HIGH-ORDER CENO RECONSTRUCTION SCHEME FOR THREE-DIMENSIONAL UNSTRUCTURED MESH

by

Al-Amin Aziz

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 © 2014 by Al-Amin Aziz
Abstract

High-Order CENO Reconstruction Scheme For Three-Dimensional Unstructured Mesh

Al-Amin Aziz
Masters of Applied Science
Graduate Department of Aerospace Engineering
University of Toronto
2014

Research and development of high-order methods in the past decade have focused on obtaining more accurate solutions with the additional advantage of decreasing overall computational cost and memory requirements. The present research focuses on implementation of the CENO reconstruction scheme for three-dimensional unstructured meshes. The CENO reconstruction scheme makes use of a hybrid reconstruction technique based on a fixed single central stencil which is particularly advantageous for unstructured meshes. A smoothness indicator facilitates the switching from a $k$-exact reconstruction in smooth regions of the solution to a limited piecewise linear reconstruction in unresolved or discontinuous regions. This scheme has been implemented to reconstruct both continuous and discontinuous functions on three-dimensional unstructured tetrahedral meshes.
Acknowledgements

This thesis would not have been possible without the support and encouragement from the following people.

First and foremost, I would like to express my gratitude to my supervisor, Professor C. P. T. Groth, for his guidance and expertise during my stay at University of Toronto. It has been an honour to be part of your CFD group.

I would also like to express my gratitude to my parents and grandparents, who were present for my every step through this journey. I am truly indebted and grateful to your love, support, encouragement and motivation.

Finally, I would like to thank my colleagues and the community of The University of Toronto Institute for Aerospace Studies for their invaluable assistance and friendship. It has truly been an honour to work with these great minds and I do hope to stay in touch with everyone as they now have become part of my own family.

Toronto, April 2014

Al-Amin Aziz
Contents

Abstract ii

Acknowledgments iii

Contents iv

List of Tables vii

List of Figures viii

1 Introduction 1

1.1 Background .................................................. 2

1.2 Motivation and Thesis Objectives .......................... 3

1.3 Overview of Thesis .......................................... 4

2 CENO FV Scheme for Conservation Laws in 1D 5

2.1 Euler Equations in One Dimension ........................ 5

2.1.1 Thermodynamic Relations ............................... 6

2.2 Finite-Volume Methods ..................................... 6

2.3 Godunov-Type Finite-Volume Methods ..................... 7

2.4 Second-Order Scheme ...................................... 8

2.4.1 Piecewise Linear Reconstruction ....................... 9
4.2.1 Determination of Reconstruction Stencil .................................. 38
4.2.2 Determination of the Derivatives for the Reconstruction .......... 39
4.2.3 Quadrature Rules ............................................................. 44
4.3 High-Order CENO Reconstruction ......................................... 45
  4.3.1 Hybrid Reconstruction Scheme ....................................... 46
  4.3.2 Smoothness Indicator in Three Dimensions ....................... 46

5 Numerical Results ........................................................................ 48
  5.1 Verification Process for High-Order Reconstruction ............... 49
    5.1.1 Determining the Order of Accuracy .............................. 49
    5.1.2 Accuracy Assessment ................................................... 49
  5.2 Visualization of Reconstructed Solutions .............................. 50
  5.3 Function Reconstruction in Three Dimensions ....................... 51
    5.3.1 Radial Cosine Function .............................................. 52
    5.3.2 Exponential Function ............................................... 55
    5.3.3 Abgrall’s Function .................................................... 56
  5.4 Summary of Results .......................................................... 61

6 Conclusions and Future Research ............................................. 62
  6.1 Conclusions ................................................................. 62
  6.2 Future Research ............................................................. 63

References ................................................................. 68
List of Tables

2.2 Orders of accuracy for the function reconstruction. .......................... 15
2.3 Initial conditions for shock-tube problem. ...................................... 23
5.1 Orders of accuracy for the Spherical Cosine Function. ...................... 53
5.2 Orders of accuracy for the Exponential Function. ............................ 55
5.3 Orders of accuracy for the Abgrall’s Function. ............................... 59
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Graph of $f(\alpha) = \frac{\alpha}{1-\alpha}$</td>
<td>14</td>
</tr>
<tr>
<td>2.2</td>
<td>20 cell function reconstruction</td>
<td>16</td>
</tr>
<tr>
<td>2.3</td>
<td>2nd-order slope limiters</td>
<td>17</td>
</tr>
<tr>
<td>2.4</td>
<td>CENO reconstruction</td>
<td>17</td>
</tr>
<tr>
<td>2.5</td>
<td>Convergence study</td>
<td>18</td>
</tr>
<tr>
<td>2.6</td>
<td>Depiction of the Riemann problem</td>
<td>19</td>
</tr>
<tr>
<td>2.7</td>
<td>Depiction of five possible solutions to the Riemann problem</td>
<td>20</td>
</tr>
<tr>
<td>2.8</td>
<td>Reconstructed solutions to a shock-tube problem</td>
<td>23</td>
</tr>
<tr>
<td>2.9</td>
<td>Reconstructed solutions to a shock-tube problem</td>
<td>24</td>
</tr>
<tr>
<td>2.10</td>
<td>Riemann solvers used for the shock-tube problem</td>
<td>25</td>
</tr>
<tr>
<td>3.1</td>
<td>Example of a two-dimensional geometric profile created using Gmsh.</td>
<td>28</td>
</tr>
<tr>
<td>3.2</td>
<td>Example of a three-dimensional geometric model created using Gmsh.</td>
<td>29</td>
</tr>
<tr>
<td>3.3</td>
<td>Example of a three-dimensional surface mesh created using Gmsh.</td>
<td>30</td>
</tr>
<tr>
<td>3.4</td>
<td>Example of a three-dimensional volume mesh created using Gmsh.</td>
<td>31</td>
</tr>
<tr>
<td>3.5</td>
<td>Three-dimensional cubed geometries and unstructured tetrahedral mesh used in the function reconstruction considered in this thesis.</td>
<td>33</td>
</tr>
<tr>
<td>4.1</td>
<td>Schematic of a tetrahedral cell</td>
<td>36</td>
</tr>
<tr>
<td>4.2</td>
<td>Nearest and next to nearest neighbours</td>
<td>39</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

Computational Fluid Dynamics (CFD) has become a key analysis tool in many fields of science and engineering. Nevertheless, the application of CFD has to date mainly focused on dealing with simple fluid flows. Physically-complex flows have been found to be difficult to solve and challenging to predict using standard numerical methods. These challenging complex flows include but are not limited to:

- multi-phase, turbulent, and combustion flows that have been encountered in propulsion systems,
- compressible flows of conducting fluids and plasma, and
- micro-scale non-equilibrium flows that have been observed in micro-electromechanical systems [1].

The current generation of second-order solution methods have been developed and applied to a wide range of practical flow problems; however, when applied to more challenging and complex problems, such as Direct Numerical Simulation (DNS) [2, 3, 4] and/or the Large Eddy Simulation (LES) [5, 6, 7] of non-reactive and reactive flows, such lower-order methods have been found to be both too dissipative as well as too dispersive. Their reduced accuracy makes them generally too computationally expensive for many practical physically-complex flow applications, particularly unsteady applications requiring high-resolution of turbulence.

Improved computational efficiency can be achieved by turning to high-order spatial and temporal discretization schemes [1]. These methods are developed by increasing the order of accuracy of the discrete representation. They offer the potential to reduce the number of computational cells required to attain a desired level of accuracy. Unfortunately, the computational
costs of high-order methods remains a challenge, particularly for more complex and practical flow geometries. For this reason, the development and implementation of effective and efficient high-order methods remains an active area of research [1].

1.1 Background

Much of the CFD algorithm development in the past decades has focused on obtaining more accurate solutions with the additional advantage of decreasing overall computational cost and memory usage requirements. The development of high-order spatial discretization schemes has been part of this research activity and high-order finite-difference, finite-volume and finite-element methods have all been considered. High-order finite-volume methods examined in previous studies include the essentially non-oscillatory (ENO) scheme by Harten et al. [8], the weighted essentially non-oscillatory (WENO) schemes [9, 10, 11] and the central-essentially non-oscillatory (CENO) schemes recently proposed by Ivan and Groth [1, 12, 13, 14]. The major challenge these schemes face is to achieve high-accuracy while preserving solution monotonicity of the reconstructed solution.

The ENO, WENO and CENO schemes are similar high-order schemes, but rely on different techniques for preserving solution monotonicity. The ENO scheme avoids the issues at discontinuities by performing reconstructions on several stencils and selecting the stencil that is the smoothest [8, 15, 16]. Conversely, WENO scheme uses a solution-dependent weighing of the reconstructions for each stencil [9, 10, 11]. The CENO scheme uses a high-order \( k \)-exact reconstruction scheme based on a single central stencil that locally reduces the order of the reconstruction near discontinuities to that of a second-order linear piecewise reconstruction where necessary [1, 12, 13, 14].

These three high-order finite-volume schemes have all been applied to both structured and unstructured meshes. The advantage of using unstructured over structured topologies is that the generation of the mesh is greatly simplified for complex geometries [17]. In addition, local nodes in the mesh can easily be generated and introduced for adaptive grid development [18]. Therefore, the implementation of these high-order schemes for unstructured meshes is deemed more applicable for practical use.

The application of the ENO scheme to unstructured mesh was first investigated by Abgrall [19], and subsequently investigated by others [15, 20]. WENO schemes have also been developed on unstructured meshes [21], but both ENO and WENO schemes have encountered difficulties for multi-dimensional unstructured meshes due to selection of appropriate stencils [19, 21, 22, 23].
Chapter 1. Introduction

The requirement of multiple stencils can also lead to the poor conditioning of the linear systems that are involved in the solution reconstruction [21, 23]. In addition, the computational cost and complexities involved in the ENO and WENO for unstructured mesh methods have limited the range of applications to finite-volume schemes [12, 24].

The CENO scheme developed by Ivan and Groth [12, 24] attempts to address some of the disadvantages associated with ENO and WENO schemes. The CENO scheme avoids the complexity associated with the usage of multiple stencils by using a fixed central stencil. The central stencil provides the most accurate reconstruction due to cancellation of truncation errors. The CENO scheme uses a hybrid technique where an unlimited, high-order, $k$-exact least-squares reconstruction is first performed [25], and this is automatically reverted to a monotonicity preserving limited piecewise linear least-squares reconstruction in cells near shocks or in discontinuous regions. The switching in the hybrid technique is controlled by a solution smoothness indicator.

Hence, the CENO reconstruction is an effective method for eliminating the appearance of spurious numerical oscillations in under-resolved regions and in regions that contain strong discontinuities and/or shocks. Moreover, because the reconstruction is based on a single central stencil (also the most accurate stencil), the CENO scheme would therefore seem well suited for use with both multi-dimensions and unstructured meshes. The latter is deemed, as previously mentioned, to be very advantageous for the solution of flows through more complex and practical three-dimensional geometries.

1.2 Motivation and Thesis Objectives

The CENO scheme has proven to be an effective and efficient method in obtaining high-order accuracy on multi-dimensional structured meshes in both two and three spatial dimensions [26, 27]. The high-order finite-volume method has also been extended for applications to two-dimensional unstructured meshes [27, 28]. Thus, this thesis aims to build on the previous work of Rashad [26] and McDonald [28] and further extend the method for application to three-dimensional unstructured meshes.

In particular, the objective of this research is to develop and verify the higher-order CENO reconstruction procedure for general three-dimensional unstructured meshes. The development of the CENO scheme has been implemented within the in-house code of the University of Toronto Institute for Aerospace Studies (UTIAS) CFD and propulsion group. Following implementation, the CENO scheme was verified and assessed by performing solution reconstruction for both continuous and discontinuous functions in three-dimensions.
The extension of the CENO scheme to three-dimensional unstructured meshes is an important step in the advancement of the method. High-order methods have been shown to significantly improve the quality of numerical solutions [25] and the CENO scheme developed [1, 12, 13, 14] also supports this improvement. In the longer term, the future application of this research could eventually lead to the construction of high-order techniques for the Euler and Navier-Stoke equations on three-dimensional unstructured meshes.

### 1.3 Overview of Thesis

Following this chapter, which provides an outline for the proposed research related to the development of the high-order CENO reconstruction method for general three-dimensional unstructured mesh, Chapter 2 first provides a review of the high-order CENO scheme and its implementation for the solution of the one-dimensional form of euler equations governing compressible gas dynamics. The latter is a very commonly used representative system for hyperbolic systems of conservation laws, for which the CENO scheme has been developed herein. Chapter 3 then provides an overview of meshing techniques used in the development of general three-dimensional unstructured meshes. Chapter 4 provides an overview of the extension of the CENO reconstruction scheme procedure to three-dimensional unstructured meshes. Chapter 5 focuses mainly on a discussion of numerical results achieved for the reconstruction of both continuous and discontinuous functions. The final chapter, Chapter 6, summarizes the overall conclusions of this thesis and the future research related to further the development of the CENO scheme.
Chapter 2

High-Order CENO Finite-Volume Scheme for Conservation Laws in One Dimension

This chapter will review the development and implementation of the high-order CENO finite-volume method for hyperbolic conservation laws in one dimension. In particular, the CENO finite-volume scheme applied to the solution of the one-dimensional form of the Euler equations governing inviscid compressible gas dynamics is reviewed. Details of the finite-volume scheme and high-order evaluation of the numerical fluxes and residual integrals are described and discussed.

2.1 Euler Equations in One Dimension

To understand Godunov-type finite-volume methods [29, 30], numerical solutions of the Euler equations will first be examined for the case of a single spatial dimension. The differential form of Euler equations in one dimension is given by

\[ \frac{\partial U}{\partial t} + \frac{\partial F}{\partial x} = 0, \quad (2.1) \]

where the vector \( U \) consists of the conserved variables, and \( F \) is the inviscid flux vector. The components of \( U \) and \( F \) are
\[ U = \begin{bmatrix} \rho \\ \rho u \\ e \end{bmatrix}, \quad F = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ u(E + p) \end{bmatrix}. \] (2.2)

In the above set of equations, \( u \) is the velocity component in the \( x \)-coordinate direction, \( p \) is the pressure term, \( \rho \) is the mass density, \( E \) is the total specific energy represented by the relation \( E = e + \frac{1}{2}u^2 \) and \( e \) is the specific internal energy.

### 2.1.1 Thermodynamic Relations

To obtain a solution to the Euler equations, the system of equations must be closed. This can be achieved by developing an equation of state that relates the pressure, density, and temperature terms. In the present study, the gas is assumed to behave as a thermally and calorically perfect gas. As such, it obeys the ideal gas law\[ p = \rho RT, \] (2.3)

where \( R \) is the gas constant, \( p \) is the pressure, \( \rho \) is the density, and \( T \) is the temperature. In addition, terms such as the specific energy, \( e \), the enthalpy, \( h \), and the speed of sound, \( a \), of a calorically perfect gas can be specified using

\[ e = \frac{p}{\rho(\gamma - 1)}, \quad h = \frac{\gamma}{\gamma - 1} \frac{p}{\rho}, \quad a^2 = \frac{\gamma p}{\rho}, \] (2.4)

where \( \gamma \) is the specific heat ratio for the gas. For air, which is of interest here, the specific heat ratio is taken to be \( \gamma = 1.4 \).

Equations (2.3) and (2.4) provide the necessary closure for the Euler equations. Note that with these thermodynamic relations, the Euler equations can be described in terms of three dependent variables (\( \rho, u, \) and \( p \) or \( T \)) and two independent variables (\( x \) and \( t \)).

### 2.2 Finite-Volume Methods

Finite-volume methods discretize the computational domain into many control volumes, enabling the integral form of the conservation laws to be satisfied for each control volume. The two main advantages of using finite volume methods are:

1. finite-volume methods ensure that mass, momentum and energy is conserved discretely in each computational cell; and
2. finite-volume methods do not require a coordinate transformation as required in finite difference methods for the treatment of non-cartesian mesh [32].

As discussed in the introduction of this thesis, high-order schemes have shown the potential to improve solution accuracy for finite-volume methods. Therefore, gaining a thorough understanding of finite-volume methods is very important to the development of high-order schemes. In addition, the one-dimensional CENO scheme developed in this chapter will provide a framework for the development and implementation of this scheme and, in particular, the high-order reconstruction procedure for three-dimensional unstructured meshes.

The finite-volume methods of interest will be the Godunov-type methods [29, 30]. These methods rely on the solution to the non-linear Riemann problem at the interfaces of the computational cell in the evaluation of the numerical fluxes. These methods have successfully been applied to invicid flows described by the Euler equations in previous studies [33, 34, 35].

2.3 Godunov-Type Finite-Volume Methods

Godunov-type finite-volume methods discretize the solution domain into control volumes (or cells) and apply an integral form of the conservation equations to each of the discrete control volumes [29]. This finite-volume method consists of three key stages. The first stage involves the reconstruction of the solution in each cell. From the initial solution at a given time level, the cell-average values for the conserved variables are known. Therefore, a piecewise polynomial can be constructed, and allows for the value to be determined for each conserved solution variable at any point within the cell.

The second stage consists of evaluating the fluxes obtained at each cell. The main problem that arises with evaluating fluxes is to deal with discontinuities that occur across the neighbouring cells. This can be resolved by obtaining the solution to the Riemann problem.

The final stage is to evolve the solution through time. This is achieved by using a time-marching scheme. There are various time-marching schemes, but for this thesis the explicit Runge-Kutta time-marching methods are used.

For the one-dimensional conservation laws of the Euler equations given in Equation (2.1), application of the original first-order Godunov method [29] yields the following expression for the update of the cell-averaged solution for cell $i$: 
\[
U_i^{n+1} = U_i^n - \frac{\Delta t}{\Delta x} \left[ F_{i+\frac{1}{2}}^n - F_{i-\frac{1}{2}}^n \right],
\]  

where the cell-averaged conserved solution vector is defined by \( \bar{U}_i \)

\[
\bar{U}_i = \frac{1}{\Delta x} \int_{x_{i-\Delta x/2}}^{x_{i+\Delta x/2}} U_i(x) \, dx,
\]

the time level is identified by the superscript \( n \), and the inviscid solution flux is identified by the term \( F \). The solution flux \( F \) at the cell interfaces is obtained from the solution to the Riemann problem. For a piecewise constant reconstruction which results in a first-order scheme, the solution flux, \( F_{i\pm\frac{1}{2}}^n \), at the cell interfaces are

\[
F_{i+\frac{1}{2}}^n = F(\ U_{\text{left}} = U^n_{i}, \ U_{\text{right}} = \bar{U}^n_{i+1}, \ x^L = 0 ) \quad (2.7)
\]

\[
F_{i-\frac{1}{2}}^n = F(\ U_{\text{left}} = \bar{U}^n_{i-1}, \ U_{\text{right}} = U^n_{i}, \ x^R = 0 ) \quad (2.8)
\]

The expression of Equation (2.5) can be used to update the solution in each cell throughout the computational domain and in this way the solution can be advanced to the next time step.

### 2.4 Second-Order Scheme

Second-order variants of the Godunov method require the reconstruction of the conserved solution vector, \( U \), based on the solution information from neighbouring cells. For one dimension as well as multi-dimensions, cell reconstruction can be achieved by identifying a stencil of cells and using the cell-averaged values of the conserved solution vector in each of these neighbouring cells. The solution reconstruction is then obtained by using either a least-squares approach or Green-Gauss technique based on solution information from the cell-centered stencil. Further discussion of these techniques can be found in the paper by Barth [25].

It is noted that for second-order or higher-order reconstruction, solution monotonicity is a concern. Schemes that preserve solution monotonicity assure that no new extrema are formed when the reconstruction of the stencil is carried out. To achieve a monotone solution, the stencil undergoing reconstruction is modified by a slope limiter. Further discussion of slope limiters is given in Section 2.4.2 to follow.
2.4.1 Piecewise Linear Reconstruction

For one-dimensional problems, a second-order scheme can be achieved by performing a piecewise linear reconstruction of the solution within each cell having the form

$$U_i(x) = U_i + \frac{\partial U}{\partial x} \bigg|_{i} (x - x_i) , \quad (2.9)$$

where $\frac{\partial U}{\partial x} \bigg|_{i}$ is the unknown slope. The slope can be determined by performing a least-squares reconstruction using information from the neighbouring cells. For the one-dimensional case and using just nearest neighbours, the least square solution yields

$$\frac{\partial U}{\partial x} \bigg|_{i} = \frac{U_{i+1} + U_{i-1}}{2\Delta x} , \quad (2.10)$$

which is the usual centred finite-difference approximation for the first derivative of $U$. By making use of Equation (2.9), $U$ can then be computed within any cell $i$ in terms of the average value, slope, and distance from the neighbouring cells. Once these values are found for every cell in the domain, the piecewise linear reconstruction of the entire computational domain is computed.

Using the reconstruction of Equation (2.9), the spatial order of accuracy of this scheme will be that of second-order. For a continuous function, the solution obtained will be monotone. In the case of a discontinuous function and/or under-resolved content, solution monotonicity will not be maintained and will therefore require the use of slope limiters. The latter are discussed in the following section.

2.4.2 Slope Limiters

For a one-dimensional second-order linear reconstruction, the solution states must be found at each of the cell interfaces. In order to accomplish this while maintaining solution monotonicity, the concept of slope limiter is introduced. A slope limiter term, $\phi$, can be introduced in the previously discussed piecewise linear reconstruction of cell $i$ that was shown in Equation (2.9). The modified form would be

$$U_i(x) = \bar{U}_i + \phi_i \frac{\partial U}{\partial x} \bigg|_{i} (x - x_i) , \quad (2.11)$$

In Equation (2.11), the limiter, $\phi$, behaves like a switch. In areas of a discontinuity, $\phi$ approaches
zero. This produces a first-order reconstruction. Whereas, when $\phi$ corresponds to unity, the solution is considered to be continuous. The latter leads to a second-order piecewise linear reconstruction. Therefore, this switching mechanism assures that monotonicity of the solution is maintained.

The following two slope limiters are considered for evaluating the value of $\phi_i$:

1. the slope limiter of Barth-Jesperson [25]; and
2. the slope limiter of Venkatakrishnan [36].

These limiters behave in a way such that the value of the solution at the cell faces is within the maximum and minimum average solution values of the reconstruction stencil.

**Barth-Jesperson Limiter**

The Barth-Jesperson Limiter calculates the slope limiter, $\phi$, using the relation

$$
\phi_{ik} = \begin{cases} 
\min(1, \frac{u_{\text{max}} - u_i}{u_{ik} - u_i}) & u_{ik} > u_i \\
\min(1, \frac{u_{\text{min}} - u_i}{u_{ik} - u_i}) & u_{ik} < u_i \\
1 & u_{ik} = u_i 
\end{cases},
$$

where the value for $\phi_i$ is determined from

$$
\phi_i = \min(\phi_{ik}).
$$

In the above Equation (2.12), the $u$ terms represent the primitive solution variable of interest. These terms are further defined as

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

The Barth-Jesperson slope limiter only limits the solution data when required. Therefore, it is classified as a low-dissipation limiter and has shown to be very effective in removing spurious oscillations from reconstructed solutions [37].
Venkatakrishnan Limiter

The other slope limiter of interest is the Venkatakrishnan limiter. This slope limiter is a modified version of the well known Van-Albada limiter [38]. The Venkatakrishnan limiter calculates the slope limiter, $\phi$, using the relations

$$
\phi_{ik} = \begin{cases} 
\frac{y^2+2y}{y^2+y+2} & u_k > u_i \quad y = \frac{u_{\text{max}}-u_i}{u_k-u_i} \\
\frac{y^2+2y}{y^2+y+2} & u_k < u_i \quad y = \frac{u_{\text{min}}-u_i}{u_k-u_i} \\
1 & u_{ik} = u_i
\end{cases},
$$

(2.14)

where the value for $\phi_i$ is determined from

$$
\phi_i = \min(\phi_{ik}).
$$

(2.15)

The various terms appearing in the Equation (2.14) are similar to the terms used in the Barth-Jesperson limiter. It is important to note that because of the smooth nature of Venkatakrishnan slope limiter, the solution obtained would be slightly more dissipative than the Barth-Jesperson limiter.

2.5 High-Order CENO Scheme

Various high-order finite-volume schemes have previously been developed, such as the ENO scheme by Harten [39] and the WENO scheme [9, 10, 11]. However, these schemes require either selecting or weighing the reconstructions. The CENO scheme developed by Ivan and Groth [1, 12, 13, 14] eliminates these problems of selecting and weighing by introducing a hybrid technique. The technique makes use of a fixed central stencil, and performs reconstruction within the center cell. If the reconstructed solution is determined to be discontinuous or unresolved, the technique switches to a piecewise limited linear reconstruction. The CENO scheme can be summarized in three steps:

1. first, an unlimited $k$-exact reconstruction is performed on all the cells;

2. second, the smoothness indicator is evaluated for each of the cells to determine if the solution is either smooth or discontinuous; and

3. finally, a piecewise limited linear reconstruction is applied to the cells that are flagged as discontinuous.

Note that the CENO reconstruction performs the $k$-exact reconstruction depending on the order of solution accuracy required.
2.5.1 Fourth-Order Reconstruction

For high-order reconstruction, a polynomial approximation is considered to be of an order or degree greater than one. This is referred to as the $k$th-order polynomial reconstruction. The $k$-exact reconstruction is given by the Taylor Series expansion of the conserved solution vector about the cell centers. For one-dimensional problems, the reconstruction is given by

$$\mathbf{U}_i^k(x) = \sum_{p_1=0}^{k} (x - x_i)^{p_1} D_{p_1},$$  \hspace{1cm} (2.16)

where $D_{p_1}$ is the unknown derivative and at $p_1 = 0$ the unknown will be $D_{p_1} = D_0$.

For a fourth-order accurate scheme ($k = 3$), which will be the focus of this initial one-dimensional study, the reconstruction can be obtained by including the first four terms of the Taylor series which are

$$\mathbf{U}_i(x) = D_0 + (x - x_i)^1 D_1 + (x - x_i)^2 D_2 + (x - x_i)^3 D_3.$$  \hspace{1cm} (2.17)

In Equation (2.17) above, the four unknowns are the three derivatives $D_1, D_2, D_3$ and the constant value $D_0$. These unknowns are found by extending the representation of the solution in cell $i$ to its neighboring cells. In this case, at least three neighboring cells would be required to facilitate the reconstruction of cell $i$. However, to preserve reconstruction symmetry, four neighboring cells will be used. This leads to an overdetermined system of equations. In addition, to preserve the average value for cell $i$, the following centered cell constraint will be applied

$$D_0 = \bar{U}_i - \frac{1}{\Delta x} \sum_{p_1=1}^{k} D_{p_1} \int_{x_i - \frac{\Delta x}{2}}^{x_i + \frac{\Delta x}{2}} (x - x_i)^{p_1} \, dx.$$  \hspace{1cm} (2.18)

Using the above constraint, an overdetermined system of four equations is developed and has the form:

$$\begin{align*}
\bar{U}_{i-2} - \bar{U}_i &= \frac{1}{\Delta x} \sum_{p_1=1}^{k} D_{p_1} \int_{x_i - \frac{\Delta x}{2}}^{x_i + \frac{\Delta x}{2}} (x - x_i)^{p_1} \, dx - \frac{1}{\Delta x} \sum_{p_1=1}^{k} D_{p_1} \int_{x_i - \frac{\Delta x}{2}}^{x_i + \frac{\Delta x}{2}} (x - x_i)^{p_1} \, dx \\
\bar{U}_{i-1} - \bar{U}_i &= \frac{1}{\Delta x} \sum_{p_1=1}^{k} D_{p_1} \int_{x_i - \frac{\Delta x}{2}}^{x_i + \frac{\Delta x}{2}} (x - x_i)^{p_1} \, dx - \frac{1}{\Delta x} \sum_{p_1=1}^{k} D_{p_1} \int_{x_i - \frac{\Delta x}{2}}^{x_i + \frac{\Delta x}{2}} (x - x_i)^{p_1} \, dx \\
\bar{U}_{i+1} - \bar{U}_i &= \frac{1}{\Delta x} \sum_{p_1=1}^{k} D_{p_1} \int_{x_i - \frac{\Delta x}{2}}^{x_i + \frac{\Delta x}{2}} (x - x_i)^{p_1} \, dx - \frac{1}{\Delta x} \sum_{p_1=1}^{k} D_{p_1} \int_{x_i - \frac{\Delta x}{2}}^{x_i + \frac{\Delta x}{2}} (x - x_i)^{p_1} \, dx \\
\bar{U}_{i+2} - \bar{U}_i &= \frac{1}{\Delta x} \sum_{p_1=1}^{k} D_{p_1} \int_{x_i - \frac{\Delta x}{2}}^{x_i + \frac{\Delta x}{2}} (x - x_i)^{p_1} \, dx - \frac{1}{\Delta x} \sum_{p_1=1}^{k} D_{p_1} \int_{x_i - \frac{\Delta x}{2}}^{x_i + \frac{\Delta x}{2}} (x - x_i)^{p_1} \, dx 
\end{align*}$$  \hspace{1cm} (2.19)
By solving this overdetermined system of equations, the unknowns $D_0, D_1, D_2$ and $D_3$ are obtained and the reconstruction of the solution within cell $i$ can be performed.

### 2.5.2 Smoothness Indicator

The smoothness indicator, $S$, is part of the CENO scheme developed by Ivan and Groth [1, 12] and is based on the Taylor series approximation used in the $k$-exact reconstruction. The latter is only valid for smooth functions that are continuous up to a certain desired-order of the polynomial approximation $k$. The smoothness indicator $S$ is based on the smoothness parameter $\alpha$, and can be calculated as follows:

$$S = \frac{\alpha}{\max((1 - \alpha), \epsilon)} \frac{(SOS - DOF)}{DOF - 1},$$

where $\epsilon$ is a tolerance factor to avoid division by zero and is taken as $10^{-8}$. The variable $SOS$ represents the size of stencil used in the reconstruction, and $DOF$ represents the degree of freedom or known derivatives.

The smoothness parameter is represented by term $\alpha$ and is obtained from

$$\alpha = 1 - \frac{\sum_p (U^k_p(x_p) - U^k_i(x_p))^2}{\sum_p (U^k_p(x_p) - \bar{U}_i)^2},$$

where the other terms in Equation (2.21) are defined as follows:

- $U^k_p(x_p)$: is the reconstruction performed on cell $p$ and evaluated at center of cell $p$;
- $U^k_i(x_p)$: is the projected value of the reconstruction performed on cell $i$ and evaluated by extending $U^k_i(x_p)$ on the cell center $p$; and
- $\bar{U}_i$: is the average value of cell $i$.

Figure 2.1 shows the variation of $\alpha/(1 - \alpha)$. From the plot, it can be seen that $\alpha/(1 - \alpha)$ increases rapidly as $\alpha$ approaches unity. This would also cause the smoothness indicator $S$ to grow rapidly. Once the smoothness indicator $S$ is calculated, it is compared to a pass / no-pass cut-off limit where the typical values of $S$ are in the range of 1,000-5,000 [1, 12].
One-Dimensional Function Reconstruction

The solution reconstruction in each cell is important in evaluating the numerical fluxes at the cell boundaries. To gain a broader understanding of the proposed CENO reconstruction techniques described herein, reconstruction of a prescribed function has been performed here. From the reconstructed solutions obtained from the function, the formal level of accuracy of the CENO scheme can be assessed. The following function is used as the basis for assessing the one-dimensional reconstruction:

$$u(x) = \sin(2x) + 2\cos(x).$$  \hfill (2.22)

The range of $x$ was specified to be $-4 \leq x \leq 4$.

Second-order and fourth-order reconstructions using 20 cells were performed using Equation (2.22) and are illustrated in Figure 2.2(a) and Figure 2.2(b), respectively. Notice that both the reconstructed solutions have achieved solutions that correspond closely to the exact solution. Also,
Table 2.2: Orders of accuracy for the function reconstruction.

| $k$ | $|E|_1$ | $|E|_2$ |
|-----|--------|--------|
| 0   | -1.004 | -1.0010 |
| 1   | -2.0028 | -2.0034 |
| 2   | -3.0024 | -3.0031 |
| 3   | -4.0049 | -4.0055 |

the fourth-order reconstruction provides a better fit to the exact solution when compared to the second-order reconstruction.

Now that the second- and fourth- order reconstructed solutions have been obtained, the next stage would be to implement the slope limiters and the smoothness indicator. The Barth-Jesperson and Venkatakrishnan were implemented in the second-order scheme and are shown in Figure 2.3. Notice that at the extrema, both the slope limiters trigger the solution reconstruction to reduce to that of piecewise constant reconstruction in order to maintain monotonicity of the solution. A fourth-order CENO reconstruction solution has also been created for the above mentioned function and is illustrated in Figure 2.4. Notice the improved representation of the solution provided by the fourth-order scheme.

The L1 and L2 error norms based on the difference between the exact and reconstructed solutions were also obtained for the function reconstruction problem and are shown in Figure 2.5. The order of accuracy obtained from these reconstructions are summarized in the Table 2.2. The results verify that the correct level of accuracy is achieved for the various orders of reconstructions performed herein.

### 2.7 Flux Evaluation

The fluxes of mass, momentum, and energy from adjacent cells need to be evaluated at the faces of each cell volume. Solution to the Riemann problem is used in the evaluation of these fluxes. The fluxes are conserved as fluxes exiting a cell will enter the cell sharing the same face.
Figure 2.2: 20 cell function reconstruction
Figure 2.3: 2nd-order slope limiters

Figure 2.4: CENO reconstruction
Figure 2.5: Convergence study
2.7.1 The Riemann Problem

The Riemann problem is a special form of an initial-value problem such that the initial data is given by two piecewise constant states. The initial data has the form

\[
U_o(x) = U(x, t = 0) = \begin{cases} 
U_{left} & \text{if } x \leq 0, \\
U_{right} & \text{if } x > 0.
\end{cases}
\]  

(2.23)

By considering the initial-value problem depicted in Figure 2.6, the two adjacent cells are found to have a discontinuity in their solution states. This discontinuity must be resolved in order to compute the fluxes found in the Godunov-type finite-volume methods. Solution to this discontinuity can be achieved by using a variety of Riemann solvers.

2.7.2 Riemann Solvers

The exact solution to the Riemann problem for the Euler equations can be obtained by analyzing wave patterns composed of simple waves [40]. These waves can either be rarefaction waves, contact waves, or shock waves. The solution to the Riemann problem involves one of the five wave patterns as shown in Figure 2.7 where a shock is represented by S, a contact surface or entropy wave by C, and a rarefaction wave by R.
Chapter 2. CENO FV Scheme for Conservation Laws in 1D

2.7 Riemann Problem

Figure 2.7: Depiction of five possible solutions to the Riemann problem.

The Riemann solvers implemented herein are the exact Riemann solver developed by Gottlieb and Groth [41] and several approximate Riemann solvers. The approximate solvers include the Roe approximate Riemann solver [42], the HLLE Riemann solver [43] and the HLLL Riemann solver (Linde) [44]. An illustration of the performance of these solvers is provided in Section 2.9.

2.8 Time Marching Schemes

To achieve high-order accurate solution, implementation of both high-order time-marching schemes as well as high-order spatial reconstruction are important. In order to implement a high-order time-marching scheme, the Godunov-type method from Equation (2.5) can be re-expressed in the semi-discrete form

\[
\frac{d\mathbf{U}}{dt} = -\frac{1}{\Delta x} \left[ \mathcal{F}_{i+\frac{1}{2}} - \mathcal{F}_{i-\frac{1}{2}} \right].
\]  

Time-accurate solutions to the coupled nonlinear system of ordinary differential equations (ODEs) of the type given above can be obtained by integrating the solution in time by means of a standard time-marching methods [32]. Such time-marching methods can either be
explicit or implicit. Explicit time marching schemes use the information from a previous time step to solve for the solution at a current time step. Implicit methods use information from the current time step to solve for a solution at that time step. For this thesis, the time-marching schemes that have been implemented are all explicit Runge-Kutta methods [32].

2.8.1 Runge-Kutta Methods

In numerical analysis, Runge-Kutta methods are an important family of explicit and implicit time marching methods. They are a special subset of predictor-corrector methods [32]. The Runge-Kutta methods that have been implemented here are the second- and fourth-order explicit Runge-Kutta methods. The second-order Runge-Kutta time-marching method is a two-stage scheme of the form

\[
\begin{align*}
U^{n+1} &= U^n + \frac{1}{2}(k_1 + k_2), \\
k_1 &= \Delta t F(t^n, U^n), \\
k_2 &= \Delta t F(t^n + \frac{1}{2}\Delta t, U^n + \frac{1}{2} k_1),
\end{align*}
\]

(2.25)

where the various terms are

- \(U^n\) is the solution;
- \(U^{n+1}\) is the updated solution;
- the \(k_1\) and \(k_2\) terms are required for the second-order time marching scheme;
- \(\Delta t\) is the time step;
- \(F\) is a function of \(U\); and
- \(t\) is the time at which the time marching is applied.

The fourth-order Runge-Kutta time-marching scheme is a four-stage scheme and has the form

\[
\begin{align*}
U^{n+1} &= U^n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4), \\
k_1 &= \Delta t F(t^n, U^n), \\
k_2 &= \Delta t F(t^n + \frac{1}{2}\Delta t, U^n + \frac{1}{2} k_1), \\
k_3 &= \Delta t F(t^n + \frac{1}{2}\Delta t, U^n + \frac{1}{2} k_2), \\
k_4 &= \Delta t F(t^n + \Delta t, U^n + k_3),
\end{align*}
\]

(2.26)

The fourth-order scheme terms are similar to the terms used in second-order scheme with the addition of the \(k_3\) and \(k_4\) terms.
For the purpose of the present study, the accuracy of the time-marching scheme is made to match or be higher than the spatial order of accuracy. Therefore to achieve a second-order accurate scheme, the second-order spatial scheme requires the use of the second-order Runge-Kutta time-marching scheme. Similarly, the fourth-order spatial scheme requires the use of the fourth-order Runge-Kutta time-marching scheme.

### 2.8.2 Time Step and Courant Number

The stability of the explicit time-marching schemes are based on the appropriate selection of the time step, $\Delta t$. To calculate the time step, the Courant-Friedrichs-Lewy number (CFL) is used. The time-step $\Delta t$ is obtained as follows:

$$\Delta t = \text{CFL} \frac{\Delta x}{(|u| + a)_{\text{max}}}.$$  \hfill (2.27)

The CFL number in Equation (2.27) ensures that the waves from adjacent Riemann problems do not travel far enough to interact with one another. The CFL value used in this thesis is chosen to be either one or less than one in order to maintain the numerical stability of the conditionally-stable Runge-Kutta time marching scheme.

### 2.9 Numerical Predictions for One-Dimensional Shock-Tube Problem

In order to demonstrate the correctness of the implementation and performance of the high-order CENO finite-volume method applied to the one-dimensional form of the Euler equation, numerical predictions of a one-dimensional shock-tube problem are considered. The exact Riemann solver developed by Gottlieb and Groth [41] was used to evaluate the fluxes at the cell interfaces for this case. The length of the one-dimensional domain was set to 10 m with $0 \leq x \leq 10$ and a grid size of 100 cells was used. The initial conditions are provided in Table 2.3 and the simulation time for was in the range $0 \leq t \leq 7$ ms.

The density variations in the $x$-direction for the exact, second-order and fourth-order CENO solutions are illustrated in Figure 2.8. From these computed solutions, it is apparent that in general the CENO scheme more closely corresponds to the exact solution than the results of the second-order scheme. In addition, notice that the smoothness indicator within the CENO scheme successfully triggered limited piecewise linear reconstructions in areas that were found
Chapter 2. CENO FV Scheme for Conservation Laws in 1D

<table>
<thead>
<tr>
<th>Left state</th>
<th>Right state</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\rho_l = 4.696 \text{ kg/m}^3$</td>
<td>$\rho_r = 1.408 \text{ kg/m}^3$</td>
</tr>
<tr>
<td>$u_l = 0 \text{ m/s}$</td>
<td>$u_r = 0 \text{ m/s}$</td>
</tr>
<tr>
<td>$p_l = 404.4 \text{ kPa}$</td>
<td>$p_r = 101.1 \text{ kPa}$</td>
</tr>
</tbody>
</table>

Table 2.3: Initial conditions for shock-tube problem.

Figure 2.8: Reconstructed solutions to a shock-tube problem

to be unresolved or discontinuous so maintain solution monotonicity. Figure 2.9 illustrates the second-order and CENO scheme for the primitive variables $p, \rho$ and $u$.

Note that for the one-dimensional finite-volume scheme, various Riemann solvers as discussed in Section 2.7.2, were implemented to evaluate the fluxes at the cell interfaces. Figure 2.10 illustrates the solutions obtained using these solvers with the same initial value problem discussed above.
Figure 2.9: Reconstructed solutions to a shock-tube problem
Figure 2.10: Riemann solvers used for the shock-tube problem
Chapter 3

Three-Dimensional Unstructured Mesh

This chapter provides an overview of the mesh generation process for three-dimensional unstructured meshes and the Gmsh tool used to create tetrahedral meshes used in this thesis.

3.1 Discretization of Physical Domain and Mesh Generation

A mesh is a spatial discretization of a geometric or physical domain into smaller shapes or elements [13, 17, 27, 45, 46]. For CFD, as with other applications, meshes can be defined for either two- or three-dimensional geometries, depending on the problem. Three-dimensional meshes are created for finite-volume, finite-element and finite difference methods and consist of tetrahedra, hexahedra or pyramids. Prism structures can also be used. In general, the two types of meshes are [17]:

- structured meshes; and
- unstructured meshes.

Structured meshes are characterized by the implicit arrangement of data that can either be expressed as a two- or three-dimensional array. The cell-element choice for two-dimensional meshes are quadrilaterals and that for three-dimensions are hexahedra. Because of this particular cell arrangement, the cells can easily be connected with their neighbouring cells.
Unstructured meshes are characterized by the explicit definition of the connectivity of the points within the grid \[17, 47\]. The cell-element choices for two-dimensional meshes are triangles or hexagons, and that for three-dimensional meshes are tetrahedra, pyramids and/or prisms. Because of the unstructured cell-data arrangement, building cell-connections between the neighbouring cells must be performed manually. Thus, the development solution methods that make use of unstructured meshes is generally much more complex and difficult when compared to methods for structured meshes.

The advantage of using unstructured meshes over structured meshes is the greater flexibility offered in dealing with complex geometries \[17\]. Therefore, unstructured meshes are more commonly used when solving computational fluid dynamics problems for practical engineering applications \[13, 17, 27, 45, 46, 48\].

### 3.2 Unstructured Mesh Generation

Mesh generation, or grid generation, is performed by creating a polygonal or polyhedral mesh that approximates the geometric domain. A range of unstructured mesh generating tools have been developed to date that make use of a variety of meshing algorithms \[17, 49\].

For this research, the Gmsh tool has been selected to generate three-dimensional unstructured meshes. Gmsh was developed by Geuzaine and Remacle \[49\]. It is a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities \[49\]. It also has several unstructured mesh generation algorithms implemented that will be discussed in the following sections.

#### 3.2.1 Overview of Gmsh

Gmsh is a three-dimensional grid generator that has a built-in computational aided design(CAD) engine and a post-processor. Gmsh was developed by Geuzaine and Remacle \[49\] and has been designed to provide a fast, light, and user friendly meshing experience. Gmsh also includes parametric input as well as advanced visualization capabilities available \[49\].

The main advantages of using Gmsh is that it can either provide an interactive Graphical User Interface (GUI) to create a mesh or a scripting language can be used to automate the mesh generating process. The automated meshing process via the scripting language is achieved by performing loops, applying conditions and carrying out external system calls.
Gmsh is composed of four major modules which are as follows:

1. geometry module;
2. mesh generation module;
3. finite-element solver; and
4. post processor.

The geometry module and the mesh generation module will be of particular interest in this research. The geometry module is used to define the geometry of a three-dimensional object. The mesh generating module is used to convert this geometric object to a three-dimensional unstructured mesh.

### 3.2.2 Geometry Module

The geometry module in Gmsh uses a boundary representation to describe both two-and three-dimensional geometries. Geometric models are created using a bottom-up flow technique. First,
points are defined to create a layout of the geometric model. Once the layout has been created, the points are connected by oriented lines such as line segments and splines to form a one-dimensional profile. For a two-dimensional profile, a surface is developed by connecting a group of these oriented lines. In the case of a three-dimensional profile, a volume is defined by a group of surfaces developed. Finally, physical groups can be defined based on the groups of the elementary geometric entities. These geometric entities can be fully parametrized using the scripting language in Gmsh.

As this thesis focuses on three-dimensional geometric models for unstructured mesh generations, developing these models requires defining points, lines, surfaces and volumes. To illustrate a complex geometry that has curved boundaries and sharp ends, a two-dimensional profile is first created. This profile is depicted in Figure 3.1. The two-dimensional profile is then extended to three-dimensions using a group of surface profiles. The resulting three-dimensional model is illustrated in Figure 3.2.
3.2.3 Mesh Generation Module

The Gmsh mesh module can create both two-dimensional and three-dimensional unstructured meshes. The elementary geometrical elements in these unstructured meshes are defined only by an ordered list of their nodes. Therefore, there is no predefined order on the relationship between any two elements.

The mesh generation uses various algorithms (to be discussed in the section to follow) and is performed using a similar bottom-up technique as described for the geometric module. First, the lines are discretized. These lines are then used to mesh the surface. For three-dimensional geometric models, the meshed surfaces are then used to mesh the volume.

To illustrate the generation of a three-dimensional mesh, the geometric model developed in the previous section is again considered. An example of the surface mesh created using Gmsh is illustrated in Figure 3.3. Using the surface meshes, a three dimensional volume mesh was then created created. The resulting mesh is depicted in Figure 3.4. The mesh is made up of 3,698 tetrahedra with 478 points and 952 faces. An overview of the various algorithms used in generating three-dimensional meshes is discussed in the next section.
3.2.4 Mesh Generation Procedure

As this thesis is focused on three-dimensional unstructured meshes, only the three-dimensional algorithms available within Gmsh will be reviewed here. There are two unstructured mesh generation algorithms available in Gmsh:

1. the Delaunay algorithm; and
2. the frontal algorithm.

The Delaunay algorithm makes use of two steps to develop the three-dimensional unstructured mesh. First, an initial mesh is created by using the Tetgen algorithm developed by Si [50]. Then a three-dimensional version of the Delauney algorithm is applied to the initial mesh.

The frontal algorithm makes use of the the Netgen algorithm developed by Schoeberl [51]. The Netgen algorithm generates an unstructured mesh by solving four sub-problems. These sub-problems are defined as:

1. special point calculation;
2. edge following;

3. surface meshing; and

4. volume meshing.

For details of the solution of the latter refer to the work by Schoeberl [51].

3.3 Unstructured Tetrahedral Mesh for Cubed Domain

The Delaunay algorithm was selected as the main mesh generating algorithm for this thesis because of its robustness and relative speed [49]. Using this algorithm, two simple cubic meshes were created and used in the evaluation of the high-order CENO reconstruction procedure to be described later in Chapter 5 of the thesis. The boundaries of the first cube are defined by the corners located at (0,0,0) and (1,1,1) and its geometry is illustrated in Figure 3.5(a). The boundaries of the second cube are defined by the corners located at (-1,-1,-1) and (1,1,1) and its geometry is illustrated in Figure 3.5(b). Figures 3.5(c) and 3.5(d) show a generic surface mesh and the final volume mesh created using the geometry of the first cube.
Figure 3.5: Three-dimensional cubed geometries and unstructured tetrahedral mesh used in the function reconstruction considered in this thesis.
Chapter 4

High-Order CENO Reconstruction Scheme For Three-Dimensional Unstructured Mesh

This chapter considers the extension of the CENO high-order reconstruction scheme of Ivan and Groth \[1, 12, 13, 14\] to three-dimensional unstructured meshes. The contents of this chapter has been divided into three sections. Section 4.1 provides a overview of Godunov-type finite-volume methods for three-dimensional meshes for which the CENO reconstruction scheme has been developed. Section 4.2 provides a summary of the \(k\)-exact least-square reconstruction scheme developed by Barth \[25\] and its current implementation for three-dimensional unstructured meshes. Finally, Section 4.3 provides an overview of the CENO scheme for three-dimensional unstructured meshes.

4.1 Overview of Godunov-Type Finite-Volume Scheme for Three-Dimensional Flows

This section provides an overview of the Godunov-type finite-volume methods, as discussed in Chapter 2, applied to the solution of the Euler equations on three-dimensional unstructured meshes. The three-dimensional integral form of the Euler equations can be written as

\[
\iiint_{V} \left[ \frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F} \right] dV = 0, \tag{4.1}
\]
where \( V \) is the volume of the region of interest and \( \mathbf{U} \) represents the vector of conserved solution variables given by
\[
\mathbf{U} = [\rho \quad \rho u \quad \rho v \quad \rho w \quad E]^T.
\] (4.2)

The term \( \mathbf{F} = (\mathbf{F}, \mathbf{G}, \mathbf{H}) \) is the inviscid flux dyad and has the components
\[
\mathbf{F} = 
\begin{bmatrix}
\rho u \\
\rho u^2 + p \\
\rho u v \\
\rho u w \\
u(E + p)
\end{bmatrix},
\mathbf{G} = 
\begin{bmatrix}
\rho v \\
\rho v^2 + p \\
\rho v w \\
\rho v w \\
v(E + p)
\end{bmatrix},
\mathbf{H} = 
\begin{bmatrix}
\rho w \\
\rho w^2 + p \\
\rho w v \\
\rho w w \\
w(E + p)
\end{bmatrix}.
\] (4.3)

The elements of the vectors appearing in Equation (4.3) are defined as follows:

- \( u, v, \) and \( w \) are the velocity components in the respective \( x, y \) and \( z \) directions;
- \( p \) is the pressure;
- \( \rho \) is the density;
- \( E \) is the total specific energy where \( E = e + \frac{1}{2} |\mathbf{u}|^2 \); and
- \( e \) is the specific internal energy.

For the three-dimensional unstructured mesh considered herein, tetrahedral elements are used. A schematic of a tetrahedral element is illustrated in Figure 4.1. Using a \( N_g \)-point Gaussian quadrature numerical integration, the semi-discrete form of the Euler equations applied to a tetrahedral cell or volume element, \( i \), can be written in the following compact form
\[
\frac{d\mathbf{U}_i}{dt} = -\frac{1}{V_i} \sum_{l=1}^{N_f} \sum_{m=1}^{N_g} (\mathbf{F} \cdot \mathbf{n}_l A)_{i,l,m},
\] (4.4)

where \( V_i \) is the volume of cell \( i \), \( N_f \) is the number of faces (for tetrahedral, \( N_f = 4 \)), \( \mathbf{n}_l \) is the normal unit vector pointing away from the face area \( A \). The semi-discrete form of Equation (4.4) represents the application of a Godunov-type finite-volume method to a hyperbolic system in three dimensions. It is analogous to Equation (2.24) of Chapter 2 for the one-dimensional case. The scheme again consists of the following three stages: performing cell reconstruction, evaluating fluxes and applying a time marching scheme. The remainder of this chapter will primarily focus on the cell reconstruction stage.
4.2 \( k \)-Exact Least-Square Reconstruction

The finite-volume procedure outlined above can make use of the \( k \)-exact reconstruction developed by Barth and Fredrickson [25, 37] to achieve high-order spatial accuracy. The \( k \)-exact reconstruction uses the \( k^{th} \)-order Taylor series expansion for the centre of the cell \( i \) being reconstructed. In solving the Euler equations using the semi-discrete method of Equation (4.4), the \( k^{th} \)-order reconstruction would result in a \((k+1)\)-order accurate scheme in terms of spatial accuracy.

As previously indicated, development of the \( k \)-exact reconstruction technique for three-dimensional unstructured meshes is of particular interest. The \( k \)-exact higher-order reconstruction for the solution in cell \( i \) can be written as

\[
U_i^k(x, y, z) = \sum_{p_1=0}^{N_1} \sum_{p_2=0}^{N_2} \sum_{p_3=0}^{N_3} (x - x_i)^{p_1} (y - y_i)^{p_2} (z - z_i)^{p_3} D_{p_1p_2p_3} \quad (4.5)
\]

The various terms in Equation (4.5) are summarized as follows:
$U^k_i$ : represents the primitive solution variable of the cell being reconstructed;

$x_i, y_i, z_i$ : are the coordinates of the cell centroid of cell $i$;

$k$ : determines the order of the polynomial interpolant;

$N_1, N_2, N_3$ : are the upper bounds of the summations that must satisfy the condition that $(N_1 + N_2 + N_3) \leq k$; and

$D_{p_1p_2p_3}$ : are the unknown derivatives (polynomial coefficients) that are derived from the expansion of the Taylor series in three-dimensions.

The order of accuracy of the reconstruction performed is based on the number of derivatives, $N$, in the representation of the solution. The number of derivatives are found using

$$N = \frac{1}{d!} \prod_{n=1}^{d} (k + n), \quad (4.6)$$

where $d$ represents the number of spatial dimensions. For three dimensions, $d = 3$ and the number of derivatives, $N$, would become

$$N = \frac{(k + 1)(k + 2)(k + 3)}{6} \quad (4.7)$$

Since the order of accuracy exhibited is equal to $k+1$, a second-order scheme would have four unknown derivatives and a fourth scheme would have twenty unknown derivatives.

To determine these unknown derivatives, the following conditions must be satisfied by the $k$-exact reconstruction [37]:

1. the solution should be a polynomial of degree $N \leq k$;

2. the cell average value must be preserved; and

3. the reconstruction procedure should have a compact support.

Using the information provided from neighbouring cells, a polynomial of degree $N \leq k$ can easily be created. For the case of structured meshes, a fixed group of neighbouring cells can be used as was shown previously by Rashad [26]. This grouping is called the “reconstruction stencil”. For unstructured meshes, there is no easily identifiable fixed group of cells and thus the stencil must be manually developed by linking the neighbouring cells. Further discussion on the determination of the reconstruction stencil for unstructured mesh is found in Section 4.2.1.
In order to preserve the cell average values of the solution in each cell, the following constraint must be imposed on the reconstruction:

$$\bar{U}_i = \frac{1}{V_i} \iiint_{V_i} U_k^i(x, y, z) \, dV$$ \hspace{1cm} (4.8)

Finally, for “compact support” system, the neighbouring cells used in the reconstruction stencil should be those nearest to the cell being reconstructed. A further discussion on “compact support” is also provided in Section 4.2.1.

### 4.2.1 Determination of Reconstruction Stencil

In multi-dimensional meshes, the $k$-exact reconstruction process leads to an over-determined system of equations as described by Barth [37]. By using a least-squares approach the error taken over the entire reconstructed stencil can be minimized. One of the challenges for unstructured mesh is to determine this reconstruction stencil. This challenge can be overcome by establishing a link between neighbouring cells and the next to the nearest neighbouring cells.

For structured meshes, determining the neighbours and next to the nearest neighbours would be a straightforward process. A simple $(i, j, k)$ cell block system can be used as shown in Figure 4.2(a). The neighbouring cells are allocated by taking steps either to the left, right, forward or backward from the coordinates of the reconstructed cell. This procedure is illustrated for the two-dimensional structured mesh face of Figure 4.2(a). Notice that cell $i$ has 8 neighbouring cells and 16 next to the nearest neighbouring cells.

For an unstructured mesh, stencil selection algorithm is required to overcome the challenge of obtaining the neighbouring and next to the nearest neighbouring cells. The generic unstructured mesh illustrated in Figure 4.2(b) shows the cell of interest represented by the cell that is coloured red. In addition there are also two rings of cells surrounding the red cell. The first ring consists of the nearest neighbouring cells and is represented by the green coloured cells. The second ring consists of the next to the nearest neighbouring cells and is represented by the blue coloured cells.

To determine all the neighbouring cells surrounding the red cell in the unstructured mesh, an algorithm has been developed using a structured pointer storage system. The algorithm works by checking if the face of a neighbouring cell is shared with the cell of interest. Once a match is determined, both cells store a pointer linking them as neighbouring cells. This algorithm continues for all the cells in the mesh and adds their neighbours using the shared face technique. Once all these pointers are stored, a data structure connects the neighbouring
cells and the next to the nearest neighbouring cells. For a fourth-order accurate scheme, at least twenty cells are required to successfully achieve the reconstruction of the solution. Using just the nearest neighbouring cells and the next to nearest neighbouring cells is just not sufficient to produce an overdetermined system of equations. Therefore an additional third ring of cells is required to satisfy the requirements for an overdetermined system of equation. This third ring includes the nearest neighbours of the blue ring of cells illustrated in Figure 4.2(b).

4.2.2 Determination of the Derivatives for the Reconstruction

To determine the derivatives for the reconstruction, the following three steps must be performed:

(i) construction of the overdetermined system of equations;

(ii) calculation of the geometric coefficients and moments; and

(iii) solution of the overdetermined system of equations using the least-squares approach.

The overdetermined system can be expressed in the form $A\mathbf{x} - \mathbf{b} = \mathbf{c}$. Here, $A$ is a coefficient matrix, $\mathbf{b}$ is a vector of the right-hand sides of the equations, $\mathbf{x}$ is the unknown vector of derivatives and $\mathbf{c}$ is the error vector. Because the length of unknown vector $\mathbf{x}$ is less than the rows of $A$ and the length of $\mathbf{b}$, a unique solution does not exist (i.e., the system is overdetermined). Since, there is no unique solution to the system of equations, a least-squares approach is used.
to solve this system of equations. In this research, the “dgelsy” routine found in the Linear
Algebra Package (LAPACK) [52] libraries is used in the solution of the least-squares problem.

The Overdetermined System

Let the cell of interest that is reconstructed be represented by cell \( i \) and the neighbouring cells
by cell \( j \). The reconstruction polynomial of cell \( i \) for three-dimensional meshes can be expressed
as

\[
U^k_i(x, y, z) = \sum_{p_1=0}^{k} \sum_{p_2=0}^{k} \sum_{p_3=0}^{k} (x - x_i)^{p_1} (y - y_i)^{p_2} (z - z_i)^{p_3} D_{p_1p_2p_3}
\] (4.9)

where the cell-averaged value of cell \( i \) is defined by

\[
\bar{U}_i = \frac{1}{V_i} \iiint_{V_i} U^k_i(x, y, z) \, dV,
\] (4.10)

and the neighbouring cell-averaged values are given by the subscript \( j \) and are determined as follows

\[
\bar{U}_j = \frac{1}{V_j} \iiint_{V_j} U^k_i(x, y, z) \, dV.
\] (4.11)

By substituting Equation (4.9) into Equation (4.10), the cell-averaged value of cell \( i \) can be
written as

\[
\bar{U}_i = \frac{1}{V_i} \iiint_{V_i} \left[ \sum_{p_1=0}^{k} \sum_{p_2=0}^{k} \sum_{p_3=0}^{k} (x - x_i)^{p_1} (y - y_i)^{p_2} (z - z_i)^{p_3} D_{p_1p_2p_3} \right] \, dV,
\] (4.12)

and by substituting Equation (4.9) into Equation (4.11), the cell-averaged value of the neigh-
bouring cell \( j \) can be written as

\[
\bar{U}_j = \frac{1}{V_j} \iiint_{V_j} \left[ \sum_{p_1=0}^{k} \sum_{p_2=0}^{k} \sum_{p_3=0}^{k} (x - x_i)^{p_1} (y - y_i)^{p_2} (z - z_i)^{p_3} D_{p_1p_2p_3} \right] \, dV.
\] (4.13)

Because the unknown derivatives are constants, Equation (4.13) can be rearranged by trans-
ferring the the derivatives outside of the integrand. Thus, by doing so we achieve the following
form for the neighbouring cells \( j \)
\[
\bar{U}_j = \sum_{p_1=0}^{k} \sum_{p_2=0}^{k} \sum_{p_3=0}^{k} \left[ D_{p_1 p_2 p_3} \cdot \frac{1}{V_j} \iiint_{V_j} (x - x_i)^{p_1} (y - y_i)^{p_2} (z - z_i)^{p_3} \right] \, dV.
\] (4.14)

The first term of the reconstruction is represented by \( D_{000} \). This term represents the value of the reconstruction at the cell centre. It is important to note that because the reconstruction is of high-order, the cell centred value is not to be equal to the cell-averaged value \( (D_{000} \neq \bar{U}_i) \).

By determining an expression for \( D_{000} \) in terms of the other derivatives, the number of unknown derivatives can be reduced by one. This ensures that the average value of cell \( i \) is recovered when an integration over the cell is performed.

By modifying the terms in Equation (4.12), the following expression for the cell-averaged value of cell \( i \) is formed

\[
\bar{U}_i = D_{000} + \sum_{p_1=0}^{k} \sum_{p_2=0}^{k} \sum_{p_3=0}^{k} D_{p_1 p_2 p_3} \cdot \frac{1}{V_i} \iiint_{V_i} (x - x_i)^{p_1} (y - y_i)^{p_2} (z - z_i)^{p_3} \, dV.
\] (4.15)

The equation above can be expressed using the following geometric moments

\[
(x^{p_1} y^{p_2} z^{p_3})_i = \frac{1}{V_i} \iiint_{V_i} (x - x_i)^{p_1} (y - y_i)^{p_2} (z - z_i)^{p_3} \, dV.
\] (4.16)

A further discussion of geometric moments is given later in this section. Using the geometric moments of Equation (4.16) and substituting into Equation (4.15), the cell-averaged value for cell \( i \) can be written as

\[
\bar{U}_i = D_{000} + \sum_{p_1=0}^{k} \sum_{p_2=0}^{k} \sum_{p_3=0}^{k} D_{p_1 p_2 p_3} \cdot (x^{p_1} y^{p_2} z^{p_3})_i
\] (4.17)

By re-arranging the terms in Equation (4.17), an expression for \( D_{000} \) can be obtained as follows

\[
D_{000} = \bar{U}_i - \sum_{p_1=0}^{k} \sum_{p_2=0}^{k} \sum_{p_3=0}^{k} D_{p_1 p_2 p_3} (x^{p_1} y^{p_2} z^{p_3})_i
\] (4.18)

The cell-averaged value of the solution for the neighbouring cell \( j \) can also be written as follows
Finally, an expression for the complete system of equations is achieved and can be written as

$$\bar{U}_j = \sum_{p_1=0}^{k} \sum_{p_2=0}^{k} \sum_{p_3=0}^{k} D_{p_1p_2p_3} \cdot \left[ (x^{p_1}y^{p_2}z^{p_3})_j \right]$$  \hspace{1cm} (4.19)

where the term $\left( x^{p_1}y^{p_2}z^{p_3} \right)_j$ is of the form

$$\left( x^{p_1}y^{p_2}z^{p_3} \right)_j = \frac{1}{V_j} \iiint_{V_j} (x-x_i)^{p_1}(y-y_i)^{p_2}(z-z_i)^{p_3} \, dV. \quad (4.20)$$

The difference in cell-averaged values between the cell of interest and the neighbouring cell can then be expressed as

$$\bar{U}_j - \bar{U}_i = \sum_{p_1=0}^{k} \sum_{p_2=0}^{k} \sum_{p_3=0}^{k} D_{p_1p_2p_3} \cdot \left[ (x^{p_1}y^{p_2}z^{p_3})_j - (x^{p_1}y^{p_2}z^{p_3})_i \right]$$  \hspace{1cm} (4.21)

The Equation (4.21) can further be expressed using

$$\left( x^{p_1}y^{p_2}z^{p_3} \right)_{ij} = \left[ (x^{p_1}y^{p_2}z^{p_3})_j - (x^{p_1}y^{p_2}z^{p_3})_i \right]$$  \hspace{1cm} (4.22)

Finally, an expression for the complete system of equations is achieved and can be written as

$$\bar{U}_j - \bar{U}_i = \sum_{p_1=0}^{k} \sum_{p_2=0}^{k} \sum_{p_3=0}^{k} D_{p_1p_2p_3} \cdot \left[ (x^{p_1}y^{p_2}z^{p_3})_{ij} \right]$$  \hspace{1cm} (4.23)

For the case of an overdetermined system, a system of linear equations of the form $Ax - b = c$ can be found based on Equation (4.23). This $M \times N$ overdetermined system can be written as

$$
\begin{bmatrix}
(x^{0}y^{0}z^{1})_{i1} & \cdots & (x^{1}y^{1}z^{2})_{i1} & \cdots & (x^{k}y^{0}z^{0})_{i1} \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
(x^{0}y^{0}z^{1})_{ij} & \cdots & (x^{1}y^{1}z^{2})_{ij} & \cdots & (x^{k}y^{0}z^{0})_{ij} \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
(x^{0}y^{0}z^{1})_{iM} & \cdots & (x^{1}y^{1}z^{2})_{iM} & \cdots & (x^{k}y^{0}z^{0})_{iM}
\end{bmatrix}
\begin{bmatrix}
D_{001} \\
\vdots \\
D_{k00}
\end{bmatrix}
\begin{bmatrix}
(\bar{U}_1 - \bar{U}_i) \\
\vdots \\
(\bar{U}_M - \bar{U}_i)
\end{bmatrix}
= 
\begin{bmatrix}
c_1 \\
\vdots \\
N \times 1
\end{bmatrix} - 
\begin{bmatrix}
M \times N \\
N \times 1 \\
M \times 1 \\
M \times 1
\end{bmatrix} \begin{bmatrix}
c_1 \\
\vdots \\
N \times 1
\end{bmatrix}.$$
where \( M \) represents the total number of neighbouring cells in the reconstruction stencil and \( N \) represents the total number of unknown derivatives (minus the expression for \( D_{000} \)).

Once the overdetermined system has been identified, geometric weighting can be applied to each equation. Geometric weighing is specific to each control volume with the purpose of improving the locality of the reconstruction. The geometric weights, \( w_{ij} \), are multiplied with their corresponding terms in matrix \( A \) and vector \( b \). Geometric weights typically used are either the inverse distance or the inverse distance squared weighting as given by

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

or

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

respectively, where \( \vec{r} \) is the position vector from the indicated cell. The next subsection will discuss the geometric coefficients and moments that need to be calculated to solve the overdetermined system.

**Geometric Coefficients and Moments**

Matrix \( A \) depends entirely on the geometry of the cells in the stencil. Hence, to reduce computational cost and memory usage, an efficient approach is required to evaluate the components of this matrix. This approach would involve pre-computing and storing only the geometric coefficients of each cell about its own cell centre. Consider the following equation

\[
\left(\frac{x^{p_1}y^{p_2}z^{p_3}}{\nu_j}\right)_j \equiv \frac{1}{V_j} \iiint_{V_j} (x - x_j)^{p_1} (y - y_j)^{p_2} (z - z_j)^{p_3} \, dV.
\]

The pre-calculated stored coefficients can be used to calculate the elements found in matrix \( A \), using the appropriate powers \((p_1, p_2, \text{and } p_3)\) of \( \left(\frac{x^{p_1}y^{p_2}z^{p_3}}{\nu_j}\right)_j \). The elements of matrix \( A \) can once again be written as

\[
\left(\frac{x^{p_1}y^{p_2}z^{p_3}}{\nu_j}\right)_{ij} = \frac{1}{V_j} \iiint_{V_j} (x - x_i)^{p_1} (y - y_i)^{p_2} (z - z_i)^{p_3} \, dV - \left(\frac{x^{p_1}y^{p_2}z^{p_3}}{\nu_i}\right)_{ij}.
\]
\[
\left( x^{p_1} y^{p_2} z^{p_3} \right)_{ij} = \sum_{\xi=0}^{p_1} \sum_{\ell=0}^{p_2} \sum_{\varphi=0}^{p_3} \left[ C_\xi^{p_1} C_\ell^{p_2} C_\varphi^{p_3} \cdot \Delta x_j^{\xi} \Delta y_j^{\ell} \Delta z_j^{\varphi} \cdot \left( x (p_1 - \xi) y (p_2 - \ell) z (p_3 - \varphi) \right)_j \right.
\]
\[
- \left. \left( x^{p_1} y^{p_2} z^{p_3} \right)_i \right]. \quad (4.28)
\]

By examining Equation (4.28), the RHS only contains the geometric coefficients of neighboring cells taken about their own cell centres where the term \(\left( x (p_1 - \xi) y (p_2 - \ell) z (p_3 - \varphi) \right)_j\) in Equation (4.28) is of the form
\[
\left( x (p_1 - \xi) y (p_2 - \ell) z (p_3 - \varphi) \right)_j \equiv \frac{1}{V_j} \iiint_{V_j} (x - x_j)^{p_1 - \xi} (y - y_j)^{p_2 - \ell} (z - z_j)^{p_3 - \varphi} \, dV. \quad (4.29)
\]

Notice that Equation (4.29) is simply just a shift of the stored powers of the \(\left( x^{p_1} y^{p_2} z^{p_3} \right)_j\) term. Therefore computing the elements of matrix A can be seen to be a rather straightforward process.

To compute the \(\left( x^{p_1} y^{p_2} z^{p_3} \right)_j\) terms for a three-dimensional unstructured mesh, an integral would have to be performed over the tetrahedral elements. This can be achieved by obtaining the quadrature points and applying a high-order numerical quadrature formula. Further discussion of the tetrahedral quadrature rules is given in Section 4.2.3 to follow.

Finally, the coefficients of the binomial expansion can be computed in a recursive manner, with initial coefficients \(C_0^{p_1} = C_0^{p_2} = C_0^{p_3} = 1\). The remaining coefficients are found as follows:
\[
C_\xi^{p_1} = \frac{p_1 - \xi + 1}{\xi} C_{\xi-1}^{p_1}, \quad C_\ell^{p_2} = \frac{p_2 - \ell + 1}{\ell} C_{\ell-1}^{p_2}, \quad C_\varphi^{p_3} = \frac{p_3 - \varphi + 1}{\varphi} C_{\varphi-1}^{p_1}. \quad (4.30)
\]

### 4.2.3 Quadrature Rules

Quadrature rules give an approximation of the definite integral of a function. A quadrature rule is set of quadrature points \(i\) and weights \(W_i\) such that an integral of a function \(f(x, y, z)\) over the region of interest can be performed. The quadrature rule for performing an integral over a canonical tetrahedral element \(\Omega_e\) has the form
\[
\iiint_{\Omega_e} f(x, y, z) \, dxdydz = V_e \sum_{i=1}^{n} W_i f(\zeta_1^i, \zeta_2^i, \zeta_3^i, \zeta_4^i) + E \quad (4.31)
\]
where \(V_e\) is the volume of the element \(e\), the \((\zeta_1^i, \zeta_2^i, \zeta_3^i, \zeta_4^i)\) are the tetrahedral coordinates and \(E\) is the error from approximating the integration \((E = 0\) for exact integration of the integrand).
Figure 4.3: Quadrature points on a tetrahedron

Quadrature rules for integrating polynomials of orders one through six are described in the research conducted by Jinyun [53] and for orders up to eight are described by Keast [54]. An example of a tetrahedron with fifteen quadrature points is illustrated in Figure 4.3. Approximation of a definite integral of an arbitrary function using these fifteen quadrature points leads to a fifth-order accurate solution.

4.3 High-Order CENO Reconstruction

As for the one-dimensional case, discussed in Chapter 2, high-order central essentially non-oscillatory finite-volume scheme developed by Ivan and Groth [1, 12] can be implemented for three-dimensional unstructured meshes using the following steps:

1. first, an unlimited \( k \)-exact reconstruction is performed for all the cells;

2. the smoothness indicator is then evaluated for each cell to determine if the solution is continuous or discontinuous; and

3. finally, a piecewise linear reconstruction is applied to the cells that are flagged as discontinuous.
Note that a three-dimensional CENO scheme for structured meshes was previously implemented by Rashad [26] and a two-dimensional variation of the CENO scheme for unstructured meshes was implemented by McDonald [28]. This section is focuses on extending the implementation of the CENO reconstruction scheme to three-dimensional unstructured meshes.

4.3.1 Hybrid Reconstruction Scheme

As described previously, the reconstruction performed in the CENO scheme uses a hybrid solution reconstruction process. First a high-order $k$-exact least-squares reconstruction technique by Barth [25] is performed over a fixed central stencil. If the reconstruction is found to be discontinuous by the smoothness indicator, a second-order limited piecewise linear reconstruction is performed. To enforce monotonicity, the limited piecewise linear reconstruction uses the limiters of either Barth-Jesperson or Venkatakrishnan. These slope limiters have been discussed in Chapter 2 of this thesis. The next section will review the implementation of the smoothness indicator for three-dimensional meshes.

4.3.2 Smoothness Indicator in Three Dimensions

The smoothness indicator, $S$, as developed by Ivan and Groth [1, 12] for the CENO scheme and described in Chapter 2 is again used in the three-dimensional case. The smoothness indicator validates functions that are continuous up to some desired order of the polynomial approximation $k$. For the fully three-dimensional case, the smoothness indicator can be calculated as follows:

$$S = \frac{\alpha}{\max((1 - \alpha), \epsilon)} \frac{(SOS - DOF)}{DOF - 1}$$  \hspace{1cm} (4.32)

where the smoothness parameter is represented by $\alpha$. For three-dimensional meshes, $\alpha$ can be calculated as follows:

$$\alpha = 1 - \frac{\sum_p(U_p^k(x_p, y_p, z_p) - U_i^k(x_p, y_p, z_p))^2}{\sum_p(U_p^k(x_p, y_p, z_p) - \bar{U}_i)^2}$$  \hspace{1cm} (4.33)

where the various terms in Equation (4.33) are

$U_p^k(x_p, y_p, z_p)$ : is the reconstruction performed on cell $p$ and evaluated at center of cell $p$;

$U_i^k(x_p, y_p, z_p)$ : is the projected value of the reconstruction performed on cell $i$ evaluated on cell center $p$; and
$\bar{U}_i$: is the average value of cell $i$.

Once the smoothness indicator $S_c$ is calculated, it can once again be compared to a pass/no-pass cut-off limit that is in the range of 1,000-5,000 [1, 12]. Note that when $S_c$ is calculated in smooth regions, it is typically found to be orders of magnitude larger than the cutoff value and this has also been confirmed by the previous numerical experiments in three dimensions.
Chapter 5

Numerical Results for Three-Dimensional CENO Reconstruction on Unstructured Mesh

Important elements required to establish the credibility and performance of high-order schemes are the verification and validation processes [55]. The verification process is defined as the “process of determining that a model implementation accurately represents the developer’s conceptual description of the model and solution” [55] and the validation process is defined as the “process of determining the degree to which a model is an accurate representation of the real world from the prospective of the intended uses of the model” [55]. This chapter will present only the verification process for the CENO reconstruction scheme applied to three-dimensional unstructured meshes and will make use of several test cases from the previous work of Ivan and Groth [12, 24].

This chapter has been divided into three sections. The first section, Section 5.1, focuses on the methods used to conduct the accuracy assessment and convergence studies for the low-order and high-order schemes. The second, Section 5.2, presents the method used in the visualization of the high-order reconstructed solutions for three-dimensional unstructured meshes; and for the last section, Section 5.3, the reconstruction of several predefined functions is considered using the CENO approach applied to three-dimensional unstructured mesh. For the latter, the reconstruction of both continuous and discontinuous functions are examined.
5.1 Verification Process for High-Order Reconstruction

Various methods for verifying high-order methods have previously been developed and a few of these methods have been described by Zingg [56] and Roach [57]. For this thesis, a convergence study is used to assess the levels of accuracy achieved by the low-order and high-order schemes.

5.1.1 Determining the Order of Accuracy

An important step that is required to determine the order of actual numerical accuracy is to show the expected order of convergence. The order of convergence can be obtained by monitoring the $L_1$ and $L_2$ error norms with an increase in grid size. By developing a log-log plot of increase in grid size versus the error norms gives a slope that shows the level of accuracy achieved.

Therefore, to obtain the convergence study, two important components need to be successfully obtained. First, an exact solution of the reconstructed cell should be known, or at least estimated. Using the exact solution, cell average values can be calculated for each computational cell using the following relation:

$$U_i = \frac{1}{V_i} \iiint_{V_i} U_{\text{exact}}(x, y, z) dV,$$

(5.1)

The second important component consist of determining the $L_1$ and $L_2$ error norms of the solution error based on the reconstructed and the exact solutions. Error norm evaluation is further discussed in Section 5.1.2 below.

For three-dimensional unstructured meshes, it is important to note that increasing the grid size is achieved by decreasing characteristic length in the grid generating software Gmsh [49]. The same characteristic length is used to evaluate the error norms for all the orders of reconstruction.

5.1.2 Accuracy Assessment

The objective is to assess the level of solution accuracy for the various reconstructed solutions. This is achieved by performing reconstruction on several functions that may be either continuous or discontinuous and by obtaining the $L_1$ and $L_2$ error norms. These error norms are computed based on the difference between the exact and reconstructed solutions.

Thus for the reconstructed solutions, the $L_1$ and $L_2$ error-norms can be obtained as follows:
\[ L_1 = |E|_1 = \frac{1}{V_T} \sum_i \iint_{V_i} \left[ U^k_i(x, y, z) - U_{\text{exact}}(x, y, z) \right] dV, \] \hfill (5.2)

\[ L_2 = |E|_2 = \left\{ \frac{1}{V_T} \sum_i \iint_{V_i} \left[ U^k_i(x, y, z) - U_{\text{exact}}(x, y, z) \right]^2 dV \right\}^{1/2}. \] \hfill (5.3)

where \( V_T \) is the total volume of the solution domain, \( U^k_i(x, y, z) \) is the reconstructed solution and \( U_{\text{exact}}(x, y, z) \) is the corresponding exact solution of the evaluated function at cell \( i \). The integrals above can be evaluated using the quadrature rules described in Chapter 4 of this thesis.

### 5.2 Visualization of Reconstructed Solutions

Due to the limitations of most plotting software, it is difficult to visualize high-order solution results (most plotting software is limited to linear representation of data). One possible approach to visualize the high-order solution content within a cell is to subdivide the cell into subelements and plot the solution in each subelement. This approach is adapted for the visualization of high-order reconstructed data in this thesis. In particular, a regular refinement technique is applied to the tetrahedral elements [58, 59]. To illustrate this technique, a general tetrahedron \( T \) illustrated in Figure 5.1(a) is used. This tetrahedron \( T \) is subdivided into eight sub-tetrahedra \( T_1...T_8 \) of equal volumes as shown in Figure 5.1(b). The refinement algorithm that is used to subdivide tetrahedron \( T \), whose vertices are \([x_0, x_1, x_2, x_3]\), can be simplified as:

\[ x_{ij} = \frac{1}{2}(x_i + x_j). \] \hfill (5.4)

Using the above equation, the tetrahedron can be split into the following eight sub-tetrahedra:

\[
\begin{align*}
T_1 &= [x_0, x_{01}, x_{02}, x_{03}] & T_5 &= [x_{01}, x_{02}, x_{03}, x_{13}], \\
T_2 &= [x_{01}, x_1, x_{12}, x_{13}] & T_6 &= [x_{01}, x_{02}, x_{12}, x_{13}], \\
T_3 &= [x_{02}, x_{12}, x_2, x_{23}] & T_7 &= [x_{02}, x_{03}, x_{13}, x_{23}], \\
T_4 &= [x_{03}, x_{13}, x_{23}, x_3] & T_8 &= [x_{02}, x_{12}, x_{13}, x_{23}]
\end{align*}
\] \hfill (5.5)

Where \( T_1...T_4 \) tetrahedra are created from the corner and are congruent with tetrahedron \( T \) and \( T_5...T_8 \) tetrahedra are created from the octahedral shown by the blue color in Figure 5.1(b). Once these tetrahedra components of a cell are obtained, the reconstructed solution of the cell can be plotted in each of these components.
5.3 Function Reconstruction in Three Dimensions

In order to verify the present implementation and performance of the high-order CENO reconstruction schemes for three-dimensional unstructured meshes, accuracy assessment for the reconstruction of both continuous and discontinuous functions has been performed. This is to ensure that the reconstruction order is maintained to the desired level of accuracy when a mesh convergence study is performed. The accuracy is assessed by first initializing the mesh with the function cell average values, performing the required order of reconstruction, and computing the error as the difference between the exact solution and the reconstructed solution.

As noted, both continuous and discontinuous solutions have been examined in this thesis. The first continuous function (radial cosine function) was assessed for a range of reconstruction orders from first-order to fifth-order to demonstrate the level of solution accuracy achieved. A second continuous function was also examined in which the CENO smoothness indicator was incorporated to show that the level of solution accuracy is still maintained for smooth solution content when the smoothness indicator is active.

Finally, a discontinuous function was used to assess the effectiveness of the smoothness indicator. The evaluation of a discontinuous function examines whether the limited piecewise reconstruction is correctly triggered by the smoothness indicator in the presence of discontinuities and examines the robustness of the reconstruction procedure. Note that, for all the reconstructions examined in this chapter, a smoothness indicator cut-off value of 4,000 was
used.

5.3.1 Radial Cosine Function

A continuous radial cosine function reconstruction is first considered for the assessment of the various orders of reconstruction. The function is continuous in the \( x, y \) and \( z \) directions. The radial cosine function is given as follows:

\[
    u_{\text{exact}} = 3 + \cos(2\pi r), \tag{5.6}
\]

where \( r \) is the radial position and represents the relation \( r = \sqrt{x^2 + y^2 + z^2} \). The geometric boundaries are located at \((0, 0, 0)\) and \((1, 1, 1)\). The reconstructed solution for this function is computed on a domain with a grid size that ranges from around 4,945 cells to 119,762 cells. These grids are also used for the convergence study.

Five different orders of reconstructions from the zeroth to fourth-order methods were performed for the cosine function. The reconstructions of interest are shown in Figure 5.2. The exact solution is illustrated in Figure 5.2(a), this is followed by a first-order \((k = 0)\) reconstruction in Figure 5.2(b), second-order \((k = 1)\) reconstruction in Figure 5.2(c) and finally the fourth-order \((k = 3)\) reconstruction shown in Figure 5.2(d). The illustrated reconstructions were all computed using the same mesh resolution that was made up of 10,080 cells. From the results of these figures, it is evident that the smoothness of the reconstructed solution increases considerably with the degree of reconstruction polynomial and the fourth-order result agrees most closely with the exact solution.

To determine the levels accuracy of the reconstructed solutions, error norms are calculated. The \( L_1 \) and \( L_2 \) error norms associated with the reconstruction of the radial cosine function are shown in Figure 5.3. The log-log plot shown in these figures, consist of meshes ranging from a coarse mesh to a finer mesh. By observing this log-log plot, an increase in the mesh size correlates with an increase in solution accuracy and vice versa.

Using the slopes obtained from the \( L_1 \) and \( L_2 \) log-log plots, the levels of accuracy for the reconstructed solutions are determined and tabulated in Table 5.1. From the tabulated results, the level of accuracy for the reconstructed solutions matched the reconstruction order for both \( L_1 \) and \( L_2 \) error-norms. Hence, the reconstructed solutions obtained are consistent with what is expected.

For the next case a smoothness indicator is implemented for the fourth-order CENO scheme. The scheme is then going to be applied to another continuous function reconstruction to verify
Chapter 5. Numerical Results

(a) Exact solution.  
(b) First-order \((k = 0)\) reconstruction.  
(c) Second-order \((k = 1)\) reconstruction.  
(d) Fourth-order \((k = 3)\) reconstruction.

Figure 5.2: Reconstruction of spherical cosine function.

Table 5.1: Orders of accuracy for the Spherical Cosine Function.

| \(k\) | \(|E|_1\)  | \(|E|_2\)   |
|------|----------|----------|
| 0    | -1.0105  | -1.0238  |
| 1    | -2.1221  | -2.0508  |
| 2    | -3.0156  | -3.0863  |
| 3    | -4.0286  | -4.0411  |
| 4    | -5.0050  | -5.0896  |

Table 5.1: Orders of accuracy for the Spherical Cosine Function.
Chapter 5. Numerical Results

Figure 5.3: Convergence studies for spherical cosine function.
Chapter 5. Numerical Results

5.3.2 Exponential Function

This section deals with another continuous function for determining the effectiveness of the smoothness indicator in the CENO scheme. The exponential function is used in this section. The function has the following $u_{exact}$ relation in the $x$, $y$ and $z$ coordinate system:

$$ u_{exact} = e^{xyz}, $$

The reconstruction is performed on the squared domain that has its boundaries at [-1,-1,-1] and [1,1,1] and is centered at [0,0,0]. The grid size ranged from a coarse mesh size of 4,762 cells to a finer mesh size of 92,418 cells. These meshes are once again used to determine the order of accuracy in the convergence study.

Three reconstructed solutions to this function are shown in Figure 5.4. The first Figure 5.4(a) shows the exact solution. The exact solution is followed by a first-order reconstruction in Figure 5.4(b), second-order reconstruction in Figure 5.4(c) and finally the fourth-order CENO reconstruction in Figure 5.4(d). The figures depict an unstructured mesh visualization of 10,080 cells. From the solutions obtained it is once again deduced that with an increase in the order of reconstruction the solution, a smooth transition of the contours is formed.

The $L_1$ and $L_2$ norms are once again calculated for the convergence study of the exponential function. The log-log plots are created for the error norms, and are shown in Figure 5.5. Using the slopes, the levels of accuracy are determined for the various orders of reconstruction and are tabulated in Table 5.2. The table shows the orders of accuracy obtained from the $L_1$ and $L_2$ error plots for the first-order, second-order and CENO reconstructions. From the above tabulated results, the levels of accuracy for the reconstructed solutions once again match the order of reconstructions as should be expected.

| $k$ | $|E|_1$   | $|E|_2$ |
|-----|---------|---------|
| 0   | -1.0106 | -1.0008 |
| 1   | -1.9933 | -2.0066 |
| 3   | -4.0856 | -4.1615 |

Table 5.2: Orders of accuracy for the Exponential Function.

that the actual level of numerical accuracy is once again consistent with the expected theoretical order of accuracy.
Figure 5.4: Reconstruction of an exponential function.

For the last case, reconstruction of a discontinuous function is examined to assess the effectiveness of the smoothness indicator in regions of discontinuities. Abgrall’s function [19] has been used for this case.

### 5.3.3 Abgrall’s Function

In this section, first-order and fourth-order CENO reconstructions of discontinuous function are considered. This is a challenge for high-order schemes, as the monotonicity of the solution should to be maintained in regions in the vicinity of the discontinuities. Abgrall function [19] is an appropriate candidate for this challenge due to its multiple discontinuous properties.
Chapter 5. Numerical Results

Figure 5.5: Convergence studies for exponential function.
Chapter 5. Numerical Results

(a) Exact solution on unstructured mesh. 

(b) Fourth-order CENO ($k = 3$) reconstruction.

(c) Smoothness indicator.

Figure 5.6: Reconstruction of Abgrall's function.

The objective is to illustrate the ability of the CENO scheme to capture these discontinuities accurately. This is an important test for the smoothness indicator of the CENO scheme as it
Table 5.3: Orders of accuracy for the Abgrall’s Function.

| Slopes | $|E|_1$  | $|E|_2$  |
|--------|---------|---------|
| 0      | -0.9555 | -0.6407 |
| 3      | -1.2326 | -0.6610 |

is supposed to be triggered regions of discontinuities. Abgrall function is defined as follows:

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

with

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

where

$$r = \sqrt{x^2 + y^2}.$$  

The exact solution and CENO reconstruction are shown in Figure 5.6. The discontinuous areas triggered by the smoothness indicator are shown in Figure 5.6(c). The illustrated figures depicts a mesh of 210,080 cells. From the two sub-figures, the fourth-order reconstruction solution shows similar contours as compared to the exact solution.

The $L_1$ and $L_2$ error norms are shown in Figure 5.7. The results of the $L_1$ and $L_2$ error-norms are obtained and tabulated in Table 5.3. The two reconstruction methods, first-order ($k = 0$) and fourth-order ($k = 3$) reconstruction, both obtain the same first-order level of solution accuracy. This was expected due to the discontinuities in Abgrall’s function. However, for the CENO scheme the error norm is shifted downwards. This implies that although the fourth-order slope reverts to first-order accurate in the presence of discontinuities, the method provides a more accurate reconstructed solution for the same mesh size. This reduction in error is advantageous and indicates that few cells are required for the CENO scheme to achieve the same error as compared to the standard first-order scheme.

Another important thing to consider in this section is the triggering of the smoothness indicator. As state previously, when the solution is determined to be discontinuous the smoothness indicator is triggered and a limited linear piecewise reconstruction procedure is carried out in order to preserve solution monotonicity. From Figure 5.6(c), the smoothness indicator behaves very effectively in discontinuous regions of the Abgrall function.
Figure 5.7: Convergence studies for Abgrall’s function.
5.4 Summary of Results

From the various reconstructions performed, it has been shown that the high-order CENO scheme for three-dimensional unstructured mesh has achieved the expected level of solution accuracy for both the continuous and discontinuous functions. In addition, the high-order CENO scheme has been shown to provide levels of solution accuracy that exceed the standard first- and second- order methods. As such, the results provide a useful first step in the verification of the high-order method for unstructured meshes.
Chapter 6

Conclusions and Future Research

As stated in the introduction, the objective of thesis was to build on the previous CENO high-order methods developed by Ivan and Groth [1, 12, 13, 14] and extend the reconstruction method to three-dimensional unstructured meshes. As such, the thesis represents an extension of the previous research conducted by Rashad [26] that focused on implementing the CENO scheme for three-dimensional structured mesh and by McDonald [28] which extended the CENO scheme to two-dimensional unstructured mesh.

Section 6.1 provides the conclusions of the results obtained from implementing and verifying the CENO reconstruction scheme for three-dimensional unstructured meshes and Section 6.2 provides a possible outline for the further development and implementation of the CENO high-order scheme.

6.1 Conclusions

The primary objective of this thesis was to develop and implement the CENO reconstruction scheme for three-dimensional unstructured meshes. The high-order CENO approach would seem particularly advantageous for unstructured mesh. The CENO scheme was successfully implemented in the in-house code at UTIAS. The implementation was then verified through the application to various function reconstructions. The verification process made use of three functions, of which two were continuous and one was discontinuous. Results obtained from the verification process showed that the CENO scheme successfully achieved its required fourth-order level of accuracy for the reconstruction of the continuous functions and first-order level of accuracy for the reconstruction of the discontinuous function. In addition, the smoothness indi-
cator within the CENO scheme was triggered and worked effectively to limit solution oscillations as expected in under-resolved and discontinuous regions.

The results obtained from the verification process assure that the high-order CENO reconstruction scheme for three-dimensional unstructured meshes was successfully implemented and obtained the required level of solution accuracy which was the primary objective outlined in Chapter 1 of this thesis.

\section*{6.2 Future Research}

This thesis provides a foundation for the development and implementation of the CENO high-order scheme for three-dimensional unstructured meshes. Clearly, there is still more research required for the development and implementation of the CENO scheme and a few of these recommendations are listed as follows:

- the CENO scheme could be implemented in the in-house code for solutions of the Euler equations on three-dimensional unstructured meshes;
- the CENO scheme could then be implemented to obtain solutions to the Navier-Stokes equations; and
- finally, the CENO scheme could be extended to hybrid three-dimensional meshes consisting of both structured and unstructured grid blocks \cite{60, 61}.

These are just a few suggested applications and there is still much more research required for the development and implementation of the CENO scheme. Therefore, the further development of the CENO scheme does show the potential to achieve solutions to practical flow problems in a more effective and efficient way.
References


