The Eighth International Conference on Computational Fluid Dynamics (ICCFD8), Chengdu, Sichuan,China, July 14-18, 2014

ICCFD8-2014-0093

A High-Order Accurate Unstructured Finite-Volume Algorithm for Turbulent Flames

M.R.J. Charest* and C.P.T. Groth**

* Los Alamos National Laboratory
P.O. Box 1663, Los Alamos, NM 87545, U.S.A.

** University of Toronto Institute for Aerospace Studies
4925 Dufferin Street, Toronto, Ontario, Canada M3H 5T6

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, universally-applicable, high-order discretizations remain somewhat illusive, especially for more arbitrary unstructured meshes and for large-eddy simulation (LES) of turbulent reacting 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 turbulent, reactive, low speed flows on three-dimensional unstructured meshes. The proposed scheme is applied to the pseudo-compressibility formulation of the Favre-filtered governing equations and the resulting discretized equations are solved with a parallel implicit Newton-Krylov algorithm. Temporal derivatives are discretized using the family of high-order backward difference formulas (BDF) and the resulting equations are solved via a dual-time-stepping-like approach. Large-eddy simulations of a laboratory-scale turbulent flame are carried out and the proposed finite-volume scheme is validated against experimental measurements. The high-order scheme is demonstrated to provide both reliable and accurate solutions for LES of turbulent reactive flows.

Keywords: Numerical Algorithms, Computational Fluid Dynamics, High-Order Methods, Turbulent Flames.

1 Introduction

Most practical combustion devices — such as those found in furnaces, gas turbines, diesel engines, and solid propellant rocket motors — make use of turbulent flames. These complex flames involve a wide range of scales and physical/chemical phenomena that present a challenge to model numerically. While computational fluid dynamics (CFD) has proven to be an important enabling technology for these devices, there is currently an upper limit on the problem size that may be simulated because of the complexities in the physics and the range of scales that must be resolved. For example, direct numerical simulation (DNS) of full-scale practical turbulent flames is not yet possible. For these reasons, mathematical models of reactive flows must rely heavily on engineering approximations and sophisticated numerical methods to represent the underlying physics and ensure that the computations remain tractable.

The primary approaches for modelling practical turbulent flows are Reynolds-averaged Navier-Stokes (RANS) and large-eddy simulation (LES) [1, 2]. LES is considered to be intermediate between DNS and RANS since LES resolves the large-scale motions of the flow and models the small scales. However, despite these physical simplifications, even LES of full-scale practical turbulent reactive flows requires significant computational resources and is therefore not practical for industry-relevant problems on a routine basis.

*Corresponding author's email: charest@utias.utoronto.ca. This work was performed while the author held a postdoctoral position at the University of Toronto Institute for Aerospace Studies.

One promising route to reducing the computational costs associated with LES of reactive flows is offered by high-order discretization methods (i.e., methods higher than second order). Being more more accurate than standard first- and second-order methods, high-order methods have the potential to offer improved numerical efficiency since fewer computational cells are required to achieve a desired level of accuracy [3]. However, these advantages are often difficult to fully realize because of the conflicting relationship between accuracy and robustness. Challenges include the requirement that for discretizations of hyperbolic partial differential equations (PDEs) and operators remain robust in the presence of discontinuities [4] while discretizations of elliptic PDEs should ideally satisfy a maximum principle [5]. Additionally, reactive flow modelling requires the integration of numerically stiff and mathematically complex source terms over the computational domain. Many different high-order formulations have been developed for both structured and unstructured mesh — see, for example, the review by Wang et al. [6] and references therein. Despite recent advances, there is still no universally accepted scheme that solves all of the aforementioned issues and is applicable to arbitrary meshes.

Due to the aforementioned challenges, the application of high-order discretization methods to realistic flows — such as the practical turbulent flames of interest here — has been somewhat limited. Nevertheless, to date there have been some applications of of high-order techniques to more practical turbulent flows. For example, Moureau et al. [7], Desjardins et al. [8], and Chen et al. [9] have successfully applied high-order finite-difference schemes to the LES and/or DNS of both turbulent reactive and non-reacting flows. As the preceding schemes are finite-difference based, a significant limitation is their applicability only to regular Cartesian and body-fitted curvilinear grids. For unstructured meshes, Tsoutsanis et al. [10] have developed and applied a high-order, essentially non-oscillatory (ENO) based, finite-volume scheme for turbulent flows; however, these authors only considered the case of non-reacting flows.

Recently, Charest et al. [11, 12] have developed a high-order central ENO (CENO), cell-centered, finite-volume scheme for low-speed, viscous flows on three-dimensional unstructured mesh. The scheme, which is based on the original CENO formulation of Ivan and Groth [13, 14, 15, 16] for two-dimensional structured mesh and was recently applied to ideal magnetohydrodynamics by Susanto et al. [17], has been demonstrated to remain both accurate and robust for a variety of physically-complex flows. This robustness is provided by a hybrid procedure that switches between two different reconstructions. A high-order k-exact reconstruction is used in smooth regions and a monotonicity-preserving piecewise linear reconstruction is used in under-resolved/discontinuous regions. The CENO method offers computational efficiencies over standard ENO and weighted ENO (WENO) schemes [4, 18–20] because only a single central stencil is required for performing the solution reconstruction, the same for each variable. The ENO and WENO methods both use variable/multiple stencils to ensure smooth reconstructions which may be different for each solution variable. Additionally, selection of appropriate stencils on general multi-dimensional unstructured meshes is not straightforward [21–24] and in many cases can result in poorly conditioned linear systems [23, 24].

The present study considers the extension of the high-order CENO finite-volume scheme to the solution of the Favre-filtered form of the Navier-Stokes equations suitable for performing LES of low-Mach-number, turbulent, reactive flows on unstructured mesh. A simple eddy-dissipation-based combustion model is is for the treatment of the treatment of turbulence/chemistry interaction and evaluation of the filtered reaction rates. The proposed algorithm is applied to a laboratory-scale premixed turbulent flame and analyzed in terms of accuracy and robustness. In particular, low- and high-order numerical predictions were compared with experimental measurements for quantities such as flame surface density, flame curvature, flame height, temperature, fluid velocity, and species concentrations. The potential of the high-order CENO scheme for LES of practical reactive flows is illustrated.

2 Pseudo-Compressibility Approach for LES of Reactive Low Speed Flows

2.1 Governing Equations and Turbulent Closure Models

In the present study, the equations governing turbulent reactive flows at low Mach numbers are considered. Large-scale turbulence is captured using LES with two different closure models for the subfilter-scale stresses, depending upon the particular application. In three space dimensions, the Favre-filtered PDEs that govern

turbulent reactive flows in this case are

$$\frac{\partial \bar{\rho}}{\partial t} + \frac{\partial}{\partial x_i}(\bar{\rho}\tilde{u}_i) = 0 \tag{1a}$$

$$\frac{\partial}{\partial t}(\bar{\rho}\tilde{u}_i) + \frac{\partial}{\partial x_j}(\bar{\rho}\tilde{u}_j\tilde{u}_i) + \frac{\partial \bar{p}}{\partial x_i} = \frac{\partial \lambda_{ji}}{\partial x_j} \tag{1b}$$

$$\frac{\partial}{\partial t}\left(\bar{\rho}\tilde{h}\right) + \frac{\partial}{\partial x_j}\left(\bar{\rho}\tilde{u}_j\tilde{h}\right) = -\frac{\partial q_j}{\partial x_j} \tag{1c}$$

$$\frac{\partial}{\partial t}\left(\bar{\rho}\tilde{f}\right) + \frac{\partial}{\partial x_j}\left(\bar{\rho}\tilde{u}_j\tilde{f}\right) = -\frac{\partial \mathcal{I}_j}{\partial x_j} \tag{1d}$$

$$\frac{\partial}{\partial t}(\bar{\rho}\tilde{y}_\mathrm{f}) + \frac{\partial}{\partial x_j}(\bar{\rho}\tilde{u}_j\tilde{y}_\mathrm{f}) = -\frac{\partial \mathcal{J}_j}{\partial x_j} + \bar{\rho}\dot{\omega}_\mathrm{T} \tag{1e}$$

where t is the time, p is the total pressure, ρ is the fluid density (which varies according to the ideal gas law), u_j is the bulk fluid velocity, f is the mixture fraction, y_f is the fuel mass fraction, $h = \int_{T_0}^{T} c_p \,\mathrm{d}T$ is the fluid enthalpy, T is the fluid temperature, c_p is the fluid specific heat, and $\dot{\omega}_\mathrm{T}$ is the filtered fuel reaction rate source term. For the simplified combustion models considered herein, knowledge of the local values of the mixture fraction and fuel mass fraction is sufficient to fully define the extent or progress of reactions and the composition of the mixture. In Eq. (1), the over-bar and over-tilde symbols denote Reynolds-averaged and Favre-averaged quantities, respectively.

The total stress tensor, λ_{ij}, of Eq. (1), includes both the resolved laminar and modelled filter-scale turbulent stresses and is given by

$$\lambda_{ij} = (\mu + \mu_\mathrm{T})\left[\left(\frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i}\right) - \frac{2}{3}\delta_{ij}\frac{\partial \tilde{u}_k}{\partial x_k}\right] \tag{2}$$

where μ is the fluid dynamic viscosity and μ_T is the subgrid turbulent eddy-viscosity. The total heat flux vector due to both molecular and subfilter-scale turbulent motion, q_j, is evaluated using using Fourier's law of thermal diffusion combined with a gradient-based assumption for the unresolved turbulent component. It is defined as

$$q_j = -\left(\frac{\mu c_p}{\mathrm{Pr}_\mathrm{L}} + \frac{\mu_\mathrm{T} c_p}{\mathrm{Pr}_\mathrm{T}}\right)\frac{\partial \tilde{T}}{\partial x_j} \tag{3}$$

where constant Prandtl numbers for the laminar, Pr_L, and turbulent, Pr_T, components were assumed. The laminar Prandtl number is taken to be $\mathrm{Pr}_\mathrm{L} = 0.7$, which is typical for most gases, and the turbulent Prandtl number is assumed to be $\mathrm{Pr}_\mathrm{T} = 0.9$ based on recommendations by Wilcox [1].

The resolved and subfilter-scale diffusive fluxes for the mixture fraction and fuel mass fraction, $\mathcal{I}_j$ and $\mathcal{J}_j$, were determined using the usual Fick's law combined and gradient-based assumption for the unresolved turbulent fluxes. The total diffusive fluxes are taken to have the forms

$$\mathcal{I}_j = \left(\frac{\mu}{\mathrm{Sc}_\mathrm{L}} + \frac{\mu_\mathrm{T}}{\mathrm{Sc}_\mathrm{T}}\right)\frac{\partial \tilde{f}}{\partial x_j} \tag{4}$$

$$\mathcal{J}_j = \left(\frac{\mu}{\mathrm{Sc}_\mathrm{L}} + \frac{\mu_\mathrm{T}}{\mathrm{Sc}_\mathrm{T}}\right)\frac{\partial \tilde{y}_\mathrm{f}}{\partial x_j} \tag{5}$$

where Sc_L and Sc_T are Schmidt numbers for the laminar and turbulent components, respectively. Constant laminar and subfilter turbulent Schmidt numbers are assumed with values of 0.9 and 1, respectively.

The standard Smagorinsky model [25] is used here with wall damping to evaluate the eddy viscosity associated with the subfilter-scale turbulence for which μ_T is defined by

$$\mu_\mathrm{T} = \bar{\rho} L_\mathrm{s}^2 \sqrt{2 S_{ij} S_{ij}} \tag{6}$$

where L_s is the mixing length for the subgrid scales and S_{ij} is the local resolved strain rate tensor, given by

$$S_{ij} = \frac{1}{2}\left(\frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i}\right) \tag{7}$$

The length scale, L_s, is computed using a two-layer approach to account for the presence of solid walls [26, 27]. It is defined as

$$L_s = \min\left(\kappa y_{\text{wall}},\ f_\mu c_s \Delta\right) \tag{8}$$

where $\kappa = 0.41$ is the von Kármán constant, y_{wall} is the local distance from the wall, f_μ is the Van Driest [28] damping function, $c_s = 0.17$ is the usual Smagorinsky coefficient, and Δ is the filter width. The Van Driest [28] damping function is

$$f_\mu = 1 - \exp\left(-y^+/A\right) \tag{9}$$

where y^+ is the dimensionless wall distance and $A = 25$. For the implicit filtering approach employed here, the width of the low-pass filter is taken to be related to the grid spacing. A filter width equal to twice the measure of the local mesh size was employed, i.e.,

$$\Delta = 2\sqrt[3]{V} \tag{10}$$

where V is the local cell volume.

For cases involving significant interactions between the flow and the walls, such as flames near bluff bodies, the wall-adapting local eddy-viscosity (WALE) model of Nicoud and Ducros [29] for the turbulent eddy viscosity provided improved numerical stability and more accurate results. The WALE eddy viscosity is defined as

$$\mu_T = \bar{\rho} L_s^2 \frac{\left(\delta_{ij}^d \delta_{ij}^d\right)^{3/2}}{\left(S_{ij} S_{ij}\right)^{5/2} + \left(\delta_{ij}^d \delta_{ij}^d\right)^{5/4}} \tag{11}$$

where δ_{ij}^d is the traceless, symmetric part of the velocity gradient tensor squared. It is given by

$$\delta_{ij}^d = \frac{1}{2}\left(g_{ij}^2 + g_{ji}^2\right) - \frac{1}{3}\delta_{ij} g_{kk}^2 \tag{12}$$

where $g_{ij}^2 = g_{ik} g_{kj}$ and $g_{ij} = \partial \tilde{u}_i / \partial x_j$ is the velocity gradient tensor. The definition of the WALE length scale is similar to Eq. (8) and is given by the following relation:

$$L_s = \min\left(\kappa y_{\text{wall}},\ f_\mu c_w \Delta\right) \tag{13}$$

where the constant c_w is equal to 0.5.

As in all LES for reactive flows, the modelling of the filtered reaction rates requires special attention. In the present work, the chemical reactions are assumed to be very fast compared with the fluid scales. As such, the combustion is controlled only by the rate of mixing between fresh reactants and/or hot combustion products. The rate of mixing, and consequently the rate of combustion, is then taken to be described by the eddy-dissipation or eddy-break-up model [30, 31]. The filtered reaction rate for the fuel mass fraction, $\dot{\omega}_T$, is then given by

$$\dot{\omega}_T = \tau_m^{-1}\, a\, \min\left[\tilde{y}_f,\ \tilde{y}_o/s,\ b\tilde{y}_p/(1+s)\right] \tag{14}$$

where s is the reaction stoichiometric coefficient, τ_m is a turbulent mixing time scale, and $\tilde{y}_o$ and $\tilde{y}_p$ are the mass fractions of oxygen and products, respectively. For the computations described herein, methane is the gaseous fuel of interest and a simple one-step chemical mechanism [32] is used to represent its chemistry with air. The coefficients a and b were set to 2.0 and 0.5, respectively, and the turbulent mixing time scale was determined from the local eddy-viscosity and length scale as

$$\tau_m = \frac{\mu_T}{\bar{\rho}(c_s \Delta)^2} \tag{15}$$

2.2 Pseudo-Compressibility Approach

For incompressible and/or low-Mach-number flows, derivatives of density with respect to time are zero or vanishingly small. This leads to ill-conditioning of the governing equations that is overcome here by modifying these derivatives according to the so-called 'pseudo-compressibility' method [33–36]. Following application of the pseudo-compressibility method, the modified governing equations are however no longer time-accurate. A dual-time-stepping-like approach is therefore adopted here to regain temporal accuracy [37–41]. Application

of the pseudo-compressibility and dual-time-stepping-like approaches to the Navier-Stokes system given previously in Eq. (1) results in the following governing set of equations expressed in matrix-vector form:

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbf{\Gamma}\frac{\partial \mathbf{W}}{\partial \tau} + \vec{\nabla}\cdot\left(\vec{\mathbf{F}} - \vec{\mathbf{F}}_{\mathbf{v}}\right) = \mathbf{S} \tag{16}$$

where $\mathbf{U}$ and $\mathbf{W}$ are the vectors of conserved and primitive variables, $\vec{\mathbf{F}} = [\mathbf{E}, \mathbf{F}, \mathbf{G}]$ and $\vec{\mathbf{F}}_{\mathbf{v}} = [\mathbf{E}_{\mathbf{v}}, \mathbf{F}_{\mathbf{v}}, \mathbf{G}_{\mathbf{v}}]$ are the inviscid and viscous solution flux dyads, $\mathbf{S}$ is the source vector, and $\mathbf{\Gamma}$ is the transformation matrix associated with the pseudo-compressible modifications. The various vector and matrix quantities appearing in Eq. (16) are defined as

$$\mathbf{U} = \begin{bmatrix} \bar{\rho} \\ \bar{\rho}\tilde{u} \\ \bar{\rho}\tilde{v} \\ \bar{\rho}\tilde{w} \\ \bar{\rho}\tilde{h} \\ \bar{\rho}\tilde{f} \\ \bar{\rho}\tilde{y}_{\mathrm{f}} \end{bmatrix}, \ \mathbf{W} = \begin{bmatrix} \bar{p} \\ \tilde{u} \\ \tilde{v} \\ \tilde{w} \\ \tilde{T} \\ \tilde{f} \\ \tilde{y}_{\mathrm{f}} \end{bmatrix}, \ \mathbf{E} = \begin{bmatrix} \bar{\rho}\tilde{u} \\ \bar{\rho}\tilde{u}^2 + \bar{p} \\ \bar{\rho}\tilde{u}\tilde{v} \\ \bar{\rho}\tilde{u}\tilde{w} \\ \bar{\rho}\tilde{u}\tilde{h} \\ \bar{\rho}\tilde{u}\tilde{f} \\ \bar{\rho}\tilde{u}\tilde{y}_{\mathrm{f}} \end{bmatrix}, \ \mathbf{F} = \begin{bmatrix} \bar{\rho}\tilde{v} \\ \bar{\rho}\tilde{v}\tilde{u} \\ \bar{\rho}\tilde{v}^2 + \bar{p} \\ \bar{\rho}\tilde{v}\tilde{w} \\ \bar{\rho}\tilde{v}\tilde{h} \\ \bar{\rho}\tilde{v}\tilde{f} \\ \bar{\rho}\tilde{v}\tilde{y}_{\mathrm{f}} \end{bmatrix}, \ \mathbf{G} = \begin{bmatrix} \bar{\rho}\tilde{w} \\ \bar{\rho}\tilde{w}\tilde{u} \\ \bar{\rho}\tilde{w}\tilde{v} \\ \bar{\rho}\tilde{w}^2 + \bar{p} \\ \bar{\rho}\tilde{w}\tilde{h} \\ \bar{\rho}\tilde{w}\tilde{f} \\ \bar{\rho}\tilde{w}\tilde{y}_{\mathrm{f}} \end{bmatrix},$$

$$\mathbf{E}_{\mathbf{v}} = \begin{bmatrix} 0 \\ \lambda_{xx} \\ \lambda_{xy} \\ \lambda_{xz} \\ -q_x \\ -\mathcal{I}_x \\ -\mathcal{J}_x \end{bmatrix}, \ \mathbf{F}_{\mathbf{v}} = \begin{bmatrix} 0 \\ \lambda_{yx} \\ \lambda_{yy} \\ \lambda_{yz} \\ -q_y \\ -\mathcal{I}_y \\ -\mathcal{J}_y \end{bmatrix}, \ \mathbf{G}_{\mathbf{v}} = \begin{bmatrix} 0 \\ \lambda_{zx} \\ \lambda_{yz} \\ \lambda_{zz} \\ -q_z \\ -\mathcal{I}_z \\ -\mathcal{J}_z \end{bmatrix}, \ \mathbf{S} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \bar{\rho}\dot{\omega}_{\mathrm{T}} \end{bmatrix},$$

$$\mathbf{\Gamma} = \begin{bmatrix} \frac{1}{\beta} & 0 & 0 & 0 & \bar{\rho}_T & \bar{\rho}_f & \bar{\rho}_{y_{\mathrm{f}}} \\ \frac{\alpha}{\beta}\tilde{u} & \bar{\rho} & 0 & 0 & \bar{\rho}_T\tilde{u} & \bar{\rho}_f\tilde{u} & \bar{\rho}_{y_{\mathrm{f}}}\tilde{u} \\ \frac{\alpha}{\beta}\tilde{v} & 0 & \bar{\rho} & 0 & \bar{\rho}_T\tilde{v} & \bar{\rho}_f\tilde{v} & \bar{\rho}_{y_{\mathrm{f}}}\tilde{v} \\ \frac{\alpha}{\beta}\tilde{w} & 0 & 0 & \bar{\rho} & \bar{\rho}_T\tilde{w} & \bar{\rho}_f\tilde{w} & \bar{\rho}_{y_{\mathrm{f}}}\tilde{w} \\ \frac{\alpha}{\beta}\tilde{h} & 0 & 0 & 0 & \bar{\rho}c_p + \bar{\rho}_T\tilde{h} & \bar{\rho}\tilde{h}_f + \bar{\rho}_f\tilde{h} & \bar{\rho}\tilde{h}_{y_{\mathrm{f}}} + \bar{\rho}_{y_{\mathrm{f}}}\tilde{h} \\ \frac{\alpha}{\beta}\tilde{f} & 0 & 0 & 0 & \bar{\rho}_T\tilde{f} & \bar{\rho} + \bar{\rho}_f\tilde{f} & \bar{\rho}_{y_{\mathrm{f}}}\tilde{f} \\ \frac{\alpha}{\beta}\tilde{y}_{\mathrm{f}} & 0 & 0 & 0 & \bar{\rho}_T\tilde{y}_f & \bar{\rho}_f\tilde{y}_f & \bar{\rho} + \bar{\rho}_{y_{\mathrm{f}}}\tilde{y}_{\mathrm{f}} \end{bmatrix}.$$

where β is the pseudo-compressibility factor, α is a preconditioning parameter, and τ denotes the pseudo-time since the original governing equations (Eq. (1)) are no longer time-accurate. The subscripts T, f, and y_{f} denote partial derivatives with respect to temperature, mixture fraction, and fuel mass fraction, respectively.

The preconditioning parameter, α, controls how the original governing equations are modified as described by Turkel [36]. A value for α equal to 1 or 2 corresponds to preconditioning the conservative or primitive form of the governing equations, respectively. As in previous research [12], $\alpha = 1$ was selected for its convergence and stability characteristics [36, 42–45].

2.3 Inviscid Eigenstructure

The Jacobian matrix of the steady physical-time, unsteady pseudo-time inviscid system is given by

$$\mathbf{A} = \mathbf{\Gamma}^{-1}\frac{\partial\left(\vec{\mathbf{F}}\cdot\hat{n}\right)}{\partial\mathbf{W}} = \begin{bmatrix} 0 & n_x\bar{\rho}\beta & n_y\bar{\rho}\beta & n_z\bar{\rho}\beta & 0 & 0 & 0 \\ \frac{n_x}{\bar{\rho}} & v_n & 0 & 0 & 0 & 0 & 0 \\ \frac{n_y}{\bar{\rho}} & 0 & v_n & 0 & 0 & 0 & 0 \\ \frac{n_z}{\bar{\rho}} & 0 & 0 & v_n & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & v_n & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & v_n & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & v_n \end{bmatrix} \tag{17}$$

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

$$\mathbf{X} = \begin{bmatrix} -\frac{\rho}{2}(v_n - c) & -\frac{\rho}{2}(v_n + c) & 0 & 0 & 0 & 0 & 0 \\ n_x & n_x & -n_y & -n_z & 0 & 0 & 0 \\ n_y & n_y & n_x & 0 & 0 & 0 & 0 \\ n_z & n_z & 0 & n_x & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{18}$$

and the corresponding eigenvalues of Eq. (17) are given by

$$\boldsymbol{\lambda} = \left[\frac{1}{2}v_n - \frac{1}{2}\sqrt{{v_n}^2 + 4\beta},\quad \frac{1}{2}v_n + \frac{1}{2}\sqrt{{v_n}^2 + 4\beta},\quad v_n,\quad v_n,\quad v_n,\quad v_n,\quad v_n\right] \tag{19}$$

The minimum value of β was chosen here based on the previous analysis by Turkel [36]:

$$\beta \geq \max\left[2\left(u^2 + v^2 + w^2\right),\ \varepsilon\right] \tag{20}$$

where ε is a smallness parameter, equal to 10^{-3}, to avoid singularities at stagnation points.

3 CENO Finite-Volume Scheme

3.1 Semi-Discrete Approach

The proposed cell-centered finite-volume scheme follows the approach for unstructured mesh described by Charest et al. [12], modified to include contributions from source terms associated with the chemical kinetics. Discretizing the domain into finite-sized computational cells and applying the integral form of Eq. (16) to

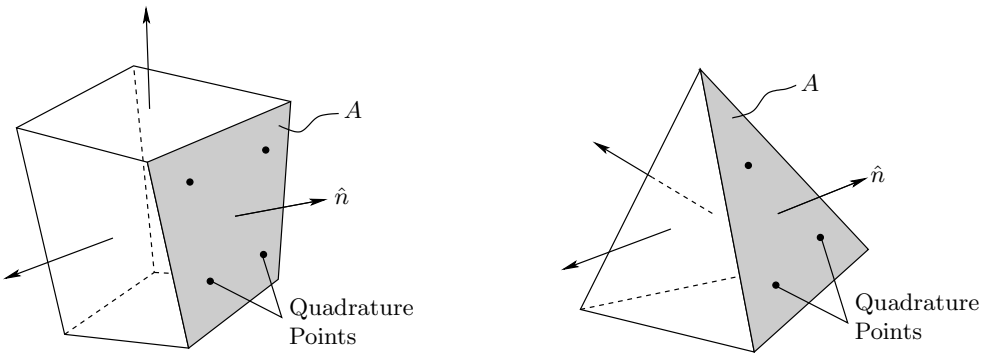


Fig. 1. Computational cells: (left) arbitrary hexahedron and (right) tetrahedron.

Table 1. Gauss quadrature rules used for integration.

Reconstruction	**Tetrahedra**			**Cartesian / Hexahedra**		
	G_f	G_v	Degree of Precision	G_f	G_v	Degree of Precision
Constant (k=0)	1	1	1	1	1	1
Linear (k=1)	1	1	1	1	1	1
Quadratic (k=2)	3	4	2	4	8	3
Cubic (k=3)	4	8	3	4	8	3
Quartic (k=4)	6	14	4	9	27	5

each cell Q_i gives the following coupled system of ordinary differential equations (ODEs) for cell-averaged solution quantities:

$$\frac{\mathrm{d}\overline{\mathbf{U}}_i}{\mathrm{d}t} + \mathbf{\Gamma}_i \frac{\mathrm{d}\overline{\mathbf{W}}_i}{\mathrm{d}\tau} = -\frac{1}{V_i} \oiint_{\partial Q_i} \left(\vec{\mathbf{F}} - \vec{\mathbf{F}}_\mathbf{v} \right) \cdot \hat{n} \, \mathrm{d}\Gamma + \frac{1}{V_i} \iiint_{Q_i} \mathbf{S} \, \mathrm{d}V = -\mathbf{R}_i, \qquad i = 1, 2, \ldots, N_\mathrm{c} \tag{21}$$

where N_c is the number of cells, V_i is the cell volume, $\hat{n}$ is a unit vector normal to the control volume surface, ∂Q_i. Applying Gauss quadrature to evaluate the surface and volume integrals in Eq. (21) produces a semi-discrete set of nonlinear ODEs given by

$$\frac{\mathrm{d}\overline{\mathbf{U}}_i}{\mathrm{d}t} + \mathbf{\Gamma}_i \frac{\mathrm{d}\overline{\mathbf{W}}_i}{\mathrm{d}\tau} = -\frac{1}{V_i} \sum_{l=1}^{N_\mathrm{f}} \sum_{m=1}^{G_\mathrm{f}} \left[\omega_\mathrm{f} \left(\vec{\mathbf{F}} - \vec{\mathbf{F}}_\mathbf{v} \right) \cdot \hat{n} \right]_{i,l,m} + \frac{1}{V_i} \sum_{n=1}^{G_\mathrm{v}} \left[\omega_\mathrm{v} \mathbf{S} \right]_{i,n} = -\mathbf{R}_i \tag{22}$$

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

Both hexahedral- and tetrahedral-shaped computational cells are considered herein, which are shown schematically in Fig. 1. For integration on tetrahedra and rectangular cells, standard Gauss quadrature rules are mapped directly from the canonical form to the local Cartesian coordinate system. More general hexahedra can have non-rectangular faces, i.e., faces composed of vertices that do not all lie on a particular plane, and thus require special consideration to maintain accuracy. For these irregular hexahedra, standard quadrature rules for cubes and squares are mapped using a trilinear coordinate transformation [46, 47]. This is described in more detail below and a similar procedure is used by Ivan et al. [48] in the recent application of the high-order CENO scheme to three-dimensional block-structured hexahedral cubed-sphere meshes. The coefficients corresponding to the quadrature rules applied here are as given by Felippa [49] and summarized in Table 1.

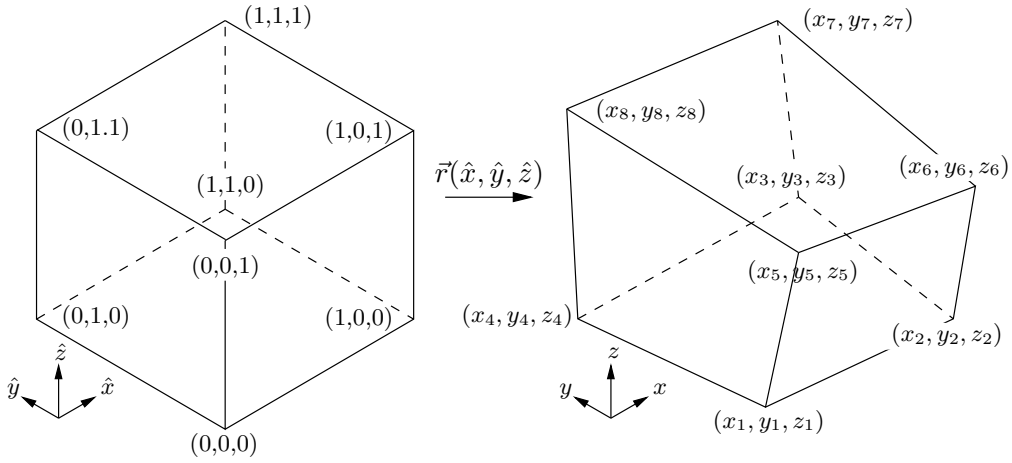


Fig. 2. (Left) Reference cube $\hat{Q}$ and (right) arbitrary hexahedron Q.

3.2 Trilinear Coordinate Transformation for Arbitrary Hexahedra

The geometric representation of general hexahedral cells, Q, used here is based on a trilinear mapping to a reference hexahedron, $\hat{Q}$ [46–48]. Although Q may have faces which are non-planar, $\hat{Q}$ has planar faces, since it is modelled as a unit cube. As shown in Fig. 2, the mapping associates any reference location $(\hat{x}, \hat{y}, \hat{z})$ in $\hat{Q}$ to a point (x, y, z) in Q:

$$\vec{r}(\hat{x},\hat{y},\hat{z}) = \vec{a} + \vec{b}\hat{x} + \vec{c}\hat{y} + \vec{d}\hat{z} + \vec{e}\hat{x}\hat{y} + \vec{f}\hat{x}\hat{z} + \vec{g}\hat{y}\hat{z} + \vec{h}\hat{x}\hat{y}\hat{z} \tag{23}$$

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. (23), integrals over the volume of an arbitrary hexahedral cell, Q, were evaluated by performing the integration over the reference element, $\hat{Q}$, as

$$\iiint_{Q} f(x,y,z)\,\mathrm{d}Q = \iiint_{\hat{Q}} f(\vec{r}(\hat{x},\hat{y},\hat{z}))\; J(\hat{x},\hat{y},\hat{z})\,\mathrm{d}\hat{Q} \tag{24}$$

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

$$J(\hat{x},\hat{y},\hat{z}) = \vec{X}(\hat{y},\hat{z}) \cdot \left(\vec{Y}(\hat{x},\hat{z}) \times \vec{Z}(\hat{x},\hat{y})\right) \tag{25}$$

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 were evaluated in a similar manner. For example, integrating $f(x, y, z)$ over a particular face, Γ_x, gives

$$\iint_{\Gamma_x} f(x,y,z)\,\mathrm{d}\Gamma_x = \iint_{\Gamma_{\hat{x}}} f(\vec{r}(\hat{x},\hat{y},\hat{z}))\; J_x(\hat{x},\hat{y},\hat{z})\,\mathrm{d}\Gamma_{\hat{x}} \tag{26}$$

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

$$J_x(\hat{x},\hat{y},\hat{z}) = \left|\vec{Y}(\hat{x},\hat{z}) \times \vec{Z}(\hat{x},\hat{y})\right| \tag{27}$$

The unit normal to the surface Γ_x at a particular location is related to the surface Jacobian by

$$\hat{n} = \frac{\vec{Y}(\hat{x},\hat{z}) \times \vec{Z}(\hat{x},\hat{y})}{\left|\vec{Y}(\hat{x},\hat{z}) \times \vec{Z}(\hat{x},\hat{y})\right|} \tag{28}$$

Using Eqs. (24) and (26), relations can easily be derived for the cell volume and areas of the individual cell faces.

3.3 CENO Reconstruction

In Eq. (22), numerical fluxes and sources are integrated over the control-volume to high order using Gauss quadrature but, from the semi-discrete finite-volume discretization, only cell-averaged quantities are known. The CENO method make use of a hybrid solution reconstruction procedure to produce a high-order representation within each cell which is then used to interpolate values of the primitive solution at each of the Gauss quadrature points [11–16].

3.3.1 k-Exact Reconstruction

The CENO spatial discretization scheme is based on the high-order k-exact least-squares reconstruction technique of Barth [50], which assumes that the solution within each cell can be represented by the following three-dimensional Taylor polynomial:

$$u_i^k(x,y,z) = \sum_{p=0}^{p+q+r\leq k} \sum_{q=0} \sum_{r=0} (x-x_i)^p (y-y_i)^q (z-z_i)^r D_{pqr} \tag{29}$$

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_{pqr} are the unknown polynomial coefficients. The summation indices, p, q and r, must always satisfy the condition that $(p+q+r) \leq k$.

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

$$\overline{u}_i = \frac{1}{V_i} \iiint_{Q_i} u_i^k(x,y,z)\, \mathrm{d}V \tag{30}$$

where $\overline{u}_i$ is the cell average. Additional constraints are introduced by the second condition, that is,

$$u_i^k(x,y,z) = u_{\text{exact}} + \mathcal{O}(l^{k+1}) \tag{31}$$

in the vicinity of cell i. The length scale, l, is defined as the maximum diameter of cell circumspheres in the vicinity of i. From Eq. (31), the reconstruction polynomial for cell i must also recover the averages of neighboring cells. That is,

$$\overline{u}_j = \frac{1}{V_j} \iiint_{Q_j} u_i^k(x,y,z)\, \mathrm{d}V + \mathcal{O}(l^{k+1}) \qquad \forall j \in S_{\text{neigh},i} \tag{32}$$

where $S_{\text{neigh},i}$ is the collection of neighboring cells in the vicinity of cell i. This property ensures that the difference between the predicted solution states at cell interfaces diminishes at a rate proportional to l^{k+1}.

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. (30)). For any type of mesh, the total number of unknown coefficients for a particular order is given by

$$N = \frac{1}{d!} \prod_{n=1}^{d} (k+n) \tag{33}$$

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 [13–16]. However, stencil selection for multi-dimensional unstructured mesh is not straightforward. The stencil is constructed here 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

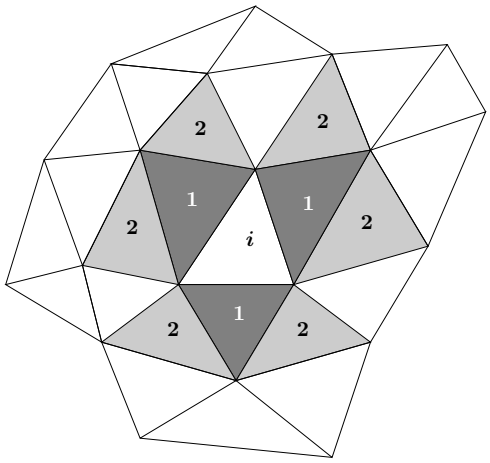


Fig. 3. Supporting stencil for reconstruction about cell i. First- and second-level neighbors are indicated in the figure.

the stencil is not biased in any direction and that the reconstruction remains reliable on poor quality meshes with high aspect ratio cells [13–16]. A sample of such a stencil construction about an unstructured cell i is illustrated in in Fig. 3 for the two-dimensional case. First, the layer of nearest neighbors surrounding cell i are selected, and then, if more neighbors are required, the next nearest neighbors are selected. This process is repeated until the stencil is deemed sufficient.

The constraints given by Eqs. (30) and (32) lead to an over-determined system of linear equations of the form:

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{34}$$

where $\mathbf{A}$ is the coefficient matrix, $\mathbf{x}$ is the vector of unknown polynomial coefficients, and $\mathbf{b}$ is a vector depending on control volume averages. Generally, a least-squares solution to the over-determined problem for $\mathbf{x}$ is sought for each variable in each control-volume.

The final form of Eq. (34) is obtained by subtracting Eq. (30) from Eq. (32), so that Eq. (30) is enforced exactly, and rearranging to give

$$\begin{bmatrix} w_{1i}\,\widehat{x^0y^0z^1}_{1i} & \cdots & w_{1i}\,\widehat{x^py^qz^r}_{1i} & \cdots & w_{1i}\,\widehat{x^ky^0z^0}_{1i} \\ \vdots & & \vdots & & \vdots \\ w_{ji}\,\widehat{x^0y^0z^1}_{ji} & \cdots & w_{ji}\,\widehat{x^py^qz^r}_{ji} & \cdots & w_{ji}\,\widehat{x^ky^0z^0}_{ji} \\ \vdots & & \vdots & & \vdots \\ w_{ni}\,\widehat{x^0y^0z^1}_{ni} & \cdots & w_{ni}\,\widehat{x^py^qz^r}_{ni} & \cdots & w_{ni}\,\widehat{x^ky^0z^0}_{ni} \end{bmatrix} \cdot \begin{bmatrix} D_{001} \\ \vdots \\ D_{pqr} \\ \vdots \\ D_{k00} \end{bmatrix} = \begin{bmatrix} w_{1i}(\overline{u}_1 - \overline{u}_i) \\ \vdots \\ w_{ji}(\overline{u}_j - \overline{u}_i) \\ \vdots \\ w_{ni}(\overline{u}_n - \overline{u}_i) \end{bmatrix} \tag{35}$$

where n is the number of neighbors in the stencil and w_{ji} are least-squares weights. The geometric coefficients are given by

$$\widehat{x^py^qz^r}_{ji} = \widetilde{x^py^qz^r}_{ji} - \overline{x^py^qz^r}_i \tag{36}$$

where

$$\widetilde{x^py^qz^r}_{ji} = \frac{1}{V_j}\int_{\Omega_j} (x - x_i)^{p_1}(y - y_i)^{p_2}(z - z_i)^{p_3}\,\mathrm{d}\Omega \tag{37}$$

$$\overline{x^py^qz^r}_i = \frac{1}{V_i}\int_{\Omega_i} (x - x_i)^{p_1}(y - y_i)^{p_2}(z - z_i)^{p_3}\,\mathrm{d}\Omega \tag{38}$$

Only the geometric moments about each individual control-volume, $\overline{x^py^qz^r}_i$, are actually computed and stored prior to solving Eq. (22). The remaining geometric coefficients are computed using a binomial expan-

sion [51], i.e., using

$$\widetilde{x^p y^q} z^r{}_{ji} = \sum_{a=0}^{p} \sum_{b=0}^{q} \sum_{c=0}^{r} \binom{p}{a} \binom{q}{b} \binom{r}{c} \cdot (x_j - x_i)^q \cdot (y_j - y_i)^b \cdot (z_j - z_i)^c \cdot \overline{x^{p-a} y^{q-b} z^{r-c}}_j \tag{39}$$

Inverse distance weighting is applied to each individual constraint equation in Eq. (35) to improve the locality of the reconstruction [52]. For solution reconstruction in cell i, the weight applied to row j is given by

$$w_{ji} = \frac{1}{|\vec{x}_j - \vec{x}_i|}, \tag{40}$$

where $\vec{x}_i$ and $\vec{x}_j$ are the postions of the respective cell centroids.

It is well established in the literature that the condition of the least-squares problem for the reconstruction coefficients can be an issue. The conditioning of Eq. (35) is improved herein by scaling the columns of $\mathbf{A}$, which effectively makes the condition number of the problem independent of the mesh size and aspect ratio [16, 19, 53]. Application of the column scaling procedure to $\mathbf{A}$ yields the modified linear system

$$\left(\mathbf{A}\mathbf{D}_\mathrm{C}\right)\left(\mathbf{D}_\mathrm{C}{}^{-1}\mathbf{x}\right) = \mathbf{b} \tag{41}$$

where $\mathbf{D}_\mathrm{C}$ is a diagonal matrix whose entries are the inverse of the largest absolute values of each column of $\mathbf{A}$, i.e.,

$$D_{\mathrm{C}jj} = \frac{1}{\max_{\forall i}\left(|A_{ij}|\right)} \tag{42}$$

Equation (41) is solved using either a QR factorization algorithm based on Householder transformations or singular value decomposition (SVD) [54]. Since the coefficient matrix, $\mathbf{A}$, and the scaling matrix, $\mathbf{D}_\mathrm{C}$, only depend on the mesh geometry, they only need to be inverted once [15, 16, 51, 55]. Thus, when SVD is used, the pseudoinverse of the scaled coefficient matrix is computed and stored prior to integrating Eq. (22), and polynomial coefficients are simply determined from the following matrix-vector product at each iteration:

$$\mathbf{x} = \mathbf{D}_\mathrm{C}\left(\mathbf{A}\mathbf{D}_\mathrm{C}\right)^\dagger \mathbf{b} \tag{43}$$

where $\dagger$ denotes the pseudoinverse and $\mathbf{D}_\mathrm{C}\left(\mathbf{A}\mathbf{D}_\mathrm{C}\right)^\dagger$ is the pre-computed and stored result of SVD. While incurring a storage cost, this procedure is considerably less computationally intensive than continually performing a full QR factorization or SVD decomposition for each control-volume at every iteration.

3.3.2 Reconstruction at Boundaries

To enforce conditions at physical boundaries, the least-squares reconstruction in adjacent control volumes is constrained without altering the order of accuracy of the reconstruction [14–16, 51, 56]. Constraints are implemented as general Robin boundary conditions

$$f\left(\vec{x}\right) = a\left(\vec{x}\right) f_\mathrm{D}\left(\vec{x}\right) + b\left(\vec{x}\right) f_\mathrm{N}\left(\vec{x}\right) \tag{44}$$

where $a\left(\vec{x}\right)$ and $b\left(\vec{x}\right)$ are weights that define the contribution of the Dirichlet, $f_\mathrm{D}\left(\vec{x}\right)$, and Neumann, $f_\mathrm{N}\left(\vec{x}\right)$, components, respectively. These coefficients are simply $(a, b) = (1, 0)$ for Dirichlet- and $(a, b) = (0, 1)$ for Neumann-type boundary conditions. The Dirichlet condition is simply expressed as

$$f_\mathrm{D}\left(\vec{x}_\mathrm{g}\right) = u^k\left(\vec{x}_\mathrm{g}\right) \tag{45}$$

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

$$\begin{aligned} f_\mathrm{N}\left(\vec{x}_\mathrm{g}\right) &= \vec{\nabla} u^k\left(\vec{x}_\mathrm{g}\right) \cdot \hat{n}_\mathrm{g} \\ &= \sum_{p+q+r=1}^{(p+q+r)\leq k} \sum \sum \Delta x^{p-1} \Delta y^{q-1} \Delta z^{r-1} \left[p\Delta y \Delta z n_x + q\Delta x \Delta z n_y + r\Delta x \Delta y n_z\right] D_{pqr} \end{aligned} \tag{46}$$

where $\Delta(\cdot) = (\cdot)_g - (\cdot)_i$ is the distance between the cell's centroid and the Gauss quadrature point.

Exact solutions to the boundary constraints described by Eq. (44) are sought. This adds linear equality constraints to the original least-squares problem (Eq. (35)). The new linear system with boundary constraints is then given by

$$\min \|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2 \qquad \text{subject to } \mathbf{C}\mathbf{x} = \mathbf{d} \tag{47}$$

where $\mathbf{C}$ and $\mathbf{d}$ are the coefficient matrix and solution vector for the boundary constraints, respectively. Gaussian elimination with full pivoting is first applied to remove the boundary constraints and the remaining least-squares problem is then solved as described in Section 3.3.1 above.

For inflow/outflow or farfield-type boundary conditions where the reconstructed variables are not related, the constraints can be applied separately to each variable. A separate least-squares problem with equality constraints can be set up and solved for each variable in and independent fashion. More complex boundary conditions involving linear combinations of solution variables, such as symmetry planes or inviscid solid walls where $\vec{v} \cdot \hat{n} = 0$, couple the polynomial coefficients for different variables. For these types of coupled boundary conditions, the full least squares problem for the coupled variables must be solved together in a coupled fashion [14–16, 51, 56].

To illustrate the latter, consider the symmetry boundary conditions used in the present study. Along symmetry planes,

$$\vec{\nabla}\rho \cdot \hat{n} = 0, \qquad \vec{v} \cdot \hat{n} = 0, \qquad \vec{\nabla}p \cdot \hat{n} = 0, \qquad \vec{\nabla}f \cdot \hat{n} = 0, \qquad \vec{\nabla}y_f \cdot \hat{n} = 0$$

where $\hat{n}$ is a unit vector normal to the boundary. All variables except for $\vec{v}$ are independent and were reconstructed separately using the boundary constraints defined in Eq. (44), but the three components of velocity are coupled via a linear combination of each other. The constraints for a zero normal velocity at the boundary are given by

$$u\left(\vec{x}_g\right) n_x\left(\vec{x}_g\right) + v\left(\vec{x}_g\right) n_y\left(\vec{x}_g\right) + w\left(\vec{x}_g\right) n_z\left(\vec{x}_g\right) = 0 \tag{48}$$

As such, the coupled least-squares problem for the unknown polynomial coefficients of the three velocity components is as follows:

$$\min \left\| \begin{bmatrix} \epsilon\mathbf{b}_u \\ \epsilon\mathbf{b}_v \\ \epsilon\mathbf{b}_w \end{bmatrix} - \begin{bmatrix} \mathbf{A}_u & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_v & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_w \end{bmatrix} \cdot \begin{bmatrix} \mathbf{x}_u \\ \mathbf{x}_v \\ \mathbf{x}_w \end{bmatrix} \right\|_2 \tag{49}$$

subject to

$$\begin{bmatrix} \mathbf{C}_u & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_v & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{C}_w \\ \multicolumn{3}{c}{\text{Coupled constraints, Eq. (48)}} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{x}_u \\ \mathbf{x}_v \\ \mathbf{x}_w \end{bmatrix} = \begin{bmatrix} \mathbf{d}_u \\ \mathbf{d}_v \\ \mathbf{d}_w \\ 0 \end{bmatrix} \tag{50}$$

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

3.3.3 Smoothness Indicator

After performing the k-exact reconstruction in every computational cell, the smoothness of the reconstruction is then assessed for every reconstructed variable to identify under-resolved solution content. The smoothness indicator proposed by Ivan and Groth [15, 16] is used for these purposes as given by

$$S_i = \frac{\sigma_i}{\max\left[(1-\sigma_i), \delta\right]} \frac{(\text{SOS} - \text{DOF})}{\text{DOF} - 1} \tag{51}$$

where σ is a smoothness parameter, δ is a tolerance to avoid division by zero (equal to 10^{-8}), DOF is the number of degrees of freedom and SOS is the size of the stencil. The smoothness parameter, σ, for a cell i

is given by

$$\sigma_i = 1 - \frac{\sum\limits_{\forall j \in S_{\text{neigh},i}} \left[u_j^k(\vec{x}_j) - u_i^k(\vec{x}_j)\right]^2}{\sum\limits_{\forall j \in S_{\text{neigh},i}} \left[u_j^k(\vec{x}_j) - \overline{u}_i\right]^2} \tag{52}$$

where u is the solution variable of interest and the subscript j refers to the cells in the reconstruction stencil. The numerator measures the ability of cell i's reconstruction to predict the values at neighboring cell centers, and the denominator acts to normalize σ. By definition, σ varies between negative infinity and one. A value of unity indicates that the solution is smooth whereas a small or negative value indicates a large variation in solution content within the reconstruction stencil.

A simple analysis, such as the one performed by Ivan and Groth [16], confirms the behavior of σ. It follows from Eq. (31) that

$$\sigma \approx 1 - \frac{\left[\mathcal{O}(h^{k+1})\right]^2}{\left[\mathcal{O}(h)\right]^2} \approx 1 - \mathcal{O}(h^{2k}) \tag{53}$$

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

$$\sigma \approx 1 - \frac{\left[\mathcal{O}(1)\right]^2}{\left[\mathcal{O}(1)\right]^2} \approx 1 - \mathcal{O}(1) \tag{54}$$

In the CENO approach, the high-order reconstructed solution is deemed smooth when the value of S is above a critical value, S_c. Previous studies found that values for S_c between 1000–5000 provided an excellent balance between stability and accuracy [13, 16]. Typical values for S in smooth regions tend to be orders of magnitude greater than these cutoff limits since the smoothness indicator grows rapidly as σ approaches unity. Unless otherwise specified, a value of S_c equal to 4000 was used here.

3.4 Limited Piecewise Linear Reconstruction

Even though most features of low-speed flows are relatively smooth, there are situations where discontinuities can occur, such as across flame fronts or fluid interfaces. Under-resolved solution content can also be problematic and lead to non-montonic or oscillatory solutions. In regions where $S < S_c$, monotonicity is preserved in the hybrid CENO approach by reverting the high-order k-exact reconstruction to a limited piecewise linear (k=1) reconstruction. Limiting was performed here using the procedure developed by Park et al. [57] specifically for multiple dimensions in conjunction with the slope limiter function of Venkatakrishnan [58]. The multi-dimensional limiter formulation was found necessary to correctly recover linear functions on unstructured mesh without unnecessary limiting or clipping of solution content [59]. The limited linear reconstruction is only applied in non-smooth regions to the inviscid terms in Eq. (22). The viscous fluxes and source terms are still evaluated using the full high-order representation since it is generally the inviscid terms that generate instabilities.

3.5 Inviscid Numerical Flux Evaluation

An upwind Godunov scheme is used in the evaluation of the inviscid numerical flux, $\vec{\mathbf{F}}$, at the cell faces. Given the left and right solution values, $\mathbf{W_L}$ and $\mathbf{W_R}$, the numerical flux at the cel interface is given by

$$\vec{\mathbf{F}} \cdot \hat{n} = \mathcal{F}\left(\mathbf{W_L}, \mathbf{W_R}, \hat{n}\right) \tag{55}$$

where $\mathcal{F}$ is a flux function which solves a Riemann problem in a direction aligned along the face normal, $\hat{n}$. The left and right solution values are determined using the hybrid CENO reconstruction procedure described in Section 3.3 above. As a result, the leading truncation error due to the inviscid operator is $\mathcal{O}\left(l^{k+1}\right)$ in smooth regions and reduces to $\mathcal{O}\left(l\right)$ near discontinuities and/or for under-resolved solution content.

The inviscid numerical flux, $\mathcal{F}$, is formulated based here on the pseudo-compressible system (i.e., steady physical-time and unsteady psuedo-time) using the eigensystem presented in Section 2.3. Similar to the

approach taken by Lee and Lee [45], the numerical flux was derived based on Roe's approximate Riemann solver [60, 61] using the eigensystem in Eq. (16). The numerical flux at the interface between two adjacent cells is given by

$$\mathbf{F}\left(\mathcal{R}\left(\mathbf{W_L}, \mathbf{W_R}\right)\right) = \frac{1}{2}\left(\mathbf{F_R} + \mathbf{F_L}\right) - \frac{1}{2}|\hat{\mathbf{A}}|\Delta\mathbf{W} \tag{56}$$

where $\mathbf{F_L}$ and $\mathbf{F_R}$ are the inviscid fluxes evaluated based on $\mathbf{W_L}$ and $\mathbf{W_R}$, respectively, $\Delta\mathbf{W} = \mathbf{W_R} - \mathbf{W_L}$, $|\hat{\mathbf{A}}| = \hat{\mathbf{X}}|\hat{\mathbf{\Lambda}}|\hat{\mathbf{X}}^{-1}$, $\hat{\mathbf{X}}$ 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 is used for $\hat{\mathbf{W}}$ in the present work.

3.6 Viscous Numerical Flux Evaluation

As proposed by Ivan and Groth [14, 16], the viscous fluxes at each quadrature point are evaluated by averaging the interface solution and gradients

$$\mathcal{G}\left(\mathbf{W_L}, \mathbf{W_R}, \vec{\nabla}\mathbf{W_L}, \vec{\nabla}\mathbf{W_R}\right) = \mathbf{F_v}\left\{\frac{1}{2}\left(\mathbf{W_L} + \mathbf{W_R}\right), \frac{1}{2}\left(\vec{\nabla}\mathbf{W_L}, \vec{\nabla}\mathbf{W_R}\right)\right\} \tag{57}$$

Because derivatives of the reconstructed polynomial are required, the leading truncation error due to the viscous operator is then only $\mathcal{O}\left(l^k\right)$. To match the truncation error introduced by the inviscid operator, the degree of the reconstruction polynomial is therefore increased by one (we call this k+1-exact reconstruction). However, the original Gauss quadrature rule selected for integrating over the cell faces is not changed. The preceding procedure maintains an order of accuracy of $\mathcal{O}\left(l^{k+1}\right)$ and was found to provide stable, although not strictly positive, discretizations of viscous fluxes without the occurrence of any even/odd decoupling phenomena [14, 16].

For piecewise-linear ($k = 1$) representations, second-order accuracy ($\mathcal{O}\left(l^{k+1}\right)$) of the viscous operator is achieved here without increasing the degree of the polynomial interpolant. In this case, the average gradient at the interface is evaluated using [62]

$$\vec{\nabla}\mathbf{W}_{i+1/2} = \left(\mathbf{W}_j - \mathbf{W}_i\right)\frac{\hat{n}}{\hat{n}\cdot\vec{r}_{ij}} + \left(\overline{\vec{\nabla}\mathbf{W}} - \overline{\vec{\nabla}\mathbf{W}}\cdot\vec{r}_{ij}\frac{\hat{n}}{\hat{n}\cdot\vec{r}_{ij}}\right) \tag{58}$$

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

$$\overline{\vec{\nabla}\mathbf{W}} = \chi\vec{\nabla}\mathbf{W}_i + (1-\chi)\vec{\nabla}\mathbf{W}_j \tag{59}$$

where

$$\chi = V_i/(V_i + V_j)$$

Equation (58) is second-order accurate if the gradient representation is also second-order accurate and thus k+1-exact reconstruction is not required.

3.7 Dual-Time Stepping Approach for Unsteady Problems

For the unsteady LES problems of interest, a dual-time-stepping-like approach is used for transient continuation [37–41]. The dual-time-stepping-like scheme solves a series of steady problems in pseudo time, τ, defined by

$$\mathbf{Q}\left(\mathbf{W}\right) = \mathbf{R}\left(\mathbf{W}\right) + \frac{\mathrm{d}\mathbf{U}\left(\mathbf{W}\right)}{\mathrm{d}t} = \mathbf{0} \tag{60}$$

where $\mathbf{Q}$ is the unsteady residual vector. The overbars denoting cell-averaged solution quantities have been here dropped for clarity.

Implicit time integration methods are considered here for the discretization of the physical time derivative, $\mathrm{d}\mathbf{U}/\mathrm{d}t$, due to their improved stability characteristics and ability to cope with numerically-stiff systems of equations. Even though more efficient implicit high-order time-marching schemes certainly exist — see, for example, the studies by Bijl et al. [63, 64] — the well-known backward difference formulas (BDF) family

Table 2. Coefficients for the BDF methods.

$$u^{n+1} + b^n u^n + b^{n-1} u^{n-1} + \cdots = a\Delta t^n \frac{\mathrm{d}u}{\mathrm{d}t}^{n+1}$$

Steps	a	b^n	b^{n-1}	b^{n-2}	b^{n-3}
1	1	-1			
2	$\frac{2}{3}$	$-\frac{4}{3}$	$\frac{1}{3}$		
3	$\frac{6}{11}$	$-\frac{18}{11}$	$\frac{9}{11}$	$-\frac{2}{11}$	
4	$\frac{12}{25}$	$-\frac{48}{25}$	$\frac{36}{25}$	$-\frac{16}{25}$	$\frac{3}{25}$

of methods is used to discretize the physical time derivative [65]. Application of the BDF time-marching schemes to Eq. (60) yields

$$\mathbf{Q}^{n+1} = \mathbf{R}^{n+1} + \frac{\mathbf{U}^{n+1} + b^n \mathbf{U}^n + b^{n-1}\mathbf{U}^{n-1} + \cdots}{a\Delta t^n} = \mathbf{0} \tag{61}$$

where n is the outer time level and Δt^n is the physical time step. The coefficients a and b depend on the particular BDF method and are provided in Table 2. Unless otherwise specified, a particular BDF method was chosen to match the order of accuracy of the spatial discretization. BDF1 was selected for $k = 0$, BDF2 was selected for $k = 1$ and $k = 2$, while BDF3 and BDF4 where selected for $k = 3$ and $k = 4$, respectively.

Either a full Newton method or a time-marching algorithm in pseudo time can be applied to solve the non-linear system of algebraic equations given by Eq. (61). Since Newton's method can fail when the initial solution estimates fall outside the radius of convergence, the latter approach was adopted. This approach effectively improves the conditioning of the non-linear problem and helps ensure global convergence for numerically-stiff systems. Re-introducing the pseudo time derivative, one can write

$$\mathbf{Q}^{n+1} + \mathbf{\Gamma}^{n+1} \frac{\mathrm{d}\mathbf{W}}{\mathrm{d}\tau}^{n+1} = \mathbf{0} \tag{62}$$

The particular temporal discretization applied to the pseudo-time derivative in Eq. (62) is arbitrary, since Eq. (62) is relaxed to a steady-state at each physical time step n. A simple first-order implicit Euler discretization is therefore applied to this derivative here because of its excellent stability characteristics, i.e., the method is A-stable. Application of the implicit Euler discretization to Eq. (62) gives

$$\mathbf{Q}^{(n+1,s+1)} + \mathbf{\Gamma}^{(n+1,s+1)} \frac{\mathbf{W}^{(n+1,s+1)} - \mathbf{W}^{(n+1,s)}}{\Delta\tau^s} = \mathbf{0} \tag{63}$$

where s is the inner iteration level and $\Delta\tau^s$ is the psuedo-time step.

Eq. (63) is itself now a non-linear system of algebraic equations and is then solved using a Jacobian-free inexact Newton method coupled with an iterative Krylov subspace linear solver. The particular implementation adopted here follows the algorithm originally developed by Groth *et al.* [66–68] specifically for use on large multi-processor parallel clusters. This algorithm was originally developed for two-dimensional multi-block body-fitted mesh and has since been extended to three-dimensional unstructured mesh by Charest et al. [11, 12].

3.7.1 Jacobian-Free Quasi-Newton-Krylov Method

Application of Newton's method to Eq. (63) followed by some rearrangement of terms gives

$$\left[\frac{1}{a\Delta t^n}\left(\frac{\partial \mathbf{U}}{\partial \mathbf{W}}\right)^{(n+1,s)} + \frac{\mathbf{\Gamma}^{(n+1,s)}}{\Delta\tau^s} + \left(\frac{\partial \mathbf{R}}{\partial \mathbf{W}}\right)^{(n+1,s)}\right] \Delta\mathbf{W}^{(n+1,s)} = -\mathbf{Q}^{(n+1,s)} \tag{64}$$

or

$$\mathbf{J}^{(n+1,s)} \Delta\mathbf{W}^{(n+1,s)} = -\mathbf{Q}^{(n+1,s)} \tag{65}$$

where $\mathbf{J}$ is the left-hand-side (LHS) coefficient matrix of the linear system of algebraic equations that must be solved at each Newton step. At each physical time step n, a steady solution to Eq. (60) is sought by iteratively solving the sequence of linear systems defined by Eq. (65). Given an initial estimate, $\mathbf{W}^{(n+1,0)}$, successively improved estimates at inner step s are given by

$$\mathbf{W}^{(n+1,s+1)} = \mathbf{W}^{(n+1,s)} + \Delta\mathbf{W}^{(n+1,s)} \tag{66}$$

The inner Newton iterations proceed until some desired reduction of the non-linear residual is achieved and the following condition is met:

$$\left\|\mathbf{Q}^{(n+1,s)}\right\|_2 < \epsilon \left\|\mathbf{Q}^{(n+1,0)}\right\|_2 \tag{67}$$

where $\epsilon = 10^{-3}$ is the convergence tolerance. While tighter tolerances — between 10^{-4} and 10^{-5} — may provide a better balance between accuracy and computation time in some cases [69], decreasing ϵ further provided no significant improvement of the predictions. In general, the level of convergence should match the accuracy of the physical time marching scheme.

As noted above, each step of Newton's method requires the solution of the linear problem defined by Eq. (65). This system tends to be relatively large, sparse, and non-symmetric, for which iterative methods are 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 *et al.* [70–73]. This is an Arnoldi-based solution technique which generates orthogonal bases of the Krylov subspace to determine the solution. The technique is particularly attractive because it can be used in Jacobian-free Newton-Krylov methods where $\mathbf{J}$ does not need to be explicitly formed. Only matrix-vector products are required at each iteration to create new trial vectors, which drastically reduces the required storage. Another advantage of GMRES is that iterations are terminated based on an estimate of the residual that does not require explicit construction of the intermediate residual vectors or solutions.

Based on the aforementioned advantages, a Jacobian-free, Krylov-based version of Newton's method was used in combination with GMRES for the solution of the linear equations. A restarted variant of the GMRES algorithm was used, GMRES(m), that minimizes storage by restarting every m iterations. Matrix-vector products, which are required by the GMRES algorithm, were computed using the following first-order approximation:

$$\mathbf{J}\mathbf{v} = \frac{\mathbf{Q}\left(\mathbf{W} + \varepsilon\mathbf{v}\right) - \mathbf{Q}\left(\mathbf{W}\right)}{\varepsilon} \tag{68}$$

where $\mathbf{v}$ is a Krylov vector and ε is a perturbation parameter. For ε, the following definition was used [74]:

$$\varepsilon = \frac{1}{N\|\mathbf{v}\|_2} \sum_{i=1}^{N} b|W_i| + b \tag{69}$$

where N is the dimension of the linear system and $b = 10^{-6}$ is an estimate of the square root of machine roundoff. An inexact convergence tolerance was applied to the linear solve of each inner non-linear iteration. It is related to the current non-linear residual, $\mathbf{Q}^{(n+1,s)}$, by

$$\left\|\mathbf{Q}^{(n+1,s)} + \mathbf{J}^{(n+1,s)}\Delta\mathbf{W}^{(n+1,s)}\right\|_2 < \zeta \left\|\mathbf{Q}^{(n+1,s)}\right\|_2,$$

where ζ is equal to 0.5 [75].

Right preconditioning of $\mathbf{J}$ was also performed to help facilitate the solution of the linear system without affecting the solution residual, $\mathbf{Q}$. The preconditioning takes the following form:

$$\left(\mathbf{J}\mathbf{M}^{-1}\right)\left(\mathbf{M}\,\Delta\mathbf{W}\right) = -\mathbf{Q} \tag{70}$$

where $\mathbf{M}$ is the preconditioning matrix. The right preconditioner is based on block incomplete lower-upper factorization with fixed fill levels, ILU(p), of the Jacobian [73]. For first- or second-order spatial discretizations, i.e., $k = 0$ or 1, the Jacobian was based on an analytical approximation of the employed discretization. However, higher-order approximations of the Jacobian were too complex and expensive to construct analytically. As such, a second-order approximation of $\mathbf{J}$ was also used for $k \geq 1$.

The level of fill, p, for the ILU preconditioner was specified between 0 and 2 to reduce storage requirements, although larger values of p typically offer improved convergence characteristics of the linear system at the expense of storage. To further reduce computational storage, reverse Cuthill-McKee (RCM) matrix reordering was used to permute the Jacobian's sparsity pattern into a banded matrix form with a small bandwith [76]. The RCM reordering was performed prior to ILU(p) factorization to make the Jacobian more diagonally dominant and improve the accuracy/stability of the factorization. This *a priori* reordering also decreases the number of non-zero entries in the factorized result, which reduces the storage requirements for $\mathbf{M}$ and improves the efficiency of matrix operations involving $\mathbf{M}$.

3.7.2 Parallel Implementation via Domain Decomposition

A parallel implementation of the proposed solution method for the governing equations has been carried out in order to take advantage of modern distributed-memory computer architectures. This was accomplished by subdividing the computational domain, distributing the computational cells among the available processors, and solving Eq. (64) on each computational sub-domain simultaneously. Subdivision was performed using a standard and well-established parallel graph partitioning algorithm, called Parmetis [77]. Ghost cells, which surround each individual sub-domain and overlap cells on neighboring sub-domains, were used to share solution content via inter-partition communication with the message passing interface (MPI) library [78].

Domain decomposition effectively distributes the coefficient matrix of the linear system, $\mathbf{J}$, among the available processors. However, the localized ILU preconditioner that was used here does not include the matrix information from other domains. An additive Schwarz global preconditioner helped account for this loss of information from domain decomposition. The additive Schwarz global preconditioner is defined as follows:

$$\mathbf{M}^{-1} = \sum_{q=1}^{N_\mathrm{d}} \mathbf{B}_q^T \mathbf{M}_q^{-1} \mathbf{B}_q \tag{71}$$

where N_d is the number of sub-domains, $\mathbf{B}_q$ is the gather matrix for the qth domain, and $\mathbf{M}_q$ is the local preconditioner matrix based on ILU(p) factorization. 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.

3.7.3 Stability of the Pseudo-Compressible System

In the quasi-Newton method described above, 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 was determined by

$$\Delta\tau_i^s \leq \mathrm{CFL} \cdot \min\left[\frac{\Delta_i}{\max(\boldsymbol{\lambda}_i)},\ \frac{\bar{\rho}_i \Delta_i^2}{\mu_i}\right] \tag{72}$$

where CFL is a constant greater than zero which determines the time step size. A value for CFL of 1000 was used here.

3.7.4 Stability of the Unsteady System

The stability of the unsteady system is governed by the physical time step size, which is also determined based on a CFL and viscous Von Neumann stability criteria. The maximum global time step size for the entire computational domain is given by

$$\Delta t^n \leq \mathrm{CFL}_\mathrm{phys} \cdot \min_{\forall i}\left(\frac{\Delta_i}{u_i^2 + v_i^2 + w_i^2},\ \frac{\bar{\rho}_i \Delta_i^2}{\mu_i}\right) \qquad i = 1, 2, \ldots, N_\mathrm{c} \tag{73}$$

where $\mathrm{CFL}_\mathrm{phys}$ is the physical CFL number.

It is well known that BDFm methods with $m > 2$ are conditionally stable. A stability analysis of the BDF methods applied to the one-dimensional form of the linear advection equation is shown in Fig. 4. The linear advection equation is defined as

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = 0 \tag{74}$$

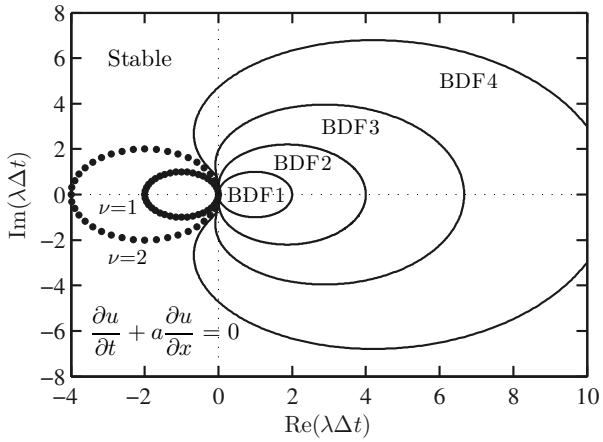


Fig. 4. Stability diagram for the BDF methods applied to the linear advection equation. Lines enclose the unstable regions for each method; symbols are the eigenvalues for a first-order upwind spatial discretization with uniform mesh spacing Δx and periodic boundaries. λ represents the eigenvalues, Δt is the time step size, a is the wave speed, and $\nu = a\Delta t/\Delta x$ is the CFL number.

where u is the unknown scalar and a is the wave speed. A first-order upwind discretization was applied to this system with uniform mesh spacing Δx and periodic boundary conditions. The eigenvalues, λ, are also shown in Fig. 4. 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 (i.e., sources) 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 the governing equations (Eq. (16)) involve a system of reaction-diffusion-advection equations. Additionally, the time step size for LES of turbulent flows is typically governed by the desired accuracy, not the stability requirements. Large-scale motions of the fluid need to be resolved for accurate LES, which usually requires fine grids and smaller time step sizes. In any event, no instabilities related to the combination of high-order BDF and CENO methods were encountered in this research.

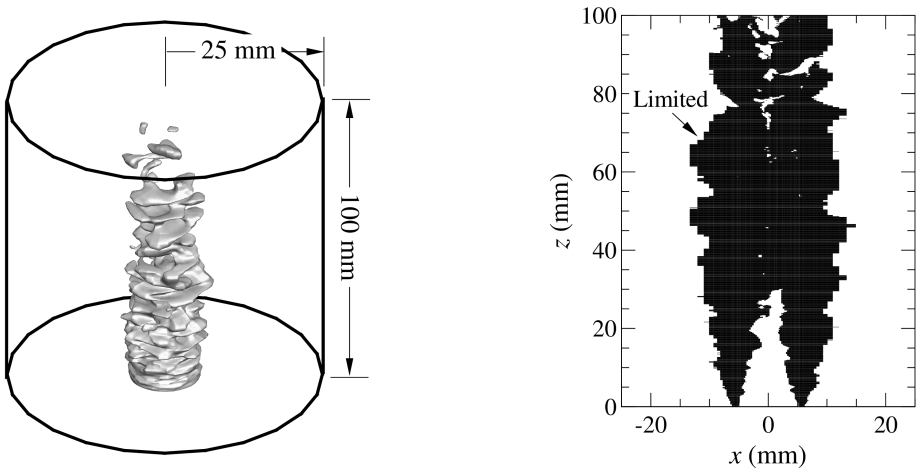
4 Results For Three-Dimensional Unstructured Mesh

The proposed finite-volume scheme for LES of reactive flows has been assessed in terms of both accuracy and stability through its application to a laboratory-scale turbulent fully-premixed flame. 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.

4.1 Premixed Bunsen Flame

The lean turbulent premixed methane flame of Yuen and Gülder [79, 80] was chosen for validation purposes in this research. It was modelled using both the second- and fourth-order numerical schemes that were described previously in Section 3.1. The flame — which is a piloted, conical jet flame — is generated using an axisymmetric Bunsen-type burner with an inner nozzle diameter of 11.2 mm. Yuen and Gülder [79, 80] employed various perforated plates to generate upstream turbulence over a range of intensities. A co-annular pilot flame surrounds the central jet in order to stabilize the main flame. The original investigators used Rayleigh scatting to capture planar images of the flame and particle image velocimetry of the cold flow to measure the instantaneous velocity field.

The particular flame configuration studied here is an atmospheric lean premixed methane-air flame with an equivalence ratio of 0.7. It is the same configuration that Hernández-Pérez et al. [81] and Shahbazian et al. [82] employed for their detailed numerical analysis. The turbulence intensity and length scale at the burner exit are $u'/s_L = 14.38$ and $L = 1.79$ mm, where s_L is the laminar flame speed. The temperature and



(a) computational domain (b)

Fig. 5. LES of lean premixed bunsen flame showing: (a) computational domain; and (b) and instantaneous snap shot at $t=6$ ms of computational cells flagged as being non-smooth by smoothness indicator based on temperature.

mean velocity of the reactants at the burner exit plane is 300 K and 15.58 m/s, respectively. This flame lies in the thickened wrinkled flame or thin reaction zone regime and has a turbulent Reynolds number of 324.

Both the second- and fourth-order numerical schemes described in Section 3.1 were applied to LES of this Bunsen flame. A cylindrical computational domain with a diameter of 0.05 m and height of 0.1 m was subdivided into approximately 1.8 million hexahedral cells. The computation domain is depicted in Fig. 5(a) and, for informational purposes, an instantaneous snap shot at $t = 6$ ms of computational cells flagged in the fourth-order results as being non-smooth by the smoothness indicator based on temperature are given in Fig. 5(b). It is clear from the latter that the rapid variations of thermodynamic quantities through the flame front are somewhat under-resolved on the current mesh and limiting is therefore being applied due to hyperbolic fluxes in these regions. Nevertheless, as will be shown, the fourth-order scheme appears to yield superior LES solutions. It is important to note that, due to the limitations in the scope of this paper, thorough mesh and time step convergence studies, computational costs, and comparison of high- and low-order results were not been performed: the fourth-order predictions are merely compared to the second-order results on the same mesh with the same time step. More thorough comparisons of LES results and assessments of computational costs will be the subjects of future follow-on studies.

In the mathematical model of the premixed Bunsen flame, the pilot flame was approximated by hot combustion products entering the domain at a velocity of 16.81 m/s. Turbulent velocity fluctuations that matched the experimental measurements were prescribed at the inlet using the synthetic eddy method developed by Jarrin et al. [83]. A time step size of 12.5 μs was employed for both the second and fourth-order calculations, which corresponds to a $\mathrm{CFL}_{\mathrm{phys}} \approx 0.8$. The total simulated physical time in each case was 20 ms (0.02 s), corresponding to a total of 4,000 time steps. Note that a quasi-steady flame is rapidly established in this case after just 4-5 ms.

The unfiltered instantaneous measurements are compared with the Favre-filtered predictions for temperature in Fig. 6. Similar structures are observed in both the experiments and predictions. The predictions underestimate the degree of wrinkling, with the second-order predictions displaying significantly less wrinkling than the fourth-order predictions. The fourth-order predictions have a thicker flame front and display more penetration of hot combustion products into the inner core of fresh reactants. Low in the flame, the two sets of predictions are similar. However, pockets of fresh unburned reactants begin to break off much lower in the fourth-order predictions. As a result, the fourth-order scheme predicts a shorter flame which agrees better with the experimental measurements. The shorter flame structure predicted by the fourth-order LES results is quite evident in comparison of the measured and predicted time-averaged distributions of the flow temperature depicted in Fig. 7.

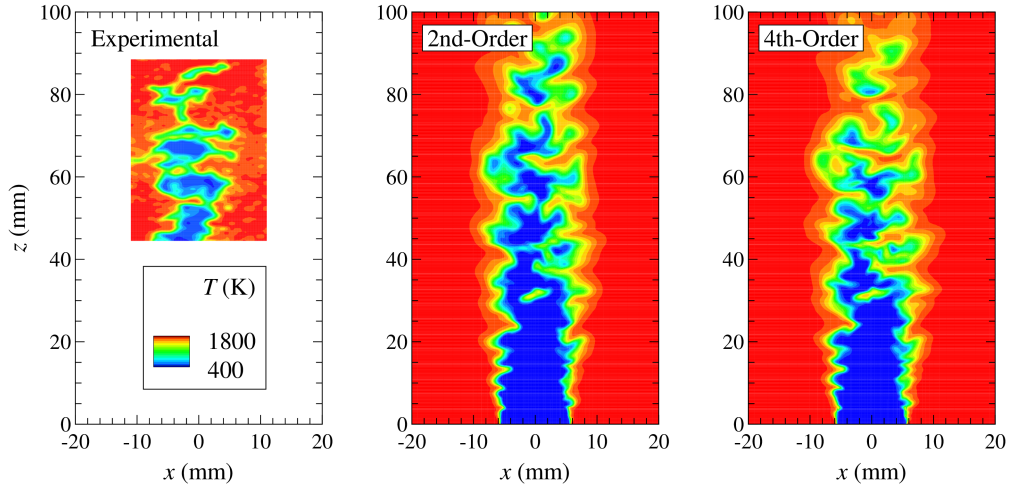


Fig. 6. LES of lean premixed bunsen flame showing a comparison of measured and predicted instantaneous distributions of the flow temperature at $t=6$ ms. Experimental results are unfiltered in the figure.

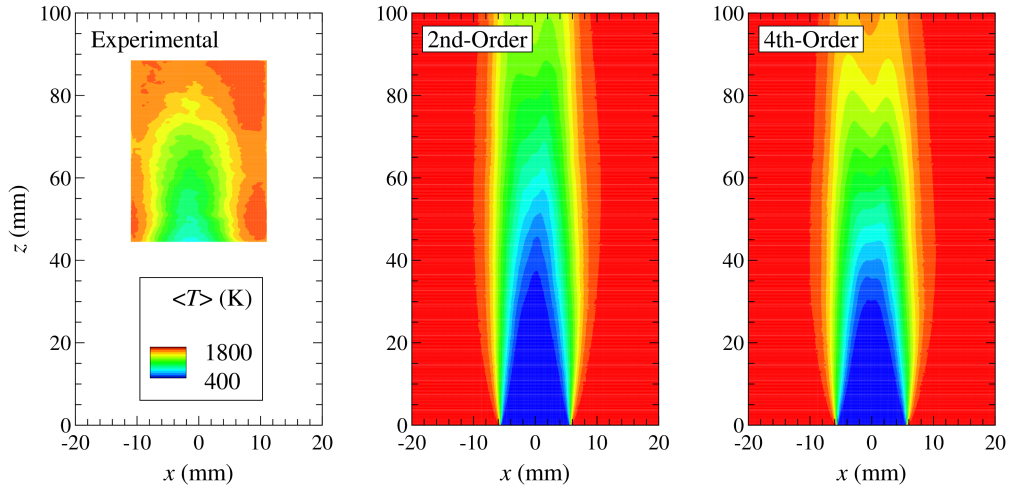


Fig. 7. LES of lean premixed bunsen flame showing a comparison of measured and predicted time-averaged distributions of the flow temperature.

The differences in predicted flame height are even more evident in Fig. 8, which illustrates the time-averaged flame surface. Here, the flame surface was defined by the location where the time-averaged value of the progress variable based on temperature was equal to 0.5. The progress variable is defined in this research as

$$c = \frac{T - T_{\mathrm{u}}}{T_{\mathrm{bu}} - T_{\mathrm{u}}} \tag{75}$$

where T_{u} and T_{b} are the temperatures of the unburned and burned gases. The second-order scheme greatly over-predicts the measured flame height while the fouth-order scheme slightly under-predict the average height. Overall, the fourth-order results show excellent agreement with the measured flame height.

The two-dimensional (2D) flame surface density (FSD) was also extracted from the planar Rayleigh scattering images and is compared with the similarly processed LES predictions in Fig. 9(a). As in Fig. 8, the flame surface was defined as the location where $\langle c \rangle = 0.5$. The values of the 2D FSD were determined using the method developed by Shepherd [84]; the same procedure was applied to both the experimental measurements and LES calculations. Here, the instantaneous flame front edges are superimposed onto the time-averaged progress variable field, $\langle c \rangle$, and the FSD is computed based on the ratio of the flame length

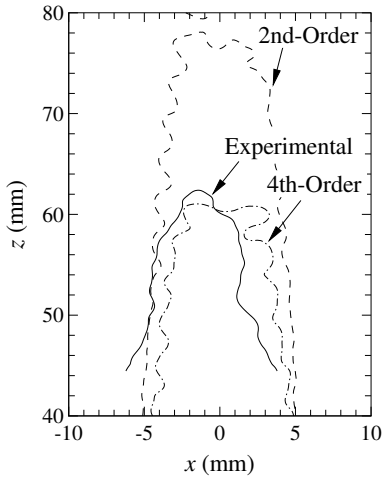
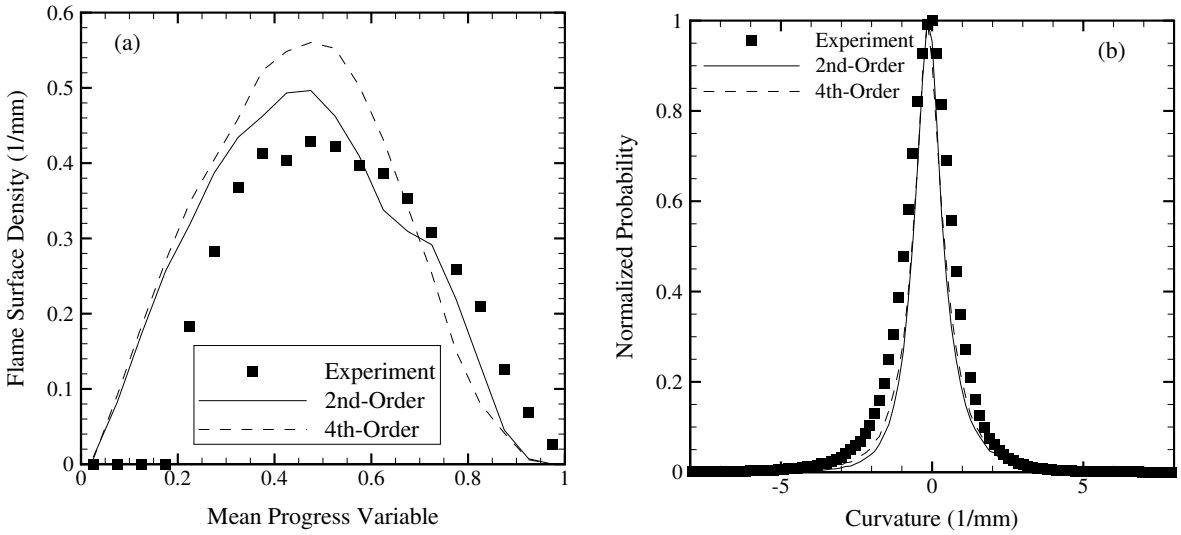


Fig. 8. LES of lean premixed bunsen flame showing a comparison between measured and predicted average flame heights for the Yuen and Gülder [79, 80] flame. The flame height is taken to be defined by the location of the $\langle c \rangle = 0.5$ contour line in each case.



(a) 2D FSD (b) 2D flame curvature

Fig. 9. LES of lean premixed bunsen flame showing comparisons between: (a) measured and predicted 2D flame surface density; and (b) normalized PDF of 2D curvature for the Yuen and Gülder [79, 80] flame. Results correspond to a flame surface defined by a value for the progress variable of $c = 0.5$.

to the flame area. For direct and meaningful comparison with the numerical predictions, the measurements for the instantaneous temperature were first filtered using a moving average filter with a width equal to the value used in the computations.

It can be seen from Fig. 9(a) that both low- and high-order LES predictions of the FSD reproduce qualitatively the experimentally-observed trends. The FSD is zero in the reactants and products, and peaks near a progress variable equal to $c = 0.5$. However, the predicted peak FSD obtained from both numerical schemes is higher than the measured peak. The predictions provided by the second-order scheme are in better agreement with the measurements. They over-predict the experimental value by a factor of approximately 1.2 whereas the fourth-order results over-predict the peak by a factor of 1.3. A higher predicted peak value for the FSD is expected from the fourth-order scheme since a larger degree of flame wrinkling was observed

by this scheme in Fig. 6. This general over-prediction of the FSD is thought to be attributed to errors introduced by the eddy-break-up turbulent combustion model.

Predicted and measured normalized probability density functions (PDFs) for the flame curvature are compared in Fig. 9(b). These PDFs were computed for a flame surface corresponding to $c = 0.5$. The predicted curvature PDFs agree favorably with the experiments, but both numerical schemes predict a slightly narrower PDF than observed in the experiments. As with the predictions for the 2D FSD, these discrepancies in predicted flame curvature statistics are attributed to errors introduced by the turbulent combustion model.

In terms of algorithm robustness and numerical stability, the fourth-order scheme was able to accurately and robustly capture the nearly discontinuous flame surface without generating any unphysical numerical oscillations as illustrated in Fig. 6. Additionally, the smoothness indicator correctly identified regions where large turbulent fluctuations were under-resolved by the employed computational mesh. Refer again to Fig. 5(b). This ensured the reliable convergence of the Newton-Krylov algorithm at each physical time step. In terms of computational costs, the fourth-order method was about 6–10 times more expensive per time step than the second-order scheme for this case. Further assessment of the trade-offs between accuracy and cost are certainly warranted.

5 Conclusions

A high-order finite-volume solver has been developed for the LES of low-Mach-number turbulent reactive flows on unstructured meshes using the pseudo-compressibility approach. The scheme is a high-order variant of cell-centered, ENO-like, finite-volume, methods and uses the hybrid CENO reconstruction method recently proposed by Ivan and Groth [13–16] and represents an extension of the CENO approach developed by Charest et al. [11, 12] for low-speed, non-reactive, viscous flows on three-dimensional unstructured mesh.

The proposed finite-volume formulation was validated by comparing predicted LES solutions of a laboratory-scale premixed turbulent flame to experimental measurements for temperature, flame surface density, flame curvature, and flame height. Predictions were obtained using both the second- and fourth-order variants of the proposed algorithm on the same mesh. Overall, both schemes predicted many of the experimentally observed trends. However, the fourth-order scheme provided significantly improved predictions for average flame height and degree of wrinkling. Discrepancies between predictions and measurements are largely attributed to the eddy-break-up turbulent combustion model.

Robust and monotone fourth-order LES solutions were maintained, even in regions with sharp gradients such as across flame fronts. 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. 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, the proposed finite-volume scheme is able to obtain accurate solutions to the equations governing low-Mach-number turbulent reactive flow.

While the validity and promise of the proposed Newton-Krylov-Schwarz, high-order CENO algorithm for LES of reactive flows on unstructured meshes has been demonstrated, there is still much work to be done. Future research will involve the further development and validation of the solution method for three-dimensional unstructured meshes. This will include the application of the pseudo-compressibility approach to more complex reactive flows and incorporating a multi-block adaptive mesh refinement (AMR) algorithm [85–88]. 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 [13–16]. Thorough mesh and time step convergence studies, computational cost assessments, and more complete comparisons of high- and low-order LES results are a high priority and will also be carried out for a range of flames. More accurate turbulent combustion models will also be considered in the future, such as the presumed conditional moment, flame prolongation of ILDM flamelet-based model [81, 89].

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. 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.C. Wilcox. *Turbulence Modeling for CFD*. DCW Industries, 3rd edition, 2006.
[2] S.B. Pope. *Turbulent Flows*. Cambridge University Press, 2000.
[3] S. Pirozzoli. *J. Comput. Phys.*, 219(2):489–497, 2006.
[4] A. Harten, B. Engquist, S. Osher, and S.R. Chakravarthy. *J. Comput. Phys.*, 71(2):231–303, 1987.
[5] W.J. Coirier and K.G. Powell. *AIAA J.*, 34(5):938–945, May 1996.
[6] Z. Wang, K. Fidkowski, R. Abgrall, F. Bassi, D. Caraeni, A. Cary, H. Deconinck, R. Hartmann, K. Hillewaert, H. Huynh, N. Kroll, G. May, P.O. Persson, B. van Leer, and M. Visbal. *Int. J. Numer. Meth. Fluids*, 72(8):811–845, 2013.
[7] V. Moureau, G. Lartigue, Y. Sommerer, C. Angelberger, O. Colin, and T. Poinsot. *J. Comput. Phys.*, 202(2):710–736, 2005.
[8] O. Desjardins, G. Blanquart, G. Balarac, and H. Pitsch. *J. Comput. Phys.*, 227(15):7125–7159, 2008.
[9] J.H. Chen, A. Choudhary, B. de Supinski, M. DeVries, E.R. Hawkes, S. Klasky, W.K. Liao, K.L. Ma, J. Mellor-Crummey, N. Podhorszki, R. Sankaran, S. Shende, and C.S. Yoo. *Comput. Sci. Discov.*, 2(1): 015001, 2009.
[10] P. Tsoutsanis, A.F. Antoniadis, and D. Drikakis. *J. Comput. Phys.*, 256(1):254–276, 2014.
[11] M.R.J. Charest, C.P.T. Groth, and P.Q. Gauthier. A high-order central ENO finite-volume scheme for three-dimensional turbulent reactive flows on unstructured mesh. In *21st AIAA Computational Fluid Dynamics Conference*, San Diego, California, June 24–27 2013.
[12] M.R.J. Charest, C.P.T. Groth, and P.Q. Gauthier. *Commun. Comput. Phys.*, 2013. Submitted for publication.
[13] L. Ivan and C.P.T. Groth. AIAA paper 2007-4323, 2007.
[14] L. Ivan and C.P.T. Groth. AIAA paper 2011-0367, 2011.
[15] L. Ivan and C.P.T. Groth. *Commun. Comput. Phys.*, 2013. submitted for publication.
[16] L. Ivan and C.P.T. Groth. *J. Comput. Phys.*, 257:830–862, 2013.
[17] A. Susanto, L. Ivan, H. De Sterck, and C.P.T. Groth. *J. Comput. Phys.*, 250(1):141 – 164, 2013.
[18] G.S. Jiang and C.W. Shu. *J. Comput. Phys.*, 126(1):202–228, 1996.
[19] O. Friedrich. *J. Comput. Phys.*, 144:194–212, 1998.
[20] C. Hu and C.W. Shu. *J. Comput. Phys.*, 150:97–127, 1999.
[21] R. Abgrall. *J. Comput. Phys.*, 114:45–58, 1994.
[22] T. Sonar. *Comp. Meth. Appl. Mech. Eng.*, pages 140–157, 1997.
[23] D. Stanescu and W. Habashi. *AIAA J.*, 36:1413–1416, 1998.
[24] A. Haselbacher. AIAA paper 2005-0879, 2005.
[25] J. Smagorinski. *Monthly Weather Review*, 91(3):99–165, 1979.
[26] E. Balaras, C. Benocci, and U. Piomelli. *AIAA J.*, 34(6):1111–1119, 1996.
[27] U. Piomelli and E. Balaras. *Ann. Rev. Fluid Mech.*, 34:349–374, 2002.
[28] E.V. Driest. *J. Appl. Sci.*, 23(11):1007–1011, 1956.
[29] F. Nicoud and F. Ducros. *Flow Turbul. Combust.*, 62(3):183–200, 1999.
[30] B.F. Magnussen and B.H. Hjertager. *Proc. Combust. Inst.*, 16(1):719–729, 1977.
[31] D.B. Spalding. *Proc. Combust. Inst.*, 13(1):649–657, 1971.
[32] C.K. Westbrook and F.L. Dryer. *Combust. Sci. Tech.*, 27:31–43, 1981.
[33] A.J. Chorin. *J. Comput. Phys.*, 2(1):12–26, 1967.
[34] J.L. Steger and P. Kutler. *AIAA J.*, 15(4):581–590, 1977.
[35] S.L. Chang and K.T. Rhee. *Int. Commun. Heat Mass Transfer*, 11(5):451–455, 1984.

[36] E. Turkel. *J. Comput. Phys.*, 72(2):277–298, 1987.
[37] A. Jameson. AIAA paper 1991-1596, 1991.
[38] C.L. Merkle and M. Athavale. AIAA paper 87-1137, 1987.
[39] W.Y. Soh and J.W. Goodrich. *J. Comput. Phys.*, 79(1):113–134, 1988.
[40] S.E. Rogers and D. Kwak. *AIAA J.*, 28(2):253–262, 1990.
[41] S.E. Rogers, D. Kwak, and C. Kiris. *AIAA J.*, 29(4):603–610, 1991.
[42] A.G. Malan, R.W. Lewis, and P. Nithiarasu. *Int. J. Numer. Meth. Engin.*, 54(5):695–714, 2002.
[43] A.G. Malan, R.W. Lewis, and P. Nithiarasu. *Int. J. Numer. Meth. Engin.*, 54(5):715–729, 2002.
[44] Z. Qian, J. Zhang, and C. Li. *Science China: Physics, Mechanics and Astronomy*, 53(11):2090–2102, 2010.
[45] H. Lee and S. Lee. *Int. J. Aeronaut. Space Sci.*, 12(4):318–330, 2011.
[46] R.L. Naff, T.F. Russell, and J.D. Wilson. Shape functions for three-dimensional control-volume mixed finite-element methods on irregular grids. In *Developments in Water Science*, volume 47, pages 359–366. Elsevier, 2002.
[47] R. Naff, T. Russell, and J. Wilson. *Computat. Geosci.*, 6:285–314, 2002.
[48] L. Ivan, H. De Sterck, A. Susanto, and C.P.T. Groth. *J. Comput. Phys.*, submitted, 2014.
[49] C.A. Felippa. *Eng. Computation.*, 21(8):867–890, 2004.
[50] T.J. Barth. AIAA Paper 93-0668, 1993.
[51] C.F. Ollivier-Gooch and M. Van Altena. *J. Comput. Phys.*, 181(2):729–752, 2002.
[52] D.J. Mavriplis. AIAA paper 2003-3986, 2003.
[53] A. Jalali and C. Ollivier-Gooch. AIAA Paper 2013-2565, 2013.
[54] C.L. Lawson and R.J. Hanson. *Solving least squares problems*. Prentice-Hall, 1974.
[55] L. Ivan. *Development of high-order CENO finite-volume schemes with block-based adaptive mesh refinement*. PhD thesis, University of Toronto, 2011.
[56] A. Nejat and C. Ollivier-Gooch. *J. Comput. Phys.*, 227(4):2582–2609, 2008.
[57] J.S. Park, S.H. Yoon, and C. Kim. *J. Comput. Phys.*, 229(3):788–812, 2010.
[58] V. Venkatakrishnan. AIAA Paper 93-0880, 1993.
[59] M. Berger, M.J. Aftosmis, and S.M. Murman. Analysis of slope limiters on irregular grids. In *43rd AIAA Aerospace Sciences Meeting*, Reno, NV, Jan. 10-13 2005. AIAA paper 2005-0490.
[60] P.L. Roe. *J. Comput. Phys.*, 43:357–372, 1981.
[61] P.L. Roe and J. Pike. Efficient construction and utilisation of approximate Riemann solutions. In R. Glowinski and J.L. Lions, editors, *Computing Methods in Applied Science and Engineering*, volume VI, pages 499–518, Amsterdam, 1984. North-Holland.
[62] S.R. Mathur and J.Y. Murthy. *Numer. Heat Transfer, Part B*, 31(2):195–215, 1997.
[63] H. Bijl, M.H. Carpenter, and V.N. Vatsa. AIAA Paper 2001-2612, 2001.
[64] H. Bijl, M.H. Carpenter, V.N. Vatsa, and C.A. Kennedy. *J. Comput. Phys.*, 179:313–329, 2002.
[65] H. Lomax, T.H. Pulliam, and D.W. Zingg. *Fundamentals of Computational Fluid Dynamics*. Springer, New York, 2003.
[66] C.P.T. Groth and S.A. Northrup. Parallel implicit adaptive mesh refinement scheme for body-fitted multi-block mesh. In *17th AIAA Computational Fluid Dynamics Conference*, Toronto, Ontario, Canada, 6-9 June 2005. AIAA paper 2005-5333.
[67] M.R.J. Charest, C.P.T. Groth, and Ö.L. Gülder. *Combust. Theor. Modelling*, 14(6):793–825, 2010.
[68] M.R.J. Charest, C.P.T. Groth, and Ö.L. Gülder. *J. Comput. Phys.*, 231(8):3023–3040, 2012.
[69] M. Tabesh and D.W. Zingg. Efficient implicit time-marching methods using a Newton-Krylov algorithm. In *47th AIAA Aerospace Sciences Meeting and Exhibit*, Orlando, Florida, 5-8 January 2009. AIAA paper 2009-0164.
[70] Y. Saad and M.H. Schultz. *SIAM J. Sci. Stat. Comput.*, 7(3):856–869, 1986.
[71] Y. Saad. *SIAM J. Sci. Stat. Comput.*, 10(6):1200–1232, 1989.
[72] P.N. Brown and Y. Saad. *SIAM J. Sci. Stat. Comput.*, 11(3):450–481, 1990.
[73] Y. Saad. *Iterative Methods for Sparse Linear Systems*. PWS Publishing Company, Boston, 1996.
[74] D.A. Knoll and D.E. Keyes. *J. Comput. Phys.*, 193(2):357–397, 2004.
[75] R.S. Dembo, S.C. Eisenstat, and T. Steihaug. *SIAM J. Numer. Anal.*, 19(2):400–408, 1982.
[76] E. Cuthill and J. McKee. Reducing the bandwidth of sparse symmetric matrices. In *Proceedings of the 1969 24th National Conference*, ACM '69, pages 157–172, New York, NY, USA, 1969. ACM.

[77] G. Karypis and K. Schloegel. http://www.cs.umn.edu/~metis, 2011.
[78] W. Gropp, E. Lusk, and A. Skjellum. *Using MPI*. MIT Press, Cambridge, Massachussets, 1999.
[79] F.T. Yuen and Ö.L. Gülder. *AIAA J.*, 47(12):2964–2973, 2009.
[80] F.T. Yuen and Ö.L. Gülder. *Proc. Combust. Inst.*, 32:1747–1754, 2009.
[81] F.E. Hernández-Pérez, F.T.C. Yuen, C.P.T. Groth, and Ö.L. Gülder. 33:1365–1371, 2011.
[82] N. Shahbazian, C.P.T. Groth, and Ö.L. Gülder. AIAA Paper 2011-0781, 2011.
[83] N. Jarrin, S. Benhamadouche, D. Laurence, and R. Prosser. *Int. J. of Heat Fluid Fl*, 27(4):585–593, 2006.
[84] I. Shepherd. *Proc. Combust. Inst.*, 26:373–379, 1996.
[85] J.S. Sachdev, C.P.T. Groth, and J.J. Gottlieb. *Int. J. Comput. Fluid Dyn.*, 19(2):159–177, 2005.
[86] X. Gao and C.P.T. Groth. *Int. J. Comput. Fluid Dyn.*, 20(5):349–357, 2006.
[87] X. Gao and C.P.T. Groth. *J. Comput. Phys.*, 229(9):3250–3275, 2010.
[88] X. Gao, S. Northrup, and C.P.T. Groth. *Prog. Comput. Fluid. Dy.*, 11(2):76–95, 2011.
[89] P. Domingo, L. Vervisch, S. Payet, and R. Hauguel. *Combust. Flame*, 143(4):566–586, 2005.