

IMPROVEMENT OF NESTED CARTESIAN FINITE DIFFERENCE GRID SOLVERS

by

Alejandro Agustin Figueroa
A Dissertation
Submitted to the
Graduate Faculty
of
George Mason University
In Partial Fulfillment of
The Requirements for the Degree
of
Doctor of Philosophy
Physics

Committee:

_____ Dr. Rainald Löhner, Committee Chair
_____ Dr. Fernando Camelli, Committee Member
_____ Dr. Juan R. Cebal, Committee Member
_____ Dr. Chi Yang, Committee Member
_____ Dr. Paul So, Department Chair
_____ Dr. Donna M. Fox, Associate Dean,
Office of Student Affairs & Special
Programs, College of Science
_____ Dr. Ali Andalibi, Dean, College of Science

Date: _____ Fall Semester 2019
George Mason University
Fairfax, VA

Improvement of Nested Cartesian Finite Difference Grid Solvers

A dissertation submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy at George Mason University

By

Alejandro Agustin Figueroa
Bachelor of Science
Instituto Universitario Aeronautico, 2014

Director: Dr. Rainald Löhner, Distinguished Professor
Department of Physics and Astronomy

Fall Semester 2019
George Mason University
Fairfax, VA

Copyright © 2019 by Alejandro Agustin Figueroa
All Rights Reserved

Dedication

I dedicate this dissertation to my family who has encouraged me during these years and to Diane who has closely supported me during the last stages of my PhD.

Acknowledgments

I would like to thank the following people who made this dissertation possible. I am grateful to my advisor, Dr. Rainald Löhner, Director of the Center for CFD at George Mason University for his unconditional support, encouragement, advice and for openly sharing his experience throughout my time as a PhD student at George Mason University.

My sincere thanks also go to Dr. Fernando Camelli, Dr. Juan Cebral and Dr. Chi Yang, Professors at George Mason University, for serving as members of my PhD committee.

Last but not least, my friends here in the US that have made these years unforgettable.

Table of Contents

	Page
List of Tables	vii
List of Figures	viii
Abstract	xi
1 Introduction	1
1.1 Computational Fluid Dynamics in Industry	1
1.2 Flow Modeling	3
1.2.1 Navier-Stokes Equations	3
1.2.2 Model for Turbulence	4
1.2.3 Energy Cascade	5
1.3 Motivation: A Practical Example	6
1.4 Issues Adressed	7
2 Finite Difference Solver	10
2.0.1 Numerical Techniques	10
2.1 Discretization in Space	11
2.1.1 Central Difference Discretization and Artificial Dissipation	11
2.1.2 Roe Type Solver for Low Mach Applications	14
2.2 Temporal Discretization	19
2.2.1 Traditional and Low-Storage Runge-Kutta Methods	19
2.2.2 Multivalued Multistage Methods	22
2.2.3 Lax-Wendroff Schemes	28
3 Interface Treatment for Nested Cartesian Grids	33
3.1 Introduction	33
3.2 Multiblock Option	33
3.3 $h/2h$ Interpolation Between Cartesian Grids	34
3.3.1 Post Processing Interpolation	34
3.4 Interpolation Limiters	39
4 Examples and Performance	41
4.1 Convergence Study	41

4.1.1	2D Lamb Vortex	41
4.1.2	2D Convergence Study with Stationary Lamb Vortex	46
4.1.3	3D Lamb Vortex	50
4.1.4	2D Lamb Vortex in 3D	53
4.1.5	Cylinder	56
4.1.6	Flow Around a Sphere	59
5	Energy Cascade	67
5.1	Introduction	67
5.2	Problem Set-up	67
5.3	Results	68
5.3.1	Energy Cascade	76
6	The Ahmed Body	80
6.1	Introduction	80
6.1.1	Geometry and Test Set-up	84
6.2	Results	85
6.2.1	Flow Structures	85
6.2.2	Drag Coefficient and Run Time	95
7	Conclusion	99
7.1	Conclusions	99
A	Roe Solver Matrices	101
A.1	Matrix set up for B term	101
A.2	Matrix set up for C term	103
B	Interpolation Weights	106
	Bibliography	107

List of Tables

Table	Page
4.1 L_2 and L_∞ norms for the 3D Lamb Vortex.	51
4.2 L_2 and L_∞ norms for the 2D Lamb Vortex in 3D.	53
4.3 Results for the Sphere Case at $Re = 3 \cdot 10^2$	61
4.4 Required Time for Solving a Sphere at $Re = 1 \cdot 10^4$ for 10secs.	66
5.1 Runtimes for the Taylor Green Vortex.	77
5.2 Kolmogorov Length Scales, at $t = 9secs$	78
6.1 Vortex Position Behind the Ahmed Body, Fine Mesh, Position in mm	88
6.2 Drag Coefficients of the Ahmed Body. ‘4 th Order + Interpolation’ Corresponds to a Discretization of Fourth Order With High Order Interpolation Between Grids Activated.	95
6.3 Run Times for the Ahmed Body Simulations.	97
6.4 Run Times for the Ahmed Body. FV2 = second order finite volume method, PS = pseudo spectral method, BDF2/3 = second/third order backward Euler, CN = implicit Crank-Nicolson, RK2/3 = second/fourth order Runge Kutta.	98

List of Figures

Figure	Page
1.1 Energy Spectrum. Figure from [1].	5
2.1 Discretization Errors for a Linear Problem.	27
2.2 Discretization Errors for a Non Linear Problem.	27
2.3 Density for several LW schemes at $t = 0.2$	30
2.4 Pressure for several LW schemes at $t = 0.2$	31
2.5 Velocity for several LW schemes at $t = 0.2$	31
3.1 Multiblock Option.	33
3.2 Interpolation Between FD Grids (1D).	34
3.3 Interpolation Between FD Grids (2D).	35
3.4 Interpolation Factors for Edges of Type E_1	37
3.5 Interpolation Factors for Edges of Type E_2	37
3.6 Interpolation Factors for Faces.	38
3.7 E_2 Interpolation in 3D.	38
4.1 Lamb Vortex: Initial Conditions, Mesh $h2h$	42
4.2 Lamb Vortex: Comparison of Pressures	43
4.3 Lamb Vortex: Comparison of Velocities	44
4.4 Lamb Vortex: Zoom Around the Vortex for Mesh $2h$ and $h2h$	45
4.5 Stationary Lamb Vortex: Typical Initial Configuration.	46
4.6 Stationary Lamb Vortex: L_2 Convergence for 6th Order.	47
4.7 Stationary Lamb Vortex: L_∞ Convergence for 6th Order.	47
4.8 Stationary Lamb Vortex: L_2 Convergence for 8th Order.	48
4.9 Stationary Lamb Vortex: L_∞ Convergence for 8th Order.	48
4.10 Stationary Lamb Vortex: Comparison of the L_2 Norm for 6th and 8th Order.	49
4.11 3D Lamb Vortex: Initial Conditions, $h2h$ Mesh.	50
4.12 3D Lamb Vortex: Comparison of Pressure Magnitudes at $T = 120$	51
4.13 3D Lamb Vortex: Comparison of Velocities Magnitudes at $T = 120$	52
4.14 Lamb Vortex: Comparison of Pressures.	54

4.15	Lamb Vortex: Comparison of Pressures.	55
4.16	Cylinder: Grid System Used.	57
4.17	Cylinder: Results at Time $T = 100$	57
4.18	Cylinder: Station Time History for Station 5: $\mathbf{x} = (1, 0, 0)$	57
4.19	Cylinder: Station Time History for Station 7: $\mathbf{x} = (10, 0, 0)$	58
4.20	System of Nested Cartesian Grids Used for the Simulation of Flow Around a Sphere.	60
4.21	Coarse Mesh and Velocity Initial Condition.	60
4.22	Velocity Field and Pressure Contour at $t = 70secs$, Medium Mesh.	62
4.23	Velocity Field and Pressure Contour at $t = 70secs$, Coarse Mesh.	63
4.24	Instantaneous Pressure Coefficient at $t = 70secs$, Medium Mesh.	63
4.25	Averaged Drag Coefficient, C_d . Experimental Data from [2] is Compared With Lattice-Boltzman Method LB [3] on a Fine, Medium and Coarse Mesh, and Several Finite Difference Discretization Orders FD.	64
4.26	Velocity Field and Pressure Contour at $t = 30secs$, Fine Mesh.	65
5.1	Taylor Green Vortex: Initial Conditions.	68
5.2	Taylor Green Vortex: Pressure (Left) and Velocity (Right) at $t = 8secs$, Coarse Mesh. Top: 2^{nd} Order, Middle 4^{th} Order, Bottom: 8^{th} Order FD Approximation.	69
5.3	Taylor Green Vortex: Pressure (Left) and Velocity (Right) at $t = 16secs$, Coarse Mesh. Top: 2^{nd} Order, Middle: 4^{th} Order, Bottom: 8^{th} Order FD Approximation.	70
5.4	Taylor Green Vortex: Pressure (Left) and Velocity (Right) at $t = 8secs$, Fine Mesh. Top: 2^{nd} Order, Middle: 4^{th} Order, Bottom: 8^{th} Order FD Approximation.	71
5.5	Taylor Green Vortex: Pressure (Left) and Velocity (Right) at $t = 16secs$, Fine Mesh. Top: 2^{nd} Order, Middle: 4^{th} Order, Bottom: 8^{th} Order FD Approximation.	72
5.6	Taylor Green Vortex: Kinetic Energy vs Time on a Coarse (100^3 Nodes), Medium (200^3 Nodes) and Fine (400^3 Nodes) Mesh.	74
5.7	Taylor Green Vortex: Kinetic energy (tke), vorticity (vor) and numerical (num. diss) dissipation rate for different approximation orders and mesh sizes. ‘FR-SD-63X4’ is a p_3 solution using the FR-SD method on a 64^3 mesh.	75

5.8	Taylor Green Vortex: Kinetic energy (tke), vorticity (vor) and numerical (num. diss) dissipation rate vs time. Left: Comparison between a 2^{nd} order FD approximation on a 200^3 mesh and a 8^{th} order FD approximation on a 100^3 mesh. Right: Comparison between a 2^{nd} order FD approximation on a 400^3 mesh and a 8^{th} order FD approximation on a 200^3 mesh.	76
5.9	Taylor Green Vortex: Energy cascade for different approximation orders and mesh sizes at $t = 9secs$. ‘OFR’ refers to an optimized FR scheme on a coarse (16^3) and medium (32^3) mesh with a p_3 and p_5 solution. Spectral refers to the reference solution obtained in [4] with a Spectral method. The last value corresponds to the slope value predicted by the Kolmogorov hypothesis. . .	79
6.1	Ahmed Body geometry, dimensions are in mm . Figure from [5].	81
6.2	C_d curve for several φ angles. Here, c_W is the drag coefficient, c_R is the friction drag coefficient, c_B is the vertical base pressure drag coefficient, c_K is the forebody pressure drag coefficient and c_S is the slant surface pressure drag coefficient. Figure from [6].	82
6.3	Flow macro structure at the rear slant of the Ahmed Body. Figure from [6].	83
6.4	System of Cartesian Grids for the Ahmed Body.	84
6.5	Coarse Mesh, Ahmed Body.	85
6.6	Experimental Streamlines Behind the Ahmed Body (Source [7]).	86
6.7	LIC Surfaces of the Flow Behind the Body	87
6.8	Normalized Mean Velocity at $x/h = -0.135$. Experimental Data at $x/h = -0.13$	89
6.9	Normalized Mean Velocity at $x/h = 0.278$. Experimental Data at $x/h = 0.27$.	90
6.10	Normalized Mean Velocity at $x/h = 1.74$. Experimental Data at $x/h = 1.74$.	91
6.11	Normalized Mean Velocity at $x/h = 3.08$	92
6.12	Q-Criterion at $t = 0.4$	93
6.13	Instantaneous Velocity Field at $t = 0.4$	94
6.14	Drag Coefficient.	96

Abstract

IMPROVEMENT OF NESTED CARTESIAN FINITE DIFFERENCE GRID SOLVERS

Alejandro Agustin Figueroa, PhD

George Mason University, 2019

Dissertation Director: Dr. Rainald Löhner

Computational Fluid Dynamics (CFD) has been a successful tool for industry applications during the last decades. However, still accurate solutions involving vortex propagation, separated and turbulent flows are associated with high computing costs. LES simulations of complex geometries, such as an automobile, at high Reynolds number require several days to obtain a solution with information statistically relevant on an arbitrary high number of cores.

FDFLO is a High Order Cartesian Finite Difference (FD) code, developed for industrial applications at the CFD Center at George Mason University. FDFLO is able to perform LES simulation with the aim of overnight turnarounds. A study of the capabilities of an FD solver to obtain accurate turbulent results in complex geometries and the main implementations and improvements introduced in the code are the subject of this thesis.

The two central issues addressed in this thesis are: a) Proper schemes for the transition between grids of different spatial resolution, and b) Development of high order methods for the time discretization with minimal stage count. The first issue was resolved using high order interpolation post-processing interpolation schemes. The second issue was addressed by employing multivalued multistage Runge-Kutta (MMRK) schemes.

The methods developed were tested on canonical analytical test cases such as the Lamb

vortex and the Taylor-Green vortex, as well as experimental results for the Ahmed Body. The results obtained showed the expected spatial and temporal convergence characteristics of the methods, as well as very good agreement (overall drag, location of flow structures, etc.) for the Ahmed body.

The gains in computational performance for the MMRK methods used as compared to classical Runge-Kutta or Low-Storage RK schemes reached up to 25% with no loss in accuracy.

An interesting empirical result observed for the Taylor-Green vortex was that for fully turbulent flows the accuracy of 8th order schemes on grids of size $2h$ was equivalent to 2nd order schemes on grids of size h . The resulting overall speedup of the 8th order scheme over the 2nd order scheme was more than a factor of 5:1.

Chapter 1: Introduction

1.1 Computational Fluid Dynamics in Industry

Computational Fluid Dynamics (CFD) has been applied extensively in several fields that include academic research and industry applications. First implementations arose in the the first half of the twenty-first century in the areas of weather prediction, and to model flow around an airfoil. Since then, the complexity of the numerical methods used to solve the fundamental system of equations in almost all CFD problems, the Navier Stokes (NS) equations, has increased. Initially, the problems to be solved were restricted to the potential equations, later generalized to inviscid Euler equations, and finally stretching to the general NS equation in the 1950s. Furthermore, a large number of discretization methods of these equations have been developed, including:

- The Finite Difference Method (FD),
- The Finite Volume Method (FV),
- The Finite Element Method (FE), and
- The Spectral Element Method.

All of these methods have been extended to second or higher order schemes, in order to solve problems with high resolution. These extensions are usually required for accurately capturing discontinuities, sharp changes or shocks, and more recently to solve challenging problems such as separation and turbulence. However, legacy codes and worldwide commercial codes are generally limited to second order accuracy. Based on the conclusions of the NASA CFD Vision 2030 Study [8] from 2014 these codes are not therefore adequate for vortex dominated and transitional flows.

In recent years, Witherden and Jameson [9], Dawes [10] and Löhner [11] addressed the high cost of running simulations that involve separation, turbulence and vortex dominated flows which are commonly required by industry. A paradigm shift has been affecting the product development cycles, for example in the automotive industry [12, 13], leading to:

- improvement in performance and quality of the product,
- lower development costs,
- reduced time between product conception and release to market.

In the early 90s the total vehicle development process used to take approximately 60 months [13]. By the middle 2000s the process had shrunk to 18 months or less. The development was fueled in part by the introduction of computational methods during design. In the 2016 report ‘Global CFD Market in Automotive Industry 2017-2021’ [14], it is noted that CFD contributed to the automotive market reducing product development cycle time and cost, and reducing the number of product recalls. Still, turnaround times of industrial simulations with an accurate flow description in the car industry are in order of weeks, a cost that companies are interested to reduce.

Although simulations can be accelerated using models for the turbulence added to the Navier Stokes equations, as in the Reynolds Averaged Navier Stokes (RANS) equations, the reproduction of turbulent or separated flows is not reliable. Nowadays is it still difficult to predict the maximum lift for airfoils at large angle of attack or the correct flow field around a car with RANS models [15, 16].

To address this problem several strategies have been developed in recent years. The introduction of hybrid RANS-LES (Large Eddy Simulation) or DES (Detached Eddy Simulation) methods are examples of new approaches used by the industry to reduce simulation times and resources. A second approach, the Lattice-Boltzmann method (LBMs) has been improved considerably over the last three decades. For low Mach-number flows, such as those found in external aerodynamics and aeroacoustics of cars [17–19] LBM has yielded accurate results at a cost comparable to or below that of traditional Navier-Stokes based

solvers (Finite Difference, Finite Volume or Finite Element). Due to their beneficial properties, LBM has become an industry standard. As reported by [20], with the use of GPUs and LBM, an accurate solution of the flow around a car can be obtained in the order of 2 ~ 3 days. The similarities of the Lattice-Boltzmann schemes with Finite Difference solvers of the Navier-Stokes equations have been described in several studies [21, 22]. These similarities explain why on grids with similar stencil size these methods produce results of similar accuracy [23, 24].

In conclusion, massively separated flows have been at the forefront of CFD research for decades. However, there are not practically utilized in industry due to the associated computing time and high costs of the simulations. Based on the urgency of the industry to reduce costs a Finite Difference Solver (FDFLO) has been developed at the Center of Computational Fluid Mechanics at George Mason University with the goal of fulfilling industrial requirements. A study of the capabilities of a FDFLO to obtain accurate turbulent results in complex geometries and the main implementations and improvements introduced in the code are the subject of this thesis.

1.2 Flow Modeling

1.2.1 Navier-Stokes Equations

The governing equations of the flow are the Navier-Stokes equations and are defined as a system of partial differential equations by,

$$\frac{1}{c^2}p_{,t} + \rho \nabla \cdot \mathbf{v} = 0, \quad (1.1)$$

$$\rho \mathbf{v}_{,t} + \rho \mathbf{v} \cdot \nabla \mathbf{v} + \nabla p = \nabla \mu \nabla \cdot \mathbf{v} + \rho \mathbf{g} + \beta \rho \mathbf{g} (T - T_0) + S_v, \quad (1.2)$$

$$\rho c_p T_{,t} + \rho c_p \mathbf{v} \nabla T = \nabla \lambda \nabla T + S_T, \quad (1.3)$$

where $\rho, \mathbf{v} = (u, v, w), p, T, c, \mu, c_p, \lambda, \beta, T_0$ denote, respectively, the density, velocity, pressure, temperature, (constant) speed of sound, viscosity, heat capacitance, conductivity, thermal expansion and reference temperature for the fluid, and \mathbf{g}, S_v, S_T the gravity vector and source terms for velocities and temperature. The temperature is included as an option, as well as the Boussinesq approximation for natural convection.

1.2.2 Model for Turbulence

Combustion, acoustics and aerodynamics design of complex geometries with large separation regions are examples of engineering application where turbulence is a dominant phenomena. The model used in this study is LES, first proposed by Joseph Smagorinsky in 1963 [25] in the weather prediction area. Navier-Stokes results require solving a wide range of length and times scales, but solving all the scales is computational expensive and comes with prohibitive costs. Therefore, in these situations LES is used as it ignores the smallest length scales, which are the most computationally costly, with a low-pass filtering. Finally, small scale effects on the flow field are modeled with a sub-grid scale model, the Smagorinsky-Lilly model [25].

To that aim LES resolves scales from domain size L to a cutoff filter size Δ and define the higher wave number flow fluctuation that can be obtained. The Smagorinsky-Lilly model represents the turbulent viscosity as,

$$\nu_t = (C_s \Delta_g)^2 \sqrt{2 \bar{S}_{ij} \bar{S}_{ij}} = (C_s \Delta_g)^2 |S|, \quad (1.4)$$

$$\bar{S}_{ij} = \frac{1}{2} \left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right), \quad (1.5)$$

where S is the rate of strain tensor, Δ_g is the grid size and C_s is a constant that depends on the flow configuration.

1.2.3 Energy Cascade

The energy cascade is an important assumption in the study of turbulence and describes the transfer of energy between different scales of motion. There are two types of energy cascade,

- the direct energy cascade: when the transfer of energy occurs from the large scales to the small scales,
- the indirect energy cascade: when the transfer of energy take place from the small scales to the large scales.

This cascade is usually represented with an energy spectrum as in 1.1, where $E(k)$ denotes the kinetic energy and k the wave number. Large eddies have low wave number and small eddies have larger wave numbers. LES resolves all scales in the energy spectrum up

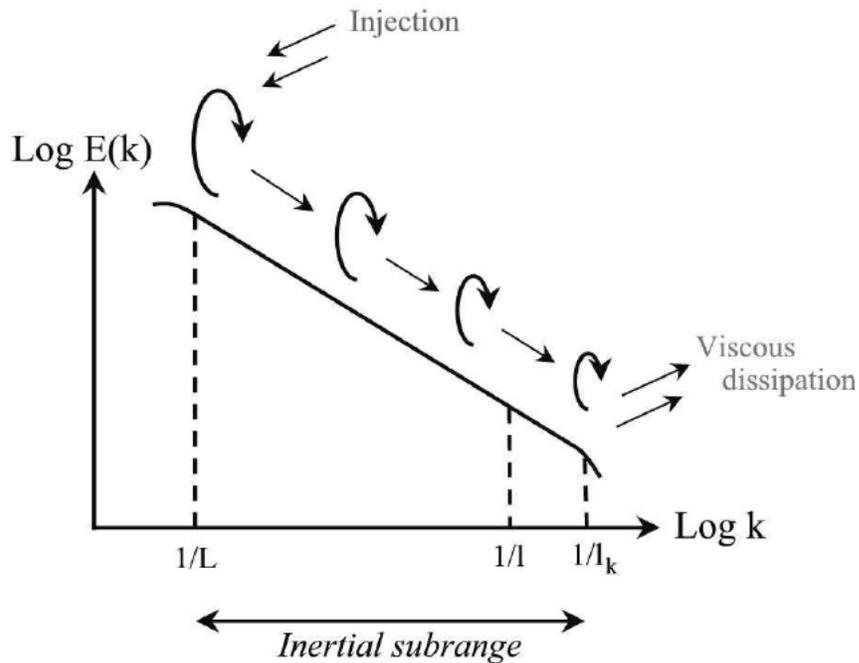


Figure 1.1: Energy Spectrum. Figure from [1].

to the cutoff wave number Δk . It is expected that LES simulations are able to resolve eddies

in the complete inertial subrange of the energy cascade. Large scale eddies produced by flow separation introduce the kinetic energy in the flow, whereas at high wave numbers energy is dissipated by viscous mechanisms into thermal energy. The behavior of the turbulence in the intermediate range is explained by the Kolmogorov hypothesis and is defined as the inertial subrange. In the inertial subrange of isotropic turbulence, Kolmogorov proposed that the rate at which kinetic energy is transferred into heat, depending on the scale of the eddy and the dissipation, follows

$$E(k) = C\varepsilon^{2/3}k^{-5/3}, \quad (1.6)$$

where ε is the dissipation rate and C is a constant that has to be determined experimentally. Kolmogorov introduced another important quantity to the turbulence theory, the Kolmogorov microscale. This is the scale where viscosity dominates the flow behavior and all the energy is converted into heat. It defines the lower limit of the inertial subrange, and is given by

$$\eta = \left(\frac{\nu^3}{\varepsilon}\right)^{1/4}, \quad (1.7)$$

where ν is the kinematic viscosity.

1.3 Motivation: A Practical Example

Based on this description of the current status of the CFD applied to the car industry and their goals for the next years, an analysis will be performed using the following description of the problem. The automotive industry set the objective of overnight LES simulations ($4 \cdot 10^4$ secs) of a car, with the subsequent requirements: a total surface of $O(20)m^2$ and minimum element size of $O(1)mm$, obtaining an estimated number of $2 \cdot 10^9$ points, and a fine mesh layer of $10cm$. With a Courant number of 0.1, approximately $O(3 \cdot 10^3)$ points along the longitude of the car, and 10 advective flow passages across the car, the total

number of time steps required to initialize the flow are $O(3 \cdot 10^5)$ to initialize the flow. However, for statistically relevance 100 advective passages are usually required, increasing the total number of time steps to $O(3 \cdot 10^6)$. In a more complex or detailed simulation the number of time steps can be of order $O(10^7)$ and with the industrial goal of overnight runs the required number of time steps per second are $250 \text{ timesteps/sec}$ or 4 msec/timestep .

A recent study confirmed that if only communication is taken into consideration, the aim of 4 msec/timestep is achievable [11]. Therefore, the focus at present is to minimize the CPU requirements per point per timestep. Solvers based on nested cartesian grid systems [26–30] are ideal candidates for the flow solvers that are sought, due to their scalability to easily increase the approximation order and their low floating point operation (FLOP) overhead.

1.4 Issues Adressed

High-order cartesian finite difference solvers have shown considerable advantages in speed and simplicity for the solution of partial differential equations (PDE). Block-wise adaptive mesh refinement combined with prismatic meshing near the body or usage of immersed or embedded body techniques appear to remove the main obstacles to attain geometrical flexibility and solution accuracy at the same time. Four issues are addressed in this thesis.

Issue 1

The transition between grids of different spatial resolutions is still a source of difficulty. Recent results show that for a high-order FD flow solver at the transition between grids, the numerical diffusion and dispersion based on simple interpolation schemes completely overwhelm the physics. This prompted a search for better interpolation techniques and interface treatments for such problems.

If one considers the interpolation problem, the immediate inclination is to use standard classic high-order interpolation techniques. High-order interpolations for locally refined

Cartesian grids have been reported repeatedly in the literature [31–33], where techniques such as Fourier basis, cubic least squares and limited monotone (WENO type) schemes were considered for designing the interpolations. In three dimensions, this often led to large stencil sizes (of the order of 20 to 30 coefficients) or iterative schemes that needed to be synthesized for each fine grid point. Moreover, interpolation schemes were fixed to maintain a particular order of accuracy. Extending them to even higher orders often required revisiting the derivations and re-implementation of new schemes.

A theme often discussed in the literature is the need of using conservative interpolation schemes. In a series of articles from 1994 (Chesshire and Henshaw [34], Brown [35], Pärt-Enander and Sjögreen [36]), the common conclusion is that for smooth solutions the standard interpolation procedures accurately capture the physical properties of the problem. As expected, conservative interpolations are only required near discontinuities. A typical situation are very slow shocks crossing a grid transition area. A frequent disadvantage of conservative interpolation is that it requires the use of a special viscosity or filter operator because the solutions are not strongly stable at the interface. Sebastian and Shu [37] compared a simple Lagrange interpolation procedure and a WENO interpolation scheme for strong shocks and complex geometry. The multi-dimensional tests performed show correct order of accuracy, robustness and non-oscillatory properties and obtained similar results between the Lagrange interpolation and the more expensive WENO interpolation.

Having a large stencil footprint or more complex schemes often defeats the purpose of using fast and cheap FD methods. This has led to the search for simpler interpolation techniques, utilizing immediately available data and constructing fast schemes that reuse results performed previously. The aim here is to assess whether improvements in accuracy with a modest increase in complexity are possible using such schemes.

Issue 2

Another topic discussed in this thesis is the viability of several high order methods for the time discretization. Typical explicit methods utilized in the literature are the Runge-Kutta

(RK) methods and low-storage versions thereof. However, the increase in the order of spatial discretization has to be accompanied by a rise in the order of the temporal discretizations. Runge-Kutta methods with order higher than five requires one extra stage (six stages) and there is no known eighth order method with less than eleven stages. Computationally, all these extra stages mean that more communication time between processor, application of boundary conditions, interpolation between meshes and storage of information are needed. In order to reduce total simulation time, high order RK methods with low stage order and Lax-Wendroff schemes are considered as substitutes of classical RK methods.

Issue 3

The claim in the literature is that in order to resolve the high wave numbers and to propagate the many vortices associated with turbulent flows high order methods are required. This claim is tested with the schemes implemented on the well-known Taylor-Green vortex by comparing the results of different grids and temporal and spatial discretization order with the results available in the bibliography.

Issue 4

Finally, a complex geometry, the Ahmed Body, is simulated in order to test the capabilities of the methods developed and compare them with experimental results. There are several wind tunnel publications on this geometry, which is considered a benchmark in the automotive industry to test the accuracy of CFD solvers. The flow around this geometry exhibits diverse complex patterns, mainly in the proximity of the rear slant and the side where separation and rotating vortices interact. The value of the drag coefficient and the flow behavior constitute the main characteristics, and are compared with experiments as well as numerical RANS and DES simulations.

Chapter 2: Finite Difference Solver

It is not clear why Finite Difference methods are not used as frequently as LBMs. Both methods use structured grids for the calculations but the number of FLOPS required per point per timestep is greater in LBMs. These schemes update 25 or 19 (with or without temperature) variables whereas for FDs the number of advected variables are 4 or 5 depending if the temperature is added or not. Furthermore, it is an easy and inexpensive matter to increase the order of approximation of Finite Difference methods for the Navier-Stokes equations, something that is hardly feasible for Lattice-Boltzmann methods.

FDFO is a FD code optimized for speed that solves the weakly compressible Navier-Stokes equations. This code was compared in [28, 29] to a LBM code for several test cases and similar results with either method were obtained.

2.0.1 Numerical Techniques

The numerical techniques implemented in FDFLO may be summarized as follows:

- Explicit timestepping via low-storage Runge-Kutta schemes;
- Conservative formulation for advection and divergence;
- Easy extensions to high-order stencils;
- Ordered access to memory;
- Minimum access to memory;
- Long 1D loops (for optimal vector, OMP and GPU performance);
- Use of halo points to impose boundary conditions and enable easy extension to massively parallel machines.

In the next sections, we describe in detail the spatial and temporal discretizations and boundary conditions employed.

2.1 Discretization in Space

Spatial discretizations are carried out using several techniques:

- Central Difference and Artificial Dissipation,
- Roe type solver,
- Lax-Wendroff method.

These approaches are implemented and tested in FDFLO in order to verify and evaluate their performance for low Mach applications. The goal of this section is to identify a robust high order fluid solver which is able to obtain a solution in a ‘realistic’ time frame.

2.1.1 Central Difference Discretization and Artificial Dissipation

The spatial discretization is carried out via Finite Differences on a cartesian grid with equal mesh size in all directions:

$$h_x = h_y = h_z = \Delta x. \quad (2.1)$$

All fluxes are written in conservative form as:

$$\mathbf{r}_i = \frac{1}{\Delta x} \left[\mathbf{f}_{i+1/2}^x - \mathbf{f}_{i-1/2}^x + \mathbf{f}_{i+1/2}^y - \mathbf{f}_{i-1/2}^y + \mathbf{f}_{i+1/2}^z - \mathbf{f}_{i-1/2}^z \right]. \quad (2.2)$$

Each flux is composed of the physical and the artificial dissipation flux, e.g.:

$$\mathbf{f}_{i+1/2} = \mathbf{f}_{i+1/2}^p - \mathbf{f}_{i+1/2}^d. \quad (2.3)$$

The advective physical fluxes are obtained from central difference operators of 2nd, 4th, 6th and 8th order:

$$\mathbf{f}_{i+1/2}^p \Big|^{II} = \frac{1}{2} (\mathbf{f}_{i+1}^p + \mathbf{f}_i^p), \quad (2.4)$$

$$\mathbf{f}_{i+1/2}^p \Big|^{IV} = \frac{7}{12} (\mathbf{f}_{i+1}^p + \mathbf{f}_i^p) - \frac{1}{12} (\mathbf{f}_{i+2}^p + \mathbf{f}_{i-1}^p), \quad (2.5)$$

$$\mathbf{f}_{i+1/2}^p \Big|^{VI} = \frac{37}{60} (\mathbf{f}_{i+1}^p + \mathbf{f}_i^p) - \frac{8}{60} (\mathbf{f}_{i+2}^p + \mathbf{f}_{i-1}^p) + \frac{1}{60} (\mathbf{f}_{i+3}^p + \mathbf{f}_{i-2}^p), \quad (2.6)$$

$$\mathbf{f}_{i+1/2}^p \Big|^{VIII} = \frac{533}{840} (\mathbf{f}_{i+1}^p + \mathbf{f}_i^p) - \frac{139}{840} (\mathbf{f}_{i+2}^p + \mathbf{f}_{i-1}^p) + \frac{29}{840} (\mathbf{f}_{i+3}^p + \mathbf{f}_{i-2}^p) - \frac{3}{840} (\mathbf{f}_{i+4}^p + \mathbf{f}_{i-3}^p). \quad (2.7)$$

For the 2nd order (Laplacian) operators the physical fluxes of 2nd, 4th, 6th and 8th order are:

$$\mathbf{f}_{i+1/2}^p \Big|^{II} = (\mathbf{f}_{i+1}^p - \mathbf{f}_i^p), \quad (2.8)$$

$$\mathbf{f}_{i+1/2}^p \Big|^{IV} = \frac{15}{12} (\mathbf{f}_{i+1}^p - \mathbf{f}_i^p) - \frac{1}{12} (\mathbf{f}_{i+2}^p - \mathbf{f}_{i-1}^p), \quad (2.9)$$

$$\mathbf{f}_{i+1/2}^p \Big|^{VI} = \frac{245}{180} (\mathbf{f}_{i+1}^p - \mathbf{f}_i^p) - \frac{25}{180} (\mathbf{f}_{i+2}^p - \mathbf{f}_{i-1}^p) + \frac{2}{180} (\mathbf{f}_{i+3}^p - \mathbf{f}_{i-2}^p), \quad (2.10)$$

$$\mathbf{f}_{i+1/2}^p \Big|^{VIII} = \frac{7175}{5040} (\mathbf{f}_{i+1}^p - \mathbf{f}_i^p) - \frac{889}{5040} (\mathbf{f}_{i+2}^p - \mathbf{f}_{i-1}^p) + \frac{119}{5040} (\mathbf{f}_{i+3}^p - \mathbf{f}_{i-2}^p) - \frac{9}{5040} (\mathbf{f}_{i+4}^p - \mathbf{f}_{i-3}^p). \quad (2.11)$$

These (unstable) approximations are stabilized by adding an appropriate artificial viscosity or damping [27, 38–40] of the form:

$$\mathbf{f}_{i+1/2}^d \Big|^{II} = c_d \lambda_{i+1/2} [(\mathbf{u}_{i+1} - \mathbf{u}_i)], \quad (2.12)$$

$$\mathbf{f}_{i+1/2}^p \Big|^{IV} = c_d \lambda_{i+1/2} [3(\mathbf{u}_{i+1} - \mathbf{u}_i) - (\mathbf{u}_{i+2} - \mathbf{u}_{i-1})], \quad (2.13)$$

$$\mathbf{f}_{i+1/2}^p \Big|^{VI} = c_d \lambda_{i+1/2} [10(\mathbf{u}_{i+1} - \mathbf{u}_i) - 5(\mathbf{u}_{i+2} - \mathbf{u}_{i-1}) + (\mathbf{u}_{i+3} - \mathbf{u}_{i-2})], \quad (2.14)$$

$$\mathbf{f}_{i+1/2}^p \Big|^{VIII} = c_d \lambda_{i+1/2} [35(\mathbf{u}_{i+1} - \mathbf{u}_i) - 21(\mathbf{u}_{i+2} - \mathbf{u}_{i-1}) + 7(\mathbf{u}_{i+3} - \mathbf{u}_{i-2}) - (\mathbf{u}_{i+4} - \mathbf{u}_{i-3})], \quad (2.15)$$

where λ is the maximum eigenvalue of the system

$$\lambda = |\mathbf{v}| + c, \quad (2.16)$$

and c_d the artificial viscosity or damping coefficient. Typical values are: $c_d^{II} = 0.2$, $c_d^{IV} = 0.10$, $c_d^{VI} = 0.02$, $c_d^{VIII} = 0.02$. For viscous cases, the artificial viscosity or damping coefficient of the momentum equations is reduced:

$$c_d^* = c_d \cdot r(u) \quad , \quad r(u) = \max(0, \min(1, Re_h - 1)) \quad , \quad Re_h = \frac{\rho u \Delta x}{\mu}. \quad (2.17)$$

Note that as the mesh is refined and the cell Reynolds number Re_h falls below $Re_h = 1$, the artificial viscosity vanishes. The same type of advection and artificial viscosity is also used for the temperature equation, but limited by the local Peclet number.

Finally, for the Boussinesq terms the used approximation is simply:

$$\mathbf{r}_i = \mathbf{g}\beta(T_B - T_i). \quad (2.18)$$

The same approximation is used for the source terms S_v, S_T .

In order to achieve long vector loops the formation of right-hand sides (RHSs) is carried out by forming a single array of point data. Given \mathbf{nx} , \mathbf{ny} , \mathbf{nz} , and defining $\mathbf{nxny} = \mathbf{nx} * \mathbf{ny}$, the points are traversed as $\mathbf{ip} = \mathbf{nxny}*(\mathbf{iz}-1) + \mathbf{ny}*(\mathbf{iy}-1) + \mathbf{ix}$. In order to minimize the use of registers, the RHSs are formed dimension by dimension.

2.1.2 Roe Type Solver for Low Mach Applications

A common approach to solve low Mach applications is to model incompressibility. The standard process starts adding an artificial compressibility parameter β to the continuum equation of the Navier Stokes equations. Here, a simpler set of formulas, the Euler equations, are used to obtain the Roe approximate Riemann solver. The Euler system is defined as,

$$\frac{1}{c^2\beta^2}p_t + \rho(u_x + v_y + w_z) = 0, \quad (2.19)$$

$$u_t + (u^2 + \frac{p}{\rho})_x + (uv)_y + (uw)_z = 0, \quad (2.20)$$

$$v_t + (uv)_x + (v^2 + \frac{p}{\rho})_y + (vw)_z = 0, \quad (2.21)$$

$$w_t + (uw)_x + (vw)_y + (w^2 + \frac{p}{\rho})_z = 0. \quad (2.22)$$

If $Q = (p, u, v, w)$, the previous continuity and momentum equations can be represented in vector form,

$$P^{-1}Q_t + A_0Q_x + B_0Q_y + C_0Q_z = 0, \quad (2.23)$$

where P^{-1} and P are,

$$P^{-1} = \begin{pmatrix} \frac{1}{\beta^2} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad P = \begin{pmatrix} \beta^2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad (2.24)$$

and A_0, B_0, C_0 are the following matrices,

$$A_0 = \begin{pmatrix} 0 & c^2\rho & 0 & 0 \\ \frac{1}{\rho} & 2u & 0 & 0 \\ 0 & v & u & 0 \\ 0 & w & 0 & u \end{pmatrix}, \quad B_0 = \begin{pmatrix} 0 & 0 & c^2\rho & 0 \\ 0 & v & u & 0 \\ \frac{1}{\rho} & 0 & 2v & 0 \\ 0 & 0 & w & v \end{pmatrix}, \quad C_0 = \begin{pmatrix} 0 & 0 & 0 & c^2\rho \\ 0 & w & 0 & u \\ 0 & 0 & w & v \\ \frac{1}{\rho} & 0 & 0 & 2w \end{pmatrix}. \quad (2.25)$$

Low Mach Preconditioner

If the Roe's flux difference splitting [41] is adopted, the flux H_n with the Low Mach Number Preconditioner defined in 1D is

$$H_n = \frac{1}{2}[H(Q_l) + H(Q_r)] - \frac{1}{2}P^{-1}|PA_i|(Q_r - Q_l), \quad (2.26)$$

where Q_r and Q_l are the right and left states with $H(Q_l)$, $H(Q_r)$ their respective fluxes, and the Jacobian matrices defined with $A_1=A=PA_0$, $A_2=B=PB_0$ and $A_3=C=PC_0$. The implemented preconditioner is a form of Turkel's preconditioning [42–45] and after expanding

Eq. (2.26) into its characteristic configuration,

$$H = \frac{1}{2}[H(Q_l) + H(Q_r)] - \frac{1}{2} \sum_{i=1}^4 |\lambda_i| \Delta w_i P^{-1} \vec{r}_i, \quad (2.27)$$

where Δw_i are the wave strengths $\Delta w_i = \vec{l}_i^T \Delta Q$ and λ_i , \vec{r}_i and \vec{l}_i are the eigenvalues, right eigenvectors and left eigenvectors of the matrix PA_i . In the next section it is shown how the eigenvalues, eigenvectors and wave strengths are obtained for matrix A. For matrices B and C the results can be found in the Appendix.

Matrix set-up for A term

The procedure used to obtain the wave strengths, eigenvalues and eigenvectors is similar to the approach presented in [46] and [47]. First, the matrix A is

$$A = PA_0 = \begin{pmatrix} 0 & \beta^2 c^2 \rho & 0 & 0 \\ \frac{1}{\rho} & 2u & 0 & 0 \\ 0 & v & u & 0 \\ 0 & w & 0 & u \end{pmatrix},$$

with eigenvalues,

$$\lambda_1 = u, \quad (2.28)$$

$$\lambda_2 = u, \quad (2.29)$$

$$\lambda_3 = u - \sqrt{\beta^2 c^2 + u^2} = u - l, \quad (2.30)$$

$$\lambda_4 = u + \sqrt{\beta^2 c^2 + u^2} = u + l, \quad (2.31)$$

with c the speed of sound. The preconditioning left and right eigenvectors are

$$R = \begin{pmatrix} 0 & 0 & \frac{\rho l(u+l)}{w} & \frac{\rho l(-u+l)}{w} \\ 0 & 0 & \frac{-l}{w} & \frac{l}{w} \\ 0 & 1 & \frac{v}{w} & \frac{v}{w} \\ 1 & 0 & 1 & 1 \end{pmatrix},$$

$$L = R^{-1} = \begin{pmatrix} \frac{-w}{l^2 \rho} & \frac{-uw}{l^2} & 0 & 1 \\ \frac{-v}{l^2 \rho} & \frac{-uv}{l^2} & 1 & 0 \\ \frac{w}{2l^2 \rho} & \frac{-w(l-u)}{2l^2} & 0 & 0 \\ \frac{w}{2l^2 \rho} & \frac{w(l+u)}{2l^2} & 0 & 0 \end{pmatrix},$$

Then, eigenvectors \vec{r}_i are replaced in Eq. (2.27),

$$\begin{aligned}
\sum_{i=1}^4 |\lambda_i| \Delta w_i P^{-1} \vec{r}_i = & \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \lambda_1 \Delta w_1 + \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \lambda_2 \Delta w_2 + \\
& + \begin{pmatrix} \frac{\rho l(l+u)}{\beta^2 w} \\ -\frac{l}{w} \\ \frac{v}{w} \\ 1 \end{pmatrix} \lambda_3 \Delta w_3 + \begin{pmatrix} \frac{\rho l(l-u)}{\beta^2 w} \\ \frac{l}{w} \\ \frac{v}{w} \\ 1 \end{pmatrix} \lambda_4 \Delta w_4.
\end{aligned} \tag{2.32}$$

Finally, the wave strengths are

$$\Delta w_1 = -\frac{w \Delta p}{l^2 \rho} - \frac{uw \Delta u}{l^2} + \Delta w, \tag{2.33}$$

$$\Delta w_2 = -\frac{v \Delta p}{l^2 \rho} - \frac{uv \Delta u}{l^2} + \Delta v, \tag{2.34}$$

$$\Delta w_3 = \frac{w \Delta p}{2l^2 \rho} - \frac{w(l-u) \Delta u}{2l^2}, \tag{2.35}$$

$$\Delta w_4 = \frac{w \Delta p}{2l^2 \rho} + \frac{w(l+u) \Delta u}{2l^2}. \tag{2.36}$$

Beta calculation

In 1987 [42], Turkel mentioned that β should be proportional to M^2 for a correct preconditioning and stiffness removal for low Mach numbers. Three β coefficients were implemented.

- Turkel, Vatsa, Radespiel β [48]:

$$\beta^2 = \min \left[\max \left(K_1(u^2 + v^2 + w^2) \left(1 + \frac{1 - M_0^2}{M_0^4} \right), K_2(u_\infty^2 + v_\infty^2 + w_\infty^2) \right), a^2 \right], \quad (2.37)$$

where M_0 is the cutoff value for the mach number and K_1 is between 1 and 1.1 and K_2 is between 0.4 and 1.

- Darmofal, Siu β [46]:

$$\beta^2 = \begin{cases} \frac{2M^2}{1-2M^2}, & \text{for } M < 0.5 \\ 1, & \text{for } M \geq 0.5, \end{cases} \quad (2.38)$$

where M is the Mach number.

- A third option implemented in FDFLO is:

$$\text{if } M \ll 1 \quad \Rightarrow \quad \beta = M_\infty. \quad (2.39)$$

2.2 Temporal Discretization

2.2.1 Traditional and Low-Storage Runge-Kutta Methods

Runge-Kutta (RK) type schemes are the most used explicit algorithms to perform time integration of the Navier Stokes equations. Since Jameson et al. [49] provided accurate solutions for steady transonic flow past a NACA 0012 airfoil using RK explicit schemes,

time stepping schemes have not been changed substantially in the last decades. In this subsection it is assumed the spatial discretization is correctly accomplished so that the system of equations may be expressed as a system of nonlinear ordinary differential equations (ODEs) of the form

$$U_t = r(t, U). \quad (2.40)$$

Typical RK methods are expressed as:

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t b_i \mathbf{r}^i, \quad (2.41)$$

$$\mathbf{r}^i = \mathbf{r}(t^n + c_i \Delta t, \mathbf{u}^n + \Delta t a_{ij} \mathbf{r}^j), \quad i = 1, s, \quad j = 1, s - 1. \quad (2.42)$$

Based on the last equations it is observed that RK methods require a certain number of stages s . For example, the classic fourth order RK scheme in Butcher tableau form is [50]:

$$\begin{array}{c|ccc} 0 & & & \\ \frac{1}{2} & \frac{1}{2} & & \\ \frac{1}{2} & 0 & \frac{1}{2} & \\ 1 & 0 & 0 & 1 \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array}$$

However, it is proven in [51] that for an explicit RK scheme of order p the number of stages s must satisfy $s \geq p$, and for $p \geq 5$ then $s \geq p + 1$. For instance, Luther [52] presented a sixth order RK algorithm with seven stages.

Traditional RK schemes require the storage of several copies of the unknowns and more important, demand multiple exchanges of unknowns between the computing nodes for large

simulations between stages. It is reported in [11] that pure communication between cores, on a machine with the latest communication network, can take on average $4msec$. As noted in the introduction in order to accomplish the industrial goal of overnight LES runs with $O(10^7)$ advective timesteps, one needs to run at $250 \text{ timesteps}/sec$, or $4 \text{ msec}/\text{timestep}$. Eventually, it is important to recognize that minimizing the number of stages reduces the overall runtime of CFD simulations.

Low-storage explicit Runge-Kutta (LSRK) methods are also implemented in FDFLO. This procedure is of the form,

$$\Delta \mathbf{u}^{n+i} = \alpha_i \Delta t \mathbf{r}(\mathbf{u}^n + \Delta \mathbf{u}^{n+i-1}) \quad , \quad i = 1, s \quad , \quad \Delta \mathbf{u}^0 = 0, \quad (2.43)$$

It is important to remark that for linear ODEs the choice

$$\alpha_i = \frac{1}{s+1-i} \quad , \quad i = 1, s, \quad (2.44)$$

in combination with Eqn. (2.43) leads to a scheme that is s -th order accurate in time. Like all explicit schemes, the allowable timestep is bounded by the condition:

$$\Delta t < CFL \cdot \min \left(\frac{h}{|\mathbf{v}| + c} \quad , \quad \frac{\rho h^2}{\mu} \quad , \quad \frac{\rho c_p h^2}{\lambda} \right). \quad (2.45)$$

where the allowable CFL factor is proportional and increasing with the number of stages s . One complete timestep is given by the following steps:

- Apply BC / Transfer Info from Domain to Halo Points
- Get Allowable Timestep
- Set Timestep $\Delta t = 0$ for Halo Points
- Loop Over the Stages:
 - Set $\mathbf{r} = 0$

- Compute \mathbf{r}
- Obtain $\Delta\mathbf{u} = \alpha_i \Delta t \mathbf{r}(\mathbf{u})$
- Apply Boundary Conditions
- Update \mathbf{u}

2.2.2 Multivalued Multistage Methods

One of the possible ways to achieve high order timestepping with a reduced number of stages is combining the concepts of RK methods and linear multistep methods. Using stage information from previous timesteps make it possible to obtain a better approximation of the current step. There are two types of generalizations of this procedure: Pseudo Runge-Kutta methods and ‘modified multistep methods’ [51]. Only pseudo RK schemes will be described in detail in the next sections.

Pseudo Runge-Kutta Methods

The first work in this area can be found in a 1966 paper by Byrne and Lambert [53]. In this method stage derivatives from previous and the current step are used together, to calculate the final result in the current step. Here, the notation differs in comparison to the last section to be consistent with the literature. The changes are: u is replaced with y , Δt with h and r with F . The general definition of these schemes is:

If F_j^{n-1} , $i = 1, 2, \dots, s$, are the stage derivatives from previous steps and F_j^n , $i = 1, 2, \dots, s$, the derivatives in the current step n , then

$$Y_i = y_{n-1} + h \sum_{j=1}^s a_{ij} F_j^n, \quad i = 1, 2, \dots, s, \quad (2.46)$$

$$F_i^n = f(x_{n-1} + hc_i, Y_i), \quad i = 1, 2, \dots, s, \quad (2.47)$$

$$y_n = y_{n-1} + h \left(\sum_{i=1}^s b_i F_i^n + \sum_{i=1}^s \bar{b}_i F_i^{n-1} \right). \quad (2.48)$$

An example of this method is the following fourth order $p = 4$ pseudo RK with three stages $s = 3$ with the coefficients:

c	A	
	\bar{b}^T	
		b^T

0			
$\frac{1}{2}$	$\frac{1}{2}$		
1	$-\frac{1}{3}$	$\frac{4}{3}$	
	$\frac{1}{12}$	$-\frac{1}{3}$	$-\frac{1}{4}$
	$\frac{11}{12}$	$\frac{1}{3}$	$\frac{1}{4}$

(2.49)

More details of the scheme implementation can be found in [51].

Two-Step Runge-Kutta Method

The second type of pseudo RK schemes is the Two-Step RK [54, 55]. This method introduces a dependency between Y_i and stage derivatives from previous steps in addition to dependencies on y_n . The Butcher tableau for this scheme is different compared with pseudo RK schemes and can be written as follows:

$$\begin{array}{c|c|c} c & \bar{A} & A \\ \hline & \bar{b}^T & b^T \end{array}$$

The Y_i , F_i and y_n for this method are

$$Y_i = y_{n-1} + h \sum_{j=1}^s \bar{a}_{ij} F_j^{n-1} + h \sum_{j=1}^s a_{ij} F_j^n, \quad i = 1, 2, \dots, s, \quad (2.50)$$

$$F_i^n = f(x_{n-1} + hc_i, Y_i), \quad i = 1, 2, \dots, s, \quad (2.51)$$

$$y_n = y_{n-1} + h \left(\sum_{i=1}^s \bar{b}_i F_i^{n-1} + \sum_{i=1}^s b_i F_i^n \right). \quad (2.52)$$

The coefficients for a fourth order ($p = 4$) Two-Step RK are [51]:

$$\begin{array}{c|ccc|ccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & -\frac{1}{4} & 0 & \frac{3}{4} & 0 & 0 \\ 1 & 0 & 1 & 0 & -2 & 2 & 0 \\ \hline & 0 & 0 & 0 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{array}, \quad (2.53)$$

To assess the order of convergence of these methods a unsteady heat equation is used. The study is presented in the next subsection.

Measuring ODE Discretization Error

The aim of this section is to measure the discretization error and the order of approximation of the methods presented before. The procedure used in this study is based on [56].

The unsteady heat equation is considered,

$$\rho c_p T_{,t} = \nabla k \nabla T + s, \quad (2.54)$$

where ρ is density, c_p is the pressure coefficient, T temperature, t time, k conductivity and s the source term. In 1D an exact unsteady solution only dependent on time (i.e. constant in space) would be

$$s = \cos(wt) \quad , \quad T = \frac{1}{\rho c_p w} \sin(wt). \quad (2.55)$$

The approximation error of this PDE is of the form

$$E(T^h - T) = \frac{c_t}{p!} \Delta t^p \frac{\partial^p T}{\partial t^p} + HOT, \quad (2.56)$$

where c_t is a constant that depends on the problem, HOT are the higher order terms and p is the discretization error. If the problem is initialized with the exact solution and integrated, after one timestep the solution will be given in Eq. (2.56).

The following procedure is implemented to measure the temporal discretization error:

- Pick an exact unsteady solution of the partial differential equation, which is constant in the spatial domain. In this case a source term that varies in time is chosen.
- Set the initial values of the unknowns equal to an exact solution of the source term.
- Advance the solution one timestep and obtain some norm of the difference between the initial values and the final values of the temperature.

Two different source terms will be used to measure the convergence order of different RK schemes; one is linear and the other nonlinear. The following parameters were chosen: $\rho = 1$, $c_p = 1$, $T_0 = 50$ and $k = 1$, $A = 5$, $\omega = 10$. The linear source term is given by

$$s_L = -T \quad , \quad T(t) = T_0 e^{-t} \quad , \quad T(0) = T_0. \quad (2.57)$$

The nonlinear source used as an example for this study is

$$s_{NL} = A\omega \cos(\omega t) \quad , \quad T(t) = A \sin(\omega t) \quad , \quad T(0) = 0. \quad (2.58)$$

For these cases the domain is given by $1 \leq x \leq 1$, and the stepsize $\Delta t = 0.1$. The schemes compared are the Pseudo RK (Pseudo RK), Two-Step RK (2Step RK), low storage RK of different orders (LSRK) and "traditional" RK schemes.

In Figures 2.1 and 2.2 the L^∞ norm is obtained for the linear and the nonlinear case presented before and used to measure the discretization error. One can see that, as expected, for the linear ($-T$) source the predicted convergence rates are obtained, while for the nonlinear ($A\omega \cos(\omega t)$) source the low-storage Runge-Kutta schemes do not go beyond 2nd order. From this simple study it is important to recognize the potential of Two-Step RK, since it can solve linear and nonlinear PDEs with the correct convergence rate, with fewer number of stages compared to LSRK and traditional RK schemes, and with the same or higher step size.

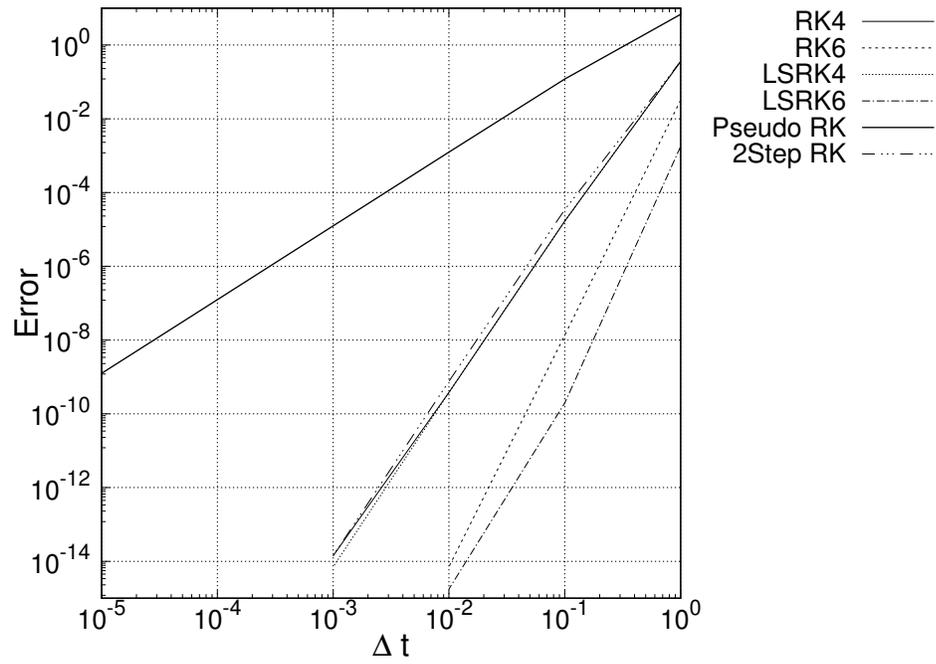


Figure 2.1: Discretization Errors for a Linear Problem.

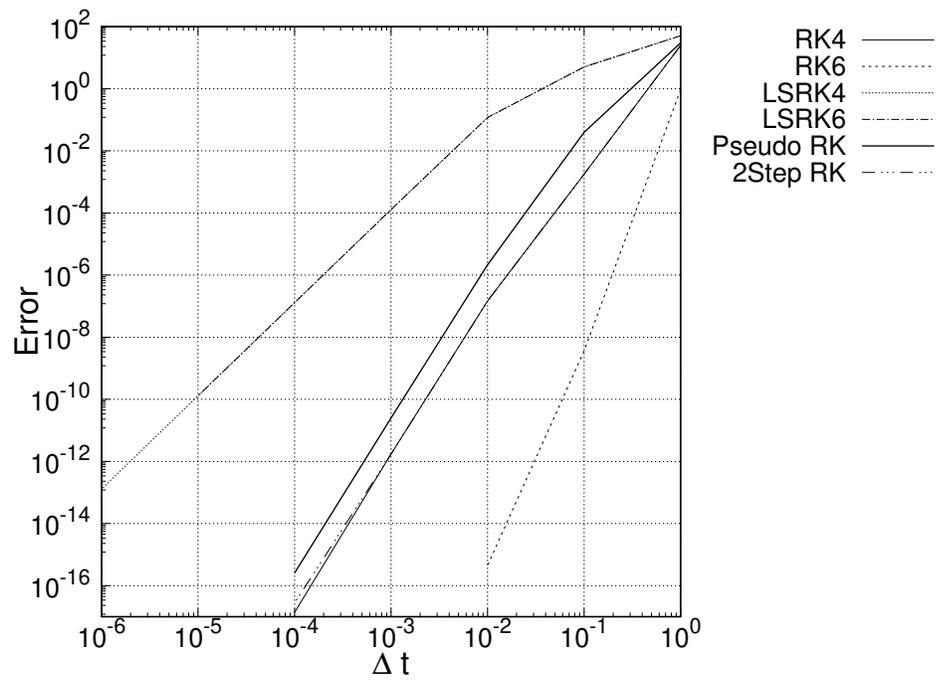


Figure 2.2: Discretization Errors for a Non Linear Problem.

2.2.3 Lax-Wendroff Schemes

Traditional Runge-Kutta algorithms separate the spatial and temporal discretizations which results in an improvement of the stability compared to other high order discretization methods such as Adams-Bashforth schemes. Unfortunately, RK methods require a minimum of n stages for a s th order numerical accuracy. For example, for a sixth order explicit Runge-Kutta method, 7 stages are required to solve the non-trivial system of equations [52].

Lax-Wendroff schemes are a alternative to RK that provide high order accuracy with lower number of stages. For instance, for a system of conservation laws in 1D of the form

$$U_t + F(U)_x = 0, \quad (2.59)$$

where $()_t$ and $()_x$ denote the partial differentiation with respect to time and space respectively, the finite difference approximation is written as

$$U_i^{n+1} = U_i^n - \frac{\lambda}{2}(F_{i+1}^n - F_{i-1}^n) + \frac{\lambda^2}{2}[A_{i+1/2}^n(F_{i+1}^n - F_i^n) - A_{i-1/2}^n(F_i^n - F_{i-1}^n)], \quad (2.60)$$

$$A(U) = \frac{\partial F}{\partial U}. \quad (2.61)$$

Note that A is the Jacobian matrix of the system and λ is the maximum eigenvalue of A . This is a 2nd order accuracy method in time and space and is based on a Taylor expansion in time of the nonlinear hyperbolic system (2.59). A key property used in the derivation of the method is the fact that multiple time derivatives of the conservatives variables can be written as spatial derivatives,

$$\frac{\partial U}{\partial t^n} = (-1)^n \frac{\partial^{n-1}}{\partial x^{n-1}}(A^{n-1}F_x). \quad (2.62)$$

This classic scheme can be extended to higher order approximations. For example, a fourth order finite difference scheme is represented in [57] as,

$$\begin{aligned}
U_i^{n+1} = U_i^n + \lambda & \left(\frac{1}{2}(F_{i+1}^n - F_{i-1}^n) - \frac{1}{12}(F_{i+2}^n - 2F_{i+1}^n + 2F_{i-1}^n - F_{i-2}^n) \right) \\
& + \frac{\lambda^2}{2} \left(A_{i+1/2}^n (F_{i+1}^n - F_i^n) - A_{i-1/2}^n (F_i^n - F_{i-1}^n) \right. \\
& - \frac{1}{12} \left\{ A_{i+3/2}^n (F_{i+2}^n - F_{i+1}^n) - 3A_{i+1/2}^n (F_{i+1}^n - F_i^n) \right. \\
& \left. \left. + 3A_{i-1/2}^n (F_i^n - F_{i-1}^n) - A_{i-3/2}^n (F_{i-1}^n - F_{i-2}^n) \right\} \right) \quad (2.63) \\
& + \frac{\lambda^3}{6} \left(\frac{(A_{i+1}^n)^2}{2} (F_{i+2}^n - F_i^n) - (A_i^n)^2 (F_{i+1}^n - F_{i-1}^n) + \frac{(A_{i-1}^n)^2}{2} (F_i^n - F_{i-2}^n) \right) \\
& + \frac{\lambda^4}{24} \left((A_{i+3/2}^n)^3 (F_{i+2}^n - F_{i+1}^n) - 3(A_{i+1/2}^n)^3 (F_{i+1}^n - F_i^n) \right. \\
& \left. + 3(A_{i-1/2}^n)^3 (F_i^n - F_{i-1}^n) - (A_{i-3/2}^n)^3 (F_{i-1}^n - F_{i-2}^n) \right),
\end{aligned}$$

with

$$A_{i\pm 1/2} = \frac{1}{2}(A_{i\pm 1} + A_i) - \frac{1}{16}(A_{i\pm 2} - A_{i\pm 1} - A_i + A_{i\mp 1}). \quad (2.64)$$

The main advantage of these type of schemes is that they require only one stage and are CPU intensive. A Sod's Shock Tube test is proposed in order to compare and test how well LW methods handle the simpler Euler equations in one dimension.

Sod's Shock Tube Test

A Sod Shock Tube test is used to compare the performance of LW schemes of different order of approximation with a Van Albada 2 flux limiter scheme. The initial conditions at the

left and right of the diaphragm are

$$\rho(x, 0) = \begin{cases} 1.0 & \text{for } x \leq \frac{1}{2} \\ 0.125 & \text{for } x > \frac{1}{2}, \end{cases} \quad (2.65)$$

$$p(x, 0) = \begin{cases} 1.0 & \text{for } x \leq \frac{1}{2} \\ 0.1 & \text{for } x > \frac{1}{2}, \end{cases} \quad (2.66)$$

$$u(x, 0) = \begin{cases} 0.75 & \text{for } x \leq \frac{1}{2} \\ 0.0 & \text{for } x > \frac{1}{2}, \end{cases} \quad (2.67)$$

The number of grid points used is 400 and for the fine grid simulation the number of intervals is 1000. The schemes compared are the first order Lax-Friedrichs, 2nd order Lax-Wendroff and 2-step Richtmyer and a 4th order LW. Results obtained are in Figures 2.3, 2.4 and 2.5. However despite obtaining good results in 1D series of factors render these LW

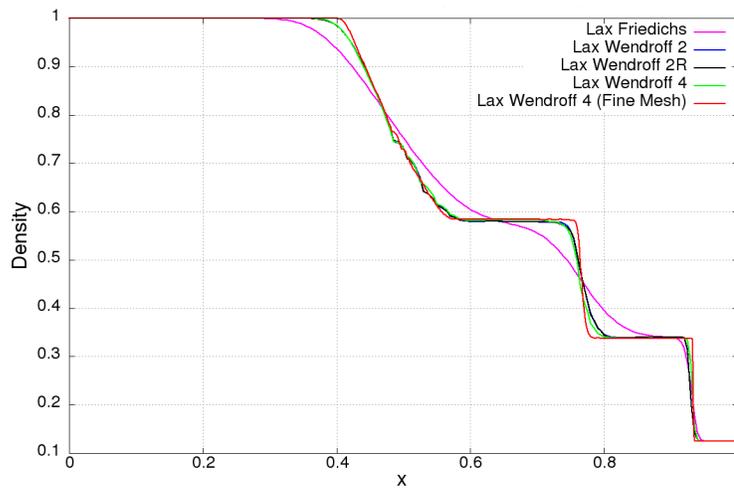


Figure 2.3: Density for several LW schemes at $t = 0.2$.

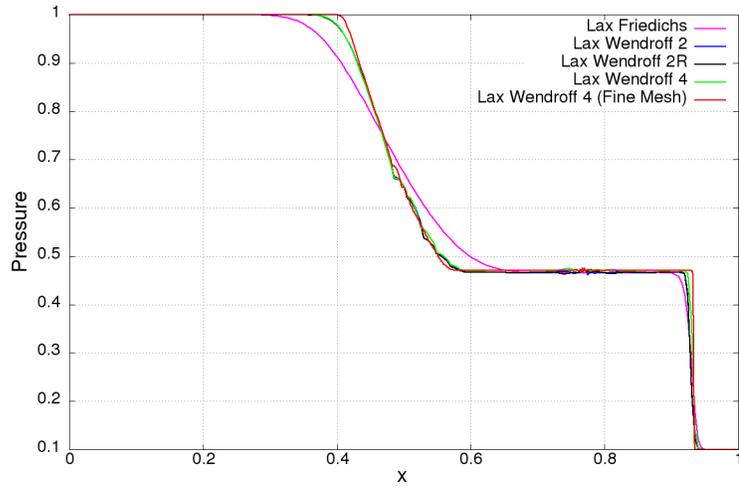


Figure 2.4: Pressure for several LW schemes at $t = 0.2$.

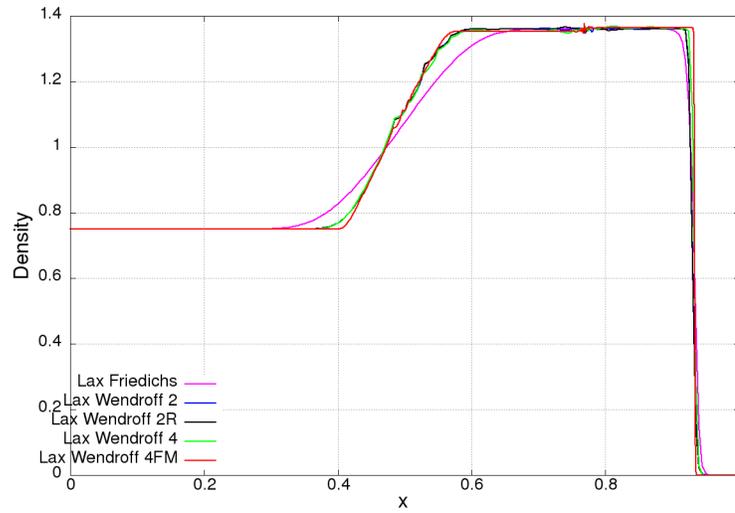


Figure 2.5: Velocity for several LW schemes at $t = 0.2$.

schemes ‘inadequate’ to the purpose of this research. If an extension to multiple dimensions is desired, the number of operations needed to obtain the stencils increase considerably. In a study by Lörcher and Munz [58] the number of operations per point and time step for the linearized Euler equations is calculated. For a fourth order scheme the number of operations are 400 for a 2D case and 3,125 for a 3D case. For an eighth order scheme these

numbers increase to 1,296 for a 2D problem and 18,225 in the 3D situation. The cost in 2D can be considered acceptable, but in 3D due to the large interpolation stencil the cost is prohibitive if the goal is to obtain a 'fast and cheap' solver.

Chapter 3: Interface Treatment for Nested Cartesian Grids

3.1 Introduction

3.2 Multiblock Option

A solver based on Cartesian Finite Differences may be very fast, and may be made applicable to complex geometries via immersed or embedded techniques. However, its use is still very limited when considering problems with varying spatial length scales. The best way of addressing this problem while keeping the speed advantages of the basic solver is via multiblocking [59,60]. The key idea is to consider each cartesian grid or block independently, and to combine these by interpolating the unknowns of the halo points from the adjacent blocks (Fig. 3.1).

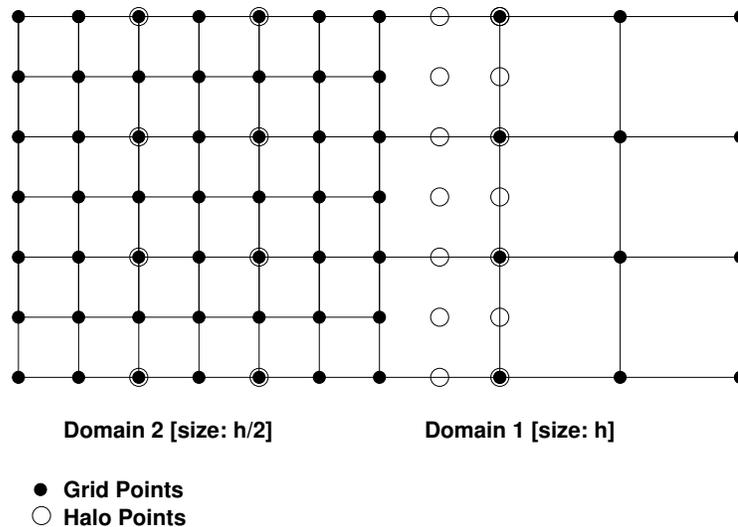


Figure 3.1: Multiblock Option.

3.3 $h/2h$ Interpolation Between Cartesian Grids

The situation commonly encountered is shown in Figures 3.2–3.3 for the 1D and 2D cases respectively. We denote the fine grids as grid h and coarse grid as grid $2h$. At the $h/2h$ boundary, the Cartesian grids need to exchange information. The objective here is that in order to maintain code modularity, halo points are used to transfer information between grids (and also for boundary conditions). This way, the ‘update’ and ‘boundary condition’ stages are separated in a clean, modular fashion. At the beginning of each timestep, iteration, or Runge-Kutta stage the information required for the halo points is obtained from the appropriate neighbouring grids. Furthermore, it is assumed that the information given at gridpoints is the most accurate and should therefore not be changed. This implies that for points that coincide (labeled D in Figure 3.3) a direct injection / transfer of information is desirable. On the other hand, for the points along edges or faces (labeled E, F in Figure 3.3), one is at liberty to apply interpolation schemes of different order. Note that this will only be required for grid h on a $h/2h$ boundary, i.e. only for the finer grid.

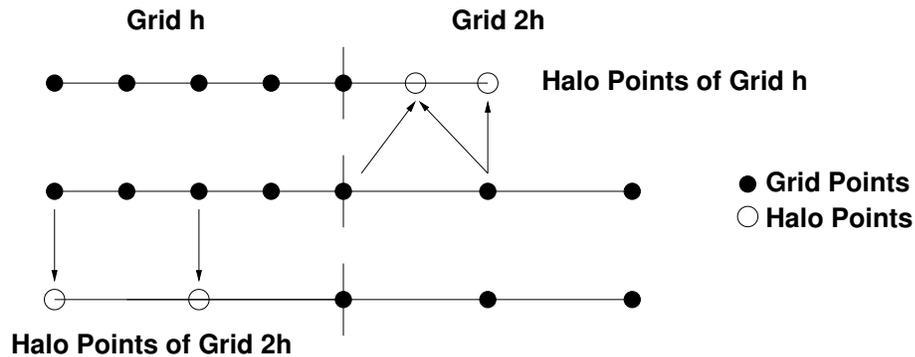


Figure 3.2: Interpolation Between FD Grids (1D).

3.3.1 Post Processing Interpolation

An interesting option when trying to maximize modularity is to transfer all direct injection points first from $2h$ to h , and then obtain the missing information by post processing this

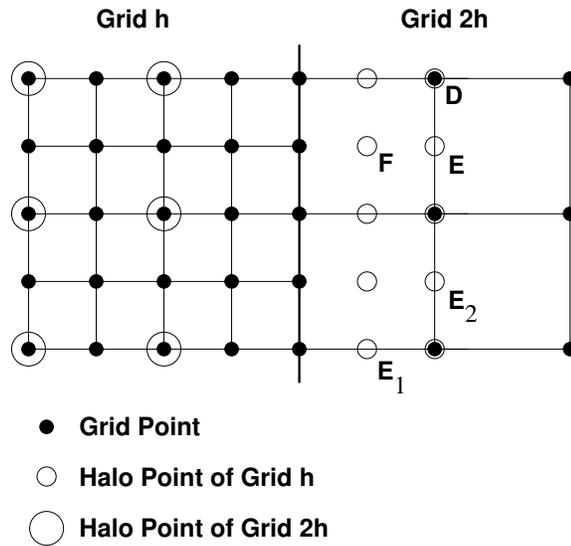


Figure 3.3: Interpolation Between FD Grids (2D).

data on grid h . In 2D, the cases that need to be considered are (see Figure 3.3):

- E_1 : Edge-points aligned with grid-lines from grid h .
- E_2 : Edge-points not aligned with grid-lines from grid h
- F : Face-points.

The distinguishing factor for points of type E_1 is that information from the interior of grid h is readily available and can be used to improve the interpolation order. Figure 3.4 shows some of the possibilities, together with the interpolation weights. For the points of type E_2 regular high order Lagrangian interpolation schemes are employed. Figure 3.5 shows some of the possibilities, together with the interpolation weights. Points of type F may be interpolated either via a weighted average of the surrounding edge points, or by treating them as points of type E_1 with the additional required information obtained previously for the points of type E_2 (Figure 3.6).

For a 2D cell information is ‘known’ at 4 points, while 5 points need to be interpolated. However, in 3D information is ‘known’ at 8 points, while 19 points require interpolation. Based on this, the options for 3D cases are to use a simple bi/trilinear interpolation or to

develop different techniques. For example, for a E_2 point (the blue point in Figure 3.7), the information needed to obtain the required value is obtained as a combination of linear (red points) and high order (blue edges) interpolation.

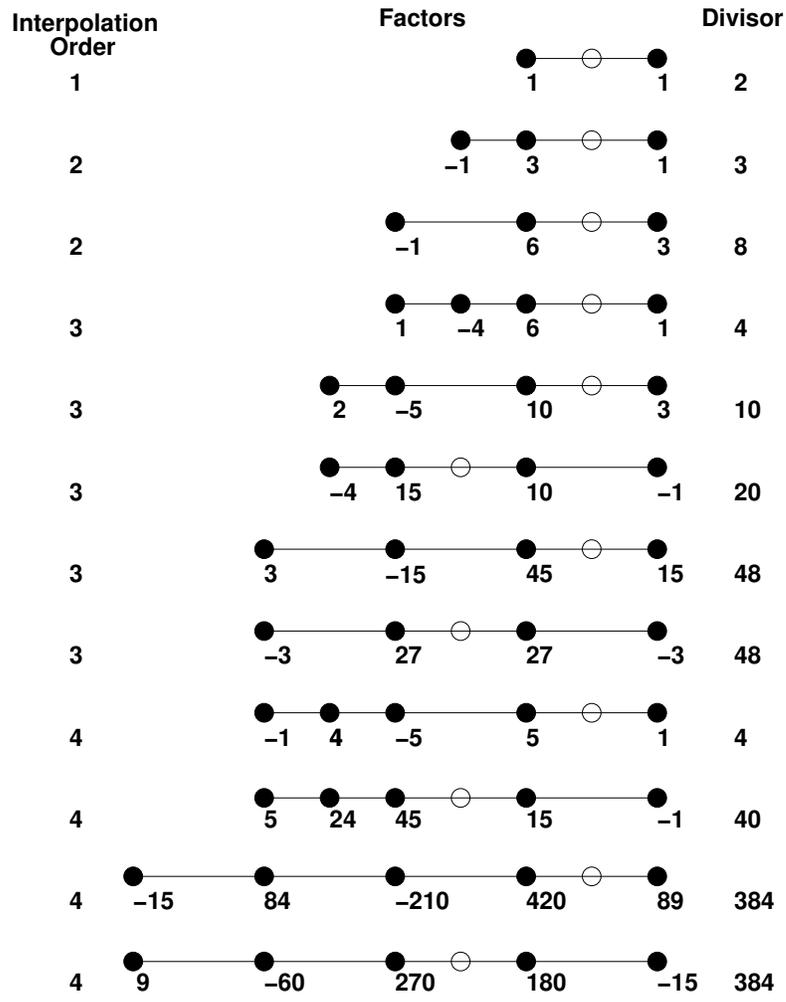


Figure 3.4: Interpolation Factors for Edges of Type E_1 .

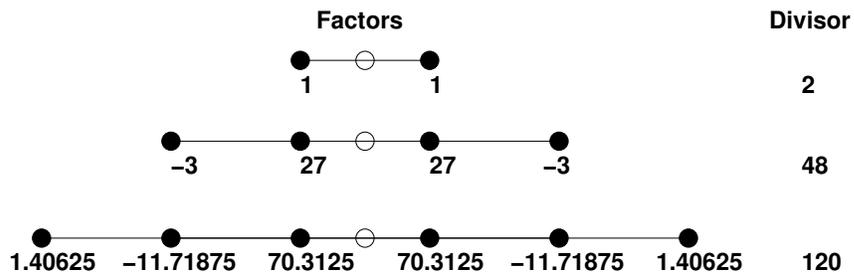


Figure 3.5: Interpolation Factors for Edges of Type E_2 .

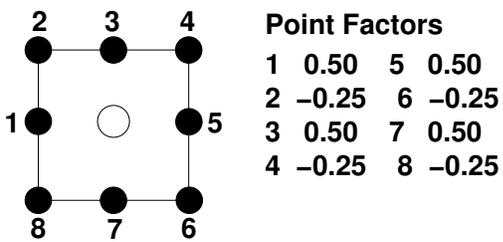
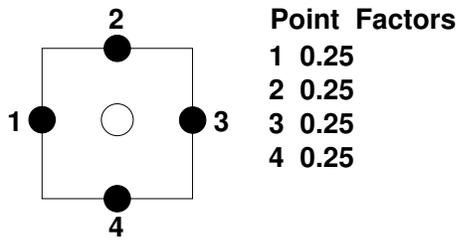


Figure 3.6: Interpolation Factors for Faces.

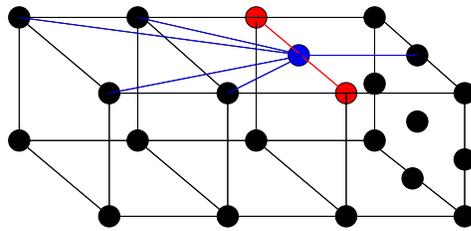


Figure 3.7: E_2 Interpolation in 3D.

3.4 Interpolation Limiters

In order to avoid spurious oscillations or sharp changes in the high order interpolation due to Runge's phenomenon, several limiters were implemented and tried.

- Barth-Jespersen limiter: This is a simple limiter where the maximum and minimum (U_{max}, U_{min}) values from the closest points around the interpolated point are selected and then used to limit the high order solution, U_h ,

$$U_i = \max(U_{min}, \min(U_{max}, U_h)). \quad (3.1)$$

- Radius of curvature limiter: Introduced in [61], this method constrains the high order solution with the argument that the interpolated value is accurate as long as the radius of curvature of an interface region is higher than three grid cells. It requires the calculation of a linear or low order interpolation, U_l ,

$$U_i = \begin{cases} U_h & \text{if } |U_h - U_l| < \beta \Delta x, \\ U_l & \text{otherwise,} \end{cases} \quad (3.2)$$

where β is heuristically chosen as $\beta = 1/20$.

- Modified radius of curvature limiter: Proposed in [62], this method attempts to overcome possible discontinuities that can appear with the previous technique. A modification is made to obtain a more regular interpolation

$$U_i = \begin{cases} U_l + \beta \Delta x & \text{if } |U_h - U_l| > \beta \Delta x, \\ U_h & \text{otherwise.} \end{cases} \quad (3.3)$$

- QMSL limiter: With the aim of using a simple and efficient limiter algorithm, and

based on the Flux Corrected Transport technique, Bermejo [63] proposed the following method. First define U_{max} and U_{min} and then set,

$$Q_+ = U_{max} - U_l \quad , \quad Q_- = U_{min} - U_l \quad , \quad P = U_h - U_l. \quad (3.4)$$

The value of the interpolated point is defined as

$$C = \begin{cases} \min(1, \frac{Q_+}{P}) & \text{if } P > 0, \\ \min(1, \frac{Q_-}{P}) & \text{if } P < 0, \\ 1 & \text{if } P = 0, \end{cases} \quad (3.5)$$

$$U_i = U_l + C(U_h - U_l). \quad (3.6)$$

Chapter 4: Examples and Performance

4.1 Convergence Study

The interpolation schemes developed were included and tested in FDFLO. In the following, we show the performance of the different options implemented.

4.1.1 2D Lamb Vortex

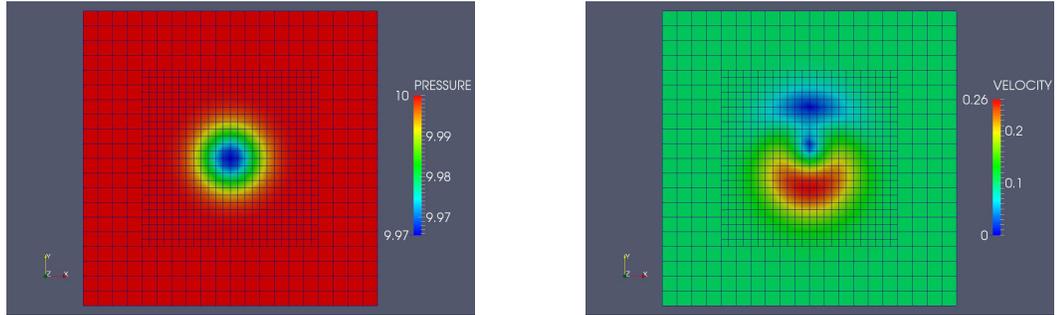
The so-called Lamb vortex, centered at $x, y = 0$, was chosen to assess the accuracy of the interpolation schemes developed. The unknowns parameters are given by:

$$u = u_0 - \frac{\alpha}{2\pi} y e^{\phi(1-x^2-y^2)} \quad , \quad v = \frac{\alpha}{2\pi} x e^{\phi(1-x^2-y^2)} \quad , \quad p = - \left(\frac{\alpha}{2\pi} \right)^2 \frac{1}{4\pi} e^{2\phi(1-x^2-y^2)} \quad (4.1)$$

and $\mu = 0$. For the particular case tested, the domain was given by $-5 \leq x, y \leq 5$, and $\alpha = 1$, $\phi = 0.5$, $c = 1$, $\rho = 1$, and the grids were of size $h = 0.125$ and $2h = 0.250$. The vortex was propagated for $T = 200$ time units. Given that the domain is doubly periodic, the vortex should reappear in the exact location as at time $T = 0$ after traversing the mesh twice.

Figure 4.1 shows the initial conditions for the mesh $h2h$, where the discretizations used are clearly visible. Figures 4.2–4.4 show the pressure and velocity obtained using an 8th order spatial discretization and a 5th order low-storage Runge-Kutta timestepping scheme. From left to right, the cases are: mesh $2h$, mesh $h2h$ (i.e. mesh h inside mesh $2h$) with usual bilinear interpolation, mesh $h2h$ with cubic interpolation, and mesh h . One can see that for the case with bilinear interpolation, **running the case on the mesh $2h$ yields better results than the mesh $h2h$, defeating the purpose of mesh refinement.** This should be a cause of alarm if one considers complex flow problems where vortices and

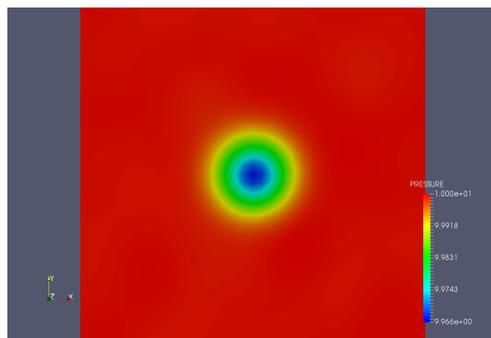
other flow structures will traverse grids with different mesh sizes. On the other hand, the cubic interpolation does yield results on the mesh $h2h$ that are demonstrably better than those on the mesh $2h$, indicating the potential of the procedures developed.



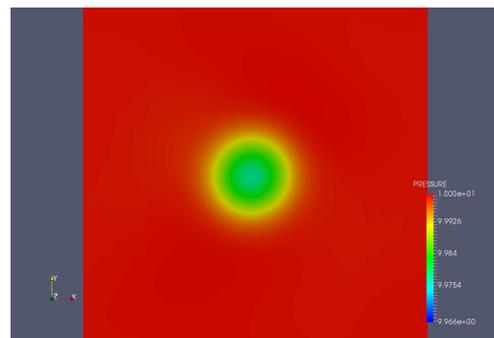
(a) Pressure

(b) Velocity

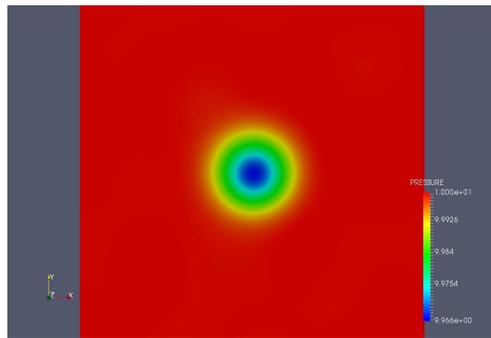
Figure 4.1: Lamb Vortex: Initial Conditions, Mesh $h2h$.



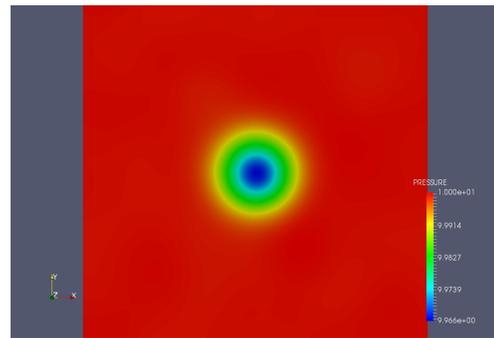
(a) Mesh $2h$



(b) Mesh $h2h$ *Bilinear*



(c) Mesh $h2h$ *Cubic*



(d) Mesh h

Figure 4.2: Lamb Vortex: Comparison of Pressures

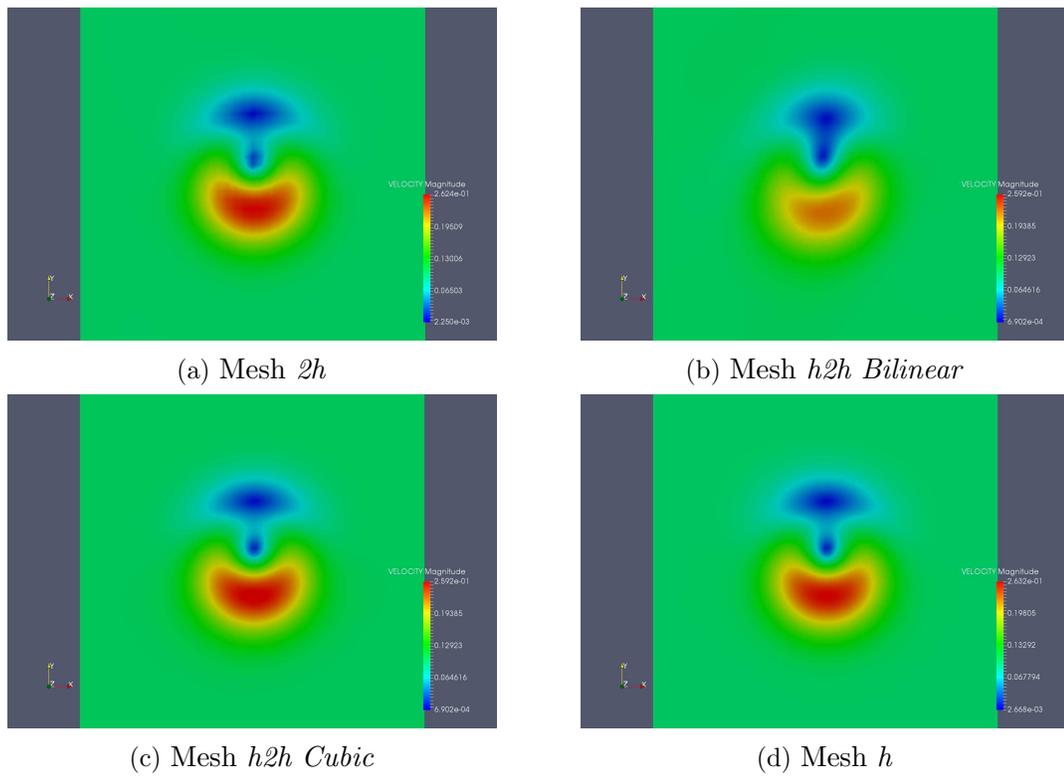
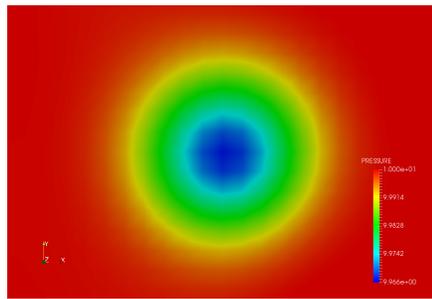
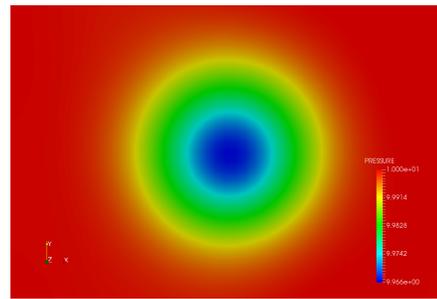


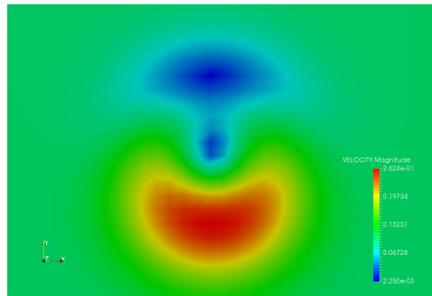
Figure 4.3: Lamb Vortex: Comparison of Velocities



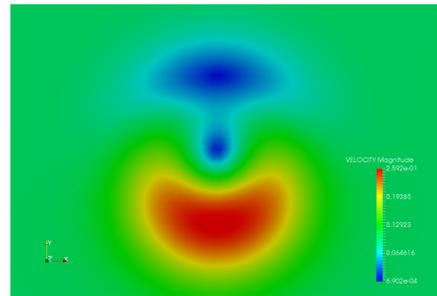
(a) Pressure: Mesh $2h$



(b) Pressure: Mesh $h2h$ Cubic



(c) Velocity: Mesh $2h$



(d) Velocity: Mesh $h2h$ Cubic

Figure 4.4: Lamb Vortex: Zoom Around the Vortex for Mesh $2h$ and $h2h$.

4.1.2 2D Convergence Study with Stationary Lamb Vortex

In order to quantify the relative merit of the different interpolation schemes, a series of convergence studies were carried out. The same domain as before was used, but the boundary conditions were changed from periodic to gliding wall. The right half of the domain was of size h , the left of size $2h$. At the same time, uniform grids of size h and $2h$ were run for comparison. A stationary Lamb Vortex (Eq. 4.1) was set as the initial condition. This is an exact steady solution, so the initial residual can be used to measure the convergence of the schemes. A typical configuration is shown in Figure 4.5.

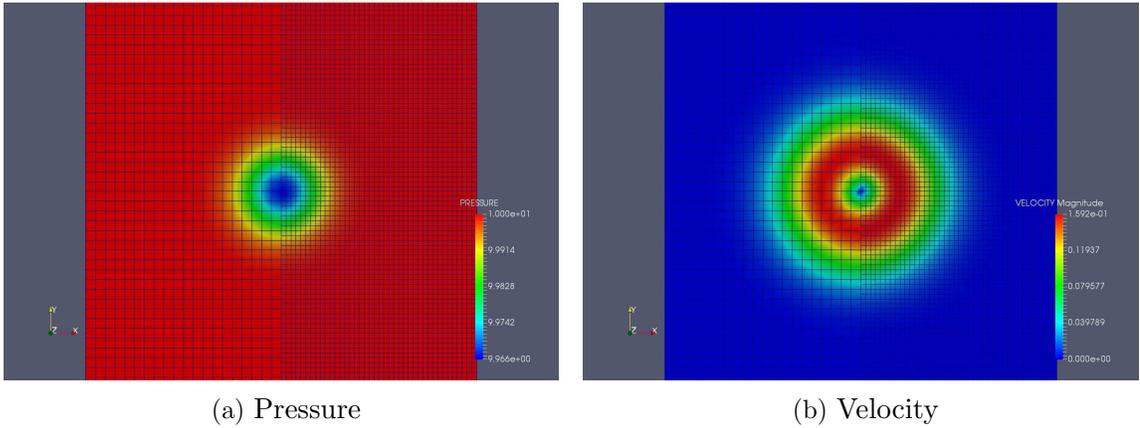
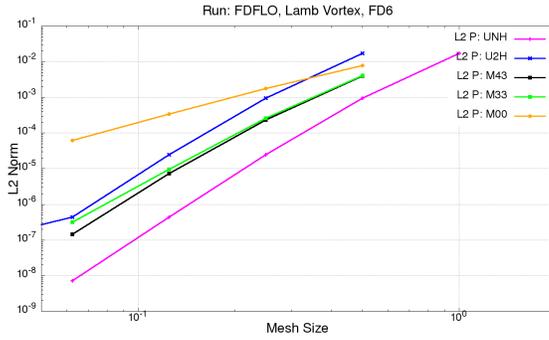


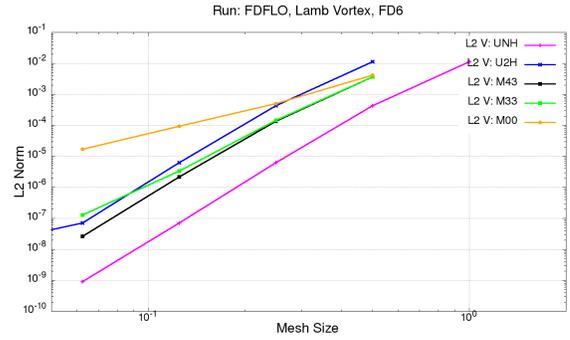
Figure 4.5: Stationary Lamb Vortex: Typical Initial Configuration.

Figures 4.6–4.9 show the convergence obtained for the 6th and 8th order solvers, together with different interpolation schemes. The notation is as follows: L2 denotes the L_2 norm, LI the L_∞ norm, P the pressure, V the velocity, UNH and U2H the convergence on uniform grids of size h and $2h$, 62 the 6th order discretization in time and space, 82 the 8th order discretization in time and space and M00, M33, M43 the cases of mixed $h, 2h$ grids with simple bi/trilinear interpolation, cubic interpolation and quartic interpolation. As expected, for the L_2 norm the errors of the high order interpolation schemes fall between the values for uniform grids of size h and $2h$. This is not always the case for the L_∞ norm. Furthermore, in Figure 4.10 one can see the serious negative effect on convergence of the bi/trilinear

interpolation. The results show that the aim of interpolation schemes that are balanced and appropriate to the spatial discretization while being local and fast has been achieved.

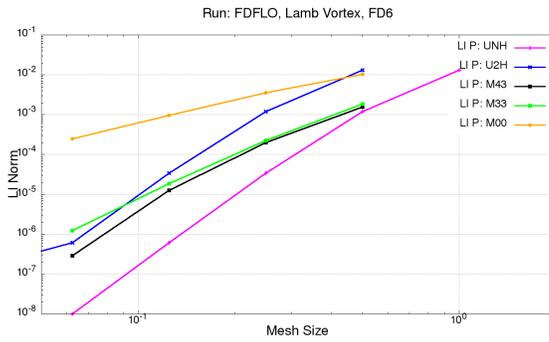


(a) Pressure

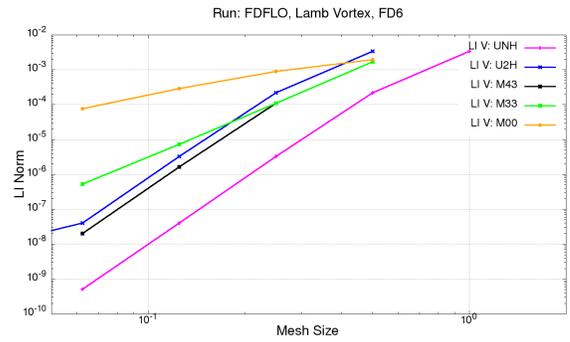


(b) Velocity

Figure 4.6: Stationary Lamb Vortex: L_2 Convergence for 6th Order.

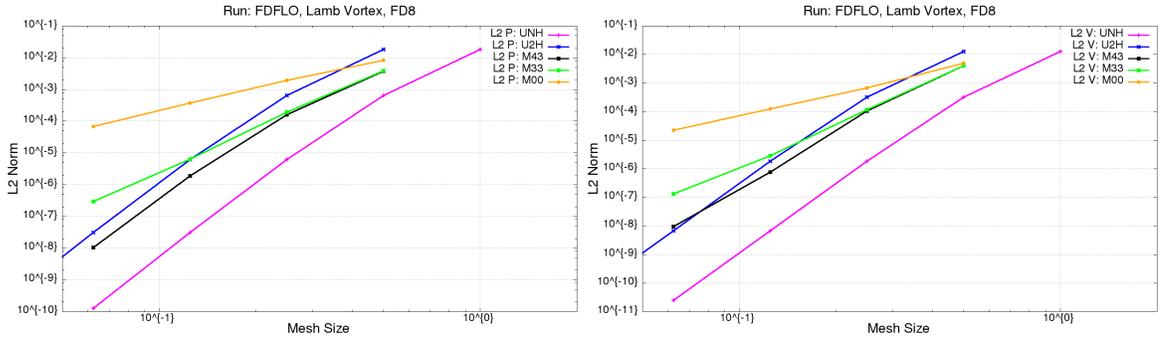


(a) Pressure



(b) Velocity

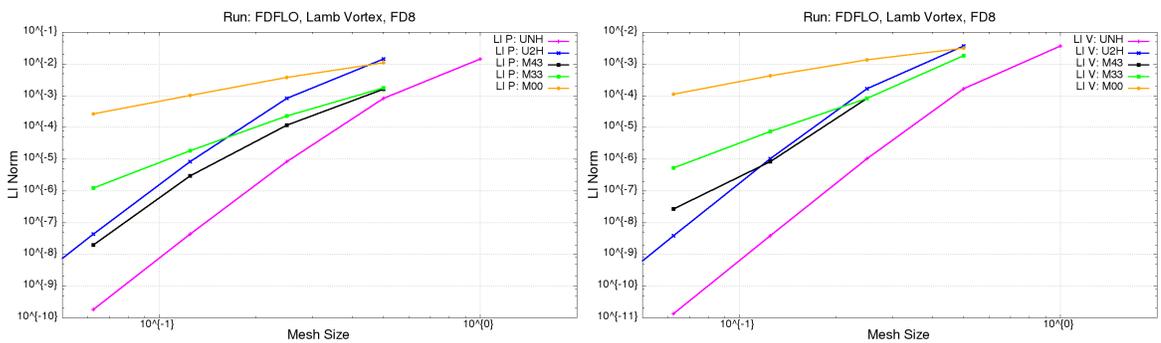
Figure 4.7: Stationary Lamb Vortex: L_∞ Convergence for 6th Order.



(a) Pressure

(b) Velocity

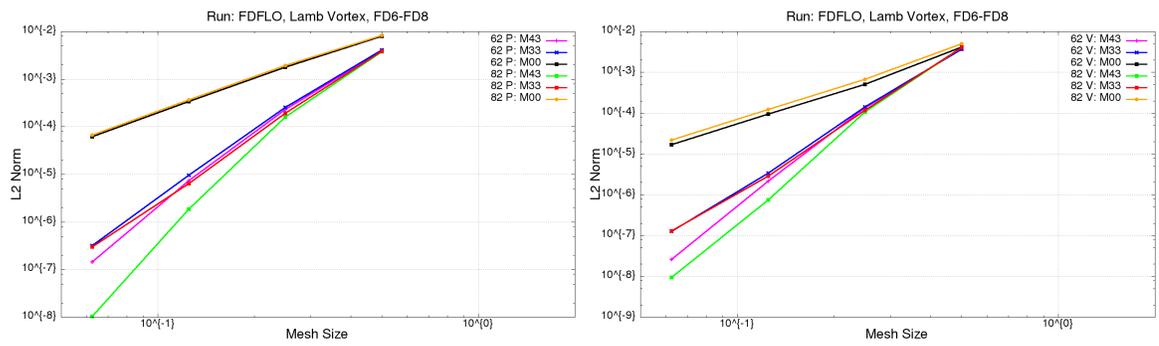
Figure 4.8: Stationary Lamb Vortex: L_2 Convergence for 8th Order.



(a) Pressure

(b) Velocity

Figure 4.9: Stationary Lamb Vortex: L_∞ Convergence for 8th Order.



(a) Pressure

(b) Velocity

Figure 4.10: Stationary Lamb Vortex: Comparison of the L_2 Norm for 6th and 8th Order.

4.1.3 3D Lamb Vortex

The idea is similar to the 2D Lamb vortex but with triple periodic condition and z -velocity $w = 0$. Two domains are defined, a coarser domain with coordinates $-6 \leq x, y, z \leq 6$ and an inner finer domain $-3 \leq x, y, z \leq 3$. The vortex is propagated $T = 120$ time units and the grid sizes are $h = 0.25$ and $2h = 0.5$.

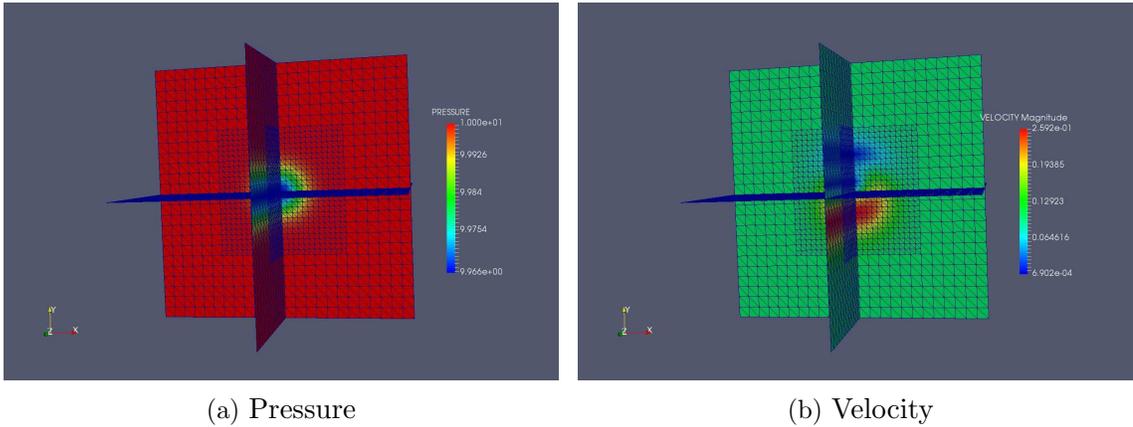


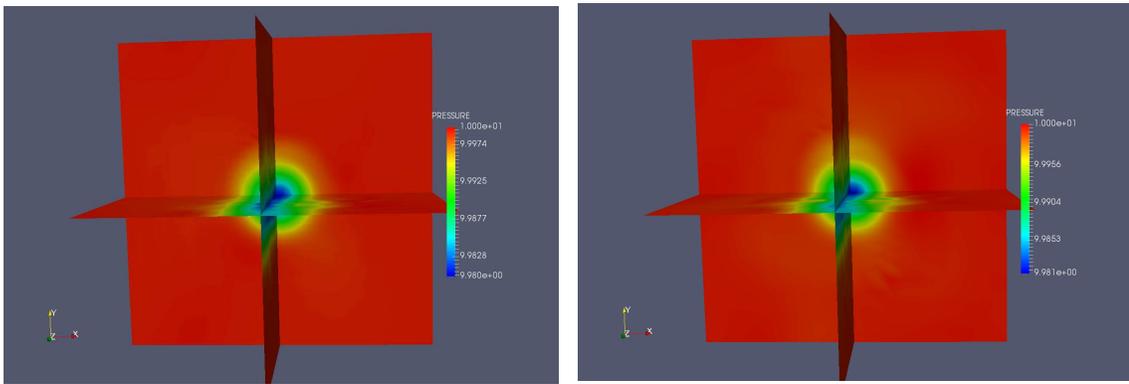
Figure 4.11: 3D Lamb Vortex: Initial Conditions, $h2h$ Mesh.

Figure 4.11 shows the initial condition and the mesh $h2h$ in 3D. For this situation a 4^{th} order polynomial interpolation is used for the points, but using 4 and 6 Halo points (HP) for the interface between the two meshes. The schemes adopted for this example were a 6^{th} order spatial discretization and a 6^{th} order low-storage Runge-Kutta for the timestepping. The L_2 and L_∞ norms are compared to a simple linear interpolation in Table 4.1.

It is important to note here that it is slightly better to use 4 Halo points instead of 6 Halo points for a 4^{th} order polynomial interpolation. However, a simple linear/bilinear/trilinear interpolation provides a better result for this problem with less computational power required. This is mainly because the z -velocity should be zero in the whole domain but due to interpolation in the interface area, the pressure gradient in the z -direction is not zero, leading to an increased error. It is necessary to continue working on better interpolation

Table 4.1: L_2 and L_∞ norms for the 3D Lamb Vortex.

norm/interpolation	linear	4 HP	6 HP
vel_{L_2}	1.0340	1.1106	1.2940
vel_{L_∞}	0.1454	0.1578	0.1767
$press_{L_2}$	0.1100	0.1107	0.1205
$press_{L_\infty}$	0.0192	0.0211	0.0208

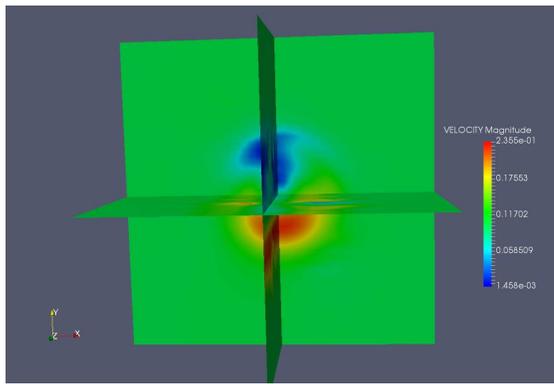


(a) Linear Interpolation

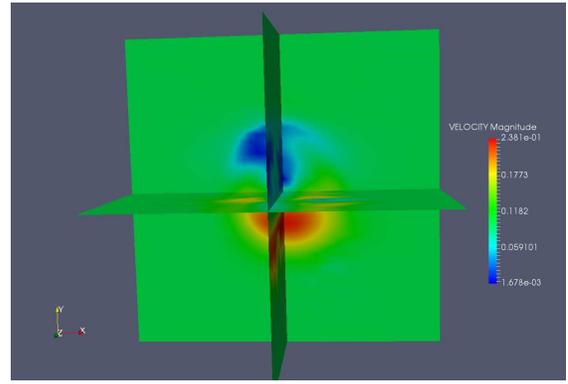
(b) 4th Order Interpolation with 4 Halo Points

Figure 4.12: 3D Lamb Vortex: Comparison of Pressure Magnitudes at $T = 120$.

schemes and limiters in order to obtain a more accurate solution and provide a competitive alternative to solve problems with mesh $h2h$ in 3D.



(a) Linear Interpolation



(b) 4th Order Interpolation with 4 Halo Points

Figure 4.13: 3D Lamb Vortex: Comparison of Velocities Magnitudes at $T = 120$.

4.1.4 2D Lamb Vortex in 3D

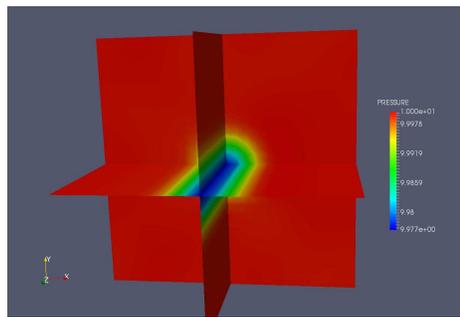
Intrigued by the unsatisfying results of the previous case, a 2D Lamb vortex was run in 3D. The x, y extent is the same as, the first case above, and the fine mesh extends all the way to z_{min}, z_{max} . Note that unlike in the 2D case one now has points that require interpolation along other faces and along the volume. As before, the vortex is propagated $T = 120$ time units and the grid sizes are $h = 0.25$ and $2h = 0.5$.

Figures 4.14–4.15 show the pressure and velocity obtained using an 6th order spatial discretization and a 6th order low-storage Runge-Kutta timestepping scheme. From left to right, the cases are: mesh $2h$, mesh $h2h$ (i.e. mesh h inside mesh $2h$) with usual bilinear interpolation, mesh $h2h$ with cubic interpolation, and mesh h . One can see that running on a $2h$ mesh yields better results than a $h2h$ mesh, whether the interpolation is bilinear or higher order. On the other hand, the high order interpolation yields better results than the linear interpolation.

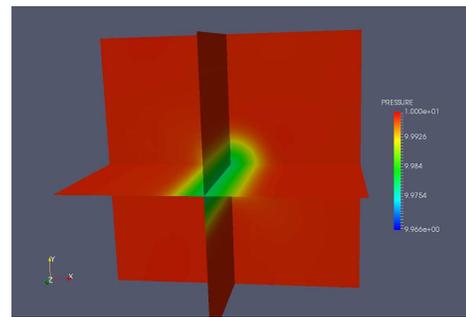
The L_2 and L_1 norms for all four combinations are compared in Table 4.2.

Table 4.2: L_2 and L_∞ norms for the 2D Lamb Vortex in 3D.

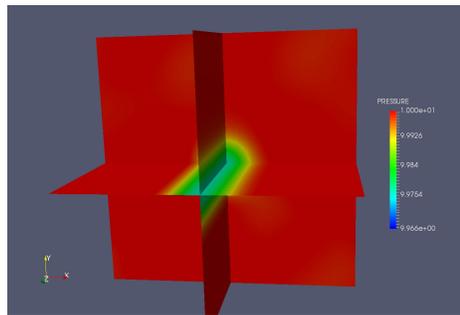
norm/interpolation	$2h$	$h2h$ lin	$h2h$ 4th	h
vel_{L_2}	0.3972	0.6987	0.4999	0.1263
vel_{L_∞}	0.0444	0.0517	0.0394	0.0112
$press_{L_2}$	0.0531	0.1174	0.0669	0.0167
$press_{L_\infty}$	0.0110	0.0134	0.0092	0.0018



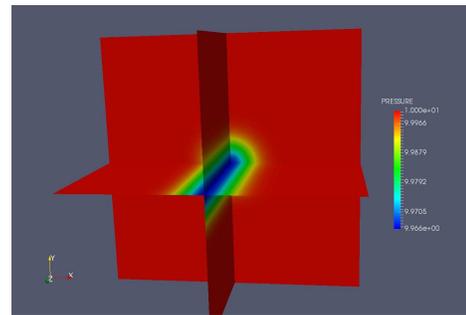
(a) Mesh $2h$



(b) Mesh $h2h$ *Bilinear*

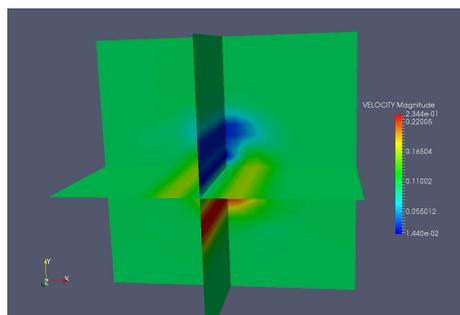


(c) Mesh $h2h$ *Quartic*

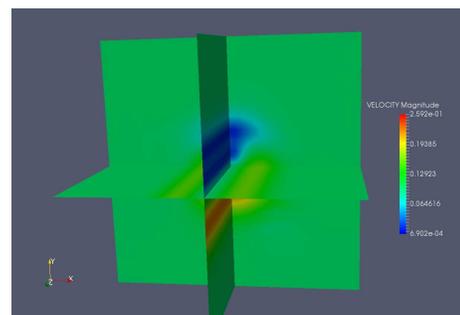


(d) Mesh h

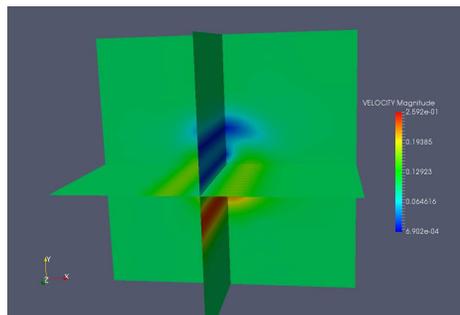
Figure 4.14: Lamb Vortex: Comparison of Pressures.



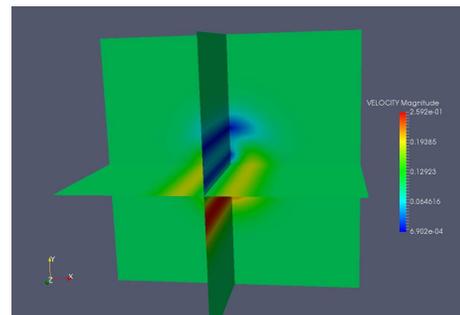
(a) Mesh $2h$



(b) Mesh $h2h$ Bilinear



(c) Mesh $h2h$ Quartic



(d) Mesh h

Figure 4.15: Lamb Vortex: Comparison of Pressures.

4.1.5 Cylinder

This classic testcase was added in order to show the effect of high-order interpolation for wake flows. The domain considered was $-4 \leq x \leq 10$, $-2 \leq y \leq 2$, with gliding wall boundary conditions at y_{min}, y_{max} , prescribed uniform inflow and prescribed pressure at outflow. As can be seen from Figure 4.16, the mesh consisted of 10 domains, with three levels of refinement: $\Delta x = 0.100, 0.050, 0.025$. The physical parameters were set as follows: $\rho = 1.0, \mathbf{v}_\infty = (1, 0, 0), \mu = 0.01, c = 5$, and the diameter of the cylinder was $d = 1.0$, yielding a Reynolds number of $Re = 100$. A 6-stage, low-storage RK scheme was used to integrate in time with a Courant number of $C = 0.4$. The immersed boundary option was used with a spatial discretization of 6th order. The case was run with the usual, low-order bi/trilinear interpolation, the high-order interpolation with and without the QMSL limiter and with a homogeneous fine mesh of $h = 0.025$. The results obtained for the latter one at $T = 100$ are shown in Figure 4.17. A number of station time history points were placed in the flow and the results recorded. Figures 4.18–4.19 show the values for the pressure, x - and y -velocities for two stations. The most pronounced difference can be observed in the pressures and x -velocities where the differences compared with the fine mesh results are around 15% of the peak value. These differences are larger for the points far away from the body. However, even the y -velocities show larger variations in time for the high-order interpolation, indicating less dissipation. Note also that a slight change of frequency is incurred when changing interpolation order. Due to the smooth behavior of the wake the interpolated and limiter results are very similar to the values obtained with the high order interpolation.

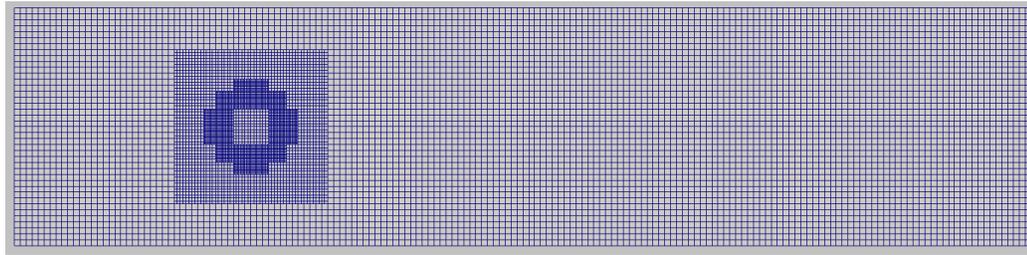


Figure 4.16: Cylinder: Grid System Used.

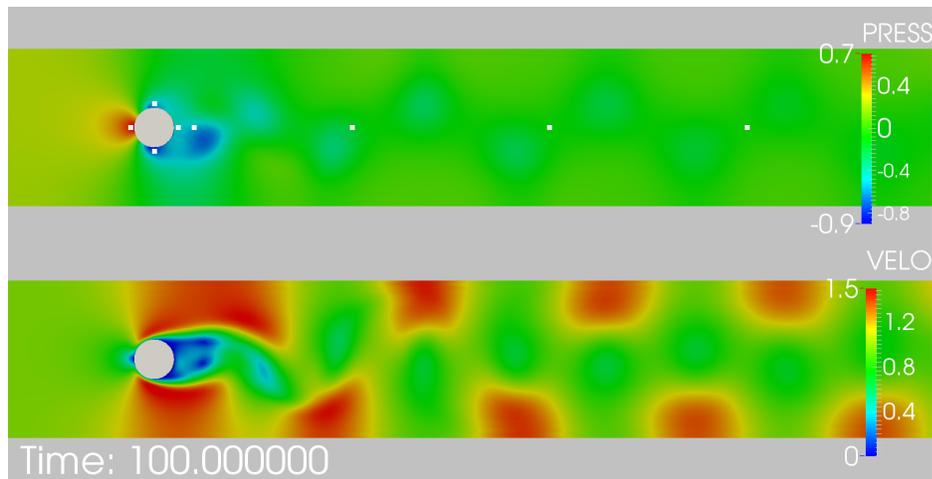


Figure 4.17: Cylinder: Results at Time $T = 100$.

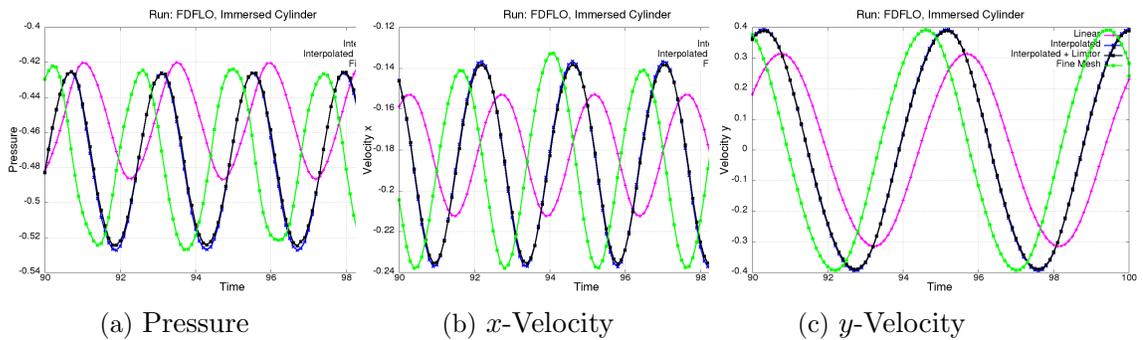
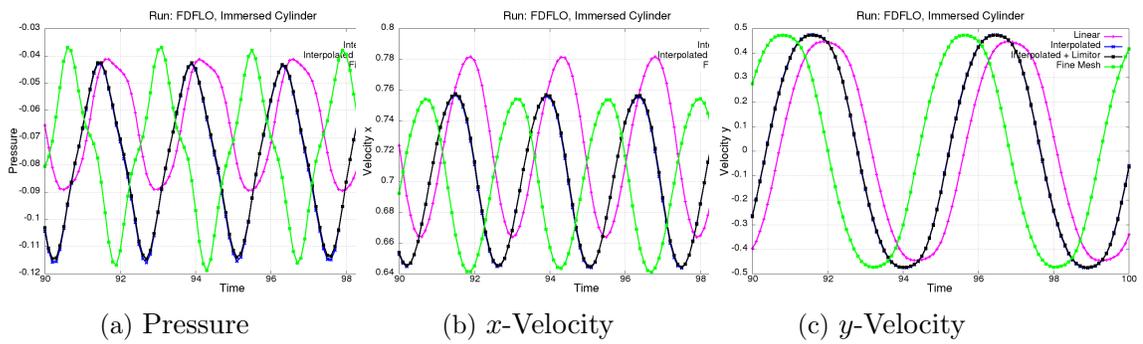


Figure 4.18: Cylinder: Station Time History for Station 5: $\mathbf{x} = (1, 0, 0)$.



(a) Pressure (b) x -Velocity (c) y -Velocity

Figure 4.19: Cylinder: Station Time History for Station 7: $\mathbf{x} = (10, 0, 0)$.

4.1.6 Flow Around a Sphere

In the previous subsection solutions of ‘simple’ 2D and 3D problems solved with a high order FD solver were shown. To study the behavior of more complex geometries, flow around a sphere before the drag crisis at two different Reynolds numbers ($Re = 3 \cdot 10^3$ and $Re = 10^5$) is used as a test case.

The sphere considered is of diameter $d = 1m$ and is embedded in a domain of size $-5.2m \leq x \leq 11.5m$, $-3.8m \leq y, z \leq 3.8m$. The minimum Δx is $0.01m$ in the fine mesh, $0.03m$ in the medium mesh and $0.04m$ in the coarse mesh. The system of nested cartesian grids with 5 levels of refinement and the coarse mesh are shown in Fig. 4.20 and Fig. 4.21, respectively. The total number of active points in the fine mesh is $nactv = 20,698,271$, in the medium mesh $nactv = 4,088,775$ and in the coarse mesh $nactv = 2,796,928$. The initial conditions are $\rho = 1kg/m^3$, $u = 1m/s$, $c_{sound} = 10m/s$ and $C_s = 0.1$.

For the low Reynolds situations the obtained solution is compared to data from [64] and to the recommended drag correlation presented in [65] for $260 \leq Re \leq 1500$,

$$\log_{10}C_d = 1.6435 - 1.1242w + 0.1558w^2, \quad (4.2)$$

where $w = \log_{10}Re$.

Results for the high Reynolds case are compared to experimental data from [2], a numerical solution obtained with a Lattice Boltzmann solver [3] and the following drag correlation from [65], valid for $1.5 \cdot 10^3 \leq Re \leq 1.2 \cdot 10^4$,

$$\log_{10}C_d = -2.4571 + 2.5558w - 0.9295w^2 + 0.1049w^3. \quad (4.3)$$

Three cartesian meshes are used in the Lattice Boltzmann simulation: a coarse grid with 410 nodes, a medium grid with 512 nodes and a medium grid with 640 nodes across the sphere diameter, leading to a total of 40,769,886, 73,855,027 and 133,438,032 nodes, respectively.

The simulation with FDFLO is solved with a second order LSRK, two fourth order RK

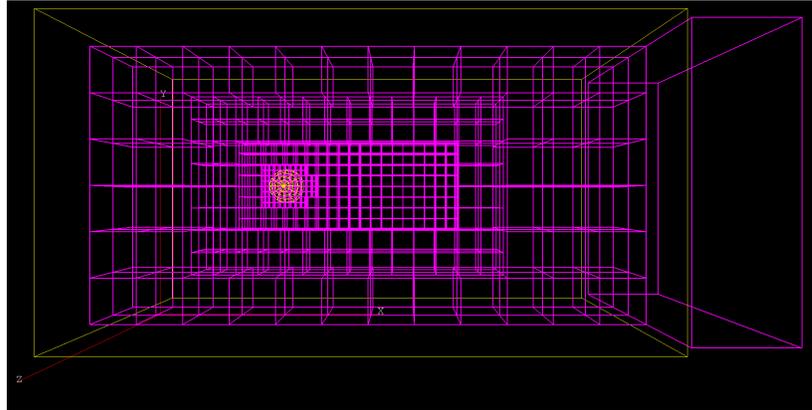


Figure 4.20: System of Nested Cartesian Grids Used for the Simulation of Flow Around a Sphere.

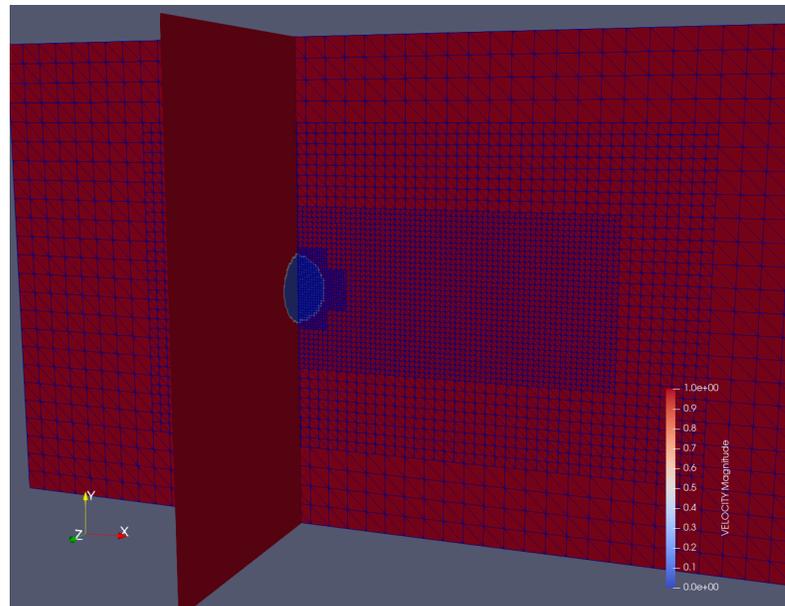


Figure 4.21: Coarse Mesh and Velocity Initial Condition.

schemes in time (LSRK and 2 Step RK), an eight stage LSRK scheme and three spatial discretizations (2^{nd} , 4^{th} and 8^{th} order). The variables used for diagnostics are the drag coefficient C_d and the pressure distribution around the sphere.

Results at Reynolds Number $3 \cdot 10^2$

Between Reynolds numbers 270 and 1000 the flow through a sphere becomes unsteady. This unsteadiness occurs as a waviness in the wake past the sphere. Table 4.3 displays the drag coefficient C_d and run times for several discretizations and mesh configurations. In [64], the computed average C_d is 0.656 and the value obtained with the formula given in [65] is $C_d = 0.6527$. Other experimental data [66] result in an interpolated value of 0.629 and [67] obtained $C_d = 0.671$. In order to obtain a averaged solution of the drag coefficient this test is run for $t = 70sec$, although only the last $40sec$ are used for the average C_d . The entirety of these simulations were run in four nodes. Each node has two IBM Power9 8335-GTH with a total CPU count of 80 physical cores.

Table 4.3: Results for the Sphere Case at $Re = 3 \cdot 10^2$.

FD Order	Time Discretization	Mesh	Courant	C_d	Time (secs)
2 nd Order	4 nd LSRK	Medium	0.2	0.675	4020
4 th Order	4 th LSRK	Medium	0.2	0.657	4258
4 th Order	4 th LSRK	Medium	0.25	0.660	3829
4 th Order	2Step RK	Medium	0.3	0.658	5297
4 th Order	2Step RK	Medium	0.5	0.658	3202
8 th Order	8 th LSRK	Medium	0.3	0.656	7610

All the schemes achieved an acceptable C_d in the medium mesh, however the costs are almost double with the eighth order method compared with second or fourth order schemes. The 2Step Runge Kutta method is more expensive than the regular fourth LSRK at similar Courant number. However if the Courant number is increased, due to a larger stability zone, 2Step RK can be considerable faster than the low storage variant. The highest stable Courant number for the fourth order LSRK found is 0.25.

Figure 4.23 demonstrates visually that the results in the coarser mesh are correct. Yet C_d and C_p values are far from the experimental values. This is mainly due to a coarser

discretization of the sphere and an incorrect representation of the forces over the sphere. Discretization of embedded and immersed surfaces for higher order schemes requires further detailed examination and testing. This is ongoing research.

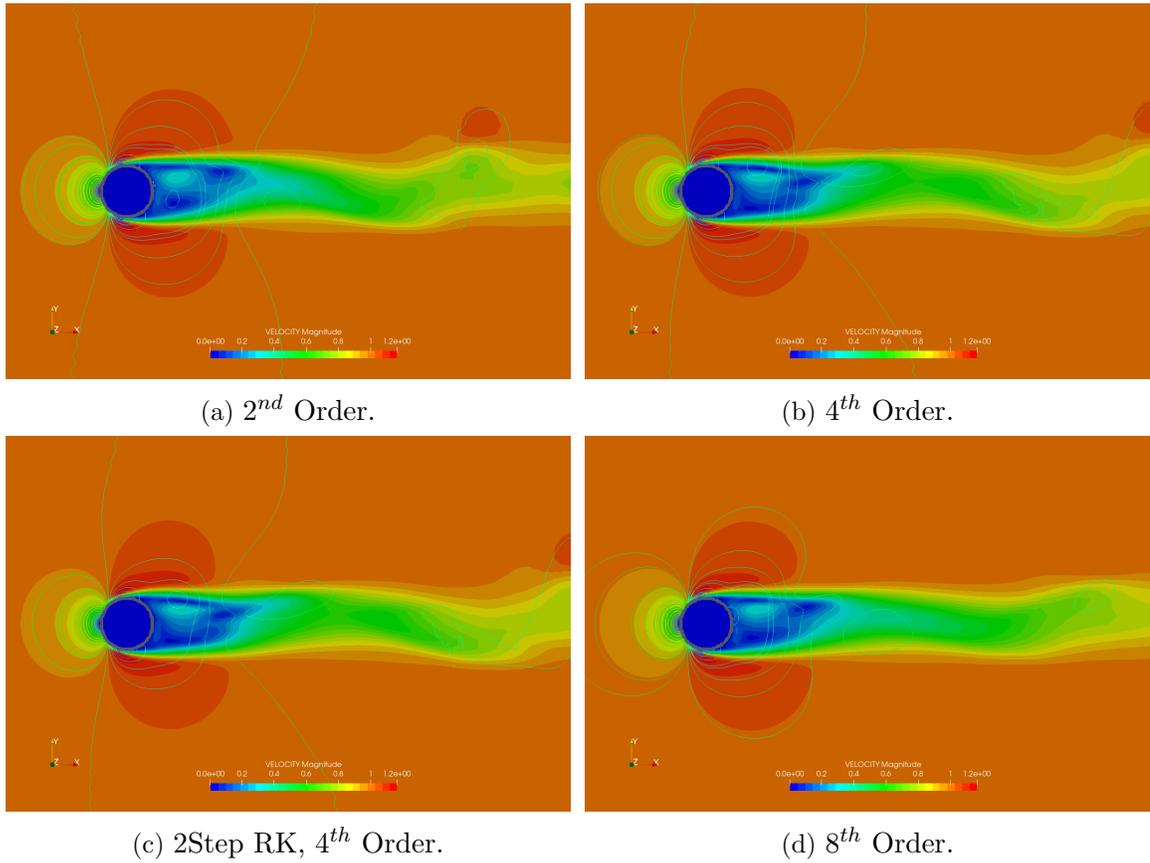
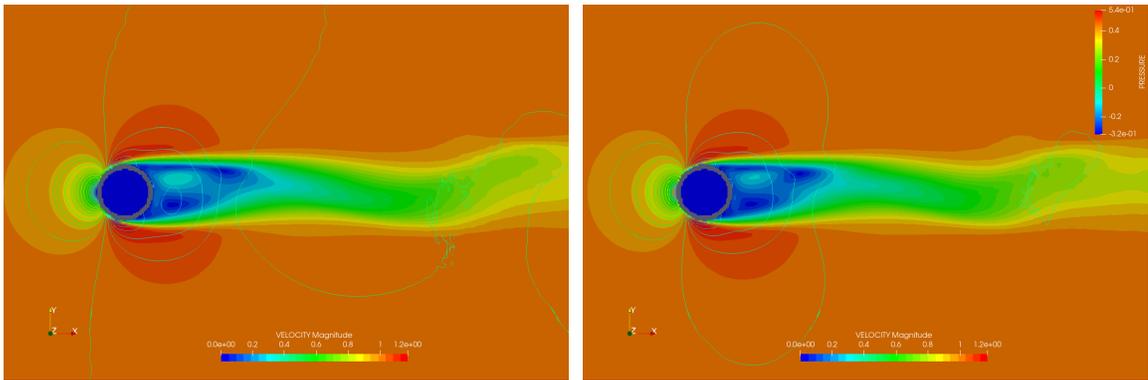
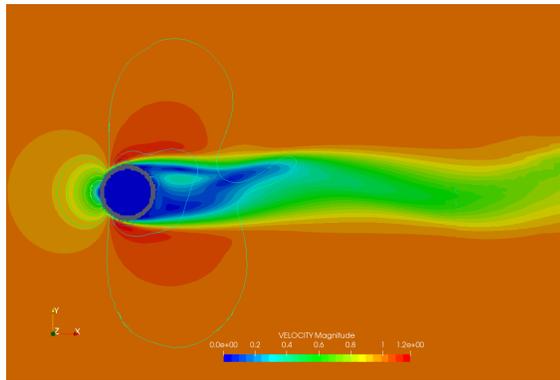


Figure 4.22: Velocity Field and Pressure Contour at $t = 70\text{secs}$, Medium Mesh.



(a) 4th Order.

(b) 2Step RK, 4th Order.



(c) 8th Order.

Figure 4.23: Velocity Field and Pressure Contour at $t = 70secs$, Coarse Mesh.

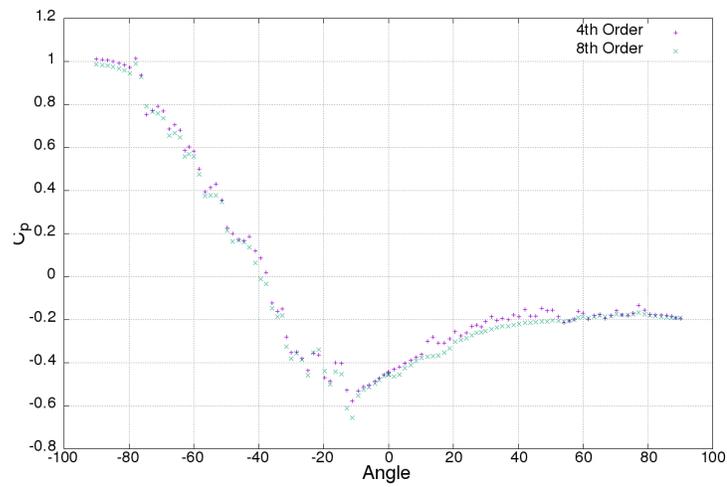


Figure 4.24: Instantaneous Pressure Coefficient at $t = 70secs$, Medium Mesh.

Results at Reynolds Number 10^4

At Reynolds $Re = 10^4$ the shear layer and the separation are in laminar conditions, however downstream the wake becomes fully turbulent and unsteady. The formula presented in [65] gives a C_d of 0.405 with the prescribed Reynolds number. The obtained drag coefficients are displayed in Fig. 4.25 and are compared with measurements from [3] and [2].

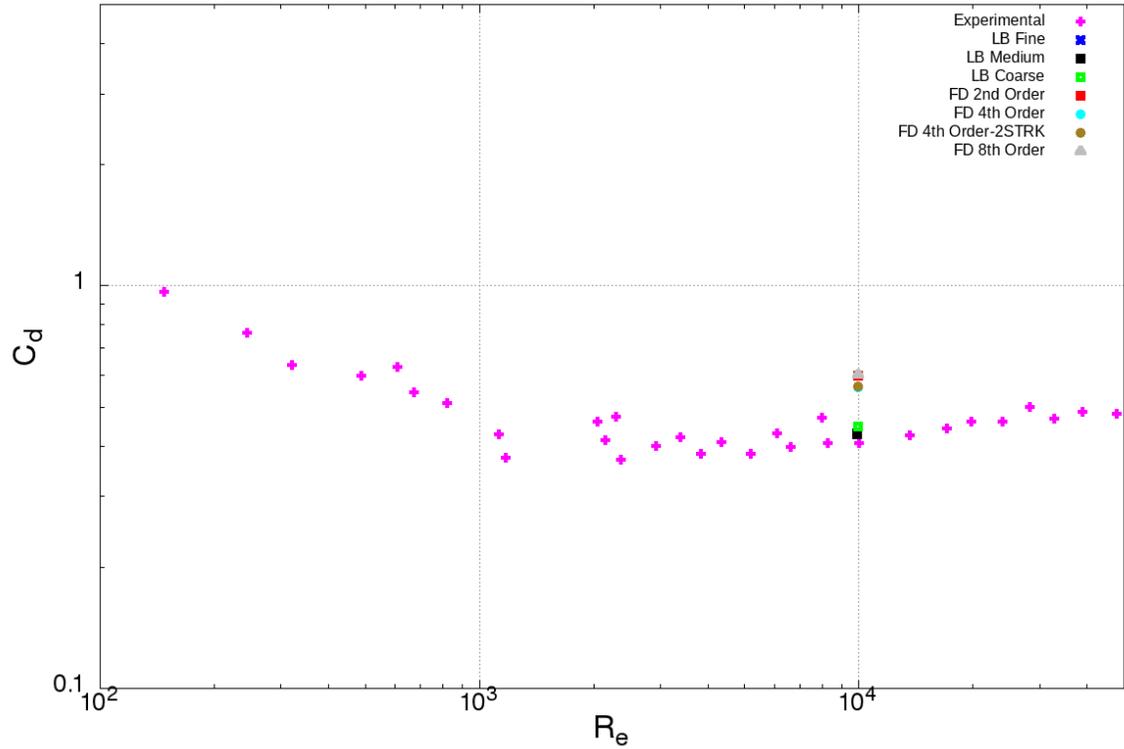


Figure 4.25: Averaged Drag Coefficient, C_d . Experimental Data from [2] is Compared With Lattice-Boltzman Method LB [3] on a Fine, Medium and Coarse Mesh, and Several Finite Difference Discretization Orders FD.

Results shown in Figure 4.25 are not close to the drag coefficients obtained experimentally and with a Lattice Boltzmann method. One of the causes of this are that despite the mesh used is the ‘finest possible’ mesh based on hardware limitations, it is still coarse for this simulation. With a minimum $\Delta x = 0.01cm$ the obtained y^+ is 6 which is not enough

to represent the boundary layer correctly. On the other hand, in the LB simulation on the coarser mesh, with $y^+ \sim 1.5$, the obtained C_d is still reasonably higher compared to the experimental measurement. In this example there are no clear advantages using higher discretization methods and, as noted before, more effort has to be made on how to compute the unknowns at the body boundaries. Despite this, the characterization of the flow past a sphere at $Re = 10000$ is correct. Close to the sphere there is a laminar area and in the wake area the flow becomes fully turbulent. This phenomena is visualized in Fig. 4.26.

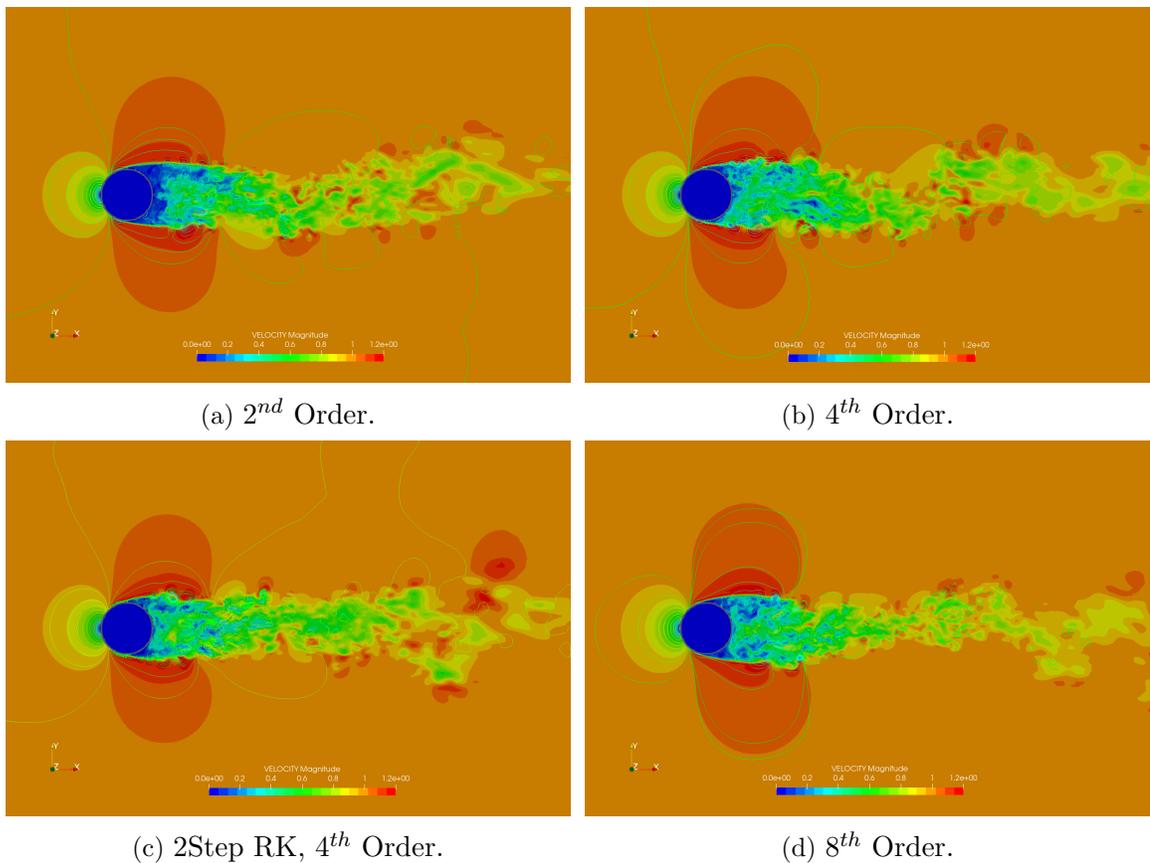


Figure 4.26: Velocity Field and Pressure Contour at $t = 30secs$, Fine Mesh.

Due to limited computing resources this test is run for $t = 30sec$, however only the last $10sec$ are used for the average C_d calculation. These simulations were run on 16 nodes, with

a total CPU count of 640 physical cores. As is reported in Table 4.4, 2Step RK order has two main advantages when compared to low storage RK schemes; it has a larger stability zone (i.e. a higher stable Courant number can be utilized) and fewer communication interactions between computing nodes are needed. This second characteristics is more relevant when the number of used nodes are higher. An example of this can be seen comparing two schemes with a fourth order discretization in space and different RK schemes where the difference in run time is $\sim 15\%$. The variation in run time between a second order and a fourth order (2Step RK) in space is $\sim 8\%$.

Table 4.4: Required Time for Solving a Sphere at $Re = 1 \cdot 10^4$ for 10secs.

FD Order	Time Discretization	Mesh	Courant	Time (secs)
2 nd Order	4 nd LSRK	Fine	0.2	4210
4 th Order	4 th LSRK	Fine	0.2	4635
4 th Order	2Step RK	Fine	0.5	3875
8 th Order	8 th LSRK	Fine	0.3	16457

Chapter 5: Energy Cascade

5.1 Introduction

In this chapter, the behavior of finite difference methods in low Mach number compressible turbulent flows is analyzed. The benchmark used is the Taylor Green Vortex (TGV) at $Re = 1600$ in 3D, which represents how a simple initial flow condition causes a fully isotropic decay to turbulent state in time. This case has been used repeatedly as a validation for high order schemes, see for example [4, 68–70]. Furthermore it has been presented as a difficult and baseline type test case in the first, second, fourth and fifth International Workshop on High Order CFD Methods.

The objective of this chapter is to show how FD methods predict the turbulent energy cascade and the main flow characteristics on different uniform cartesian meshes, and compare with the available data.

5.2 Problem Set-up

The geometry of the problem consists in a triply periodic domain with size $0 \leq x, y, z \leq 2\pi$ with initial conditions,

$$u(t_0) = u_0 \sin(x/L) \cos(y/L) \cos(z/L), \quad (5.1)$$

$$v(t_0) = -u_0 \cos(x/L) \sin(y/L) \cos(z/L), \quad (5.2)$$

$$w(t_0) = 0, \quad (5.3)$$

$$p(t_0) = p_0 + \rho_0 \left(\cos\left(\frac{2x}{L}\right) + \cos\left(\frac{2y}{L}\right) \right) \left(\cos\left(\frac{2z}{L}\right) + 2 \right) / 16, \quad (5.4)$$

with constants $L = 1$, $u_0 = 1$, $\rho_0 = 1$ and $p_0 = 100$. The velocity of sound is $c_s = 10$, the viscosity is set to 6.2510^{-4} in order to obtain the desired Reynolds number $Re = 1600$ and the total simulation time is $t = 20$. Three meshes are used in the simulations: a coarse mesh of 100^3 nodal points, a medium mesh of 200^3 nodal points and a fine mesh of 400^3 nodal points. Initial conditions and the outline of the coarse mesh are shown in Fig. 5.1.

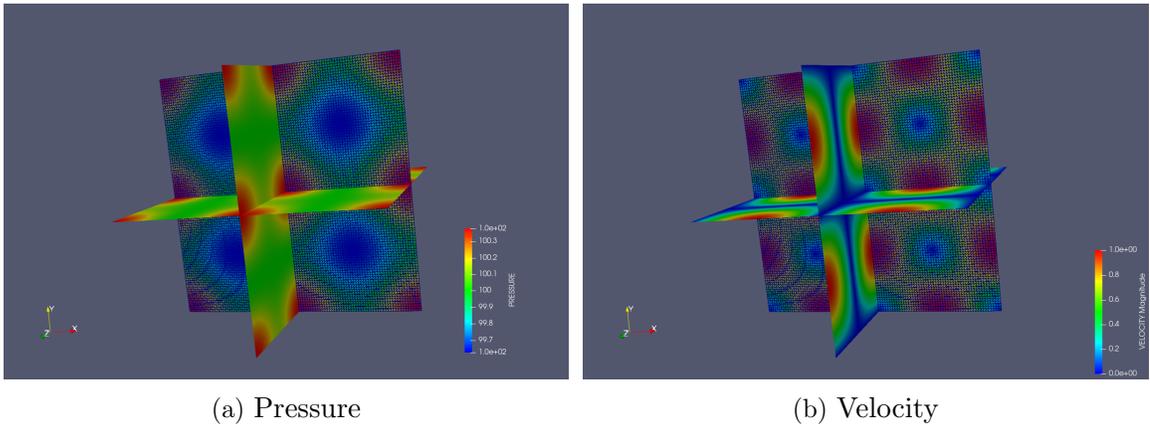


Figure 5.1: Taylor Green Vortex: Initial Conditions.

5.3 Results

Simulations solved with second and fourth order spatial discretization are handled with a fourth order LSRK scheme and the results with a eight order spatial discretization make use of a eight order LSRK scheme.

The expected results as time advances consist of a series of key physical processes in turbulence: vortex roll-up, vortex stretch and interaction, and finally total dissipation of the energy in the fluid leading to a steady state. Solutions obtained at $t = 8secs$ and $t = 16secs$

using several FD orders of approximation and mesh sizes are shown in Figures 5.2, 5.4, 5.3 and 5.5.

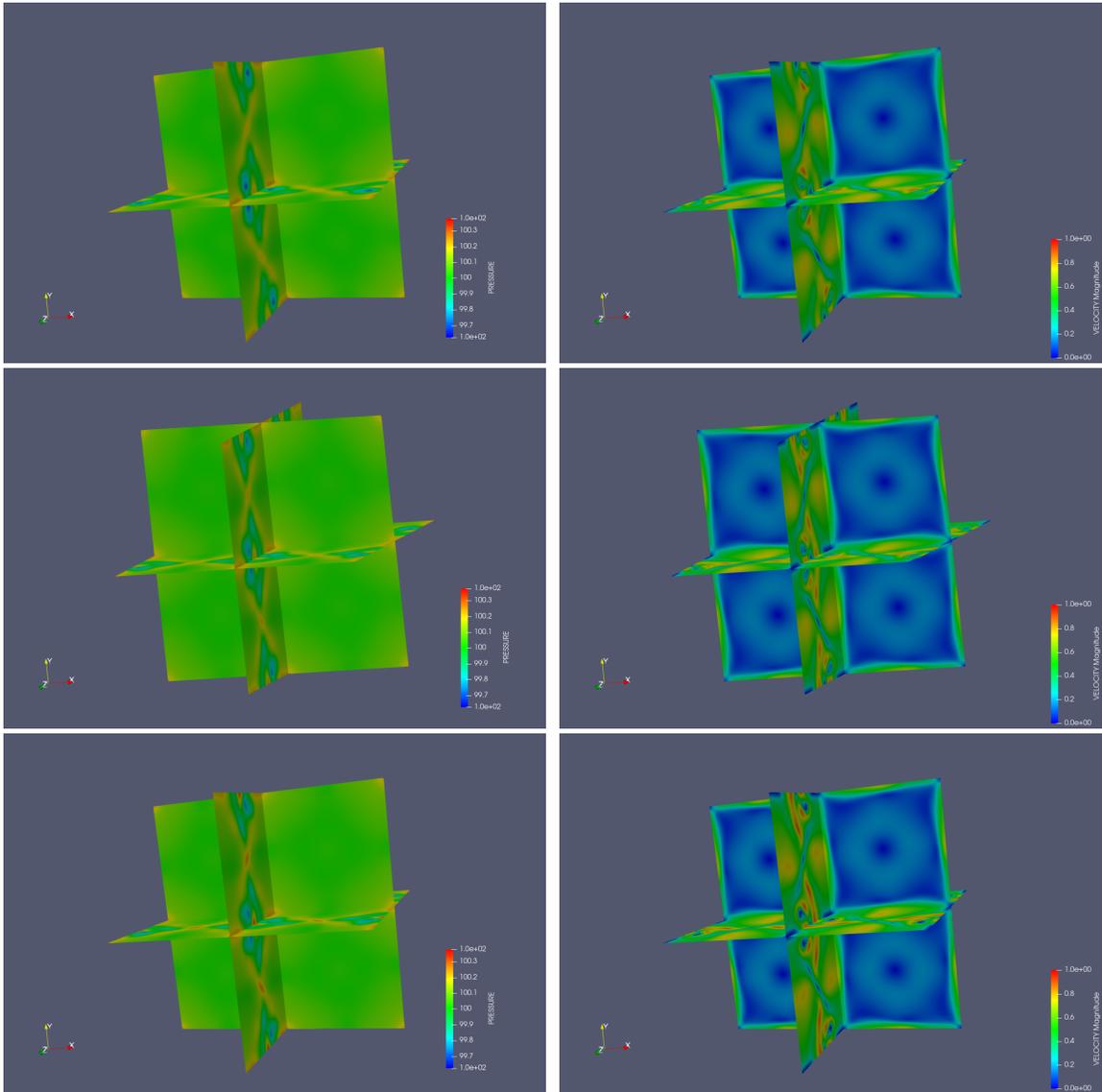


Figure 5.2: Taylor Green Vortex: Pressure (Left) and Velocity (Right) at $t = 8\text{secs}$, Coarse Mesh. Top: 2^{nd} Order, Middle 4^{th} Order, Bottom: 8^{th} Order FD Approximation.

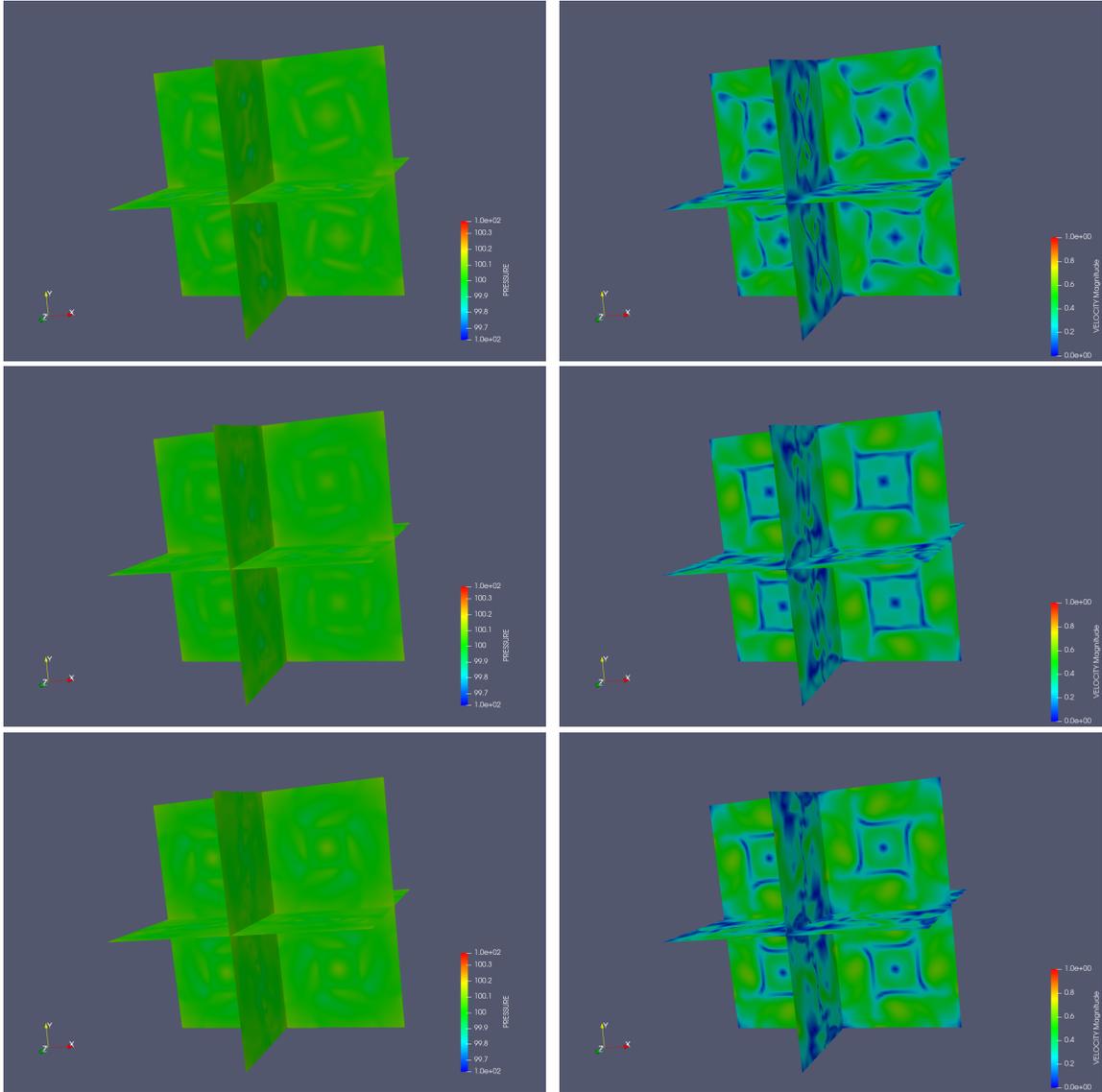


Figure 5.3: Taylor Green Vortex: Pressure (Left) and Velocity (Right) at $t = 16\text{secs}$, Coarse Mesh. Top: 2^{nd} Order, Middle: 4^{th} Order, Bottom: 8^{th} Order FD Approximation.

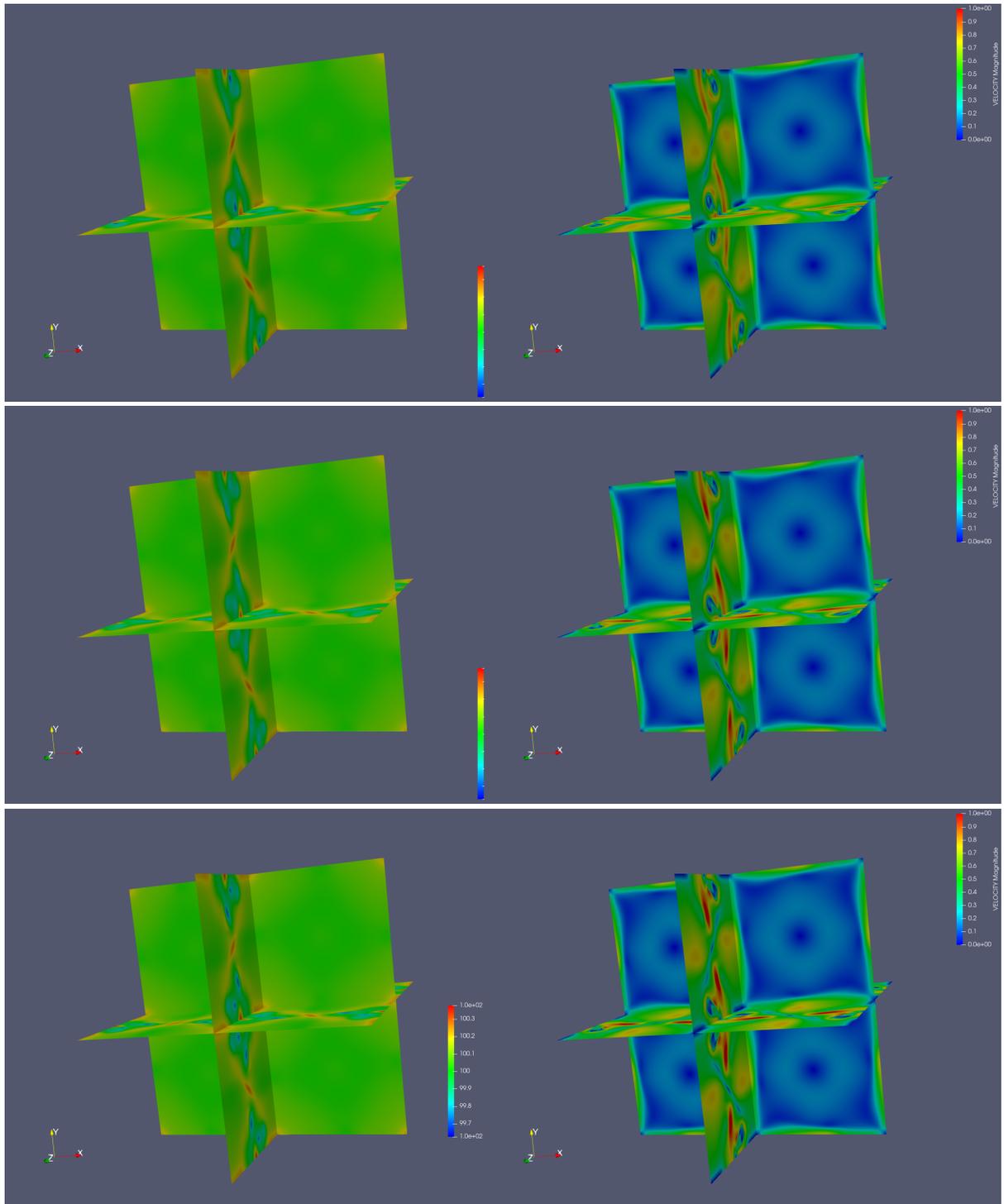


Figure 5.4: Taylor Green Vortex: Pressure (Left) and Velocity (Right) at $t = 8$ secs, Fine Mesh. Top: 2nd Order, Middle: 4th Order, Bottom: 8th Order FD Approximation.

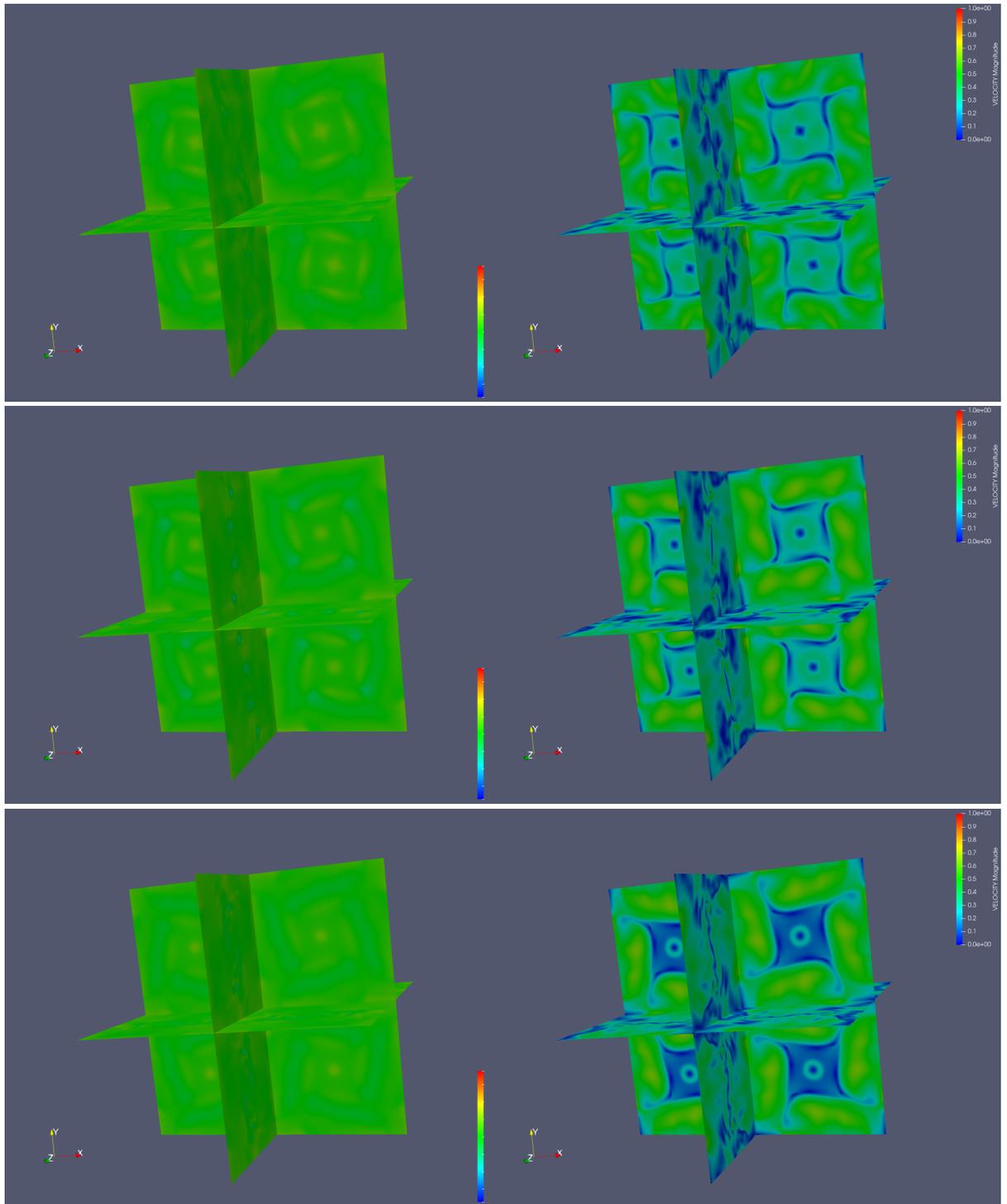


Figure 5.5: Taylor Green Vortex: Pressure (Left) and Velocity (Right) at $t = 16$ secs, Fine Mesh. Top: 2nd Order, Middle: 4th Order, Bottom: 8th Order FD Approximation.

Several volume averaged quantities are compared with other published results:

- The average kinetic energy,

$$E_k = \frac{1}{\rho_0 V} \int_V \frac{1}{2} \rho \mathbf{u} \cdot \mathbf{u} dV, \quad (5.5)$$

- The energy dissipation rate,

$$D_e = -\frac{dE_k}{dt} \quad (5.6)$$

- The vorticity dissipation rate, which is a measure of how well the inertial range of turbulence is resolved,

$$D_v = \frac{2\mu}{\rho_0} \frac{1}{\rho_0 V} \int_V \frac{1}{2\rho} \boldsymbol{\omega} \cdot \boldsymbol{\omega} dV, \quad (5.7)$$

$$\omega_i = \varepsilon_{ijk} \frac{\partial u_k}{\partial x_j}, \quad (5.8)$$

where V is the total volume, $\boldsymbol{\omega}$ is the vorticity and μ is the dynamic viscosity. Fig. 5.6 shows the average kinetic energy with 2^{nd} , 4^{th} and 8^{th} order FD using a coarse (100^3 nodes), medium (200^3 nodes) and fine mesh (400^3 nodes). Results are compared with [71] which uses a ‘Dispersion-Relation-Preserving’ (DRP) scheme on a 512^3 element mesh and [70] which implements a Flux Reconstruction scheme that recovers a Spectral Difference method (FR-SD).

In [70], Bull noted that $D_e - D_v$ is an important error measure and corresponds to an estimation of the numerical dissipation introduced by the numerical method selected. In Fig 5.7, the vorticity, kinetic energy and numerical dissipation rate obtained using different approximation orders and mesh sizes are compared.

As expected when the order of the approximation or the size of the mesh increase the results approach the solution provided by the DRP scheme on a 512^3 mesh. Furthermore,

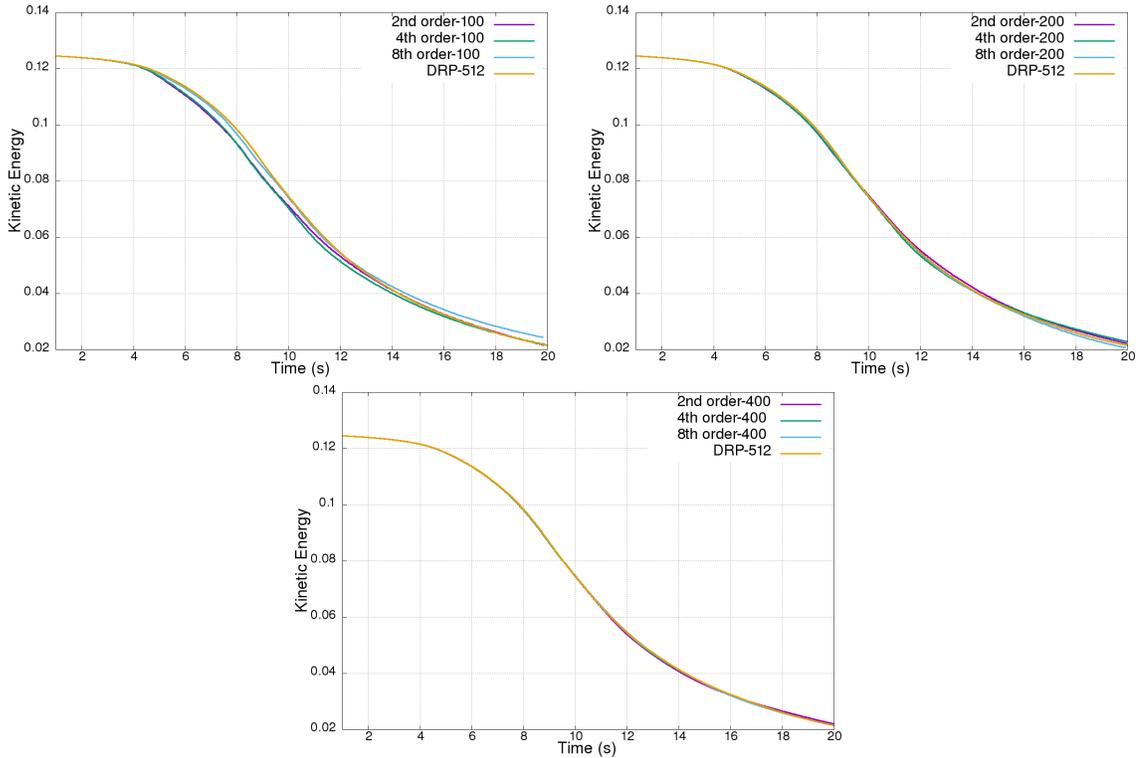


Figure 5.6: Taylor Green Vortex: Kinetic Energy vs Time on a Coarse (100^3 Nodes), Medium (200^3 Nodes) and Fine (400^3 Nodes) Mesh.

the numerical dissipation decreases and, as can be seen for the 400^3 mesh, is similar to the numerical dissipation introduced by the FR-SD method in a 64^3 mesh with a p_3 solution (i.e. corresponding to a mesh of 256^3 degrees of freedom).

Another important observation contained in Fig. 5.7 is that 8^{th} order FD approximations are similar to 2^{nd} order FD approximations with a mesh size 2^3 times coarser. Figure 5.8 substantiates the last statement by comparing kinetic energy, vorticity and numerical dissipation rate using the 2^{nd} and 8^{th} order FD method and varying the mesh size.

Simulation with a coarse and medium grid were performed on a Intel Xeon E5-2637 v4 and the runtimes are shown in Table 5.1. If the mesh size is maintained, an eight order FD simulation is around 2.8 – 3.5 times slower than a second order simulation. However

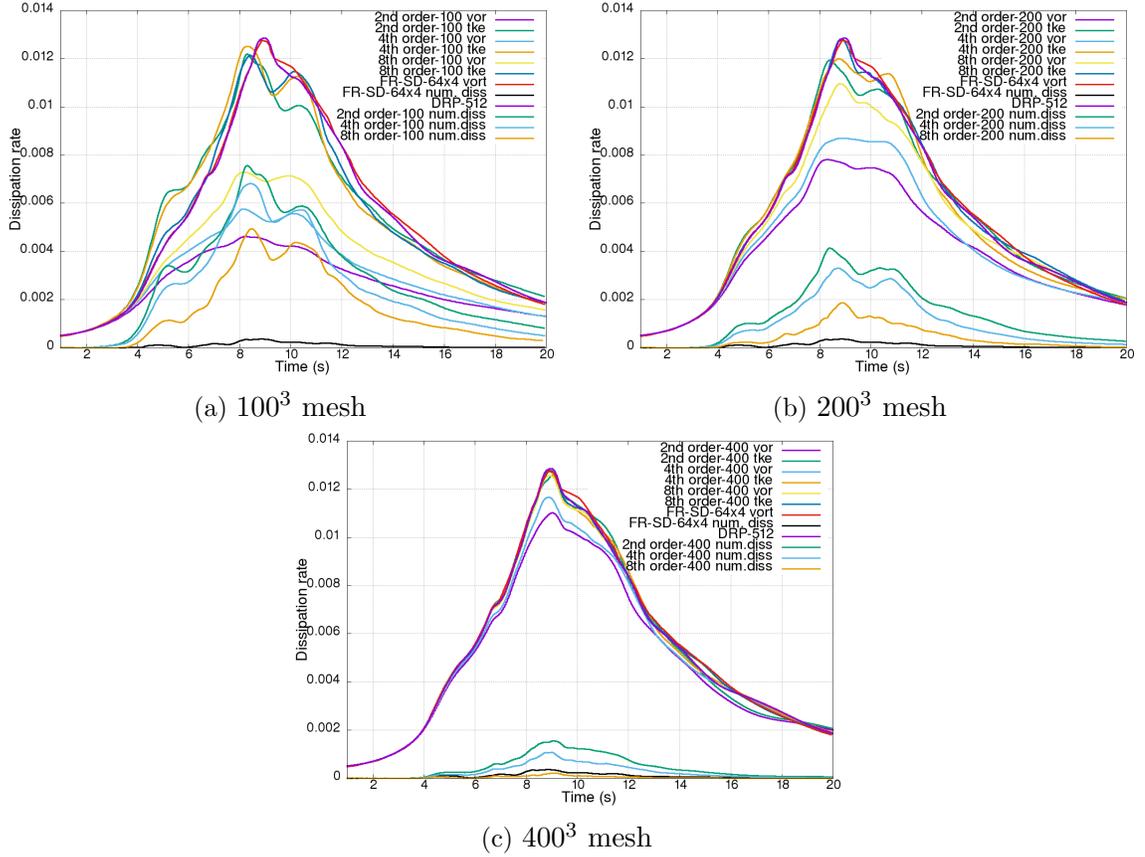


Figure 5.7: Taylor Green Vortex: Kinetic energy (tke), vorticity (vor) and numerical (num. diss) dissipation rate for different approximation orders and mesh sizes. ‘FR-SD-63X4’ is a p_3 solution using the FR-SD method on a 64^3 mesh.

if the approximation order is constant now and the mesh size is increased 2^3 times, the computation time increases by a factor of roughly $2^3 * 2 = 16$ (the last 2 is due to the reduction in the timestep).

In conclusion, there is a clear advantage of using high order FD methods for unsteady turbulent flows compared with solutions obtained with low order FD methods in finer meshes. Run times are around 5 times faster for similar solution properties while maintaining hardware configuration and simulation time.

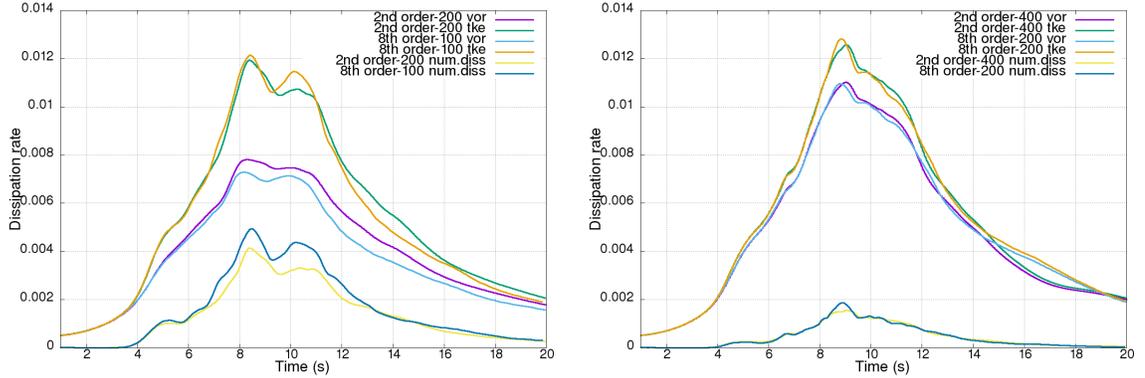


Figure 5.8: Taylor Green Vortex: Kinetic energy (tke), vorticity (vor) and numerical (num. diss) dissipation rate vs time. Left: Comparison between a 2^{nd} order FD approximation on a 200^3 mesh and a 8^{th} order FD approximation on a 100^3 mesh. Right: Comparison between a 2^{nd} order FD approximation on a 400^3 mesh and a 8^{th} order FD approximation on a 200^3 mesh.

5.3.1 Energy Cascade

The energy cascade allows the observation of how the transfer of energy from large scales to the small scales of motion occurs. This concept is important in the theory of turbulence and displays how eddies generated in turbulent conditions dissipate at the Kolmogorov microscales for the most part due to viscosity.

The energy spectrum of turbulence $E(k)$ is obtained as a function of the wave number k . Low wave numbers represent the large eddies and high wave numbers the small eddies. The $E(k)$ is associated to the mean turbulence kinetic energy as,

$$\int_0^{\infty} E(k)dk = 1/2(\overline{u_i u_i}). \quad (5.9)$$

The transfer of energy from the low to the high wave numbers represented in the energy cascade is performed at a certain rate. This is supported by Kolmogorov's hypothesis,

$$E(k) = C\varepsilon^{2/3}k^{-5/3}, \quad (5.10)$$

Table 5.1: Runtimes for the Taylor Green Vortex.

Mesh Size	FD Order	Time (sec)	Factor
100 ³	2 nd Order	1542	
	4 th Order	1936	1.26
	8 th Order	5314	3.45
200 ³	2 nd Order	28840	
	4 th Order	39362	1.36
	8 th Order	82296	2.85

where C is a constant and ε is the dissipation rate. To obtain the flow of energy of the domain in the wave number space and calculate the "Energy Cascade" it is necessary to perform a Discrete Fourier Transform (DFT). This is performed here using the 'Fastest Fourier Transform in The West' [72], a C library of routines for computing DFT in one or more dimensions. The DFT of a 1D real array X of size n is,

$$Y_k = \sum_{j=0}^{n-1} X_j e^{-2\pi j k \sqrt{-1}/n}, \quad (5.11)$$

where Y_k is the unnormalized output array. The highest resolvable wave number is defined by the relation $k_{max} = NDOF/2$, where $NDOF$ are the number of degree of freedom in 1D. After each velocity component u_i is transformed into the Fourier space \hat{u}_i , then the energy spectrum is obtained as,

$$E(k) = \int \int \hat{u}_i \hat{u}_i^* dA(k), \quad (5.12)$$

where \hat{u}_i^* is the conjugate of \hat{u}_i and $dA(k) = 4\pi k^2 dk$.

Figure 5.9 displays how the numerical methods implemented follow the reference solution presented in [4] and compared against the Optimized Flux Reconstruction schemes

presented in [70]. This energy spectrum is obtained at $t = 9secs$ close to the peak of the dissipation rate. Note how well higher order methods maintain the energy content of the high wave number scales on coarse meshes and when the number of DOF increases, the energy at low scales gets closer to the reference solution. The Kolmogorov length scale at $t = 9secs$ for each spatial discretization method and mesh are shown in Table 5.2 .

Table 5.2: Kolmogorov Length Scales, at $t = 9secs$.

Mesh Size	FD Order	Length
100 ³	2 nd Order	0.01207
	4 th Order	0.01217
	8 th Order	0.01218
200 ³	2 nd Order	0.01212
	4 th Order	0.01198
	8 th Order	0.01178
400 ³	2 nd Order	0.01181
	4 th Order	0.01178
	8 th Order	0.01176

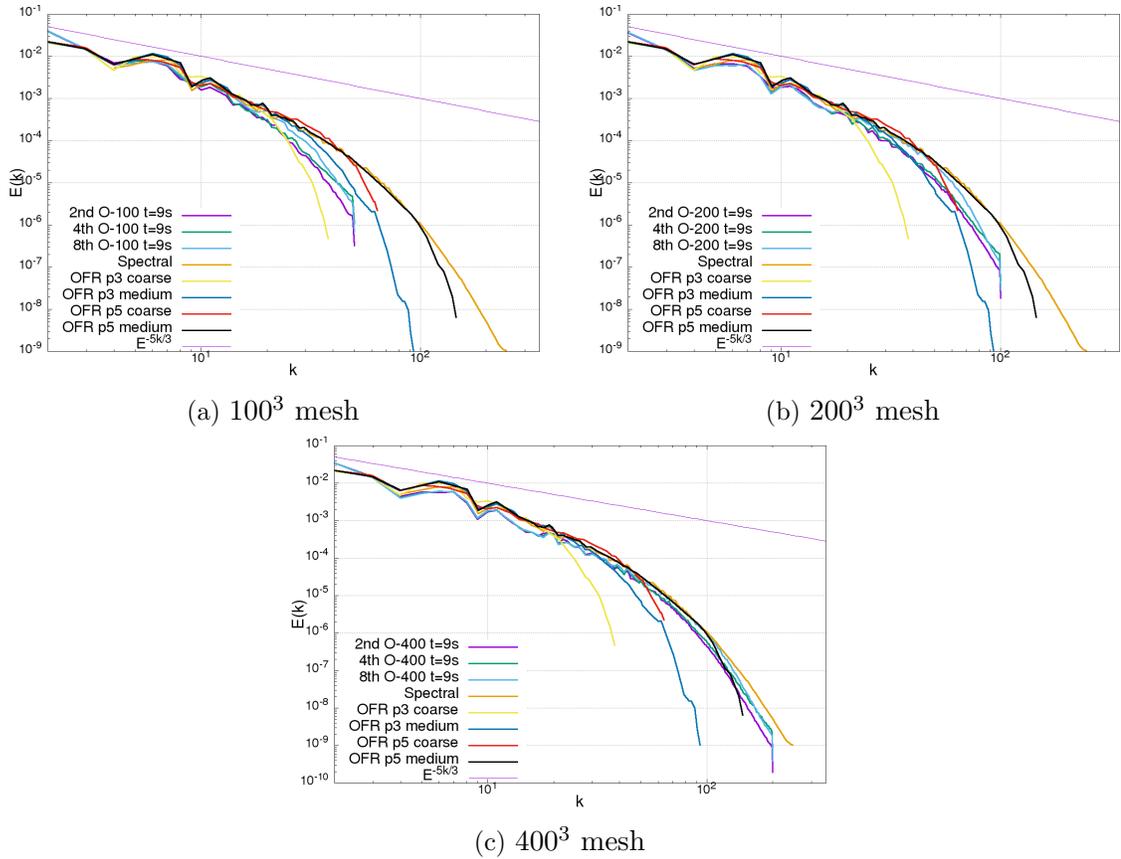


Figure 5.9: Taylor Green Vortex: Energy cascade for different approximation orders and mesh sizes at $t = 9secs$. ‘OFR’ refers to an optimized FR scheme on a coarse (16^3) and medium (32^3) mesh with a p_3 and p_5 solution. Spectral refers to the reference solution obtained in [4] with a Spectral method. The last value corresponds to the slope value predicted by the Kolmogorov hypothesis.

Chapter 6: The Ahmed Body

6.1 Introduction

The Ahmed Body was first presented as a basic ground vehicle type by Ahmed et al. [6]. It consists in a bluff body and is used as a benchmark to study the flow field, mainly around the regions of separated flow. A sketch of the geometry is presented in Fig. 6.1. It consists of a flat front with rounded corners and a base slant angle in the rear end. The first experimental article about its flow behavior presented force and pressure measurements and a detailed wake survey [6].

This model configuration is selected because it can replicate the main features of flow past real vehicles without geometrical complexities, such as: rotating wheels, a complex chassis, lateral mirrors and the receptacle under the hood. Nevertheless the study of this bluff body provides a qualitative understanding of the relation between different aerodynamic characteristics: pressure distribution, drag and wake structure.

The first experimental tests were performed in subsonic wind tunnels at Braunschweig and Göttingen [6]. These open test section tunnels have a frontal surface of $3m$ by $3m$ and the length of the test-section is $5.8m$. Based on the geometric characteristics the blockage factor of the tunnel (defined as the fraction between the Ahmed body frontal area and tunnel one) is 1.245%. All the tests presented in [6] were performed at a wind speed of $60m/s$ and a Reynolds number of 4.29 million based on the longitude of the model. Figure 6.2 shows the variation of the drag coefficient (C_d) with the slant angle φ .

Even though the flow is unsteady the time-averaged measurements shows a defined flow macro structure. There are two variations of this macro structure and they are dependent on the angle φ . For angles between 5° and 25° Figure 6.3 displays the vortex system created at the rear end of the bluff body. The main features of the wake are:

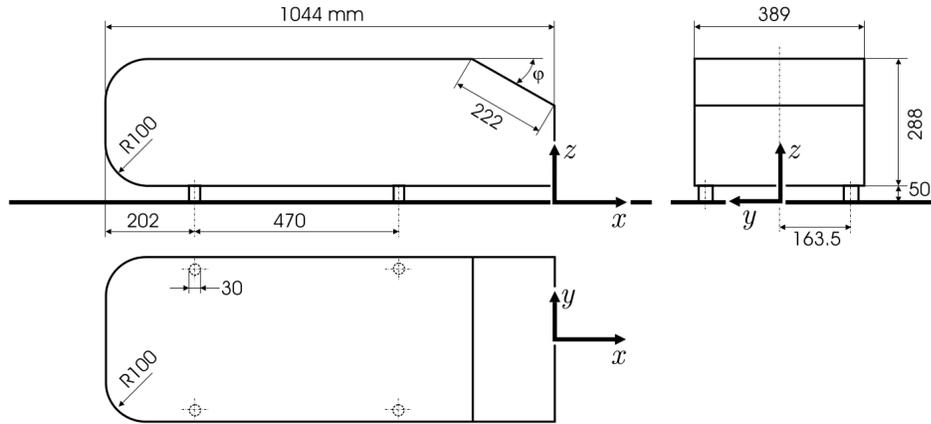


Figure 6.1: Ahmed Body geometry, dimensions are in *mm*. Figure from [5].

- An upper "A" and a lower "B" recirculatory flow region. This configuration is generated by two horseshoe vortices in the separation bubble "D".
- A roll up of vortices "C" at the slant corner edge.

For angles φ above 25° there is a large area of separated flow above the rear slant. This configuration is not studied in the present thesis.

In 2000, Lienhart et al. [73] presented the ‘Models for Vehicle Aerodynamics’ (MOVA) Project, a program established by several European consortium partners (TU Delft, University of Manchester, Universität Erlangen-Nürnberg, Electricite de France, AVL List and PSA Peugeot Citroen). The aim of this project was to develop, refine and validate turbulence models that accurately represent the vehicle aerodynamics. The Ahmed Body is one of the benchmarks selected in the MOVA Project and experimental results are presented in [73–75].

In recent years CFD simulations of the Ahmed Body have been common in the literature. Examples of this are the following articles [7,15,16,76–85]. A study of drag and flow pattern is conducted with the results collected from these articles. In addition, computing times, when available, are compared.

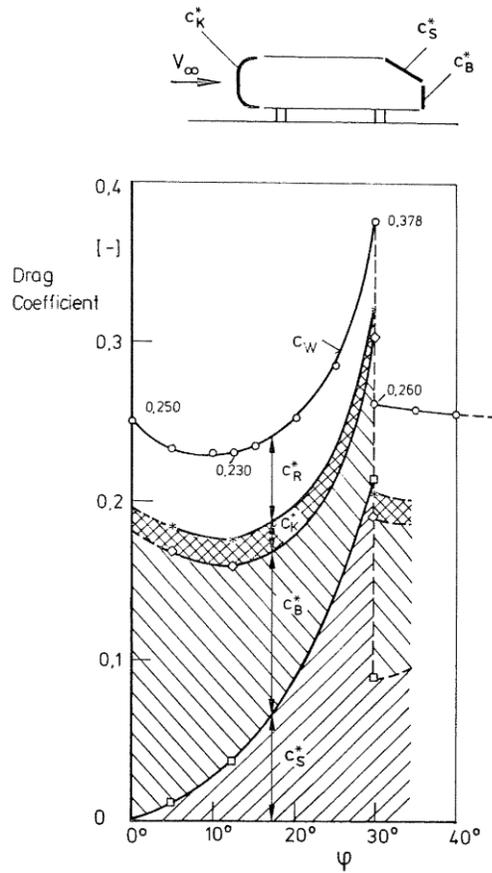


Figure 6.2: C_d curve for several φ angles. Here, c_W is the drag coefficient, c_R is the friction drag coefficient, c_B is the vertical base pressure drag coefficient, c_K is the forebody pressure drag coefficient and c_S is the slant surface pressure drag coefficient. Figure from [6].

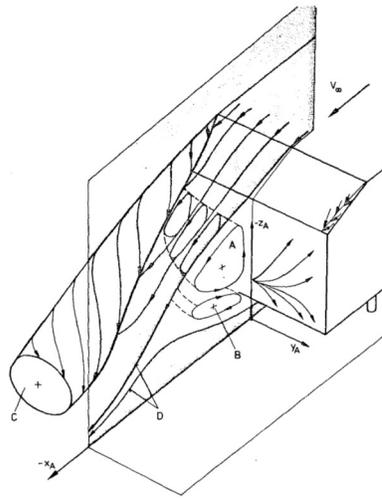


Figure 6.3: Flow macro structure at the rear slant of the Ahmed Body. Figure from [6].

6.1.1 Geometry and Test Set-up

The main dimensions of the Ahmed Body are length $l = 1044mm$, height $h = 288mm$, width $w = 389mm$ and it is placed $50mm$ above ground. The angle φ is set to 25 degrees in all the simulations. The flow initial conditions are $u = 30m/s$, $\rho = 1.225kg/m^3$, $\mu = 6.67 \cdot 10^{-5}kg/(m \cdot s)$, $c_{sound} = 300m/s$ and the Smagorinsky constant is set to $C_s = 0.13$. The system of cartesian grids is displayed in Fig. 6.4 and the number of domains is 3158. Two grids will be used for the simulations: a fine mesh with a total number of active points $n_{activ} = 29584137$ and a coarse mesh with $n_{activ} = 5849745$. The minimum mesh size is $2.61mm$ ($y^+ \approx 83$) in the fine mesh and $5.2mm$ ($y^+ \approx 165$) in the coarse mesh (Fig. 6.5). The size of the domain is $-4.8m \leq x \leq 8.6m$, $0m \leq y \leq 2m$ and $-2m \leq z \leq 2m$, with a blockage factor of 1.4%.

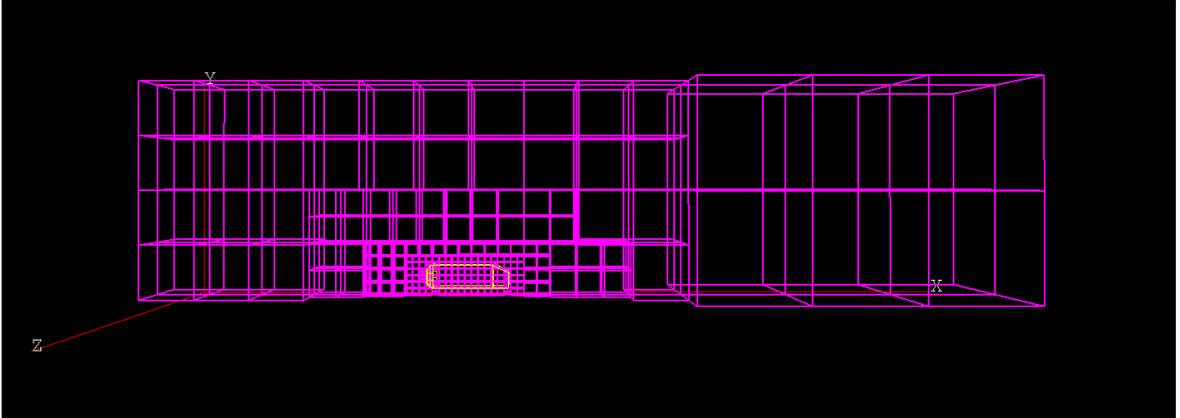


Figure 6.4: System of Cartesian Grids for the Ahmed Body.

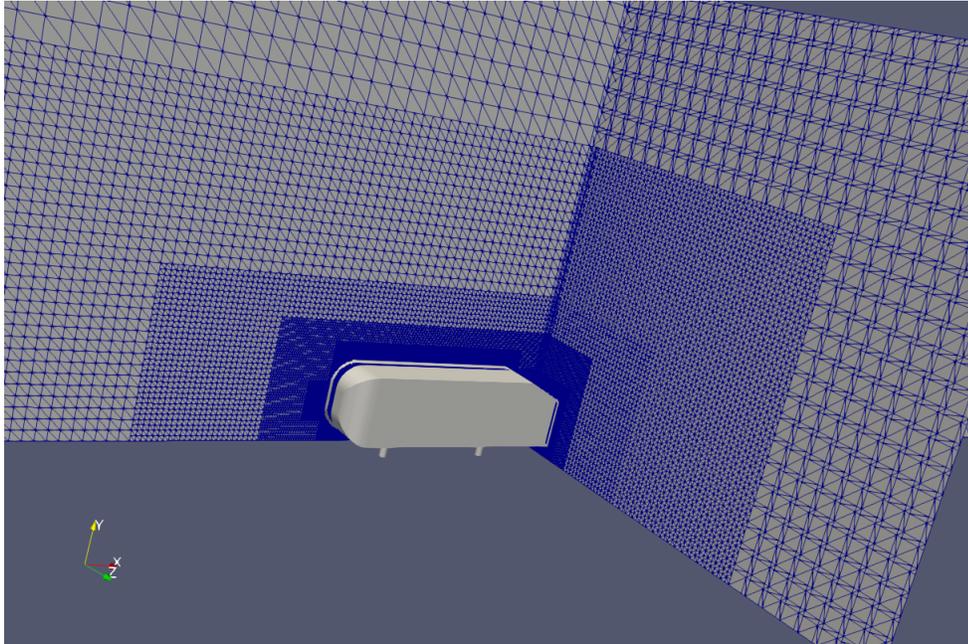


Figure 6.5: Coarse Mesh, Ahmed Body.

6.2 Results

In this section three main characteristics of the simulation results are detailed,

- the flow structures around the Ahmed Body are reported and compared with experimental data,
- the drag coefficient, and
- run times.

6.2.1 Flow Structures

The vortex structure in the rear of the Ahmed Body is shown in Figure 6.3. At the angle $\varphi = 25^\circ$ in the mean flow in the center plane, the reattachment of the flow over the slant and the recirculation areas behind the body are important characteristics reported in the literature. In Figure 6.6, two counter rotating vortices are observed, which are created by the flow evacuating the top and bottom edges of the rear vertical plane.

Results from the simulations are displayed in Fig. 6.7. In this figure the line integral convolution (LIC) is visualized. This is a vector field technique that convolves noise with a vector field, producing patterns that follow vector field tangents [86]. The expected vortices are presented in all the cases, but better defined in the fine meshes compared with the coarse ones. In the coarse mesh, there is a large separation region above the rear slant and no re attachment of the flow. This is a phenomenon also observed in [7], where DES - $k-\omega$ SST, LES-NWM (LES with Smagorinsky model and wall function) and LES-NWR (wall resolving LES with dynamic Smagorinsky model) simulations predicted fully separated flow or attached flow over the entire slant. [83] shows that RANS models over predict the separate region and DDES (Delayed DES) methods failed to predict the correct re attachment point.

Meanwhile, results in the fine mesh are similar to the experimental data in [73–75], where all the methods predict a re attached flow and two counter-rotating vortices clearly defined. Location of the vortices and comparison with experimental data are found in Table 6.1. All the schemes in the fine mesh accurately capture the lower vortex, however the solution of the upper vortex has greater discrepancies between the method used. This issue is also discussed in [78] were simulations are performed with LES schemes.

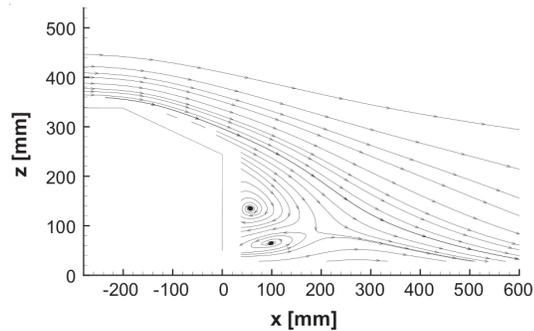
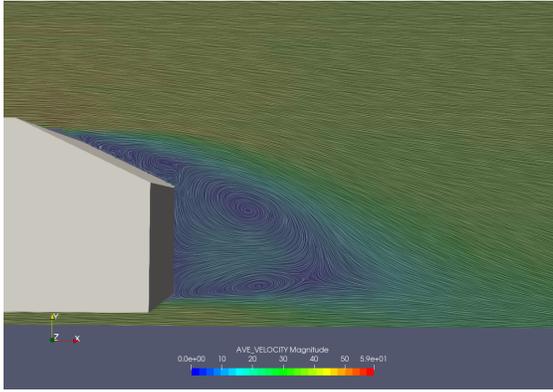
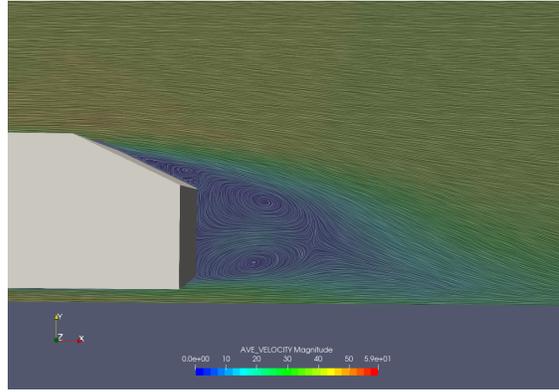


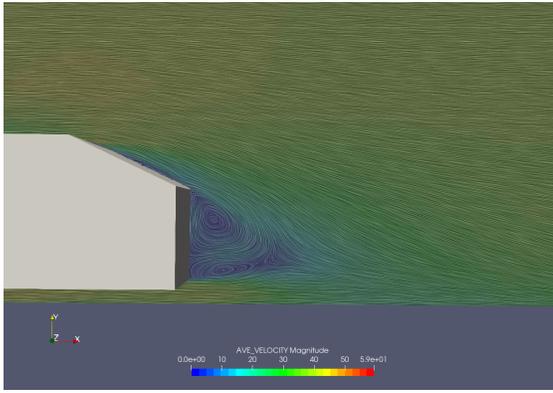
Figure 6.6: Experimental Streamlines Behind the Ahmed Body (Source [7]).



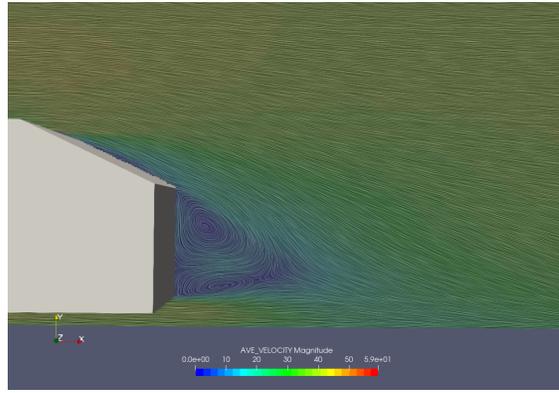
(a) Coarse Mesh, 4th Order.



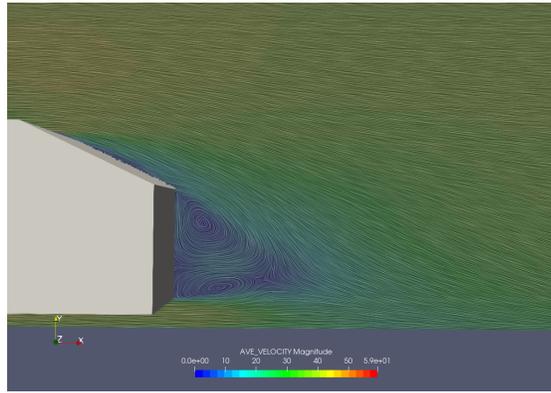
(b) Coarse Mesh, 8th Order.



(c) Fine Mesh, 2nd Order



(d) Fine Mesh, 4th Order



(e) Fine Mesh, 4th Order, 2Stage RK

Figure 6.7: LIC Surfaces of the Flow Behind the Body

Another important feature are a pair of counter rotating vortices which are located above and behind the lateral of the rear slant. These vortices are characteristic of the mean flow

Table 6.1: Vortex Position Behind the Ahmed Body, Fine Mesh, Position in *mm*.

Vortex	Method	x/h	y/h
Upper Vortex	Experiment	0.2	0.6
	2 nd Order	0.17	0.64
	4 th Order	0.18	0.61
	2Step RK, 4 th Order	0.18	0.64
Lower Vortex	Experiment	0.35	0.25
	2 nd Order	0.33	0.25
	4 th Order	0.34	0.25
	2Step RK, 4 th Order	0.34	0.25

and their shape depends on how well the flow and the turbulent fluctuations are defined. The velocity contours at three different positions behind the body are compared in Figures 6.8, 6.9 and 6.10. The main flow structures are represented at the four studied positions and it can be seen how higher order methods conserve the vortex resolution better than the lower order algorithms. At $x/h = 1.74$ and $x/h = 3.08$ with the 8th order method vortices are clearly visible and better defined than with the second or fourth order approximation. The differences between a fourth order method with linear interpolation and with high order interpolation are indistinctly close and far from the body.

To conclude, in order to display the turbulent flow structure behind the body, the Q criterion is applied. This is the second invariant of the velocity gradient tensor, and is represented by

$$Q = \frac{1}{2}(\|\Omega\|^2 - \|S\|^2). \quad (6.1)$$

S is the strain tensor, Ω the vorticity and Q represents the areas dominated by the vorticity if $Q > 0$ or by the strain tensor if $Q < 0$. The Q criterion and the instantaneous velocity field at $t = 0.4$ are shown in Fig. 6.12 and 6.13.

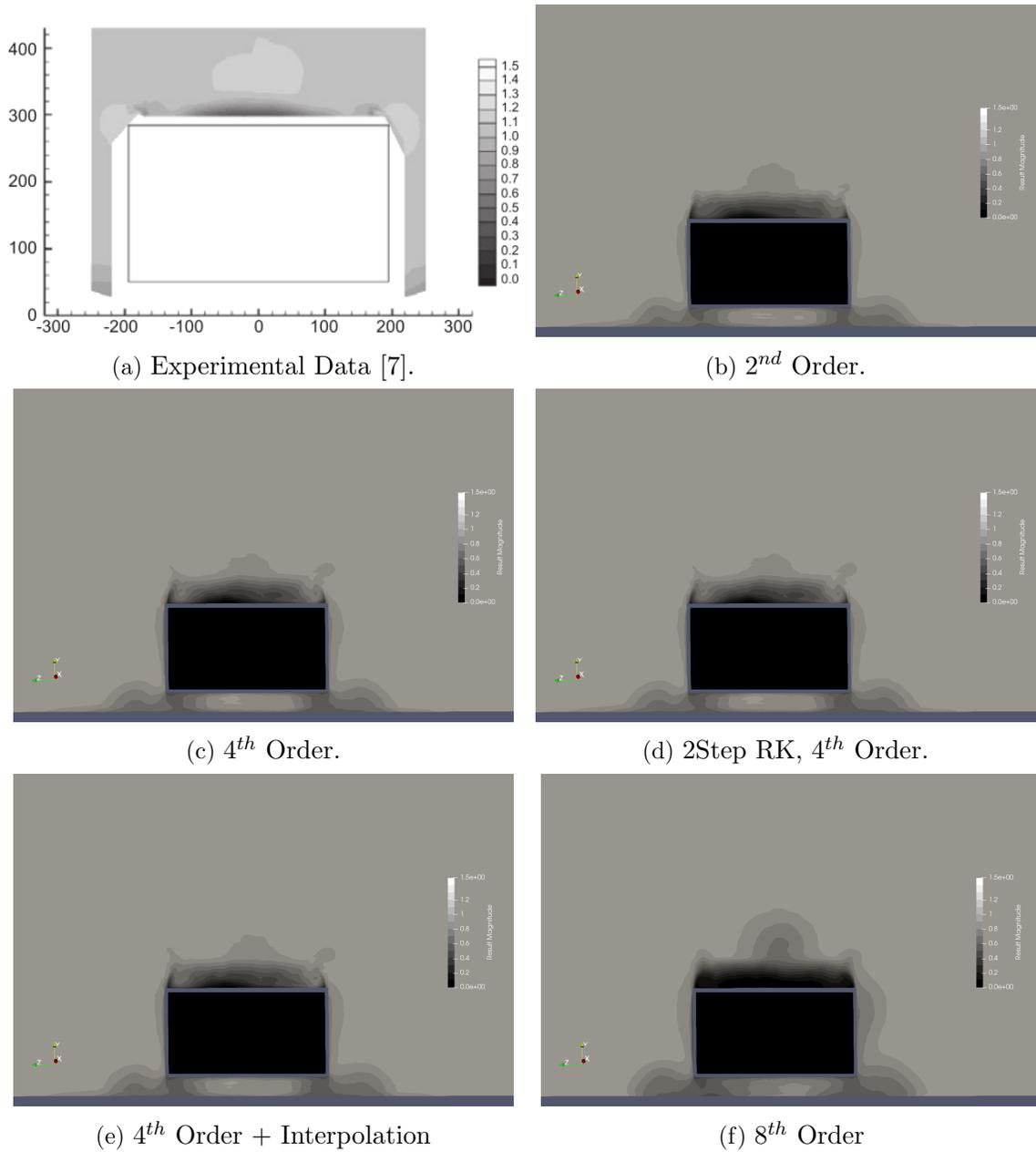


Figure 6.8: Normalized Mean Velocity at $x/h = -0.135$. Experimental Data at $x/h = -0.13$.

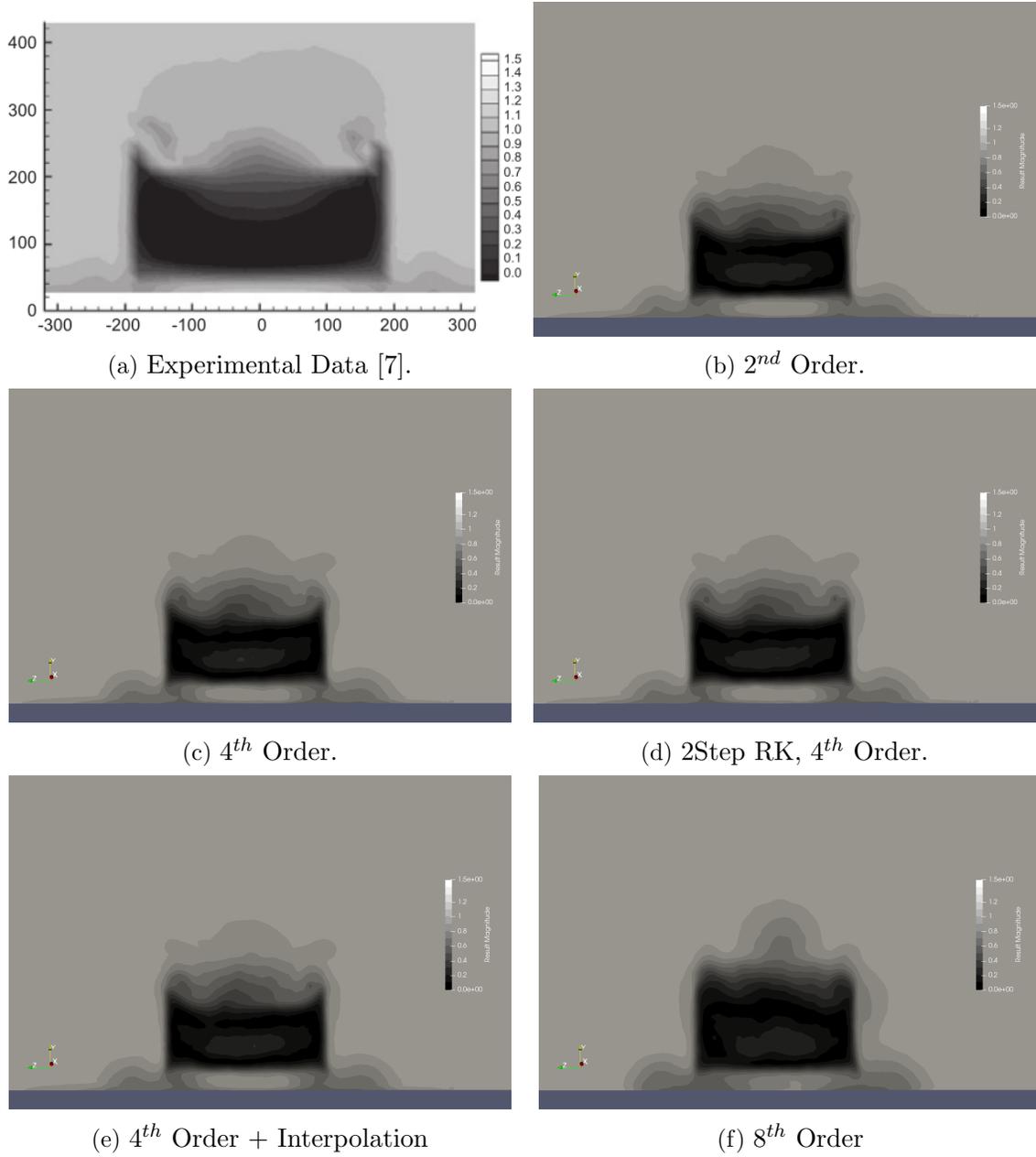


Figure 6.9: Normalized Mean Velocity at $x/h = 0.278$. Experimental Data at $x/h = 0.27$.

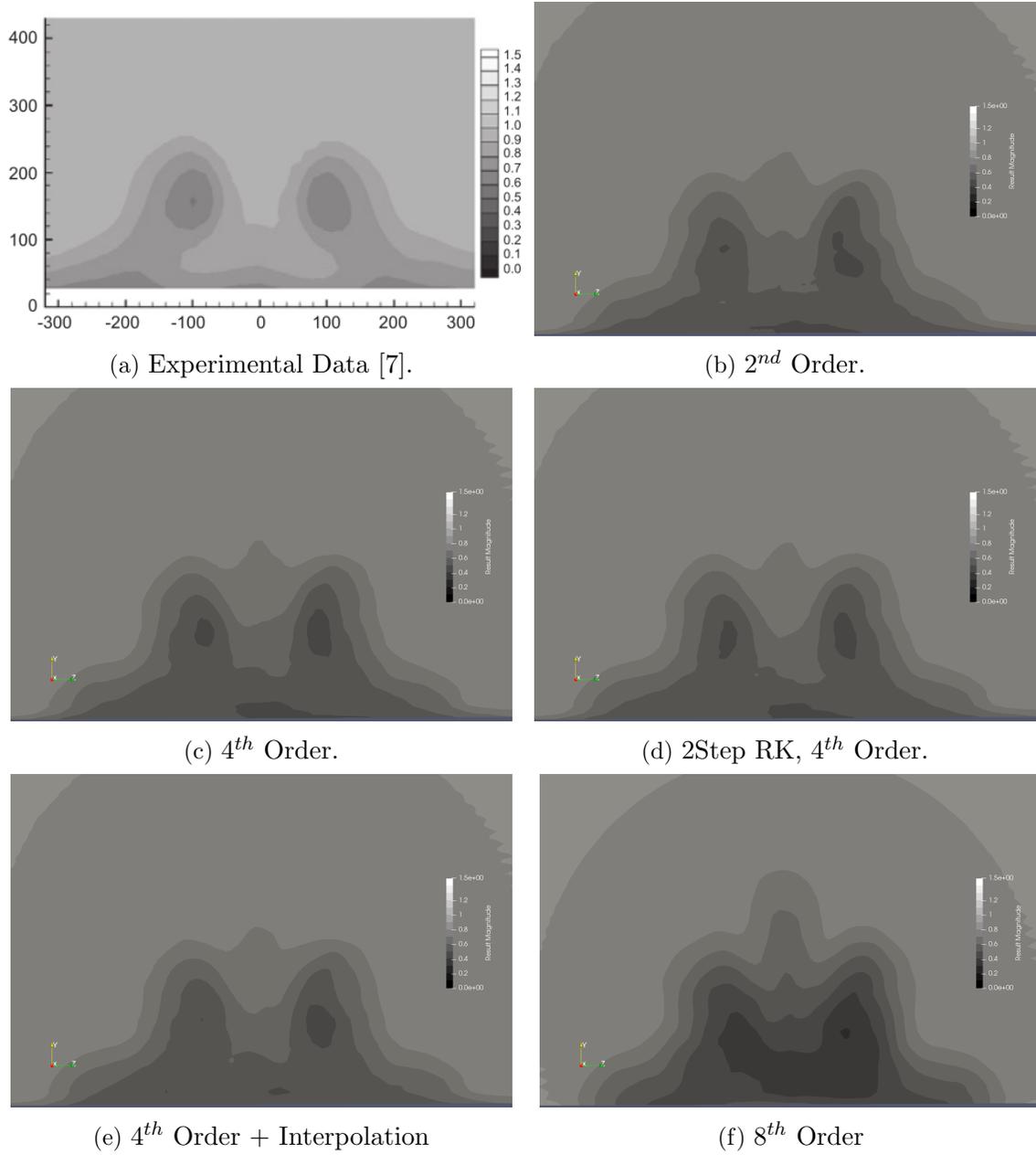


Figure 6.10: Normalized Mean Velocity at $x/h = 1.74$. Experimental Data at $x/h = 1.74$.



(a) 2nd Order.



(b) 4th Order.



(c) 2Step RK, 4th Order.

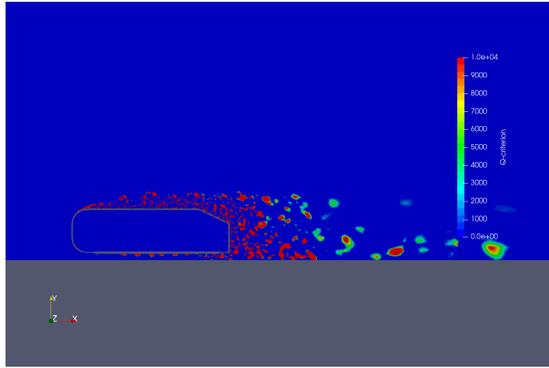


(d) 4th Order + Interpolation

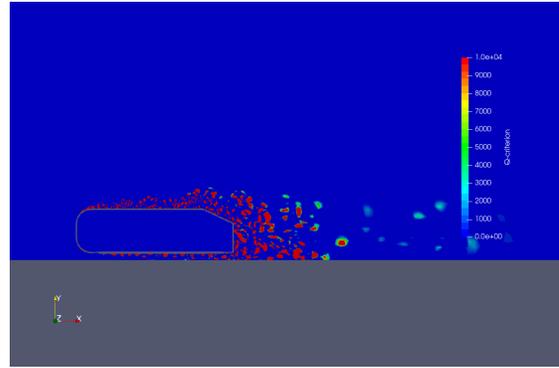


(e) 8th Order

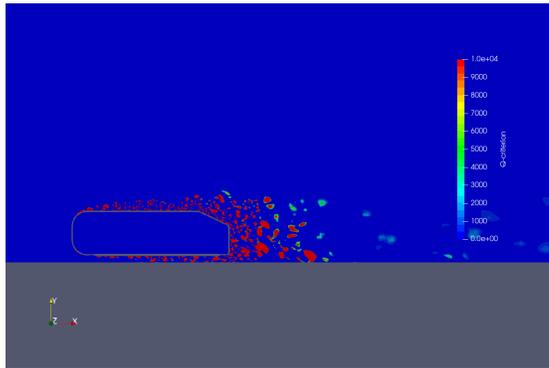
Figure 6.11: Normalized Mean Velocity at $x/h = 3.08$.



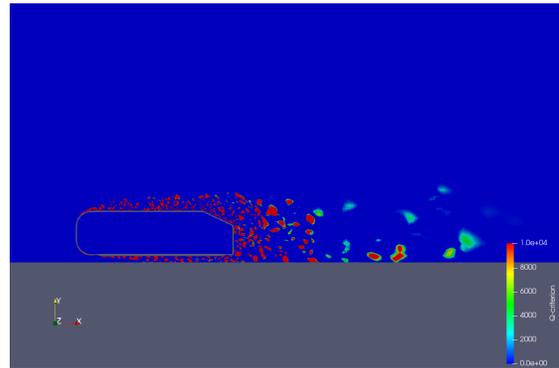
(a) 2nd Order.



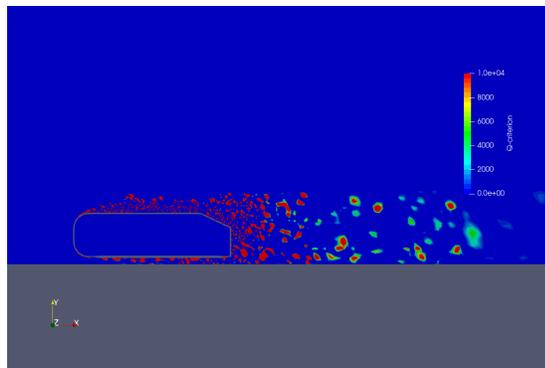
(b) 4th Order.



(c) 2Step RK, 4th Order.

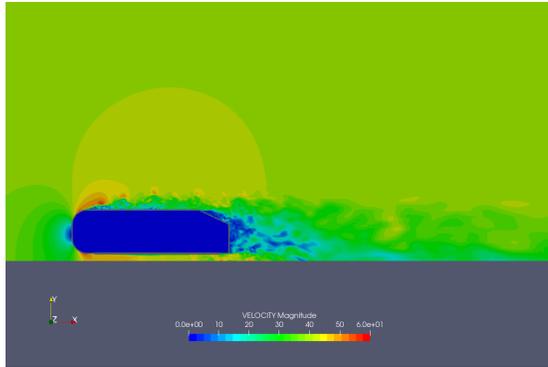


(d) 4th Order + Interpolation

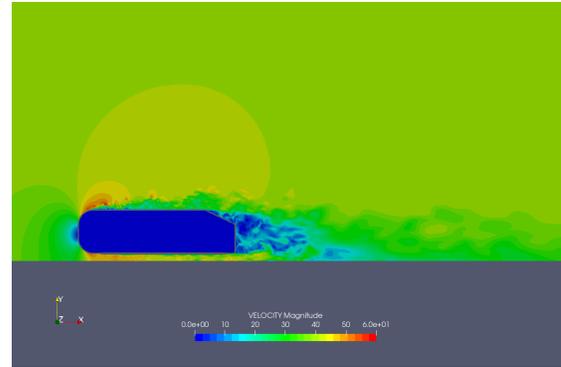


(e) 8th Order

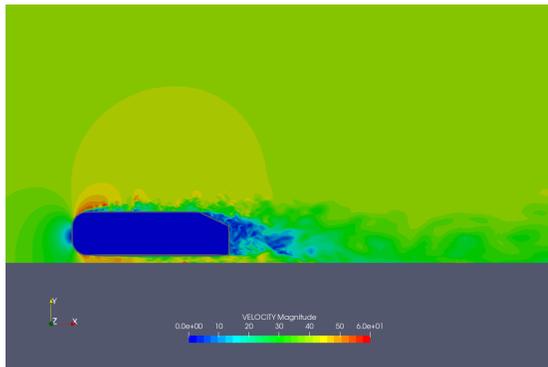
Figure 6.12: Q-Criterion at $t = 0.4$.



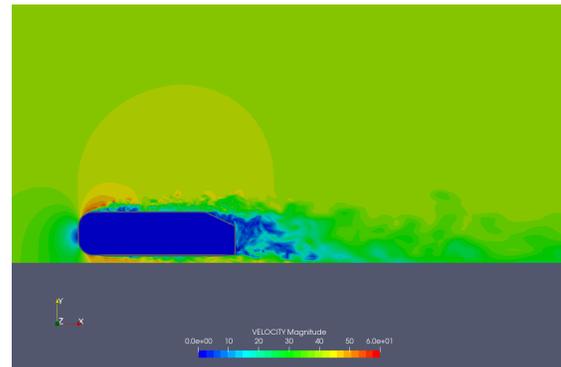
(a) 2nd Order.



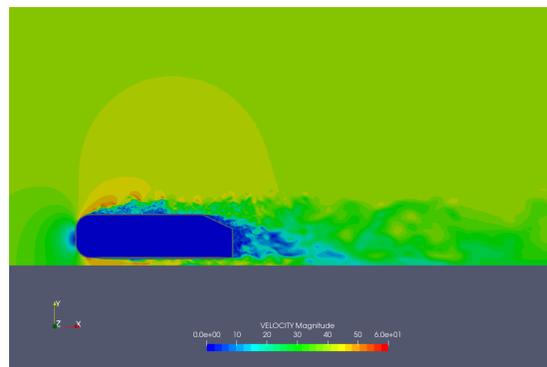
(b) 4th Order.



(c) 2Step RK, 4th Order.



(d) 4th Order + Interpolation



(e) 8th Order.

Figure 6.13: Instantaneous Velocity Field at $t = 0.4$.

6.2.2 Drag Coefficient and Run Time

The drag coefficient reported in the first experimental study of the Ahmed body is introduced in 6.2. The approximate values of the drag coefficients due to different components of the geometry at $\varphi = 25^\circ$ are:

- The slant surface $c_S = 0.14$.
- The vertical base pressure drag coefficient $c_B = 0.073$.
- The forebody pressure drag coefficient $c_K = 0.02$.
- The friction drag coefficient $c_R = 0.053$.

The total experimental drag coefficient of the body is $C_d = 0.286$. The results obtained in the simulations with FDFLO are represented in Table 6.2. As is noted in the case of the sphere, high order schemes do not estimate the drag coefficient correctly, requiring a more detailed study of the stencils and calculation of the forces over the geometry.

Table 6.2: Drag Coefficients of the Ahmed Body. ‘4th Order + Interpolation’ Corresponds to a Discretization of Fourth Order With High Order Interpolation Between Grids Activated.

Mesh	Method	C_d
	Experiment	0.286
Coarse Mesh	4 th Order	0.330
Coarse Mesh	8 th Order	0.389
Fine Mesh	2 nd Order	0.256
Fine Mesh	4 th Order	0.287
Fine Mesh	4 th Order + Interpolation	0.333
Fine Mesh	2Step RK, 4 th Order	0.291
Fine Mesh	8 th Order	0.35

All the simulations were run from $t = 0s$ to $t = 0.4s$ with only the last 0.1s used for the calculation of the drag coefficient. The variation of the C_d during the time used for

statistics is displayed in Fig. 6.14 for the 4th Order scheme in the fine mesh. It is important to comment that the solution has not converged but limited hardware access hindered simulations to run for longer periods of time.

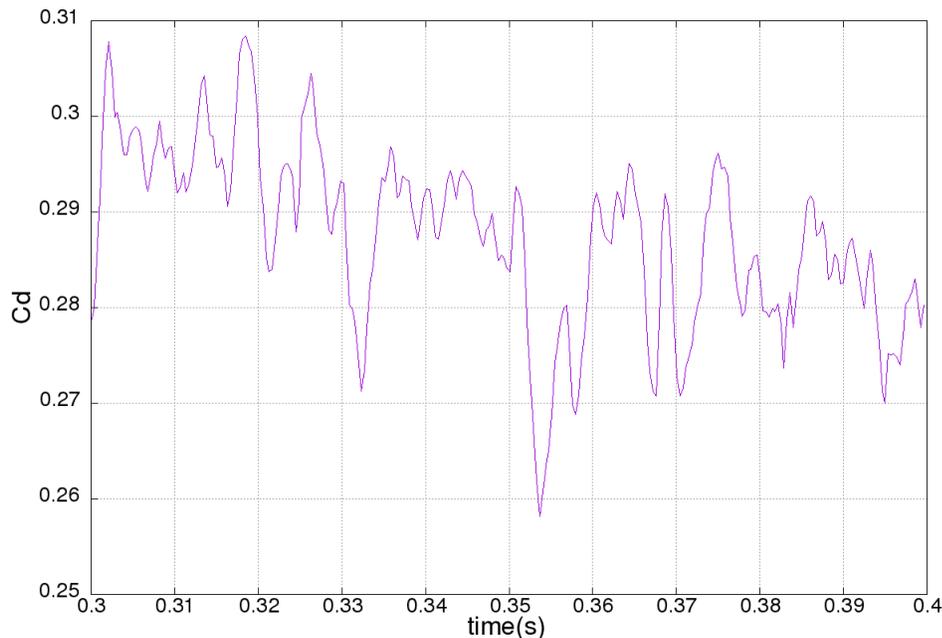


Figure 6.14: Drag Coefficient.

The number of cores used to perform the simulation were in all the cases $n_{core} = 320$. Wall time to obtain the solutions and the corresponding Courant number used are presented in Table 6.3.

As comparisons of runtimes with those presented in other articles are difficult, a dimensionless time is defined as

$$\bar{t} = (t_f - t_0) \cdot U_\infty / L, \quad (6.2)$$

with t_0 the initial time, t_f the final time and L the longitude of the car. Then, the following relations are used to estimate the total core time per point per time step and per dimensionless total time,

Table 6.3: Run Times for the Ahmed Body Simulations.

Mesh	Method	Courant	Turbulence	Time(s)
Coarse Mesh	4 th Order	0.2	no	930
Coarse Mesh	8 th Order	0.2	no	4398
Fine Mesh	2 nd Order	0.2	no	7726
Fine Mesh	4 th Order	0.2	yes	12540
Fine Mesh	2Step RK, 4 th Order	0.5	no	7512
Fine Mesh	8 th Order	0.3	no	19620

$$\bar{\Delta}t = \frac{t_{wall}n_{core}}{1 \cdot n_{point}}, \quad (6.3)$$

$$\bar{T} = \frac{t_{wall}n_{core}}{\bar{t} \cdot n_{point}}. \quad (6.4)$$

Here, t_{wall} is the wall time in hours, n_{core} the number of cores and n_{point} the number of points in the mesh. These metrics are listed in Table 6.4 contrasted with information provided in other publications. 2Step Runge Kutta method achieves a speed up of 1.7 times higher compared with regular fourth order low storage Runge Kutta mainly due to a greater Courant number and lower communication overhead. Furthermore, it has a similar run time as a second order scheme but with the advantage of using a higher order of fine discretization.

Table 6.4: Run Times for the Ahmed Body. FV2 = second order finite volume method, PS = pseudo spectral method, BDF2/3 = second/third order backward Euler, CN = implicit Crank-Nicolson, RK2/3 = second/fourth order Runge Kutta.

Code	Numerical Method	Npoin (10^6)	Δt	\bar{T}	C_d
FDFO	4 th Order	5.8	$2.8 \cdot 10^{-5}$	$2.8 \cdot 10^{-6}$	0.330
FDFO	8 th Order	5.8	$6.7 \cdot 10^{-5}$	$8.7 \cdot 10^{-6}$	0.389
FDFO	2 nd Order	29.6	$2.3 \cdot 10^{-5}$	$3.0 \cdot 10^{-6}$	0.256
FDFO	4 th Order	29.6	$3.8 \cdot 10^{-5}$	$5.0 \cdot 10^{-6}$	0.287
FDFO	2Step RK, 4 th Order	29.6	$2.3 \cdot 10^{-5}$	$3.0 \cdot 10^{-6}$	0.291
ISIS-CFD [7]	FV2-BDF3	23	$8.7 \cdot 10^{-4}$	$2.5 \cdot 10^{-5}$	0.343
FASTEST [7]	FV2-CN	40	$1.3 \cdot 10^{-4}$	$1.2 \cdot 10^{-4}$	0.346
LESOC2 [7]	FV2-RK2	18.5	$1.6 \cdot 10^{-3}$	$2.1 \cdot 10^{-4}$	0.317
SVVLES [7]	PS-BDF2/RK4	21.3	$2.3 \cdot 10^{-5}$	$2.1 \cdot 10^{-6}$	0.431

Chapter 7: Conclusions

7.1 Conclusions

Computational Fluid Dynamics (CFD) has been a successful tool for industry applications during the last decades. However, at present accurate solutions involving vortex propagation, separated and turbulent flows are still associated with high computing costs. LES simulations of complex geometries, such as an automobile, at high Reynolds number require several days to obtain a solution with statistically relevant information.

FDFLO is a Finite Difference code, developed for industrial applications at the CFD Center at George Mason University. FDFLO is able to perform LES simulation with the aim of overnight turnarounds. This code is optimized for speed and solves the weakly compressible Navier Stokes equations. The following strategies were implemented during the development of FDFLO and are part of this thesis.

Simple interpolation schemes based on post-processing raw bi-/trilinear halo-point transfer for nested cartesian grid systems have been developed. This allows to maximize modularity while preserving locality. Results obtained for model problems indicate that the simple interpolation schemes improve the convergence rates and thus preserve the overall accuracy of finite difference codes with varying grid sizes.

High order schemes of second, fourth and eighth order, were applied to simple and complex configurations, showing the expected overall performance. Turbulent flows at low Mach number were successfully resolved using the Taylor Green Vortex as an example. The resulting turbulent energy cascade and the main flow characteristics were in good agreement with the bibliography.

The Ahmed body, a bluff body used for the automotive industry as benchmark of CFD codes, was studied and compared with experimental results. Regions of separate flow,

vortex and aerodynamic characteristics were in accordance with published experimental results. Despite using a coarser mesh size than other LES simulations the physics of the flow was correctly predicted in less simulation time. However, better immersed and embedded techniques have to be developed for discretization schemes of order higher than two. Furthermore, additional validations should be performed for complex geometries.

Future work in order to achieve the main goal of a FD solver with overnight turn around times for industrial applications includes:

- Further extensions of Runge-Kutta type schemes to higher than fourth order approximations with reduced stage number.
- Development of immersed or embedded boundary methods compatible with high order discretization.
- Additional code optimization. There is a 20% run time difference between simulations with and without an activated turbulence model. This is an example of a section of the code that can still be optimized.
- Validation of complex geometries. DrivAer is an openly available car geometry developed by BMW, Audi AG and the Technische Universität München (TUM). This is an example of a realistic geometry that can be evaluated with FDFLO.
- Development of robust boundary conditions.
- Assessment of the code performance with an arbitrary high number of cores ($n_{core} \gg 1000$).

LES simulations that run overnight have been envisioned by the industry and the CFD community in the last decades. It is expected that with further code optimization and hardware developments FDFLO may achieve the aim of overnight LES runs in the near future.

Appendix A: Roe Solver Matrices

A.1 Matrix set up for B term

If B is given by,

$$B = PB_0 = \begin{pmatrix} 0 & 0 & \beta^2 c^2 \rho & 0 \\ 0 & v & u & 0 \\ \frac{1}{\rho} & 0 & 2v & 0 \\ 0 & 0 & w & v \end{pmatrix},$$

with corresponding eigenvalues,

$$\lambda_1 = v, \tag{A.1}$$

$$\lambda_2 = v, \tag{A.2}$$

$$\lambda_3 = v - \sqrt{\beta^2 c^2 + v^2} = v - l, \tag{A.3}$$

$$\lambda_4 = v + \sqrt{\beta^2 c^2 + v^2} = v + l. \tag{A.4}$$

The preconditioning right and left eigenvectors are defined by,

$$R = \begin{pmatrix} 0 & 0 & \frac{\rho l(v+l)}{w} & \frac{\rho l(-v+l)}{w} \\ 0 & 1 & \frac{u}{w} & \frac{u}{w} \\ 0 & 0 & \frac{-l}{w} & \frac{l}{w} \\ 1 & 0 & 1 & 1 \end{pmatrix},$$

$$L = R^{-1} = \begin{pmatrix} \frac{-w}{l^2\rho} & 0 & \frac{-vw}{l^2} & 1 \\ \frac{-u}{l^2\rho} & 1 & \frac{-uv}{l^2} & 0 \\ \frac{w}{2l^2\rho} & 0 & \frac{-w(l-v)}{2l^2} & 0 \\ \frac{w}{2l^2\rho} & 0 & \frac{w(l+v)}{2l^2} & 0 \end{pmatrix}.$$

Equation Eq. (2.27) for the B term is,

$$\begin{aligned} \sum_{i=1}^4 |\lambda_i| \Delta w_i P^{-1} \vec{r}_i = & \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \lambda_1 \Delta w_1 + \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \lambda_2 \Delta w_2 + \\ & + \begin{pmatrix} \frac{\rho l(l+v)}{\beta^2 w} \\ \frac{u}{w} \\ \frac{-l}{w} \\ 1 \end{pmatrix} \lambda_3 \Delta w_3 + \begin{pmatrix} \frac{\rho l(l-v)}{\beta^2 w} \\ \frac{u}{w} \\ \frac{l}{w} \\ 1 \end{pmatrix} \lambda_4 \Delta w_4. \end{aligned} \tag{A.5}$$

Finally the wave strengths are,

$$\Delta w_1 = -\frac{w\Delta p}{l^2\rho} - \frac{vw\Delta v}{l^2} + \Delta w, \tag{A.6}$$

$$\Delta w_2 = -\frac{u\Delta p}{l^2\rho} - \frac{uv\Delta v}{l^2} + \Delta u, \tag{A.7}$$

$$\Delta w_3 = \frac{w\Delta p}{2l^2\rho} - \frac{w(l-v)\Delta v}{2l^2}, \quad (\text{A.8})$$

$$\Delta w_4 = \frac{w\Delta p}{2l^2\rho} + \frac{w(l+v)\Delta v}{2l^2}. \quad (\text{A.9})$$

A.2 Matrix set up for C term

The same procedure as in the term A and B is detailed next.

$$C = PC_0 = \begin{pmatrix} 0 & 0 & 0 & \beta^2 c^2 \rho \\ 0 & w & 0 & u \\ 0 & 0 & w & v \\ \frac{1}{\rho} & 0 & 0 & 2w \end{pmatrix},$$

with eigenvalues,

$$\lambda_1 = w, \quad (\text{A.10})$$

$$\lambda_2 = w, \quad (\text{A.11})$$

$$\lambda_3 = w - \sqrt{\beta^2 c^2 + w^2} = w - l, \quad (\text{A.12})$$

$$\lambda_4 = w + \sqrt{\beta^2 c^2 + w^2} = w + l. \quad (\text{A.13})$$

The right and left eigenvectors are

$$R = \begin{pmatrix} 0 & 0 & -(w+l)\rho & (-w+l)\rho \\ 0 & 1 & \frac{-u}{l} & \frac{u}{l} \\ 1 & 0 & \frac{-v}{l} & \frac{v}{l} \\ 0 & 0 & 1 & 1 \end{pmatrix},$$

$$L = R^{-1} = \begin{pmatrix} \frac{-v}{l^2\rho} & 0 & 1 & \frac{-vw}{l^2} \\ \frac{-u}{l^2\rho} & 1 & 0 & \frac{-uw}{l^2} \\ \frac{-1}{2l\rho} & 0 & 0 & \frac{(l-w)}{2l} \\ \frac{1}{2l\rho} & 0 & 0 & \frac{(l+w)}{2l} & 0 \end{pmatrix}.$$

so, the final result is,

$$\begin{aligned}
\sum_{i=1}^4 |\lambda_i| \Delta w_i P^{-1} \vec{r}_i = & \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \lambda_1 \Delta w_1 + \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \lambda_2 \Delta w_2 + \\
& + \begin{pmatrix} \frac{-\rho(l+w)}{\beta^2} \\ \frac{-u}{l} \\ \frac{-v}{l} \\ 1 \end{pmatrix} \lambda_3 \Delta w_3 + \begin{pmatrix} \frac{\rho(l-w)}{\beta^2} \\ \frac{u}{l} \\ \frac{v}{l} \\ 1 \end{pmatrix} \lambda_4 \Delta w_4.
\end{aligned} \tag{A.14}$$

Finally the wave strengths are,

$$\Delta w_1 = -\frac{v \Delta p}{l^2 \rho} - \frac{vw \Delta w}{l^2} + \Delta v, \tag{A.15}$$

$$\Delta w_2 = -\frac{u \Delta p}{l^2 \rho} - \frac{uw \Delta w}{l^2} + \Delta u, \tag{A.16}$$

$$\Delta w_3 = -\frac{\Delta p}{2l\rho} + \frac{(l-w)\Delta w}{2l}, \tag{A.17}$$

$$\Delta w_4 = \frac{\Delta p}{2l\rho} + \frac{(l+w)\Delta w}{2l}. \tag{A.18}$$

Appendix B: Interpolation Weights

The procedure used to derive the 1D interpolation weights is presented in this section. Using similar arguments the method can be extended to 2D and 3D. Given n points x_i , $i = 1, n$ and n values u^i at these points, the Lagrange Polynomial is given by:

$$u(x) = \sum_{i=1}^n u^i L_i(x). \quad (\text{B.1})$$

where the Lagrange polynomial associated with point x_i is:

$$L_i(x) = \frac{N_i(x)}{D_i} = \frac{\prod_{j=1, j \neq i} (x - x_j)}{\prod_{j=1, j \neq i} (x_i - x_j)}. \quad (\text{B.2})$$

This may also be written as:

$$L_i(x) = \sum_{k=0}^{n-1} a_{ik} x^k. \quad (\text{B.3})$$

where a_{ik} are the product entries of $\prod_{j=1, j \neq i} (x - x_j)$ ordered in ascending powers of x , divided by D_i .

Bibliography

Bibliography

- [1] “Energy Cascade,” https://en.wikipedia.org/wiki/Energy_cascade.
- [2] H. Schlichting and K. Gersten, *Fundamentals of Boundary-Layer Theory*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2017, 10.1007/978-3-662-52919-5_2.
- [3] M. Geier, A. Pasquali, and M. Schönherr, “Parametrization of the cumulant lattice boltzmann method for fourth order accurate diffusion part ii: Application to flow around a sphere at drag crisis,” *Journal of Computational Physics*, July 2017, 10.1016/j.jcp.2017.07.004.
- [4] C. C. de Wiart, K. Hillewaert, M. Duponcheel, and G. Winckelmans, “Assessment of a Discontinuous Galerkin Method for the Simulation of Vortical Flows at High Reynolds Number,” *International Journal for Numerical Methods in Fluids*, vol. 74, no. 7, pp. 469–493, 2014, 10.1002/fld.3859. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.3859>
- [5] C. Hinterberger, M. García-Villalba, and W. Rodi, “Large eddy simulation of flow around the Ahmed Body,” R. McCallen, F. Browand, and J. Ross, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 77–87, the Aerodynamics of Heavy Vehicles: Trucks, Buses, and Trains.
- [6] S. Ahmed, G. Ramm, and G. Faltin, “Some Salient Features Of The Time-Averaged Ground Vehicle Wake,” in *SAE Technical Paper*. SAE International, February 1984, <https://doi.org/10.4271/840300>.
- [7] E. Serre, M. Minguéz, R. Pasquetti, E. Guilmineau, G. Deng, M. Kornhaas, M. Schäfer, J. Fröhlich, C. Hinterberger, and W. Rodi, “On Simulating the Turbulent Flow Around the Ahmed body: A French–German Collaborative Evaluation of LES and DES,” *Computers & Fluids*, vol. 78, pp. 10–23, 2013.
- [8] J. Slotnick, A. Khodadoust, J. Alonso, D. Darmofal, W. Gropp, E. Lurie, and D. Mavriplis, “CFD Vision 2030 Study: A Path to Revolutionary Computational Aero-science,” *NASA/CR-2014-218178, NF1676L-18332*, 2014.
- [9] F. D. Witherden and A. Jameson, “Future Directions in Computational Fluid Dynamics,” 2017, presented at AIAA-AVIATION 2017 Special Session.
- [10] B. Dawes, L. Yi, K. Liu, and C. Vasco, “Progress Towards an Industry-Friendly LES Capability,” CFD-19. Towards Industrial LES and DNS for Aeronautics, 2017, presented at AIAA-AVIATION 2017 Special Session.

- [11] R. Löhner, “Towards overcoming the les crisis,” *International Journal of Computational Fluid Dynamics*, vol. 33, no. 3, pp. 87–97, 2019.
- [12] T. Larsson, C. Bienz, T. Sato, and B. Ullbrand, “In Front of the Grid-CFD at SAUBER PETRONAS F1 Leading the Aerodynamic Development,” June 2003.
- [13] T. Larsson, T. Sato, and B. Ullbrand, “Supercomputing in F1 – Unlocking the Power of CFD,” June 2005.
- [14] technavio, “Global CFD Market in Automotive Industry 2017-2021,” December 2016, <https://www.technavio.com/report/global-product-lifecycle-management-global-cfd-market-automotive-industry-2017-2021>.
- [15] N. Ashton, A. West, S. Lardeau, and A. Revell, “Assessment of RANS and DES Methods for Realistic Automotive Models,” *Computers & Fluids*, vol. 128, pp. 1–15, 2016.
- [16] E. Guilmineau, G. Deng, A. Leroyer, P. Queutey, M. Visonneau, and J. Wackers, “Assessment of RANS and DES Methods for the Ahmed Body,” in *ECCOMAS Congress, June*, 2016, pp. 5–10.
- [17] H. Aono, A. Gupta, D. Qi, and W. Shyy, “The Lattice Boltzmann Method for Flapping Wing Aerodynamics,” *AIAA-2010-4867*, 2010.
- [18] B. Duncan, A. Fischer, and S. Kandasamy, “Validation of Lattice-Boltzmann Aerodynamics Simulation for Vehicle Lift Prediction.” ASME, August 2010, pp. 2705–2716, <http://dx.doi.org/10.1115/FEDSM-ICNMM2010-30891> (ASME 2010 3rd Joint US-European Fluids Engineering Summer Meeting).
- [19] D. Lockard, L. Luo, and B. Singer, “Evaluation of the Lattice-Boltzmann Equation Solver PowerFLOW for Aerodynamic Applications,” *NASA/CR-2000-210550 ICASE Report No. 2000-40 (also NASA/CR-2000-210550)*, 2000.
- [20] A. Pasquali, M. Schönherr, M. Geier, and M. Krafczyk, “Simulation of external aerodynamics of the DrivAer model with the LBM on GPGPUs,” *Parallel Computing: On the Road to Exascale*, vol. 27, pp. 391–400, 2016.
- [21] H. Xiaoyi, G. Doolen, and T. Clark, “Comparison of the lattice boltzmann method and the artificial compressibility method for navierstokes equations,” *J. Comp. Phys.*, vol. 179, pp. 439–451, 2002.
- [22] M. Junk, “A Finite Difference Interpretation of the Lattice Boltzmann Method,” *Num. Meth. Part. Diff. Equations*, vol. 17, pp. 383–402, 2001.
- [23] S. Geller, M. Krafczyk, J. Tölke, S. Turek, and J. Hron, “Benchmark Computations Based on Lattice-Boltzmann, Finite Element and Finite Volume Methods for Laminar Flows,” *Computers & Fluids*, vol. 35, pp. 888–897, 2006.
- [24] S. Marie, D. Ricot, and P. Sagaut, “Comparison Between Lattice Boltzmann Method and NavierStokes High Order Schemes for Computational Aeroacoustics,” *J. Comp. Phys.*, vol. 228, pp. 1056–1070, 2009.

- [25] J. Smagorinsky, “General Circulation Experiments With the Primitive Equations: I. The Basic Experiment,” *Monthly Weather Review*, vol. 91, no. 3, pp. 99–164, 1963.
- [26] K. Nakahashi, “Building-Cube Method for Flow Problems with Broadband Characteristic Length,” in *Computational Fluid Dynamics 2002*. Springer, 2003, pp. 77–81.
- [27] J. Sitaraman, A. Katz, B. Jayaraman, A. M. Wissink, and V. Sankaran, “Evaluation of a Multi-Solver Paradigm for CFD Using Overset Unstructured and Structured Adaptive Cartesian Grids,” *AIAA-2008-0660*, 2008.
- [28] R. Löhner, A. Corrigan, K. R. Wichmann, and W. Wall, “On the Achievable Speeds of Finite Difference Solvers on CPUs and GPUs,” *AIAA-2013-2852*, 2013.
- [29] —, “Comparison of Lattice-Boltzmann and Finite Difference Solvers,” *AIAA-2014-1439*, 2014.
- [30] J. Sitaraman, V. Lakhminarayan, B. Roget, and A. Wissink, “Progress in Strand Mesh Generation and Domain Connectivity For Dual-Mesh CFD Simulations,” *AIAA-2017-0288*, 2017.
- [31] J. Ray, C. A. Kennedy, S. Lefantzi, and H. N. Najm, “Using High-Order Methods on Adaptively Refined Block Structured Meshes Discretizations, Interpolations and Filters,” *SAND2005-7981*, Sandia National Laboratories, CA., 2005.
- [32] P. McCorquodale and P. Collella, “A High-Order Finite-Volume Method For Conservation Laws on Locally Refined Grids,” *Comm. in Applied Mathematics and Computational Science*, vol. 6, p. 1, 2011.
- [33] J. Hittinger and J. Banks, “Block-Structured Adaptive Mesh Refinement Algorithms For Vlasov Simulation,” *J. Comp. Phys.*, vol. 241, pp. 118–140, 2013.
- [34] G. Chesshire and W. D. Henshaw, “A Scheme for Conservative Interpolation on Overlapping Grids,” *SIAM J. Sci. Comput.*, vol. 15(4), pp. 819–845, 1994.
- [35] D. L. Brown, “An Unsplit Godunov Method for Systems of COnservation Laws on Curvilinear Grids,” *Mathl. Comput. Modelling*, vol. 20(10), pp. 29–48, 1994.
- [36] E. Pärt-Enander and B. Sjögreen, “Conservative and Non-Conservative Interpolation Between Overlapping Grids For Finite Volume Solutions of Hyperbolic Problems,” *Computer Fluids*, vol. 23(3), pp. 551–574, 1994.
- [37] K. Sebastian and C. Shu, “Multidomain WENO Finite Difference Method with Interpolation at Subdomain Interfaces,” *J. Scientific Computing*, vol. 19, pp. 405–438, 2003.
- [38] C. Hirsch, *Numerical Computation of Internal and External Flow*. J. Wiley & Sons, 1991.
- [39] Y. Kallinderis and A. Chen, “An Incompressible 3-D Navier-Stokes Method with Adaptive Hybrid Grids,” *AIAA-96-0293*, 1996.
- [40] R. Löhner, *Applied CFD Techniques, Second Edition*. J. Wiley & Sons, 2008.

- [41] P. Roe, “Approximate Riemann Solvers, Parameter Vector, and Difference Schemes,” *Journal of Computational Physics*, vol. 43, pp. 357–372, October 1981.
- [42] E. Turkel, “Preconditioned Methods for Solving the Incompressible and Low Speed Compressible Equations,” *Journal of Computational Physics*, vol. 72, no. 2, pp. 277 – 298, 1987, [https://doi.org/10.1016/0021-9991\(87\)90084-2](https://doi.org/10.1016/0021-9991(87)90084-2).
- [43] E. Turkel and A. Arnone, “Pseudo-Compressibility Methods for the Incompressible Flow Equations,” *NASA-CR-191533, NAS 1.26:191533, ICASE-93-66, AD-A273657*, October 1993.
- [44] E. Turkel, A. Fiterman, and B. van Leer, “Preconditioning and the Limit to the Incompressible Flow Equations,” *Journal of Applied Numerical Mathematics*, August 1993.
- [45] E. Turkel and V. Vatsa, “Local Preconditioners for Steady and Unsteady Flow Applications,” *ESAIM: M2AN*, vol. 39, no. 3, pp. 515–535, 2005, 10.1051/m2an:2005021.
- [46] D. Darmofal and K. Siu, “A Robust Multigrid Algorithm for the Euler Equations with Local Preconditioning and Semi-coarsening,” *J. Comput. Phys.*, vol. 151, no. 2, pp. 728–756, May 1999. [Online]. Available: <http://dx.doi.org/10.1006/jcph.1999.6216>
- [47] J. E. Kress, “An Unstructured Grid Incompressible Navier-Stokes Algorithm for Convective Heat Transfer Based on Artificial Compressibility,” Master’s thesis, The University of Tennessee, Chattanooga, December 2012.
- [48] E. Turkel, V. Vatsa, and R. Radespiel, “Preconditioning Methods for Low-Speed Flows.” 14th AIAA Applied Aerodynamics Conference, June 18-20, 1996, New Orleans, LA, 1996.
- [49] A. Jameson, W. Schmidt, and E. Turkel, “Numerical Solution of the Euler Equations by Finite Volume Methods Using Runge Kutta Time Stepping Schemes,” *14th Fluid and Plasma Dynamics Conference*, 1981. [Online]. Available: <https://arc.aiaa.org/doi/abs/10.2514/6.1981-1259>
- [50] J. C. Butcher, *Numerical Methods for Ordinary Differential Equations*. J. Wiley, 2003.
- [51] ——. John Wiley & Sons, Ltd, 2016, 10.1002/9781119121534. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119121534>
- [52] H. A. Luther, “An Explicit Sixth-Order Runge-Kutta Formula,” *Math. Comp.*, vol. 2, pp. 434–436, 1968, <https://doi.org/10.1090/S0025-5718-68-99876-1>.
- [53] G. D. Byrne and R. J. Lambert, “Pseudo-Runge-Kutta Methods Involving Two Points,” *J. ACM*, vol. 13, no. 1, pp. 114–123, January 1966, 10.1145/321312.321321. [Online]. Available: <http://doi.acm.org/10.1145/321312.321321>
- [54] Z. Jackiewicz, *General Linear Methods for Ordinary Differential Equations*. Wiley Online Library, 2009.

- [55] Z. Jackiewicz and S. Tracogna, “A General Class of Two-Step RungeKutta Methods for Ordinary Differential Equations,” *SIAM Journal on Numerical Analysis*, vol. 32, no. 5, pp. 1390–1427, 1995. [Online]. Available: <https://doi.org/10.1137/0732064>
- [56] R. Löhner, “Getting Discretization Errors From Black-Box Field Solvers,” unpublished - Internal Technical Report.
- [57] G. Zwas and S. Abarbanel, “Third and Fourth Order Accurate Schemes for Hyperbolic Equations of Conservation Law Form,” *Mathematics of Computation*, vol. 25, pp. 229–236, 1971.
- [58] F. Lörcher and C.-D. Munz, “Lax–Wendroff-type Schemes of Arbitrary Order in Several Space Dimensions,” *IMA journal of Numerical Analysis*, vol. 27, no. 3, pp. 593–615, 2007.
- [59] M. J. Berger and J. Olinger, “Adaptive Mesh Refinement for Hyperbolic Partial Differential Equations,” *J. Comp. Phys.*, vol. 53(3), pp. 484–512, 1984.
- [60] R. Pember, J. Bell, P. Colella, W. Crutchfield, and M. Welcome, “An Adaptive Cartesian Grid Method for Unsteady Compressible Flow in Irregular Regions,” *J. Comp. Phys.*, vol. 120, p. 278, 1995.
- [61] Y. Wang, S. Simakhina, and M. Sussman, “A Hybrid Level Set-Volume Constraint Method for Incompressible Two-Phase Flow,” *J. Comp. Phys.*, vol. 241, pp. 6438–6471, 2012.
- [62] M. O. K. Terekhov, K. Nikitin and Y. Vassilevski, “A Semi-Lagrangian Method on Dynamically Adapted Octree Meshes,” *Russ. J. Numer. Anal. Math. Modelling*, vol. 30(6), pp. 363–380, 2015.
- [63] R. Bermejo and A. Staniforth, “The Conversion of Semi-Lagrangian Schemes to Quasimonotone Schemes,” *Monthly Weather Review*, vol. 120, pp. 2622–2632, 1992.
- [64] T. Johnson and V. Patel, “Flow Past a Sphere up to a Reynolds Number of 300,” *Journal of Fluid Mechanics*, vol. 378, p. 1970, 1999.
- [65] R. Clift, J. Grace, and M. Weber, *Bubbles, drops, and particles*. Courier Corporation, 2005.
- [66] F. Roos and W. Willmarth, “Some Experimental Results on Sphere and Disk Drag,” *AIAA journal*, vol. 9, no. 2, pp. 285–291, 1971.
- [67] A. Tomboulides, S. Orszag, and G. Karniadakis, “Direct and Large-Eddy Simulations of Axisymmetric Wakes,” in *31st Aerospace Sciences Meeting*, 1993, p. 546.
- [68] G. Gassner and A. Beck, “On the Accuracy of High-Order Discretizations for Underresolved Turbulence Simulations,” *Theoretical and Computational Fluid Dynamics*, vol. 27, June 2012.
- [69] J. Bull and A. Jameson, “Simulation of the Compressible Taylor Green Vortex using High-Order Flux Reconstruction Schemes,” in *7th AIAA Theoretical Fluid Mechanics Conference*. [Online]. Available: <https://arc.aiaa.org/doi/abs/10.2514/6.2014-3210>

- [70] —, “Simulation of the Taylor-Green Vortex Using High-Order Flux Reconstruction Schemes,” *AIAA Journal*, vol. 53, no. 9, pp. 2750–2761, 2015. [Online]. Available: <https://doi.org/10.2514/1.J053766>
- [71] J. DeBonis, “Solutions of the Taylor-Green Vortex Problem Using High-Resolution Explicit Finite Difference Methods,” in *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, 2013, p. 382.
- [72] “The Fastest Fourier Transform in the West,” <http://fftw.org>.
- [73] H. Lienhart, C. Stoots, and S. Becker, “Flow and Turbulence Structures in the Wake of a Simplified Car Model Ahmed Model,” 2000, DGLR Fach Symp. der AG STAB, Stuttgart University.
- [74] —, “Flow and Turbulence Structures in the Wake of a Simplified Car Model Ahmed Model,” *Notes on Numerical Fluid Mechanics*, vol. 77, 2002, 10.1007/978-3-540-45466-3_39.
- [75] H. Lienhart and S. Becker, “Flow and Turbulence Structure in the Wake of a Simplified Car Model.” SAE International, March 2003, 10.4271/2003-01-0656.
- [76] S. Krajnović and L. Davidson, “Large-Eddy Simulation of the Flow Around Simplified Car Model,” SAE Technical Paper, Tech. Rep., 2004.
- [77] G. Franck, F. Carazo, N. Nigro, M. Storti, and J. D’Elia, “Simulacion Numerica del Flujo Alrededor del Modelo de Ahmed para un Angulo de Inclinacion Critico,” *Mecnica Computacional*, vol. XXIII, pp. 2189–2209, 2004.
- [78] M. Minguez, R. Pasquetti, and E. Serre, “High Order Large-Eddy Simulation of Flow over the Ahmed Body Car Model,” *Physics of Fluids*, vol. 20, no. 095101, 2008.
- [79] G. Franck, N. Nigro, M. Storti, and J. D’Elia, “Numerical Simulation of the Flow Around the Ahmed Vehicle Model,” *Latin American Applied Research*, vol. 39, pp. 295–306, 2009.
- [80] A. Thacker, “Contribution Expérimentale à l’analyse Stationnaire et Instationnaire de l’écoulement à l’arrière d’un Corps de Faible Allongement.” Ph.D. dissertation, 2010.
- [81] A. Thacker, S. Aubrun, A. Leroy, and P. Devinant, “Experimental Characterization of Flow Unsteadiness in the Centerline Plane of an Ahmed Body Rear Slant,” *Experiments in fluids*, vol. 54, no. 3, p. 1479, 2013.
- [82] D. Aljure, O. Lehmkuhl, I. Rodríguez, and A. Oliva, “Flow and Turbulent Structures Around Simplified Car Models,” *Computers & Fluids*, vol. 96, pp. 122–135, 2014.
- [83] N. Ashton, A. Revell, and R. Poletto, “Grey-Area Mitigation for the Ahmed Car Body Using Embedded DDES,” in *Progress in Hybrid RANS-LES Modelling*. Springer, 2015, pp. 119–129.
- [84] N. Ashton and A. Revell, “Key Factors in the Use of DDES for the Flow Around a Simplified Car,” *International Journal of Heat and Fluid Flow*, vol. 54, pp. 236–249, 2015.

- [85] O. Lehmkuhl, G. Houzeaux, H. Owen, G. Chrysokentis, and I. Rodriguez, “A Low-Dissipation Finite Element Scheme for Scale Resolving Simulations of Turbulent Flows,” *Journal of Computational Physics*, vol. 390, pp. 51–65, 2019.
- [86] “Line Integral Convolution, Paraview,” http://www.paraview.org/Wiki/ParaView/Line_Integral_Convolution.

Biography

Alejandro A. Figueroa graduated from La Salle High School, Cordoba, Argentina, in 2009. He finished his undergraduate studies in Mechanical Aeronautical Engineering from Instituto Universitario Aeronautico, Cordoba, Argentina, in 2014.

In 2015, he enrolled in the Physics Ph.D. Program at George Mason University. For the last three years of the Ph.D. Program, he was working on improving FDFLO, a high order cartesian finite difference code developed for industrial applications at the CFD Center at George Mason University. This work resulted in the present dissertation: ‘Improvement of nested cartesian finite difference grid solvers’, which was completed in November 2019.