which is a second-order error term and denoted by $\mathcal{T}_s$:

$$\mathcal{T}_s = \frac{2(b_R + c_R + c_L) + a_R}{4V_j}\sum_{k\in\{k_j\}} \partial^2_{jk} s_j V_{jk}, \tag{A.45}$$

does not vanish, but can be simplified on regular simplex grids. In two dimensions, we have

$$\sum_{k\in\{k_j\}} \partial^2_{jk} s_j V_{jk} = \frac{1}{4}\sum_{k\in\{k_j\}} \left[\left(n_x\Delta x^3 + n_y\Delta x^2\Delta y\right)\partial_{xx} + \left(n_y\Delta y^3 + n_x\Delta y^2\Delta x\right)\partial_{yy} + 2\left(n_x\Delta x^2\Delta y + n_y\Delta x\Delta y^2\right)\partial_{xy}\right] s_j,$$

which can be simplified on regular grids, by the identities (A.12-A.13), as

$$\sum_{k\in\{k_j\}} \partial^2_{jk} s_j V_{jk} = \frac{1}{3} \sum_{k\in\{k_j\}} \left[n_x \Delta x^3 \partial_{xx} + 3 n_x \Delta x^2 \Delta y \partial_{xy} + n_y \Delta y^3 \partial_{yy} \right] s_j, \tag{A.46}$$

and thus,

$$\mathcal{T}_s = \frac{2(b_R + c_R + c_L) + a_R}{4V_j} \left(\frac{1}{3} \right) \sum_{k\in\{k_j\}} \left[Q_{xx} + Q_{yy} + Q_{xy} \right] s_j. \tag{A.47}$$

In three dimensions, we have

$$\begin{aligned} \sum_{k\in\{k_j\}} \partial^2_{jk} s_j V_{jk} = \frac{1}{6} \sum_{k\in\{k_j\}} \big[& \left(n_x \Delta x^3 + n_y \Delta x^2 \Delta y + n_z \Delta x^2 \Delta z \right) \partial_{xx} + \left(n_y \Delta y^3 + n_z \Delta y^2 \Delta z + n_x \Delta y^2 \Delta x \right) \partial_{yy} \\ & + \left(n_z \Delta z^3 + n_x \Delta z^2 \Delta x + n_y \Delta z^2 \Delta y \right) \partial_{zz} + 2 \left(n_x \Delta x^2 \Delta y + n_y \Delta x \Delta y^2 + n_z \Delta x \Delta y \Delta z \right) \partial_{xy} \\ & + 2 \left(n_y \Delta y^2 \Delta z + n_z \Delta y \Delta z^2 + n_x \Delta x \Delta y \Delta z \right) \partial_{yz} + 2 \left(n_x \Delta x^2 \Delta z + n_z \Delta x \Delta z^2 + n_y \Delta x \Delta y \Delta z \right) \partial_{zz} \big] s_j, \end{aligned}$$

which can be simplified on regular tetrahedral grids, by the identities (A.19-A.30), as

$$\sum_{k\in\{k_j\}} \partial^2_{jk} s_j V_{jk} = \frac{5}{18} \sum_{k\in\{k_j\}} \left[n_x \Delta x^3 \partial_{xx} + n_y \Delta y^3 \partial_{yy} + n_z \Delta z^3 \partial_{zz} + 6(n_x \partial_{yz} + n_y \partial_{zx} + n_z \partial_{xy}) \Delta x \Delta y \Delta z \right] s_j.$$

Hence, the second-order truncation error is given by

$$\mathcal{T}_s = \frac{2(b_R + c_R + c_L) + a_R}{4V_j} \left(\frac{5}{18} \right) \sum_{k\in\{k_j\}} \left[Q_{xx} + Q_{yy} + Q_{zz} + Q_{xy} + Q_{yz} + Q_{zx} \right] s_j. \tag{A.48}$$

The coefficients 1/3 in Equation (A.47) and 5/18 in Equation (A.48) can be expressed in terms of the dimension constant D:

$$\frac{D+2}{6D}. \tag{A.49}$$

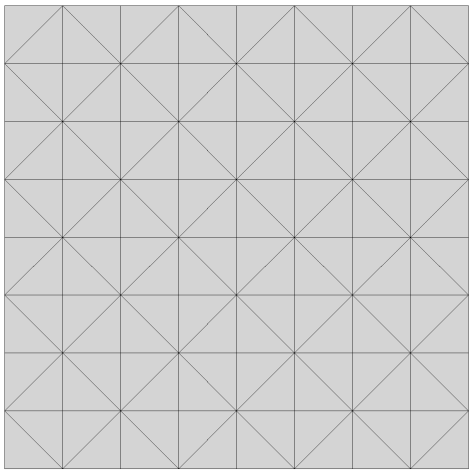
Therefore, the second-order truncation error can be written as in Equation (5.15). Comparing the second-order truncation error $\mathcal{T}_s$ with that for the conservation law, Equations (A.15) and (A.32), we find the compatibility condition:

$$\frac{2(b_R + c_R + c_L) + a_R}{4} \left(\frac{D+2}{6D} \right) = -\frac{1}{24}, \tag{A.50}$$

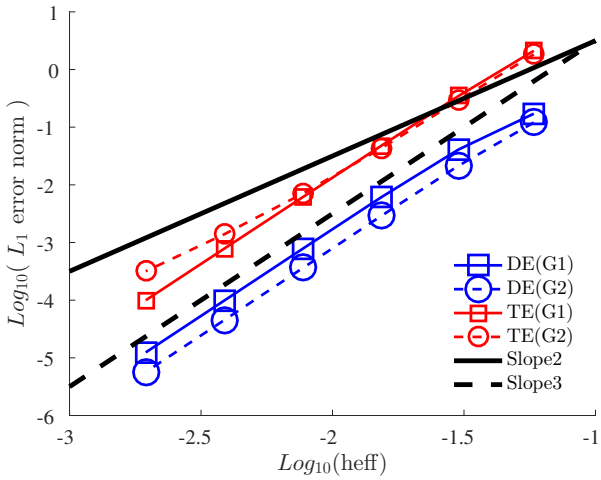
which leads to the equation (5.18).

Appendix B: Non-Regular Grid That Admits Regular Formula

There exist non-regular grids, on which the second-order truncation error cannot be factored as described above, but the regular formula ('Regular' in Table 1) yields third-order accuracy. An example shown in Figure B.1(a) is constructed from a Cartesian grid by inserting alternating diagonals. The grid is not regular because the stencil is not the same at all nodes; some nodes have four neighbors and others have eight neighbors. It is straightforward to show that the identities (A.13) do not hold on this grid. Therefore, the third-order accuracy has to rely on vanishing coefficient of the first-order error term: $a_R + b_L + b_R = 0$. However, the first-order error in the source discretization (A.38) vanishes for any $a_R + b_L + b_R$ because the identities (A.37) still hold on such a grid. Hence, the condition $a_R + b_L + b_R = 0$ is redundant, and so the regular formula can provide third-order accuracy. To verify this, numerical experiments were performed for the linear advection problem used in Section 7.2. The source term was discretized by the regular formula. Two grids are considered: (G1) the regular grid in Figure 10(a), and (G2) the irregular grid in Figure B.1(a). Truncation and discretization error convergence results obtained for $n \times n$ grids with $n = 17, 33, 65, 129, 257, 513$ are shown in Figure B.1(b). As can be observed, the discretization error is $O(h^3)$ on both grids, but the truncation error is $O(h^2)$ on G2 while $O(h^3)$ on G1. The results confirm that the grid in Figure B.1(b) is an irregular grid but allows the regular formula to achieve third-order accuracy.



(a) Irregular right-isosceles triangular grid generated from a Cartesian grid by inserting alternating diagonals (G2).



(b) Error convergence results for interior nodes obtained with the regular formula on two types of grids, G1 (Figure 10(a)) and G2 (Left).

Figure B.1: Grid and error convergence results for regular and irregular right-isosceles triangular grids.