

A VERTEX-BASED HIGH-ORDER FINITE-VOLUME SCHEME FOR THREE-DIMENSIONAL COMPRESSIBLE FLOWS ON TETRAHEDRAL MESH

MARC R.J. CHAREST¹, THOMAS R. CANFIELD²,
NATHANIEL R. MORGAN³, JACOB WALTZ⁴,
AND JOHN G. WOHLBIER⁵

¹ Los Alamos National Laboratory, Los Alamos, NM, U.S.A., charest@lanl.gov

² Los Alamos National Laboratory, Los Alamos, NM, U.S.A., canfield@lanl.gov

³ Los Alamos National Laboratory, Los Alamos, NM, U.S.A., nmorgan@lanl.gov

⁴ Los Alamos National Laboratory, Los Alamos, NM, U.S.A., jwaltz@lanl.gov

⁵ Los Alamos National Laboratory, Los Alamos, NM, U.S.A., wohlbier@lanl.gov

Key words: Numerical Algorithms, Computational Fluid Dynamics, High-Order Methods, Compressible Flows, Shock Hydrodynamics.

Abstract. High-order discretization methods offer the potential to reduce the computational cost associated with modelling compressible flows. However, it is difficult to obtain accurate high-order discretizations of conservation laws that do not produce spurious oscillations near discontinuities, especially on multi-dimensional unstructured mesh. To overcome this issue, a novel, high-order, central essentially non-oscillatory (CENO) finite-volume method was proposed for tetrahedral mesh. The proposed unstructured method is vertex-based, which differs from existing cell-based CENO formulations, and uses a hybrid reconstruction procedure that switches between two different solution representations. It applies a high-order k -exact reconstruction in smooth regions and a limited linear one when discontinuities are encountered. Both reconstructions use a single, central stencil for all variables, making the application of CENO to arbitrary unstructured meshes relatively straightforward. The new approach was applied to the conservation equations governing compressible flows and assessed in terms of accuracy and computational cost. For all problems considered, which included various function reconstructions and idealized flows, CENO demonstrated excellent reliability and robustness. High-order accuracy was achieved in smooth regions and essentially non-oscillatory solutions were obtained near discontinuities. The high-order schemes were also more computationally efficient for high-accuracy solutions, i.e., they took less wall time than the lower-order schemes to achieve a desired level of error.

1 INTRODUCTION

Finite-volume methods are a popular discretization technique for computational fluid dynamics, especially for compressible flows. Numerous formulations exist, and one of the main differing characteristics is the approach used to discretize the computational domain. Either cell- or vertex-based discretizations are typically employed. Cell-based approaches apply conservation laws to the individual elements or cells of the mesh, whereas vertex-based approaches apply them to control volumes constructed surrounding the vertices of the mesh. The choice is not always straightforward, since both techniques are widely used and both have their respective advantages/disadvantages. For example, vertex-based schemes are often favored for use with unstructured tetrahedral mesh since there is approximately 5 to 6 times fewer vertices than elements. But because there are more elements than vertices in a tetrahedral mesh, cell-based schemes have more degrees of freedom. As such, cell-based schemes tend to be slightly more accurate on tetrahedral meshes, although this increased accuracy comes at the expense of additional computational effort [1]. The direct comparison between the two types of schemes is complicated, however, because they both use different stencils. The larger, denser stencils that vertex-based schemes are more robust and accurate per degree of freedom, which may make vertex centered schemes more computationally efficient for a given accuracy [1]. Nonetheless, whatever the chosen finite-volume formulation, current production codes rely mostly on standard first- or second-order accurate discretization schemes. These discretizations are often not practical for physically-complex, multi-dimensional flows with disparate scales as they tend to exhibit excessive numerical dissipation.

High-order discretization methods for conservation laws have the potential to significantly reduce the cost of modelling physically-complex flows. They offer improved numerical efficiency to obtain high-resolution solutions since fewer computational cells are required to achieve a desired level of accuracy [2]. However, this potential is challenging to fully realize as it is difficult to obtain accurate and robust discretizations of hyperbolic conservation laws near discontinuities [3]. Although there are many different high-order schemes for both structured and unstructured mesh that attempt to address this issue [3–31], there is still no consensus on a robust, efficient, and accurate scheme that deals with the aforementioned issues and is universally applicable to arbitrary meshes.

One promising high-order discretization is the central essentially non oscillatory (CENO) finite-volume approach [32–40]. It was originally developed for two-dimensional structured mesh by Ivan et al. [36–40] and then extended to three-dimensional unstructured mesh by Charest et al. [32–35]. In all formulations, CENO remained both accurate and robust throughout a variety of physically-complex flows. This robustness is provided by a hybrid reconstruction procedure that switches between two algorithms: an unlimited high-order k -exact reconstruction in smooth regions, and a monotonicity-preserving limited piecewise linear reconstruction in regions with discontinuities or shocks. Switching between the two reconstructions is facilitated by a smoothness indicator that measures the ability of the

of the k -exact reconstruction to locally resolve the flow. Fixed central stencils are used for both reconstruction algorithms, which makes its extension to arbitrary unstructured meshes straightforward.

The CENO approach avoids many of the complexities associated with other essentially non-oscillatory (ENO) [3] and weighted ENO (WENO) [8, 10, 11] finite-volume schemes because it does not require a high-order reconstruction on multiple stencils. Both ENO and WENO schemes have difficulty selecting stencils on general multi-dimensional unstructured meshes [5, 6, 9, 41], and some of these stencils produce poorly conditioned linear systems for solution reconstruction [9, 41].

The existing CENO formulations for structured [36–40] and unstructured [32–35] meshes were developed for cell-based finite-volume schemes only. In the present research, CENO was extended to a vertex-based finite-volume discretization for three-dimensional unstructured mesh and applied to solve the equations governing compressible flows. The resulting algorithm was applied to various function reconstructions, as well as steady and unsteady flows, and then analyzed with respect to accuracy and computational cost. This research was performed using CHICOMA, a computational framework for compressible fluid flow, i.e., shock hydrodynamics [42]

2 GOVERNING EQUATIONS

The Euler equations governing compressible fluid flow were considered for the present research. In three space dimensions, these partial-differential equations (PDEs) are given by

$$\frac{\partial}{\partial t} \mathbf{U}(\mathbf{W}) + \vec{\nabla} \cdot \vec{\mathbf{F}}(\mathbf{W}) = \mathbf{S}(\mathbf{W}) \quad (1)$$

where t is the time, \mathbf{U} and \mathbf{W} are the vectors of conserved and primitive variables, respectively, $\vec{\mathbf{F}}(\mathbf{W}) = [\mathbf{E}, \mathbf{F}, \mathbf{G}]$ is the inviscid solution flux dyad, and $\mathbf{S}(\mathbf{W})$ is a vector of source terms. These terms are defined as

$$\begin{aligned} \mathbf{U} &= [\rho, \rho u, \rho v, \rho w, \rho e_t], \\ \mathbf{W} &= [\rho, u, v, w, e], \\ \mathbf{E} &= \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ u(\rho e_t + p) \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} \rho v \\ \rho vu \\ \rho v^2 + p \\ \rho vw \\ v(\rho e_t + p) \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} \rho w \\ \rho wu \\ \rho wv \\ \rho w^2 + p \\ w(\rho e_t + p) \end{bmatrix} \end{aligned}$$

where ρ is the fluid density, p is the pressure, $\vec{v} = (u, v, w)$ is the fluid velocity vector, e is the internal energy, and e_t is the total energy. The total energy is the sum of the internal and kinetic energies, i.e.,

$$e_t = e + \frac{1}{2}(u^2 + v^2 + w^2) \quad (2)$$

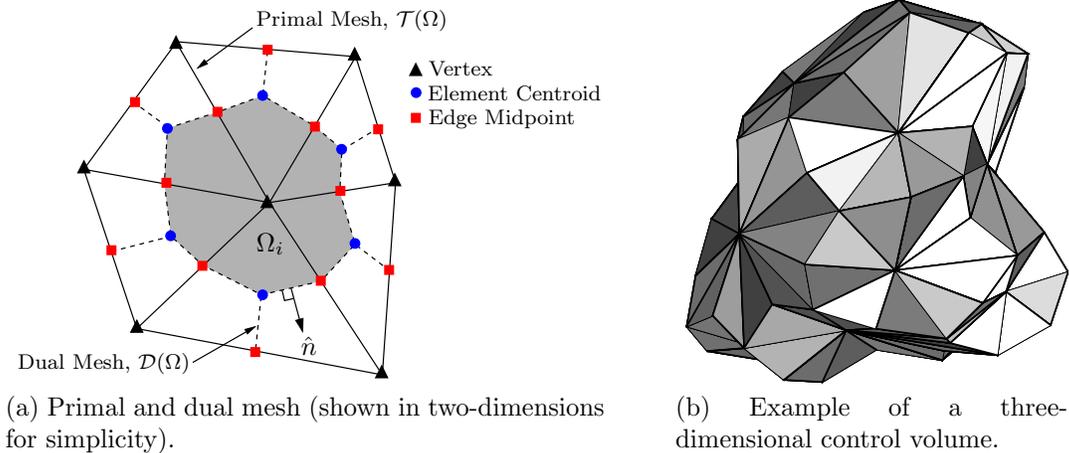


Figure 1: Computational mesh and local control volume configuration.

Internal energy is related to pressure and density through the following relation for an ideal gas:

$$e = \frac{p}{\rho(\gamma - 1)} \quad (3)$$

where γ is the ratio of specific heats. The sound speed of an ideal gas is given by

$$a = \sqrt{\gamma p / \rho} \quad (4)$$

Although Eqs. (3) and (4) describe an ideal gas, the numerical formulation described herein is designed to support arbitrary analytic or tabular equations of state. Unless otherwise specified, $\gamma = 1.4$.

The source term vector, \mathbf{S} , is typically treated as zero throughout this work. There is one particular case — which will be discussed in the following sections — where it was used to generate a known solution for validation purposes.

3 CENO FINITE-VOLUME SCHEME

In the proposed vertex-based finite-volume approach, the physical domain, Ω , was discretized into non-overlapping, finite-sized control volumes, Ω_i , such that

$$\Omega = \cup \Omega_i \quad (5)$$

$$\Omega_i \cap \Omega_j \quad \text{for } i \neq j \quad (6)$$

The individual control volumes were formed by constructing the median dual, $\mathcal{D}(\Omega)$, of a three-dimensional triangulation of the domain, $\mathcal{T}(\Omega)$, which is illustrated in two-dimensions in Fig. 1a. Only primal meshes composed of tetrahedral elements were considered, and they have a corresponding dual mesh composed of complex polyhedrons with

Table 1: Gauss quadrature rules used for integrating over triangles and tetrahedrons.

Reconstruction	Number of Points		Degree of Precision
	Triangle	Tetrahedron	
Constant ($k=0$)	1	1	1
Linear ($k=1$)	1	1	1
Quadratic ($k=2$)	3	4	2
Cubic ($k=3$)	4	8	3
Quartic ($k=4$)	6	14	4

triangular faces. A control volume surrounding a vertex i , Ω_i , was constructed from the polyhedron whose vertices are the centroids of incident tetrahedra and triangles, plus the midpoints of incident edges. A sample control volume surrounding an individual vertex in a three-dimensional tetrahedral mesh is illustrated in Fig. 1b.

Equation (1) was integrated over each individual control volume to give the following system of ordinary differential equations (ODEs) for control-volume-averaged solution quantities, $\bar{\mathbf{U}}_i$:

$$\frac{d\bar{\mathbf{U}}_i}{dt} = -\frac{1}{V_i} \oint_{\partial\Omega_i} (\vec{\mathbf{F}} \cdot \hat{\mathbf{n}}) d\Gamma + \frac{1}{V_i} \int_{\Omega_i} \mathbf{S} d\Omega = \mathbf{R}_i, \quad i = 1, 2, \dots, N_v \quad (7)$$

where N_v is the number of control volumes (i.e., vertices of the primal mesh), V_i is the volume and $\hat{\mathbf{n}}$ is the unit vector normal to the surface of the control volume, $\partial\Omega_i$. Applying Gauss quadrature to evaluate the surface and volume integrals in Eq. (7) produces a set of nonlinear ODEs given by

$$\frac{d\bar{\mathbf{U}}_i}{dt} = -\frac{1}{V_i} \sum_{j=1}^{N_f} \sum_{k=1}^{G_f} [\omega_f \vec{\mathbf{F}} \cdot \hat{\mathbf{n}}]_{i,j,k} + \frac{1}{V_i} \sum_{m=1}^{G_v} [\omega_v \mathbf{S}]_{i,m} = \mathbf{R}_i \quad (8)$$

where N_f is the number of faces, G_f and G_v are the number of quadrature points and ω_f and ω_v are the corresponding quadrature weights for the face and volume integrals, respectively.

In Eq. (8), the number of quadrature points required for each rule is a direct function of the number of spatial dimensions and the reconstruction order — i.e., the quadrature rule must be able to integrate a k -degree polynomial exactly (k -exactness). Integrating over the individual faces of the polyhedral-shaped control volume is relatively straightforward. Since the faces are triangular, standard quadrature rules for triangles were used. However, general quadrature rules for integrating over complex polyhedrons do not exist. As such, numerical integrals over the volume of these complex elements were evaluated by subdividing the polyhedrons into tetrahedrons and applying standard Gauss quadrature

rules to each individual tetrahedron. The coefficients for the quadrature rules applied herein are as given by Felippa [43] and summarized in Table 1.

3.1 CENO Reconstruction

Evaluating Eq. (8) requires numerically integrating the fluxes and source terms over the control-volumes, and this numerical integration requires interpolating the solution at quadrature points. Only control-volume averages are known in the proposed finite-volume approach, so the solution at these quadrature points was interpolated using the high-order CENO method [32–40]. The reconstruction was applied to the primitive solution quantities, \mathbf{W} , to ensure that both pressure and internal energy remain positive.

3.1.1 k -Exact Reconstruction

The CENO spatial discretization scheme is based on the high-order k -exact least-squares reconstruction technique of Barth [4, 44]. The k -exact reconstruction algorithm begins by assuming that the solution within each control-volume is represented by piecewise Taylor polynomials. In three space dimensions, the polynomials are defined as

$$u_i^k(x, y, z) = \sum_{p=0}^{p+q+r \leq k} \sum_{q=0} \sum_{r=0} (x - x_i)^p (y - y_i)^q (z - z_i)^r D_{pqr} \quad (9)$$

where u_i^k is the reconstructed solution quantity, (x_i, y_i, z_i) is the geometric reference point, k is the degree of the piecewise polynomial interpolant, and D_{pqr} are the unknown coefficients of the Taylor series expansion. Any geometric reference point can be chosen; the vertex about which the control volume is constructed was used here.

The following conditions were applied to determine the unknown coefficients: (i) the mean or average value within the computational volume must be preserved; (ii) the solution reconstruction must reproduce polynomials of degree $\leq k$ exactly (i.e., k -exactness); and (iii) the reconstruction must have compact support. The first condition introduces a constraint on the reconstruction which states that

$$\bar{u}_i = \frac{1}{V_i} \int_{\Omega_i} u_i^k(x, y, z) \, d\Omega \quad (10)$$

where \bar{u}_i is the control-volume average in Ω_i . Additional constraints are introduced by the second condition, requiring that

$$u_i^k(x, y, z) = u_{\text{exact}} + \mathcal{O}(h^{k+1}) \quad (11)$$

in the vicinity of Ω_i . The length scale, h , is defined as the maximum diameter of the control-volume circumspheres in the vicinity of Ω_i . From Eq. (11), the reconstruction polynomial for Ω_i must also recover the averages of neighboring control volumes. That is,

$$\bar{u}_j = \frac{1}{V_j} \int_{\Omega_j} u_i^k(x, y, z) \, d\Omega + \mathcal{O}(h^{k+1}) \quad \forall j \in S_{\text{neigh},i} \quad (12)$$

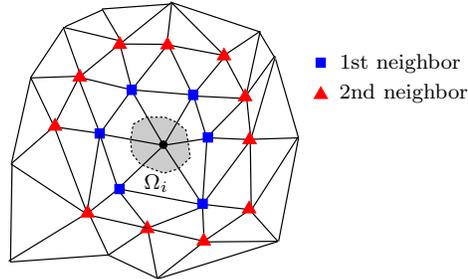


Figure 2: Supporting stencil for the reconstruction in Ω_i . First- and second-level neighbors are indicated in the figure.

where $S_{\text{neigh},i}$ is the collection of control-volume indices in the vicinity of Ω_i . This property ensures that the difference between the predicted solution states at control-volume interfaces diminishes at a rate proportional to h^{k+1} .

The third condition merely specifies the number and location of neighbors included in the reconstruction. For a compact stencil, the minimum number of neighbors is equal to the number of unknowns minus one (because of the constraint imposed by Eq. (10)). For any type of mesh, the total number of unknown coefficients for a particular order is given by

$$N = \frac{1}{d!} \prod_{n=1}^d (k + n) \quad (13)$$

where d represents the number of space dimensions. In three-dimensions, there are four, ten, twenty and thirty-five unknown coefficients for $k=1$, $k=2$, $k=3$ and $k=4$, respectively.

A sample stencil for Ω_i is illustrated in two-dimensions in Fig. 2. The stencil was constructed by recursively selecting nearest neighbors until at least the minimum number of neighbors was met. The closest neighbors were selected first, and then, if more neighbors were required, the next nearest neighbors were selected. This process continued until the stencil was deemed sufficient. Additional neighbors were typically included to ensure that the stencil was not biased in any particular direction and that the reconstruction remained reliable on poor quality meshes with high aspect ratio cells [44]. In the present research, a stencil at least 1.5 times larger than the minimum size was employed. The employed stencil sizes are listed in Table 2.

The constraints given by Eqs. (10) and (12) create an over-determined system of linear equations of the form,

$$\mathbf{A}\mathbf{X} = \mathbf{B} \quad (14)$$

where \mathbf{A} is the coefficient matrix, \mathbf{X} is the vector of unknown polynomial coefficients, and \mathbf{B} is a vector which depends on control volume averages. Since the system is over-determined, a least-squares solution for \mathbf{X} was obtained in each control-volume. Equation (10) must be strictly enforced, while a minimum-error solution to the remaining

Table 2: Minimum stencil sizes used for reconstructions.

Reconstruction	Minimum Stencil Size	
	Theoretical	Actual
Linear ($k=1$)	3	5
Quadratic ($k=2$)	9	14
Cubic ($k=3$)	19	29
Quartic ($k=4$)	34	51

constraint equations was sought. The final form of Eq. (14) for each control volume i was derived from Eqs. (10) and (12). It is given by

$$\begin{bmatrix} w_{1i} \widehat{x^0 y^0 z^1}_{1i} & \cdots & w_{1i} \widehat{x^p y^q z^r}_{1i} & \cdots & w_{1i} \widehat{x^k y^0 z^0}_{1i} \\ \vdots & & \vdots & & \vdots \\ w_{ji} \widehat{x^0 y^0 z^1}_{ji} & \cdots & w_{ji} \widehat{x^p y^q z^r}_{ji} & \cdots & w_{ji} \widehat{x^k y^0 z^0}_{ji} \\ \vdots & & \vdots & & \vdots \\ w_{ni} \widehat{x^0 y^0 z^1}_{ni} & \cdots & w_{ni} \widehat{x^p y^q z^r}_{ni} & \cdots & w_{ni} \widehat{x^k y^0 z^0}_{ni} \end{bmatrix} \cdot \begin{bmatrix} D_{001} \\ \vdots \\ D_{pqr} \\ \vdots \\ D_{k00} \end{bmatrix} = \begin{bmatrix} w_{1i} (\bar{u}_1 - \bar{u}_i) \\ \vdots \\ w_{ji} (\bar{u}_j - \bar{u}_i) \\ \vdots \\ w_{ni} (\bar{u}_n - \bar{u}_i) \end{bmatrix} \quad (15)$$

where n is the number of neighbors in the stencil, $S_{\text{neigh},i}$, and w_{ji} are least-squares weights. The geometric coefficients, $\widehat{x^p y^q z^r}_{ji}$, are given by

$$\widehat{x^p y^q z^r}_{ji} = \widetilde{x^p y^q z^r}_{ji} - \overline{x^p y^q z^r}_i \quad (16)$$

where

$$\widetilde{x^p y^q z^r}_{ji} = \frac{1}{V_j} \int_{\Omega_j} (x - x_i)^p (y - y_i)^q (z - z_i)^r d\Omega \quad (17)$$

$$\overline{x^p y^q z^r}_i = \frac{1}{V_i} \int_{\Omega_i} (x - x_i)^p (y - y_i)^q (z - z_i)^r d\Omega \quad (18)$$

Only the geometric moments about each individual control-volume, $\overline{x^p y^q z^r}_i$, were stored prior to solving Eq. (8). The remaining geometric coefficients were computed using a binomial expansion [10, 12]:

$$\widetilde{x^p y^q z^r}_{ji} = \frac{1}{V_j} \int_{\Omega_j} [(x - x_j) + (x_j - x_i)]^p \cdot [(y - y_j) + (y_j - y_i)]^q \cdot [(z - z_j) + (z_j - z_i)]^r d\Omega \quad (19)$$

$$= \sum_{a=0}^p \sum_{b=0}^q \sum_{c=0}^r \binom{p}{a} \binom{q}{b} \binom{r}{c} \cdot (x_j - x_i)^a \cdot (y_j - y_i)^b \cdot (z_j - z_i)^c \cdot \overline{x^{p-a} y^{q-b} z^{r-c}}_j \quad (20)$$

Weighting was applied to each individual constraint equation to improve the locality of the reconstruction [45]. The weights for the reconstruction in Ω_i are

$$w_{ji} = \frac{1}{|\vec{x}_j - \vec{x}_i|^p}, \quad (21)$$

where \vec{x}_i and \vec{x}_j are the vertex locations. The exponent, p , was set equal to 1.

The condition of the least-squares problem for the reconstruction coefficients was improved via the application of a simple column scaling [10, 46], which effectively makes the condition number independent of the mesh size and control-volume aspect ratio. Scaling the columns of the matrix \mathbf{A} gives the new linear system

$$(\mathbf{AP})(\mathbf{P}^{-1}\mathbf{X}) = \mathbf{B} \quad (22)$$

where \mathbf{P} is a diagonal matrix of size $N - 1$ whose entries are the inverse of the largest absolute values of each column of \mathbf{A} . The scaling matrix is given by

$$P_{jj} = \frac{1}{\max_{\forall i} |A_{ij}|} \quad i = 1, 2, \dots, n \quad (23)$$

where P_{jj} and A_{ij} are the individual elements of \mathbf{P} and \mathbf{A} , respectively.

A least-squares solution to Eq. (22) was sought using either QR factorization based on Householder transformations or the singular value decomposition (SVD) method [47]. Since the coefficient matrix, \mathbf{A} , and the scaling matrix, \mathbf{P} , only depend on the mesh geometry, they can be inverted and stored prior to solving Eq. (8) [37]. Thus, when SVD was used, the pseudoinverse was stored and polynomial coefficients were simply determined from the following matrix-vector product at each iteration:

$$\mathbf{X} = \mathbf{P}(\mathbf{AP})^\dagger \mathbf{B} \quad (24)$$

where \dagger denotes the pseudoinverse and the matrix $\mathbf{P}(\mathbf{AP})^\dagger$ is the pre-computed and stored result of SVD. This operation was considerably less computationally intensive than performing a full QR factorization or SVD decomposition for each control-volume at every iteration. Once the least-squares solution for \mathbf{X} in Eq. (22) was obtained, the remaining polynomial coefficient, D_{000} , was obtained from Eq. (10).

3.1.2 Reconstruction at Boundaries

To enforce conditions at the boundaries of the computational domain, the least-squares reconstruction was constrained at Gauss quadrature points along the boundary without altering the reconstruction's order of accuracy [12, 13, 37]. The constraints were implemented as Robin-type boundary conditions and are given by

$$f(\vec{x}) = a(\vec{x}) f_D(\vec{x}) + b(\vec{x}) f_N(\vec{x}) \quad (25)$$

where $a(\vec{x})$ and $b(\vec{x})$ are coefficients which define the contribution of the Dirichlet, $f_D(\vec{x})$, and Neumann, $f_N(\vec{x})$, components, respectively. These coefficients are simply $(a, b) = (1, 0)$ for Dirichlet- and $(a, b) = (0, 1)$ for Neumann-type boundary conditions. The Dirichlet condition is expressed as

$$f_D(\vec{x}_g) = u^k(\vec{x}_g) \quad (26)$$

where \vec{x}_g is the location of the Gauss quadrature point. The Neumann condition is

$$\begin{aligned} f_N(\vec{x}_g) &= \vec{\nabla} u^k(\vec{x}_g) \cdot \hat{n}_g \\ &= \sum_{p+q+r=1}^{p+q+r \leq k} \Delta x^{p-1} \Delta y^{q-1} \Delta z^{r-1} [p \Delta y \Delta z n_x + q \Delta x \Delta y n_y + r \Delta x \Delta y n_z] D_{pqr} \end{aligned} \quad (27)$$

where $\Delta(\cdot) = (\cdot)_g - (\cdot)_i$ is the distance between the vertex of the control volume adjacent to the boundary and the Gauss quadrature point, and \hat{n}_g is the outward surface normal at the quadrature point.

Exact solutions to the boundary constraints described by Eq. (25) were sought, which adds linear equality constraints to the original over-determined system (Eq. (15)). This resulting equality-constrained least-squares problem was solved using the method of weights [47]. It was solved in the same manner as described in Section 3.1.1, except the original equations in Eq. (15) were multiplied by an additional weight. The new over-determined linear system with boundary constraints is given by

$$\begin{bmatrix} \epsilon \mathbf{A} \\ \mathbf{C} \end{bmatrix} \cdot [\mathbf{X}] = \begin{bmatrix} \epsilon \mathbf{B} \\ \mathbf{D} \end{bmatrix} \quad (28)$$

where \mathbf{C} and \mathbf{D} are the coefficient matrix and solution vector for the boundary constraints, respectively. A weight, ϵ , equal to 10^{-3} was applied to the original equations defined by Eq. (15), which gives the boundary constraints a large influence.

For boundary conditions where the reconstructed variables are not related, such as inflow/outflow or farfield-type conditions, the constraints were applied separately to each variable. Thus, a separate least-squares problem with equality constraints was set up for each variable and solved independently of the others. More complex boundary conditions involve linear combinations of solution variables that couple the reconstruction coefficients of different variables. For example, the individual velocity components for reflection or solid wall conditions are coupled because $\vec{v} \cdot \hat{n} = 0$. These types of coupled boundary conditions were handled via constraints — which requires reconstructing all variables together — in combination with an appropriately prescribed flux [12, 13, 38, 48].

For coupled boundary conditions, the unknown polynomial coefficients for the uncoupled variables were determined independently first, and then the coupled variables were

reconstructed together. To illustrate this procedure, consider a reflection boundary condition, which was applied in the present study. Along reflecting boundaries,

$$\begin{aligned}\vec{\nabla}\rho \cdot \hat{n} &= 0 \\ \vec{v} \cdot \hat{n} &= 0 \\ \vec{\nabla}e \cdot \hat{n} &= 0\end{aligned}$$

where \hat{n} is a unit vector normal to the boundary. Both ρ and e are independent, so they were reconstructed separately by solving Eq. (28), but the three components of velocity are coupled via a linear combination of each other. The constraints for a zero normal velocity at the boundary are given by

$$u(\vec{x}_g)n_x(\vec{x}_g) + v(\vec{x}_g)n_y(\vec{x}_g) + w(\vec{x}_g)n_z(\vec{x}_g) = 0 \quad (29)$$

As such, the coupled, over-determined linear system for the unknown polynomial coefficients of the three velocity components is as follows:

$$\begin{bmatrix} \epsilon\mathbf{A}_u & \mathbf{0} & \mathbf{0} \\ \mathbf{C}_u & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \epsilon\mathbf{A}_v & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_v & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \epsilon\mathbf{A}_w \\ \mathbf{0} & \mathbf{0} & \mathbf{C}_w \\ \text{Coupled constraints, Eq. (29)} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{X}_u \\ \mathbf{X}_v \\ \mathbf{X}_w \end{bmatrix} = \begin{bmatrix} \epsilon\mathbf{B}_u \\ \mathbf{D}_u \\ \epsilon\mathbf{B}_v \\ \mathbf{D}_v \\ \epsilon\mathbf{B}_w \\ \mathbf{D}_w \\ 0 \end{bmatrix} \quad (30)$$

where the subscripts u , v , and w refer to the solution quantities with which the components of the linear system are associated with.

The prescribed flux at each quadrature point along the reflecting boundary is

$$\mathbf{F}_{\text{reflect}} = [0, pn_x, pn_y, pn_z, 0] \quad (31)$$

where p is calculated at the wall boundary by extrapolating ρ and e .

Solid walls were treated the same as reflecting boundaries, except that no constraints were applied to ρ and e along the boundary, i.e., only $\vec{v} \cdot \hat{n} = 0$ was enforced.

3.1.3 Smoothness Indicator

After performing a k -exact reconstruction in each control volume, the smoothness indicator was computed for every reconstructed variable to identify under-resolved solution content. It was evaluated as [38]

$$S = \frac{\sigma}{\max[(1 - \sigma), \delta]} \frac{\text{SOS} - \text{DOF}}{\text{DOF} - 1} \quad (32)$$

where σ is a smoothness parameter, δ is a tolerance to avoid division by zero (equal to 10^{-8}), DOF is the number of degrees of freedom and SOS is the size of the stencil. The factor, $(\text{SOS} - \text{DOF})/(\text{DOF} - 1)$, adjusts σ to account for the number of polynomial coefficients relative to the size of the reconstruction stencil.

The smoothness parameter is based on the coefficient of determination or R^2 parameter, which is a statistical parameter used for assessing how well lines or curves fit data points [49]. For a control-volume Ω_i , the smoothness parameter is given by

$$\sigma = 1 - \frac{\sum_{\forall j \in S_{\text{neigh},i}} [u_j^k(\vec{x}_j) - u_i^k(\vec{x}_j)]^2}{\sum_{\forall j \in S_{\text{neigh},i}} [u_j^k(\vec{x}_j) - \bar{u}_i]^2} \quad (33)$$

where u is the solution variable of interest. The numerator of the fraction in Eq. (33) measures how well the reconstruction polynomial for Ω_i predicts the values in nearby control volumes, while the denominator in Eq. (33) measures the variance from some reference point — \bar{u}_i in this case — and normalizes σ .

By definition, σ can have a value between negative infinity and one. A value of unity indicates that the solution is smooth whereas a small or negative value indicates large variations in solution content within the reconstruction stencil. An order of magnitude analysis, similar to the ones performed by Ivan and Groth [39] and Charest et al. [35] for cell-based CENO formulations, confirms the correct behavior of σ with changes in mesh size, h . It follows from Eq. (11) that

$$\sigma \approx 1 - \frac{[\mathcal{O}(h^{k+1})]^2}{[\mathcal{O}(h)]^2} \approx 1 - \mathcal{O}(h^{2k}) \quad (34)$$

for smooth solution content, so $\sigma \rightarrow 1$ as $\Delta x \rightarrow 0$ at a rate much faster than the formal order of accuracy of the scheme. Conversely, when the solution is not smooth, σ is much less than unity because

$$\sigma \approx 1 - \frac{[\mathcal{O}(1)]^2}{[\mathcal{O}(1)]^2} \approx 1 - \mathcal{O}(1) \quad (35)$$

Solutions were deemed smooth when the value of S was above a critical value, S_c . Previous studies found that values for S_c between 1000–5000 provided an excellent balance between stability and accuracy [38]. And because of the form of Eq. (32), S grows rapidly as $\sigma \rightarrow 1$, so S tends to be orders of magnitude greater than these cutoff limits in smooth regions. Unless otherwise specified, S_c was equal to 4000.

In cases where the solution was not varying, such as in the free-stream, the smoothness indicator sometimes incorrectly indicated that solutions with small deviations due numerical noise were under-resolved. This occurred because both the denominator and

numerator of the fraction in Eq. (33) approached zero, and σ was close to zero or negative. To alleviate this issue, the solution was automatically deemed smooth if the local variation within the stencil was below a tolerance. That is, if

$$\max_{\forall j \in S_{\text{neigh},i}} |\bar{u}_j - \bar{u}_i| < t_{\text{abs}} u_{\text{ref}} + t_{\text{rel}} \bar{u}_{\text{stencil}}, \quad (36)$$

the smoothness indicator was not computed and the high-order k -exact reconstruction was used. Here, u_{ref} is a reference solution, \bar{u}_{stencil} is the stencil average, $t_{\text{abs}} = 10^{-5}$ is an absolute tolerance, and $t_{\text{rel}} = 10^{-3}$ is a relative tolerance. The reference solution, u_{ref} , was the average value within the computational domain and was only computed once prior to solving the governing ODEs given by Eq. (8).

3.1.4 Limited Piecewise Linear Reconstruction

In regions where the smoothness indicator was below the critical value, monotonicity was preserved by switching to a limited piecewise linear ($k=1$) reconstruction on a smaller stencil (see Table 2). The least-squares reconstruction procedure described by Barth [50] was used in these regions, since it was found to be more computationally efficient than directly solving Eq. (15) via QR factorization or SVD.

The limited piecewise linear representation in each control volume is given by

$$u^{k=1}(\vec{x}) = \bar{u}_i + \phi_i \vec{\nabla} u \cdot (\vec{x} - \vec{x}_{c,i}) \quad (37)$$

where ϕ_i is the slope limiter and $\vec{x}_{c,i}$ is the location of the control-volume centroid. In this particular case, $k=1$, the control-volume-averaged solution is equal to the solution at the centroid, not the solution at the control volume's associated vertex. The two locations do not coincide with each other, i.e., $\vec{x}_{c,i} \neq \vec{x}_i$.

The new over-determined matrix equation for the solution gradients in Ω_i is given by

$$\begin{bmatrix} w_{i1} \Delta x_{i1} & w_{i1} \Delta y_{i1} & w_{i1} \Delta z_{i1} \\ \vdots & \vdots & \vdots \\ w_{ij} \Delta x_{ij} & w_{ij} \Delta y_{ij} & w_{ij} \Delta z_{ij} \\ \vdots & \vdots & \vdots \\ w_{in} \Delta x_{in} & w_{in} \Delta y_{in} & w_{in} \Delta z_{in} \end{bmatrix} \cdot \begin{bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial u}{\partial y} \\ \frac{\partial u}{\partial z} \end{bmatrix} = \begin{bmatrix} w_{i1} (\bar{u}_1 - \bar{u}_i) \\ \vdots \\ w_{ij} (\bar{u}_j - \bar{u}_i) \\ \vdots \\ w_{in} (\bar{u}_n - \bar{u}_i) \end{bmatrix} \quad (38)$$

where $\Delta(\cdot)_{ij} = (\cdot)_j - (\cdot)_i$ is the distance between control volume centroids. This system was solved in a least-squares sense using the Gram-Schmidt process outlined in [50].

Uncoupled Dirichlet- and Neumann-type boundary conditions were incorporated by adding constraint equations to Eq. (38) for each quadrature point:

$$\text{Dirichlet:} \quad \vec{\nabla} u \cdot \Delta \vec{x}_i = u(\vec{x}_g) - \bar{u}_i \quad (39)$$

$$\text{Neumann:} \quad \vec{\nabla} u \cdot \Delta \vec{x}_n = \vec{\nabla} u(\vec{x}_g) \cdot \Delta \vec{x}_n \quad (40)$$

where $\Delta\vec{x}_i = \vec{x}_g - \vec{x}_{c,i}$ and $\Delta\vec{x}_n = (\Delta\vec{x}_i \cdot \hat{n}_g)\hat{n}_g$. More complicated boundary conditions were treated using ghost cells to influence the reconstruction. For example, reflecting boundaries or solid walls were treated by reflecting the solution at the control volume's centroid about the boundary and injecting the reflected solution into the ghost cell.

Limiting was performed using the multi-dimensional limiting process (MLP) developed by Park et al. [51] in conjunction with the slope limiter function of Venkatakrishnan [52]. Although MLP was developed specifically for cell-based finite-volume schemes on structured and unstructured mesh, it is easily extended to vertex-based formulations. The general form of the MLP condition states that monotonicity is preserved if the following condition is true for every vertex v_j of a control-volume Ω_i :

$$\bar{u}_{i,\text{neigh}}^{\min} \leq u_{v_j} \leq \bar{u}_{i,\text{neigh}}^{\max} \quad \forall v_j \in \Omega_i \quad (41)$$

where u_{v_j} is the interpolated value at the vertex v_j , and $\bar{u}_{i,\text{neigh}}^{\min}$ and $\bar{u}_{i,\text{neigh}}^{\max}$ are the minimum and maximum control-volume-averaged values among the control-volumes that share a vertex with Ω_i , respectively. Essentially, the interpolated values at the vertices of the control volume must be bounded by the maximum and minimum \bar{u} of the surrounding control volumes. These vertices, v_j , are the vertices of the dual mesh, $\mathcal{D}(\Omega)$, not the primal mesh, $\mathcal{T}(\Omega)$.

The final MLP slope limiter for the i th control volume is expressed as

$$\phi_i = \min_{\forall v_j \in \Omega_i} \begin{cases} \Phi \left(\frac{\bar{u}_{i,\text{neigh}}^{\max} - \bar{u}_i}{u_{v_j} - \bar{u}_i} \right) & \text{if } u_{v_j} - \bar{u}_i > a, \\ \Phi \left(\frac{\bar{u}_{i,\text{neigh}}^{\min} - \bar{u}_i}{u_{v_j} - \bar{u}_i} \right) & \text{if } u_{v_j} - \bar{u}_i < -a, \\ 1 & \text{otherwise} \end{cases} \quad (42)$$

where Φ is the Venkatakrishnan limiter function and $a = 10^{-7}$ is tolerance to avoid limiter chatter caused by numerical noise.

3.2 Numerical Flux and Sources

An upwind Godunov-type scheme was used to integrate the inviscid numerical flux, $\vec{\mathbf{F}}$, over the control-volume [53]. Given the left and right solution states, \mathbf{W}_L and \mathbf{W}_R , the numerical flux at the interface between two control-volumes is defined as

$$\vec{\mathbf{F}} \cdot \hat{n} = \mathcal{F}(\mathbf{W}_L, \mathbf{W}_R, \hat{n}) \quad (43)$$

where \mathcal{F} is a flux function which solves a Riemann problem in a direction aligned along the face normal, \hat{n} . Both the Rusanov [54, 55] and HLL [56] approximate Riemann solvers were implemented for the numerical flux, \mathcal{F} . The HLL flux function was found to be slightly more accurate while the Rusanov flux provided additional stability.

The left and right solution states at the interface were determined using the k -exact reconstruction procedure described in Section 3.1. As a result, the leading truncation error due to the inviscid operator is $\mathcal{O}(h^{k+1})$ in smooth regions. When the solution is under-resolved and deemed not smooth, the limited piecewise linear reconstruction was used and the truncation error of the inviscid operator is between $\mathcal{O}(h^2)$ (unlimited) and $\mathcal{O}(h)$ (limited).

The limited linear reconstruction of Section 3.1.4 is only applied in non-smooth regions to the inviscid terms in Eq. (8). The source terms are still evaluated using the higher-order representation since they don't generally generate instabilities. Thus, the truncation error of the source term operator is $\mathcal{O}(h^{k+1})$.

3.3 Transient Continuation and Steady-State Relaxation

Equation (8) defines a finite set of ODEs. For unsteady problems, the temporal derivative was discretized using the classic fourth-order Runge-Kutta (RK4) scheme [57]. This four-stage RK scheme is written as

$$\mathbf{U}^1 = \mathbf{U}^n + \frac{\Delta t}{2} \mathbf{R}^n \quad (44a)$$

$$\mathbf{U}^2 = \mathbf{U}^n + \frac{\Delta t}{2} \mathbf{R}^1 \quad (44b)$$

$$\mathbf{U}^3 = \mathbf{U}^n + \Delta t \mathbf{R}^2 \quad (44c)$$

$$\mathbf{U}^{n+1} = \mathbf{U}^n + \frac{\Delta t}{6} (\mathbf{R}^n + 2\mathbf{R}^1 + 2\mathbf{R}^2 + \mathbf{R}^3) \quad (44d)$$

where the superscript n denotes the time level.

Steady-state problems were relaxed using the two-stage optimally smoothing scheme of Van Leer et al. [58].

$$\mathbf{U}^{n+1} = \mathbf{U}^n + \sum_{\alpha=1}^{n_s} \beta^\alpha \Delta t \mathbf{R}^\alpha \quad (45)$$

where $n_s = 2$ is the number of stages, the superscript α denotes the intermediate stage, and β^k are the stage coefficients.

In both cases, steady and unsteady, the time step was determined by considering the inviscid Courant-Friedrichs-Lewy (CFL) stability criteria. The maximum permissible time step for each control volume is given by

$$\Delta t_i = \text{CFL} \cdot \left(\frac{\Delta_i}{\|\vec{v}_i\| + a_i} \right), \quad i = 1, 2, \dots, N_v \quad (46)$$

where $\Delta_i = \sqrt[3]{V_i}$ and CFL is a constant greater than zero. Time-accurate problems use a global time step given by

$$\Delta t = \min_{\forall i} \Delta t_i \quad (47)$$

4 RESULTS FOR THREE-DIMENSIONAL UNSTRUCTURED MESH

The proposed finite-volume scheme was assessed in terms of accuracy, stability, and computational efficiency. Numerical results for smooth and discontinuous function reconstructions, as well as steady and unsteady idealized flows, were obtained on three-dimensional unstructured tetrahedral mesh. All computations were performed on an HP DL980 G7 compute node with eight Intel Xeon X6550 (2.00GHz) processors and 128 GB of random-access memory (RAM).

Depending on the problem, accuracy was assessed based on the L_1 , L_2 , and/or L_∞ norms of the error between the exact solution and the numerical solution. The L_p norm of the error evaluated over the entire computational domain is given by

$$L_p = \|\text{Error}\|_p = \left[\frac{1}{\mathcal{V}_T} \sum_i^{N_v} \int_{\Omega_i} |u_i^k(\vec{x}) - u_{\text{exact}}(\vec{x})|^p \, d\Omega \right]^{1/p} \quad (48)$$

where \mathcal{V}_T is the total volume of the domain and $u_{\text{exact}}(\vec{x})$ is the exact solution. This integration is performed using the adaptive cubature algorithm developed by Berntsen et al. [59] for integrating functions over a collection of three-dimensional simplices.

4.1 Spherical Cosine Function

The first case considered was the reconstruction of a smooth spherical cosine function. The function, which is smooth in all directions, is illustrated in Fig. 3a and described by

$$u(r) = 1 + \frac{1}{3} \cos(r) \quad (49)$$

where $r = 10\sqrt{x^2 + y^2 + z^2}$ is the radial position. The solution was computed on a unit cube centered at (0.5, 0.5, 0.5) using grids composed of tetrahedral cells with varying levels of resolution. A sample mesh is illustrated in Fig. 3b.

Unlimited k -exact reconstructions of the spherical cosine function that were obtained on a coarse mesh (995 vertices and 4,515 tetrahedral elements) are illustrated in Fig. 3c. As the order of the piecewise polynomial interpolant was increased from $k=0$ to $k=3$, the reconstructed solution rapidly approached the exact solution. There is almost no visible difference between the exact solution and the reconstructed solution for $k=4$ (not shown in figure).

The behavior of the discretization error as the mesh resolution was increased, illustrated in Fig. 3d for various values of k , confirms that k -exact reconstruction of a smooth function yields an order of accuracy equal to $k+1$. A convergence rate of approximately $k+1$ was observed in all of the error norms, including the L_∞ error norm.

4.2 Abgrall's Function

The Abgrall function [60] possesses a number of solution discontinuities that test a high-order spatial discretization's ability to maintain monotonicity. As such, reconstruc-

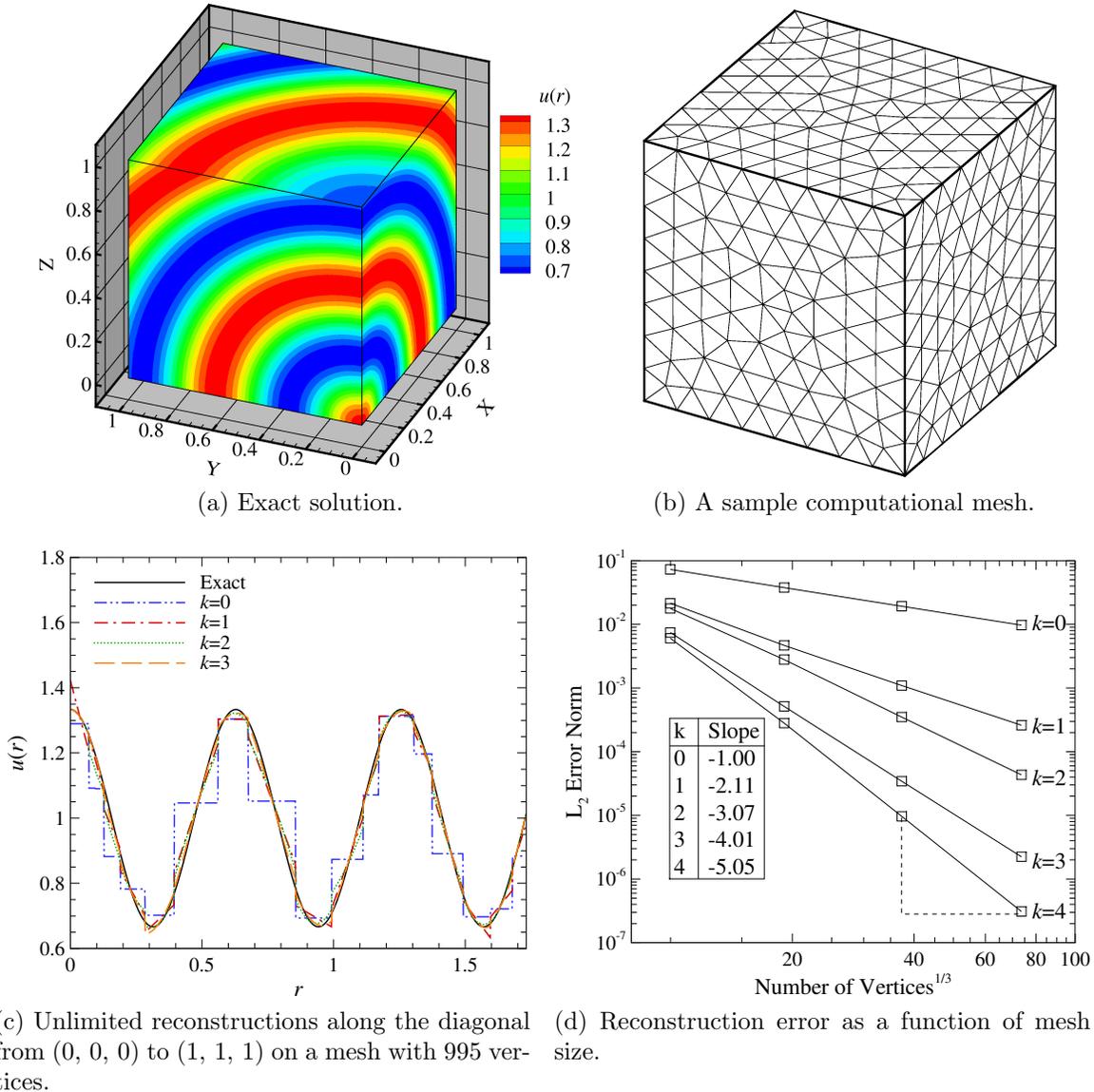


Figure 3: Results for k -exact reconstruction of the spherical cosine function.

tions of this function were performed using the proposed CENO algorithm to ensure the effectiveness of the smoothness indicator defined in Eq. (32). While the performance of the smoothness indicator was already verified using the Abgrall function on both structured [36] and unstructured [32–35] mesh, it has not been verified for vertex-based approach. All of the previous CENO approaches applied cell-based finite-volume formulations only.

Since the Abgrall function was originally designed to vary in two space dimensions only, it was modified to include a variation and discontinuity along the third dimension.

The resulting three-dimensional, discontinuous function is given by

$$u(x, y, z) = g(z) \cdot \begin{cases} f \left[x - \cot \left(\sqrt{\pi/2} y \right) \right] & x \leq \cos(\pi y)/2 \\ f \left[x + \cot \left(\sqrt{\pi/2} y \right) \right] + \cos(2\pi y) & x > \cos(\pi y)/2 \end{cases} \quad (50)$$

where

$$f(r) = \begin{cases} -r \cdot \sin(3\pi r^2/2) & r \leq -1/3 \\ |\sin(2\pi r)| & |r| < 1/3 \\ 2r - 1 + \sin(3\pi r)/6 & r \geq 1/3 \end{cases} \quad (51)$$

and

$$g(z) = \begin{cases} \sin(z\pi/2)/2 + 1 & z < -1/2 \\ -z/2 + 1 & z \geq -1/2 \end{cases} \quad (52)$$

Eq. (50) was discretized on a cube with length 2 and centered about the origin, using computational meshes similar to those used for the spherical cosine function (Fig. 3b). A reconstructed solution that was obtained with $k = 4$ is compared with the exact solution along $z = 0$ in Figs. 4a and 4b. This reconstruction was performed on a mesh with approximately 3 million vertices and 18 million tetrahedral elements, and, as observed in Fig. 4b, it was able to accurately represent the Abgrall function without producing spurious oscillations. This is because the smoothness indicator, illustrated in Fig. 4c, correctly identified the discontinuities in both u and $\partial u/\partial x_i$.

The solution obtained with $k = 0$ to 4 on a mesh with 18 million tetrahedral elements is compared with the original function along a line in Fig. 4d. The proposed CENO scheme was able to ensure oscillation-free solutions despite the large discontinuities observed. This confirms the effectiveness of the smoothness indicator.

The effect of mesh resolution on the L_1 norm of the solution error is illustrated in Fig. 5. A large improvement in the error was achieved by increasing k from 0 to 1. This improvement became less pronounced as k was increased further to 2 since a large portion of the domain possessed discontinuous features. In fact, because of the large number of discontinuities, there was only a slight improvement in the solution error as k was increased beyond 2. This indicates that the solution error has not yet reached the asymptotic regime for this case.

The convergence rate of the error norms is also provided in Fig. 5. An order of accuracy of 1 was observed for all values of k , which was expected after applying a limited piecewise linear reconstruction near discontinuous. Nonetheless, the main highlight is that the hybrid reconstruction procedure was able to produce non-oscillatory solutions despite the presence of discontinuities, using only a single, central stencil.

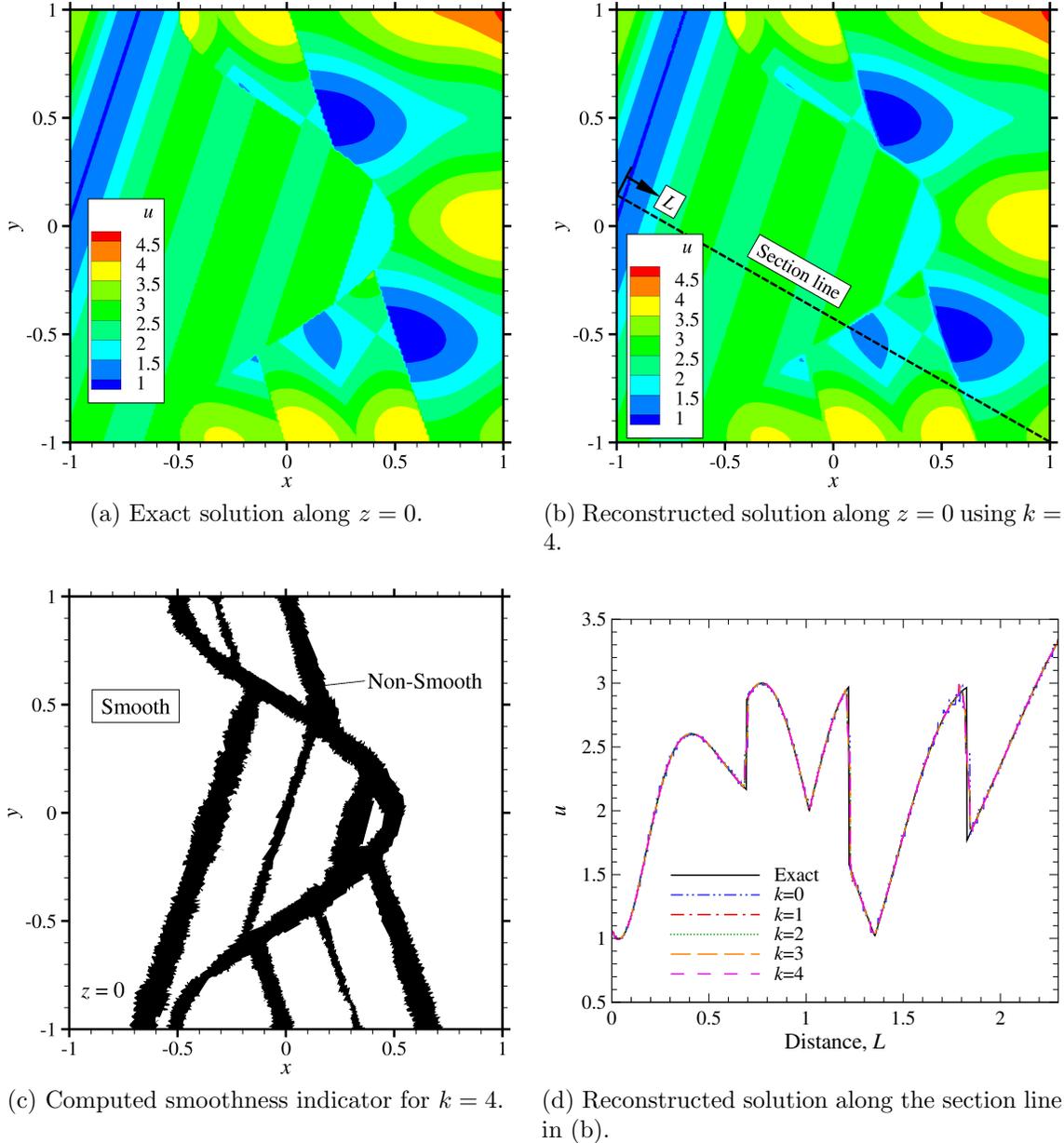


Figure 4: Results for CENO reconstruction of the modified Abgrall function. Numerical results shown here were obtained using a mesh with 3 million vertices and 18 million tetrahedral elements.

4.3 Smooth Supersonic Flow

The spatial accuracy of the proposed finite-volume formulation was verified for smooth flows using the method of manufactured solutions (MMS) [61–63]. In MMS, analytical source terms are derived which, when added to the governing equations, produce a desired

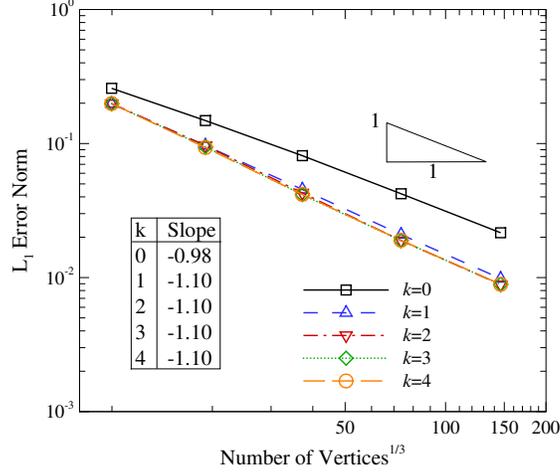


Figure 5: Effect of grid resolution on solution accuracy for CENO reconstruction of the modified Abgrall function. Accuracy is measured using the L_1 norm of the error.

Table 3: The coefficients for the three-dimensional manufactured solution to Eq. (53).

ϕ	ϕ_0	ϕ_x	ϕ_y	ϕ_z	a_{ϕ_x}	a_{ϕ_y}	a_{ϕ_z}
ρ	1	0.15	-0.1	-0.12	1	0.5	1.5
u	800	50	-30	-18	1.5	0.6	0.5
v	800	-75	40	-30	0.5	2/3	1.25
w	800	15	-25	35	1/3	1.5	1
p	1×10^5	0.2×10^5	0.5×10^5	-0.35×10^5	2	1	1/3

solution. The particular MMS approach outlined by Roy et al. [64, 65] was applied to produce sinusoidal solutions of the following form:

$$\rho = \rho_0 + \rho_x \sin(a_{\rho_x} \pi x/L) + \rho_y \cos(a_{\rho_y} \pi y/L) + \rho_z \sin(a_{\rho_z} \pi z/L) \quad (53a)$$

$$u = u_0 + u_x \sin(a_{u_x} \pi x/L) + u_y \cos(a_{u_y} \pi y/L) + u_z \cos(a_{u_z} \pi z/L) \quad (53b)$$

$$v = v_0 + v_x \cos(a_{v_x} \pi x/L) + v_y \sin(a_{v_y} \pi y/L) + v_z \sin(a_{v_z} \pi z/L) \quad (53c)$$

$$w = w_0 + w_x \sin(a_{w_x} \pi x/L) + w_y \sin(a_{w_y} \pi y/L) + w_z \cos(a_{w_z} \pi z/L) \quad (53d)$$

$$p = p_0 + p_x \cos(a_{p_x} \pi x/L) + p_y \sin(a_{p_y} \pi y/L) + p_z \cos(a_{p_z} \pi z/L) \quad (53e)$$

where ϕ_0 , ϕ_x , ϕ_y , ϕ_z , a_{ϕ_x} , a_{ϕ_y} , and a_{ϕ_z} are constants for the variable $\phi \in \{\rho, u, v, w, p\}$. The coefficients were chosen based on those given by Roy et al. [64] for smooth, three-dimensional, supersonic, inviscid flow. They are provided in Table 3 for completeness. The length scale, L , was chosen as unity.

This particular manufactured solution describes a steady supersonic flow at an angle of approximately 45 degrees to the coordinate axes, and with a Mach number varying between 3 and 6. The flow was modeled on a unit cube over the range $0 \leq x, y, z \leq 1$;

although, any domain could be used since the solutions exist for all x , y and z . An example of this smoothly varying solution is illustrated in Fig. 6a, which depicts the internal energy distribution.

The spatial accuracy was assessed by performing calculations with different mesh sizes and measuring the changes in solution error. Solutions were obtained on meshes of varying resolution, similar to those in Fig. 3b, with supersonic inflow and outflow boundary conditions at the corresponding upstream and downstream boundaries of the domain. All solutions were relaxed to a steady-state using the two-stage optimally smoothing scheme of Van Leer et al. [58] with a CFL = 0.5 and the HLL numerical flux. Any numerical flux can be used for this error analysis, because, according to Eq. (12), the dissipation vanishes with $\mathcal{O}(h^{k+1})$ for smooth solutions.

For each calculation, the analytical solution at every vertex of the primal mesh was prescribed and the governing equations were relaxed until all equation residuals were reduced by four orders of magnitude. Tighter tolerances were also tested, but no significant gain in accuracy was observed using them.

The L_2 norm of the error in the predicted internal energy, e , is illustrated in Fig. 6b. All of the primitive solution quantities displayed the same relationship between mesh size and total solution error, but e displayed the largest errors, and was therefore chosen for this analysis. The slopes of the lines in Fig. 6b are provided in Fig. 6c, along with those observed for the other norms, i.e., the L_1 and L_∞ norms. For all values of interest for k , the formal order of accuracy was achieved by the L_2 norm. The other norms also displayed similar convergence characteristics, although some degradation of the slopes of the L_∞ norms were observed as the mesh spacing decreased. This is largely attributed to the finite precision of the adaptive cubature algorithm used to evaluate the numerical errors.

The solution error is plotted as a function of the wall-clock time in Fig. 6d. The high-order schemes become more efficient in terms of accuracy vs computational cost as the target accuracy gets smaller. That is, there is a particular range of accuracy for which a particular value of k is the most efficient, and this optimal value of k increases with the desired level of accuracy. For all the meshes considered, the first-order ($k = 0$) scheme was the least efficient, while the second-order ($k = 1$) scheme was the most efficient for a target error above about 10^3 . Just below this level of error, the fourth-order ($k = 3$) scheme was the most efficient. The fifth-order ($k = 4$) scheme was the most efficient for errors below 10. For this particular problem, there was no range over which the third-order ($k = 2$) scheme was optimal. These results confirm that, for smooth problems, higher-order schemes are more efficient for higher levels of desired accuracy.

4.4 Shock Tube

The robustness and accuracy of the algorithm was demonstrated for non-smooth problems with a one-dimensional shock-tube [66]. This time-dependent problem was solved

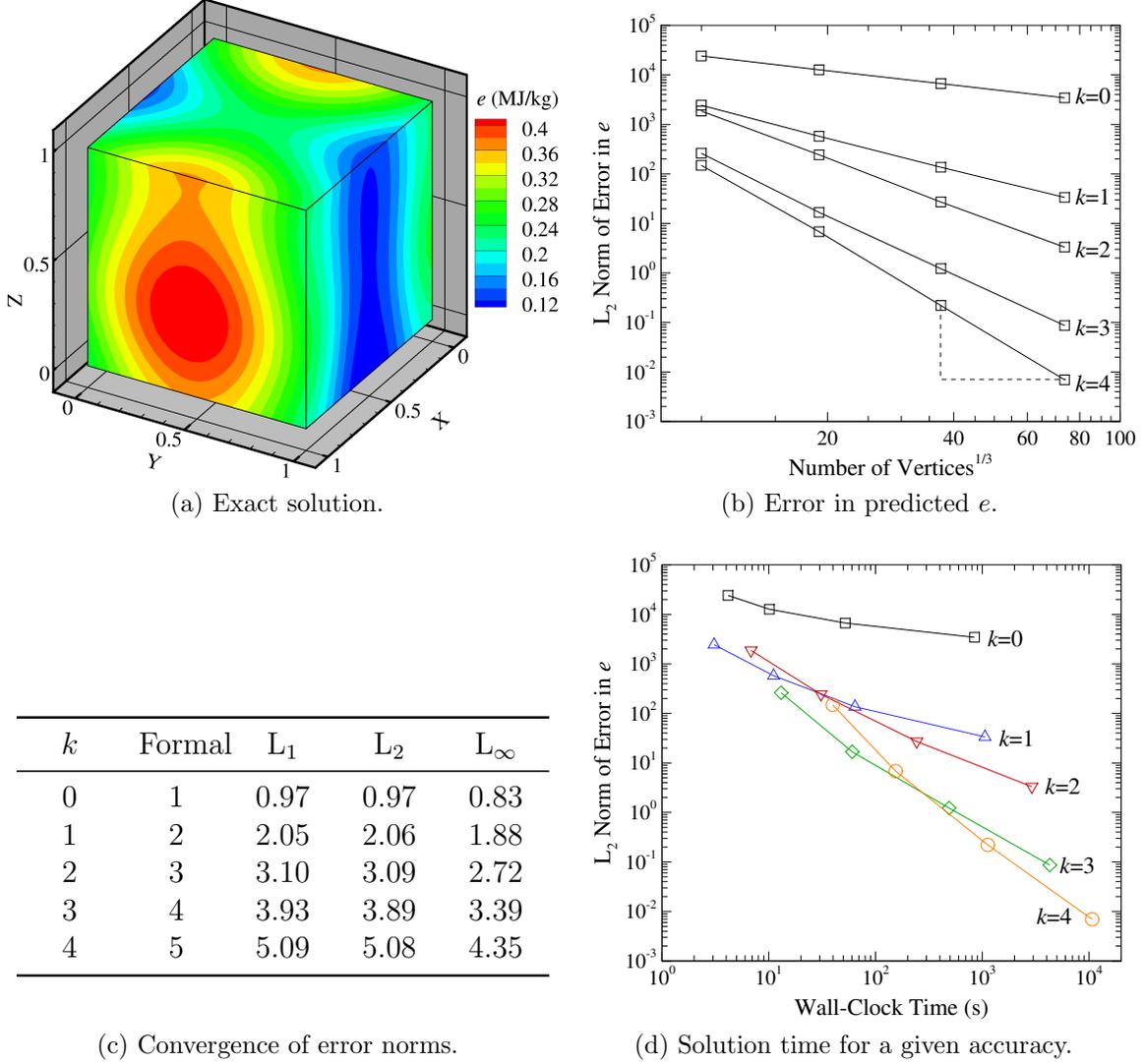


Figure 6: Results for smooth supersonic flow.

on a rectangular domain of length 1, with the following initial conditions:

$$\mathbf{W}(x, 0) = \begin{cases} \mathbf{W}_L & \text{if } x \leq 0.45, \\ \mathbf{W}_R & \text{if } x > 0.45 \end{cases} \quad (54)$$

where $\mathbf{W}_L = [1, 0, 0, 0, 1]$ and $\mathbf{W}_R = [0.1, 0, 0, 0, 0.125]$. A sample of the meshes used is illustrated in Fig. 7 along with the dimensions of the computational domain.

Solutions were obtained for different values of k , using meshes of increasing resolution. All solutions were integrated in time until $t = 0.2$ s with the RK4 time-marching scheme, a CFL of 0.2, and the HLL numerical flux. Reflection boundary conditions were applied to the surrounding surfaces while the solution was free to vary at both ends of the tube.

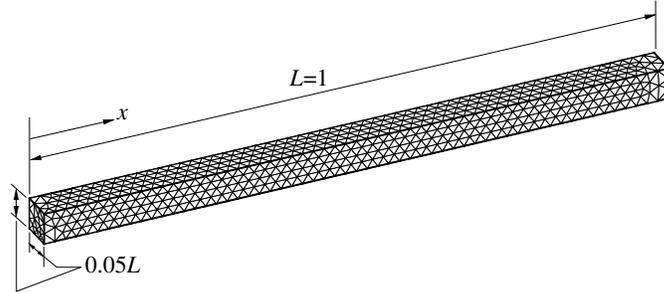


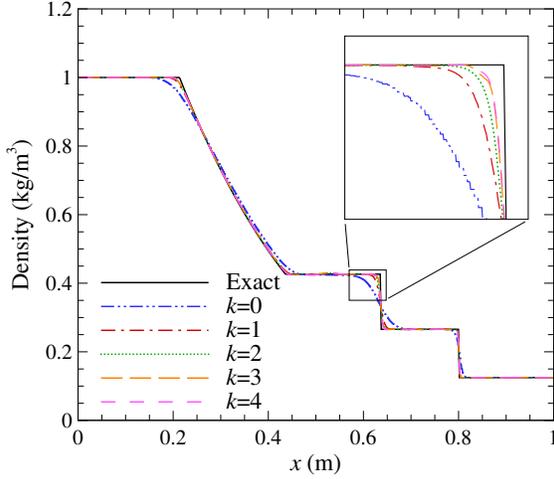
Figure 7: Sample computational mesh and domain used for shock tube problem.

Solutions on the finest mesh considered are compared with the exact solution at $t = 0.2$ s in Fig. 8a. Overall, there was a distinct improvement in the numerical solution as the polynomial degree was increased, even near discontinuities. This is highlighted for the contact surface in the inset of Fig. 8a. There was an initial large improvement in the solution as k was increased from 0 to 1. Further increases in k provided smaller and smaller improvements in the solution.

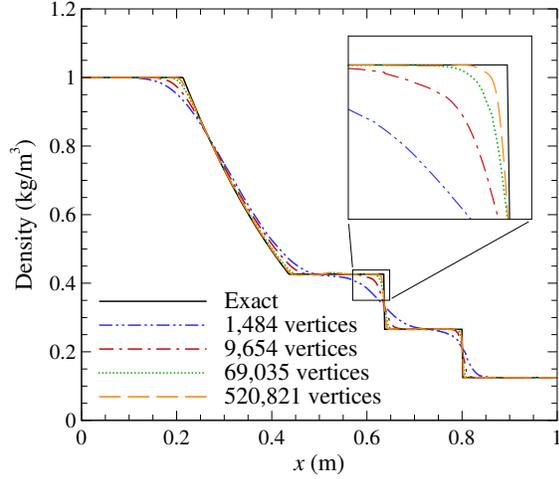
A similar comparison is made in Fig. 8b, which illustrates the effect of mesh resolution on the fifth-order ($k = 4$) scheme. As expected, increasing the mesh resolution improves the agreement of the numerical solution with the exact solution. This occurs because there is less dissipation introduced by the time marching scheme, i.e., smaller time steps, and the spatial discretization, i.e., smaller grid spacing.

The behavior of the L_1 error norms with mesh size is demonstrated for ρ in Fig. 8c. The first-order ($k = 0$) scheme does not quite reach the asymptotic region. It only achieves an order of accuracy of approximately 0.7. The $k = 1$ scheme achieved a significant reduction in error over the $k = 0$ scheme, but only converges at a rate of $\mathcal{O}(h)$ due to the discontinuities present in the solution. Using the coarsest mesh, all higher-order ($k > 1$) schemes had the same error as the linear representation. This is because the smoothness indicator detected under-resolved data throughout most of the domain and the linear reconstruction was used everywhere. However, as the mesh resolution was increased, the higher-order schemes had lower errors. All higher-order schemes only achieved first-order accuracy, which was expected because of the discontinuities in the solution, but there was still a decrease in overall error as k was increased beyond 1. Even though CENO drops to first-order near discontinuities, the size of the region influenced by the discontinuity decreases with mesh size. As such, there is a net reduction in error.

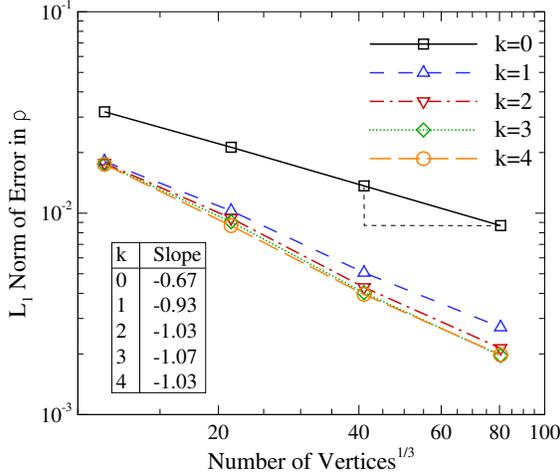
The computational efficiency was assessed in terms of the wall-clock time to a given level of error. Over the range of meshes studied, the $k = 1$ scheme was the most efficient. However, extrapolating to lower error levels, the $k = 2$ scheme is expected to be more efficient for error levels below 0.02. The efficiency of the other high-order schemes, i.e., $k = 3$ and $k = 4$, is expected to improve as the desired error is lowered further.



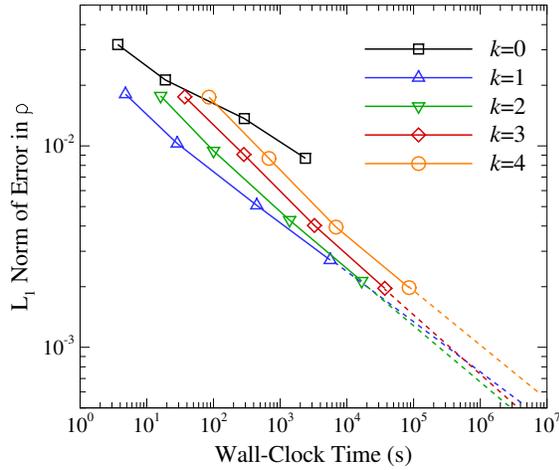
(a) Effect of polynomial degree on predictions using a mesh with 520,821 vertices.



(b) Effect of mesh size on predictions for $k = 4$.



(c) Convergence of the error norms.



(d) Accuracy as a function of solution time. Dashed lines represent extrapolations.

Figure 8: Results for one-dimensional shock-tube at $t=2$ s.

4.5 Sedov Blast Wave

The Sedov explosion problem [67] involves the evolution of a spherical blast wave from an initial pressure perturbation in an otherwise homogeneous medium. The blast wave was generated by an initial energy source, e_{blast} , located in a small region of radius r_0 near

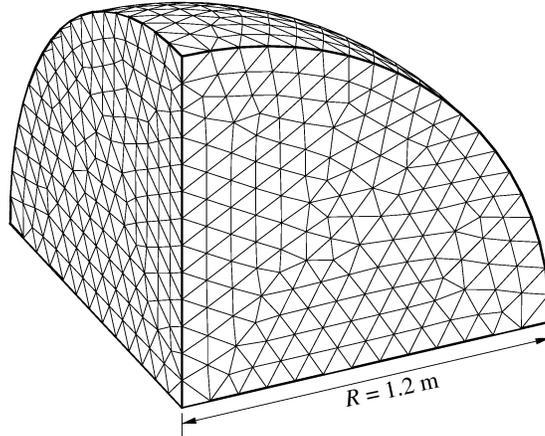


Figure 9: Sample computational mesh used for Sedov problem.

the origin. The initial conditions at time $t = 0$ are

$$\rho(r, 0) = 1, \quad \vec{v}(r, 0) = \vec{0}, \quad p(r, 0) = \begin{cases} \frac{3\rho(\gamma - 1)e_{\text{blast}}}{4\pi r_0^3} & \text{if } r \leq r_0 \\ 10^{-5} & \text{if } r > r_0 \end{cases}$$

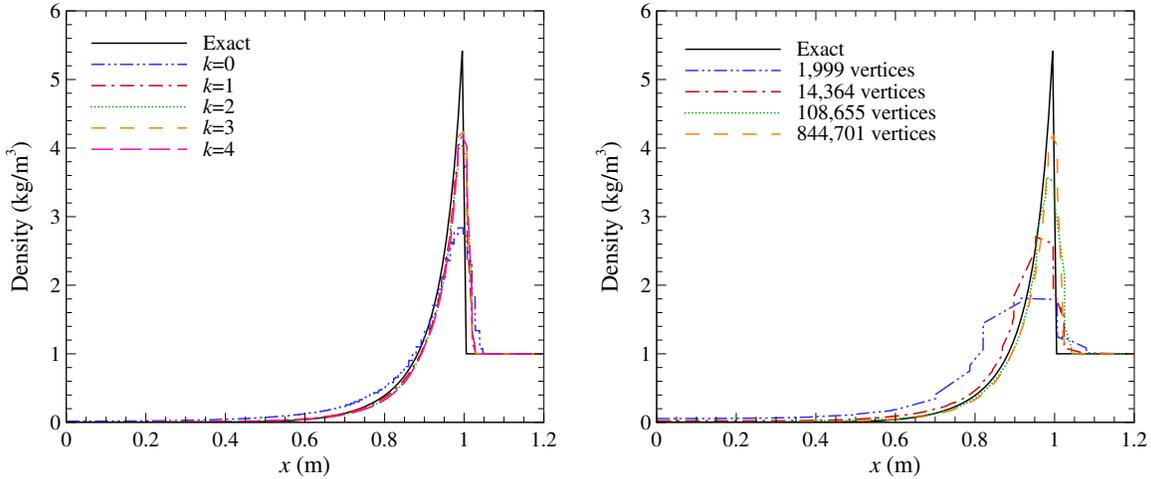
where $r = \sqrt{x^2 + y^2 + z^2}$, $e_{\text{blast}} = 0.851072$. This configuration gives a blast wave that reaches $r = 1$ at $t = 1$ s.

In practice, it is difficult to define a small radius r_0 without an overly fine mesh near the origin, especially when using tetrahedral mesh. So the energy e_{blast} was deposited into the control volume at the origin only. Simulations were obtained with $k = 0, 1, \dots, 4$ on four different, successively-refined meshes. All solutions were integrated in time until $t = 1$ s with the RK4 time-marching scheme, a CFL of 0.1, and the Rusanov numerical flux. Only an octant of a sphere was modeled, with reflecting boundary conditions to enforce symmetry. The outer surface of the sphere, located at $R = 1.2$ m, was also treated as a reflecting wall. A sample computational mesh is illustrated in Fig. 9.

Predictions for density are compared with the analytical solution in Fig. 10. The exact solution for this spherical blast wave was obtained using the numerical algorithm outlined by Kamm [68]. For all values of k and meshes employed, no oscillations were observed in front of or behind the diverging shock wave. For the high-order solutions, i.e., $k > 1$, the smoothness indicator correctly identified the large solution discontinuity at the moving shock front

The effect of polynomial order on the predicted density is illustrated in Fig. 10a for the finest mesh investigated (844,701 vertices and 4,922,880 tetrahedra). As also observed for the shock tube test problem in Section 4.4, increasing the order of the polynomial provided a significant improvement in the predicted solution.

Figure 10b illustrates the effect of mesh resolution on the predicted density, which



(a) Effect of polynomial degree on predictions using a mesh with 844,701 vertices.

(b) Effect of mesh size on predictions for $k = 4$.

Figure 10: Results for the Sedov problem at $t = 1$ s.

was obtained using the 5th-order ($k = 4$) CENO reconstruction. At low resolutions, the smoothness indicator flagged a large portion of the domain as under-resolved. However, as the mesh resolution was increased, the size of the region treated using the lower-order limited piecewise linear reconstruction diminished.

4.6 Triple-Point Shock Interaction

As a final test of the algorithm’s robustness, a three-dimensional, three-state Riemann problem was studied. The initial conditions, which are illustrated in Fig. 11 along with the geometry, generate a shock that propagates parallel to a contact discontinuity, which in turn generates a high-speed vortex. This problem does not have an exact solution, but it is studied here because of the difficulty in resolving the interaction between the shocks and contact discontinuities without generating spurious oscillations, especially for high-order solution methods.

Two simulations were obtained on mesh with approximately 347,811 vertices and 1,960,914 tetrahedra, one with a low-order representation ($k = 1$) and one with a high-order ($k = 4$) polynomial representation. Both solutions were integrated in time until $t = 5$ s with the RK4 time-marching scheme, a CFL of 0.25, and the HLL numerical flux. Reflection and solid wall boundary conditions were applied as indicated in Fig. 11.

Numerical predictions for density at $t = 5$ s are compared in Fig. 12. No significant differences between the two solutions were visible, because the high-order CENO scheme deemed a large portion of the flow non-smooth and it was treated using the limited linear reconstruction instead. These results highlight the robustness of the high-order CENO algorithm, since the $k = 4$ scheme was able to reliably obtain a solution without producing

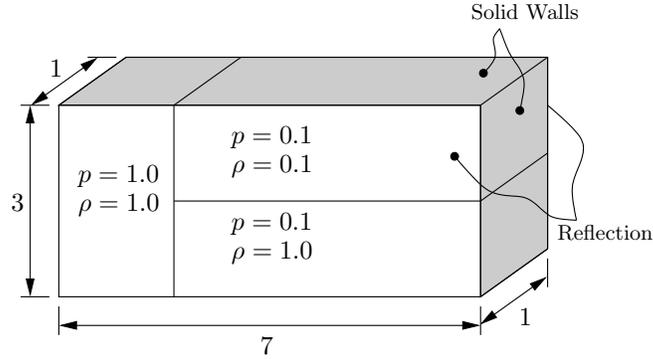


Figure 11: Computational domain and initial conditions for the triple-point problem.

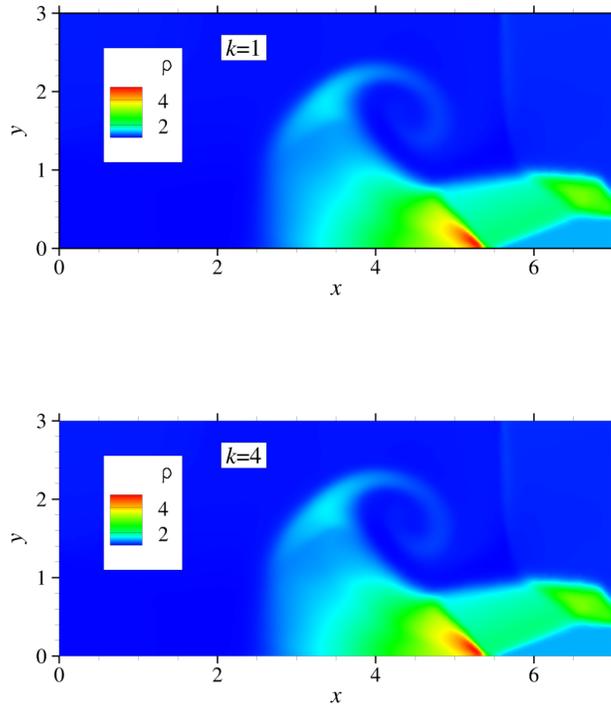


Figure 12: Predicted density field at $t = 5$ s for triple-point problem.

any unphysical oscillations.

5 CONCLUSIONS

A high-order finite-volume scheme was developed for the mathematical description of compressible fluids on unstructured meshes. It is a vertex-based variant of the cell-based,

Godunov-type, finite-volume methods developed by Ivan et al. [36–40] and Charest et al. [32–35], which use a hybrid CENO reconstruction procedure to avoid spurious oscillations. The scheme was assessed in terms of accuracy and computational cost for a variety of problems, including smooth and discontinuous function reconstructions, and solutions to idealized flow problems.

Up to fifth-order accuracy was demonstrated. For smooth flows and function reconstructions, $(k + 1)$ th order of accuracy was achieved using piecewise polynomial representations of degree k . Only first-order accuracy was observed for all problems that contained discontinuities, but there was still a measured advantage provided by the high-order schemes. They displayed lower errors for a given mesh.

In terms of computational efficiency, i.e., wall time for a given accuracy, there was an optimal value of k which varied depending upon the desired error, and the particular problem. The standard second-order scheme was the most efficient for higher error levels, and the high-order schemes became more efficient as the desired error was decreased. This was demonstrated for smooth and discontinuous problems, although the mesh sizes at which the high-order schemes were more efficient was significantly larger for discontinuous problems.

Overall, this research highlights the main advantages of the CENO finite-volume algorithm. High-order accuracy was achieved in smooth regions, while robust and monotone solutions were maintained near discontinuities and under-resolved solution content. Future work consists of further development and validation of the proposed algorithm, including its extension to multi-material problems, arbitrary equations of state, moving meshes, and adaptive mesh refinement.

ACKNOWLEDGMENTS

This research was supported by the United States Department of Energy, through the Advanced Simulation & Computing (ASC) and Metropolis postdoctoral fellowship programs.

REFERENCES

- [1] D.J. Mavriplis. AIAA J., 46(6):1281–1298, 2008.
- [2] S. Pirozzoli. J. Comput. Phys., 219(2):489–497, 2006.
- [3] A. Harten, B. Engquist, S. Osher, and S.R. Chakravarthy. J. Comput. Phys., 71(2): 231–303, 1987.
- [4] T.J. Barth. AIAA Paper 93-0668, 1993.
- [5] R. Abgrall. J. Comput. Phys., 114:45–58, 1994.
- [6] T. Sonar. Comp. Meth. Appl. Mech. Eng., pp. 140–157, 1997.

- [7] C.F. Ollivier-Gooch. *J. Comput. Phys.*, 133:6–17, 1997.
- [8] G.S. Jiang and C.W. Shu. *J. Comput. Phys.*, 126(1):202–228, 1996.
- [9] D. Stanescu and W. Habashi. *AIAA J.*, 36:1413–1416, 1998.
- [10] O. Friedrich. *J. Comput. Phys.*, 144(1):194–212, 1998.
- [11] C. Hu and C.W. Shu. *J. Comput. Phys.*, 150:97–127, 1999.
- [12] C.F. Ollivier-Gooch and M. Van Altena. *J. Comput. Phys.*, 181(2):729–752, 2002.
- [13] A. Nejat and C. Ollivier-Gooch. *J. Comput. Phys.*, 227(4):2582–2609, 2008.
- [14] B. Cockburn and C.W. Shu. *Math. Comp.*, 52:411, 1989.
- [15] B. Cockburn, S. Hou, and C.W. Shu. *J. Comput. Phys.*, 54:545, 1990.
- [16] R. Hartmann and P. Houston. *J. Comput. Phys.*, 183:508–532, 2002.
- [17] H. Luo, J.D. Baum, and R. Löhner. *J. Comput. Phys.*, 225:686–713, 2007.
- [18] G. Gassner, F. Lörcher, and C.D. Munz. *J. Comput. Phys.*, 224(2):1049–1063, 2007.
- [19] F. Bassi and S. Rebay. *J. Comput. Phys.*, 131:267–279, 1997.
- [20] B. Cockburn and C.W. Shu. *SIAM J. Numer. Anal.*, 35(6):2440–2463, 1998.
- [21] B. Leervan , M. Lo, and M. Raaltevan . *AIAA Paper 2007-4083*, 2007.
- [22] M. Raaltevan and B. Leervan . *Commun. Comput. Phys.*, 5(2–4):683–693, 2009.
- [23] H. Liu and J. Yan. *SIAM J. Numer. Anal.*, 41(1):675–698, 2009.
- [24] Z.J. Wang. *J. Comput. Phys.*, 178:210–251, 2002.
- [25] Z.J. Wang and Y. Liu. *J. Comput. Phys.*, 179:665–697, 2002.
- [26] Z.J. Wang, L. Zhang, and Y. Liu. High-order spectral volume method for 2d euler equations. Paper 2003–3534, *AIAA*, June 2003.
- [27] Z.J. Wang and Y. Liu. *Journal of Scientific Computing*, 20(1):137–157, 2004.
- [28] Y. Sun, Z.J. Wang, and Y. Liu. *J. Comput. Phys.*, 215(1):41–58, 2006.
- [29] H. Huynh. *AIAA Paper 2007-4079*, 2007.
- [30] Z.J. Wang and H. Gao. *AIAA Paper 2009–401*, 2009.

- [31] Z.J. Wang and H. Gao. *J. Comput. Phys.*, 228:8161–8186, 2009.
- [32] S.D. McDonald, M.R.J. Charest, and C.P.T. Groth. High-order CENO finite-volume schemes for multi-block unstructured mesh. 20th AIAA Computational Fluid Dynamics Conference, Honolulu, Hawaii, June 27–30 2011. doi: 10.2514/6.2011-3854. AIAA-2011-3854.
- [33] M.R.J. Charest, C.P.T. Groth, and P.Q. Gauthier. High-order CENO finite-volume scheme for low-speed viscous flows on three-dimensional unstructured mesh. ICCFD7 - International Conference on Computational Fluid Dynamics, Hawaii, July 9–13 2012. Paper ICCFD7-1002.
- [34] M.R.J. Charest and C.P.T. Groth. A high-order central ENO finite-volume scheme for three-dimensional turbulent reactive flows on unstructured mesh. 21st AIAA Computational Fluid Dynamics Conference, San Diego, California, June 24–27 2013. doi: 10.2514/6.2013-2567. AIAA 2013-2567.
- [35] M.R.J. Charest, C.P.T. Groth, and P.Q. Gauthier. *Commun. Comput. Phys.*, 2013. Submitted for publication.
- [36] L. Ivan and C.P.T. Groth. AIAA paper 2007-4323, 2007.
- [37] L. Ivan and C.P.T. Groth. AIAA paper 2011-0367, 2011.
- [38] L. Ivan and C.P.T. Groth. *Commun. Comput. Phys.*, 2013. submitted for publication.
- [39] L. Ivan and C.P.T. Groth. *J. Comput. Phys.*, 257:830–862, 2013.
- [40] A. Susanto, L. Ivan, H. De Sterck, and C.P.T. Groth. *J. Comput. Phys.*, 250(1):141–164, 2013.
- [41] A. Haselbacher. AIAA paper 2005-0879, 2005.
- [42] J. Waltz, N.R. Morgan, T.R. Canfield, M.R.J. Charest, L.D. Risinger, and J.G. Wohlbiel. *Comput. Fluids*, 92:172–187, 2013.
- [43] C.A. Felippa. *Eng. Computation.*, 21(8):867–890, 2004.
- [44] T.J. Barth and P.O. Fredrickson. AIAA Paper 90-0013, 1990.
- [45] D.J. Mavriplis. AIAA paper 2003-3986, 2003.
- [46] A. Jalali and C. Ollivier-Gooch. AIAA Paper 2013-2565, 2013.
- [47] C.L. Lawson and R.J. Hanson. *Solving least squares problems*. Prentice-Hall, 1974.
- [48] K. Michalak and C. Ollivier-Gooch. AIAA paper 2007-3943, 2007.

- [49] N.R. Draper and H. Smith. Applied regression analysis. Wiley, New York, 3rd edition, 1998.
- [50] T.J. Barth. AIAA Paper 91-1548, 1991.
- [51] J.S. Park, S.H. Yoon, and C. Kim. J. Comput. Phys., 229(3):788–812, 2010.
- [52] V. Venkatakrishnan. AIAA Paper 93-0880, 1993.
- [53] S.K. Godunov. Mat. Sb., 47:271–306, 1959.
- [54] V.V. Rusanov. J. Comput. Math. Phys. USSR, 1:267–279, 1961.
- [55] S.F. Davis. SIAM J. Sci. Stat. Comput., 9(3):445–473, 1988.
- [56] A. Harten, P.D. Lax, and B. Leervan . SIAM Rev., 25(1):35–61, 1983.
- [57] H. Lomax, T.H. Pulliam, and D.W. Zingg. Fundamentals of Computational Fluid Dynamics. Springer, New York, 2003.
- [58] B. Van Leer, C. Tai, and K.G. Powell. AIAA Paper 89-1933, 1989.
- [59] J. Berntsen, R. Cools, and T.O. Espelid. ACM Trans. Math. Softw., 19(3):320–332, 1993.
- [60] R. Abgrall. ICASE Contractor Report 189574, 1991.
- [61] P.J. Roache and S. Steinberg. AIAA J., 22(10):1390–1394, 1984.
- [62] W.L. Oberkampf and F.G. Blottner. AIAA J., 36(5):687–695, 1998.
- [63] P.J. Roache. J. Fluids Eng., 124(1):4–10, 2001.
- [64] C.J. Roy, T.M. Smith, and C.C. Ober. AIAA Paper 2002-3110, 2002.
- [65] C.J. Roy, C.C. Nelson, T.M. Smith, and C.C. Ober. Int. J. Numer. Meth. Fluids, 44(6):599–620, 2004.
- [66] G.A. Sod. J. Comput. Phys., 27(1):1 – 31, 1978.
- [67] L.I. Sedov. Similarity and Dimensional Methods in Mechanics. Academic Press, New York, 1959.
- [68] J. Kamm. Evaluation of the Sedov-von Neumann-Taylor blast wave solution. Technical Report LA-UR-00-6055, Los Alamos National Laboratory, 2000.