Eleventh International Conference on Computational Fluid Dynamics (ICCFD11), Maui, HI, USA, July 11-15, 2022

ICCFD11-1403

Analysis of Edge-Based Method on Tetrahedra for Diffusion

Boris Diskin*, Hiroaki Nishikawa*, and Yi Liu*
Corresponding author: yi.liu@nianet.org

*National Institute of Aerospace, Hampton, Virginia 23666, USA

Abstract: This paper establishes consistency and formal second-order accuracy of a novel, efficient, edge-based viscous (EBV) discretization method that has been recently developed, implemented in a practical, unstructured-grid, finite-volume flow solver, and demonstrated multifold acceleration of all viscous-kernel computations that include evaluations of meanflow viscous fluxes, turbulence-model and chemistry-model diffusion terms, and the corresponding Jacobian contributions. The EBV method uses simplicial grids, computes viscous fluxes and diffusion terms in an efficient loop over edges, and features a compact discretization stencil based on the nearest neighbors. The second-order accuracy of the EBV solutions is shown by truncation error analysis and by the method of manufactured solutions.

Keywords: Numerical Analysis, Edge-Based Discretization Methods, Diffusion.

1 Introduction

Edge-based and cell-based methods are widely used within unstructured-grid, node-centered, finite-volume solvers for discretizing elliptic second-order partial differential operators that represent viscous effects in computational fluid dynamics (CFD) equations. In the literature, there are many examples of edge-based schemes for inviscid and viscous fluxes [1-6]. A typical edge-based scheme for diffusion requires viscosity and solution gradients at grid edges. The edge-based viscosity is averaged from the edge endpoints. The edge gradients are evaluated either by averaging gradients computed at the edge endpoints or by defining an edge-based stencil [7]. The edge gradient can be augmented with an edge-based derivative [8] or by adding an adjustable term to damp oscillations [9]. The residual contributions from edge-based diffusion terms are computed in an edge loop that avoids duplicate computations inherent in point and cell loops. These traditional edge-based methods typically result in large discretization stencils that include neighbors of neighbors. Alternatively, a thin-layer gradient approximation can be used for approximating diffusion terms. The thin-layer approximation includes only edge-based derivatives, resulting in a compact stencil, but lacks consistency on general unstructured grids, and may degrade solution accuracy.

A consistent and compact-stencil alternative for finite-volume discretization of diffusion terms is a cell-based viscous (CBV) method that is equivalent to the Galerkin finite-element approximation on simplicial grids. The CBV method computes diffusion fluxes in a loop over cells. At each cell, the solution gradient and the cell-averaged viscosity are computed using the solution and viscosity at cell vertices. This approach results in a compact discretization stencil that is suitable for massively parallel computations and conveniently supports the exact linearization. The main disadvantage of the CBV approach is a relatively high computational cost of the cell loop. The CBV method is the baseline method of FUN3D [10], a large-scale CFD code that is developed and supported at NASA Langley

Research Center. FUN3D solutions using the CBV method have been extensively verified and validated through formal analysis and practical applications [11-17].

The edge-based viscous (EBV) method considered in this paper combines benefits of the established cell- and edge-based methods. The EBV residual is efficiently computed in an edge loop, and the EBV discretization stencil involves only immediate neighbors. The EBV method follows the methodology introduced by Barth [1] and uses an edge-based implementation of the cell-based gradients on simplicial grids. The EBV method has been implemented for viscous kernel of Reynolds-averaged Navier-Stokes (RANS) equations in FUN3D. The viscous kernel includes evaluation of the viscous fluxes of the meanflow equations, the diffusion terms of the turbulence-model and chemistry equations, and the corresponding Jacobian terms.

Initial implementation of the EBV method was assessed for the RANS solutions on a family of tetrahedral grids [18]. The EBV solutions closely resembled the reference solutions computed with the baseline CBV method. The time spent on the viscous-kernel computations was reduced by more than a factor of three, and the fraction of viscous-kernel computations was reduced from over 33% to 13% of a typical nonlinear iteration. A relative downside of that EBV implementation was a modest increase in memory required to store EBV coefficients at each grid edge. Later, the EBV implementation was optimized [19]: the EBV memory requirements have been decreased dramatically, the EBV speedup of the viscous-kernel computations on finer grids has been improved to a factor of six, and the fraction of viscous-kernel computations has been reduced to less than 2.5%. Recently, the EBV method was further modified and applied to simulations of complex chemically reacting flows on mixed-element grids [20]. On a mixed-element grid, the EBV method is implemented on a derived tetrahedral grid that is automatically generated by local and consistent division on nontetrahedral cells into tetrahedra. This EBV implementation produces an important additional benefit over the baseline CBV method: the viscous-flux discretization stencil is significantly reduced, resulting in smaller Jacobian, noticeable memory saving, and acceleration of the linear solver of FUN3D.

Although EBV RANS solutions have been verified for several benchmark flows [18-20], a formal analysis of the EBV discretization has been lacking. This paper focuses on systematically establishing consistency and formal second-order accuracy of EBV solutions for scalar diffusion and laminar Navier-Stokes equations. The EBV solutions and truncation- and discretization-error analyses reported in this paper are compared with the corresponding CBV solutions and analyses. The CBV method has rigorous estimates of second-order accuracy on simplicial grids due to its relation to the family of finite-element Galerkin methods and provides high-quality references.

In this paper, the truncation error analysis is conducted on regular simplicial grids by Taylor expansion and by assessing grid convergence of discrete residuals computed for the continuous PDE solution restricted to specific grids in a family. The discretization error analysis is conducted by the method of manufactured solutions on regular and highly irregular simplicial grids. The analyses of the original EBV implementation [18, 19] for scalar diffusion equations indicate second-order accuracy. However, detailed analyses of the EBV implementation for the Navier-Stokes equations reveal that some correction terms are required for second-order accuracy. The correction terms, which are products of spatial derivatives of viscosity and velocity, vanish for scalar diffusion. In the modified EBV implementation, the correction terms have been implemented as source terms for the momentum and energy conservation equations. The EBV solutions in Ref. [20] have been computed with the correction terms. As shown in this paper, without correction terms, the accuracy deterioration is clearly observed in the truncation error and in the discretization error for a manufactured solution. However, only minimal differences are observed for EBV RANS solutions computed with and without correction terms for flow conditions and grids that are used in applications and verification studies.

The material in this paper is presented in the following order. Section 2 describes the CBV and EBV methods for linear and nonlinear diffusion equations on simplicial grids and shows a complexity analysis and truncation- and discretization-error analyses in two dimensions (2D) and three dimensions

(3D). Section 3 analyzes the CBV and EBV methods for the laminar Navier-Stokes equations on 3D tetrahedral grids and indicates necessity of correction terms for consistency and second-order accuracy. Second-order accuracy of the EBV solutions with the corrections terms is shown. Section 4 compares RANS solutions with and without correction terms. Concluding remarks are provided in Section 5.

2 Scalar linear and nonlinear diffusion

2.1 EBV and CBV finite-volume methods

A scalar diffusion partial differential equation (PDE) for a twice differentiable function u is defined in Eq. 1.

$$\mathrm{div}(\mu \boldsymbol{\nabla} u) = g \tag{1}$$

Here, μ is a viscosity function that can be constant, spatially varying, or dependent on solution u; g is a force function introduced to allow manufactured solutions. The equation is discretized in an unstructured-grid, node-centered framework where all functions are defined at grid points. For short notation, functions evaluated at a grid point use the same set of indices as the grid point itself, i.e., function u evaluated at a grid point $\boldsymbol{p}_i$ is denoted as $u_i = u(\boldsymbol{p}_i)$.

Figure 1 illustrates discretization on a general tetrahedron with vertices $\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3,$ and $\boldsymbol{p}_4$. The point $\boldsymbol{m}_{i,j} = \frac{\boldsymbol{p}_i + \boldsymbol{p}_j}{2}$ is the median of the edge connecting vertices $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$. The point $\boldsymbol{f}_{ijk} = \frac{\boldsymbol{p}_i + \boldsymbol{p}_j + \boldsymbol{p}_k}{3}$ is the centroid of the face that has vertices $\boldsymbol{p}_i, \boldsymbol{p}_j,$ and $\boldsymbol{p}_k$. The point $\boldsymbol{c}_{1,2,3,4} = \frac{\boldsymbol{p}_1 + \boldsymbol{p}_2 + \boldsymbol{p}_3 + \boldsymbol{p}_4}{4}$ is the centroid of the tetrahedron. The three quadrilateral shapes shaded in Fig. 1 show the portion of the control-volume surface that is inside the tetrahedron; the control volume is centered at $\boldsymbol{p}_1$; $\boldsymbol{d}_1$ is the corresponding directed-area vector pointing outward from $\boldsymbol{p}_1$.

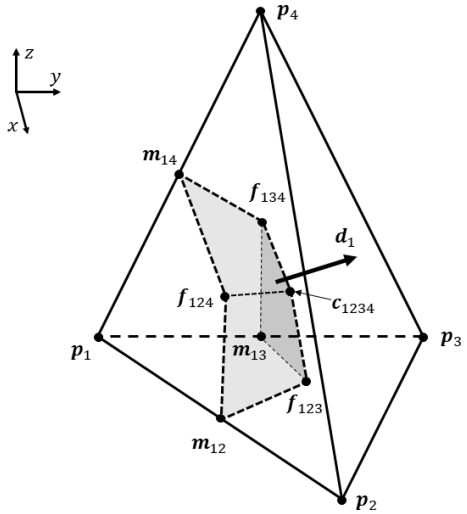


Figure 1. Control-volume boundaries within tetrahedron.

In the CBV method, the tetrahedron provides the following contribution to the residual at $\boldsymbol{p}_1$.

$$R_1 = R_1 + \mu_{1,2,3,4}\left(\boldsymbol{\nabla}_{1,2,3,4} u \cdot \boldsymbol{d}_1\right) \tag{2}$$

Here, the viscosity and the solution gradient at the tetrahedron are evaluated as follows.

$$\mu_{1,2,3,4} = \frac{\mu_1 + \mu_2 + \mu_3 + \mu_4}{4} \tag{3}$$

$$\boldsymbol{\nabla}_{1,2,3,4} u = \frac{1}{Vol}\left[\frac{u_2 + u_3 + u_4}{3}\boldsymbol{n}_1 + \frac{u_1 + u_3 + u_4}{3}\boldsymbol{n}_2 + \frac{u_1 + u_2 + u_4}{3}\boldsymbol{n}_3 + \frac{u_1 + u_2 + u_3}{3}\boldsymbol{n}_4\right] \tag{4}$$

This gradient approximation uses the Green-Gauss theorem; Vol is the tetrahedron volume, $\boldsymbol{n}_i$ is the

outward directed-area vector of the triangular face opposite to $\boldsymbol{p}_i$,

$$\boldsymbol{n}_1 + \boldsymbol{n}_2 + \boldsymbol{n}_3 + \boldsymbol{n}_4 = 0 \tag{5}$$

and the directed-area vector, $\boldsymbol{d}_i$, relates to the directed area of the opposite face as shown in Eq. 6.

$$\boldsymbol{d}_i = \frac{1}{3}\boldsymbol{n}_i \tag{6}$$

Introducing the edge-based difference operator

$$\Delta_{i,j}u \equiv u_j - u_i \tag{7}$$

and substituting $\boldsymbol{n}_1 = -(\boldsymbol{n}_2 + \boldsymbol{n}_3 + \boldsymbol{n}_4)$ (Eq. 5) in Eq. 4, one can obtain a gradient expression in terms of edge-based differences.

$$\boldsymbol{\nabla}_{1,2,3,4}u = \frac{-1}{3\,Vol}\Big((\Delta_{1,2}u)\boldsymbol{n}_2 + (\Delta_{1,3}u)\boldsymbol{n}_3 + (\Delta_{1,4}u)\boldsymbol{n}_4\Big) \tag{8}$$

The residual contribution (Eq. 2) is reformulated in Eq. 9.

$$R_1 = R_1 - \frac{\mu_{1,2,3,4}}{9\,Vol}\left[(\boldsymbol{n}_2 \cdot \boldsymbol{n}_1)\Delta_{1,2}u + (\boldsymbol{n}_3 \cdot \boldsymbol{n}_1)\Delta_{1,3}u + (\boldsymbol{n}_4 \cdot \boldsymbol{n}_1)\Delta_{1,4}u\right] \tag{9}$$

Eq. 9 is a version of the CBV method. The terms in the square brackets are edge-based terms. The only solution-dependent cell-based term is viscosity. The EBV method modifies Eq. 9 as in Eq. 10.

$$R_1 = R_1 - \frac{1}{9\,Vol}\left[\mu_{1,2}(\boldsymbol{n}_2 \cdot \boldsymbol{n}_1)\Delta_{1,2}u + \mu_{1,3}(\boldsymbol{n}_3 \cdot \boldsymbol{n}_1)\Delta_{1,3}u + \mu_{1,4}(\boldsymbol{n}_4 \cdot \boldsymbol{n}_1)\Delta_{1,4}u\right] \tag{10}$$

Coefficients $\mu_{i,j}$ are defined in Eq. 11.

$$\mu_{i,j} = \frac{\mu_i + \mu_j}{2} \tag{11}$$

The EBV method for a scalar diffusion equation combines all grid metrics into a single coefficient at each edge. At the edge $[\boldsymbol{p}_i, \boldsymbol{p}_j]$, the EBV coefficient is computed as in Eq. 12.

$$c_{i,j} = \sum_{\{T\}}\left[\frac{1}{9\,Vol}(\boldsymbol{n}_i \cdot \boldsymbol{n}_j)\right]_T \tag{12}$$

Here, summation is over all tetrahedra that share $[\boldsymbol{p}_i, \boldsymbol{p}_j]$. The EBV coefficient can be precomputed for static and rigidly moving grids. The contributions to residuals at $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$ from $[\boldsymbol{p}_i, \boldsymbol{p}_j]$ are computed in Eq. 13.

$$R_i = R_i - \mu_{i,j}c_{i,j}\Delta_{i,j}u, \qquad R_j = R_j + \mu_{i,j}c_{i,j}\Delta_{i,j}u \tag{13}$$

2.2 Complexity of EBV and CBV methods

To estimate the speedup of the EBV method over the CBV method for a 3D diffusion equation (Eq. 1) on tetrahedral grids, one can compare the number of basic computing operations (additions and multiplications) required for evaluating CBV and EBV residuals. For simplicity, the analysis does not distinguish between multiplications and divisions, between additions and subtractions, and between operations with integer and floating-point numbers and ignores the cost of memory access and sign change.

Ignoring the boundary effects, the number of grid points and the number of cells in a hexahedral grid are asymptotically the same. A tetrahedral grid can be derived by dividing each hexahedron into six tetrahedra. If all hexahedra are divided in the same consistent way, each grid point in the tetrahedral grid has exactly 14 edges attached; each edge is counted for two grid points. Thus, a tetrahedral grid

has approximately six tetrahedra and seven edges per grid point.

At each tetrahedron, the CBV method accesses the solution, viscosity, and the spatial coordinates of the four vertices. The vertices are locally preordered to uniquely determine the orientation of the tetrahedron. Storage of local vertex indexes (four integers per cell, 24 integers per grid point) is the memory used by the CBV method; all grid metrics required for residual updates are computed as needed. The EBV method accesses solution and viscosity at the two endpoints of an edge. All required grid metrics are precomputed and represented by one EBV coefficient at each edge. Storing one EBV coefficient per edge translates into storing seven floating-point numbers per grid point.

At a tetrahedron, the CBV implementation follows Eq. 9. The following three steps are performed: (1) compute solution related quantities, namely, the cell-averaged viscosity and six edge-based solution differences; (2) compute grid metrics, namely, directed area vectors for the four faces, the cell volume, and six inner products of the directed area vectors; and (3) update residuals at the four vertices. To minimize the operation count, multiplications by common coefficients are delayed, and their collective action is accounted for at the final step.

The cell-averaged viscosity (Eq. 3) requires three additions; the division by four is delayed. Six edge-based solution differences require six additions. Overall, the first step requires nine additions and no multiplications.

An edge vector from $\boldsymbol{p}_i$ to $\boldsymbol{p}_k$ is denoted as $\boldsymbol{r}_{i,k}$.

$$\boldsymbol{p}_i = (x_i, y_i, z_i)^{\mathrm{T}}, i = 1,2,3,4; \qquad \boldsymbol{r}_{i,k} = \left(r_{i,kx}, r_{i,ky}, r_{i,kz}\right)^{\mathrm{T}} = (x_k - x_i, y_k - y_i, z_k - z_i)^{\mathrm{T}} \quad (14)$$

The superscript T denotes transposition to indicate vertical vectors. A face directed-area vector, $\boldsymbol{n}_i = \left(n_{ix}, n_{iy}, n_{iz}\right)^{\mathrm{T}}$, is computed as a vector product of two edge vectors.

$$\boldsymbol{n}_2 = \frac{1}{2}\left(\boldsymbol{r}_{1,4} \times \boldsymbol{r}_{1,3}\right), \qquad \boldsymbol{n}_3 = \frac{1}{2}\left(\boldsymbol{r}_{1,2} \times \boldsymbol{r}_{1,4}\right), \qquad \boldsymbol{n}_4 = \frac{1}{2}\left(\boldsymbol{r}_{1,3} \times \boldsymbol{r}_{1,2}\right) \quad (15)$$

There are six edge vectors in a tetrahedron, but only $\boldsymbol{r}_{1,2}, \boldsymbol{r}_{1,3}$, and $\boldsymbol{r}_{1,4}$ are needed for computing $\boldsymbol{n}_2, \boldsymbol{n}_3$, and $\boldsymbol{n}_4$. Computation of these edge vectors requires nine additions. The vector-product evaluation requires six multiplications and five additions; division by two is delayed. Computing $\boldsymbol{n}_1 = -(\boldsymbol{n}_2 + \boldsymbol{n}_3 + \boldsymbol{n}_4)$ requires six additions. The operation count for computing the four directed-area vectors is 18 multiplications and 20 additions.

The Eq. 8 gradient is exact for linear functions. One can compute the cell volume as in Eq. 16.

$$Vol = -\frac{1}{3}\left(r_{1,2x}n_{2x} + r_{1,3x}n_{3x} + r_{1,4x}n_{4x}\right), r_{1,2x} = x_2 - x_1, r_{1,3x} = x_3 - x_1, r_{1,4x} = x_4 - x_1 \quad (16)$$

This volume evaluation requires three multiplications and two additions; division by three is delayed.

Computing an inner product of two vectors requires three multiplications and two additions. Six inner products, $\left(\boldsymbol{n}_i \cdot \boldsymbol{n}_j\right)$, are needed to compute residual contributions at the four vertices. The overall operation count for the grid metrics is 39 multiplications and 34 additions.

In the final step, two multiplications are used to modify the cell-averaged viscosity, which is divided by volume and by 48; the latter factor represents the collective action of all delayed multiplications. The expression in the square brackets of Eq. 9 is evaluated, multiplied by the modified cell-based viscosity, and added to the residual at $\boldsymbol{p}_1$, totaling four multiplication and three additions. Accounting for updates at four nodes, the overall operation count for the final step is 18 multiplications and 14 additions. Table 1 summarizes the CBV operation count. The total CBV operation count for the 3D diffusion residual is 57 multiplications and 53 additions per tetrahedron or 342 multiplications and 318 additions per grid point.

Table 1. CBV operation count for residual contributions from tetrahedron.

	Multiplications	Additions
Solution quantities	0	9
Directed area vectors	18	20
Cell volume	3	2
Inner products	18	12
Residual update	18	14
Total per cell	57	53
Total per grid point	342	318

At each edge, the EBV method performs the following two steps: (1) compute solution related quantities, namely, the edge difference of the solution and the edge-averaged viscosity, and (2) update residuals at the edge endpoints. The solution edge difference requires one addition. The edge-averaged viscosity (Eq. 11) requires one addition. The precomputed EBV coefficient accounts for division by two. The residual update at the edge endpoints (Eq. 13) requires two multiplications and two additions. The total EBV count is two multiplications and four additions per edge or 14 multiplications and 28 additions per grid point. Table 2 summarizes the EBV operation count. Given that multiplications are more costly than additions, the EBV method for a scalar 3D diffusion equation is estimated to be at least 20 times faster than the CBV method.

Table 2. EBV operation count for residual contributions from edge.

	Multiplications	Additions
Edge-based quantities	0	2
Residual update	2	2
Total per edge	2	4
Total per grid point	14	28

Although not shown in this paper, a similar complexity analysis conducted for the Navier-Stokes equations [19] predicts that the EBV operation count is at least five times lower than the CBV operation count (140 multiplications and 217 additions for the EBV method vs. 792 multiplications and 924 additions for the CBV method). The EBV memory consumption for the Navier-Stokes equations is estimated as six EBV coefficients per interior edge, which translates into 42 floating-point numbers per grid point.

In general, the benefits of precomputing grid metrics for unit computations, such as the CBV computations at a tetrahedron or the EBV computations at an edge, depend on the solution input and output for the unit computations. Certain grid metrics should be precomputed and certain operations should be performed for computing the contribution from a solution input to an output. Formally, the EBV method at an edge has four solution inputs (solution and viscosity at two endpoints) and two outputs (endpoint residuals). One might expect that 8 EBV coefficients are required for each edge; such storage requirements would render the EBV method impractical. However, Eq. 13 reveals that the EBV method requires only a single solution input, namely, the product of the edge-averaged viscosity and the solution edge difference, and the two EBV outputs are symmetric, i.e., differ only by sign. These unique EBV properties result in dramatic reduction in the operation count and a modest memory consumption.

It appears that precomputing grid metrics for the CBV method is not beneficial. The CBV method at a tetrahedron has relatively many solution inputs (solution and viscosity at four vertices) and contributes to relatively many outputs (residuals at four vertices). For these reasons, storing grid metrics would require a large additional memory and would result in a modest reduction in the operation count. For example, considering contributions to the four residuals at cell vertices as in Eq. 9, there are six

independent solution inputs, namely, the products of the cell-averaged viscosity and six edge-based solution differences. One can choose to precompute grid metrics $\frac{(\boldsymbol{n}_i \cdot \boldsymbol{n}_j)}{36\,Vol}$; these metrics account for the factor four from the denominator of Eq. 3. In this scenario, six coefficients are stored per cell, and 16 multiplications and 18 additions are needed to compute the residual updates at the four vertices. The total additional storage of this CBV implementation is 36 floating-point numbers per grid point, and the total operation count is 96 multiplications and 108 additions per grid point.

2.3 Truncation and Discretization Errors

The main accuracy measure for a discrete PDE solution is the discretization error. On a given grid, the discretization error function is defined as the difference between the zero-residual discrete solution computed on the grid and the continuous PDE solution restricted to the grid. Another commonly used accuracy measure is the truncation error. The truncation error characterizes the local accuracy of PDE approximation. The truncation error can be defined as the leading term of the Taylor expansion of a discrete residual expression (to approximate the target PDE, a discrete flux-balance residual should be divided by control volume) or as a norm of the discrete residuals evaluated for the continuous solution restricted to the grid. Truncation error analysis on regular grids can accurately predict the convergence order of discretization errors. A grid is considered regular if it can be smoothly mapped on a periodic grid that has identical discretization stencils at the interior grid points. Regular grids include, but are not limited to, grids derived from Cartesian ones, triangular grids obtained by diagonal splitting with a periodic pattern, smoothly stretched grids, skewed grids, smooth curvilinear grids, etc. Grids that cannot be smoothly mapped to a periodic grid are called irregular grids. Note that on irregular (e.g., unstructured) grids, truncation error analysis underestimates convergence of discretization errors [15].

2.3.1 Truncation Error Analysis in 2D

In the 2D truncation-error analysis, a regular anisotropic stencil on a triangular grid uses seven grid points listed in Table 3; h_x and h_y are mesh sizes in the x- and y-directions, respectively. The stencil is illustrated in Fig. 2. The shaded area designates the control volume around $\boldsymbol{p}_0$. The corresponding six triangles are listed in Eq. 17.

$$\{T_1, T_2, \cdots, T_6\} = \{[\boldsymbol{p}_0, \boldsymbol{p}_1, \boldsymbol{p}_2], [\boldsymbol{p}_0, \boldsymbol{p}_2, \boldsymbol{p}_3], [\boldsymbol{p}_0, \boldsymbol{p}_3, \boldsymbol{p}_4], [\boldsymbol{p}_0, \boldsymbol{p}_4, \boldsymbol{p}_5], [\boldsymbol{p}_0, \boldsymbol{p}_5, \boldsymbol{p}_6], [\boldsymbol{p}_0, \boldsymbol{p}_6, \boldsymbol{p}_1]\} \quad (17)$$

Table 3. Grid points of regular anisotropic 2D triangular grid.

	$\boldsymbol{p}_0$	$\boldsymbol{p}_1$	$\boldsymbol{p}_2$	$\boldsymbol{p}_3$	$\boldsymbol{p}_4$	$\boldsymbol{p}_5$	$\boldsymbol{p}_6$
x	0	$-h_x$	0	h_x	h_x	0	$-h_x$
y	0	$-h_y$	$-h_y$	0	h_y	h_y	0

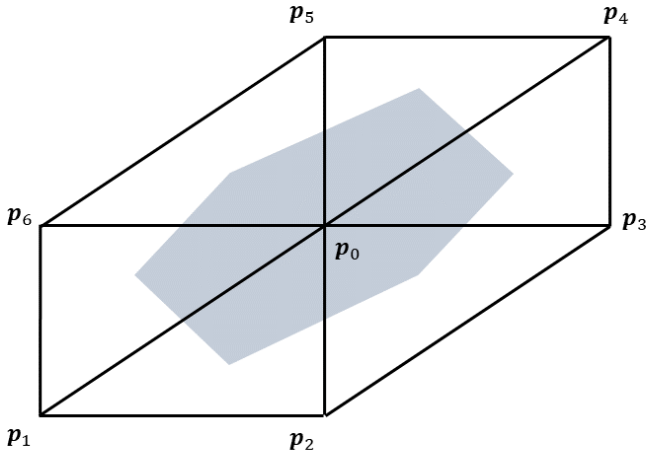


Figure 2. Regular anisotropic 2D triangular grid stencil.

The CBV and EBV discretization schemes on this stencil are defined in Eqs. 18 and 19, respectively.

$$\frac{1}{Vol}\sum_{i=1}^{6} R_{CBV,T_i} = \frac{(\mu_{0,1,2}+\mu_{0,2,3})\Delta_{0,2}u + (\mu_{0,4,5}+\mu_{0,5,6})\Delta_{0,5}u}{2h_y^2} + \frac{(\mu_{0,2,3}+\mu_{0,3,4})\Delta_{0,3}u + (\mu_{0,5,6}+\mu_{0,6,1})\Delta_{0,6}u}{2h_x^2} \tag{18}$$

$$\frac{1}{Vol}\sum_{i=1}^{6} R_{EBV,T_i} = \frac{\mu_{0,2}\Delta_{0,2}u + \mu_{0,5}\Delta_{0,5}u}{h_y^2} + \frac{\mu_{0,3}\Delta_{0,3}u + \mu_{0,6}\Delta_{0,6}u}{h_x^2} \tag{19}$$

Here, $Vol = h_x h_y$, R_{CBV,T_i} and R_{EBV,T_i} are contributions to the CBV and EBV residuals at $\boldsymbol{p}_0$ from the triangle T_i,

$$\mu_{0,i,j} = \frac{\mu_0 + \mu_i + \mu_j}{3}, \qquad \mu_{0i} = \frac{\mu_0 + \mu_i}{2} \tag{20}$$

are cell- and edge-averaged viscosity coefficients, and $\Delta_{ij}u$ is the edge-based solution difference defined in Eq. 7. Eqs. 18 and 19 offer an interpretation of the EBV coefficient at an edge as an approximation to the CBV coefficients averaged over all cells that share the edge. The following Taylor series are used in computing truncation errors.

$$\Delta_{0,2}u = u_2 - u_0 \approx -h_y u_y + \frac{1}{2}h_y^2 u_{yy} - \frac{1}{6}h_y^3 u_{yyy} + \frac{1}{24}h_y^4 u_{yyyy}$$
$$\Delta_{0,3}u = u_3 - u_0 \approx h_x u_x + \frac{1}{2}h_x^2 u_{xx} + \frac{1}{6}h_x^3 u_{xxx} + \frac{1}{24}h_x^4 u_{xxxx}$$
$$\Delta_{0,5}u = u_5 - u_0 \approx h_y u_y + \frac{1}{2}h_y^2 u_{yy} + \frac{1}{6}h_y^3 u_{yyy} + \frac{1}{24}h_y^4 u_{yyyy}$$
$$\Delta_{0,6}u = u_6 - u_0 \approx -h_x u_x + \frac{1}{2}h_x^2 u_{xx} - \frac{1}{6}h_x^3 u_{xxx} + \frac{1}{24}h_x^4 u_{xxxx}$$
$$\mu_{0,2} = \frac{\mu_0+\mu_2}{2} \approx \mu_0 - \frac{1}{2}h_y\mu_y + \frac{1}{4}h_y^2\mu_{yy} - \frac{1}{12}h_y^3\mu_{yyy}$$
$$\mu_{0,3} = \frac{\mu_0+\mu_3}{2} \approx \mu_0 + \frac{1}{2}h_x\mu_x + \frac{1}{4}h_x^2\mu_{xx} + \frac{1}{12}h_x^3\mu_{xxx}$$
$$\mu_{0,5} = \frac{\mu_0+\mu_5}{2} \approx \mu_0 + \frac{1}{2}h_y\mu_y + \frac{1}{4}h_y^2\mu_{yy} + \frac{1}{12}h_y^3\mu_{yyy}$$
$$\mu_{0,6} = \frac{\mu_0+\mu_6}{2} \approx \mu_0 - \frac{1}{2}h_x\mu_x + \frac{1}{4}h_x^2\mu_{xx} - \frac{1}{12}h_x^3\mu_{xxx}$$
$$\mu_{0,1,2}+\mu_{0,2,3} = \frac{2\mu_0+\mu_1+2\mu_2+\mu_3}{3} \approx 2\mu_0 - h_y\mu_y + \left(\frac{1}{3}h_x^2\mu_{xx} + \frac{1}{3}h_x h_y\mu_{xy} + \frac{1}{2}h_y^2\mu_{yy}\right) - \left(\frac{1}{6}h_x^2 h_y\mu_{xxy} + \frac{1}{6}h_x h_y^2\mu_{xyy} + \frac{1}{6}h_y^3\mu_{yyy}\right)$$
$$\mu_{0,4,5}+\mu_{0,5,6} = \frac{2\mu_0+\mu_4+2\mu_5+\mu_6}{3} \approx 2\mu_0 + h_y\mu_y + \left(\frac{1}{3}h_x^2\mu_{xx} + \frac{1}{3}h_x h_y\mu_{xy} + \frac{1}{2}h_y^2\mu_{yy}\right) + \left(\frac{1}{6}h_x^2 h_y\mu_{xxy} + \frac{1}{6}h_x h_y^2\mu_{xyy} + \frac{1}{6}h_y^3\mu_{yyy}\right)$$
$$\mu_{0,2,3}+\mu_{0,3,4} = \frac{2\mu_0+\mu_2+2\mu_3+\mu_4}{3} \approx 2\mu_0 + h_x\mu_x + \left(\frac{1}{2}h_x^2\mu_{xx} + \frac{1}{3}h_x h_y\mu_{xy} + \frac{1}{3}h_y^2\mu_{yy}\right) + \left(\frac{1}{6}h_x^3\mu_{xxx} + \frac{1}{6}h_x^2 h_y\mu_{xxy} + \frac{1}{6}h_x h_y^2\mu_{xyy}\right)$$
$$\mu_{0,5,6}+\mu_{0,6,1} = \frac{2\mu_0+\mu_5+2\mu_6+\mu_1}{3} \approx 2\mu_0 - h_x\mu_x + \left(\frac{1}{2}h_x^2\mu_{xx} + \frac{1}{3}h_x h_y\mu_{xy} + \frac{1}{3}h_y^2\mu_{yy}\right) - \left(\frac{1}{6}h_x^3\mu_{xxx} + \frac{1}{6}h_x^2 h_y\mu_{xxy} + \frac{1}{6}h_x h_y^2\mu_{xyy}\right)$$

The accumulated truncation errors are represented as follows.

$$\frac{1}{Vol}\sum_{i=1}^{6} R_{CBV,T_i} \approx \mu_y u_y + \mu_x u_x + \mu_0\left(u_{yy}+u_{xx}\right) + \frac{1}{12}\mu_0\left(h_y^2 u_{yyyy} + h_x^2 u_{xxxx}\right) \tag{21}$$
$$+\frac{1}{6}\left(h_y^2\mu_y u_{yyy} + h_x^2\mu_x u_{xxx}\right) + \left(\frac{1}{6}h_x^2\mu_{xx} + \frac{1}{6}h_x h_y \mu_{xy} + \frac{1}{4}h_y^2\mu_{yy}\right)u_{yy}$$
$$+\left(\frac{1}{4}h_x^2\mu_{xx} + \frac{1}{6}h_x h_y \mu_{xy} + \frac{1}{6}h_y^2\mu_{yy}\right)u_{xx}$$
$$+\left(\frac{1}{6}h_x^2\mu_{xxy} + \frac{1}{6}h_x h_y \mu_{xyy} + \frac{1}{6}h_y^2\mu_{yyy}\right)u_y$$
$$+\left(\frac{1}{6}h_x^2\mu_{xxx} + \frac{1}{6}h_x h_y \mu_{xxy} + \frac{1}{6}h_y^2\mu_{xyy}\right)u_x + O(h^4)$$

$$\frac{1}{Vol}\sum_{i=1}^{6} R_{EBV,T_i} \approx \mu_y u_y + \mu_x u_x + \mu_0\left(u_{yy}+u_{xx}\right) + \frac{\mu_0}{12}\left(h_y^2 u_{yyyy} + h_x^2 u_{xxxx}\right) + \frac{1}{6}h_y^2\mu_y u_{yyy} \tag{22}$$
$$+\frac{1}{6}h_x^2\mu_x u_{xxx} + \frac{1}{4}h_y^2\mu_{yy}u_{yy} + \frac{1}{4}h_x^2\mu_{xx}u_{xx} + \frac{1}{6}h_y^2\mu_{yyy}u_y + \frac{1}{6}h_x^2\mu_{xxx}u_x + O(h^4)$$

The CBV and EBV truncation errors indicate consistent, second-order accurate discretization methods.

2.3.2 Truncation Error Analysis in 3D

A regular isotropic tetrahedral-grid stencil used in the 3D truncation-error analysis has mesh size h and contains 14 grid points $\boldsymbol{p}_i$ listed in Table 4. A set of 24 tetrahedra forms a closed polyhedron around $\boldsymbol{p}_0$ and determines the edges between grid points. Each tetrahedron is uniquely defined by its four vertices.

Table 4. Grid points of regular isotropic 3D tetrahedral grid.

	$\boldsymbol{p}_0$	$\boldsymbol{p}_1$	$\boldsymbol{p}_2$	$\boldsymbol{p}_3$	$\boldsymbol{p}_4$	$\boldsymbol{p}_5$	$\boldsymbol{p}_6$	$\boldsymbol{p}_7$	$\boldsymbol{p}_8$	$\boldsymbol{p}_9$	$\boldsymbol{p}_{10}$	$\boldsymbol{p}_{11}$	$\boldsymbol{p}_{12}$	$\boldsymbol{p}_{13}$	$\boldsymbol{p}_{14}$
x	0	0	0	$-h$	$-h$	0	0	$-h$	$-h$	0	0	h	h	h	h
y	0	0	$-h$	0	0	$-h$	0	h	h	h	h	$-h$	0	$-h$	0
z	0	$-h$	0	$-h$	0	$-h$	h	$-h$	0	$-h$	0	0	0	h	h

$$\begin{aligned}\{T_1, T_2, \cdots, T_{24}\} = \{&[\boldsymbol{p}_0,\boldsymbol{p}_3,\boldsymbol{p}_1,\boldsymbol{p}_2],[\boldsymbol{p}_0,\boldsymbol{p}_3,\boldsymbol{p}_2,\boldsymbol{p}_4],[\boldsymbol{p}_0,\boldsymbol{p}_2,\boldsymbol{p}_5,\boldsymbol{p}_4],[\boldsymbol{p}_0,\boldsymbol{p}_5,\boldsymbol{p}_6,\boldsymbol{p}_4],\\ &[\boldsymbol{p}_0,\boldsymbol{p}_4,\boldsymbol{p}_7,\boldsymbol{p}_3],[\boldsymbol{p}_0,\boldsymbol{p}_4,\boldsymbol{p}_8,\boldsymbol{p}_7],[\boldsymbol{p}_0,\boldsymbol{p}_7,\boldsymbol{p}_9,\boldsymbol{p}_1],[\boldsymbol{p}_0,\boldsymbol{p}_9,\boldsymbol{p}_7,\boldsymbol{p}_{10}],\\ &[\boldsymbol{p}_0,\boldsymbol{p}_7,\boldsymbol{p}_1,\boldsymbol{p}_3],[\boldsymbol{p}_0,\boldsymbol{p}_8,\boldsymbol{p}_{10},\boldsymbol{p}_7],[\boldsymbol{p}_0,\boldsymbol{p}_{10},\boldsymbol{p}_8,\boldsymbol{p}_6],[\boldsymbol{p}_0,\boldsymbol{p}_4,\boldsymbol{p}_6,\boldsymbol{p}_8],\\ &[\boldsymbol{p}_0,\boldsymbol{p}_1,\boldsymbol{p}_{11},\boldsymbol{p}_2],[\boldsymbol{p}_0,\boldsymbol{p}_{11},\boldsymbol{p}_1,\boldsymbol{p}_{12}],[\boldsymbol{p}_0,\boldsymbol{p}_2,\boldsymbol{p}_{13},\boldsymbol{p}_5],[\boldsymbol{p}_0,\boldsymbol{p}_6,\boldsymbol{p}_5,\boldsymbol{p}_{13}],\\ &[\boldsymbol{p}_0,\boldsymbol{p}_{13},\boldsymbol{p}_{11},\boldsymbol{p}_{12}],[\boldsymbol{p}_0,\boldsymbol{p}_{14},\boldsymbol{p}_{13},\boldsymbol{p}_{12}],[\boldsymbol{p}_0,\boldsymbol{p}_{13},\boldsymbol{p}_2,\boldsymbol{p}_{11}],[\boldsymbol{p}_0,\boldsymbol{p}_{14},\boldsymbol{p}_6,\boldsymbol{p}_{13}],\\ &[\boldsymbol{p}_0,\boldsymbol{p}_{12},\boldsymbol{p}_1,\boldsymbol{p}_9],[\boldsymbol{p}_0,\boldsymbol{p}_{12},\boldsymbol{p}_9,\boldsymbol{p}_{10}],[\boldsymbol{p}_0,\boldsymbol{p}_{10},\boldsymbol{p}_6,\boldsymbol{p}_{14}],[\boldsymbol{p}_0,\boldsymbol{p}_{12},\boldsymbol{p}_{10},\boldsymbol{p}_{14}]\}\end{aligned} \tag{23}$$

The CBV and EBV discretization schemes on this stencil are defined as follows.

$$\frac{1}{Vol}\sum_{i=1}^{24} R_{CBV,T_i} = \frac{\left(\mu_{0,1,2,3}+\mu_{0,1,7,9}+\mu_{0,1,3,7}+\mu_{0,1,2,11}+\mu_{0,1,11,12}+\mu_{0,1,9,12}\right)}{6h^2}\Delta_{0,1}u \tag{24}$$
$$+\frac{\left(\mu_{0,1,2,3}+\mu_{0,2,3,4}+\mu_{0,2,4,5}+\mu_{0,1,2,11}+\mu_{0,2,5,13}+\mu_{0,2,11,13}\right)}{6h^2}\Delta_{0,2}u$$
$$+\frac{\left(\mu_{0,2,3,4}+\mu_{0,2,4,5}+\mu_{0,4,5,6}+\mu_{0,3,4,7}+\mu_{0,4,7,8}+\mu_{0,4,6,8}\right)}{6h^2}\Delta_{0,4}u$$
$$+\frac{\left(\mu_{0,4,5,6}+\mu_{0,6,8,10}+\mu_{0,4,6,8}+\mu_{0,5,6,13}+\mu_{0,6,13,14}+\mu_{0,6,10,14}\right)}{6h^2}\Delta_{0,6}u$$
$$+\frac{\left(\mu_{0,7,9,10}+\mu_{0,7,8,10}+\mu_{0,6,8,10}+\mu_{0,9,10,12}+\mu_{0,6,10,14}+\mu_{0,10,12,14}\right)}{6h^2}\Delta_{0,10}u$$
$$+\frac{\left(\mu_{0,1,11,12}+\mu_{0,11,12,13}+\mu_{0,12,13,14}+\mu_{0,1,9,12}+\mu_{0,9,10,12}+\mu_{0,10,12,14}\right)}{6h^2}\Delta_{0,12}u$$

$$\frac{1}{Vol}\sum_{i=1}^{24} R_{EBV,T_i} = \mu_{0,1}\frac{\Delta_{0,1}u}{h^2} + \mu_{0,2}\frac{\Delta_{0,2}u}{h^2} + \mu_{0,4}\frac{\Delta_{0,4}u}{h^2} + \mu_{0,6}\frac{\Delta_{0,6}u}{h^2} + \mu_{0,10}\frac{\Delta_{0,10}u}{h^2} + \mu_{0,12}\frac{\Delta_{0,12}u}{h^2} \quad (25)$$

Evaluating the truncation errors at each tetrahedron, adding up contributions from all tetrahedra, and dividing by the control volume ($Vol = h^3$) results in the following accumulated truncation errors.

$$\frac{1}{Vol}\sum_{i=1}^{24} R_{CBV,T_i} \approx \mathrm{div}(\mu\boldsymbol{\nabla}u) + h^2\left(T_{u_x} + T_{u_y} + T_{u_z} + T_{u_{xx}} + T_{u_{yy}} + T_{u_{zz}} + T_3 + T_4\right) + O(h^4) \quad (26)$$

$$\frac{1}{Vol}\sum_{i=1}^{24} R_{EBV,T_i} \approx \mathrm{div}(\mu\boldsymbol{\nabla}u) + h^2(T_1' + T_2' + T_3' + T_4') + O(h^4) \quad (27)$$

$$T_{u_x} = \frac{u_x}{6}\left(\mu_{xxx} - \mu_{xxy} + \mu_{xxz} + \mu_{xyy} - \mu_{xyz} + \mu_{xzz}\right)$$

$$T_{u_y} = \frac{u_y}{6}\left(\mu_{xxy} - \mu_{xyy} + \mu_{xyz} + \mu_{yyy} - \mu_{yyz} + \mu_{yzz}\right)$$

$$T_{u_z} = \frac{u_z}{6}\left(\mu_{xxz} - \mu_{xyz} + \mu_{xzz} + \mu_{yyz} - \mu_{yzz} + \mu_{zzz}\right)$$

$$T_{u_{xx}} = \frac{u_{xx}}{6}\left(-\mu_{xy} - \mu_{yz} + \mu_{xz} + \mu_{yy} + \mu_{zz} + \frac{3}{2}\mu_{xx}\right)$$

$$T_{u_{yy}} = \frac{u_{yy}}{6}\left(-\mu_{xy} - \mu_{yz} + \mu_{xz} + \mu_{zz} + \mu_{xx} + \frac{3}{2}\mu_{yy}\right)$$

$$T_{u_{zz}} = \frac{u_{zz}}{6}\left(-\mu_{xy} - \mu_{yz} + \mu_{xz} + \mu_{yy} + \mu_{xx} + \frac{3}{2}\mu_{zz}\right)$$

$$T_3 = T_3' = \frac{1}{6}\left(\mu_x u_{xxx} + \mu_y u_{yyy} + \mu_z u_{zzz}\right), \qquad T_4 = T_4' = \frac{1}{12}\mu_0\left(u_{xxxx} + u_{yyyy} + u_{zzzz}\right)$$

$$T_1' = \frac{1}{6}\left(u_x\mu_{xxx} + u_y\mu_{yyy} + u_z\mu_{zzz}\right), \qquad T_2' = \frac{1}{4}\left(u_{xx}\mu_{xx} + u_{yy}\mu_{yy} + u_{zz}\mu_{zz}\right)$$

The leading terms of the CBV and EBV truncation errors indicate second-order accuracy. Although not shown, 2D and 3D truncation-error analyses have also been conducted on regular skewed grids and confirmed second-order accuracy of the CBV and EBV methods for diffusion equations.

2.3 Discretization error analysis in 3D

The analysis of CBV and EBV discretization errors is conducted using the method of manufactured solutions. In all tests in this section, two PDEs are considered, both of them can be described by Eq. 1. A linear PDE corresponds to $\mu = 1$ and a nonlinear PDE corresponds to $\mu = 1 + u^2$. The same manufactured solution is used for the linear and nonlinear equations resulting in different force functions g. Strong Dirichlet boundary conditions are used; discrete solutions at boundary nodes are specified from the manufactured solution.

The discretization error function is computed as the pointwise difference between the zero-residual discrete solution on a given grid and the manufactured solution evaluated at grid points. For the linear equation, the CBV and EBV solutions are expected to be identical. The discretization error convergence is assessed on families of uniformly refined grids for an isotropic cube. Two grid families are generated: a family of four regular tetrahedral grids and a family of irregular grids derived by a random perturbation of grid points of the corresponding regular grids. The L_1-norm of the discretization error function is plotted versus the effective mesh size, which is computed as the L_1-norm of the cubic roots of the control volumes. The L_1-norm of a discrete function φ_i is defined as in Eq. 28.

$$\|\varphi\|_{L_1} = \frac{1}{N}\sum_{i=1}^{N}|\varphi_i| \tag{28}$$

N is the number of grid points.

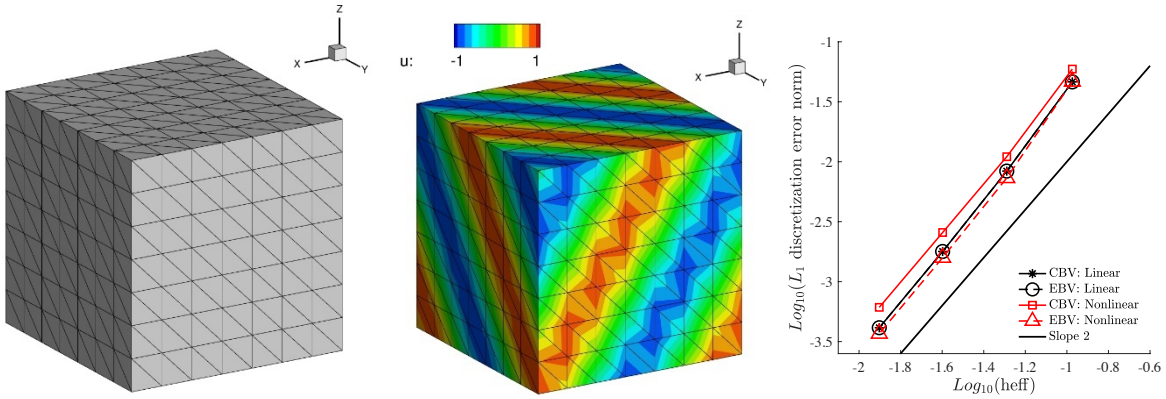
For the unit cube $(x, y, z) \in [0, 1] \times [0, 1] \times [0, 1]$, the manufactured solution is defined in Eq. 29.

$$u = \sin(2.2\,x + 2.4\,y + 1.4\,z) \tag{29}$$

The regular grids are uniform tetrahedral grids with $8, 16$, 32, and 64 grid points in each dimension. Each interior grid point has exactly 24 tetrahedra that share the point and 14 edge neighbors. The irregular grids are derived from the regular grids by perturbing coordinates of the interior grid points as shown in Eq. 30.

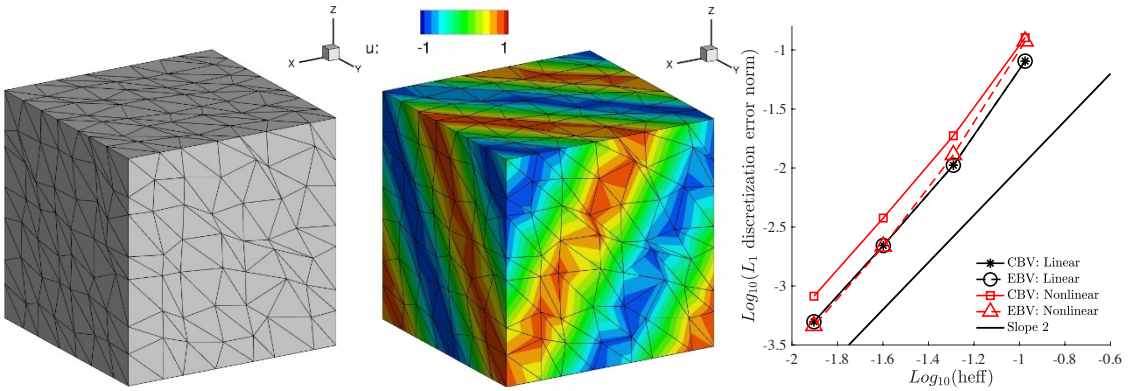
$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} + 0.6 \begin{bmatrix} \rho_1 h \\ \rho_2 h \\ \rho_3 h \end{bmatrix} \tag{30}$$

Here, h is the regular-grid mesh size and ρ_1, ρ_2, and ρ_3 are random real numbers between -1 and 1. The boundary grid points are perturbed within the corresponding boundary planes, so the shape of the computational domain is preserved. The degrees of freedom and the grid connectivity remain the same. The grid perturbation is done sequentially, point by point. Volumes of perturbed tetrahedra are checked after each point perturbation. If the perturbation produces a tetrahedron with a negative volume, the perturbation of this specific grid point is cancelled. The coarsest regular grid, contours of the manufactured solution, and convergence of the discretization errors on the regular grids are shown in Fig. 3. Figure 4 shows the corresponding data for irregular grids. All discretization errors converge with second order. As expected, the EBV and CBV solutions for the linear PDE are identical. The magnitude of the EBV error for the nonlinear PDE is slightly lower than the magnitude of the CBV error.



a. Coarse grid b. Manufactured solution c. Discretization error convergence

Figure 3. Regular isotropic tetrahedral grids.



a. Coarse grid b. Manufactured solution c. Discretization error convergence

Figure 4. Irregular isotropic tetrahedral grids.

3 Laminar Navier-Stokes equations

The three-dimensional compressible unsteady Navier-Stokes equations are given by [21].

$$\partial_t \boldsymbol{Q} + \partial_x \boldsymbol{F} + \partial_y \boldsymbol{G} + \partial_z \boldsymbol{H} = \boldsymbol{0} \tag{31}$$

In a nondimensional form, the vectors $\boldsymbol{F}, \boldsymbol{G}$, and $\boldsymbol{H}$ are defined in Eq. 32.

$$\boldsymbol{F} = \begin{pmatrix} \rho u \\ \rho uu + p - \tau_{xx} \\ \rho uv - \tau_{xy} \\ \rho uw - \tau_{xz} \\ (E+p)u - \left(u\tau_{xx} + v\tau_{xy} + w\tau_{xz}\right) + q_x \end{pmatrix}$$

$$\boldsymbol{G} = \begin{pmatrix} \rho v \\ \rho uv - \tau_{xy} \\ \rho vv + p - \tau_{yy} \\ \rho vw - \tau_{yz} \\ (E+p)v - \left(u\tau_{xy} + v\tau_{yy} + w\tau_{yz}\right) + q_y \end{pmatrix} \tag{32}$$

$$\boldsymbol{H} = \begin{pmatrix} \rho w \\ \rho uw - \tau_{xz} \\ \rho vw - \tau_{yz} \\ \rho ww + p - \tau_{zz} \\ (E+p)w - \left(u\tau_{xz} + v\tau_{yz} + w\tau_{zz}\right) + q_z \end{pmatrix}$$

Here, p is the static pressure, $\boldsymbol{u} = (u, v, w)^{\mathrm{T}}$ is the velocity vector, $\boldsymbol{q} = \left(q_x, q_y, q_z\right)^{\mathrm{T}}$ is the local heat flux vector, and $\boldsymbol{Q} \equiv (\rho, \rho u, \rho v, \rho w, E)^{\mathrm{T}}$ is the vector of conserved variables that includes the density ρ, the momentum $\rho\boldsymbol{u} = (\rho u, \rho v, \rho w)^{\mathrm{T}}$, and the total energy per unit volume E. For a perfect gas, equations are closed using the following relations:

$$p = (\gamma - 1)\left(E - \frac{\rho}{2}(u^2 + v^2 + w^2)\right), \qquad \mathrm{T} = a^2 = \gamma\frac{p}{\rho} \tag{33}$$

where T is a nondimensional temperature, a is the speed of sound and $\gamma = 1.4$ is the ratio of specific heats.

The viscous fluxes in the Navier-Stokes equations include the shear stress tensor and the heat flux vector as defined in Eq. 34.

$$\tau_{xx} = \frac{2}{3}\frac{M_{ref}}{Re}\mu\left(2\partial_x u - \partial_y v - \partial_z w\right)$$

$$
\begin{aligned}
\tau_{yy} &= \frac{2}{3}\frac{M_{ref}}{Re}\mu\left(2\partial_y v - \partial_x u - \partial_z w\right) \\
\tau_{zz} &= \frac{2}{3}\frac{M_{ref}}{Re}\mu\left(2\partial_z w - \partial_x u - \partial_y v\right) \\
\tau_{yx} &= \tau_{xy} = \frac{M_{ref}}{Re}\mu\left(\partial_x v + \partial_y u\right) \\
\tau_{zx} &= \tau_{xz} = \frac{M_{ref}}{Re}\mu(\partial_x w + \partial_z u) \\
\tau_{zy} &= \tau_{yz} = \frac{M_{ref}}{Re}\mu\left(\partial_z v + \partial_y w\right) \\
q_x &= \frac{M_{ref}}{Re(\gamma-1)}\frac{\mu}{Pr}\partial_x T \\
q_y &= \frac{M_{ref}}{Re(\gamma-1)}\frac{\mu}{Pr}\partial_y T \\
q_z &= \frac{M_{ref}}{Re(\gamma-1)}\frac{\mu}{Pr}\partial_z T
\end{aligned}
\tag{34}
$$

Here, μ is the dynamic laminar viscosity computed by Sutherland's law [22], M_{ref} is the reference Mach number, Re is the Reynolds number, and Pr is the Prandtl number. In Sutherland's law, the nondimensional local dynamic viscosity, μ, relates to the local nondimensional temperature, T, through the following formula

$$
\mu = \mu_{ref} T^{\frac{3}{2}}\left(\frac{1+S^*}{T+S^*}\right), \qquad S^* = \frac{S}{T_{ref}} \tag{35}
$$

where $S = 198.6$ °R, and the reference dimensional viscosity, μ_{ref}, is assumed at the reference dimensional temperature T_{ref}.

3.1 EBV and CBV finite-volume methods

For the tetrahedron in Fig. 1, the gradients can be computed in an edge-based form as in Eq. 8. Applying the edge-based gradients, $\boldsymbol{\nabla}^h = \left(\partial_x^h, \partial_y^h, \partial_z^h\right)$, to the shear stress terms of the Navier-Stokes equations (Eq. 34) results in the following contribution from the cell to the CBV momentum conservation residual evaluated at $\boldsymbol{p}_1$.

$$
\boldsymbol{R}_{CBV,m1} = \boldsymbol{R}_{CBV,m1} + \frac{\mu}{9\,Vol}\left\{\boldsymbol{M}_{2,1}\Delta_{2,1}\boldsymbol{u} + \boldsymbol{M}_{3,1}\Delta_{3,1}\boldsymbol{u} + \boldsymbol{M}_{4,1}\Delta_{4,1}\boldsymbol{u}\right\} \tag{36}
$$

where $\Delta_{j,i}\boldsymbol{u} = \left(\Delta_{j,i}u, \Delta_{j,i}v, \Delta_{j,i}w\right)^{\mathrm{T}}$, $\Delta_{j,i}u = u_j - u_i, \Delta_{j,i}v = v_j - v_i, \Delta_{j,i}w = w_j - w_i$, and

$$
\boldsymbol{M}_{j,i} = \left[\left(\boldsymbol{n}_j \cdot \boldsymbol{n}_i\right)\boldsymbol{I} - \frac{2}{3}\boldsymbol{n}_i\boldsymbol{n}_j^{\mathrm{T}} + \boldsymbol{n}_j\boldsymbol{n}_i^{\mathrm{T}}\right] \tag{37}
$$

is a 3×3 matrix of edge-based grid-metric coefficients associated with the edge $\left[\boldsymbol{p}_j, \boldsymbol{p}_i\right]$, $\boldsymbol{I}$ is the 3×3 identity matrix, $\left(\boldsymbol{n}_j \cdot \boldsymbol{n}_i\right)$ and $\boldsymbol{n}_i\boldsymbol{n}_j^{\mathrm{T}}$ are the scalar and tensor products of vectors $\boldsymbol{n}_i$ and $\boldsymbol{n}_j$, respectively.

The EBV method modifies Eq. 36 as

$$
\boldsymbol{R}_{EBV,m1} = \boldsymbol{R}_{EBV,m1} + \left\{\frac{\mu_{2,1}}{9\,Vol}\boldsymbol{M}_{2,1}\Delta_{2,1}\boldsymbol{u} + \frac{\mu_{3,1}}{9\,Vol}\boldsymbol{M}_{3,1}\Delta_{3,1}\boldsymbol{u} + \frac{\mu_{4,1}}{9\,Vol}\boldsymbol{M}_{4,1}\Delta_{4,1}\boldsymbol{u}\right\} \tag{38}
$$

where $\mu_{j,i}$ represents the edge-based average of the viscosity coefficients defined at $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$. The contribution from the edge $[\boldsymbol{p}_1, \boldsymbol{p}_2]$ to the EBV momentum residual at $\boldsymbol{p}_1$ is represented in Eq. 39.

$$
\boldsymbol{R}_{EBV,m1} = \boldsymbol{R}_{EBV,m1} + \mu_{2,1}\boldsymbol{E}_{2,1}\Delta_{2,1}\boldsymbol{u} \tag{39}
$$

The EBV matrix of coefficients, $\boldsymbol{E}_{2,1} = \sum \frac{\boldsymbol{M}_{2,1}}{9\,Vol}$, where summation is over all tetrahedra that share the edge $[\boldsymbol{p}_1, \boldsymbol{p}_2]$. Similarly, the contribution from the edge $[\boldsymbol{p}_1, \boldsymbol{p}_2]$ to the EBV momentum residual at $\boldsymbol{p}_2$ is represented in Eq. 40.

$$\boldsymbol{R}_{EBV,m2} = \boldsymbol{R}_{EBV,m2} + \mu_{1,2}\boldsymbol{E}_{1,2}\Delta_{1,2}\boldsymbol{u} \tag{40}$$

The CBV method for the energy-residual update at $\boldsymbol{p}_1$ is formulated in Eq. 41.

$$\begin{aligned} R_{CBV,e1} = R_{CBV,e1} &+ \frac{\mu}{9\,Vol}\bar{\boldsymbol{u}}^T\{\boldsymbol{M}_{2,1}\Delta_{2,1}\boldsymbol{u} + \boldsymbol{M}_{3,1}\Delta_{3,1}\boldsymbol{u} \quad + \boldsymbol{M}_{4,1}\Delta_{4,1}\boldsymbol{u}\} \\ &- \frac{\mu}{9\,PrVol(\gamma-1)}\{(\boldsymbol{n}_2\cdot\boldsymbol{n}_1)\Delta_{2,1}T + (\boldsymbol{n}_3\cdot\boldsymbol{n}_1)\Delta_{3,1}T + (\boldsymbol{n}_4\cdot\boldsymbol{n}_1)\Delta_{4,1}T\} \end{aligned} \tag{41}$$

Here, $\bar{\boldsymbol{u}}$ is the velocity vector averaged over the cell. The EBV formulation for the contributions to the energy residual at $\boldsymbol{p}_1$ from the edge $[\boldsymbol{p}_1, \boldsymbol{p}_2]$ is represented as follows.

$$R_{EBV,e1} = R_{EBV,e1} + \mu_{2,1}\boldsymbol{E}_{2,1}\bar{\boldsymbol{u}}_{2,1}^T\Delta_{2,1}\boldsymbol{u} - \frac{1}{(\gamma-1)}\frac{\mu_{2,1}}{Pr}\left(\sum\frac{(\boldsymbol{n}_2\cdot\boldsymbol{n}_1)}{9\,Vol}\right)\Delta_{2,1}T, \;\; \bar{\boldsymbol{u}}_{2,1} = \frac{\boldsymbol{u}_2+\boldsymbol{u}_1}{2} \tag{42}$$

The EBV matrix $\boldsymbol{E}_{2,1}$ in Eq. 42 is the same as in the momentum equations (Eq. 37), and summation is again over all tetrahedra that share the edge $[\boldsymbol{p}_1, \boldsymbol{p}_2]$.

There is a rigorous proof [19] that, for any interior edge fully surrounded by tetrahedra, the EBV matrix collected over all tetrahedra that surround and share the edge is symmetric, $\boldsymbol{E}_{i,j} = \boldsymbol{E}_{j,i}$. Thus, there are only six independent coefficients corresponding to the diagonal and upper triangular part of the EBV matrix. For boundary edges that are not fully surrounded by tetrahedra, the EBV matrices corresponding to the two edge endpoints of the edge $[\boldsymbol{p}_i, \boldsymbol{p}_j]$ are transpose of each other, $\boldsymbol{E}_{i,j} = \boldsymbol{E}_{j,i}^T$. Thus, only one EBV matrix (nine EBV coefficients) needs to be stored for boundary edges.

Finally, the trace (the sum of diagonal elements) of $\boldsymbol{M}_{j,i}$ (Eq. 37) can be expressed as follows.

$$tr(\boldsymbol{M}_{j,i}) = 3(\boldsymbol{n}_j\cdot\boldsymbol{n}_i) - \frac{2}{3}(\boldsymbol{n}_j\cdot\boldsymbol{n}_i) + (\boldsymbol{n}_j\cdot\boldsymbol{n}_i) = \frac{10}{3}(\boldsymbol{n}_j\cdot\boldsymbol{n}_i) \tag{43}$$

Thus, the heat-flux EBV coefficient is proportional to $tr(\boldsymbol{E}_{j,i})$ and requires no additional storage. In summary, six EBV coefficients per edge are needed to compute the EBV meanflow residual contributions from edges fully surrounded by tetrahedra, and nine EBV coefficients per edge are needed to compute the contributions from boundary edges.

3.2 EBV correction terms

The Taylor expansion for CBV and EBV viscous fluxes has been computed on a 3D regular-grid stencil described in Table 4 using triangles listed in Eq. 23. The specific formulas are not shown because of its complexity. As expected the leading terms of the Taylor expansion computed for the CBV discretization (Eqs. 36 and 41) are second-order terms. In the view of the truncation-error analysis for the EBV diffusion discretization described in Section 2.3.2, similar results were expected for the EBV discretization of the Navier-Stokes viscous fluxes. However, although there are no first-order terms, the Taylor expansion of the EBV discretization of Eqs. 39 and 42 shows the following zeroth-order error terms.

$$x-\text{momentum}: \frac{5}{6}\left[\partial_x\mu\,\partial_y v - \partial_y\mu\,\partial_x v + \partial_x\mu\,\partial_z w - \partial_z\mu\,\partial_x w\right], \tag{44}$$

$$y-momentum: \frac{5}{6}\left[\partial_y\mu\,\partial_x u - \partial_x\mu\,\partial_y u + \partial_y\mu\,\partial_z w - \partial_z\mu\,\partial_y w\right], \tag{45}$$

$$z-\text{momentum}: \frac{5}{6}\left[\partial_z\mu\,\partial_x u - \partial_x\mu\,\partial_z u + \partial_z\mu\,\partial_y v - \partial_y\mu\,\partial_z v\right], \tag{46}$$

$$\text{energy}: \frac{5}{6}\{2\mu\left[\left(\partial_x u\,\partial_y v+\partial_x u\,\partial_z w+\partial_y v\,\partial_z w\right)-\left(\partial_y u\,\partial_x v+\partial_z u\,\partial_x w+\partial_z v\,\partial_y w\right)\right]$$
$$+\partial_x\mu\left[u\left(\partial_y v+\partial_z w\right)-v\,\partial_y u-w\,\partial_z u\right]$$
$$+\partial_y\mu\left[v(\partial_x u+\partial_z w)-u\,\partial_x v-w\,\partial_z v\right]$$
$$+\partial_z\mu\left[w\left(\partial_z w+\partial_y v\right)-u\,\partial_x w-v\,\partial_y v\right]\}. \quad (47)$$

To recover consistency and second-order accuracy the operators compensating Eqs. 44-47 have been implemented as correction source terms in the corresponding residual equations. The spatial derivatives of the viscosity and vorticity are computed at grid points using a least-squares method based on the nearest-neighbor stencils.

3.3 Method of manufactured solutions

The method of manufactured solution has been implemented in FUN3D and is used here to assess the order of accuracy for the CBV and EBV methods. The manufactured solution for the Navier-Stokes equations takes the following form.

$$\boldsymbol{w}=\boldsymbol{c_0}+\boldsymbol{c_s}\exp\left(\boldsymbol{c_x}x+\ \boldsymbol{c_y}y+\ \boldsymbol{c_z}z\ \right) \quad (48)$$

Here, $\boldsymbol{w}=(\rho,\boldsymbol{u},p)$ is differentiable solution vector, $\boldsymbol{c_0},\boldsymbol{c_s},\boldsymbol{c_x},\boldsymbol{c_y}$ and $\boldsymbol{c_z}$ are vectors of selected parameters. The specific parameters chosen for this study are the following.

$$\boldsymbol{c_0}=\begin{bmatrix}2.0\\1.0\\-3.0\\2.0\\1.0\end{bmatrix},\quad \boldsymbol{c_s}=\begin{bmatrix}1.0\\1.0\\0.0\\1.0\\2.0\end{bmatrix},\quad \boldsymbol{c_x}=\begin{bmatrix}1.0\\2.0\\2.0\\-1.0\\2.0\end{bmatrix},\quad \boldsymbol{c_y}=\begin{bmatrix}2.0\\-1.0\\3.0\\-2.0\\-2.0\end{bmatrix},\quad \boldsymbol{c_s}=\begin{bmatrix}1.5\\-1.5\\2.5\\-1.5\\-1.0\end{bmatrix} \quad (49)$$

This manufactured solution is the exact solution to the following nonhomogeneous steady-state PDE.

$$\frac{d\boldsymbol{F}(\boldsymbol{w})}{dx}+\frac{d\boldsymbol{G}(\boldsymbol{w})}{dy}+\frac{d\boldsymbol{H}(\boldsymbol{w})}{dz}=\boldsymbol{f} \quad (50)$$

Here, $\boldsymbol{F},\boldsymbol{G}$, and $\boldsymbol{H}$ are defined in Eq. 32 and $\boldsymbol{f}$ is the vector known analytically.

The truncation error is estimated as the residual of discretized Eq. 50 evaluated for the manufactured solution restricted to 3D stencil of grid points. Small regular tetrahedral grids of $11\times11\times11$ grid points are generated on a sequence of shrinking isotropic cubes. The initial cube has the edge length of one. The edge of each next smaller cube is half of the edge of the previous bigger cube, implying that the characteristic mesh size is also reduced by factor two. Figure 5 illustrates the grids on the two largest cubes. The central point in all cubes is $\boldsymbol{p_c}=(0.5,0.5,0.5)$, which is surrounded by 24 tetrahedra and has 14 edge-connected neighbors forming the stencil. The discrete solutions on each grid are injected from the manufactured solution. Discrete residuals of the conservation laws are computed at $\boldsymbol{p_c}$ as flux balance divided by the median-dual control volume constructed around $\boldsymbol{p_c}$. The reference Mach number is 0.3, the Reynolds number per unit length is one. Inviscid fluxes of the residuals use least-squares solution gradients computed at the grid points, a second-order solution reconstruction to the edge median, and Roe's Riemann solver. The same inviscid fluxes are used with the CBV and EBV methods.

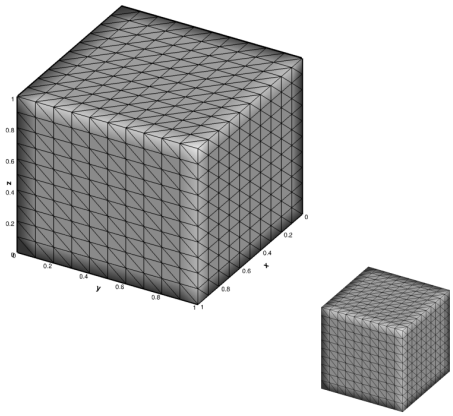


Figure 5. Regular tetrahedral grids for truncation-error analysis.

Figure 6 shows grid convergence of residuals computed with the CBV method (denoted as _CBV) and two EBV methods, with (_EBV) and without (_EBV_no_corr) correction. Lines indicating first- and second-order slopes are shown for reference. The magnitudes of the x-momentum (R2) and energy (R5) conservation residuals are shown versus the characteristic mesh size that is computed as cubic root of the control volume. The truncation errors of the CBV method and the corrected EBV method are second order and overplotted on all grids. The truncation error of the EBV method without correction is similar to the other truncation errors on the coarser grids, but clearly deteriorates on fine grids.

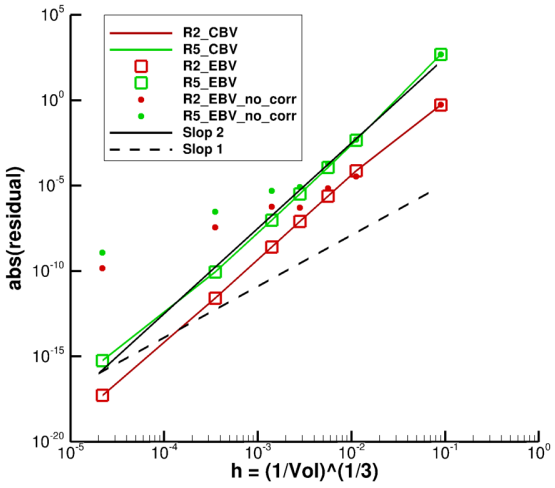
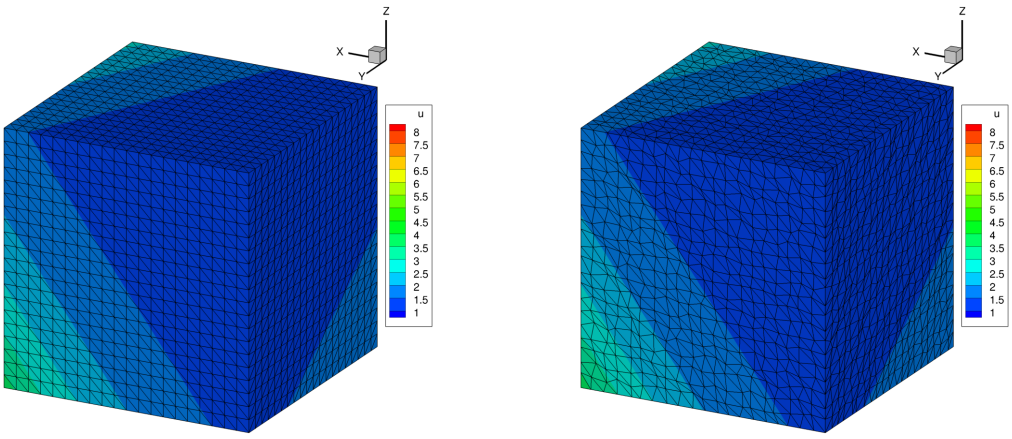


Figure 6. Truncation errors for x-momentum and energy conservation equations.

The same manufactured solution (Eqs. 48, 49) and flow conditions are used to assess grid convergence of the discretization error. For discretization error analysis, a family of five uniformly refined nested regular grids is generated on the unit isotropic cube. The coarsest grid in the family has $11 \times 11 \times 11$ grid points; the finest grid has $161 \times 161 \times 161$ grid points. Irregular grids are derived by 3D random perturbation of the points of the corresponding regular grids. The boundary points are not allowed to move from the original boundary planes. Discrete solutions at all boundary points are overspecified from the manufactured solution. Representative regular and irregular grids are shown in Fig. 7; grids are colored with contours of the x-velocity component.

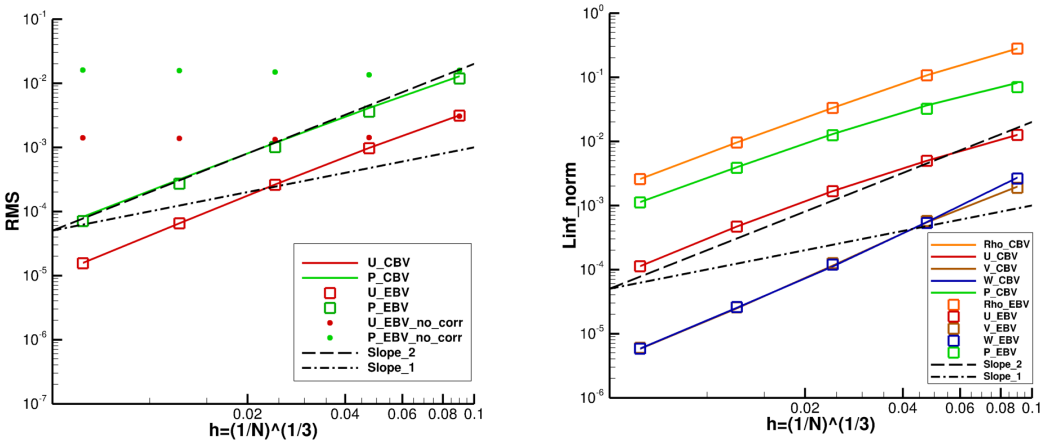


a. Regular grid b. Irregular grid

Figure 7. Regular and irregular grids for discretization-error analysis.

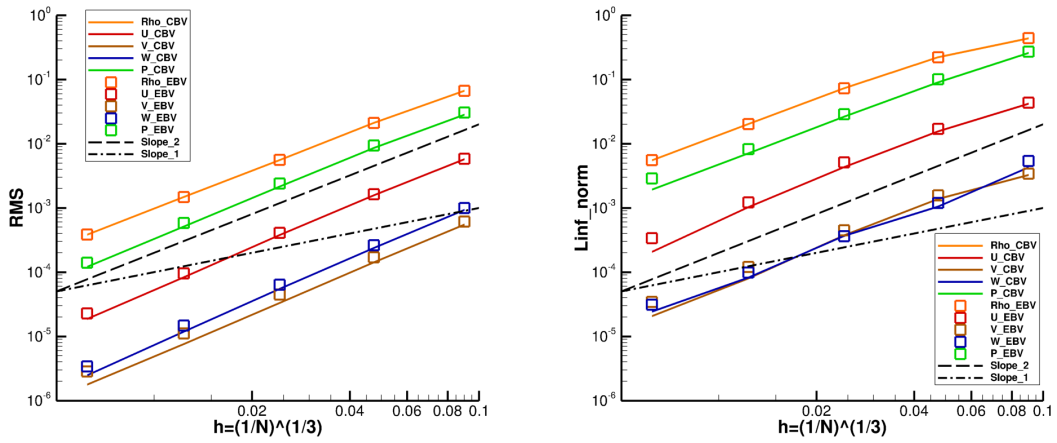
Figure 8 shows grid convergence of the root-mean-square (RMS) and L-infinity (maximum) norms of the discretization error on the regular grids. Discretization errors for the CBV and corrected EBV solutions converge with second order and close to each other in both the norms. Discretization errors for solutions of the EBV method with no corrections deteriorate to the zeroth order.

Figure 9 shows discretization errors of the CBV and corrected EBV solutions on irregular grids. All discretizations errors converge with second order in both the RMS and L-infinity norms. On the same grids, the magnitude of the CBV discretizations error is somewhat smaller than the magnitude of the discretization error of solutions computed with the corrected EBV method. The main conclusion is that adding correction terms to the EBV discretization fully recovers second-order accuracy.



a. RMS norm b. L_∞ norm

Figure 8. Discretization errors on regular grids.



a. RMS norm b. L_∞ norm

Figure 9. Discretization errors for the irregular mesh.

4 RANS solutions

Previous simulations of a subsonic separated flow around a hemisphere-cylinder configuration reported in Refs. [18, 19] have been repeated to systematically assess the effect of the EBV correction terms on RANS solutions. The family of the nested tetrahedral grids uses the same T1-T3 grids and adds a finer grid (T0). The grid statistics and the number of CPU cores are shown in Table 1.

Table 1. Family of tetrahedral grids for hemisphere cylinder.

Grid	Points	Cells	Edges	CPU cores
T0	71,368,353	424,673,280	497,217,184	400
T1	8,995,153	53,084,160	62,373,200	320
T2	1,143,081	6,635,520	7,852,072	40
T3	147,637	829,440	995,444	5

Figure 10 shows the volume and surface meshes corresponding to the T3 grid, where red color indicates the cylinder surface, blue color shows the symmetry boundary, green color shows the outflow boundary and orange color marks the farfield boundary. Three sets of solutions are computed on each grid: the CBV solution, the EBV solution with correction terms, and the EBV_no_corr solution without correction terms. The latter set corresponds to tetrahedral-grid solutions computed previously [18, 19]. All solutions use the same UMUSCL $\kappa = 0.75$ discretization of inviscid fluxes and the same boundary conditions.

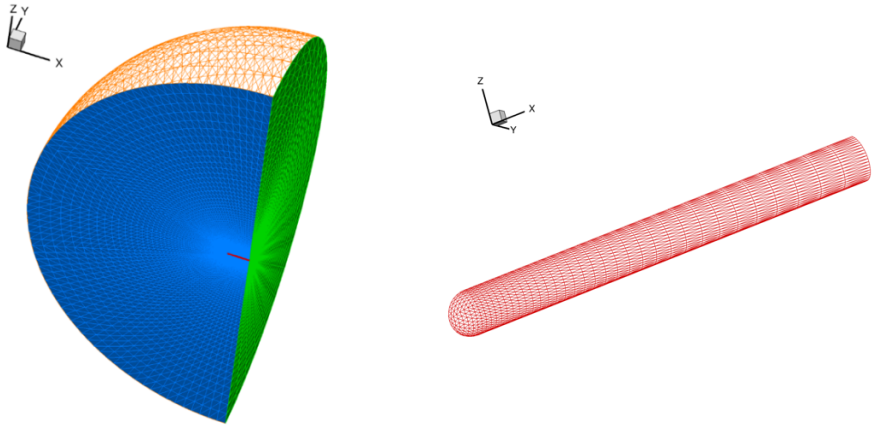
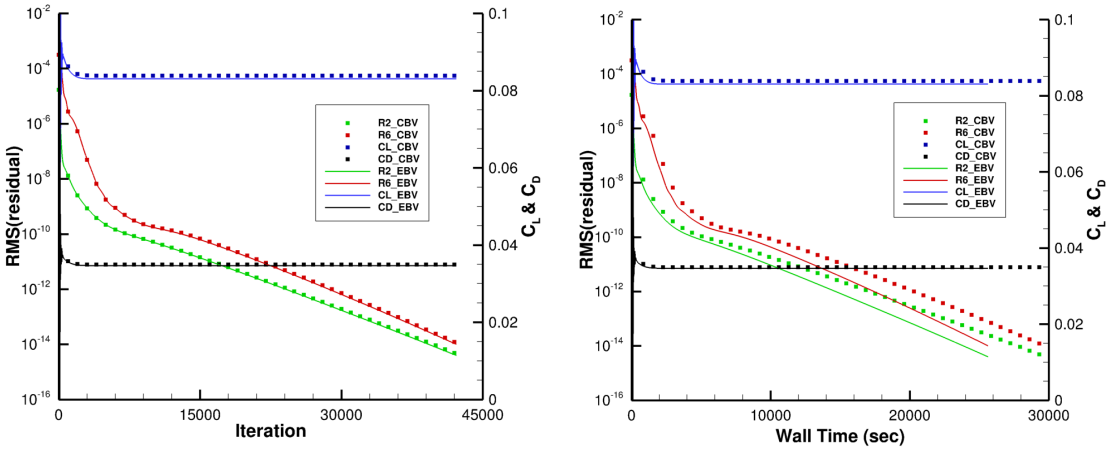


Figure 10. Volume and surface mesh for benchmark flow around hemisphere cylinder.

The baseline iterative solver of FUN3D is used to compute all solutions. The iterations stop when the RMS residual norm for all equations is below 10^{-14}. The CFL number is set to 100. The decoupled linear system is solved using 30 multicolor sweeps for the meanflow and 15 multicolor sweeps for the SA-neg equation. Figure 11 shows convergence of the aerodynamic coefficients (CD and CL denote the drag and lift coefficients) and the RMS norm of the x-momentum (R2) and SA-neg (R6) residuals of the CBV and EBV solutions on the T2 grid. As expected, the EBV and CBV methods show almost identical convergence per iteration and a close agreement between the converged aerodynamic coefficients. In Fig. 11a, the plots of the EBV and CBV convergence versus iterations are hardly distinguishable and similar to those reported in Refs. [18, 19], where the EBV method did not incorporate correction terms. Table 2 shows the converged aerodynamic coefficients for all four tetrahedral grids. The bold font highlights matching digits in the EBV and EBV_no_cor coefficients computed on the same grids. The difference between the lift and drag coefficients predicted from the same-grid EBV solutions with and without correction terms are very small. The maximum difference of 0.1% is observed for the drag coefficient on the T0 grid, the difference is much less on coarser grids. The difference between the CBV and EBV aerodynamic coefficients is also small, less than 1%, but an order of magnitude greater than the difference between the EBV coefficients. The iterative convergence history versus wall time is shown in Fig. 11b. The EBV method takes less time per iteration than the CBV method and significantly reduces the time to convergence.



a. Nonlinear iterations, T2 grid b. Wall time, T2 grid

Figure 11. Baseline iterations.

Table 2. Aerodynamic coefficients.

Grid	Lift Coefficient			Drag Coefficient		
	CBV	EBV_no_corr	EBV	CBV	EBV_no_corr	EBV
T0	0.0853535	**0.0847**002	**0.0847**416	0.0344473	**0.034**1969	**0.034**2351
T1	0.0848187	**0.0840**070	**0.0840**053	0.0343821	**0.0340**752	**0.0340**965
T2	0.0838571	**0.0830**166	**0.0830**149	0.0349403	**0.0346**277	**0.0346**446
T3	0.0845361	**0.083**7683	**0.083**8032	0.0383544	**0.038**0831	**0.038**1029

Table 3 shows the total iteration number and the wall time to achieve the converged solutions for three tetrahedral grids. The T0-grid data are not shown because solutions on the T0 grid much larger partitions (more grid points per CPU core) and different communication-to-computation ratio. The T0 timing is not directly comparable with the timing of solutions on coarser grids that were computed in similar computing environments. The "smart" Jacobian update is activated for all simulations. The Jacobian update is scheduled depending on residual reductions in the previous nonlinear iterations, resulting in variation between iterations and time to solution. Overall, the cost of the EBV correction source terms is hardly noticeable, the time-per-iteration fluctuations are comparable with usual fluctuations in the CPU core performance, which are roughly 2%. Similar to the results reported in Refs. [18, 19], the EBV method achieves 10-30% speedup over the CBV method depending on the frequency of the Jacobian update.

Table 3. Time-to-solution statistics.

Grid	Iterations			Wall time (sec)			EBV speed-up	
	CBV	EBV_no_corr	EBV	CBV	EBV_no_corr	EBV	No_corr	With_corr
T1	37491	37142	36913	28031	24153	23745	13.8%	15.3%
T2	42528	42308	42188	29671	25661	25623	13.5%	13.6%
T3	28784	28332	28337	12156	10117	10356	16.8%	14.8%

5 Concluding remarks

A recently developed novel, efficient, edge-based viscous (EBV) discretization method is applied and analyzed for scalar diffusion and viscous fluxes of the Navier-Stokes equations discretized on simplicial grids. The EBV method is compared with a cell-based viscous (CBV) method that is equivalent to a Galerkin finite-element method and represents the current state of the art. The complexity analysis for scalar diffusion shows that the complexity of the EBV method is at least 20 times lower than the complexity of the CBV method. The truncation-error analysis for linear and nonlinear diffusion equations indicates second-order accuracy of the EBV method in the two dimensions (2D) and in the three dimensions (3D). The method of manufactured solutions confirms second-order accuracy of EBV solutions on regular and irregular grids.

Truncation-error analysis of the EBV method for the Navier-Stokes equations indicates that some correction terms are needed to ensure second-order accuracy for the momentum and energy conservation equations. The correction terms are combinations of products of spatial derivatives of viscosity and velocity which vanish for scalar diffusion. The EBV correction terms have been implemented as source terms accompanying the EBV discretization of the viscous fluxes in the corresponding conservation equations. Truncation-error analysis and discretization-error analysis conducted by the method of manufactured solutions confirm second-order accuracy of the EBV discretizations with correction terms and the corresponding solutions. Although, without correction terms, accuracy clearly deteriorates in the EBV truncation errors and the EBV discretization errors for a manufactured solution, only minimal differences are observed for the EBV Reynolds-averaged Navier-Stokes solutions computed with and without correction terms for flow conditions and grids that are used in practical applications and verification studies.

References

[1] T.J. Barth. Numerical aspects of computing viscous high Reynolds number flows on unstructured meshes. AIAA 91–0721. https://doi.org/10.2514/6.1991-721

[2] H. Luo, J.D. Baum, R. Löhner, and J. Cabello. Adaptive edge-based finite element scheme for the Euler and Navier-Stokes equations on unstructured grids. AIAA 93-0336. https://doi.org/10.2514/6.1993-336

[3] W.K. Anderson and D.L. Bonhaus. An implicit upwind algorithm for computing turbulent flows on unstructured grids. Comput. Fluids, 23(1):1-21, 1994. https://doi.org/10.1016/0045-7930(94)90023-X

[4] P. Eliasson. EDGE. A Navier-Stokes solver for unstructured grids. FOI-R-0298-SE, Dec. 2001. https://www.foi.se/rest-api/report/FOI-R--0298--SE

[5] D.J. Mavriplis. Grid resolution study of a drag prediction workshop configuration using the NSU3D unstructured mesh solver. AIAA 2005-4729. https://doi.org/10.2514/6.2005-4729

[6] D. Schwamborn, T. Gerhold, and R.K. Heinrich. The DLR TAU code: recent applications in research and industry. ECCOMAS CFD 2006, 619. https://elib.dlr.de/22421/

[7] M.E. Braaten and S.D. Connel. “Three-dimensional unstructured adaptive multigrid scheme for the Navier-Stokes equations. AIAA J., 34(2):281–290, 1996. https://doi.org/10.2514/3.13062

[8] A. Haselbacher and J. Blazek. Accurate and efficient discretization of Navier-Stokes equations on mixed grids. AIAA J., 38(11):2094–2102, 2000. https://doi.org/10.2514/2.871

[9] H. Nishikawa. Beyond interface gradient: a general principle for constructing diffusion schemes.

AIAA 2010-5093. https://doi.org/10.2514/6.2010-5093
[10] 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, K.B. Thompson, A.C. Walden, L. Wang, and W.A. Wood. FUN3D manual: 13.7. NASA TM 2020-5010139. https://fun3d.larc.nasa.gov/chapter-2.html#manual_13_7
[11] B. Diskin, N.N. Ahmad, W.K. Anderson, J. M. Derlaga, M.J. Pandya, C. L. Rumsey, L. Wang, S.L. Wood, Y. Liu, H. Nishikawa, and M.C. Galbraith. Verification test suite for Spalart-Allmaras QCR2000 turbulence model. AIAA 2021-1552. https://doi.org/10.2514/6.2021-1552
[12] B. Diskin, W.K. Anderson, M.J. Pandya, C.L. Rumsey, J.L. Thomas, Y. Liu, and H. Nishikawa. Grid convergence for three dimensional benchmark turbulent flows. AIAA 2018-1102. https://doi.org/10.2514/6.2018-1102
[13] C.L. Rumsey, J.-R. Carlson, and N.N. Ahmad, "FUN3D juncture flow computations compared with experimental data. AIAA 2019-0079. https://doi.org/10.2514/6.2019-0079
[14] K.S. Abdol-Hamid, J.-R. Carlson, C.L. Rumsey, E.M. Lee-Rausch, and M.A. Park. Sixth drag prediction workshop results using FUN3D with k-kL-MEAH2015 turbulence model. J. Airc., 55(6): 1458-1468, 2018. https://doi.org/10.2514/1.C034481
[15] B. Diskin, J.L. Thomas, E.J. Nielsen, H. Nishikawa, and J.A. White. Comparison of node-centered and cell-centered unstructured finite-volume discretizations: viscous fluxes. AIAA J., 48(7):1326-1338. 2010. https://doi.org/10.2514/1.44940
[16] J.L. Thomas, B. Diskin, and C.L. Rumsey. Towards verification of unstructured-grid solvers. AIAA J., 46(12):3070-3079, 2008. https://doi.org/10.2514/1.36655
[17] B. Diskin and J.L. Thomas. Accuracy analysis for mixed-element finite-volume discretization schemes. NIA Report 2007-08. https://fun3d.larc.nasa.gov/papers/nia_2007_08.pdf
[18] Y. Liu, B. Diskin, W.K. Anderson, E.J. Nielsen, and L. Wang. Edge-based viscous method for node-centered formulations. AIAA 2021–2728. https://doi.org/10.2514/6.2021-2728
[19] Y. Liu, B. Diskin, H. Nishikawa, W.K. Anderson, E.J. Nielsen, and L. Wang. Edge-based viscous method for node-centered finite-volume formulations," AIAA J. (submitted 2/11/2022).
[20] Y. Liu, B. Diskin, H. Nishikawa, W.K. Anderson, G.C. Nastac, E.J. Nielsen, and L. Wang. Edge-based viscous method for mixed-element node-centered finite-volume solvers. AIAA 2022-4083. https://doi.org/10.2514/6.2022-4083.
[21] J.C. Tannehill, D.A. Anderson, and R.H. Pletcher. *Computational Fluid Mechanics and Heat Transfer*, third edition. Taylor & Francis, 2012.
[22] F.M. White. *Viscous Fluid Flow*. McGraw Hill, New York, 1974, p. 28.