Very High-Order Methods for 3D Arbitrary Unstructured Grids

Panagiotis Tsoutsanis

Submitted for the degree of PhD

Department of Aerospace Sciences

Cranfield University

Cranfield, UK

2009

Cranfield University

School of Engineering

PhD

2009

Panagiotis Tsoutsanis

Very High-Order Methods for 3D Arbitrary Unstructured Grids

Supervisors: Dimitris Drikakis, Vladimir Titarev

August 28, 2009

Abstract

Understanding the motion of fluids is crucial for the development and analysis of new designs and processes in science and engineering. Unstructured meshes are used in this context since they allow the analysis of the behaviour of complicated geometries and configurations that characterise the designs of engineering structures today. The existing numerical methods developed for unstructured meshes suffer from poor computational efficiency, and their applicability is not universal for any type of unstructured meshes. High-resolution high-order accurate numerical methods are required for obtaining a reasonable guarantee of physically meaningful results and to be able to accurately resolve complicated flow phenomena that occur in a number of processes, such as resolving turbulent flows, for direct numerical simulation of Navier-Stokes equations, acoustics etc.

The aim of this research project is to establish and implement universal, high-resolution, very high-order, non-oscillatory finite-volume methods for 3D unstructured meshes. A new class of linear and WENO schemes of very high-order of accuracy (5^{th}) has been developed. The key element of this approach is a high-order reconstruction process that can be applied to any type of meshes. The linear schemes which are suited for problems with smooth solutions, employ a single reconstruction polynomial obtained from a close spatial proximity. In the WENO schemes the reconstruction polynomials, arising from different topological regions, are non-linearly combined to provide high-order of accuracy and shock capturing features.

The performance of the developed schemes in terms of accuracy, non-oscillatory behaviour and flexibility to handle any type of 3D unstructured meshes has been assessed in a series of test problems. The linear and WENO schemes presented achieve very high-order of accuracy (5^{th}). This is the first class of WENO schemes in the finite volume context that possess high-order of accuracy and robust non-oscillatory behaviour for any type of unstructured meshes.

The schemes have been employed in a newly developed 3D unstructured solver (UCNS3D). UCNS3D utilises unstructured grids consisted of tetrahedrals, pyramids, prisms and hexahedral elements and has been parallelised using the MPI framework. The high parallel efficiency achieved enables the large scale computations required for the analysis of new designs and processes in science and engineering.

Acknowledgments

I would like to take this opportunity to thank and express my sincere gratitude to my supervisor Professor Dimitris Drikakis for giving me the opportunity to do a research degree in a project where I feel very lucky to be a part of, and for his expert guidance and support.

I am deeply indebted to Dr Vladimir Titarev for his constant support and constructive suggestions that have greatly enriched my knowledge with this exceptional insights in high-order schemes for unstructured grids. His guidance has helped me construct a robust theoretical background and he has given much time and devotion to this project.

To my parents Dimitrios and Aggeliki who instill in me
the drive and determination to fulfill my dreams
and pursue my goals.

Abbreviations

ADER	Arbitrary Order Derivative Riemann Problem
API	Application Programming Interface
CFD	Computational Fluid Dynamics
CFL	Courant, Friedrichs and Lewy Condition Number
DG	Discontinuous Galerkin
FD	Finite Difference
FEM	Finite Element
FVM	Finite Volume
MPI	Message Passing Interface
TVD	Total Variation Deminishing
WENO	Weighted Essentially Non-Oscillatory

Contents

List of Algorithms

List of Tables

List of Figures

Chapter 1

Introduction

Understanding the motion of fluids is crucial for the development and analysis of new designs and processes in science and engineering. Computational fluid dynamics (CFD) is the branch of fluid dynamics concerned with the numerical solution of flow problems. Due to the leap in computing processing power in the last years CFD is employed for a staggering amount of applications. The numerical solution of the Navier-Stokes equations -that model the fluid flow- is performed by algebraic approximations that give numerical solution at discrete points in the flow field. The collection of those discrete points is the grid. Depending on the pattern that the grids follow there are distinguished in two types; structured and unstructured.

The grid is the Achilles heel of accurate and efficient numerical simulation of flow problems. There is a sensitive balance amongst accuracy of solution, efficiency of simulation and accuracy of geometrical representation. Numerical methods have been introduced in the structured grid context, which has resulted in robust schemes. A shift of the CFD community towards unstructured grids is noticed over the last years for large scale computations [55, 54] of complicated structures. The motivation for this transition lies in their efficiency for representing complicated geometrical structures. The existing unstructured grid numerical methods suffer from low accuracy compared to structured grids and their requirements in terms of computing power and resources are higher. The majority of the existing finite volume unstructured grid methods can not simultaneously achieve higher than 2^{nd}-order of accuracy, demonstrate non-oscillatory properties and have the capability to handle any type of unstructured mesh.

The development of very high-order FV schemes for unstructured meshes is associated with numerous challenges. The design of the reconstruction process that can achieve high-order approximations in arbitrary shaped domains will drive the spatial accuracy of the scheme. The generation of spurious oscillations leading to unphysical results [25] poses an additional challenge in the design of the numerical scheme. The generic nature of the methods in the sense that should be able to handle any type of unstructured is another challenging task. The efficiency of 3D unstructured solvers is an issue that still needs to be addressed. Several

approaches has been developed to circumvent the aforementioned limitations and can be categorised to the discontinuous Galerkin finite element framework, the spectral finite-volume, spectral finite-difference framework and the finite volume framework. The basic characteristic of the majority of the approaches in the DG framework [12, 14, 13, 11, 15, 28, 37, 76] is the combination of the finite element methods for local data representation and the Riemann problem solution at the discontinuous intercell states. The advantages of the DG approaches are their compactness and locality which is translated to very high-order of accuracy, and their ability to handle any type of unstructured meshes. These schemes suffer from the generation os spurious oscillations in the vicinity of discontinuities unless an appropriate limiter is employed. Although some recent approaches such as the HERMITE-WENO schemes [48, 59, 61] try to overcome this problem, these schemes are still unable to combine higher than 3^{rd}-order of accuracy and non-oscillatory properties in an efficient and robust manner.

In the spectral finite-volume framework [73, 65, 45] each spectral volume is further subdivided into control volumes. The data are cell-averaged over the control volumes to reconstruct high-order approximation in the spectral volume. The fluxes at the spectral volume boundaries are calculated by employing Riemann solvers across them. Spurious oscillation are reduced by TVD limiters around discontinuities. This class of methods along with the spectral-difference ones have been applied in a series of complicated problems [29, 66, 74, 65]. The main advantages of this framework are their compactness and locality, and their superior efficiency especially for the spectral-difference methods. However the issues that remain unresolved are the subdivision of arbitrary shaped spectral volumes into control volumes and the employment of more sophisticated limiters in order to attain very high-order of accuracy and non-oscillatory behaviour.

The finite volume framework is principally the framework of choice for commercial and industrial applications since a large variety of robust schemes exist that have been introduced in the structured grid context and have been tested for a various applications. However the finite volume framework is still very immature compared to the finite element framework in terms of high-order schemes for unstructured meshes. The first class of high-resolution methods developed for unstructured grids included the ENO type of schemes [1, 57] and later the WENO [23, 32] type of schemes. In the WENO type of schemes the high-order of accuracy was achieved by non-linearly combining a series of high-order reconstruction polynomials arising from a series of reconstruction stencils. However most of the approaches are limited to 3D applications of tetrahedral meshes [26, 56, 32] only and up to 3rd-order of accuracy with a few notable exceptions [49, 22]. These limitations motivates the research regarding the stencil selection influence, the combination of reconstruction polynomials and the applicability to generic unstructured meshes.

The parallel efficiency of 3D unstructured solvers has been addressed in a series of approaches [9, 50] , and have been applied to a series of large-scale computations of complicated flow problems [55, 54, 53, 51, 2] but no in the context of very-high order finite volume schemes.

1.1 Aim and Objectives

The aim of this research project is to establish and implement universal, high-resolution, very high-order, non-oscillatory finite-volume methods for 3D unstructured meshes. The main objectives are:

- Research and develop very-high-order methods such as WENO and TVD schemes in the unstructured grid context
- Develop a new 3D unstructured flow solver employing the developed schemes (UCNS3D)
- Verify the accuracy of the developed methods towards a series of well established test cases
- Verify the non-oscillatory properties of the schemes through a series of test cases that contain strong discontinuities
- Assess the performance of the schemes in terms of robustness and reliability for resolving complicated flow patterns around complicated geometries and configurations
- Study the grid-dependency of the schemes
- Identify the parameters that dictate the performance of the methods for test cases containing smooth solutions and test cases containing discontinuous solutions
- Carry out a parallel implementation of UCNS3D suited for large-scale computations
- Assess the parallel efficiency and scalability of the parallel UCNS3D solver

1.2 Thesis Contributions

The work conducted in this thesis has made significant contributions to the field of high-order methods for unstructured grids. In particular

- A very high-order (5^{th}-order) set of linear and WENO schemes has been developed. The key aspect of the developed schemes is a high-order reconstruction process that is applicable to any type of unstructured meshes.
- The performance of the schemes in terms of accuracy and non-oscillatory properties has been assessed in a series of smooth and discontinuous test cases. The results demonstrate that the schemes reach their designed accuracy and do not produce any spurious oscillations even for strong discontinuous test problems.
- The schemes are employed in a a newly developed 3D unstructured flow solver. A parallel implementation of the solver has been carried out and results demonstrate high parallel efficiency and scalability.
- This is the first class of successfull WENO schemes that achieve higher than 2^{nd}-order of accuracy on any type of unstructured grids in the finite volume context. The developed schemes can be extended to any desired order of accuracy.
- Results from this thesis are used for the preparation of two journal articles concerning the 3D WENO schemes for hybrid meshes, and the parallelisation of high-order hybrid unstructured solver.
- Results from this thesis have been presented in a number of international conferences including the 8th World Congress on Computational Mechanics, 30 June-4 July 2008, Venice, Italy and the 3rd International Conference on High-Order Non-Oscillatory Methods for Wave Propagation, Transport and Flow Problems, 30 March- 2 April 2009, Trento, Italy.

1.3 Thesis Overview

This thesis is organised as follows.

Chapter 2 presents a detailed description of the geometrical operations and the reconstruction process. The geometrical operations involve all the aspects for dealing with any type of unstructured meshes. Those are the element decomposition for volume and surface integrals computations, the stencil construction algorithms and the geometrical parameters that govern them, and the utilisation of spherical coordinate system for computation of the normal vector at the surfaces. The reconstruction process is the basic ingredient of the developed schemes since it the one that provides the high-order of spatial accuracy. Three implementations of the reconstruction process are presented the linear type, a TVD type and a WENO type. The linear type utilises only one reconstruction stencil, the central one in order to achieve high-order of spatial accuracy and is mostly suited for problems containing smooth solutions. The TVD implementation is a 2^{nd}-order accurate scheme which is characterised by its robustness and computational efficiency and is mostly suited for problems dominated by discontinuities. Finally the WENO implementation is described which combines reconstruction polynomials from different reconstruction stencils in a non-linear manner. Both the linear and WENO scheme in their present implementation can be extended to any desired order of accuracy.

Chapter 3 presents the extension of the developed schemes to the 3D compressible Euler equations. This involves the implementation of the reconstruction process for the Euler equations, the numerical flux approximation, the time advancement techniques and the implementation of the boundary conditions. The developed WENO scheme performs the reconstruction in terms of characteristic variables rather than the conserved ones. The numerical flux approximation is done by the HLLC approximate Riemann solver of [70]. The solution is advanced in time by explicit Runge-Kutta schemes up to 4th-order. Finally the boundary conditions implementation in the context of unstructured meshes is described since the are some challenges for high-order finite volume schemes when certain types of boundary conditions are encountered.

Chapter 4 presents the assessment of the developed schemes in terms of accuracy, non-oscillatory properties and capability for handling any type of boundary conditions and unstructured meshes. Firstly the schemes are applied to the 3D linear advection equation for a smooth test problem and a discontinuous one. Next the 3D Euler equations are solved for a series of test problems in order to assess specific aspects of the developed schemes. The order of accuracy of the schemes are analysed by computing the smooth test problem of the evolution of a vortex. The non-oscillatory properties, the influence of the central stencil linear weights, and the influence of the geometrical directionality condition of the directional stencils used in WENO schemes are assessed in a shock tube test problem, a 3D spherical explosion and a 3D spherical explosion. Finally the capability of the schemes to handle complicated geometries using hybrid unstructured meshes is assessed in the test problem of a blunted-cone-cylinder-flare geometry. It is noticed that both the linear and WENO schemes achieve up to 5th-order of accuracy and WENO schemes robust shock capturing features. The most efficient schemes in terms of computing resources and accuracy for practical applications are the 2^{nd}-order TVD and the 3^{rd}-order WENO scheme.

Chapter 5 presents the parallel implementation of the developed 3D unstructured flow solver (UCNS3D). The mesh decomposition strategy and the load balancing achieved for various types of unstructured meshes are detailed. The most crucial factors that have been taken into account when designing a series of parallel algorithms are described. Finally the parallel performance of the the 3D unstructured flow solver is assessed in a fixed problem computed in various processors at the Astral-HPC facility at Cranfield University. The parallel efficiency achieved is almost linear for higher-order schemes and it is also noticed that higher-order schemes scale much better than lower-order.

Chapter 6 presents the conclusions drawn from this research project and possible future research directions in the context of very high-order finite volume schemes for 3D unstructured meshes.

Chapter 2

General Framework of Developed Schemes

Introduction

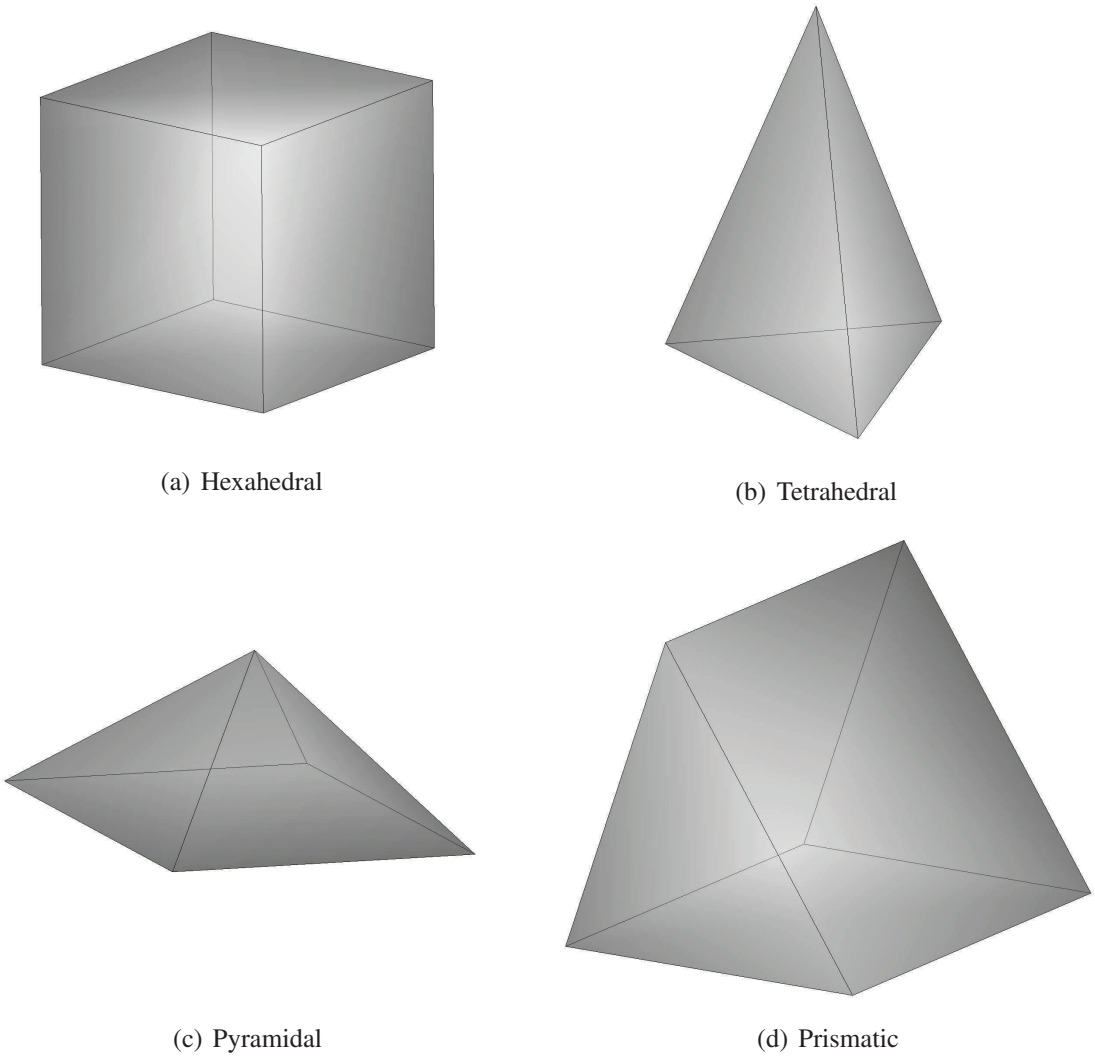
In this chapter we describe the general framework of the developed schemes tailored for mixed-element unstructured meshes in three-space dimensions. The chapter is structured as follows. We first present the geometrical operations performed in the context of unstructured hybrid meshes. Next we outline the linear reconstruction procedure for mixed-element unstructured meshes in three-space dimensions for scalar which will then be used for the construction of TVD and WENO schemes that are described in the following sections.

2.1 Geometrical Operations

Since the developed approach is flexible to handle mixed-element unstructured grids this results in greater requirements in terms of geometrical related operations compared to structured grids. In the following subsections the strategy for performing a series geometrical computations for different element shapes is described in great detail.

2.1.1 Element Shapes

The schemes are tailored for unstructured meshes that can consist of various element shapes. Those are hexahedral, tetrahedral, pyramidal and prismatic elements as shown in Figure 2.1 on page 19. When constructing numerical schemes the distinctive geometrical characteristics of each element must be taken into account. These include the number of nodes, the number faces, the number of tetrahedral volumes that each element can be decomposed and the number of triangular faces that each element face can be decomposed to. We are only considering strictly conforming combinations of these element shapes. Therefore the direct-side neighbour of an element although it can be of different shape it must have the same face.This is also one of the reasons that pyramids and prisms are used in order to facilitate the transition



(a) Hexahedral

(b) Tetrahedral

(c) Pyramidal

(d) Prismatic

Figure 2.1: Element Shapes

between hexahedrals and tetrahedrals elements and ensure conformity. The basic geometrical characteristics of each element shape are outlined in Table 2.1 on page 20.

2.1.2 Facecentres & Barycentres Computations

For a series of operations of the finite volume schemes developed the position of the centre of each surface and the barycenter of each element is required. The facecentre coordinates of each surface area is given by the following expression

Geometrical Characteristic	Hexahedral	Tetrahedral	Pyramidal	Prismatic
Number of Faces	6	4	5	5
Number of Nodes	8	4	5	6
Number of Triangular Faces	0	4	4	2
Number of Quadrilateral Faces	6	0	1	3
Tetrahedral Decompositions	6	1	2	3
Triangular Decompositions	12	4	6	8

Table 2.1: Geometrical Characteristics of Elements

$$X_c = \frac{1}{ns}\sum_{w=1}^{ns} X_w \tag{2.1}$$

$$Y_c = \frac{1}{ns}\sum_{w=1}^{ns} Y_w \tag{2.2}$$

$$Z_c = \frac{1}{ns}\sum_{w=1}^{ns} Z_w \tag{2.3}$$

Where X_c, Y_c, Z_c is the position of the facecentre in x, y and z Cartesian coordinates respectively, n_s is the number of nodes that this surface area has which takes the value 3 for triangular faces and 4 for quadrilateral faces. We always require that the facecentres and barycentres lie within the volume and the face respectively. Therefore highly stretched elements that can result in the barycentre outside the volume are avoided in the mesh generation, and the faces are always planar. Having the barycentre lying outside the volume would results in wrong geometrical computations that would greatly impact the accuracy of the scheme.

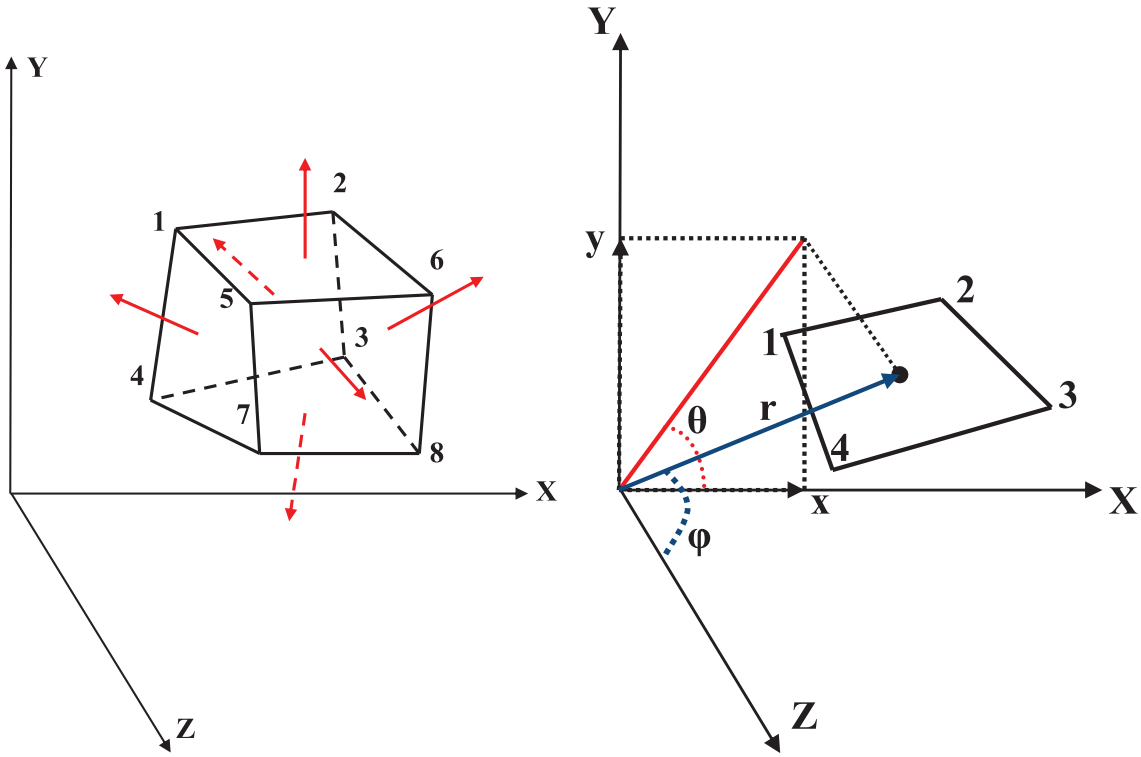
The computation of the barycentre of each element depends on the shape of the element. For example for tetrahedrals and hexahedrals no decomposition is required, but for pyramidal, prismatic elements a decomposition is required. The reason is that by decomposing the element into other shapes it is more convenient to compute the barycentre. The barycentre position is given by

$$X_{bc} = \frac{1}{nd}\sum_{d=1}^{nd}\frac{1}{nv}\sum_{w=1}^{nv} X_w \tag{2.4}$$

$$Y_{bc} = \frac{1}{nd}\sum_{d=1}^{nd}\frac{1}{nv}\sum_{w=1}^{nv} Y_w \tag{2.5}$$

$$Z_{bc} = \frac{1}{nd}\sum_{d=1}^{nd}\frac{1}{nv}\sum_{w=1}^{nv} Z_w \tag{2.6}$$

Where X_{bc}, Y_{bc}, Z_{bc}, is the position of the barycentre in x, y and z Cartesian coordinates respectively, nd is the number of decompositions that have been made, and nv is the number of nodes that each decomposed element has.



(a) Surface Normal Vectors (b) Spherical Coordinates

Figure 2.2: Surface Orientation

2.1.3 Surface Orientation Operations

The orientation of each surface with respect to the origin of the Cartesian axis is required in order to establish the direction of the vectors normal to the surfaces. Since the normal vector of each surface should point outwards the origin is taken to be barycentre of each control volume. Consider the faces of a uniform hexahedral cell in Cartesian axis as shown in Figure 2.2 on page 21 , for each surface the equation of plane is solved by using the Cartesian coordinates of three vertices of the surface. The equation of plane is solved as shown below

$$Ax + By + Cz + D = 0 \tag{2.7}$$

$$A = y1(z2 - z3) + y2(z3 - z1) + y3(z1 - z2) \tag{2.8}$$

$$B = z1(x2 - x3) + z2(x3 - x1) + z3(x1 - x2) \tag{2.9}$$

$$C = x1(yz2 - y3) + x2(y3 - y1) + x3(y1 - y2) \tag{2.10}$$

$$D = -x1(y2z3 - y3z2) + x2(y3z1 - y1z3) + x3(y1z2 - y2z1) \tag{2.11}$$

$$xn = x_{crt} - x_{bc},\ yn = y_{crt} - y_{bc}\ , zn = z_{crt} - z_{bc} \tag{2.12}$$

$$A^n = A/D,\ B^n = B/D,\ C^n = C/D \tag{2.13}$$

Where xn, yn, zn with $n = 1,2,3$ are the coordinates of each vertex n with respect to the position of the barycentre (x_{bc}, y_{bc}, z_{bc}). A ,B, C and D and are the coefficients for the equation of plane and A^n , B^n , C^n are the normalised coefficients for the equation of plane. Now by using the spherical coordinate convention the radial r, the azimuthal angle θ and polar angle φ are given by the following expressions

$$r = \sqrt{(A^n)^2 + (B^n)^2 + (C^n)^2} \tag{2.14}$$

$$\theta = tan^{-1}\left(\frac{B^n}{A^n}\right) \tag{2.15}$$

$$\varphi = cos^{-1}\left(\frac{C^n}{R}\right) \tag{2.16}$$

It must be noted that the appropriate modifications must be made in Equation 2.14 to 2.16, depending on which quadrant the plane lies in. It is always ensured that the elements are such that the barycentre is always inside the volume, in the case that it was not the computation of the normal vector would fail leading to the normal facing inwards to the element rather outwards which is undesirable.

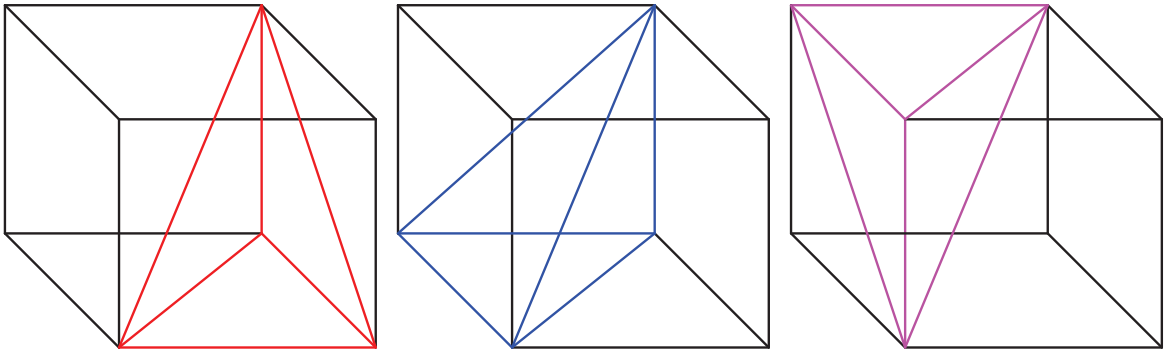
Another limitation is that in the case of non co-planar face surface orientation computation will not be correct since not all vertices will lie in the same plane. Hence the usage of three vertices for the computation of the normal vector (given that all three vertices lie in the same plane) however this will not be correct since the face will not be unique. Therefore we always ensure that the faces are planar in order to avoid this problem.

2.1.4 Element Decomposition Strategy

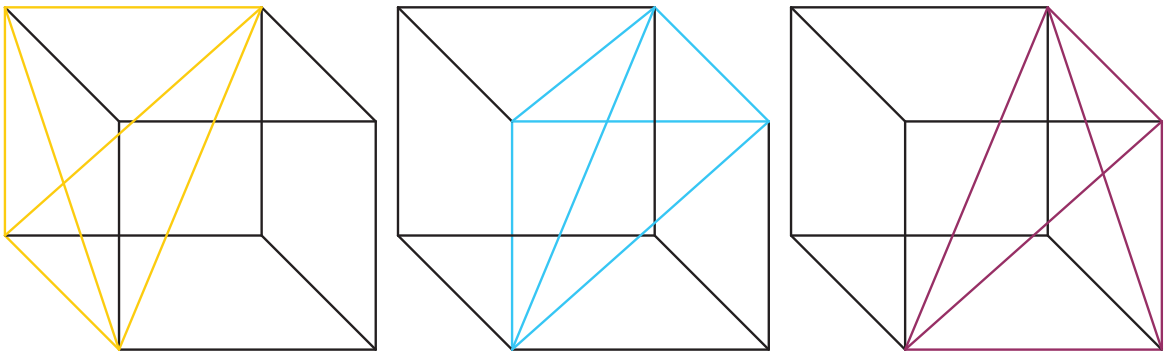
As mentioned earlier the computations involving the surface area and the volume of the element are done in an efficient, accurate and convenient way by decomposition. By decomposition we mean that the element is decomposed into a number of elements. For instance consider a hexahedral cell which can be decomposed into 5 or 6 tetrahedral cells. When dealing with complicated arbitrary shaped domains this decomposition is essential for the following reasons:

1. Unified framework for determining the Gaussian quadrature points, barycentres of an element

2. Polynomial basis functions that are independent of the element



(a) First Decomposition (b) Second Decomposition (c) Third Decomposition

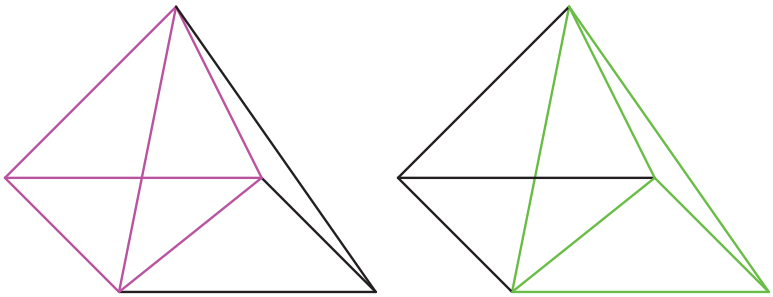


(d) Fourth Decomposition (e) Fifth Decomposition (f) Sixth Decomposition

Figure 2.3: Hexahedral Decomposition

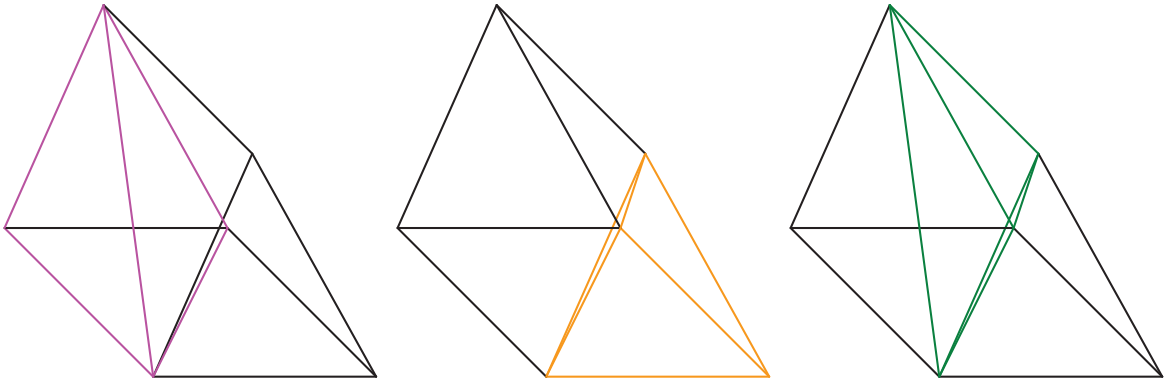
3. Transformation from physical to computational plane wrt one of the decomposed elements

Therefore a strategy is required on how the elements and surface areas can be decomposed. There are two techniques that are available, the first one being the decomposition of each element into elements of lower order, and the decomposition into elements of higher order. By order in this case we mean the node count of the element, therefore when decomposing a hexahedral cell described by 8 nodes, into an arbitrary number of tetrahedral cells described by 4 nodes this is a lower order decomposition. On the other hand a tetrahedral cell can be decomposed into 4 hexahedral cells which is a higher order decomposition. Generally the lower order decomposition is preferred since it is the most efficient in terms of computing resources. A typical example of lower order decompositions for the basic shapes of hexahedral, prisms, and pyramids is shown in Figure 2.3 on page 23 to Figure 2.5 on page 24. The decomposed tetrahedral elements are defined by colour lines in each case.



(a) First Decomposition (b) Second Decomposition

Figure 2.4: Pyramid Decomposition



(a) First Decomposition (b) Second Decomposition (c) Third Decomposition

Figure 2.5: Prism Decomposition

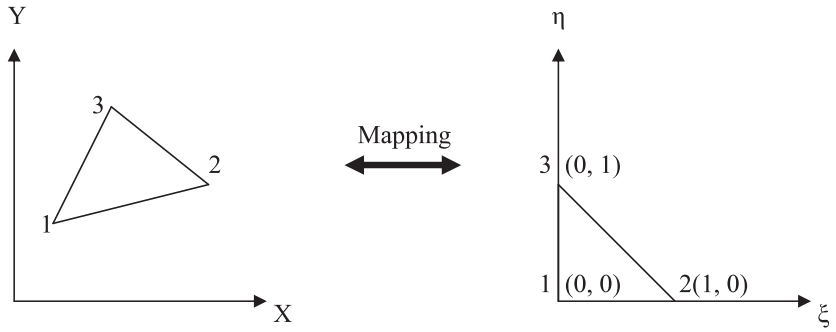


Figure 2.6: Triangle Mapping

The surface area and volume for any type of element are given by the following expressions

$$S_s = \sum_{nf=1}^{nd} S_T \tag{2.17}$$

$$V_{ks} = \sum_{nv=1}^{nd} V_T \tag{2.18}$$

Where s stands for the shape of the surface ranging from 1 to n depending on the element shape, nf is the index of the decomposed triangular area, nd the number of decompositions. S_T is the area of the decomposed triangular surface, ks stands for the shape of the element ranging from 1 to 5 depending on the element shape, nv is the index of the decomposed tetrahedrals , and V_T the volume of the decomposed tetrahedral element.

2.1.5 Gaussian Quadrature Points

The volume and surface integrals that appear in the finite volume schemes are approximated by a Gaussian quadrature technique of appropriate order. The reason for that is since high-order of spatial accuracy is required the approximation of these integrals should also be of high-order of accuracy. The Gaussian quadrature rules used are defined on the unit triangle for surface integrals and on unit tetrahedral for volume integrals. Therefore the Cartesian coordinates of a triangle are mapped onto the coordinates of the unit triangle in order to establish the position of the Gaussian quadrature points, the same procedure is followed for the tetrahedral as well. The Gaussian quadrature rules used can either be based on the Legendre , Jacobi, Hermite, Chebyshev, Leguerre polynomials or any shifted formulation of them. For the shifted Legendre polynomials in the interval $[0,1]$ the mapping from physical Cartesian coordinates x, y and z to the computational coordinates ξ, η and ζ is illustrated in Figure 2.6 on page 25 and Figure 2.7 on page 26.

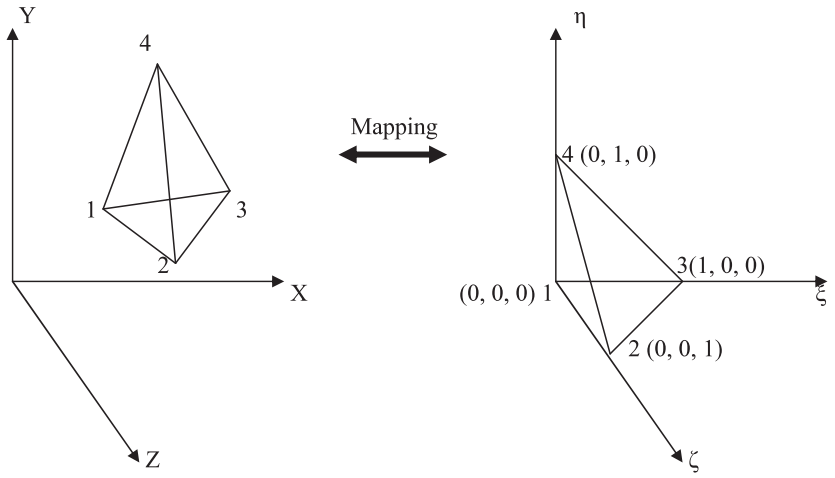


Figure 2.7: Tetrahedral Mapping

The Gaussian quadrature points are defined in the computational space with coordinates ξ, η and ζ . The position of the quadrature points in the physical space with coordinates x, y and z is given by the following equations.

$$q_x = x_1 + (x_2 - x_1)\,\xi_q + (x_3 - x_1)\,\eta_q + (x_4 - x_1)\,\zeta_q \tag{2.19}$$

$$q_y = y_1 + (y_2 - y_1)\,\xi_q + (y_3 - y_1)\,\eta_q + (y_4 - y_1)\,\zeta_q \tag{2.20}$$

$$q_z = z_1 + (z_2 - z_1)\,\xi_q + (z_3 - z_1)\,\eta_q + (z_4 - z_1)\,\zeta_q \tag{2.21}$$

Where q_x, q_y and q_z are the coordinates of the quadrature points in physical space, ξ_q, η_q and ζ_q are the coordinates of the Gaussian quadrature points in computational space as given by the appropriate rule, and x_n , y_n and z_n the coordinates in physical space of the nodes. The origin of the axis in computational space is vertex 1 , therefore the coordinates of all the vertices are normalised wrt the coordinates of vertex 1. Equations 2.19 to 2.21 provide the mapping for tetrahedral elements. The number of quadrature points and weights for triangles and tetrahedrals and the corresponding total number of surface and volume points for the elements based on the decomposition for various orders of accuracy are outlined in Table 2.2 on page 27 and Table 2.3 on page 27. The weights and coordinates for the Gaussian quadrature rules in the shifted Legendre interval of $[0,1]$can be found in [40].

Order	Tetrahedral	Hexahedral	Pyramidal	Prismatic
2	12	36	18	24
3	16	48	24	32
4	24	72	36	48
5	28	84	42	56
6	48	144	72	96
7	52	156	78	104
8	64	192	96	128

Table 2.2: Element comparison in terms of surface Gaussian quadrature points

Order	Tetrahedral	Hexahedral	Pyramidal	Prismatic
2	5	30	10	15
3	10	60	20	30
4	11	66	22	33
5	15	90	30	45
6	24	144	48	72
7	31	186	64	93
8	45	270	90	135

Table 2.3: Element comparison in terms of volume Gaussian quadrature points

2.1.6 Stencil Selection Approach

Since we are dealing with high-order methods we need to know the variation of information (data) in the close spatial proximity (neighbourhood) of each element in the mesh. Therefore we would be able to obtain high quality approximations of how this information varies within each element. Hence it is required to be able to construct a region (neighbourhood) of elements surrounding each element in the mesh. This region is named the stencil and is constructed by recursively adding the direct side neighbours of any considered element until a number of elements has been reached. The basic steps of the stencil construction procedure are outlined in Algorithm 1. Typical examples of central stencils in three-dimensions can be seen in Figure 2.16 on page 36.

There is also another category of stencils named the directional stencils or WENO stencils where there is an additional conditions that the elements must satisfy apart from the repetition condition as seen in Algorithm 1. The additional condition that must be satisfied is that the candidate element must lie within a specified geometrical sector. For analysing the geometrical sector condition consider a quadrilateral cell i that we want to construct a directional stencil with respect to this cell. Most of the approaches employing the sectorial stencils [22, 59, 61, 78, 75, 57] utilise sectors that either arise from the planes between the edge centre the edge node and the barycentre of the elements, or from the edges that make a side and the sectors outside from them as seen in Figure 2.8 on page 29. This technique results in 8 sectors for the quadrilateral cell in two-dimensions. We have adopted another technique for defining

Algorithm 1 Central Stencil Selection Algorithm

1. For cell i we want to construct a set of elements S (stencil) consisting of $N(S)$elements
2. With $c = 1, 2, ...N$ being the index of the numbering of the elements in the stencil
3. $S_1 = i$, the considered cell i is always the first element in the stencil $c = 1$
4. Assume that the element in the stencil S_c has M number of direct-side neighbours
   (a) Check which of the M elements do not belong in the set (Repetition Condition)
5. Assign as the next elements in the stencil only the ones that do not belong in the set
6. Repeat steps 4 to 8 until N number of elements have been assigned

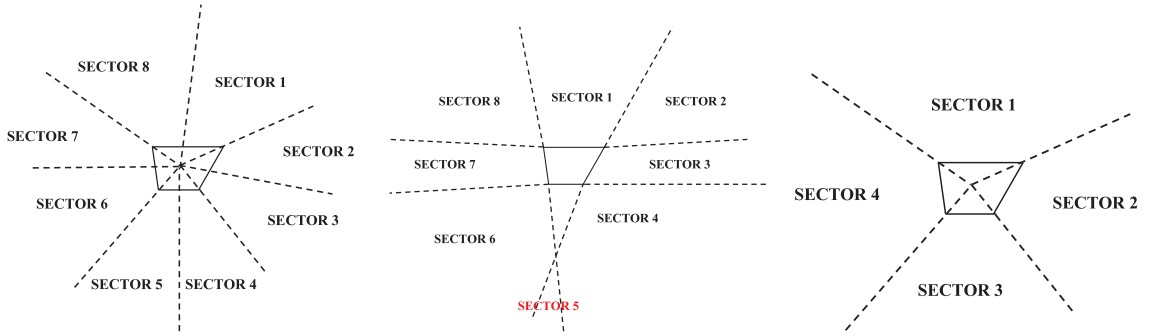
the directional sectors which is at least two times and three times more efficient in terms of resources required for storage of additional matrices for the other stencils, in two-dimensions and three-dimensions respectively. We define the sectors by the plane of the two nodes that define an edge and the barycentre of an element in two dimensions as seen in Figure 2.8 on page 29. For three dimensions this technique extends in the same manner by having three edges for each triangular face and four edges for each quadrilateral face as it is illustrated in Figure 2.9 on page 29 respectively. A directional stencil is admissible when it has at least the required number of elements satisfying the geometrical condition. For the triangular face the edges $C12$, $C13$ and $C23$ and for the quadrilateral face the edges $C12$, $C23$, $C34$, $C41$ make the sector where C is the barycentre of the element that this face belongs to. Having defined the sectors of each face the next step involves to set the condition that must be satisfied.

The geometrical condition that each candidate element for a directional stencil must satisfy leads to the introduction of a new geometrical parameter D_c. D_c is the parameter defined as the percentage of the vertices that must lie within the planes that define the sector. For uniform unstructured and structured meshes D_c could be equal to 1 (100%) where for highly stretched meshed this value must be relaxed. The effect of this parameter in terms of the chosen elements for the stencil can be seen in Figure 2.10 on page 30 where the elements that satisfy the condition set by D_c are depicted in red. The basic steps of the directional stencil construction procedure are outlined in Algorithm 2.

$$D_c = \frac{number\ of\ nodes\ that\ lie\ within\ the\ sector}{total\ number\ of\ nodes\ of\ the\ element} \tag{2.22}$$

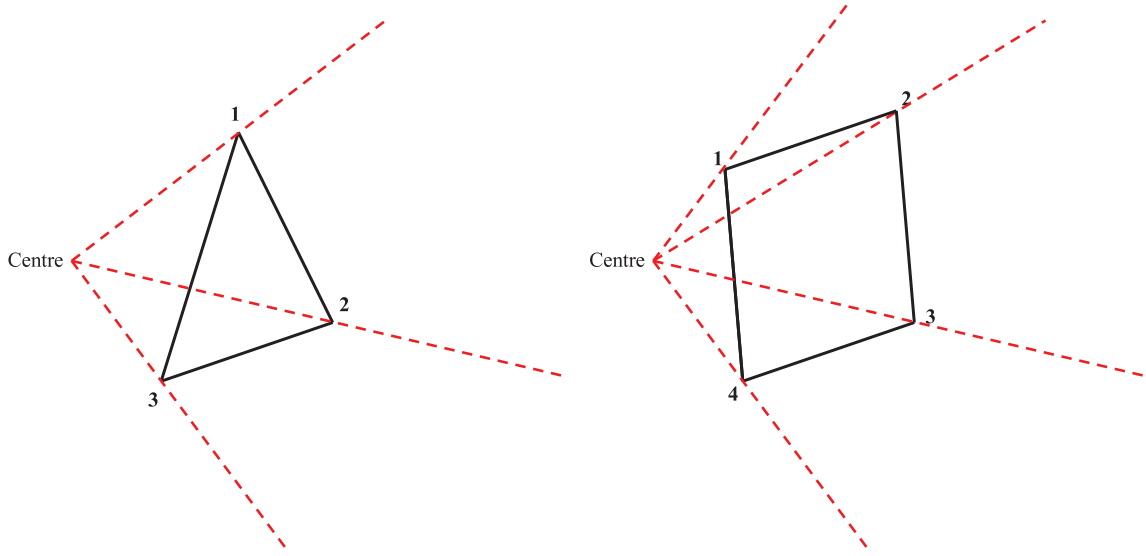
2.2 Reconstruction

Upwind finite volume methods evolve in time spatial cell averages of the solution. The use of these averages in the calculation of the numerical flux leads to the first-order methods.



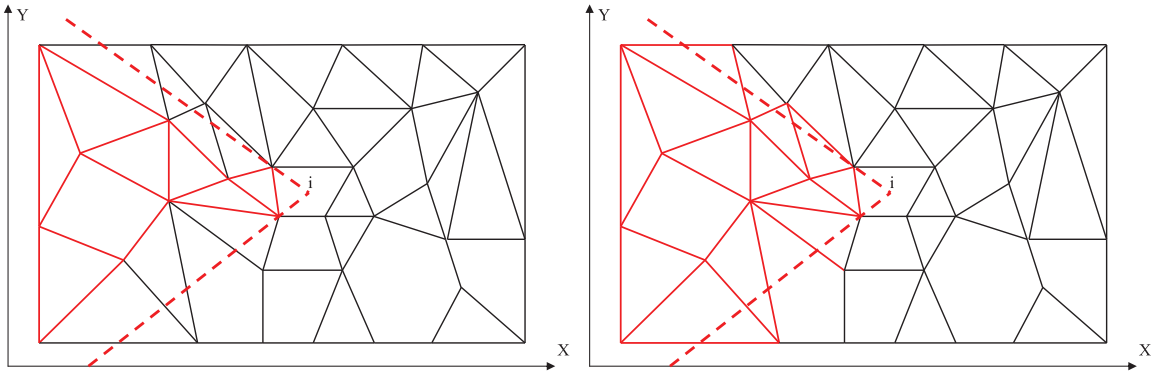
(a) Barycentre and edgecentre configuration (b) Edge and reverse edge configuration (c) Barycentre and edge configuration

Figure 2.8: Examples of directional sectors for stencils



(a) Planes of Triangular Face (b) Planes of Quadrilateral Face

Figure 2.9: Directional planes for directional stencils

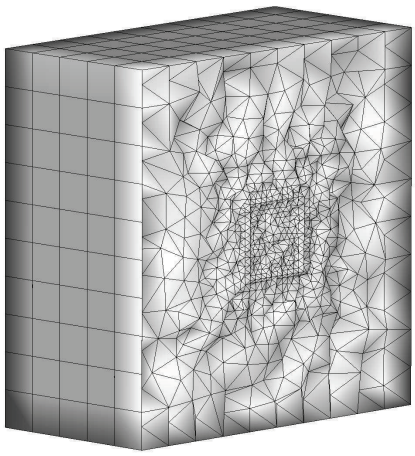


(a) 1 (b) 0.66

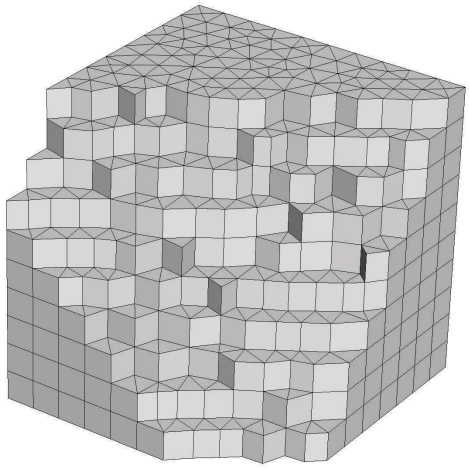
Figure 2.10: Directional stencils for different values of D_c

Algorithm 2 Directional Stencil Selection Algorithm

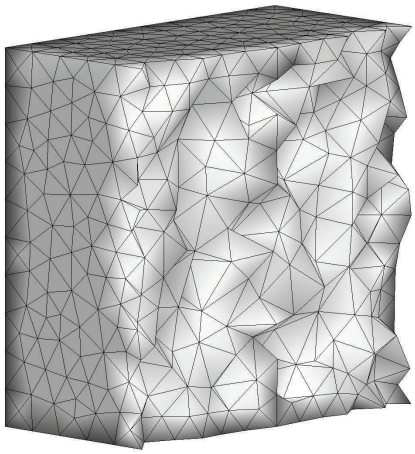
1. For cell i we want to construct a set of elements S (stencil) consisting of $N(S)$elements
2. With $c = 1, 2, ...N$ being the index of the numbering of the elements in the stencil
3. $S_1 = i$, the considered cell i is always the first element in the stencil $c = 1$
4. Assume that the element in the stencil S_c has M number of direct-side neighbours
   (a) Check which of the M elements do not belong in the set (Repetition Condition)
   (b) Check which of the elements that do not belong in the set satisfy the geometrical condition (D_c condition)
5. Assign as the next elements in the stencil only the ones that do not belong in the set
6. Repeat steps 4 to 8 until N number of elements have been assigned



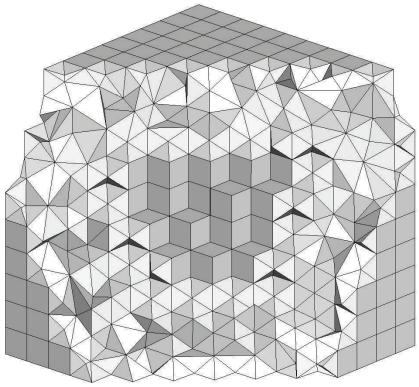
(a) Hybrid Configuration



(b) Prismatic Configuration



(c) Tetrahedral Configuration



(d) Hybrid Configuration

Figure 2.11: Types of Unstructured Meshess

Modern higher-order accurate finite-volume methods usually need a non-oscillatory reconstruction procedure to recover high-order accurate point-wise values of the solution from cell averages. The quality of reconstruction, together with that of the numerical flux and time advance method, defines the overall accuracy of the resulting finite-volume method.

In this section we describe a universal reconstruction procedure that is tailored for any type of structured and unstructured meshes in three-space dimensions which will then be used for the construction of the TVD and WENO schemes . Suppose that the spatial computational domain is discretised by conforming elements V_i of the volume $|V_i|$, indexed by a unique mono-index i. The center of the element has coordinates (x_i, y_i, z_i). The mesh can consist of hexahedral, tetrahedral, pyramidal, prismatic or any combination of them as shown in Figure 2.11 on page 31 . More general polyhedral shapes can also be considered, but are

omitted in the present work. In order to simplify the notation, we omit the global spatial index i and introduce the local numbering of cells in order to simplify notation. We note that it is sufficient to explain the idea of arbitrary high-order reconstruction for a scalar variable $u(x,y,z)$. The reconstruction problem can thus be reformulated as follows: for a target cell V_0 we would like to build a high-order polynomial $p(x,y,z)$ that has the same cell average as u on the target cell

$$\overline{u}_0 = \frac{1}{V_0} \int_{V_0} u(x,y,z)\, dV \tag{2.23}$$

The reconstruction procedure will use the cell averages of $u(x,y,z)$ on the target cell V_0 as well as averages $\bar{u}_m$ from the reconstruction stencil formed by neighboring cells V_m.

2.2.1 Linear (Central) Reconstruction

In general, the reconstruction can be carried out in the physical coordinates $\mathbf{x} = (x,y,z)$, taking special measures against scaling effects. However, a more elegant and computationally accurate approach is to use the so-called reference coordinate system (ξ,η,ζ), as was suggested in [22] for triangular (2D) and tetrahedral (3D) elements. Here we extend the transformation technique from [22] to deal with general mesh elements. The basic steps of our new procedure are as follows:

1. Decompose the considered cell into a number of simpler elements, which can be either tetrahedrals or hexahedrals.
2. Choose one of the resulting decomposed elements
3. Transform the chosen decomposed element from the physical space described by the Cartesian coordinates x,y,z into a reference space described by ξ,η,ζ
4. Based on the Jacobian matrix of the transformation of the chosen decomposed element, map the coordinates of the entire element into the reference space described by coordinates ξ,η,ζ
5. Based on the same Jacobian all the elements in the stencil are transformed to the reference space and their volumes, and barycentres positions are recomputed in the new reference space

In what follows we always decompose the general mesh element into tetrahedral elements. Let $\mathbf{v}_{ij}$, $j = 1,2,\ldots J_i$ be the vertices of the considered (general) element, which can be either tetrahedral, hexahedral, prismatic or pyramidal. Let also $\mathbf{w}_1 = (x_1,y_1,z_1)$, $\mathbf{w}_2 = (x_2,y_2,z_2)$, $\mathbf{w}_3 = (x_3,y_3,z_3)$, $\mathbf{w}_4 = (x_4,y_4,z_4)$ be the four vertices of one of the tetrahedrals this element is consisted of. Obviously, these vertices are between $\mathbf{v}_{ij}$ ones. The transformation from the Cartesian coordinates x, y, z into a reference space ξ, η, ζ is given by the following equations

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} + J \cdot \begin{pmatrix} \xi \\ \eta \\ \zeta \end{pmatrix} \tag{2.24}$$

with the Jacobian matrix given by

$$J = \begin{bmatrix} x_2 - x_1 & x_3 - x_1 & x_4 - x_1 \\ y_2 - y_1 & y_3 - y_1 & y_4 - y_1 \\ z_2 - z_1 & z_3 - z_1 & z_4 - z_1 \end{bmatrix} \tag{2.25}$$

Via the inverse mapping the element V_0 can be transformed to the element V_0' in the reference coordinate system

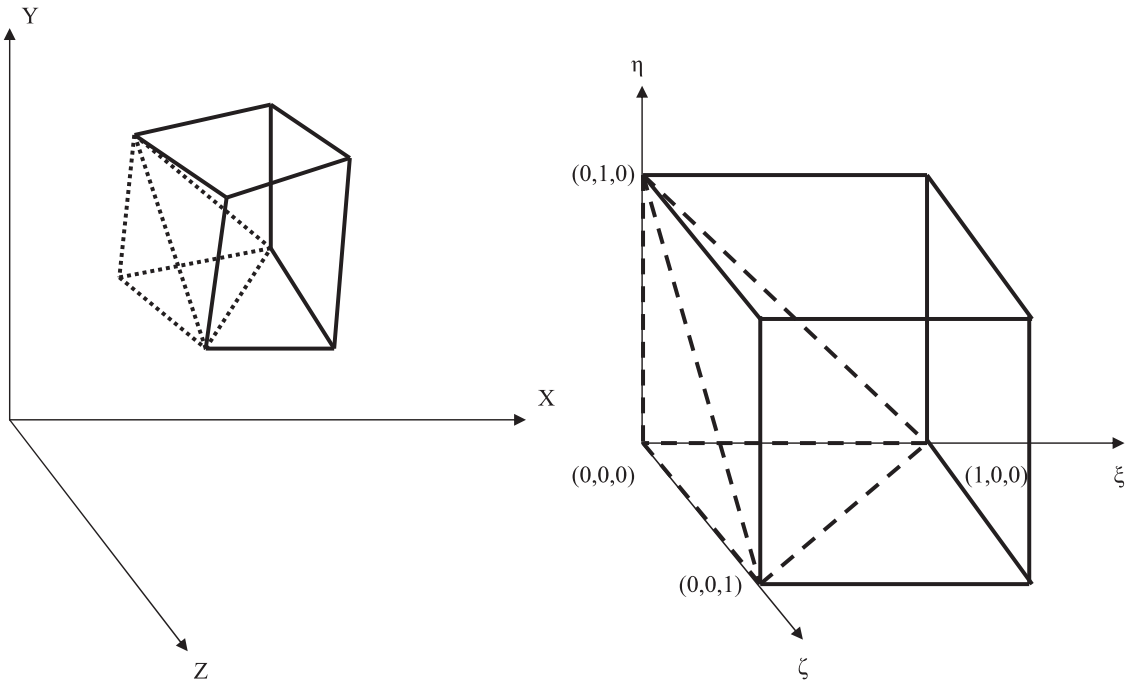
$$\mathbf{v}'_{ij} = J^{-1} \cdot \left(\mathbf{v}_{ij} - \mathbf{w}_1 \right), \quad j = 1, 2, \ldots J_i \tag{2.26}$$

Figure 2.12 on page 34 to Figure 2.15 on page 35 show the results of the transformation for various element types. Note that for the uniform (Cartesian) hexahedral mesh the transformed element V_0' is just a unit cube in the reference space ξ, η, ζ. For general non-uniform meshes the transformed element will not be unit cube, however, four of its vertices will still be from the unit square. This is different from the case of tetrahedral meshes in which each cell is transformed in the unique reference triangle, see [22] for more details. Note that spatial averages of $u(x,y,z)$ does not change during transformation:

$$\bar{u}_0 = \frac{1}{|V_0|} \int\limits_{V_0} u(x,y,z)\, dV \equiv \frac{1}{|V_0'|} \int\limits_{V_0'} u(\xi,\eta,\zeta)\, d\xi d\eta d\zeta$$

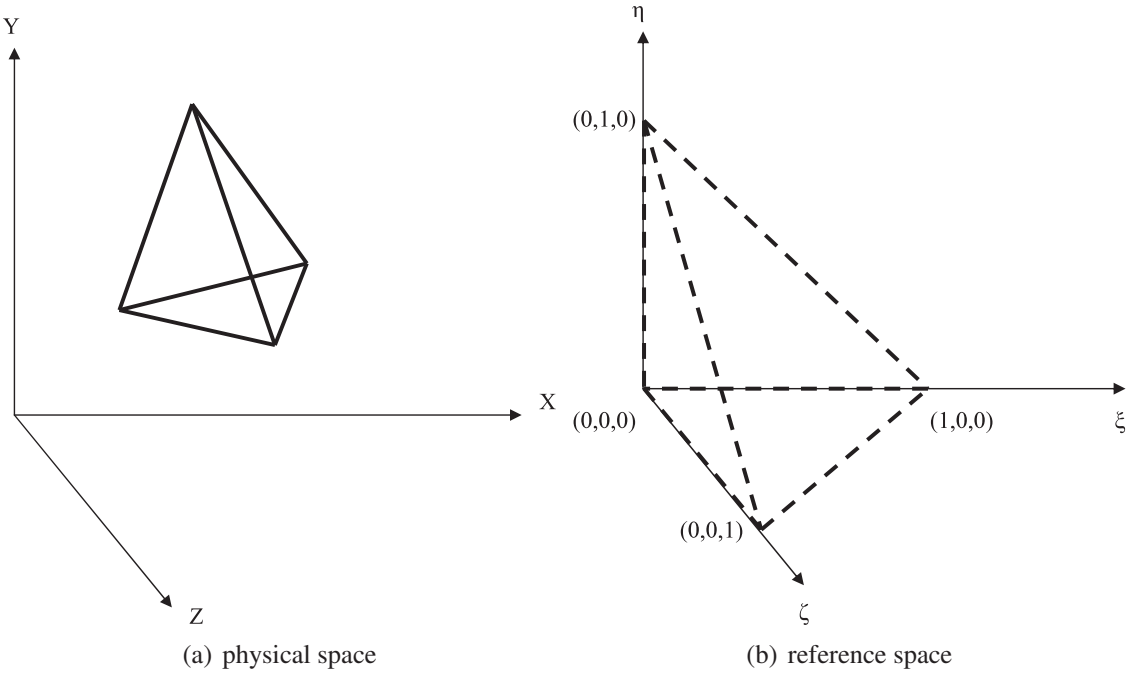
For performing the reconstruction on the target element E_0, we form the so-called central reconstruction stencil $\mathscr{S}$ which will consist of $M+1$ elements, including the target element E_0:

$$\mathscr{S} = \bigcup_{m=0}^{M} V_m$$



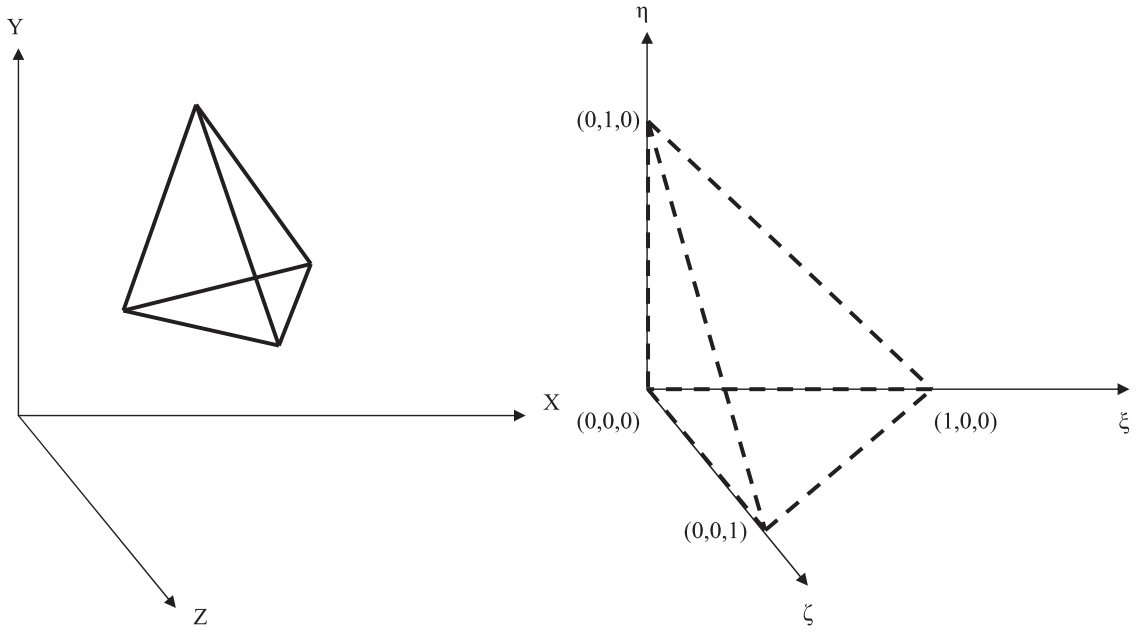
(a) physical space (b) reference space

Figure 2.12: Hexahedral transformation



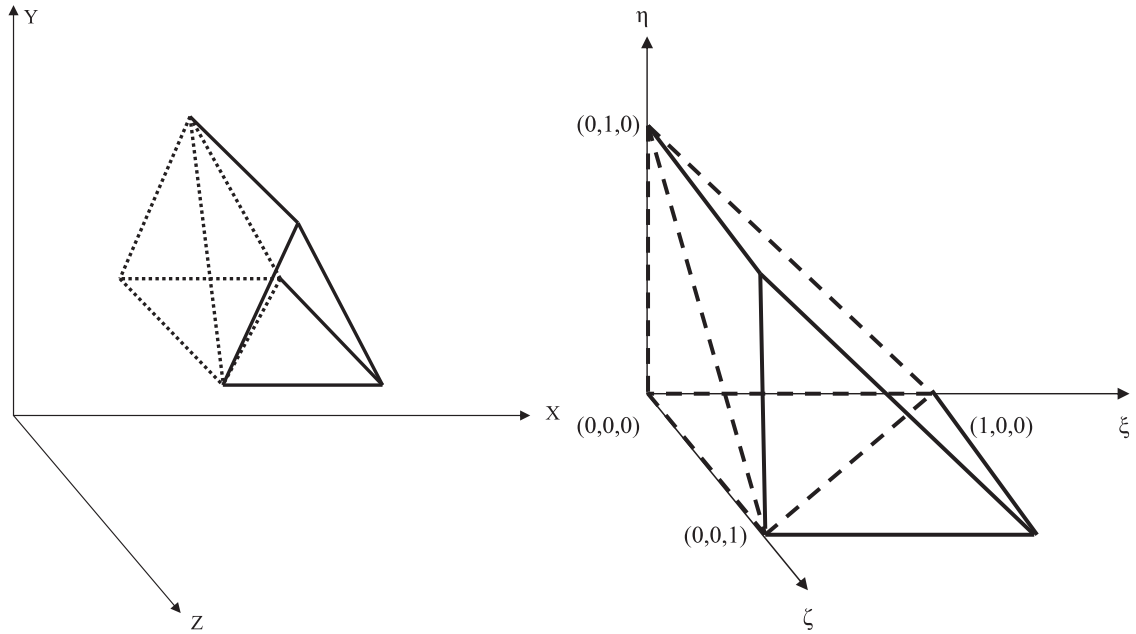
(a) physical space (b) reference space

Figure 2.13: Tetrahedral transformation



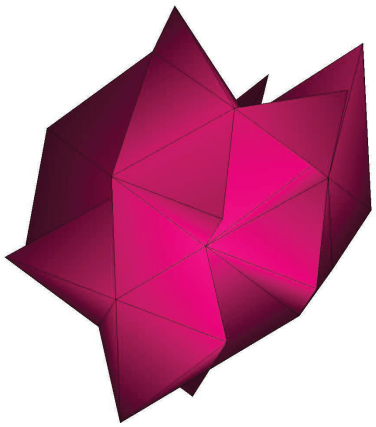
(a) physical space (b) reference space

Figure 2.14: Tetrahedral transformation

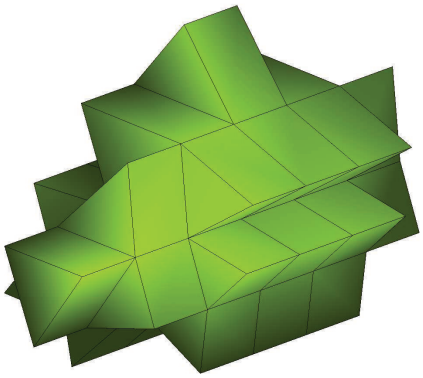


(a) physical space (b) reference space

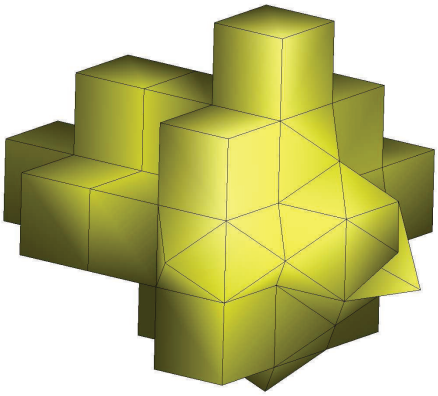
Figure 2.15: Prism transformation



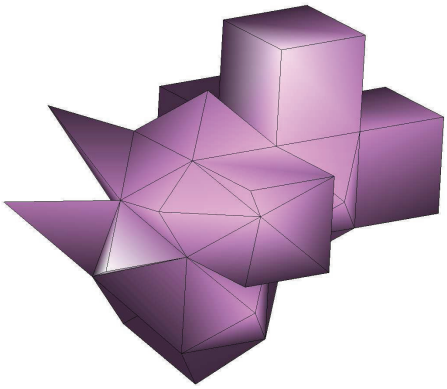
(a) Tetrahedral elements



(b) Prismatic



(c) mixed-elements



(d) mixed-elements

Figure 2.16: Central stencils of various element shapes

where the local index m counts the elements in the stencil $\mathscr{S}$. This central stencil is build up by recursively adding the direct side neighbors of the element V_0 and all of the elements added to the stencil so far, until the desired number of stencil elements is reached. Typical examples of central stencils can be seen in Figure 2.16 on page 36. We then apply the inverse mapping 2.26 to all the elements V_m from the reconstruction stencil $\mathscr{S}$ and denote the transformed elements and stencil as E'_m and $\mathscr{S}'$, respectively:

$$\mathscr{S}' = \bigcup_{m=0}^{M} V'_m$$

One of the contributions of the present research is the development of very high-order methods which can use mixed-element meshes consisting of hexahedrals, tetrahedrals, prisms and pyramids. The transformation from cartesian coordinates to reference coordinates is valid for tetrahedral elements (since a tetrahedral is always mapped into a unit tetrahedral in reference space) that are not highly stretched and we always ensure in the mesh generation that we do not have highly stretched elements. It should be stressed that there is no reduction of the formal order of accuracy at the interface between cells of two different types. Our stencil construction procedure allows to use stencils comprising of cells of different types. The central stencil is build up by adding neighbors in the physical space x, y irrespective of their shape.

The r^{th} order reconstruction polynomial at the transformed cell V'_0 is sought as an expansion over local polynomial basis functions $\phi_k(\xi,\eta,\zeta)$:

$$p(\xi,\eta,\zeta) = \sum_{k=0}^{K} a_k \phi_k(\xi,\eta,\zeta) = \bar{u}_0 + \sum_{k=1}^{K} a_k \phi_k(\xi,\eta,\zeta) \tag{2.27}$$

where a_k are degrees of freedom and the upper index in the summation of expansion K is related to the order of the polynomial r by the expression $K = \frac{1}{6}(r+1)(r+2)(r+3)-1$. The conservation condition 2.23 impose an important constraint on the basis functions: they must have zero mean value over the cell V'_0. On purely tetrahedral meshes hierarchical orthogonal reconstruction basis functions defined on the reference element satisfy this requirement automatically [22]. Since our general cells are not necessarily transformed onto a unit tetrahedron or cube, we need to construct basis functions ϕ_k in such a way that condition 2.23 is satisfied identically irrespective of values of degrees of freedom. We define the basis functions as follows:

$$\phi_k(\xi,\eta,\zeta) \equiv \psi_k(\xi,\eta,\zeta) - \frac{1}{|V'_0|}\int_{V'_0} \psi_k \, d\xi d\eta d\zeta, \quad k = 1,2,\ldots \tag{2.28}$$

where

$$\{\psi_k\} = \xi,\ \eta,\ \zeta,\ \xi^2,\ \eta^2,\ \zeta^2,\ \xi\cdot\eta,\ \xi\cdot\zeta,\ \zeta\cdot\eta,\ \xi\cdot\eta\cdot\zeta\ldots$$

The resulting expression 2.28 for basis functions is suitable for cells of arbitrary shape.

To find the unknown degrees of freedom a_k we require that for each cell V'_m from the stencil the cell average of the reconstruction polynomial $p(\xi,\eta,\zeta)$ be equal to the cell average of the solution $\bar{u}_m$:

$$\int\limits_{E'_m} p(\xi,\eta,\zeta)\,d\xi d\eta d\zeta = |V'_m|\bar{u}_0 + \sum_{k=1}^{K} \int\limits_{V'_m} a_k\phi_k\,d\xi d\eta d\zeta = |V'_m|u_m, \quad m = 1,\dots M$$

Denoting the integrals of the basis function k over the cell m in the stencil the vector of right-hand side by A_{mk} and $\mathbf{b}$, respectively

$$A_{mk} = \int\limits_{V'_m} \phi_k\,d\xi d\eta d\zeta, \quad b_m = |V'_m|(\bar{u}_m - \bar{u}_0)$$

we can rewrite the equations for degrees of freedom a_k in the matrix form as

$$\sum_{k=1}^{K} A_{mk}a_k = b_m, \quad m = 1,2,\dots M \tag{2.29}$$

The three-dimensional integrals on the left-hand side of (2.29) are calculated using Gaussian quadratures of appropriate orders [64].

In general, in order to compute the degrees of freedom a_k we need at least K cells in the stencil, different from the target cell E_0. However, the use of the minimum possible number of cells in the stencil $M \equiv K$ results in a scheme which may become unstable on general meshes. It is therefore recommended to use more cells in the stencil then the minimal required number [5, 22]. Although it is usually sufficient to use 50% more cells, for mixed-element meshes it is safer to increase the stencil further. We typically select $M = 2 \cdot K$.

Since the resulting system 2.29 becomes over-determined, the least-square procedure is invoked to solve it. The least-square reconstruction of 2.29 is obtained by seeking the minimum of the following functional

$$\mathscr{F} = \sum_{m=1}^{M} \omega_m \cdot \left(\sum_{k=1}^{K} A_{mk}a_k - b_m \right)^2$$

where the weights ω_m are squared reciprocals of the distance between cells E'_0 and E_m. The advantage of the weighted least square reconstruction is that the influence of the data farther from the considered E'_0 is reduced [24, 17, 57], although a central least square reconstruction is materialised with the weights ω_m being equal to unity. Minimization of $\mathscr{F}$ gives a linear system for finding a_k:

$$\sum_{k=1}^{K} C_k a_k = \sum_{m=1}^{M} A_{mp}\omega_m b_m, \quad C_k = \left(\sum_{m=1}^{M} \omega_m A_{mk}A_{mp} \right), \quad p = 1,\dots K \tag{2.30}$$

Although the least square reconstruction has been adopted for various schemes for unstructured grids [33, 18, 63, 17, 24, 77] there are certain issues that must be addressed. This

procedure for solving the resulting least square is not suitable since for unstructured grids the resulting system can be ill-conditioned , hence the procedure can be inaccurate. Therefore a QR decomposition method is employed to solve this system of equations. One numerically stable technique that is also suitable for ill-conditions systems is the Householder transformation in which a vector is reflected in some plane in such a way that all coordinates but one disappear. We remark, that the coefficients of the resulted linear symmetric matrix A are precomputed and stored for each element during the preprocessing stage of the calculation increasing the computational efficiency of the method. Having solved numerically the linear system 2.30, we can form the reconstruction polynomial 2.27 .

2.2.2 TVD Reconstruction

It is well known from Godunov's theorem [25] that linear high-order scheme produce spurious oscillations when applied to discontinuous solutions. To circumvent this non-linear solution adaptive methods are designed. It appears that the first second-order non-oscillatory Godunov-type scheme on unstructured meshes was proposed in [67] and uses an extension of the minmod slope of Kolgan originally devised for structured meshes [39, 38]. More elaborate version which uses higher-order polynomials was proposed in [5, 19]. We have employed a TVD reconstruction process based on the central reconstruction as a more cost-efficient alternative to the non-linear WENO reconstruction. Although this TVD implementation can employ the central reconstruction process of any order of accuracy in practise it is 2nd-order accurate in space.

The key ingredient of this technique is that we perform the linear-(central) reconstruction as before but we restrict the reconstructed value to lie within a minimum and maximum limit. Although the central stencil is used as previously for the reconstruction the minimum and maximum values that restrict the reconstructed solution are obtained only from the direct side neighbours (TVD stencil) of the considered element as shown in Figure 2.17 on page 40.

The basic steps of the TVD reconstruction are as follows:

1. Perform the linear reconstruction process as before by using the central stencil and we obtain the values for the degrees of freedom a_k
2. Obtain the minimum and maximum values u_{min}^{TVD}, u_{max}^{TVD} of the scalar $u(x,y,z)$ from the TVD stencil
3. Determine the reconstructed value u_p at the vertices of the considered element
4. Compute the slope limiter ψ_i of cell i with cell average $\bar{u}_i$

$$\psi_i = \begin{cases} if\ u_p - \bar{u}_i > 0 \ \Rightarrow\ min\left(\frac{u_{max}^{TVD} - u_p}{\bar{u}_i - u_p}\right) \\ if\ u_p - \bar{u}_i < 0 \ \Rightarrow\ min\left(\frac{u_{min}^{TVD} - u_p}{\bar{u}_i - u_p}\right) \\ if\ u_p - \bar{u}_i = 0 \ \Rightarrow\ 1 \end{cases} \tag{2.31}$$

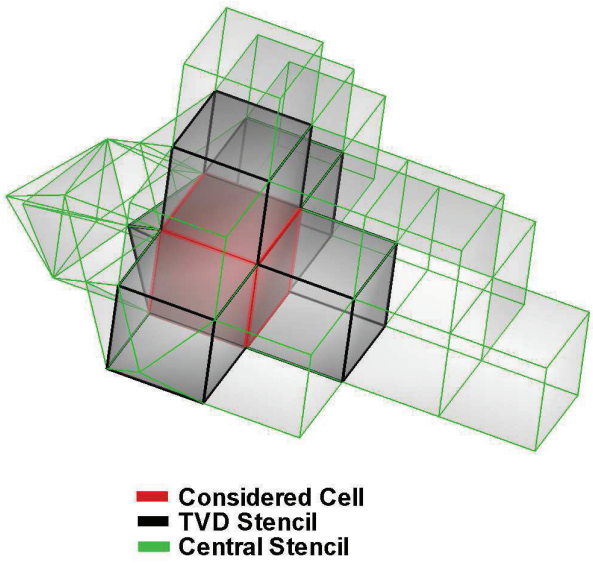


Figure 2.17: TVD Central Stencil

5. The reconstructed value u_p is then limited by the slope limiter ψ_i

$$u_p = \bar{u}_i + \psi_i \cdot \sum_{k=1}^{K} a_k \phi_k (\xi, \eta, \zeta) \tag{2.32}$$

2.2.3 WENO Reconstruction

Although TVD type of methods are quite robust and cost-efficient schemes they are not very suitable for certain applications involving long-time evolution of smooth structures. Significant improvement in accuracy may be obtained by considering the so-called essentially non-oscillatory (ENO) and weighted ENO methods [30, 44]. These reconstructions use reconstruction polynomials from several different stencils. In particular, in WENO schemes the actual reconstructed value is a convex combination of reconstructed values from stencils, with nonlinear (solution-adaptive) WENO weights.These nonlinear weights are constructed from the linear (constant) weights by taking into account smoothness of the solution in each of the reconstruction stencils. The key difference of WENO approach from TVD-type reconstructions is that the resulting methods are uniformly high-order accurate while still maintaining non-oscillatory behaviour at discontinuities.

Details on existing types of WENO reconstructions for tetrahedral meshes can be found as well as references therein [22, 78]. The reconstruction proposed in the present work is an extension of the approach from [22] to mixed-element meshes, consisting of elements of arbitrary shapes. The WENO reconstruction stencils is a union of several reconstruction stencils $\mathscr{S}_m$, $m = 0, 1, \ldots, m_s$. These are one central stencil and several one-sided, or sectorial,

stencils. The construction of the central stencil $\mathscr{S}_0$ was outlined in the previous sections. The sectorial stencils are obtained by adding only those neighboring cells, centres of which lie inside the given sector. Each sector in the present work is defined by the cell centre and a face of the cell, which is different from [22] . Since the sectors constructed in such a way cover all possible directions, we do not need to use the so-called reverse sectors suggested in [36] . The number of sectorial stencils in our schemes is usually equal to the number of faces of the cell. Note, that the number of stencils may be smaller near solid boundaries. Overall, our reconstruction procedure thus uses a significantly smaller number of stencils compared to the original construction, where $m_s = 6$ is used for triangular elements. Figure 2.18 on page 42 and Figure 2.19 on page 43 illustrate the directional stencils for a mixed-element and an tetrahedral unstructured mesh in the physical coordinate system.

The WENO reconstruction polynomial is now defined as a non-linear combination of reconstruction polynomials $p_m(\xi,\eta,\zeta)$, obtained by using individual stencils $\mathscr{S}_m$:

$$p_{\text{weno}} = \sum_{m=0}^{m_s} \omega_m p_m(\xi,\eta,\zeta) \tag{2.33}$$

Substituting the form of the individual polynomial corresponding to the stencil $\mathscr{S}_m$

$$p_m(\xi,\eta,\zeta) = \sum_{k=0}^{K} a_k^{(m)} \phi_k(\xi,\eta,\zeta)$$

and using the condition where $\sum\limits_m \omega_m \equiv 1$, we obtain

$$p_{\text{weno}} = \sum_{m=0}^{m_s} \omega_m \left(\sum_{k=0}^{K} a_k^{(m)} \phi_k(\xi,\eta,\zeta) \right) = \bar{u}_0 + \sum_{m=0}^{m_s} \omega_m \left(\sum_{k=1}^{K} a_k^{(m)} \phi_k(\xi,\eta,\zeta) \right) \tag{2.34}$$

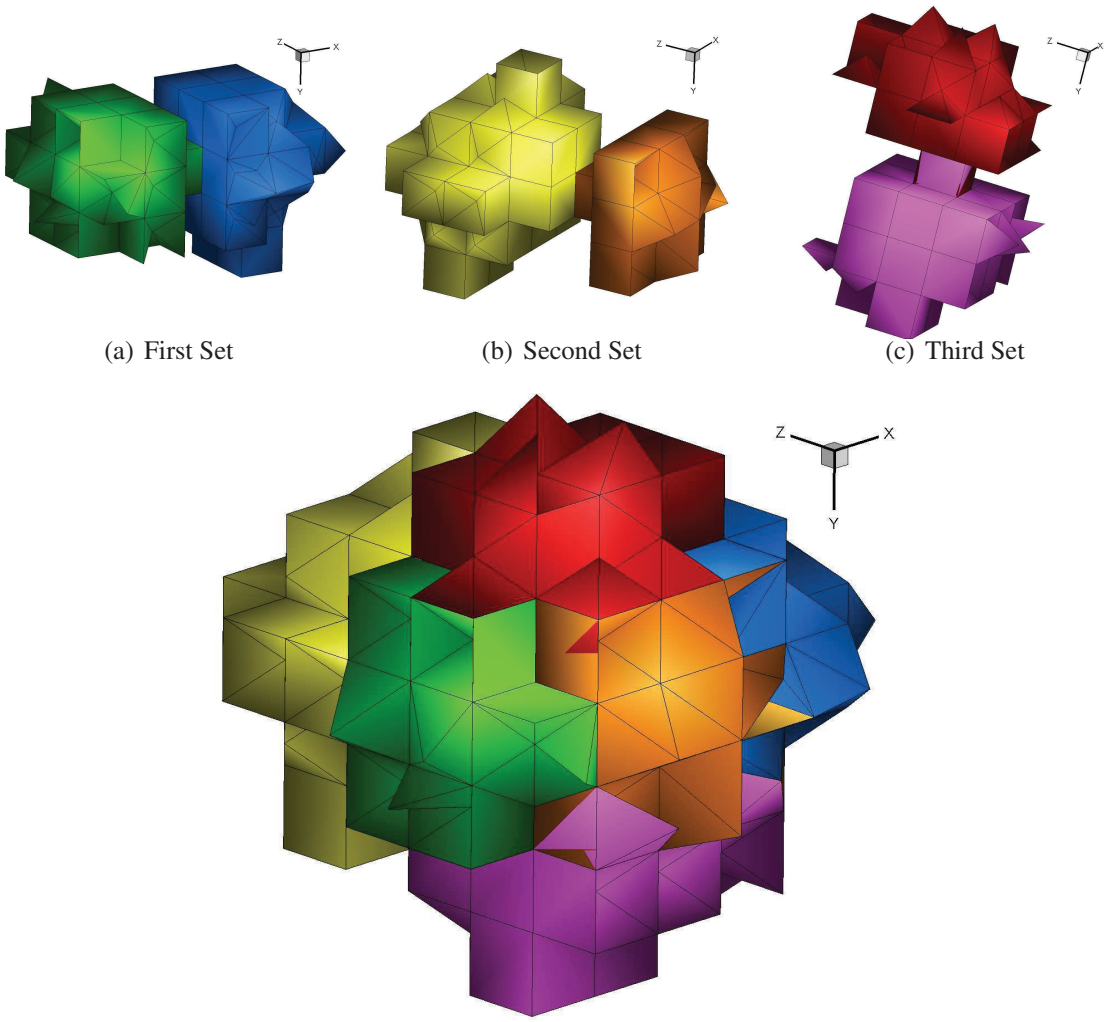
Further reordering yields

$$p_{\text{weno}} = \bar{u}_0 + \sum_{k=1}^{K} \left(\sum_{m=0}^{m_s} \omega_m a_k^{(m)} \right) \phi_k(\xi,\eta,\zeta) \equiv \bar{u}_0 + \sum_{k=1}^{K} \tilde{a}_k \phi_k(\xi,\eta,\zeta) \tag{2.35}$$

Here $\tilde{a}_k$ are the new values of degrees of freedom, modified according to the WENO procedure. As is usual in WENO methods [34, 32] , the nonlinear weights ω_m are defined as

$$\omega_m = \frac{\gamma_m}{\sum\limits_{m=0}^{m_s} \gamma_m}, \quad \gamma_m = \frac{d_m}{(\varepsilon + IS_m)^p}$$

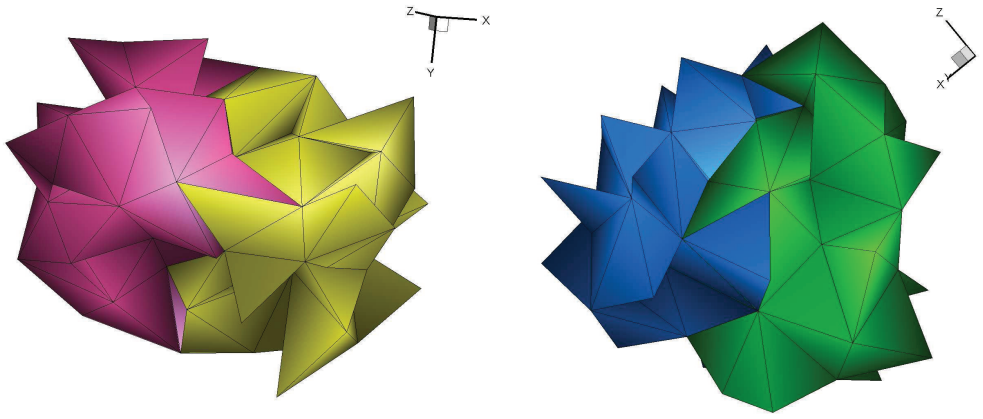
where d_m are the so-called linear weights, IS_m are smoothness indicators, ε is a small number used to avoid division by zero and finally p is an integer parameter, controlling how fast the non-linear weights decay for non-smooth stencils. We typically use $\varepsilon = 10^{-6}$ and $p = 4$. Note, that for some applications the choice of these two parameters may have a profound effect on the numerical solution, see e.g [69] .



(a) First Set (b) Second Set (c) Third Set

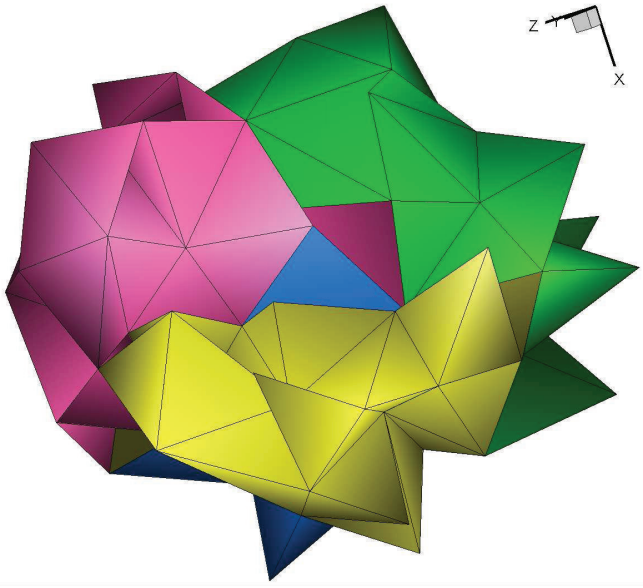
(d) Complete Set

Figure 2.18: WENO directional stencils for a mixed element mesh



(a) First Set

(b) Second Set



(c) Complete Set

Figure 2.19: WENO directional stencils for a tetrahedral element mesh

Our selection of linear weights d_m follows [22]. The central stencil is assigned a large linear weight $d_0 = 10^2 \dots 10^5$ whereas the sectorial stencils are assigned smaller weights $d_m = 1$. This selection of the weights is motivated by the fact that for smooth solutions the central stencil is usually the most accurate one. We also note that a similar concept was used in in two spatial dimensions with equal weights assigned to all stencils [36].

The oscillation indicators IS_m of each stencil is a measurement of how smooth the solution is on this stencil. Due to the use of the reference coordinate system, scaling is already taken out of the problem and IS_m can be computed in a mesh-independent manner as

$$IS_m = \sum_{1<|\beta|<r} \int_{V_0'} \left(D^\beta p_m(\xi,\eta,\zeta) \right)^2 d\xi d\eta d\zeta \tag{2.36}$$

where β is a multi-index [40, 57], r is the order of the polynomial and D is the derivative operator. The general form of D in three space dimensions can be found in [22]. It is easily seen that the smoothness indicators are quadratic functions of degrees of freedom $a_k^{(m)}$ and thus the expression can be rewritten in terms of the so-called universal oscillation indicator matrix [22] . If the mesh consists of tetrahedral elements only, then this matrix does not depend on the element. For general elements it will, however, depend on the element. For efficiency, it can be precomputed and stored at the beginning of the calculations for each element V_i.

Chapter 3

Extension to the Compressible Euler Equations

Introduction

In this chapter we describe the application of the developed schemes for the compressible Euler equations in three-space dimensions. The chapter is structured as follows. We first present the application of the reconstruction procedure to the Euler equations with respect to characteristic variables. Next we outline the numerical flux approximation by approximate Riemann solvers, the time advancement of the solution and finally the implementation of the boundary conditions in the context of three-dimensional mixed-element unstructured meshes.

3.1 Euler Equations

In this section we consider the three-dimensional Euler equations in the following formulation

$$\frac{\partial}{\partial t}\mathbf{U}+\frac{\partial}{\partial x}\mathbf{F}(\mathbf{U})+\frac{\partial}{\partial y}\mathbf{G}(\mathbf{U})+\frac{\partial}{\partial z}\mathbf{H}(\mathbf{U})=\mathbf{0} \tag{3.1}$$

where $\mathbf{U}$ is the vector of the conserved variables, $\mathbf{F}$, $\mathbf{G}$, $\mathbf{H}$ are the flux vectors in x, y and z Cartesian coordinates directions respectively given by

$$\left.\begin{aligned} \mathbf{U} &= \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ E \end{bmatrix}, \qquad \mathbf{F} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ u(E+p) \end{bmatrix} \\ \mathbf{G} &= \begin{bmatrix} \rho v \\ \rho vu \\ \rho v^2 + p \\ \rho vw \\ v(E+p) \end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix} \rho w \\ \rho wu \\ \rho wv \\ \rho w^2 + p \\ w(E+p) \end{bmatrix} \end{aligned}\right\}$$

here $\rho$is density, u,v,w -velocity components in x,y and z directions respectively, p - pressure, $E = p/(\gamma-1)+(1/2)\rho(u^2+v^2+w^2)$ - total energy per unit mass, γ is the ratio of specific heats. We use $\gamma = 1.4$ throughout.

Integrating (3.1) in space over a mesh element V_i, and be exploiting the rotational invariance property of the Euler equations [70] we obtain the following semi-discrete finite-volume method :

$$\frac{d}{dt}\mathbf{U}_i + \frac{1}{|V_i|}\oint_{\partial V_i} \mathbf{F}_n dA = \mathbf{0}, \quad \mathbf{F}_n(\mathbf{U}) = \mathbf{F}(\mathbf{U})n_x + \mathbf{G}(\mathbf{U})n_y + \mathbf{H}(\mathbf{U})n_z = \mathbf{T}^{-1}\mathbf{F}(\mathbf{TU}) \tag{3.2}$$

where $\mathbf{n} = (n_x, n_y, n_z)$ is outward unit normal vector, $\mathbf{U}_i(t)$ are the cell averages of the solution at time t, $\mathbf{F}_n$ - projection of the flux tensor on the normal direction, $\mathbf{T}$ is the rotation matrix and $\mathbf{T^{-1}}$ its inverse given by

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & cos\theta sin\varphi & sin\theta sin\varphi & cos\varphi & 0 \\ 0 & -sin\theta & cos\theta & 0 & 0 \\ 0 & cos\theta cos\varphi & sin\theta cos\varphi & -sin\varphi & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.3}$$

$$\mathbf{T^{-1}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & cos\theta sin\varphi & -sin\theta & cos\theta cos\varphi & 0 \\ 0 & sin\theta sin\varphi & cos\theta & sin\theta cos\varphi & 0 \\ 0 & cos\varphi & 0 & -sin\varphi & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.4}$$

where θ is the azimuthal angle and φ is the polar angle as defined in (2.15) and (2.16) respectively.

Assume that the element's surface consists of L faces (here we omitted the spatial index i for simplicity):

$$\partial V_i = \sum_j^L A_j$$

Also denote by $\mathbf{n}_j$ the outward unit vector for face A_j. Then the integral over the element boundary ∂V_i splits into the sum of integrals over each face resulting in the following expression:

$$\frac{d}{dt}\mathbf{U}_i = \mathbf{R}_i, \quad \mathbf{R}_i = -\frac{1}{|V_i|}\sum_{j=1}^{L}\int_{A_j}\mathbf{F}_{n,j}dA = -\frac{1}{|V_i|}\sum_{j=1}^{L}\mathbf{K}_{ij} \tag{3.5}$$

Here the numerical flux $\mathbf{K}_{ij}$ corresponding to the face j of the cell V_i is the surface integral of the projection of the tensor of fluxes onto $\mathbf{n}_j$. In a numerical method the exact integral expression for the numerical flux $\mathbf{K}_{ij}$ for the face j of a cell V_i is approximated by a suitable Gaussian numerical quadrature:

$$\mathbf{K}_{ij} = \int_{A_j}\mathbf{F}_{n,j}dA = \sum_{\beta}\mathbf{F}_{n,j}\left(\mathbf{U}(\mathbf{x}_\beta,t)\right)\omega_\beta|A_j| \tag{3.6}$$

where the subscript β corresponds to different Gaussian integration points $\mathbf{x}_\beta$ and weights ω_β over the face A_j.

3.2 Reconstruction for systems

Calculation of a numerical flux (3.6) through the face A_j of a cell V_i requires the knowledge of point-wise values of the conserved vector $\mathbf{U}$ at the Gaussian points. However, the numerical method advances in time the cell averages of the conserved vector. Therefore, we have to obtain high-order approximation to the point-wise values of the conserved vector at each Gaussian point of a face by using some high-order non-oscillatory reconstruction procedure. Apart from the accuracy requirements, this procedure must also satisfy the conservation condition, namely the cell average of the reconstruction polynomial over the cell V_i is equal to $\mathbf{U}_i$.

Here we employ the TVD and WENO reconstruction procedure on mixed-element meshes, based on reconstruction procedure for a scalar function u, developed in the previously and extended here in to vector variables, which are solutions of the compressible Euler equations. The reconstruction produces the high-order reconstruction polynomials $\mathbf{P}_i(\xi,\eta,\zeta)$ defined in the reference coordinate system. The simplest approach to the construction of a reconstruction polynomial $\mathbf{P}_i$ is to apply the scalar reconstruction procedure, developed in the previously, to each component of $\mathbf{U}$. In other words, the conventional component-wise WENO reconstruction polynomial is given by applying (2.34), (2.35) to each component of the conserved vector $\mathbf{U}$. For the TVD scheme we perform the reconstruction process in a component-wise manner, hence we employ the TVD slope-limiter (2.31) for each conserved

variable. However for higher than 2^{nd}-order of accuracy the reconstruction procedure should be carried out in characteristic variables rather than conservative variables. It can be shown that the use of conservative variables (in a component-wise manner) in the reconstruction results in considerable spurious oscillations even for simple shock-tube problems, see e.g. [60]. Moreover, these spurious oscillations do not vanish as the mesh is refined. Therefore, in the present thesis the WENO reconstruction is carried out in characteristic variables. Our approach for extending the scalar reconstruction to the characteristic-based reconstruction is very similar to that of [49], although different in some respects, and thus we only outline the main steps.

Consider the cell V_i and the corresponding set of directional stencils in the local reference coordinate system $\{S'_m\}$, $m = 0, 1, \ldots m_s$. Calculate the vector degrees of freedom $\mathbf{A}_{ik}^{(m)}$ for each stencil, applying the linear scalar reconstruction procedure in the component-wise fashion. Then, the corresponding polynomials are given by

$$\mathbf{P}_{im}(\xi,\eta,\zeta) = \sum_{k=0}^{K} \mathbf{A}_{ik}^{(m)} \phi_{ik}(\xi,\eta,\zeta) = \bar{\mathbf{U}}_i + \sum_{k=1}^{K} \mathbf{A}_{ik}^{(m)} \phi_{ik}(\xi,\eta,\zeta), \tag{3.7}$$

where ϕ_{ik} are basis functions for cell V_i in the local reference coordinate system.

Define as the arithmetic average of the conserved vector $\mathbf{U_i}$ and the conserved vector $\hat{\rho}$, corresponding to the computational cell, adjacent to the face A_j of the current cell V_i:

$$\mathbf{U}'_n = \frac{1}{2}(\mathbf{U}_i + \mathbf{U}_{i'}).$$

Hence

$$\rho_s = \frac{1}{2}\left(\hat{\rho}_i + \hat{\rho}_{i'}\right) \tag{3.8}$$

$$u_s = \frac{1}{2}\left(\hat{u}_L + \hat{u}_{i'}\right) \tag{3.9}$$

$$v_s = \frac{1}{2}\left(\hat{v}_L + \hat{v}_{i'}\right) \tag{3.10}$$

$$w_s = \frac{1}{2}\left(\hat{w}_L + \hat{w}_{i'}\right) \tag{3.11}$$

$$E_s = \frac{1}{2}\left(\hat{E}_L + \hat{E}_{i'}\right) \tag{3.12}$$

$$\frac{1}{2}V_s = \frac{1}{2}\left(u_s^2 + v_s^2 + w_s^2\right) \tag{3.13}$$

Where $\hat{\rho}$, $\hat{u}$, $\hat{v}$, $\hat{e}$, $\hat{E}$ correspond to the rotated density, u-velocity, v-velocity, w-velocity, and Energy. Let $\mathbf{R_j}$, $\mathbf{L_j}$ be the matrices containing the right and left eigenvectors of the Jacobian

matrix $\mathbf{H}_j$, corresponding to the normal projection of the flux tensor calculated at this average state

where

$$\mathbf{H}_j = \frac{\partial \mathbf{F}_n}{\partial \mathbf{U}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ (\gamma-1)H_s - u_s^2 - a_s^2 & (3-\gamma)u_s & -(\gamma-1)v_s & -(\gamma-1)w_s & (\gamma-1) \\ -u_s v_s & v_s & u_s & 0 & 0 \\ -u_s w_s & w_s & 0 & u_s & 0 \\ \frac{1}{2}u_s\left[(\gamma-3)H_s - a^2\right] & H_s - (\gamma-1)u_s^2 & -(\gamma-1)uv & -(\gamma-1)u_s w_s & \gamma u_s \end{bmatrix} \tag{3.14}$$

$$H_s = (E_s + p_s)/\rho_s = \frac{1}{2}V_s^2 + \frac{a_s^2}{(\gamma-1)} \tag{3.15}$$

$$a_s = \sqrt{\frac{\gamma p_s}{\rho_s}} \tag{3.16}$$

$$\mathbf{R_j} = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 \\ u_s - a_s & u_s & 0 & 0 & u_s + a_s \\ v_s & v_s & 1 & 0 & v_s \\ w_s & w_s & 0 & 1 & w_s \\ H_s - u_s a_s & \frac{1}{2}V_s^2 & v_s & w_s & H_s + u_s a_s \end{bmatrix} \tag{3.17}$$

$$\mathbf{L_j} = \frac{(\gamma-1)}{2a_s^2} \begin{bmatrix} H_s + \frac{a_s}{(\gamma-1)}(u_s - a_s) & -\left(u_s - \frac{a_s}{(\gamma-1)}\right) & -v_s & -w_s & 1 \\ -2H_s + \frac{4}{(\gamma-1)}a_s^2 & 2u_s & 2v_s & 2w_s & -2 \\ -\frac{2v_s a_s^2}{(\gamma-1)} & 0 & \frac{2a_s^2}{(\gamma-1)} & 0 & 0 \\ -\frac{2w_s a_s^2}{(\gamma-1)} & 0 & 0 & \frac{2a_s^2}{(\gamma-1)} & 0 \\ H - \frac{a_s}{(\gamma-1)}(u_s - a_s) & -u_s + \frac{a_s}{(\gamma-1)} & -v_s & -w_s & 1 \end{bmatrix} \tag{3.18}$$

where $\mathbf{F}_n$ is defined in (3.2) H_s is the enthalpy and a_s is the speed of sound. The crucial step now is to compute the characteristic projections of the vector of the degrees of freedom of each stencil S_m, including the cell averaged value $\mathbf{U_I}$ as

$$\mathbf{B}_{ikj}^{(m)} = \mathbf{L}_j \mathbf{A}_{ik}^{(m)}, \quad m = 0, \ldots, m_s, \quad k = 0, \ldots K.$$

We now apply the scalar WENO reconstruction algorithm to each component of the projected degrees of freedom. The resulting modified degrees of freedom $\tilde{\mathbf{B}}_{ikj}^{(m)}$ are projected back to by multiplying them by $\mathbf{R_j}$. The resulting WENO reconstruction polynomial for the face A_j is given by

$$\mathbf{P}_{ij}(\xi,\eta,\zeta) = \bar{\mathbf{U}}_i + \sum_{k=1}^{K} \tilde{\mathbf{A}}_{ikj}\phi_{ik}(\xi,\eta,\zeta), \quad \tilde{\mathbf{A}}_{ikj} = \mathbf{R}_j \mathbf{B}_{ikj}. \tag{3.19}$$

Note, that the degrees of freedom in (3.19) depend on the face index j. Finally, the reconstructed values at Gaussian integration points are then given by

$$\mathbf{P}_i(\xi_\beta, \eta_\beta, \zeta_\beta) = \bar{\mathbf{U}}_i + \sum_{k=1}^{K} \mathbf{A}_{ikj} \phi_{ik}(\xi_\beta, \eta_\beta, \zeta_\beta) \tag{3.20}$$

where $(\xi_\beta,\ \eta_\beta,\ \zeta_\beta)$ are the coordinates of Gaussian points in the reference coordinate system for the face A_j of the cell V_i. We note that the values of the basis functions at Gaussian integration points can be calculated and stored during the pre-processing step, increasing the efficiency of the method.

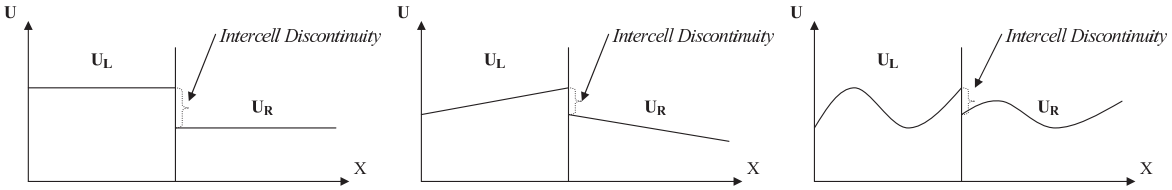
An additional step in the reconstruction process was used in [49]. Namely, for each cell V_i the least oscillatory of all $\mathbf{P}_{ij}$ is taken as the unique reconstruction polynomial $\mathbf{P}_i$ and then used for all faces. In our calculations we omit this part of the characteristic-wise reconstruction in order to reduce the computational cost.

It is well known that the WENO reconstruction as applied to nonlinear systems may fail if the solution contains two discontinuities which are too close to each other. This is because the reconstruction procedure will not be able to find a smooth stencil and spurious oscillations will appear. As a result, the scheme may crash. To remedy this problem we adopt (with appropriate modifications for the present study) a modification of the reconstruction originally proposed in [30]for one-dimensional ENO schemes and later successfully extended to the three-dimensional finite-volume WENO methods on structured meshes. Essentially, we check if the reconstructed values of gas density and pressure differ too drastically from the cell averaged values and if this is the case, we locally reduce the order of the reconstruction polynomial. It can be shown that the use of the above procedure does not in any way degrade the high order of accuracy of the schemes for smooth solutions; see [71] for details.

3.3 Numerical flux

After the reconstruction is carried out, for each computational cell the point-wise values of the conserved vector $\mathbf{U}$ are represented by high-order reconstruction polynomials. Since these polynomials are different, at each Gaussian point β in the expression for the numerical flux (3.6) for the face A_j of cell V_i we have two approximate values for the conserved vector $\mathbf{U}$. The first value $\mathbf{U}_\beta^-$ corresponds to the spatial limit to the cell boundary from inside the cell V_i and is given by the reconstruction polynomial $\mathbf{P}_i$. The second value $\mathbf{U}_\beta^+$ corresponds to the spatial limit from outside the element and is obtained by using the reconstruction polynomial of the neighboring element $V_{i'}$. The values $\mathbf{U}_\beta^\pm$ are usually called left and right boundary extrapolated values. In upwind Godunov-type methods the resulting discontinuity as illustrated in Figure 3.1 on page 51 is resolved by replacing the physical flux at each Gaussian integration point by using a monotone function of left and right boundary extrapolated values so that (3.6) can be rewritten as

$$\mathbf{K}_{ij} \approx \sum_\beta \mathbf{F}_{n,j}\left(\mathbf{U}_\beta^-, \mathbf{U}_\beta^+\right) \omega_\beta |A_j| \tag{3.21}$$



(a) Piecewise Constant Discontinuous States (b) Piecewise Linear Discontinuous States (c) Arbitrary-Order Discontinuous States

Figure 3.1: Discontinuous Intercell States

The function $\tilde{\mathbf{F}}_{n,j}\left(\mathbf{U}_\beta^-, \mathbf{U}_\beta^+\right)$ is called the Riemann solver, or a building block of a high-order scheme.

Review of existing exact and approximate Riemann solvers for for various hyperbolic systems can be found in [20, 70]. In this thesis we use the HLLC Riemann solver. A detailed and up-to-date description can be found [70]. Using the concept of the rotational invariance [70] , for each face A_j and the local azimuthal angle θ and the polar angle φ we replace the normal projection of the flux tensor $\mathbf{F}_{n,j}$ by

$$\mathbf{F}_{n,j} = \mathbf{T}^{-1}\mathbf{F}\left(\mathbf{T}_j\mathbf{U}\right) \tag{3.22}$$

where $\mathbf{T}_j$ is the (constant) rotation matrix for face j. Then the expression (3.21) for $\mathbf{K}_{ij}$ is rewritten as

$$\mathbf{K}_{ij} = \sum_\beta \mathbf{F}_{n,j}\left(\mathbf{U}_\beta^-, \mathbf{U}_\beta^+\right)\omega_\beta|A_j| = \sum_\beta \mathbf{T}^{-1}\mathbf{F}\left(\hat{\mathbf{U}}_L, \hat{\mathbf{U}}_R\right)\omega_\beta|A_j| \tag{3.23}$$

where $\hat{\mathbf{U}}_j$ is the rotated conserved variable and

$$\hat{\mathbf{U}}_L = \mathbf{T}_j\mathbf{U}_\beta^-, \quad \hat{\mathbf{U}}_R = \mathbf{T}_j\mathbf{U}_\beta^+$$

It follows from (3.23) that the flux function for the Gaussian point β can be computed from the augmented one-dimensional Riemann problem

$$\frac{\partial}{\partial t}\hat{\mathbf{U}} + \frac{\partial}{\partial s}\hat{\mathbf{F}} = \mathbf{0}, \quad \hat{\mathbf{F}} = \mathbf{F}(\hat{\mathbf{U}}), \quad \hat{\mathbf{U}}(s,0) = \begin{cases} \hat{\mathbf{U}}_L, & s < 0, \\ \hat{\mathbf{U}}_R, & s > 0 \end{cases} \tag{3.24}$$

Assuming a three-wave structure with wave speed estimates S_L, S_* and S_R the HLLC flux is given by

$$\hat{\mathbf{F}}^{HLLC} = \begin{cases} \hat{\mathbf{F}}_L, & if \quad 0 \le S_L, \\ \hat{\mathbf{F}}_{*L} = \hat{\mathbf{F}}_L + S_L(\hat{\mathbf{U}}_{*L} - \hat{\mathbf{U}}_L), & if \quad S_L \le 0 \le S_*, \\ \hat{\mathbf{F}}_{*R} = \hat{\mathbf{F}}_R + S_R(\hat{\mathbf{U}}_{*R} - \hat{\mathbf{U}}_R), & if \quad S_* \le 0 \le S_R, \\ \hat{\mathbf{F}}_R, & if \quad 0 \ge S_R, \end{cases} \tag{3.25}$$

where

$$\hat{\mathbf{U}}_{*K} = \rho_K \left(\frac{S_K - u_K}{S_K - S_*} \right) \begin{bmatrix} 1 \\ S_* \\ v_K \\ w_K \\ \frac{E_K}{\rho_K}(S_* - u_K)[S_* + \frac{p_K}{\rho_K(S_K - u_K)}] \end{bmatrix}$$

for $K = L$ and $K = R$. The wave speeds S_L, S_* and S_R are estimated using the procedure for pressure-velocity estimates of [70].

We remark that HLLC flux contains all waves in the Riemann problem solution, does not use linearization of the equations and works well for low-density problems and sonic points without any fixes. The HLLC flux has been recently used in a number of very high-order methods, with good results, see e.g. [71, 68] .

3.4 Time Advancement

Having constructed the numerical fluxes $\mathbf{F}_{n,j}$as expressed in the semi-discrete conservative formulation (3.5) the next step involves the advancement of the solution in time. Depending on the spatial-order of accuracy of the scheme utilised we employ a time-advancement scheme of the same order of accuracy (up to 3^{rd}-order). Therefore the schemes used are the explicit forward-euler, the explicit 2^{nd}-order TVD Runge-Kutta and 3^{rd}-order TVD Runge-Kutta and the 4^{th}-order Runge-Kutta. For higher than third-order schemes matching time accuracy to space accuracy is limited due to the use Runge-Kutta schemes. To avoid spurious oscillations the Runge-Kutta schemes must be TVD and this leads to accuracy barriers[68] the accuracy of such methods cannot be higher than fifth. Moreover, fourth and fifth order methods are quite complicated and have reduced stability range.

$$\mathbf{U}^{n+1} = \mathbf{U}^n + \Delta t \cdot L(\mathbf{U}^n) \tag{3.26}$$

$$\left.\begin{aligned} \mathbf{U}^1 &= \mathbf{U}^n + \tfrac{\Delta t}{2} \cdot L(\mathbf{U}^n) \\ \mathbf{U}^{n+1} &= \mathbf{U}^n + \Delta t \cdot L\left(\mathbf{U}^1\right) \end{aligned}\right\} \tag{3.27}$$

$$\left.\begin{aligned} \mathbf{U}^1 &= \mathbf{U}^n + \Delta t \cdot L(\mathbf{U}^n) \\ \mathbf{U}^2 &= \tfrac{3}{4}\mathbf{U}^n + \tfrac{1}{4}\mathbf{U}^1 + \tfrac{\Delta t}{4} \cdot L\left(\mathbf{U}^1\right) \\ \mathbf{U}^{n+1} &= \tfrac{1}{3}\mathbf{U}^n + \tfrac{2}{3}\mathbf{U}^2 + \tfrac{2\Delta t}{3} \cdot L\left(\mathbf{U}^2\right) \end{aligned}\right\} \tag{3.28}$$

$$\left.\begin{aligned}
\mathbf{U}^1 &= \mathbf{U}^n + \tfrac{\Delta t}{2}\cdot L\left(\mathbf{U}^n\right)\\
\mathbf{U}^2 &= \mathbf{U}^n + \tfrac{\Delta t}{2}\cdot L\left(\mathbf{U}^1\right)\\
\mathbf{U}^3 &= \mathbf{U}^n + \Delta t\cdot L\left(\mathbf{U}^2\right)\\
\mathbf{U}^{n+1} &= \tfrac{1}{3}\left(-\mathbf{U}^n + \mathbf{U}^1 + 2\mathbf{U}^2 + \mathbf{U}^3\right) + \tfrac{\Delta t}{6}\cdot L\left(\mathbf{U}^3\right)
\end{aligned}\right\} \tag{3.29}$$

The time step Δt is selected according to the formula

$$\Delta t = K \min_i \frac{h_i}{S_i \cdot V_i} \tag{3.30}$$

where S_i is an estimate of the maximum propagation speed in cell V_i, $K \leq 1/3$ is the CFL number, h_i is the characteristic length of the element V_i. The maximum propagation speed in each cell is given by

$$S_i = sp_x \cdot n_x + sp_y \cdot n_y + sp_z \cdot n_z \tag{3.31}$$

with

$$sp_x = |u+a|, \quad , sp_y = |v+a|, \quad sp_x = |w+a|$$

where $\mathbf{n} = (n_x, n_y, n_z)$ is the outward unit normal vector and a is the speed of sound and the characteristics are running inwards to the domain. However for negative velocities the local maximum of the eigenvalues has to be taken.

Similarly to other approaches [49, 74, 43, 45] the characteristic length h_i of each element is taken to be the radius of the inscribed sphere of each element. We remark that although the semi-discrete scheme ((3.5),(3.26)-(3.29)) advances in time cell averages of the conserved quantities, the integrals of the flux functions over cell faces use point-wise values. The description of the scheme is complete once a reconstruction procedure to calculate the point-wise values from cell averages and a numerical flux (building block) of the scheme are specified.

For higher than 3^{rd}-order spatial accurate schemes however we employ the 3^{rd}-order TVD Runge-Kutta . For convergence studies and in cases where the temporal-order of accuracy is of crucial importance we enforce a time-step size smaller than the one imposed by the CFL condition. As documented by [42, 14, 62, 32] in order to match the spatial to the temporal order of accuracy we use a time-step size given by

$$\Delta t = K \cdot (\Delta x)^{\frac{n}{3}} \tag{3.32}$$

where n stands for the order of the scheme for $n > 3$.

3.5 Implementation of Boundary Conditions

The implementation of boundary conditions in the context of three-dimensional unstructured meshes is treated in a different manner from structured grids. Since the existence of valid fictitious cells (widely used for structured grids [20, 3]) can not be guaranteed a different approach is exploited. The types of boundary conditions encountered for the compressible Euler equations are the following:

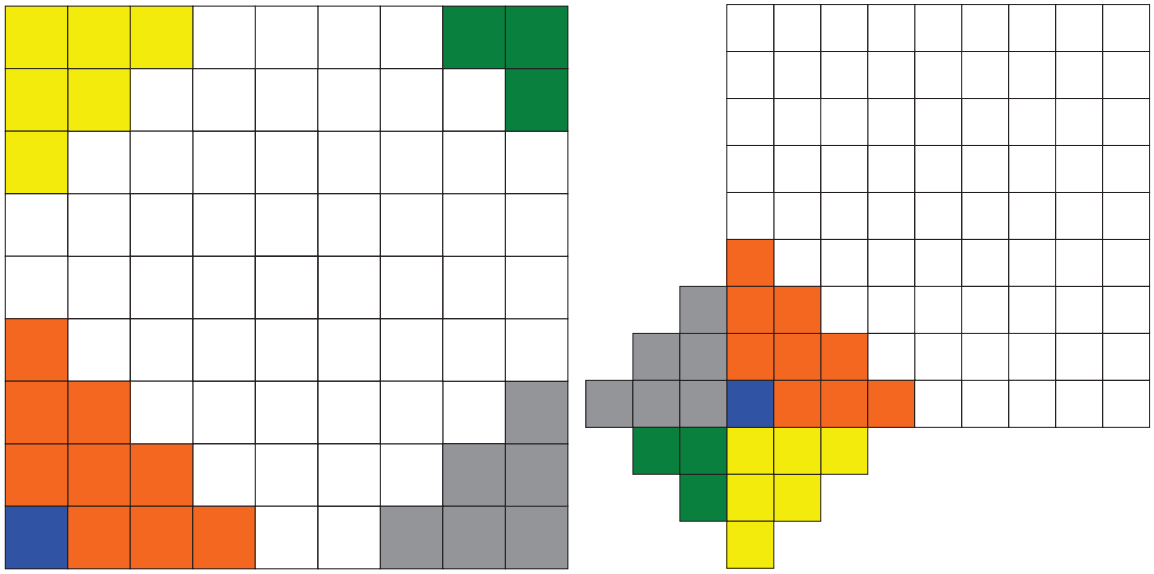
1. Inflow
2. Transmissive
3. Solid (Wall)
4. Periodic

The central and directional stencil selection algorithms remain unchanged in the presence of non-periodic boundary conditions and therefore the search algorithm is constrained within the boundaries of the computational domain; this results in one-sided central stencils. The total number of admissible directional stencils is reduced at the presence of boundaries. On the other hand for periodic boundary conditions the stencils selection algorithm is as follows

1. For every cell with a periodic boundary face $\partial\Omega$ find the corresponding periodic element inside the computational domain
2. Include the periodic element in the stencil selection algorithm
3. Recursively add the direct side neighbours (and the ones arising from the included periodic element) satisfying the appropriate conditions until the required number of elements in the stencil has been reached
4. Shift the coordinates of every element in the stencil that is periodic (as shown in the schematic in Figure 3.2 on page 55 where different directional stencils are colour coded)

For the computation of the numerical flux at the boundaries the inverse Riemann problem is solved by prescribing data outside the computational domain as [49, 22]. Consider the rotated left (inside the considered cell) and right (outside the considered cell) intercell conserved vector data states $\hat{\mathbf{U}}_L$ and $\hat{\mathbf{U}}_R$ with :

$$\hat{\mathbf{U}}_L = \begin{bmatrix} \rho \\ \rho\hat{u}_L \\ \rho\hat{v}_L \\ \rho\hat{w}_L \\ E \end{bmatrix}, \ \hat{\mathbf{U}}_R = \begin{bmatrix} \rho \\ \rho\hat{u}_R \\ \rho\hat{v}_R \\ \rho\hat{w}_R \\ E \end{bmatrix}$$



(a) Periodic Stencils Configuration (b) Shifted Periodic Stencils Configuration

Figure 3.2: Stencil Selection at the Presence of Periodic Boundaries

$$\hat{\mathbf{U}}_R = \hat{\mathbf{U}}_L \quad for\, transmissive\, boundary \tag{3.33}$$

$$\hat{\mathbf{U}}_R = \begin{bmatrix} \rho \\ -\rho\hat{u}_L \\ \rho\hat{v}_L \\ \rho\hat{w}_L \\ E \end{bmatrix}, \quad for\, no-slip(solid\, wall)\, boundary \tag{3.34}$$

$$\hat{\mathbf{U}}_R = \begin{bmatrix} \rho_{inflow} \\ \rho_{inflow}\hat{u}_{inflow} \\ \rho_{inflow}\hat{v}_{inflow} \\ \rho_{inflow}\hat{w}_{inflow} \\ E_{inflow} \end{bmatrix}, \quad for\, inflow\, boundary \tag{3.35}$$

Where the subscript $inflow$ declares that these values are prescribed to be equal to the ones specified at the boundary and corresponds to supersonic inflow boundary condition since all characteristics run inwards to the domain and need to specified. For subsonic inflow however the far-field "inflow" boundary conditions the fixed and extrapolated Riemann invariants corresponding to the incoming and outgoing waves traveling in characteristic directions defined normal to the boundary must be used since there are also characteristics running outwards of the domain. We remark that when periodic boundary conditions are used in the context of unstructured meshes the surface meshes that are periodic must be exactly the same. This will ensure that a for each element that has a periodic boundary surface a periodic element exists.

Chapter 4

3D Applications

Introduction

In this section we present numerical results of our schemes up to 5^{th}-order of accuracy as applied to linear and non-linear hyperbolic conservation laws with both smooth and discontinuous solutions in three space dimensions. In all our computations the multidimensional Gaussian quadrature rule used for the approximation of surface and volume integrals has twice the order of accuracy of the numerical scheme. For the Euler equations the HLLC approximate Riemann solver is employed.

Below we denote the schemes of order r as WENO$-r$, e.g. the spatially 5^{th}-order scheme is denoted as WENO-5. The 3^{rd}-order explicit TVD Runge-Kutta is employed for all the numerical schemes except from the 1^{st}-order Godunov scheme where 1^{st}-order Forward Euler is used. For all the numerical schemes up to 3^{rd}-order of accuracy we run all convergence tests with a fixed Courant number, which is chosen to be $C_{cfl} = 0.3$ since we are using an unsplit finite volume with a stability condition that requires that CFL should be less than 1/3. However for higher-order numerical schemes the time-step size is reduced as defined in (3.32) so that the spatial-order of accuracy dominates the computation.

The results illustrate that our schemes can compute discontinuous solutions without oscillations and at the same time maintain the designed very high order of accuracy in multiple space dimensions.

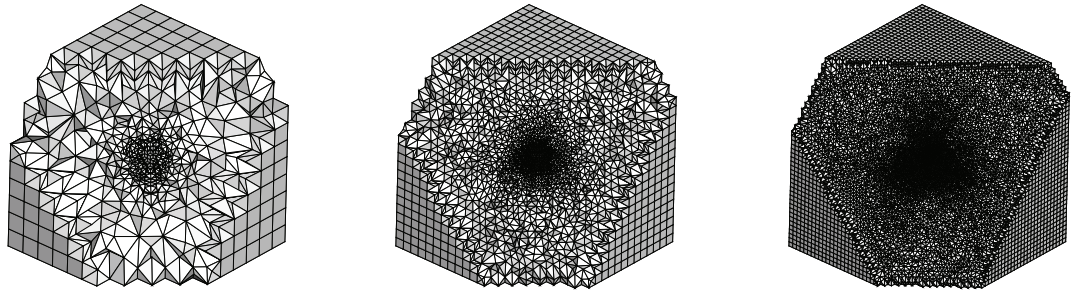
4.1 3D Linear Advection Equation

4.1.1 Smooth Solution Test Case

We solve

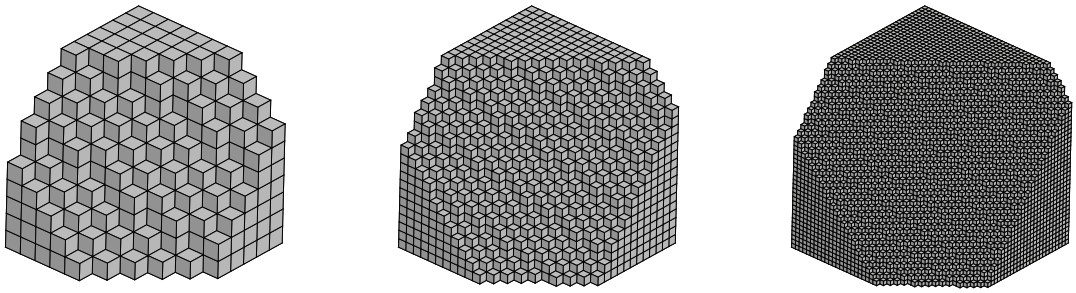
$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} + \frac{\partial u}{\partial z} = 0 \tag{4.1}$$

with a smooth initial condition



(a) N=10 (b) N=20 (c) N=40

Figure 4.1: Sequence of hybrid meshes (cutaway sections) used for convergence study of the model equation (4.1) with initial conditions (4.2)



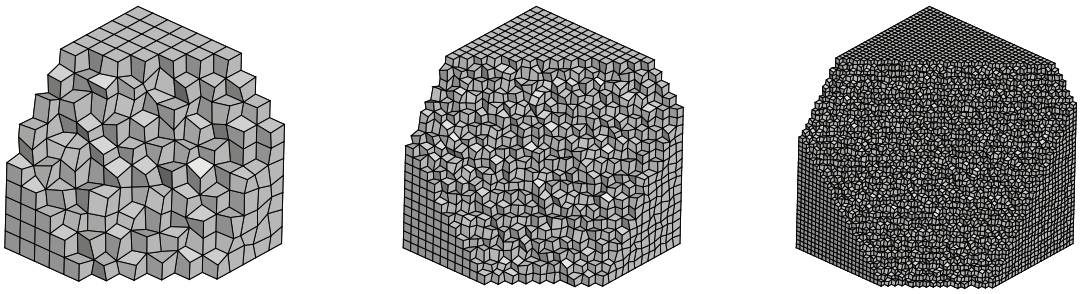
(a) N=10 (b) N=20 (c) N=40

Figure 4.2: Sequence of uniform hexahedral meshes (cutaway sections) used for convergence study of the model equation (4.1) with initial conditions (4.2)

$$u_0(x,y,z) = sin\,(2\pi x) \cdot sin(2\pi y) \cdot sin(2\pi z) \tag{4.2}$$

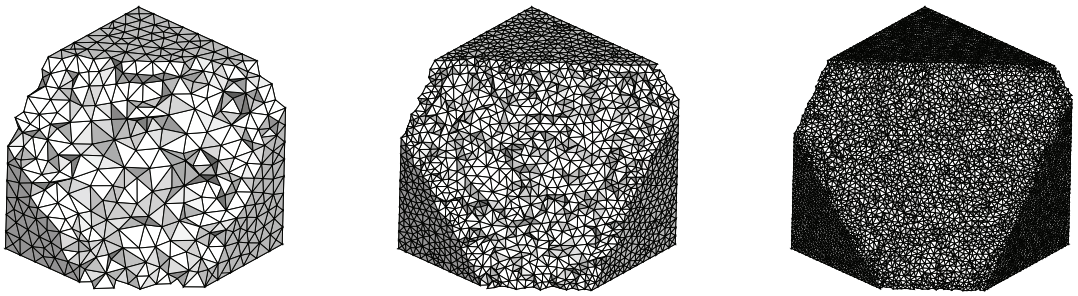
The computational domain is a cube $[0,1]^3$. Periodic boundary conditions are used. The error is measured at time $t = 1$. The domain is meshed by five types of unstructured meshes. Figure 4.1 on page 57 to Figure 4.5 on page 58 provide a cut view of the meshes used for $N = 10, 20, 40$ where N specifies the number of cells over each edge of the cube. Then, the interior is meshed with hexahedral, tetrahedral, prismatic and pyramidal elements or any combination of them. The total number of cells is then denoted as N_{tot}.

Table 4.1 on page 59 to Table 4.5 on page 63 show convergence rates and errors for a sequence of meshes for linear and non-linear schemes, used in calculations. We observe that the schemes reach the designed order of accuracy. It must be noted that for this test case where periodic boundary conditions the coarsest hexahedral and prismatic meshes $N = 10$ are not sufficient (N/S) to employ LINEAR-5 and WENO-5 schemes since the stencils extend more than half a period in length, leading to wrong results.



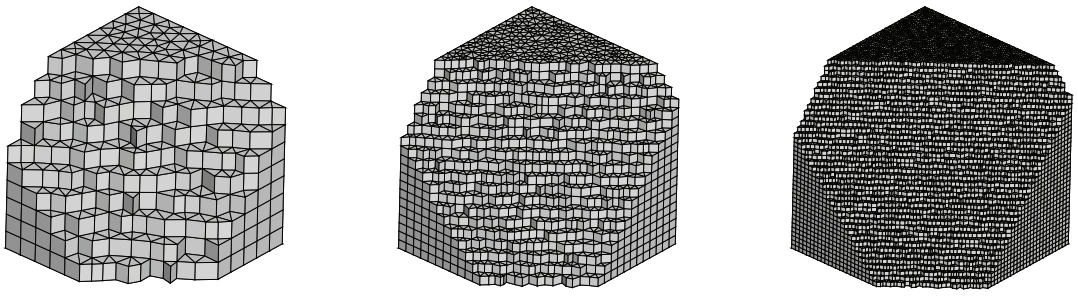
(a) N=10 (b) N=20 (c) N=40

Figure 4.3: Sequence of unstructured hexahedral meshes (cutaway sections) used for convergence study of the model equation (4.1) with initial conditions (4.2)



(a) N=10 (b) N=20 (c) N=40

Figure 4.4: Sequence of tetrahedral meshes (cutaway sections) used for convergence study of the model equation (4.1) with initial conditions (4.2)



(a) N=10 (b) N=20 (c) N=40

Figure 4.5: Sequence of prismatic meshes (cutaway sections) used for convergence study of the model equation (4.1) with initial conditions (4.2)

Method	N	L_1 error	L_1 order	L_∞ error	L_∞ order
1st-Order	20	1.30×10^{-1}	1.026	2.10×10^{-1}	1.013
	40	6.54×10^{-2}	0.991	9.89×10^{-2}	1.086
	80	3.33×10^{-2}	0.986	5.20×10^{-2}	0.927
Linear-2	20	4.45×10^{-2}	1.724	7.61×10^{-2}	1.843
	40	9.89×10^{-3}	2.169	2.33×10^{-2}	1.705
	80	2.61×10^{-3}	1.921	5.89×10^{-3}	1.983
Linear-3	20	1.20×10^{-2}	2.762	5.73×10^{-2}	2.477
	40	1.61×10^{-3}	2.897	8.32×10^{-3}	2.783
	80	1.98×10^{-4}	3.041	1.11×10^{-3}	2.906
Linear-4	20	8.4×10^{-3}	3.287	2.6×10^{-2}	3.491
	40	5.3×10^{-4}	3.986	2.1×10^{-3}	3.653
	80	3.21×10^{-5}	4.045	1.45×10^{-4}	3.856
Linear-5	20	1.2×10^{-3}	N/S	2.12×10^{-3}	N/S
	40	4.2×10^{-5}	4.836	7.4×10^{-5}	4.840
	80	1.53×10^{-6}	4.778	2.41×10^{-6}	4.948
TVD-2	20	7.7×10^{-2}	1.425	4.0×10^{-1}	0.997
	40	1.7×10^{-2}	2.179	1.9×10^{-1}	1.074
	80	4.34×10^{-3}	1.969	9.45×10^{-2}	1.007
WENO-2	20	5.74×10^{-2}	1.931	8.59×10^{-2}	1.872
	40	1.52×10^{-2}	1.916	2.18×10^{-2}	1.978
	80	4.01×10^{-3}	1.922	5.31×10^{-3}	2.037
WENO-3	20	1.4×10^{-2}	2.535	5.1×10^{-2}	2.622
	40	2.1×10^{-3}	2.722	7.3×10^{-3}	2.796
	80	2.65×10^{-4}	2.986	1.03×10^{-3}	2.825
WENO-4	20	2.4×10^{-3}	3.816	9.3×10^{-3}	3.599
	40	2.1×10^{-4}	3.610	5.9×10^{-4}	3.978
	80	1.42×10^{-5}	3.886	3.56×10^{-5}	4.050
WENO-5	20	9.56×10^{-4}	N/S	4.30×10^{-3}	N/S
	40	3.52×10^{-5}	4.763	1.56×10^{-4}	4.784
	80	1.21×10^{-6}	4.862	4.89×10^{-6}	4.995

Table 4.1: Convergence study for various schemes using a uniform hexahedral mesh as applied to the model equation (4.1) with initial conditions (4.2) at output time $t = 1.0$.

Method	N	L_1 error	L_1 order	L_∞ error	L_∞ order
1st-Order	20	2.20×10^{-1}	0.937	3.59×10^{-1}	1.123
	40	1.21×10^{-1}	0.874	2.09×10^{-1}	0.784
	80	6.54×10^{-2}	0.875	1.06×10^{-1}	0.979
Linear-2	20	4.67×10^{-2}	1.839	8.69×10^{-2}	1.732
	40	1.23×10^{-2}	1.924	2.41×10^{-2}	1.851
	80	3.54×10^{-3}	1.796	6.98×10^{-3}	1.787
Linear-3	20	1.39×10^{-2}	2.857	4.66×10^{-2}	2.962
	40	1.59×10^{-3}	3.127	5.55×10^{-3}	3.059
	80	1.89×10^{-4}	3.072	7.67×10^{-4}	2.865
Linear-4	20	6.77×10^{-3}	3.943	1.89×10^{-2}	3.842
	40	4.21×10^{-4}	4.007	1.87×10^{-3}	3.344
	80	2.98×10^{-5}	3.820	1.22×10^{-4}	3.938
Linear-5	20	8.56×10^{-4}	N/S	2.09×10^{-3}	N/S
	40	3.34×10^{-5}	4.679	6.89×10^{-5}	4.920
	80	1.08×10^{-6}	4.950	1.95×10^{-6}	5.145
TVD-2	20	8.2×10^{-2}	1.821	4.6×10^{-1}	0.912
	40	2.4×10^{-2}	1.772	2.2×10^{-1}	1.064
	80	6.52×10^{-3}	1.880	1.03×10^{-1}	1.094
WENO-2	20	6.16×10^{-2}	1.907	9.02×10^{-2}	1.693
	40	2.25×10^{-2}	1.453	2.66×10^{-2}	1.761
	80	5.67×10^{-3}	1.988	6.79×10^{-3}	1.969
WENO-3	20	9.43×10^{-3}	2.579	3.67×10^{-2}	2.622
	40	1.55×10^{-3}	2.604	5.88×10^{-3}	2.641
	80	1.96×10^{-4}	2.983	8.23×10^{-4}	2.836
WENO-4	20	8.56×10^{-4}	3.977	7.33×10^{-3}	3.874
	40	6.97×10^{-5}	3.618	4.12×10^{-4}	4.153
	80	4.52×10^{-6}	3.946	2.77×10^{-5}	3.894
WENO-5	20	7.73×10^{-4}	N/S	3.47×10^{-3}	N/S
	40	2.69×10^{-5}	4.844	1.27×10^{-4}	4.772
	80	1.03×10^{-6}	4.706	4.23×10^{-6}	4.908

Table 4.2: Convergence study for various schemes using a unstructured hexahedral mesh as applied to the model equation (4.1) with initial conditions (4.2) at output time $t = 1.0$.

Method	N	L_1 error	L_1 order	L_∞ error	L_∞ order
1st-Order	20	3.90×10^{-1}	0.679	4.80×10^{-1}	0.754
	40	1.97×10^{-1}	0.985	2.31×10^{-1}	1.055
	80	1.05×10^{-1}	0.907	1.22×10^{-1}	0.921
Linear-2	20	3.77×10^{-2}	1.887	7.48×10^{-2}	1.905
	40	8.54×10^{-3}	2.142	2.28×10^{-2}	1.714
	80	2.29×10^{-3}	1.898	5.96×10^{-3}	1.935
Linear-3	20	9.85×10^{-2}	2.844	3.983×10^{-2}	2.958
	40	9.73×10^{-4}	3.339	6.47×10^{-3}	2.620
	80	1.29×10^{-4}	2.915	7.26×10^{-4}	3.155
Linear-4	20	4.51×10^{-3}	3.911	1.27×10^{-2}	3.632
	40	3.62×10^{-4}	3.639	9.24×10^{-4}	3.780
	80	2.47×10^{-5}	3.873	6.56×10^{-5}	3.816
Linear-5	20	7.54×10^{-4}	N/S	1.53×10^{-3}	N/S
	40	2.57×10^{-5}	4.874	5.39×10^{-5}	4.828
	80	9.38×10^{-7}	4.776	1.79×10^{-6}	4.912
TVD-2	20	9.3×10^{-2}	1.966	4.78×10^{-1}	0.798
	40	2.54×10^{-2}	1.872	2.64×10^{-1}	0.856
	80	7.51×10^{-3}	1.757	1.31×10^{-1}	1.010
WENO-2	20	5.21×10^{-2}	1.741	7.27×10^{-2}	1.933
	40	1.48×10^{-2}	1.815	1.79×10^{-2}	2.022
	80	4.29×10^{-3}	1.786	5.02×10^{-3}	1.831
WENO-3	20	6.93×10^{-3}	2.671	2.49×10^{-2}	2.542
	40	9.27×10^{-4}	2.902	4.73×10^{-3}	2.396
	80	1.41×10^{-4}	2.716	6.77×10^{-4}	2.804
WENO-4	20	5.87×10^{-4}	3.994	6.97×10^{-3}	3.309
	40	4.68×10^{-5}	3.648	6.42×10^{-4}	3.440
	80	3.05×10^{-6}	3.939	3.91×10^{-5}	4.035
WENO-5	20	6.99×10^{-4}	N/S	1.96×10^{-3}	N/S
	40	2.36×10^{-5}	4.888	8.55×10^{-5}	4.518
	80	9.71×10^{-7}	4.603	2.69×10^{-6}	4.990

Table 4.3: Convergence study for various schemes using a unstructured prismatic mesh as applied to the model equation (4.1) with initial conditions (4.2) at output time $t = 1.0$.

Method	N	L_1 error	L_1 order	L_∞ error	L_∞ order
1st-Order	20	2.39×10^{-1}	0.621	3.30×10^{-1}	0.579
	40	1.36×10^{-1}	0.819	1.45×10^{-1}	1.186
	80	6.65×10^{-2}	1.032	9.58×10^{-2}	0.958
Linear-2	20	1.29×10^{-2}	1.689	4.75×10^{-2}	1.733
	40	3.27×10^{-3}	1.980	1.23×10^{-2}	1.949
	80	8.69×10^{-4}	1.911	2.85×10^{-3}	2.109
Linear-3	20	4.32×10^{-3}	2.942	1.66×10^{-2}	2.746
	40	5.96×10^{-4}	2.857	2.38×10^{-3}	2.802
	80	7.12×10^{-5}	3.065	2.91×10^{-4}	3.031
Linear-4	20	1.09×10^{-3}	3.843	9.61×10^{-3}	3.961
	40	8.22×10^{-5}	3.729	5.94×10^{-4}	4.016
	80	5.39×10^{-6}	3.930	3.66×10^{-5}	4.020
Linear-5	20	1.93×10^{-4}	4.991	7.56×10^{-4}	4.322
	40	8.67×10^{-6}	4.476	2.72×10^{-5}	4.796
	80	2.99×10^{-7}	4.853	1.06×10^{-6}	4.681
TVD-2	20	6.85×10^{-2}	1.779	3.66×10^{-1}	0.874
	40	1.67×10^{-2}	2.036	1.79×10^{-1}	1.031
	80	3.98×10^{-3}	2.069	8.94×10^{-2}	1.001
WENO-2	20	1.87×10^{-2}	1.832	3.77×10^{-2}	1.643
	40	5.67×10^{-3}	1.721	7.92×10^{-3}	1.978
	80	1.36×10^{-3}	2.059	2.16×10^{-3}	1.874
WENO-3	20	1.96×10^{-3}	2.899	8.67×10^{-2}	2.541
	40	2.54×10^{-4}	2.947	1.03×10^{-3}	3.073
	80	3.75×10^{-5}	2.759	1.36×10^{-4}	2.918
WENO-4	20	2.97×10^{-4}	3.923	3.69×10^{-3}	3.772
	40	2.11×10^{-5}	3.815	2.29×10^{-4}	4.010
	80	1.23×10^{-6}	4.100	1.43×10^{-5}	4.012
WENO-5	20	3.72×10^{-4}	4.712	8.63×10^{-4}	4.963
	40	1.69×10^{-5}	4.460	2.97×10^{-5}	4.860
	80	5.33×10^{-7}	4.986	8.94×10^{-7}	5.053

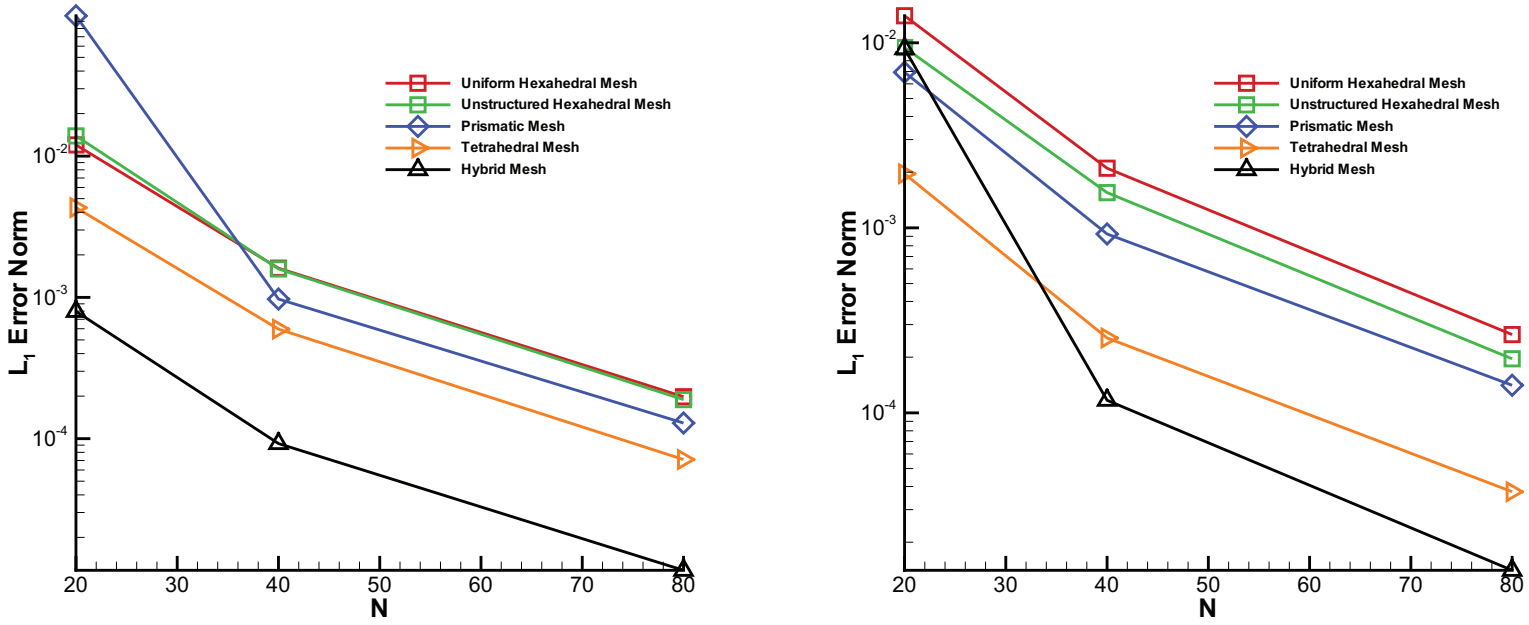
Table 4.4: Convergence study for various schemes using a unstructured tetrahedral mesh as applied to the model equation (4.1) with initial conditions (4.2) at output time $t = 1.0$.

Method	N	L_1 error	L_1 order	L_∞ error	L_∞ order
1st-Order	20	3.54×10^{-1}	0.759	4.98×10^{-1}	0.598
	40	1.69×10^{-2}	1.066	2.53×10^{-1}	0.977
	80	8.23×10^{-2}	1.038	1.33×10^{-1}	0.927
Linear-2	20	6.87×10^{-3}	1.892	1.44×10^{-2}	1.963
	40	1.66×10^{-3}	2.049	2.97×10^{-3}	2.277
	80	4.21×10^{-4}	1.979	6.98×10^{-4}	2.089
Linear-3	20	7.96×10^{-4}	2.892	5.78×10^{-3}	2.642
	40	9.23×10^{-5}	3.108	6.99×10^{-4}	3.047
	80	1.17×10^{-5}	2.979	8.32×10^{-5}	3.070
Linear-4	20	7.82×10^{-4}	3.968	2.11×10^{-3}	3.667
	40	4.91×10^{-5}	3.990	1.33×10^{-4}	3.982
	80	3.23×10^{-6}	3.928	8.57×10^{-6}	3.961
Linear-5	20	4.85×10^{-5}	4.692	2.61×10^{-4}	4.518
	40	1.44×10^{-6}	5.073	8.67×10^{-6}	4.911
	80	4.57×10^{-8}	4.977	2.55×10^{-7}	5.087
TVD-2	20	5.34×10^{-2}	1.563	4.11×10^{-1}	0.732
	40	1.31×10^{-2}	2.130	2.22×10^{-1}	0.888
	80	3.28×10^{-3}	2.038	1.09×10^{-1}	1.026
WENO-2	20	1.02×10^{-2}	1.875	6.33×10^{-2}	1.821
	40	2.33×10^{-3}	2.130	1.56×10^{-2}	2.020
	80	5.67×10^{-4}	2.038	4.07×10^{-3}	1.938
WENO-3	20	9.29×10^{-3}	2.638	6.75×10^{-3}	2.778
	40	1.17×10^{-4}	2.990	7.63×10^{-4}	3.145
	80	1.41×10^{-5}	3.050	9.41×10^{-5}	3.019
WENO-4	20	8.57×10^{-4}	3.529	4.29×10^{-3}	3.617
	40	4.96×10^{-5}	4.110	2.76×10^{-4}	3.958
	80	2.94×10^{-6}	4.073	1.75×10^{-5}	3.979
WENO-5	20	5.62×10^{-5}	4.887	3.43×10^{-4}	4.925
	40	2.35×10^{-6}	4.578	1.08×10^{-5}	4.784
	80	7.56×10^{-8}	4.959	3.62×10^{-7}	4.898

Table 4.5: Convergence study for various schemes using a hybrid unstructured mesh as applied to the model equation (4.1) with initial conditions (4.2) at output time $t = 1.0$.

A comparison between the five different types of meshes for the 3^{rd}-order schemes with a fixed CFL is shown in Figure 4.6 on page 65. It is clear at this stage that for the same mesh resolution N, the hybrid and tetrahedral meshes are superior to structured and unstructured hexahedral meshes in terms of L_1 and L_∞ error norms. This is justified by the fact that the reconstruction stencils of hybrid and tetrahedral meshes for the same order of accuracy are more compact since for the same mesh resolution more elements exist and therefore the reconstruction process is more accurate. We must remark that hybrid and tetrahedral meshes for the same resolution in terms of N are 4 to 14 times more expensive in terms of computing resources than the hexahedral meshes which has a great impact on the total simulation time. Another comparison between the five different types of meshes shown in Figure 4.7 on page 66 for the same scheme reveals that for the same number of total elements N_{tot} the hexahedral meshes produces more accurate results than any other unstructured mesh.This is justified by the fact that the time-step size of hexahedral elements is greater for a fixed CFL and by the fact that hexahedral elements have more nodes and more faces than any of the other elements used. In Table 4.6 on page 67 the statistics of each mesh used for this test problem are illustrated. Figure 4.8 on page 68 to Figure 4.12 on page 72 illustrate slices of the isolines of the solution for 2nd-order and 5th-order linear and WENO schemes for all the types of meshes used for $N = 20$. It is clear that the 5th-order linear and WENO schemes provide superior results compared to their 2nd-order counterparts.

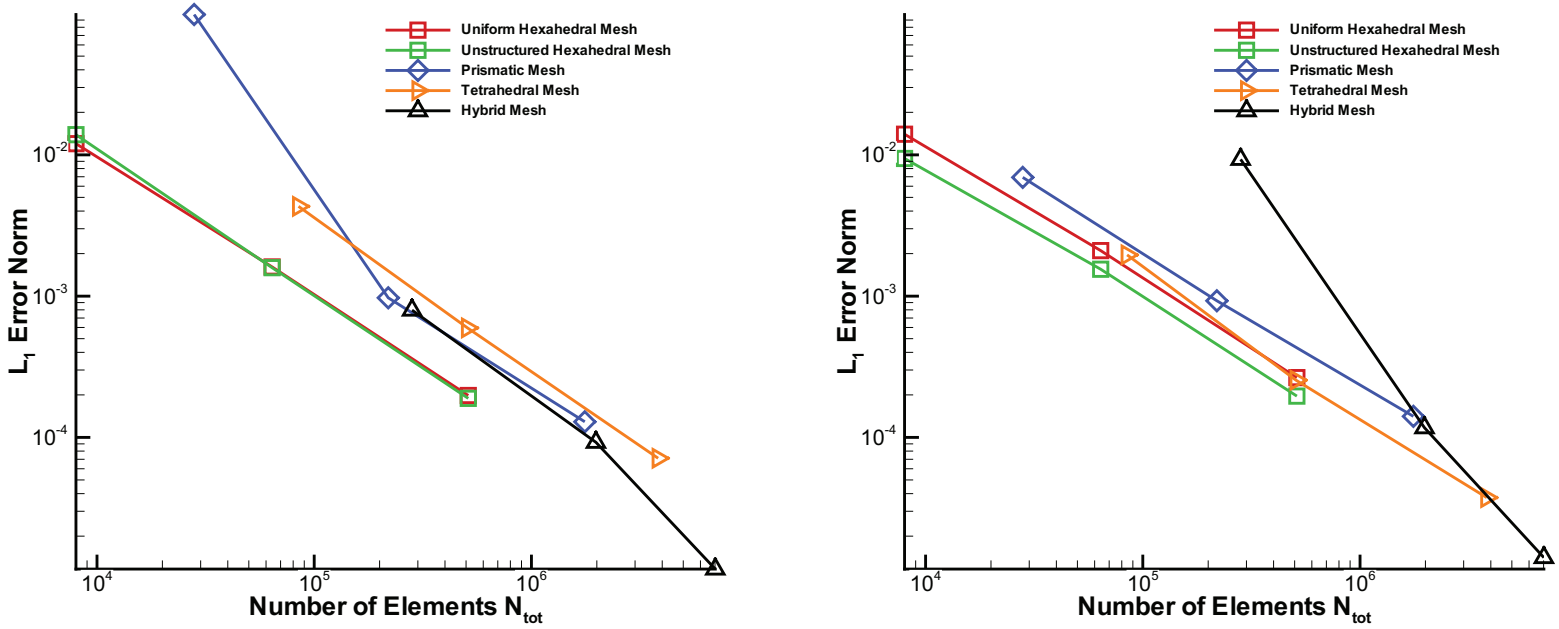
Finally we must note that the 1st-order Godunov type of method produces worse results for unstructured meshes than structured ones. This leads to the conclusion that unstructured meshes have greater need for higher-order schemes than their structured counterparts; since the complexity of the variation of spatial information must be treated with highly sophisticated techniques that only the higher-order reconstruction schemes offer. The main purpose of increasing the spatial resolution is to demonstrate the convergence rates of the high-order schemes. It should be noted that although high-order schemes will provide more accurate solution on coarser meshes compared to lower-order methods they require at least such a mesh resolution that the stencils do not extend more than a period in length for these type of problems.



(a) LINEAR-3

(b) WENO-3

Figure 4.6: Comparison of the performance of various schemes for all types of meshes used in terms of L_1error norm and mesh resolution N



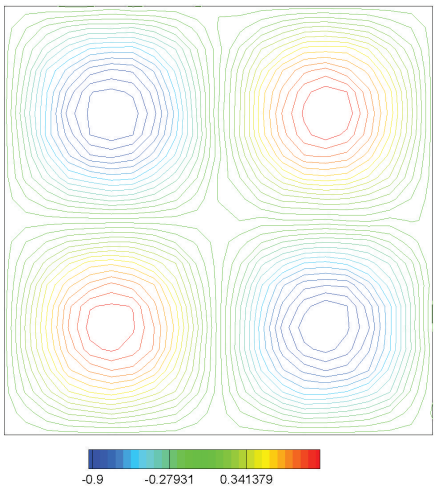
(a) LINEAR-3

(b) WENO-3

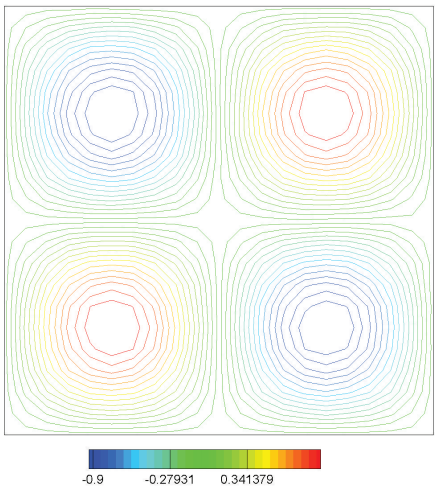
Figure 4.7: Comparison of the performance of various schemes for all types of meshes used in terms L_1error norm and total number of elements N_{tot}

Mesh Type	N	Tetrahedrals	Pyramids	Prisms	Hexahedrals	N_{tot}
Uniform Hexahedral	20	0	0	0	8000	8000
	40	0	0	0	64000	64000
	80	0	0	0	512000	512000
Hexahedral	20	0	0	0	8000	8000
	40	0	0	0	64000	64000
	80	0	0	0	512000	512000
Tetrahedral	20	84669	0	0	0	84669
	40	507337	0	0	0	507337
	80	3830397	0	0	0	3830397
Prismatic	20	0	0	28000	0	28000
	40	0	0	219200	0	219200
	80	0	0	1760480	0	1760480
Hybrid	20	218016	40000	16000	8000	282016
	40	1629213	160000	128000	64000	1981213
	80	4844598	640000	1024000	512000	7020598

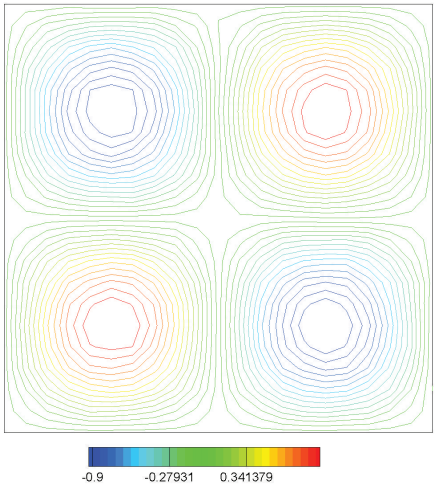
Table 4.6: Statistics for the meshes used for the computations of the model equation (4.1) with initial conditions (4.2)



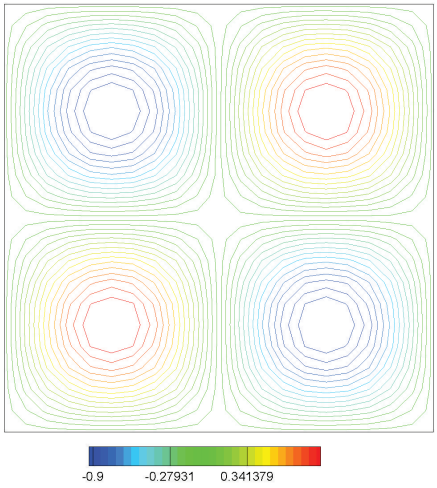
(a) LINEAR-2



(b) LINEAR-5

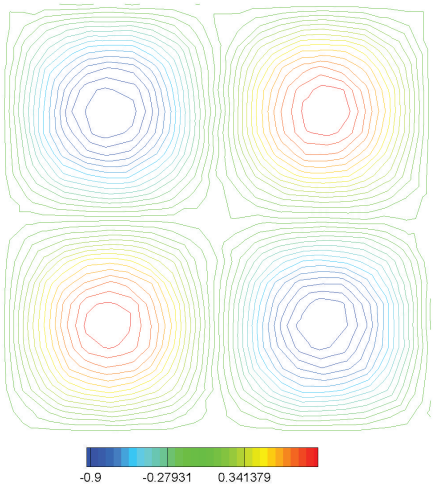


(c) WENO-2

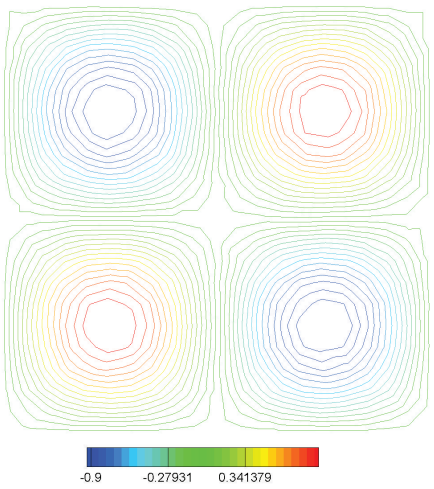


(d) WENO-5

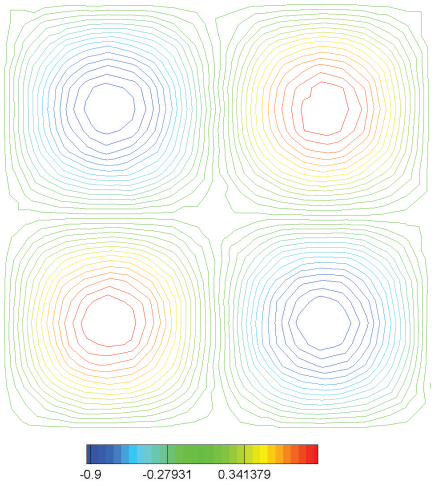
Figure 4.8: Slice of isolines of solution for various schemes on uniform hexahedral mesh at position $z = 0.25$ as applied to model equation (4.1) with initial conditions (4.2) at time $t = 1.0$.



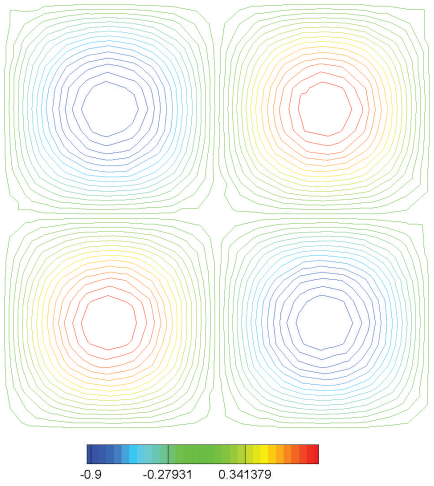
(a) LINEAR-2



(b) LINEAR-5

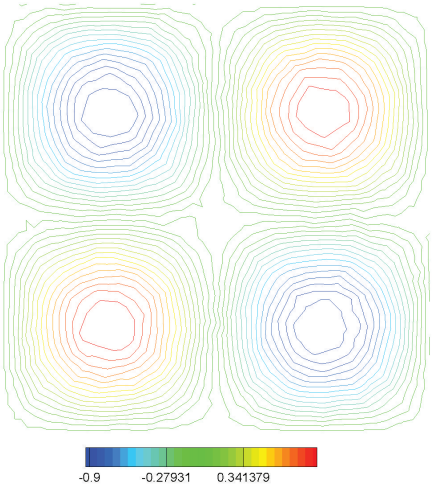


(c) WENO-2

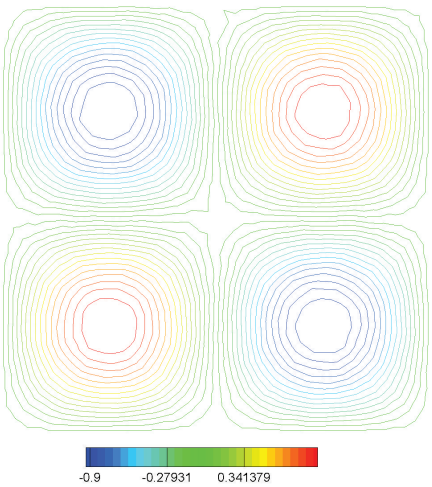


(d) WENO-5

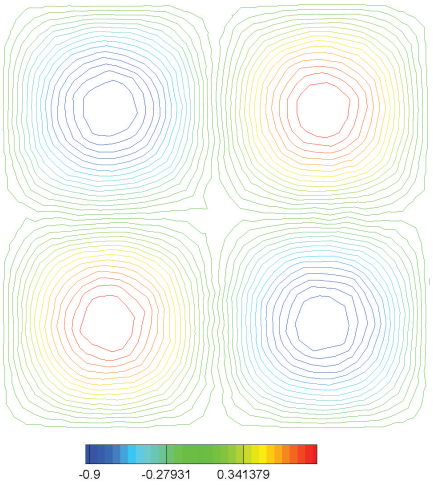
Figure 4.9: Slice of isolines of solution for various schemes on unstructured hexahedral mesh at position $z = 0.25$ as applied to model equation (4.1) with initial conditions (4.2) at time $t = 1.0$.



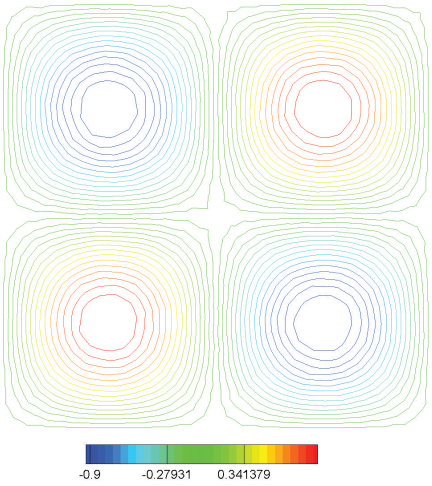
(a) LINEAR-2



(b) LINEAR-5

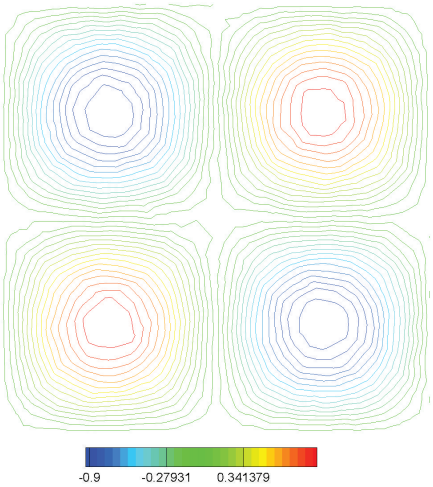


(c) WENO-2

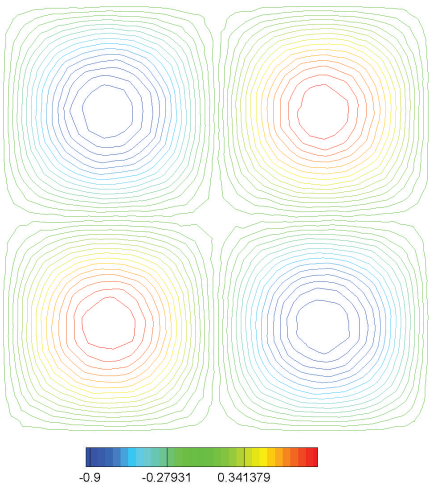


(d) WENO-5

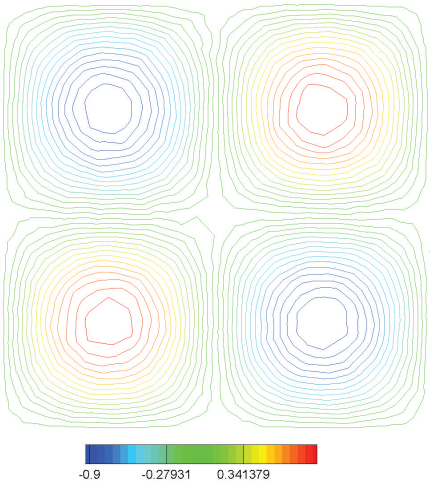
Figure 4.10: Slice of isolines of solution for various schemes on prismatic mesh at position $z = 0.25$ as applied to model equation (4.1) with initial conditions (4.2) at time $t = 1.0$.



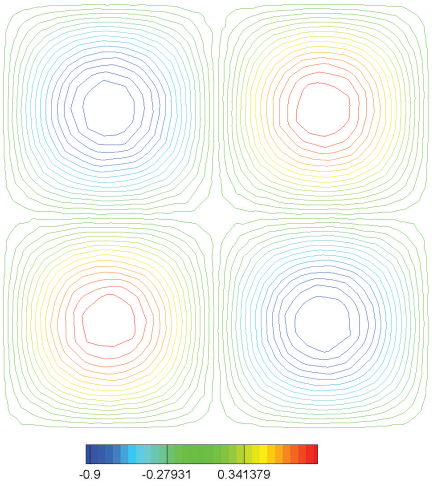
(a) LINEAR-2



(b) LINEAR-5

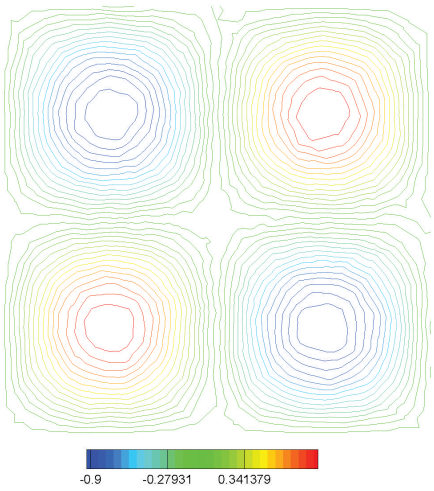


(c) WENO-2

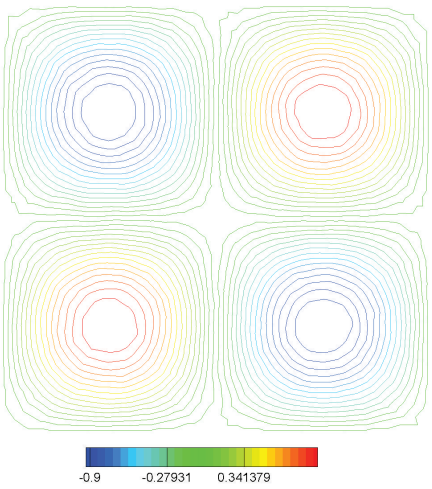


(d) WENO-5

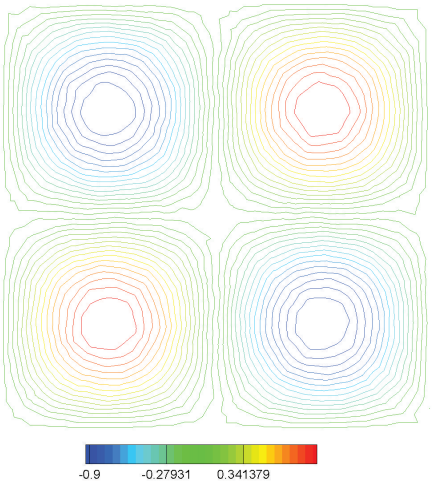
Figure 4.11: Slice of isolines of solution for various schemes on uniform tetrahedral mesh at position $z = 0.25$ as applied to model equation (4.1) with initial conditions (4.2) at time $t = 1.0$.



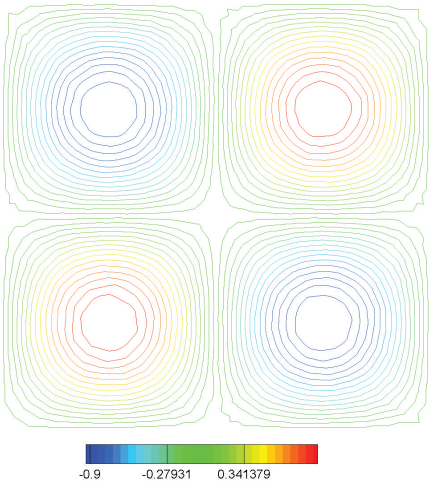
(a) LINEAR-2



(b) LINEAR-5



(c) WENO-2



(d) WENO-5

Figure 4.12: Slice of isolines of solution for various schemes on hybrid mesh at position $z = 0.25$ as applied to model equation (4.1) with initial conditions (4.2) at time $t = 1.0$.

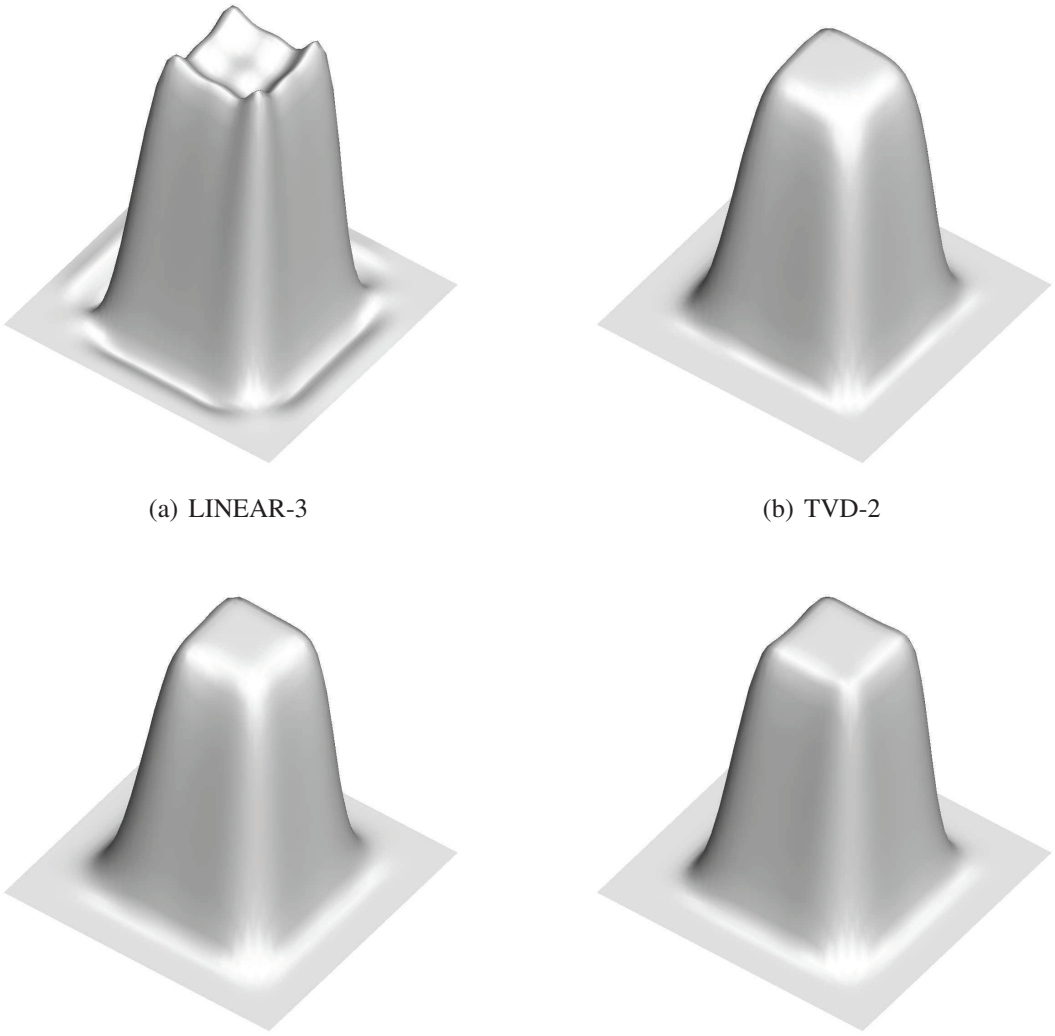
4.1.2 Discontinuous Solution Test Case

We solve the constant coefficient equation (4.1) with a discontinuous initial solution

$$u_0(x,y,z) = \begin{cases} 1, & if\ 0.25 \leq x, z \leq 0.75 \\ 0, & otherwise \end{cases} \tag{4.3}$$

The computational domain is a cube $[0,1]^3$. Periodic boundary conditions are used. The domain is meshed by five types of unstructured meshes as for the smooth test case (4.2) and the solution is computed at time $t = 1$. The computed solution from various schemes on a uniform hexahedral mesh ($N = 50$) is illustrated in Figure 4.13 on page 74. We observe that the non-linear schemes produce non-oscillatory solutions and the higher-order WENO schemes produce a much sharper resolution of the discontinuity.

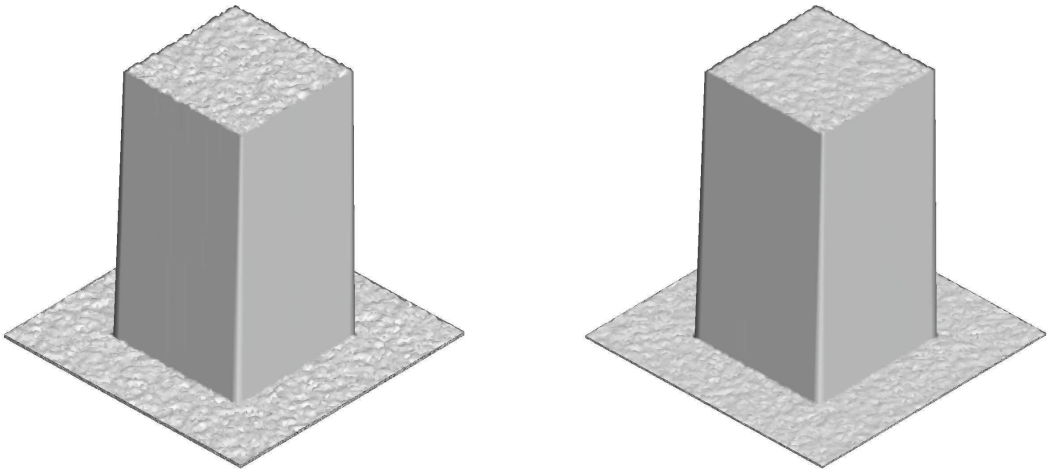
Figure 4.14 on page 75 show the effect of the central stencil linear weight d_0 where $A = d_0 = 10^5$ and $B = d_0 = 10^3$. It is noticed that with the central stencil linear weight A some slight oscillations arise in the solution profile. It has been documented [21] that for smooth solution profiles the central stencil weight should takes values in the top range $\sim 10^5$ where for strong discontinuous problems should be takes values of the lower range $\sim 10^2$. The central stencil linear weight A, B has an influence on the contribution of the central-linear reconstruction on the total reconstruction process. In smooth regions of the flow the central linear reconstruction process would be the most accurate one where in discontinuous regions of the flow the it would produce spurious oscillations.



(a) LINEAR-3 (b) TVD-2

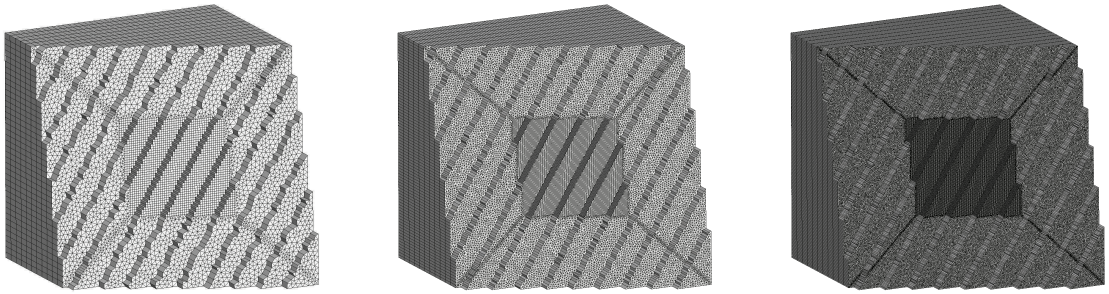
(c) WENO-2 (d) WENO-3

Figure 4.13: Computed solution for various schemes on a uniform hexahedral mesh $N = 50$ as applied to model equation (4.1) with initial conditions (4.3) at time $t = 1.0$.



(a) WENO-3 Central Stencil Linear Weight A (b) WENO-3 Central Stencil Linear Weight A

Figure 4.14: Solution profile for WENO-3 on hybrid mesh $N = 80$ as applied to model equation (4.1) with initial conditions (4.3) at time $t = 1.0$.



(a) 40x40x20 (b) 80x80x20 (c) 160x160x20

Figure 4.15: Sequence of hybrid meshes (cutaway sections) used for convergence study of the model equation (4.4) with initial conditions (4.5)

4.2 3D Euler Equations

4.2.1 Vortex Evolution

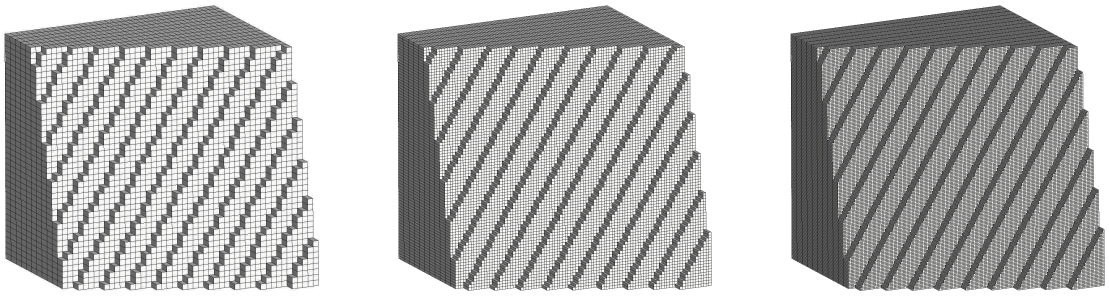
We solve the three-dimensional Euler equations

$$\frac{\partial}{\partial t}\mathbf{U}+\frac{\partial}{\partial x}\mathbf{F}(\mathbf{U})+\frac{\partial}{\partial y}\mathbf{G}(\mathbf{U})+\frac{\partial}{\partial z}\mathbf{H}(\mathbf{U})=\mathbf{0} \tag{4.4}$$

defined on $[0,10]\times[0,10]\times[0,0.5]$with periodic boundary conditions. The initial condition corresponds to a smooth two-dimensional vortex placed at the centre of the x-y plane $[5,5]$ and is defined as the following isentropic perturbation of unit values of primitive variables [4]:

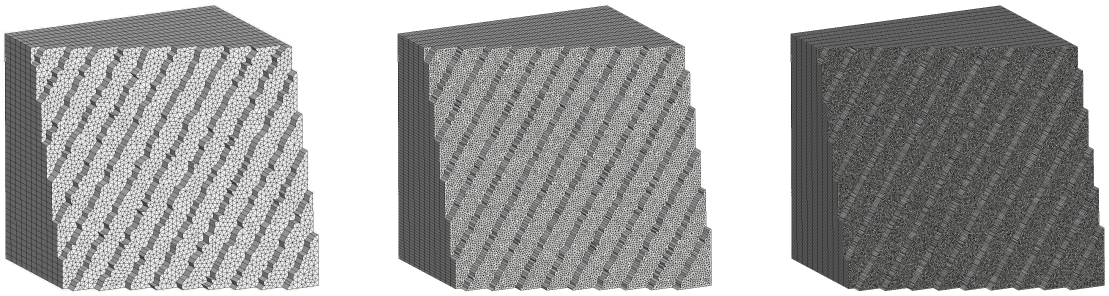
$$\begin{gathered} u=\frac{\varepsilon}{2\pi}e^{\frac{1-r^2}{2}(5-y)},\ \ v=\frac{\varepsilon}{2\pi}e^{\frac{1-r^2}{2}(x-5)},\ \ w=0, \\ T=\frac{(\gamma-1)\varepsilon^2}{8\gamma\pi^2}e^{\left(1-r^2\right)},\ \ \frac{p}{\rho^\gamma}=1,\ \ r^2=(x-5)^2+(y-5)^2 \end{gathered} \tag{4.5}$$

where the vortex strength is $\varepsilon=5$. The exact solution is a vortex movement in the $x-y$ plane with a constant velocity at 45^o to the Cartesian axis. We compute the numerical solution at the output time $t=10$ (one period) for which the vortex returns to the initial position. For this test case the following meshes used are illustrated in Figure 4.15 on page 76 to Figure 4.18 on page 77. Periodic boundary conditions are applied. The statistics of the meshes used can be found in Table 4.7 on page 78.



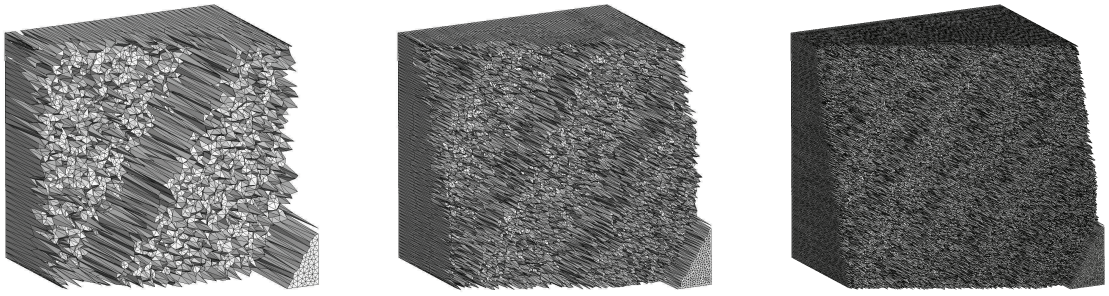
(a) 40x40x20 (b) 80x80x20 (c) 160x160x20

Figure 4.16: Sequence of hexahedral meshes (cutaway sections) used for convergence study of the model equation (4.4) with initial conditions (4.5)



(a) 40x40x20 (b) 80x80x20 (c) 160x160x20

Figure 4.17: Sequence of prismatic meshes (cutaway sections) used for convergence study of the model equation (4.4) with initial conditions (4.5)



(a) 40x40x20 (b) 80x80x20 (c) 160x160x20

Figure 4.18: Sequence of tetrahedral meshes (cutaway sections) used for convergence study of the model equation (4.4) with initial conditions (4.5)

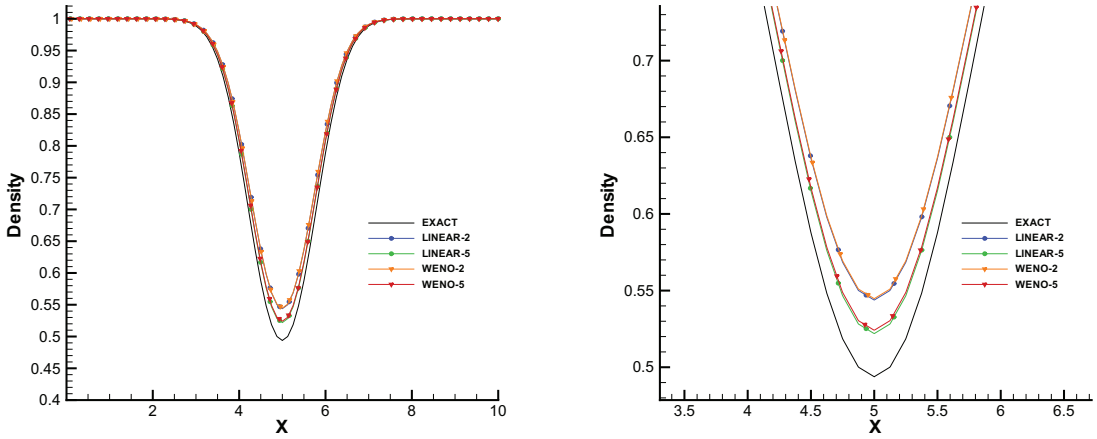


Figure 4.19: One dimensional profile cuts of density for various schemes using a uniform hexahedral mesh for the model equation (4.4) with initial conditions (4.5) at $t = 10.0$

Mesh Type	N	Tetrahedrals	Pyramids	Prisms	Hexahedrals	N_{tot}
Hexahedral	40x40x20	0	0	0	8000	8000
	80x80x20	0	0	0	64000	64000
	160x160x20	0	0	0	512000	512000
Prismatic	40x40x20	110800	0	0	0	110800
	80x80x20	440840	0	0	0	440840
	160x160x20	1761280	0	0	0	1761280
Tetrahedral	40x40x20	0	0	51844	0	51844
	80x80x20	0	0	350153	0	350153
	160x160x20	0	0	2085769	0	2085769
Hybrid	40x40x20	97343	39217	0	8000	137360
	80x80x20	351405	109755	0	64000	525160
	160x160x20	1041529	486391	0	512000	2039920

Table 4.7: Statistics for the meshes used for the computations of the model equation (4.4) with initial conditions (4.5)

Table 4.8 on page 79 to Table 4.11 on page 82 show errors and convergence rates in L_1 and L_∞ norm for cell averages of density. Both linear and WENO schemes are employed. We observe that the schemes achieve higher than expected convergence rates. Figure 4.19 on page 78 to Figure 4.22 on page 84 shows one dimensional profile cuts of density for different schemes at $y = 5, z = 0.25$.

Method	N	L_1 error	L_1 order	L_∞ error	L_∞ order
1st-Order	40x40x20	4.75×10^{-1}	-	5.31×10^{-1}	-
	80x80x20	2.89×10^{-1}	0.711	3.2×10^{-1}	0.730
	160x160x20	1.5×10^{-1}	0.951	1.8×10^{-1}	0.830
Linear-2	40x40x20	2.26×10^{-1}	-	3.7×10^{-1}	-
	80x80x20	5.86×10^{-2}	1.947	1.03×10^{-1}	1.844
	160x160x20	1.33×10^{-2}	2.139	2.96×10^{-2}	1.798
Linear-3	40x40x20	1.87×10^{-1}	-	2.45×10^{-1}	-
	80x80x20	2.61×10^{-2}	2.837	2.93×10^{-2}	3.061
	160x160x20	3.26×10^{-3}	3.001	3.6×10^{-3}	3.027
Linear-4	40x40x20	1.59×10^{-1}	-	1.82×10^{-1}	-
	80x80x20	8.72×10^{-3}	4.188	1.19×10^{-2}	3.926
	160x160x20	4.98×10^{-4}	4.129	7.31×10^{-4}	4.033
Linear-5	40x40x20	1.24×10^{-1}	-	1.49×10^{-1}	-
	80x80x20	3.57×10^{-3}	5.118	4.68×10^{-3}	4.992
	160x160x20	1.08×10^{-4}	5.046	1.47×10^{-4}	4.984
WENO-2	40x40x20	2.89×10^{-1}	-	3.5×10^{-1}	-
	80x80x20	7.34×10^{-2}	1.977	8.57×10^{-2}	2.029
	160x160x20	1.86×10^{-2}	1.980	2.21×10^{-2}	1.956
WENO-3	40x40x20	1.67×10^{-1}	-	1.96×10^{-1}	-
	80x80x20	2.31×10^{-2}	2.853	2.98×10^{-2}	2.717
	160x160x20	2.52×10^{-3}	3.196	3.76×10^{-3}	2.986
WENO-4	40x40x20	1.02×10^{-1}	-	1.25×10^{-1}	-
	80x80x20	5.67×10^{-3}	4.169	6.25×10^{-3}	4.321
	160x160x20	3.47×10^{-4}	4.034	4.16×10^{-4}	3.909
WENO-5	40x40x20	8.65×10^{-2}	-	9.64×10^{-1}	-
	80x80x20	2.33×10^{-3}	5.214	3.29×10^{-3}	4.872
	160x160x20	8.14×10^{-5}	4.839	9.33×10^{-5}	5.140

Table 4.8: Convergence study for various schemes using a uniform hexahedral mesh as applied to the model equation (4.4) with initial conditions (4.5) at output time $t = 10.0$.

Method	N	L_1 error	L_1 order	L_∞ error	L_∞ order
1st-Order	40x40x20	4.01×10^{-1}	-	5.19×10^{-1}	-
	80x80x20	2.02×10^{-1}	1.021	3.10×10^{-1}	0.746
	160x160x20	1.12×10^{-1}	0.850	1.62×10^{-1}	0.936
Linear-2	40x40x20	2.54×10^{-1}	-	3.96×10^{-1}	-
	80x80x20	6.37×10^{-2}	1.995	1.21×10^{-1}	1.710
	160x160x20	1.66×10^{-2}	1.940	3.21×10^{-2}	1.914
Linear-3	40x40x20	1.96×10^{-1}	-	2.51×10^{-1}	-
	80x80x20	2.87×10^{-2}	2.771	3.35×10^{-2}	2.905
	160x160x20	4.41×10^{-3}	2.702	4.63×10^{-3}	2.855
Linear-4	40x40x20	1.44×10^{-1}	-	1.81×10^{-1}	-
	80x80x20	8.52×10^{-3}	4.079	1.62×10^{-2}	3.481
	160x160x20	5.17×10^{-4}	4.042	9.44×10^{-4}	4.101
Linear-5	40x40x20	1.18×10^{-1}	-	1.28×10^{-1}	-
	80x80x20	2.86×10^{-3}	5.366	4.17×10^{-3}	4.939
	160x160x20	1.11×10^{-4}	4.687	1.56×10^{-4}	4.740
WENO-2	40x40x20	1.99×10^{-1}	-	2.63×10^{-1}	-
	80x80x20	5.31×10^{-2}	1.905	7.42×10^{-2}	1.825
	160x160x20	1.37×10^{-2}	1.954	1.83×10^{-2}	2.019
WENO-3	40x40x20	1.78×10^{-1}	-	2.57×10^{-1}	-
	80x80x20	1.95×10^{-2}	3.190	3.11×10^{-2}	3.046
	160x160x20	2.54×10^{-3}	2.864	4.07×10^{-3}	2.933
WENO-4	40x40x20	1.18×10^{-1}	-	1.39×10^{-1}	-
	80x80x20	6.89×10^{-3}	4.098	8.12×10^{-3}	4.097
	160x160x20	4.35×10^{-4}	3.985	5.31×10^{-4}	3.939
WENO-5	40x40x20	7.99×10^{-2}	-	1.27×10^{-1}	-
	80x80x20	2.46×10^{-3}	5.021	4.514×10^{-3}	4.814
	160x160x20	7.22×10^{-5}	5.090	1.28×10^{-5}	5.140

Table 4.9: Convergence study for various schemes using a prismatic mesh as applied to the model equation (4.4) with initial conditions (4.5) at output time $t = 10.0$.

Method	N	L_1 error	L_1 order	L_∞ error	L_∞ order
1st-Order	40x40x20	4.56×10^{-1}	-	5.47×10^{-1}	-
	80x80x20	2.38×10^{-1}	0.940	3.70×10^{-1}	0.562
	160x160x20	1.19×10^{-1}	0.994	1.76×10^{-1}	1.071
Linear-2	40x40x20	2.73×10^{-1}	-	4.2×10^{-1}	-
	80x80x20	7.17×10^{-2}	1.932	1.42×10^{-1}	1.581
	160x160x20	1.86×10^{-2}	1.946	3.69×10^{-2}	1.943
Linear-3	40x40x20	2.18×10^{-1}	-	2.84×10^{-1}	-
	80x80x20	3.01×10^{-2}	2.857	3.42×10^{-2}	3.051
	160x160x20	5.20×10^{-3}	2.533	5.49×10^{-3}	2.641
Linear-4	40x40x20	1.56×10^{-1}	-	2.12×10^{-1}	-
	80x80x20	9.24×10^{-3}	4.081	1.91×10^{-2}	3.476
	160x160x20	5.30×10^{-4}	4.123	1.04×10^{-3}	4.198
Linear-5	40x40x20	1.41×10^{-1}	-	1.53×10^{-1}	-
	80x80x20	2.92×10^{-3}	5.605	4.71×10^{-3}	5.024
	160x160x20	1.19×10^{-4}	4.604	1.59×10^{-4}	4.890
WENO-2	40x40x20	2.09×10^{-1}	-	3.03×10^{-1}	-
	80x80x20	6.32×10^{-2}	1.724	8.88×10^{-2}	1.782
	160x160x20	1.45×10^{-2}	2.118	1.99×10^{-2}	2.145
WENO-3	40x40x20	1.31×10^{-1}	-	2.93×10^{-1}	-
	80x80x20	2.13×10^{-2}	3.299	3.26×10^{-2}	3.168
	160x160x20	2.80×10^{-3}	2.927	4.49×10^{-3}	2.860
WENO-4	40x40x20	1.31×10^{-1}	-	1.63×10^{-1}	-
	80x80x20	7.37×10^{-3}	4.157	8.64×10^{-3}	4.240
	160x160x20	4.42×10^{-4}	4.059	5.87×10^{-4}	3.887
WENO-5	40x40x20	8.63×10^{-2}	-	1.44×10^{-1}	-
	80x80x20	2.71×10^{-3}	4.989	4.80×10^{-3}	4.096
	160x160x20	7.7×10^{-5}	5.141	1.49×10^{-4}	5.014

Table 4.10: Convergence study for various schemes using a tetrahedral mesh as applied to the model equation (4.4) with initial conditions (4.5) at output time $t = 10.0$.

Method	N	L_1 error	L_1 order	L_∞ error	L_∞ order
1st-Order	40x40x20	3.38×10^{-1}	-	4.63×10^{-1}	-
	80x80x20	2.31×10^{-1}	0.639	2.6×10^{-1}	0.833
	160x160x20	1.23×10^{-1}	0.901	1.55×10^{-1}	0.744
Linear-2	40x40x20	1.60×10^{-1}	-	2.83×10^{-1}	-
	80x80x20	5.52×10^{-2}	1.541	7.9×10^{-2}	1.843
	160x160x20	1.16×10^{-2}	2.251	2.31×10^{-2}	1.770
Linear-3	40x40x20	1.73×10^{-1}	-	2.28×10^{-1}	-
	80x80x20	1.99×10^{-2}	3.120	2.86×10^{-2}	2.994
	160x160x20	2.74×10^{-3}	2.864	3.42×10^{-3}	3.063
Linear-4	40x40x20	1.43×10^{-1}	-	1.46×10^{-1}	-
	80x80x20	8.52×10^{-3}	4.072	9.41×10^{-3}	3.958
	160x160x20	3.83×10^{-4}	4.474	5.66×10^{-4}	4.056
Linear-5	40x40x20	9.38×10^{-2}	-	1.11×10^{-1}	-
	80x80x20	2.89×10^{-3}	5.019	4.05×10^{-3}	4.767
	160x160x20	9.23×10^{-5}	4.970	1.23×10^{-4}	5.036
WENO-2	40x40x20	2.75×10^{-1}	-	3.06×10^{-1}	-
	80x80x20	7.21×10^{-2}	1.931	7.24×10^{-2}	2.082
	160x160x20	1.65×10^{-2}	2.127	1.69×10^{-2}	2.099
WENO-3	40x40x20	1.59×10^{-1}	-	1.53×10^{-1}	-
	80x80x20	1.97×10^{-2}	3.010	2.80×10^{-2}	2.457
	160x160x20	1.92×10^{-3}	3.357	3.71×10^{-3}	2.916
WENO-4	40x40x20	8.55×10^{-1}	-	1.14×10^{-1}	-
	80x80x20	4.5×10^{-3}	4.248	6.02×10^{-3}	4.248
	160x160x20	2.8×10^{-4}	4.008	3.82×10^{-4}	3.977
WENO-5	40x40x20	6.53×10^{-2}	-	7.12×10^{-1}	-
	80x80x20	2.22×10^{-3}	4.870	3.27×10^{-3}	4.442
	160x160x20	5.83×10^{-5}	5.258	8.03×10^{-5}	5.350

Table 4.11: Convergence study for various schemes using a hybrid mesh as applied to the model equation (4.4) with initial condition (4.5) at output time $t = 10.0$.

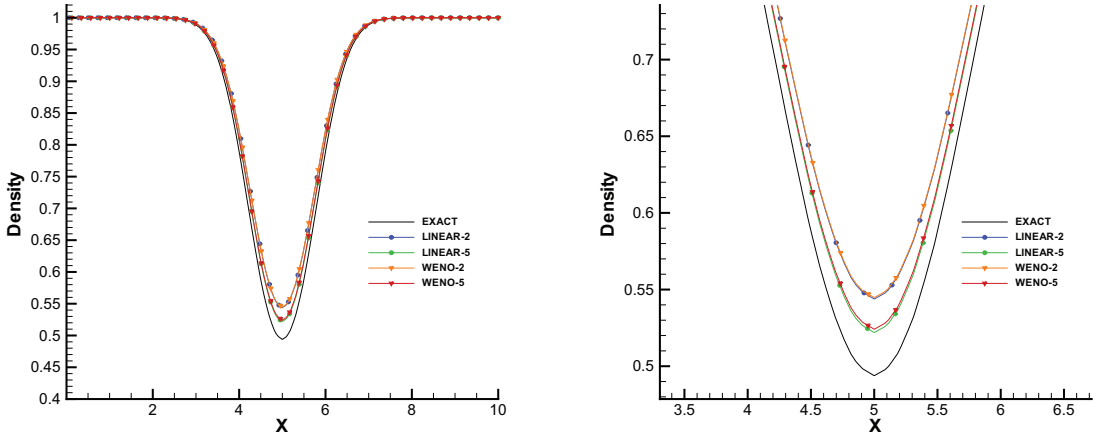


Figure 4.20: One dimensional profile cuts of density for various schemes using a prismatic mesh for the model equation (4.4) with initial conditions (4.5) at $t = 10.0$

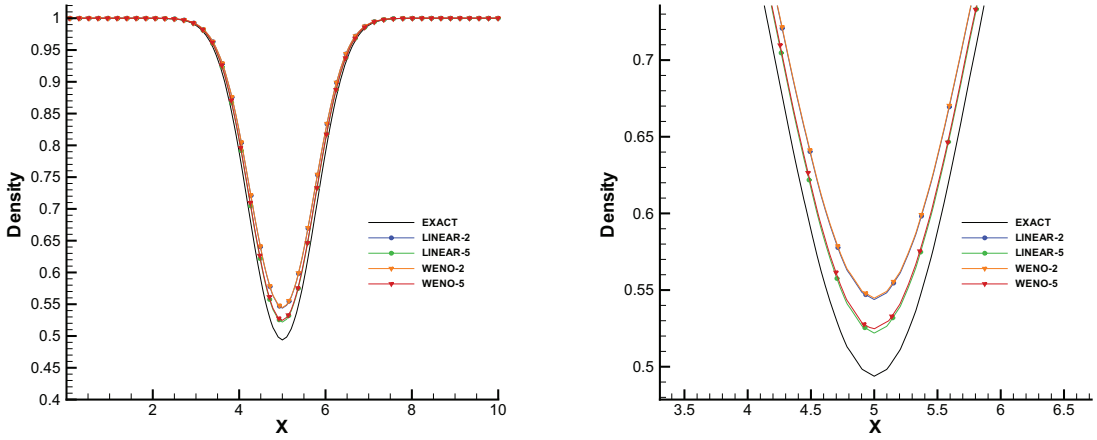


Figure 4.21: One dimensional profile cuts of density for various schemes using a tetrahedral mesh for the model equation (4.4) with initial conditions (4.5) at $t = 10.0$

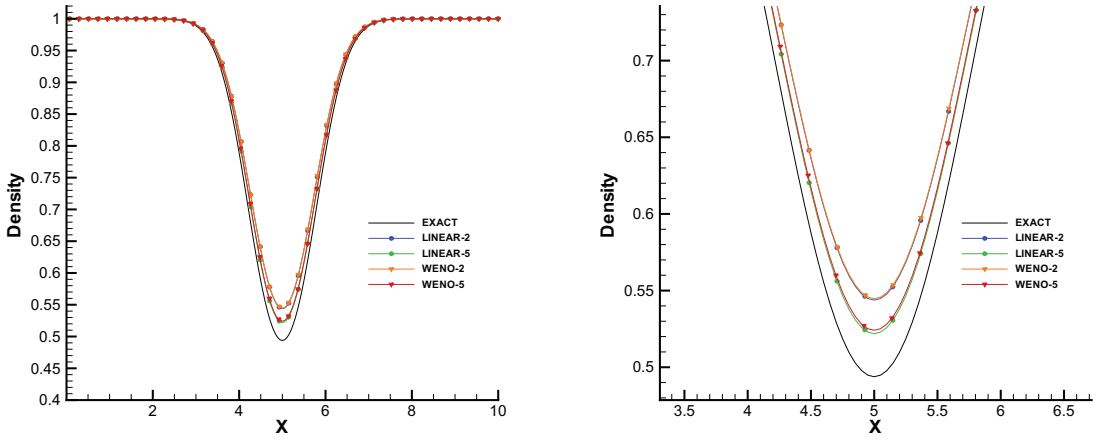


Figure 4.22: One dimensional profile cuts of density for various schemes using a hybrid mesh for the model equation (4.4) with initial conditions (4.5) at $t = 10.0$

4.2.2 Shock tube

We consider the one-dimensional shock tube problem computed in a 3D manner which is a a modification of the original Sod test problem proposed by [70]. The computational domain has the shape of a rectangular tube with length $L = 1$, from $x = 0$ to $x = 1$, and width $W = 0.5$. The interior is meshed by an unstructured hexahedral mesh as shown in Figure 4.23 on page 85 with $N = 100$ across x-direction . Transmissive boundary conditions are used along x-axis and periodic boundary conditions along y, and z-axis . We solve the three-dimensional Euler equations (4.4) with the initial condition

$$(\rho, u, v, w, p)(\vec{x}, 0) = \begin{cases} (1,0,0,0,1) & if\ x \leq 0.5 \\ (0.125,0,0,0,0.1) & if\ x > 0.5 \end{cases} \tag{4.6}$$

Figure 4.24 on page 85 shows a comparison of the density profile of WENO-3 across x-direction. We observe the non-oscillatory properties of the developed scheme and the influence of the linear weight assigned to the central stencils. The solution profile is slightly sharper with the larger linear weight but with some slight oscillations on the other hand the smaller linear weight does not produce any oscillations but also does not have such a sharp profile. In other words the nature of the problem to be solved drives the values for the linear weights of the central stencil as also documented by [49, 57]. For the majority of the problems the most robust value for the central stencil linear weight should be 100 as it has been mentioned in the literature to be the most robust [49, 22].

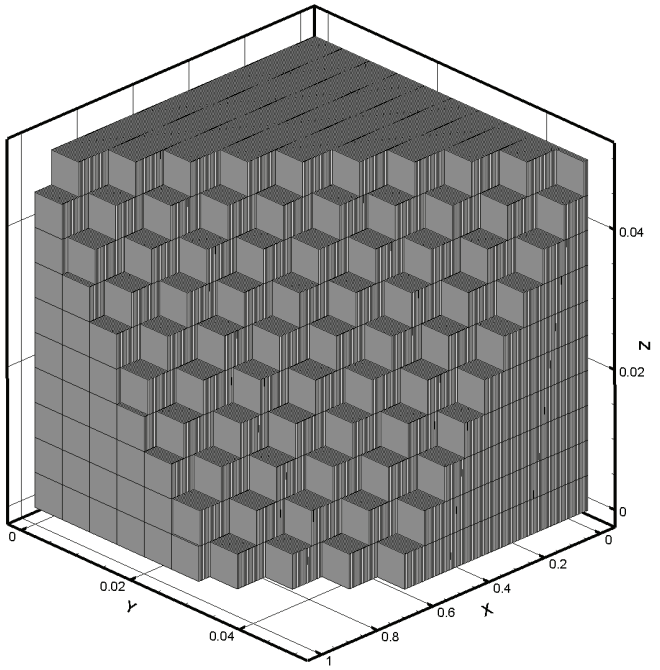


Figure 4.23: Cutaway section of unstructured hexahedral mesh used for convergence study of the model equation (4.4) with initial conditions (4.6)

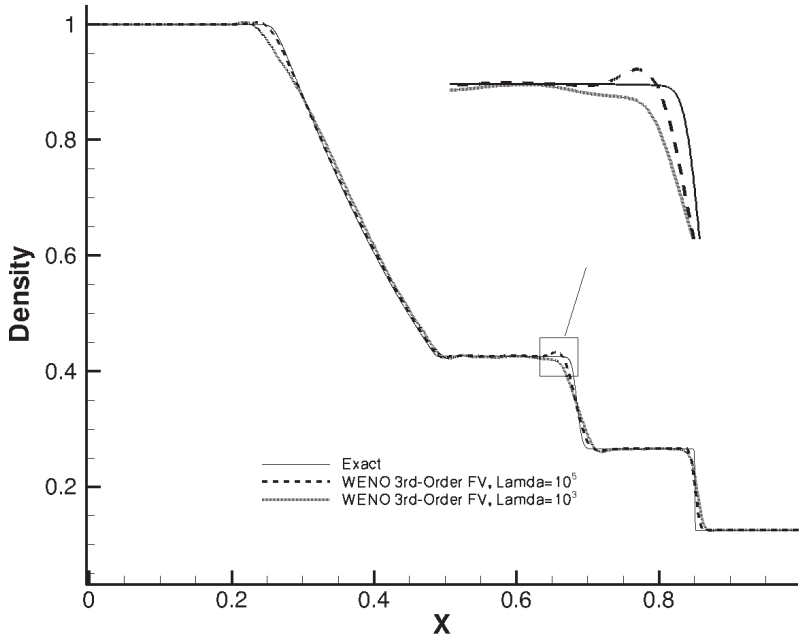
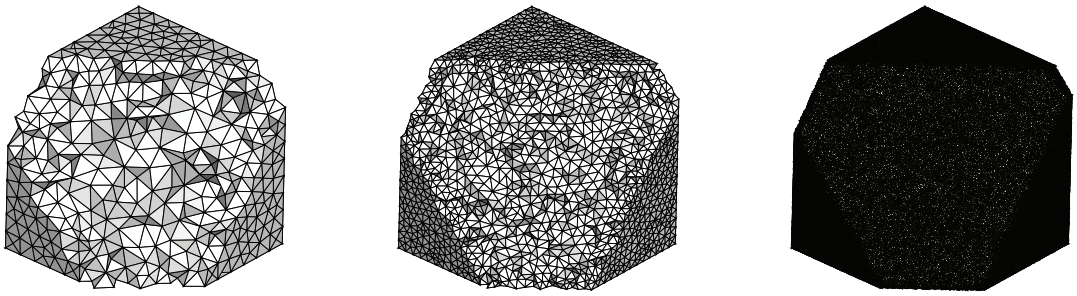


Figure 4.24: Density profile across x-direction for WENO-3 with different values of linear weights assigned to the central stencil applied to the model equation (4.4) with initial conditions (4.6)



(a) N=10 (b) N=20 (c) N=80

Figure 4.25: Sequence of tetrahedral meshes (cutaway sections) used for the study of the non-oscillatory properties of the schemes applied to the model equation (4.4) with initial conditions (4.7)

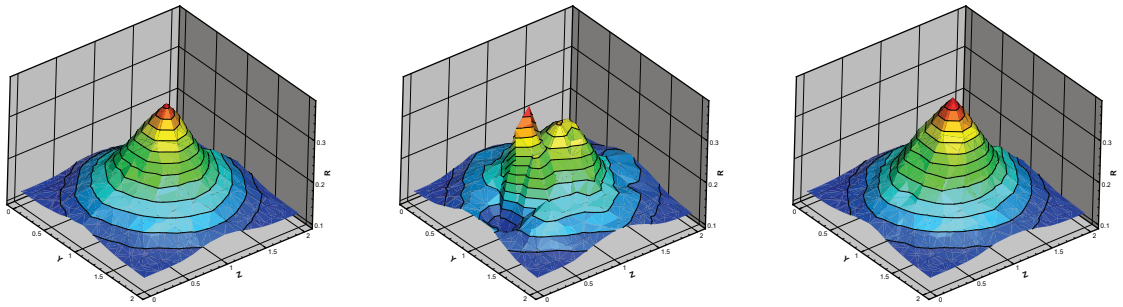
4.2.3 Explosion

We calculate the solution of the so-called spherical explosion test problem [70]. The initial condition defined on $[0:2]^3$ consists of two regions of constant but different values of gas parameters separated by a sphere of radius 0.4 :

$$(\rho,u,v,w,p)(\vec{x},0) = \begin{cases} (1,0,0,0,1) & if\ r \leq 0.4 \\ (0.125,0,0,0,0.1) & if\ r > 0.4 \end{cases}, \ r^2 = x^2+y^2+z^2. \tag{4.7}$$

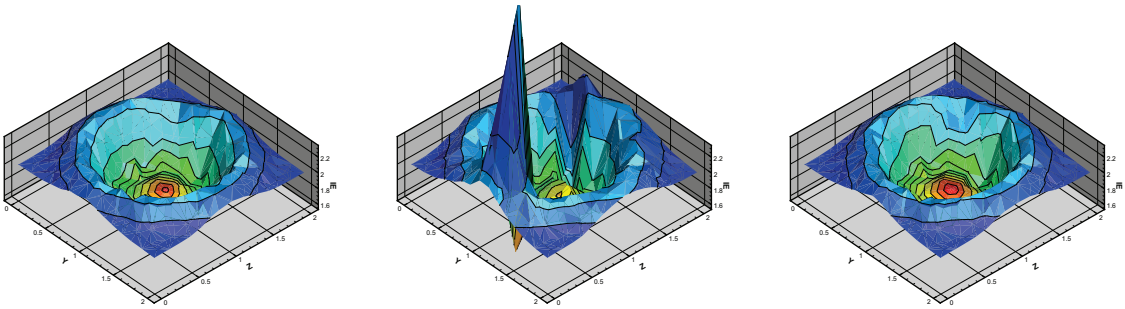
The Euler equations (4.4) are solved, transmissive boundary conditions are applied and numerical solution is computed at the output time $t = 0.25$ on a sequence of refined tetrahedral meshes with $N = 10, 20, 80$ cells along each edge of the cube as shown in Figure 4.25 on page 86 .

We present distributions of gas density ρ and internal energy $e = T/(\gamma-1)$ in Figure 4.26 on page 87 to Figure 4.30 on page 89. The solution contains a spherical shock wave and a contact surface traveling away from the centre and a spherical rarefaction wave traveling towards the origin $(1,1,1)$. First of all it is noticed that TVD-2 is producing the correct flow pattern, and that the WENO-3 scheme depends on the geometrical directionality condition D_c (2.22). Ensuring that all the nodes of a candidate element for the directional stencils lie within this sector results in more robust scheme and hence the possibility for having at least one stencil within a smooth region (unless discontinuities are too close to each other in terms of grid spacing) is greater. Although other approaches [1, 32, 62, 31, 75, 42] use the barycentre to determine if an element lies within a sector, for arbitrary shaped elements this is not sufficient since the barycentre could lie within a sector but at the same time a node could not and this could dramatically impact the reliability of the scheme as it is illustrated in Figure 4.26 on page 87 to Figure 4.29 on page 88. We observe that the WENO-3 scheme with $D_c = 1.0$ produces a much sharper profile than the corresponding TVD-2 and at the same time without any oscillations. It must be stressed that the geometrical parameter D_c since it is



(a) TVD-2 (b) WENO-3 with Dc=0.2 (c) WENO-3 with Dc=1.0

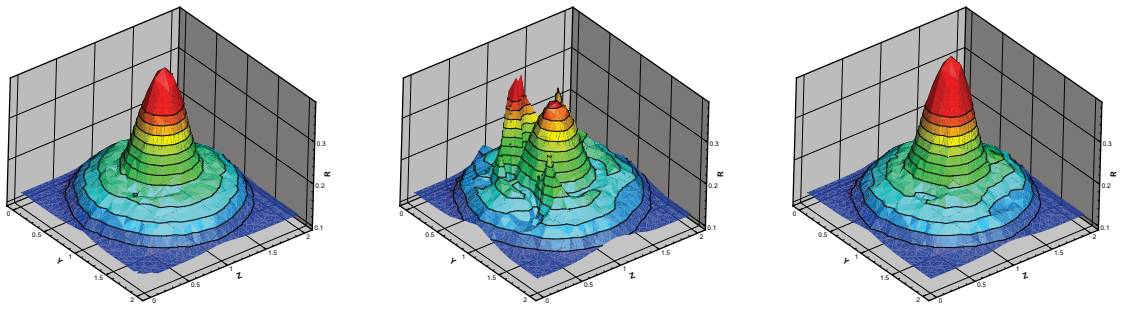
Figure 4.26: Density profile at $x = 1$ for various schemes for tetrahedral mesh for $N = 10$ for the model equation (4.4) with initial conditions (4.7) at time $t = 0.25$



(a) TVD-2 (b) WENO-3 with Dc=0.2 (c) WENO-3 with Dc=1.0

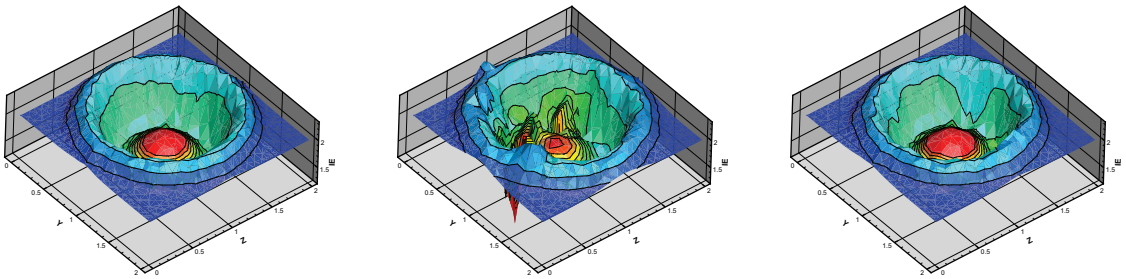
Figure 4.27: Internal energy profile at $x = 1$ for various schemes for tetrahedral mesh for $N = 10$ for the model equation (4.4) with initial conditions (4.7) at time $t = 0.25$

dependent on the mesh elements it should be a local adaptive parameter where every element depending on the surrounding elements would adjust this value so that no one-sided stencils along a straight line are constructed (high condition number for least square reconstruction) and that most admissible stencils are constructed.



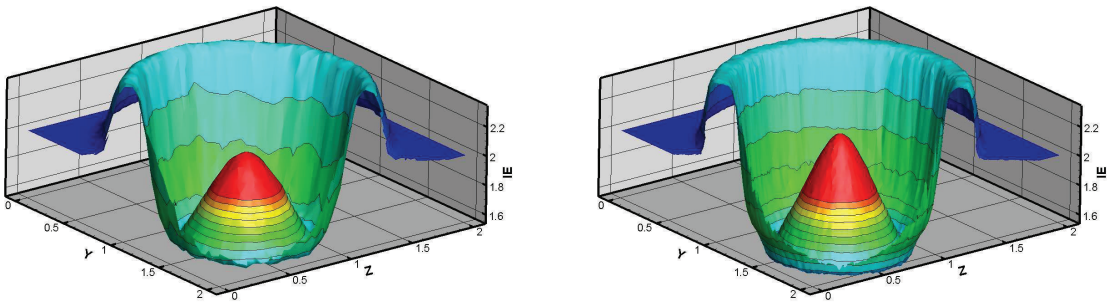
(a) TVD-2 (b) WENO-3 with Dc=0.2 (c) WENO-3 with Dc=1.0

Figure 4.28: Density profile at $x = 1$ for various schemes for tetrahedral mesh for $N = 20$ for the model equation (4.4) with initial conditions (4.7) at time $t = 0.25$



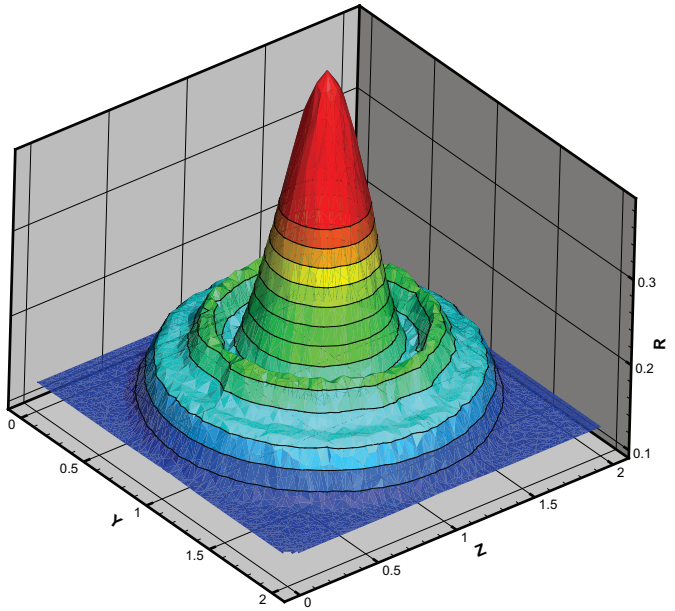
(a) TVD-2 (b) WENO-3 with Dc=0.2 (c) WENO-3 with Dc=1.0

Figure 4.29: Internal energy profile at $x = 1$ for various schemes for tetrahedral mesh for $N = 20$ for the model equation (4.4) with initial conditions (4.7) at time $t = 0.25$

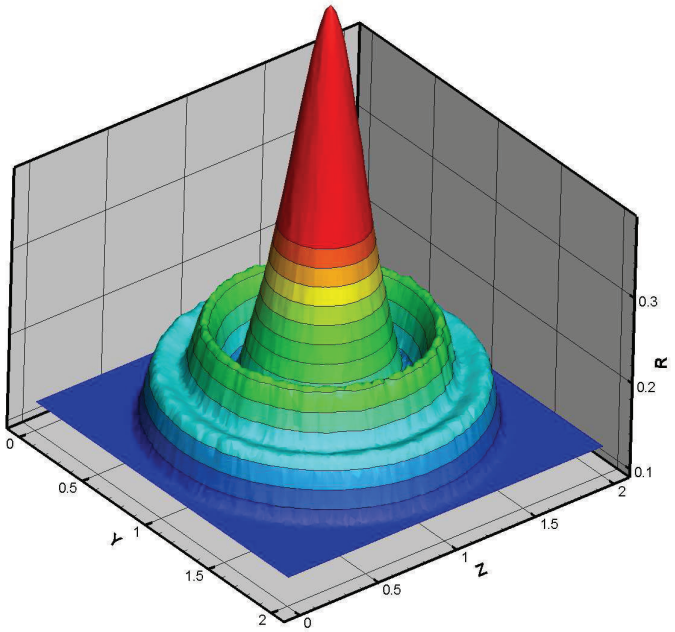


(a) TVD-2 (b) WENO-3 with Dc=1.0

Figure 4.30: Internal energy profile at $x = 1$ for various schemes for tetrahedral mesh for $N = 80$ for the model equation (4.4) with initial conditions (4.7) at time $t = 0.25$

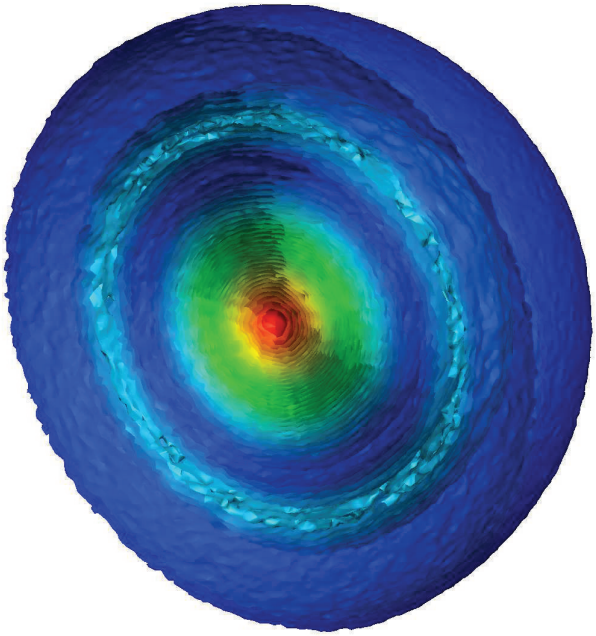


(a) TVD-2

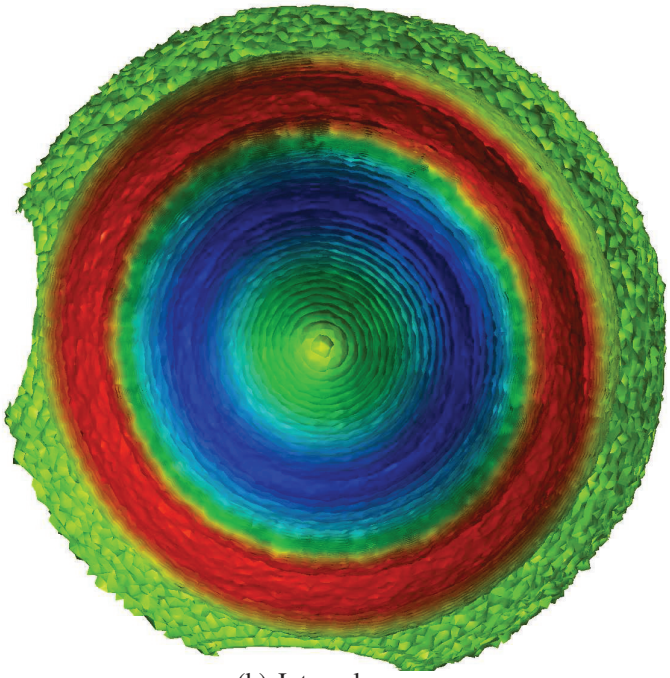


(b) WENO-3 with Dc=1.0

Figure 4.31: Density profile at $x = 1$ for various schemes for tetrahedral mesh for $N = 80$ for the model equation (4.4) with initial conditions (4.7) at time $t = 0.25$



(a) Density



(b) Internal energy

Figure 4.32: Isosurfaces cutaway sections of tetrahedral mesh for $N = 80$ for the model equation (4.4) with initial conditions (4.7) at time $t = 0.25$using a WENO-3 scheme.

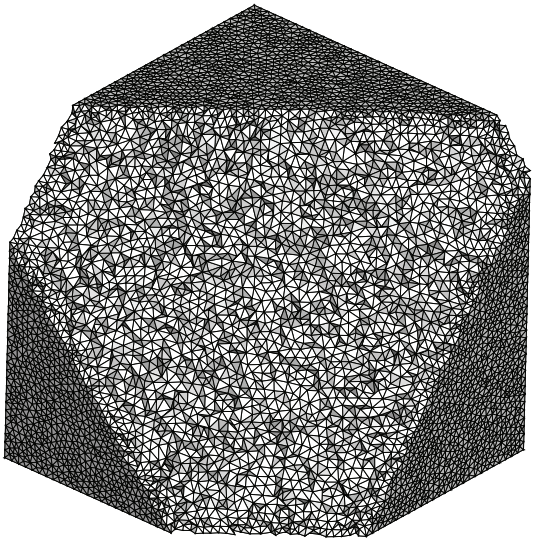


Figure 4.33: Tetrahedral mesh (cutaway sections) used for the study of the non-oscillatory properties of the WENO-3 applied to the model equation (4.4) with initial conditions (4.8)

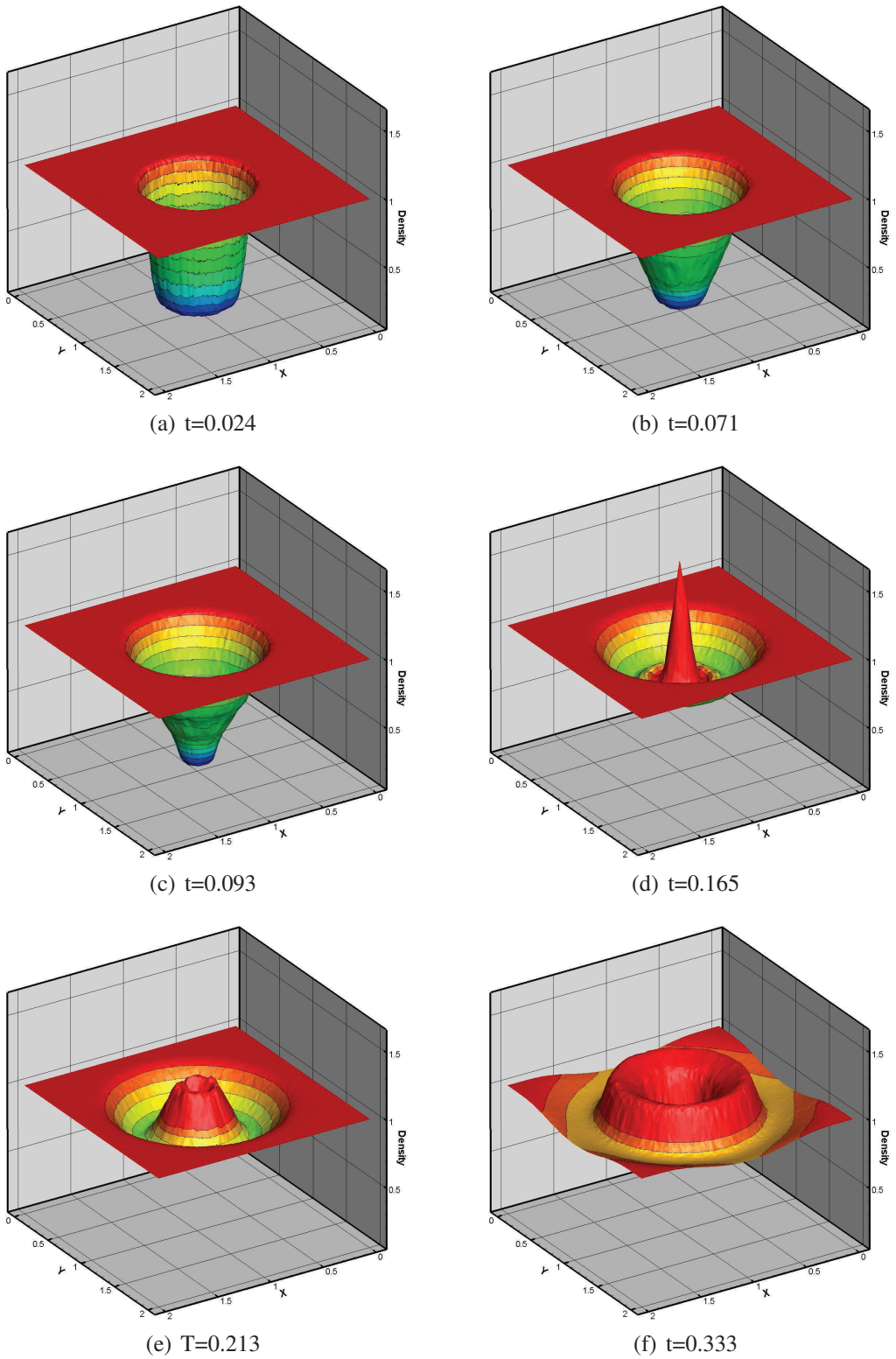
4.2.4 Implosion

We calculate the solution of the so-called spherical implosion test problem [70]. The initial condition defined on $[0:2]^3$ consists of two regions of constant but different values of gas parameters separated by a sphere of radius 0.4 :

$$(\rho,u,v,w,p)(\vec{x},0) = \begin{cases} (1,0,0,0,1) & if\ r > 0.4 \\ (0.125,0,0,0,0.1) & if\ r \leq 0.4 \end{cases}, \ r^2 = x^2+y^2+z^2. \tag{4.8}$$

The Euler equations (4.4) are solved, transmissive boundary conditions are applied and numerical solution is computed on a tetrahedral mesh of 3710329 cells as shown in Figure 4.33 on page 92 .

We present distributions of gas density ρ and internal energy $e = T/(\gamma-1)$ in the Figure 4.34 on page 93 to Figure 4.37 on page 96 . The purpose for computing the spherical implosion test problem is not to gain an insight in the processes involved during this complicated phenomena that occurs in nuclear physics, type-II supernova, black holes etc but to illustrate the robustness of the schemes for this challenging problem. The challenge in this problem is that the discontinuities are not moving farther away but they move towards each other until they collide and then an explosion process occurs. A WENO-3 scheme is employed with directionality condition $D_c = 1.0$ and a linear weight assigned to the central stencil $d_0 = 10^2$ since this is test with strong discontinuities. We remark that any spurious oscillation that could occur would result in a blown-up solution and the robustness of the WENO-3 scheme is demonstrated by the fact that no spurious oscillations are created.

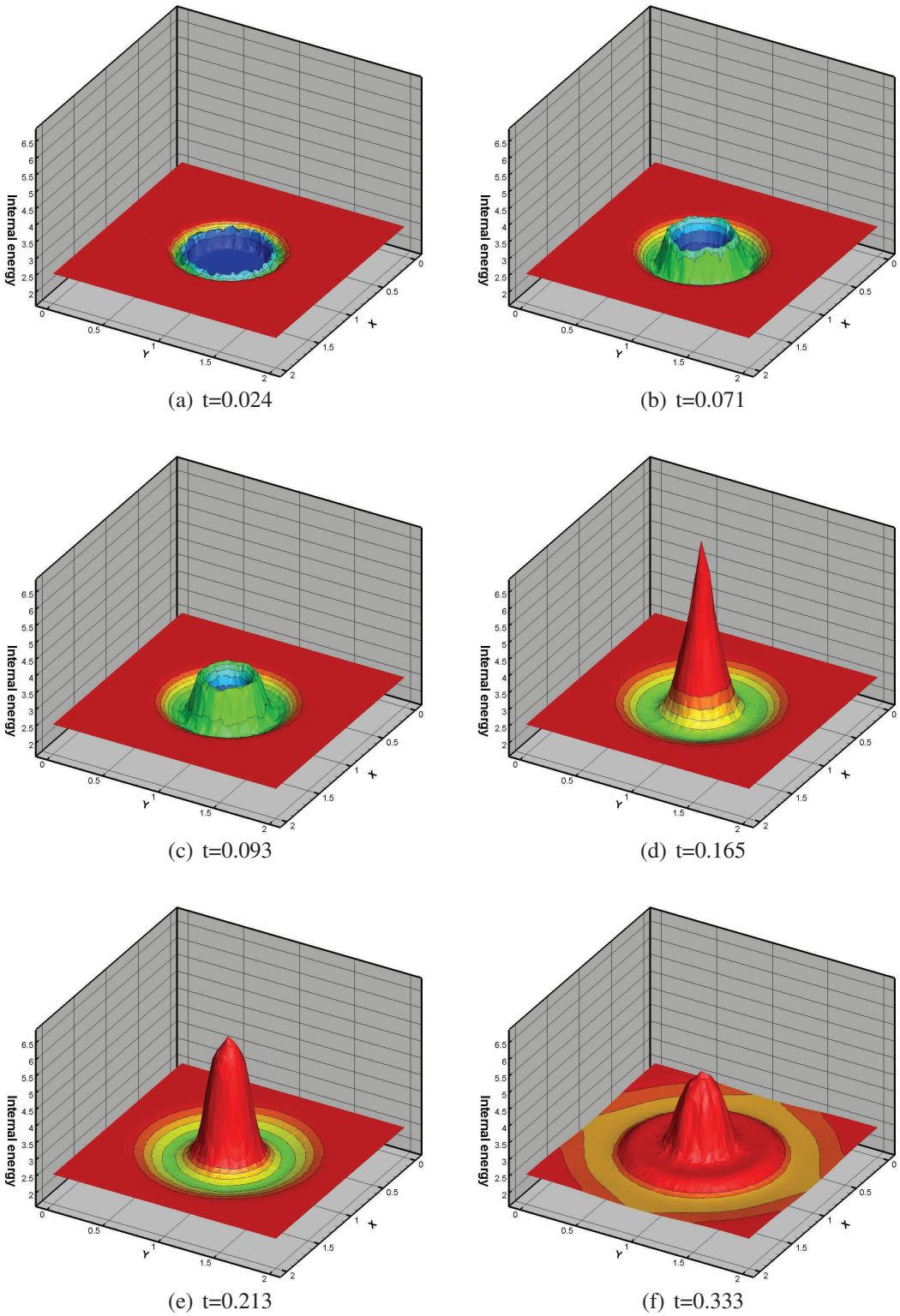


(a) t=0.024 (b) t=0.071

(c) t=0.093 (d) t=0.165

(e) T=0.213 (f) t=0.333

Figure 4.34: Density profile at $z = 1$ for WENO-3 for the model equation (4.4) with initial conditions (4.8) at various instants



(a) t=0.024 (b) t=0.071

(c) t=0.093 (d) t=0.165

(e) t=0.213 (f) t=0.333

Figure 4.35: Internal energy profile at $z = 1$ for WENO-3 for the model equation (4.4) with initial conditions (4.8) at various instants

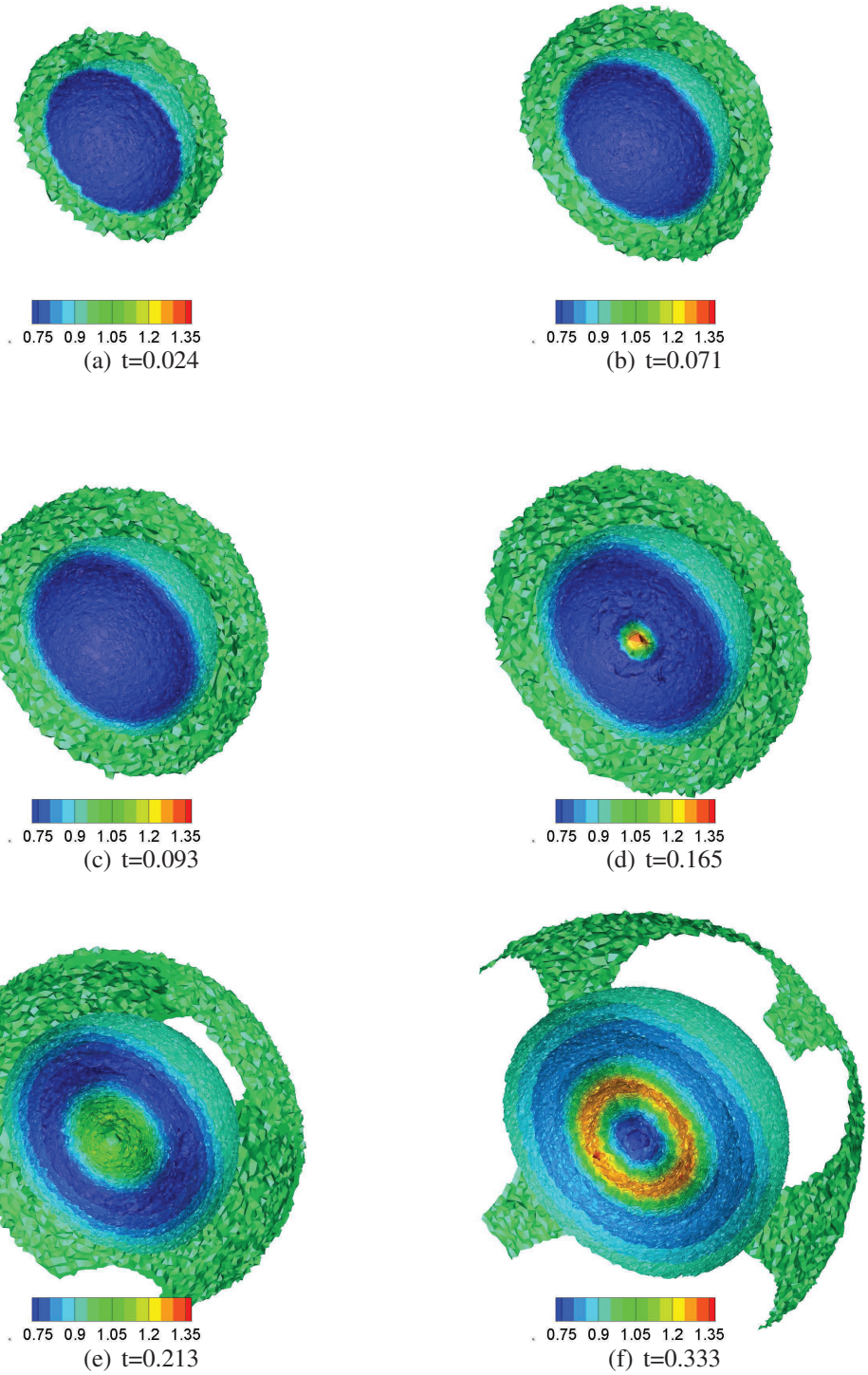


Figure 4.36: Density isosurfaces cutaway section at various instants for the 3D implosion test problem using a WENO-3 scheme

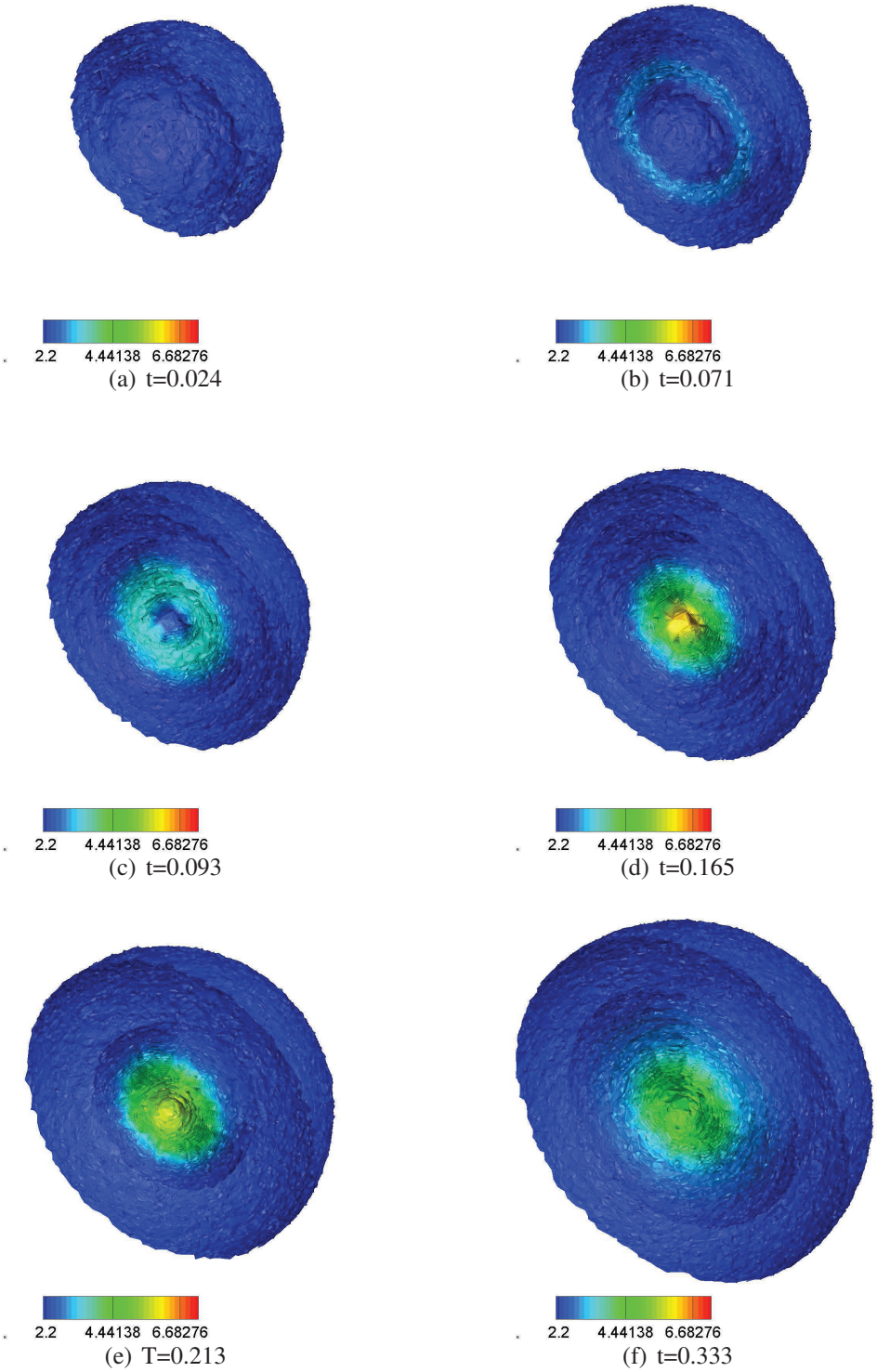


Figure 4.37: Internal energy isosurfaces cutaway section at various instants for the 3D implosion test problem using a WENO-3 scheme

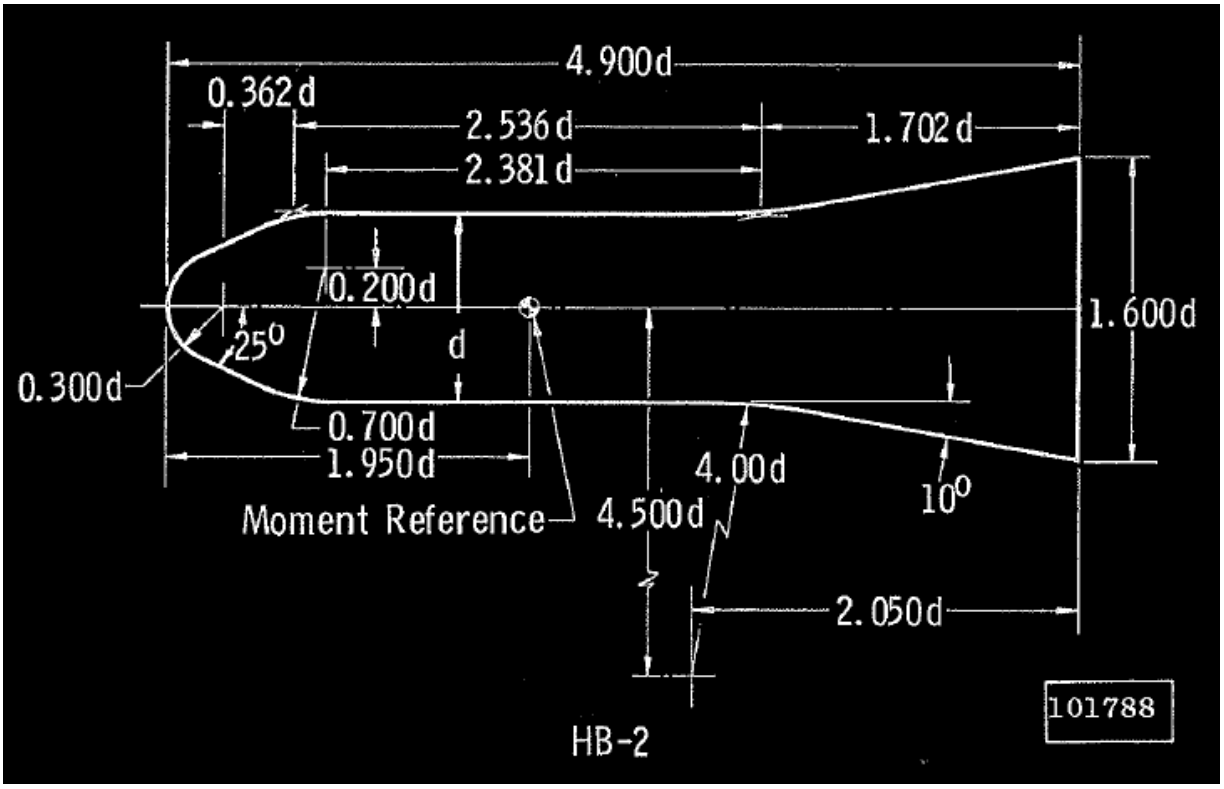


Figure 4.38: HB2 Geometry taken from [27]

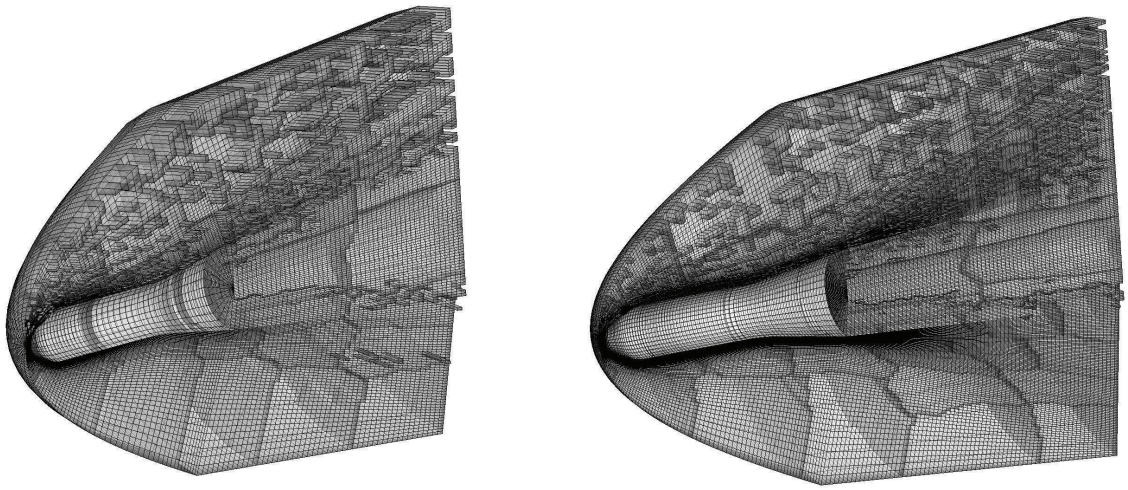
4.2.5 Blunted-cone-cylinder-flare Test Case

Finally we apply the third-order WENO method to the flow over a realistic geometry. We consider the so-called blunted-cone-cylinder-flare geometry, designated HB-2. This geometry has been used extensively in aerodynamic test-facilities [27]. Vast amounts of data gathered from experiments conducted under axissymetric three dimensional conditions are available in the literature. This high-speed flow problem is used to test the robustness of the proposed methods as applied to real-life applications.

The geometry of the problem is shown in Figure 4.38 on page 97. In our computational setup the x-axis is directed along the body. The computational domain (including the wake region) is meshed by two hybrid unstructured meshes of different resolution; Mesh 1 having 64431 cells and Mesh 2 having 690040 cells as shown in Figure 4.39 on page 98 . Although most of the computational cells are hexahedral, near the nose and the base of the body the mesh contains prisms and is thus of mixed-element type. We present the computational results for the case with the free-stream Mach number equal to 5 at zero angle of attack based on the compressible Euler equations.

We monitor the convergence of the total normal pressure force of the HB2 geometry. The pressure force is defined by the following formula:

$$F_x = (p \cdot S) \cdot n_x \tag{4.9}$$



(a) Mesh 1 (b) Mesh 2

Figure 4.39: Meshes used for the HB2 Geometry

Where p is the pressure, S is the surface area of the HB2 and $n_x$is the x-axis normal vector. The convergence of the normal pressure force of the WENO-3 scheme can be seen in Figure 4.40 on page 99 .

Figure 4.41 on page 100 and Figure 4.42 on page 101 show the pressure distribution along the body, normalised by the post-shock stagnation pressure and ploted against longitudinal position for Mesh 1 and Mesh 2 respectively. We notice that for both meshes WENO-3 scheme produces much sharper pressure profile as opposed to TVD-2 scheme although the performance of the schemes is dramatically improved by the increase in the mesh resolution. Figure 4.43 on page 102 shows the density distribution obtained from TVD-2 and WENO-3 schemes where it is noticed that WENO-3 produces slightly sharper profile compared to TVD-2 scheme. Finally the U velocity isosurfaces for WENO-3 scheme are shown in Figure 4.44 on page 103.

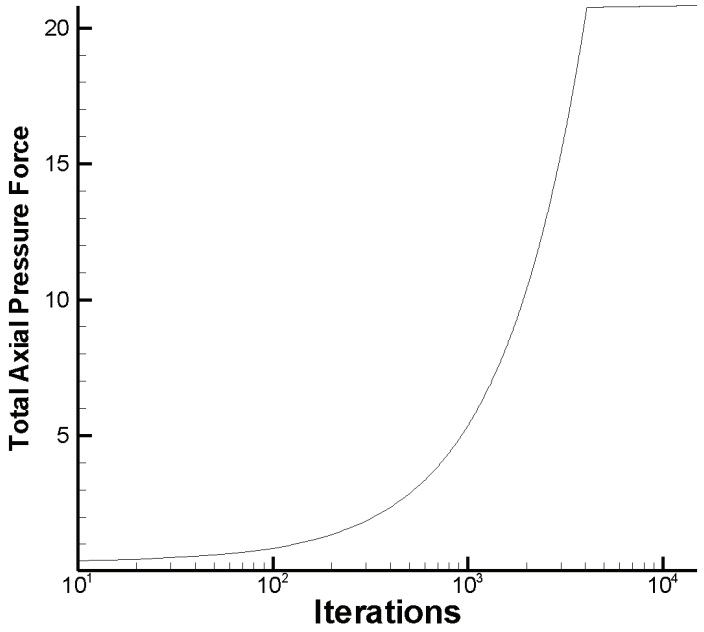


Figure 4.40: Normal Pressure force convergence for the WENO-3 scheme

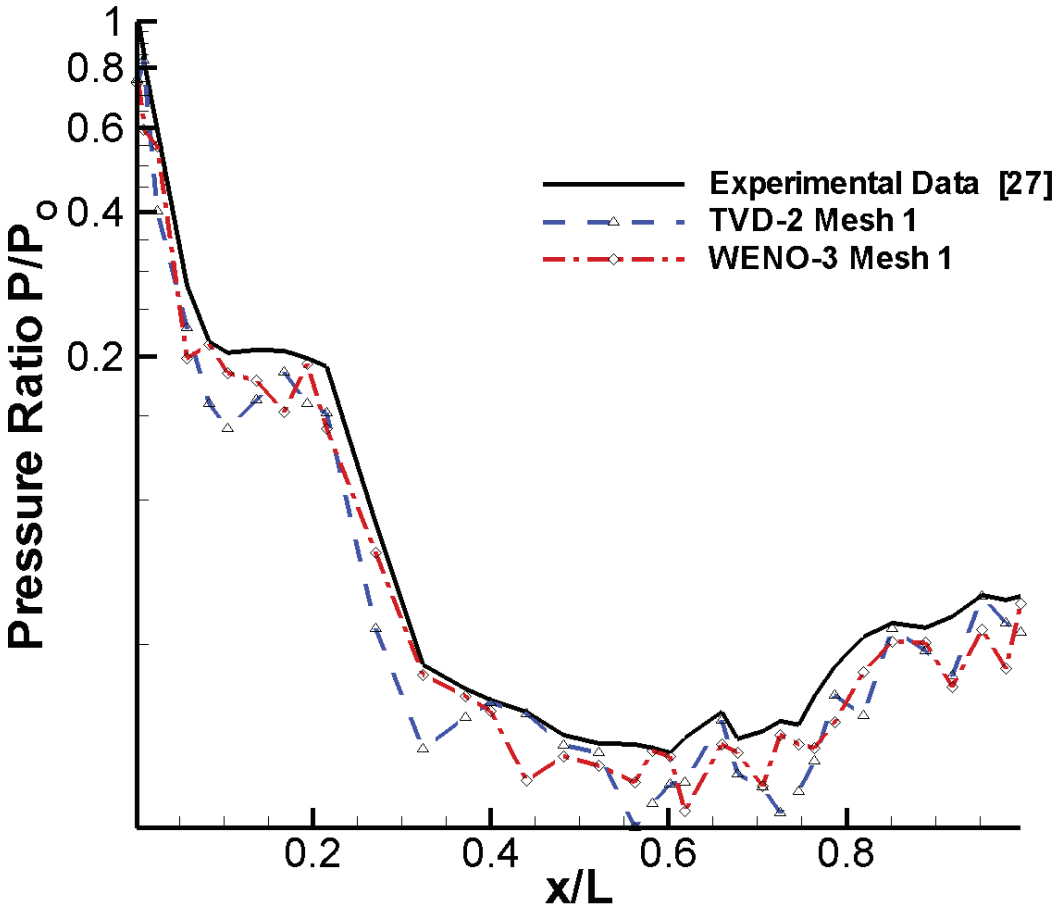


Figure 4.41: Normalised Pressure Distribution for HB2 using Mesh 1

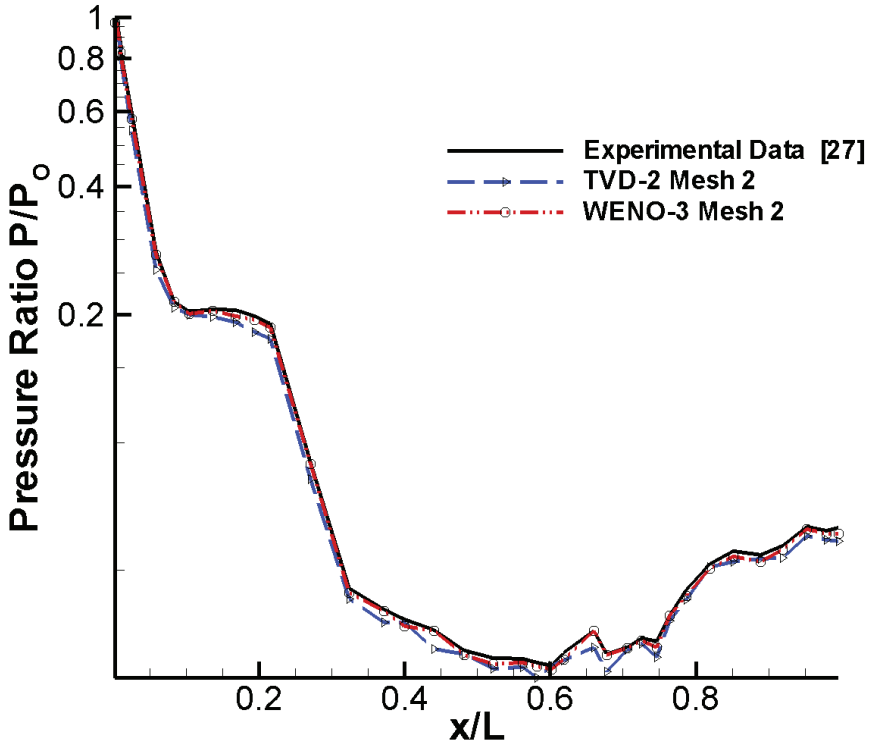
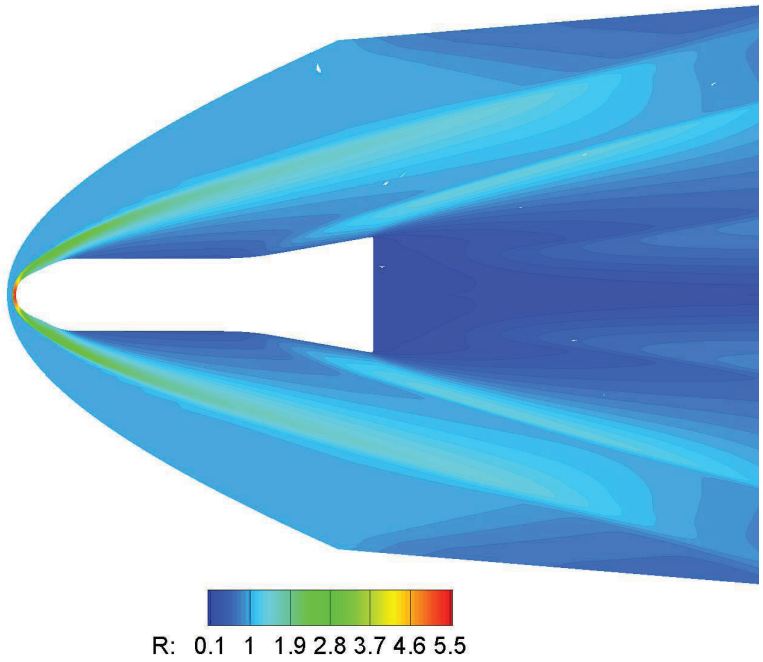
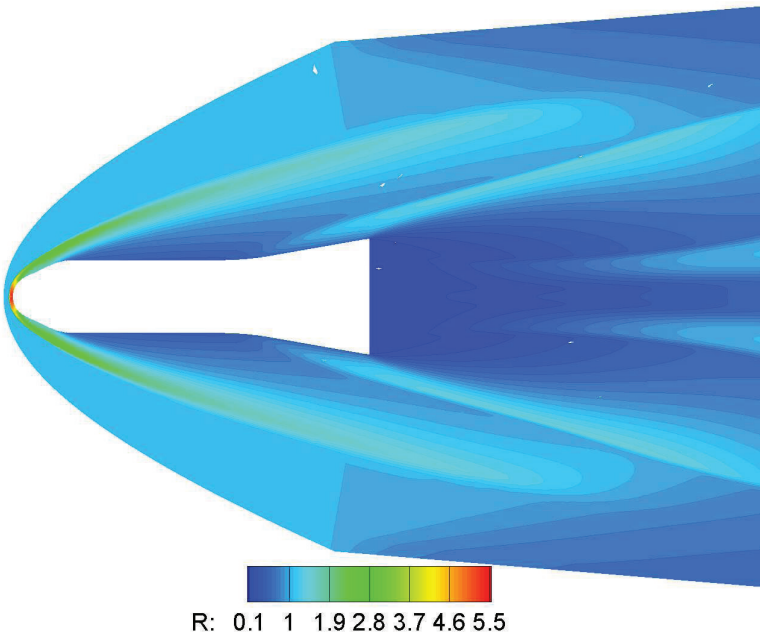


Figure 4.42: Normalised Pressure Distribution for HB2 using Mesh 2



(a) TVD-2



(b) WENO-3

Figure 4.43: Density profile at Y=0, for HB2 from various schemes

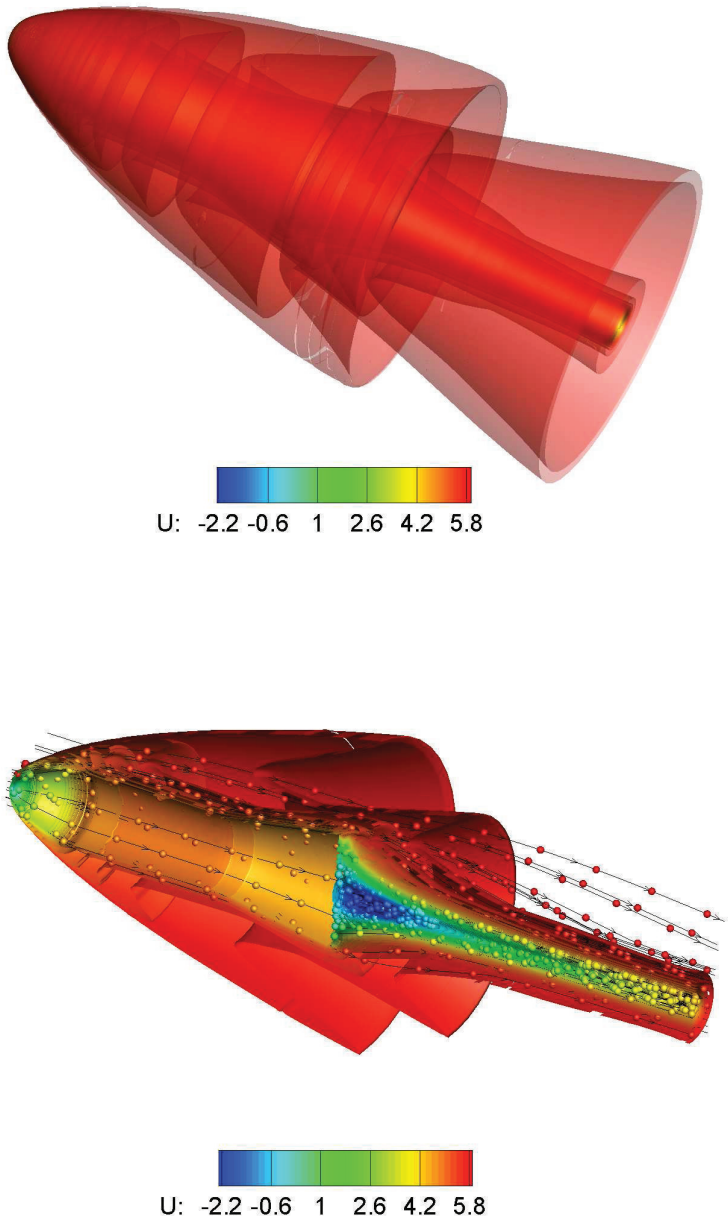


Figure 4.44: U Velocity isosurfaces for HB2 geometry using WENO-3 scheme

Chapter 5

Parallel Implementation

5.1 Introduction

The requirements for large-scale three-dimensional high-quality CFD analysis for industrial applications have increased due to the huge leap in computing processing power of the last decade. Parallelisation of existing computational methods and software is a subject of active research. However in the context of very high-order finite volume schemes tailored for any type of unstructured meshes the research activity is still very limited. One of the reasons for that is the arbitrary nature of the meshes used and the arbitrariness of the load balancing between the processes which leads to poor parallel performance of unstructured solvers.

Most of the algorithms employed in various highly sophisticated unstructured solvers perform worse than a structured solver [16, 6, 79, 7, 8, 46]. One of the advantages of explicit methods used here is that they can be easily parallelised based on domain decomposition. In this chapter we present the parallelisation strategy. We have exploited the potential of parallel algorithms for unstructured meshes and have designed a new family of algorithms that offer similar parallelisation efficiency if not better than structured solvers [2, 41]. In this chapter we first present the mesh decomposition employed and the corresponding load balancing achieved. Next we outline the boundary exchange strategy which is required in the context of any type of scheme, and then we describe the reconstruction exchange strategy which is required only by high-order schemes which is the most expensive process in terms of computing power. Finally we present results from various schemes for a fixed test-problem obtained from the ASTRAL-HPC at Cranfield University where the performance of the developed algorithms is assessed. The Message Passing Interface API is used for the parallelisation of the UCNS3D solver since it is suited for High Performance Computing Facilities with distributed memory architecture.

5.2 Mesh decomposition

The mesh decomposition process is the preprocessor step which is one of the most essential elements for an equal load balance between processes in terms of memory requirements and

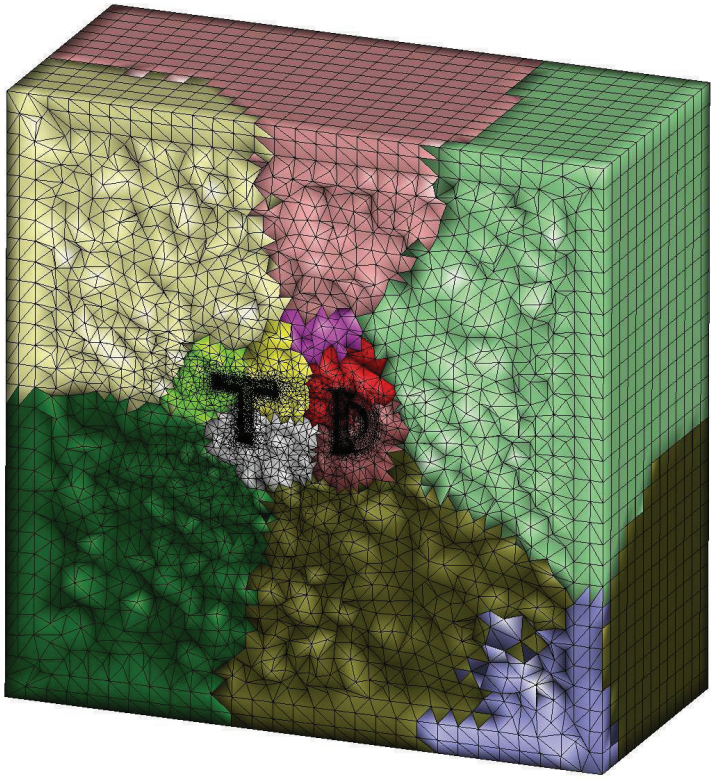


Figure 5.1: Mesh decomposition of complicated geometry

communications between processes. The software package used for partitioning uniform unstructured meshes and even hybrid meshes is the METIS software package [35]. The process chosen for decomposition is to convert the mesh into a nodal graph rather than using a dual graph in order to derive a partitioning of the nodes. The reason for doing that is that by using a nodal graph we are not limited by uniform unstructured meshes and therefore hybrid unstructured meshes can also be decomposed. A typical example of a hybrid mesh decomposition is illustrated in Figure 5.1 on page 105.

The load balancing achieved with this software package for the meshes used ranges between 1.00 and 1.07 for any type of unstructured mesh. The performance of the METIS software package has produced highly-efficient unstructured solvers with very high-order schemes [49]. ParMETIS could also have been used but since optimisation of the mesh partitioning was not a priority it was not employed, however for future optimisation of the mesh partitioning it is highly desirable to employ ParMETIS. Although the load balancing between processes could reach the ideal value of 1.00, that does not imply that the actual load balance between processes would be equal. The reason for that is that although each process has roughly the same number of elements and vertices, it is other parameters that define the balance between the processes such as the position of the physical boundaries of the computational domain, the boundaries between the processes and the most important the type of the scheme employed.

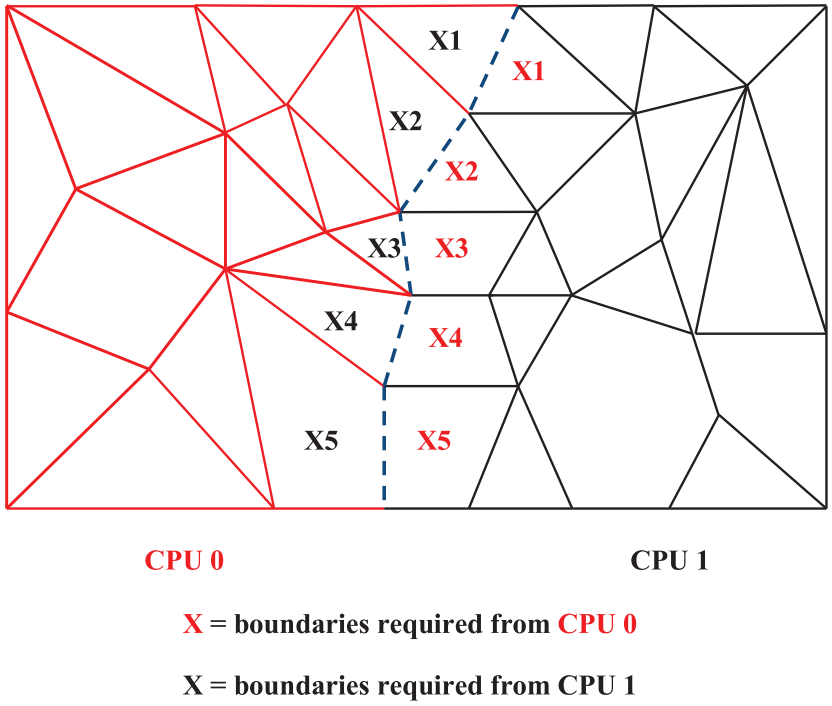


Figure 5.2: Typical example of boundary interface between processes

5.3 Boundary exchange strategy

Having decomposed the mesh into a number of blocks every block is assigned to a specific process. In this section we only describe the strategy for exchanging reconstructed values at the intercell faces of the boundaries between processes in order to compute the numerical flux through an approximate Riemann solver. To illustrate the basic ingredients of the algorithm consider a two dimensional hybrid mesh decomposed for two processes 0 and 1 as illustrated in Figure 5.2 on page 106.

Every process requires from the other one the reconstructed solution at the intercell boundaries with another process. Since only conforming unstructured meshes are considered in this study the number of boundaries required between processes is the same. This means that since CPU 0 requires the reconstructed solution at intercell faces of 5 elements from CPU1 then CPU1 would also require the reconstructed solution from 5 elements of CPU0. For first-order scheme only one value per variable is required at the intercell face. On the other hand for higher-order schemes the reconstructed solution is required at every Gaussian quadrature point.

The basic steps for exchanging the reconstructed solution at every intercell face are as follows:

1. For every process determine which elements are needed from which process

2. Allocate the appropriate memory for receiving the solution at the boundaries from other

processes

3. Allocate the appropriate memory for sending the required solution at the boundaries to other processes

4. Set pointers to the memory allocated for sending boundary information pointing to the boundary values in the current process required from the other processes

5. Set pointers at the boundary elements of the current process that require information from other processes to point to the memory allocated for receiving this information

6. Perform a combined MPI_SENDRECV to send the boundary information required from other processes and receive the required boundary information from the current process by using the pointers assigned at step 4

7. Use the pointers assigned at step 5 to retrieve the values at the intercell boundaries that have been received

The steps 1 to 5 are performed only before the computations and steps 6 and 7 are performed every time that the information at the intercell boundaries between processes is required. The reason for choosing a combined MPI_SENDRECV is that as mentioned earlier the number of boundaries that need to be send and the number of boundaries required are the same therefore by using only one call the sending and receive takes place at once and the MPI communication subsystem makes sure that the possibility for deadlock occurrence is relatively small.

5.4 Reconstruction-related exchange strategy

As it has been described higher-order schemes require various stencils consisting of a number of elements in order to perform a high-order interpolation of cell averages. The challenges imposed by this requirement are, firstly how to recursively construct the stencils when some elements belong to other processes and secondly how to exchange the information required to solve the linear system (2.30). The strategy for facing the first challenge is described in Algorithm 3.

Having constructed the stencils the coefficients of the resulted linear symmetric matrix A in the linear system (2.30) are stored for each element during the preprocessing stage of the calculation. Therefore the only information that needs to be exchanges during the stage of calculation is the cell averages of the elements of the stencils that belong to other processes. However there is an important issue when dealing with complicated domains and very high-order schemes that must be taken into account. This issue is that the number of elements required from each process for the reconstruction stencils is completely arbitrary and by arbitrary we mean that different number of elements are required between processes as illustrated in Figure 5.3 on page 109. In some extreme cases some processes require elements from other processes that do not require any elements back. This usually occurs when WENO schemes are used that they have directionality conditions that must be satisfied and

Algorithm 3 Parallel Stencil Construction Algorithm

1. For each cell i in the mesh we want to construct a set of elements S (stencil) consisting of $N(S)$elements
2. With $c = 1, 2, ...N$ being the index of the numbering of the elements in the stencil
3. $S_1 = i$, the considered cell i is always the first element in the stencil $c = 1$
4. Recursively start adding the direct side neighbours of each element
   (a) In the case that an element belongs to another process send a requirement to receive the direct side neighbours of this element from the other process together with the coordinates of the elements
   (b) Check if any element is required from other processes that belongs to the current process and send the element and its direct side neighbours to them together with their coordinates
5. Check which of the elements already exist in the stencil
6. Check if the candidate elements satisfy the directionality conditions(for directional stencils only)
7. In the case that the candidate elements do not belong in the set S and satisfy the directionality condition store them in the next available memory location
8. Repeat steps 4 to 8 until N number of elements have been assigned to each stencil
9. Repeat steps 1 to 8 until the stencils for the last element have been constructed
10. Send to every process that the current process has constructed the stencils for all the cells
11. Receive from other processes their status(if they have finished constructing the stencils)
    (a) If any of the other processes has not finished
        i. Check if any element is required from other processes that belongs to the current process and send the element and its direct side neighbours to them together with their coordinates
    (b) If all of the other processes have finished then exit

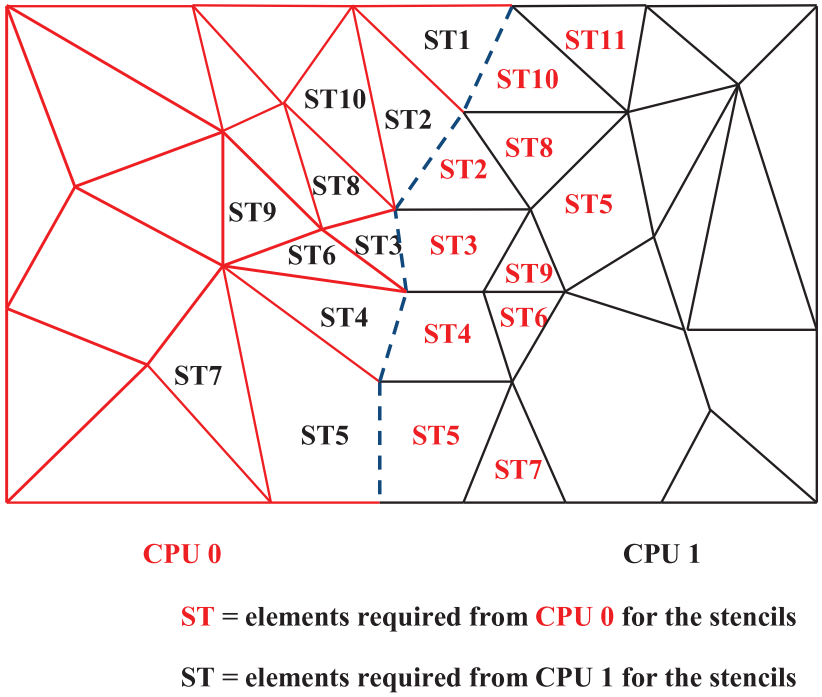


Figure 5.3: Typical example of stencil elements requirements between processes

in the presence of boundaries the stencils of some elements might extend to more than just one block.

The basic steps for exchanging the cell averages for stencils elements between processes are as follows:

1. For each cell in the domain find the stencil elements required from other processes
2. Having found the total number of stencil elements required from other processes, sort the total number of elements required per process by making sure that common ones are removed (stencil elements required from more than one cells in the current process are only stored once)
3. Allocate appropriate memory for each process the current process requires stencil elements
4. Send to the processes that the current process requires stencil elements the number of the elements required and which elements are required
5. Check if any process requires stencil elements from the current one
6. Allocate appropriate memory for each process that requires stencil elements from the current one

7. Set pointers to the memory allocated for sending stencil elements cell averages to other processes to point to the cell averages of these elements in the current process

8. Set pointers to the linear systems of the current process that require stencil elements cell averages from other processes to point to the memory allocated for receiving this information

9. Perform a non-blocking MPI_ISEND to all processes that require the cell averages from stencil elements within the current process

10. Perform a non-blocking MPI_IRECV from all processes that the current process requires cell averages from their stencil elements

11. Use the pointers assigned at step 7 to retrieve the cell averages of the stencil elements required for the solution of the linear system (2.30)

Steps 1 to 8 are performed only once before the computations and steps 9 to 11 are performed every time that cell averages from stencil elements from other processes are required. In the flow charts in Figure 5.4 on page 111 and Figure 5.5 on page 112 the differences between a serial and the parallel process involving the algorithms described can be viewed.

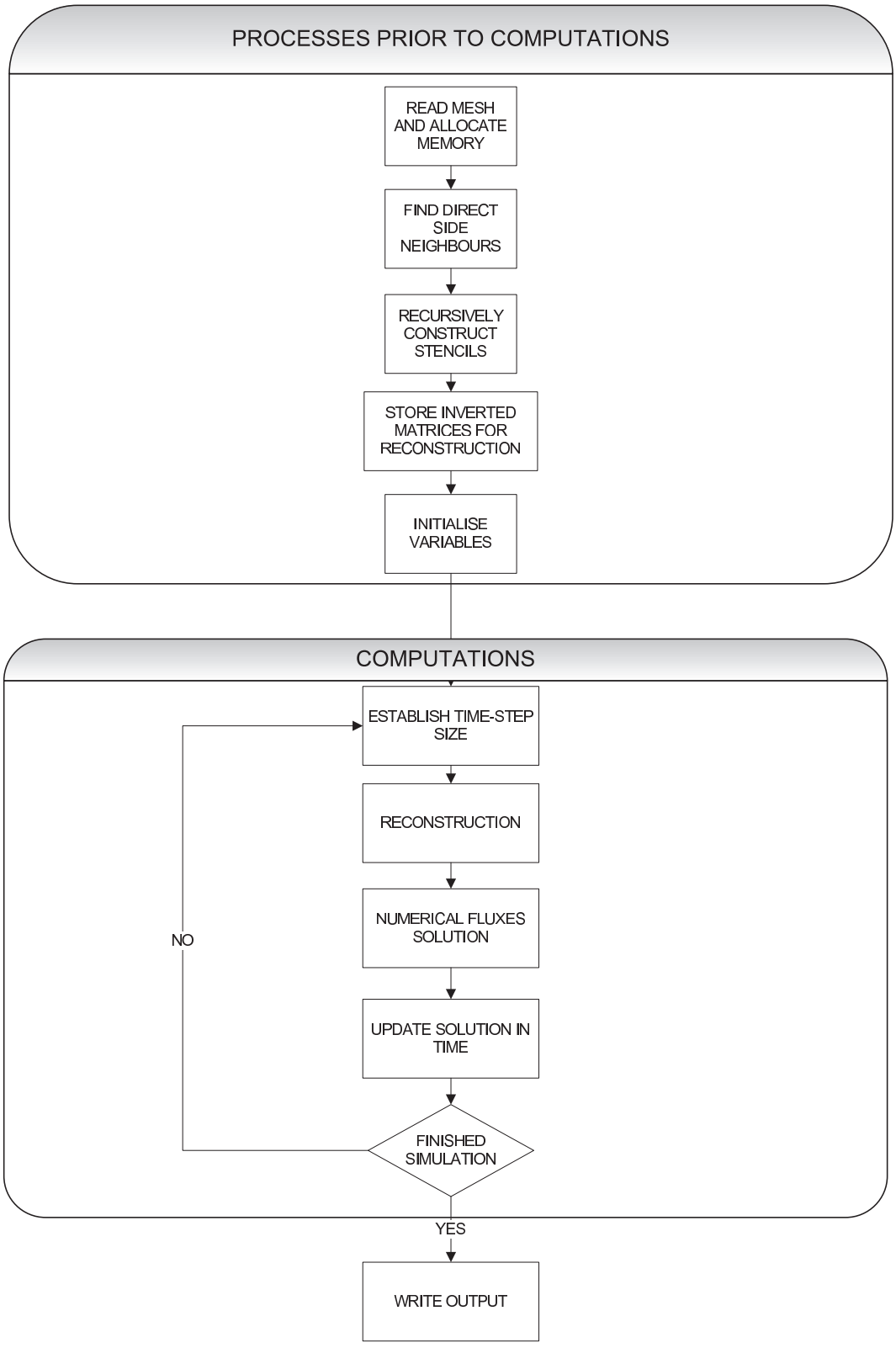


Figure 5.4: Serial Process Flow Chart

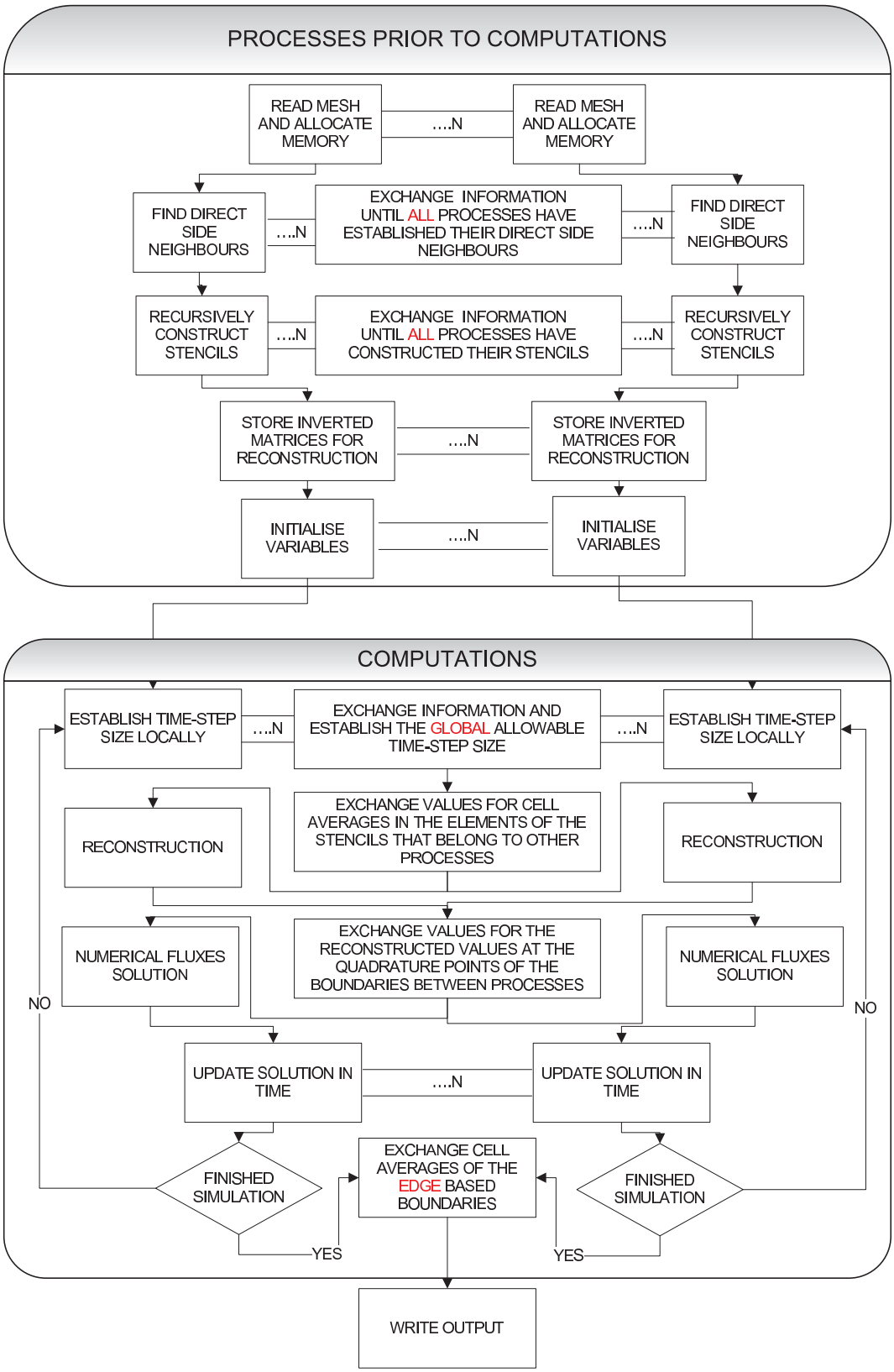
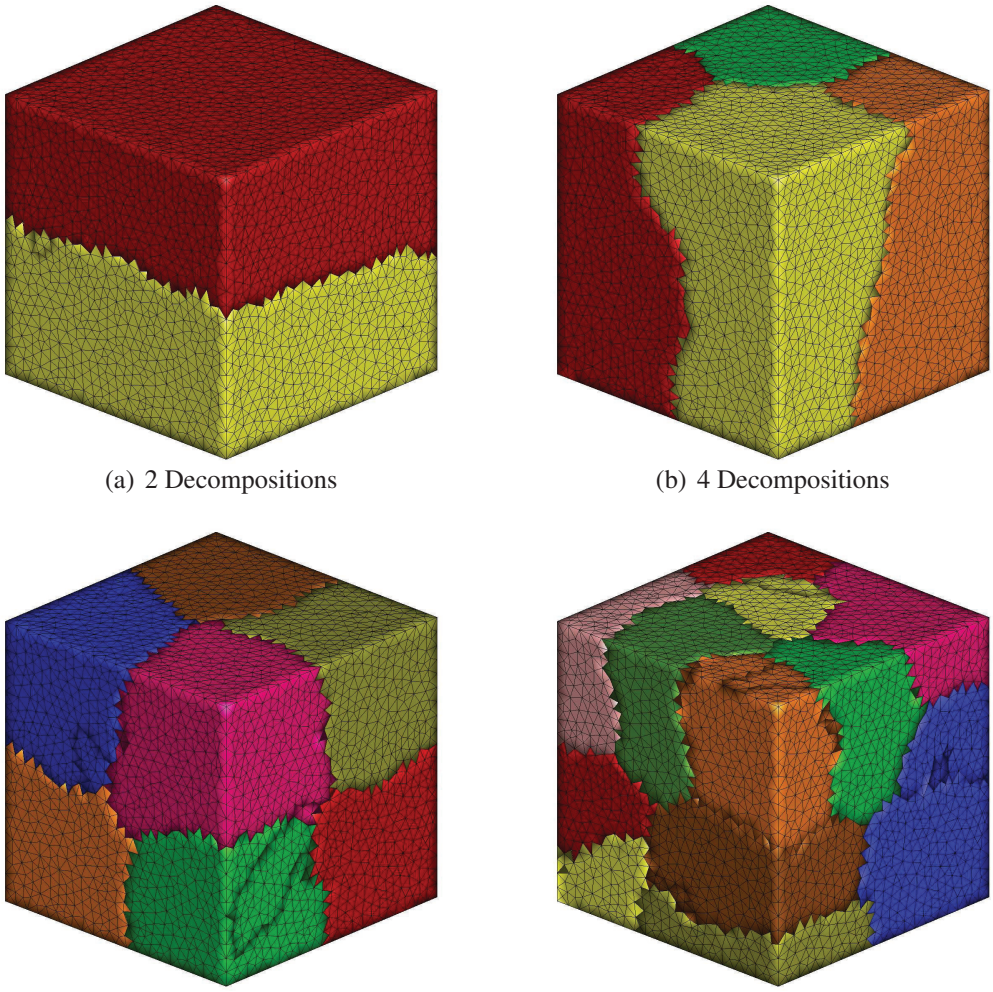


Figure 5.5: Parallel Processes Flow Chart



(a) 2 Decompositions (b) 4 Decompositions

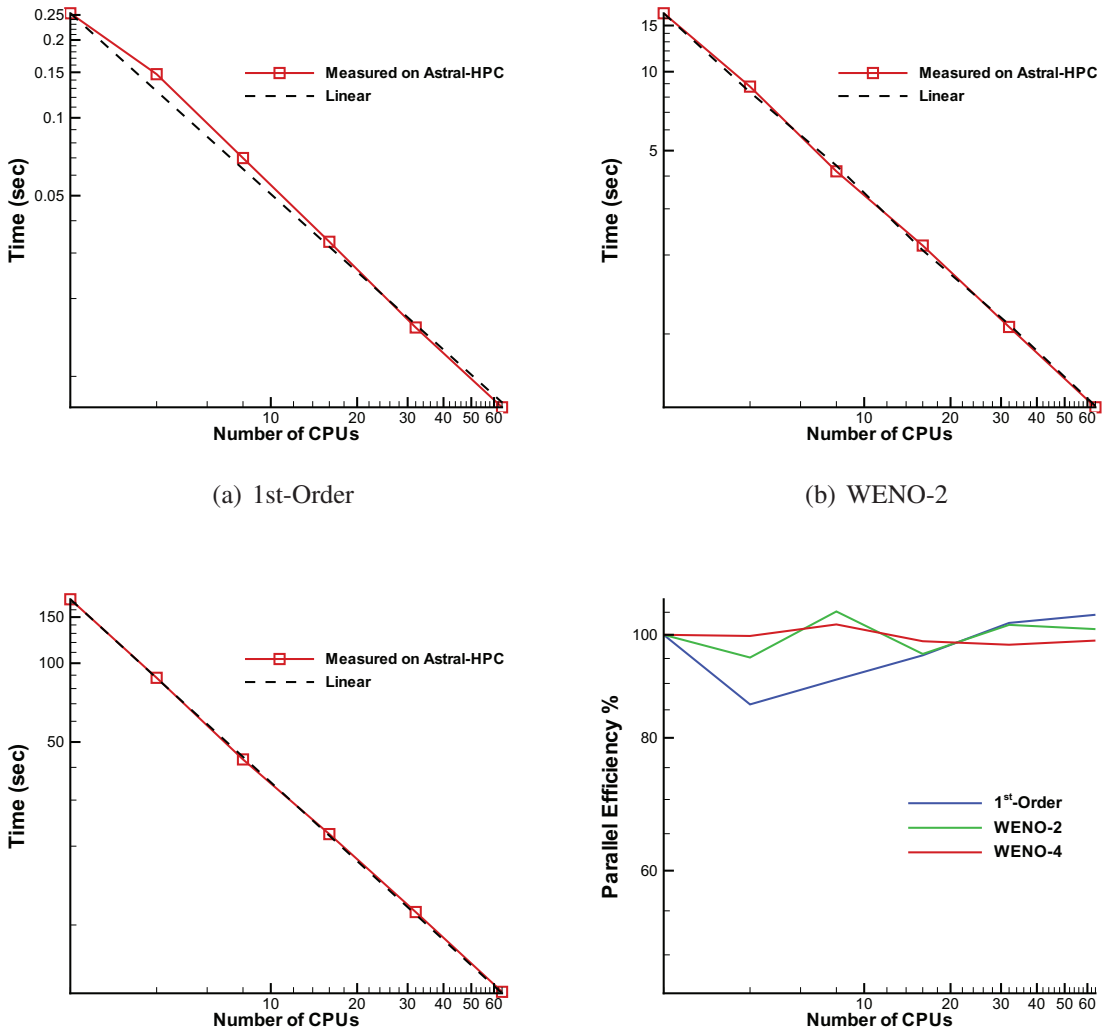
(c) 8 Decompositions (d) 16 Decompositions

Figure 5.6: Mesh decompositions employed for the study of the parallel performance of the developed schemes.

5.5 Parallel efficiency study

In order to assess the parallel performance of the developed algorithms, we run a three-dimensional problem using the various schemes on an unstructured tetrahedral mesh with 84771 elements. The problem is the same as used for the explosion test problem presented in Section 4.2.3 corresponding to the $N = 20$. This test problem is computed on 2,4,8,16,32, and 64 CPUs and the mesh decomposition used for this study are in shown in Figure 5.6 on page 113. In Figure 5.7 on page 114 the MPI speedup and parallel efficiency for various schemes is illustrated.

The MPI speedup and parallel efficiency is measured in terms of the time required for advancing the solution in the whole computational domain for one timestep without taking into account the pre-computations processes such as the stencil construction since they are



(a) 1st-Order (b) WENO-2

(c) WENO-4 (d) Comparison of schemes

Figure 5.7: MPI speedup and Parallel efficiency measured on the ASTRAL-HPC Cranfield University, using various schemes on an unstructured tetrahedral mesh in 3D.

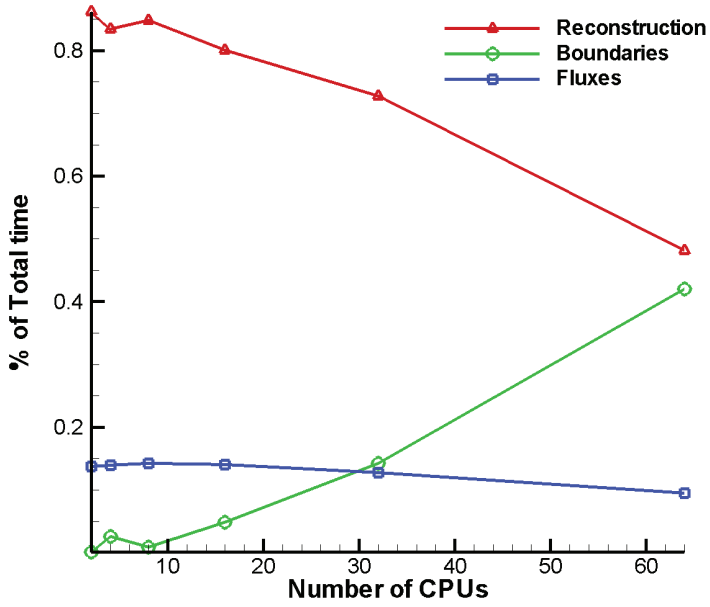


Figure 5.8: Individual processes contribution to the total time taken for the solution advancement in the WENO-4 scheme

only done once. We notice that all the schemes achieve almost linear speedup. The 1st-order scheme which is the cheapest in terms of computing resources seems to perform worse than the more expensive WENO-2 and WENO-4 schemes in terms of parallel efficiency under 20 CPUs. On the other hand the WENO-2 and WENO-4 schemes seem to scale up quite well even at few CPUs. In order to have a better insight of the cost of each of the processes we investigated the percentage of the total time for the update of the solution that it is taken in the reconstruction process, the boundary exchange(including stencil elements cell averages exchange) process and the fluxes for the WENO-4 scheme. The finding are are illustrated in Figure 5.8 on page 115.This behaviour can be justified by the fact that for the WENO-4 scheme the time taken for the reconstruction scales down similar to the way that the boundary exchange scales up beyond 16 CPUs. Both of this processes are the most expensive in terms of computing power. The superlinear acceleration observed can be justified by the fact that the computations part is taking advantage of the cache memory, since there are less computation data that each CPU is handling therefore the CPU does not address the main memory so frequently as it happens when fewer CPUs are used. This is tranlated that the data can easier fit into CPU cache memory which has lower access time therefore for this test problem we notice that this superlinear acceleration is due to the more efficient usage of cache memory.

5.6 Conclusions

We have materialised a parallel implementation of the UCNS3D code suited for large-scale CFD simulations on any type of unstructured meshes. It has been documented [41, 6] that the arbitrariness of the unstructured meshes limits the parallel efficiency of unstructured solver we have found this to be true but only for lower-order schemes. For the very high-order finite volume schemes developed the most expensive process in terms of memory and computing power is the reconstruction process itself rather than the exchange of boundary and cell averages of the stencil elements between processes.

There is always a trade-off between the operations that are computed every time and the data that is stored in memory and can be retrieved that any developer should take into account. We have decided to store in memory the most computing intensive parts such as the stencils, the symmetric linear matrices and other geometrical information which for three-dimensional domains it is expensive to compute every time.

Our parallel implementation takes advantage of the fact that the linear symmetric matrix for the least-square reconstruction is stored and the only information required to be exchanged every time is only the cell averages of stencil elements between processes rather than any topological information. Therefore our parallel implementation is characterised by an almost linear speedup and a parallel efficiency higher than 95% for the high-order schemes. We expect that even higher-order schemes such as WENO-5, LINEAR-5 would be even more efficient since the reconstruction process is becoming even more expensive.

Chapter 6

Conclusions & Future Work

This thesis has focused in the development of three-dimensional very high-order finite volume schemes that can be applied to any type of unstructured meshes. The main results can be summarised as follows:

1. A new set of schemes of very high-order of accuracy which are universal in the sense that they can be applied to any type of unstructured mesh has been presented. They can be viewed as a very high-order generalisation of unsplit multidimensional schemes. Three different categories of schemes have been developed a linear type, a TVD type and a WENO type.

2. The crucial process for achieving high-order of accuracy is a reconstruction process that can combine elements of different shapes by removing any scaling problems through a mapping of the problem from physical domain to a computational domain.

3. The linear schemes make use only of the central stencil for the reconstruction process and are mostly suited for problems with smooth solutions.

4. The TVD type makes use of the central stencil for reconstruction process but restricts the reconstructed solution to lie within the minimum and maximum values present in the direct side neighbours of each cell, and is a cost-efficient alternative of WENO suited for problems that contain discontinuous solutions.

5. The WENO schemes make use of a series of stencils and the reconstruction polynomials from every one of them are combined in a non-linear way by taking into account the smoothness of information in each one of them. The implementation of the WENO scheme for the Euler equations has the elegant feature that reconstruction is done with respect to the characteristic variables rather than the conserved variables.

6. Convergence studies of the schemes for the model 3D linear advection equation and the 3D Euler equations have been performed. All the schemes presented up to 5^{th}-order achieve their theoretical order of accuracy for any type of unstructured meshes. This shows that the schemes are universal and achieve their theoretical order of accuracy independent of the mesh used.

7. Assessment of the non-oscillatory properties of the TVD and WENO schemes has been performed. Both TVD and WENO schemes do not produce any spurious oscillations around discontinuities with the WENO schemes resolving the solution profile much sharper than the TVD schemes.

8. For problems containing discontinuities we have found that the directionality criteria when choosing admissible directional stencils and the linear weight assigned to the central stencil have a great impact on the non-oscillatory performance of the WENO schemes.

9. Qualitative and quantitative analysis of the capabilities of the developed schemes to handle complicated shapes and configurations with mixed-element type of meshes has also been performed. Both TVD and WENO schemes do not produce any spurious oscillations for strong discontinuous problems involving complicated geometries, with the WENO scheme being able to resolve more complicated flow structures.

10. A new 3D hybrid unstructured flow solver (UCNS3D) has been developed employing the very high-order schemes constructed. The solver has been parallelised by using highly sophisticated algorithms, mesh decomposition software tools and by using the MPI API. The results demonstrate that the higher the order of the schemes the greater the parallel efficiency.

Interesting future research directions worth considering in the context of very high-order schemes on unstructured meshes are:

1. Development of a Hybrid WENO scheme by combining the schemes constructed here and the WENO schemes of [78, 32] where lower order polynomials are combined to achieve higher-order of accuracy. The manner at which those schemes could be combined could be that the constructed WENO scheme could be used in discontinuous regions of the flow and the the WENO schemes of [78, 32] in smooth regions of the flows.

2. Employment of adaptive mesh refinement techniques which have been successfully applied to a series of challenging and complicated problems [72, 58, 47, 52, 10] in the context of very high-order schemes in the finite volume framework. Although one of the main challenges is the utilisation of the stencil construction algorithm at various stages that the mesh is refined, the overall computational efficiency would be improved. Adaptive mesh refinement is a desirable technique so that the mesh is refined at critical regions of the flow where mesh resolution is important such as close to discontinuities so that the whole flow features would be resolved with a greater detail.

3. Inclusion of polyhedral unstructured elements currently offered by state-of-the-art mesh generation software packages for the constructed high-order schemes. The efficiency of this high node count elements in terms of geometry representation, and computational resources is a desirable feature of any high-order scheme.

4. Extension of the application of the schemes for other computationally challenging tasks in science and engineering.

Bibliography

[1] R. Abgrall. On Essentially Non-Oscillatory Schemes on Unstructured Meshes: Analysis and Implementation. *Journal of Computational Physics*, 114(1):45–58, 1994.

[2] Barnewitz H.-Schwarten H. Becker K. Heinrich R. Roll B. Galle M. Kroll N. Gerhold Th. Schwamborn D. Franke M. Aumann, P. Megaflow: Parallel complete aircraft cfd. *Parallel Computing*, 27(4):415–440, 2001.

[3] A. Bagabir and D. Drikakis. Numerical Experiments Using High Resolution Schemes for Unsteady, Inviscid, Compressible Flows. *Computer Methods in Applied Mechanics and Engineering*, 2004.

[4] D. S. Balsara and C. W Shu. Monotonicity Preserving Weighted Essentially Non-Oscillatory Schemes with Increasingly High Order of Accuracy. *Journal of Computational Physics*, 160(2):405–452, 2000.

[5] T. Barth and P. Frederickson. Higher Order Solution of the Euler Equations on Unstructured Grids using Quadratic Reconstruction. *AIAA paper no.90-0013, 28th Aerospace Sciences Meeting*, 1990.

[6] Carpentieri B.-Sosonkina M. Bonfiglioli, A. Eulfs: A parallel cfd code for the simulation of euler and navier-stokes problems on unstructured grids. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 4699 LNCS:676–685, 2007.

[7] T. T. Bui. A parallel, finite-volume algorithm for large-eddy simulation of turbulent flows. *Computers and Fluids*, 29(8):877–915, 2000.

[8] J. Cabello. Parallel Explicit Unstructured Grid Solvers on Distributed Memory Computers. *Advances in Engineering Software*, 26(3):189–200, 1996.

[9] J. C. Carrete, D. J. Mavriplis, Y. Kallinderis, R. Lohner, and M. J. Aftosmis. Algorithms and Data Structures for Structured and Unstructured Grid Generation. *Selected Special Topics From VKI Lecture Series*, 1998.

[10] A. J. Chen and Y. Kallinderis. Adaptive Hybrid (Prismatic-Tetrahedral) Grids for Incompressible Flows. *International Journal for Numerical Methods in Fluids*, 26(9): 1085–1105, 1998.

[11] B. Cockburn. Devising Discontinuous Galerkin Methods for Non-Linear Hyperbolic Conservation Laws. *Journal of Computational and Applied Mathematics*, 128(1-2): 187–204, 2001.

[12] B. Cockburn. Discontinuous galerkin methods. *Zeitschrift Fur Angewandte Mathematik Und Mechanik*, 83(11):731–754, 2003.

[13] B. Cockburn and C. W Shu. The Local Discontinuous Galerkin Method for Time-Dependent Convection-Diffusion Systems. *SIAM Journal on Numerical Analysis*, 35 (6):2440–2463, 1998.

[14] B. Cockburn and C. W Shu. The Runge-Kutta Discontinuous Galerkin Method for Conservation Laws V: Multidimensional Systems. *Journal of Computational Physics*, 141(2):199–224, 1998.

[15] B. Cockburn and C. W Shu. Runge-Kutta Discontinuous Galerkin Methods for Convection-Dominated Problems. *Journal of Scientific Computing*, 16(3):173–261, 2001.

[16] P. Colli-Franzone, L. F. Pavarino, and B. Taccardi. A parallel solver for anisotropic cardiac models. volume 30, pages 781–784, 2003.

[17] L. Cueto-Felgueroso, I. Colominas, J. Fe, F. Navarrina, and M. Casteleiro. High-order finite volume schemes on unstructured grids using moving least-squares reconstruction. application to shallow water dynamics. *International Journal for Numerical Methods in Engineering*, 65(3):295–331, 2006.

[18] L. Cueto-Felgueroso, I. Colominas, X. Nogueira, F. Navarrina, and M. Casteleiro. Finite volume solvers and moving least-squares approximations for the compressible navier-stokes equations on unstructured grids. *Computer Methods in Applied Mechanics and Engineering*, 196(45-48):4712–4736, 2007.

[19] M. S. Darwish and F. Moukalled. Tvd schemes for unstructured grids. *International Journal of Heat and Mass Transfer*, 46(4):599–611, 2003.

[20] Dimitris Drikakis and William Rider. *High-Resolution Methods for Imcompressible and Low-Speed Flows*. Springer-Verlag Berlin Heidelberg, 2005.

[21] M. Dumbser and M. Kaiser. An Arbitrary High-Order Discontinuous Galerkin Method for Elastic Waves on Unstructured Meshes - ii. The Three-Dimensional Isotropic Case. *Geophysical Journal International*, 167(1):319–336, 2006.

[22] M. Dumbser and M. Kaser. Arbitrary High Order Non-Oscillatory Finite Volume Schemes on Unstructured Meshes for Linear Hyperbolic Systems. *Journal of Computational Physics*, 221(2):693–723, 2007.

[23] O. Friedrich. Weighted Essentially Non-Oscillatory Schemes for the Interpolation of Mean Values on Unstructured grids. *Journal of Computational Physics*, 144(1):194–212, 1998.

[24] J. Furst. A weighted least square scheme for compressible flows. *Flow, Turbulence and Combustion*, 76(4):331–342, 2006.

[25] S. K. Godunov. A Difference Method for Numerical Calculation of Discontinuous Equations of Hydrodynamics. *Mat. Sb.*, 47(3):271–306, 1959.

[26] C. Ollivier Gooch and M. Van Altena. A High-Order-Accurate Unstructured Mesh Finite-Volume Scheme for the Advection-Diffusion Equation. *Journal of Computational Physics*, 181(2):729–752, 2002.

[27] D. Gray, J. Summary Report on Aerodynamic Characteristics of Standard Models HB-1 and HB-2. *Technical Report AEDC-TDR*, 64-137, 1964.

[28] L. Grinberg and G. E. Karniadakis. Hierarchical Spectral Basis and Galerkin Formulation Using Barycentric Quadrature Grids in Triangular Elements. *Journal of Engineering Mathematics*, 56(3):289–306, 2006.

[29] T. Haga, N. Ohnishi, K. Sawada, and A. Masunaga. Spectral volume computation of flowfield in aerospace application using earth simulator. volume 1, pages 82–90, 2006.

[30] A. Harten, B. Engquist, S. Osher, and S. Chakravarthy. Uniformly High Order Accurate Essentially Non-Oscillatory Schemes iii. *J. Comput. Phys.*, 71(2):231–303, 1987.

[31] A. Haselbacher. A weno reconstruction algorithm for unstructured grids based on explicit stencil construction. pages 3369–3378, 2005.

[32] C. Hu and C. W Shu. Weighted Essentially Non-Oscillatory Schemes on Triangular Meshes. *Journal of Computational Physics*, 150(1):97–127, 1999.

[33] P. A. Jayantha and I. W. Turner. Second order control-volume finite-element least-squares strategy for simulating diffusion in strongly anisotropic media. *Journal of Computational Mathematics*, 23(1):1–16, 2005.

[34] G. S Jiang and C. W Shu. Efficient Implementation of Weighted ENO Schemes. *Journal of Computational Physics*, 126(1):202–228, 1996.

[35] Kumar V. Karypis, G. Multilevel k-way partitioning scheme for irregular graphs. *J. Parallel Distrib. Comput.*, 48:96–129, 1998.

[36] A. Kaser and M. Iske. Ader Schemes for the Solution of Conservation Laws on Adaptive Triangulations. *Mathematical Methods and Modeling in Hydrocarbon Exploration and Production, Series: Mathematics in Industry, Springer-Verlag*, 7, 2004.

[37] R. M. Kirby, T. C. Warburton, I. Lomtev, and G. E. Karniadakis. Discontinuous Galerkin Spectral/HP Method on Hybrid Grids. *Applied Numerical Mathematics*, 33(1):393–405, 2000.

[38] V. P. Kolgan. Application of the Minimum-Derivative Principle in the Construction of Finite-Difference Schemes for Numerical Analysis of Discontinuous Solutions in Gas Dynamics. *Transactions of the Central Aerodynamics Institute*, 3(6):68–77, 1972.

[39] V. P. Kolgan. Finite-Difference Schemes for Computatuon of Three Dimensional Solutions of Gas Dynamics and Calculation of a Flow over a Body under an Angle of Attack. *Transactions of the Central Aerodynamics Institute*, 6(2):1–6, 1975.

[40] E. Kreyszig. *Advanced Engineering Mathematics (9th Edition)*. 2005.

[41] Rossow C.-C. Becker K. Thiele F. Kroll, N. The megaflow project. *Aerospace Science and Technology*, 4(4):223–237, 2000.

[42] D. Levy, S. Nayak, C. . Shu, and Y. . Zhang. Central weno schemes for hamilton-jacobi equations on triangular meshes. *SIAM Journal of Scientific Computing*, 28(6): 2229–2247, 2006.

[43] D. C. Liu. Unstructured grid solution scheme for hybrid meshes. *Zhongguo Hangkong Taikong Xuehui Huikan/Transactions of the Aeronautical and Astronautical Society of the Republic of China*, 35(1):73–80, 2003.

[44] X. Liu, S. Osher, and T. Chan. Weighted Essentially Non-Oscillatory Schemes. *Journal of Computational Physics*, 115:200–212, 1994.

[45] Y. Liu, M. Vinokur, and Z. J. Wang. Spectral (Finite) Volume Method for Conservation Laws on Unstructured Grids V: Extension to Three-Dimensional Systems. *Journal of Computational Physics*, 212(2):454–472, 2006.

[46] R. Lohner and M. Galle. Minimization of Indirect Addressing for Edge-Based Field Solvers. *Communications in Numerical Methods in Engineering*, 18(5):335–343, 2002.

[47] R. Lohner, J. D. Baum, E. Mestreau, D. Sharov, C. Charman, and D. Pelessone. Adaptive Embedded Unstructured Grid Methods. *International Journal for Numerical Methods in Engineering*, 60(3):641–660, 2004.

[48] H. Luo, J. D. Baum, and R. Lo?hner. A hermite weno-based limiter for discontinuous galerkin method on unstructured grids. volume 9, pages 6148–6175, 2007.

[49] V. A. Titarev E. F. Toro M. Dumbser, M. Kaser. Quadrature-Free Non-Oscillatory Finite Volume Schemes on Unstructured Meshes for Nonlinear Hyperbolic Systems. *Journal of Computational Physics*, 226(1):204–243, 2007.

[50] D. J. Mavriplis. Unstructured Grid Techniques. *Annual Review of Fluid Mechanics*, 29: 473–514, 1997.

[51] D. J. Mavriplis. Viscous Flow Analysis Using a Parallel Unstructured Multigrid Solver. *AIAA journal*, 38(11):2067–2076, 2000.

[52] D. J. Mavriplis. Adaptive Meshing Techniques for Viscous Flow Calculations on Mixed Element Unstructured Meshes. *International Journal for Numerical Methods in Fluids*, 34(2):93–111, 2000.

[53] D. J. Mavriplis. Parallel Performance Investigations of an Unstructured Mesh Navier-Stokes Solver. *International Journal of High Performance Computing Applications*, 16 (4):395–407, 2002.

[54] D. J. Mavriplis. Grid Resolution Study of a Drag Prediction Workshop Configuration using the NSU3D Unstructured Mesh Solver, 2005.

[55] D. J. Mavriplis, M. J. Aftosmis, and M. Berger. High Resolution Aerospace Applications using the NASA Columbia Supercomputer. *International Journal of High Performance Computing Applications*, 21(1):106–126, 2007.

[56] A. Nejat and C. Ollivier Gooch. A high-Order Sccurate Unstructured GMRES Algorithm for Inviscid Compressible Flows. In *17th AIAA Computational Flow Dynamics Conference; Toronto, Canada*, 6-9 June 2005 2005.

[57] C. F. Ollivier-Gooch. Quasi-eno schemes for unstructured meshes based on unlimited data-dependent least-squares reconstruction. *Journal of Computational Physics*, 133 (1):6–17, 1997.

[58] Vahdati M. Morgan K. Zienkiewicz O.C. Peraire, J. Adaptive remeshing for compressible flow computations. *Journal of Computational Physics*, 72(2):449–466, 1987.

[59] J. Qiu and C. W Shu. Hermite WENO Schemes and their Application as Limiters for Runge-Kutta Discontinuous Galerkin Method: One Dimensional Case. *J. Comput. Phys.*, 193:115–135, 2003.

[60] J. Qiu and C. W. Shu. Finite Difference Schemes with Lax-Wendroff Type Time Discretization. *SIAM Journal of Scientific Computing*, (24):2185–2198, 2003.

[61] J. Qiu and C. W Shu. Hermite WENO Schemes and their Application as Limiters for Runge-Kutta Discontinuous Galerkin Method II: Two Dimensional Case. *Computers and Fluids*, 34(6):642–663, 2005.

[62] J. Qiu and C. W Shu. Runge-Kutta Discontinuous Galerkin Method using WENO Limiters. *SIAM Journal of Scientific Computing*, 26(3):907–929, 2005.

[63] Y. Shi and C. . Liang. The finite-volume time-domain algorithm using least square method in solving maxwell's equations. *Journal of Computational Physics*, 226(2): 1444–1457, 2007.

[64] A. Stroud. Approximate Calculation of Multiple Integrals. *Prentice-Hall Inc., Englewood Cliffs, NJ*, 1971.

[65] Y. Sun, Z. J. Wang, and Y. Liu. Spectral (Finite) Volume Method for Conservation Laws on Unstructured Grids VI: Extension to Viscous Flow. *Journal of Computational Physics*, 215(1):41–58, 2006.

[66] Y. Sun, Z. J. Wang, and Y. Liu. High-order multidomain spectral difference method for the navier-stokes equations on unstructured hexahedral grids. *Communications in Computational Physics*, 2(2):310–333, 2007.

[67] N. Tillaeva. A Generalization of the Modified Godunov Scheme to Arbitrary Unstructured Meshes. *Transactions of the Central Aerodynamics Institute*, 17(2):18–26, 1986.

[68] V. A. Titarev and E. F. Toro. ADER Schemes for Three-Dimensional Non-Linear Hyperbolic Systems. *Journal of Computational Physics*, 204(2):715–736, 2005.

[69] V. A. Titarev, E. I. Romenski, and E. Toro. Musta-Type Upwind -Fluxes for Non-Linear Elasticity. *International Journal of Numerical Methods in Engineering*, 73:897–926, 2008.

[70] Eleuterio F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics - A Practical Introduction*. Berlin: Springer-Verlag, 1997.

[71] E. F. Toro V. A. Titarev. Finite-Volume WENO Schemes for Three-Dimensional Conservation Laws. *Journal of Computational Physics*, 201(1):238–260, 2004.

[72] P. Wang, Z. . Zhu, S. . Tuo, and X. . Yin. 3d solutions of the euler equations on the adaptive tetrahedral meshes. *Hangkong Xuebao/Acta Aeronautica et Astronautica Sinica*, 22 (6):495–499, 2001.

[73] Z. J. Wang. Spectral (finite) volume method for conservation laws on unstructured grids. basic formulation. *Journal of Computational Physics*, 178(1):210–251, 2002.

[74] Z. J. Wang, Y. Liu, G. May, and A. Jameson. Spectral difference method for unstructured grids ii: Extension to the euler equations. *Journal of Scientific Computing*, 32(1): 45–71, 2007.

[75] W. R. Wolf and J. L. F. Azevedo. High-order eno and weno schemes for unstructured grids. *International Journal for Numerical Methods in Fluids*, 55(10):917–943, 2007.

[76] A. Wolkov, Ch Hirsch, and B. Leonard. Discontinuous galerkin method on unstructured hexahedral grids for 3d euler and navier-stokes equations. volume 1, pages 678–697, 2007.

[77] S. Zhang and K. Jain. Least square data reconstruction approach for unstructured grid. volume 1 SYMPOSIA, pages 39–46, 2007.

[78] Y. T. Zhang and C. W. Shu. Third Order WENO Scheme on Three Dimensional Tetrahedral Meshes. *Communications in Computational Physics*, 5(2-4):836–848, 2008.

[79] X. Zhao and S. Zhang. A parallelized, dynamic solution adaptive, multi-grid, overset chimera unstructured grid solver. volume 1 SYPMOSIA, pages 1293–1302, 2006.