DEVELOPMENT OF A HIGH-ORDER FINITE-VOLUME METHOD FOR THE NAVIER-STOKES EQUATIONS IN THREE DIMENSIONS

by

Ramy Rashad

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

Copyright © 2009 by Ramy Rashad
Abstract

Development of a High-Order Finite-Volume Method for the Navier-Stokes Equations in Three Dimensions

Ramy Rashad
Masters of Applied Science
Graduate Department of Aerospace Engineering
University of Toronto
2009

The continued research and development of high-order methods in Computational Fluid Dynamics (CFD) is primarily motivated by their potential to significantly reduce the computational cost and memory usage required to obtain a solution to a desired level of accuracy. In this work, a high-order Central Essentially Non-Oscillatory (CENO) finite-volume scheme is developed for the Euler and Navier-Stokes equations in three dimensions. The proposed CENO scheme is based on a hybrid solution reconstruction procedure using a fixed central stencil. A solution smoothness indicator facilitates the hybrid switching between a high-order $k$-exact reconstruction technique, and a monotonicity preserving limited piecewise linear reconstruction algorithm. The resulting scheme is applied to the compressible forms of the Euler and Navier-Stokes equations in three dimensions. The latter of which includes the application of this high-order work to the Large Eddy Simulation (LES) of turbulent non-reacting flows.
Acknowledgements

While carrying out this research I have been very fortunate to work under the supervision of Prof. Clinton P. T. Groth at the University of Toronto Institute for Aerospace Studies. For his excellent guidance, tutelage, motivation, and resolve, I am truly grateful.

I would also like to thank my colleagues in the CFD lab for providing me with invaluable help and support throughout this work. More importantly, I thank them for making this experience truly enjoyable.

An extended thank-you goes out to Lucian Ivan for taking the time to explain his implementation and development of the CENO scheme, and for answering so many of my questions regarding the various facets of our work.

Finally, and most importantly, I’d like to thank my mom, dad, and sister. They have always been the biggest proponents of my endeavours, and for that I am forever grateful.

Toronto, August 2009

Ramy Rashad
Contents

Abstract ii

Acknowledgments iii

Contents iv

List of Tables vii

List of Figures viii

1 Introduction 1
   1.1 Motivation ........................................... 1
   1.2 Objectives ......................................... 2
   1.3 Overview of Thesis ................................ 3

2 Godunov-Type Finite-Volume Methods 4
   2.1 Conservation Equations ................................... 4
       2.1.1 Navier-Stokes Equations in Cartesian Coordinate Frame 5
       2.1.2 Thermodynamic Relations ................................ 6
       2.1.3 Euler Equations in Cartesian Coordinate Frame .......... 6
       2.1.4 Hyperbolic and Elliptic Behaviour ..................... 7
   2.2 Godunov-Type Methods .................................. 7
### 2.3 Flux Evaluation

- **2.3.1 The Riemann Problem**
- **2.3.2 Approximate Riemann Solvers**

### 2.4 Reconstruction Basics

- **2.4.1 The Euler Equations in One Dimension**
- **2.4.2 Godunov's Method and Piecewise Constant Reconstruction**
- **2.4.3 Piecewise Linear Reconstruction**
- **2.4.4 Monotonicity and Godunov's Theorem**
- **2.4.5 Enforcing Monotonicity: Slope Limiters**

### 2.5 Time Marching Schemes

### 3 High-Order CENO Scheme

- **3.1 Previous High-Order Work**
- **3.2 Classification of Schemes**
- **3.3 Overview of the CENO Scheme**
- **3.4 Motivation of the CENO Scheme**
- **3.5 \(k\)-Exact Least-Squares Reconstruction**
  - **3.5.1 Reconstruction Stencils**
  - **3.5.2 Determination of the Derivatives: The Least-Squares Problem**
  - **3.5.3 Forming the Overdetermined System, \(Ax-b=c\)**
  - **3.5.4 Calculating the Geometric Coefficients and Moments**
  - **3.5.5 Solving the Least-Squares Problem**
- **3.6 The Smoothness Indicator**
- **3.7 Flux Evaluation**
  - **3.7.1 Inviscid Flux Evaluation**
  - **3.7.2 Viscous Flux Evaluation**
4 Verification and Numerical Results

4.1 Accuracy Assessment Methods ........................................ 56
4.2 Convergence Studies .................................................. 56
4.3 Demonstration of High-Order Accuracy ................................. 57
  4.3.1 Spherical Cosine Function ......................................... 58
  4.3.2 A Smooth 3D Function ............................................... 61
4.4 Demonstration of Smoothness Indicator ................................. 63
  4.4.1 Abgrall Function .................................................... 64
  4.4.2 Hyperbolic Tangent Function ..................................... 67
4.5 Application to the Euler Equations .................................... 68
  4.5.1 1D Shock-Tube Problem ........................................... 68
  4.5.2 3D Shock-Box Problem ............................................ 71
  4.5.3 Periodic Density Wave ............................................. 72
4.6 Application to the Navier-Stokes Equations ......................... 76
  4.6.1 A Brief Introduction to Turbulence Modelling ................. 77
  4.6.2 Application to Isotropic Turbulence Decay .................... 78

5 Conclusions and Future Research ....................................... 84

5.1 Conclusions .......................................................... 84
5.2 Future Research ...................................................... 85

References .................................................................................. 91
List of Tables

3.2 Number of derivatives (including the cell-centered value) required for $k$-exact reconstruction. .................................................. 34

4.1 Orders of accuracy for the spherical cosine function reconstruction. .................. 60

4.2 Spherical cosine reconstruction: Number of cells flagged by the smoothness indicator as unfit for unlimited $k^{th}$-order reconstruction ($S_c=1,500$). ............. 61

4.3 Orders of accuracy for the smooth 3D function reconstruction. ...................... 62

4.4 Smooth 3D function reconstruction: Number of cells flagged by the smoothness indicator as unfit for unlimited $k^{th}$-order reconstruction ($S_c=1,500$). ............. 64

4.5 Table of results comparing the limited second-order scheme and the fourth-order CENO scheme. .................................................. 76

4.6 Specifics of the model spectrum. .................................................. 79
List of Figures

2.1 Wave patterns for the one-dimensional Riemann problem. .......................... 11
2.2 HLLE flux function diagram in the \( x-t \) plane showing the three-states (left, middle, and right), two waves (\( \lambda^- \) and \( \lambda^+ \)), and the closed integration path, \( S \). ............................ 13
2.3 Depiction of a piecewise constant reconstruction. ........................................ 15
2.4 First-order accurate comparison of flux functions: shock tube problem: \( t= 7 \) ms, 100 cells, CFL = 0.7. ................................................................. 16
2.5 (a) Depiction of piecewise linear reconstruction, (b) backward-biased (blue-line) and forward-biased (red-line) reconstructions, (c) depiction of centered-stencil and errors (red-bars) minimized using the least-squares method, and (d) optimized slope found using least-squares method for the given centered-stencil. ............................. 18
2.6 Depiction of the oscillations that occur in the vicinity of a shock wave for an unlimited second-order FV scheme. ......................................................... 19
2.7 Solution reconstruction using (a) unlimited piecewise linear and (b) limited piecewise linear reconstructions. ................................................................. 20
2.8 Comparison of limiters for the shocktube problem: \( t= 7 \) ms, 200 cells; density distribution. ................................................................. 22
2.9 The clipping of smooth extrema occurring as a result of slope limiters being applied to a smooth maximum (right), compared to the case of a discontinuity (left). ................................................................. 23
3.1 Numbering convention used for the one and two-layer reconstruction stencils. . 35
3.2 Variation of the function \( f(\alpha) = \frac{\alpha}{1-\alpha} \). ........................................ 45
4.17 Comparison of turbulent kinetic energy spectra of isotropic turbulence decay at 100 ms for different orders of spatial reconstruction on a $64 \times 64 \times 64$ grid. 81

4.18 Comparison of turbulent kinetic energy spectra of isotropic turbulence decay at 100 ms for different orders of spatial reconstruction on a $128 \times 128 \times 128$ grid. 82

4.19 Comparison of turbulent kinetic energy spectra of isotropic turbulence decay at 100 ms for second-order LPL scheme and fourth-order CENO scheme on different grid resolutions. 82
Chapter 1

Introduction

Fluid dynamics is the branch of physics and engineering which deals with gases and liquids in motion. The study and development of this subject has relied on three complementary approaches: experimental, theoretical, and computational [1]. These approaches provide an integral framework for advancing our knowledge of fluid dynamics, as well as our ability to apply that knowledge to the engineering design of aerospace technology, the application for which this work is motivated.

The current push for greener aerospace technology requires serious efforts to mitigate the escalating effects of such technology on climate change and natural resources. Both economic and environmental factors are calling for improved, more efficient aircraft and propulsion systems. Meanwhile, the engineers involved in the detailed design of these systems have become heavily reliant on computer assisted design tools such as Computational Fluid Dynamics (CFD); that is, the branch of fluid dynamics which uses numerical methods and algorithms in combination with high-speed computers to predict and study fluid flow. As such, the development of more efficient and robust CFD methods is of immediate consequence and benefit to the advancement of aerodynamic and other related aerospace technology. This project serves as a stepping-stone to these ends, with the primary focus of developing a high-order finite-volume method for the compressible forms of the Euler and Navier-Stokes equations in three dimensions.

1.1 Motivation

The continued research and development of high-order methods in CFD is primarily motivated by their potential to significantly reduce the computational cost and memory usage required
to obtain a solution to a desired level of accuracy. The realization of this potential, in turn, promises more routine calculations of complex fluid flow problems with the goal of accelerating and optimizing the design of future aerospace technology. Furthermore, the growing trend to move toward three-dimensional modeling is in hopes of improving our understanding of natural fluid flow phenomena, such as turbulence, as well as predicting flow over and through complex three-dimensional geometry, such as those encountered in gas-turbine engines.

Over the past two decades, researchers have considered and investigated a variety of high-order methods in CFD. A more thorough discussion of previous high-order research may be found in Section 3.1, examples of which include the Essentially Non-Oscillatory (ENO) and Weighted Essentially Non-Oscillatory (WENO) schemes, Discontinuous Galerkin (DG) schemes, and high-order Finite-Difference (FD) schemes. Despite the consistent and continuous research effort, there is still no consensus on a robust and effective way of providing high-order accuracy without compromising the solution monotonicity \[2\]. Given the plethora of CFD applications, perhaps a consensus is too much to ask. Nevertheless we press on with the aim of developing a high-order method that presents a compromise between the often competing goals of accuracy, computational efficiency, and robustness, in turn arriving at a scheme which is practical and well suited to a variety of exciting aerospace applications.

Indeed, the high-order Central Essentially Non-Oscillatory (CENO) scheme developed in this work has great potential in its applicability, including, but not limited to: gas turbine and rocket propulsion systems (viz. turbulent combusting flows), hypersonic flows, micron-scale flows, and magnetohydrodynamics. In Chapter 4, an example of this high-order work in combination with the Large Eddy Simulation (LES) of turbulent non-reacting flow will be presented. In addition, a natural extension and future objective (outside the scope of this research) will be the combination of the high-order CENO scheme with the LES of turbulent reacting flows involving disparate spatial and temporal scales.

The specific design of the high-order CENO scheme has several advantages that motivate its continued development and use, and these shall be presented in Section 3.4.

### 1.2 Objectives

Although Finite-Volume (FV) methods have seen widespread use in many advanced codes written for CFD \[3\], high-order variants of these schemes continue to present a difficult computational challenge; that is, the development and implementation of efficient and robust FV methods, of order higher than two, in three dimensions \[4\]. The primary objective of this
research is to attempt to address this computational challenge.

More specifically, the objective of this research is to develop, implement, and verify a high-order CENO FV scheme, in three-dimensional space, for several representative test flow problems. A high-order CENO FV scheme has recently been developed by Ivan and Groth [5, 6, 2, 7] for the application to the one- and two-dimensional forms of the Euler and Navier-Stokes equations. This previous work provides the starting point for this thesis and the focus of this follow-on research has been the extension of the high-order CENO scheme from the two-dimensional to the three-dimensional forms of the Euler and Navier-Stokes equations. High-order CENO schemes up to and including fourth-order accuracy in space are considered.

Due to the time constraints of the thesis, this work makes exclusive use of Cartesian grids (implying no curved boundaries) and is restricted to non-reacting flows. In an attempt to further maximize computational efficiency, the high-order scheme involves a parallel multi-block implementation on large scale, distributed-memory clusters with a message passing interface. The use of these techniques in the current implementation have been developed by Groth et al. [8, 9, 10].

1.3 Overview of Thesis

This thesis provides full descriptions of the mathematical models and numerical methods for the development and implementation of the proposed high-order CENO scheme. The implementation of the scheme is then subjected to a thorough verification process using a variety of test cases. In Chapter 2 the Godunov-type FV methodology is developed, with the aim of providing a clear understanding of the overall solution process. In addition, several important concepts and equations necessary for the development of the high-order CENO scheme are presented. Chapter 3 begins with a review of previous high-order work, the various monotonicity criteria, and the overview and motivations of the CENO scheme. The proposed high-order CENO scheme is then described in detail, including the $k$-exact least-squares reconstruction procedure, the inviscid and viscous flux evaluations, and the so-called smoothness indicator employed by the CENO scheme. Chapter 4 presents a verification of the implementation, including a demonstration of the high-order accuracy and of the smoothness indicator. Also included in Chapter 4 are the numerical results from test cases which have applied the high-order CENO scheme to the Euler and Navier-Stokes equations. In closing, Chapter 5 contains the final remarks and conclusions, as well as the recommendations for future work and further development of the proposed CENO scheme.
Chapter 2

Godunov-Type Finite-Volume Methods

Godunov-type FV methods for hyperbolic systems of conservation laws were first introduced by Godunov in 1959 [11]. These methods discretize the solution domain into control volumes (or cells) and apply the integral form of the conservation equations to each of the discrete control volumes [4]. This chapter begins with a presentation of the conservation equations, then moves on to the elements of the inviscid flux evaluation, reconstruction process, and time evolution employed by Godunov-type FV methods. This chapter aims to provide a clear understanding of the overall solution process, as well as the important concepts and equations necessary for the development of the high-order CENO scheme, which follows in the next chapter.

To clarify the content of this chapter in relation to its title, the reader is advised that this chapter shall only consider the evaluation of the inviscid (hyperbolic) fluxes appearing in both the Euler and Navier-Stokes equations (viz. the solution to the Riemann problem). The detailed presentation of the approach taken to extend the Godunov-type method and the evaluation of the viscous (elliptic) fluxes is reserved for Chapter 3. It is, however, convenient to present both the Euler and the Navier-Stokes equations together, and now and again, the reader will find some brief, preliminary discussions of the viscous flux evaluation throughout this chapter.

2.1 Conservation Equations

The behaviour of a compressible viscous gas is governed by the conservation of mass, momentum, and energy [3]. These well known governing equations are collectively referred to as
the Navier-Stokes equations. The Navier-Stokes equations are a coupled system of non-linear Partial Differential Equations (PDEs); providing a mathematical description for the manner in which the basic fluid properties relate and evolve in both space and time. What follows is a presentation of the three-dimensional form of the Navier-Stokes (viscous) and the Euler (inviscid) equations for a compressible Newtonian fluid, in Cartesian coordinates.

### 2.1.1 Navier-Stokes Equations in Cartesian Coordinate Frame

Using matrix-vector notation, which is amenable to the discussion of FV algorithms, the integral form of the Navier-Stokes equations may be written as follows:

\[
\int \int \int_V \left[ \frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot (\overrightarrow{F} - \overrightarrow{F_v}) \right] dV = 0,
\]

(2.1)

where \( \mathbf{U} \) is the vector of conserved solution variables,

\[
\mathbf{U} = \begin{bmatrix} \rho & \rho u & \rho v & \rho w & E \end{bmatrix}^T,
\]

(2.2)

\( \overrightarrow{F} = (\mathbf{F}, \mathbf{G}, \mathbf{H}) \) is the inviscid flux dyad, with components

\[
\mathbf{F} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho u v \\ \rho u w \\ u(E + p) \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} \rho v \\ \rho v^2 + p \\ \rho v w \\ \rho v w \\ v(E + p) \end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix} \rho w \\ \rho w^2 + p \\ \rho w v \\ \rho w w \\ w(E + p) \end{bmatrix},
\]

(2.3)

and \( \overrightarrow{F_v} = (\mathbf{F_v}, \mathbf{G_v}, \mathbf{H_v}) \) is the viscous flux dyad, with components

\[
\mathbf{F_v} = \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xz} \\ u\tau_{xx} + v\tau_{xy} + w\tau_{xz} - q_x \end{bmatrix}, \quad \mathbf{G_v} = \begin{bmatrix} 0 \\ \tau_{xy} \\ \tau_{yy} \\ \tau_{yz} \\ u\tau_{xy} + v\tau_{yy} + w\tau_{yz} - q_y \end{bmatrix},
\]

\[
\mathbf{H_v} = \begin{bmatrix} 0 \\ \tau_{xz} \\ \tau_{yz} \\ \tau_{zz} \\ u\tau_{xz} + v\tau_{yz} + w\tau_{zz} - q_z \end{bmatrix},
\]

(2.4)
In the above equations, \( u, v, \) and \( w \) are the velocity components in the \( x, y, \) and \( z \) directions, respectively, \( p \) is the pressure, \( \rho \) is the mass density, \( E \) is the total specific energy given by \( E = e + \frac{1}{2}|\vec{u}|^2 \), and \( e \) is the specific internal energy. Introducing the fluid viscosity, \( \mu \), for an isotropic fluid and making use of Stoke’s hypothesis, the stress tensor terms (i.e., components of \( \vec{\tau} \)) observed in the viscous flux dyad can be related by the so-called constitutive equations, as follows [12]:

\[
\begin{align*}
\tau_{xx} &= \frac{\mu}{3} \left( 4 \frac{\partial u}{\partial x} - 2 \frac{\partial v}{\partial y} - 2 \frac{\partial w}{\partial z} \right), \\
\tau_{xy} &= \frac{\mu}{3} \left( \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \right), \\
\tau_{yy} &= \frac{\mu}{3} \left( 4 \frac{\partial v}{\partial y} - 2 \frac{\partial u}{\partial x} - 2 \frac{\partial w}{\partial z} \right), \\
\tau_{xz} &= \frac{\mu}{3} \left( \frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} \right), \\
\tau_{zz} &= \frac{\mu}{3} \left( 4 \frac{\partial w}{\partial z} - 2 \frac{\partial u}{\partial x} - 2 \frac{\partial v}{\partial y} \right),
\end{align*}
\]

Finally, using Fourier’s law of heat conduction, and introducing the coefficient of thermal expansion, \( k \), and the temperature, \( T \), we define the components of the heat flux vector, \( \vec{q} \), as

\[
q_x = -k \frac{\partial T}{\partial x}, \quad q_y = -k \frac{\partial T}{\partial y}, \quad q_z = -k \frac{\partial T}{\partial z}.
\]

### 2.1.2 Thermodynamic Relations

The Navier-Stokes equations must be closed by an equation of state, relating the pressure, density, and temperature, as well as the empirical relations for the variation of viscosity and thermal conductivity with temperature: \( \mu(T) \) and \( k(T) \). This work considers only polytropic gases (that is, calorically perfect gases) obeying an ideal-gas equation given by

\[
p = \rho RT,
\]

where \( R \) is the gas constant.

With these thermodynamic relations, we may describe the Navier-Stokes equations as a system of equations with four independent variables \( (x, y, z, \) and \( t) \) and five unknown, dependent variables \( (\rho, u, v, w, \) and \( p \) or \( T) \). Hence, the five equations concisely expressed by Equation (2.1), along with the closing relations provided by the thermodynamic and constitutive relations above, provide a complete mathematical description of the fluid dynamics relevant to this work.

### 2.1.3 Euler Equations in Cartesian Coordinate Frame

The Euler equations are a simplification of the Navier-Stokes equations that are applicable to compressible inviscid fluid flow. They may be deduced from the Navier-Stokes equations
Chapter 2. Godunov-Type Finite-Volume Methods

presented in the previous section; we simplify Equation (2.1) by eliminating the viscous flux dyad. In doing so, we arrive at the integral form of the Euler equations given by

$$\iiint_V \left[ \frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F} \right] dV = 0,$$

(2.8)

where the vector of conserved variables, $\mathbf{U}$, and the inviscid flux dyad, $\mathbf{F}$, are given by Equations (2.2) and (2.3), respectively.

2.1.4 Hyperbolic and Elliptic Behaviour

The integral form of the conservation equations presented thus far, represent PDEs having a mathematical behaviour that is dependent on the order of the derivatives found in the solution fluxes [1]. For the Euler equations, Equation (2.8), the terms associated with the inviscid fluxes involve derivatives of the solution that are at most first-order. Hence, these equations are classified as hyperbolic with respect to time (viz. unsteady flows). Physically, this implies that solution changes (or disturbances) associated with these terms are transported via wave phenomena at finite speeds.

On the other hand, the solution fluxes of the Navier-Stokes equations, Equation (2.1), involve terms of both first and second-order derivatives of the solution. The second-order derivatives observed in the viscous fluxes introduce an elliptic behaviour to the Navier-Stokes equations. Physically, this implies that disturbances are felt instantaneously everywhere throughout the domain [1] and are effectively associated with infinite propagation speeds. As a result, the Navier-Stokes equations are classified as having a mixed-type behaviour.

It is important to note that the inviscid (hyperbolic) flux terms are evaluated separately, using different techniques, than the viscous (elliptic) flux terms. As previously mentioned, the detailed treatment of the viscous flux terms is reserved for Chapter 3. In the next section, we will concentrate on the treatment of the inviscid fluxes and will describe how Godunov-type FV methods incorporate the physical properties of the flow equations into the discretized formulation [13].

2.2 Godunov-Type Methods

Godunov-Type FV methods discretize the solution domain into control volumes (or cells) and apply the integral form of the conservation equations to each of the discrete control volumes [4]. They make use of the solution to a Riemann problem, in order to evaluate the inviscid fluxes
into and out of each control volume. Although these methods were designed for hyperbolic systems, extensions are possible for the treatment of elliptic operators, as described in Chapter 3. We begin by manipulating the Navier-Stokes equations given in the form of Equation (2.1). Using the divergence theorem and noting that \( t \) is an independent variable, Equation (2.1) may be re-written as:

\[
\frac{d}{dt} \iiint_V \mathbf{U} dV = - \iint_S \left( \mathbf{F} - \mathbf{F}_v \right) \cdot \mathbf{n} dS .
\]

Examining Equation (2.9), the right hand side (RHS) represents the net flux (or transport by the mean flow) of density, momentum, and energy. Due to the conservative nature of the governing equations, it is duly noted that the net flux through each control volume is the sole contributor to the time rate of change of \( \mathbf{U} \) in each control volume. Furthermore, the left hand side (LHS) of Equation (2.9) may be written in terms of the cell average values, \( \bar{\mathbf{U}} \), as follows:

\[
\frac{d}{dt} \iiint_V \mathbf{U} dV = V \frac{d\bar{\mathbf{U}}}{dt} .
\]

Therefore, Equation (2.9) may be re-written as

\[
\frac{d\bar{\mathbf{U}}}{dt} = - \frac{1}{V} \iint_S \left( \mathbf{F} - \mathbf{F}_v \right) \cdot \mathbf{n} dS .
\]

Application of a spatial discretization procedure to the RHS of Equation (2.11) results in a semi-discrete form of the conservation equations for the average solution value in each cell. This coupled system of non-linear, first-order Ordinary Differential Equations (ODE), for the cell-averaged solution may be written for a three-dimensional uniform grid, aligned with the Cartesian coordinate system, as follows:

\[
\frac{d\bar{\mathbf{U}}}{dt} = - \frac{1}{\Delta x} \left\{ \left[ F_{x,(i+\frac{1}{2},j,k)} - F_{x,(i-\frac{1}{2},j,k)} \right] - \left[ F_{x,(i+\frac{1}{2},j,k)} - F_{x,(i-\frac{1}{2},j,k)} \right] \right\} \\
- \frac{1}{\Delta y} \left\{ \left[ F_{y,(i,j+\frac{1}{2},k)} - F_{y,(i,j-\frac{1}{2},k)} \right] - \left[ F_{y,(i,j+\frac{1}{2},k)} - F_{y,(i,j-\frac{1}{2},k)} \right] \right\} \\
- \frac{1}{\Delta z} \left\{ \left[ F_{z,(i,j,k+\frac{1}{2})} - F_{z,(i,j,k-\frac{1}{2})} \right] - \left[ F_{z,(i,j,k+\frac{1}{2})} - F_{z,(i,j,k-\frac{1}{2})} \right] \right\}.
\]

The control volume here is taken as a cube with dimensions of \( \Delta x, \Delta y, \Delta z \), with a center identified by \((i,j,k)\), and whose faces are identified by the subscripts of \( \mathcal{F} \); representing the flux through and normal to each face. The viscous fluxes are identified in Equation 2.12 by an additional subscript \( v \).

In Godunov-type methods, the fundamental unknowns are the cell-averaged values of the solution variables \([14]\), that is, \( \bar{\mathbf{U}} \). The above ODEs describe the time evolution of these unknowns, in each computational cell, and may be integrated forward in time using an appropriate time-marching method. Having arrived at the semi-discrete form, given by Equation (2.12), the remainder of the Godunov-type FV solution procedure consists of three stages \([14]\):
1. **Reconstruction in each cell**: Given cell average values, provide a mathematical description for the spatial variation of the solution, using piecewise polynomials, within each cell (Section 2.4).

2. **Flux evaluation across cell interfaces**: Using the reconstructed polynomials of the cells, evaluate the numerical fluxes using numerical integration over each cell face. For the inviscid fluxes, this requires the solution to a Riemann problem at each quadrature point used in the numerical integration of the inviscid fluxes (Section 2.3). The viscous fluxes are treated by other means, such as the high-order method described in Section 3.7.

3. **Time evolution**: Having integrated the RHS of Equation (2.11), update the cell average values to the next time-step using an appropriate time marching method (Section 2.5). In this step-wise manner, we can construct the solution variation in time.

Although in practice the reconstruction step is carried out prior to the flux evaluation, it is instructive to first discuss the details of the latter. We then present the basics of the reconstruction process and the concept of monotonicity, before ending the chapter with a brief discussion of time marching methods.

### 2.3 Flux Evaluation

The Godunov-type FV scheme in three-dimensional space may employ arbitrary polyhedral cells; for example, tetrahedral or hexahedral cells may be used. This research will consider only hexahedral cells on Cartesian grids and the equations that follow reflect that purpose. Using Gaussian quadrature integration, the semi-discrete form of the Navier-Stokes Equations (2.12), in three dimensions, may be written for a particular cell \((i,j,k)\) in the compact form

\[
\frac{d\overline{U}_{i,j,k}}{dt} = -\frac{1}{V_{i,j,k}} \sum_{l=1}^{N_f} \sum_{m=1}^{N_g} \left( \left( \overline{F} - \overline{F}_{\nu} \right) \cdot \overline{n} A_{i,j,k,l,m} \right),
\]

where \(N_f = 6\) is the number of faces, \(N_g\) is the number of Gauss quadrature points, \(V\) is the cell-volume, \(A\) is the face area, and \(\overline{n}\) is the normal unit vector pointing outward from a given face. The numerical flux evaluation in three-dimensional space is carried out in alignment with \(\overline{n}\), and depends on the reconstructed values of the solution at the quadrature points which lie on the cell faces.

Following the reconstruction of the solution, there may exist discontinuities in the solution state at the cell interfaces. This is due to each cell having its own polynomial reconstruction. For
Chapter 2. Godunov-Type Finite-Volume Methods

simplicity, the cell on one side of the interface is referred to as the left-state and the cell on the other side of the interface is referred to as the right-state (labels inherited from the study of one-dimensional problems). Therefore, at a particular Gauss quadrature point, the flux is a function of both the left and right solution states provided by the reconstructed polynomials of the adjoining cells.

In addition, note that the number of Gauss quadrature points is increased as a function of the order of the reconstruction. Current practice in three-dimensional space is to use \( N_g = 1 \) for first- and second-order schemes, and \( N_g = 4 \) for third- and fourth-order schemes \([2]\), to ensure that the spatial accuracy of the scheme is preserved. Additional details of the numerical integration procedure will be presented in Chapter 3.

For the inviscid flux evaluation, the above dependencies may be summarized as

\[
\bar{F} \cdot \bar{n} = \bar{F}(U_{left}, U_{right}, \bar{n}, \eta = 0). \tag{2.14}
\]

On a Cartesian grid, \( \eta = 0 \) corresponds to \( \frac{x}{t} = 0, \frac{y}{t} = 0, \) and \( \frac{z}{t} = 0 \) for the \( x, y, \) and \( z \) direction fluxes, respectively (more on this in the next section). Regarding the evaluation of the viscous fluxes, the gradient of the solution at the quadrature points is also required, such that

\[
\bar{F}_v \cdot \bar{n} = \bar{F}_v(U_{left}, \nabla U_{left}, U_{right}, \nabla U_{right}, \bar{n}, \eta = 0). \tag{2.15}
\]

The remainder of this section focuses on the solution to the Riemann problem employed by Godunov-type FV methods for the evaluation of the inviscid fluxes. For the treatment of the viscous fluxes, the reader is referred to Section 3.7.

### 2.3.1 The Riemann Problem

The Riemann problem is a building block used in the evaluation of the inviscid flux terms appearing on the RHS of Equation (2.13). It is solved in order to resolve the discontinuities in the solution variables that exist at the cell boundaries upon reconstruction. The Riemann problem is an initial value problem for which an exact solution may be found using iterative techniques, in the case of the one-dimensional unsteady Euler equations and a polytropic gas \([15]\). There are also various approximate approaches to solving the Riemann problem that have gained popularity \([16, 17]\), a select few of which will be described in the next section.

Let us consider the Riemann problem in one-dimensional space, which is defined by the following initial data:

\[
U_0(x) = U(x, t = 0) = \begin{cases} 
U_{left} & \text{if } x \leq 0 \\
U_{right} & \text{if } x > 0.
\end{cases} \quad \text{for } -\infty \leq x \leq \infty \tag{2.16}
\]
Chapter 2. Godunov-Type Finite-Volume Methods

Figure 2.1: Wave patterns for the one-dimensional Riemann problem.

Given this initial data for the Euler equations, three centered waves will form a particular pattern (in the $x$-$t$ plane) which consists of rarefaction (R), contact (C), and/or shock waves (S). Figure 2.1 shows the four different wave patterns that may occur at the cell boundaries in the one-dimensional case, including: RCS, SCR, RCR, and SCS. The movement from $\mathbf{U}_{\text{left}}$ to $\mathbf{U}_{\text{right}}$ in the $x$-$t$ plane is fully contained by these four patterns. It should be noted that a fifth wave pattern, not shown in Figure (2.1), is also possible and consists of a vacuum (V) between two contact waves, all between two rarefaction waves $RCVCR$. However, this fifth case is not realizable in most practical situations (occurring in the limit of a perfect gas at zero pressure and temperature) [15], and will not be considered herein.

The solution to the Riemann problem is self-similar, such that the solution is constant along $\xi = \text{constant}$. Therefore, the particular pattern of waves combined with the Rankine-Hugoniot shock jump conditions and the Riemann invariants, are used to determine $\mathbf{U}(x,t)$ for $t > 0$. If we know $\mathbf{U}(x,t)$ for $t > 0$ on a given cell boundary, we may then calculate the required inviscid fluxes, using Equation (2.3).

The extension to three dimensions considered in this work is straight forward, in that we treat the flux through each face in an approximate manner. More specifically, we treat the flux through each Gauss quadrature point by assuming a locally one-dimensional flow aligned with the normal vector to the given cell-face. This approach appears to work well for solutions with smooth variation. For flows with discontinuities and shock waves, Roe makes the following remarks regarding the above approximation [16]:

This gives good results so long as the shock waves remain aligned with the computing grid: such shock waves are accurately recognized and appropriately treated. Problems arise when the shock wave lies obliquely across the grid, and are particularly severe as the solution attempts to reach a steady state.
Note that alternative methods have been proposed that are more multi-dimensional in their treatment of the Riemann problem in multiple dimensions, such as Residual Distribution (RD) schemes [18, 19]. These schemes remain an active area of research and are not considered in the scope of this work.

2.3.2 Approximate Riemann Solvers

Approximate Riemann solvers were engendered by the realization that it may not be necessary to solve the Riemann problem exactly, since the conserved solution is being updated in an approximate manner. For the Euler equations governing polytropic gases, it has been found that the time saved by using an approximate solver versus the exact solver of Gottlieb and Groth is minimal, if any at all [15]. Approximate Riemann solvers are useful when dealing with non-polytropic gases, or hyperbolic partial differential equations for which exact Riemann solvers are unavailable. The following is a concise overview of the methodologies employed by the various Riemann solvers investigated and used in this work.

Roe’s Approximate Riemann Solver

Roe’s approximate Riemann solver [16] is motivated by the trivial solution that is obtained for linear systems via their eigensystem decomposition. Here, the non-linear system of equations is expressed as \( \frac{\partial \mathbf{U}}{\partial t} + \mathbf{A} \frac{\partial \mathbf{U}}{\partial x} = 0 \), where \( \mathbf{A} = \frac{\partial \mathbf{F}}{\partial \mathbf{U}} \) is the flux Jacobian. Roe proposed a local linearization of the non-linear system of equations; let \( \mathbf{A}_* \) be the local linearization of \( \mathbf{A} \) about \( \mathbf{U}_{left} \) and \( \mathbf{U}_{right} \) such that a linear decomposition can be applied. This approach is not completely linear, since \( \mathbf{A}_* \) still depends on an averaged solution state; that is, \( \mathbf{A}_*(\mathbf{U}_*(\mathbf{U}_{left}, \mathbf{U}_{right})) \). The calculation of Roe’s averaged state and the various properties that it satisfies are outlined in detail in [16]. This average state was chosen in such a way that, for the case of an isolated discontinuity, the exact solution is recovered. If \( \mathbf{U}_* \) is known, then the change in inter-cellular flux may be computed and the flux may be evaluated using eigensystem decomposition. Furthermore, the situation in which a sonic point exists inside of a rarefaction waves can produce the so-called “rarefaction shocks”; an entropy-violating solution to the Euler equations. Harten’s entropy correction is employed to resolve this issue (for further details, the reader is referred to [20]).

The HLLE and Linde Flux Functions

The HLLE and Linde flux functions may be derived by first taking the integral of a closed path, \( S \), surrounding the breakup of the initial discontinuity of a Riemann problem [21]:
Figure 2.2: HLLE flux function diagram in the $x$-$t$ plane showing the three-states (left, middle, and right), two waves ($\lambda^-$ and $\lambda^+$), and the closed integration path, $S$.

\[ \oint_s (U_t dx - F dt) = 0. \]

The HLLE flux function \cite{22,23} assumes that there are only three states (labeled left, middle, and right), separated by two waves ($\lambda^-$ and $\lambda^+$). Figure 2.2 shows the three states, two waves, and closed integration path $S$ \cite{21}.

The wave speeds are calculated by making use of **Roe’s average state**. Finally, to evaluate the flux along $x/t = 0$ a second path integral is taken using only one side of the previous domain \cite{21}. The Linde flux function \cite{24} modifies the HLLE flux function by adding a correction term that incorporates the intermediate wave; for a one-dimensional shock-tube problem, this corresponds to the contact wave.

### 2.4 Reconstruction Basics

Given cell average solution values, $U$, the purpose of reconstruction is to determine the spatial distribution of the solution in each cell, $U(x, y, z)$, using polynomial approximations. The accuracy of the reconstruction is important because it determines the spatial order of accuracy of the FV scheme. For hyperbolic systems and smooth functions, a $k^{th}$-order polynomial reconstruction results in a $k+1$-order spatially accurate FV scheme \cite{2}. For the remainder of this
report, the accuracy of the solution will refer specifically to the spatial accuracy of the solution, unless otherwise stated.

The semi-discrete forms of Equations (2.12) and (2.13) have yet to introduce any assumptions or approximations which formally introduce truncation error. In order to evaluate the flux at each cell interface, the left and right solution states must be provided to a given Riemann solver, and these states are found using the reconstruction of \( \mathbf{U} \) in each cell. Note that since we are only given the cell average values, \( \mathbf{U} \), we are forced to make assumptions regarding the distribution of \( \mathbf{U} \) within each cell. Also note that the cell average values are either provided as initial conditions, or via the time integration of the solution.

### 2.4.1 The Euler Equations in One Dimension

In order to demonstrate the various facets of reconstruction, we will first detail the methodology used in one-dimensional reconstruction; returning to three-dimensional high-order reconstruction in Chapter 3. As such, we consider the solution to the one-dimensional form of the Euler equations. Using an explicit Euler time marching scheme (which is first-order accurate in time), Equation (2.12) can be re-expressed as

\[
\mathbf{U}_i^{n+1} = \mathbf{U}_i^n - \frac{\Delta t}{\Delta x} \left[ F_{i+\frac{1}{2}}^n - F_{i-\frac{1}{2}}^n \right],
\]

(2.17)

where the control volume of interest is identified by the index \( i \), the time level is identified by \( n \), and the inviscid flux values, \( F \), at the cell interfaces are found from the solution to the Riemann problem.

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

A first-order accurate FV scheme can be achieved by using a piecewise constant (zeroth-order) reconstruction and assumes that the solution is a constant value within each cell. The right hand side of Equation (2.17) can be found in a straightforward manner by specifying the following:

\[
F_{i+\frac{1}{2}}^n = F( \mathbf{U}_{left} = \mathbf{U}_i^n, \mathbf{U}_{right} = \mathbf{U}_{i+1}^n, \frac{x}{t} = 0 ) \quad (2.18)
\]

\[
F_{i-\frac{1}{2}}^n = F( \mathbf{U}_{left} = \mathbf{U}_{i-1}^n, \mathbf{U}_{right} = \mathbf{U}_i^n, \frac{x}{t} = 0 ). \quad (2.19)
\]

Figure 2.3 provides a depiction of a piecewise constant reconstruction scheme. The combination of piecewise constant reconstruction, explicit Euler time marching, and an exact Riemann solver for the evaluation of the inviscid fluxes is often referred to distinctly as Godunov’s Method (from
which other Godunov-type FV methods have evolved). Godunov's Method is a first-order accurate scheme in both space and time.

Having described the discretized form of the one-dimensional Euler equations, we take a quick detour in order to present a comparison of the flux functions previously described in Section 2.3. The comparison is carried out by solving the one-dimensional form of the Euler equations using Godunov's method. In Figure 2.4, the results of an initial value problem (representing a shock tube problem) using the various flux functions are compared to each other and to the exact solution. The initial left and right states were $\rho_l = 4.696 \text{ kg/m}^3$, $u_l = 0 \text{ m/s}$, $p_l = 404.4 \text{ kPa}$, $\rho_r = 1.408 \text{ kg/m}^3$, $u_r = 0 \text{ m/s}$, $p_r = 101.1 \text{ kPa}$, and $0 \leq t \leq 7\text{ms}$; taken from [25]. It is evident from this comparison, that the exact and approximate Riemann solvers yield very similar results for the one-dimensional shock-tube problem. In addition, it can be seen that the Linde flux function does indeed provide improved results in resolving the contact wave, as compared to the HLLE flux function.

### 2.4.3 Piecewise Linear Reconstruction

A second-order accurate FV scheme can be achieved by using a piecewise linear (first-order) reconstruction and assumes that the solution varies linearly with distance throughout each cell. In this case, the solution varies linearly about the mid-point value (which is also the cell-average
Figure 2.4: First-order accurate comparison of flux functions: shock tube problem: t = 7 ms, 100 cells, CFL = 0.7.

value) and the reconstruction in cell $i$ is then given by

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

(2.20)

The only unknown in Equation (2.20) is the slope, $\left. \frac{\partial U}{\partial x} \right|_{i}$, as depicted in Figure 2.5(a). In order to determine the slope, it is possible to use information from only one neighbouring cell, for the one unknown, such as the backward or forward finite-difference approximations for the slope, as depicted in Figure 2.5(b). In doing so, we have biased the reconstruction stencil to one side, or the other, resulting in two quite different values for the slope. This, in turn, raises the question: Given only cell average values, what is the optimal value for the unknown slope?

One possibility, (referred to herein as Least-Squares Reconstruction) is to extend the representation of cell $i$ to its neighbours in both directions, and to impose the following constraints:

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

(2.21)

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

(2.22)

Equations (2.21) and (2.22) represent constraints which state that the reconstructed polynomial in cell $i$ should recover the neighbouring cell average values when it is integrated over the domain of the given neighbouring cell. This approach can be used to avoid biasing the solution to one side. However, note that we are left with a system of two linear equations for only one unknown;
an overdetermined system of equations. The overdetermined system may be solved using a least-squares method, where the goal is to minimize the sum of the squares of the errors. The error in this case is the difference between the actual cell average values, and the average values obtained when integrating the reconstructed polynomial of cell \( i \) over the domain of the given neighbour; represented by the red bars in Figure 2.5(d). The least-squares method is to be discussed in more detail in Chapter 3. For this particular case, involving just one unknown, the least squares solution is simply the average of the forward- and backward-difference approximations for the slope (equivalently, the centered finite-difference approximation), as shown in Figure 2.5(d) and given by

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

(2.23)

In this manner, the size of the stencil (that is, the number of neighbours used in determining the unknown slope of cell \( i \)) may be increased to include even more neighbours; we simply include additional constraint equations when formulating the over-determined system of equations. As we will see in Chapter 3, the size of the stencil will depend on the total number of derivatives (unknowns) to be found, which is of course, a function of the dimensionality of the problem and the order of the polynomial reconstruction. As a final note, since the piecewise linear reconstruction is taken about the cell center locations, the average values in each cell are automatically preserved.

### 2.4.4 Monotonicity and Godunov’s Theorem

In moving to second and even higher-order spatial discretizations of FV schemes there are issues related to solution monotonicity (oscillations) that occur near large solution gradients and discontinuities, such as shock waves. Figure 2.6 depicts the oscillatory behaviour (often referred to as the Gibbs phenomenon [26]) that are typical of second and higher-order schemes in the vicinity of shocks. The magnitude of the oscillations are typically proportional to the size of the solution jump [27]. If one considers the Fourier representation of the numerical solution, then the oscillatory behaviour can be attributed to numerical error in the propagation speeds of different phases of the solution content [1]; also known as numerical dispersion. If not properly dealt with, the robustness of the scheme may be seriously compromised, in that non-physical fluid properties may result, such as negative pressures, densities, and species concentrations (the latter are important for reactive flows). A potential scenario is shown in Figure 2.6, where a second-order scheme has produced a negative pressure. The next time the algorithm attempts to evaluate the sound speed, \( a = \sqrt{\gamma p/\rho} \), in the vicinity of the shock, the result will lead to the square root of a negative number, and the method cannot (or at least should not) proceed.
Godunov was interested in determining if it was possible to have a second-order scheme that was also monotone (i.e., monotonicity preserving); he went on to prove that one did not exist by considering an explicit constant coefficient scheme applied to the linear advection equation: \[ \frac{\partial u}{\partial t} + a \frac{\partial u}{\partial x} = 0. \] Godunov’s theorem states that for the class of constant coefficient schemes applied to hyperbolic equations, there are no schemes which are both second-order accurate and monotonicity preserving [11].

It should be noted that a constant coefficient scheme applied to the linear advection equation is itself a linear scheme. The important implication of Godunov’s theorem is that in order to control the unwanted oscillations, we must make use of non-linear methods, which have coefficients that depend on the solution and locally revert to a first-order scheme based on the solution content (that is, in the vicinity of discontinuities). A more rigorous discussion on monotonicity...
may be found in Section 3.2, which presents an overview of the various criteria related to the enforcement of monotonicity (including the requirements of a \textit{monotonicity preserving} scheme). For now, it shall suffice to appreciate that we need to attempt to enforce the solution monotonicity, and that this must be done using non-linear means, such as the slope limiters, which are described in the next section.

\subsection*{2.4.5 Enforcing Monotonicity: Slope Limiters}

As a result of Godunov’s theorem, second-order accurate Godunov-type FV schemes typically employ non-linear methods (such as flux or slope limiting) to control the spurious oscillations that occur in the vicinity of discontinuities. The limiter functions described in this section are considered non-linear in the sense that they themselves depend on the solution values that they are limiting. Although we have yet to discuss the matter, it should be noted that the proposed high-order CENO scheme will make use of the limited piecewise linear reconstruction presented in this section.

The approach used in this work is to limit the values of the slopes in the piecewise linear reconstruction polynomial. In the one-dimensional case, the reconstruction polynomial in cell \( i \), with the use of a slope limiter, is given by

\[
U_i(x) = U_i + \psi_i \frac{\partial U}{\partial x} \bigg|_{x_i} (x - x_i),
\]  

\begin{equation}
(2.24)
\end{equation}

where \( \psi_i \) is the \textit{slope limiter} that has been explicitly introduced to modify the solution gradient in the cell. Note that \( \psi_i = 0 \) corresponds to piecewise constant reconstruction and \( \psi_i = 1 \).
corresponds to unlimited piecewise linear reconstruction. In general, these limiters are imposed such that the reconstructed values of the solution in a given cell, do not exceed the maximum and minimum of neighbouring cell average values [14]. In this work, the primitive variables are reconstructed and limited so as to maintain tighter control on the positivity of certain flow variables (namely, pressure and density). Figure 2.7, re-illustrated from [14], depicts the typical application of slope limiters to the reconstruction of a smoothly varying function. Notice that at smooth extrema, the slope limiter is assigned a value of zero, thereby forcing piecewise constant reconstruction in order to enforce the solution monotonicity.

There are many different limiter formulations, however, we will consider only two of them in detail here; known as the Barth-Jespersen [28] and Venkatakrishnan [29] slope limiters. Both of these limiters can be described, for a given cell $i$, as

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

(2.25)

where $k$ indexes the quadrature points at which the flux is evaluated. In one-dimensional space, there are simply two quadrature points for a given cell (one at each boundary). It is duly noted that, for the multi-dimensional case, we check our solution monotonicity (that is, calculate the limiter) only at the Gauss quadrature points used in the numerical integration of the flux. We
then select the minimum of $\psi_{i,k}$ considering all $k$ quadrature points of the given cell. In the multi-dimensional case this implies that it is possible for the solution to have overshoots and undershoots at other locations, even if the solution is properly limited at the quadrature points.

For both limiters, the value of $\psi_{i,k}$ is a function of $r_k$ (such that $\psi_{i,k} = \psi_{i,k}(r_k)$), where $r_k$ is defined as follows:

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

(2.26)

Here $\bar{u}$ represents the primitive solution variable of interest and the various terms in Equation (2.26) are defined as follows:

- $\bar{u} =$ cell average value (for cell $i$),
- $u_{\text{max}} = \max(\bar{u}, \bar{u}_{\text{neighbours}}) =$ maximum cell average value amongst all cells used in the reconstruction of cell $i$,
- $u_{\text{min}} = \min(\bar{u}, \bar{u}_{\text{neighbours}}) =$ minimum cell average value amongst all cells used in the reconstruction of cell $i$, and
- $u_k =$ unlimited reconstructed value at the $k^{th}$ quadrature point.

### Barth-Jespersen Limiter:

The Barth-Jespersen limiter [28] is a limiter that tends to limit only where required, i.e., minimal dissipation is added to the scheme. It follows a discontinuous path from zero to one and may be defined as follows:

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

(2.27)

### Venkatakrishnan Limiter:

The Venkatakrishnan limiter [29] is somewhat similar in form to the well-known Van Albada limiter and is a more dissipative limiter than that of Barth-Jespersen. The Venkatakrishnan
Figure 2.8: Comparison of limiters for the shocktube problem: t = 7 ms, 200 cells; density distribution.

The limiter follows a smooth path from zero to one and may be defined as follows:

\[
\psi_{i,k}(r_k) = \begin{cases} 
1 & \text{for } (u_k - \overline{u}) = 0 \\
\frac{r_k^2 + 2r_k}{r_k^2 + r_k + 2} & \text{for } (u_k - \overline{u}) \neq 0.
\end{cases}
\] (2.28)

We now carry out a comparison of the above limiters and of the unlimited second-order accurate FV scheme, using the author’s own 1D code. Figure 2.8 features a comparison of the exact, first-order and second-order results (with and without slope limiting) for the same shocktube problem presented Section 2.3.2. Note that the second-order schemes are significantly less dissipative than the first-order scheme and that the unlimited second-order scheme is significantly more dispersive than the limited schemes. In this particular case, the Barth-Jespersen and Venkatakrishnan slope limiters behave very similarly and provide similar solution quality.

Clipping of Smooth Extrema:

A significant drawback regarding the use limiters in piecewise linear reconstruction is that they inhibit the ability to “capture smooth extrema without clipping” [29]. As designed, slope lim-
Chapter 2. Godunov-Type Finite-Volume Methods

Iters do an excellent job of preserving the monotonicity of the solution content at discontinuities; locally reverting to a piecewise constant reconstruction wherever overshoots and undershoots occur. However, the limiters cannot distinguish between a sharp discontinuity and a local smooth extrema. This results in a degradation of the scheme’s spatial accuracy in smooth regions. Figure 2.9 demonstrates the difference between the two indistinguishable scenarios; in this case, we observe the clipping of a smooth maximum. Harten explains that “the perpetual damping of local extrema determines the cumulative global error” \[30\] of the scheme, and quantifies the orders of accuracy as \(O(\Delta x)\) in the \(L_\infty\)-norm, \(O(\Delta x^{3/2})\) in the \(L_2\)-norm and \(O(\Delta x^2)\) in the \(L_1\)-norm \[30\].

![Figure 2.9: The clipping of smooth extrema occurring as a result of slope limiters being applied to a smooth maximum (right), compared to the case of a discontinuity (left).](image)

As a prelude to the next chapter, the clipping of smooth extrema is altogether avoided by the use of the so-called smoothness indicator employed in the proposed high-order CENO scheme, described herein for its application to three-dimensional flows. The smoothness indicator is described in Section 3.6.

2.5 Time Marching Schemes

Godunov-type FV methods may apply either implicit or explicit time marching schemes to the semi-discrete form of the conservation laws, Equation (2.12). The objective is to evolve the solution forward in time from some initial condition to either a steady-state solution, in which \(\frac{dU}{dt} = 0\), or to some desired final time for an unsteady problem. The considerations of this work have been restricted to explicit time marching methods. The high-order CENO scheme itself is not necessarily restricted to explicit methods, but there are additional considerations to be addressed in order to incorporate the high-order CENO scheme with an implicit time marching method. Such considerations have been left to future work.
For consistency, in terms of accuracy, the order of accuracy in time of the explicit time marching methods have been matched (as best as possible) to the order of accuracy in space of the CENO scheme, for any given test case. For the fourth-order spatially accurate CENO scheme, a fourth-order Runge-Kutta explicit time marching method is used. The third-order spatially accurate CENO scheme also makes use of the fourth-order Runge-Kutta scheme. The second-order CENO scheme uses a second-order Runge-Kutta scheme, and finally, the first-order (piecewise constant) spatial discretization uses a first-order explicit Euler time marching method. The details of these well-known time marching schemes are not included herein, as the time marching is not the primary focus of this research. For the details of these (and other) time marching methods, the reader is referred to several textbooks on the subject [3, 31, 13].

With regards to the solution monotonicity, discussed in Section 2.4.4, it should be noted that the time-marching method may have an effect. Although the limiters can ensure that the left and right states (provided to a Riemann solver) introduce no new local extrema at the quadrature points, one still has to solve the Riemann problem and on top of that, update the solution in time. Upon updating the solution, there is no guarantee that our previous efforts to enforce monotonicity have not been thwarted. This is an important consideration to keep in mind during the verification process of any second or higher-order scheme.

As a final note, the stability of the explicit time marching schemes used herein are controlled by the appropriate selection of the time step, $\Delta t$. From the perspective of a Godunov-type FV scheme, the time step is selected such that the outermost waves from adjacent Riemann problems (i.e., adjacent cell interfaces) do not have enough time to interact with each other. This is illustrated in Figure 2.1 by the red horizontal line depicting the subsequent time level, $t + \Delta t$. We select the time step by making use of the Courant-Friedrichs-Lewy number, $CFL$, such that

$$\Delta t = CFL \frac{\Delta x}{(|u| + a)_{max}}. \quad (2.29)$$

In this work, given the aforementioned combinations of time-marching and reconstruction procedures, the value of the CFL number is typically taken to be less than or equal to unity, in order to maintain the numerical stability of the scheme. In general, the maximum value of the CFL number is dependent on both the time marching method and the spatial reconstruction procedure.
Chapter 3

High-Order CENO Scheme

Our aim in this chapter is to develop a high-order CENO scheme in three dimensions by extending the previous work of Ivan and Groth [5, 6, 2]. The CENO scheme is a shock-capturing Godunov-type FV scheme which addresses and controls the Gibbs phenomenon and altogether avoids the clipping of smooth extrema. The CENO scheme is well suited to the various applications mentioned in Chapter 1, and is also designed to be a compromise between the often competing goals of accuracy, computational efficiency, and robustness. In addition, the CENO scheme is valid for both structured and unstructured grids. In this work we have restricted our attention to regular and uniform Cartesian grids in three dimensions. Nonetheless, the development that follows is carried out in a general fashion, without exploiting many of the potential simplifications associated with uniform Cartesian grids. In doing so, the CENO scheme may be readily extended to general hexahedral- and tetrahedral-type cells for structured, unstructured, and hybrid grids, which will be topics for future research.

A “high-order” scheme shall herein refer to any scheme with a spatial order of accuracy greater than second-order. Furthermore, this work considers up to and including fourth-order accuracy for both the Euler and Navier-Stokes equations, however, the computational framework has been developed with the flexibility of being extended to a spatial accuracy of any desired order.

To elucidate the content of this chapter, we shall consider it in a three-part presentation. In the first part we will focus on a review of previous work, the various monotonicity criteria, and the overview and motivation of the proposed CENO scheme. The second part, which includes Sections 3.5 to 3.7, is then dedicated to the detailed development of the CENO scheme; including the $k$-exact reconstruction and its least-squares solution, the high-order inviscid and viscous flux evaluations, and the definition and use of the so-called smoothness indicator. Finally, the third part of this chapter shall consider the treatment of the boundary conditions and the parallel
3.1 Previous High-Order Work

Many researchers have journeyed down this road with the same overarching objective of achieving more accurate solutions at lower computational costs. Over the past two decades a number of schemes have been developed for hyperbolic conservation laws, with the goal of obtaining high-order spatial accuracy while maintaining solution monotonicity near discontinuities. The original ENO schemes were first introduced by Harten et al. [32]. These ENO schemes perform reconstruction on multiple stencils and then select the “smoothest” of the stencils in an attempt to enforce monotonicity [2]. Abgrall [33] and Sonar [34] have extended ENO schemes for the application to unstructured grids. A variation of the ENO schemes, referred to as Weighted ENO (WENO) schemes have also been developed [35, 36, 37], which assign a weighting to multiple reconstruction stencils in an attempt to ameliorate the convergence to steady-state. Unfortunately, these schemes tend to be computationally expensive and complex when applied to multiple space dimensions and may result in poorly conditioned coefficient matrices when solving for the piecewise polynomial solution representation in each cell [2]. Although this has ultimately limited the widespread use of ENO and WENO schemes [2], they have still been implemented in combination with AMR for both structured and unstructured grids, as described in [38].

A more flexible high-order scheme that is more easily extended to multiple dimensions, the so-called $k$-exact reconstruction procedure, has been developed by Barth and Fredrickson [14, 39]. Note that the $k$-exact reconstruction is used as part of the CENO formulation herein proposed [2]. There has also been significant advances in high-order spatial accuracy applied to viscous flows using spectral discretization methods, known as Discontinuous Galerkin (DG) methods [40, 41, 42]. The finite difference approach to high-order spatial discretization has also been considered for the prediction of turbulent aerodynamic flows by De Rango and Zingg [43]. Despite the consistent and continuous research effort, there is still no consensus (if ever one is expected) on a robust and effective way of providing high-order accuracy without compromising the solution monotonicity [2].

3.2 Classification of Schemes

The high-order CENO scheme makes use of a fixed central stencil. As such, there may be some confusion as to why the proposed CENO scheme is referred to as Essentially Non-Oscillatory
(ENO); since we do not make use of adaptive or weighted stencils. However, Harten [27] provides an explicit definition of what is required of an ENO-satisfying scheme, and has investigated ENO schemes on fixed stencils with hybrid reconstruction [26]. In this light, ENO schemes do not necessitate, per se, the use of multiple stencils. Furthermore, the proposed CENO scheme borrows heavily from several ENO-related ideas in its formulation and, therefore, the reference to ENO is retained in the naming of the CENO scheme.

At this point, it is beneficial to take a step back and try to define some of the terms which are commonly used to classify CFD methods in terms of their ability to enforce monotonicity. The formal theory presented below has been developed for scalar equations or for linear systems in one-dimensional space [13], and as such, we will make use of the classical model equation for hyperbolic PDEs, that is, the linear advection equation:

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

Note that any explicit constant coefficient scheme for the solution of the linear advection equation may be written as:

$$u_i^{n+1} = \sum_p c_{ip} u_{i+p}^n,$$

(3.1)

where $i$ indexes the cells in one-dimensional space, $n$ indexes the time-levels, $p$ indexes the neighbouring cells which make up the stencil of cell $i$, and $c_{ip}$ represents the coefficients for the given cell in the stencil, that is, cell $i + p$.

It is possible to define a hierarchy of varying degrees of strictness (or weakness) in the enforcement of monotonicity. In order, from the most strict monotonicity requirements to the weakest, a scheme may be classified as either

- Monotone;
- Total Variation Diminishing (TVD);
- Monotonicity Preserving; or
- Essentially Non-Oscillatory (ENO).

We now consider a brief description of each of these criteria.

**Monotone:** A monotone scheme is one in which a monotonically increasing solution remains monotonically increasing, or, a monotonically decreasing solution remains monotonically decreasing, for all time. For Equation (3.1) we therefore require that the values of $c_{ip}$ are positive, such that the difference between any two adjacent cells always maintains its sign (that is, positive for an increasing solution, and negative for a decreasing solution). If we
have two adjacent cells whose solutions are expressed by

\[ u_{i+1}^{n+1} = \sum_{p} c_{ip} u_{i+1+p}^{n} \quad \text{and} \quad u_{i}^{n+1} = \sum_{p} c_{ip} u_{i+p}^{n}, \quad (3.2) \]

then taking the difference, we have

\[ u_{i+1}^{n+1} - u_{i}^{n+1} = \sum_{p} c_{ip} (u_{i+1+p}^{n} - u_{i+p}^{n}), \quad (3.3) \]

where \( c_{ip} \geq 0 \) for all \( i \) and \( p \).

**Total Variation Diminishing (TVD):** A scheme is total variation diminishing if the total variation in \( x \) (TV) of any physically admissible solution does not increase in time [13]. If we let

\[ TV(u) \equiv \sum_{i} |u_{i+1} - u_{i}|, \quad (3.4) \]

then a numerical scheme is said to be total variation diminishing if

\[ TV(u^{n+1}) \leq TV(u^{n}). \quad (3.5) \]

Note that in the scalar case, the Runge-Kutta time marching methods are “order-preserving, and consequently also TVD” [27]; hence the total variation of a Godunov-type scheme “is dominated by that of the reconstruction step.” [27].

**Monotonicity Preserving:** Defined somewhat more loosely than TVD, or at least without a strict quantitative criteria; a monotonicity preserving scheme satisfies the following requirements [13]:

1. The scheme does not introduce new extrema. That is, the number of solution extrema is preserved in time [44], and;
2. Local maximums are non-increasing and local minimums are non-decreasing.

Note that for linear systems, monotonicity preservation leads to the same conditions as monotonicity, which by Godunov’s theorem, implies first-order accuracy [13].

**Essentially Non-Oscillatory (ENO):** An essentially non-oscillatory scheme is one in which the piecewise polynomial interpolation, \( R_{k}(x; u) \), is designed to provide high-order accuracy in smooth regions, but is also designed to avoid the Gibbs phenomenon at solution discontinuities [44]. To better explain the ENO criteria, let us replace the solution described by Equation (3.1), with a “solution operator” which updates \( u^{n} \) to \( u^{n+1} \), such that \( u^{n+1} = E_{h} \cdot u^{n} \). Let the operator, \( E_{h} \), for a Godunov-type scheme include all three stages (reconstruction, flux evaluation, and time evolution). Harten states that the ENO
criteria on the reconstruction operator, $R_k(x; u)$, may be expressed in terms of the total variation as

$$TV(R_k(x; u)) \leq TV(u) + O(\Delta x^{k+1}), \quad (3.6)$$

where $k$ is the order of the reconstruction polynomial. The resulting Godunov-type scheme, which uses an ENO satisfying reconstruction, is itself an ENO satisfying scheme, if it satisfies the following ENO criteria \cite{27}:

$$TV(u^{n+1}) = TV(E_k \cdot u) \leq TV(u) + O(\Delta x^{k+1}). \quad (3.7)$$

Harten has proved that all monotone schemes are TVD, and that all TVD schemes are monotonicity preserving, and as such, a hierarchy has been established \cite{13}. However, all TVD schemes are restricted to at most second-order accuracy \cite{27} (see Section 2.4.5). In moving to high-order schemes, the weaker ENO criteria was established by Harten \cite{27}. It is important to note that ENO-type schemes (including the CENO scheme) take measures to avoid the Gibbs phenomenon in the vicinity of discontinuities. However, note that in the vicinity of smooth extrema, the ENO criteria implies that spurious oscillations may in fact exist, thereby allowing the scheme to provide high-order accuracy in those regions, all the while satisfying a monotonicity criteria. Equation (3.6) states that the oscillations that may occur in the vicinity of smooth extrema diminish as the mesh is refined, and they do so at an order which is related to the order of the polynomial reconstruction. Also, notice that if we eliminate the last term on the right hand side of Equation (3.7), we recover the TVD criteria. From this perspective, we can acknowledge why Harten refers to ENO satisfying schemes as “almost TVD” \cite{27}. A more rigorous development of the above criteria may be found in the literature \cite{13, 44, 32, 27}. The purpose of this section was to provide a brief overview of the terminology commonly used in relation to monotonicity and - in relevancy to this work - to introduce the ENO criteria satisfied by the CENO scheme.

### 3.3 Overview of the CENO Scheme

A high-order CENO FV scheme is proposed for the numerical solution of the Euler and Navier-Stokes equations in three dimensions. Rather than adapting or weighting multiple reconstruction stencils, the CENO scheme makes use of a hybrid reconstruction approach such that a fixed central stencil is maintained, and the reconstruction within that stencil is switched to avoid the Gibbs phenomenon. In regions where the solution content is decidedly smooth or fully-resolved, the unlimited high-order reconstruction technique is applied; for this the CENO scheme uses a $k$-exact least-squares reconstruction procedure. However, if the solution is deemed
discontinuous or under-resolved then the reconstruction method switches, in those regions, to a limited piecewise linear reconstruction scheme which makes use of either the Barth-Jespersen or Venkatakrishnan slope-limiters. The switching to limited piecewise linear reconstruction provides a means of eliminating the spurious oscillations that occur near discontinuities. The switching procedure is facilitated by a so-called smoothness indicator. The objective of the smoothness indicator is implied by its name, in that it must determine whether or not the solution content in each cell is either smooth/fully-resolved or discontinuous/under-resolved.

The high-order CENO reconstruction procedure, for a given state variable, may be summarized as follows:

1. Perform an unlimited high-order reconstruction, in all cells, using the high-order $k$-exact least-squares reconstruction method. Section 3.5.

2. Calculate the smoothness indicator in order to determine if the solution is smooth/fully-resolved or discontinuous/under-resolved. Section 3.6.

3. If any cells have been flagged as discontinuous/under-resolved, then discard the high-order reconstructions (in those cells only) and revert to a limited piecewise linear reconstruction.

### 3.4 Motivation of the CENO Scheme

The design of the CENO scheme, described in the previous section, has several advantages that motivate its continued development and use. The CENO scheme:

- **Eliminates the spurious oscillations** that occur near solution discontinuities while, at the same time, is capable of capturing smooth solution extrema with high-order accuracy.

- Uses a **fixed central stencil**, which results in the most accurate reconstructions (as compared to biased stencils of the same size), and “such schemes are excellent numerical solvers for problems with smooth solutions.” [26]

- Takes a **simple approach**, relatively speaking, which avoids the complexities associated with other ENO and WENO schemes that require reconstruction on multiple stencils as well as many of the difficulties associated with poorly conditioned coefficient matrices for the least squares problems [2].
Chapter 3. High-Order CENO Scheme

- Uses a robust smoothness indicator that can be used to identify not only discontinuities, but also regions which are under-resolved (i.e., too few cells or grid points). This makes the high-order CENO scheme particularly well suited for its use with Adaptive Mesh Refinement (AMR); the benefits of which have been demonstrated by Ivan and Groth for the two-dimensional scheme [6, 2].

- Makes use of \textit{k}-exact reconstruction which is readily extendable to multiple dimensions (as demonstrated in this work) and unstructured grids [14] (a future consideration for the CENO scheme).

- Provides a practical compromise between accuracy, computational efficiency, and robustness that is well suited to the applications mentioned in Chapter 1.

Of course, no scheme is without its disadvantages. Those of the proposed CENO scheme include:

- Uniform accuracy, which is a feature of ENO and WENO schemes, is formally lost by using the hybrid reconstruction procedure.

- Additional computational cost associated with the evaluation of the smoothness indicator, and also in carrying out a second reconstruction (the limited piecewise linear reconstruction) when and where the switching occurs.

- The CENO scheme is not a compact scheme in comparison to say, the DG method [40, 41, 42]. The compactness of the scheme (or lack there of) is a point that will be made clear by our discussion in Section 3.8.

In considering the above advantages and disadvantages, it is acknowledged that the proposed CENO scheme is in fact a compromise. This compromise is itself considered to be another advantage of the CENO scheme, and is listed above as such. In the underlying pursuit of accelerating and optimizing the design of future aerodynamic and aerospace related technology, it is the author's opinion that a practical compromise between the often competing goals of accuracy, computational efficiency, and robustness is precisely what is needed in the development and application of a high-order scheme. The next several sections, 3.5 to 3.7, will focus on the detailed development of the various facets of the CENO scheme proposed herein.

3.5 \textit{k}-Exact Least-Squares Reconstruction

In moving to high-order FV schemes, this work considers the so-called \textit{k}-exact least-squares reconstruction technique developed by Barth and Fredrickson in the early 1990’s [14, 39]. The
$k$-exact reconstruction is given by the $k^{th}$-order Taylor series expansion of the variable $U$, about the cell center; where throughout this work, $U$ shall denote a scalar quantity when not in bold-face.

Recall from Section 2.4 that the order of the polynomial reconstruction determines the spatial order of accuracy of the scheme. In smooth regions, a $k^{th}$-order reconstruction results in a $k+1$-order scheme for the Euler equations, and a $k^{th}$-order scheme for the Navier-Stokes equations (more on this in Section 3.7). Henceforth, such discussions of high-order accurate results shall imply smooth regions of the solution (in which no switching occurs). Furthermore, we note that Godunov-type FV methods have the flexibility of carrying out the solution reconstruction on either the conserved variable vector, $\mathbf{U} = [\rho \; \rho u \; \rho v \; \rho w \; E]^T$, or the primitive variable vector, $\mathbf{W} = [\rho \; u \; v \; w \; p]^T$. In either case, each component of the vector is reconstructed individually, and therefore, the variable $U$ (as it is used here) may represent any given component of either vector, $\mathbf{U}$ or $\mathbf{W}$. As mentioned in Section 2.4.5, in this work, the primitive variables are reconstructed (and limited) so as to maintain tighter control on the positivity of certain flow variables (namely, pressure and density). However, the current implementation of the CENO scheme requires very few changes in order to reconstruct the conserved variables, rather than the primitive.

In either case, the $k^{th}$-order Taylor series reconstruction polynomial, in three dimensions, may be conveniently expressed in a compact and general form as follows [2]:

$$
U^k_{ijk}(x,y,z) = \sum_{p_1=0}^{k} \sum_{p_2=0}^{k} \sum_{p_3=0}^{k} (x - x_{ijk})^{p_1} (y - y_{ijk})^{p_2} (z - z_{ijk})^{p_3} D_{p_1p_2p_3}.
$$

(3.8)

Here the various terms in Equation (3.8) are defined as follows:

- $U$ : the primitive or conservative variable being reconstructed
- $k$ : the order of the reconstruction
- $ijk$ : the indices in the $x$-, $y$-, and $z$-coordinate directions, identifying the cell that is being reconstructed (commas between indices omitted for clarity)
- $U^k_{ijk}(x,y,z)$ : the $k$-exact polynomial reconstruction of the variable $U$ in the cell $ijk$
- $pqr$ : the indices in the $x$-, $y$-, and $z$-coordinate directions, identifying the neighbouring cells that form the reconstruction stencil of cell $ijk$ (commas between indices omitted for clarity)
- $x_{ijk}, y_{ijk}, z_{ijk}$ : the coordinates of the cell centroid of cell $ijk$
- $p_1, p_2, p_3$ : the summation indices which must always satisfy the condition that $(p_1 + p_2 + p_3) \leq k$. 

The unknowns to be determined in Equation (3.8) are the constant coefficients, \( D_{p_1 p_2 p_3} \), belonging to each term in the series, also referred to as the unknown derivatives. When determining the unknown derivatives, the following requirements must be satisfied by the \( k \)-exact reconstruction procedure [2]:

1) the solution reconstruction must represent polynomials of degree \( N \leq k \) exactly;
2) the solution reconstruction must preserve the average value in the cell to which it is applied.
3) the reconstruction procedure must have compact support.

### 3.5.1 Reconstruction Stencils

The determination of the unknown derivatives follows the same reasoning presented in Section 2.4.3. We use the information from neighbouring cells (that is, their average values) in conjunction with the constraint equations, to provide a system of equations which has at least as many equations as the number of unknown derivatives. To be clear, the term “reconstruction stencil” used herein, refers to the group of neighbouring cells (indexed by \( pqr \)) used in the reconstruction of cell \( ijk \).

Based on the second requirement of \( k \)-exact reconstruction (as listed above), we must preserve the cell average value for the cell being reconstructed (cell \( ijk \)) by imposing the following constraint:

\[
\bar{U}_{ijk} = \frac{1}{V_{ijk}} \iiint_{V_{ijk}} U_{ijk}^k(x, y, z) \, dV.
\]  

(3.9)

This constraint is not sufficient to determine all of the required unknowns, \( D_{p_1 p_2 p_3} \). As such, the unknown derivatives are found by extending the representation in cell \( ijk \) to its neighbouring cells \( pqr \) and imposing additional constraints in the following manner: the average value recovered by integrating \( U_{ijk}^k(x, y, z) \) over the given neighbouring cell domain, \( V_{pqr} \), should equal the known average value in that neighbouring cell, \( \bar{U}_{pqr} \). In this manner, the number of constraints imposed (and number of neighbours in the stencil) must be at least equal to the number of unknowns derivatives.
Note that the total number of unknown derivatives, $N$, is a function of the order of the reconstruction, $k$. In three-dimensional space the relationship is expressed as

$$N = \frac{(k + 1)(k + 2)(k + 3)}{6},$$

(3.10)

which results in the number of derivatives listed in Table 3.2. Note that here, the value for $N$ includes the cell-centered value (that is, the zero-derivative).

In practice, there are advantages to including information from a larger number of neighbouring cells than the minimum required. For example, using a larger stencil may make results of the reconstruction more robust in the case of stretched meshes and/or solution gradients that are not aligned with the mesh [2]. If this is done, the result is an overdetermined system of equations (more constraint equations than unknowns). The coefficients are then found using a least-squares formulation, as discussed in Section 3.5.2.

The third requirement of the $k$-exact reconstruction procedure concerns the number and locality of the neighbouring cells used to form the reconstruction stencil. The condition of “compact support” implies that the neighbouring cells should be physically near to the cell being reconstructed [45]. Since this work considers only centered stencils on structured grids, the stencil size may be described by the number of layers of cells, surrounding the cell $ijk$, which form the stencil. Current practice is to use one-layer of cells for the $k=0, 1,$ and 2 reconstructions, and two-layers of cells for the $k=3$ and 4 reconstructions. In three dimensions, a one-layer stencil consists of 27 cells ($3^3$), and a two-layer stencil consists of 125 cells ($5^3$); where both numbers include the cell $ijk$ that is being reconstructed. Figure 3.1 provides the numbering convention used to identify the cells that make up the reconstruction stencil. The count starts from zero at the cell being reconstructed (shown in green) and the positive $x$, $y$, and $z$ directions are identified by the $I$, $J$, and $K$ directions, respectively.

3.5.2 Determination of the Derivatives: The Least-Squares Problem

The goal of the remaining subsections of Section 3.5 is to present a more rigorous mathematical development of the overdetermined system of equations described qualitatively in the previous
Chapter 3. High-Order CENO Scheme

Section. The development is divided into three tasks:

i) forming the matrices of the overdetermined system,

ii) calculating the geometric coefficients and moments, and

iii) solving the overdetermined system in a least-squares sense.

We begin by considering the matrix form, $\mathbf{A} \mathbf{x} \neq \mathbf{b}$, where the inequality symbol is used intentionally in order to make clear that, for an overdetermined system of equations, a unique solution to $\mathbf{A} \mathbf{x} - \mathbf{b} = 0$ does not exist. The vector $\mathbf{x}$ here represents the vector of unknown derivatives, the length of which is actually less than the number of rows in $\mathbf{A}$ and length of $\mathbf{b}$, which represent the number of constraint equations. Since a unique solution does not exist, we instead make use of the solution to the linear least-squares problem. The linear least-squares problem may be defined as follows:

$$\text{Find the vector } \mathbf{x}, \text{ such that } ||\mathbf{A} \mathbf{x} - \mathbf{b}||^2 \text{ is minimized.} \quad (3.11)$$

Figure 3.1: Numbering convention used for the one and two-layer reconstruction stencils.
For convenience, we now introduce the vector $c$, which allows us to write our over-determined system of equations in the form of an equality: $Ax - b = c$. The vector $c$ is not actually formed prior to solving the least-squares problem, but is rather representative of the resulting error incurred by solving the least-squares problem. Note that the vector $c$ is of the same length as vector $b$ (or equivalently, the number of rows in matrix $A$). In this work, we make use of two different approaches for solving the least-squares problem: i) A Householder QR factorization method, and ii) The Pseudo-Inverse Matrix via Singular Value Decomposition (SVD). Both methods will be described in Section 3.5.5.

### 3.5.3 Forming the Overdetermined System, $Ax - b = c$

We begin forming the required matrices of the overdetermined system by considering some new notation amenable to our purpose. Let us label the cell being reconstructed as cell $i$ (formerly labeled $ijk$) and any neighbouring cell in the reconstruction stencil as cell $j$ (formerly labeled $pqr$). Note that the manner in which we form a set of neighbouring cells, indexed by $j$, is made clear by the numbering convention presented in Figure 3.1. Restating Equation (3.8), the reconstruction polynomial of cell $i$ may be expressed as

$$U_i^k(x, y, z) = \sum_{p_1=0}^{k} \sum_{p_2=0}^{k} \sum_{p_3=0}^{k} (x - x_i)^{p_1} (y - y_i)^{p_2} (z - z_i)^{p_3} D_{p_1p_2p_3}.$$  \hfill (3.12)

The average value in cell $i$ is defined by the strict constraint that

$$\bar{U}_i = \frac{1}{V_i} \iiint_{V_i} [U_i^k(x, y, z)] dV,$$  \hfill (3.13)

and all of the additional constraint equations (using neighbouring cells) are defined as

$$\bar{U}_j = \frac{1}{V_j} \iiint_{V_j} [U_i^k(x, y, z)] dV.$$  \hfill (3.14)

Substituting Equation (3.12) into Equation (3.14) we have

$$\bar{U}_j = \frac{1}{V_j} \iiint_{V_j} \left[ \sum_{p_1=0}^{k} \sum_{p_2=0}^{k} \sum_{p_3=0}^{k} (x - x_i)^{p_1} (y - y_i)^{p_2} (z - z_i)^{p_3} D_{p_1p_2p_3} \right] dV.$$  \hfill (3.15)

Noting that the unknown derivatives are simply constants (since they are evaluated at the cell centroid of cell $i$) we are free to manipulate Equation (3.15) by moving the derivatives out of the integrand, and by doing the same to the summations, we arrive at the following form:

$$\bar{U}_j = \sum_{p_1=0}^{k} \sum_{p_2=0}^{k} \sum_{p_3=0}^{k} \left[ D_{p_1p_2p_3} \frac{1}{V_j} \iiint_{V_j} (x - x_i)^{p_1} (y - y_i)^{p_2} (z - z_i)^{p_3} dV \right].$$  \hfill (3.16)
Equation (3.16) represents the reconstruction of cell $i$ integrated over the volume of the given neighbouring cell $j$. It is duly noted that the first term of the reconstruction polynomial is simply $D_{000}$, since the remaining terms of the integrand (when $p_1 = p_2 = p_3 = 0$) are all equal to unity. The term $D_{000}$ represents the value of the reconstruction polynomial evaluated at the cell center. Furthermore, since we are no longer necessarily using piecewise constant or piecewise linear polynomial reconstructions, we can no longer assume that the cell center value is equal to the cell average value; that is, in general,

$$D_{000} \neq \bar{U}_i.$$  \hspace{1cm} (3.17)

Let us now find an expression for $D_{000}$ in terms of the other derivatives, and in doing so we will effectively reduce the number of unknown derivatives (and constraint equations) by one. This is a necessary step in order to strictly ensure that the average value of cell $i$ is recovered when integrating its own reconstruction polynomial over its own cell domain (viz. the right hand side of Equation (3.13)).

Looking at cell $i$ we can re-write the Equation (3.13) for the cell average value as

$$\bar{U}_i = D_{000} + \sum_{p_1=0}^{k} \sum_{p_2=0}^{k} \sum_{p_3=0}^{k} D_{p_1 p_2 p_3} \cdot \frac{1}{V_i} \iiint_{V_i} (x - x_i)^{p_1} (y - y_i)^{p_2} (z - z_i)^{p_3} dV.$$  \hspace{1cm} (3.18)

Examining Equation (3.18) we can define the so-called geometric moments of cell $i$ taken about its own cell center as

$$\left(\tilde{x}^{p_1} y^{p_2} z^{p_3}\right)_i \equiv \frac{1}{V_i} \iiint_{V_i} (x - x_i)^{p_1} (y - y_i)^{p_2} (z - z_i)^{p_3} dV.$$  \hspace{1cm} (3.19)

For convenience, we may re-write Equation (3.18) in the following form:

$$\bar{U}_i = D_{000} + \sum_{p_1=0}^{k} \sum_{p_2=0}^{k} \sum_{p_3=0}^{k} D_{p_1 p_2 p_3} \left(\tilde{x}^{p_1} y^{p_2} z^{p_3}\right)_i.$$  \hspace{1cm} (3.20)

In doing so, we have arrived at the sought-after expression for $D_{000}$ in terms of the other derivatives:

$$D_{000} = \bar{U}_i - \sum_{p_1=0}^{k} \sum_{p_2=0}^{k} \sum_{p_3=0}^{k} D_{p_1 p_2 p_3} \left(\tilde{x}^{p_1} y^{p_2} z^{p_3}\right)_i.$$  \hspace{1cm} (3.21)

By similar reasoning, we can write the average value in cell $j$ as follows:

$$\bar{U}_j = D_{000} + \sum_{p_1=0}^{k} \sum_{p_2=0}^{k} \sum_{p_3=0}^{k} D_{p_1 p_2 p_3} \left(\tilde{x}^{p_1} y^{p_2} z^{p_3}\right)_j.$$  \hspace{1cm} (3.22)
where the term \( \left( x_{p1} y_{p2} z_{p3} \right)_j \) represents the geometric moments of cell \( j \) taken about the cell center of cell \( i \), and is defined as

\[
\left( x_{p1} y_{p2} z_{p3} \right)_j = \frac{1}{V_j} \iiint_{V_j} (x - x_i)^{p1} (y - y_i)^{p2} (z - z_i)^{p3} \, dV. \quad (3.23)
\]

If we substitute Equation (3.21) into Equation (3.22) we can get an expression for \( U_j - U_i \):

\[
U_j - U_i = \sum_{p1=0}^{k} \sum_{p2=0}^{k} \sum_{p3=0}^{k} D_{p1p2p3} \cdot \left( \left( x_{p1} y_{p2} z_{p3} \right)_j - \left( x_{p1} y_{p2} z_{p3} \right)_i \right), \quad (3.24)
\]

and defining,

\[
\left( x_{p1} y_{p2} z_{p3} \right)_{ij} \equiv \left( x_{p1} y_{p2} z_{p3} \right)_j - \left( x_{p1} y_{p2} z_{p3} \right)_i, \quad (3.25)
\]

we can write an expression which represents the system of equations to be solved:

\[
\begin{bmatrix}
(x_{0} y_{0} z_{1})_{i1} & \cdots & (x_{p1} y_{p2} z_{p3})_{i1} & \cdots & (x_{k} y_{0} z_{0})_{i1} \\
\vdots & \cdots & \vdots & \cdots & \vdots \\
(x_{0} y_{0} z_{1})_{ij} & \cdots & (x_{p1} y_{p2} z_{p3})_{ij} & \cdots & (x_{k} y_{0} z_{0})_{ij} \\
\vdots & \cdots & \vdots & \cdots & \vdots \\
(x_{0} y_{0} z_{1})_{iM} & \cdots & (x_{p1} y_{p2} z_{p3})_{iM} & \cdots & (x_{k} y_{0} z_{0})_{iM}
\end{bmatrix}
\begin{bmatrix}
D_{001} \\
\vdots \\
D_{p1p2p3} \\
\vdots \\
D_{k00}
\end{bmatrix}
= \begin{bmatrix}
(U_1 - U_i) \\
\vdots \\
(U_j - U_i) \\
\vdots \\
(U_M - U_i)
\end{bmatrix}
\begin{bmatrix}
c_1 \\
\vdots \\
c_j \\
\vdots \\
c_M
\end{bmatrix},
\]

Equation (3.26) represents an overdetermined system of equations only when the number of constraint equations becomes larger than the number of unknown derivatives, in which case the equality in Equation (3.26) is no longer strictly valid. In the overdetermined case, Equation (3.26) may be re-arranged and written in the sought-after matrix form, \( Ax - b = c \), as follows:

\[
M 	imes N \quad N \times 1 \quad M \times 1 \quad M \times 1
\]

where \( M \) represents the total number of neighbouring cells in the reconstruction stencil, and \( N \) represents the total number of unknown derivatives, minus one, since \( D_{000} \) has been eliminated by using Equation (3.21). Notice that the matrix \( A \) depends only on the mesh geometry, and the vector \( b \) depends only on the known cell average values. At this point we may also apply a
geometric weighting to each cell in our reconstruction stencil. This takes into consideration the distance between cell centers in the reconstruction stencil by applying greater weights to the neighbouring cells that are physically closer to cell $i$. The geometric weights, $w_{ij}$ are multiplied with each corresponding term in the matrix $A$ and the vector $b$. The two options currently implemented for the geometric weighting are the inverse distance and inverse distance squared, given by

$$w_{ij} = \frac{1}{|\vec{r}_j - \vec{r}_i|},$$  

(3.27)

or

$$w_{ij} = \frac{1}{|\vec{r}_j - \vec{r}_i|^2},$$  

(3.28)

respectively. Here $\vec{r}$ is the position vector of the indicated cell. Note that the geometric weighting is of greater importance when dealing with curved boundaries and with unstructured grids (neither of which is addressed in this work). We will soon discuss two methods for solving the overdetermined system of equations, however, in the next section we first explore a computationally efficient technique by which the elements of the matrix $A$ are calculated.

### 3.5.4 Calculating the Geometric Coefficients and Moments

Since the matrix $A$ depends only on the mesh geometry, we seek to compute its elements in a computationally efficient manner, taking into account both computational cost and memory usage. In theory we might pre-compute and store the entire matrix $A$ for every cell in the mesh in order to significantly reduce the computational cost of each reconstruction and hence, the cost of the overall scheme. However, as we shall see, the matrix $A$ is manipulated (by orthogonal decomposition) when used to solve the least-squares problem, and as such, it makes more sense to store the manipulated matrices (such as the pseudo-inverse matrix discussed in Section 3.5.5). So from a broad perspective, we have two options: i) pre-compute and store manipulations of the matrix $A$ for every cell, prior to time-marching, or ii) form the matrix $A$ for every cell, at every time step. The trade-off is between computational cost and the amount of required memory storage; the first option requires more memory, but is computationally cheaper (see Section 3.5.5 for more details).

Having digressed slightly, we can acknowledge that in either case, the matrix $A$ must be formed, and the proposed computationally efficient approach (described in this section) involves pre-computing and storing only the geometric coefficients of each cell about its own cell center. That is, the quantity $\left(\frac{x^{p_1} y^{p_2} z^{p_3}}{V_j}\right)_j$ given by

$$\left(\frac{x^{p_1} y^{p_2} z^{p_3}}{V_j}\right)_j \equiv \frac{1}{V_j} \iiint_{V_j} (x - x_j)^{p_1} (y - y_j)^{p_2} (z - z_j)^{p_3} dV.  \quad (3.29)$$


To re-iterate, these are coefficients that are pre-computed and stored regardless of whether we choose the first or second option above. If, say, the available memory resources prohibit the use of the first option, then the stored geometric coefficients become a significant cost-saver in the computation of the matrix $A$ at every time-step. We can make use of these stored coefficients to calculate the elements found in matrix $A$ by using the appropriate values for the powers $(p_1$, $p_2$, and $p_3)$ of $\left(\frac{x^{p_1}}{V_j}y^{p_2}z^{p_3}\right)$. In order to demonstrate how this is done, let us first substitute Equation (3.23) into Equation (3.19) so that we may write the elements of matrix $A$ as

$$
\left(\frac{x^{p_1}}{V_j}y^{p_2}z^{p_3}\right)_{ij} = \frac{1}{V_j} \int \int \int_{V_j} (x - x_i)^p_1 (y - y_i)^p_2 (z - z_i)^p_3 dV - \left(\frac{x^{p_1}}{V_j}y^{p_2}z^{p_3}\right)_{i}. 
$$

We are now going to re-express the geometry in terms of “geometric moments” about cell $j$, by making use of binomial expansions. To do so, we consider the following substitutions [46]:

\begin{align}
(x - x_i) &\Rightarrow (x - x_j) + (x_j - x_i), \\
(y - y_i) &\Rightarrow (y - y_j) + (y_j - y_i), \\
(z - z_i) &\Rightarrow (z - z_j) + (z_j - z_i),
\end{align}

into Equation (3.30). Using the relations above results in the following expression for the elements of matrix $A$:

$$
\left(\frac{x^{p_1}}{V_j}y^{p_2}z^{p_3}\right)_{ij} = \frac{1}{V_j} \int \int \int_{V_j} [(x - x_j) + (x_j - x_i)]^{p_1} [(y - y_j) + (y_j - y_i)]^{p_2} [(z - z_j) + (z_j - z_i)]^{p_3} dV - \left(\frac{x^{p_1}}{V_j}y^{p_2}z^{p_3}\right)_{i}. 
$$

In Equations (3.32), the terms of the integrand may be written in terms of binomial expansions as follows:

\begin{align}
(x - x_j) + (x_j - x_i) &\Rightarrow (x - x_j) + \Delta x_{ij} = \sum_{\xi=0}^{p_1} C_{p_1}^{\xi} \Delta x_{ij}^{\xi}(x - x_j)^{p_1-\xi}, \\
(y - y_j) + (y_j - y_i) &\Rightarrow (y - y_j) + \Delta y_{ij} = \sum_{\ell=0}^{p_2} C_{p_2}^{\ell} \Delta y_{ij}^{\ell}(y - y_j)^{p_2-\ell}, \\
(z - z_j) + (z_j - z_i) &\Rightarrow (z - z_j) + \Delta z_{ij} = \sum_{\varphi=0}^{p_3} C_{p_3}^{\varphi} \Delta z_{ij}^{\varphi}(z - z_j)^{p_3-\varphi},
\end{align}

Here we have defined $\Delta x_{ij} = (x_j - x_i)$, $\Delta y_{ij} = (y_j - y_i)$, and $\Delta z_{ij} = (z_j - z_i)$, and $C_{p_1}$, $C_{p_2}$, and $C_{p_3}$ are the coefficients of the binomial expansion (still to be defined). Substituting Equations
(3.33) into Equation (3.32), we arrive at a new expression for the elements of the matrix $A$:

\[
\begin{bmatrix}
  x^{p_1} y^{p_2} z^{p_3}
\end{bmatrix}_{ij} = \sum_{\xi=0}^{p_1} \sum_{\ell=0}^{p_2} \sum_{\varphi=0}^{p_3} \left[ C_{p_1}^{\xi} C_{p_2}^{\ell} C_{p_3}^{\varphi} \cdot \Delta x_i^{\xi} \Delta y_j^{\ell} \Delta z_j^{\varphi} \cdot \left( x (p_1 - \xi) y (p_2 - \ell) z (p_3 - \varphi) \right)_{ij} \right] - \left( x^{p_1} y^{p_2} z^{p_3} \right)_{ij}. 
\]

Upon examination of Equation (3.34), it is duly noted that the RHS no longer contains any geometric coefficients of neighbouring cells taken about cell $i$, but rather of neighbouring cells taken about their own cell centers; which is precisely what we have proposed to pre-compute and store. To clarify this point, notice that the term denoted \( \left( x (p_1 - \xi) y (p_2 - \ell) z (p_3 - \varphi) \right)_{ij} \) is defined by

\[
\left( x (p_1 - \xi) y (p_2 - \ell) z (p_3 - \varphi) \right)_{ij} \equiv \frac{1}{V_j} \iiint_{V_j} (x - x_j)^{p_1 - \xi} (y - y_j)^{p_2 - \ell} (z - z_j)^{p_3 - \varphi} dV.
\]

which is simply a shift in the powers of the stored \( x^{p_1} y^{p_2} z^{p_3} \) term. Hence, when computing the elements of matrix $A$, it is a straightforward matter of carrying out the summations on the RHS of Equation (3.34), by selecting the appropriate values for the powers of the pre-computed \( x^{p_1} y^{p_2} z^{p_3} \) belonging to the given neighbouring cell $j$. Note that this is significantly cheaper than carrying out the volume integrals on the RHS of Equation (3.29) for every neighbouring cell $j$ relative to cell $i$, and repeating the process for all cells $i$ in the solution domain.

Finally, we can compute the coefficients of the binomial expansion in a recursive manner, with initial coefficients $C_{p_1}^0 = C_{p_2}^0 = C_{p_3}^0 = 1$, as follows:

\[
C_{p_1}^{\xi} = \frac{p_1 - \xi + 1}{\xi} C_{p_1}^{\xi-1}, \quad C_{p_2}^{\ell} = \frac{p_2 - \ell + 1}{\ell} C_{p_2}^{\ell-1}, \quad C_{p_3}^{\varphi} = \frac{p_3 - \varphi + 1}{\varphi} C_{p_3}^{\varphi-1}. \tag{3.36}
\]

At this stage we have formed the required matrices that describe the overdetermined system of equations, $Ax - b = c$, and we are now ready to go ahead and solve the least-squares problem.

### 3.5.5 Solving the Least-Squares Problem

In this section we will briefly discuss the two methods employed in this work for solving the least squares problem; namely, the Householder QR factorization method, and the pseudo-inverse matrix method via singular value decomposition.
Householder QR Factorization

In the householder QR factorization method we re-write the matrix $A$ as a product of an orthogonal matrix, $Q$, and an upper-triangular matrix, $R$, such that

$$ A = QR. \quad (3.37) $$

There are different methods available for determining the matrices $Q$ and $R$, and in this work we make use of the Householder method. Such details, however, are omitted for the sake of brevity and may be found in the literature \[47, 48\].

Assuming that we have available $Q$ and $R$, we then pre-multiply the quantity that we wish to minimize, $||Ax - b||^2$, by $Q^T$ to get

$$ \min ||Q^T Ax - Q^T b||^2. \quad (3.38) $$

Substituting Equation (3.37) into Equation (3.38), we have

$$ \min ||Q^T QRx - Q^T b||^2. \quad (3.39) $$

Since $Q$ is orthogonal, its transpose equals its inverse and therefore its transpose multiplied by itself, $Q^T Q$, equals the identity matrix, and we have

$$ \min ||Rx - Q^T b||^2. \quad (3.40) $$

Furthermore, since $Q$ is orthogonal, the minimization of $||Rx - Q^T b||^2$ is equivalent to the minimization of $||Ax - b||^2$. In Equation (3.40) the matrix $Q^T$ is a square matrix with dimensions $M \times M$, and the matrix $R$ is partitioned into an $N \times N$ upper-right triangular matrix and an $(M - N) \times N$ zero block. Where $M$ and $N$ are defined by the number of equations and unknowns. Furthermore, it can be shown that the least squares solution is found by solving the first $N$ rows of the following system \[49\]:

$$ Rx - Q^T b = Q^T c. \quad (3.41) $$

Since the matrix $R$ is upper-triangular, the solution to Equation (3.41) by back-substitution is straightforward and computationally inexpensive. The resulting vector $x$, which is of length $N$, is such that we have minimized $||Ax - b||^2$. Using the above method gives more accurate solutions to the least-squares problem than forming the normal equations, especially for ill-conditioned matrices \[49\]; implying that this method adds robustness to the numerical scheme.

As a final note, although not yet implemented in the scheme, since the matrix $A$ depends only on the geometry, one could pre-compute and store the matrices $R$ and $Q^T$ in order to speed up the computation of the derivatives at each time-step. This would be at the expense, of course, of additional memory storage.
Pseudo-Inverse Matrix via Singular Value Decomposition

The second method implemented in this work for solving the least-squares problem considers the use of the so-called pseudo-inverse matrix, which is found by the orthogonal Singular Value Decomposition (SVD) of matrix $A$. The pseudo-inverse matrix, denoted $A^+$, which has dimensions $N \times M$, may be used to solve the least-squares problem as follows:

$$x = A^+ b,$$

(3.42)

where the resulting vector $x$, which is of length $N$, is such that we have minimized $||Ax - b||^2$. The SVD method used to obtain the pseudo-inverse matrix, consists of a decomposition of the matrix $A$, such that

$$A = Q \Sigma V^T,$$

(3.43)

where $Q$ is an $M \times M$ orthogonal matrix, $\Sigma$ is an $N \times N$ diagonal matrix with non-negative real values along the diagonal, and $V$ is an $N \times N$ orthogonal matrix. Note that the values on the diagonal of the matrix $\Sigma$ are the so-called “singular values” for which the method is named. The details of finding the matrices $Q$ and $V$ are omitted for brevity and may be found in the literature [48]. We may now calculate the pseudo-inverse by taking the inverse of the above decomposition. Since $Q$ and $V$ are orthogonal, their inverses equal their transpose, and since $\Sigma$ is a diagonal matrix, its inverse is obtained by simply taking the reciprocal of each non-zero diagonal value, such that

$$A^+ = V \Sigma^{-1} Q^T.$$

(3.44)

One of the advantages of the SVD method, which makes the method both robust and numerically stable, is that one can control the cut-off for the “non-zero” singular values on the diagonal of $\Sigma$; any value below the cut-off may be set to zero, rather than taking the reciprocal of a very small number (resulting in a very large number). Furthermore, since the pseudo-inverse depends only the geometry of the mesh, one can pre-compute and store it, making for extremely quick computation of the unknown derivatives. In studies carried out by Ivan and Groth [2], it was found that storing the pseudo-inverse “can provide a computational speed-up by a factor of about 5 for high-order two-dimensional Euler computations compared with the situation in which the coefficient matrix $A$ is formed and a least-squares subroutine is called for each spatial reconstruction,” at every time step [2].

The descriptions of the above two methods have been kept very brief, with the intent being to provide the reader with a general overview of the procedures used in this work for the solution to the least-squares problem. For additional details of these procedures the reader is referred...
to the literature [47, 50, 48]. As a final note, the results presented in Chapter 4 have been produced by the method of storing the pseudo-inverse matrix found via SVD.

### 3.6 The Smoothness Indicator

The smoothness indicator, developed by Ivan and Groth [6, 2], is calculated as part of a post-analysis step for every reconstructed variable, individually, within each cell. Note that at this stage, the reconstruction in each cell is complete and available for use. The smoothness indicator is based on the notion that the Taylor series approximation used in the $k$-exact reconstruction is only valid for smooth functions that are continuously differentiable up to some desired order of the polynomial approximation, $k$.

In this light, the smoothness indicator, $S$, is calculated based on a solution smoothness parameter, $\alpha$, as follows:

$$S = \frac{\alpha}{\max((1-\alpha), \epsilon)} \frac{(M - N)}{N - 1}.$$  \hspace{1cm} (3.45)

where, $\epsilon$ is a tolerance to avoid division by zero (taken as $10^{-8}$), and $M$ is the size of the stencil, $N$ is the number of unknown derivatives (or degrees of freedom). The solution smoothness parameter, $\alpha$, for cell $ijk$ and solution variable $U$ has the form:

$$\alpha = 1 - \frac{\sum_{pqr} (U_{pqr}^k(x_{pqr}, y_{pqr}, z_{pqr}) - U_{ijk}^k(x_{pqr}, y_{pqr}, z_{pqr}))^2}{\sum_{pqr} (U_{pqr}^k(x_{pqr}, y_{pqr}, z_{pqr}) - \bar{U}_{iijk})^2}. $$ \hspace{1cm} (3.46)

where, the cells in the stencil are indexed (as before) in the $x$, $y$, and $z$ directions by $p$, $q$, and $r$ respectively, and the commas between indices have been omitted for clarity. The various terms in Equation (3.46) are defined as follows:

- $U_{pqr}^k(x_{pqr}, y_{pqr}, z_{pqr})$: the reconstruction polynomial belonging to cell $pqr$ evaluated at the cell center of cell $pqr$
- $U_{ijk}^k(x_{pqr}, y_{pqr}, z_{pqr})$: the projected value of the reconstruction polynomial of cell $ijk$, obtained by extending $U_{ijk}^k(x, y, z)$ out to the cell center of $pqr$
- $\bar{U}_{iijk}$: the average value (or piecewise constant projection) of cell $ijk$.

Observe that the numerator in the second term of Equation (3.46) is the difference between an actual reconstructed value for cell center $pqr$ and the projected reconstructed value at that location from cell $ijk$. The denominator is a normalization assuming piecewise constant projection from cell $ijk$. Figure 3.2 depicts the variation of $\frac{\alpha}{1-\alpha}$, and the key point here, is that $\frac{\alpha}{1-\alpha}$ increases rapidly as $\alpha$ approaches unity [2]. Once $S$ has been calculated, it is compared
Chapter 3. High-Order CENO Scheme

Figure 3.2: Variation of the function $f(\alpha) = \frac{\alpha}{1 - \alpha}$.

to a pass/no-pass cutoff value, $S_c$ (with typical values of $S_c$ in the range of 1,000-5,000) [2]. Furthermore, when $S$ is calculated in smooth regions it is typically orders of magnitude larger than the cutoff value, as found by Ivan and Groth [2], and confirmed by numerical experiments in three dimensions. In this manner, the CENO reconstruction makes use of the smoothness indicator to control the reconstruction in the proposed hybrid approach.

Although in general all cells in the reconstruction stencil should be considered when evaluating the smoothness indicator, recent work by Ivan and Groth in two dimensions has shown that the smoothness indicator works well when only a single-layer of the stencil is considered (see Figure 3.1). That is to say that the neighbouring cells $pqr$ used to evaluate the smoothness of the solution, need not take into account all of the neighbouring cells required for the reconstruction of cell $ijk$. The same has been observed in the current three-dimensional case; the smoothness indicator performs in a robust fashion with a one-layer stencil. Although both the two-layer and one-layer options are made available in the three-dimensional CENO implementation, the one-layer stencil for the smoothness indicator is preferred in order to reduce the computational cost of the scheme (a point that will be made clear by our discussion of the boundary conditions in Section 3.8).

This section has focused on the theoretical development of the smoothness indicator. Of more import, perhaps, is the reliable and robust manner with which it operates; this is demonstrated in Chapter 4 and is the sole focus of Section 4.4.
3.7 Flux Evaluation

In this section we consider the evaluation of the inviscid and viscous fluxes employed by the high-order CENO scheme. Our development in Chapter 2 serves us well here; recall that the inviscid fluxes exhibit a hyperbolic mathematical behaviour, whereas the viscous fluxes are elliptic (Section 2.1.4). The semi-discrete form of the governing equations resulting from the finite-volume spatial discretization is given by Equation (2.12) and repeated here for convenience:

\[
\frac{dU}{dt} = -\frac{1}{\Delta x} \left\{ \left[ F_{x,i+\frac{1}{2},j,k} - F_{x,i-\frac{1}{2},j,k} \right] - \left[ F_{y,i+\frac{1}{2},j,k} - F_{y,i-\frac{1}{2},j,k} \right] \right\} v_x \\
- \frac{1}{\Delta y} \left\{ \left[ F_{y,i,j+\frac{1}{2},k} - F_{y,i,j-\frac{1}{2},k} \right] - \left[ F_{y,i,j+\frac{1}{2},k} - F_{y,i,j-\frac{1}{2},k} \right] \right\} v_y \\
- \frac{1}{\Delta z} \left\{ \left[ F_{z,i,j,k+\frac{1}{2}} - F_{x,i,j,k-\frac{1}{2}} \right] - \left[ F_{z,i,j,k+\frac{1}{2}} - F_{x,i,j,k-\frac{1}{2}} \right] \right\} v_z. 
\]

(3.47)

3.7.1 Inviscid Flux Evaluation

In Godunov-type FV schemes the reconstruction stage is “completely decoupled from the physical stage where the Riemann problems are solved at the interfaces of the cells” [13]. As such, we are free to employ the \( k \)-exact reconstruction technique, “without modifying the Riemann solver, in order to generate higher spatial approximations” [13]. That is to say that our development of the Riemann problem and its solvers in Chapter 2 remains unchanged. The above statements are applicable to high-order reconstructions on one condition: that the accuracy of the method by which we numerically integrate the fluxes at the cell interfaces, is such that it does not degrade the spatial accuracy of the scheme.

Gauss Quadrature Integration in Two Dimensions

In three-dimensional space the terms on the RHS of Equation (3.47) involve the numerical integration over each cell face; that is, the integral over the area defined by the given face. In this work we employ a Gauss-Legendre quadrature integration [51], with a sufficient number of Gauss quadrature points to maintain the high-order spatial accuracy of our scheme. Note that since this work considers Cartesian grids, every cell face forms a plane (that is, all four nodes bounding the given cell face lie in the same plane). As such, our objective is to carry out a two-dimensional Gauss quadrature integration, which is defined as follows (in the \( xy \)-plane) by
applying successively the formulas for one-dimensional integration [51]:

\[
I = \int_{y_a}^{y_b} \int_{x_a}^{x_b} \mathcal{F}(x, y) dx dy = \int_{y_a}^{y_b} \left[ \sum_{i=0}^{m} W_i^x \mathcal{F}(x_i, y) + E_x^m \right] dy \\
= \sum_{j=0}^{l} W_j^y \left[ \sum_{i=0}^{m} W_i^x \mathcal{F}(x_i, y_j) + E_x^m \right] + E_y^l \\
= \sum_{j=0}^{l} \sum_{i=0}^{m} W_j^y W_i^x \mathcal{F}(x_i, y_j) + E_{xy}^{lm}.
\]  

(3.48)

Here \( I \) is the result of the numerical integration, \( x_i \) and \( y_j \) are the locations of the quadrature points, and \( W_i^x \) and \( W_j^y \) are the associated weight factors in the \( x \) and \( y \) directions, respectively, and \( E_{xy}^{lm} \) is the combined error of the numerical integration. For our purposes, we use the same number of quadrature points in both directions. For \( k = 0 \) and \( k = 1 \) reconstructions it is sufficient to use only one Gauss quadrature point located at the center or mid-point of each cell-face [6, 46]. As illustrated in Figure 3.3, for \( k = 2, 3, \) and 4 reconstructions, in three dimensions, we use four Gauss quadrature points distributed symmetrically over the cell face (that is \( N_g = 4 \) and \( l = m = 2 \)). For all \( k \)-exact reconstruction orders and corresponding Gauss quadrature integrations, the order of accuracy of the scheme has been verified by numerical experiments. Note that the number of quadrature points in each direction determines the positions \( (x_i, y_j) \) and the associated weights as used in Equation (3.48). For a Cartesian cell with dimensions in the \( xy \)-plane of \( \Delta x \) and \( \Delta y \), and with four Gauss quadrature points, the relative...
positions of the quadrature points (with respect to the cell center) are \( x_i = \pm \frac{1}{2}(1 - \frac{1}{\sqrt{3}})\Delta x \) and \( y_i = \pm \frac{1}{2}(1 - \frac{1}{\sqrt{3}})\Delta y \) and the weights are taken as \( W_i^x = W_j^y = 1 \) [51].

### 3.7.2 Viscous Flux Evaluation

The high-order treatment of the elliptic operators is required in order to evaluate the viscous fluxes in the Navier-Stokes equations, given by Equations (2.4) through (2.7). Recall that the viscous fluxes depend on both the solution states and the solution gradients at the cell interfaces:

\[
\mathbf{F}_v \cdot \mathbf{n} = \mathbf{F}_v(\mathbf{U}_{left}, \nabla \mathbf{U}_{left}, \mathbf{U}_{right}, \nabla \mathbf{U}_{right}, \mathbf{n}, \eta = 0).
\]

(3.49)

The objective is then to obtain a consistent order of accuracy for both the hyperbolic and elliptic terms. The discretization procedure used in this work, follows directly from the work of Ivan and Groth in two dimensions [7]. In 2002, Ollivier-Gooch and Van Altena [46] proposed that a \( k \)-th order accurate gradient for the diffusive flux evaluation, can be obtained directly by differentiating the \( k \)-exact reconstruction polynomial that is readily available. At the quadrature points, the solution state and gradient are then calculated by taking their respective averages (the arithmetic means) of the left and right state and gradient values [7]. The result can then be used to evaluate all viscous flux terms at each quadrature point. The final step is to carry out the Gauss quadrature integration over each cell interface, as described in Section 3.7.1.

In this light, with the reconstruction polynomial defined as,

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

(3.50)

the terms of the left and right state gradients may be calculated - using the primitive solution variables - in the following manner:

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

(3.51a)

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

(3.51b)

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

(3.51c)
Chapter 3. High-Order CENO Scheme

49

Note that all $D_{000}$ terms drop out since the derivative of a constant is zero. Considering a cell interface in the $x$-direction the arithmetic means of both the solution states and gradients are obtained as follows:

$$U_{i+rac{1}{2},j,k} = \frac{(U_{\text{left}} + U_{\text{right}})}{2}, \quad (3.52)$$

$$\nabla U_{i+rac{1}{2},j,k} = \frac{(\nabla U_{\text{left}} + \nabla U_{\text{right}})}{2}. \quad (3.53)$$

As a result of calculating the above partial derivatives, Equations (3.51) have a leading truncation error term which are one order less than the $k^{th}$-order reconstruction polynomial, given by Equation (3.50). As such, in order to obtain a $k^{th}$-order accurate spatial discretization (for elliptic operators and the Navier-Stokes equations), we use a $k^{th}$-order reconstruction polynomial.

To reach our goal of fourth-order accuracy, we require the use of piecewise quartic reconstruction, and this is the reason that we have included the $k=4$ case in all prior discussions.

Ollivier-Gooch and Van Altana [46], and Ivan and Groth [7] have considered the application of the above elliptic discretization procedure to the Laplace operator, as well as the advection-diffusion equation [7]. Results have shown that the $k=2$ and $k=4$ cases converge at the expected orders (namely, 2 and 4). However, on uniform Cartesian grids, it was observed that error cancellations in the $k=1$ and $k=3$ cases result in $k+1$-order accuracy for these two cases. Nevertheless, the advantages of using quartic ($k=4$) reconstruction are not lost, since the studied test cases have also demonstrated that the magnitudes of the error when using quartic-reconstruction “are consistently lower than those of the cubic interpolants by at least one order” (the order of the magnitude is not to be confused with the order of convergence of the error, see Section 4.2 for clarification). Furthermore, the analysis by Ivan and Groth indicate that “this behaviour can only really be expected on Cartesian grids, for which error cancellation occurs, and not on arbitrary grids” [52]. Unlike Ollivier-Gooch and Van Altana, Ivan and Groth make sure to recover the correct order of accuracy, by always increasing the order of the reconstruction scheme by one order, that is $k^{th}$-order accurate reconstruction when specifying a desired order of accuracy of $k$ for the Navier-Stokes calculations. Recall that for the $k=4$ reconstruction, the reconstruction stencil is still only two-layers and the number of Gauss quadrature points is still just four. So the additional computational cost required is due primarily to finding the additional fifteen derivatives (see Table 3.2). Also note that the number of Gauss quadrature points need not be increased for the $k=4$ case, since the order of the accuracy of our scheme remains at most fourth-order.
3.8 Boundary Conditions

In general, the high-order treatment of boundary conditions is essential if any scheme is to maintain its global high-order accuracy. In the development of the CENO scheme, Ivan and Groth have investigated different means for addressing this issue on straight and curved boundaries, including:

1. The use of a constrained reconstruction technique [46] to impose the physical boundary conditions while using a modified (or one-sided) stencil. Here we force the reconstructed polynomials to recover certain values at certain locations along the boundary.

2. Making use of the so-called “ghost-cells” (additional layers of cells extending beyond the physical solution/block domain), in which we define average values using the specified type of boundary condition. Figure 3.4 illustrates a two-dimensional multi-block grid with two layers of ghost cells. To visualize the three-dimensional case, the reader may consider a “box within a box”, where the interior box represents the physical solution domain of interest. The three-dimensional single-block case is depicted in Figure 3.6.

Although the constrained reconstruction technique is a future consideration in three dimensions, this work employs the second method; the use of ghost cells onto which the appropriate boundary conditions are applied. To this end, we must add a sufficient number of “layers” of ghost
cells such that the flux along the boundary of the physical solution domain remains to be function of both a left-state and a right-state provided by the piecewise polynomial reconstructions. The question is then: How many ghost cell layers are required?

The answer depends on the size of the reconstruction stencil and the size of the stencil used in evaluating the smoothness indicator (as discussed in Sections 3.5.1 and 3.6, respectively). Assuming that we are using a two-layer stencil for the “high-order” reconstruction (that is, \( k \geq 2 \)) then the total number of ghost cell layers required at the boundary of the solution domain is five (for a two-layer smoothness indicator stencil), and four (for a one-layer smoothness indicator stencil). It is instructive to understand how we arrive at these numbers; consider the Figure 3.5, representing the ghost cells extended out in a single direction. The dashed red line indicates the boundary of the physical solution domain, and notice that the right-state (and its gradient) must comes from the reconstruction in the first layer of the ghost cells, labeled as “Ghost Cell 1” in Figure (3.5). The purpose of each ghost cell is then described as follows:

**Ghost Cell 1:** We need to reconstruct this cell in order to provide the flux functions with the right-side (R) solution state (and its gradient).

**Ghost Cells 2 and 3:** These cells are used to form the high-order two-layer reconstruction stencil belonging to Cell 1.

**Ghost Cells 4 and 5:** These cells are used to calculate the smoothness indicator in Cell 1. This may seem disconcerting at first, but consider the fact that the smoothness parameter, \( \alpha \), given by Equation (3.46) must make use of the reconstruction polynomials of both Cells 2 and 3 (for a two-layer smoothness indicator stencil). Therefore, we must reconstruct Cells 2 and 3. So in essence, we can describe the outermost Cells 4 and 5 as belonging to the reconstruction stencil of Cell 3, which in turn, allows us to compute the smoothness indicator in Cell 1. Furthermore, if the smoothness indicator only makes use of a single layer stencil (as discussed in Section 3.6), then Ghost Cell 5 is no longer required.
Note that there is a significant difference between four and five ghost cells in terms of the total number of extra cells required; for example, on a $100 \times 100 \times 100$ single-block mesh, the increase in the total number of required cells is 71,288, or 5.7% (calculated as calculated as $\frac{110^3-108^3}{108^3}$). In the multi-block case (described in the next section), this matter becomes even more prominent (since each block will require its own set of ghost cells). As such, the advantage of using a one-layer stencil for the smoothness indicator is particularly evident in three dimensions. In Chapter 5, alternative methods for further reducing the number of ghost cells are discussed as recommendations for future research.

For the test cases used to verify the high-order CENO implementation in three dimensions, three types of boundary conditions are considered: periodic, reflection, and constant extrapolation. Each of these boundary conditions are calculated based on the cell average values of the interior cells (within the solution domain of interest) and applied to the ghost cells on the exterior. The parallel multi-block implementation, discussed in the next section, ensures that the correct boundary conditions are applied only to the blocks which make up the solution domain’s boundary. Care is also taken to resolve any potential conflicts in assigning the boundary conditions to the ghost cells which extend out from the twelve edges and eight corners of the solution domain.

### 3.9 Parallel Multi-Block Implementation

In this final section on the development of the high-order CENO scheme we briefly consider the parallel multi-block implementation. In the multi-block grid, the solution domain is partitioned into several smaller blocks that can be farmed out to multiple processors (with more than one block permitted on each processor) [53]. Figure 3.6 depicts a $2 \times 2 \times 2$ multi-block hexahedral domain decomposition in three dimensions, and highlights the structure of the interior cells and overlapping ghost cells by comparison to the two- and three-dimensional single-block cases. This domain decomposition is also amenable to Adaptive Mesh Refinement (AMR). A parallel implementation of the block-based AMR scheme has been developed using the C++ programming language and the MPI (message passing interface) library for the three-dimensional solvers [10, 53]. In this work, all block boundaries (both interior and exterior) are handled by the addition of ghost cells; where the exterior ghost cells are handled by the boundary conditions discussed in the previous section. For the interior blocks boundaries, the ghost cells overlap with the physical cells in adjacent blocks, as depicted in both Figures 3.4 and 3.6. At every stage of the multi-stage time integration procedure (that is, after the interior physical cells have been updated), we then update the average values in the ghost cells via MPI [53]. Although
there may be overlap between the ghost cells of multiple blocks, it is duly noted that any interior ghost cell can only occupy the same space as one physical cell, and it is from this physical cell that the information is copied. Note that the above description is not strictly true when making use of AMR, where adjacent blocks may have different mesh densities, see for example, [10]. Finally, we acknowledge that for the high-order CENO scheme, the only modification required to the existing parallel multi-block implementation has been the message passing of the solution and geometry associated with the additional ghost cells.
Figure 3.6: Depiction of ghost cells for the 2D single-block quadrilateral grid, and the 3D single-block and multi-block hexahedral grids.
Chapter 4

Verification and Numerical Results

The verification and validation of CFD solutions and/or simulations are important components of solution algorithm development, in order to establish credibility and performance of newly proposed numerical methods [10]. The verification process of CFD solutions has been defined in the AIAA guidelines [54] as the “process of determining that a model implementation accurately represents the developer’s conceptual description of the model and the solution to the model” [54]. Defined differently, is the validation process, which is the “process of determining the degree to which a model is an accurate representation of the real world from the perspective of the intended uses of the model” [54]. In this chapter, we present the verification of some key aspects of the high-order CENO scheme and the numerical results from selected test cases. The validation process for specific applications will be an important part of future follow-on research.

Studies by Ivan and Groth [5, 6, 2, 7] have previously verified the high-order CENO scheme as applied to various test cases in one- and two-dimensional space. Several of the test cases used in this chapter have been inspired from this work. The previous verification of Ivan and Groth also adds credibility to the scheme, established even prior to the outset of this research.

We begin this chapter with a brief description of the accuracy assessment methods and convergence studies used to evaluate the numerical error and order of accuracy of the scheme. Next we consider the reconstruction of functions, which serve to verify that we are successfully achieving up to fourth-order accuracy for smooth functions, and also that the smoothness indicator is functioning in a robust manner when applied to both discontinuous and under-resolved functions. Next we consider three unsteady test cases applied to the Euler equations, including: the one-dimensional shock-tube problem, three-dimensional shock-box problem, and advection of a smooth periodic density function used to demonstrate the savings in computational cost achieved by using the fourth-order CENO scheme (as compared to a second-order scheme).
Finally, we present the application to the Navier-Stokes equations, via the combination of the high-order CENO scheme with the Large Eddy Simulation (LES) of compressible turbulent flows.

### 4.1 Accuracy Assessment Methods

Since the main objective of this work is to obtain high-order spatial accuracy in three dimensions, it makes sense that the verification process of the CENO scheme involves the selection of test cases for which the exact solutions are available. As such, we are able to calculate the $L_1$, $L_2$ and $L_\infty$ error norms for the purpose of evaluating the numerical error of the scheme. These error norms are defined as follows:

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

(4.1)

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

(4.2)

$$L_\infty = |E|_\infty = \max \left\{ \frac{1}{V_T} \iiint_{V_{ijk}} \left[ U^k_{ijk}(x,y,z) - U_{\text{exact}}(x,y,z) \right] dV \right\} \text{ over all cells } ijk,$$  

(4.3)

where $V_T$ is the total volume of the solution domain. The above integrals are evaluated numerically using an adaptive Gaussian quadrature integration scheme, from which an accurate numerical result can be obtained to a desired tolerance (taken as eight digits in this work). Although this numerical integration process can be computationally expensive, particularly in three dimensions, the accuracy assessments of the numerical solutions are only performed as a post-processing step. For more specific details regarding the adaptive Gaussian quadrature integration scheme, the reader is referred to the literature [55].

### 4.2 Convergence Studies

A convergence study is carried out by incrementally refining the mesh (and recomputing the solution and the corresponding error) so that we can determine the “order of accuracy” of the scheme for the test case of interest. Graphically, we can plot the error-norms versus the number of cells - on a log-log scale - and the order of accuracy of the scheme will then correspond to the slope of the best-fit line going through the plotted data. Note that, provided that the asymptotic limit has been reached, the magnitude of the numerical error is proportional to the grid spacing, $\Delta$, such that Error $\propto \Delta^m$, where $m$ is the order of the scheme. Taking
the logarithm of both sides, we have \( \log(\text{Error}) \propto m \log(\Delta) \), where here, \( m \) also corresponds to the slope of the resulting linear log-log relationship. Note that the order of accuracy, on its own, does not indicate the magnitude of the error, it only indicates the rate at which the magnitude of the error will decrease as the mesh is refined. Furthermore, note that convergence studies typically involve more than two data points, and in this work, we make use of a least-squares approximation to the straight line that “best-fits” the data points, and in turn, best-approximates the slope of the line, and hence, the order of accuracy of the CENO scheme for the problem of interest.

### 4.3 Demonstration of High-Order Accuracy

In order to demonstrate that the expected theoretical orders of convergence are achieved in three dimensions with the proposed hybrid reconstruction scheme, the application of the CENO scheme to the reconstruction of various functions is considered in this section. To this end, this section shall consider smooth functions. The reconstruction of discontinuous and under-resolved functions, with the purpose of verifying the functionality of the smoothness indicator, will be presented in Section 4.4. Note that by reconstructing specified functions, the reconstruction stage is isolated from the flux evaluation and time-evolution stages.

Given an exact representation of the given function, denoted as \( U_{\text{exact}}(x, y, z) \), we must first obtain accurate values for the cell averaged solution as follows:

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

from which we can then reconstruct the solution. The integration of the RHS of Equation (4.4) is carried out numerically to the desired level of accuracy using the adaptive Gaussian quadrature integration scheme discussed in Section 4.1.

We also note that the maximum grid resolution under consideration will be \( 128 \times 128 \times 128 \) cells in the \( x \), \( y \), and \( z \) coordinate directions, respectively. This grid consists of over two million three-dimensional hexahedral cells, making the accuracy assessment of the reconstructed solution somewhat computationally expensive and time-consuming. However, limiting the number of cells to only 128 in each coordinate direction does not, in practice, necessarily achieve the asymptotic limits of the convergence orders. Nevertheless, due to the time constraints of this thesis, more refined grids were not investigated. It should be noted, however, that Ivan and Groth have studied several similar test cases in which the asymptotic limits were reached using the one- and two-dimensional high-order CENO schemes. In addition, test cases representing...
one-dimensional problems have been studied using the CENO scheme in its three-dimensional implementation. In these test cases it was possible to refine the grid (in only one coordinate direction) to achieve the asymptotic limit of the convergence orders. For brevity, we shall not present the reconstruction of one-dimensional functions herein, but rather focus on the convergence orders obtained by applying the high-order CENO scheme to the reconstruction of three-dimensional functions. As we evaluate the results, it is important to keep in mind the aforementioned factors regarding grid resolution and the asymptotic limits of the convergence orders.

Unless otherwise stated, throughout this chapter we make use of the Venkatakrishnan limiter, a smoothness indicator stencil consisting of only one layer of cells (as discussed in Section 3.6), a smoothness indicator cut-off value of $S_c = 1500$, and a geometric weighting of $\frac{1}{\Delta r}$ (as discussed in Section 3.5.3).

### 4.3.1 Spherical Cosine Function

We first consider the reconstruction of the spherical cosine function, having smooth solution variation in all three coordinate directions. The exact solution in this case is given by

$$U_{\text{exact}}(x, y, z) = 1 + \frac{1}{3} \cos(r),$$

where $r = \sqrt{(x + 5)^2 + (y + 5)^2 + (z + 5)^2}$ is the radial position. Due to the symmetry of the problem, we have modeled only one eighth of the spherical cosine function. The shift in radial position allows the $10 \times 10 \times 10$ solution domain to remain centered on $x = y = z = 0$ for convenience.

In Figure 4.1 the contour plots are presented for the exact solution as well as the reconstructed solutions using the first-order and fourth-order CENO methods. Notice the increased resolution achieved by using the fourth-order CENO scheme as compared to a first-order scheme. Figure 4.2 shows the results from the convergence study carried out for this first test case, using the series of mesh refinements from $32^3$, to $64^3$, to $96^3$, and finally $128^3$. The slopes (and hence, the orders of accuracy) of the various reconstructions and error-norms are presented in Table 4.3.1. These results verify that the expected orders of accuracy are obtained by using the corresponding order of reconstruction; that is, $k = 0$ for first-order, $k = 1$ for second-order, $k = 2$ for third-order, and $k = 3$ for fourth-order accuracy. The values of the slopes have been calculated using the best-fit line through the last three data-points (as we attempt to reach the asymptotic limit).

It can be observed in Figure 4.2 that, for the coarsest $8^3$ grid, the magnitude of the errors of
the schemes converge to the same point. In this case, the smoothness indicator has flagged all of the cells as unfit for unlimited $k^{th}$-order reconstruction (due to the fact that the grid is under-resolved) and the scheme has reverted to a limited piecewise linear reconstruction in all of the cells. In Table 4.3.1, we present the number of unfit cells flagged by the smoothness indicator in relation to the total number of cells, and the order of the scheme. Notice that as the order of the CENO schemes is increased, the number of cells flagged as “unfit” is reduced more rapidly as the mesh is refined.

(a) Exact solution.  
(b) First-order CENO ($k = 0$) reconstruction.  
(c) Fourth-order CENO ($k = 3$) reconstruction.

Figure 4.1: Reconstruction of spherical cosine function.
Figure 4.2: Convergence studies for spherical cosine function reconstruction.

Table 4.1: Orders of accuracy for the spherical cosine function reconstruction.
4.3.2 A Smooth 3D Function

Let us now consider the reconstruction of a second smooth function in three dimensions, which is a combination of sine and cosine functions. This function represents a more complex and more interesting test case than the spherical cosine function previously considered. We define the smooth function in this case as follows:

\[ U_{\text{exact}}(x, y, z) = 5 + \cos^2(2\pi x) + 3\sin(2\pi y) + 0.5\cos(2\pi z). \]  

Here the solution domain is taken as \(10 \times 10 \times 10\) centered on \(x=y=z=0\).

Figure 4.3 presents the contour plots of the exact solution, first-order, and fourth-order CENO reconstructions. Here, two iso-surfaces (at values of 2 and 9) have been included to highlight some features of the interior solution distribution. Figure 4.4 shows the convergence studies carried out for this test case. Again, the second-, third-, and fourth-order schemes converge to the same point on the coarsest \(8^3\) grid, for exactly the same reasons discussed in Section 4.3.1. The slopes resulting from the convergence studies are listed in Table 4.3.2. As in the previous test case, these results demonstrate that the expected theoretical orders of convergence are achieved in three dimensions with the proposed hybrid reconstruction scheme. Here we also present, in Table 4.3.2, the number of unfit cells flagged by the smoothness indicator in relation to the total number of cells, and the order of the scheme. As we refine the mesh, the same behaviour is again observed, in that the number of cells flagged as “unfit” begins to drop-off much more rapidly with increasing order of the CENO scheme.
Chapter 4. Verification and Numerical Results

(a) Exact solution.  
(b) First-order CENO \((k = 0)\) reconstruction.  
(c) Fourth-order CENO \((k = 3)\) reconstruction.

Figure 4.3: Reconstruction of the smooth 3D function.

| Slopes | \(k \) | \(|E|_1\) | \(|E|_2\) | \(|E|_\infty\) |
|--------|-----|-----|-----|-----|
|        | 0   | -1.000 | -1.000 | -0.998 |
|        | 1   | -2.025 | -2.025 | -1.999 |
|        | 2   | -2.993 | -2.993 | -2.989 |
|        | 3   | -4.013 | -4.007 | -3.989 |

Table 4.3: Orders of accuracy for the smooth 3D function reconstruction.
Figure 4.4: Convergence studies for the smooth 3D function reconstruction.

4.4 Demonstration of Smoothness Indicator

In this section, we consider the reconstruction of analytical functions which includes discontinuities and for which the grid is under-resolved. Our objective is to verify that the smoothness indicator is in fact capable of identifying such regions, and that the switching to a limited piecewise linear reconstruction is carried out as designed in the proposed hybrid reconstruction procedure.
Table 4.4: Smooth 3D function reconstruction: Number of cells flagged by the smoothness indicator as unfit for unlimited $k^{th}$-order reconstruction ($S_c=1,500$).

<table>
<thead>
<tr>
<th>Grid</th>
<th>Total</th>
<th>Second-Order CENO Unfit</th>
<th>Second-Order CENO Unfit %</th>
<th>Third-Order CENO Unfit</th>
<th>Third-Order CENO Unfit %</th>
<th>Fourth-Order CENO Unfit</th>
<th>Fourth-Order CENO Unfit %</th>
</tr>
</thead>
<tbody>
<tr>
<td>$8 \times 8 \times 8$</td>
<td>512</td>
<td>512</td>
<td>512</td>
<td>512</td>
<td>512</td>
<td>512</td>
<td>512</td>
</tr>
<tr>
<td>$16 \times 16 \times 16$</td>
<td>4,096</td>
<td>3,744</td>
<td>91.4</td>
<td>1,536</td>
<td>37.5</td>
<td>128</td>
<td>3.1</td>
</tr>
<tr>
<td>$32 \times 32 \times 32$</td>
<td>32,768</td>
<td>21,312</td>
<td>65.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$64 \times 64 \times 64$</td>
<td>262,144</td>
<td>100,288</td>
<td>38.3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$96 \times 96 \times 96$</td>
<td>884,736</td>
<td>209,376</td>
<td>23.7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$128 \times 128 \times 128$</td>
<td>2,097,152</td>
<td>284,320</td>
<td>13.6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

4.4.1 Abgrall Function

The Abgrall function [33] is a function that, by design, has number of solution discontinuities, and presents a significant challenge for any scheme hoping to enforce solution monotonicity in its reconstruction. For our purposes, the reconstruction of this function is used to demonstrate and verify the reliability and robustness of the smoothness indicator. The Abgrall function is defined in two dimensions as follows [33]:

$$U_{\text{exact}}(x, y) = \begin{cases} 
  f(x - \cot \sqrt{\frac{\pi}{2}} y) & x \leq \cos \frac{\pi y}{2}, \\
  f(x + \cot \sqrt{\frac{\pi}{2}} y) + \cos(2\pi y) & x > \cos \frac{\pi y}{2},
\end{cases} \quad (4.7)$$

with

$$f(r) = \begin{cases} 
  -r \sin \left(\frac{3\pi}{2} r^2\right) & r \leq -\frac{1}{3}, \\
  \left|\sin(2\pi r)\right| & |r| < \frac{1}{3}, \\
  2r - 1 + \frac{1}{6} \sin(3\pi r) & r \geq \frac{1}{3},
\end{cases} \quad (4.8)$$

where

$$r = \sqrt{x^2 + y^2}. \quad (4.9)$$

To reconstruct this function in a $1 \times 1 \times 1$ solution domain using the three-dimensional CENO implementation, we have extruded the function along the $z$-axis such that the $xy$-plane distribution provided by the Abgrall function (which is independent of $z$) is constant in the $z$ direction. Figure 4.5(a) shows the contour plot of the exact solution in the $xy$-plane. To observe the complexity of the function, notice the sharp discontinuities surrounding the smoother regions. A first and fourth-order reconstruction of the Abgrall function are presented in Figures 4.5(b) and 4.5(c), on a $256 \times 256 \times 4$ grid. In Figure 4.5(d), the regions in red identify the non-smooth
solution content, flagged by the smoothness indicator in the fourth-order CENO scheme, using a smoothness indicator cut-off value of \( S_c = 4000 \). In these red regions the scheme reverts to a limited piecewise linear reconstruction in order to enforce the solution monotonicity. We can acknowledge that the enforcement of solution monotonicity is particularly challenging in this case, due to the number and distribution of discontinuities that arise from the definition of the Abgrall function. The high-order CENO scheme appears to be quite capable of meeting this challenge by making use of the smoothness indicator (as demonstrated in Figure 4.5(d)) and the proposed hybrid reconstruction procedure.

Figure 4.5: Reconstruction and smoothness indicator of the Abgrall function on a 256\( \times \)256\( \times \)4 grid.

If we now turn our attention to the order of accuracy that results when the reconstruction
switching occurs, we discover an important aspect of the hybrid reconstruction technique; regardless of the order of reconstruction, we always obtain first-order accuracy. In such cases, unlike ENO and WENO schemes, the CENO scheme does not achieve (or strictly attempt to achieve) uniform high-order accuracy throughout the domain. The global order of accuracy of the high-order hybrid scheme will be dictated by the lowest order of reconstruction used in the solution. In the red regions flagged by the smoothness indicator, the use of limiters (when reverting to a piecewise linear reconstruction) will effectively apply a piecewise constant first-order reconstruction in order to enforce solution monotonicity in the vicinity of the discontinuities. Although this may seem to negate the advantages of using the high-order CENO scheme, Figure 4.6 tells quite a different story. Although the lines all have similar slopes, they are in fact shifted downwards as the order of the reconstruction is increased, indicating an overall reduction in the magnitude of the reconstruction error. From a practical standpoint this demonstrates that for a desired level of accuracy (chosen arbitrarily here and illustrated by the horizontal dashed line in Figure 4.6), we require significantly fewer cells, when we make use of increasingly higher orders of reconstruction. In this example, the first-order scheme requires approximately 320 cells in both the $x$ and $y$ directions, while the fourth-order scheme requires only 100 cells. If we cube these numbers, by assuming what may be typically required in each direction for a challenging three-dimensional problem, and take the percentage difference, we see a $\frac{(320^3 - 100^3)}{320^3} \times 100 \approx 97\%$ reduction in the total number of cells required. The percentage difference in the total number of cells required between the second-order and fourth-order schemes would then be $\frac{(110^3 - 100^3)}{110^3} \times 100 \approx 33\%$.

The relative shift downwards between the schemes can be attributed to the form of the function, or more specifically, to the abundance of discontinuous and under-resolved regions, and also to the solution variation in smooth-regions. If one were to consider a function with even more discontinuities and non-smooth features than the Abgrall function, one would expect fewer differences between the error magnitudes of the schemes, and the same is true in the opposite sense. As a final note, although the computational cost associated per cell increases with the order of the reconstruction, this additional cost is offset by the significant reduction in the total number of cells required. In Section 4.5.3 we will quantitatively compare the computational costs associated with a second-order limited piecewise linear scheme and the fourth-order CENO scheme in the consideration of the advection of a periodic density wave.
4.4.2 Hyperbolic Tangent Function

To further demonstrate the behaviour and capabilities of the smoothness indicator, it is instructive to consider the third-order CENO reconstruction of the hyperbolic tangent function in one dimension, given by

\[ U_{\text{exact}}(x, y, z) = 2 + \tanh\left(\frac{1}{2}x\right). \]  

(4.10)

Figure 4.7(a) shows the hyperbolic tangent function and the specific region (outlined in red) in which we will focus our attention. For this test case, we incrementally reduce the number of cells in the solution domain, as shown in the sequence of Figures 4.7(b) to 4.7(f). As we continuously coarsen the grid, eventually the grid becomes under-resolved (or too coarse), and the smoothness indicator value, \( S \), will become less than the cut-off, \( S_c = 4000 \), as indicated. In the under-resolved cases, the smoothness indicator effectively sees the solution as discontinuous, and in turn, flags the region as unfit for high-order reconstruction. It can be seen in Figure 4.7(e), and even more clearly in Figure 4.7(f), that the reconstruction scheme has in fact switched to a piecewise linear reconstruction.

This test case illuminates one of the most difficult situations that must be addressed when considering the spatial discretization of a given solution domain; that is, the situation in which we do not know, \textit{a priori}, how dense our mesh should be in order to resolve the solution.
content of our fluid flow problem. In the CENO scheme, if the initial mesh turns out to be under-resolved, then the smoothness indicator facilitates the switch to limited piecewise linear reconstruction. This feature of the smoothness indicator is particularly well suited for AMR, where the mesh is initially and intentionally coarse (and under-resolved) and then subsequently refined, where ever needed, based on the solution content. The combination of AMR with the high-order CENO scheme in the three-dimensional implementation is a near future consideration for this work, the benefits of which have been demonstrated by Ivan and Groth for the two-dimensional scheme \[2, 6\].

4.5 Application to the Euler Equations

Having successfully verified the CENO reconstruction scheme on a variety of functions, we now consider the application of the CENO scheme to the compressible form of the Euler equations in three dimensions.

4.5.1 1D Shock-Tube Problem

It is instructive to first consider the one-dimensional shock-tube problem, making use of the three-dimensional implementation of the high-order CENO scheme. This test-case shall serve to verify the inviscid flux evaluation, which includes a Gauss-quadrature integration over the cell, and also serves to further verify the $k$-exact reconstruction procedure and smoothness indicator, applied to an unsteady fluid flow problem. Here we consider a shock tube problem with a pressure ratio of $\frac{p_L}{p_R} = 4$ and the following initial conditions: $\rho_l = 4.696 \text{ kg/m}^3$, $u_l = v_l = w_l = 0 \text{ m/s}$, $p_l = 404.4 \text{ kPa}$, $\rho_r = 1.408 \text{ kg/m}^3$, $u_r = v_r = w_r = 0 \text{ m/s}$, $p_r = 101.1 \text{ kPa}$, and $0 \leq t \leq 7\text{ms}$, taken from \[25\].

To solve this problem using the three-dimensional CENO implementation, on a 128x4x4 grid, we use a constant extrapolation of the solution in both the y and z directions. The simulation is run on a $10 \times 10 \times 10 \text{ m}$ domain for 7 ms, and the scheme makes use of Roe’s approximate Riemann solver, the Venkatakrishnan slope limiter, and a smoothness indicator cut-off value of $S_c = 4000$. Figure 4.8 shows the density variation in the $x$-direction for both the third- and fourth-order CENO schemes, with and without the hybrid switching to limited piecewise linear reconstruction. It can be observed that when the hybrid switching is active (facilitated by the smoothness indicator), the spurious solution oscillations are significantly reduced. This is indicative of the scheme’s ability to enforce monotonicity in the vicinity of solution discontinuities and shock waves. Although not presented here, the one-dimensional shock tube problem was
Figure 4.7: Resolved and under-resolved grids for the hyper-tangent function.

also tested in both the $y$ and $z$ directions in order to verify the consistency of the smoothness indicator and solution content in each direction. Such consistency shall be more clearly
demonstrated using the shock-box problem in the next section.

Figure 4.8: Density variation: shock tube problem: $t = 7ms, 128 \text{ cells}, CFL = 0.7$. 
4.5.2 3D Shock-Box Problem

In this section we consider the application of the high-order CENO scheme to a three-dimensional shock-box problem. The initial conditions for the shock-box are such that one quadrant of the cube (that is one-eighth of the three-dimensional solution domain) is set equal to the so-called left-state, with the remaining quadrants of the domain are set to the so-called right state. For \( x \leq 0, \ y \leq 0, \ z \leq 0 \), the solution is initialized to the left-state, defined as

\[
\begin{align*}
\rho_l &= 1.225 \text{ kg/m}^3, \\
u_l &= v_l = w_l = 0 \text{ m/s}, \\
p_l &= 101.325 \text{ kPa},
\end{align*}
\]

and the remainder of the \( 10 \times 10 \times 10 \) m domain is initialized to right-state, defined as

\[
\begin{align*}
\rho_r &= 9.8 \text{ kg/m}^3, \\
u_r &= v_r = w_r = 0 \text{ m/s}, \\
p_r &= 1,013.25 \text{ kPa}.
\end{align*}
\]

For this test case, we make use of Roe’s approximate Riemann solver, the Venkatakrishnan slope limiter, and a smoothness indicator cut-off value of \( S_c = 4000 \). The boundary conditions for this case are set everywhere to that of constant-extrapolation, such that the solution content in the ghost cells is set equal to the interior cells forming the boundary of the solution domain. The simulation is then run for 6 ms, such that the propagating waves interact with each other, but are not given enough time to propagate out of the solution domain’s boundaries.

Figures 4.9, 4.10, and 4.11 show the variation in density, pressure, and Mach number after 6 ms on a \( 64 \times 64 \times 64 \) grid. Observe the increased solution resolution exhibited by the fourth-order CENO as compared to a first-order scheme, when solving the shock-box problem on the same \( 64 \times 64 \times 64 \) grid. This type of increased resolution applied to more complex three-dimensional flow fields can provide additional insight into the complex fluid dynamic processes that may be taking place, and this is one of the advantages that motivate the use of the high-order CENO scheme. Furthermore, in Figure 4.12 we show the cells (in red) in which the smoothness indicator has flagged as unfit for high-order reconstruction and in which a limited piecewise linear reconstruction has been used. This serves to demonstrate that the smoothness indicator is able to flag the appropriate regions in a three-dimensional unsteady application to the compressible form of the Euler equations, it also demonstrates that the smoothness indicator is directionally consistent. That is to say that the smoothness indicator flagging observed on the three faces shown in Figure 4.12 is applied symmetrically in each direction. This is indeed
an important step in the verification and testing of the smoothness indicator employed by the proposed CENO scheme in three dimensions.

4.5.3 Periodic Density Wave

In this final test problem for the Euler equations, we seek to investigate the computational efficiency of the fourth-order CENO scheme by comparing the scheme to a second-order limited
Figure 4.11: Mach number variation for the shock-box problem: $t = 6\, ms$, $64^3$ grid.

Figure 4.12: Comparison of density variation and corresponding smoothness indicator flagging for the shock-box problem: $t = 6\, ms$, $64^3$ grid.
piecewise linear, Godunov-type scheme, as described in Chapter 2. Here we employ Roe’s Riemann solver, along with the Venkatakrishnan slope limiter. The initial conditions for this test case are as follows:

\[
\begin{align*}
\rho &= 1.0 + 0.5 \cos(\pi x) \sin(5\pi x) \text{ kg/m}^3, \\
 u &= 100 \text{ m/s}, \\
 v &= w = 0 \text{ m/s}, \\
 p &= 101.325 \text{ kPa}.
\end{align*}
\] (4.13)

Here the density has been initialized using a smooth, modulated-amplitude, one-dimensional function. The density wave is then propagated in the \(x\)-direction using periodic boundary conditions for one-full period, such that the time-marching is halted when the wave returns to its original starting place; that is, after 20 ms (at 100 m/s) on a \(2 \times 2 \times 2\) m domain.

Figure 4.13 compares the initial condition (which is also the exact solution) to the numerical solution of the propagating density wave problem obtained using the fourth-order CENO scheme after one-full period. It can be observed that there is very little difference in the two density profiles. The primary considerations in our comparison are the magnitude of the error-norms, the time required to propagate the density wave one-full period, and the required memory storage. Table 4.5.3 presents the results of the accuracy assessment, along with the time and memory storage required to obtain the solutions making use of a single-processor. Notice that the fourth-order CENO scheme, on a \(200 \times 4 \times 4\) grid, obtains comparable error levels (that is, the same order of magnitude) as the second-order scheme on a \(4000 \times 4 \times 4\) grid. As previously discussed, there is certainly additional cost required on a per cell basis with the fourth-order CENO scheme, however, as demonstrated in this comparison, this is far-outweighed by the significant reduction in the total number of cells required. Observe, in Table 4.5.3, that the total time required to reach an equivalent magnitude of error is 49 hours 1 minute for the second-order scheme and only 4 hours 34 minutes for the fourth-order CENO scheme. This demonstrates a reduction in time by a factor of 10.7 for this particular test case.

When considering the available computer resources and hardware, one must also consider the required memory storage for the given scheme. There are two points to consider here. The first is that, in three dimensions, the fourth-order CENO scheme requires more memory than the second-order scheme (326 vs. 233 MB). Although the CENO scheme requires significantly fewer cells, the storage of the geometric coefficients and the pseudo-inverse matrix for each cell accounts for the increased memory. As compared to the two-dimensional CENO scheme, the storage of the pseudo-inverse matrix accounts for an increase in memory by a factor of approximately 11; the size of the pseudo-inverse matrix is increased from \(9 \times 24 = 216\) in two
Chapter 4. Verification and Numerical Results

(a) Initial Conditions and Exact Solution

Figure 4.13: Contour plots of the periodic density wave.

dimensions, to $19 \times 124 = 2,356$ in three dimensions (see Section 3.5.3).

The second point to consider, which concerns both time and memory, is that this particular test case represents a one-dimensional problem solved using the three-dimensional implementations of the two schemes. As such, the fourth-order CENO scheme is computing several derivatives in the $y$ and $z$ directions which do not serve to increase the accuracy of the polynomial reconstruction. If we project the numbers in Table 4.5.3 to a three-dimensional problem, not only should we expect the difference in time be significantly greater, but the memory requirements of the fourth-order CENO scheme are projected to be significantly less than the second-order scheme. With modern super-computers, based on distributed-memory parallel clusters, the fourth-order CENO scheme remains well within reach; as demonstrated by the solution of the $128^3 = 2,097,152$ grid resolutions demonstrated in Section 4.3. On the other hand, for the second-order scheme, the number of cells (for example, $4000^3 = 64 \times 10^9$) and corresponding memory requirements can quickly escalate and become out of reach for even the largest and most powerful of super-computers. As more grid topologies and boundary conditions are developed and implemented within the CENO computational framework, similar and additional studies may be carried out in order to further demonstrate the aforementioned advantages of using the high-order CENO scheme.
Table 4.5: Table of results comparing the limited second-order scheme and the fourth-order CENO scheme.

<table>
<thead>
<tr>
<th># Cells</th>
<th>$O(\Delta^2)$</th>
<th>$O(\Delta^4)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>3,200</td>
<td>L1: 2.76E-02</td>
<td>L1: 1.91E-04</td>
</tr>
<tr>
<td>(200x4x4)</td>
<td>L2: 3.35E-02</td>
<td>L2: 2.13E-04</td>
</tr>
<tr>
<td></td>
<td>LMax: 7.38E-02</td>
<td>LMax: 3.34E-04</td>
</tr>
<tr>
<td></td>
<td>Time: 00:07:12</td>
<td>Time: 04:34:28</td>
</tr>
<tr>
<td></td>
<td>Mem: 28</td>
<td>Mem: 326</td>
</tr>
<tr>
<td>6,400</td>
<td>L1: 9.68E-03</td>
<td>L1: 1.19E-05</td>
</tr>
<tr>
<td>(400x4x4)</td>
<td>L2: 1.20E-02</td>
<td>L2: 1.33E-05</td>
</tr>
<tr>
<td></td>
<td>LMax: 2.99E-02</td>
<td>LMax: 2.10E-05</td>
</tr>
<tr>
<td></td>
<td>Time: 00:14:06</td>
<td>Time: 20:26:32</td>
</tr>
<tr>
<td></td>
<td>Mem: 38</td>
<td>Mem: 628</td>
</tr>
<tr>
<td>64,000</td>
<td>L1: 1.14E-04</td>
<td>L1: -</td>
</tr>
<tr>
<td>(4000x4x4)</td>
<td>L2: 2.26E-04</td>
<td>L2: -</td>
</tr>
<tr>
<td></td>
<td>LMax: 1.33E-03</td>
<td>LMax: -</td>
</tr>
<tr>
<td></td>
<td>Time: 49:01:12</td>
<td>Time: -</td>
</tr>
<tr>
<td></td>
<td>Mem: 233</td>
<td>Mem: -</td>
</tr>
</tbody>
</table>

Time format - hh:mm:ss Memory - MegaBytes (MB)

4.6 Application to the Navier-Stokes Equations

In this work we have also considered the implementation and treatment of an appropriate high-order spatial discretization for the elliptic operators, that is, for the viscous fluxes that are present in the Navier-Stokes equations (see Section 3.7). As such, in this final results section, we carry out a partial verification of the the high-order CENO scheme in the application to the compressible form of the Navier-Stokes equations in three dimensions. The application to be considered is the Large Eddy Simulation (LES) of turbulent non-reacting flow, and more specifically, the prediction of the decay of homogeneous isotropic turbulence. To keep this presentation tractable, we begin with a brief introduction to turbulence and LES, before presenting the results of the specific problem of interest.
4.6.1 A Brief Introduction to Turbulence Modelling

Turbulence involves a wide range of time and length scales, varying from the large energy-containing eddies all the way down to the smallest eddies at the Kolmogorov scales. The so-called energy cascade describes the manner in which kinetic energy enters the turbulence at the largest scales of motion, and is then transferred largely by “inviscid processes to smaller and smaller scales until at the smallest scales, the energy is dissipated by viscous processes [56].” Furthermore, in every turbulent flow at sufficiently high Reynolds number, the small-scale turbulent motions are statistically isotropic (Kolmogorov’s hypothesis of local isotropy). It has also been postulated that the smaller scale turbulent motions have a universal form that is uniquely determined by the rate of dissipation, $\varepsilon$, and the kinematic viscosity of the fluid, $\nu$ (in the so-called dissipation range). Only motions in the dissipation range experience significant viscous effects, and are responsible for essentially all of the dissipation [56]. The structure of turbulent eddies are depicted in Figure 4.14 for an isotropic homogeneous flow field [56].

In the attempt to model the complex phenomenon of turbulence, including the aforementioned energy cascade, three main approaches to turbulence modelling have been established: Direct Numerical Simulation (DNS), Reynold’s Averaged Navier-Stokes (RANS), and LES. In LES, the unsteady large scale motions are resolved, while the small scales are modelled [56].
approach offers a compromise between the DNS approach, in which all of the turbulent scales are resolved (requiring a grid spacing commensurate with the Kolmogorov length scale), and the RANS approach, in which none of the turbulent fluctuations are resolved, and instead, the time-averaged RANS approach attempts to model the influence of all turbulent fluctuations on the mean flow.

An important component of the LES approach is the spatial filtering operation used to separate the small scales from the large scales. For the results presented herein, we have used an implicit filtering approach applied to uniform Cartesian grids. In this case, the filter width, $\Delta$, is coupled with the mesh spacing, $\Delta$, by the so-called Filter-Grid Ratio ($FGR = \frac{\Delta}{\Delta}$). Following the spatial filtering of the solution, the effects of the unresolved filtered scales must be modelled by a so-called Subfilter-Scale (SFS) model. For the results presented herein, the standard Smagorinsky model is used to approximate the SFS stresses. Considering the scope of this work, the above descriptions are all that we need to continue our demonstration in a meaningful manner. For more detailed and complete discussions of the exciting and complex subject of turbulence and turbulence modelling, the reader is refereed to several textbooks on the subject [57, 58].

4.6.2 Application to Isotropic Turbulence Decay

For the purpose of verifying the high-order CENO scheme, we will make use of the fact that no LES simulation is valid if it cannot reproduce the correct behaviour of isotropic turbulence decay [56]. “Experiments and DNS results agree that the rate at which isotropic turbulence decays is proportional to $t^{-1.25}$. The convergence of the decay rate from the LES calculations to this value indicates the proper behaviour” [56]. Hence, the objective here is to ensure that such behaviour is recovered when making use of the high-order CENO scheme. For this demonstration we make use of an established and previously implemented test problem considered in previous studies by Deconinck [56]. As an additional basis for comparison, we also include the results obtained using the second-order Limited Piecewise Linear reconstruction scheme which uses a compact diamond path viscous flux evaluation (labeled herein as the “second-order LPL” scheme).

The total turbulent kinetic energy of the flow field is the metric by which we may measure the isotropic turbulence decay rate. The total turbulent kinetic energy, $K$, represents the kinetic energy in the flow, per unit mass, associated with the turbulent velocity fluctuations, and is defined as

$$K = \frac{3}{2} u_{\text{rms}}^2,$$  \hspace{1cm} (4.14)

where $u_{\text{rms}}$ is the root-mean-square (RMS) fluctuating velocity field in three dimensions. Note that $K$ consists of the sum of the resolved turbulent kinetic energy and the modelled SFS-kinetic
energy [56].

A homogeneous isotropic turbulence field is obtained by initializing a $2\pi \times 2\pi \times 2\pi$ m solution domain with Rogallo’s random procedure [59], using the model spectrum proposed by Hawroth-Poisnot [60], the details of both are omitted for brevity. Table 4.6 presents the turbulence-related parameters, with brief descriptions, as used for this problem. It is noted that since the proposed turbulence decay problem employs periodic boundary conditions and does not require any curved boundaries, it is particularly well suited to our verification purposes.

In this work, we simulate the turbulent flow field for a duration of 100 ms. The following three grid resolutions will be considered by making use of different orders of the CENO scheme, along with the second-order LPL scheme:

- $32 \times 32 \times 32$ cells, FGR = 1.
- $64 \times 64 \times 64$ cells, FGR = 2.
- $128 \times 128 \times 128$ cells, FGR = 4.

In the above cases, as we refine the mesh (by halving the uniform mesh spacing in each case), the FGR is intentionally doubled in order to maintain a constant filter width for all of the simulations. In doing so, we are attempting to resolve/model the same scales of the turbulent motion in each LES simulation, in turn allowing us to make comparisons of the isotropic turbulence decay rates.

<table>
<thead>
<tr>
<th>parameter</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specific total turbulent kinetic energy</td>
<td>$K$</td>
</tr>
<tr>
<td>Root-mean-square of velocity fluctuations</td>
<td>$u_{rms}$</td>
</tr>
<tr>
<td>Characteristic length scale of largest eddies</td>
<td>$l_0$</td>
</tr>
<tr>
<td>Wave number of largest eddies</td>
<td>$\kappa_0$</td>
</tr>
<tr>
<td>Integral length scale</td>
<td>$l_{11}$</td>
</tr>
<tr>
<td>Dissipation rate</td>
<td>$\varepsilon$</td>
</tr>
<tr>
<td>Kolmogorov length scale</td>
<td>$\eta$</td>
</tr>
<tr>
<td>Kolmogorov wave number</td>
<td>$\kappa_\eta$</td>
</tr>
</tbody>
</table>

Table 4.6: Specifics of the model spectrum.

Figures 4.16, 4.17, and 4.18 show the comparison of the turbulent kinetic energy versus time, on a log-log scale, for each grid and for the different schemes under consideration. Both the
Chapter 4. Verification and Numerical Results

Figure 4.15: Initial turbulent velocity field. Contour values show the velocity components normal to the faces, arrows show the velocity vectors on the faces.

target slope of -1.25 and the originally implemented second-order LPL scheme are included in the figures.

In each case, we can observe that initially, the total turbulent kinetic energy remains relatively constant (decreasing only slightly), which corresponds to a period in which energy is being transferred from the larger resolved eddies to the smaller eddies represented by the SFS model. The total kinetic energy thus migrates from larger scales to smaller scales, decreasing the resolved turbulent kinetic energy and increasing the SFS-kinetic energy [56]. After about 10 ms, the SFS-model begins dissipating the turbulent kinetic energy, and both components of the turbulent kinetic energy start decreasing. Furthermore, we can acknowledge that the correct asymptotic decay rate is recovered, in each case, when using the high-order CENO schemes. This result can be interpreted as a verification that the gradient calculations of the solution at the Gauss quadrature points and the ensuing high-order viscous flux evaluations (described in Section 3.7.2) are not polluting the LES prediction of homogeneous isotropic turbulence decay, that is, as we compare the CENO schemes to both the theoretical decay rate and the second-order LPL scheme. In fact, it can be seen in the coarser grids (where greater numerical dissipation error may be observed) that the third and fourth-order CENO schemes are in fact less dissipative than the second-order schemes. This is to be expected since a higher-order
Figure 4.16: Comparison of turbulent kinetic energy spectra of isotropic turbulence decay at 100 ms for different orders of spatial reconstruction on a $32\times32\times32$ grid.

Figure 4.17: Comparison of turbulent kinetic energy spectra of isotropic turbulence decay at 100 ms for different orders of spatial reconstruction on a $64\times64\times64$ grid.
Figure 4.18: Comparison of turbulent kinetic energy spectra of isotropic turbulence decay at 100 ms for different orders of spatial reconstruction on a $128 \times 128 \times 128$ grid.

Figure 4.19: Comparison of turbulent kinetic energy spectra of isotropic turbulence decay at 100 ms for second-order LPL scheme and fourth-order CENO scheme on different grid resolutions.
reconstruction scheme is typically expected to be less dissipative than a correspondingly lower-order scheme.

Finally, Figure 4.19 compares the three grids using a fourth-order CENO reconstruction and the second-order LPL scheme. Here we can observe that as we refine the mesh, the transitions to the correct asymptotic decay rate occur later in time, which is in agreement with the findings of Deconinck and Groth [56], and can be attributed to a reduction in the numerical dissipation as we refine the grid. Note that in the 128³ grid, the two schemes show nearly identical results, indicating convergence toward a grid-independent solution. Furthermore, comparing the two schemes on the coarser grids (32³ and 64³), we can observe that the fourth-order CENO scheme converges toward a grid-independent solution more rapidly than the second-order LPL scheme.

These results hold much promise for the future development and application of the high-order CENO scheme in combination with LES. In the future, as curved three-dimensional boundaries, additional grid topologies, and boundary conditions are developed and implemented, additional turbulent flow problems will duly serve to further demonstrate and verify the applicability and advantages of the high-order CENO scheme. The long-term objectives (outside the scope of this research) will be to combine the high-order CENO scheme with AMR for the LES of turbulent reacting flows involving disparate spatial and temporal scales.
Chapter 5

Conclusions and Future Research

5.1 Conclusions

The use of high-order methods in CFD has yet to see widespread use by the engineers involved in the design of aerospace technology. The reasons for this may be attributed to the underlying issues that appear to be common to all high-order CFD methods. These include, but are not limited to, the issues regarding solution monotonicity, computational cost and memory usage, as well as the robustness, applicability, and simplicity of the available high-order methods (many of which are still in development). These issues are particularly pertinent in the application of high-order methods to complex three-dimensional flow fields. This research has attempted to address the ultimate challenge of achieving high-order accuracy without compromising the solution monotonicity, and has attempted to do so in a robust and efficient manner in three dimensions. The result is a scheme that presents a practical compromise between accuracy, computational efficiency, and robustness that is well suited to the exciting applications mentioned in Chapter 1.

In summary, the various components of the CENO scheme, including the $k$-exact reconstruction procedure and its least-squares solution, the high-order flux evaluations, and the smoothness indicator, have each been successfully extended, further developed, and implemented in three dimensions based on their two-dimensional counter-parts. The combination of these components has resulted in a new high-order CENO FV scheme for the compressible forms of the Euler and Navier-Stokes equations in three dimensions.

A verification of the CENO scheme has determined that its implementation in three dimensions accurately represents the author’s conceptual and theoretical description of the scheme. The
verification process has made use of several representative test cases. It has been shown that the $k$-exact reconstruction procedure successfully achieves the corresponding high-orders of spatial accuracy for smooth solutions. Furthermore, the robust, reliable, and versatile manner with which the smoothness indicator operates has been exemplified by its application to both discontinuous and under-resolved solution content. The high-order CENO scheme has been successfully applied to both the Euler and Navier-Stokes equations in three dimensions, in turn verifying the high-order implementations of the inviscid and viscous flux evaluations. The application to the Navier-Stokes equations has included the combination of the high-order CENO scheme with the LES of turbulent compressible non-reacting flows.

This work has successfully achieved its primary objectives as outlined in Section 1.2. The results we have presented are very promising, but it is noted that we have yet to realize the full potential of the proposed CENO scheme, nor have we applied it to the multitude of far-reaching applications described in Section 1.1. That being said, this work provides a firm base from which the CENO scheme may be further developed; some recommendations may be found in the next section. It is the author’s hope that in time, the high-order CENO scheme will serve out its ultimate purpose of providing a means of better understanding complex fluid dynamic phenomena (such as turbulence) and as an engineering design tool used to accelerate and optimize the design of future aerodynamic and other related aerospace technology. There is little doubt that the research community is still in the process of creatively developing high-order CFD methods. We are evidently still waiting for that moment when all uncertainty subsides and a clear consensus on the ultimate high-order methodology prevails; a moment which may exist only in the author’s own naive hopes.

5.2 Future Research

In this final section, we present a list of some of the recommendations for the future research and development of the high-order CENO scheme in three dimensions.

- Ways to reduce the number of required ghost cells should certainly be investigated. It seems viable to reduce the number of ghost cells required at the block boundaries by message passing (and applying boundary conditions) in such a way that we explicitly provide the smoothness indicator with the information required to evaluate the solution smoothness parameter, rather than reconstructing the solution in additional ghost cells. More specifically, we would need to pass the cell-centered reconstructed solution values of ghost cell layers numbered 2 and 3 (as depicted in Figure 3.4). This would altogether
eliminate the need for ghost cell layers 4 and 5, and presents a potentially significant cost savings, particularly for the parallel multi-block implementation in three dimensions.

- Development and implementation of the high-order CENO scheme with AMR in three dimensions should also be considered. This combination has the potential to significantly reduce the computational cost and memory usage of the scheme, as demonstrated by Ivan and Groth, who have successfully combined these techniques in their two-dimensional implementations [2, 6].

- This work has been restricted to Cartesian grids, and as such the development and implementation of the high-order CENO scheme with general hexahedral- and tetrahedral-type cells for structured, unstructured, and hybrid grid topologies should be considered. An additional extension of this work would be the high-order treatment of curved boundaries in three dimensions; a necessary step toward a scheme capable of handling the complex three-dimensional geometries that are often encountered in aerospace applications.

- The high-order treatment of additional types of boundary conditions should be investigated. As mentioned in Section 3.8 an alternate approach to addressing boundary conditions on straight and curved boundaries is the so-called “constrained reconstruction” technique, the extension from its two-dimensional implementation, should also be considered in the future development of the CENO scheme for three-dimensional applications.

- As curved three-dimensional boundaries, additional grid topologies, and boundary conditions are developed and implemented, a more rigorous and detailed verification and validation process may then be carried out by making use of more complex and realistic test cases. This is particularly necessary in order to evaluate and demonstrate the potential savings in computational cost and memory usage of the CENO scheme (as was done for a one-dimensional test case in Section 4.5.3).

- With the promising results presented in Section 4.6 for the combination of the high-order CENO scheme with the LES of turbulent compressible non-reacting flows, future research should continue to investigate the advantages of this combination. The long term objectives of this work should be the LES of turbulent compressible reacting flows, involving disparate spatial and temporal scales, using the high-order CENO scheme with AMR.

The above list is by no means exhaustive. In this section we have merely tried to suggest some of the exciting avenues and applications that will help unlock and demonstrate the full potential of the high-order CENO scheme in three dimensions.
References


