AIAA 2013-2567

A High-Order Central ENO Finite-Volume Scheme for Three-Dimensional Turbulent Reactive Flows on Unstructured Mesh

M. R.J. Charest* and C. P.T. Groth†

University of Toronto Institute for Aerospace Studies

4925 Dufferin Street, Toronto, Ontario, Canada, M3H 5T6

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 approach. Large-eddy simulations of a laboratory-scale turbulent flame is 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 complex turbulent reactive flows.

I. Introduction

Many practical flows that are relevant to science and engineering present significant challenges to model numerically, primarily due to the wide range of scales and physical/chemical phenomena involved. Such flows include, but are not limited to, multiphase, turbulent, and combusting flows encountered in propulsion systems (e.g., gas turbine engines and solid propellant rocket motors). Nonetheless, computational fluid dynamics (CFD) has proven to be an important enabling technology in these areas. But for most of these practical flows, there is currently a limit on what can 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 flows is not possible. Therefore, mathematical models of these complex flows must rely heavily on engineering approximations and sophisticated numerical methods to represent the underlying physics and ensure that computations remain tractable.

The primary approaches for modelling turbulent flows are Reynolds-Averaged Navier-Stokes (RANS) and Large-Eddy Simulation (LES)[1,2]. RANS-based methods are more commonly used in engineering applications due to their computational efficiency. However, they cannot resolve any part of the turbulent fluctuations since information regarding fluctuations is lost through the averaging process. LES separates the turbulent scales via a low-pass filtering process applied to the governing equations and, as a result, provides information about the large-scale turbulent fluctuations. As such, LES is considered an intermediate between DNS and RANS since LES resolves the large-scale motions of the flow and models the small scales.

Despite the physical simplifications employed by LES, LES of full-scale practical turbulent reactive flows still requires significant computational resources. One promising numerical technique to reduce the computational cost associated with LES calculations is the use of high-order methods (i.e., methods higher than second order). High-order methods are more accurate than standard low-order methods (i.e, methods up to second order) and also offer improved numerical efficiency since fewer computational cells are required to achieve a desired level of accuracy[3].

*Post-Doctoral Fellow, UTIAS, charest@utias.utoronto.ca, Member AIAA.
†Professor, UTIAS, groth@utias.utoronto.ca, Senior Member AIAA.

However, these advantages of high-order methods are difficult to fully realize because of the conflicting relationship between accuracy and robustness. 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[4]. High-order schemes for elliptic partial differential equations (PDEs) should satisfy a maximum principle while remaining accurate[5]. Additionally, modelling reacting flows requires integrating numerically stiff and mathematically complex source terms over the computational domain. There are many studies of high-order schemes developed for finite-volume[4,6–15], discontinuous Galerkin[16–20], and spectral finite-difference/finite-volume methods[21–25] on both structured and unstructured mesh. In spite of 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.

Ivan and Groth[26,27] proposed a high-order Central Essentially Non Oscillatory (CENO), cell-centered, finite-volume scheme that was demonstrated to remain both accurate and robust in a variety of physically-complex flows. The CENO scheme uses a fixed central stencil which avoids the difficulties encountered by standard and weighted ENO schemes[4,10,12,13] that result from the use of variable stencils. With ENO and WENO schemes, selecting the appropriate stencils on general multi-dimensional unstructured meshes is not straightforward[7,8,11,28] and using these stencils can produce poor conditioning of the linear systems involved in performing the solution reconstruction[11,28].

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 the unlimited high-order k-exact reconstruction and the limited piecewise linear reconstruction. Switching between the two reconstruction algorithms is determined by a solution smoothness indicator that indicates whether or not the solution is resolved on the computational mesh. Originally developed for structured two-dimensional mesh, this scheme has been successfully extended to two- and three-dimensional unstructured mesh and applied to a variety of inviscid and viscous flows by McDonald et al.[29], Charest et al.[30].

In the present research, the high-order CENO finite-volume scheme is extended to solve the equations governing low-Mach-number, turbulent, reactive flows on unstructured mesh. The ability of the scheme to accurately model turbulent flames is demonstrated using a LES approach coupled with an eddy-dissipation-based model for turbulence-chemistry interactions. The resulting governing equations are solved using the pseudo-compressibility approach coupled with an implicit Newton-Krylov algorithm. The algorithm is applied to a laboratory scale turbulent premixed flame and analyzed in terms of accuracy and robustness. In particular, low- and high-order numerical predictions are compared with experimental measurements for quantities such as flame surface density, flame curvature, and flame height.

II. Pseudo-Compressibility Approach for Reactive Low Speed Flows

II.A. Governing Equations and Turbulent Closure Models

In the present research, the equations governing turbulent, reactive flows at low Mach numbers are considered. Large-scale turbulence is captured using LES with the constant-coefficient Smagorinsky model[31] for the subgrid-scale stresses. For the premixed turbulent flame investigated in the present study, chemical reactions are assumed to occur very fast. As a result of this assumption, the rate of combustion is controlled by the rate mixing between fresh reactants and hot combustion products only. The rate of mixing, and consequently the rate of combustion, is described by the eddy-dissipation or eddy-break-up model[32,33].

In three space dimensions, the resulting Favre-filtered partial-differential equations which govern the reactive tur-

bulent flow are

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

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

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

$$\frac{\partial}{\partial t}\left(\bar{\rho}\tilde{f}\right) + \frac{\partial}{\partial x_j}\left(\bar{\rho}\tilde{u}_j\tilde{f}\right) = \frac{\partial}{\partial x_j}\left[D_{\mathrm{T}}\frac{\partial \tilde{f}}{\partial x_j}\right] \tag{1d}$$

$$\frac{\partial}{\partial t}(\bar{\rho}\tilde{y}_{\mathrm{f}}) + \frac{\partial}{\partial x_j}\left(\bar{\rho}\tilde{u}_j\tilde{y}_{\mathrm{f}}\right) = \frac{\partial}{\partial x_j}\left[D_{\mathrm{T}}\frac{\partial \tilde{y}_{\mathrm{f}}}{\partial x_j}\right] + \bar{\rho}\dot{\omega}_{\mathrm{T}} \tag{1e}$$

where t is the time, p is the total pressure, ρ is the fluid density, 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, D_{T} is the turbulent eddy-diffusivity, and $\dot{\omega}_{\mathrm{T}}$ is the turbulent source term. The fluid stress tensor, λ_{ij}, 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 dynamic viscosity and μ_{T} is the turbulent eddy-viscosity. The heat flux vector due to molecular and turbulent motion is computed using Fick's law of diffusion and a gradient-based assumption for the turbulent component

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

where constant Prandtl numbers for the laminar, Pr_{L}, and turbulent, Pr_{T}, components were assumed. Here, $\mathrm{Pr}_{\mathrm{L}} = 0.7$ and $\mathrm{Pr}_{\mathrm{T}} = 0.9$.

The turbulent eddy-diffusivity is determined from the eddy-viscosity using a constant subfilter scale Schmidt number, Sc_{T}, by the following relation:

$$D_{\mathrm{T}} = \frac{\mu_{\mathrm{T}}}{\bar{\rho}\mathrm{Sc}_{\mathrm{T}}} \tag{4}$$

The turbulent eddy viscosity, μ_{T}, defined by the constant-coefficient Smagorinsky model[31] is

$$\mu_{\mathrm{T}} = \bar{\rho}L_{\mathrm{s}}^2\sqrt{2S_{ij}S_{ij}} \tag{5}$$

where L_{s} is the mixing length for the subgrid scales and S_{ij} is the local strain rate tensor. The local strain rate tensor is defined as

$$S_{ij} = \frac{1}{2}\left(\frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i}\right) \tag{6}$$

The length scale is computed using a two-layer approach to account for the presence of walls[34,35],

$$L_{\mathrm{s}} = \min\left(\kappa y_{\mathrm{wall}},\ f_\mu c_{\mathrm{s}}\Delta\right) \tag{7}$$

where $\kappa = 0.41$ is the von Kármán constant, y_{wall} is the local distance from the wall, f_μ is the Van Driest[36] damping function, $c_{\mathrm{s}} = 0.17$ is the Smagorinsky coefficient, and Δ is the filter width. The Van Driest[36] damping function is

$$f_\mu = 1 - \exp\left(-y^+/A\right) \tag{8}$$

where y^+ is the dimensionless wall distance and $A = 25$.

An implicit filtering approach is employed here whereby the grid is assumed to be the low-pass filter. A filter width equal to twice the average local mesh size was employed,

$$\Delta = 2\sqrt[3]{\Delta V} \tag{9}$$

where ΔV is the local cell volume.

The rate of turbulent combustion, as defined by Magnussen and Hjertager[32], is given by

$$\dot{\omega}_{\mathrm{T}} = \tau_{\mathrm{m}}^{-1}\, a\, \min\left(\tilde{y}_{\mathrm{f}},\ \tilde{y}_{\mathrm{o}}/s,\ b\tilde{y}_{\mathrm{p}}/(1+s)\right) \tag{10}$$

where s is the reaction stoichiometric coefficient, $\tilde{y}_{\mathrm{o}}$ and $\tilde{y}_{\mathrm{p}}$ are the mass fractions of oxygen and products, respectively, and τ_{m} is a turbulent mixing time scale. The reaction coefficients, a and b, were set to 2.0 and 0.5, respectively. The turbulent mixing time scale is determined from the local eddy-viscosity and length scale as

$$\tau_{\mathrm{m}} = \frac{\mu_{\mathrm{T}}}{\bar{\rho}(c_{\mathrm{s}}\Delta)^2} \tag{11}$$

II.B. Pseudo-Compressibility Approach

Incompressible and low-Mach-number flows are challenging to solve numerically because the partial derivative of density with respect to time vanishes. As a result, the governing equations (Eq. (1)) are ill-conditioned. The pseudo-compressibility method is employed here to overcome this ill-conditioning by modifying partial derivatives of density with respect to time[37–40]. In the original formulation of Chorin[37] for incompressible flows, a pressure time derivative was added to the steady form of the continuity equation and the primitive form of the governing equations were solved using a time-marching procedure. Turkel[40] derived the conservative form of Chorin's modified governing equations and showed that time derivatives of pressure should also be added to the other equations.

Since the the pseudo-compressibility method modifies the original partial derivatives with respect to time in Eq. (1), a dual-time-stepping approach is employed to regain time-accuracy[41–45]. Applying the pseudo-compressibility and dual-time-stepping approach to Eq. (1), the resulting governing equations are

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbf{\Gamma}\frac{\partial \mathbf{W}}{\partial \tau} + \frac{\partial}{\partial x}(\mathbf{E}-\mathbf{E}_{\mathbf{v}}) + \frac{\partial}{\partial y}(\mathbf{F}-\mathbf{F}_{\mathbf{v}}) + \frac{\partial}{\partial z}(\mathbf{G}-\mathbf{G}_{\mathbf{v}}) = \mathbf{S} \tag{12}$$

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. They are defined as

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

$$\mathbf{E}_{\mathbf{v}} = \begin{bmatrix} 0 \\ \lambda_{xx} \\ \lambda_{xy} \\ \lambda_{xz} \\ -q_x \\ D_{\mathrm{T}}\frac{\partial \tilde{f}}{\partial x} \\ D_{\mathrm{T}}\frac{\partial \tilde{y}_{\mathrm{f}}}{\partial x} \end{bmatrix},\ \mathbf{F}_{\mathbf{v}} = \begin{bmatrix} 0 \\ \lambda_{yx} \\ \lambda_{yy} \\ \lambda_{yz} \\ -q_y \\ D_{\mathrm{T}}\frac{\partial \tilde{f}}{\partial y} \\ D_{\mathrm{T}}\frac{\partial \tilde{y}_{\mathrm{f}}}{\partial y} \end{bmatrix},\ \mathbf{G}_{\mathbf{v}} = \begin{bmatrix} 0 \\ \lambda_{zx} \\ \lambda_{yz} \\ \lambda_{zz} \\ -q_z \\ D_{\mathrm{T}}\frac{\partial \tilde{f}}{\partial z} \\ D_{\mathrm{T}}\frac{\partial \tilde{y}_{\mathrm{f}}}{\partial z} \end{bmatrix},\ \mathbf{\Gamma} = \begin{bmatrix} \frac{1}{\beta} & 0 & 0 & 0 & \bar{\rho}_T & \bar{\rho}_f & \bar{\rho}_{y_{\mathrm{f}}} \\ \frac{\alpha}{\beta}\tilde{u} & \bar{\rho} & 0 & 0 & \bar{\rho}_T\tilde{u} & \bar{\rho}_f\tilde{u} & \bar{\rho}_{y_{\mathrm{f}}}\tilde{u} \\ \frac{\alpha}{\beta}\tilde{v} & 0 & \bar{\rho} & 0 & \bar{\rho}_T\tilde{v} & \bar{\rho}_f\tilde{v} & \bar{\rho}_{y_{\mathrm{f}}}\tilde{v} \\ \frac{\alpha}{\beta}\tilde{w} & 0 & 0 & \bar{\rho} & \bar{\rho}_T\tilde{w} & \bar{\rho}_f\tilde{w} & \bar{\rho}_{y_{\mathrm{f}}}\tilde{w} \\ \frac{\alpha}{\beta}\tilde{h} & 0 & 0 & 0 & \bar{\rho}c_p+\bar{\rho}_T\tilde{h} & \bar{\rho}\tilde{h}_f+\bar{\rho}_f\tilde{h} & \bar{\rho}\tilde{h}_{y_{\mathrm{f}}}+\bar{\rho}_{y_{\mathrm{f}}}\tilde{h} \\ \frac{\alpha}{\beta}\tilde{f} & 0 & 0 & 0 & \bar{\rho}_T\tilde{f} & \bar{\rho}+\bar{\rho}_f\tilde{f} & \bar{\rho}_{y_{\mathrm{f}}}\tilde{f} \\ \frac{\alpha}{\beta}\tilde{y}_{\mathrm{f}} & 0 & 0 & 0 & \bar{\rho}_T\tilde{y}_f & \bar{\rho}_f\tilde{y}_f & \bar{\rho}+\bar{\rho}_{y_{\mathrm{f}}}\tilde{y}_{\mathrm{f}} \end{bmatrix}.$$

where the subscripts T, f, and y_f denote partial derivatives with respect to temperature, mixture fraction, and fuel mass fraction, respectively. β is the pseudo-compressibility factor, α is a preconditioning parameter, and τ denotes the pseudo-time since the modified equations are no longer time-accurate. The preconditioning parameter, α, controls how the original governing equations are modified. The original pseudo-compressibility method of Chorin [37] 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.

II.C. Inviscid Eigenstructure

Based on the analysis conducted by Turkel [40] and the numerical results obtained by Qian et al. [46] and Lee and Lee [47], the optimal value of α is 2. However, Malan et al. [48,49] and Lee and Lee [47] found that a loss of robustness can occur for $\alpha > 1$ if β is too small. This loss of robustness occurs because the determinant of the modal matrix can be zero when $\alpha > 1$. Since larger values of α display better convergence characteristics [46,47], $\alpha = 1$ was selected for the current work. For $\alpha = 1$, the Jacobian matrix of the steady-state inviscid system with respect to the primitive variables is

$$\mathbf{A} = \boldsymbol{\Gamma}^{-1}\frac{\partial\left(\vec{\mathbf{F}}\cdot\hat{n}\right)}{\partial\mathbf{W}} = \begin{bmatrix} 0 & n_x\bar{\rho}\beta & n_y\bar{\rho}\beta & n_z\bar{\rho}\beta & 0 & 0 & 0 \\ \frac{n_x}{\bar{\rho}} & q & 0 & 0 & 0 & 0 & 0 \\ \frac{n_y}{\bar{\rho}} & 0 & q & 0 & 0 & 0 & 0 \\ \frac{n_z}{\bar{\rho}} & 0 & 0 & q & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & q & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & q & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & q \end{bmatrix} \tag{13}$$

where $q = \vec{v}\cdot\hat{n}$ and $c^2 = q^2 + 4\beta$. The resulting matrix of right eigenvectors is

$$\mathbf{R} = \begin{bmatrix} -\frac{\rho}{2}(q-c) & -\frac{\rho}{2}(q+c) & 0 & 0 & 0 & 0 & 0 \\ n_x & n_x & -n_y & -n_z & 0 & 0 & 0 \\ n_y & n_y & n_x & 0 & 0 & 0 & 0 \\ n_z & n_z & 0 & n_x & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{14}$$

The eigenvalues of the steady-state inviscid system defined by Eq. (12) in a particular direction are

$$\boldsymbol{\lambda} = \left[\frac{1}{2}u_n - \frac{1}{2}\sqrt{u_n{}^2 + 4\beta},\quad u_n,\quad u_n,\quad u_n,\quad u_n,\quad u_n,\quad \frac{1}{2}u_n + \frac{1}{2}\sqrt{u_n{}^2 + 4\beta}\right] \tag{15}$$

where u_n is the velocity of the bulk flow projected onto the direction vector of interest.

The minimum value of β was chosen based on the following formulation proposed by Turkel[40]:

$$\beta = \max\left[2\left(u^2 + v^2 + w^2\right),\ \varepsilon\right] \tag{16}$$

where ε is a smallness parameter.

III. 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. For a cell i, the approach results in the following coupled system of partial differential equations (PDEs) for cell-averaged solution quantities:

$$\frac{\mathrm{d}\overline{\mathbf{U}}_i}{\mathrm{d}t} + \mathbf{\Gamma}_i \frac{\mathrm{d}\overline{\mathbf{W}}_i}{\mathrm{d}\tau} = -\frac{1}{V_i} \oiint \left(\vec{\mathbf{F}} - \vec{\mathbf{F}}_\mathbf{v}\right) \cdot \hat{n}\, \mathrm{d}A + \frac{1}{V_i} \iiint \mathbf{S}\, \mathrm{d}V = -\mathbf{R_i} \tag{17}$$

where the overbar denotes cell-averaged quantities, V 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 and volume integrals in Eq. (17) produces a set of nonlinear ordinary differential equations (ODEs) given by

$$\frac{\mathrm{d}\overline{\mathbf{U}}_i}{\mathrm{d}t} + \mathbf{\Gamma}_i \frac{\mathrm{d}\overline{\mathbf{W}}_i}{\mathrm{d}\tau} = -\frac{1}{V_i} \sum_{l=1}^{N_\mathrm{f}} \sum_{m=1}^{N_{\mathrm{G_f}}} \left[\omega_\mathrm{f} \left(\vec{\mathbf{F}} - \vec{\mathbf{F}}_\mathbf{v}\right) \cdot \hat{n} A\right]_{i,l,m} + \frac{1}{V_i} \sum_{n=1}^{N_{\mathrm{G_v}}} \left[\omega_\mathrm{v} \mathbf{S} V\right]_{i,n} \tag{18}$$

where N_f is the number of faces (equal to 4 for tetrahedra and 6 for hexahedra), $N_{\mathrm{G_f}}$ and $N_{\mathrm{G_v}}$ are the number of quadrature points for the surface and volume integrals, respectively, and ω_f and ω_v are the corresponding surface and volume integral quadrature weights. In Eq. (18), the number of quadrature points required along each face or over each cell is a function of the reconstruction order and number of spatial dimensions. For tetrahedra and Cartesian (hexahedra with rectangular faces) cells, Gauss quadrature points can be directly mapped from the canonical form to the 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[50,51]. The coefficients for the quadrature rules applied here are tabulated by Felippa[52] and summarized in Table 1.

Table 1. Gauss quadrature rules used for integration.

Reconstruction	Tetrahedra			Cartesian / Hexahedra		
	$N_{\mathrm{G_f}}$	$N_{\mathrm{G_v}}$	Polynomial Degree	$N_{\mathrm{G_f}}$	$N_{\mathrm{G_v}}$	Polynomial Degree
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

III.A. Trilinear Coordinate Transformation for Arbitrary Hexahedra

Geometric representation of a general hexahedral cell shown in Fig. 1 is based on the trilinear transformation

$$\vec{r}(p,q,r) = \vec{a} + \vec{b}p + \vec{c}q + \vec{d}r + \vec{e}pq + \vec{f}pr + \vec{g}qr + \vec{h}pqr \tag{19}$$

where p, q and r are Cartesian coordinates in the canonical space of the reference hexahedron and $\vec{a}$, $\vec{b}$, $\vec{c}$, $\vec{d}$, $\vec{e}$, $\vec{f}$, $\vec{g}$ and $\vec{h}$ are the transformation coefficient vectors. The reference hexahedron, shown in Fig. 1, consists of a unit cube with faces aligned with the p, q and r coordinate axes. Using the transformation defined by Eq. (19), integrals over the volume of an arbitrary hexahedral cell, Q, are evaluated over the reference element $\hat{Q}$ as

$$\iiint_Q f(x,y,z)\, \mathrm{d}Q = \iiint_{\hat{Q}} f(\vec{r}(p,q,r))\ J(p,q,r)\, \mathrm{d}\hat{Q} \tag{20}$$

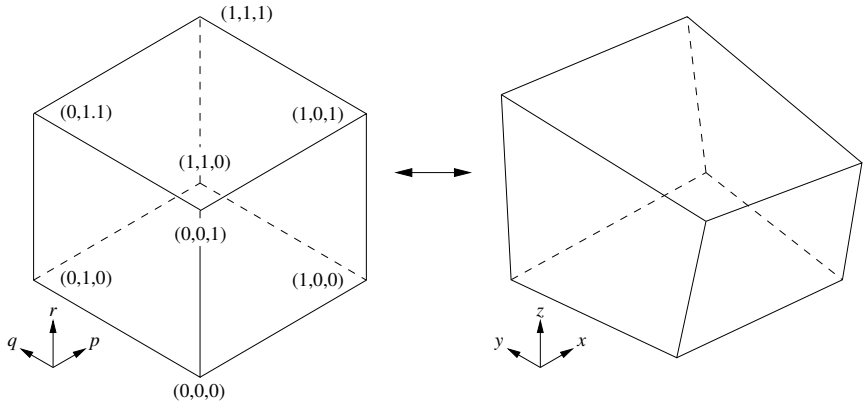

Figure 1. (Left) Reference cube $\hat{Q}$ and (right) arbitrary hexahedron Q.

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

$$J(p, q, r) = \vec{X}(q, r) \cdot \left(\vec{Y}(p, r) \times \vec{Z}(p, q)\right) \tag{21}$$

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

$$\iint_{\Gamma_x} f(x, y, z)\, \mathrm{d}\Gamma_x = \iint_{\Gamma_p} f(\vec{r}(p, q, r))\; J_x(p, q, r)\; \mathrm{d}\Gamma_p \tag{22}$$

where Γ_p is the corresponding surface on the reference cube. The surface Jacobian, J_x, is

$$J_x(p, q, r) = \left|\vec{Y}(p, r) \times \vec{Z}(p, q)\right| \tag{23}$$

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

$$\hat{n} = \frac{\vec{Y}(p, r) \times \vec{Z}(p, q)}{\left|\vec{Y}(p, r) \times \vec{Z}(p, q)\right|} \tag{24}$$

Using Eqs. (20) and (22), relations can be derived for the cell volume as well as the area of cell faces by setting $f(x, y, z)$ to unity. They are also used to transform the quadrature points and weights from the canonical element to the arbitrary hexahedron.

III.B. CENO Reconstruction

Evaluating Eq. (18) requires integration of the numerical flux along the cell faces, but 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 [26,27]. This hybrid approach involves a fixed central stencil in smooth or fully-resolved regions which is switched to a limited piecewise linear reconstruction when discontinuities in solution content are encountered. This 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 called 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. Oscillations can even occur for relatively smooth flows when there is insufficient mesh resolution.

III.B.1. k-Exact Reconstruction

The CENO spatial discretization scheme is based on the high-order k-exact least-squares reconstruction technique of Barth [6]. The k-exact higher-order reconstruction algorithm begins by assuming that the solution within each cell can

be represented by the following Taylor series expansion in three dimensions:

$$u_i^k(x,y,z) = \sum_{p_1=0}^{(p_1+p_2+p_3)\le k} \sum_{p_2=0} \sum_{p_3=0} (x-x_i)^{p_1}(y-y_i)^{p_2}(z-z_i)^{p_3} D_{p_1p_2p_3} \tag{25}$$

where u_i^k is the reconstructed solution quantity, (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 $(p_1 + p_2 + p_3) \le k$.

The following conditions are applied to determine the unknown coefficients: i) the solution reconstruction must reproduce polynomials of degree $N \le k$ 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

$$\overline{u}_i = \frac{1}{V_i} \iiint_{V_i} u_i^k(x,y,z)\,\mathrm{d}V \tag{26}$$

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

$$N = \frac{1}{d!} \prod_{n=1}^{d} (k+n) \tag{27}$$

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. Additional neighbors are included to ensure that the stencil is not biased in any particular direction and that the reconstruction remains reliable on poor quality meshes with high aspect ratio cells[26]. For each neighboring cell, p, a constraint is formed by requiring that

$$\overline{u}_p = \frac{1}{V_p} \iiint_{V_p} u_i^k(x,y,z)\,\mathrm{d}V \tag{28}$$

Since the constraints of Eqs. (26) and (28) result in an over-determined system of linear equations, a least-squares solution for the coefficients, $D_{p_1p_2p_3}$, is obtained in each cell. Equation (26) is strictly enforced by Gaussian elimination and a minimum-error solution to the remaining constraint equations is sought. Substituting Eq. (25) into Eq. (26) and rearranging for D_{000} gives

$$D_{000} = \overline{u}_i - \sum_{(p_1+p_2+p_3)>0}^{(p_1+p_2+p_3)\le k} \sum \sum \sum D_{p_1p_2p_3} \frac{1}{V_i} \iiint_{V_i} (x-x_i)^{p_1}(y-y_i)^{p_2}(z-z_i)^{p_3}\,\mathrm{d}V \tag{29}$$

A similar relation is obtained using Eqs. (25) and (28) to give

$$D_{000} = \overline{u}_j - \sum_{(p_1+p_2+p_3)>0}^{(p_1+p_2+p_3)\le k} \sum \sum \sum D_{p_1p_2p_3} \frac{1}{V_j} \iiint_{V_j} (x-x_i)^{p_1}(y-y_i)^{p_2}(z-z_i)^{p_3}\,\mathrm{d}V \tag{30}$$

Equating Eq. (29) and Eq. (30) gives

$$\overline{u}_j - \overline{u}_i = \sum_{(p_1+p_2+p_3)>0}^{(p_1+p_2+p_3)\le k} \sum \sum \sum D_{p_1p_2p_3} (\widehat{x^{p_1}y^{p_2}z^{p_3}})_{ij} \tag{31}$$

where

$$(\widehat{x^{p_1}y^{p_2}z^{p_3}})_{ij} = \frac{1}{V_j} \iiint_{V_j} (x-x_i)^{p_1}(y-y_i)^{p_2}(z-z_i)^{p_3}\,\mathrm{d}V - \frac{1}{V_i} \iiint_{V_i} (x-x_i)^{p_1}(y-y_i)^{p_2}(z-z_i)^{p_3}\,\mathrm{d}V \tag{32}$$

are geometric coefficients which only depend on the mesh geometry. The new overdetermined linear system is formed using Eq. (31) and given by

$$\begin{bmatrix} (\widehat{x^0y^0}z^1)_{i1} & \cdots & (x^{p_1}\widehat{y^{p_2}z^{p_3}})_{i1} & \cdots & (\widehat{x^ky^0z^0})_{i1} \\ \vdots & & \vdots & & \vdots \\ (\widehat{x^0y^0}z^1)_{ij} & \cdots & (x^{p_1}\widehat{y^{p_2}z^{p_3}})_{ij} & \cdots & (\widehat{x^ky^0z^0})_{ij} \\ \vdots & & \vdots & & \vdots \\ (\widehat{x^0y^0}z^1)_{iN_n} & \cdots & (x^{p_1}\widehat{y^{p_2}z^{p_3}})_{iN_n} & \cdots & (\widehat{x^ky^0z^0})_{iN_n} \end{bmatrix} \cdot \begin{bmatrix} D_{001} \\ \vdots \\ D_{p_1p_2p_3} \\ \vdots \\ D_{k00} \end{bmatrix} = \begin{bmatrix} \overline{u}_1 - \overline{u}_i \\ \vdots \\ \overline{u}_j - \overline{u}_i \\ \vdots \\ \overline{u}_{N_n} - \overline{u}_i \end{bmatrix} \tag{33}$$

where N_n is the number of neighbors in the stencil. The remaining polynomial coefficient, D_{000}, is obtained from Eq. (29) after obtaining the least-squares solution to the overdetermined system given by Eq. (33).

The resulting coefficient matrix of the linear system defined in Eq. (33) depends only on the mesh geometry, so it can be inverted and stored prior to computations [14,53]. Either a Householder QR factorization algorithm or orthogonal decomposition by the SVD method was used to solve the weighted least-squares problem [54]. Weighting is applied here to each individual constraint equation to improve the locality of the reconstruction [55]. An inverse distance weighting formula is applied. For the reconstruction in cell i,

$$w_j = \frac{1}{|\vec{x}_j - \vec{x}_i|}, \tag{34}$$

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

III.B.2. Reconstruction at Boundaries

To enforce conditions at physical boundaries, the least-squares reconstruction was constrained in adjacent control volumes without altering the reconstruction order of accuracy [14,15,27]. 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

$$f(\vec{x}) = a(\vec{x})\, f_D(\vec{x}) + b(\vec{x})\, f_N(\vec{x}) \tag{35}$$

where $a(\vec{x})$ and $b(\vec{x})$ define the contribution of the Dirichlet, $f_D(\vec{x})$, and Neumann, $f_N(\vec{x})$, components, respectively. In terms of the cell reconstruction, the Dirichlet condition is simply expressed as

$$f_D\left(\vec{x}_g\right) = u^k\left(\vec{x}_g\right) \tag{36}$$

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

$$\begin{aligned} f_N\left(\vec{x}_g\right) &= \nabla u^k\left(\vec{x}_g\right) \cdot \hat{n}_g \\ &= \sum_{p_1+p_2+p_3=1}^{(p_1+p_2+p_3)\le k} \sum \sum \Delta x^{p_1-1} \Delta y^{p_2-1} \Delta z^{p_3-1} \left[p_1 \Delta y \Delta z n_x + p_2 \Delta x \Delta y n_y + p_3 \Delta x \Delta y n_z \right] D_{p_1p_2p_3} \end{aligned} \tag{37}$$

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

Exact solutions to the boundary constraints described by Eq. (35) are sought, which adds linear equality constraints to the original least-squares problem (Section III.B.1). 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 III.B.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 and 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. (25) for different variables to become coupled. For these types of coupled boundary conditions, the reconstruction for all of the coupled solution variables is performed together [15,27]. Thus the final matrix in Eq. (33) for the constrained least-squares reconstruction contains the individual constraints for each variable (Eq. (35)), the relational constraints, and the approximate mean conservation equations for each variable (Eq. (31)).

III.B.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

$$S = \frac{\sigma}{\max\left[(1-\sigma),\delta\right]}\frac{(\mathrm{SOS}-\mathrm{DOF})}{\mathrm{DOF}-1} \tag{38}$$

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

$$\sigma = 1 - \frac{\sum_p \left[u_p^k(x_p,y_p,z_p) - u_i^k(x_p,y_p,z_p)\right]^2}{\sum_p \left[u_p^k(x_p,y_p,z_p) - \overline{u}_i\right]^2} \tag{39}$$

where u is the solution variable of interest, the subscript p refers to the cells in the reconstruction stencil, $u_p^k(x_p,y_p,z_p)$ is the reconstructed solution in cell p evaluated at the cell's centroid (x_p,y_p,z_p), $u_i^k(x_p,y_p,z_p)$ is the projected value of the reconstruction polynomial for cell i evaluated at (x_p,y_p,z_p), and $\overline{u}_i$ is the average value for cell i. By definition, σ can have a value 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.

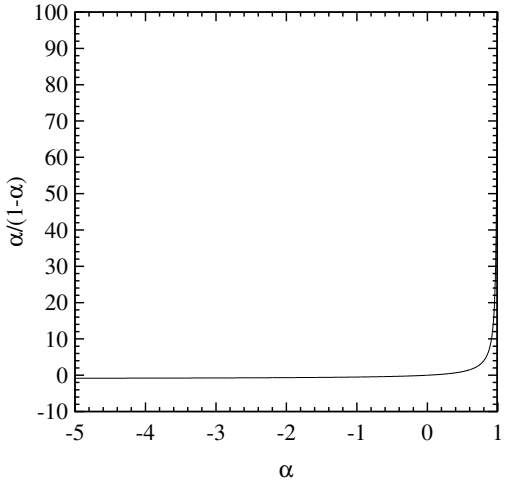

Figure 2. Variation of smoothness indicator with σ.

The behavior of the smoothness indicator is demonstrated in Fig. 2. As σ 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[26]. Typical values for S in smooth regions tend to be orders of magnitude greater than these cutoff limits. Unless otherwise specified, a value for S_c of 1000 was used.

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.[56] specifically for multiple dimensions in conjunction with the slope limiter function of Venkatakrishnan[57]. Note that the limited linear reconstruction in non-smooth regions is only applied to the inviscid terms in Eq. (18). The viscous flux and source term is still evaluated using the higher-order representation since it is generally the inviscid terms that generate instabilities.

III.C. Numerical Flux

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

$$\vec{\mathbf{F}}\cdot\hat{n} = \mathcal{F}\left(\mathbf{W_L},\mathbf{W_R},\hat{n}\right) \tag{40}$$

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 III.B. As a result, the leading truncation error due to the inviscid operator is $\mathcal{O}\left(\Delta x^{k+1}\right)$.

The flux function, $\mathcal{F}$, was derived by applying Roe's approximate Riemann solver[58,59] to the new modified inviscid eigensystem for Eq. (12). The numerical flux at the interface between two cells is given by

$$\mathbf{F}\left(\mathcal{R}\left(\mathbf{W_L}, \mathbf{W_R}\right)\right) = \frac{1}{2}\left(\mathbf{F_R} + \mathbf{F_L}\right) - \frac{1}{2}|\hat{\mathbf{A}}|\Delta\mathbf{W} \tag{41}$$

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

$$\mathcal{G}\left(\mathbf{W_L}, \mathbf{W_R}, \nabla\mathbf{W_L}, \nabla\mathbf{W_R}\right) = \mathbf{F_v}\left\{\frac{1}{2}\left(\mathbf{W_L} + \mathbf{W_R}\right), \frac{1}{2}\left(\nabla\mathbf{W_L}, \nabla\mathbf{W_R}\right)\right\} \tag{42}$$

Because derivatives of the reconstructed polynomial are required, the leading truncation error due to the viscous operator is only $\mathcal{O}\left(\Delta x^{k}\right)$. The degree of the reconstruction polynomial is therefore increased by one to match the leading truncation error introduced by the inviscid operator. The Gauss quadrature rule is selected to maintain an order of accuracy of $k + 1$ when integrating the fluxes over the cell faces.

For piecewise-linear ($k = 1$) representations, second-order ($k+1$) accuracy of the viscous operator can be achieved without increasing the degree of the polynomial interpolatant. In this case, the average gradient at the interface is evaluated by[60]

$$\nabla\mathbf{W}_{i+1/2} = \left(\mathbf{W}_n - \mathbf{W}_p\right)\frac{\hat{n}}{\hat{n}\cdot\vec{r}_{pn}} + \left(\overline{\nabla\mathbf{W}} - \overline{\nabla\mathbf{W}}\cdot\vec{r}_{pn}\frac{\hat{n}}{\hat{n}\cdot\vec{r}_{pn}}\right) \tag{43}$$

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 weighted average of the neighboring cell gradients, $\overline{\nabla\mathbf{W}}$, is evaluated as

$$\begin{gathered}\overline{\nabla\mathbf{W}} = \chi\nabla\mathbf{W}_p + (1-\chi)\nabla\mathbf{W}_n \\ \chi = V_p/(V_p + V_n)\end{gathered} \tag{44}$$

Equation (43) is second-order accurate if the gradient representation is also second-order accurate. Thus, $k + 1$ reconstruction is not required for $k = 1$.

III.D. Dual-Time Stepping Approach for Unsteady Problems

Integration of 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 parallel graph partitioning algorithm, called Parmetis[61], 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[62]. Ghost cells, which surround an individual local solution domain and overlap cells on neighboring domains, are used to share solution content through inter-block communication.

Newton's method is applied in this work for transient continuation using a dual-time-stepping approach[41–45] with the family of high-order backwards difference formulas to discretize the physical time derivative. Newton's method is used to relax

$$\mathbf{R}^*(\mathbf{W}) = \mathbf{R} + \frac{\mathrm{d}\overline{\mathbf{W}}}{\mathrm{d}t} = \mathbf{0} \tag{45}$$

at each physical time level. This particular implementation follows the algorithm developed by Groth et al.[63–65] specifically for use on large multi-processor parallel clusters.

Applying the BDF temporal discretization and Newton's method to the semi-discrete form of the governing equations, Eq. (45), gives

$$\left[\frac{a}{\Delta t^n}\left(\frac{\partial\mathbf{U}}{\partial\mathbf{W}}\right)^{(n+1,k)} + \frac{\mathbf{\Gamma}}{\Delta\tau^k} + \left(\frac{\partial\mathbf{R}}{\partial\mathbf{W}}\right)^{(n+1,k)}\right]\Delta\mathbf{W}^{(n+1,k)} = -\mathbf{R}^{(n+1,k)} - \left.\frac{\mathrm{d}\overline{\mathbf{U}}}{\mathrm{d}t}\right|^{(n+1,k)} = -\mathbf{R}^{*(n+1,k)} \tag{46}$$

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). Since Newton's method can fail when initial solution estimates fall outside the radius of convergence, an implicit Euler discretization was applied to the pseudo-time term in Eq. (46).

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. Note however, it is well known that BDF methods with accuracy higher than second-order can be unstable when the real component of the eigenvalues of the system are negative. A stability analysis of the BDF methods applied to the one-dimensional form of the linear advection equation is shown in Fig. 3. The eigenvalues, λ, for a first-order upwind discretization applied to this system with uniform mesh spacing Δx and periodic boundary conditions are also shown in Fig. 3. BDF4 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. (12) involves a system of advection-diffusion-reaction equations. No issues related to the stability of the BDF methods were encountered throughout this research.

Table 2. Coefficients for the BDF methods.

$$b^{n+1}u^{n+1} + b^n u^n + \ldots = \Delta t f^{n+1}$$

Order	u^{n+1}	u^n	u^{n-1}	u^{n-2}	u^{n-3}
1	1	-1			
2	$\frac{2}{3}$	-2	$\frac{1}{2}$		
3	$\frac{11}{6}$	-3	$\frac{3}{2}$	$-\frac{1}{3}$	
4	$\frac{25}{12}$	-4	3	$\frac{4}{3}$	$\frac{1}{4}$

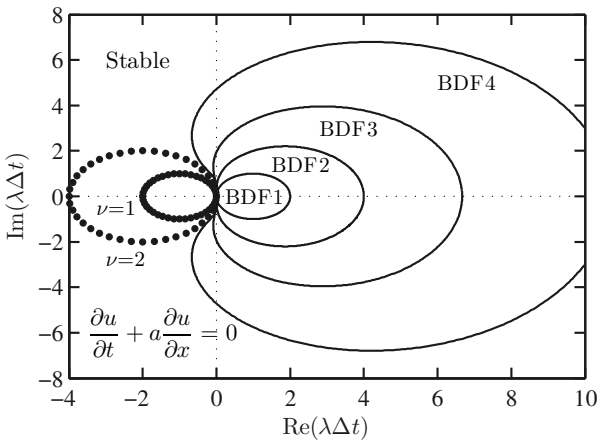

Figure 3. Stability diagram for the BDF methods applied to the linear advection equation. Lines enclose the unstable regions for each method; symbols are the eigenvalues for a first-order upwind spatial discretization with uniform mesh spacing Δx and periodic boundaries. λ represents the eigenvalues, Δt is the time step size, a is the wave speed, and $\nu = a\Delta t/\Delta x$ is the CFL number.

In Eq. (46), $\Delta\tau^k$ is the inner pseudo time step while Δt^n is the physical time step. The pseudo time step size was determined by considering the inviscid Courant-Friedrichs-Lewy (CFL) and viscous Von Neumann stability criteria based on the steady pseudo-compressible system. The maximum permissible time step for each local cell is determined by

$$\Delta\tau^k \leq \text{CFL} \cdot \min\left[\frac{\Delta x}{\max(\lambda)}, \frac{\rho\Delta x^2}{\mu}\right] \tag{47}$$

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. A value for CFL of 10^3 is typically used. The stability of the unsteady system is governed by the physical time

step size, which is determined based on the physical CFL criterion:

$$\Delta t^n \leq \mathrm{CFL}_{\mathrm{phys}} \cdot \min\left(\frac{\Delta x}{u^2 + v^2 + w^2}\right) \tag{48}$$

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

III.D.1. Inexact Newton Method

In the dual time-stepping procedure, at each physical time step n, a steady problem is solved using a Newton procedure. A solution to Eq. (46) is sought by iteratively solving a sequence of linear systems given an initial estimate, $\mathbf{W}^{(n+1,0)}$, at the beginning of each physical time step. Successively improved estimates are obtained by solving

$$\left(\frac{\partial \mathbf{R}^*}{\partial \mathbf{W}}\right)^{(n+1,k)} \Delta\mathbf{W}^{(n+1,k)} = \mathbf{J}(\mathbf{W}^{(n+1,k)})\, \Delta\mathbf{W}^{(n+1,k)} = -\mathbf{R}^*(\mathbf{W}^{(n+1,k)}) \tag{49}$$

where $\mathbf{J} = \partial\mathbf{R}^*/\partial\mathbf{W}$ is the residual Jacobian. After solving Eq. (49) for $\Delta\mathbf{W}^{(n+1,k)}$, the improved solution at step $(n+1,k)$ is determined from

$$\mathbf{W}^{(n+1,k+1)} = \mathbf{W}^{(n+1,k)} + \Delta\mathbf{W}^{(n+1,k)} \tag{50}$$

The Newton iterations proceed at each physical time level until some desired reduction of the norm of the residual is achieved and the condition $\|\mathbf{R}^*(\mathbf{W}^{(n+1,k)})\| < \epsilon\|\mathbf{R}^*(\mathbf{W}^{(n+1,0)})\|$ is met. The tolerance, ϵ, used in this work was 10^{-2}.

For a system of nonlinear equations, each step of Newton's method requires the solution of the linear problem $\mathbf{J}\Delta\mathbf{W} = -\mathbf{R}^*(\mathbf{W})$. 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[66–69]. 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 the matrix $\mathbf{J}$ 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 ζ is typically in the range $0.1 - 0.5$[70]. We use a restarted version of the GMRES algorithm here, GMRES(m), that minimizes storage by restarting every m iterations.

Right preconditioning of $\mathbf{J}$ is performed to help facilitate the solution of the linear system without affecting the solution residual, $\mathbf{b}$. The preconditioning takes the form

$$\left(\mathbf{J}\mathbf{M}^{-1}\right)(\mathbf{M}\mathbf{x}) = \mathbf{b} \tag{51}$$

where $\mathbf{M}$ 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

$$\mathbf{M}^{-1} = \sum_{k=1}^{N_\mathrm{b}} \mathbf{B}_k^T \mathbf{M}_k^{-1} \mathbf{B}_k \tag{52}$$

where N_b is the number of blocks and $\mathbf{B}_k$ is the gather matrix for the kth domain. The local preconditioner, $\mathbf{M}_k^{-1}$, in Eq. (52) is based on block ILU(p) factorization[69] of the Jacobian for the first order approximation of each domain. The level of fill, p, was maintained between 0–1 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 matrix reordering is used to permute the Jacobian's sparsity pattern into a band matrix form with a small bandwith[71].

IV. Results For Three-Dimensional Unstructured Mesh

The proposed finite-volume scheme was assessed in terms of both accuracy and stability through its application to a laboratory-scale turbulent premixed flame. All computations were performed on a high performance parallel cluster

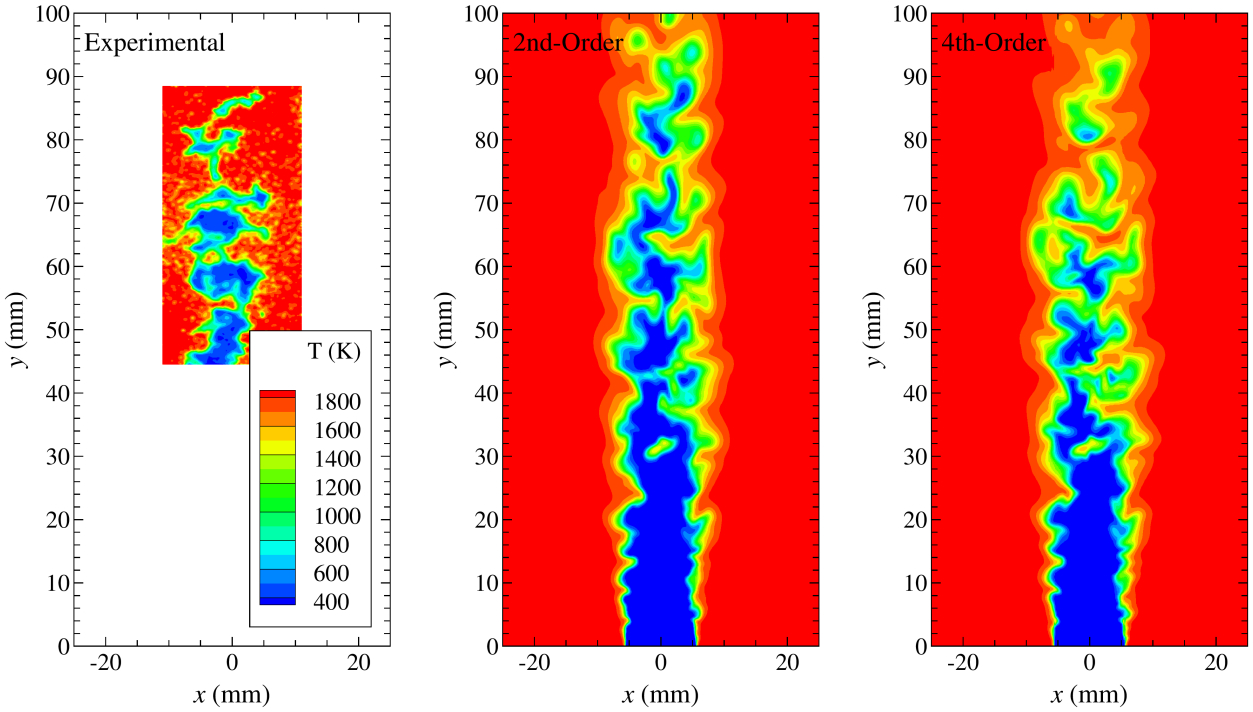

Figure 4. Measured and predicted instantaneous temperature contours. Instantaneous experimental results are unfiltered in this figure.

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.

The lean turbulent premixed methane flame of Yuen and Gülder[72,73] 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 III. 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[72,73] 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.[74] and Shahbazian et al.[75] 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 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 III were applied to 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. In the mathematical model, 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.[76]. A time step size of 12.5 μs was employed for all calculations, which corresponds to a $\mathrm{CFL}_{\mathrm{phys}} \approx 0.8$.

The unfiltered instantaneous measurements are compared with the Favre-filtered predictions for temperature in Fig. 4. Similar structures are observed in both the experiments and predictions. The predictions underestimate the degree of wrinkling, with the 2nd-order predictions displaying significantly less wrinkling than the 4th-order predictions. The 4th-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 4th-order predictions. As a result, the 4th-order scheme predicts a shorter flame which agrees better with the experimental measurements.

The differences in predicted flame height are evident in Fig. 5, 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

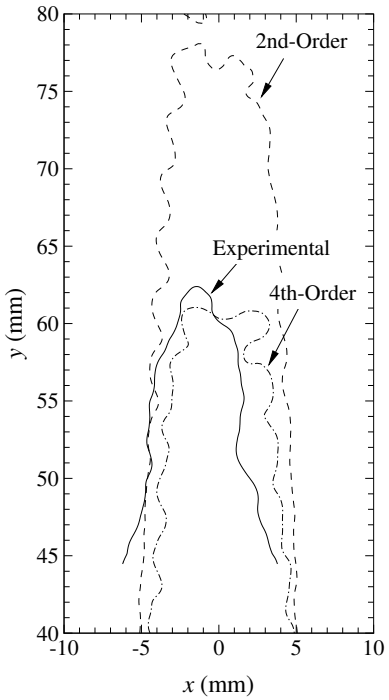

Figure 5. Comparison between measured and predicted average flame heights for the Yuen and Gülder [72,73] flame. Contour lines define the location where $\langle c \rangle = 0.5$.

temperature was equal to 0.5. The progress variable is defined in this research as

$$c = \frac{T - T_u}{T_{bu} - T_u} \tag{53}$$

where T_u and T_b are the temperatures of the unburned and burned gases. The 2nd-order scheme greatly over-predicts the measured flame height while the 4th-order scheme slightly under-predict the average height. Overall, the 4th-order results show excellent agreement with the measured flame height.

The two-dimensional (2D) flame surface density (FSD) was extracted from the planar Rayleigh scattering images and compared with predictions in Fig. 6(a). As in Fig. 5, the flame surface was defined as the location where $\langle c \rangle = 0.5$. 2D FSD values were determined using the method developed by Shepherd [77]. 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 to the flame area. A similar procedure was applied to 2D slices extracted from the numerical predictions. For direct comparison with the numerical predictions, the measurements for the instantaneous temperature were filtered using a moving average filter with a width equal to the value used in the computations.

Both sets of predicted FSD values qualitatively reproduce 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 2nd-order scheme are in better agreement with the measurements. They over-predict the experimental value by a factor of approximately 1.2 whereas the 4th-order results over-predict the peak by a factor of 1.3. A higher predicted peak value for the FSD is expected from the 4th-order scheme since a larger degree of flame wrinkling was observed by this scheme in Fig. 4. This general over-prediction of the FSD is 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. 6(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, illustrated in Fig. 4, without generating any unphysical numerical oscillations. Additionally, the smoothness indicator correctly identified regions where large turbulent fluctuations

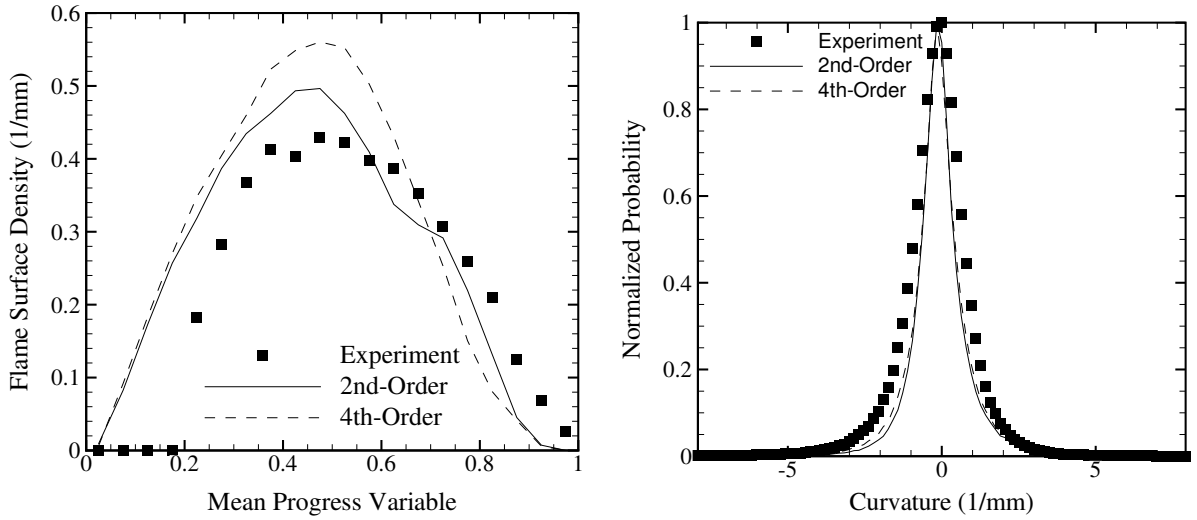

Figure 6. Measured and predicted (a) 2D flame surface density and (b) normalized PDF of 2D curvature for the Yuen and Gülder [72,73] flame. Results correspond to a flame surface defined by a progress variable of $c = 0.5$.

were under-resolved by the employed computational mesh. This ensured the reliable convergence of the Newton-Krylov algorithm at each physical time step.

V. 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, Godunov-type, finite-volume, methods and uses the hybrid CENO reconstruction method recently proposed by Ivan and Groth [26,27].

The proposed finite-volume formulation was validated by comparing predicted 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. 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 solutions were maintained throughout this study 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.

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 and incorporating a multi-block adaptive mesh refinement (AMR) algorithm [78–81]. 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 [26,27]. More accurate models for turbulent combustion that incorporate complex chemistry and detailed turbulence-chemistry interactions will also be included. One example is the presumed conditional moment, flame prolongation of ILDM combustion model [74,82].

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. Wilcox, D. C., *Turbulence Modeling for CFD*, DCW Industries, 3rd ed., 2006.

2. Pope, S. B., *Turbulent Flows*, Cambridge University Press, 2000.

3. Pirozzoli, S., "On the spectral properties of shock-capturing schemes," *J. Comput. Phys.*, Vol. 219, No. 2, 2006, pp. 489–497.

4. Harten, A., Engquist, B., Osher, S., and Chakravarthy, S. R., "Uniformly high order accurate essentially non-oscillatory schemes, III," *J. Comput. Phys.*, Vol. 71, No. 2, 1987, pp. 231–303.

5. Coirier, W. J. and Powell, K. G., "Solution-Adaptive Cartesian Cell Approach for Viscous and Inviscid Flows," *AIAA J.*, Vol. 34, No. 5, May 1996, pp. 938–945.

6. Barth, T. J., "Recent Developments in High Order K-Exact Reconstruction on Unstructured Meshes," AIAA Paper 93-0668, 1993.

7. Abgrall, R., "On essentially non-oscillatory schemes on unstructured meshes: analysis and implementation," *J. Comput. Phys.*, Vol. 114, 1994, pp. 45–58.

8. Sonar, T., "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.*, 1997, pp. 140–157.

9. Ollivier-Gooch, C. F., "Quasi-ENO Schemes for Unstructured Meshes Based on Unlimited Data-Dependent Least-Squares Reconstruction," *J. Comput. Phys.*, Vol. 133, 1997, pp. 6–17.

10. Jiang, G. S. and Shu, C. W., "Efficient implementation of weighted ENO schemes," *J. Comput. Phys.*, Vol. 126, No. 1, 1996, pp. 202–228.

11. Stanescu, D. and Habashi, W., "Essentially Nonoscillatory Euler solutions on unstructured meshes using extrapolation," *AIAA J.*, Vol. 36, 1998, pp. 1413–1416.

12. Friedrich, O., "Weighted essentially non-oscillatory schemes for the interpolation of mean values on unstructured grids," *J. Comput. Phys.*, Vol. 144, 1998, pp. 194–212.

13. Hu, C. and Shu, C.-W., "Weighted essentially non-oscillatory schemes on triangular meshes," *J. Comput. Phys.*, Vol. 150, 1999, pp. 97–127.

14. Ollivier-Gooch, C. F. and Van Altena, M., "A high-order accurate unstructured mesh finite-volume scheme for the advection-diffusion equation," *J. Comput. Phys.*, Vol. 181, No. 2, 2002, pp. 729–752.

15. Nejat, A. and Ollivier-Gooch, C., "A high-order accurate unstructured finite volume Newton-Krylov algorithm for inviscid compressible flows," *J. Comput. Phys.*, Vol. 227, No. 4, 2008, pp. 2582–2609.

16. Cockburn, B. and Shu, C.-W., "TVB Runge-Kutta Local Projection Discontinous Galerkin Finite-Element Method for Conservation Laws II: General Framework," *Math. Comp.*, Vol. 52, 1989, pp. 411.

17. Cockburn, B., Hou, S., and Shu, C.-W., "TVB Runge-Kutta Local Projection Discontinous Galerkin Finite-Element Method for Conservation Laws IV: The Multidimensional Case," *J. Comput. Phys.*, Vol. 54, 1990, pp. 545.

18. Hartmann, R. and Houston, P., "Adaptive Discontinuous Galerkin Finite Element Methods for the Compressible Euler Equations," *J. Comput. Phys.*, Vol. 183, 2002, pp. 508–532.

19. Luo, H., Baum, J. D., and Löhner, R., "A Hermite WENO-based limiter for discontinuous Galerkin method on unstructured grids," *J. Comput. Phys.*, Vol. 225, 2007, pp. 686–713.

20. Gassner, G., Lörcher, F., and Munz, C.-D., "A contribution to the construction of diffusion fluxes for finite volume and discontinuous Galerkin schemes," *J. Comput. Phys.*, Vol. 224, No. 2, 2007, pp. 1049–1063.

21. Wang, Z. J., "Spectral (Finite) Volume Method for Conservation Laws on Unstructured Grids – Basic Formulation," *J. Comput. Phys.*, Vol. 178, 2002, pp. 210–251.

22. Wang, Z. J. and Liu, Y., "Spectral (Finite) Volume Method for Conservation Laws on Unstructured Grids – II. Extenstion to Two-Dimensional Scalar Equation," *J. Comput. Phys.*, Vol. 179, 2002, pp. 665–697.

23. Wang, Z. J., Zhang, L., and Liu, Y., "High-Order Spectral Volume Method for 2D Euler Equations," Paper 2003–3534, AIAA, June 2003.

24. Wang, Z. J. and Liu, Y., "Spectral (Finite) Volume Method for Conservation Laws on Unstructured Grids – III. One Dimensional Systems and Partition Optimization," *Journal of Scientific Computing*, Vol. 20, No. 1, 2004, pp. 137–157.

25. Sun, Y., Wang, Z. J., and Liu, Y., "Spectral (finite) volume method for conservation laws on unstructured grids VI: extension to viscous flow," *J. Comput. Phys.*, Vol. 215, No. 1, 2006, pp. 41–58.

26. Ivan, L. and Groth, C. P. T., "High-order central CENO finite-volume scheme with adaptive mesh refinement," AIAA paper 2007-4323, 2007.

27. Ivan, L. and Groth, C. P. T., "High-order solution-adaptive central essentially non-oscillatory (CENO) method for viscous flows," AIAA paper 2011-0367, 2011.

28. Haselbacher, A., "A WENO Reconstruction Algorithm for Unstructed Grids Based on Explicit Stencil Construction," AIAA paper 2005-0879, 2005.

29. McDonald, S. D., Charest, M. R. J., and Groth, C. P. T., "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.

30. Charest, M. R. J., Groth, C. P. T., and Gauthier, P. Q., "High-order CENO finite-volume scheme for low-speed viscous flows on three-dimensional unstructured mesh," *ICCFD7 - International Conference on Computational Fluid Dynamics*, Hawaii, July 9–13 2012, Paper ICCFD7-1002.

31. Smagorinski, J., "General Circulation Experiments with the Primitive Equations. I: The Basic Experiment," *Monthly Weather Review*, Vol. 91, No. 3, 1979, pp. 99–165.

32. Magnussen, B. F. and Hjertager, B. H., "On mathematical modeling of turbulent combustion with special emphasis on soot formation and combustion," *Proc. Combust. Inst.*, Vol. 16, No. 1, 1977, pp. 719–729.

33. Spalding, D. B., "Mixing and chemical reaction in steady confined turbulent flames," *Proc. Combust. Inst.*, Vol. 13, No. 1, 1971, pp. 649–657.

34. Balaras, E., Benocci, C., and Piomelli, U., "Two-layer approximate boundary conditions for large-eddy simulations," *AIAA J.*, Vol. 34, No. 6, 1996, pp. 1111–1119.

35. Piomelli, U. and Balaras, E., "Wall-layer models for large-eddy simulations," *Ann. Rev. Fluid Mech.*, Vol. 34, 2002, pp. 349–374.

36. Driest, E. V., "On turbulent flow near a wall," *J. Appl. Sci.*, Vol. 23, No. 11, 1956, pp. 1007–1011.

37. Chorin, A. J., "A numerical method for solving incompressible viscous flow problems," *J. Comput. Phys.*, Vol. 2, No. 1, 1967, pp. 12–26.

38. Steger, J. L. and Kutler, P., "Implicit finite-difference procedures for the computation of vortex wakes," *AIAA J.*, Vol. 15, No. 4, 1977, pp. 581–590.

39. Chang, S. L. and Rhee, K. T., "Blackbody radiation functions," *Int. Commun. Heat Mass Transfer*, Vol. 11, No. 5, 1984, pp. 451–455.

40. Turkel, E., "Preconditioned methods for solving the incompressible and low speed compressible equations," *J. Comput. Phys.*, Vol. 72, No. 2, 1987, pp. 277–298.

41. Jameson, A., "Time dependent calculations using multigrid, with applications to unsteady flows past airfoils and wings," AIAA paper 1991-1596, 1991.

42. Merkle, C. L. and Athavale, M., "Time-accurate unsteady incompressible flow algorithms based on artificial compressibility," AIAA paper 87-1137, 1987.

43. Soh, W. Y. and Goodrich, J. W., "Unsteady solution of incompressible Navier-Stokes equations," *J. Comput. Phys.*, Vol. 79, No. 1, 1988, pp. 113–134.

44. Rogers, S. E. and Kwak, D., "Upwind differencing scheme for the time-accurate incompressible Navier-Stokes equations," *AIAA J.*, Vol. 28, No. 2, 1990, pp. 253–262.

45. Rogers, S. E., Kwak, D., and Kiris, C., "Steady and unsteady solutions of the incompressible Navier-Stokes equations," *AIAA J.*, Vol. 29, No. 4, 1991, pp. 603–610.

46. Qian, Z., Zhang, J., and Li, C., "Preconditioned pseudo-compressibility methods for incompressible Navier-Stokes equations," *Science China: Physics, Mechanics and Astronomy*, Vol. 53, No. 11, 2010, pp. 2090–2102.

47. Lee, H. and Lee, S., "Convergence characteristics of upwind method for modified artificial compressibility method," *Int. J. Aeronaut. Space Sci.*, Vol. 12, No. 4, 2011, pp. 318–330.

48. Malan, A. G., Lewis, R. W., and Nithiarasu, P., "An improved unsteady, unstructured, artificial compressibility, finite volume scheme for viscous incompressible flows: Part I. Theory and implementation," *Int. J. Numer. Meth. Engin.*, Vol. 54, No. 5, 2002, pp. 695–714.

49. Malan, A. G., Lewis, R. W., and Nithiarasu, P., "An improved unsteady, unstructured, artificial compressibility, finite volume scheme for viscous incompressible flows: Part II. Application," *Int. J. Numer. Meth. Engin.*, Vol. 54, No. 5, 2002, pp. 715–729.

50. Naff, R. L., Russell, T. F., and Wilson, J. D., "Shape functions for three-dimensional control-volume mixed finite-element methods on irregular grids," *Developments in Water Science*, Vol. 47, Elsevier, 2002, pp. 359–366.

51. Naff, R., Russell, T., and Wilson, J., "Shape functions for velocity interpolation in general hexahedral cells," *Computat. Geosci.*, Vol. 6, 2002, pp. 285–314.

52. Felippa, C. A., "A compendium of FEM integration formulas for symbolic work," *Eng. Computation.*, Vol. 21, No. 8, 2004, pp. 867–890.

53. Ivan, L., *Development of high-order CENO finite-volume schemes with block-based adaptive mesh refinement*, Ph.D. thesis, University of Toronto, 2011.

54. Lawson, C. L. and Hanson, R. J., *Solving least squares problems*, Prentice-Hall, 1974.

55. Mavriplis, D. J., "Revisiting the least-squares procedure for gradient reconstruction on unstructured meshes," AIAA paper 2003-3986, 2003.

56. Park, J. S., Yoon, S.-H., and Kim, C., "Multi-dimensional limiting process for hyperbolic conservation laws on unstructured grids," *J. Comput. Phys.*, Vol. 229, No. 3, 2010, pp. 788–812.

57. Venkatakrishnan, V., "On the Accuracy of Limiters and Convergence to Steady State Solutions," AIAA Paper 93-0880, 1993.

58. Roe, P. L., "Approximate Riemann solvers, parameter vectors, and difference schemes," *J. Comput. Phys.*, Vol. 43, 1981, pp. 357–372.

59. Roe, P. L. and Pike, J., "Efficient Construction and Utilisation of Approximate Riemann Solutions," *Computing Methods in Applied Science and Engineering*, edited by R. Glowinski and J. L. Lions, Vol. VI, North-Holland, Amsterdam, 1984, pp. 499–518.

60. Mathur, S. R. and Murthy, J. Y., "A pressure-based method for unstructured meshes," *Numer. Heat Transfer, Part B*, Vol. 31, No. 2, 1997, pp. 195–215.

61. Karypis, G. and Schloegel, K., "ParMeTis: Parallel graph graph partitioning and sparse matrix ordering library, Version 4.0," http://www.cs.umn.edu/~metis, 2011.

62. Gropp, W., Lusk, E., and Skjellum, A., *Using MPI*, MIT Press, Cambridge, Massachussets, 1999.

63. Groth, C. P. T. and Northrup, S. A., "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.

64. Charest, M. R. J., Groth, C. P. T., and Gülder, Ö. L., "A computational framework for predicting laminar reactive flows with soot formation," *Combust. Theor. Modelling*, Vol. 14, No. 6, 2010, pp. 793–825.

65. Charest, M. R. J., Groth, C. P. T., and Gülder, Ö. L., "Solution of the equation of radiative transfer using a Newton-Krylov approach and adaptive mesh refinement," *J. Comput. Phys.*, Vol. 231, No. 8, 2012, pp. 3023–3040.

66. Saad, Y. and Schultz, M. H., "GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems," *SIAM J. Sci. Stat. Comput.*, Vol. 7, No. 3, 1986, pp. 856–869.

67. Saad, Y., "Krylov Subspace Methods on Supercomputers," *SIAM J. Sci. Stat. Comput.*, Vol. 10, No. 6, 1989, pp. 1200–1232.

68. Brown, P. N. and Saad, Y., "Hybrid Krylov Methods for Nonlinear Systems of Equations," *SIAM J. Sci. Stat. Comput.*, Vol. 11, No. 3, 1990, pp. 450–481.

69. Saad, Y., *Iterative Methods for Sparse Linear Systems*, PWS Publishing Company, Boston, 1996.

70. Dembo, R. S., Eisenstat, S. C., and Steihaug, T., "Inexact Newton Methods," *SIAM J. Numer. Anal.*, Vol. 19, No. 2, 1982, pp. 400–408.

71. Cuthill, E. and McKee, J., "Reducing the bandwidth of sparse symmetric matrices," *Proceedings of the 1969 24th National Conference*, ACM '69, ACM, New York, NY, USA, 1969, pp. 157–172.

72. Yuen, F. T. and Gülder, Ö. L., "Investigation of dynamics of lean turbulent premixed flames by Rayleigh imaging," *AIAA J.*, Vol. 47, No. 12, 2009, pp. 2964–2973.

73. Yuen, F. T. and Gülder, Ö. L., "Premixed turbulent flame front structure investigation by Rayleigh scattering in the thin reaction zone regime," *Proc. Combust. Inst.*, Vol. 32, 2009, pp. 1747–1754.

74. Hernández-Pérez, F. E., Yuen, F. T. C., Groth, C. P. T., and Gülder, Ö. L., "LES of a laboratory-scale turbulent premixed Bunsen flame using FSD, PCM-FPI and thickened flame models," Vol. 33, 2011, pp. 1365–1371.

75. Shahbazian, N., Groth, C. P. T., and Gülder, Ö. L., "Assessment of presumed PDF models for large eddy simulation of turbulent premixed flames," AIAA Paper 2011-0781, 2011.

76. Jarrin, N., Benhamadouche, S., Laurence, D., and Prosser, R., "A synthetic-eddy-method for generating inflow conditions for large-eddy simulations," *Int. J. of Heat Fluid Fl*, Vol. 27, No. 4, 2006, pp. 585–593.

77. Shepherd, I., "Flame surface density and burning rate in premixed turbulent flames," *Proc. Combust. Inst.*, Vol. 26, 1996, pp. 373–379.

78. Sachdev, J. S., Groth, C. P. T., and Gottlieb, J. J., "A parallel solution-adaptive scheme for multi-phase core flows in solid propellant rocket motors," *Int. J. Comput. Fluid Dyn.*, Vol. 19, No. 2, 2005, pp. 159–177.

79. Gao, X. and Groth, C. P. T., "A parallel adaptive mesh refinement algorithm for predicting turbulent non-premixed combusting flows," *Int. J. Comput. Fluid Dyn.*, Vol. 20, No. 5, 2006, pp. 349–357.

80. Gao, X. and Groth, C. P. T., "A parallel solution-adaptive method for three-dimensional turbulent non-premixed combusting flows," *J. Comput. Phys.*, Vol. 229, No. 9, 2010, pp. 3250–3275.

81. Gao, X., Northrup, S., and Groth, C. P. T., "Parallel solution-adaptive method for two-dimensional non-premixed combusting flows," *Prog. Comput. Fluid. Dy.*, Vol. 11, No. 2, 2011, pp. 76–95.

82. Domingo, P., Vervisch, L., Payet, S., and Hauguel, R., "DNS of a premixed turbulent V flame and LES of a ducted flame using a FSD-PDF subgrid scale closure with FPI-tabulated chemistry," *Combust. Flame*, Vol. 143, No. 4, 2005, pp. 566–586.