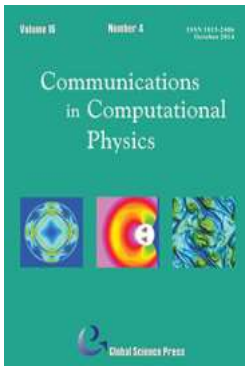



# A High-Order Central ENO Finite-Volume Scheme for Three-Dimensional Low-Speed Viscous Flows on Unstructured Mesh


Marc R. J. Charest, Clinton P. T. Groth and Pierre Q. Gauthier



Communications in Computational Physics / Volume 17 / Issue 03 / March 2015, pp 615 - 656
DOI: 10.4208/cicp.091013.281114a, Published online: 02 April 2015

Commun. Comput. Phys.
doi: 10.4208/cicp.091013.281114a

Vol. **17**, No. 3, pp. 615-656
March 2015

# A High-Order Central ENO Finite-Volume Scheme for Three-Dimensional Low-Speed Viscous Flows on Unstructured Mesh

Marc R. J. Charest[1,*], Clinton P. T. Groth[1] and Pierre Q. Gauthier[2]

[1] *University of Toronto Institute for Aerospace Studies, 4925 Dufferin Street, Toronto, Ontario, Canada M3H 5T6.*
[2] *Energy Engineering & Technology, Rolls-Royce Canada Limited, 9245 Côte-de-Liesse, Dorval, Québec, Canada H9P 1A5.*

Received 9 October 2013; Accepted (in revised version) 28 November 2014

Communicated by Chi-Wang Shu

**Abstract.** High-order discretization techniques offer the potential to significantly reduce the computational costs necessary to obtain accurate predictions when compared to lower-order methods. However, efficient and universally-applicable high-order discretizations remain somewhat illusive, especially for more arbitrary unstructured meshes and for incompressible/low-speed flows. A novel, high-order, central essentially non-oscillatory (CENO), cell-centered, finite-volume scheme is proposed for the solution of the conservation equations of viscous, incompressible flows on three-dimensional unstructured meshes. Similar to finite element methods, coordinate transformations are used to maintain the scheme's order of accuracy even when dealing with arbitrarily-shaped cells having non-planar faces. The proposed scheme is applied to the pseudo-compressibility formulation of the steady and unsteady Navier-Stokes equations and the resulting discretized equations are solved with a parallel implicit Newton-Krylov algorithm. For unsteady flows, a dual-time stepping approach is adopted and the resulting temporal derivatives are discretized using the family of high-order backward difference formulas (BDF). The proposed finite-volume scheme for fully unstructured mesh is demonstrated to provide both fast and accurate solutions for steady and unsteady viscous flows.

**AMS subject classifications**: 35Q30, 65Z05, 65M08, 65N08, 76M12, 76D05, 76G25

**Key words**: Numerical algorithms, computational fluid dynamics, high-order methods, incompressible flows.

*Corresponding author. *Email addresses:* charest@utias.utoronto.ca (M. R. J. Charest), groth@utias.utoronto.ca (C. P. T. Groth), pierre.gauthier@rolls-royce.com (P. Q. Gauthier)

# 1 Introduction

Computational fluid dynamics (CFD) has proven to be an important enabling technology in many areas of science and engineering. In spite of the relative maturity and widespread success of CFD in these areas, there is a variety of physically-complex flows which are still not well understood and are very challenging to predict with numerical methods. Such flows include, but are certainly not limited to, multiphase, turbulent, and combusting flows encountered in aerospace propulsion systems (e.g., gas turbine engines and solid propellant rocket motors). These flows present numerical challenges as they generally involve a wide range of complicated physical/chemical phenomena and scales.

Many flows of engineering interest are incompressible or can be approximated as incompressible to a high degree of accuracy, i.e. low-speed flows. Incompressible flows are challenging to solve numerically because the partial derivative of density with respect to time vanishes. As a result, the governing equations themselves are ill-conditioned. Various methods for solving the incompressible Navier-Stokes equations have been successfully developed to overcome this ill-conditioning [1, 2]. These include but are not limited to the pressure-Poisson [3, 4], fractional-step [5, 6], vorticity-based [7, 8], pseudo-compressibility [9], and characteristic-based methods [10, 11]. The equations governing fully-compressible flows have also been successfully applied to incompressible and low-speed flows by using preconditioning techniques [12–17]. The pseudo-compressible formulation [9, 18–25] is attractive because it is easily extended to three dimensions and applied in conjunction with high-order schemes. This method was originally referred to as the artificial compressibility method by Chorin [9], but Chang and Kwak [26] introduced the more accurate name "pseudo-compressibility method".

High-order methods have the potential to significantly reduce the cost of modelling physically-complex flows, but this potential is challenging to fully realize. As such, the development of robust and accurate high-order methods remains an active area of research. Standard lower-order methods (i.e, methods up to second order) can exhibit excessive numerical dissipation for multi-dimensional problems and are often not practical for physically-complex flows. High-order methods offer improved numerical efficiency for accurate solution representations since fewer computational cells are required to achieve a desired level of accuracy [27]. For hyperbolic conservation laws and/or compressible flow simulations, the main challenge involves obtaining accurate discretizations while ensuring that discontinuities and shocks are handled reliably and robustly [28]. High-order schemes for elliptic partial differential equations (PDEs) that govern diffusion processes should satisfy a maximum principle, even on stretched/distorted meshes, while also remaining accurate [29]. There are many studies of high-order schemes developed for finite-volume [28, 30–39], discontinuous Galerkin [40–49], spectral-difference/spectral-volume [50–54], flux reconstruction [55], and lifting collocation penalty methods [56, 57] on both structured and unstructured mesh. In spite of these many advances, there is still no consensus for a robust, efficient, and accurate scheme that fully deals with

all of the aforementioned issues and is universally applicable to arbitrary meshes.

Harten et al. [28] originally proposed the essentially non-oscillatory (ENO) high-order finite-volume scheme which achieves monotonicity by avoiding the use of computational stencils that contain discontinuities. Weighted ENO (WENO) schemes attempt to simplify the ENO procedure by adopting a stencil-weighting approach [34, 36, 37]. However, both the ENO and WENO schemes encounter difficulties when selecting appropriate stencils on general multi-dimensional unstructured meshes [31, 32, 35, 58] and using these stencils can produce poor conditioning of the linear systems involved in performing the solution reconstruction [35, 58]. These difficulties, along with the associated computational cost and complexities of the ENO and WENO finite-volume schemes, have somewhat limited their application.

Ivan and Groth [59–62] have recently proposed a high-order central essentially non oscillatory (CENO), cell-centered, finite-volume scheme that was demonstrated to remain both accurate and robust for a variety of physically-complex flows. The CENO scheme is based on a hybrid solution reconstruction procedure that combines an unlimited high-order $k$-exact least-squares reconstruction technique with a monotonicity preserving limited piecewise linear least-squares reconstruction algorithm. Fixed central stencils are used for both reconstructions and switching between the two is decided by a smoothness indicator that indicates whether or not the solution is resolved on the computational mesh. This hybrid approach avoids the complexities associated with reconstruction on multiple stencils that other ENO and WENO schemes can encounter. Although originally developed for structured two-dimensional mesh, CENO has been successfully extended to two- and three-dimensional unstructured mesh by McDonald et al. [63]. The CENO method has also been applied to compressible ideal magnetohydrodynamics (MHD) on two-dimensional structured grids by Susanto et al. [64].

The application of high-order solution methods to the pseudo-compressibility approach is not new. Rogers and Kwak [23, 65] and Qian and Zhang [25] employed high-order finite-difference discretizations up to fifth- and sixth-order accuracy, respectively. Using a finite-volume discretization, Chen et al. [66] applied a fifth-order WENO scheme on two-dimensional structured mesh. However, these discretizations are not easily applied to three-dimensional unstructured mesh.

Implicit solution algorithms are commonly applied to improve the stability and convergence of pseudo-compressibility approaches. Implicit algorithms that have been applied to the pseudo-compressible formulation of the Navier-Stokes equations include: approximate factorization [21,67], LU-SGS/SSOR algorithms [25,66,68] and line-relaxation techniques [24, 65]. Due to various approximations and/or linearizations, these schemes are not fully-implicit and their application to unstructured mesh is not straightforward. Jacobian-free Newton-Krylov methods [69–72] offer significant improvements over these types of implicit schemes in terms of rapid convergence. They can robustly handle stiff-wave systems, strong non-linear couplings between equations and offer the potential of quadratic convergence [72, 73].

All of the applications of the pseudo-compressibility formulation discussed previ-

ously have focused on constant density flows. They are not directly applicable to more general low-speed flows that involve combustion or multiple fluids/species. Several researchers have applied the standard pseudo-compressibility approach in conjunction with interface-capturing methods to track the discontinuities in density encountered in multi-fluid flows [74,75]. There are other approaches that are more applicable to the combusting flows encountered in propulsion systems. For example, Riedel [76] applied the pseudo-compressibility approach to reacting flows using an artificial-dissipation-based finite-volume method. A characteristic-based scheme using the pseudo-compressibility approach was derived by Shapiro and Drikakis [77] which is applicable for variable-density, multi-species, isothermal flows. A similar characteristic-based scheme for constant density flows with heat transfer was developed by Azhdarzadeh and Razavi [78].

In this study, the high-order CENO finite-volume scheme is extended to solve the equations governing incompressible, viscous, laminar flows with variable density on three-dimensional general unstructured mesh. For steady flows, the equations are solved using the pseudo-compressibility approach coupled with a parallel implicit Newton-Krylov algorithm. The proposed scheme is extended to unsteady flows via a dual-time stepping approach. The resulting algorithm is applied to both steady and unsteady flows and analyzed in terms of accuracy, computational cost, and rate of convergence. In particular, the spatial and temporal accuracy of solutions are examined and the influence of mesh resolution on accuracy is assessed for several idealized flow problems. The steady flow over an isothermal flat plate, the unsteady decay of Taylor vortices, and the lid-driven flow inside a cube are studied here.

## 2 Pseudo-compressibility approach for variable density low speed flows

In the present research, the equations governing viscous, laminar, compressible flows at low Mach numbers are considered. In three space dimensions, the governing partial-differential equations are

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{v}) = 0, \tag{2.1a}$$

$$\frac{\partial}{\partial t}(\rho \vec{v}) + \nabla \cdot (\rho \vec{v}\vec{v} + p\vec{\mathbf{I}}) = \nabla \cdot \vec{\boldsymbol{\tau}}, \tag{2.1b}$$

$$\frac{\partial}{\partial t}(\rho h) + \nabla \cdot (\rho \vec{v} h) = -\nabla \cdot \vec{q}, \tag{2.1c}$$

where $t$ is the time, $p$ is the total pressure, $\rho$ is the fluid density, $\vec{v}$ is the bulk fluid velocity vector, $h = \int_{T_0}^{T} c_p \mathrm{d}T$ is the fluid enthalpy, $c_p$ is the fluid specific heat, $T$ is the temperature, $\vec{q} = -\kappa \nabla T$ is the heat flux vector, and $\kappa$ is the fluid thermal conductivity. The fluid stress

tensor is given by

$$\tau_{ij}=\mu\left[\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right)-\frac{2}{3}\delta_{ij}\frac{\partial u_k}{\partial x_k}\right], \tag{2.2}$$

where $\mu$ is the dynamic viscosity. At low speeds, density becomes weakly coupled to pressure via the ideal gas law. Here, it is assumed that pressure is constant and density is a function of temperature only, $\rho=\rho(T)$.

For low-Mach-number and incompressible flows, the pseudo-compressibility method modifies the partial derivatives of density with respect to time [9, 18, 20, 79]. While steady-state or time-invariant solutions are unaffected, this modification to Eq. (2.1) destroys the temporal accuracy of the original equations. However, time-accuracy can be regained by using a dual-time-stepping approach [23, 24, 80–82]. Applying the pseudo-compressibility and dual-time-stepping approaches to Eq. (2.1) gives the following governing equations:

$$\frac{\partial \mathbf{U}}{\partial t}+\mathbf{\Gamma}\frac{\partial \mathbf{W}}{\partial \tau}+\frac{\partial}{\partial x}(\mathbf{E}-\mathbf{E_v})+\frac{\partial}{\partial y}(\mathbf{F}-\mathbf{F_v})+\frac{\partial}{\partial z}(\mathbf{G}-\mathbf{G_v})=\mathbf{0}, \tag{2.3}$$

where $\mathbf{U}$ and $\mathbf{W}$ are the vectors of conserved and primitive variables, $\vec{\mathbf{F}}=[\mathbf{E},\mathbf{F},\mathbf{G}]$ and $\vec{\mathbf{F}}_\mathbf{v}=[\mathbf{E_v},\mathbf{F_v},\mathbf{G_v}]$ are the inviscid and viscous solution flux dyads, and $\mathbf{\Gamma}$ is the transformation matrix associated with the pseudo-compressible modifications to the original Navier-Stokes equations. The various vector and matrix quantities are defined as

$$\mathbf{U}=\begin{bmatrix}\rho\\ \rho u\\ \rho v\\ \rho w\\ \rho h\end{bmatrix},\ \mathbf{W}=\begin{bmatrix}p\\ u\\ v\\ w\\ T\end{bmatrix},\ \mathbf{E}=\begin{bmatrix}\rho u\\ \rho u^2+p\\ \rho uv\\ \rho uw\\ \rho uh\end{bmatrix},\ \mathbf{F}=\begin{bmatrix}\rho v\\ \rho vu\\ \rho v^2+p\\ \rho vw\\ \rho vh\end{bmatrix},\ \mathbf{G}=\begin{bmatrix}\rho w\\ \rho wu\\ \rho wv\\ \rho w^2+p\\ \rho wh\end{bmatrix},$$

$$\mathbf{E_v}=\begin{bmatrix}0\\ \tau_{xx}\\ \tau_{xy}\\ \tau_{xz}\\ \kappa\dfrac{\partial T}{\partial x}\end{bmatrix},\ \mathbf{F_v}=\begin{bmatrix}0\\ \tau_{yx}\\ \tau_{yy}\\ \tau_{yz}\\ \kappa\dfrac{\partial T}{\partial y}\end{bmatrix},\ \mathbf{G_v}=\begin{bmatrix}0\\ \tau_{zx}\\ \tau_{yz}\\ \tau_{zz}\\ \kappa\dfrac{\partial T}{\partial z}\end{bmatrix},\ \mathbf{\Gamma}=\begin{bmatrix}\dfrac{1}{\beta} & 0 & 0 & 0 & 0\\ \dfrac{\alpha}{\beta}u & \rho & 0 & 0 & 0\\ \dfrac{\alpha}{\beta}v & 0 & \rho & 0 & 0\\ \dfrac{\alpha}{\beta}w & 0 & 0 & \rho & 0\\ \dfrac{\alpha}{\beta}h & 0 & 0 & 0 & \rho c_p\end{bmatrix},$$

where $\beta$ is the pseudo-compressibility factor, $\alpha$ is a preconditioning parameter, and $\tau$ denotes the pseudo-time since the modified equations are no longer time-accurate. The preconditioning parameter, $\alpha$, controls how the original governing equations are modified. The original pseudo-compressibility method of Chorin [9] corresponds to $\alpha=0$.

When $\alpha=1$ or 2, the pressure time derivatives are added directly to the conserved or primitive formulation of the governing equations, respectively [20].

## 2.1 Eigenstructure

Based on the analysis conducted by Turkel [20] and the numerical results obtained by Qian et al. [83] and Lee and Lee [84], the optimal value of $\alpha$ is 2. However, Malan et al. [85, 86] and Lee and Lee [84] found that a loss of robustness can occur for $\alpha>1$ if $\beta$ is too small. Since larger values of $\alpha$ display better convergence characteristics [83, 84], $\alpha=1$ was selected for the current work. For $\alpha=1$, the Jacobian matrix with respect to the primitive variables of the steady physical-time, unsteady pseudo-time, inviscid system is then

$$\mathbf{A}=\mathbf{\Gamma}^{-1}\frac{\partial\left(\vec{\mathbf{F}}\cdot\hat{n}\right)}{\partial\mathbf{W}}=\begin{bmatrix}0 & n_x\rho\beta & n_y\rho\beta & n_z\rho\beta & 0\\ \frac{n_x}{\rho} & u_n & 0 & 0 & 0\\ \frac{n_y}{\rho} & 0 & u_n & 0 & 0\\ \frac{n_z}{\rho} & 0 & 0 & u_n & 0\\ 0 & 0 & 0 & 0 & u_n\end{bmatrix}, \tag{2.4}$$

where $u_n=\vec{v}\cdot\hat{n}$, $c^2=u_n^2+4\beta$, and $\hat{n}=(n_x,n_y,n_z)$ is a unit vector in the particular direction of interest. The corresponding matrix of right eigenvectors is

$$\mathbf{R}=\begin{bmatrix}-\frac{\rho}{2}(u_n-c) & -\frac{\rho}{2}(u_n+c) & 0 & 0 & 0\\ n_x & n_x & -n_y & -n_z & 0\\ n_y & n_y & n_x & 0 & 0\\ n_z & n_z & 0 & n_x & 0\\ 0 & 0 & 0 & 0 & 1\end{bmatrix}. \tag{2.5}$$

The eigenvalues of the inviscid system defined by Eq. (2.3) in the direction $\hat{n}$ are

$$\boldsymbol{\lambda}=\begin{bmatrix}\frac{1}{2}u_n-\frac{1}{2}\sqrt{{u_n}^2+4\beta}\\ u_n\\ u_n\\ u_n\\ \frac{1}{2}u_n+\frac{1}{2}\sqrt{{u_n}^2+4\beta}\end{bmatrix}. \tag{2.6}$$

The minimum value of $\beta$ was chosen based on the following formulation as proposed by Turkel [20]:

$$\beta = \max\left[2\left(u^2+v^2+w^2\right),\varepsilon\right], \tag{2.7}$$

where $\varepsilon$ is a smallness parameter.

## 3 CENO finite-volume scheme

In the proposed cell-centered finite-volume approach, the physical domain is discretized into finite-sized computational cells and the integral forms of conservation laws are applied to each individual cell. Both hexahedral- and tetrahedral-shaped computational cells are considered here, which are shown schematically in Fig. 1. For any cell $i$, the approach results in the following coupled system of ordinary differential equations (ODEs) for cell-averaged solution quantities:

$$\frac{\mathrm{d}\overline{\mathbf{U}}_i}{\mathrm{d}t}+\mathbf{\Gamma}_i\frac{\mathrm{d}\overline{\mathbf{W}}_i}{\mathrm{d}\tau}=-\frac{1}{V_i}\oiint\left(\vec{\mathbf{F}}-\vec{\mathbf{F}}_\mathbf{v}\right)\cdot\hat{n}\,\mathrm{d}A=-\mathbf{R_i}, \tag{3.1}$$

where the overbar denotes cell-averaged quantities, $V_i$ is the cell volume, $A$ is the area of the face and $\hat{n}$ is the unit vector normal to a given face. Applying Gauss quadrature to evaluate the surface integral in Eq. (3.1) produces a set of nonlinear ODEs given by

$$\frac{\mathrm{d}\overline{\mathbf{U}}_i}{\mathrm{d}t}+\mathbf{\Gamma}_i\frac{\mathrm{d}\overline{\mathbf{W}}_i}{\mathrm{d}\tau}=-\frac{1}{V_i}\sum_{l=1}^{N_\mathrm{f}}\sum_{m=1}^{N_\mathrm{G}}\left[\omega\left(\vec{\mathbf{F}}-\vec{\mathbf{F}}_\mathbf{v}\right)\cdot\hat{n}A\right]_{i,l,m}, \tag{3.2}$$

where $N_\mathrm{f}$ is the number of faces (equal to 4 for tetrahedra and 6 for hexahedra), $N_\mathrm{G}$ is the number of Gauss quadrature points and $\omega$ is the corresponding quadrature weight. In Eq. (3.2), the number of quadrature points required along each face is a function of the reconstruction order and number of spatial dimensions. For tetrahedra and Cartesian (hexahedra with rectangular faces) cells, the Gauss quadrature points can be directly

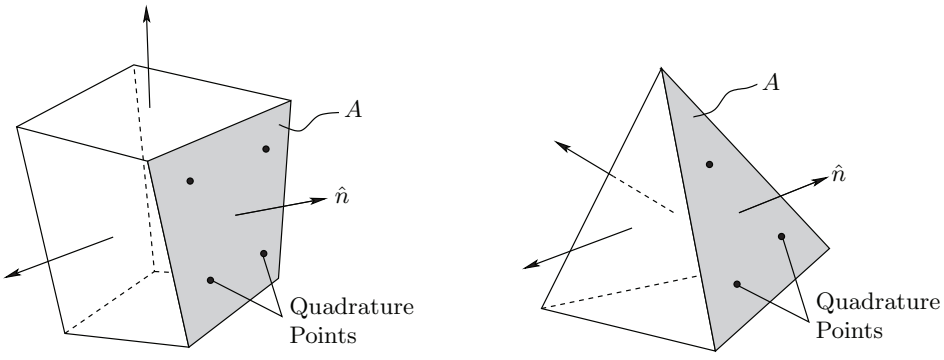



Figure 1: Computational cells. (Left) Arbitrary hexahedron and (right) tetrahedron.

Table 1: Gauss quadrature rules used for cell face integration.

| Reconstruction | Tetrahedra | | Cartesian / Hexahedra | |
|---|---|---|---|---|
| | Points | Polynomial Degree | Points | Polynomial Degree |
| Constant ($k$=0) | 1 | 1 | 1 | 1 |
| Linear ($k$=1) | 1 | 1 | 1 | 1 |
| Quadratic ($k$=2) | 3 | 2 | 4 | 3 |
| Cubic ($k$=3) | 4 | 3 | 4 | 3 |
| Quartic ($k$=4) | 6 | 4 | 9 | 5 |

mapped from the canonical form to the local Cartesian coordinate system. More general hexahedra can have non-rectangular faces or faces composed of vertices that do not all lie on a particular plane. In this case, the Gauss quadrature points are mapped to the Cartesian coordinate system using a trilinear coordinate transformation [87,88]. The coefficients corresponding to the quadrature rules applied here are as given by Felippa [89] and summarized in Table 1.

### 3.1 Trilinear coordinate transformation for arbitrary hexahedra

As mentioned previously, the geometric representation of a general hexahedral cell, $Q$, is based on a trilinear mapping to a reference hexahedron, $\hat{Q}$ [87,88]. Although $Q$ may have faces which are non-planar, $\hat{Q}$ is a unit cube. As shown in Fig. 2, the mapping associates any reference location $\hat{r}=(\hat{x},\hat{y},\hat{z})$ in $\hat{Q}$ to a point $\vec{r}=(x,y,z)$ in $Q$:

$$\vec{r}=(x,y,z)=\vec{a}+\vec{b}\hat{x}+\vec{c}\hat{y}+\vec{d}\hat{z}+\vec{e}\hat{x}\hat{y}+\vec{f}\hat{x}\hat{z}+\vec{g}\hat{y}\hat{z}+\vec{h}\hat{x}\hat{y}\hat{z}, \tag{3.3}$$

where $\vec{a}$, $\vec{b}$, $\vec{c}$, $\vec{d}$, $\vec{e}$, $\vec{f}$, $\vec{g}$ and $\vec{h}$ are the transformation coefficient vectors. Using the transformation defined by Eq. (3.3), integrals over the volume of an arbitrary hexahedral cell, $Q$, are instead evaluated by performing the integration over the reference element, $\hat{Q}$, as

$$\iiint_{Q} f(x,y,z)\,\mathrm{d}Q=\iiint_{\hat{Q}} f(\vec{r}(\hat{x},\hat{y},\hat{z}))\,J(\hat{x},\hat{y},\hat{z})\,\mathrm{d}\hat{Q}, \tag{3.4}$$

where $f(x,y,z)=f(\vec{r})$ is an arbitrary function defined in the general Cartesian coordinate system. The determinant of the transformation Jacobian, $J$, is given by

$$J(\hat{x},\hat{y},\hat{z})=\vec{X}(\hat{y},\hat{z})\cdot\left(\vec{Y}(\hat{x},\hat{z})\times\vec{Z}(\hat{x},\hat{y})\right), \tag{3.5}$$

where $\vec{X}=\partial\vec{r}/\partial\hat{x}$, $\vec{Y}=\partial\vec{r}/\partial\hat{y}$ and $\vec{Z}=\partial\vec{r}/\partial\hat{z}$. Integrals over the cell faces are evaluated in a similar manner. For example, integrating $f(x,y,z)$ over a face in the logical $x$-direction, $\Gamma_x$, gives

$$\iint_{\Gamma_x} f(x,y,z)\,\mathrm{d}\Gamma_x=\iint_{\Gamma_{\hat{x}}} f(\vec{r}(\hat{x},\hat{y},\hat{z}))\,J_x(\hat{x},\hat{y},\hat{z})\,\mathrm{d}\Gamma_{\hat{x}}, \tag{3.6}$$

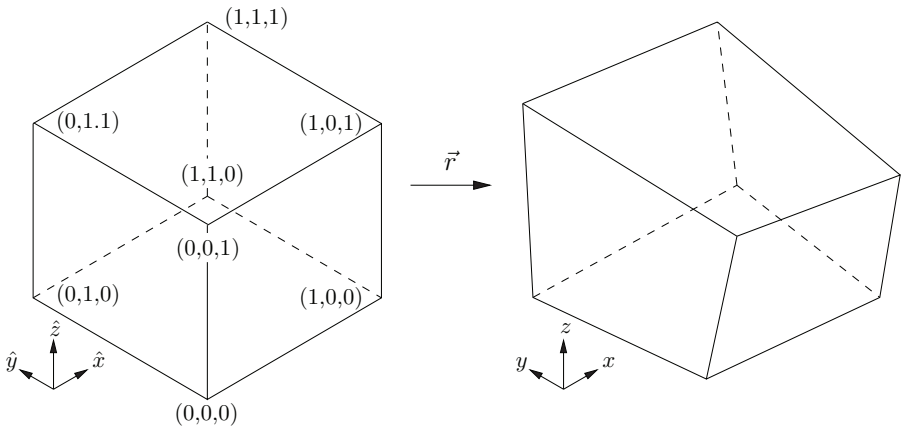



Figure 2: (Left) Reference cube $\hat{Q}$ and (right) arbitrary hexahedron $Q$.

where $\Gamma_{\hat{x}}$ is the corresponding surface on the reference cube. The surface Jacobian, $J_x$, is

$$J_x(\hat{x},\hat{y},\hat{z}) = \left|\vec{Y}(\hat{x},\hat{z}) \times \vec{Z}(\hat{x},\hat{y})\right|. \tag{3.7}$$

The unit normal to the surface $\Gamma_x$ at a particular location is related to the surface Jacobian by

$$\hat{n} = \frac{\vec{Y}(\hat{x},\hat{z}) \times \vec{Z}(\hat{x},\hat{y})}{\left|\vec{Y}(\hat{x},\hat{z}) \times \vec{Z}(\hat{x},\hat{y})\right|}. \tag{3.8}$$

Using Eqs. (3.4) and (3.6), relations can be derived for the cell volume as well as the area of cell faces by setting $f(x,y,z)$ to unity. The transformation defined in Eq. (3.3) is also used to transform quadrature points and weights from the canonical element to those for the arbitrary hexahedron.

### 3.2 CENO reconstruction

Evaluating Eq. (3.2) requires integrating the numerical flux along the cell faces, but in the proposed finite-volume approach only cell-averaged quantities are known. The high-order CENO method uses a hybrid solution reconstruction process to interpolate the primitive solution state at the Gauss quadrature points along each face [59–62]. This hybrid approach involves a $k$-exact reconstruction in smooth or fully-resolved regions which is switched to a limited piecewise linear reconstruction when discontinuities in solution content are encountered and identified. Fixed central stencils are used for both reconstructions. The switching provides a means of eliminating spurious oscillations that can occur near regions where the solution is under-resolved. It is facilitated by a parameter referred to here as the "smoothness indicator" which indicates the current level of resolution.

Even though most features of low-speed flows are relatively smooth, there are cases where discontinuities can occur, such as across flame fronts or fluid interfaces. Under-resolved solution content is also an issue. Oscillations can occur for relatively smooth flows when there is insufficient mesh resolution.

### 3.2.1 *k*-exact reconstruction

The CENO spatial discretization scheme is based on the high-order *k*-exact least-squares reconstruction technique of Barth [30]. The *k*-exact high-order reconstruction algorithm begins by assuming that the solution within each cell can be represented by the following Taylor series expansion in three dimensions:

$$u_i^k(x,y,z) = \sum_{p_1=0}^{(p_1+p_2+p_3)\leq k} \sum_{p_2=0} \sum_{p_3=0} (x-x_i)^{p_1}(y-y_i)^{p_2}(z-z_i)^{p_3} D_{p_1p_2p_3}, \tag{3.9}$$

where $u_i^k$ is the reconstructed solution quantity in cell $i$, $(x_i,\ y_i,\ z_i)$ are the coordinates of the cell centroid, $k$ is the degree of the piecewise polynomial interpolant and $D_{p_1p_2p_3}$ are the unknown coefficients of the Taylor series expansion. The summation indices, $p_1$, $p_2$ and $p_3$, must always satisfy the condition that $(p_1+p_2+p_3)\leq k$.

The following conditions are applied to determine the unknown coefficients: i) the solution reconstruction must reproduce polynomials of degree $\leq k$ exactly; ii) the mean or average value within the computational cell must be preserved; and iii) the reconstruction must have compact support. The second condition states that

$$\overline{u}_i = \frac{1}{V_i} \iiint_{V_i} u_i^k(x,y,z)\,\mathrm{d}V, \tag{3.10}$$

where $\overline{u}_i$ is the cell average.

The third condition dictates the number and location of neighboring cells 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. (3.10)). For any type of mesh, the total number of unknown coefficients for a particular order is given by

$$N = \frac{1}{d!} \prod_{n=1}^{d} (k+n), \tag{3.11}$$

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.

For structured mesh, the stencil is generally derived directly from the underlying data structures [59–62]. However, stencil selection is not as straightforward for the unstructured mesh of interest here. Rather, the stencil is constructed by recursively selecting nearest neighbors — i.e. two cells which share a common face — until at least the minimum number of neighbors is met. Additional neighbors are included to ensure that the

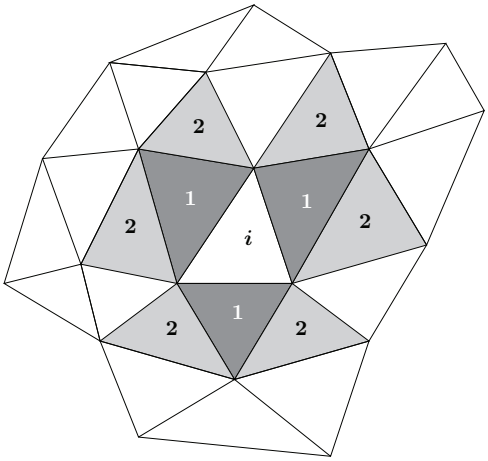



Figure 3: Supporting stencil for reconstruction about cell $i$. First- and second-level neighbors are indicated in the figure.

stencil is not biased in any particular direction and that the reconstruction remains reliable on poor quality meshes with high aspect ratio cells [59]. A sample stencil about a cell $i$ is illustrated in two-dimensions in Fig. 3. The first layer of surrounding cells, denoted by 1, are selected first. If more neighbors are required, the next nearest neighbors are selected, denoted by 2. This process repeats until the stencil is deemed sufficient.

The unknown coefficients in Eq. (3.9) are determined by assuming that the reconstruction polynomial, $u_i^k$, extends to all of the computational cells within the supporting stencil. For each neighboring cell, $j$, of the central cell, $i$, a constraint is formed by requiring that

$$\overline{u}_j = \frac{1}{V_j} \iiint_{V_j} u_i^k(x,y,z)\,\mathrm{d}V. \tag{3.12}$$

Since the constraints of Eqs. (3.10) and (3.12) result in an over-determined system of linear equations for cell $i$, a least-squares solution for the coefficients, $D_{p_1p_2p_3}$, is obtained in each cell. Equation (3.10) is strictly enforced by Gaussian elimination and a minimum-error solution to the remaining constraint equations is sought. Substituting Eq. (3.9) into Eq. (3.10) and rearranging for $D_{000}$ gives

$$D_{000} = \overline{u}_i - \sum_{(p_1+p_2+p_3)>0}^{(p_1+p_2+p_3)\le k} \sum\sum D_{p_1p_2p_3} \frac{1}{V_i} \iiint_{V_i} (x-x_i)^{p_1}(y-y_i)^{p_2}(z-z_i)^{p_3}\,\mathrm{d}V. \tag{3.13}$$

A similar relation is obtained using Eqs. (3.9) and (3.12) to give

$$D_{000} = \overline{u}_j - \sum_{(p_1+p_2+p_3)>0}^{(p_1+p_2+p_3)\le k} \sum\sum D_{p_1p_2p_3} \frac{1}{V_j} \iiint_{V_j} (x-x_i)^{p_1}(y-y_i)^{p_2}(z-z_i)^{p_3}\,\mathrm{d}V. \tag{3.14}$$

Equating Eq. (3.13) and Eq. (3.14) gives

$$\overline{u}_j-\overline{u}_i=\sum_{(p_1+p_2+p_3)>0}^{(p_1+p_2+p_3)\leq k}\sum\sum\sum D_{p_1p_2p_3}(\widehat{x^{p_1}y^{p_2}z^{p_3}})_{ij}, \tag{3.15}$$

where

$$\begin{aligned}(\widehat{x^{p_1}y^{p_2}z^{p_3}})_{ij}=&\frac{1}{V_j}\iiint_{V_j}(x-x_i)^{p_1}(y-y_i)^{p_2}(z-z_i)^{p_3}\,\mathrm{d}V\\&-\frac{1}{V_i}\iiint_{V_i}(x-x_i)^{p_1}(y-y_i)^{p_2}(z-z_i)^{p_3}\,\mathrm{d}V\end{aligned} \tag{3.16}$$

are geometric coefficients which only depend on the mesh geometry. The new overdetermined linear system is formed using Eq. (3.15) and given by

$$\begin{bmatrix}(\widehat{x^0y^0z^1})_{i1} & \cdots & (\widehat{x^{p_1}y^{p_2}z^{p_3}})_{i1} & \cdots & (\widehat{x^ky^0z^0})_{i1}\\ \vdots & & \vdots & & \vdots\\ (\widehat{x^0y^0z^1})_{ij} & \cdots & (\widehat{x^{p_1}y^{p_2}z^{p_3}})_{ij} & \cdots & (\widehat{x^ky^0z^0})_{ij}\\ \vdots & & \vdots & & \vdots\\ (\widehat{x^0y^0z^1})_{iN_\mathrm{n}} & \cdots & (\widehat{x^{p_1}y^{p_2}z^{p_3}})_{iN_\mathrm{n}} & \cdots & (\widehat{x^ky^0z^0})_{iN_\mathrm{n}}\end{bmatrix}\cdot\begin{bmatrix}D_{001}\\ \vdots\\ D_{p_1p_2p_3}\\ \vdots\\ D_{k00}\end{bmatrix}=\begin{bmatrix}\overline{u}_1-\overline{u}_i\\ \vdots\\ \overline{u}_j-\overline{u}_i\\ \vdots\\ \overline{u}_{N_\mathrm{n}}-\overline{u}_i\end{bmatrix}, \tag{3.17}$$

where $N_\mathrm{n}$ is the number of neighbors in the stencil. The remaining polynomial coefficient, $D_{000}$, is obtained from Eq. (3.13) after obtaining the least-squares solution to the overdetermined system given by Eq. (3.17).

The resulting coefficient matrix of the linear system defined in Eq. (3.17) depends only on the mesh geometry, so it can be inverted and stored prior to computations [38, 90]. Either a Householder QR factorization algorithm or orthogonal decomposition by the SVD method was used to solve the weighted least-squares problem [91].

To improve the locality of the reconstruction, inverse distance weighting was applied to each individual constraint equation in Eq. (3.17) [92]. For the reconstruction in cell $i$, the weight applied to row $j$ is

$$w_j=\frac{1}{|\vec{x}_j-\vec{x}_i|}, \tag{3.18}$$

where $\vec{x}_j$ is the centroid of the neighbor cell $j$.

It is well established that conditioning of the least-squares problem for the reconstruction coefficients can be an issue. Ivan and Groth [62] found that the use of inverse distance weighting slightly improved the condition number of Eq. (3.17). More recently, Jalali and Ollivier-Gooch [93] showed that the conditioning can be greatly improved via the application of a simple column-scaling that effectively makes the condition number of the problem independent of the mesh size and aspect ratio. Based on these findings,

the conditioning of the least-squares reconstruction problem can be well controlled by a combination of inverse-distance weighting, column-scaling, and the use the single central stencil. Using this combination, accurate reconstructions are readily obtained on a wide range of unstructured mesh, at least for orders up to four.

### 3.2.2 Reconstruction at boundaries

To enforce conditions at physical boundaries, the least-squares reconstruction is constrained in adjacent control volumes without altering the reconstruction's order of accuracy [38, 39, 60]. Constraints are placed on the least-squares reconstruction for each variable to obtain the desired value/gradient (Dirichlet/Neumann) at each Gauss integration point. Here we implement them as Robin boundary conditions

$$f(\vec{x}) = a(\vec{x}) f_{\mathrm{D}}(\vec{x}) + b(\vec{x}) f_{\mathrm{N}}(\vec{x}), \tag{3.19}$$

where $a(\vec{x})$ and $b(\vec{x})$ are weights which define the contribution of the Dirichlet, $f_{\mathrm{D}}(\vec{x})$, and Neumann, $f_{\mathrm{N}}(\vec{x})$, components, respectively. In terms of the cell reconstruction, the Dirichlet condition is simply expressed as

$$f_{\mathrm{D}}\left(\vec{x}_{\mathrm{g}}\right) = u^k\left(\vec{x}_{\mathrm{g}}\right), \tag{3.20}$$

where $\vec{x}_{\mathrm{g}}$ is the location of the Gauss quadrature point. The Neumann condition is

$$\begin{aligned} f_{\mathrm{N}}\left(\vec{x}_{\mathrm{g}}\right) &= \nabla u^k\left(\vec{x}_{\mathrm{g}}\right) \cdot \hat{n}_{\mathrm{g}} \\ &= \sum_{p_1+p_2+p_3=1}^{(p_1+p_2+p_3)\leq k} \sum \sum \Delta x^{p_1-1} \Delta y^{p_2-1} \Delta z^{p_3-1} \left[ p_1 \Delta y \Delta z n_x + p_2 \Delta x \Delta z n_y + p_3 \Delta x \Delta y n_z \right] D_{p_1 p_2 p_3}, \end{aligned} \tag{3.21}$$

where $\Delta(\cdot) = (\cdot)_{\mathrm{g}} - (\cdot)_i$, the subscript $i$ denotes the location of the cell's centroid adjacent to the boundary, and the subscript $g$ denotes the location of the Gauss quadrature point.

Exact enforcement of the boundary constraints described by Eq. (3.19) is sought, which adds linear equality constraints to the original least-squares problem (Eq. (3.17)). Gaussian elimination with full pivoting is first applied to remove these additional boundary constraints and the remaining least-squares problem is solved as described in Section 3.2.1.

For inflow/outflow or farfield-type boundary conditions where the reconstructed variables are not related, the constraints may be applied separately to each variable. Thus, a separate least-squares problem with equality constraints can be set up for each variable that is solved independently of the other variables. More complex boundary conditions involving linear combinations of solution variables, such as symmetry or inviscid solid walls (where $\vec{v} \cdot \hat{n} = 0$), can cause the reconstruction coefficients in Eq. (3.9) for different variables to become coupled. For these types of coupled boundary conditions, the reconstruction for all variables is performed together [39, 60]. Thus the final matrix in Eq. (3.17) for the constrained least-squares reconstruction contains the individual constraints for each variable (Eq. (3.19)), the relational constraints, and the approximate mean conservation equations for each variable (Eq. (3.15)).

### 3.2.3 Smoothness indicator

After performing a $k$-exact reconstruction for each solution variable in each computational cell, the smoothness indicator is computed for every reconstructed variable to identify under-resolved solution content. It is evaluated as

$$S=\frac{\sigma}{\max[(1-\sigma),\delta]}\frac{(\text{SOS}-\text{DOF})}{\text{DOF}-1}, \tag{3.22}$$

where $\sigma$ is a smoothness parameter, $\delta$ 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 smoothness parameter, $\sigma$, for a cell $i$ is given by

$$\sigma=1-\frac{\sum_j\left[u_j^k(x_j,y_j,z_j)-u_i^k(x_j,y_j,z_j)\right]^2}{\sum_j\left[u_j^k(x_j,y_j,z_j)-\overline{u}_i\right]^2}, \tag{3.23}$$

where $u$ is the solution variable of interest, the subscript $j$ refers to the cells in the reconstruction stencil, $u_j^k(x_j,y_j,z_j)$ is the reconstructed solution in cell $j$ evaluated at the cell's centroid, $u_i^k(x_j,y_j,z_j)$ is the projected value of the reconstruction polynomial for cell $i$ evaluated at $(x_j,y_j,z_j)$, and $\overline{u}_i$ is the average value for cell $i$. By definition, $\sigma$ can take on values between negative infinity and one. A value of unity indicates that the solution is smooth whereas small or negative values indicate large variations in solution content within the reconstruction stencil.

Note that, based on the analysis of Ivan and Groth [62], $\sigma\approx 1-\mathcal{O}(\Delta x^{2k})$ for smooth solution content. Thus, $\sigma\to 1$ as $\Delta x\to 0$ at a rate much faster than the formal order of accuracy of the scheme. Conversely, $\sigma\approx 1-\mathcal{O}(1)$ when the solution is not smooth, so $\sigma$ will be much less than unity.

The behavior of the smoothness indicator is demonstrated in Fig. 4. As $\sigma$ approaches unity, the smoothness indicator grows rapidly. Solutions are deemed smooth when the value of $S$ is above critical value, $S_c$. Previous studies found that values for $S_c$ between 1000–5000 provided an excellent balance between stability and accuracy [59]. Typical values for $S$ in smooth regions tend to be orders of magnitude greater than these cutoff limits. For all test cases considered, a value for $S_c$ of 1000 was used.

All prior implementations of the smoothness indicator presented in Eqs. (3.22) and (3.23) were applied to compressible flows only. The smoothness indicator functions and behaves exactly the same for incompressible flows: it detects discontinuities and under-resolved content. Although there are no shocks or large discontinuities present in any of the idealized flows considered here, the smoothness indicator still identifies under-resolved solution content that could cause unphysical oscillations. It also provides numerical stability early in the solution process when intermediate solutions are far from the final, possibly smooth, solution. Large unphysical jumps or discontinuities similar to shocks can occur in these intermediate unconverged solutions, especially when inexact

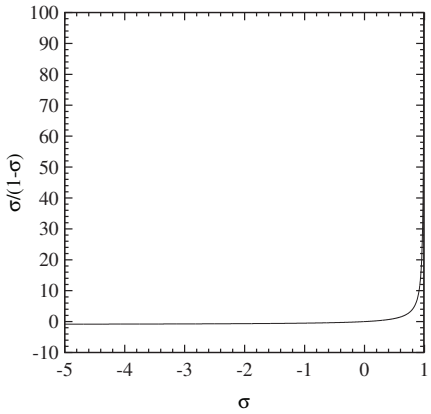



Figure 4: Variation of smoothness indicator with $\sigma$.

Newton's methods are employed (inexact Newton methods are the focus of the current research).

In regions where $S$ is below the critical value, monotonicity is preserved by reverting the high-order $k$-exact reconstruction to a limited piecewise linear ($k$=1) reconstruction. Limiting is performed using the procedure developed by Park et al. [94] specifically for multiple dimensions in conjunction with the slope limiter function of Venkatakrishnan [95]. The use of the multi-dimensional limiter formulation was required to correctly recover linear functions on unstructured mesh without unnecessary limiting or clipping of solution content. Note that the limited linear reconstruction in non-smooth regions is only applied to the inviscid terms in Eq. (3.2). The viscous flux is still evaluated using the higher-order representation since it is generally the inviscid terms that generate instabilities.

### 3.3 Inviscid numerical flux evaluation

An upwind Godunov scheme is used to integrate the inviscid numerical flux, $\vec{\mathbf{F}}$, over the cell face. Given the left and right solution states, $\mathbf{W_L}$ and $\mathbf{W_R}$, the numerical flux at the cell interface is defined as

$$\vec{\mathbf{F}}\cdot\hat{n} = \mathcal{F}(\mathbf{W_L}, \mathbf{W_R}, \hat{n}), \tag{3.24}$$

where $\mathcal{F}$ is a flux function which solves a Riemann problem, $\mathcal{R}$, in a direction aligned along the face normal, $\hat{n}$. The left and right solution states at the interface are determined using the $k$-exact reconstruction procedure described in Section 3.2. As a result, the leading truncation error due to the inviscid operator is $\mathcal{O}\left(\Delta x^{k+1}\right)$.

The inviscid numerical flux, $\mathcal{F}$, is formulated based on the pseudo-compressible system (i.e., steady physical-time and unsteady pseudo-time) using the eigensystem presented in Section 2.1. Note that without applying the pseudo-compressibility approach to Eq. (2.1), it is not possible to formulate such a numerical flux in the low-Mach-number/

incompressible limit. Similar to the approach taken by Lee and Lee [84], the numerical flux is derived based on Roe's approximate Riemann solver [96,97] using the eigensystem in Eq. (2.3). The numerical flux at the interface between two cells is given by

$$\mathbf{F}(\mathcal{R}(\mathbf{W_L},\mathbf{W_R}))=\frac{1}{2}(\mathbf{F_R}+\mathbf{F_L})-\frac{1}{2}|\hat{\mathbf{A}}|\Delta\mathbf{W}, \tag{3.25}$$

where $\mathbf{F_L}$ and $\mathbf{F_R}$ are the inviscid fluxes evaluated based on $\mathbf{W_L}$ and $\mathbf{W_R}$, $\Delta\mathbf{W}=\mathbf{W_R}-\mathbf{W_L}$, $|\hat{\mathbf{A}}|=\hat{\mathbf{R}}|\hat{\mathbf{\Lambda}}|\hat{\mathbf{R}}^{-1}$, $\hat{\mathbf{R}}$ is the matrix of primitive variable right eigenvectors and $\hat{\mathbf{\Lambda}}$ is the eigenvalue matrix. The matrix $\hat{\mathbf{A}}$ is the linearized flux Jacobian evaluated at a reference state, $\hat{\mathbf{W}}$. A simple arithmetic average between the left and right states was chosen as the reference state.

The pseudo-compressibility factor, $\beta$, essentially alters the wave-speeds of the incompressible system. As such, it effectively controls the degree of dissipation added to the inviscid flux. It also controls the degree of preconditioning applied to the time-invariant system.

### 3.4 Viscous numerical flux evaluation

The viscous fluxes at each quadrature point are evaluated by averaging the interface solution and gradients

$$\mathcal{G}(\mathbf{W_L},\mathbf{W_R},\nabla\mathbf{W_L},\nabla\mathbf{W_R})=\mathbf{F_v}\left\{\frac{1}{2}(\mathbf{W_L}+\mathbf{W_R}),\frac{1}{2}(\nabla\mathbf{W_L},\nabla\mathbf{W_R})\right\}. \tag{3.26}$$

Because derivatives of the reconstructed polynomial are required, the leading truncation error due to the viscous operator is only $\mathcal{O}(\Delta x^k)$. The degree of the reconstruction polynomial is therefore increased by one to match the leading truncation error introduced by the inviscid operator. However, the original Gauss quadrature rule selected for integrating over the cell faces remains unchanged. It is selected to maintain an order of accuracy of $\mathcal{O}(\Delta x^{k+1})$ when integrating the fluxes over the cell faces. The preceding approach was found to provide stable, although not strictly positive, discretizations of viscous fluxes without the occurrence of any even/odd decoupling phenomena [60,62].

For piecewise-linear ($k=1$) representations, second-order accuracy ($\mathcal{O}(\Delta x^{k+1})$) of the viscous operator can be achieved without increasing the degree of the polynomial interpolant. In this case, the average gradient at the interface is evaluated by [98]

$$\nabla\mathbf{W}_{i+1/2}=(\mathbf{W}_n-\mathbf{W}_p)\frac{\hat{n}}{\hat{n}\cdot\vec{r}_{pn}}+\left(\overline{\nabla\mathbf{W}}-\overline{\nabla\mathbf{W}}\cdot\vec{r}_{pn}\frac{\hat{n}}{\hat{n}\cdot\vec{r}_{pn}}\right), \tag{3.27}$$

where $\mathbf{W}_n$ and $\mathbf{W}_p$ are the solutions at the center of the two adjacent cells, and $\vec{r}_{pn}=\vec{x}_n-\vec{x}_p$ is the vector between the neighboring cell centers. The volume-weighted average of the neighboring cell gradients, $\overline{\nabla\mathbf{W}}$, is evaluated as

$$\overline{\nabla\mathbf{W}}=\chi\nabla\mathbf{W}_p+(1-\chi)\nabla\mathbf{W}_n, \tag{3.28}$$

where

$$\chi = V_p/(V_p+V_n).$$

Eq. (3.27) is second-order accurate if the gradient representation is also second-order accurate. Thus, $k+1$ reconstruction is not required for $k=1$.

## 3.5 Inexact Newton method for steady and unsteady flows

Integrating the governing equations is performed in parallel to fully take advantage of modern computer architectures. This is carried out by dividing the computational domain up using a standard and well-established parallel graph partitioning algorithm, called Parmetis [99], and distributing the computational cells among the available processors. Solutions for each computational sub-domain are simultaneously computed on each processor. The proposed computational algorithm was implemented using the message passing interface (MPI) library and the Fortran 90 programming language [100]. Ghost cells, which surround an individual local solution domain and overlap cells on neighboring domains, are used to share solution content through inter-partition communication.

Newton's method is applied in this work for both steady state relaxation and transient continuation. For transient calculations, a dual-time-stepping approach is used [23,24,80–82] with the family of high-order backwards difference formulas (BDF) to discretize the physical time derivative. In both cases, steady and unsteady, Newton's method is used to relax

$$\mathbf{R}^*(\mathbf{W}) = \mathbf{R} + \frac{\mathrm{d}\overline{\mathbf{U}}}{\mathrm{d}t} = \mathbf{0}, \tag{3.29}$$

where $\mathrm{d}\overline{\mathbf{U}}/\mathrm{d}t = \mathbf{0}$ for steady problems.

This particular implementation follows the algorithm developed by Groth et al. [101–103] specifically for use on large multi-processor parallel clusters. The implementation makes use of a Jacobian-free inexact Newton method coupled with an iterative Krylov subspace linear solver. It was originally developed for two-dimensional multi-block body-fitted mesh and is extended here for application to three-dimensional unstructured mesh.

### 3.5.1 Inexact Newton method for steady problems

For steady problems, a solution to Eq. (3.29) is sought by iteratively solving a sequence of linear systems given an initial estimate, $\mathbf{W}^0$. Successively improved estimates are obtained by solving

$$\left(\frac{\partial \mathbf{R}}{\partial \mathbf{W}}\right)^k \Delta\mathbf{W}^k = \mathbf{J}(\mathbf{W}^k)\,\Delta\mathbf{W}^k = -\mathbf{R}(\mathbf{W}^k), \tag{3.30}$$

where $\mathbf{J} = \partial\mathbf{R}/\partial\mathbf{W}$ is the residual Jacobian. After solving Eq. (3.30) for $\Delta\mathbf{W}^k$, the improved solution at step $k$ is determined from

$$\mathbf{W}^{k+1} = \mathbf{W}^k + \Delta\mathbf{W}^k. \tag{3.31}$$

The Newton iterations proceed until some desired reduction of the norm of the residual is achieved and the condition $\|\mathbf{R}(\mathbf{W}^k)\| < \epsilon\|\mathbf{R}(\mathbf{W}^0)\|$ is met. The tolerance, $\epsilon$, used in this work was typically $10^{-7}$ for steady problems.

Each step of Newton's method requires the solution of the linear problem defined in Eq. (3.30). This system tends to be relatively large, sparse, and non-symmetric for which iterative methods have proven much more effective than direct methods. One effective method for a large variety of problems, which is used here, is the generalized minimal residual (GMRES) technique of Saad and co-workers [71, 104–106]. This is an Arnoldi-based solution technique which generates orthogonal bases of the Krylov subspace to construct the solution. The technique is particularly attractive because **J** does not need to be explicitly formed and instead only matrix-vector products are required at each iteration to create new trial vectors. This drastically reduces the required storage. Another advantage is that iterations are terminated based on only a by-product estimate of the residual which does not require explicit construction of the intermediate residual vectors or solutions. Termination also generally only requires solving the linear system to some specified tolerance, $\|\mathbf{R}^k + \mathbf{J}^k\Delta\mathbf{W}^k\|_2 < \zeta\|\mathbf{R}(\mathbf{W}^k)\|_2$, where $\zeta$ is typically in the range $0.1-0.5$ [69]. We use a restarted version of the GMRES algorithm here, GMRES($m$), that minimizes storage by restarting every $m$ iterations.

Right preconditioning of **J** is performed to help facilitate the solution of the linear system without affecting the solution residual, **b**. The preconditioning takes the form

$$\left(\mathbf{J}\mathbf{M}^{-1}\right)(\mathbf{M}\mathbf{x}) = \mathbf{b}, \tag{3.32}$$

where **M** is the preconditioning matrix. A combination of an additive Schwarz global preconditioner and a block incomplete lower-upper (BILU) local preconditioner is used. In additive Schwarz preconditioning, the solution in each block is updated simultaneously and shared boundary data is not updated until a full cycle of updates has been performed on all domains. The preconditioner is defined as follows

$$\mathbf{M}^{-1} = \sum_{k=1}^{N_\mathrm{b}} \mathbf{B}_k^T \mathbf{M}_k^{-1} \mathbf{B}_k, \tag{3.33}$$

where $N_\mathrm{b}$ is the number of blocks and $\mathbf{B}_k$ is the gather matrix for the $k$th domain. The local preconditioner, $\mathbf{M}_k^{-1}$, in Eq. (3.33) is based on block ILU($p$) factorization [106] of the Jacobian for the first- or second-order approximation of each domain. A first-order Jacobian approximation is used when $k=0$ whereas a second-order approximation is used when $k>0$. The level of fill, $p$, was maintained between 0–2 to reduce storage requirements. Larger values of $p$ typically offer improved convergence characteristics for the linear system at the expense of storage. To further reduce computational storage, reverse Cuthill-McKee (RCM) matrix reordering is used to permute the Jacobian's sparsity pattern into a banded matrix form with a small bandwidth [107].

### 3.5.2 Implicit-Euler startup

Newton's method can fail when initial solution estimates fall outside the radius of convergence. To ensure global convergence of the algorithm, the implicit Euler startup procedure of Mulder and Van Leer [108] with successive evolution/relaxation (SER) was used. Application of this startup procedure to the semi-discrete form of the governing equations gives

$$\left[\frac{\mathbf{\Gamma}}{\Delta\tau^k}+\left(\frac{\partial\mathbf{R}}{\partial\mathbf{W}}\right)^k\right]\Delta\mathbf{W}^k=-\mathbf{R}^k, \tag{3.34}$$

where $\Delta\tau^k$ is the time step. In the SER approach, the time step is varied from some small finite value and gradually increased as the steady state solution is approached. As $\Delta\tau^k\to\infty$, Newton's method is recovered.

In the quasi-Newton and SER methods, the time step size was determined by considering the inviscid Courant-Friedrichs-Lewy (CFL) and viscous Von Neumann stability criteria based on the pseudo-compressible system. The maximum permissible time step for each local cell is determined by

$$\Delta\tau^k\leq\text{CFL}\cdot\min\left[\frac{\Delta x}{\max(\boldsymbol{\lambda})},\frac{\rho\Delta x^2}{\mu}\right], \tag{3.35}$$

where CFL is a constant greater than zero which determines the time step size and $\Delta x=V^{1/3}$ is a measure of the grid size. Using SER, the CFL number for the $k$th iteration is computed using the following relation:

$$\text{CFL}^k=\text{CFL}^0\frac{\|\mathbf{R}(\mathbf{W^0})\|}{\|\mathbf{R}(\mathbf{W^k})\|}. \tag{3.36}$$

Typically, a value for $\text{CFL}^0$ between 10–100 is used.

### 3.5.3 Dual-time stepping approach for unsteady problems

Even though possibly more efficient implicit high-order time-marching schemes exist (see, for example, the studies by Bijl et al. [109, 110]), the well-known high-order BDF time-marching schemes are considered here [111]. Applying the high-order BDF temporal discretization and Newton's method to the semi-discrete form of the governing equations, Eq. (3.29), gives

$$\left[\frac{a}{\Delta t^n}\left(\frac{\partial\mathbf{U}}{\partial\mathbf{W}}\right)^{(n+1,k)}+\frac{\mathbf{\Gamma}^{(n+1,k)}}{\Delta\tau^k}+\left(\frac{\partial\mathbf{R}}{\partial\mathbf{W}}\right)^{(n+1,k)}\right]\Delta\mathbf{W}^{(n+1,k)}=-\mathbf{R}^{*(n+1,k)}, \tag{3.37}$$

where $n$ is the outer time level, $k$ is the now the inner iteration level, and $a$ is a constant which depends on the temporal discretization ($a$=1 for implicit Euler, 3/2 for BDF2, 11/6 for BDF3 and 25/12 for BDF4).

In the dual time-stepping procedure, at each physical time step $n$, a steady problem is solved using the Newton procedure described in Section 3.5.1. The stability of the unsteady system is now governed by the physical time step size, which is determined based on the CFL criterion:

$$\Delta t^n \leq \mathrm{CFL}_{\mathrm{phys}} \cdot \min\left(\frac{\Delta x}{u^2+v^2+w^2}\right), \tag{3.38}$$

where the $\mathrm{CFL}_{\mathrm{phys}}$ is the physical CFL number. The convergence tolerance for the inner iteration loop used here was $\epsilon = 10^{-4}$. A tolerance of $10^{-4}$ was shown to provide a good balance between accuracy and computation time by Tabesh and Zingg [112].

Coefficients for the different BDF schemes are provided in Table 2. The order of accuracy of the temporal discretization scheme was chosen to match the accuracy of the spatial discretization. However, it is well known that BDF methods with accuracy higher than second-order can be unstable.

Table 2: Coefficients for the BDF methods.

| $b^{n+1}u^{n+1}+b^nu^n+\cdots=\Delta t f^{n+1}$ | | | | | |
|---|---|---|---|---|---|
| Order | $u^{n+1}$ | $u^n$ | $u^{n-1}$ | $u^{n-2}$ | $u^{n-3}$ |
| 1 | 1 | $-1$ | | | |
| 2 | $\frac{3}{2}$ | $-2$ | $\frac{1}{2}$ | | |
| 3 | $\frac{11}{6}$ | $-3$ | $\frac{3}{2}$ | $-\frac{1}{3}$ | |
| 4 | $\frac{25}{12}$ | $-4$ | 3 | $-\frac{4}{3}$ | $\frac{1}{4}$ |

A stability analysis of the BDF methods applied to the one-dimensional form of the linear advection equation is shown in Fig. 5. The linear advection equation is defined as

$$\frac{\partial u}{\partial t}+a\frac{\partial u}{\partial x}=0, \tag{3.39}$$

where $u$ is the unknown scalar and $a$ is the wave speed. The eigenvalues, $\lambda$, for a first-order upwind discretization applied to this system with uniform mesh spacing $\Delta x$ and periodic boundary conditions are also shown in Fig. 5. The BDF4 scheme is stable for a CFL, $\nu = a\Delta t/\Delta x$, up to 2. This condition may be relaxed for systems involving diffusion and relaxation processes since they tend to shift the eigenvalues further into the negative portion of the real $\lambda\Delta t$-plane. As such, the BDF methods are still suitable for the present application since Eq. (2.3) involves a system of advection-diffusion equations. No issues related to the stability of the BDF methods used in combination with the high-order CENO schemes were encountered in this research.

## 3.6 Computational algorithm

The overall proposed solution algorithm can be described as follows for time-dependent flows:

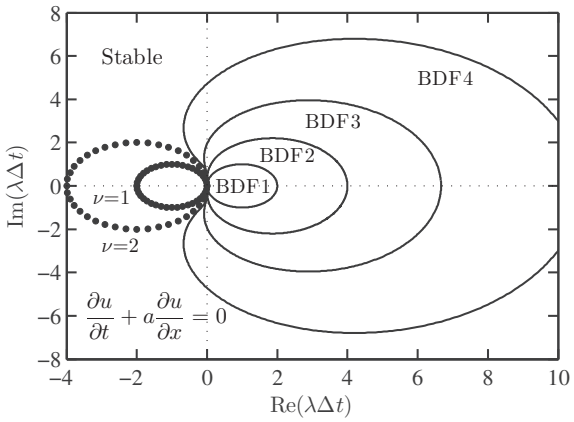



Figure 5: Stability diagram for the BDF methods applied to the linear advection equation. Lines enclose the unstable regions for each method; symbols are the eigenvalues for a first-order upwind spatial discretization with uniform mesh spacing $\Delta x$ and periodic boundaries. $\lambda$ represents the eigenvalues, $\Delta t$ is the time step size, $a$ is the wave speed, and $\nu = a\Delta t/\Delta x$ is the CFL number.

(1) Compute the initial solution, $\mathbf{W}^0$, set the initial time level $n=0$, and set the initial time $t=t_0$.

(2) Begin physical time loop, $n=n+1$.

(3) Set the initial Newton iteration level, $k=0$, choose the initial guess for the next time level, $\mathbf{W}^{(n,0)}=\mathbf{W}^{n-1}$, and compute the initial non-linear residual, $\mathbf{R}^{*(n,0)}=\mathbf{R}^*\left(\mathbf{W}^{(n,0)}\right)$, using Eq. (3.29).

(4) Begin Newton iteration loop, $k=k+1$.

(5) Compute the local preconditioner via BILU, i.e.,

$$M_j^{-1}=\text{BILU}\left[\frac{a}{\Delta t^n}\left(\frac{\partial \mathbf{U}}{\partial \mathbf{W}}\right)^{(n,k)}+\frac{\mathbf{\Gamma}^{(n,k)}}{\Delta\tau^k}+\left(\frac{\partial \mathbf{R}}{\partial \mathbf{W}}\right)^{(n,k)}\right],$$

and form the global preconditioner, $M^{-1}$, using Eq. (3.33).

(6) Solve Eq. (3.37) for $\Delta\mathbf{W}^{(n,k)}$ via preconditioned GMRES($m$).

(7) Update the solution, $\mathbf{W}^{(n,k+1)}=\mathbf{W}^{(n,k)}+\Delta\mathbf{W}^{(n,k)}$, and compute the new non-linear residual, $\mathbf{R}^{*(n,k+1)}=\mathbf{R}^*\left(\mathbf{W}^{(n,k+1)}\right)$.

(8) Check convergence. If $\|\mathbf{R}^*(\mathbf{W}^{(n,k+1)})\|\geq\epsilon\|\mathbf{R}^*(\mathbf{W}^{(n,0)})\|$, go to step (4).

(9) Finish time loop:

  (a) The solution at time level $n$ is $\mathbf{W}^n=\mathbf{W}^{(n,k+1)}$.

  (b) If $t<t_{\text{final}}$, go to step (2).

(10) End computation.

The same algorithm was used for steady flows, except the loop over physical time levels was removed and the temporal discretization coefficient, $a$, was set to zero. In either case, i.e., steady or unsteady, a high-order CENO reconstruction was performed before the non-linear residual, $\mathbf{R}^*$, was evaluated. This procedure is summarized as follows:

(1) For each variable in the set $\mathbf{W}$, and each computational cell, perform a $k$-exact reconstruction using the procedure outlined in Section 3.2.1.

   (a) Form the system defined by Eq. (3.17), adding additional constraint equations if the cell is adjacent to the boundary.
   (b) Solve the over-determined system given by Eq. (3.17) for the minimum-error solution via QR factorization or SVD. If this reconstruction is constrained at the boundaries, apply Gaussian elimination with full pivoting first to remove the constraints.

(2) Compute the smoothness indicator as described in Section 3.2.3 for each variable in each computational cell.

   (a) Evaluate $S$ using Eq. (3.22).
   (b) If $S > S_c$, flag the variable in a particular cell for limited linear reconstruction.

(3) Perform a limited linear reconstruction (described in Section 3.2.3) for each variable in each computational cell where the high-order reconstruction was flagged as not smooth.

# 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 both smooth and discontinuous function reconstructions as well as solutions for steady and unsteady viscous flows on three-dimensional unstructured mesh were obtained. All computations were performed on a high performance parallel cluster consisting of 3,780 Intel Xeon E5540 (2.53GHz) nodes with 16GB RAM per node. The cluster is connected with a high-speed InfiniBand switched fabric communications link.

An error analysis is performed whenever exact solutions are present. Accuracy is assessed based on the $L_1$ and $L_2$ norms of the error between the exact solution and the numerical solution. The $L_p$ norm of the error is evaluated over all cells, $i$,

$$L_p = \|\text{Error}\|_p = \left[ \frac{1}{\mathcal{V}_T} \sum_i \iiint_{V_i} \left| u_i^k(x,y,z) - u_{\text{exact}}(x,y,z) \right|^p \mathrm{d}V \right]^{1/p}, \tag{4.1}$$

where $\mathcal{V}_T$ is the total volume of the domain and $u_{\text{exact}}(x,y,z)$ is the exact solution. This integration is performed using an adaptive cubature algorithm developed by Cools et al.

[113, 114] for integrating functions over a collection of $N$-dimensional hyperrectangles and simplices.

### 4.1 Spherical cosine function

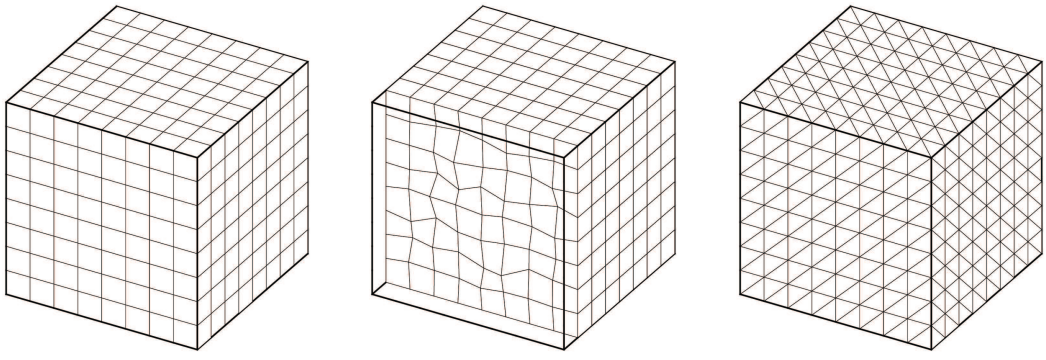
The first case considered is the reconstruction of a smooth spherical cosine function. The function, which is smooth in all directions, is described by

$$u(r) = 1 + \frac{1}{3}\cos(r), \tag{4.2}$$

where $r = 10\sqrt{x^2+y^2+z^2}$ is the radial position. The solution is computed on a unit cube centered about $(0.5, 0.5, 0.5)$ using grids composed of tetrahedral, Cartesian, and irregular hexahedral cells with varying levels of resolution. Coarse meshes of the three types used are illustrated in Fig. 6. The irregular hexahedral meshes were generated by randomly perturbing the internal nodes of an initial Cartesian mesh.

The results for the unlimited $k$-exact reconstruction of the three-dimensional spherical cosine function performed on a coarse mesh with 3,072 tetrahedral elements is illustrated in Fig. 7(a). As the order of the piecewise polynomial interpolant is increased from $k$=0 to $k$=3, the reconstructed solution rapidly approaches the exact solution. There is almost no visible difference between the exact solution and the reconstructed solution for $k$=4 (not shown in figure).

An analysis of the $L_2$ norm of the error in the numerical solution as the tetrahedral mesh resolution is increased, illustrated in Fig. 7(b) for various values of $k$, confirms that $k$-exact reconstruction of a smooth function yields an order of accuracy equal to $k$+1. Similar results for the error analysis are observed for meshes composed of Cartesian, Fig. 7(c), and irregular hexahedrals, Fig. 7(d).



(a) Cartesian ($8\times8\times8$ cells) (b) Hexahedra ($8\times8\times8$ cells) (c) Tetrahedra (3,072 cells)

Figure 6: Sample computational meshes used for function reconstructions and the study of Taylor-Green vortex decay.

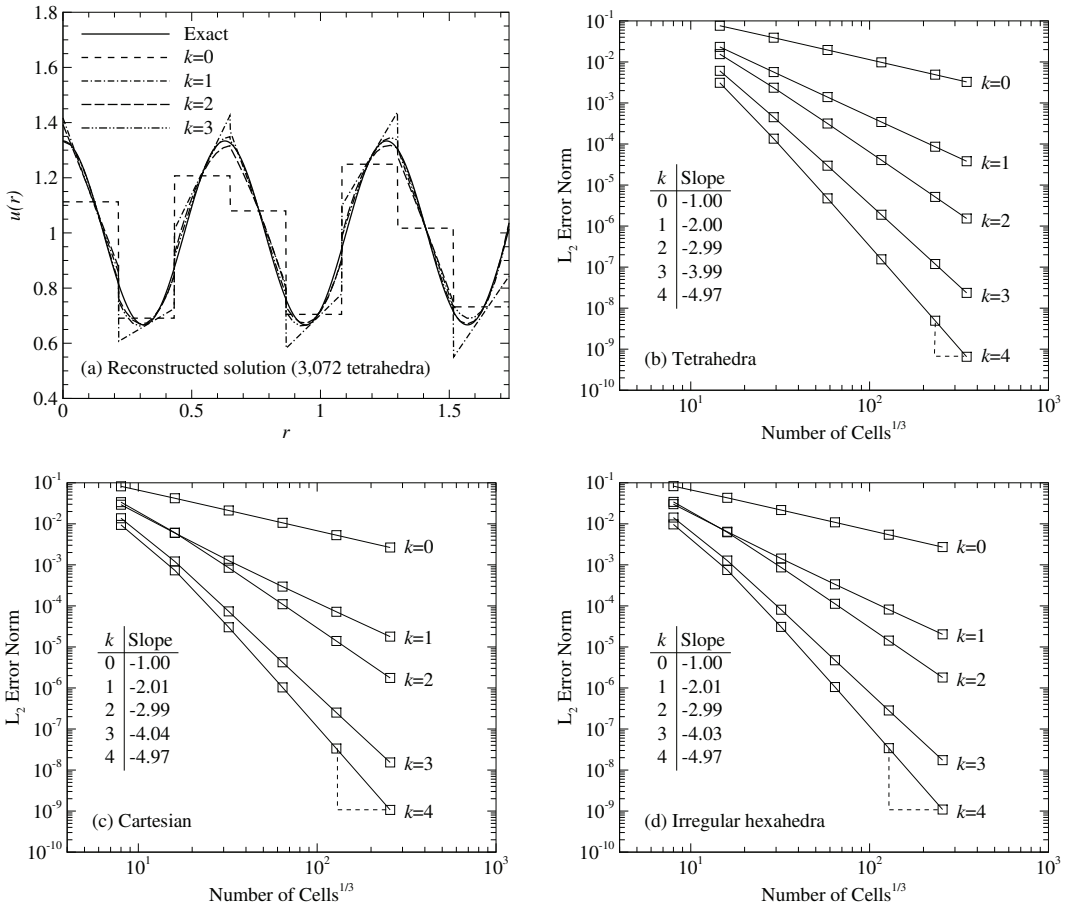



Figure 7: Results for $k$-exact reconstruction of the spherical cosine function. (a) Unlimited reconstructed solution along the diagonal from $(0,0,0)$ to $(1,1,1)$ obtained on a mesh with 3,072 tetrahedral cells; $L_2$ norm of reconstruction error for (b) tetrahedral, (c) Cartesian, and (d) irregular hexahedral shaped cells.

## 4.2 Abgrall's function

The Abgrall function [115] possesses a number of solution discontinuities which test a high-order spatial discretization's ability to maintain monotonicity. Reconstructions of this function using the proposed high-order CENO algorithm for unstructured meshes are obtained to ensure the effectiveness of the smoothness indicator defined in Eq. (3.22). Even though the performance of this formulation for $S$ was already verified using the Abgrall function on structured mesh [59], it has not been fully evaluated for unstructured mesh. McDonald et al. [63] only obtained preliminary results for this function on tetrahedral meshes. The Abgrall function is defined as

$$u(x,y)=\begin{cases} f(x-\cot\sqrt{\pi/2}\,y) & \text{if } x\leq\cos(\pi y)/2, \\ f(x+\cot\sqrt{\pi/2}\,y)+\cos(2\pi y) & \text{if } x>\cos(\pi y)/2, \end{cases} \tag{4.3}$$

where

$$f(r)=\begin{cases} -r\cdot\sin\left(3\pi r^2/2\right) & \text{if } r\leq -1/3, \\ |\sin(2\pi r)| & \text{if } |r|<1/3, \\ 2r-1+\sin(3\pi r)/6 & \text{if } r\geq 1/3. \end{cases} \tag{4.4}$$

Here, it is applied in three dimensions to a cube with length 2 by extruding the two-dimensional function along the $z$ axis. Similar computational meshes as those used in the previous section were employed (Fig. 6).

The reconstructed solution obtained using the proposed high-order CENO algorithm for unstructured meshes with $k=4$ is compared with the exact solution in Figs. 8(a) and 8(b). The CENO scheme is able to accurately reconstruct the Abgrall function even

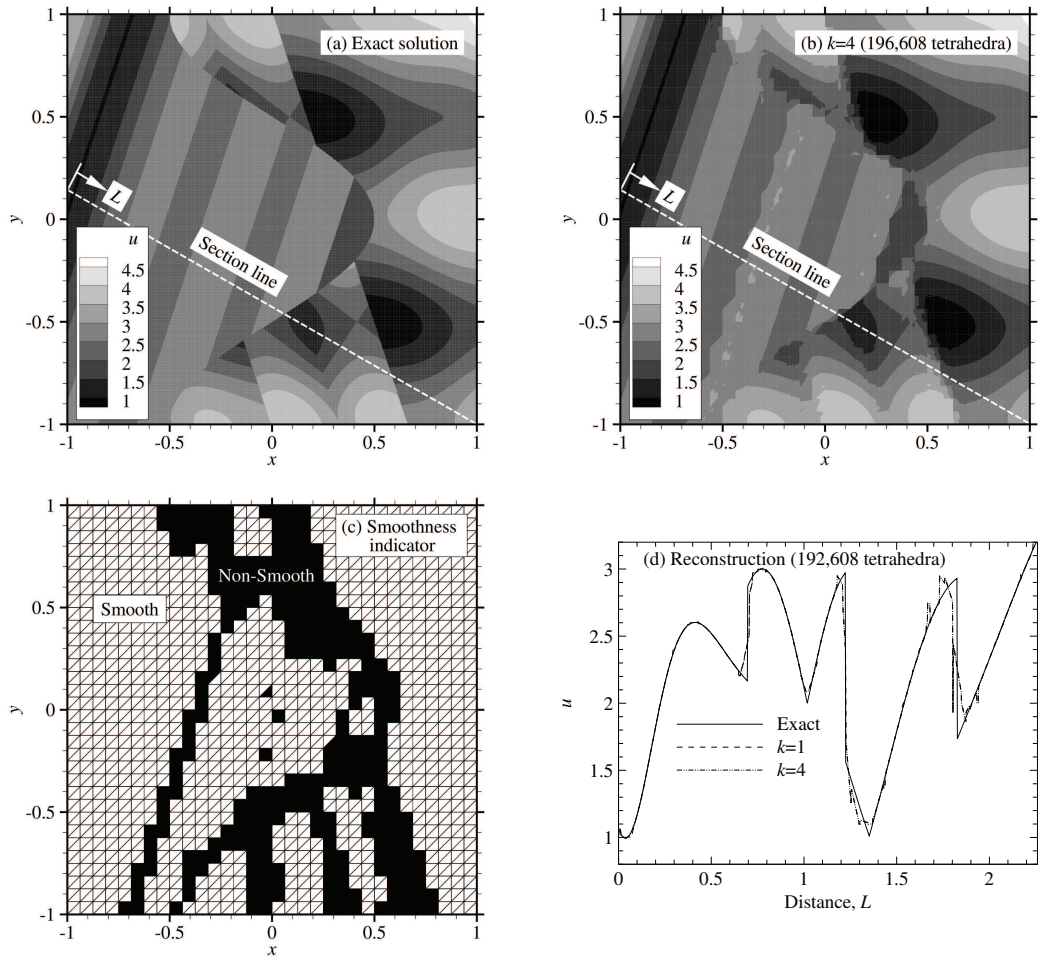



Figure 8: Results for CENO reconstruction of Abgrall's function obtained using a three-dimensional mesh with 196,608 tetrahedral cells. (a) Exact solution; (b) numerical solution and (c) computed smoothness indicator for $k$=4; (d) reconstructed solution along the line identified in (a).

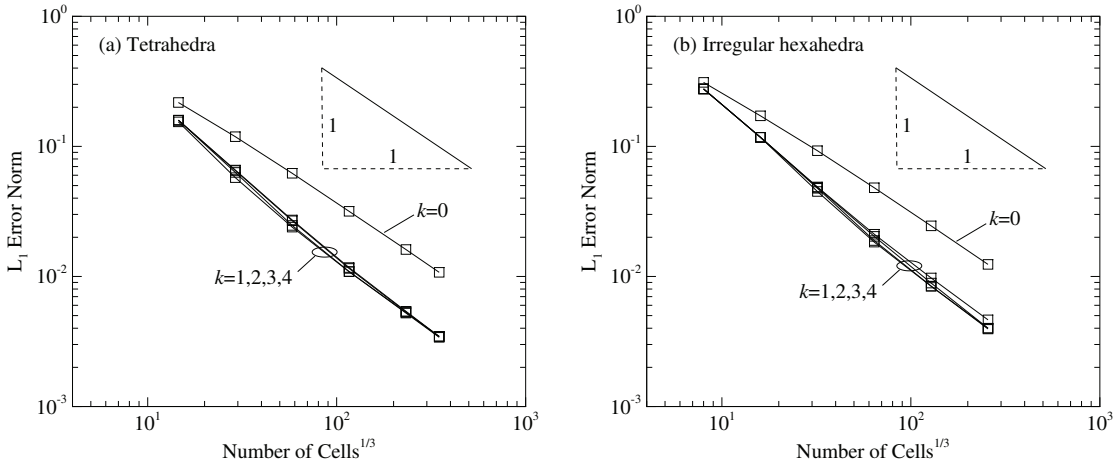



Figure 9: Effect of grid resolution on solution accuracy for CENO reconstruction of the Abgrall function. Accuracy is measured using the $L_1$ norm of the error.

on a relatively coarse mesh consisting of only 196,608 tetrahedral cells without producing spurious oscillations. For this mesh, there is only approximately 58 cells in each direction with an average edge length of 0.03. The smoothness indicator, illustrated in Fig. 8(c), correctly identified the discontinuities in both $u$ and $\partial u/\partial x_i$.

The reconstructed solution obtained with $k$=0 to $k$=4 and 196,608 tetrahedral cells along a line is illustrated in Fig. 8(d). The proposed CENO scheme is able to ensure oscillation-free solutions despite the large discontinuities observed. The small spikes observed in the numerical solution are a result of post-processing and occur when the line intersects the faces between cells at acute angles.

The effect of mesh resolution on the $L_1$ norm of the solution error is shown in Fig. 9 for tetrahedral and irregular hexahedral cells. A large improvement in the error is achieved by increasing $k$ from 0 to 1. This improvement becomes less pronounced as $k$ is increased further to 4 since a large portion of the domain possesses discontinuous features. An order of accuracy of 1 is expected for all values of $k$ due to discontinuities in the function and the use of a limited piecewise linear reconstruction in these regions. Nonetheless, the hybrid reconstruction procedure maintains the monotonicity of the higher-order solutions. This highlights the robustness of the proposed scheme.

### 4.3 Isothermal flow over a flat plate

Numerical results for the steady laminar flow over a flat plate were obtained using the second- and fourth-order CENO finite volume scheme and compared with the Blasius analytic solution. The computational domain and boundary conditions are illustrated in Fig. 11(a). A rectangular domain was used with dimensions $4L\times 2L\times L/4$. The far-field and upstream of the plate are modeled by a reflection/slip condition while the presence

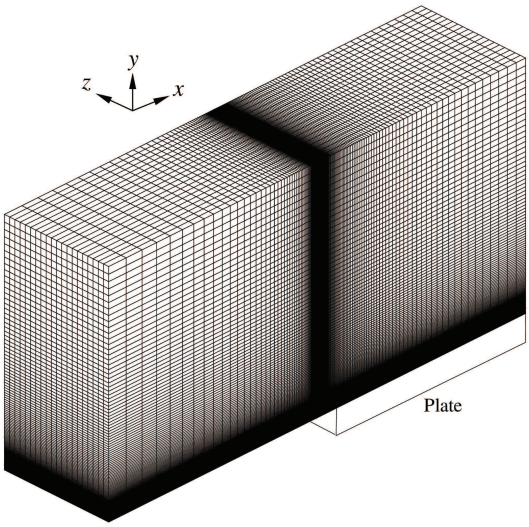



Figure 10: Computational mesh used to study steady laminar flow over a flat plate. The mesh consists of 512,000 Cartesian cells clustered near the leading edge of the plate and near the plate surface.

of the plate is simulated using a no-slip condition. A uniform velocity profile is specified at the inlet and a zero-gradient condition on the fluid velocity is applied at the outlet. Pressure is held fixed and equal to 0 at the outlet. The free-stream Mach number and Reynolds number based on the plate length, $L$, are $M_\infty$=0.2 and $Re_L$=10,000, respectively. Solutions were obtained on a stretched mesh, illustrated in Fig. 10, with 512,000 Cartesian cells (160×160×20) clustered near the leading edge of the plate and near the plate surface. The CFL was initially equal to 100 for start-up and then increased to 400 using the SER procedure outlined in Section 3.5.2.

The predicted non-dimensional boundary layer profiles, $u/U_\infty$ and $\theta = v\sqrt{Re_x}/U_\infty$, at $Re_x = 8{,}000$ is compared with the Blasius analytic solution in Fig. 11(c). Here, $\nu = \mu/\rho$ is the kinematic viscosity and $U_\infty$ is the free-stream velocity. The non-dimensional distance from the plate, $\eta$, is defined as $y\sqrt{Re_x}/x$. Excellent agreement is obtained with the analytic solution.

The convergence histories obtained using the second- and fourth-order schemes are provided in Fig. 11(b). Excellent convergence characteristics are displayed by the parallel Newton-Krylov-Schwarz solution algorithm, requiring only 121 Newton iterations and 1269 evaluations of Eq. (3.29) to reduce the equation residuals by approximately eight orders of magnitude. Note that a non-zero residual is observed for the enthalpy equation even though the flow is isothermal. This is a result of introducing a pseudo-time derivative with respect to pressure to Eq. (2.1c). The fourth-order CENO scheme required more Newton iterations and evaluations of **R** than the second-order scheme, 293 and 3468, respectively, to obtain the same order of reduction. This is a result of the added stiffness introduced by the higher degree interpolation polynomials. Nonetheless, the proposed Newton algorithm does an excellent job of coping with this added stiffness.

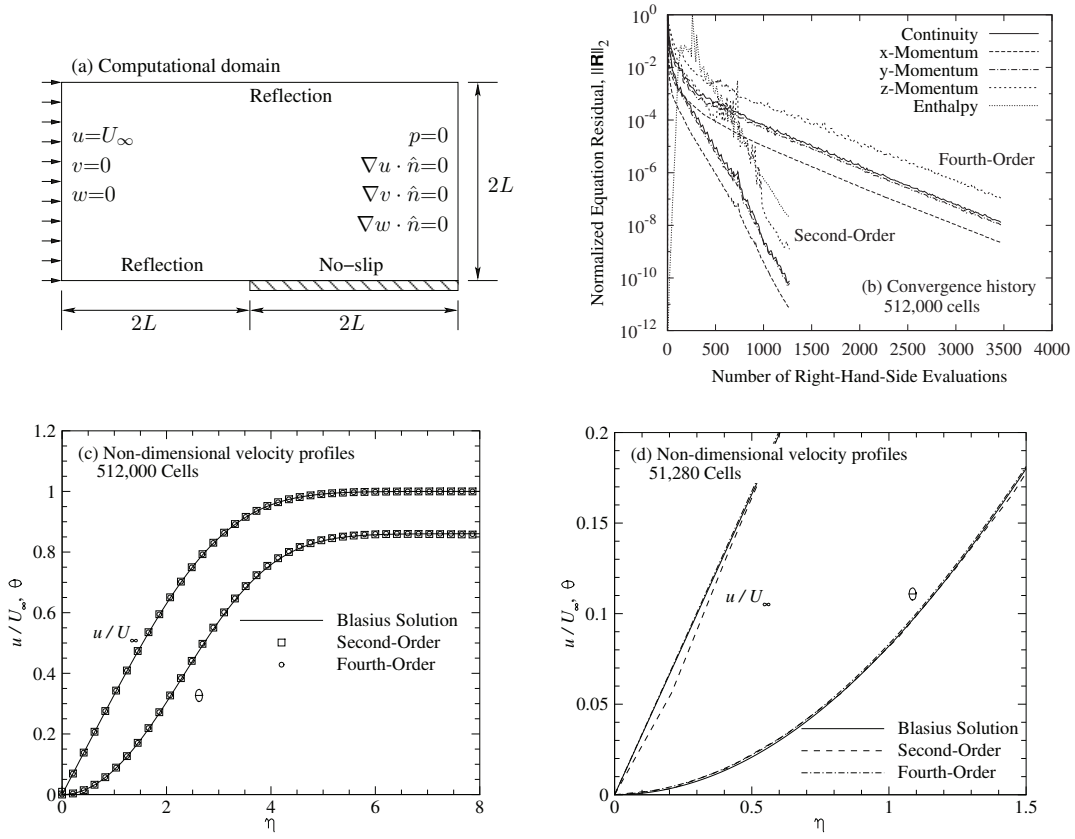



Figure 11: Numerical results for laminar flow over a flat plate. (a) Computational domain and boundary conditions; (b) convergence histories for the second- and fourth-order accurate solutions; and Blasius and numerical solutions for the non-dimensionalized boundary layer profile at $\mathrm{Re}_x$=8,000 obtained on a mesh with (c) 512,000 and (d) 51,280 computational cells.

As mentioned in Section 3.2.3, the smoothness indicator provides numerical stability early in the solution process when the initial/intermediate solution is far from the final result. However, in this particular case, convergence of the fourth-order CENO scheme initially stalled after four orders of reduction in the residual. To alleviate this, the smoothness indicator was simply frozen at this point and the convergence history displayed in Fig. 11(b) was obtained. The value of the cutoff, $S_c$, had little affect on the overall convergence history, provided it was large enough to ensure non-oscillatory solutions, i.e., $S_c \gtrsim 500$.

Both sets of numerical results — i.e., second- and fourth-order — for the relatively fine mesh with 512,000 cells are nearly indistinguishable from the Blasius solution. To highlight the advantages of applying the proposed high-order CENO finite-volume discretization to this particular problem, second- and fourth-order solutions were also obtained on a coarse mesh with only 51,280 computational cells. The non-dimensional

boundary layer profiles at $\mathrm{Re}_x = 8{,}000$, predicted using this coarse mesh, are compared with the Blasius analytic solution in Fig. 11(d). Although the second-order solution fails to accurately predict $u/U_\infty$ and $\theta$ near the plate, the fourth-order CENO solution is still in excellent agreement with the Blasius solution. This shows the ability of the fourth-order CENO discretization to retain high accuracy even on coarse meshes.

### 4.4 Taylor-Green vortex decay

The Taylor-Green vortex solution [116] is used for testing and validation of the spatial and temporal accuracy of the proposed solution algorithm for the incompressible Navier-Stokes equations. The Taylor-Green vortex solution is given by

$$u = \sin(x)\cos(y)\,e^{-2\nu t}, \tag{4.5}$$

$$v = -\cos(x)\sin(y)\,e^{-2\nu t}, \tag{4.6}$$

$$p = \frac{\rho}{4}[\cos(2x)+\cos(2y)]e^{-4\nu t} \tag{4.7}$$

over the domain $0 \leq x, y \leq 2\pi$, where $\nu$ is the kinematic viscosity of the fluid. Although the solution is two-dimensional, it was extended to three dimensions by extrapolating the solution in the $z$-direction and setting $w$=0. Solutions were obtained on a periodic cube domain with length $2\pi$ using varying grid resolutions with either regular hexahedral (Cartesian) or tetrahedral cells. Sample computational meshes are illustrated in Figs. 6(a) and 6(c). The time-step size was selected to maintain a constant $\mathrm{CFL}_{\mathrm{phys}}$ such that $\Delta t/\Delta x = 0.02$. The kinematic viscosity, $\nu = \mu/\rho$, is set to 1.0 $\mathrm{m/s^2}$ and $\rho$=1 $\mathrm{kg/m^3}$.

The initial conditions for $u$ at time $t$=0 is illustrated in Fig. 12(a). As time progresses, the magnitude of the velocity decays exponentially through the effects of viscous damping. This decay is apparent in Fig. 12(b) which shows the exact solution along $y$=$z$=$\pi$ at $t$=0.4 s, which is close to the decay half-life. The predicted solutions obtained using the CENO finite-volume scheme with different orders of accuracy on a mesh with 196,608 tetrahedra are also provided in Fig. 12(b). The first-order solution, which uses a piecewise-constant representation to evaluate the inviscid fluxes and a piecewise-linear representation for the viscous fluxes, greatly over-predicts the exponential decay of the velocity field because of the scheme's dissipative nature. A significant improvement in accuracy is observed when the second-order scheme is used. However, the second-order scheme still cannot properly represent the sinusoidal form of the solution using a piecewise-linear representation. Excellent agreement is obtained between the predicted solutions obtained using the higher-order schemes (third- and fourth-order) and the exact solution.

An error analysis was performed to verify the correct spatial order-of-accuracy and is shown in Fig. 13 for both Cartesian, Fig. 13(a), and tetrahedral, Fig. 13(b), meshes. For both types of meshes, the first-order scheme yields an order-of-accuracy of 0.99, which is close to the expected value of 1. The second-order scheme provides an order-of-accuracy equal to approximately 2 on both Cartesian and tetrahedral meshes.

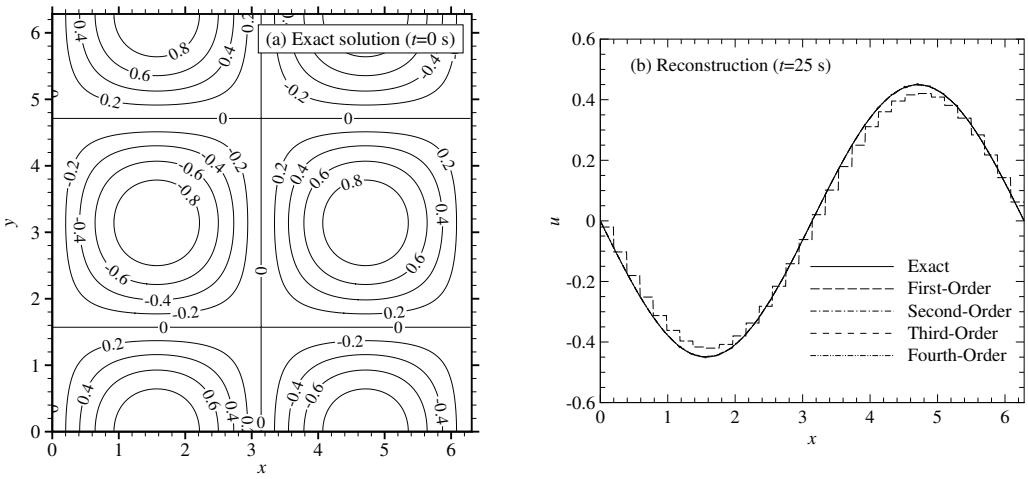



Figure 12: Results for CENO reconstruction of the Taylor-Green vortex decay. (a) Exact solution for $u$ at $t$=0 s and (b) reconstructed solution for $u$ along $y$=$z$=$\pi$ m at $t$=0.4 s obtained on a mesh composed of 196,608 tetrahedral cells.

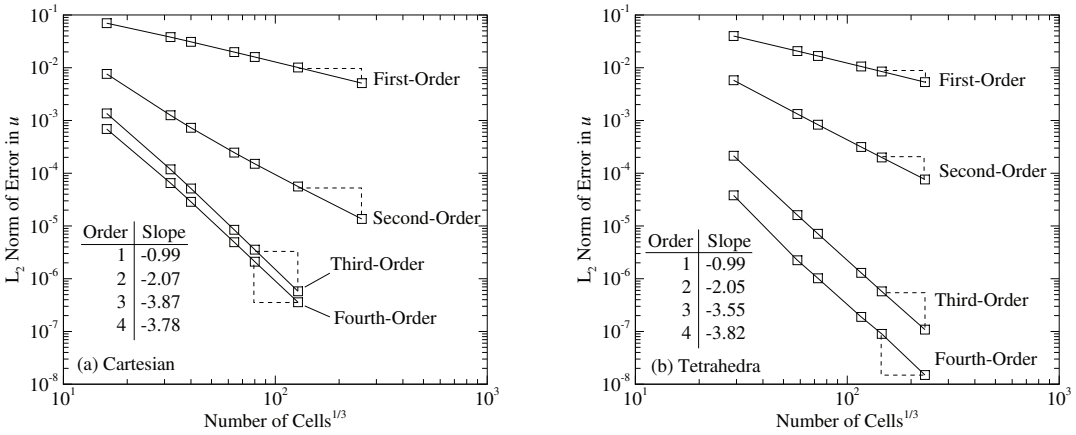



Figure 13: Effect of mesh size on the $L_2$-norm of the error in $u$ for the Taylor-Green vortex decay.

When the third- and fourth-order schemes are applied to uniform Cartesian mesh, a similar trend regarding the magnitude of the slopes of the lines corresponding to error vs mesh-size is observed in Fig. 13(a). The third- and fourth-order schemes achieve an order of accuracy slightly less than 4. However, it is apparent in Fig. 13(a) that the finest meshes used in this study has not yet reached the asymptotic regime since the slopes of their corresponding lines is still decreasing. Both schemes are expected to yield the same order of accuracy as a result of the cancellation of truncation errors on uniform Cartesian mesh.

With the tetrahedral meshes, Fig. 13(b), an order of accuracy close to 4 is observed for both the third- and fourth-order schemes. Cancellation of truncation errors are also

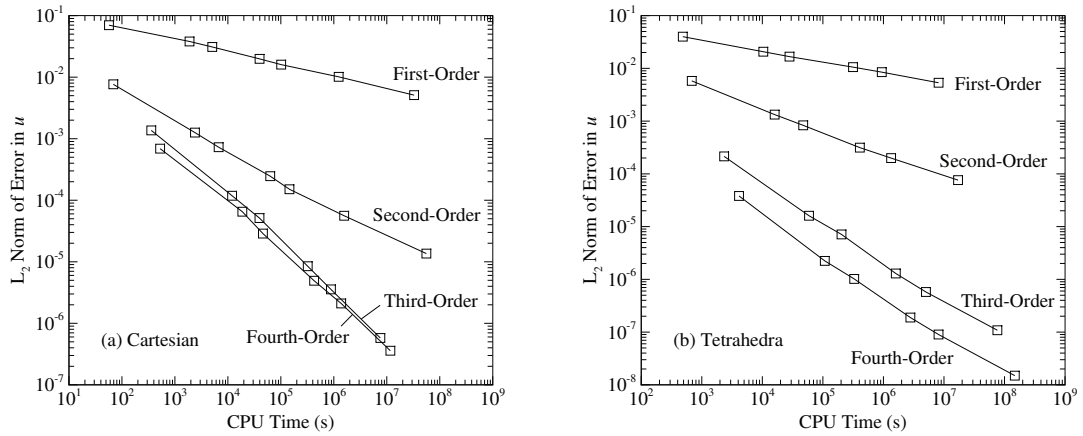



Figure 14: Required CPU time for a desired level of accuracy for the Taylor-Green vortex decay.

expected for tetrahedral mesh, since the they were created by subdividing a uniform Cartesian mesh. Similar to the findings obtained when using Cartesian mesh, the results for the high-order schemes obtained on tetrahedral mesh suggest that the largest mesh used in this study may not be sufficiently fine to reach the asymptotic regime. Further study is warranted to clarify these findings. It is certainly clear that the observed convergence rates for all orders either matches the theoretical expectations for this unsteady problem.

The computational efficiency is assessed in Fig. 14 which illustrates the error as a function of CPU time for each scheme. As expected, lower levels of error have a higher associated computational costs for all the schemes tested here. If the desired level of error is large, higher than $10^{-1}$, the first- and second-order solution methods offer the lowest computational cost. However, the third- and fourth-order schemes become the most cost effective in terms of accuracy vs computational time when higher levels of accuracy are desired. For example, for an $L_2$ error of approximately $10^{-5}$ on a uniform Cartesian mesh, the third- and fourth-order schemes are 285 and 450 times faster than the second-order scheme, respectively. This level of error is achieved by both the third- and fourth-order schemes using a mesh 90 and 140 times smaller, respectively, than the mesh required by the second-order scheme. Similar results were obtained when using tetrahedral mesh. An $L_2$ error of approximately $10^{-5}$ would be obtained with 6,800 and 45,000 times less computational effort with the third- and fourth-order schemes when compared to the second-order scheme, respectively. The required mesh sizes to obtain this error are 600 (third-order) and 2,600 (fourth-order) times smaller than the mesh size required by the second-order scheme. It is apparent that for unstructured meshes of both Cartesian and tetrahedral computational cells, the higher-order schemes (third-order accuracy and higher) offer a superior advantage over the low-order ones in terms of accuracy and computational cost.

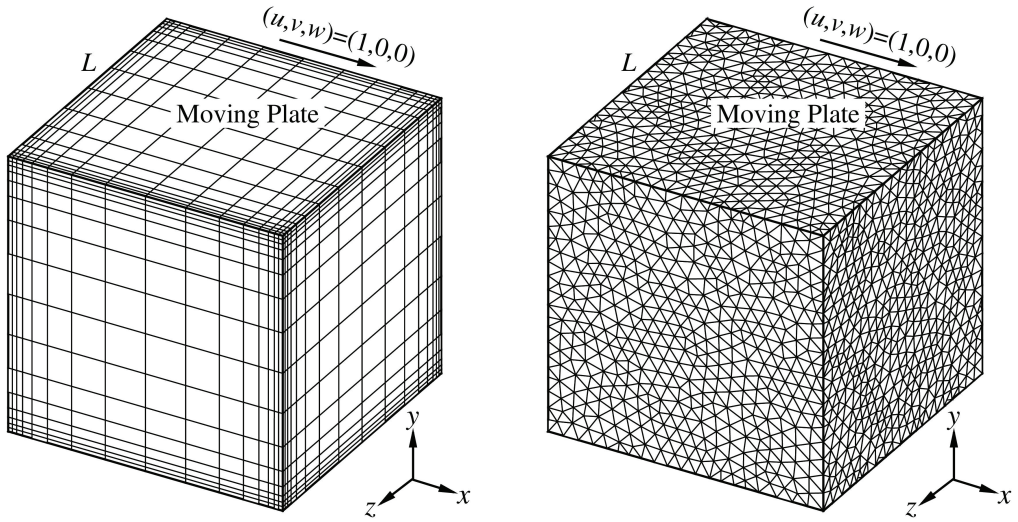
### 4.5 Lid driven cavity flow

The lid-driven cavity flow is a standard benchmark test for laminar viscous fluid flow. It demonstrates complex flow physics in a simple geometry with well-defined boundary conditions. For this particular test case, a simple three-dimensional box is filled with fluid. All walls are stationary, except for the lid, which is driven at a prescribed velocity. The resulting flow pattern inside the box contains eddies, secondary flows, and can even display turbulent behavior if the plate velocity is high enough [117].

These flows are characterized by their Reynolds number, which is defined as $\mathrm{Re}=\rho\|\vec{v}_{\mathrm{lid}}\|L/\mu$, where $\vec{v}_{\mathrm{lid}}$ is the velocity of the lid and $L$ is the length scale of the box. A cube-shaped cavity was considered, and computations were carried out for a Reynolds number of 1000. In this case, $\rho$=1 kg/m$^3$, $\mu=10^{-3}$ Pa·s, the velocity of the plate is $\vec{v}_{\mathrm{lid}}=(1,0,0)$ m/s, and the sides of the cube are of length $L$=1 m. The domain was meshed using Cartesian and tetrahedral cells, and solutions were obtained using a series of successively refined meshes. Non-uniform mesh spacing was used for both types of meshes, with the computational cells clustered near the cube walls. The geometry and computational meshes are illustrated in Fig. 15.

The Cartesian mesh sizes were 16x16x16, 32x32x32, 64x64x64, and 128x128x128. Three different tetrahedral meshes were used. They consisted of 31,574, 228,107, and 970,567 tetrahedra. Fixed wall boundary conditions were used everywhere, with no condition on pressure. That is,

$$\vec{v}(x,y,z)=\begin{cases}\vec{v}_{\mathrm{lid}} & \text{if } y=1,\\ \vec{0} & \text{if } x=0,\ x=1,\ y=0,\ z=0, \text{ or } z=1.\end{cases} \tag{4.8}$$




(a) Cartesian ($16\times16\times16$ cells) (b) Tetrahedra (31,574 cells)

Figure 15: Geometry and sample computational meshes used for the study of lid-driven cavity flow.

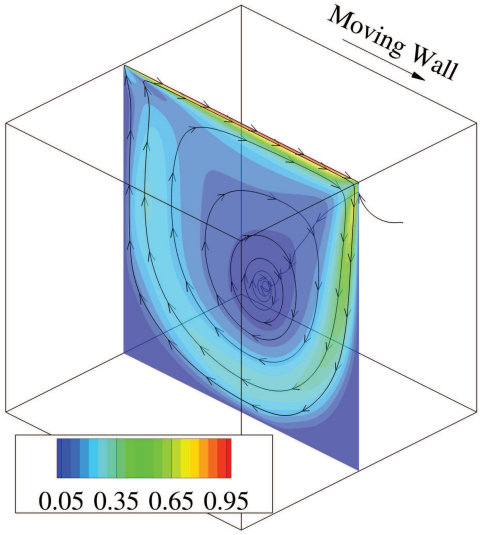



Figure 16: Contours of velocity magnitude, $|\vec{v}|$, and streamlines along $z=0.5$m for the lid-driven cavity (Re$=$ 1000). Units in m/s.

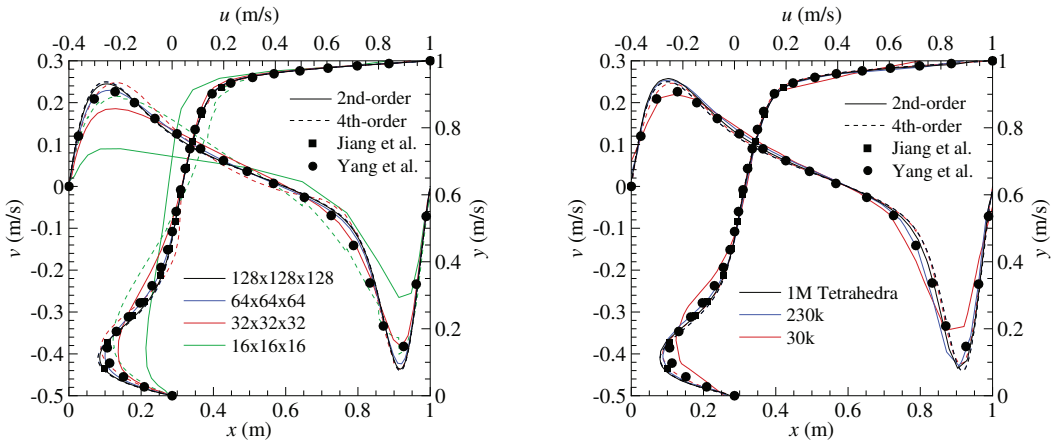
All solutions were obtained using the inexact Newton's method described in Section 3.5.1 for steady flows. Full Newton iteration was used without any implicit Euler startup, starting from an initial solution given by $\vec{v}(\vec{x})=\vec{0}$. Newton iteration was stopped when the non-linear residual was reduced by 5 orders of magnitude ($\epsilon=10^{-5}$). Reducing it any further had no noticeable effect on the predicted solution.

Numerical results for the lid-driven cavity were obtained using the second- and fourth-order CENO finite volume scheme and compared with the previous numerical predictions of Jiang et al. [118] and Yang et al. [119]. Fig. 16 illustrates the velocity magnitude along the plane defined by $z=0.5$m. The complex recirculating flow pattern is also depicted in the figure.

In Fig. 17, the $x$- and $y$-components of velocity are compared with the previous numerical predictions of Jiang et al. [118] and Yang et al. [119] along $y=z=0.5$ and $x=z=0.5$ for different mesh sizes. Jiang et al.'s solution was obtained using an almost uniform mesh with $50\times52\times25$ hexahedral elements to model half of the domain, while Yang et al. modelled the entire computational domain using a stretched mesh with $33\times33\times33$ hexahedral elements.

With both types of meshes considered here, the second-order solutions at the coarsest resolution do not agree well with the numerical results of Jiang et al. [118] and Yang et al. [119], which is expected since these benchmark solutions were obtained with considerably finer meshes. The second-order predictions improve greatly as the mesh is refined; the results using $64\times64\times64$ Cartesian elements and those using 228,107 tetrahedra agree well with prior numerical results.

The fourth-order scheme offers a significant improvement in agreement as it requires




(a) Cartesian (b) Tetrahedra

Figure 17: Predicted velocity profiles along centerlines ($y=z=0.5$ and $x=z=0.5$) for the lid-driven cavity (Re$=1000$).

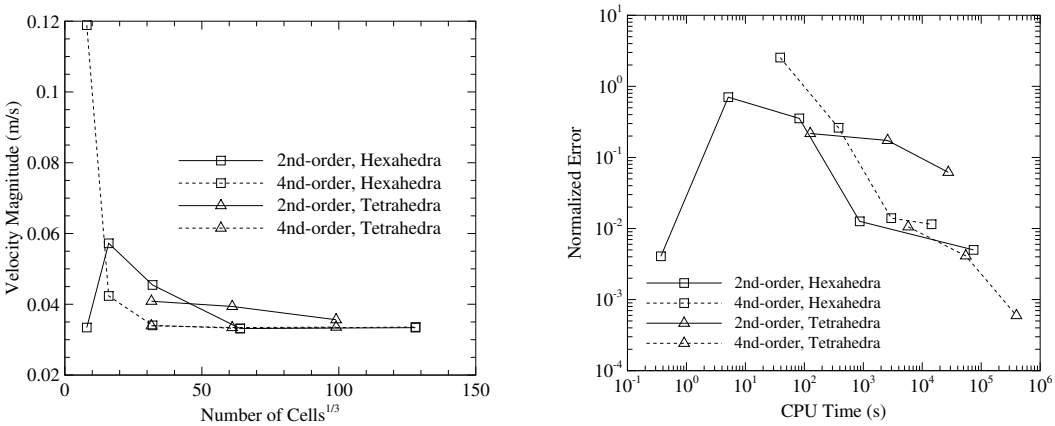



Figure 18: The (a) predictions for the magnitude of velocity at $x=y=z=0.5$m and (b) the required CPU time for a desired level of estimated accuracy for the lid-driven cavity (Re$=1000$). The error was estimated using the results obtained with $128\times128\times128$ Cartesian elements as the reference solution.

fewer computational cells to resolve the flow than the second-order scheme. Even using the coarsest meshes, i.e., the meshes with $16\times16\times16$ Cartesian and 31,574 tetrahedral elements, the fourth-order solutions closely match the prior numerical results of Jiang et al. and Yang et al.. And these prior solutions were obtained using much finer meshes. In all cases, the predictions converge asymptotically as the meshes are refined.

To assess the accuracy of the numerical predictions, despite the fact that an exact solution is not known, the predicted magnitude of velocity at $x=y=z=0.5$m was compared in Fig. 18(a). This location was chosen arbitrarily; any reference point could have been chosen. For both discretization schemes and types of meshes, the solutions all converge

asymptotically towards a value of approximately 0.034 m/s. It is clear that the high-order solutions converge at a much faster rate than the low-order ones.

The error in predicted velocity magnitude was plotted as a function of CPU time in Fig. 18(b). Since an exact solution is not known, the error was computed using the fourth-order results obtained on the mesh with $128\times128\times128$ Cartesian elements as the reference solution. As demonstrated for the Taylor-Green vortex decay in the previous section, the high-order scheme is more efficient for lower desired levels of error. The fourth-order scheme using tetrahedra was the most efficient for error levels below $4\times10^{-3}$.

## 5 Conclusions

A parallel, implicit, high-order, finite-volume scheme has been developed for the numerical solution of the incompressible Navier-Stokes equations on unstructured meshes using the pseudo-compressibility approach. The scheme is a high-order variant of cell-centered, Godunov-type, finite-volume methods and uses the hybrid CENO reconstruction method recently proposed by Ivan and Groth [59–62].

A validation of the CENO reconstruction procedure and finite-volume scheme has been performed for a variety of functions and idealized flow problems. For smooth functions, up to fourth-order reconstruction accuracy was achieved on general unstructured hexahedral and tetrahedral mesh. Robust and monotone reconstructions were maintained throughout this study even for functions with discontinuities. This demonstrates the effectiveness of the smoothness indicator, which lowers the order of accuracy of the scheme near areas of discontinuities or under-resolved solution content in order to maintain monotonicity.

Validation of predicted solutions to the three-dimensional Navier-Stokes equations was also performed. Numerical results obtained for the unsteady decay of Taylor-Green vortices confirmed that high-order solutions for smooth viscous incompressible fluids are achieved with the proposed finite-volume scheme, even on unstructured mesh. The higher-order schemes also displayed excellent computational efficiency in terms of both accuracy and CPU time. For an $L_2$ error of approximately $10^{-5}$, the third- and fourth-order schemes are up to 6,800 and 45,000 times faster than the second-order scheme, respectively. This level of error was achieved with the third- and fourth-order schemes using a tetrahedral mesh that was 600 and 2,600 times smaller, respectively, than the mesh required by the second-order scheme.

In general, the proposed scheme is able to accurately represent solutions with smooth extrema while robustly handling under-resolved and/or non-smooth solution content. Combined with the parallel Newton-Krylov-Schwarz solution algorithm, which exhibited excellent convergence characteristics, the proposed finite-volume scheme is able to obtain fast and accurate solutions to the incompressible Navier-Stokes equations. Future work consists of further development and validation of the proposed Newton-Krylov-

Schwarz CENO algorithm for three-dimensional unstructured meshes. This includes applying the pseudo-compressibility approach to more complex flows, such as the large-eddy simulation of turbulent flames, and incorporating a multi-block adaptive mesh refinement (AMR) algorithm [120–123]. The applicability of CENO to AMR and the substantial benefits in terms of accuracy and computational savings have already been demonstrated for body-fitted multi-block meshes [59, 60].

## Acknowledgments

Financial support for the research described herein was provided by MITACS (Mathematics of Information Technology and Complex Systems) Network, part of the Networks of Centres of Excellence (NCE) program funded by the Canadian government, as well as by Rolls-Royce Canada Inc. This funding is gratefully acknowledged with many thanks. Computational resources for performing all of the calculations reported herein were provided by the SciNet High Performance Computing Consortium at the University of Toronto and Compute/Calcul Canada through funding from the Canada Foundation for Innovation (CFI) and the Province of Ontario, Canada.

### References

[1] D. Drikakis and W. Rider. High-Resolution Methods for Incompressible and Low-Speed Flows. Springer-Verlag, Berlin, Heidelberg, 2005.
[2] D. Kwak, C. Kiris, and J. Housman. Implicit methods for viscous incompressible flows. Comput. Fluids, 41(1):51–64, 2011.
[3] S.V. Patankar and D.B. Spalding. A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. Int. J. Heat Mass Transfer, 15(10):1787–1806, 1972.
[4] C.M. Rhie and W.L. Chow. Numerical study of the turbulent flow past an airfoil with trailing edge separation. AIAA J., 21(11):1525–1532, 1983.
[5] N.N. Yanenko. The Method of Fractional Steps. Springer-Verlag, Berlin, 1971.
[6] J. Kim and P. Moin. Application of a fractional-step method to incompressible navier-stokes equations. J. Comput. Phys., 59(2):308–323, 1985.
[7] H. Fasel. Investigation of the stability of boundary layers by a finite-difference model of the Navier-Stokes equations. J. Fluid Mech., 78(2):355–383, 1976.
[8] S.C.R. Dennis, D.B. Ingham, and R.N. Cook. Finite-difference methods for calculating steady incompressible flows in three dimensions. J. Comput. Phys., 33(3):325–339, 1979.
[9] A.J. Chorin. A numerical method for solving incompressible viscous flow problems. J. Comput. Phys., 2(1):12–26, 1967.
[10] D. Drikakis, P. Govatsos, and D. Papantonis. A characteristic-based method for incompressible flows. Int. J. Numer. Meth. Fluids, 19(8):667–685, 1994.
[11] E. Shapiro and D. Drikakis. Non-conservative and conservative formulations of characteristics-based numerical reconstructions for incompressible flows. Int. J. Numer. Meth. Engin., 66(9):1466–1482, 2006.

[12] B. Van Leer, W.T. Lee, and P.L. Roe. Characteristic time-stepping or local preconditioning of the euler equations. AIAA Paper 91-1552, 1991.
[13] Y.H. Choi and C.L. Merkle. The application of preconditioning in viscous flows. J. Comput. Phys., 105(2):207–223, 1993.
[14] E. Turkel. Review of preconditioning methods for fluid dynamics. Appl. Numer. Math., 12 (1-3):257–284, 1993.
[15] J.M. Weiss and W.A. Smith. Preconditioning applied to variable and constant density flows. AIAA J., 33(11):2050–2057, 1995.
[16] E. Turkel, R. Radespiel, and N. Kroll. Assessment of preconditioning methods for multidimensional aerodynamics. Comput. Fluids, 26(6):613–634, 1997.
[17] S. Venkateswaran, M. Deshpande, and C.L. Merkle. The application of preconditioning to reacting flow computations. 12th AIAA Computational Fluid Dynamics Conference, San Diego, CA, Jun. 19-22 1995. AIAA paper 1995-1673.
[18] J.L. Steger and P. Kutler. Implicit finite-difference procedures for the computation of vortex wakes. AIAA J., 15(4):581–590, 1977.
[19] D. Kwak, J.L.C. Chang, S.P. Shanks, and S.R. Chakravarthy. Three-dimensional incompressible Navier-Stokes flow solver using primitive variables. AIAA J., 24(3):390–396, 1986.
[20] E. Turkel. Preconditioned methods for solving the incompressible and low speed compressible equations. J. Comput. Phys., 72(2):277–298, 1987.
[21] S.E. Rogers, D. Kwak, and U. Kaul. On the accuracy of the pseudocompressibility method in solving the incompressible Navier-Stokes equations. Appl. Math. Model., 11(1):35–44, 1987a.
[22] C. Kiris, D. Kwak, S. Rogers, and I.D. Chang. Computational approach for probing the flow through artificial heart devices. J. Biomed. Eng., 119(4):452–460, 1997.
[23] S.E. Rogers and D. Kwak. Upwind differencing scheme for the time-accurate incompressible Navier-Stokes equations. AIAA J., 28(2):253–262, 1990.
[24] S.E. Rogers, D. Kwak, and C. Kiris. Steady and unsteady solutions of the incompressible Navier-Stokes equations. AIAA J., 29(4):603–610, 1991.
[25] Z. Qian and J. Zhang. Implicit preconditioned high-order compact scheme for the simulation of the three-dimensional incompressible Navier-Stokes equations with pseudocompressibility method. Int. J. Numer. Meth. Fluids, 69(7):1165–1185, 2012.
[26] J.L.C. Chang and D. Kwak. On the method of pseudo compressibility for numerically solving incompressible flows. AIAA Paper 84-0252, 1984.
[27] S. Pirozzoli. On the spectral properties of shock-capturing schemes. J. Comput. Phys., 219 (2):489–497, 2006.
[28] A. Harten, B. Engquist, S. Osher, and S.R. Chakravarthy. Uniformly high order accurate essentially non-oscillatory schemes, III. J. Comput. Phys., 71(2):231–303, 1987.
[29] W.J. Coirier and K.G. Powell. Solution-adaptive Cartesian cell approach for viscous and inviscid flows. AIAA J., 34(5):938–945, May 1996.
[30] T.J. Barth. Recent developments in high order k-exact reconstruction on unstructured meshes. AIAA Paper 93-0668, 1993.
[31] R. Abgrall. On essentially non-oscillatory schemes on unstructured meshes: analysis and implementation. J. Comput. Phys., 114:45–58, 1994.
[32] T. Sonar. On the construction of essentially non-oscillatory finite volume approximations to hyperbolic conservation laws on general triangulations: polynomial recovery, accuracy and stencil selection. Comp. Meth. Appl. Mech. Eng., pp. 140–157, 1997.
[33] C.F. Ollivier-Gooch. Quasi-ENO schemes for unstructured meshes based on unlimited

data-dependent least-squares reconstruction. J. Comput. Phys., 133:6–17, 1997.
[34] G.S. Jiang and C.W. Shu. Efficient implementation of weighted ENO schemes. J. Comput. Phys., 126(1):202–228, 1996.
[35] D. Stanescu and W. Habashi. Essentially nonoscillatory Euler solutions on unstructured meshes using extrapolation. AIAA J., 36:1413–1416, 1998.
[36] O. Friedrich. Weighted essentially non-oscillatory schemes for the interpolation of mean values on unstructured grids. J. Comput. Phys., 144:194–212, 1998.
[37] C. Hu and C.W. Shu. Weighted essentially non-oscillatory schemes on triangular meshes. J. Comput. Phys., 150:97–127, 1999.
[38] C.F. Ollivier-Gooch and M. Van Altena. A high-order accurate unstructured mesh finite-volume scheme for the advection-diffusion equation. J. Comput. Phys., 181(2):729–752, 2002.
[39] A. Nejat and C. Ollivier-Gooch. A high-order accurate unstructured finite volume newton-krylov algorithm for inviscid compressible flows. J. Comput. Phys., 227(4):2582–2609, 2008.
[40] B. Cockburn and C.W. Shu. TVB Runge-Kutta local projection discontinous Galerkin finite-element method for conservation laws II: General framework. Math. Comp., 52:411, 1989.
[41] B. Cockburn, S. Hou, and C.W. Shu. TVB Runge-Kutta local projection discontinous Galerkin finite-element method for conservation laws IV: The multidimensional case. J. Comput. Phys., 54:545, 1990.
[42] R. Hartmann and P. Houston. Adaptive discontinuous Galerkin finite element methods for the compressible Euler equations. J. Comput. Phys., 183:508–532, 2002.
[43] H. Luo, J.D. Baum, and R. Löhner. A Hermite WENO-based limiter for discontinuous Galerkin method on unstructured grids. J. Comput. Phys., 225:686–713, 2007.
[44] G. Gassner, F. Lörcher, and C.D. Munz. A contribution to the construction of diffusion fluxes for finite volume and discontinuous Galerkin schemes. J. Comput. Phys., 224(2): 1049–1063, 2007.
[45] F. Bassi and S. Rebay. A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier Stokes equations. J. Comput. Phys., 131: 267–279, 1997.
[46] B. Cockburn and C.W. Shu. The local discontinuous Galerkin method for time-dependent convection diffusion system. SIAM J. Numer. Anal., 35(6):2440–2463, 1998.
[47] B. Leervan, M. Lo, and M. Raaltevan. A discontinuous Galerkin method for diffusion based on recovery. AIAA Paper 2007-4083, 2007.
[48] M. Raaltevan and B. Leervan. Bilinear forms for the recovery-based discontinuous Galerkin method for diffusion. Commun. Comput. Phys., 5(2–4):683–693, 2009.
[49] H. Liu and J. Yan. The direct discontinuous Galerkin (DDG) methods for diffusion problems. SIAM J. Numer. Anal., 41(1):675–698, 2009.
[50] Z.J. Wang. Spectral (finite) volume method for conservation laws on unstructured grids – basic formulation. J. Comput. Phys., 178:210–251, 2002.
[51] Z.J. Wang and Y. Liu. Spectral (finite) volume method for conservation laws on unstructured grids – ii. extenstion to two-dimensional scalar equation. J. Comput. Phys., 179: 665–697, 2002.
[52] Z.J. Wang, L. Zhang, and Y. Liu. High-order spectral volume method for 2d euler equations. Paper 2003–3534, AIAA, June 2003.
[53] Z.J. Wang and Y. Liu. Spectral (finite) volume method for conservation laws on unstructured grids – iii. one dimensional systems and partition optimization. Journal of Scientific Computing, 20(1):137–157, 2004.

[54] Y. Sun, Z.J. Wang, and Y. Liu. Spectral (finite) volume method for conservation laws on unstructured grids VI: extension to viscous flow. J. Comput. Phys., 215(1):41–58, 2006.
[55] H. Huynh. A flux reconstruction approach to high-order schemes including discontinuous Galerkin methods. AIAA Paper 2007-4079, 2007.
[56] Z.J. Wang and H. Gao. A unifying lifting collocation penalty formulation for the Euler equations on mixed grids. AIAA Paper 2009–401, 2009a.
[57] Z.J. Wang and H. Gao. A unifying lifting collocation penalty formulation including the discontinuous Galerkin, spectral volume/difference methods for conservation laws on mixed grids. J. Comput. Phys., 228:8161–8186, 2009b.
[58] A. Haselbacher. A WENO reconstruction algorithm for unstructed grids based on explicit stencil construction. AIAA paper 2005-0879, 2005.
[59] L. Ivan and C.P.T. Groth. High-order central CENO finite-volume scheme with adaptive mesh refinement. AIAA paper 2007-4323, 2007.
[60] L. Ivan and C.P.T. Groth. High-order solution-adaptive central essentially non-oscillatory (CENO) method for viscous flows. AIAA paper 2011-0367, 2011.
[61] L. Ivan and C.P.T. Groth. High-order central ENO scheme with adaptive mesh refinement for hyperbolic conservation laws. Commun. Comput. Phys., 2013a. submitted for publication.
[62] L. Ivan and C.P.T. Groth. High-order solution-adaptive central essentially non-oscillatory (CENO) method for viscous flows. J. Comput. Phys., 2013b. submitted for publication.
[63] 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. AIAA-2011-3854.
[64] A. Susanto, L. Ivan, H. De Sterck, and C.P.T. Groth. High-order central ENO finite-volume scheme for ideal MHD. J. Comput. Phys., 250(1):141 – 164, 2013.
[65] S.E. Rogers and D. Kwak. An upwind differencing scheme for the incompressible navier-strokes equations. Appl. Numer. Math., 8(1):43–64, 1991.
[66] Y.N. Chen, S.C. Yang, and J.Y. Yang. Implicit weighted essentially non-oscillatory schemes for the incompressible Navier-Stokes equations. Int. J. Numer. Meth. Fluids, 31(4):747–765, 1999.
[67] S.E. Rogers, J.L.C. Chang, and D. Kwak. A diagonal algorithm for the method of pseudo-compressibility. J. Comput. Phys., 73(2):364–379, 1987b.
[68] S. Yoon and D. Kwak. LU-SGS implicit algorithm for three-dimensional incompressible Navier-Stokes equations with source term. AIAA Paper 89-1964, 1989.
[69] R.S. Dembo, S.C. Eisenstat, and T. Steihaug. Inexact newton methods. SIAM J. Numer. Anal., 19(2):400–408, 1982.
[70] T.F. Chan and K.R. Jackson. Nonlinearly preconditioned krylov subspace methods for discrete newton algorithms. SIAM J. Sci. Stat. Comput., 5(3):533–542, 1984.
[71] P.N. Brown and Y. Saad. Hybrid Krylov methods for nonlinear systems of equations. SIAM J. Sci. Stat. Comput., 11(3):450–481, 1990.
[72] D.A. Knoll and D.E. Keyes. Jacobian-free Newton-Krylov methods: A survey of approaches and applications. J. Comput. Phys., 193(2):357–397, 2004.
[73] D.A. Knoll, V.A. Mousseau, L. Chacn, and J. Reisner. Jacobian-free Newton-Krylov methods for the accurate time integration of stiff wave systems. J. Comput. Phys., 25(1):213–230, 2005.
[74] D. Pan and C.. Chang. The capturing of free surfaces in incompressible multi-fluid flows. Int. J. Numer. Meth. Fluids, 33(2):203–222, 2000.

[75] L. Qian, D.M. Causon, D.M. Ingram, and C.G. Mingham. Cartesian cut cell two-fluid solver for hydraulic flow problems. J. Hydraul. Eng., 129(9):688–696, 2003.
[76] U. Riedel. A finite volume scheme on unstructured grids for stiff chemically reacting flows. Combust. Sci. Tech., 135(1-6):99–116, 1998.
[77] E. Shapiro and D. Drikakis. Artificial compressibility, characteristics-based schemes for variable density, incompressible, multi-species flows. part i. derivation of different formulations and constant density limit. J. Comput. Phys., 210(2):584–607, 2005.
[78] M. Azhdarzadeh and S.E. Razavi. A pseudo-characteristic based method for incompressible flows with heat transfer. J. Appl. Sci., 8(18):3183–3190, 2008.
[79] S.L. Chang and K.T. Rhee. Blackbody radiation functions. Int. Commun. Heat Mass Transfer, 11(5):451–455, 1984.
[80] A. Jameson. Time dependent calculations using multigrid, with applications to unsteady flows past airfoils and wings. AIAA paper 1991-1596, 1991.
[81] C.L. Merkle and M. Athavale. Time-accurate unsteady incompressible flow algorithms based on artificial compressibility. AIAA paper 87-1137, 1987.
[82] W.Y. Soh and J.W. Goodrich. Unsteady solution of incompressible Navier-Stokes equations. J. Comput. Phys., 79(1):113–134, 1988.
[83] Z. Qian, J. Zhang, and C. Li. Preconditioned pseudo-compressibility methods for incompressible navier-stokes equations. Science China: Physics, Mechanics and Astronomy, 53 (11):2090–2102, 2010.
[84] H. Lee and S. Lee. Convergence characteristics of upwind method for modified artificial compressibility method. Int. J. Aeronaut. Space Sci., 12(4):318–330, 2011.
[85] A.G. Malan, R.W. Lewis, and P. Nithiarasu. An improved unsteady, unstructured, artificial compressibility, finite volume scheme for viscous incompressible flows: Part I. theory and implementation. Int. J. Numer. Meth. Engin., 54(5):695–714, 2002a.
[86] A.G. Malan, R.W. Lewis, and P. Nithiarasu. An improved unsteady, unstructured, artificial compressibility, finite volume scheme for viscous incompressible flows: Part II. application. Int. J. Numer. Meth. Engin., 54(5):715–729, 2002b.
[87] R.L. Naff, T.F. Russell, and J.D. Wilson. Shape functions for three-dimensional control-volume mixed finite-element methods on irregular grids. Developments in Water Science, vol. 47, pp. 359–366. Elsevier, 2002a.
[88] R. Naff, T. Russell, and J. Wilson. Shape functions for velocity interpolation in general hexahedral cells. Computat. Geosci., 6:285–314, 2002b.
[89] C.A. Felippa. A compendium of FEM integration formulas for symbolic work. Eng. Computation., 21(8):867–890, 2004.
[90] L. Ivan. Development of high-order CENO finite-volume schemes with block-based adaptive mesh refinement. PhD thesis, University of Toronto, 2011.
[91] C.L. Lawson and R.J. Hanson. Solving least squares problems. Prentice-Hall, 1974.
[92] D.J. Mavriplis. Revisiting the least-squares procedure for gradient reconstruction on unstructured meshes. AIAA paper 2003-3986, 2003.
[93] A. Jalali and C. Ollivier-Gooch. Higher-order finite volume solution reconstruction on highly anisotropic meshes. AIAA Paper 2013-2565, 2013.
[94] J.S. Park, S.H. Yoon, and C. Kim. Multi-dimensional limiting process for hyperbolic conservation laws on unstructured grids. J. Comput. Phys., 229(3):788–812, 2010.
[95] V. Venkatakrishnan. On the accuracy of limiters and convergence to steady state solutions. AIAA Paper 93-0880, 1993.
[96] P.L. Roe. Approximate Riemann solvers, parameter vectors, and difference schemes.

J. Comput. Phys., 43:357–372, 1981.
[97] P.L. Roe and J. Pike. Efficient construction and utilisation of approximate Riemann solutions. R. Glowinski and J.L. Lions, editors, Computing Methods in Applied Science and Engineering, vol. VI, pp. 499–518, Amsterdam, 1984. North-Holland.
[98] S.R. Mathur and J.Y. Murthy. A pressure-based method for unstructured meshes. Numer. Heat Transfer, Part B, 31(2):195–215, 1997.
[99] G. Karypis and K. Schloegel. ParMeTis: Parallel graph graph partitioning and sparse matrix ordering library, Version 4.0. `http://www.cs.umn.edu/ metis`, 2011.
[100] W. Gropp, E. Lusk, and A. Skjellum. Using MPI. MIT Press, Cambridge, Massachussets, 1999.
[101] C.P.T. Groth and S.A. Northrup. Parallel implicit adaptive mesh refinement scheme for body-fitted multi-block mesh. 17th AIAA Computational Fluid Dynamics Conference, Toronto, Ontario, Canada, 6-9 June 2005. AIAA paper 2005-5333.
[102] M.R.J. Charest, C.P.T. Groth, and Ö.L. Gülder. A computational framework for predicting laminar reactive flows with soot formation. Combust. Theor. Modelling, 14(6):793–825, 2010.
[103] M.R.J. Charest, C.P.T. Groth, and Ö.L. Gülder. Solution of the equation of radiative transfer using a Newton-Krylov approach and adaptive mesh refinement. J. Comput. Phys., 231(8): 3023–3040, 2012.
[104] Y. Saad and M.H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J. Sci. Stat. Comput., 7(3):856–869, 1986.
[105] Y. Saad. Krylov subspace methods on supercomputers. SIAM J. Sci. Stat. Comput., 10(6): 1200–1232, 1989.
[106] Y. Saad. Iterative Methods for Sparse Linear Systems. PWS Publishing Company, Boston, 1996.
[107] E. Cuthill and J. McKee. Reducing the bandwidth of sparse symmetric matrices. Proceedings of the 1969 24th National Conference, ACM '69, pp. 157–172, New York, NY, USA, 1969. ACM.
[108] W.A. Mulder and B. Van Leer. Experiments with implicit upwind methods for the Euler equations. J. Comput. Phys., 59:232–246, 1985.
[109] H. Bijl, M.H. Carpenter, and V.N. Vatsa. Time integration schemes for the unsteady Navier-Stokes equations. AIAA Paper 2001-2612, 2001.
[110] H. Bijl, M.H. Carpenter, V.N. Vatsa, and C.A. Kennedy. Implicit time integration schemes for the unsteady compressible Navier-Stokes equations: Laminar flow. J. Comput. Phys., 179:313–329, 2002.
[111] H. Lomax, T.H. Pulliam, and D.W. Zingg. Fundamentals of Computational Fluid Dynamics. Springer, New York, 2003.
[112] M. Tabesh and D.W. Zingg. Efficient implicit time-marching methods using a Newton-Krylov algorithm. 47th AIAA Aerospace Sciences Meeting and Exhibit, Orlando, Florida, 5-8 January 2009. AIAA paper 2009-0164.
[113] A. Genz and R. Cools. An adaptive numerical cubature algorithm for simplices. ACM Trans. Math. Softw., 29(3):297–308, 2003.
[114] R. Cools and A. Haegemans. Algorithm 824: CUBPACK: A package for automatic cubature; framework description. ACM Trans. Math. Softw., 29(3):287–296, 2003.
[115] R. Abgrall. Design of an essentially non-oscillatory reconstruction procedure on finite-element type meshes. ICASE Contractor Report 189574, 1991.
[116] G.I. Taylor and A.E. Green. Mechanism of the production of small eddies from large ones.

Proc. Royal Soc. London A, 158(895):499–521, 1937.
[117] P. Shankar and M. Deshpande. Fluid mechanics in the driven cavity. Ann. Rev. Fluid Mech., 32(1):93–136, 2000.
[118] B.N. Jiang, T. Lin, and L.A. Povinelli. Large-scale computation of incompressible viscous flow by least-squares finite element method. Comp. Meth. Appl. Mech. Eng., 114(3):213–231, 1994.
[119] J.Y. Yang, S.C. Yang, Y.N. Chen, and C.A. Hsu. Implicit weighted ENO schemes for the three-dimensional incompressible Navier–Stokes equations. J. Comput. Phys., 146(1):464–487, 1998.
[120] J.S. Sachdev, C.P.T. Groth, and J.J. Gottlieb. A parallel solution-adaptive scheme for multiphase core flows in solid propellant rocket motors. Int. J. Comput. Fluid Dyn., 19(2):159–177, 2005.
[121] X. Gao and C.P.T. Groth. A parallel adaptive mesh refinement algorithm for predicting turbulent non-premixed combusting flows. Int. J. Comput. Fluid Dyn., 20(5):349–357, 2006.
[122] X. Gao and C.P.T. Groth. A parallel solution-adaptive method for three-dimensional turbulent non-premixed combusting flows. J. Comput. Phys., 229(9):3250–3275, 2010.
[123] X. Gao, S. Northrup, and C.P.T. Groth. Parallel solution-adaptive method for two-dimensional non-premixed combusting flows. Prog. Comput. Fluid. Dy., 11(2):76–95, 2011.