A High-Order Finite-Volume Scheme for Large-Eddy Simulation of Premixed Flames on Multi-Block Cartesian Mesh

by

Prabakar Regmi

A thesis submitted in conformity with the requirements for the degree of Masters of Applied Science
Graduate Department of Institute of Aerospace Studies
University of Toronto

Copyright © 2012 by Prabakar Regmi
Abstract

A High-Order Finite-Volume Scheme for Large-Eddy Simulation of Premixed Flames on Multi-Block Cartesian Mesh

Prabhakar Regmi
Masters of Applied Science
Graduate Department of Institute of Aerospace Studies
University of Toronto
2012

Large-eddy simulation (LES) is emerging as a promising computational tool for reacting flows. High-order schemes for LES are desirable to achieve improved solution accuracy with reduced computational cost. In this study, a parallel, block-based, three-dimensional high-order central essentially non-oscillatory (CENO) finite-volume scheme for LES of premixed turbulent combustion is developed for Cartesian mesh. This LES formulation makes use of the flame surface density (FSD) for subfilter-scale reaction rate modelling. An algebraic model is used to approximate the FSD. A detailed explanation of the governing equations for LES and the mathematical framework for CENO schemes are presented. The CENO reconstruction is validated and is also applied to three-dimensional Euler equations prior to its application to the equations governing LES of reacting flows.
Acknowledgements

I owe a sincere and deepest debt of gratitude to my supervisor, Prof. Dr. C. P. T. Groth, for believing in me, and for guiding and supporting me during this research work. It is with his acceptance that I am working in the field of CFD. He has always motivated me to have an in-depth understanding of the essential topics in CFD before addressing and solving problems associated with it. I would like to thank him for patiently letting me take the courses on CFD, combustion and turbulence modelling, before beginning my graduate research. Because of him, graduate study in CFD has been a smooth transition for a physics undergraduate student like me.

Additionally, this research has greatly benefited due to many helpful suggestions and advices from colleagues, doctoral and post-doctoral candidates in the CFD Lab. Particularly, I’d like to thank Marc Charest for providing me with his implementation of high-order CENO scheme on uniform Cartesian grids. Similarly, Nasim Shahbazian has helped me with various issues related to the in-house LES codes. I would also like to extend my gratitude to Chris Lam, Jenny Zhang, Lucian Ivan, Scott Northrup, and all other members of CFD and Propulsion group who have directly or indirectly helped me.

It should be acknowledged that computational resources for performing most of the calculations reported here 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.

Last, but not the least, I would like to thank my God, family and friends for their constant love and support. My parents, Mr. Murari Regmi and Mrs. Rekha Kumari Regmi, have always been pillars of support to me, and this thesis is dedicated to them. Thanks are also due to my brother Pramesh, and my friends, Malte, Hyoyoung, Stefanie, Elisabeth, and Alamin, for being there in good and bad times. Without your support, this work wouldn’t have been possible.

Toronto, 2012

Prabhakar Regmi
Contents

Abstract ii

Acknowledgments iii

Contents iv

List of Figures vii

1 Introduction 1

1.1 Motivation ............................................... 2

1.1.1 Turbulent Premixed Combustion .......................... 2

1.1.2 Numerical Treatment of Turbulent Combustion .......... 5

1.1.3 High-Order Central Essentially Non-Oscillatory (CENO) Scheme . . . . . . . 6

1.2 Thesis Objectives ........................................ 7

1.3 Overview of the Thesis ................................... 8

2 Governing Equations for LES of Premixed Combustion 9

2.1 Favre-Filtered Navier-Stokes Equations ................... 9

2.1.1 LES Filtering ........................................... 9

2.1.2 Favre-Filtered Equations .............................. 10

2.2 Subfilter-Scale Stress Modelling .......................... 13
2.3 Subfilter-Scale Scalar Transport Modelling ................................................. 14
2.4 Subfilter-Scale Turbulent Transport Modelling ............................................. 14
2.5 Reaction Rate Modelling .............................................................................. 15
  2.5.1 Modelled Transport Equation for FSD ...................................................... 16
  2.5.2 Algebraic Model of Boger for FSD .......................................................... 17

3 Second-Order Finite-Volume Formulation ......................................................... 18
  3.1 Governing Equations in Vector Form .......................................................... 19
  3.2 Godunov-type Finite-Volume Method .......................................................... 20
  3.3 Flux Evaluation ......................................................................................... 22
    3.3.1 Inviscid (Hyperbolic) Flux Evaluation .................................................. 22
    3.3.2 Viscous (Elliptic) Flux Evaluation ........................................................ 24
  3.4 Solution Reconstruction ............................................................................. 25
    3.4.1 Godunov’s Method and Piecewise Constant Reconstruction ................. 26
    3.4.2 Piecewise Linear Reconstruction .......................................................... 26
  3.5 Time Marching Schemes ............................................................................ 30
  3.6 Numerical Example – 1D Shock-tube Problem ............................................. 31

4 High-Order CENO Schemes ............................................................................ 33
  4.1 Overview of the CENO Scheme .................................................................... 34
  4.2 $k$-Exact Least-Squares Reconstruction ..................................................... 35
    4.2.1 Stencils Used for Reconstruction ......................................................... 36
    4.2.2 The Least-Squares Problem .................................................................. 38
  4.3 Smoothness Indicator ................................................................................ 43
  4.4 Flux Evaluation ......................................................................................... 44
    4.4.1 Inviscid Flux Evaluation ...................................................................... 45
4.4.2 Viscous Flux Evaluation ........................................... 46
4.5 Source Vector Evaluation ........................................... 48
4.6 Boundary Conditions .................................................. 48

5 Results, Verification and Discussion ................................. 50

5.1 Verification of the CENO scheme ................................. 50
  5.1.1 Demonstration of High-Order Accuracy .......................... 51
  5.1.2 Application to the Euler Equations .............................. 57

5.2 Application of CENO Scheme to LES of Turbulent Premixed Flames .......................... 60
  5.2.1 LES of Non-Reacting Flows .................................... 63
  5.2.2 LES of Turbulent Premixed Combustion .......................... 66

6 Concluding Remarks .................................................... 70

6.1 Conclusions ............................................................ 70
6.2 Recommendations for Future Work ................................. 71

References ................................................................. 79
List of Figures

1.1 Regime diagram for premixed turbulent combustion as proposed by Peters. . . . 4

3.1 Piecewise constant and piecewise linear reconstructions. . . . . . . . . . . . 27

3.2 Depiction of the oscillations that occur near a shock wave for an unlimited second-
order FV scheme. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 28

3.3 First- and second-order solutions (density distributions) of the shock-tube prob-
lem at t = 7 ms. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 32

4.1 One- and two-layer reconstruction stencils. . . . . . . . . . . . . . . . . . . . . . 38

4.2 Variation of function \( f(\gamma) = \frac{\gamma}{(1-\gamma)} \) . . . . . . . . . . . . . . 44

5.1 Reconstruction of radial cosine function on a \((30 \times 30 \times 30)\) grid. . . . . . 53

5.2 Convergence studies for radial cosine function reconstruction. . . . . . . . . . . 54

5.3 Reconstruction of the Abgrall function on a \((180 \times 180 \times 6)\) grid. . . . . . 56

5.4 \(|E|_1\) convergence study for the discontinuous Abgrall function. . . . . . . . . . 57

5.5 Plots of the periodic density wave. . . . . . . . . . . . . . . . . . . . . . . . . . . . 59

5.6 Density variations in a shock-box problem. . . . . . . . . . . . . . . . . . . . . . 61

5.7 Initial Turbulent Velocity Field. Arrows denote the velocity vectors on the faces,
contour values denote the velocity component parallel to the x-axis, \(u\) . . . . . . 62

5.8 Decay of total turbulent kinetic energy computed using limited CENO scheme
on the given grid size. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 64
5.9 Smoothness Indicators acting (on $\rho$) at the last time step before 100 ms: red regions depict the non-smooth areas where smoothness indicators are switched on. 65

5.10 Comparison of decay of total turbulent kinetic energy computed using unlimited second-order LPL scheme and unlimited fourth-order CENO scheme on the mesh sizes of $(32 \times 32 \times 32)$ and $(64 \times 64 \times 64)$ cells. 65

5.11 Schematic of the testcase setup for the premixed flame in a turbulent box. 66

5.12 Fourth-order instantaneous progress variable distribution in a $xz$-plane of the three-dimensional flame at various time. 67

5.13 Predicted instantaneous flame surfaces defined as the iso-surfaces of $\tilde{c}^* = 0.5$ at $t = 0.25$ ms. 68

5.14 Depiction of the action of smoothness indicators on the progress variable, $c$, at the last time step before $t = 0.25$ ms 69
Chapter 1

Introduction

Climate change has become a major issue, globally. As such, researchers have been looking for ways to make combustion systems more eco-friendly. Generally, combustion results in emission of various environmentally harmful gases. The prominent products of combustion are the so-called greenhouse gases, like carbon dioxide (CO$_2$), which play a major role in global warming. Moreover, air-pollutants such as nitrogen oxides (NO$_x$), carbon monoxide (CO) and unburnt hydrocarbons (UHC) are produced during combustion. To achieve the goal of reduced emissions, it is imperative for researchers to have a fundamental understanding of combustion processes. However, the mechanisms with which fuels burn to produce these pollutants are very complex. Thankfully, due to the advancement in computational resources and improvement in numerical methods, scientific computing can be used to obtain detailed and multi-scale information about complex combustion processes.

Computational fluid dynamics (CFD) is the “third-approach” [1] used to study fluid-dynamics, which uses various numerical methods and high-speed computing to predict and study the nature of fluid flows. The other two approaches are theoretical and experimental fluid dynamics. Due to the lower costs and easier availability of high-performance computational resources, CFD has been extremely popular among researchers and engineers to study combustion. As such, there exists a desire to develop more efficient and robust CFD schemes. This work aims for achieving this, with the primary goal of developing “high-order finite-volume schemes” for “large-eddy simulation” of “premixed combustion”. Each of these terms will be explained in detail in the following sections.
1.1 Motivation

Generally, research in the field of high-order methods for CFD is motivated by the purpose of obtaining solutions with a desired level of accuracy at a reduced computational cost and lower memory usage. While developing high-order schemes for reacting flows, it is worthwhile to first understand the complications associated with turbulent combustion modelling. Thereafter, the usage of a high-order finite-volume method, known as central essentially non-oscillatory (CENO) scheme, will be discussed.

1.1.1 Turbulent Premixed Combustion

Designing computational models for reactive flows is challenging because of the complex interactions between chemistry and turbulence in these flows. These interactions are complex due to the wide range of length and time scales associated with turbulent flows. Turbulent flows are composed of vortex-like structures called ‘eddies’. Turbulent kinetic energy, $k$, is transferred from larger eddies to smaller eddies, via an energy-cascade process. At the scale of the smallest eddies, kinetic energy is dissipated to heat due to the action of molecular viscosity [2]. The size of the largest eddy is known as integral turbulent length scale, $\Lambda$. The ratio of the integral length scale to the root-mean-square (RMS) velocity fluctuation, $u'$, defines the turbulent time scale, $\tau_t$, as

$$\tau_t \approx \frac{\Lambda}{u'}.$$  \hspace{1cm} (1.1)

The RMS velocity fluctuation and the integral length scale can be related to the turbulent kinetic energy, $k$, and the rate of its dissipation, $\epsilon = -\frac{dk}{dt}$, in the following manner:

$$u' \approx \sqrt{k}, \quad \Lambda \approx \frac{u'^3}{\epsilon}.$$  \hspace{1cm} (1.2)

The turbulent transport and the viscous forces can be compared using a non-dimensional quantity called turbulent Reynold’s number, $Re_\Lambda$, which is given by

$$Re_\Lambda = \frac{u'\Lambda}{\nu},$$  \hspace{1cm} (1.3)

where $\nu$ is the kinematic viscosity. Based on the idea of energy cascade, Kolmogorov [3] postulated that from the larger eddies, smaller eddies should receive energy at approximately the same rate that the smallest eddies dissipate. Hence, the motion at the smallest scales should depend only on the kinematic viscosity and the rate at which the larger eddies supply energy, i.e. the dissipation rate $\epsilon$. Following dimensional analysis, the so-called Kolmogorov scales for
Chapter 1. Introduction

length, \( \eta \), time, \( \tau_\eta \), and velocity, \( u_\eta \), can then defined as

\[
\eta \approx \left( \frac{\nu^3}{\epsilon} \right), \quad \tau_\eta \approx \left( \frac{\nu}{\epsilon} \right)^{1/2}, \quad u_\eta = (\nu \epsilon)^{1/4}.
\] (1.4)

The chemical time scale, \( \tau_c \), approximates the speed at which reactants are converted into products. It is defined as the ratio of laminar flame thickness, \( \delta_L \), to the laminar flame speed, \( s_L \), given by

\[
\tau_c \approx \frac{\delta_L}{s_L}.
\] (1.5)

Using these definitions, two important dimensionless numbers are introduced by the classical scaling arguments:

- **The Damköhler number, Da**: ratio of turbulent time scale, \( \tau_t \), to chemical time scale, \( \tau_c \), given by

\[
Da = \frac{\tau_t}{\tau_c} \approx \frac{\Lambda}{u'} \frac{\delta_L}{s_L}.
\] (1.6)

It can be inferred, from this definition, that a high Damköhler number corresponds to a case where turbulent time scale is larger than the chemical time scale, which means that the reaction zones are thin. In this case, the inner flame structure locally exhibit similarities to laminar flame, whereas globally, the flame is wrinkled due to the turbulence.

- **The Karlovitz number, Ka**: ratio of chemical time scale, \( \tau_c \), to Kolmogorov time scale, \( \tau_\eta \), given by

\[
Ka = \frac{\tau_c}{\tau_\eta} \approx \left( \frac{\delta_L}{\eta} \right)^2 \approx \left( \frac{u_\eta}{s_L} \right)^2,
\] (1.7)

A very low Karlovitz number corresponds to the case where the flame is thinner than the smallest scale of turbulence and therefore, the turbulent fluctuations are not able to penetrate the inner flame structure.

Depending on when and how the reactants are mixed, combustion can be classified as premixed, non-premixed or partially-premixed. During a premixed combustion, fuels and oxidizers are perfectly mixed at the molecular level before sending them into a reactor. In a non-premixed combustion, they are introduced separately into the combustion chamber, and they mix during the reaction process. The reactants need not be mixed perfectly before combustion. They can also just be partially-premixed. This work investigates premixed combustion, and therefore it is worthwhile to review key aspects and characteristics of turbulent premixed flames.

Premixed flames can be classified using the turbulent regime diagrams, which are plotted as functions of the ratios of length, \( \Lambda/\delta_L \), and velocity, \( u'/s_L \). Figure 1.1 demonstrates such a
regime diagram as proposed by Peters [4]. In this case, five distinct regimes can be identified as follows:

- **Laminar flames regime**: This is the zone where $\text{Re}_\Lambda < 1$. The $\text{Re}_\Lambda = 1$ line separates laminar flame regime with turbulent combustion regime. Practical combustors generally operate in the turbulent regime.

- **Wrinkled flamelets regime**: This is a turbulent regime with $u' < S_L$. Here, interaction between turbulence and combustion is limited, because laminar propagation of the flame dominates. The effect of large turbulent eddies on the wrinkling of the flame-front is minimal.

- **Corrugated flamelets regime**: This is another turbulent flamelet regime ($K_a < 1$), where $u' > S_L$. Here, large turbulent motions causes flame-front interactions leading to the formation of separated pockets of fresh and burnt gases. Eventually, changes in flame topology is observed in this regime.

- **Thin reaction zones regimes**: This is the regime where $1 < K_a < 100$. In this regime, since the Kolmogorov scales are smaller than the flame-thickness, the smallest eddies can enter the preheat layer of the flame and distort the locally laminar structure of the flame-front.
• **Broken reaction zones regimes**: This is the turbulent regime for which $\mathrm{Ka} > 100$. Turbulent motions have shorter characteristic times than the chemical reaction time in this zone. The smallest eddies can penetrate the inner structure of the flame, causing significant thickening of the flame-front.

### 1.1.2 Numerical Treatment of Turbulent Combustion

In general, there are three popular ways for the numerical treatment of turbulent premixed combustion: direct numerical simulation (DNS), Reynolds-averaged Navier-Stokes (RANS), and large-eddy simulation (LES). DNS does not rely on turbulent models. Instead, it solves the full range of scales in the governing equations including both the turbulent and chemical scales. In particular, it computes the solution of the conservation equations down to the Kolmogorov scales. Therefore, DNS can be used to predict any flow quantity of interest, even those that are impossible to measure experimentally [5]. Unfortunately, DNS can be extremely expensive and is limited to the availability of high performance computing facilities, since every detail in the flow has to be incorporated. Thus, it is impractical for many engineering flow problems. In addition, to resolve all the scales of motion, the number of grid points should be proportional to $\mathrm{Re}^9/4$ for non-reacting flows. For reacting flows, the grid size should be small enough to resolve the combustion chemistry as well. For this reason, DNS is limited to low Reynolds-number flows with simple geometries.

RANS is the least expensive among the three mentioned schemes. This approach applies time averaging over a period much longer than the correlation time of turbulence. It therefore resolves large-scale mean flows. Consequently, the effects of all the turbulent fluctuations on the mean quantities must be modelled. A shortcoming of RANS approach is that this model must represent full range of turbulent scales, and therefore it is affected by boundary conditions. For this reason, a RANS model for universal use remains elusive.

Large-eddy simulation is an alternative approach for turbulence modelling, which has gained popularity in the last 10–15 years due to its balance between cost and accuracy of turbulence modelling. In LES, as its name suggests, numerical simulation of large-scale flow features are carried out exactly, and only the effects of small-scale (smaller than a prespecified filter scale) turbulence is modelled. The spatial terms in the Navier-Stokes equations are filtered using a filter-width which can be resolved on an affordable computational grid [6]. Unlike RANS, LES can explicitly resolve most of the turbulence in a system. The computational cost to carry out LES is less compared to DNS (however, LES is more expensive than RANS). Compared to RANS, LES ameliorates the involvement of modelling in a flow problem and hence reduces the
uncertainty associated to these modelling procedures.

Large-eddy simulation is used as a wide-spread tool for non-reacting flows. However, for reacting flows, it is still developing. A considerable complication for LES of turbulent premixed flame arises from the fact that the reactions take place in very thin flame-fronts, which cannot be resolved in the computational grids. Naturally then, these scales, known as the sub-filter scales (SFS), require modelling. Since 1990’s [7], researchers have developed various conceptually different SFS models to account for the chemistry-turbulence interactions, like G-equation approach [8, 9], probability density functions (PDFs) [10], thickened flame (TF) [11], eddy-dissipation concept (EDC) [12] and laminar flamelet models [13, 14]. This research makes use of the laminar flamelet model for LES. This model is based on the popular assumption that premixed turbulent flames with low turbulent intensities can be considered as a collection of one-dimensional laminar flamelets, which separate burnt and unburnt gases. It should be noted that the laminar flamelet model works well for flows in the flamelet (wrinkled and corrugated) regime shown in Figure 1.1. Lin [15] has demonstrated that this model can be used to predict flames even in the thin reaction zones. The modelling process involves an additional transport equation for the reaction progress variable. The closure for the production and the destruction terms in this transport equation can be obtained from the idea of flame surface density (FSD). There exists various algebraic expressions, assuming equilibrium between production and destruction, to model the flame surface density, like the ones proposed by Boger et al. [16] and Charlette et al. [17]. Similarly, accounting for the non-equilibrium transport of FSD, Hawkes and Cant [18] have proposed a transport equation, too. These algebraic and transport equation models for FSD is discussed in Section 2.5.

1.1.3 High-Order Central Essentially Non-Oscillatory (CENO) Scheme

For several decades now, researchers have been very keen in developing high-order CFD techniques to accomplish the goal of decreased overall computational cost and memory requirements. In the past, various high-order finite-volume schemes like essentially non-oscillatory (ENO) [19], weighted essentially non-oscillatory (WENO) [20, 21, 22], and central essentially non-oscillatory (CENO) [23, 24] schemes are proposed. A major challenge in all these schemes has been to achieve high-order spatial discretizations while maintaining the solution monotonicity near the discontinuities and shocks. High-order spatial discretizations in these schemes are achieved by high-order polynomial reconstruction of the solution, based on the average solution quantities in the chosen reconstruction stencil around the cell.

In ENO scheme, reconstruction is performed using the “adaptive-stencil” strategy, where sten-
cils of neighbouring cells leading to the “smoothest” reconstruction of the solution are selected and those containing discontinuities are avoided. ENO was extended to WENO, where a solution-dependent weighting is assigned to each and every reconstruction stencil such that all these stencils are incorporated for solution reconstruction appropriately. Unfortunately, these schemes have not gained much in popularity due to the high computational cost attached to them. Additionally, in practical problems involving multiple space dimensions, these schemes are complex as they result in poorly conditioned coefficient matrices while solving for a piecewise polynomial solution representation in each cell [24].

The central essentially non-oscillatory (CENO) scheme proposed by Ivan and Groth [23, 24, 25] is possibly a more flexible high-order alternative. This scheme uses a fixed central stencil for reconstruction. CENO is based on hybrid solution reconstruction procedure which combines \( k \)-exact reconstruction (from a fixed central stencil) with a monotonicity preserving piecewise linear reconstruction algorithm. A smoothness indicator determines whether solutions are resolved on the current computational cell. The limited piecewise linear reconstruction (with, \( k = 1 \)) is applied to cells with under-resolved solution content (i.e., in regions with discontinuities). The high-order \( k \)-exact procedure, on the other hand, is used for cells where solutions are fully resolved. In this way, CENO reconstruction avoids problems associated with capturing smooth extrema as well as discontinuities. Even if this scheme does not provide uniform high-order accuracy for non-smooth solutions, it is easily extendable to multi-dimensions and general unstructured mesh [26].

1.2 Thesis Objectives

The objective of this thesis is to apply high-order CENO scheme to the equations governing LES of premixed turbulent combustion. A second-order scheme for LES using FSD model has been developed previously by Lin [15]. A high-order CENO finite-volume method using adaptive mesh refinement for structured grids has been developed by Ivan and Groth, which has been implemented to solve Euler and Navier-Stokes equations in one- and two-dimensions [24]. This scheme is extended to three-dimensional Cartesian mesh and implemented on Euler and Navier-Stokes equations by Rashad [27]. This research uses high-order CENO scheme to solve the filtered Navier-Stokes equations governing LES of reactive turbulent flows. Within the scope of this thesis, the high-order algorithm for LES is implemented on Cartesian grids.
1.3 Overview of the Thesis

Chapter 2 introduces Favre-filtered Navier-Stokes equations that govern LES procedure. This chapter outlines the filtering process which separates resolved terms with unresolved subfilter-scale terms. Thereafter, modelling techniques for the unresolved terms are discussed. The finite-volume formulation of the governing equations (that are outlined in Chapter 2) are presented in Chapter 3. This chapter discusses the general Godunov-type finite-volume method which is first-order accurate, and explains how this scheme is extendible to second-order accuracy. Chapter 4 then focuses on the theoretical insight on the development and extension of the finite-volume scheme to high-order using the CENO approach. This chapter discusses various processes essential for the development of CENO schemes, including the $k$-exact least-squares reconstruction and smoothness indicators. This chapter concludes the theoretical background presented in this thesis.

Chapter 5 presents the results generated from the implementation of CENO schemes on LES equations. First, the validity of CENO reconstruction is assessed, using reconstruction for both smooth and discontinuous functions. Then, application of this scheme to the Euler equations is validated. Thereafter, the results produced due to the application of CENO schemes on the LES equations are presented. The LES results of both the cases involving non-reacting flows as well as reacting turbulent flows are presented. Finally, Chapter 6 summarizes the results and presents some recommendations for further research on this topic.
Chapter 2

Governing Equations for LES of Premixed Combustion

This chapter outlines and explains the governing equations for large-eddy simulation using the flame-surface density model. More in-depth discussions about turbulence and combustion modelling can be found in various textbooks including those of Pope [28], Saguat [29], and Poinsot and Veynante [30].

2.1 Favre-Filtered Navier-Stokes Equations

2.1.1 LES Filtering

According to the philosophy of LES, large scales are fully resolved and small scales are modelled. Therefore, there is a need for a filtering operation that separates these scales, such that any solution variable, $\phi$, could be decomposed to a filtered quantity, $\bar{\phi}$, which is completely resolved, and a fluctuating sub-filter scale (SFS) component, $\phi'$, that requires modelling, as follows:

$$\phi = \bar{\phi} + \phi'.$$

(2.1)

A spatial low-pass filtering operation helps determine the contribution of the resolved scales, and is given by [31]

$$\bar{\phi}(\vec{x}) = \int_{-\infty}^{+\infty}\phi(\vec{\psi})G(\vec{x} - \vec{\psi}; \Delta)d^3\vec{\psi},$$

(2.2)

where $\Delta$ is the filter width, and $G$ is a normalized LES filter function, requiring that

$$\int_{-\infty}^{+\infty} G(\vec{x} - \vec{\psi}; \Delta)d^3\vec{\psi} = 1.$$

(2.3)
In physical space, the simplest type of filter is the volume-averaged box filter, defined as

\[
G(\vec{x} - \vec{\psi} ; \Delta) = \begin{cases} 
1/\Delta^3, & \text{if } |x_i - \psi_i| \leq \Delta x_i/2, \\
0, & \text{otherwise.}
\end{cases}
\] (2.4)

Similarly, another common spatial filter is known as Gaussian filter, given by

\[
G(\vec{x} - \vec{\psi} ; \Delta) = \left( \frac{6}{\pi \Delta^2} \right)^{3/2} \exp \left[ -6 \frac{|x - \psi|^2}{\Delta^2} \right].
\] (2.5)

For compressible and reactive flows, it is a common practice to use the mass-weighted Favre filtering procedure. This filtering for compressible flows guarantees that there is no unclosed terms in the filtered-continuity equation, thereby eliminating the need to model the high-order filtered quantities in this equation. Favre-filtering is defined by

\[
\tilde{\phi}(\vec{x}) = \frac{1}{\bar{\rho}} \int_{-\infty}^{+\infty} \rho \phi(\vec{\psi}) G(\vec{x} - \vec{\psi}) d^3\vec{\psi},
\] (2.6)

where \( \rho \) is the density of the mixture in the flow.

In this work, an implicit filtering approach is used, which uses computational grid and discretization operators as filters for governing equations. The filter width is taken to be proportional to the grid size, and might be expressed as

\[
\Delta = k(\Delta x \Delta y \Delta z)^{1/3},
\] (2.7)

where \( \Delta x, \Delta y, \) and \( \Delta z \) are the cell size in \( x, \) \( y, \) and \( z \)-directions, respectively, and \( k \) is a constant known as filter-grid-ratio (FGR). In this research, the FGR is set to be 2. For an implicit filtering procedure, unlike in the explicit approach, there is no need to specify explicitly the filter function. For more information and further discussion on the filtering procedure, the reader is referred to the works by Scotti et al. [32, 33].

### 2.1.2 Favre-Filtered Equations

The mathematical description of LES of turbulent premixed combustion is provided by applying the Favre-filtering operation described above to the Navier-Stokes equations governing compressible flows of a thermally perfect reactive mixture of gases. Using the low-pass spatial filtering described in the previous section, relevant solution quantities, \( \phi \), are filtered, Favre-filtered or evaluated in terms of filtered quantities to give \( \tilde{\phi} \), \( \tilde{\phi} \) and \( \tilde{\phi} \), respectively. The filtered Navier-Stokes equations are given by
• mass (continuity) equation:

$$\frac{\partial \bar{\rho}}{\partial t} + \frac{\partial (\bar{\rho} \bar{u}_i)}{\partial x_i} = 0,$$  \hspace{1cm} (2.8)

• momentum equations:

$$\frac{\partial (\bar{\rho} \bar{u}_i)}{\partial t} + \frac{\partial (\bar{\rho} \bar{u}_i \bar{u}_j + \delta_{ij} \bar{p})}{\partial x_i} - \frac{\partial \bar{\tau}_{ij}}{\partial x_i} = \frac{\partial \sigma_{ij}}{\partial x_i} + \frac{\partial (\bar{\tau}_{ij} - \bar{\tau}_{ij})}{\partial x_i}.$$  \hspace{1cm} (2.9)

• total-energy equation:

$$\frac{\partial (\bar{\rho} \bar{E})}{\partial t} + \frac{\partial [(\bar{\rho} \bar{E} + \bar{p}) \bar{u}_i]}{\partial x_i} + \frac{\partial \bar{q}_i}{\partial x_i} = - \frac{\partial [\overline{\rho} (\bar{h}_s \bar{u}_i - \bar{h}_s \bar{u}_i)]}{\partial x_i}$$

$$\text{III} - \frac{\partial [\sum_{n=1}^{N_s} \Delta h_{f,n}^0 \bar{p} (Y_n \bar{u}_i - \bar{Y}_n \bar{u}_i)]}{\partial x_i}$$

$$\text{IV} - \frac{1}{2} \frac{\partial [\overline{\rho} (u_i \bar{u}_j + \bar{u}_i u_j \bar{u}_j)]}{\partial x_i}$$

$$\text{V} + \frac{\partial (\bar{\tau}_{ij} \bar{u}_j - \bar{\tau}_{ij} \bar{u}_j)}{\partial x_i} + \frac{\partial (\bar{\tau}_{ij} \bar{u}_j - \bar{\tau}_{ij} \bar{u}_j)}{\partial x_i}$$

$$\text{VI} - \frac{\partial (\bar{q}_i - \bar{q}_i)}{\partial x_i},$$  \hspace{1cm} (2.10)

• chemical transport equation ($N_s$ species, $n = 1, \ldots, N_s$):

$$\frac{\partial (\bar{\rho} Y_n)}{\partial t} + \frac{\partial (\bar{\rho} \bar{Y}_n \bar{u}_i)}{\partial x_i} + \frac{\partial \bar{J}_{n,i}}{\partial x_i} = - \frac{\partial [\overline{\rho} (Y_n \bar{u}_i - \bar{Y}_n \bar{u}_i)]}{\partial x_i} + \frac{\partial (\bar{J}_{n,i} - \bar{J}_{n,i})}{\partial x_i}$$

$$\text{IX} + \frac{\partial (\bar{J}_{n,i} - \bar{J}_{n,i})}{\partial x_i}$$

$$\text{X} + \frac{\partial (\bar{J}_{n,i} - \bar{J}_{n,i})}{\partial x_i}$$

$$\text{XI} - \frac{\partial (\bar{q}_i - \bar{q}_i)}{\partial x_i},$$  \hspace{1cm} (2.11)

• equation of state:

$$\bar{p} = \bar{\rho} R \bar{T} + \sum_{n=1}^{N_s} R_n \bar{p} (Y_n \bar{T} - \bar{Y}_n \bar{T}),$$  \hspace{1cm} (2.12)

where $\bar{\rho}$ is the filtered mixture density, $\bar{u}_i$ is the Favre-filtered mixture velocity, $\bar{p}$ is the filtered mixture pressure given by the equation of state (Eq. (2.12)), $\bar{h}_s$ is the sensible enthalpy and $\Delta h_{f,n}^0$ is the heat of formation for species $n$, $\bar{T}$ is the mixture temperature, $\bar{Y}_n$ is the mass
fraction of species $n$, and $\bar{\omega}$ is the filtered reaction rate. The Favre-filtered total mixture energy (including chemical energy), $\tilde{E}$, is given by

$$
\tilde{E} = \bar{h}_n + \sum_{n=1}^{N_s} \tilde{Y}_n \Delta h^0_{i,n} - \frac{\bar{p}}{\rho} + \frac{1}{2} \bar{u}_i \bar{u}_j .
$$

(2.13)

The viscous stress tensor, $\tilde{\tau}_{ij}$, heat flux, $\tilde{q}_i$, and species molecular flux, $\tilde{J}_{n,i}$, are all computed using the filtered quantities, and are given by

$$
\tilde{\tau}_{ij} = 2\tilde{\mu} \left( \tilde{S}_{ij} - \frac{1}{3} \delta_{ij} \tilde{S}_{kk} \right),
$$

(2.14)

$$
\tilde{q}_i = -\tilde{\kappa} \frac{\partial \tilde{T}}{\partial x_i} - \bar{\rho} \sum_{n=1}^{N_s} \bar{h}_n \tilde{D}_n \frac{\partial \tilde{Y}_n}{\partial x_i},
$$

(2.15)

$$
\tilde{J}_{n,i} = -\bar{\rho} \tilde{D}_n \frac{\partial \tilde{Y}_n}{\partial x_i},
$$

(2.16)

respectively, where $\tilde{\mu}$ is the mixture viscosity, $\bar{\kappa}$ is the mixture thermal conductivity, and $\tilde{D}_n$ is the diffusivity of species $n$ with respect to the mixture. The strain rate tensor, $\tilde{S}_{ij}$, is defined in terms of filtered velocity field and given by

$$
\tilde{S}_{ij} = \frac{1}{2} \left( \frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i} \right).
$$

(2.17)

The terms I – IX on the right hand side of the governing equations are the subfilter-scale terms resulting from the filtering process, which represent the effect of the unresolved scales on the resolved solution content. These unclosed quantities require modelling. Terms II, VI, VIII and X are generally neglected assuming that $\tilde{\tau}_{ij}$, $\tilde{q}_i$ and $\tilde{J}_{n,i}$ are approximated by $\bar{\tau}_{ij}$, $\bar{q}_i$ and $\bar{J}_{n,i}$, respectively. The subfilter-scale viscous diffusion term VII is much smaller than the other terms in the equation that require modelling. Therefore, it is neglected, too. Finally, there is no conventional closure for the subfilter temperature-species correlation term, term XII; however it is a common practice to neglect this term. Thus, only terms I, III, IV, V, IX and XI needs to be modelled with appropriate closure strategies. Specifically, the closure problem for LES of reacting flows involves the following terms:

- the subfilter-scale stresses, $\sigma_{ij} = \bar{\rho} (\bar{u}_i \bar{u}_j - \bar{u}_i \bar{u}_j)$;
- the subfilter-scale turbulent transport, $(\bar{u}_i \bar{u}_j \bar{u}_j - \bar{u}_i \bar{u}_j \bar{u}_j) / 2$;
- the subfilter-scale enthalpy flux, $\bar{\rho} (\bar{h}_s \bar{u}_i - \bar{h}_s \bar{u}_i)$;
Chapter 2. Governing Equations for LES of Premixed Combustion

- the subfilter-scale species flux, \( \tilde{\rho} \left( \tilde{Y}_n u_i - \tilde{Y}_n \tilde{u}_i \right) \); and
- the filtered reaction rate, \( \tilde{\omega}_n \).

The appropriate closure strategies for these terms, used in this work, are discussed in the sections to follow.

### 2.2 Subfilter-Scale Stress Modelling

The subfilter-scale stresses, \( \sigma_{ij} \), appearing in term I, arise from non-linearity of convective terms. There are various ways for modelling these stresses. In this work, an algebraic model proposed by Smagorinsky and a one-equation model are implemented. The Smagorinsky model \[34\] uses a similar concept as the Boussinesq approximation and the eddy-viscosity model (which is commonly used in RANS approach) to model the stress tensor. Using this model, the SFS stress for compressible flows is given by

\[
\sigma_{ij} = 2\tilde{\rho}\nu_t \left( \tilde{S}_{ij} - \frac{\delta_{ij}}{3} \tilde{S}_{kk} \right) - \frac{\delta_{ij}}{3} \sigma_{kk},
\]

(2.18)

where \( \nu_t \) is the subfilter-scale eddy viscosity, given by

\[
\nu_t = (C_s \Delta)^2 |\tilde{S}|,
\]

(2.19)

\( |\tilde{S}| = \sqrt{2\tilde{S}_{ij}\tilde{S}_{ij}} \) is the magnitude of the filtered strain rate, \( C_s \) is the Smagorinsky coefficient, which can be determined by computations of isotropic turbulence decay (and is generally between 0.1 to 0.25). In this work, a value of 0.18 is used for \( C_s \). The trace of the SFS stress tensor, \( \sigma_{kk} \), can be modelled using the following expression proposed by Yoshizawa \[35\]:

\[
\sigma_{kk} = 2\tilde{\rho}C_1 \Delta^2 |\tilde{S}|^2,
\]

(2.20)

where \( C_1 \) is a closure coefficient that is generally 0.005.

Because of its simplicity, the Smagorinsky model is a widely used closure technique. However, it has its limitations. A prominent shortcoming comes from the fact that the Smagorinsky coefficient needs to be specified prior to the computation. Since \( C_s \) varies for different flow conditions, there is a disagreement in its values listed in the literature. To refine the subfilter-scale stress model where the Smagorinsky model is not valid, kinetic energy one-equation model is used.

A kinetic energy one-equation model was proposed by Yoshizawa et al. \[35\] and was further developed by Menon et al. \[36\]. In this model, the subfilter-scale kinetic energy is defined as

\[
\tilde{k} = \frac{1}{2\tilde{\rho}} \sigma_{kk} = \frac{1}{2} \left( \tilde{u}_k \tilde{u}_k - \tilde{\omega}_k \tilde{u}_k \right),
\]

(2.21)
and its transport equation takes the form

$$\frac{\partial \bar{\rho} \tilde{k}}{\partial t} + \frac{\partial \bar{\rho} \tilde{u}_i \tilde{k}}{\partial x_i} = \frac{\partial}{\partial x_i} \left( \bar{\rho} \left( \frac{\nu_t}{Pr_t} + \nu \right) \frac{\partial \tilde{k}}{\partial x_i} \right) + \sigma_{ij} \tilde{S}_{ij} - C_\epsilon \bar{\rho} \tilde{k}^{3/2} / \Delta, \tag{2.22}$$

where $Pr_t$ is the turbulent Prandtl number, defined as the ratio of momentum eddy-diffusivity to thermal eddy-diffusivity, and $C_\epsilon$ is a model constant. In this research-work, these constants are set as $C_\epsilon = 0.845$ and $Pr_t = 0.9$. Eq. (2.22) is solved for $\tilde{k}$, after which the subfilter stresses are found using

$$\sigma_{ij} = 2 \bar{\rho} \nu_t \left( \tilde{S}_{ij} - \frac{\delta_{ij}}{3} \tilde{S}_{kk} \right) - \frac{2}{3} \bar{\rho} \tilde{k} \delta_{ij}, \tag{2.23}$$

with SFS turbulent viscosity, $\nu_t$, defined as

$$\nu_t = C_\nu \Delta \sqrt{\tilde{k}}, \tag{2.24}$$

where $C_\nu$ is another model constant, assumed to be 0.086. The one-equation model provides more accurate results for many flows where Smagorinsky model fails. However, one more equation needs to be solved for the use of this model, and therefore, the implementation becomes computationally slightly more expensive.

### 2.3 Subfilter-Scale Scalar Transport Modelling

In terms III, IV and IX, the subfilter-scale enthalpy flux, $\bar{\rho} \left( \tilde{h}_s u_i - \tilde{h}_s \tilde{u}_i \right)$, and the subfilter-scale species flux, $\bar{\rho} \left( \tilde{Y}_n u_i - \tilde{Y}_n \tilde{u}_i \right)$, appear. These scalar transport terms are modelled using the standard gradient-based approximations given by

$$\bar{\rho} \left( \tilde{h}_s u_i - \tilde{h}_s \tilde{u}_i \right) = -C_p \frac{\bar{\rho} \nu_t}{Pr_t} \frac{\partial \tilde{T}}{\partial x_i}, \tag{2.25}$$

$$\bar{\rho} \left( \tilde{Y}_n u_i - \tilde{Y}_n \tilde{u}_i \right) = -\frac{\bar{\rho} \nu_t}{Sc_t} \frac{\partial \tilde{Y}_n}{\partial x_i}, \tag{2.26}$$

where $C_p$ is the specific heat capacity and $Sc_t$ corresponds to the SFS turbulent Schmidt number.

### 2.4 Subfilter-Scale Turbulent Transport Modelling

The SFS turbulent transport appears in term V, due to the filtering of convective terms in the energy equation. Knight et al. [37] proposed a model for the subfilter-scale turbulent transport term given by

$$(\bar{u}_i \tilde{u}_j \tilde{u}_j - \bar{u}_i \bar{u}_j \bar{u}_j) / 2 = \sigma_{ij} \bar{u}_i. \tag{2.27}$$

This is the model used for turbulent transport in this research.
Chapter 2. Governing Equations for LES of Premixed Combustion

2.5 Reaction Rate Modelling

The filtered reaction rate, $\bar{\omega}_n$, appearing in XI, is the term that adds major complications for performing LES of turbulent premixed combustion. Modelling this term is very challenging due to its non-linear dependence on temperature, pressure and species mass fraction. The filtered reaction rate cannot be expressed directly in terms of the resolved or filtered quantities, such that

$$\bar{\omega}_n \neq \hat{\omega}_n (\bar{T}, \bar{p}, \bar{Y}_n).$$

(2.28)

Moreover, in reacting flows, reactions occur in very thin flame-fronts, as mentioned in Section 1.1.2. The laminar flame thickness, $\delta_L$, which typically ranges from $(0.1 - 1.0)$ mm, is generally smaller than the LES filter size, $\Delta$. Evidently, these flames cannot be resolved by the LES mesh. Therefore, there are several approaches proposed for modelling flame-front propagation. Using a filtered progress variable with a reaction rate that depends on flame surface density (FSD) is one such approach.

The FSD model essentially neglects the combustion mechanisms in the flame-front, and the so-called reaction progress variable, $c$, is used to track the entire combustion process. The progress variable varies from 0 to 1 monotonically across the flame-front representing the range from unburnt reactants to burnt products. Since all the internal flame structures and the detailed chemical kinetics are not considered, this approach drastically reduces the computational cost. Compared to the non-reacting case of LES, only one additional transport equation needs to be solved – the transport equation for progress variable, $c$.

The progress variable defines the degree to which the mixture of the fuel has reacted, and can be expressed as reduced fuel mass fraction or reduced temperature given by

$$c = \frac{Y_F - Y_F^u}{Y_F^b - Y_F^u},$$

(2.29)

or

$$c = \frac{T - T^u}{T^b - T^u},$$

(2.30)

where $Y_F$, $Y_F^u$ and $Y_F^b$ are the local, unburnt and burnt fuel mass fractions, and $T$, $T^u$ and $T^b$ are the local, unburnt and burnt gas temperatures, respectively. At a critical value of the progress variable, $c^*$, the flame surface is defined.

The transport equation of the progress variable can be expressed in a general form as

$$\frac{\partial (\rho c)}{\partial t} + \frac{\partial (\rho u_i c)}{\partial x_i} = \bar{\omega} + \frac{\partial}{\partial x_i} \left( \rho D \frac{\partial c}{\partial x_i} \right) = \rho s_d |\nabla c|,$$

(2.31)
where \( s_d \) is the local propagation speed of an isosurface of a progress variable. Application of the Favre-filtering procedure to Eq. (2.31) results in the filtered form of transport equation of \( \tilde{c} \), as

\[
\frac{\partial (\bar{\rho} \tilde{c})}{\partial t} + \frac{\partial (\bar{\rho} \tilde{u}_i \tilde{c})}{\partial x_i} + \frac{\partial \left[ \bar{\rho} (\tilde{u}_i \tilde{c} - \tilde{u}_i \bar{c}) \right]}{\partial x_i} = \bar{\omega} + \frac{\partial}{\partial x_i} \left( \rho D \frac{\partial \bar{c}}{\partial x_i} \right) = \rho S_d |\nabla \bar{c}|. \tag{2.32}
\]

The SFS transport term is modelled using the gradient-based approximation, similar to the approach outlined in Section 2.3. Therefore,

\[
\frac{\partial}{\partial x_i} \left[ \bar{\rho} (\tilde{u}_i \tilde{c} - \tilde{u}_i \bar{c}) \right] = \frac{\partial}{\partial x_i} \left( \bar{\rho} \nu_t \frac{\partial \tilde{c}}{\partial x_i} \right). \tag{2.33}
\]

The unclosed terms in the right hand side of Eq. (2.32), \( \rho S_d |\nabla \bar{c}| \), requires modelling. For very thin flames, this can be modelled using the filtered flame surface density, \( \Sigma_{sfs} = \tilde{\rho} \tilde{\Sigma} \), which is essentially the flame surface area per unit volume contained within the LES filtering volume. Using the laminar flamelet assumption, these terms are modelled as

\[
\rho S_d |\nabla \bar{c}| = \tilde{\omega} + \frac{\partial}{\partial x_i} \left( \rho D \frac{\partial \bar{c}}{\partial x_i} \right) \approx \rho_u s_L \Sigma_{sfs}. \tag{2.34}
\]

where \( \rho_u \) is the density of unburnt reactants, and as specified earlier, \( s_L \) is the laminar flame speed. Using these approximations, the modelled transport equation for the filtered-progress variable can be expressed as

\[
\frac{\partial}{\partial t} \left( \bar{\rho} \tilde{c} \right) + \frac{\partial}{\partial x_i} \left( \bar{\rho} \tilde{u}_i \tilde{c} \right) = \frac{\partial}{\partial x_i} \left( \bar{\rho} \nu_t \frac{\partial \tilde{c}}{\partial x_i} \right) + \rho_u s_L \Sigma_{sfs}. \tag{2.35}
\]

The closure for this equation requires the determination of flame surface density, \( \Sigma_{sfs} \). There exists both simple algebraic equations and full transport equation closures for the SFS flame surface density. These closures are briefly discussed in the following sections. Specifically, the transport equation variant proposed by Hawkes and Cant [18, 38] and the algebraic variant developed by Boger et al. [16] are outlined. The reader is referred to the PhD theses of Hawkes [38] and Lin [15] for detailed discussions about these closure techniques.

### 2.5.1 Modelled Transport Equation for FSD

Hawkes and Cant [18] suggested an approach to model unclosed terms in the LES-FSD transport equation, based on a pre-existing concept of RANS modelling of FSD transport equation [39, 40]. Of course, in case of LES, they also included the terms that account for the resolved components of FSD production and destruction. The modelled flame surface density transport equation that they developed is summarized as

\[
\frac{\partial}{\partial t} \left( \bar{\rho} \hat{\Sigma} \right) + \frac{\partial}{\partial x_i} \left( \bar{\rho} \hat{\Sigma} \bar{u}_i \right) = \frac{\partial}{\partial x_i} \left( \bar{\rho} \nu_t \frac{\partial \hat{\Sigma}}{\partial x_i} \right) + \left( \delta_{ij} - n_{ij} \right) \frac{\partial \bar{u}_i}{\partial x_j} \rho \hat{\Sigma} - s_L (1 + \tau \hat{c}) N_i \frac{\partial}{\partial x_i} \left( \bar{\rho} \hat{\Sigma} \right) \\
- s_L \tau N_i \rho \hat{\Sigma} \frac{\partial \hat{c}}{\partial x_i} + \Gamma_k \left( \frac{\hat{k}}{\Delta} \rho \hat{\Sigma} \right)^2 s_L - \alpha \beta s_L \left( \frac{\rho \hat{\Sigma}}{1 - \hat{c}} \right)^2,
\]

(2.36)
where $\bar{c}$ can be defined as
\[
\bar{c} = \frac{(1 + \tau)\tilde{c}}{1 + \tau \tilde{c}},
\] (2.37)

$N_i$ is the i-th component of the local flame normal direction given by
\[
\vec{N} = -\frac{\nabla \tilde{c}}{|\nabla \tilde{c}|},
\] (2.38)

$\tau = (T^{ad} - T^u)/T^u$ is the heat release parameter depending on the adiabatic flame temperature $T^{ad}$, $\Gamma_k$ is a curve-fitted efficiency function suggested by Angelberger et al. [41], $\alpha$ is a resolution factor and $\beta \geq 1$ is a model constant. $n_{ij}$ acts as a orientation for the flame strain with respect to the flow gradients, and is given by
\[
n_{ij} = N_i N_j + \frac{1}{3} \alpha \delta_{ij}.
\] (2.39)

In this work, there are a few examples for second-order finite-volume scheme for LES of premixed flames that use this transport equation model for FSD. Application of high-order CENO schemes to this equation is out of the scope of this research. Here, the high-order scheme for LES is implemented only for cases where the algebraic variant of Boger et al. [16] is used to model FSD.

### 2.5.2 Algebraic Model of Boger for FSD

Compared to the transport equation model just outlined, algebraic models are simpler, easier to implement and computationally less expensive. The compromise is that they may not be as general or universally acceptable. These models are usually developed based on a corresponding RANS model derived earlier. Boger et al. [16] suggested a simple expression to model flame surface density, similar to Eddy-Break-Up (EBU) and Bray-Moss-Libby (BML) models used frequently for RANS simulation. The expression is given by
\[
\Sigma_{sfs} = \tilde{\rho} \tilde{\Sigma} = K_\Sigma \frac{\bar{c} (1 - \bar{c})}{\Delta},
\] (2.40)

where $K_\Sigma$ is a model constant. $K_\Sigma$ depends on the filter width, $\Delta$, and the local turbulence, but it does not depend on observation time. Based on the infinitely thin planar flame and a Gaussian LES filter, Boger et al. computed the value of this constant to be $K_\Sigma = 4\sqrt{6}/\pi$. This work makes use of this algebraic model for reaction rate modelling, while applying CENO scheme to the equations governing LES of reacting flows.
Chapter 3

Second-Order Finite-Volume Formulation

The Favre-filtered Navier-Stokes equations and the filtered transport equations described in Chapter 2 form a basis on which LES for turbulence premixed flames can be carried out. This study uses the finite-volume method (FVM) to solve these equations. Finite-volume formulations have gained in popularity because of their two general advantages [42]:

1. the FVM naturally ensures that the discretization is conservative. This means, using FV method, one can rest assured that the conserved properties, such as mass, momentum and energy, are conserved in a discrete sense; and

2. the FVM do not require coordinate transformation to be applied to the irregular meshes. This flexibility has enabled the use of unstructured grids, consisting of arbitrary polyhedra in three dimensions or arbitrary polygons in two dimensions.

The so-called ‘Godunov-type’ finite-volume method is employed for the computation of inviscid fluxes in this research. This method was first proposed by Godunov in 1959 [43], and has been subsequently modified by various researchers since [44, 45, 46, 47]. The approach is extended here for viscous flow applications using a standard central weighting technique. This method discretizes the solution domain to various control volumes, and applies the integral form of the conservation equations to these volumes. The following sections describe the use of the proposed finite-volume method for solving the equations governing LES of premixed flames described previously in Chapter 2. The focus here is on a finite-volume method that provides standard second-order accuracy for the smooth solutions. The extension of the finite-volume approach
to high-order accuracy, which is the primary focus of the thesis, is discussed in Chapter 4 to follow.

### 3.1 Governing Equations in Vector Form

The Favre-filtered Navier-Stokes equations outlined earlier can be expressed using matrix-vector notation as

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla : (\mathbf{F}^I - \mathbf{F}^V) = \mathbf{S},$$

(3.1)

where \( \mathbf{S} \) is the vector containing all the turbulence, chemistry and source terms, \( \mathbf{U} \) is the vector of conserved solution variables, \( \mathbf{F}^I = (\mathbf{F}^I, \mathbf{G}^I, \mathbf{H}^I) \) is the inviscid flux dyad, and \( \mathbf{F}^V = (\mathbf{F}^V, \mathbf{G}^V, \mathbf{H}^V) \) is the viscous flux dyad. Here, \( \mathbf{F}^I, \mathbf{F}^V, \mathbf{G}^I, \mathbf{G}^V, \mathbf{H}^I \) and \( \mathbf{H}^V \) represent the inviscid and viscous flux-vectors in x-, y- and z-directions, respectively.

In the following lines, the components of each of these vectors are outlined for LES of reactive flows. Since this work makes use of Boger’s algebraic FSD model, the transport equation for FSD is not included in the representation. Following the expressions for the Favre-filtered Navier-Stokes equations and the filtered transport equations for \( \tilde{c} \) and \( \tilde{k} \), the following expressions are obtained for \( \mathbf{U}, \mathbf{F}^I, \mathbf{F}^V, \mathbf{G}^I, \mathbf{G}^V, \mathbf{H}^I, \mathbf{H}^V \) and \( \mathbf{S} \):

$$\mathbf{U} = \begin{pmatrix} \bar{\rho} \\ \bar{\rho} \bar{u} \\ \bar{\rho} \bar{v} \\ \bar{\rho} \bar{w} \\ \bar{\rho} \bar{E} \\ \bar{\rho} \tilde{c} \\ \bar{\rho} \tilde{k} \end{pmatrix},$$

(3.2)

$$\mathbf{F}^I = \begin{pmatrix} \bar{\rho} \bar{u} \\ \bar{\rho} \bar{u} \bar{u} + \bar{p} \\ \bar{\rho} \bar{u} \bar{v} \\ \bar{\rho} \bar{u} \bar{w} \\ \bar{u}(\bar{\rho} \bar{E} + \bar{p}) \\ \bar{\rho} \bar{u} \bar{c} \\ \bar{\rho} \bar{u} \tilde{k} \end{pmatrix}, \quad \mathbf{G}^I = \begin{pmatrix} \bar{\rho} \bar{v} \\ \bar{\rho} \bar{v} \bar{u} + \bar{p} \\ \bar{\rho} \bar{v} \bar{v} \\ \bar{\rho} \bar{v} \bar{w} \\ \bar{v}(\bar{\rho} \bar{E} + \bar{p}) \\ \bar{\rho} \bar{v} \bar{c} \\ \bar{\rho} \bar{v} \tilde{k} \end{pmatrix}, \quad \mathbf{H}^I = \begin{pmatrix} \bar{\rho} \bar{w} \\ \bar{\rho} \bar{w} \bar{u} + \bar{p} \\ \bar{\rho} \bar{w} \bar{v} \\ \bar{\rho} \bar{w} \bar{w} \\ \bar{w}(\bar{\rho} \bar{E} + \bar{p}) \\ \bar{\rho} \bar{w} \bar{c} \\ \bar{\rho} \bar{w} \tilde{k} \end{pmatrix},$$

(3.3)
Chapter 3. Second-Order Finite-Volume Formulation

\[ \mathbf{F}^V = \begin{pmatrix} 0 \\ \tilde{\tau}_{xx} + \sigma_{xx} \\ \tilde{\tau}_{xy} + \sigma_{xy} \\ \tilde{\tau}_{xz} + \sigma_{xz} \\ \tilde{u} (\tilde{\tau}_{xx} + \sigma_{xx}) + \tilde{v} (\tilde{\tau}_{xy} + \sigma_{xy}) + \tilde{w} (\tilde{\tau}_{xz} + \sigma_{xz}) - (\tilde{q}_x + \theta_x) + \tilde{\rho} \left( \frac{\nu}{Pr} + \nu \right) \frac{\partial k}{\partial x} / Sc_t \end{pmatrix}, \quad (3.4) \]

\[ \mathbf{G}^V = \begin{pmatrix} 0 \\ \tilde{\tau}_{xy} + \sigma_{xy} \\ \tilde{\tau}_{yy} + \sigma_{yy} \\ \tilde{\tau}_{yz} + \sigma_{yz} \\ \tilde{u} (\tilde{\tau}_{xy} + \sigma_{xy}) + \tilde{v} (\tilde{\tau}_{yy} + \sigma_{yy}) + \tilde{w} (\tilde{\tau}_{yz} + \sigma_{yz}) - (\tilde{q}_y + \theta_y) + \tilde{\rho} \left( \frac{\nu}{Pr} + \nu \right) \frac{\partial k}{\partial y} / Sc_t \end{pmatrix}, \quad (3.5) \]

\[ \mathbf{H}^V = \begin{pmatrix} 0 \\ \tilde{\tau}_{xz} + \sigma_{xz} \\ \tilde{\tau}_{yz} + \sigma_{yz} \\ \tilde{\tau}_{zz} + \sigma_{zz} \\ \tilde{u} (\tilde{\tau}_{xz} + \sigma_{xz}) + \tilde{v} (\tilde{\tau}_{yz} + \sigma_{yz}) + \tilde{w} (\tilde{\tau}_{zz} + \sigma_{zz}) - (\tilde{q}_z + \theta_z) + \tilde{\rho} \left( \frac{\nu}{Pr} + \nu \right) \frac{\partial k}{\partial z} / Sc_t \end{pmatrix}, \quad (3.6) \]

and

\[ \mathbf{S} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \rho u \Sigma_{sf} - \frac{C_p k^{3/2}}{\Delta} \end{pmatrix}, \quad (3.7) \]

3.2 Godunov-type Finite-Volume Method

For purely hyperbolic conservation laws, Godunov-type finite-volume method employs the integral form of conservation equations. Integration of Eq. (3.1) over an arbitrary control volume
Chapter 3. Second-Order Finite-Volume Formulation

V that is bounded by a surface \( A \) is given by

\[
\int \int \int_{V} \left[ \frac{\partial U}{\partial t} + \nabla \cdot \left( \vec{F}^I - \vec{F}^V \right) \right] \, dV = \int \int \int_{V} S \, dV.
\]  

(3.8)

Using divergence theorem, and recognizing the fact that \( t \) is an independent variable, Eq. (3.8) can be represented as

\[
\frac{d}{dt} \int \int \int_{V} U \, dV + \oint_{A} \left( \vec{F}^I - \vec{F}^V \right) \cdot \hat{n} \, dA = \int \int \int_{V} S \, dV,
\]  

(3.9)

where \( \hat{n} \) is the unit vector normal to surface \( A \). Using the cell-averaged quantities, which are defined as

\[
\bar{U} = \frac{1}{V} \int \int \int_{V} U \, dV, \text{ and }
\]  

(3.10)

\[
\bar{S} = \frac{1}{V} \int \int \int_{V} S \, dV,
\]  

(3.11)

Eq. (3.9) can be expressed as

\[
\frac{d\bar{U}}{dt} = -\frac{1}{V} \oint_{A} \left( \vec{F}^I - \vec{F}^V \right) \cdot \hat{n} \, dA + \bar{S}.
\]  

(3.12)

The right hand side (RHS) of Eq. (3.12) represents the sum of net transport by the mean flow (the net flux) of the conserved variables and the effects due to chemistry and turbulence. Therefore, this equation simply suggests that the time rate of change of \( \bar{U} \) is dependent on the net flux through each control volume and the turbulent-chemistry effects inside the control volume. Application of a spatial discretization procedure to RHS of Eq. (3.12) gives a semi-discrete form of the conservation equations. Using a three-dimensional (3D) Cartesian control volume in a structured, multi-block mesh, and using a \( N_G \)-point Gaussian quadrature numerical integration procedure to evaluate the solution flux along each of the \( N_f \) faces, the following semi-discrete form of the governing equation is obtained:

\[
\frac{dU_{ijk}}{dt} = -\frac{1}{V_{ijk}} \sum_{l=1}^{N_f} \sum_{m=1}^{N_G} \left( \omega \left( \vec{F}^I_{ijkl,m} - \vec{F}^V_{ijkl,m} \right) \cdot \hat{n}_A \right)_{ijkl,m} + \bar{S}_{ijk} = \bar{R}_{ijk} \left( \bar{U} \right),
\]  

(3.13)

where \( \omega_{lm} \) are the quadrature weighting coefficients, \( A_l \) denotes the surface area of face \( l \), and \( \bar{R}_{ijk} \) is the residual operator. After the evaluation of \( \bar{R} \), one can advance the solution in time, and therefore iteratively solve the time dependent problem that is described by the equations.

After obtaining the semi-discrete form of governing equations, the Godunov-type FV scheme employs four further steps:
1. **Solution Reconstruction**: Given the cell-averaged values of the solution, there is a need to provide spatial variation of the solution in each cell, using piecewise polynomials. Essentially, the order of this piecewise solution reconstruction accounts for the spatial order of accuracy of the scheme. Solution reconstruction is discussed in Section 3.4.

2. **Flux Evaluation**: Using the reconstructed polynomials in each cell and an appropriate numerical flux evaluation procedure, fluxes are evaluated at each quadrature points at the cell interfaces. Inviscid and viscous fluxes are evaluated differently, and are discussed in Sections 3.3.1 and 3.3.2, respectively.

3. **Source Vector Evaluation**: One of the important features in the semi-discrete equations governing LES of reacting flows, Eq. (3.13), is the existence of the cell-averaged source vector, $\mathbf{S}$. The components of this vector are the terms that incorporate effects of turbulence and chemistry in the reacting flows. As such, these terms also need to be evaluated alongside the viscous and inviscid fluxes. For a second-order scheme, these terms are evaluated simply using the cell-averaged quantities. The high-order evaluation of these terms are discussed in Chapter 4.

4. **Time marching**: Appropriate time marching scheme is used to update the cell-averaged values to the next time step, after the evaluation of $\mathbf{R}$. Time marching methods are briefly discussed in Section 3.5.

Details of each of these steps is now outlined for the standard second-order accurate approach. For the purpose of maintaining clarity in explanation, inviscid flux evaluation is discussed prior to solution reconstruction in the following sections.

## 3.3 Flux Evaluation

### 3.3.1 Inviscid (Hyperbolic) Flux Evaluation

In a Godunov-type finite-volume method, the inviscid (hyperbolic) fluxes, $\mathbf{F}^I$, are evaluated using solutions of locally one-dimensional Riemann problems. Riemann problems are initial value problems, first introduced by Godunov [43, 48] in his finite-difference schemes developed to solve unsteady inviscid flows. A Riemann problem in one-dimension can be expressed by the following initial data:

$$U_o(x) = U(x, t = 0) = \begin{cases} U_{\text{left}}, & \text{if } x \leq 0 \\ U_{\text{right}}, & \text{if } x > 0 \end{cases} \quad \text{for } -\infty \leq x \leq \infty, \quad (3.14)$$
where $U_{\text{left}}$ and $U_{\text{right}}$ are the left and the right solution states at the cell interface. Adapting this idea to a Cartesian three-dimensional mesh, where 'cell interfaces' are the six faces of the cell, there exists a Riemann problem in each Gauss quadrature point of each cell faces. For instance, cell $i$ (left-state) and cell $i + 1$ (right-state) share an interface at $i + \frac{1}{2}$, as depicted in a one-dimensional grid in Figure 3.1. At a quadrature point in this interface, there exist solutions reconstructed from the averaged values in cell $i$, and those reconstructed from cell $i + 1$. Clearly, these reconstructed solutions would not always be the same, and therefore, discontinuities occur. At each quadrature point in the cell interface, inviscid flux is a function of both the left and the right solution states, given respectively by the solution reconstruction in these cells. This can be summarized as

$$\vec{F}_I \cdot \hat{n} = \vec{F}_I (U_{\text{left}}, U_{\text{right}}, \hat{n}).$$

The introduction of Riemann problem has inspired researchers to assess and develop various Riemann solvers and flux functions. Gottlieb and Groth [49] have developed an exact Riemann solver which finds the inviscid fluxes using iterative techniques. This solver can be used efficiently for solving Euler equations governing polytropic gases. Several approximate Riemann solvers have also been developed such that the computational cost associated with the iterative procedure is eliminated. Since conservative solutions are updated in an approximate manner via time marching procedure, solving Riemann problems exactly might not be always necessary. Some popular approximate Riemann solvers, such as the one developed by Roe [50] and that by Harten-Lax-van Leer-Einfeld (HLLE) [51, 45], are available for use within the computational framework of the research group that the author is associated with. This work, however, makes use of an approximate Riemann solver called AUSM$^+$-up. This solver is briefly described in the following section.

**AUSM$^+$-up**

In 2006, Liou [52] developed the latest version of the advection upstream splitting method (AUSM) and named it AUSM$^+$-up. AUSM schemes are generally valid for all speed regimes, and are therefore independent of Mach-number. Contrary to the general AUSM scheme, in AUSM$^+$-up, inviscid fluxes are splitted into convective and pressure contribution. For instance,
the inviscid flux in the x-direction is splitted as follows

\[ \mathbf{F}^l = \mathbf{F}_c + \mathbf{P} = \dot{m}\Phi + \mathbf{P} = \hat{\rho}\hat{u} \begin{pmatrix} 1 \\ \hat{u} \\ \hat{v} \\ \hat{w} \\ \hat{E} \\ \hat{c} \\ \hat{k} \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \]  

(3.16)

where the mass flux \( \dot{m} \) is given by \( \dot{m} = \hat{\rho}\hat{u} \), and the vector \( \Phi \) is given by \( \Phi = (\hat{u}, \hat{v}, \hat{w}, \hat{E}, \hat{c}, \hat{k})^T \).

The numeric flux at the cell interface, \( \mathbf{F}_{1/2} \), is then expressed as

\[ \mathbf{F}_{1/2} = \dot{m}_{1/2}\Phi_{\text{left}/\text{right}} + \mathbf{P}_{1/2}, \]  

(3.17)

where \( \Phi_{\text{left}/\text{right}} \) is the left or the right vector that is determined with the upwind scheme as

\[ \Phi_{\text{left}/\text{right}} = \begin{cases} \Phi_{\text{left}}, & \text{if } \dot{m}_{1/2} > 0, \\ \Phi_{\text{right}}, & \text{if } \dot{m}_{1/2} \leq 0. \end{cases} \]  

(3.18)

From these equations, it can be seen that computation of the mass fluxes, \( \dot{m} \), at each cell interface is crucial for the AUSM\textsuperscript{+}-up scheme. For brevity, this computation is not presented in this work. The reader is advised to refer to the work of Liou [52] for further details. At this point, it is worthwhile to note that AUSM\textsuperscript{+}-up scheme is shown to be robust for all Mach number regimes and valid for low-speed flows. Therefore, it is an appropriate flux function to compute inviscid fluxes while performing LES of reacting flows.

### 3.3.2 Viscous (Elliptic) Flux Evaluation

The viscous fluxes represent the diffusion processes in governing equations. As such, they display elliptic nature. Their evaluation at each quadrature point of the respective cell interface depends on both the solution quantities and the gradients of the solution. For a cell interface located at \( (i + \frac{1}{2}, j, k) \), the viscous flux at a particular quadrature point can be expressed as

\[ \vec{F}^V_{i+\frac{1}{2},j,k} \cdot \hat{n} = \vec{F}^V \left( \mathbf{U}_{i+\frac{1}{2},j,k}, \vec{V} \mathbf{U}_{i+\frac{1}{2},j,k} \right). \]  

(3.19)

The solution vector at a quadrature point of the cell interface, \( \mathbf{U}_{i+\frac{1}{2},j,k} \), is evaluated averaging the left and the right reconstructed solutions, given by

\[ \mathbf{U}_{i+\frac{1}{2},j,k} = \frac{\mathbf{U}_{\text{left}} + \mathbf{U}_{\text{right}}}{2}. \]  

(3.20)
On the other hand, the solution gradient, $\nabla U_{i+\frac{1}{2},j,k}$, is evaluated differently according to the order of the scheme used. For a second-order scheme, in this study, the so called hybrid average gradient-diamond-path reconstruction is used for the gradient evaluation. This scheme is proposed by Mathur and Murthy [53], which suggests that the solution gradients can be computed as follows:

$$\hat{\nabla} U_{i+\frac{1}{2},j,k} = \frac{U_{i+1,j,k} - U_{i,j,k}}{\Delta} \frac{\hat{n} \cdot \hat{e}_s}{\hat{n} \cdot \dot{e}_s} + \left( \nabla U - \nabla U \cdot \hat{e}_s \frac{\hat{n}}{\hat{n} \cdot \dot{e}_s} \right),$$

where,

$$\nabla U = \vartheta \nabla U_{i,j,k} + (1 - \vartheta) \nabla U_{i+1,j,k},$$

the weighting factor $\vartheta$ is dependent on cell volume ratios given by

$$\vartheta = \frac{V_{i,j,k}}{V_{i,j,k} + V_{i+1,j,k}},$$

$\hat{n}$ is the normal vector at the cell interface, $\Delta$ is the distance between the cell centers of the adjacent cells, and $\dot{e}_s$ is a unit vector along this direction.

For a high-order method, the solution gradients are evaluated in a different manner. This approach is discussed in Chapter 4.

### 3.4 Solution Reconstruction

It can be noted from Eq. (3.12) and Eq. (3.13) that the Godunov-type finite-volume method makes use of and updates the average solution quantities, $\bar{U}$. These average solution quantities are provided as initial conditions, and are updated via time marching integration. However, for flux evaluation, one requires the left ($U_{\text{left}}$) and the right ($U_{\text{right}}$) states of the solution at each Gauss quadrature point in the cell interfaces. To determine these solution states at the cell interfaces, solution reconstruction is necessary. Essentially, given the average solution quantities, $\bar{U}$, in each cell, reconstruction helps to determine the spatial distribution of $U(x, y, z)$ in the cell using polynomial approximations. The order of the reconstruction polynomial dictates the spatial order of accuracy of the method. It has been shown that a $k^{th}$-order polynomial reconstruction results in $(k + 1)$-order spatial accuracy for hyperbolic systems and $k$-order accuracy for elliptic systems [23, 54, 24, 25].

The following sections will explain the piecewise constant and piecewise linear reconstructions. Note that the hyperbolic Euler equations in one-dimension are used in these sections to explain various facets of these reconstruction. The semi-discrete form of Euler equations can be easily obtained from Eq. (3.13) by eliminating the viscous and the source terms, and can be expressed
as
\[
\frac{d\mathbf{U}_{ijk}}{dt} = -\frac{1}{V_{ijk}} \sum_{l=1}^{N_f} \sum_{m=1}^{N_G} \left( \omega \hat{F} \cdot \hat{n} A \right)_{ijk,l,m} = \mathbf{R}_{ijk}(\mathbf{U}).
\] (3.24)

In one-dimension, these equations can be written as
\[
\frac{dU_i}{dt} = -\frac{1}{\Delta x} \left( F^I_{i+\frac{1}{2}} + \frac{1}{2} - F^I_{i-\frac{1}{2}} \right),
\] (3.25)
where \( F^I_{i+\frac{1}{2}} \) and \( F^I_{i-\frac{1}{2}} \) are the inviscid fluxes at left and right interfaces of cell \( i \), respectively. These fluxes need to be evaluated based on solution reconstruction and flux functions.

### 3.4.1 Godunov’s Method and Piecewise Constant Reconstruction

Godunov [43] used a first-order upwind scheme to obtain monotonic solutions, which are free of oscillations near shocks and discontinuities. A first-order FV scheme for hyperbolic equations can be achieved by a so called ‘piecewise constant reconstruction’. Piecewise constant reconstruction is the zeroth-order \((k = 0)\) reconstruction, where the solution distribution in a cell is a constant value corresponding to the cell-averaged solution value. This reconstruction is depicted in Figure 3.1. With piecewise constant reconstruction, the right hand side of Eq. (3.25) can easily be determined using a Riemann solver with the following information:
\[
F^I_{i+\frac{1}{2}} = F^I(\mathbf{U}_{\text{left}} = \bar{\mathbf{U}}_i, \mathbf{U}_{\text{right}} = \bar{\mathbf{U}}_{i+1}),
\] (3.26)
\[
F^I_{i-\frac{1}{2}} = F^I(\mathbf{U}_{\text{left}} = \bar{\mathbf{U}}_{i-1}, \mathbf{U}_{\text{right}} = \bar{\mathbf{U}}_i).
\] (3.27)

Thereafter, the use of first-order explicit Euler time marching scheme completes the Godunov’s method. Evidently, this method is first-order in both space and time. A second-order spatial accuracy can be obtained by using piecewise linear reconstruction.

### 3.4.2 Piecewise Linear Reconstruction

Building up on the idea of first-order Godunov’s scheme, it is clear that a second-order spatial accuracy for hyperbolic systems can be achieved by increasing the order of solution reconstruction to piecewise linear \((k = 1)\). This reconstruction scheme assumes that the solution in each cell varies linearly with position, \( \bar{x} = (x, y, z) \), and can be expressed as
\[
\mathbf{U}(\bar{x}) = \mathbf{U}_{ijk} + \left( \nabla \mathbf{U}_{ijk} \right) \cdot (\bar{x} - \bar{x}_{ijk}),
\] (3.28)
where $\vec{x}_{ijk}$ is the location of centroid of the cell indicated by $ijk$. In one-dimension, this equation is simply

$$U(x) = U_i + \left. \frac{\partial U}{\partial x} \right|_{i+\frac{1}{2}} (x - x_i).$$

(3.29)

These equations cannot be solved yet, as the solution gradients $\vec{\nabla}U_{ijk}$ are still unknown. To determine these values, the cell-averaged quantities of the neighbouring cells can be utilized. A method called least-squares reconstruction can be employed to find these gradients in a cell with the help of its neighbouring cells. For the purpose of explaining linear least-squares reconstruction here, only one-dimensional case is considered. A general $k$-exact least-squares reconstruction is discussed in further detail in Section 4.2.

The method of least-squares reconstruction utilizes an idea that reconstructed polynomial in cell $i$ should recover cell-averaged values of the neighboring cells ($i + 1$ and $i - 1$) if the polynomial is integrated over the domain of these neighboring cells. Therefore,

$$U_{i-1} = \frac{1}{\Delta x} \int_{x_{i-\frac{1}{2}}}^{x_{i-\frac{1}{2}}} U(x) dx = \frac{1}{\Delta x} \int_{x_{i-\frac{1}{2}}}^{x_{i-\frac{1}{2}}} \left[ U_i + \left. \frac{\partial U}{\partial x} \right|_{i+\frac{1}{2}} (x - x_i) \right] dx,$$

(3.30)

$$U_{i+1} = \frac{1}{\Delta x} \int_{x_{i+\frac{1}{2}}}^{x_{i+\frac{1}{2}}} U(x) dx = \frac{1}{\Delta x} \int_{x_{i+\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \left[ U_i + \left. \frac{\partial U}{\partial x} \right|_{i+\frac{1}{2}} (x - x_i) \right] dx.$$

(3.31)

Note that Eq. (3.30) and Eq. (3.31) are two linear equations with one unknown – the slope of the linear representation, $\left. \frac{\partial U}{\partial x} \right|_{i+\frac{1}{2}}$. This overdetermined system of equations can be solved using a least-squares method, in which the goal is primarily to minimize the sum of the squares of errors. In this case, errors are the difference between the actual cell-averaged values of the neighbours.
and its average values obtained from approximated polynomial in cell \( i \). In the one-dimensinal case, least-squares problem can be solved easily to obtain the following approximation for the gradient at cell \( i \):

\[
\frac{\partial U}{\partial x} \bigg|_i = \frac{U_{i+1} - U_{i-1}}{2\Delta x}.
\]

(3.32)

It can be noted that this expression is an arithmetic mean of the forward- and the backward-difference approximation schemes used in CFD to compute derivatives. In finite-difference method, this scheme is known as the center-difference approximation. Figure 3.1 depicts a piecewise linear reconstruction in cell \( i \) based on the cell-averaged values in cell \((i - 1)\) and cell \((i + 1)\). Notice that the higher cell-averaged solution in cell \((i + 1)\) affects the slope of the linear reconstruction in cell \( i \). The reconstruction slope is positive when the line moves from the centroid of cell \( i \) towards the cell \((i + 1)\) and it is negative when the line moves from the centroid towards the cell \((i - 1)\).

The second-order Godunov-type FV scheme violates Godunov’s theorem that states that within a class of constant coefficient explicit schemes for solution of linear advection equation, there are no scheme that are both at least second-order accurate and monotonicity preserving [43]. As a result, second-order spatially accurate scheme leads to oscillatory and non-monotonic solutions near shocks and discontinuities, as depicted in Figure 3.2. These unwanted oscillations (often known as Gibbs phenomenon [55]) might be detrimental for solution generation. Indeed, the first-order method is more accurate monotone-preserving minimum-dissipation scheme. Therefore, to enforce solution monotonicity, second-order scheme could only be used in smooth regions. In discontinuous regions near shocks, the scheme could be reverted down to first-order
spatial accuracy. This can be achieved by the usage of ‘MUSCL-type slope limiters’.

Enforcing Solution Monotonicity: Slope Limiters

A second-order Godunov-type finite-volume method typically employs a non-linear method, such as flux or slope limiting, to control the spurious oscillations occurring at the vicinity of shocks. The limiters are non-linear as they themselves depend on the solutions that they are limiting. In this research, the Monotonic Upstream Schemes for Conservation Laws of van Leer [56] (MUSCL)-type slope limiters are used.

A \textit{slope limiter}, $\psi_{ijk}$, which varies from zero to one, is added to Eq. (3.28), to explicitly modify solution gradients during linear reconstruction, as

$$
\mathbf{U}(\vec{x}) = \mathbf{U}_{ijk} + \psi_{ijk} \left( \nabla \mathbf{U}_{ijk} \right) \cdot (\vec{x} - \vec{x}_{ijk}),
$$

which is expressed in one dimension as

$$
\mathbf{U}(x) = \mathbf{U}_i + \psi_i \frac{\partial \mathbf{U}}{\partial x} \bigg|_{x=x_i} (x - x_i).
$$

It can be easily noted that $\psi_{i,j,k} = 0$ corresponds to a piecewise constant reconstruction, and $\psi_{i,j,k} = 1$ corresponds to an unlimited piecewise linear reconstruction. Essentially, the value of the limiter is 1 in the smooth regions, and is 0 in the vicinity of shocks or discontinuities. It should be noted that although it is possible to reconstruct and limit the conservative variables, primitive variables are used for these purposes in this work. Use of primitive variables for reconstruction and limitation helps in maintaining a better control on the positivity of certain flow variables, such as density and pressure.

There exists many formulations for the slope limiters. An extensive analysis about these limiters is made in a review by Waterson and Deconick [57]. The slope limiters developed by Barth and Jespersen [58] and developed by Venkatakrishnan [59] are extensively used in the computational framework in the research group that the author is associated with. This work mostly makes use of the Barth-Jespersen limiter, and therefore, it is briefly discussed in the following paragraphs.

For a given cell, $i$, Barth-Jespersen limiter is given by

$$
\psi_i = \min(\psi_{i,k}),
$$

where $k$ index refers to the quadrature points at which the flux is evaluated. In one-dimensional problem, each cell has simply two quadrature points – one at each cell boundary. For multi-dimensional case, the monotonicity is checked at each Gauss quadrature points where the
numerical flux is computed. Barth-Jespersen limiter adds only a minimal dissipation to the scheme by limiting only when required. It follows a discontinuous path (different compared to the continuous path of Venkatakrishnan limiter) from 0 to 1, and can be described as follows:

$$\psi_{i,k} = \psi_{i,k}(r_k) = \begin{cases} 1 & \text{for } (u_k - \bar{u}) = 0, \\ \min(1,r_k) & \text{for } (u_k - \bar{u}) \neq 0, \end{cases}$$ (3.36)

where $u$ represents the primitive solution variable of interest in cell $i$, $\bar{u}$ is the cell-averaged solution quantity, $u_k$ is the unlimited reconstructed value at the $k$-th quadrature point, and $r_k$ is defined as

$$r_k = \begin{cases} \frac{u_{\max} - \bar{u}}{u_k - \bar{u}} & \text{for } (u_k - \bar{u}) > 0, \\ \frac{u_{\min} - \bar{u}}{u_k - \bar{u}} & \text{for } (u_k - \bar{u}) < 0, \end{cases}$$ (3.37)

where, $u_{\max}$ and $u_{\min}$ are respectively the maximum and the minimum cell average quantities amongst all the cells used in reconstruction of cell $i$, given by

$$u_{\max} = \max(\bar{u}, \bar{u}_{\text{neighbours}}),$$
$$u_{\min} = \min(\bar{u}, \bar{u}_{\text{neighbours}}).$$

This limited piecewise linear reconstruction can be used to determine the left and the right states at the cell interfaces. For instance, at the cell interface $i + \frac{1}{2}$, the left and right states are

$$U_{i + \frac{1}{2}, \text{left}} = U_i + \psi_i \frac{\partial U}{\partial x} \bigg|_{i} (x_{i + \frac{1}{2}} - x_i),$$ (3.38)
$$U_{i + \frac{1}{2}, \text{right}} = U_{i+1} + \psi_i \frac{\partial U}{\partial x} \bigg|_{i+1} (x_{i + \frac{1}{2}} - x_{i+1}).$$ (3.39)

These states can be supplied to the flux functions to obtain the second-order accurate inviscid fluxes. A second-order time marching method, such as second-order Runge-Kutta scheme (RK2) [42] could then be used in Eq. (3.25) to obtain the solutions. Time marching schemes are briefly discussed in Section 3.5.

### 3.5 Time Marching Schemes

In any initial value problem, time marching methods are crucial to recognize the evolution of the solution quantities from their initial states. Godunov-type finite-volume methods may make use of either explicit or implicit time marching methods to solve the set of non-linear ordinary differential equations (ODEs) given by Eq. (3.13). In this work, explicit time marching schemes have been employed. In general, high-order CENO scheme is not restricted to explicit time
marching methods. But additional considerations need to be addressed to incorporate implicit
time marching scheme with CENO [25, 27]. These considerations are not within the scope of
this research.

In this work, to maintain consistency with regards to accuracy, the order of accuracy of time
marching schemes are kept the same as their spatial-accuracy, wherever possible. For a fourth-
order spatially accurate CENO scheme, a fourth-order Runge-Kutta (RK4) method is used.
A third-order CENO scheme also uses RK4 time marching, whereas a second-order CENO
utilizes second-order accurate RK method. Finally, the first-order scheme uses Explicit Euler
time marching scheme, which is first-order accurate. All these schemes are well-known, and can
be found in many CFD textbooks including the one by Lomax, Pulliam and Zingg [42] or that
by Hirsch [60, 61].

It should be noted that all these explicit time marching methods are conditionally stable.
The time-step, in case of reacting flows, is restricted by the effects of Courant-Friedrichs-Lewy
(CFL) stability, von Neumann viscous stability, and constraints arising due to the turbulent
and chemical time-steps. These turbulent and chemical time step constraints are estimated
using the inverse quantity of the maximum Jacobian of chemical source term, \( \frac{\partial S}{\partial U} \). Taking these
conditions into consideration, a stable global time step is given by

\[
\Delta t \leq \min \left( \frac{CFL}{\Delta l |\vec{u}| + a}, \alpha \frac{\rho \Delta l^2}{\max(\nu, \nu_t)}, \beta \max \left( \frac{\partial S}{\partial U} \right)^{-1} \right), \tag{3.40}
\]

where \( \Delta l \) is the length of the face of a cell, \( a \) is the local sound speed, \( \nu \) represents the kinematic
viscosity and \( \nu_t \) denotes the turbulent eddy viscosity. The scaling factors such as CFL, \( \alpha \) and
\( \beta \) are used to allow better control on the constraints.

### 3.6 Numerical Example – 1D Shock-tube Problem

This section provides an example to demonstrate the first- and the second-order finite-volume
methods discussed in this chapter. For this purpose, the one-dimensional Euler equations, given
in Eq. (3.25), are solved for a shock-tube problem. The corresponding results are displayed in
Figure 3.6.

Figure 3.6 highlights the advantages of using a second-order scheme compared to the first-order
method. This figure is generated using a one-dimensional Euler solver developed by the author,
which solves Eq. (3.25) for an initial value problem using one of the various flux functions
(Roe, HLLL, HLLE) with either first- or second-order accuracy. Figure 3.6 depicts the density
distribution solution of an initial value shock-tube problem. The initial states are adapted
Figure 3.3: First- and second-order solutions (density distributions) of the shock-tube problem at $t = 7$ ms.

from [62], given by

$$W(x, 0) = \begin{pmatrix} \rho(x, 0) \\ u(x, 0) \\ p(x, 0) \end{pmatrix} = \begin{cases} W_{\text{left}}, & x < 5 \text{ m}, \\ W_{\text{right}}, & x \geq 5 \text{ m}, \end{cases} \quad (3.41)$$

where the initial left states are $\rho_{\text{left}} = 4.696 \text{ kg/m}^3$, $u_{\text{left}} = 0 \text{ m/s}$, $p_{\text{left}} = 404.4 \text{ kPa}$, and the right states are $\rho_{\text{right}} = 1.408 \text{ kg/m}^3$, $u_{\text{right}} = 0 \text{ m/s}$, and $p_{\text{right}} = 101.1 \text{ kPa}$. This unsteady state problem is evaluated until time $t = 7$ ms in air with a grid with 100 cells using a CFL number of 0.5. HLLE flux function is used to evaluate the inviscid fluxes and Barth-Jespersen limiter is used to limit the slopes used in piecewise linear reconstruction. From this figure, one can notice that the monotonicity-preserving first-order method is not as accurate as the second-order method. Moreover, an unlimited second-order scheme exhibits oscillations near discontinuities. Clearly, the limited second-order scheme generates solutions that are more dissipative compared to the first-order results, and less dispersive compared to the solutions generated using unlimited second-order scheme.
Chapter 4

High-Order CENO Schemes

In this thesis, high-order methods refer to the schemes with order of accuracy greater than two. Development of high-order methods for CFD is currently an active field of research. These methods potentially generate better accurate results with comparatively lower computational cost. In other words, compared to the lower-order methods, high-order schemes make use of reduced computational mesh sizes to generate solutions with a desired level of accuracy. This improved computational efficiency might prove immensely beneficial for numerical treatment of complex flows such as combustion, aeroacoustic modelling and computational electromagnetics.

The central essentially non-oscillatory scheme, popularly known as CENO scheme, is a high-order finite-volume method which is robust enough to capture discontinuities as well as avoid the clipping of smooth solution extremas. The difference between CENO and other essentially non-oscillatory (ENO) schemes is presented in Section 1.1.3. CENO is developed by Ivan and Groth [23, 24, 25] utilizing $k$-order polynomial reconstruction within each computational cells. Ivan and Groth applied CENO scheme to generate high-order solutions of two-dimensional Euler and Advection-Diffusion equations on structured grids. Later, McDonald [63, 26] extended the application of this scheme to two-dimensional unstructured grids. Rashad [27] extended CENO to three-dimensional Cartesian grids, and used it to solve Euler and Navier-Stokes equations. This work will apply the three-dimensional CENO to solve equations governing LES of turbulent premixed flames on Cartesian grids.

The purpose of this chapter is to describe the extension of the second-order accurate finite-volume scheme for solving the filtered Navier-Stokes equations described in Chapter 3 to high-order accuracy via application of the CENO approach. This extension can be then be used to perform high-order LES of turbulent premixed flames.
4.1 Overview of the CENO Scheme

CENO scheme is based on a hybrid solution reconstruction procedure. A high-order $k$-exact reconstruction is applied only on cells with fully-resolved (smooth) solution content. In other cells, where solutions are under-resolved (discontinuous), the scheme reverts down to monotonicity preserving limited piecewise linear reconstruction. In this way, CENO reconstruction is able to capture smooth solution extremas as well as shocks and discontinuities. Moreover, since a fixed central stencil is used, a more representative reconstruction of the solution content is provided. This is because information of neighbouring cells from all sides of the cell of interest are utilized for reconstruction. The use of a fixed central stencil in CENO also eliminates computational cost associated with reconstruction using multiple stencils (ENO and WENO schemes use multiple stencils). To state it generally, CENO is a robust high-order scheme that takes a relatively simple approach and is flexible enough to be used on both structured and unstructured meshes.

CENO reconstruction is executed using three distinct steps:

1. perform an unlimited $k$-exact least-squares reconstruction in each cells;

2. evaluate the smoothness indicator in each cell to determine whether the solution is fully-resolved (smooth) or under-resolved (discontinuous); and

3. perform a limited piecewise linear reconstruction on the cells in which the solution is under-resolved (discontinuous).

All these steps are explained in detail in the following sections. Section 4.2 explains how the $k$-exact reconstruction can be performed. Thereafter, Section 4.3 puts light on the method used to compute smoothness indicator. Section 3.4 from Chapter 3 has information regarding piecewise linear reconstruction. Once CENO reconstruction is performed, inviscid and viscous fluxes could be evaluated using appropriate methods. These methods are discussed in Section 4.4. While solving the governing equations for LES, source terms also need to be determined in each computational cell. Evaluation of these source terms with high-order accuracy is explained in Section 4.5. The modified high-order residual computed by the evaluation of all these terms can then be used to march the solution in time, using high-order RK4 time marching method mentioned in Section 3.5.
4.2 \( k \)-Exact Least-Squares Reconstruction

CENO scheme makes use of the so-called \( k \)-exact least-squares reconstruction technique developed by Barth and Fredrickson [64, 65]. The reconstructed solution is evaluated in a cell from the \( k \)-th-order Taylor series expansion of the solution variable \( U \), about the cell center. Recall that Godunov-type FV methods have flexibility to carry out reconstructions using both primitive and conservative variables. Therefore, \( U \) is henceforth used to represent any component of the primitive solution vector, \( \mathbf{W} \), or a conservative solution vector, \( \mathbf{U} \). It should also be noted that the value of ‘\( k \)’ in \( k \)-exact reconstruction determines the spatial accuracy of the solution. In smooth regions, a \( k \)-th-order reconstruction results in a \((k+1)\)-th-order spatially accurate scheme for hyperbolic Euler equations, and a \( k \)-th-order accurate scheme for elliptic Navier-Stokes equations. This is due to the involvement of viscous terms in Navier-Stokes equations (as discussed further in Section 4.4). Since this work makes use of the Favre-filtered Navier-Stokes equations, spatial accuracy of the scheme corresponds to the order of the reconstruction polynomial, \( k \).

The \( k \)-th-order Taylor series of the spatial distribution of a solution quantity, \( U_{ijk} \), inside the cell with index \( ijk \), about the cell-centroid \((x_{ijk}, y_{ijk}, z_{ijk})\) can be expressed as

\[
U_{ijk}^k(x, y, z) = \sum_{p_1=0}^{k} \sum_{p_2=0}^{k} \sum_{p_3=0}^{k} (x - x_{ijk})^{p_1} (y - y_{ijk})^{p_2} (z - z_{ijk})^{p_3} D_{p_1p_2p_3}, \tag{4.1}
\]

where \( D_{p_1p_2p_3} \) represents the constant coefficients of the Taylor series expansion given by

\[
D_{p_1p_2p_3} = \frac{1}{p_1!p_2!p_3!} \frac{\partial^{p_1} U}{\partial x^{p_1}} \frac{\partial^{p_2} U}{\partial y^{p_2}} \frac{\partial^{p_3} U}{\partial z^{p_3}}. \tag{4.2}
\]

These constant coefficients are the only unknowns in the right-hand-side (RHS) of Eq. (4.1). They are referred to as the unknown derivatives in the following sections. The number of these unknown derivatives varies according to the order of reconstruction. In three dimensions, the number, \( N \), of unknown derivatives are given by

\[
N = \frac{1}{3!} \prod_{n=1}^{3} (k + n) = \frac{(k + 1)(k + 2)(k + 3)}{6}. \tag{4.3}
\]

Using this relation, one can easily find that there are 10 unknown derivatives for a second-order (piecewise quadratic) reconstruction, 20 unknowns for a third-order (piecewise cubic) reconstruction and 35 for a fourth-order (piecewise quartic) reconstruction. Determination of these unknown derivatives can be achieved by utilizing the following three important conditions outlined by Barth [64] for \( k \)-exact reconstruction procedure:

1. the reconstruction should conserve the mean value in the cell to which its applied;
2. the reconstruction should be \( k \)-exact, in a sense that it should represent polynomials of degree \( \delta \leq k \) exactly; and

3. the reconstruction should have a compact support from the stencils.

These three conditions are discussed in detail in the following section.

### 4.2.1 Stencils Used for Reconstruction

The unknown derivatives are determined using the least-squares method, with a similar approach outlined in Section 3.4.2. In particular, the cell-averaged values from the neighboring cells are used along with the reconstruction conditions outlined by Barth to obtain a system of equations containing at least as many equations as the number of unknown derivatives. In this work, the neighboring cells used for the reconstruction of solution in cell \( ijk \) are denoted with index \( pqr \), and the collection of these neighbouring cells is referred to as the ‘reconstruction stencil’.

The first condition outlined by Barth demands the reconstruction to preserve the cell-averaged solution values. In cell \( ijk \), this condition can be met by imposing the following constraint:

\[
U_{ijk} = \frac{1}{V_{ijk}} \iint_U U_{ijk}(x, y, z) dV
\]

\[
= \frac{1}{V_{ijk}} \iiint \left( \sum_{p_1=0}^{k} \sum_{p_2=0}^{k} \sum_{p_3=0}^{k} (x - x_{ijk})^{p_1} (y - y_{ijk})^{p_2} (z - z_{ijk})^{p_3} D_{p_1p_2p_3} \right) dV. \tag{4.4}
\]

Realizing that the derivatives are constant, Eq. (4.4) can be further simplified to

\[
U_{ijk} = \sum_{p_1=0}^{k} \sum_{p_2=0}^{k} \sum_{p_3=0}^{k} [D_{p_1p_2p_3} \frac{1}{V_{ijk}} \iiint (x - x_{ijk})^{p_1} (y - y_{ijk})^{p_2} (z - z_{ijk})^{p_3} dV]. \tag{4.5}
\]

It should be realized that this constraint is not enough to determine the unknown derivatives. For instance, 35 unknown derivatives are required to conduct a piecewise quartic reconstruction (as mentioned earlier). Therefore, for this case, there should be at least 35 constraints similar to Eq. (4.5). These constraints can be found by extending the polynomial representation of the solution in cell \( ijk \) to its neighbouring cells \( pqr \), and requiring that the cell-averaged quantity in cell \( pqr \) can also be recovered by integrating \( U_{ijk}^k(x, y, z) \) over its domain, \( V_{pqr} \). This constraint
can thus be expressed as

$$U_{pqr} = \frac{1}{V_{pqr}} \iiint U_{ijk}^k(x, y, z) dV$$

$$= \sum_{p_1=0}^{k} \sum_{p_2=0}^{k} \sum_{p_3=0}^{k} \left[ D_{p_1 p_2 p_3} \frac{1}{V_{pqr}} \iiint (x - x_{ijk})^{p_1} (y - y_{ijk})^{p_2} (z - z_{ijk})^{p_3} dV \right]. \quad (4.6)$$

In this way, to obtain a set of enough equations to solve for unknown derivatives, the number of neighbours considered in the stencil must be at least equal to the number of unknown derivatives. In practice, it is advantageous to use larger number of neighbouring cells than required. A larger size of the stencil generally makes the computational framework for reconstruction more robust for the cases using stretched meshes and/or for the cases where the solution gradients are not aligned with the mesh [24].

The second condition necessitates the $k$-exactness of the reconstruction. Mathematically, this condition is equivalent to

$$U_{ijk}^k(x, y, z) - U_{\text{exact}}(x, y, z) = O\left((\Delta_{x,y,z})^{k+1}\right), \quad (4.7)$$

where $\Delta_{x,y,z}$ is the mesh spacing that is $\Delta_{x,y,z} = (\Delta x \Delta y \Delta z)^{\frac{1}{3}}$ in three-dimensional Cartesian grids.

The third condition deals with the locality of and the number of reconstruction cells used to form the reconstruction stencil. The condition of “compact stencils” demands the reconstruction stencil to have cells that are physically near the cell being reconstructed [66]. CENO scheme reconstructs the central cell, based on the information from neighbouring cells around it. In structured grids, the stencil size may be defined by the number of layers of cells around the cell $ijk$. In general, the first layer includes the nearest neighbours of the cell $ijk$, the second layer consists of the neighbours after that, and so on. Figure 4.1, provided by Rashad [27], depicts the first and the second layer of cells in a reconstruction stencil of a three-dimensional Cartesian mesh. This figure also provides the numbering convention used to identify cells in the first and the second layers. The cell denoted with a 0 is cell $ijk$ (or, 000), on which the reconstruction should be performed. The positive $x$, $y$ and $z$ directions are identified with $I$, $J$ and $K$, respectively. From this figure, it can easily be observed that in three dimensions, a one-layer stencil consists of 27 cells (3 cells in each direction), and a two-layer stencil has 125 cells (5 cells in each direction). In this work, a one-layer stencil is used for $k = 0, 1$ and 2 reconstructions. For $k = 3$ and 4 reconstructions, a two-layer stencil is employed. The average solution quantities in the cells of a reconstruction stencil can be used in Eq. (4.5) and (4.6) to obtain an overdetermined system of equations.
4.2.2 The Least-Squares Problem

The goal of this section is to provide mathematical development of an overdetermined system of equations and to outline the technique used to solve it. This section first discusses the formation of overdetermined system of equations. Then, it explains how geometric coefficients and moments appearing in the matrix of the least-squares problem is efficiently computed. Thereafter, solving the system of equations with a least-squares technique is discussed.

At this stage, it is worthwhile to understand the basics of a least-squares problem. An overdetermined system of equations can be written in the matrix-form as

\[ \mathbf{A} \mathbf{x} - \mathbf{b} = \mathbf{e}, \quad (4.8) \]

where \( \mathbf{A} \) is a coefficient matrix of size \( M \times N \), \( \mathbf{x} \) is the vector of \( N \) unknown quantities, \( \mathbf{b} \) is a vector containing \( M \) components of known values. Since, in general, \( \mathbf{A} \mathbf{x} \neq \mathbf{b} \) for overdetermined system of equations, a residual vector, \( \mathbf{e} \), represents errors incurred while solving the least-squares problem. Here, \( M \) refers to the number of equations in the system, and \( N \) refers to the number of unknowns. Naturally then, for the system of equations to be eligible to solve
for unknowns, \( M \) should be greater than or equal to \( N \). The goal of a least-squares problem is to determine the unknowns for which squares of each components of the error vector, \( e \), are minimized. In terms of the mathematical notations used, a linear least-squares problem may be defined as:

\[
\text{Find the unknown vector } x, \text{ such that } ||Ax - b||^2 = ||e||^2 \text{ is minimized.} \tag{4.9}
\]

**Formation of the Overdetermined System**

For the evaluation of the unknown derivatives, \( D_{p1p2p3} \), the least-squares problem of the overdetermined system \( Ax - b = e \) needs to be solved. Therefore, there is a need to form an overdetermined system of equations from Eq. (4.5) and (4.6). It should be noted that in these equations, the first term is simply \( D_{000} \), since all the remaining terms equal to unity when \( p_1 = p_2 = p_3 = 0 \). Using this information, Eq. (4.5) can be expressed as

\[
\bar{U}_{ijk} = D_{000} + \sum_{p_1=0}^{k} \sum_{p_2=0}^{k} \sum_{p_3=0}^{k} [D_{p1p2p3} \frac{1}{V_{ijk}} \iiint (x - x_{ijk})^{p_1} (y - y_{ijk})^{p_2} (z - z_{ijk})^{p_3} d\mathcal{V}] \tag{4.10}
\]

At this point, the so-called geometric moments of cell \( ijk \) taken about its own cell center is defined as

\[
\left( x^{p_1} y^{p_2} z^{p_3} \right)_{ijk} \equiv \frac{1}{V_{ijk}} \iiint (x - x_{ijk})^{p_1} (y - y_{ijk})^{p_2} (z - z_{ijk})^{p_3} d\mathcal{V}. \tag{4.11}
\]

In terms of these geometric moments, Eq. (4.10) can be written as

\[
\bar{U}_{ijk} = D_{000} + \sum_{p_1=0}^{k} \sum_{p_2=0}^{k} \sum_{p_3=0}^{k} D_{p1p2p3} \left( x^{p_1} y^{p_2} z^{p_3} \right)_{ijk}. \tag{4.12}
\]

Rearranging terms in this equation provides an expression used to determine \( D_{000} \) with respect to other unknowns, \( D_{p1p2p3} \), given by:

\[
D_{000} = \bar{U}_{ijk} - \sum_{p_1=0}^{k} \sum_{p_2=0}^{k} \sum_{p_3=0}^{k} D_{p1p2p3} \left( x^{p_1} y^{p_2} z^{p_3} \right)_{ijk}. \tag{4.13}
\]

Using a similar argument about the first term being a constant \( D_{000} \), Eq. (4.6) (that represents the extension of piecewise polynomial reconstructed in cell \( ijk \) to the neighbouring cell \( pqr \)) can be expressed as

\[
\bar{U}_{pqr} = D_{000} + \sum_{p_1=0}^{k} \sum_{p_2=0}^{k} \sum_{p_3=0}^{k} D_{p1p2p3} \left( x^{p_1} y^{p_2} z^{p_3} \right)_{pqr}. \tag{4.14}
\]
where \( \left( x^{p_1} y^{p_2} z^{p_3} \right)_{pq} \) is the geometric moment of cell \( pqr \) taken about the cell center of cell \( ijk \) defined in a similar fashion as in Eq. (4.11), given by

\[
\left( x^{p_1} y^{p_2} z^{p_3} \right)_{pq} = \frac{1}{V_{pqr}} \iiint (x - x_{ijk})^{p_1} (y - y_{ijk})^{p_2} (z - z_{ijk})^{p_3} \, dV.
\] (4.15)

Substituting the expression for \( D_{000} \), given in Eq. (4.13), into Eq. (4.14) results in the following expression:

\[
\bar{U}_{pqr} - \bar{U}_{ijk} = \sum_{p_1=0}^{k} \sum_{p_2=0}^{k} \sum_{p_3=0}^{k} D_{p_1p_2p_3} \left( x^{p_1} y^{p_2} z^{p_3} \right)_{ijk,pqr} - \left( x^{p_1} y^{p_2} z^{p_3} \right)_{ijk,pqr}.
\] (4.16)

where

\[
\left( x^{p_1} y^{p_2} z^{p_3} \right)_{ijk,pqr} = \left( x^{p_1} y^{p_2} z^{p_3} \right)_{pq} - \left( x^{p_1} y^{p_2} z^{p_3} \right)_{ijk}.
\] (4.17)

Eq. (4.16) represents an overdetermined system of equations, where \( pq \) represents all the neighbouring cells used for reconstruction, and \( ijk \) represents the cell in which the solution is being reconstructed. For clarity, in the following expressions, cell index \( ijk \) is referred to as \( \alpha \) and the neighbouring cells indices \( pq \) are denoted as \( \beta \). It should be noted that according to the numbering convention depicted in Figure 4.1, \( \alpha = 0 \) and \( 1 \leq \beta \leq M \), where \( M \) is one less than the number of cells used in the reconstruction stencil. For example, for a \( k = 4 \) reconstruction, \( M = 125 - 1 = 124 \), and therefore, \( 1 \leq \beta \leq 124 \). Note that since the zeroth-derivative \( D_{000} \), is already known (and is given by Eq. (4.13)), there are only 34 unknown derivatives for a \( k = 4 \) reconstruction. Thus, there are 124 equations in the overdetermined system that are used to determine 34 unknown derivatives.

Following these conventions, Eq. (4.16) can be rearranged in the matrix-form of \( \mathbf{A}\mathbf{x} = \mathbf{b} = \mathbf{e} \), as follows:

\[
\begin{bmatrix}
(x^0 y^0 z)_{\alpha_1} & \ldots & (x^p y^p z^p)_{\alpha_1} & \ldots & (x^k y^k z^k)_{\alpha_1} \\
\vdots & & \vdots & & \vdots \\
(x^0 y^0 z)_{\alpha_\beta} & \ldots & (x^p y^p z^p)_{\alpha_\beta} & \ldots & (x^k y^k z^k)_{\alpha_\beta} \\
\vdots & & \vdots & & \vdots \\
(x^0 y^0 z)_{\alpha_M} & \ldots & (x^p y^p z^p)_{\alpha_M} & \ldots & (x^k y^k z^k)_{\alpha_M}
\end{bmatrix}
\begin{bmatrix}
D_{001} \\
\vdots \\
D_{p_1p_2p_3} \\
\vdots \\
D_{k00}
\end{bmatrix}
\begin{bmatrix}
\bar{U}_{1} - \bar{U}_{\alpha} \\
\vdots \\
\bar{U}_{\beta} - \bar{U}_{\alpha} \\
\vdots \\
\bar{U}_{M} - \bar{U}_{\alpha}
\end{bmatrix}
= \begin{bmatrix}
e_1 \\
\vdots \\
e_\beta \\
\vdots \\
e_M
\end{bmatrix},
\]

where \( M \) represents the total number of neighboring cells in the reconstruction stencil, and \( N \) is the total number of unknown derivatives minus one, as discussed earlier. At this point, it is
worthwhile to note that the matrix $A$ depends solely on mesh geometry, and vector $b$ depends only on the known cell-averaged solution quantities.

With the purpose of adding more weightage to the information from the neighboring cells that are physically closer to cell $\alpha$, a geometric weighting is applied in each cell of the reconstruction stencil. This weighting works towards fulfilling the third condition outlined by Barth [64] for $k$-exact reconstruction procedure, mentioned in the introduction of this Section. The geometric weights, $w_{\alpha\beta}$, are multiplied with each terms in matrix $A$ and vector $b$ in order to have a more localized reconstruction. In this work, the geometric weighting is taken to be

$$ w_{\alpha\beta} = \frac{1}{|\vec{r}_\beta - \vec{r}_\alpha|^2}, \quad (4.18) $$

where $\vec{r}$ is the position vector of the indicated cell. This weighting process is found to be more significant while dealing with curvilinear and unstructured meshes. Before solving the overdetermined system of equations, geometric moments present in matrix $A$ need to be efficiently computed. The techniques used for the computationally-efficient determination of these moments is the subject of the following subsection.

**Geometric Coefficients and Moments**

To achieve a balance between computational cost and memory requirements, the coefficient matrix $A$ in each cell is neither completely formed and stored before the first time-step, nor is it computed from scratch at each time-step. In turn, the geometric moment in each cell about its cell center is precomputed and stored such that they can be readily used to form matrix $A$. Therefore, in each neighboring cell $\beta$, the moments given by

$$ \left( \frac{x^{p_1} y^{p_2} z^{p_3}}{\beta} \right) \equiv \frac{1}{V_\beta} \iiint_{V_{\beta}} (x - x_\beta)^{p_1} (y - y_\beta)^{p_2} (z - z_\beta)^{p_3} \, dV. \quad (4.19) $$

is stored. Note that for Cartesian grids, where the limits of the integration are known, Eq. (4.19) could be easily evaluated analytically. These stored moments can be used to find the coefficients of matrix $A$ using appropriate values for the powers $(p_1, p_2$ and $p_3)$ of $\left( \frac{x^{p_1} y^{p_2} z^{p_3}}{\beta} \right)$. After a simple derivation, as presented in [27], the coefficients of the matrix $A$ can be expressed as

$$ \left( \frac{x^{p_1} y^{p_2} z^{p_3}}{\alpha\beta} \right) = \sum_{\xi=0}^{p_1} \sum_{\ell=0}^{p_2} \sum_{\varphi=0}^{p_3} \left[ C_{p_1}^{\xi} C_{p_2}^{\ell} C_{p_3}^{\varphi} \cdot \Delta x_{\beta}^\xi \Delta y_{\beta}^\ell \Delta z_{\beta}^\varphi \cdot \left( x^{(p_1-\xi)} (p_2-\ell) z (p_3-\varphi) \right)_{\beta} \right] - \left( \frac{x^{p_1} y^{p_2} z^{p_3}}{\alpha} \right), \quad (4.20) $$
where $C_{p1}$, $C_{p2}$ and $C_{p3}$ are the coefficients of binomial expansion (which will be defined in the following lines), $\Delta x_{\beta\alpha} = (x_{\beta} - x_{\alpha})$, $\Delta y_{\beta\alpha} = (y_{\beta} - y_{\alpha})$, $\Delta z_{\beta\alpha} = (z_{\beta} - z_{\alpha})$, and

$$
\left( x(p_1 - \xi) y(p_2 - \ell) z(p_3 - \varphi) \right)_\beta \equiv \frac{1}{V_{\beta}} \iiint (x - x_\beta)^{p_1 - \xi} (y - y_\beta)^{p_2 - \ell} (z - z_\beta)^{p_3 - \varphi} dV. \tag{4.21}
$$

The right-hand side of Eq. (4.20) no longer contains the geometric moments of neighbouring cell $\beta$ taken about cell $\alpha$. The geometric moments of the neighbouring cells are taken about their own cell centers, as further explicated by Eq. (4.21). Indeed, the term $\left( x(p_1 - \xi) y(p_2 - \ell) z(p_3 - \varphi) \right)_\beta$ is simply a shift in the powers of the stored $\left( x^{p_1} y^{p_2} z^{p_3} \right)_\beta$ term. Note that this process of storing geometric moments at each cell is significantly cheaper rather than evaluating the RHS of Eq. (4.17) by computing moments of each neighboring cells $\beta$ relative to cell $\alpha$, and repeating this for all the cells in the solution domain.

The yet-undetermined coefficients of binomial expansion can be computed recursively, with initial values, $C_{p1}^0 = C_{p2}^0 = C_{p3}^0 = 1$, as

$$
C_{p1}^\xi = \frac{p_1 - \xi + 1}{\xi} C_{p1}^{\xi-1}, \quad C_{p2}^\ell = \frac{p_2 - \ell + 1}{\ell} C_{p2}^{\ell-1}, \quad C_{p3}^\varphi = \frac{p_3 - \varphi + 1}{\varphi} C_{p3}^{\varphi-1}. \tag{4.22}
$$

This concludes the formation of the overdetermined system of equations given by $Ax - b = e$. Now, this least-squares problem ought to be solved to obtain the unknown derivatives.

**Solving The Least-Squares Problem**

This work utilizes the Householder QR factorization method provided by the Linear Algebra Package (LAPACK) numerical library [67] to solve the least-squares problem. In the computational framework that the author is associated with, use of pseudo-inverse matrix via Singular Value Decomposition (SVD) is also possible.

Householder QR factorization method decomposes matrix $A$ as a product of an orthogonal matrix $Q$ and an upper-triangular matrix $R$, which helps to solve the least-squares problem in a computationally efficient manner. The reader is referred to the textbooks by Lawson [68], Nicholson [69] or Nocedal et.al [70] for further discussion about this technique. The mathematical description of these least-squares solvers is beyond the scope of this work.

Once the unknown derivatives are determined, it is easily possible to obtain a piecewise polynomial representation of the solution quantities in a cell, using Eq. (4.1). This section concludes the implementation of unlimited high-order $k$-exact reconstruction for CENO scheme. This $k$-exact reconstruction is used in smooth-regions of the solution domain. In the vicinity of
shocks and discontinuities, the scheme reverts down to limited piecewise linear reconstruction. The so-called smoothness indicators are used to distinguish between smooth and non-smooth regions in the domain.

### 4.3 Smoothness Indicator

After $k$-exact reconstruction is performed in all the cells in the domain, smoothness indicators are determined in each of these cells. The value of each smoothness indicator dictates whether the reconstruction in that cell reverts down to limited piecewise linear reconstruction. Smoothness indicator, $S$, in each cell is calculated based on a solution smoothness parameter, $\gamma$, as follows:

$$S = \frac{\gamma}{\max((1-\gamma),\epsilon)} \frac{(M-N)}{(N-1)},$$  \hspace{1cm} (4.23)

where $\epsilon$ is a tolerance value (taken to be $10^{-8}$) added to avoid division by zero. Consistent to the previously used notations, $M$ refers to the size of the stencil (the number of neighbouring cells used for reconstruction) and $N$ is the number of unknown derivatives (degree of freedom).

The smoothness parameter, $\gamma$, for cell $\alpha$ with solution variable $U$ has the form

$$\gamma = 1 - \frac{\sum_\beta \left(U^k_\beta(x_\beta, y_\beta, z_\beta) - U^k_\alpha(x_\beta, y_\beta, z_\beta)\right)^2}{\sum_\beta \left(U^k_\beta(x_\beta, y_\beta, z_\beta) - \bar{U}_\alpha\right)^2}.$$  \hspace{1cm} (4.24)

Notice that in Eq. (4.24), the cell indices have been kept consistent to the previously defined conventions, wherein $\alpha$ refers to the cell of interest, $ijk$, and $\beta$ denotes the neighbouring cells, $pqr$. $U^k_\beta(x_\beta, y_\beta, z_\beta)$ denotes the reconstruction polynomial of the neighbouring cell $\beta$ evaluated at its own cell center. $U^k_\alpha(x_\beta, y_\beta, z_\beta)$, on the other hand, refers to the value of polynomial reconstructed at cell $\alpha$ evaluated at the cell center of the neighbouring cell $\beta$.

The smoothness parameter, $\gamma$, represents a normalized difference between the actual reconstructed value at the neighbouring cell $\beta$ and the value at cell $\beta$ projected from reconstruction in cell $\alpha$. It can be observed that the maximum value that $\gamma$ can have is 1. From Eq. (4.23), it can be inferred that as $\gamma$ approaches unity, $S$ grows very rapidly. This is further explicated by Figure 4.2, which depicts growth of the function $f(\gamma) = \frac{\gamma}{(1-\gamma)}$ when $\gamma$ approaches unity. It is worthwhile to mention that previous works by Ivan and Groth [23, 24] and Rashad [27] have suggested that smoothness indicator works well even if only one-layer-stencil of neighbouring cells is used for its computation. The use of only one layer of stencil is advantageous also because it helps in minimizing the computational cost.

The calculated value of $S$ is compared to the critical pass/no-pass cutoff value, $S_c$. Typically,
S_c is in the range of 1,000–5,000 [23, 24]. In this work, S_c is mostly set to be 4,000. It is found and verified by Ivan and Groth [23, 24] that, S ≫ S_c by some orders of magnitude in smooth regions. Therefore, all the cells having S ≤ S_c are flagged as non-smooth regions. These under-resolved cells then make use of the limited piecewise linear reconstruction. In this way, CENO reconstruction scheme makes use of both high-order k-exact reconstruction (on smooth regions) and monotonicity preserving linear reconstruction (on non-smooth regions).

This concludes the discussion on CENO reconstruction. The following section discusses the evaluation of the terms in RHS of Eq. (3.13) with high-order accuracy.

4.4 Flux Evaluation

The semi-discrete form of the governing equations presented in Eq. (3.13) can be specialized for Cartesian mesh, to obtain the following expression

$$
\frac{dU_{ijk}}{dt} = -\frac{1}{\Delta x} \left[ \sum_{m=1}^{NG} \left\{ \omega \left( \left[ F^I_{i+\frac{1}{2},j,k} - F^I_{i-\frac{1}{2},j,k} \right] - \left[ F^V_{i+\frac{1}{2},j,k} - F^V_{i-\frac{1}{2},j,k} \right] \right) \right\}_m \right] \\
-\frac{1}{\Delta y} \left[ \sum_{m=1}^{NG} \left\{ \omega \left( \left[ G^I_{i,j+\frac{1}{2},k} - G^I_{i,j-\frac{1}{2},k} \right] - \left[ G^V_{i,j+\frac{1}{2},k} - G^V_{i,j-\frac{1}{2},k} \right] \right) \right\}_m \right] \\
-\frac{1}{\Delta z} \left[ \sum_{m=1}^{NG} \left\{ \omega \left( \left[ H^I_{i,j,k+\frac{1}{2}} - H^I_{i,j,k-\frac{1}{2}} \right] - \left[ H^V_{i,j,k+\frac{1}{2}} - H^V_{i,j,k-\frac{1}{2}} \right] \right) \right\}_m \right] \\
+ S_{ijk}.
$$

\(4.25\)
The computation of RHS of Eq. (4.25) involves the high-order evaluation of inviscid fluxes, viscous fluxes and the cell-averaged source terms.

### 4.4.1 Inviscid Flux Evaluation

Once the limited high-order CENO reconstruction is obtained for the cell of interest, an approximate or exact Riemann solver can be used to compute the high-order fluxes at each quadrature point of the cell interface. Thus, after obtaining the high-order left and right solution states at a quadrature point in a cell interface, Eq. (3.15) is used to determine the inviscid flux at that quadrature point. To maintain high-order accuracy, the numerical integration scheme used should be at least as accurate as the reconstruction. The following section discusses the Gauss quadrature integration procedure utilized in this work to maintain solution accuracy.

#### Gauss Quadrature Integration in two-dimensions

The Gauss-Legendre quadrature integration [71] is used to evaluate high-order integration of the fluxes in each cell interface. The number of Gauss quadrature points, $N_G$, in these cell faces depends on the desired order of accuracy. In general, one-dimensional Gauss-Legendre quadrature integration can be extended to two-dimensions (in $xy$ plane) as follows:

\[
I = \int_{y_a}^{y_b} \int_{x_a}^{x_b} F(x, y) \, dx \, dy = (x_b - x_a) \int_{y_a}^{y_b} \left[ \sum_{i=1}^{n} \omega_i F(x_i, y) + \epsilon_x^n \right] dy \\
= (x_b - x_a)(y_b - y_a) \sum_{j=1}^{n} \omega_j \left[ \sum_{i=1}^{n} \omega_i F(x_i, y_j) + \epsilon_x^n \right] + \epsilon_y^n \\
= (x_b - x_a)(y_b - y_a) \sum_{j=1}^{n} \sum_{i=1}^{n} \omega_j \omega_i F(x_i, y_j) + \epsilon_{xy}^{nn}.
\]  

Here, $I$ is the result of two-dimensional numerical integration, $n$ is the number of quadrature points in each direction used for the integration, $x_i$ and $y_i$ are the locations of quadrature points, and $\omega_i$ and $\omega_j$ are the associated weights of quadrature points $x_i$ and $y_i$, respectively. The location of the quadrature points relative to the cell center in $x$ and $y$ directions are given by $u_i$ and $v_j$, respectively. The weights associated with these relative positions for 1, 2 and 3-point 1D Gauss quadrature rules are provided in Table 4.4.1. For Cartesian grids, a simple transformation can be used to relate $x_i$ and $y_j$ to $u_i$ and $v_j$, respectively, given by:

\[
x_i = \frac{(x_b - x_a)}{2} u_i + \frac{1}{2}(x_b + x_a), \quad \text{and} \quad (4.27a)
\]

\[
y_j = \frac{(y_b - y_a)}{2} v_j + \frac{1}{2}(y_b + y_a). \quad (4.27b)
\]
Table 4.1: Gauss-Legendre quadrature points and weights

<table>
<thead>
<tr>
<th>n</th>
<th>ω_i, ω_j</th>
<th>u_i, v_j</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0.5</td>
<td>$-\frac{1}{\sqrt{3}}$</td>
</tr>
<tr>
<td>3</td>
<td>$\frac{5}{18}$</td>
<td>$-\sqrt{\frac{3}{5}}$</td>
</tr>
</tbody>
</table>

Eq. (4.25) uses a $N_G$-point Gaussian quadrature numerical integration to evaluate high-order integration of the fluxes in each cell faces. In this equation, $N_G = n^2$, and the collective weight for each quadrature point in a cell face is simply $\omega_m = \omega_j \cdot \omega_i$. Recall that this is the case only for the cell faces corresponding to $xy$-plane. Using this example, it is straightforward to obtain the weights and locations for other faces.

For $k = 0$ and $k = 1$ reconstruction, one-point integration (with $n = 1$ and $N_G = 1$) is sufficient to preserve the accuracy [25, 54, 66]. This is to say that the fluxes could be computed at the centroid of each cell face. For $k = 2, 3$ and 4 reconstructions, $N_G = 4$ (with $n = 2$) should be used. Generally, a $(n = 2)$-point linear Gauss quadrature integration produces fourth-order accurate results. Therefore, to integrate the fluxes computed with fourth-order accurate reconstruction, four Gauss quadrature points distributed symmetrically over the cell face should be used.

### 4.4.2 Viscous Flux Evaluation

Recall that the viscous fluxes presented in Eq. (3.4), (3.5) and (3.6) depend on both the solution states and the solution gradients. At a quadrature point in a cell interface, these fluxes can therefore be expressed as

$$\vec{F}_V \cdot \hat{n} = \vec{F}_V (\vec{U}_{\text{left}}, \nabla \vec{U}_{\text{left}}, \vec{U}_{\text{right}}, \nabla \vec{U}_{\text{right}}, \hat{n}).$$

(4.28)

To obtain high-order viscous fluxes, both hyperbolic solution quantities and their elliptic gradients should be high-order accurate. Ollivier-Gooch and Van-Altena [72] have proposed that high-order accurate gradients can be obtained by differentiating the $k$-exact reconstruction
polynomial that is already available. Therefore, with the reconstruction polynomial defined as

\[ U_{ijk}^k(x, y, z) = \sum_{p_1=0}^{k} \sum_{p_2=0}^{k} \sum_{p_3=0}^{k} (x - x_{ijk})^{p_1} (y - y_{ijk})^{p_2} (z - z_{ijk})^{p_3} D_{p_1p_2p_3}, \]  

(4.29)

the components of high-order gradients can be computed as

\[ \frac{\partial U_{ijk}^k(x, y, z)}{\partial x} = \sum_{p_1=1}^{k} \sum_{p_2=0}^{k} \sum_{p_3=0}^{k} p_1 (x - x_{ijk})^{p_1-1} (y - y_{ijk})^{p_2} (z - z_{ijk})^{p_3} D_{p_1p_2p_3}, \]  

(4.30a)

\[ \frac{\partial U_{ijk}^k(x, y, z)}{\partial y} = \sum_{p_1=0}^{k} \sum_{p_2=1}^{k} \sum_{p_3=0}^{k} p_2 (x - x_{ijk})^{p_1} (y - y_{ijk})^{p_2-1} (z - z_{ijk})^{p_3} D_{p_1p_2p_3}, \]  

(4.30b)

\[ \frac{\partial U_{ijk}^k(x, y, z)}{\partial z} = \sum_{p_1=0}^{k} \sum_{p_2=0}^{k} \sum_{p_3=1}^{k} p_3 (x - x_{ijk})^{p_1} (y - y_{ijk})^{p_2} (z - z_{ijk})^{p_3-1} D_{p_1p_2p_3}. \]  

(4.30c)

Using these expressions, the gradients at each of the quadrature points of the cell interfaces can be determined. Then, at these quadrature points, there exists ‘right’ and ‘left’ solution states as well as ‘right’ and ‘left’ gradients. Here, the solution states and gradients are calculated by taking arithmetic means of these states, respectively. For instance, at a quadrature point in a cell interface parallel to the \( xy \)-plane, the solution states and the gradients are obtained as follows:

\[ U_{i+\frac{1}{2},j,k} = \frac{(U_{\text{left}} + U_{\text{right}})}{2}, \]  

(4.31a)

\[ \nabla U_{i+\frac{1}{2},j,k} = \frac{(\nabla U_{\text{left}} + \nabla U_{\text{right}})}{2}. \]  

(4.31b)

These solution states and gradients can then be directly used in Eq. (3.4), (3.5) and (3.6) to obtain viscous fluxes at the relevant quadrature point. High-order integration is performed in a similar fashion as explained earlier.

Recall that for hyperbolic systems, a \( k \)-th order reconstruction polynomial facilitates a \((k+1)\)-th order accurate scheme. The differentiation operation in Equations (4.30) causes these equations to have the leading truncation error one-order less than the error in \( k \)-exact reconstruction polynomial, Eq. (4.29). As a result, to obtain \( k \)-th order spatially accurate solutions in case of elliptic operators or Navier-Stokes equations, a \( k \)-th order reconstruction polynomial should be used.
4.5 Source Vector Evaluation

The source vector, $\mathbf{S}$, contains terms related to the chemistry and turbulence modelling. Note that, similar to the viscous fluxes, the source terms that appear in Eq. (3.7) depend on the reconstructed solutions and their gradients. However, it should be noted that these are no longer the solutions and gradients computed at each cell interface. Indeed, these are computed at the Gauss quadrature points inside the volume of the cell. Therefore, the source vector evaluation makes use of Gauss-Legendre numerical integration in three dimensions.

Recall that the cell-averaged source vector, $\overline{\mathbf{S}}_{ijk}$ is given by

$$\overline{\mathbf{S}}_{ijk} = \frac{1}{V_{ijk}} \iiint S dV.$$  \hfill (4.32)

Extending the previously explained two-dimensional Gauss-quadrature integration rule to three-dimensions, the following expression is obtained for the cell-averaged source terms:

$$\overline{\mathbf{S}}_{ijk} = \sum_{a=1}^{n} \sum_{b=1}^{n} \sum_{c=1}^{n} \omega_a \omega_b \omega_c S \left( \mathbf{U}_{ijk}^k(x_a, y_b, z_c), \nabla \mathbf{U}_{ijk}^k(x_a, y_b, z_c) \right),$$  \hfill (4.33)

where $(x_a, y_b, z_c)$ represents the location of the Gauss-quadrature points inside cell $ijk$. In Cartesian grids, this location can be determined from the relative location of Gauss quadrature points with respect to the cell centroid (given by $(u_a, v_b, w_c)$), using expressions similar to those given in Eq. (4.27). With the help of the weights given in Table 4.4.1, Eq. (4.33) can then determine the cell-averaged source vector with high-order accuracy.

It is reiterated that to get a fourth-order accuracy, a $k=4$ reconstruction is performed. Similarly, Gauss-Legendre integration using $n=2$ (two quadrature points in each direction) is exploited. This means that computation of source terms with fourth-order accuracy involves a total of eight quadrature points inside each computational cell.

4.6 Boundary Conditions

To maintain the global high-order accuracy, the high-order treatment of boundary conditions is also essential. Ivan and Groth [23, 54] have investigated different ways of dealing with high-order boundaries. This work makes use of the so called ghost cells to ensure high-order boundaries.

Ghost cells form an additional layer of cells extending beyond the block or the physical solution domain. In three-dimensional Cartesian box-grids, it can be visualized as a smaller box inside a larger box, where the interior box represents the actual solution domain, and the larger box
includes both the solution domain and ghost cells. In all the ghost-cells, the cell-averaged values are defined, according to the prespecified boundary conditions.

Depending on the size of the reconstruction stencil and that of the stencil used to compute smoothness indicator, the required number of ghost cells used can be determined. Generally, a CENO scheme using a two-layer reconstruction stencil and a two-layer smoothness indicator stencil would require five ghost cells in each direction of the boundaries. In this case, if only a one-layer smoothness indicator stencil is used, the scheme would require four ghost cells in each direction. The purpose of each of these ghost cells in Cartesian grids is discussed by Rashad [27].

This work makes use of parallel multi-block implementation, where the physical solution domain is partitioned into multiple small blocks. This partitioning proves to be advantageous in saving the computational time for a flow problem using large grid sizes. Essentially, these blocks are distributed to various processors, and they are treated as an entire solution domain in each processor. The solution in these blocks are computed separately in different processors. In this work, ghost cells serve to add the boundary conditions to each of the blocks. In the exterior blocks (having cells at the boundary of physical solution domain), ghost cells are updated via the provided boundary conditions. For the interior block boundaries, ghost cells overlap with the physical cells of the neighbouring blocks. The average solution values in these ghost-cells are updated according to the values in the corresponding physical cells in the neighbouring blocks. This is done by using message passing interface (MPI) library [73, 74] in C++. 
Chapter 5

Results, Verification and Discussion

This chapter presents the results obtained by application of the high-order finite-volume method to the Favre-filtered equations governing LES of premixed flames using algebraic FSD model. Firstly, the usage of high-order CENO scheme is verified conducting experiments similar to those by Rashad [27]. It is duly noted that the computer codes for CENO scheme developed by Rashad are different to what are used for this work. Therefore, assessment of the validity of these codes are essential. CENO is validated for reconstruction of both smooth and discontinuous functions. Thereafter, application of CENO scheme to the Euler equations in three dimensions are also verified.

CENO scheme is then applied to the LES of premixed flames using the FSD model. High-order LES is performed for both non-reacting and reacting flows. Within the scope of this work, these computations are restricted to Cartesian meshes. Moreover, flame surface density is evaluated using the algebraic model developed by Boger et al. The kinetic energy one-equation model is used for the determination of subfilter stresses. Unless otherwise stated, throughout this chapter the Barth-Jespersen slope limiter is used. The smoothness indicator uses a two-layer stencil and has a cut-off value of $S_c = 4000$.

5.1 Verification of the CENO scheme

In CFD, after a model is proposed and developed, it should be verified and validated. Verification involves assessment of the model-implementation based on how well it fits and represents the conceptual insights put in by the developer. Validation process includes the determination of the degree to which the model accurately depicts the real world [75]. Verification and
validation of the CENO scheme in two-dimensions have been extensively performed by Ivan and Groth [23, 54, 24, 25]. In three-dimensional Cartesian grids, Rashad [27] has verified and validated CENO. Since this work implements CENO with a different version of computer codes, it is necessary to verify that the expected accuracy of the scheme is achieved. This verification process serves the purpose of validating the use of CENO scheme on the equations governing LES.

### 5.1.1 Demonstration of High-Order Accuracy

The high-order accuracy of the CENO scheme can be validated by performing reconstructions of three-dimensional functions in a solution domain using various mesh sizes. The reconstructed functions are then compared to the exact functions to obtain their errors for respective mesh sizes. In particular, the second constraint for $k$-exact reconstruction provided by Barth [64], given by Eq. (4.7), dictates that numerical error is proportional to the mesh spacing, $\Delta_{x,y,z}$, such that $Error \propto (\Delta_{x,y,z})^{k+1}$. Recall that for hyperbolic systems, $(k + 1)$ denotes the order of the scheme. Taking the logarithm of the both sides of this expression, one obtains,

$$\log(\text{Error}) \propto (k + 1) \log(\Delta_{x,y,z}).$$

(5.1)

According to this equation, errors and their corresponding mesh sizes have a linear log-log relationship, with a slope corresponding to the order of the scheme. Note that this linear relationship can be obtained only when an asymptotic limit for the slope has been reached. Additionally, it is worthwhile to realize that the order of accuracy only indicates the rate at which the magnitude of error decreases when the mesh size increases – it does not signify the magnitude of the error itself.

As mentioned earlier, $k$-exact reconstruction makes use of cell-averaged values. To perform an arbitrary function reconstruction in a cell, it is important to first compute cell-averaged values of this function in each cell of its reconstruction stencil. The average value for a given function, $U_{\text{exact}}(x, y, z)$, in cell $ijk$ is given by

$$U_{ijk} = \frac{1}{V_{ijk}} \iiint_{V_{ijk}} U_{\text{exact}}(x, y, z) dV.$$  

(5.2)

The RHS of Eq. (5.2) is numerically computed with a desired level of accuracy using the Gauss-Legendre integration procedure discussed earlier.

Once the cell-averaged values of all the cells in the domain are computed, their CENO reconstructions are performed. The solution obtained, $U_{ijk}^k(x, y, z)$, are then compared to the exact
solution to get their errors. In this work, $L_1$ and $L_2$ error norms are used to evaluate the numerical errors. They are defined as,

\[
L_1 = |E|_1 = \frac{1}{V_T} \sum_{ijk} \iint_{V_{ijk}} \left| U_{ijk}^k(x, y, z) - U_{\text{exact}}(x, y, z) \right| dV,
\]

\[
L_2 = |E|_2 = \left\{ \frac{1}{V_T} \sum_{ijk} \iint_{V_{ijk}} \left[ U_{ijk}^k(x, y, z) - U_{\text{exact}}(x, y, z) \right]^2 dV \right\}^{1/2},
\]

where $V_T$ denotes the total volume of the solution domain and the integration is performed over all cells $ijk$ in the solution domain. The integrals in these equations should be numerically computed with a higher order of accuracy compared to the order of the scheme.

Theoretically, a $k$-th order CENO reconstruction should recover a $k$-th order function exactly, irrespective of the mesh size used. This fact is verified by the usage of various high-order functions and their corresponding CENO reconstructions. For example, a fourth-order function, $f(x, y, z) = 2x^3 + 3y^2z^2 - 20z^4$, is reconstructed on a $(5 \times 5 \times 5)$ solution domain with fourth-order ($k = 4$) reconstruction. Regardless of the grid sizes used, the $L_1$ and $L_2$ errors always remained in the order of machine zero ($10^{-12}$). This verifies the correct implementation of the $k$-exact reconstruction. The following subsections further analyze CENO reconstruction in smooth and non-smooth functions. The convergence studies of the computed errors are performed.

**Smooth Function Reconstruction**

A radial cosine function is used as a smooth function to be reconstructed on a $(10 \times 10 \times 10)$ solution domain. It is defined as,

\[
U_{\text{exact}} = 1 + \frac{1}{3} \cos(r),
\]

where $r = \sqrt{(x + 5)^2 + (y + 5)^2 + (z + 5)^2}$ is the radial position. This radial position is shifted according to the symmetry of the solution domain, such that it stays centered at $x = y = z = 0$.

Figure 5.1 displays contour plots of the zeroth-order and the fourth-order reconstruction of the radial cosine function along with its exact solution. Note that the reconstruction is performed in a domain consisting of one block with $(30 \times 30 \times 30)$ cells. Clearly, the resolution of the zeroth-order reconstruction demonstrated in Figure 5.1(b) is poor, due to piecewise constant reconstructions in each cell. The increased resolution in case of the fourth-order reconstruction depicted in Figure 5.1(c) very closely matches the exact function. The log-log charts of $L_1$ and $L_2$ error norms of $k = 1$, $k = 2$, $k = 3$ and $k = 4$ reconstructions plotted against the varying mesh sizes are depicted in Figure 5.2. This figure communicates various aspects of the reconstruction. Firstly, it depicts that a piecewise-quartic ($k = 4$) reconstruction produces results...
with minimal magnitudes of errors compared to other reconstructions. Secondly, the linear trends for each of the plot can be observed, as expected. According to Eq. (5.1), provided that their asymptotic limit is reached, the slopes of these lines should correspond to their respective orders of the method, \((k+1)\). Table 5.1 displays the slopes of each of these lines computed using the final three points in the charts. These slopes closely match the expected results, thereby verifying that the expected order of accuracy is achieved by using the corresponding order of reconstruction in a hyperbolic system.

Figure 5.1: Reconstruction of radial cosine function on a \((30 \times 30 \times 30)\) grid.

(a) Exact Solution.

(b) Zeroth-order CENO \((k = 0)\) reconstruction.

(c) Fourth-order CENO \((k = 4)\) reconstruction.
Chapter 5. Results, Verification and Discussion

54

(a) $|E_1|$ convergence study.

(b) $|E_2|$ convergence study.

Figure 5.2: Convergence studies for radial cosine function reconstruction.

| \( k \) | \( |E_1| \) | \( |E_2| \) |
|------|------|------|
| 0    | -0.999 | -1.000 |
| 1    | -2.064 | -2.105 |
| 2    | -3.041 | -3.163 |
| 3    | -4.069 | -4.038 |
| 4    | -4.990 | -4.989 |

Table 5.1: Orders of accuracy for the reconstruction of radial cosine function.

Discontinuous Function Reconstruction

After investigating reconstruction of a smooth function, where the smoothness indicators are mostly inactive, it is imperative to assess how CENO reconstruction behaves for discontinuous functions. For this purpose, the Abgrall function [76] is used. By definition, this function has many discontinuities, and therefore its high-order monotonous reconstruction is not possible. In the vicinity of discontinuities, CENO reconstruction has to switch down to a limited piecewise linear reconstruction. The robustness of the CENO scheme to switch between the higher and lower order reconstruction can be demonstrated by the use of this function.

The Abgrall function is defined in two dimensions as,

\[
U_{\text{exact}}(x, y) = \begin{cases} 
  f \left( x - y \cot \sqrt{\frac{\pi}{2}} \right) & x \leq \frac{\cos \pi y}{2}, \\
  f \left( x + y \cot \sqrt{\frac{\pi}{2}} \right) + \cos(2\pi y) & x > \frac{\cos \pi y}{2},
\end{cases}
\]

(5.6)
with

\[ f(r) = \begin{cases} 
- r \sin \left( \frac{3\pi}{2} r^2 \right) & r \leq -\frac{1}{3}, \\
| \sin(2\pi r) | & |r| < \frac{1}{3}, \\
2r - 1 + \frac{1}{6} \sin(3\pi r) & r \geq \frac{1}{3}, 
\end{cases} \] (5.7)

Following the implementation of Rashad [27], this two-dimensional function is constantly extruded in the third direction. In doing so, any \( xy \) plane in the \( z \)-axis corresponds to the Abgrall function. Using a domain of \((2 \times 2 \times 2)\), three-dimensional reconstruction of this function is performed using CENO. Since this function is independent of \( z \)-axis, the error caused due to reconstruction is also independent of the grid size used in the \( z \)-direction.

Figure 5.3(a) shows the discontinuities in a two-dimensional Abgrall function. This function is then reconstructed on a \((2 \times 2 \times 2)\) solution domain with various mesh sizes. In particular, the grid size in \( z \)-direction is set to be 6, and the grid sizes in the \( x \)- and \( y \)-axes are varied. Figures 5.3(b) and 5.3(c) display the zeroth- and the fourth-order CENO reconstructions of this function, using a grid size of \((180 \times 180 \times 6)\). Figure 5.3(d) shows the regions where the smoothness indicators are switched on during the fourth-order reconstruction. It can be clearly seen that the smoothness indicators are activated in the vicinity of discontinuities. At these cells, the \( k = 4 \) reconstruction is reverted down to a limited \( k = 1 \) reconstruction. Due to the use of slope limiters, this linear reconstruction essentially switches down to monotonicity preserving piecewise constant \((k = 0)\) reconstruction near discontinuities. Additionally, in the smooth regions in Figures 5.3(b) and 5.3(c), the resolution of the solutions are quite dissimilar. As in the case with the radial cosine function, a \( k = 4 \) reconstruction provides higher resolved solutions in these areas. This demonstrates the use of high-order reconstruction in the smooth regions of the solution domain.

To assess the order of accuracy of this discontinuous-reconstruction, convergence studies of the errors can be performed using a similar approach as earlier. The convergence of \( L_1 \) norms for the errors of a piecewise constant \((k = 0)\) and a piecewise quartic \((k = 4)\) reconstruction are displayed in Figure 5.4. Again, there are many facts communicated by this figure. Firstly, it can be noticed that the slopes of both the lines look qualitatively similar. Using the last three points (which have reached the asymptotic limit), the slopes of these lines are computed to be as follows:

| \( k \) | \( |E|_1 \) |
|---|---|
| 0 | -0.984 |
| 4 | -1.085 |
Chapter 5. Results, Verification and Discussion

(a) Exact Solution.  (b) Zeroth-order CENO (k = 0) reconstruction.

(c) Fourth-order CENO (k = 4) reconstruction.  (d) Smoothness indicators for k = 4 reconstruction: red regions depict the non-smooth area where smoothness indicators are switched on.

Figure 5.3: Reconstruction of the Abgrall function on a (180 × 180 × 6) grid.

Therefore, the data suggest that both the CENO schemes produce first-order spatially accurate schemes. This is due to the hybrid switching occurring in CENO schemes. The global order of accuracy of a high-order hybrid scheme is governed by the lowest order of the reconstruction used in the scheme [27, 25]. Since piecewise constant reconstruction is applied to the discontinuous regions, any high-order CENO schemes would give first-order spatial accuracy for discontinuous reconstruction. However, another key feature that can be observed in Figure 5.4 is that the magnitude of the error goes down when a high-order method is used. Even if the slopes of the
Figure 5.4: $|E|_1$ convergence study for the discontinuous Abgrall function.

lines stay the same, the plot for $k = 4$ reconstruction is shifted downwards compared to the plot for $k = 0$ reconstruction. This shift depicts that higher-order methods are more accurate compared to lower order methods. Rashad [27] has shown that when a single layer stencil is used for computing smoothness indicators, a piecewise cubic method is more accurate than a piecewise quadratic method. In general, it is noted that higher-order CENO reconstructions produce more accurate results compared to the lower-order reconstructions. This can be further verified by Figure 5.4 where, to obtain solutions with $L_1$ error of 0.0085, usage of a piecewise constant reconstruction requires $(373 \times 373 \times 6)$ cells, whereas piecewise quartic reconstruction requires only $(121 \times 121 \times 6)$ cells. The number of cells required decreases from 834,774 to 87,846, by 89.5%. This is quite a significant decrease, which speaks a lot about the reduced computational effort due to the usage of the high-order CENO scheme.

5.1.2 Application to the Euler Equations

After the verification of CENO reconstruction in smooth and non-smooth functions, it is instructive to apply this reconstruction in real flow problems. With an aim to judge the accuracy of the computation of high-order inviscid fluxes, CENO schemes are first applied to the hyperbolic Euler equations. The semi-discrete form of these Euler equations are presented in Eq. (3.24). In the following examples depicting solutions of Euler equations, the approximate Riemann solver developed by Roe [50] is used for inviscid flux computations.
Periodic Density Wave

To validate the use of high-order CENO scheme on hyperbolic equations, Euler equations are used to march a periodic density wave with time. The initial conditions of this wave is defined as

\[
\begin{align*}
\rho &= \left[ 1.0 + \frac{1}{2} \sin(5\pi x) \cos(\pi x) \right] \text{ kg/m}^3, \\
u &= 100 \text{ m/s}, \\
v &= w &= 0 \text{ m/s}, \\
p &= 1 \text{ atm} = 101.325 \text{ kPa}.
\end{align*}
\]

Here, the density is defined by a smooth one-dimensional periodic function, which is propagated in \(x\)-direction with a velocity of \(u = 100 \text{ m/s}\). When marched in a three-dimensional (2 \(\times\) 2 \(\times\) 2) m solution domain with periodic boundary conditions, this function completes a full period in 20 ms. Figure 5.5 compares the initial density distribution with the distribution obtained after 20 ms by solving Euler equations, using high-order CENO.

In particular, Figure 5.5(b) depicts the solution of Euler equations using a fourth-order CENO method \((k = 3\) reconstruction). The grid is discretized to a mesh of \((400 \times 6 \times 6)\) cells, and a CFL number of 0.8 is used for the time marching procedure. It can be observed that this distribution matches the expected solution presented in Figure 5.5(a) very well. Quantitative analysis of the results can be performed by computing the \(L_1\) and \(L_2\) error norms.

Table 5.2 compares the \(L_1\) and \(L_2\) error norms for the solutions computed using limited second-order method and fourth-order CENO scheme. Euler equations are solved using various grid sizes in a single processor with both second-order and CENO schemes. The time required for each implementation are also noted. It can be observed that for a given number of cells, the fourth-order CENO scheme takes much longer to implement compared to the limited second-order method. However, for a specific grid size, the magnitude of the errors also significantly reduces due to the high-order scheme. To obtain errors comparable to the ones generated by the fourth-order CENO using 7,200 cells, a second-order scheme has to use 90,000 cells. Table 5.2 depicts how this increases the computational time for the second-order method, thereby implying the increase in computational cost. These results demonstrate a paramount advantage of using high-order schemes, which is one of the motivations for this work: high-order methods generate better accurate results with a comparatively lower computational cost.
Chapter 5. Results, Verification and Discussion

(a) The exact density distribution.

(b) Fourth-order CENO scheme \((k = 3)\) after 20 ms.

Figure 5.5: Plots of the periodic density wave.

<table>
<thead>
<tr>
<th># Cells</th>
<th>(\mathcal{O}(\Delta x^2))</th>
<th>(\mathcal{O}(\Delta x^4))</th>
</tr>
</thead>
<tbody>
<tr>
<td>7,200 ((200 \times 6 \times 6))</td>
<td>(L_1: 1.74 \cdot 10^{-2})</td>
<td>(L_1: 2.30 \cdot 10^{-4})</td>
</tr>
<tr>
<td></td>
<td>(L_2: 2.06 \cdot 10^{-2})</td>
<td>(L_2: 2.56 \cdot 10^{-4})</td>
</tr>
<tr>
<td></td>
<td>Time: 00:13:09</td>
<td>Time: \textbf{20:22:50}</td>
</tr>
<tr>
<td>14,400 ((400 \times 6 \times 6))</td>
<td>(L_1: 6.35 \cdot 10^{-3})</td>
<td>(L_1: 1.44 \cdot 10^{-5})</td>
</tr>
<tr>
<td></td>
<td>(L_2: 7.35 \cdot 10^{-3})</td>
<td>(L_1: 1.60 \cdot 10^{-5})</td>
</tr>
<tr>
<td></td>
<td>Time: 00:59:21</td>
<td>Time: 46:05:13</td>
</tr>
<tr>
<td>90,000 ((2500 \times 6 \times 6))</td>
<td>(L_1: 2.20 \cdot 10^{-4})</td>
<td>(L_1: -)</td>
</tr>
<tr>
<td></td>
<td>(L_2: 3.44 \cdot 10^{-4})</td>
<td>(L_2: -)</td>
</tr>
<tr>
<td></td>
<td>Time: \textbf{36:03:40}</td>
<td>Time: -</td>
</tr>
</tbody>
</table>

Table 5.2: Comparison of the limited second-order scheme and the fourth-order CENO scheme.

3D Shock-Box Problem

For completeness, the one-dimensional shock-tube problem presented in Section 3.6 is extended to three dimensions here. Using the same so-called “left” and “right” conditions \((\rho_{\text{left}} = 4.696\)
kg/m³, $v_{\text{left}} = 0$ m/s, $p_{\text{left}} = 404.4$ kPa, $\rho_{\text{right}} = 1.408$ kg/m³, $v_{\text{right}} = 0$ m/s, and $p_{\text{right}} = 101.1$ kPa), the initial values of a shock-box problem is defined as

$$\mathbf{W}(\vec{x}, 0) = \begin{pmatrix} \rho(\vec{x}, 0) \\ u(\vec{x}, 0) \\ v(\vec{x}, 0) \\ w(\vec{x}, 0) \\ p(\vec{x}, 0) \end{pmatrix} = \begin{cases} \mathbf{W}_{\text{left}}, & x, y, z \leq 0 \text{ m}, \\ \mathbf{W}_{\text{right}}, & x, y, z > 0 \text{ m}. \end{cases}$$  (5.9)

The initial density distribution in the (10 × 10 × 10) m solution domain is depicted in Figure 5.6(a). This shock-box problem is then solved using first-order Godunov’s method and CENO high-order scheme, for a time of 6 ms, using a CFL number of 0.5. Figure 5.6(b) depicts the density distribution in the shock-box after 6 ms obtained using the Godunov’s method in a (64 × 64 × 64) grid. Note that multiple blocks are employed for this problem. Specifically, there are 8 blocks used for this calculation, two blocks in each direction and (32 × 32 × 32) cells in each block. Figure 5.6(c) demonstrates the usage of CENO scheme for the same. Expectedly, an increase in the resolution of the solution can be observed when high-order CENO scheme is used. Since this shock-box case is a more complex flow compared to the previous case of periodic density distribution, this numerical experiment verifies the advantages of using CENO schemes on more general engineering flows. Moreover, Figure 5.6(d) displays the activation of smoothness indicators near the regions of discontinuities in density distribution for the last time step. The smoothness indicators are turned on in all the areas marked with red. At these regions, the $k = 3$ reconstruction switches down to the limited $k = 1$ reconstruction. A qualitative analysis of this figure compared to the results obtained with the fourth-order CENO scheme (Figure 5.6(c)) can be enough to verify the correct usage of smoothness indicators. Indeed, smoothness indicators act near the vicinity of discontinuities.

This concludes the verification and validation of CENO schemes on hyperbolic systems. The following sections deal with equations governing LES, i.e. elliptic systems.

### 5.2 Application of CENO Scheme to LES of Turbulent Premixed Flames

After the verification of CENO scheme, it is finally applied to the semi-discrete form of Favre-filtered Navier-Stokes equations, given by Eq. (3.13). These are the equations that govern LES using FSD model. Note that these equations make use of viscous (elliptic) fluxes, too. These
(a) Initial density distribution (for \( t = 0 \)).

(b) First-order scheme (\( k = 0 \)) for \( t = 6 \) ms.

(c) Fourth-order CENO scheme (\( k = 3 \)) for \( t = 6 \) ms.

(d) Smoothness indicators for the fourth-order scheme (at \( t = 6 \) ms): red regions depict the non-smooth area where smoothness indicators are switched on.

Figure 5.6: Density variations in a shock-box problem.
viscous fluxes are evaluated with high-order accuracy by the approach outlined in Section 4.4.2. In the following sections, there are a few examples of second-order accurate results, too. These results are obtained by using a limited piecewise linear reconstruction for inviscid fluxes and a hybrid average gradient-diamond-path approach [53] for viscous fluxes. These schemes are henceforth referred to as “second-order LPL” schemes.

The following sections present two distinct flow problems utilizing high-order LES. One of these problems involves a turbulent non-reacting flow, in which the decay of a homogeneous isotropic turbulence is studied. The second problem deals with a turbulent premixed flame inside a box. In this case, the flame-propagation and the behaviour of the flame is observed. It should be noted that in both these cases, AUSM\(^{+}\)-up flux function is employed for the inviscid flux computations. Moreover, the turbulent velocity field is initialized by using Rogallo’s [77] procedure along with the initial turbulent kinetic energy spectrum proposed by Haworth and Poinset [78]. A three-dimensional velocity field corresponding to such an initial turbulence in a box is displayed in Figure 5.7. A low CFL number of 0.1 facilitates that the stability criteria of the method is fulfilled. In both the cases, solution domains are decomposed to multiple blocks, and parallel implementation of the codes have been conducted.
5.2.1 LES of Non-Reacting Flows

The decay of a three-dimensional homogeneous isotropic turbulence is considered to be the starting point of validation of high-order methods for solving equations governing LES. In doing so, it should be noted that the rate of decay of the isotropic turbulence is proportional to $t^{-1.25}$, according to experiments and DNS results [79]. This criteria should be recovered from LES results as well.

In a $(2\pi \times 2\pi \times 2\pi)$ m solution domain, an isotropically decaying turbulence is introduced. The boundaries of this solution domain is assumed to be periodic everywhere. The initial turbulence related parameters are summarized in Table 5.3. The definition of each of the symbols are consistent with the expressions provided earlier, where $\Lambda$ denotes integral length scale, $\lambda$ is the Taylor micro scale, $\eta$ stands for the Kolmogorov length scale, $u'$ is the RMS velocity fluctuation, and $k$ is the total turbulent kinetic energy. In this work, this turbulence decay is evaluated using two grid sizes comprising of $(32 \times 32 \times 32)$ cells and $(64 \times 64 \times 64)$ cells. At these grid sizes, the results obtained using second-order LPL scheme are compared to those obtained using second, third and fourth order CENO schemes. This turbulent flow is simulated in these grids for a duration of 100 ms.

<table>
<thead>
<tr>
<th>$\Lambda$</th>
<th>$\lambda$</th>
<th>$\eta$</th>
<th>$u'$</th>
<th>$k$</th>
</tr>
</thead>
<tbody>
<tr>
<td>mm</td>
<td>mm</td>
<td>mm</td>
<td>m/s</td>
<td>m$^2$/s$^2$</td>
</tr>
<tr>
<td>1.9625</td>
<td>0.85</td>
<td>0.00845</td>
<td>100</td>
<td>15,000</td>
</tr>
</tbody>
</table>

Table 5.3: Turbulence parameters for decaying isotropic turbulence in a box.

Figure 5.8 depicts the decay of total turbulent kinetic energy on the two grid sizes, using a log-log scale. In each case, it can be observed that initially the turbulent kinetic energy doesn’t change much, and decreases only very slightly. After approximately the time of 10 ms, the decay rate increases significantly. This decay seems to have a constant slope in the log-log scale. The decay criteria mentioned earlier dictates that this slope should be $-1.25$. For qualitative analysis, a line with this slope is also displayed in the figures. It can be easily observed that a $(64 \times 64 \times 64)$ grid (Figure 5.8(b)) provides much accurate results compared to the $(32 \times 32 \times 32)$ grid (Figure 5.8(a)).

Another important observation that one can make from Figure 5.8(a) is that irrespective of the order of CENO being used, the results show a trend demonstrated by a second-order CENO scheme (or a second-order LPL scheme). Note that this is not entirely the case in Figure 5.8(b). Here, a higher-order scheme produces slightly less dissipative solutions, with a slope closer to the target slope of $-1.25$. This is due to the action of smoothness indicators. In the coarser
Chapter 5. Results, Verification and Discussion

(a) A $32 \times 32 \times 32$ grid.

(b) A $64 \times 64 \times 64$ grid.

Figure 5.8: Decay of total turbulent kinetic energy computed using limited CENO scheme on the given grid size.

mesh, most of the cells are unresolved, and are flagged by smoothness indicators, as depicted in Figure 5.9(a). In these cells, the scheme is reverted down to the limited piecewise reconstruction. And therefore, even a higher-order CENO scheme gives results corresponding to limited first-order schemes (with $k = 1$ reconstruction). On the other hand, when the mesh is made finer to have $(64 \times 64 \times 64)$ cells, the smoothness indicators only act in a few cells (Figure 5.8(b)), as the solutions are more resolved. Therefore, in this case, $k$-exact reconstruction still dominates giving better accurate results.

The fact that high-order solutions in this elliptic system are influenced by the smoothness indicators in the coarser grid can be verified by conducting CENO without the use of smoothness indicators. The same flow problem is solved using fourth-order unlimited CENO scheme to determine its accuracy. Figure 5.10 compares the unlimited second-order LPL scheme with unlimited fourth-order CENO scheme for the two grid sizes used. Clearly, in both the coarse and fine grids, the high-order unlimited CENO scheme is less dissipative than the second-order LPL scheme.

These results verify the usage of three-dimensional CENO scheme on elliptic systems. This establishes a premise on which CENO scheme can be applied to the equations governing LES of turbulence premixed flames. This application is discussed in the following section.
Figure 5.9: Smoothness Indicators acting (on $\rho$) at the last time step before 100 ms: red regions depict the non-smooth areas where smoothness indicators are switched on.

Figure 5.10: Comparison of decay of total turbulent kinetic energy computed using unlimited second-order LPL scheme and unlimited fourth-order CENO scheme on the mesh sizes of (32 × 32 × 32) and (64 × 64 × 64) cells.
5.2.2 LES of Turbulent Premixed Combustion

Following the verification of the usage of high-order CENO schemes on LES of non-reacting flows, it is intuitive to apply CENO on LES of reacting flows. It is duly noted that even while testing the earlier flow problem corresponding to the non-reactive case, the equations governing LES of reacting flows (using the FSD model) were employed. Therefore, it is straightforward to use the same high-order implementation for this flow problem, involving reacting flows.

Specifically, the flow problem involves a freely propagating premixed flame in an isotropically decaying turbulence. The flame is assumed to propagate with a one-step combustion of methane and air, which involves a total of five species: CH$_4$, O$_2$, N$_2$, CO$_2$ and H$_2$O. This flame is initialized by introducing a planar laminar premixed flame in a turbulent velocity field, initialized using Rogallo's procedure [77] as earlier. A cubic solution domain of ($0.01 \times 0.01 \times 0.01$) m is represented by a computational domain of ($64 \times 64 \times 64$) cells (such that a fourth-order limited CENO scheme gives higher accuracy compared to the second-order LPL scheme, as seen earlier in the case of isotropic turbulence decay for this grid size). Subsonic inflow and outflow boundary conditions are applied at the boundaries of the domain perpendicular to the laminar flow direction. The boundaries of the rest of the domain are assumed to be periodic. The schematic of this computational domain is displayed in Figure 5.11. The parameters characterizing the initial turbulence are provided in Table 5.4, where $\phi$ is the equivalence ratio of the combustion, $\delta_L$ is the laminar flame thickness, and $s_L$ is the laminar flame speed. Note that these parameters are identical to those used for a similar case investigated by Lin [15].

<table>
<thead>
<tr>
<th>$\phi$</th>
<th>$\Lambda$</th>
<th>$\lambda$</th>
<th>$\eta$</th>
<th>$u'$</th>
<th>$k$</th>
<th>$s_L$</th>
<th>$\delta_L$</th>
<th>$u'/s_L$</th>
<th>$\Lambda/\delta_L$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mm</td>
<td>mm</td>
<td>mm</td>
<td>m/s</td>
<td>m$^2$/s$^2$</td>
<td>m/s</td>
<td>mm</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>1.790</td>
<td>0.460</td>
<td>0.029</td>
<td>2.92</td>
<td>12.8</td>
<td>0.403</td>
<td>0.050</td>
<td>7.25</td>
<td>35.8</td>
</tr>
</tbody>
</table>

Table 5.4: Turbulence parameters for the premixed flame in a turbulent box.
It should be noted that the computational domain is decomposed to several blocks for a faster multi-processor parallel implementation. In particular, the domain in each direction is decomposed to 8 blocks, and each block contains 8 cells in a specific direction. Therefore, the domain consists of the total of 512 blocks, each containing 512 cells. This corresponds to the total of 262,144 cells, with the grid spacing of \( \Delta x = \Delta y = \Delta z = 0.156 \text{ mm} \). Recall that the FGR of 2 is used in this work, which suggests that the filter width is \( \Delta = 0.312 \text{ mm} \), which is much larger than the laminar flame thickness, \( \delta_L \). This fact was mentioned earlier in Section 2.5, where the challenges that this poses on reaction rate modelling were discussed.

Figure 5.12: Fourth-order instantaneous progress variable distribution in a \( xz \)-plane of the three-dimensional flame at various time.
Chapter 5. Results, Verification and Discussion

Figure 5.13: Predicted instantaneous flame surfaces defined as the iso-surfaces of $\tilde{c}^* = 0.5$ at $t = 0.25$ ms.

Figure 5.12 depicts two-dimensional $xz$ slices of the instantaneous progress variables at various times computed using fourth-order CENO scheme. Figure 5.12(a) demonstrates the initial planar laminar premixed flame introduced in a isotropically decaying turbulence. It can be clearly seen that as the time progresses, the turbulence affects the flame and the flame starts wrinkling. The wrinkling becomes more and more pronounced with the increase in time. Moreover, as time progresses, the amount of products increases compared to the reactants. Note that a progress variable of zero (blue regions in Figure 5.12) represents the unburnt reactants. One can clearly observe that with the increment in time, area of the blue region decreases. These characteristics suggest that the implementation of fourth-order CENO schemes has provided expected wrinkling of the flame. The three-dimensional wrinkling of the flame is demonstrated in Figure 5.13.

Figure 5.13 displays the three-dimensional flame surfaces predicted by using the second-order LPL method and the fourth-order CENO scheme at the time of 0.25 ms. These flame surfaces are identified by the isosurface of the filtered progress variable, at its critical value $\tilde{c}^* = 0.5$. Figure 5.13(a) depicts the flame surface obtained using fourth-order CENO scheme. It can be observed that this flame looks very similar to the second-order flame given in Figure 5.13(b). This similarity shows that the expected flame prediction is achieved by the use of high-order CENO method. However, since there is no apparent difference in the flame surfaces between the low-order and high-order scheme, the usage of high-order method for LES of premixed flame inside a turbulent box seems questionable. To judge the usage of high-order CENO schemes
(a) Fourth-order CENO.  
(b) Smoothness Indicators: red regions depict the non-smooth areas where the smoothness indicators are turned on.

Figure 5.14: Depiction of the action of smoothness indicators on the progress variable, $c$, at the last time step before $t = 0.25$ ms

to predict the flame surfaces, it is natural to seek the understanding of the regions where the smoothness indicators are switched on.

A two-dimensional $xz$ slice of the instantaneous progress variable distribution at $t = 0.25$ ms is displayed again in Figure 5.14 along with the depiction of regions where the smoothness indicators are switched on. From these figures it is clear that in the vicinity of the flame, the scheme switches down to a first-order scheme. This can obviously be attributed to the relatively coarser mesh size that is being used in this simulation ($64 \times 64 \times 64$). The use of a much higher mesh size with CENO would then theoretically give a much accurate solution, since the grid size resolves the flame much better. However, this severely increases the computational cost for the usage of a high-order method. In this case, one can argue on the validity of the motivation to use high-order methods. A more efficient formulation could then be the usage of adaptive mesh refinement (AMR) with CENO. An AMR approach can be utilized to obtain increased mesh size only near the flames. This might then make CENO scheme more accurate for the flame prediction. The usage of AMR with CENO for LES of turbulent premixed flames is suggested as a topic for further research.
Chapter 6

Concluding Remarks

6.1 Conclusions

This study is an attempt towards achieving reactive flow solutions having higher accuracy with reduced computational cost. In particular, this work applies the high-order CENO finite-volume schemes to the equations governing LES of reacting flames. It should be realized that CENO scheme is still in the development phase and the potential of this high-order scheme is yet to be explored and/or fully realized. In three-dimensional Cartesian mesh, this work has demonstrated the successful implementation of CENO schemes on LES equations.

Firstly, high-order CENO scheme in three dimensions has been verified and validated. CENO reconstructions of smooth as well as discontinuous functions have given the expected results and demonstrated the corresponding order of the scheme correctly. Moreover, the smoothness indicators have shown their robustness in acting in non-smooth regions. The effect of smoothness indicators on the global order of the method has also been observed.

Secondly, CENO scheme has also been applied to Euler equations, and the results obtained have been compared to the exact solutions. These experiments have substantiated the argument that high-order schemes produce better accurate results with reduced computational costs.

Finally, CENO scheme has been applied to the equations governing LES. Two distinct flow problems, involving non-reacting and reacting flows are assessed. The case with a non-reacting flow depicted an isotropic turbulence decay, and demonstrated that high-order schemes pro-
duced less dissipative decay compared to the corresponding lower-order schemes. A case with a premixed flame in a turbulent box has been used to demonstrate the use of high-order scheme on equations governing LES of reacting flows. The results depicted the correct wrinkling behaviour of the flame. The smoothness indicators seemed to act expectedly near the discontinuities.

This work has achieved its primary objectives that was listed in Section 1.2 of Chapter 1. It should be noted that since there are no “exact solutions” or experimental results for the premixed flame in a turbulent box to access solution quality, the fourth-order solutions have been simply compared to the second-order solutions. The extension of CENO to body-fitted hexahedral meshes or to unstructured meshes would allow performing LES for flames that exists in experiments, like the Bunsen-type flames. Since the experimental results for these flames exist, comparison of high-order methods and lower-order methods to predict the real world would be possible. However, this extension is beyond the scope of this work.

6.2 Recommendations for Future Work

With this work as a stepping stone, many corresponding research work can be conducted. Some of the author’s recommendations for future research on the application of high-order CENO schemes on LES are outlined below:

- Application of high-order schemes on body-fitted hexahedral meshes, hybrid meshes and unstructured meshes should be definitely considered. These meshes have the capability of discretizing real world combustors, which employ more complicated burner configurations. Assessment of high-order accuracy can be better judged using these flow problems.

- This research makes use of the algebraic equation developed by Boger et al. for computing FSD. This expression assumes local equilibrium for FSD. To account for non-equilibrium effects, the transport equation for FSD proposed by Hawkes and Cant should be used. High-order implementation of this equation should also be investigated.

- This work makes use of implicit filtering procedure. It should be interesting to investigate the use of high-order explicit filters.

- The high-order treatment of additional kinds of boundary conditions needs to be investigated. This work makes use of ghost cells, and the boundary conditions are mostly periodic or inflow-outflow (which are dictated by the solutions inside the domain). Constrained boundary conditions for curved or straight boundaries should also be extended to higher order.
• High-order LES framework can be greatly benefited by using CENO scheme with adaptive mesh refinement (AMR) in three dimensions. This combination would significantly reduce the computational time and the memory usage required by the scheme.

• High-order framework for LES can be developed for equations employing SFS models other than the flame surface density. In particular, high-order treatment of LES equations using the thickened flame model [11] or the G-equation model [4] should be interesting to investigate.
References


References


[23] Ivan, L. and Groth, C. P. T., “High-Order Central ENO Scheme for Hyperbolic Conservation Laws on Body-Fitted Multi-Block Mesh,” *Proceedings of the 14th Annual Conference*


