

# **A Very High-Order Finite Volume Method Based on Weighted Least Squares for the Solution of Poisson Equation on Unstructured Grids**

**Artur Guilherme Rodrigues de Vasconcelos**

Thesis to obtain the Master of Science Degree in

## **Aerospace Engineering**

Supervisors: Prof. José Carlos Fernandes Pereira  
Dr. Duarte Manuel Salvador Freire da Silva de Albuquerque

### **Examination Committee**

Chairperson: Prof. Filipe Szolnoky Ramos Pinto Cunha  
Supervisor: Dr. Duarte Manuel Salvador Freire da Silva de Albuquerque  
Member of the Committee: Prof. José Paulo Baptista Moitinho de Almeida

**June 2017**



"Sabemos muito mais do que julgamos,  
podemos muito mais do que imaginamos"

José Saramago, El Mundo (1998)



## Acknowledgments

A terminar esta Tese, gostaria de deixar os meus sinceros agradecimentos às demais individualidades que contribuíram para a concretização deste que foi um projecto que resultou de muita dedicação e partilha de conhecimentos.

À minha família, em particular os meus pais e irmã por todo o apoio incondicional, exemplo e educação que me transmitiram ao longo destes anos permitindo que me tornasse na pessoa que sou hoje. Um agradecimento especial à Carla Ferreira por todo o seu carinho, amor, apoio e compreensão que demonstrou, assim como o encoragemento e incentivo nos momentos de maior desalento.

Ao Professor José Carlos Pereira por ter aceitado a desafiante proposta de ser o meu orientador. Fico grato pelo vasto conhecimento que me transmitiu e discussão de ideias sobre as mais variadas temáticas do CFD, assim como pela exigência e rigor que impôs para o enriquecimento desta Tese.

Ao Dr. Duarte Albuquerque, co-orientador desta Tese, cuja sua contribuição neste projecto foi imprescindível para a qualidade deste trabalho. Fico imensamente grato pelas suas revisões bibliográficas ao documento e pela sua enorme disponibilidade em me auxiliar nas mais diversas questões que surgiram, nomeadamente na árdua tarefa que é a busca por erros no código.

A toda a equipa do LASEF gostaria de salientar todo o seu espírito de camaradagem e alegria que tornaram esta caminhada, que por vezes incerta, mais prazerosa.

Deixo uma palavra para os meus amigos que gentilmente apoiaram-me no descorrer do meu percurso académico que ora findamos, nomeadamente ao João Gonçalo, ao Rubén Menezes, ao Lourenço, ao Couto, Daniela, Bruno Conceição, Gameiro, Verrissimo, Hugo Abreu. Pedindo desde já desculpa a possíveis esquecimentos.

Por fim gostaria de agradecer ao Professor André Marta por ter disponibilizado o template em Latex para a construção deste documento.

Foi de facto uma longa jornada que de valeu a pena.



## Resumo

É proposto um novo esquema de alta ordem com o método de volume finito para a solução da equação de Poisson em malha não-estruturada baseado no método dos mínimos quadrados ponderados. Este novo esquema consiste no cálculo do fluxo difusivo nas faces do volume de controlo, através de uma reconstrução polinomial centrada na face até oitava ordem de precisão. Foi utilizado um novo algoritmo para a extensão do *stencil* para que a alta ordem seja mantida junto das fronteiras computacionais.

Para optimizar a função de ponderação foi realizado um estudo paramétrico baseado no número de condição da matriz, ordem de convergência e nível de precisão dos esquemas. Deste estudo resulta uma nova função de ponderação para este tipo de esquemas de reconstrução. Os resultados indicam que os esquemas propostos alcançam a ordem de convergência teórica para todas as topologias de malha, validando esta metodologia. É proposta uma nova abordagem para as fronteiras de Neumann reduzindo o erro numérico quando comparada com o procedimento clássico. O estudo de qualidade de malha focado no ângulo de não-ortogonalidade e na razão de volumes indica que as ordens de convergência dos esquemas não são afectadas.

Os critérios de eficiência estudados (memória requerida e tempo de execução do *solver*) recomendam o esquema de 8<sup>a</sup> ordem em detrimento dos restantes esquemas, sendo ainda mostrado que as malhas poliédricas são mais eficientes que as malhas Cartesiana e triangular para ambos os critérios de eficiência estudados.

**Palavras-chave:** Esquemas difusivos de oitava ordem, Método de volume finito, Mínimos quadrados ponderados, Malha não-estruturada, Optimização da função de ponderação, Equação de Poisson.



## Abstract

A very high-order finite volume method is proposed for the solution of Poisson equation in unstructured grids based on the weighted least-squares method. The new method consists in a face centred reconstruction up to eight order of accuracy for the computation of the diffusive fluxes at the faces of the control volume. It uses a new stencil extension algorithm in order to maintain the local high order near the boundaries of the computational domain.

To optimize the weight function a parametric study is performed based on the matrix condition number, convergence order and error magnitude of the schemes. From this study a new proposed weight function is defined for this type of reconstruction schemes. The results showed that the proposed method achieves the theoretical convergence order for Cartesian, triangular, polyhedral and hybrid grids, validating the proposed methodology. A new approach for the Neumann boundary type is also proposed which reduces the error magnitude when compared with the general approach. A grid quality study based on non-orthogonality angle and volume ratio is done and shows that the schemes convergence order is not affected by those grid parameters.

The efficiency criteria, defined by required memory and solver-run time (SRT) indicates that the eight-order scheme is advantageous over the other lower order schemes, additionally it is demonstrated that the polyhedral grid yields more efficient solutions than the ones obtained with Cartesian or triangular ones.

**Keywords:** Eight-order diffusive scheme, Finite volume method, Weight least-squares, unstructured grids, Weighted function optimization, Poisson equation.



# Contents

Acknowledgments . . . . .	v
Resumo . . . . .	vii
Abstract . . . . .	ix
List of Tables . . . . .	xiii
List of Figures . . . . .	xv
Acronyms . . . . .	xix
Nomenclature . . . . .	xxi
Glossary . . . . .	1
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	3
1.2 Background . . . . .	5
1.3 SOL Code . . . . .	7
1.4 Objectives . . . . .	7
1.5 Present Contributions . . . . .	8
1.6 Thesis Outline . . . . .	9
<b>2 High-Order Schemes in One Dimensional Space</b>	<b>11</b>
2.1 Poisson Equation discretize by the Finite Volume Method . . . . .	11
2.2 Discretized Schemes . . . . .	12
2.2.1 Polynomial Reconstruction in 1D space . . . . .	12
2.2.2 Source Term . . . . .	13
2.3 Error Norms . . . . .	15
2.3.1 Convergence Order . . . . .	16
2.4 Results . . . . .	16
2.4.1 Explicit Face Flux Reconstruction . . . . .	17
2.4.2 Implicit Face Flux Reconstruction . . . . .	18
2.4.3 Deferred Correction (DC) Approach . . . . .	19
2.5 Conclusions . . . . .	21
<b>3 High-Order Schemes for Unstructured Grids</b>	<b>23</b>
3.1 Unstructured Grids . . . . .	23

3.1.1	Grid Quality . . . . .	26
3.2	Least Squares Approach . . . . .	27
3.2.1	Polynomial Reconstructions . . . . .	28
3.2.2	General Approach . . . . .	29
3.2.3	Interior Faces . . . . .	33
3.2.4	Boundary Treatment . . . . .	34
3.3	Source Term . . . . .	36
<b>4</b>	<b>Results and Discussion for Unstructured Grids</b>	<b>39</b>
4.1	Selection of Weight Function . . . . .	39
4.1.1	Selection by Properties of Matrix A . . . . .	40
4.1.2	Selection by Schemes Convergence Order . . . . .	42
4.1.3	Selection by Numerical Error Magnitude . . . . .	43
4.2	Influence of Grid Type . . . . .	44
4.2.1	Accuracy Order Verification . . . . .	45
4.2.2	Grid Comparison Study . . . . .	48
4.3	Influence of Neumann Boundary Conditions . . . . .	49
4.3.1	Imposed Not-Null Flux Case . . . . .	50
4.3.2	Imposed Null Flux Case . . . . .	52
4.4	Influence of Grid Quality Parameters . . . . .	54
4.4.1	Influence of Non-Orthogonality Angle . . . . .	54
4.4.2	Influence of Volume Ratio . . . . .	56
4.5	Efficiency Remarks . . . . .	58
4.5.1	Number of Non Zeros (NNZ) . . . . .	58
4.5.2	Solver-Run Time (SRT) . . . . .	60
4.5.3	Efficiency Variation with Grid Type . . . . .	61
<b>5</b>	<b>Conclusions</b>	<b>65</b>
5.1	Future Work . . . . .	66
<b>Bibliography</b>		<b>69</b>
<b>A</b>	<b>Stability Analysis of High-Order Diffusive Schemes</b>	<b>A.1</b>
A.1	Unsteady-State Heat Equation . . . . .	A.1
A.2	Stability Analysis by Von Neumann Method . . . . .	A.2
A.2.1	Fourier Series Decomposition . . . . .	A.2
A.3	Spectral Analysis of Numerical Errors . . . . .	A.3
A.4	Time Integration Methods . . . . .	A.5
A.4.1	Time Discretization . . . . .	A.6
A.5	Stability Domain . . . . .	A.9
A.6	Amplitude Error Optimization . . . . .	A.11

# List of Tables

2.1	Gauss points coordinates for an integration interval $[-1; 1]$ with the respective Gauss weights for the different orders accurate from Pina, [76] and Weisstein [77]. . . . .	14
3.1	Number of terms of the Taylor expansion required for a $p^{th}$ order polynomial at two dimensional cases. . . . .	29
3.2	Order of vertex neighbours required for a $p^{th}$ order polynomial. . . . .	33
3.3	Simplex coordinates and Gauss weight for a triangular element for a $p^{th}$ order integration, from Akin, [74]. . . . .	37
4.1	The matrix condition number obtained with $w_{LS_1}$ for the Cartesian grid (left) and hybrid grid (right). . . . .	41
4.2	The matrix condition number obtained with $w_{LS_2}$ for the Cartesian grid (left) and hybrid grid (right). . . . .	41
4.3	The matrix condition number obtained with $w_{LS_3}$ for the Cartesian grid (left) and hybrid grid (right). . . . .	41
4.4	The $\mathcal{O}_1$ of the schemes for Cartesian and hybrid grids with the weight function $w_{LS_1}$ , where the (–) indicates that a converged solution was not achieved. . . . .	42
4.5	The $\mathcal{O}_1$ of the schemes for Cartesian and hybrid grids with the weight function $w_{LS_3}$ , where (–) indicates that a converged solution was not achieved and (*) the discarded shape factors from the previous section. . . . .	43
4.6	The $\ e\ _1$ for a Cartesian grid with 25600 cells and a hybrid grid with 47534 cells for $k = 2$ and $k = q$ , where in the last line is the ratio between the errors of the two shape factors. . . . .	44
4.7	Obtained $\ e\ _1$ and $\mathcal{O}_1$ for Cartesian grid. . . . .	45
4.8	Obtained $\ e\ _\infty$ and $\mathcal{O}_\infty$ for Cartesian grid. . . . .	45
4.9	Obtained $\ e\ _1$ and $\mathcal{O}_1$ for triangular grid. . . . .	47
4.10	Obtained $\ e\ _\infty$ and $\mathcal{O}_\infty$ for triangular grid. . . . .	47
4.11	Obtained $\ e\ _1$ and $\mathcal{O}_1$ for polyhedral grid. . . . .	47
4.12	Obtained $\ e\ _\infty$ and $\mathcal{O}_\infty$ for polyhedral grid. . . . .	47
4.13	Obtained $\ e\ _1$ and $\mathcal{O}_1$ for hybrid grid. . . . .	47
4.14	Obtained $\ e\ _\infty$ and $\mathcal{O}_\infty$ for hybrid grid. . . . .	47
4.15	Reference length required for a $\ e\ _1$ of 1.00E-10 for all schemes and grid types. . . . .	49

4.16 Comparison between the two approaches for a problem with an imposed not-null Neumann BC for all schemes applied to the irregular polyhedral grids. . . . .	50
4.17 Comparison between the two approaches for a problem with an imposed not-null Neumann BC for all schemes applied to the triangular grids. . . . .	51
4.18 Comparison between the two approaches for a problem with an imposed null flux Neumann BC for all schemes applied to the irregular polyhedral grids. . . . .	53
4.19 Comparison between the two approaches for a problem with an imposed null flux Neumann BC for all schemes applied to triangular grids. . . . .	53
4.20 $\ e\ _1$ and $\mathcal{O}_1$ of the FLS4 scheme with the twisted grids. . . . .	54
4.21 $\ e\ _1$ and $\mathcal{O}_1$ of the FLS8 scheme with the twisted grids. . . . .	54
4.22 Ratio of mean and maximum errors for all schemes between an imposed $\alpha$ grid and the Cartesian one with 102400 cells. . . . .	55
4.23 $\ e\ _1$ and $\mathcal{O}_1$ for FLS4 scheme with perturbed grids. . . . .	57
4.24 $\ e\ _1$ and $\mathcal{O}_1$ for FLS8 scheme with perturbed grids. . . . .	57
4.25 Ratio of mean and maximum error norms for all schemes between grids with an imposed displacement and a Cartesian one with 25600 cells. . . . .	57
4.26 $\mathcal{O}_1^z$ and $\ e\ _1$ for Cartesian grid for FLS2, FLS4 and FLS8. . . . .	59
4.27 $\mathcal{O}_1^z$ and $\ e\ _1$ for irregular polyhedral grid for FLS2, FLS4 and FLS8. . . . .	59
4.28 Comparison of the required computational resources for all orders with a 1-norm level error of 4.30E-09 for Cartesian grid and 5.97E-09 for irregular polyhedral one, where 1Gb = 1024Mb. . . . .	60
4.29 $\ e\ _1$ and $\mathcal{O}_1^{SRT}$ for Cartesian grid for FLS2, FLS4 and FLS8. . . . .	60
4.30 $\ e\ _1$ and $\mathcal{O}_1^{SRT}$ for irregular polyhedral grid for FLS2, FLS4 and FLS8. . . . .	61
4.31 Comparison of the SRT for all orders with a 1-norm level error of 4.30E-09 for Cartesian grid and 5.97E-09 for irregular polyhedral grid. . . . .	61
4.32 Computational Resources required for a $\ e\ _1$ of 1.00E-10 for all schemes and grid types, where 1Tb = $1024^2$ Mb. . . . .	62
4.33 Solver-Run time required for a $\ e\ _1$ of 1.00E-10 for all schemes and grid types, in seconds. . . . .	63
A.1 Coefficients of the different RK schemes analysed. . . . .	A.9
A.2 Stability domains for the numerical schemes for the time integration method for the different space discretization orders. . . . .	A.9
A.3 $\beta$ that minimizes the amplitude error for high frequencies for the different time discretization methods. . . . .	A.11

# List of Figures

1.1	Simulation of the flow around the KHEOPS, for EXPERT program, during the atmospheric re-entry, using a computational code LORE, it is possible to observe the heat flux around vehicle surface and the boundary layer separation, which was determined during the hypersonic phase, figure obtained from Muyaert et al. [1]. . . . .	2
1.2	Example of growing use of CFD tools in a today's aircraft, Airbus A380, picture obtained from Jameson [2]. . . . .	3
1.3	Example of a 3D CFD simulation over an Airbus A380 performed by DLR flow, this simulation is similar to the one performed by Jameson in the 80's at time with a Boeing 747-200, picture obtained from Jameson [2]. . . . .	4
1.4	The requirements to have an accurate, robust and efficient CFD code capable to obtain realistic flow field simulations. . . . .	5
2.1	A FV discretization over the CV $i$ ( $\bullet$ - Cell Center). . . . .	12
2.2	Convergence curves for uniform grid of $\ e\ _1$ and $\ e\ _\infty$ for all schemes using an explicit face flux reconstruction, where the dotted line represents the theoretical slope for each scheme. . . . .	17
2.3	Convergence curves for a non-uniform grid of $\ e\ _1$ and $\ e\ _\infty$ for all schemes using an explicit face flux reconstruction, where the dotted line represents the theoretical slope for each scheme. . . . .	18
2.4	Convergence curves of a non-uniform grid for $\ e\ _1$ and $\ e\ _\infty$ for all schemes using an implicit face flux reconstruction, where the dotted line represents the theoretical slope of each scheme. . . . .	19
2.5	Convergence curves for a non-uniform grid of $\ e\ _1$ for a 2 <sup>nd</sup> and 4 <sup>th</sup> order implicit approach for the remaining schemes, where the dotted line represents the theoretical slope of the scheme. . . . .	20
3.1	Cell-centered FV discretization over the CV $P$ ( $\bullet$ - Cell Center). . . . .	24
3.2	Examples of a Cartesian, triangular, polyhedral and hybrid grids, which were used by the Author in this Thesis. . . . .	25
3.3	Example of the skewness and non-orthogonality angle grid quality parameters, where $\bullet$ are the centroids of the cells $P_1$ and $P_2$ and face $f$ . . . . .	26

3.4	Twist grid, with two different non-orthogonality angles: 10° and 40°. . . . .	27
3.5	Perturbed grids with two different imposed displacements applied to a Cartesian grid with 400 cells. . . . .	28
3.6	Example of the different orders of vertex neighbours of a face, marked in red, where the numbers inside of each cell represents the vertex neighbouring order of that cell relatively to the marked face. . . . .	33
3.7	Example of a stencil extension in an elliptic hybrid grid for a cubic polynomial reconstruction for the face represented with the red color and the $\triangleleft$ mark, the blue ones are the other boundary faces included in the regression. . . . .	35
3.8	Example of a division of a polyhedral cell in triangles in order to perform the numerical integration of source term at cell $P$ , in red is the face's sub-triangles. The dots represent the Gauss points coordinates of the GLQ . . . . .	36
4.1	The $\ e\ _1$ for all high-order schemes, where the $\bigcirc$ is the proposed shape factor and the $\square$ is the shape factor used by most of the other authors. . . . .	43
4.2	Convergence curves for all grid types for $\ e\ _1$ and $\ e\ _\infty$ with all schemes, where the dotted line represents the theoretical slope of each scheme. . . . .	46
4.3	Convergence curves of the error 1-norm with the $h_{ref}$ for the different grids with the FLS4 and FLS8. . . . .	48
4.4	Convergence curves of the mean error for mixed boundary conditions with irregular polyhedral and triangular grids and all schemes with an imposed not-null flux. The dotted lines are the convergence curves for the GC approach and the solid ones represent the convergence curves with the DC <sub>N</sub> approach. . . . .	51
4.5	Convergence curves of the mean error for mixed boundary conditions with irregular polyhedral and triangular grids for all schemes with an imposed null flux. The dotted lines are the convergence curves for the GC approach and the solid ones represent the convergence curves with the DC <sub>N</sub> approach. . . . .	52
4.6	Convergence curve of the $\ e\ _1$ for FLS4 and FLS8 schemes with the twisted grids. . . . .	55
4.7	Error distribution for FLS8 scheme for two different $\alpha$ . . . . .	56
4.8	Convergence curves for different imposed perturbed grids with FLS4 and FLS8 schemes. . . . .	56
4.9	Error distribution for FLS8 scheme for two different $\gamma$ with a grid with 25600 cells. . . . .	58
4.10	Convergence curves of error 1-norm with respect to $N_z$ with the Cartesian and polyhedral grids for all schemes, where the dotted line represents the theoretical slope of each scheme. . . . .	59
4.11	Mean error with SRT diagram for the proposed schemes for Cartesian and polyhedral grids. . . . .	61
4.12	Convergence curves of the error 1-norm with respect to $N_z$ (a) and SRT (b) for different grids types with the FLS8 scheme. . . . .	62
A.1	Stability domains for the different time schemes for all space discretizations. . . . .	A.10
A.2	Curves of the amplitude error where $\theta = \pi$ and function of $\beta$ for several time schemes for all space discretization orders. . . . .	A.12

A.3 Curves of the amplitude error function of  $\theta$  for the optimal  $\beta$  value for several time schemes  
for all space discretization orders, the dotted lines represents the  $\beta_{\max}$  (only applicable to  
the Euler Explicit scheme). . . . . A.13



# Acronyms

1D	1 Dimensional Space.
2D	2 Dimensional Space.
3D	3 Dimensional Space.
AB	Adams-Bashforth.
BC	Boundary Condition.
BE	Backward Euler.
CFD	Computational Fluid Dynamics.
CN	Crank-Nicholson.
CS	Control Surface.
CV	Control Volume.
DC	Deferred Correction.
DC <sub>N</sub>	Dimensional Correction.
DG	Discontinuous Galerkin.
DNS	Direct Numerical Simulation.
EE	Euler Explicit.
ENO	Essentially Non-Oscillatory.
FCS	Face Centred Scheme.
FD	Finite Difference.
FDM	Finite Difference Method.
FEM	Finite Element Method.
FLS	Face Least-Squares.

FV      Finite Volume.

FVM     Finite Volume Method.

GC      General Case.

GP      Gauss Point.

GQ      Gauss Quadrature.

GW      Gauss Weight.

HO      High-Order.

LES     Large Eddy Simulation.

LO      Low-Order.

LS      Least-Squares.

NNZ     Number Non-Zeros.

PDE     Partial Differential Equations.

RHS     Right Hand Side.

RK      Rung-Kutta.

SRT     Solver Run-Time.

WLS     Weighted Least-Squares.

# Nomenclature

## Greek symbols

$\alpha$	Non-Orthogonality Angle.
$\beta$	Diffusive Criteria.
$\Phi$	Transportable Variable, Vector Form.
$\Gamma$	Faces Displacement.
$\gamma$	Grid Perturbation.
$\kappa$	Thermal Conductivity.
$\lambda$	Wavelength.
$\Omega$	Eigenvalue of the Matrix.
$\Phi$	Wave Amplitude of Transportable Variable.
$\phi$	Transportable Variable.
$\psi$	Auxiliary Analytical Solution.
$\tau, \sigma$	Runge-Kutta Parameters.
$\theta$	Phase Angle.
$\varepsilon$	Error.
$\varphi_\phi$	Source Term.
$\zeta$	Natural Coordinates.

## Math operators

$\Delta$	Difference Operator.
$\frac{\partial}{\partial t}$	Derivative with respect to time.
cond	Condition Number.
Im()	Imaginary Part of a Variable.

$\text{Re}()$  Real part of a variable.

$\max$  Maximum of a function.

$\min$  Minimum of a function.

$\nabla$  Gradient Operator.

$\nabla \cdot$  Divergence Operator.

$\nabla^2$  Laplace Operator.

$|\bullet|$  Absolute Value.

$\|\bullet\|_x$  x-Norm.

$I$  Imaginary Number.

### Roman symbols

$A$  Global Matrix.

$b$  Right Hand Side Vector.

$c$  Vector of Reconstructions Constants.

$D$  Matrix of Combination of Distance Moments.

$d$  Distance Moment Vector.

$f$  Face Centroid Cartesian Coordinates.

$I$  Identity Matrix.

$n$  Normal Vector.

$P$  Cell Cartesian Coordinates.

$P$  Pseudo Inverse Matrix.

$r$  Weighted Residual, Vector Form.

$S$  Face Area.

$T$  Flux Reconstruction Coefficients Matrix.

$W_{LS}$  Weight Function Matrix of Weighted Least-Squares Method.

$x$  Cartesian Space Coordinates, Vector Form.

$D_w$  Matrix of Combination of Distance Moments with Weight Function.

$C$  Set of Cells.

$F$  Set of Faces.

$\mathcal{G}$	Set of Gauss Points.
$\mathcal{N}$	Set of Stencil Elements.
$\mathcal{O}$	Order of Convergence.
$\mathcal{T}$	Set of Triangular Elements in a 2D GQ.
$a$	Coefficient of Finite Difference Equation.
$C$	Reconstruction Constant.
$D$	Space Dimension.
$d$	Distance.
$e$	Error.
$f$	Generic Function.
$G$	Amplification Factor.
$h$	Hydraulic Diameter, Grid Size, Cell Size.
$k$	Shape Factor or Wave Number.
$L$	Total Length.
$n_G$	Number of Gauss points.
$n_s$	Stencil Size.
$n_x$	Size in X Direction.
$N_z$	Number of NNZ entries.
$n_{cells}$	Number of Cells.
$n_{coefs}$	Number of Reconstruction Coefficients.
$p$	Order of Reconstruction Polynomial.
$q$	Order of the Scheme.
$r$	Ratio Between two Quantities.
$r$	Weighted Residual.
$t$	Time (seconds).
$V$	Cell Volume.
$w_G$	Gauss Weight.
$w_{LS}$	Weight Function of WLS Method.

$x, y$     Cartesian Space Coordinates.

### Subscripts

$\Delta$     Triangle.

$b$     Boundary.

$e$     East Face.

$f$     Face f.

$G$     Gauss.

$i, j$     Matrix Indexes.

$n$     Norm.

$w$     West Face.

$x, y, z$     Cartesian Components.

$\max$     Maximum Value.

$\min$     Minimum Value.

$\text{opt}$     Optimal Value.

$\text{ref}$     Reference Value.

### Superscripts

$A$     Analytical.

$n$     Time Step, Iteration.

$s$     Stencil.

$SRT$     Solver Run-Time.

$z$     Number of Non-Zeros.

$-1$     Inverse.

$T$     Transpose.

# Chapter 1

## Introduction

The fluid flows can be described by a set first and second order of partial differential equations (PDEs), being possible to determine some dynamic flows properties like pressure, velocity and temperature by solving these equations. However, due to some mathematical complexities of the model and the geometry used, in a very few situations analytical solutions can be obtained. As consequence of this, it is important to develop a valid approach to obtain the flow variables for all required conditions, such as computational simulations and experimental tests, this one is beyond the scope of this Thesis.

Furthermore in a laboratory, there are some cases that are impossible to recreate the real conditions with a scale model and for these situations numerical simulations are needed in order to study the fluid flow field. Figure 1.1 is one of these examples, which shows a fluid flow simulation over a space vehicle during the atmosphere re-entry, the KHEOPS capsule, which is an unmanned capsule developed by the European Space Agency and used for the EXPERT program that is a low cost hypersonic flight program, which allows different flying trajectories with velocities varying from 5000 to 7000  $m/s$ . It is possible to observe in the figure 1.1 the heat flux around the vehicle and the boundary layer separation effect in the deflected flap that affects the flap control and the heating associated to the shear layer reattachment [1].

The numerical calculation of the flow variables using a computer, which has been gaining a prominent position due to the technological advances that have been verified in the last decades, specially with regard to computation power. The calculation of the fluid flow properties using the numerical calculation is known as Computational Fluid Dynamics (CFD) and it has become a valid tool for fluid flow simulation, specially when complex geometries or physical models are used or when the real fluid flow conditions are impossible to be applied on a scaled model in a laboratory, such as the atmosphere re-entry of a space vehicle, example presented in figure 1.1.

The CFD, has the capability to simulate the most extreme environment conditions and flow velocities, measuring several fluid flow variables at the same time allowing a cost reduction, although it may not always be true. As disadvantages, it has the time required to converge a simulation, which in some cases may be long, also the costs associated to the code development or model and without a certain methodology it is impossible to guarantee that the obtained behaviour for a fluid flow is the one that

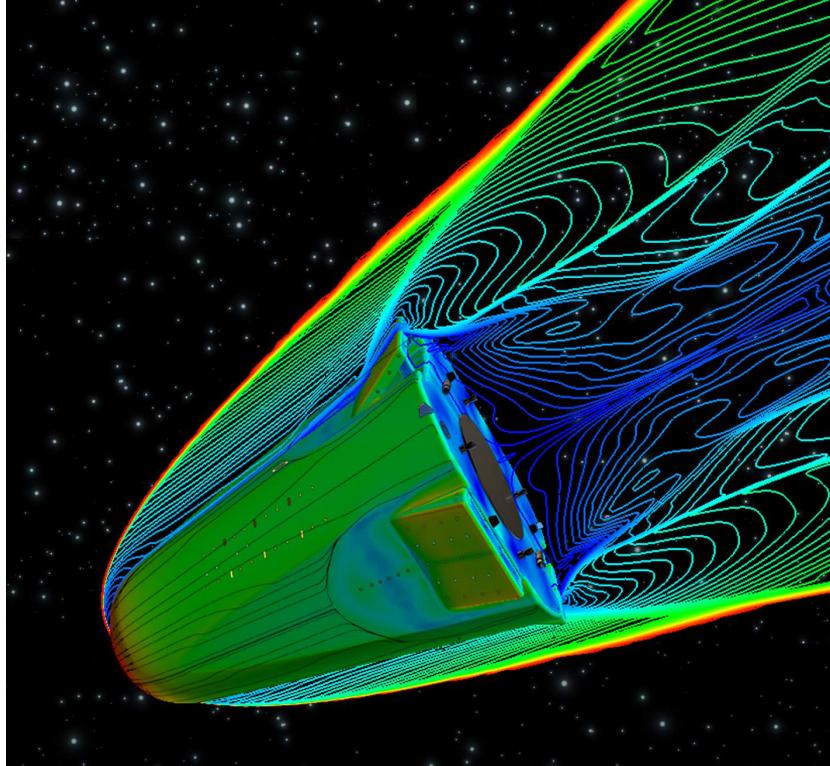


Figure 1.1: Simulation of the flow around the KHEOPS, for EXPERT program, during the atmospheric re-entry, using a computational code LORE, it is possible to observe the heat flux around vehicle surface and the boundary layer separation, which was determined during the hypersonic phase, figure obtained from Muylaert et al. [1].

really happens on the real life situation, for example it should not rely on a CFD simulation which an aircraft really flies, as it would be necessary a wind tunnel test to complement those tests. Although this is a good example where the CFD and the experimental tests must be interconnected, some preliminary designs can be eliminated from the CFD simulations and only the most promising ones tested on a wind tunnel.

Nowadays, CFD is no longer seen as an academic tool and has become reliable enough to be applied in industry, being important the emergence of the commercial CFD codes with an user friendly interface, such as StarCCM+, Fluent, among others, there is still the OpenFOAM, which is an open source option of these codes. They have became the main tool for the development process, from preliminary design stage to the shape optimization one, figure 1.2 shows where CFD is nowadays used in transport aircraft design and it is possible to observe that is used in many stages of the development of the aircraft.

Currently the goal of CFD is to find more accurate and reliable solutions to fluid dynamic problem while minimizes the computational time and required memory. Most of the simulation codes for aerospace industry are not based on general software but in own developed software. This is due to the high accuracy required for most of the applications.



Figure 1.2: Example of growing use of CFD tools in a today's aircraft, Airbus A380, picture obtained from Jameson [2].

## 1.1 Motivation

It is possible to use CFD for aerodynamic applications, where the main goal is the simulation of a flow field around a complex geometry, usually a three dimensional (3D) one, which allows the computation of the aerodynamics force coefficients applied over all surfaces of the geometry. Examples of simulation of 3D flow field around a realistic aerodynamic surface date from the 80's, see e.g. Jameson, [3, 4], at that time the aerodynamic surface studied was a Boeing 747-200 aircraft and more recently the Airbus A380, figure 1.3.

These results have "shaped" the way how the aerodynamic can be studied and with this change all the aerodynamic design phase and aerodynamic optimization of an aircraft. According to Johnson et al., [5], at Boeing Commercial Airplanes in Seattle in 1973 around 200 CFD simulations about vehicles were done and in 2002 more than 20000 CFD simulations were done for the same purpose, which means that there is an exponential growing in the CFD importance for the development of aerodynamic surfaces.

It is important to emphasize that to obtain a realistic simulation of a flow field around an aerodynamic surface it is necessary that the CFD code accomplish the following requirements: a capable mesh generator, an accurate physical model which includes a robust turbulent model and an accurate fluid flow solver as depicted in the flowchart of figure 1.4.

The mesh generator discretizes the domain, which is one of the most important tasks in a CFD simulation, since it is only possible to achieve accurate solutions in a mesh with a good quality. Even with a very good physical model a bad mesh generator can origin very inaccurate results. This phase is

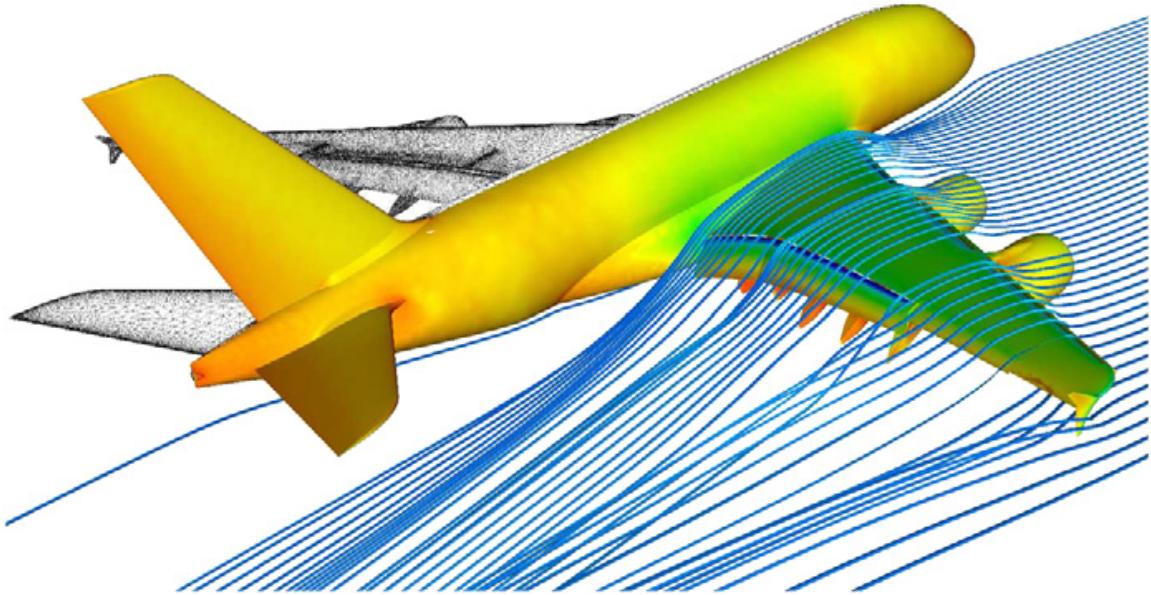


Figure 1.3: Example of a 3D CFD simulation over an Airbus A380 performed by DLR flow, this simulation is similar to the one performed by Jameson in the 80's at time with a Boeing 747-200, picture obtained from Jameson [2].

the one that consumes more time in the simulation. According to a study performed by Mavriplis, [6], in a realist aerodynamic application used in aerospace engineering the time that is required for the mesh generation could be 45 times greater than the required one for the rest of fluid flow simulation. Currently one of the research subject field is to develop new tools that decrease the computational cost of the mesh generation without decreasing the mesh quality, specially for the 3D complex geometries. Ideally the mesh generator should be able to generate a mesh, having into account the geometry and physics of the problem without many interventions from the user, this means that the code should be capable of selecting the areas where it is necessary to do mesh refinements without inputs from the user. The unstructured grids have became very popular in today's CFD applications, since they are more efficient, capable of solving more complex aerodynamic problems, and also can be applied local refinements and adaptive solvers according to Venkatakrishnan, [7].

The implementation of the physical model is important, since it contains the system of equations of the problem, being its choice crucial for the success of the method. The model is determined from the mathematical modelation of the flow physics, it specifies the system of equations and the boundary conditions. Most of the aerospace engineering applications involve turbulent flows. Direct Numerical Simulation (DNS) of turbulent flows are not feasible for next decades due the computation limitations, namely memory and speed ones, see e.g. Bailey, [8], or Spalart, [9]. Nowadays Large Eddy Simulation (LES) is a promising model but it requires high-order accuracy in time and space as well 3D complex geometries.

The last requirement is the fluid flow solver that must be accurate, efficient and robust for a generic mesh, which are important aspects of a CFD code. It should be stable and converge to a correct solution with a reasonable cost. A generic solver can be divided into three parts: equation's discretization, numerical fluxes computation and the time advancing method. Currently the CFD algorithms have some

accuracy and efficiency limitations, being more evident when the fluid flow is complex which, requires more time and memory to achieve an accurate solution. To avoid these limitations, a simplified mathematical model is used, neglecting some issues caused by this approximation, this process is known by the discretization of equations, which error can be estimated by  $\mathcal{O}(h^q)$  for problems with a smooth solution, being  $h$  the mesh reference length and  $q$  the order of accuracy of the discretization. As a consequence, the accuracy of a numerical solution can be improved by two processes: mesh refinement or a higher-order discretization.

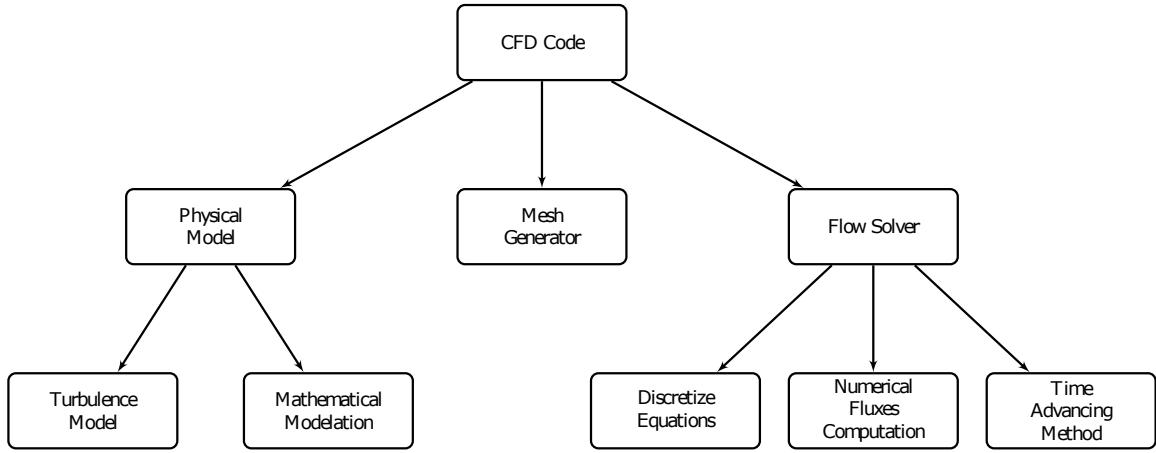


Figure 1.4: The requirements to have an accurate, robust and efficient CFD code capable to obtain realistic flow field simulations.

The scope of this Thesis is the development of high-order accurate viscous flow methods, namely the diffusive flux computation, where second-order discretization prevails. The convergence of high-order methods is more efficient than second order methods specially for unstructured grids, consequently the implementation of a higher-order discretization method for unstructured grids with an implicit reconstruction is extremely helpful, if not mandatory, to obtain an accurate solution.

## 1.2 Background

The numerical solution of transport phenomena (fluid flow, heat and mass transfer, etc.) in complex physical and geometrical configurations is a subject of continuous development regarding accuracy, robustness and efficiency. The geometrical complexity can be handled with different grid topologies such as polyhedral cells and their issues of robustness and efficiency are relevant for industrial applications. Particularly for aerospace, see e.g. Lê et al., [10], and Drikakis et al., [11], where high-order accurate methods for unstructured mesh finite volume (FV) computations have historically been focused on hyperbolic equations and in particular on advective flux.

Nowadays the focus is to obtain an accurate solution with a computational cost reduced to a minimum, which will allow to solve more complex problems and faster. Higher-order (higher than second-order) computation and particularly very high-order (higher than fourth-order) is a demanding issue, motivated by a potential cost reduction for complex CFD simulations. High-order methods have the potential to achieve an accurate solution with a coarse grid and a lower computational cost than a fine one

with a second-order method, see e.g. Zing et al., [12], for structured grids and Lipnikov and Manzini, [13], for unstructured grids.

Barth and Frederickson, [14], were the pioneers on the development of a high-order Finite Volume Methods (FVM) for unstructured grids using a quadratic polynomial, which creates a third-order method that was applied for the resolution of the Euler equation. In last decades high-order methods have become very popular. This is highlighted in the polynomial reconstruction technique applied to FVM by Ollivier-Gooch et al., [15–19], Cueto-Felgueroso et al., [20–25], Nogueira et al., [26,27], with fourth-order methods and Clain et al., [28–32] with results reported up to sixth-order.

In finite difference (FD) field many methods have been successfully developed for structured grids, see e.g. Visbal and Gaitonde, [33], which are superior in terms of computational cost and efficiency, for boundary layer simulations, see Deng et al., [34]. Still in FD framework there are the Padé compact schemes, which are well established for structured grids with a compact stencil, see e.g. Pereira et al., [35], with a fourth-order accurate scheme.

In Essentially Non-Oscillatory (ENO) schemes field, which is included all its variants, like WENO or CENO, stand out the works of Ollivier-Gooch, [36], Ivan and Groth, [37], Hu and Shu, [38] and Friedrich, [39], all of them with fourth-order methods. In the mimetic schemes, they have reached up to the fourth-order accurate methods, see Lipnikov, [13, 40]. For the case of spectral FDM the highlighted works are Liu et al., [41], and Wang et al., [42], with a fourth-order accurate scheme and the Fourier pseudo-spectral methods with the work of Mariano et al., [43].

Finally with the finite element method (FEM) there exist works that achieved up to fourth-order accurate, like Bassi and Rebay, [44], or Marques et al., [45], and more recently Bériot et al., [46], have reached to the tenth-order accurate. The Discontinuous Galerkin (DG) Method is a class of the FEM that has become popular, namely for compressible flow simulations. As examples of DG are the works of Shu et al., [47,48] with a sixth and seventh order schemes, the Arnold et al., [49], Dumbser et al., [50], Nogueira et al., [51], and J. Jaśkowiec et al., [52], that reaches up to the sixth order accurate. Finally the work of V. Etienne et al., [53], which applied to their high-order schemes a hp-adaptive method.

In the last years the resolution of parabolic and elliptic problems in unstructured grids has made significant progress, namely with the introduction of very high-order methods, see e.g. Bouzaras et al., [32], Bertolazzi et al., [54], Droniou, [55]. Poisson problems, see Batty, [56], heat transfer problems, see e.g. Chantasiriwan, [57], diffusion equations with variable coefficients, see Zhai, [58], or discontinuous coefficients, see e.g. Clain et al., [30]. These are some problems that can use these methods.

The coupling of Euler system with viscous terms see e.g. Gooch et al. [15, 17, 36, 59, 60], Barth and Frederickson, [14], Ivan and Groth, [37], Felgueroso et al., [21], and Toro and Hidalgo, [61] requires diffusive schemes for the discretization of the viscous terms. Another case where the diffusive schemes is required is the shallow-water flow, see Felgueroso et al., [20], and Cea et al., [62].

Other applications includes the incompressible Navier-Stokes equations, namely for the projection methods, like the SIMPLE and PISO, that requires the computation of face gradients (diffusive terms) for the velocity field updates, presented in works of Nogueira et al., [27], and Guermond et al., [63].

These are some examples that use elliptic operators and they are the main reasons for the devel-

opment of the very high order schemes in the Finite Volume (FV) framework. Another advantage of the development of these schemes is the possibility of the resolution of Partial Differential Equations (PDE) with high-order derivatives (greater than second one) which is not possible with the classic second order accurate schemes, see e.g. Guo and Lu, [64].

### 1.3 SOL Code

Created in 2003 by the LASEF research group, the SOL code is an in-house code which has as main objective the development of a common tool for CFD relating research. This tool is written in C language and provides to the researchers a way of benefiting from the different synergies that working in the same numerical code can generate, in opposite to working alone in their own code.

Since the beginning of this project several researches had work on its improvement through the implementation of new functionalities, which begin with Magalhães, [65], the first coordinator and the Author of this project with his PhD work on the development and implementation of numerical problems in adaptive grids. The current coordinator, Albuquerque, [66], where in his PhD Thesis created an importing grid system and reviewed all aspects already implemented, moreover the implementation of several second order accurate schemes which are able to solve problems both in unstructured and adaptive grids. More recently, several works about immersed boundaries were done by Pereira, [67], and Martins, [68], and one about turbulent jet flows with LES by Leite, [69]. Finally a mention for Reis, [70], with his Master Thesis, where a parallel computation for the resolution of the Euler equations was implemented.

It is important to list the most important features of the SOL code for the work here presented. At the moment it is possible to import unstructured grids from both CFD codes OpenFOAM and STAR CCM+® through the read of two ASCII files. It is also possible to create grids with other softwares, since they generate files in the correct format to be able the reading in SOL code. With this feature it is possible to maintain and update the grid database with several grid types for testing the different numerical schemes and new approaches.

This tool uses several libraries code packages, like AZTEC, METIS, LAPACK, BLAS, ATLAS and TECIO, where it is possible to find the sparse matrix structure and the solvers used in this Thesis. These libraries are also used by the majority of in-house CFD codes around the world used for research. Finally, at this moment the code has more than 160 thousand lines of code and 60 files.

### 1.4 Objectives

The work presented in this dissertation has the following major objective:

- Development and implementation of very high-order schemes, up to 8<sup>th</sup> order, based on WLS for one dimensional (1D) and two dimensional (2D) space for Poisson equation;

To achieve this final goal several objectives had to be met:

- i) A new approach for stencil construction on unstructured grids at the interior faces;
- ii) A novel boundary treatment, namely in the stencil with a new extension algorithm in order to maintain the local theoretical order of the scheme;
- iii) Optimization study optimization of weight function of the weighted least-squares (WLS) method used in the proposed schemes based in different functions of the literature and shape factors;
- iv) Verification of the proposed schemes for several grid types and analytical solutions;
- v) Grid quality study for the proposed schemes based on the following parameters: non-orthogonality angle and volume ratio;
- vi) Efficiency study based on computational costs, namely in the amount of memory and solver-run time (SRT);
- vii) Testing and verification of two new reconstruction approaches for the Neumann boundary conditions for the both cases of null and not-null imposed flux.

## 1.5 Present Contributions

In this section, it is intended to elucidate the reader about the work performed by the Author that was required in order to achieve the present results. The Author's work can be divided in two parts, the one done in Matlab 2015 environment and the other one done in the SOL code.

In Matlab, a numerical code was implemented for the solution of the 1D Poisson equation based on FVM with very high-order schemes capable to execute in both uniform and non-uniform spaced grids, called Solver1D. This code is able to switch between explicit, implicit flux reconstruction and deferred correction (DC) approach to solve the Poisson equation. All the 1D results presented in this Thesis for 1D problem were obtained with this numerical code.

It was also implemented a Matlab code, which goal was to test the proposed schemes in a Cartesian framework, which afterwards would help the same implementation in an unstructured framework at the SOL code. This numerical code, called Solver2D, is based on FVM and WLS reconstruction. To turn possible the execution of this code, it was necessary to implement several functions from bottom up, excluding the solvers and preconditioners. This algorithm uses an unstructured grid structure in order to follow and test the steps of the algorithm implemented in the SOL code, specially the algorithms of stencil construction and the boundary treatment with the new stencil extension technique.

The Solver2D allows the user to choose and test several aspects of the simulation like the analytical solution, the boundary conditions (Dirichlet or Neumann), the solver (GMRES or BICGSTAB), preconditioner (ILU(0)) and show the global matrix properties, such as the matrix condition number and spectral radius.

In SOL, all aspects related with the high-order methods were fully developed and implemented by the Author. These functions include the polynomial reconstructions, cells gathering for the stencil with

the required extension for the boundary treatment, the different weight functions and the calculation of Gauss points coordinates for any generic face and cell used in the numerical integration.

With regard to the implicit flux reconstruction approach, it was necessary to programme new functions like the matrix transpose, the matrix product and a generic square matrix inverse algorithm with the Gauss elimination method in order to obtain the pseudo inverse matrix. Once the SOL did not have an implicit reconstruction method, an implicit version of the explicit second order scheme was implemented.

Additionally, the grid database was updated with the introduction of new grids, namely the regular triangular grids and the ones with imposed volume ratio and non-orthogonality angle, that were programmed in Matlab. Finally an exporter of the global matrix and source term was created in order to compute sparse matrix properties with Matlab functions, such as the matrix condition number and spectral radius.

## 1.6 Thesis Outline

This Thesis is divided into five chapters, being its outline as follows. The chapter 2 introduces the Poisson equation and its discretization with the finite volume method, namely for the unidimensional case. The proposed very high-order schemes are tested in 1D space in order to verify the schemes convergence order with three different problems solving approaches. Finally the major conclusions about these results were presented.

Chapter 3 is dedicated to the unstructured grids, where all grid types at study are explained and the equations deduced in chapter 2 are extended for the two dimensional space case. The numerical method is explained with a special focus for the weighted least-squares method and the stencil selection technique for both interior and boundary faces. It also explains the high-order integration of the source term and several aspects of the numerical method implementation.

Chapter 4 shows the numerical tests performed to demonstrate the effective convergence order with the new very high-order scheme. A weight function optimization study it is also presented and it shows that a careful choice allows for a reduction of the error magnitude. The influence of mesh topology on schemes accuracy is studied and suggests that polyhedral grids have advantages over the other ones. The Neumann boundary condition is also analysed and compared with a new proposed approach, which reduces the error for both null and not-null imposed fluxes. Afterwards, a grid quality study is carried out for two parameters: the non-orthogonality angle and the volume ratio. Finally an efficiency study based on the required memory and solver run time is shown, where the advantages of the eight order and the polyhedral grids are shown and proven.

At the end, in chapter 5, the main conclusions of the Thesis are presented and they are focused on the efficiency of the proposed schemes when applied to an unstructured polyhedral grid. This chapter finishes with a section dedicated to suggestions and new topics in this for future work research field.



## Chapter 2

# High-Order Schemes in One Dimensional Space

This chapter describes the proposed schemes for diffusive term discretization of a transport equation in 1D space. Three different approaches to compute fluxes reconstructions are studied: explicit, implicit and deferred correction. Afterwards the error norms used in this Thesis are described and finally, a detailed analysis of the results obtained with the proposed scheme for 1D space is presented.

### 2.1 Poisson Equation discretize by the Finite Volume Method

The diffusive terms consists in elliptic operators and the Poisson equation is an example of a pure isotropic diffusive steady-state case that in the conservative form is given by:

$$\nabla \cdot \nabla \phi = \varphi_\phi, \quad (2.1)$$

where  $\phi$  is the transported variable,  $\varphi_\phi$  is the source term.

The Finite Volume (FV) discretization is based on the integral form of the conservative law presented in equation (2.1), which is obtained by the integration over a Control Volume (CV) or cell:

$$\int_{CV} \nabla \cdot \nabla \phi dV = \int_{CV} \varphi_\phi dV. \quad (2.2)$$

By applying the Gauss Divergence Theorem, which relates a volume integral with a surface integral to all CV's faces, to expression (2.2) results in:

$$\int_{CS} \nabla \phi \cdot d\mathbf{S} = \int_{CV} \varphi_\phi dV, \quad (2.3)$$

where  $d\mathbf{S}$  is a vector that takes into account the unitary outward normal and the surface area of the cell's faces. Finally should be noted that the mathematical operations done to obtain the equation (2.3) are exact and do not require any approximation.

Since this chapter only focus in 1D cases, the CV will be defined as a line segment and the faces will have an area equal to one, figure 2.1 shows the divisions of a typical 1D continuous domain in non-overlapping volumes (cells). The information about the transported variable  $\phi$ , is stored in each cell centroid.

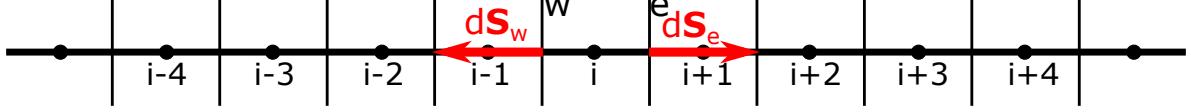


Figure 2.1: A FV discretization over the CV  $i$  ( $\bullet$  - Cell Center).

The discretization of the equation (2.1) with FVM cell centered approach requires the calculation of the east and west face gradients,  $\nabla\phi_e$  and  $\nabla\phi_w$ , respectively,

$$\nabla\phi_e - \nabla\phi_w = \int_{CV} \varphi_\phi dV, \quad (2.4)$$

where it has been taken into account the directions of the  $dS$ , for an easier understanding of the discretization of equation (2.4) the reader should see figure 2.1.

### Boundary Conditions (BC)

To have a well-posed problem of PDE's, it is required to impose boundary conditions, because in the sense of Hadamard a problem is well-posed if the solution depends in a continuous manner on the boundary condition, even if it is a steady-state problem, [72].

For an elliptic system of equations the number of required boundary conditions is half the order of the system, so a second-order elliptic equation only requires one boundary condition along the computational boundary. The Poisson equation, most of the times uses imposed values or derivatives on the boundary domain. The Dirichlet BC is when a fixed value is imposed on boundary, as  $\phi = \phi_b$ , where  $\phi_b$  is the transportable value on the boundary. The Neumann BC occurs when the flux through the boundaries is known, like  $\nabla\phi = \nabla\phi_b$ .

## 2.2 Discretized Schemes

An important characteristic of the elliptic operators is that from a perturbation inside the domain propagates over all directions, this feature can be proved through the mathematical nature of a PDE and its characteristics lines, see Hirsch, [72]. As consequence, it must be always ensured that the discretizations contains information from all directions and consequently centred discretization is used.

### 2.2.1 Polynomial Reconstruction in 1D space

To obtain the solution and its gradients at the face is necessary to perform a reconstruction of the unknown primitive variable. The polynomial reconstruction is achieved by a  $p^{th}$  order expansion of the

Taylor series, given by:

$$\phi_f^R(x) = \phi_f + (x - x_f) \frac{\partial \phi_f}{\partial x} + \frac{(x - x_f)^2}{2!} \frac{\partial^2 \phi_f}{\partial x^2} + \cdots + \frac{(x - x_f)^p}{p!} \frac{\partial^p \phi_f}{\partial x^p}, \quad (2.5)$$

where the subscript  $f$  refers that the reconstruction is made at the face  $f$  centroid.

Since the problem is in the 1D space, the problem of the reconstruction becomes direct because there is an equal number of unknowns and equations, so it is possible to obtain the flux through the faces from the FDM.

According to Moin, [73], there is an expression that allows to determine the coefficients of the finite difference equation, which is given by:

$$\nabla \phi_f + \sum_{j=1}^{p+1} a_j \phi_j = \mathcal{O}(h^{p+1}), \quad (2.6)$$

where  $a_j$  are the coefficients of a linear combination of a different Taylor series in order to maximize the order of the resultant scheme. From the expression (2.6) it is possible obtain the following east face flux reconstruction for a uniform space grid  $\Delta x$ , for second, forth, sixth and eighth order of accuracy, respectively:

$$\nabla \phi_e = \frac{\phi_{i+1} - \phi_i}{\Delta x} \quad (2.7)$$

$$\nabla \phi_e = \frac{-\phi_{i+2} + 27\phi_{i+1} - 27\phi_i + \phi_{i-1}}{24\Delta x} \quad (2.8)$$

$$\nabla \phi_e = \frac{9\phi_{i+3} - 125\phi_{i+2} + 2250\phi_{i+1} - 2250\phi_i + 125\phi_{i-1} - 9\phi_{i-2}}{1920\Delta x} \quad (2.9)$$

$$\nabla \phi_e = \frac{-75\phi_{i+4} + 1029\phi_{i+3} - 8575\phi_{i+2} + 128625\phi_{i+1} - 128625\phi_i + 8575\phi_{i-1} - 1029\phi_{i-2} + 75\phi_{i-3}}{107520\Delta x}. \quad (2.10)$$

In the case of west face, the flux is obtained through an analogous process, figure 2.1 helps the reader to understand the location of each computational point  $i$ . The proposed schemes for 1D space would be called of Face Centred Scheme (FCS) and to indicate the order of accuracy will be presented in the following form FCS( $p+1$ ), where  $p+1$  is the order of accuracy.

For the faces near the computational boundary, the FD equation ceases to be centred and expand in the opposite direction of the boundary until the minimum number of points defined by the method's order is achieved.

## 2.2.2 Source Term

A high-order Gauss Quadrature (GQ) for the integral of the source term of the Poisson equation,  $\varphi_{\phi i}$ , is required for the verification of the numerical schemes. For 1D space the integral of a generic function  $f$

takes the following form:

$$I(f) = \int_{x_1}^{x_2} f(x) dx, \quad (2.11)$$

and in the discrete form the expression (2.11) is given by:

$$I(f) = \sum_{i \in \mathcal{G}}^{n_G} w_{G_i} f(x_i), \quad (2.12)$$

where the  $\mathcal{G}$  is the set of  $n_G$  Gauss points,  $x_i$  is the coordinates of the Gauss point (GP)  $i$  and  $w_{G_i}$  is the Gauss weight (GW) for the respective coordinate.

In the GQ, the points are symmetric to the centre of integration interval and with  $n$  points it is possible integrates exactly a  $(2n - 1)$  polynomial. The GPs are tabulated for natural coordinates in the interval  $[-1; 1]$ , [74].

This requires a coordinate transformation, which according with Reddy, [75], is given by:

$$x(\zeta) = x_1 + \frac{x_2 - x_1}{2} (1 + \zeta), \quad (2.13)$$

where  $\zeta$  is the natural coordinate, and the final expression of the integral is:

$$I(f) = \int_{x_1}^{x_2} f(x(\zeta)) dx = \Delta x \sum_{i \in \mathcal{G}} w_{G_i} f(x(\zeta_i)), \quad (2.14)$$

where  $\Delta x = x_2 - x_1$ . Table 2.1 presents the GP coordinates and the respective weights for an integration through a line segment between -1 and 1.

Table 2.1: Gauss points coordinates for an integration interval  $[-1; 1]$  with the respective Gauss weights for the different orders accurate from Pina, [76] and Weisstein [77].

$n_G$	$p$	$\zeta_i$	$w_{G_i}$
1	1	0	2
2	3	$\pm\sqrt{\frac{1}{3}}$	1
3	5	0	$\frac{8}{9}$
		$\pm\sqrt{\frac{3}{5}}$	$\frac{5}{9}$
4	7	$\pm\sqrt{\frac{525-70\sqrt{30}}{35}}$	$\frac{18+\sqrt{30}}{36}$
		$\pm\sqrt{\frac{525+70\sqrt{30}}{35}}$	$\frac{18-\sqrt{30}}{36}$

Finally, the expression for the source term integration is given by:

$$\int_{CV} \varphi_\phi dV = \Delta x \sum_{i \in \mathcal{G}} w_{G_i} \varphi_\phi(x_i), \quad (2.15)$$

where  $\Delta x$  is equal to the cell volume in 1D space and the final expression for the solution of the Poisson problem is given by:

$$\nabla \phi_e - \nabla \phi_w = \Delta x \sum_{i \in \mathcal{G}(P)} w_{G_i} \varphi_\phi(x_i), \quad (2.16)$$

where the  $\mathcal{G}(P)$  is the set of Gauss points of cell  $P$ .

## 2.3 Error Norms

The errors that can be found in a numerical solution, according with Reddy, [75], appears due three basic sources: discretization, truncation or finite arithmetic, and approximation errors. When it is applied a discretization to the governing equations, will result in a discrete problem and the resultant system of algebraic equations is solved in order to obtain the numerical solution at each node of the computational domain, this will lead to the discretization error, see Jasak, [78]. Every numerical solution requires the calculation of auxiliary quantities, most of them without an exact representation, due the limitations of the number of digits that the computer is able to store in memory originating the truncation error. The approximation errors appear when the mathematical model is an approximation from the physical one, which is beyond the scope of this Thesis.

Because of these sources of error, it is important to study the quality of the numerical solution obtained with the proposed schemes. It turns necessary to have a tool capable of measuring the scheme's accuracy for a generic problem and the error norms are an example of such tool. An error norm must be capable to evaluate the absolute error magnitude and its distribution over the domain, see Zienkiewicz and Taylor, [79].

There are several ways to measure the error and it becomes important to define the numerical error as the difference between the numerical and the exact solution:

$$e_i = \phi_i - \phi_i^A, \quad (2.17)$$

where  $\phi_i$  is the numerical solution and  $\phi_i^A$  is the exact solution both at cell  $i$ . The error obtained from equation (2.17) is a field distributed along the domain and for most purposes is required a single number that evaluates the global size of error, being that measure called norm, which is a non-negative real number [75, 80].

$$\|e\|_1 = \frac{\sum_{i \in \mathcal{C}} |e_i| V_i}{\sum_{i \in \mathcal{C}} V_i}, \quad (2.18)$$

$$\|e\|_2 = \sqrt{\frac{\sum_{i \in \mathcal{C}} |e_i|^2 V_i^2}{\sum_{i \in \mathcal{C}} V_i^2}}, \quad (2.19)$$

$$\|e\|_\infty = \max_{i \in \mathcal{C}} |e_i|, \quad (2.20)$$

where  $\mathcal{C}$  is the set of all cells that domain is divided.

Equations (2.18) to (2.20) represents the most used norms to measure the error, which are: the error norm-1 or mean error, the error norm-2 or Euclidean norm, and the error norm- $\infty$  or maximum error, respectively, where the  $\mathcal{C}$  is the set of cells that the domain is divided. The Author will only use the error norm-1, which can be defined as a global error evaluation and the norm- $\infty$ , that is a local appreciation

of the error.

### 2.3.1 Convergence Order

It is assumed that the grids used in this work are fine enough to provide and guarantee monotone convergence, see Ferziger and Peric, [81]. The rate of convergence between two spaced grids,  $h_1$  and  $h_2$ , indicates the accuracy order of the scheme and it is given by:

$$\mathcal{O}_n = \frac{\log_{10} \|e\|_{n_1} - \log_{10} \|e\|_{n_2}}{\log_{10} h_1 - \log_{10} h_2}, \quad (2.21)$$

where the subscript  $n$  is the n-norm used to compute the rate of convergence. According with Ferziger and Peric, [81], the scheme's order of accuracy should be obtained from the results of three consecutive grids, which means that it is required to obtain identical results for two consecutive convergence orders, that mathematically is given by:

$$\frac{\log_{10} \|e\|_{n_1} - \log_{10} \|e\|_{n_2}}{\log_{10} h_1 - \log_{10} h_2} = \frac{\log_{10} \|e\|_{n_2} - \log_{10} \|e\|_{n_3}}{\log_{10} h_2 - \log_{10} h_3}, \quad (2.22)$$

where  $h_1$ ,  $h_2$  and  $h_3$  are three consecutive spaced grids, and it is said that the asymptotic convergence is achieved, because the convergence order has become constantly.

## 2.4 Results

In this section several numerical tests are presented to investigate the accuracy with respect to the approach of reconstruction performed, namely the explicit and implicit face flux reconstruction. This will be followed by a study of the deferred correction approach.

In this study two grid types will be used, an uniform grid and a non-uniform one with a domain of  $[0, 1]$ . For this verification exercise the following analytical solution of the Poisson equation will be used:

$$\phi(x) = \exp\left(-\frac{(x - 0.5)^2}{0.0175}\right), \quad (2.23)$$

and the source term  $\varphi_\phi$  will be equal to  $-\nabla^2 \phi$ , which is given by:

$$\varphi_\phi(x) = \left(\frac{2(x - 0.5)^2}{0.0175} - 1\right) \frac{2}{0.0175} \exp\left(-\frac{(x - 0.5)^2}{0.0175}\right). \quad (2.24)$$

As boundary conditions, it will be used a Dirichlet BC for both boundaries.

### 2.4.1 Explicit Face Flux Reconstruction

In this type of face reconstruction approach, it is assumed that the  $\phi_i$  is known for every cell and consequently, the unknown will be the  $\nabla^2\phi$ , which is given by the following expression:

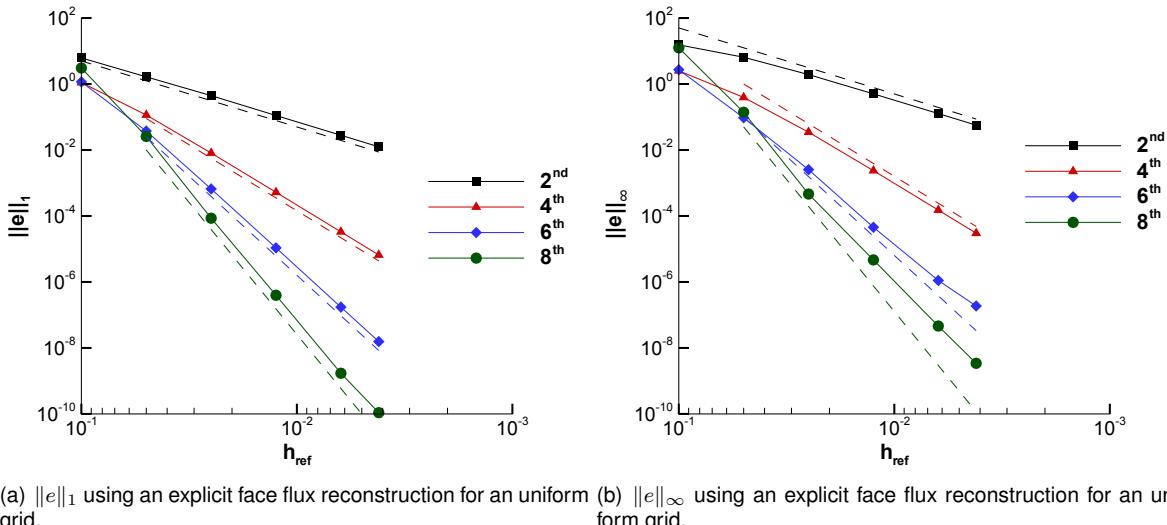
$$\nabla^2\phi_i = \frac{\nabla\phi_e - \nabla\phi_w}{V_i}, \quad (2.25)$$

where the  $V_i$  is the volume at cell  $i$ , which in this case is given by the cell length,  $h_i$ . To obtain the final result, equation (2.25) is applied at each cell and it will be done a comparison between numerical value,  $\nabla^2\phi_i$ , and the analytical one,  $\nabla^2\phi_i^A$ , that it is computed using Gauss quadrature:

$$\nabla^2\phi_i^A = \sum_{j \in \mathcal{G}(i)} w_{G_j} \varphi_{\phi_i}^A(x_j). \quad (2.26)$$

The expression to calculate the error needs to be changed and in expression (2.17) where appears  $\phi_i$  and  $\phi_i^A$  should be replaced by  $\nabla^2\phi_i$  and  $\nabla^2\phi_i^A$  respectively. For the error norms the expressions (2.18) and (2.20) for the mean and maximum error respectively, were used. For the convergence order the expression (2.21) was used.

#### Uniform Grid



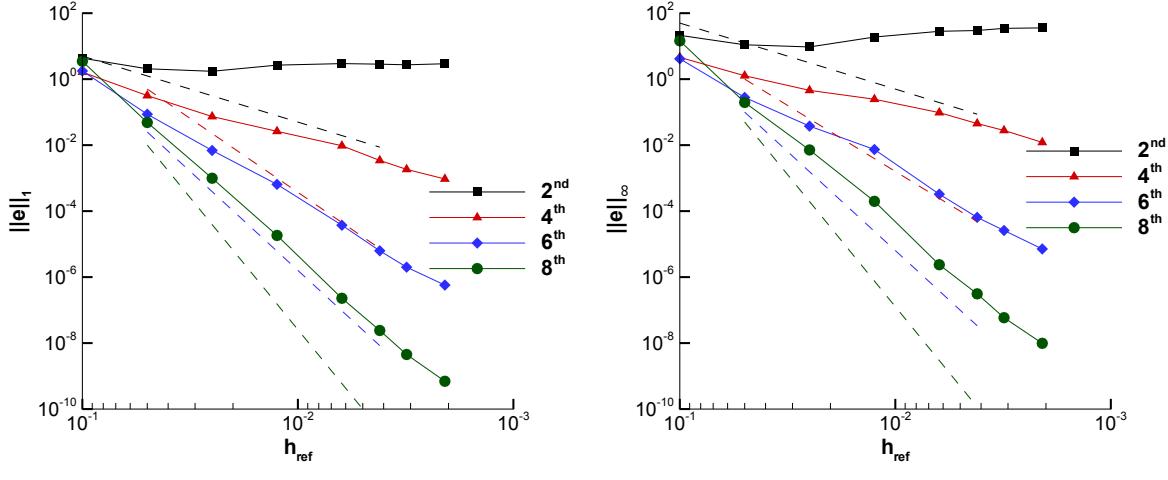
(a)  $\|e\|_1$  using an explicit face flux reconstruction for an uniform grid. (b)  $\|e\|_\infty$  using an explicit face flux reconstruction for an uniform grid.

Figure 2.2: Convergence curves for uniform grid of  $\|e\|_1$  and  $\|e\|_\infty$  for all schemes using an explicit face flux reconstruction, where the dotted line represents the theoretical slope for each scheme.

Figure 2.2 shows the convergence curves of the mean and maximum error applied to a 1D uniform grid. It is possible to observe from figure 2.2(a) that the convergence curves of the mean error for each scheme are parallel to the corresponding dotted curve, which indicates the ideal behaviour of each scheme. This allows to conclude that the proposed schemes achieved the theoretical accuracy order. For the maximum error, figure 2.2(b), the curves are close to parallel to the dotted lines, which indicates

that the theoretical value for the accuracy order is almost achieved.

### Non-uniform Grid



(a)  $\|e\|_1$  using an explicit face flux reconstruction for a non-uniform grid. (b)  $\|e\|_\infty$  using an explicit face flux reconstruction for a non-uniform grid.

Figure 2.3: Convergence curves for a non-uniform grid of  $\|e\|_1$  and  $\|e\|_\infty$  for all schemes using an explicit face flux reconstruction, where the dotted line represents the theoretical slope for each scheme.

It is shown in figure 2.3 the convergence curves for mean and maximum error applied to a non-uniform grid. Both figures point out that the theoretical slope is never achieved, which indicates that an accuracy issue is presented in the method. From figure 2.3(a) it is possible to extrapolate the proposed schemes' behaviour of  $\mathcal{O}_1 - 2$  when using an explicit flux reconstruction in a non-uniform grid.

These results indicate that this form of reconstruction cannot be applied with non-uniform grids, this result can be extrapolated to 2D case when is used unstructured grids. It becomes necessary to use a different approach to compute the flux reconstruction in order to verify the accuracy order of the schemes.

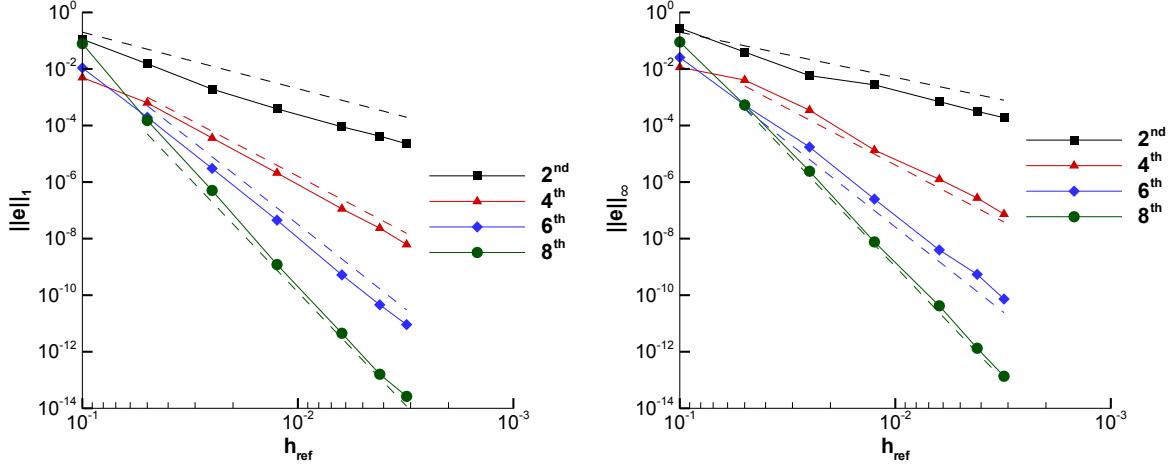
### 2.4.2 Implicit Face Flux Reconstruction

In this case the transportable variable is the unknown of the problem and consequently the computed flux reconstruction is given as function to the transportable variable. Equation (2.16) is valid for a generic CV and when it is applied the face flux, which is given by equations (2.7) to (2.10), depending of the scheme's order, results in an equation with a determined number of unknowns that represents the  $\phi$  values. Applying equation (2.16) to all CVs, it results in a linear system of equations with the form of  $\mathbf{A}\phi = \mathbf{b}$ , where  $\mathbf{A}$  is a square matrix with  $(n_{cells})^2$  entries, which is known as Global Matrix, and the  $\mathbf{b}$  is a column vector with  $n_{cells}$  values, known as right-hand side (RHS), where  $n_{cells}$  is the number of cells into domain is divided. Every line of  $\mathbf{A}$  corresponds to a cell of the domain and each column corresponds to the contribution of the neighbouring cells for the calculation of the  $\phi$  value in that cell,

since the number of equations and unknowns are the same, this will result in a direct problem. The solution of the Poisson problem with an implicit face flux reconstruction will be given by:

$$\Phi = \mathbf{A}^{-1}\mathbf{b}, \quad (2.27)$$

where  $\mathbf{b}$  takes into account the source term and the contributions of the boundary faces of the problem.



(a)  $\|e\|_1$  using an implicit face flux reconstruction for a non-uniform grid. (b)  $\|e\|_\infty$  using an implicit face flux reconstruction for a non-uniform grid.

Figure 2.4: Convergence curves of a non-uniform grid for  $\|e\|_1$  and  $\|e\|_\infty$  for all schemes using an implicit face flux reconstruction, where the dotted line represents the theoretical slope of each scheme.

Figure 2.4 shows the convergence curves of the mean and maximum error of the proposed schemes applied to a non-uniform grid. When is used an implicit approach, both sub-figures shows that the theoretical convergence order is achieved for all schemes, in opposite to what has happened with the explicit face reconstruction approach with the same grid type, see figure 2.3.

For the uniform grid, the results are not shown since they have the same behaviour as the ones obtained with an explicit approach.

The results obtained with the implicit approach fix the problems that have appeared when an explicit reconstruction approach is used and justify the requirement of to use an implicit flux reconstruction approach in order to obtain the theoretical convergence order for the proposed schemes.

### 2.4.3 Deferred Correction (DC) Approach

One way to obtain a high-order solution is through deferred correction approach, which was firstly introduced in 1974 by Khosla and Rubin, [82], where its basis consists in a low-order (LO) implicit approach combined with a high-order (HO) explicit approach. This can be mathematically describe by:

$$\mathbf{A}_{LO}\Phi^{n+1} = \mathbf{b}_{HO} + (\mathbf{A}_{LO} - \mathbf{A}_{HO})\Phi^n, \quad (2.28)$$

where the superscript  $n + 1$  and  $n$  is the current and previous iteration, respectively, the subscript  $LO$  and  $HO$  refers to the discretization method used. This approach should cause no problems in the iterative solution procedure and when a converge solution is achieved, it means that  $\Phi^{n+1} \approx \Phi^n$ , and the  $LO$  term will be cancelled since  $\mathbf{A}_{LO}\Phi^{n+1} \approx \mathbf{A}_{LO}\Phi^n$  and finally, the expression (2.28) will be like  $\mathbf{A}_{HO}\Phi^n = \mathbf{b}_{HO}$ . It is shown that the solution is achieved by a higher order approach, see Ferziger and Peric, [81], and the system to be solved can be written in the following form:

$$\Phi^{n+1} = \mathbf{A}_{LO}^{-1} (\mathbf{b}_{HO} + (\mathbf{A}_{LO} - \mathbf{A}_{HO}) \Phi^n), \quad (2.29)$$

for the iterative process the GMRES solver of Matlab was used.

With this approach it is possible to solve non-orthogonality issues, [81], pressure correction when dealing with an incompressible Navier-Stokes problem, see e.g. Ramírez et al., [27], with a fourth-order scheme. This approach avoids the implementation of an implicit high-order method.

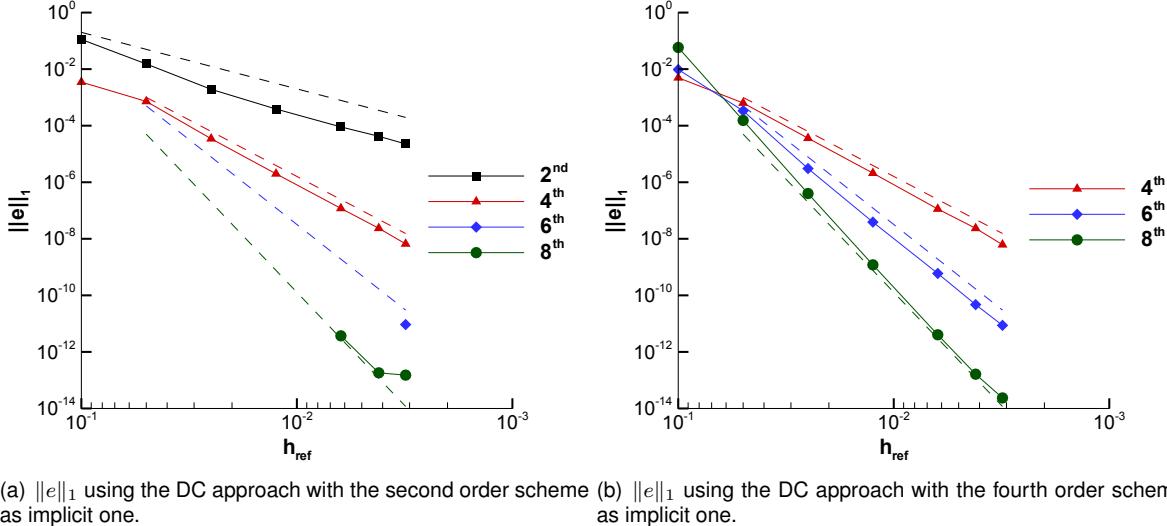


Figure 2.5: Convergence curves for a non-uniform grid of  $\|e\|_1$  for a  $2^{nd}$  and  $4^{th}$  order implicit approach for the remaining schemes, where the dotted line represents the theoretical slope of the scheme.

Figure 2.5 shows the convergence curves of the mean error of the DC approach when is used a second-order (figure 2.5(a)) and fourth-order (figure 2.5(b)) implicit approach and the remaining ones with an explicit approach.

From figure 2.5(a) it is possible to conclude that with a second-order implicit approach, it is only possible to obtain a  $4^{th}$  order solution, being the convergence order equal to the theoretical one, for the remaining orders of accuracy the results are inconsistent. Besides the solver is only capable to achieve a converged solution with the finest grids. In the figure 2.5(b), it is shown that with a  $4^{th}$  order implicit approach, it is possible to obtain a converged solution for all other orders, being the accuracy order the expected one.

Although it is not shown the results for the maximum error and for uniform grids, the results for these cases are the expected ones, not bringing another relevant result than the ones already mentioned

above.

## 2.5 Conclusions

The performed tests allows to verify that an explicit flux reconstruction approach cannot be used to validate the proposed schemes, since it only has the expected behaviour for uniform grids. Tests have been performed in an implicit flux reconstruction approach and reveals that it can fix those problems, thus allowing the validation of the proposed high-order schemes, which shows an excellent behaviour of the convergence curves indicating that the theoretical convergence is achieved for all numerical schemes and both grid types. Finally, it was tested a deferred correction approach, which results indicate that it is only possible to obtain a fine behaviour of the very high-order ( $6^{th}$  and  $8^{th}$  order) schemes if an explicit  $4^{th}$  order approach is used.

Therefore, since in order to obtain very high-order accurate schemes with non-uniform grids it is mandatory to have an implicit approach at least of  $4^{th}$  order and once this implicit flux reconstruction approach is implemented, the extension for  $6^{th}$  and  $8^{th}$  order accurate is quite straightforward. Due to these reasons only the implicit flux reconstruction will be considered for the unstructured 2D grids avoiding the implementation of an explicit approach for these grids.

Nevertheless the reader should consider the DC approach tests like a verification of conditions that allow the DC approach utilization. If a fourth-order scheme is wanted, a second-order implicit scheme needs to be used and for a eighth-order scheme is mandatory at least a fourth-order implicit one. This will lead to a major conclusion that suggest an implicit scheme at least half the order of the explicit one, which defines the method's order, is required. These results confirm the authors that have used this approach with a second-order implicit code in their own DC method to obtain fourth-order accurate scheme, such as Ramírez et al., [27], for the incompressible Navier-Stokes equations.



# **Chapter 3**

# **High-Order Schemes for Unstructured Grids**

This chapter begins with a description of the unstructured grids and their applicability to the industry applications. It will be exposed the grid quality parameters, namely the non-orthogonality angle and the volume ratio. The FVM for a 2D domain is explained and a detailed description of the proposed WLS scheme for diffusive term discretization of a transport equation will be made. Details are provided on the stencil selection and how the global coefficient matrix is constructed according to the boundary condition.

## **3.1 Unstructured Grids**

The main goal of this Thesis is the development of numerical schemes for diffusive term of a transport equation to be applied in unstructured grids, being important a detailed explanation of what is an unstructured grid. Although it has already been mentioned in chapter 1, it is important to emphasize that the mesh generator is the one of the most important requirements of a CFD code and the process of generating a mesh with good quality parameters is a research field independent of the numerical schemes used. Nowadays, the unstructured grids are the ones that are most used in CFD simulations because of they adaptability to complex geometries.

Every time that a spatial discretization is performed into a domain, like the one performed in chapter 2, it is required to have a grid formed by several CVs, which have the dimensional space of the domain. These CVs should not intersect with each other and fill the entire domain, figure 3.1 illustrates the division of a typical continuous domain in non-overlapping volumes (cells), where it is shown a cell and its neighbours.

One cell,  $P$ , can have an arbitrary shape being delimited by a set of faces, which can be defined by a line segments through two consecutive grid vertices, and each vertex is defined by its spacial coordinates. It is in the geometric center of each cell where is stored the information about the transportable variable, these points are known as the computational points.

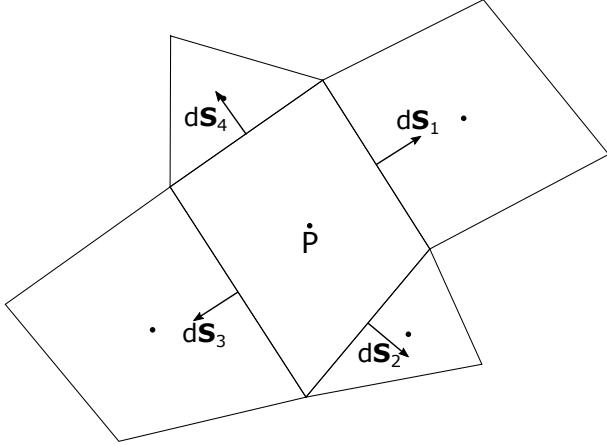


Figure 3.1: Cell-centered FV discretization over the CV  $P$  ( $\bullet$  - *Cell Center*).

In a grid there are two types of faces: the interior and boundary ones. The interior faces are the ones that separates two cells and thus it belongs to these cells, this type of cells generally does not require any special treatment for the discretization. The boundary faces are the ones that are coincident with the boundary domain and can only belong to one cell, in opposite to the interior ones this type of faces requires a special treatment, since they store the information about the BC applied to the problem. Interior faces have two neighbour cells and boundary faces have only one neighbour cell. The face normal vector,  $\mathbf{n}$ , points outwards from the cell with lower index for interior faces and points outwards from the domain for the boundary ones.

The SOL code is capable of computing some grid properties, which are fundamental for the FV discretization such as: face area,  $S$ , the normal vector,  $\mathbf{n}$ , the cell volume,  $V_P$ , and the cell centroid,  $P$ . Since, it is used unstructured grids, it is required to determine the reference length, which is a size indicator of each cell, that can be obtained from the expression of the hydraulic diameter,  $h$ , that for a 2D cell case is given by:

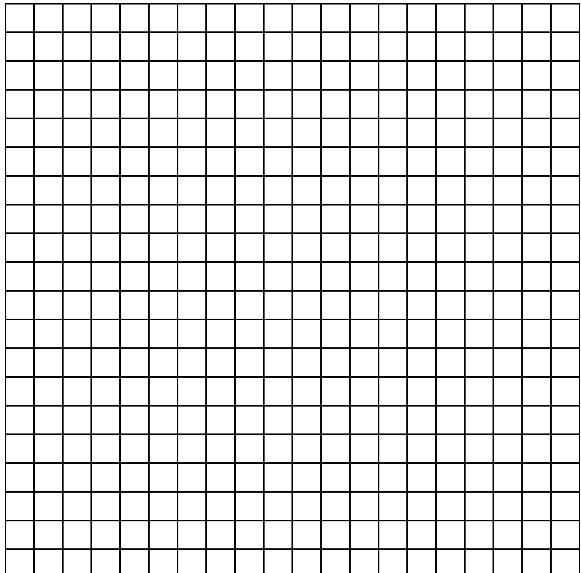
$$h_P = \frac{4V_P}{\sum_{f \in \mathcal{F}(P)} \|\mathbf{S}_f\|}, \quad (3.1)$$

where  $\mathcal{F}(P)$  is the set of faces that delimits the cell  $P$ . From the equation (3.1) it is possible to obtain the mean reference length of the grid, which is given by:

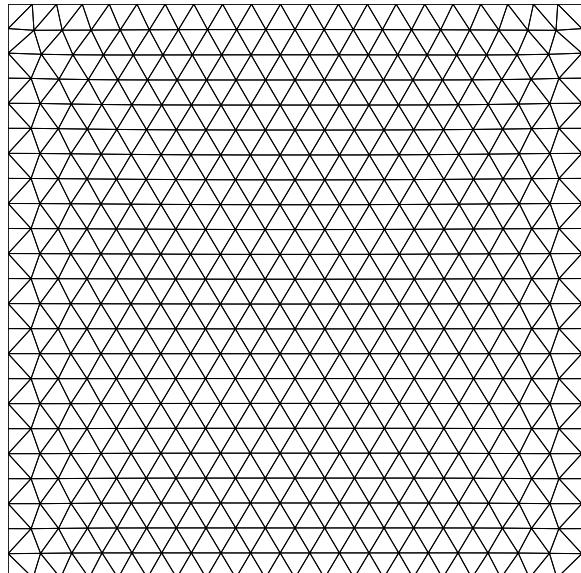
$$h_{ref} = \frac{\sum_{p \in \mathcal{C}} h_p}{n_{cells}} \quad (3.2)$$

where  $\mathcal{C}$  is the set of cells that contains the complete domain. This is frequently used in the unstructured grids framework as the cell reference length,  $h_{ref}$ , being very important for the study of the numerical error decay with successive grid refinements.

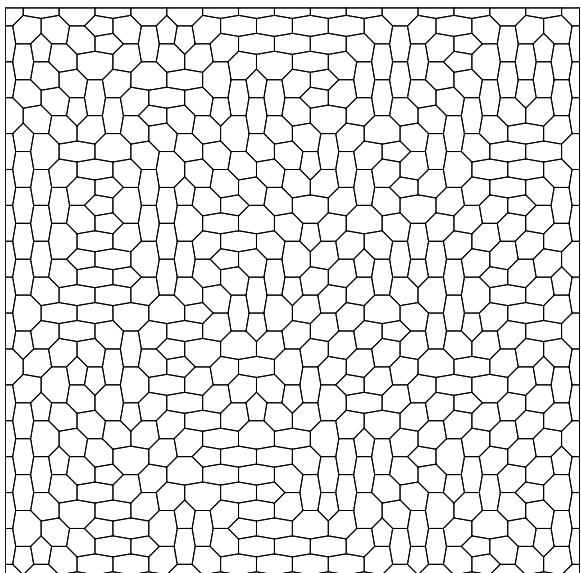
The majority of the 2D CFD codes use quadrilateral and triangular cells, because they are the simplest ones to be generated. Figure 3.2 shows the different grids types used in this Thesis. The quadrilateral cells have four vertices and are considered to be the most common ones. Figure 3.2(a) shows



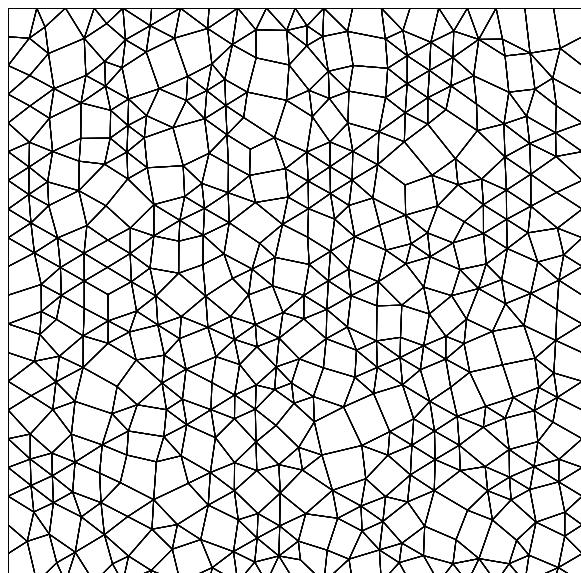
(a) Cartesian Grid with 400 cells.



(b) Triangular Grid with 899 cells.



(c) Polyhedral Grid with 545 cells.



(d) Hybrid Grid with 726 cells.

Figure 3.2: Examples of a Cartesian, triangular, polyhedral and hybrid grids, which were used by the Author in this Thesis.

an example of this grid type with 400 cells. Since the scope of this Thesis is the unstructured grids, the Cartesian one was only used for code verification. The triangular grid is formed by triangular cells, which have three vertices, this grid was generated with the DISTMESH code, [83], that is an open-source software to generate triangular grids with a constant grid size. This grid can be considered as a regular one, since almost cells have the same volume. Figure 3.2(b) shows an example of this type of grid, where it is a regular grid for interior cells, and only the cells that are influenced by the boundaries are irregular, the coarse grid has 211 cells and the finest one has 91772 cells.

The polyhedral grid is formed by irregular pentagonal and hexagonal cells with five and six vertices, respectively. A possible process to generate a polyhedral grid is through a triangular one with the dual mesh concept, while use the triangle centroids as the vertices of the polyhedral grid. The coarser grid contains 145 cells and the finest one has 131585 cells, where an example with 545 cells is shown in figure 3.2(c). The hybrid grid is built from a triangular one, where 20% of the cells are randomly selected to merge with a random face neighbour, this results in a quadrilateral cell and afterwards an elliptical softener is applied. It was used six grid refinements, where the coarser grid has 167 cells and the finest one has 200905 cells, an example of this grid with 726 cells is shown in figure 3.2(d) and it was used as a robustness test for the proposed numerical schemes.

### 3.1.1 Grid Quality

The grid quality is measured by several parameters, through them it is possible to assess the quality of the grid used in the simulation. This type of metric was described by F. Juretic and D Gosman, [84], and the parameters analysed are the skewness, non-orthogonality angle and the volume ratio.

The skewness is defined as the distance between the face centroid,  $f$ , and the intersection point of the line segment through the two cells centroids and that face,  $P_f$ , in figure 3.3 shows an example of that case where the red line represents that distance. This parameter does not affect the discretization of the elliptic operators and consequently will be neglected.

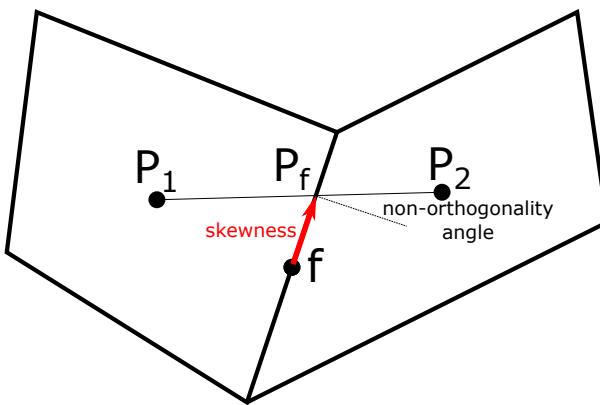


Figure 3.3: Example of the skewness and non-orthogonality angle grid quality parameters, where  $\bullet$  are the centroids of the cells  $P_1$  and  $P_2$  and face  $f$ .

The non-orthogonality angle,  $\alpha$ , is the one between the normal vector of the face and the line segment through the two cells centroid, in figure 3.3 it is shown an example of this angle. Several twist grids based

on the Cartesian one, where a constant non-orthogonality angle is imposed to the vertical faces were built, in order to test the proposed diffusive schemes. Five different non-orthogonality angles,  $\alpha$ , were used:  $0^\circ$ ,  $10^\circ$ ,  $20^\circ$ ,  $30^\circ$  and  $40^\circ$ , figure 3.4 shows two examples of this grid type, for a non-orthogonality angle of  $10^\circ$ , fig. 3.4(a) and  $40^\circ$ , fig. 3.4(b), respectively.

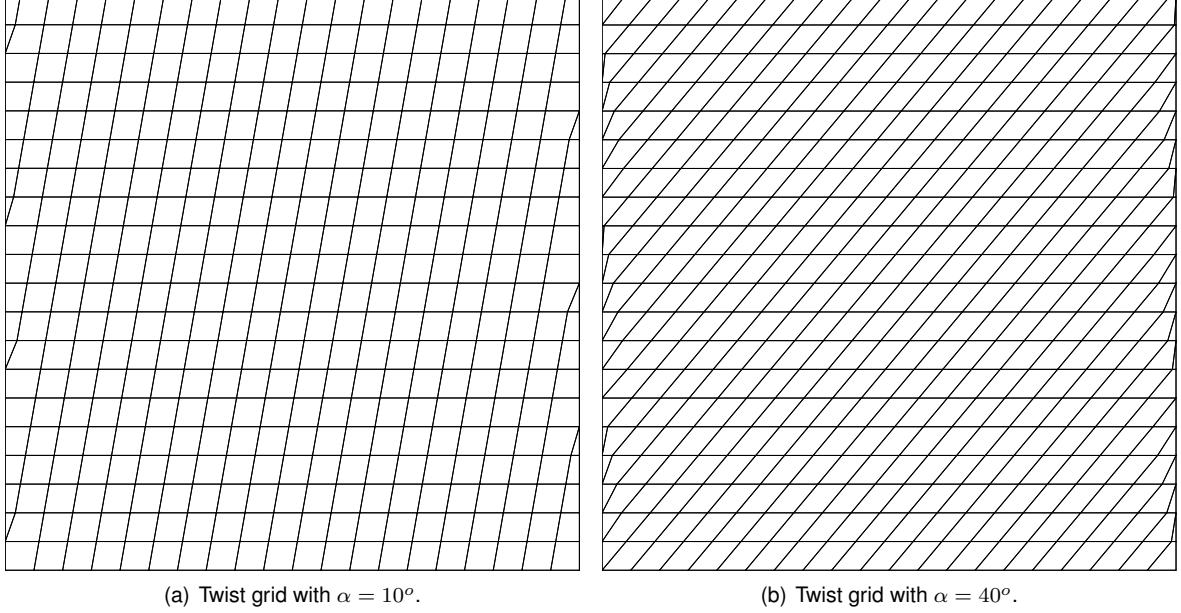


Figure 3.4: Twist grid, with two different non-orthogonality angles:  $10^\circ$  and  $40^\circ$ .

The volume ratio parameter measures the ratio between maximum and minimum cell volume of a domain and can be calculated by:

$$\frac{V_{\max}}{V_{\min}} = \frac{(h_{ref} + 2\Gamma)^2}{(h_{ref} - 2\Gamma)^2} \quad (3.3)$$

where  $\Gamma$  is the displacement applied to the grid given by  $\Gamma = \gamma h_{ref}$ , where the  $\gamma$  is the percentage of the reference length that the face is displaced. This grid type, known as perturbed grid, is also based on the Cartesian one where every face group is moved a certain percentage of the grid parameter,  $\gamma$ , and a face group is defined as a set of faces which have in common one coordinate. It was used four different percentages, 0, 10, 20 and 30%, and from equation (3.3) it is possible to conclude that in the worst case scenario the volume ratio can be equal to 2.25, 5 and 16 for 0, 10, 20 and 30%, respectively. Figure 3.5 shows two examples of this grid type, one with a displacement of 10% and another one with a displacement of 30%.

## 3.2 Least Squares Approach

This section describes the proposed WLS scheme for diffusive term of a transport equation. Details are provided on the stencil selection and how the global coefficient matrix is constructed according to the boundary condition.

Starting from the Poisson equation, given by equation (2.1), and the discretization performed to it for a 1D domain, which is explained in chapter 2 and results in the discretized Poisson equation, given by

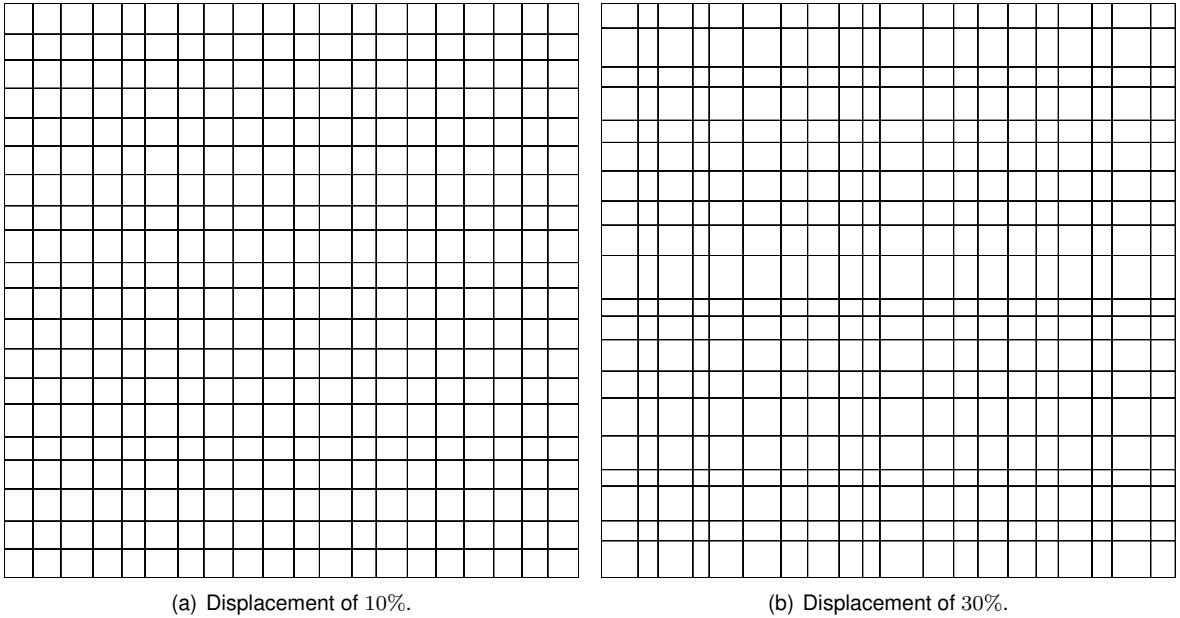


Figure 3.5: Perturbed grids with two different imposed displacements applied to a Cartesian grid with 400 cells.

equation (2.4). This discretization can be extended for a 2D domain with a FVM cell centered approach, which requires the calculation of the face gradient,  $\nabla\phi_f$ , for every face and will result in:

$$\sum_{f \in \mathcal{F}(P)} \sum_{g \in \mathcal{G}(f)} \nabla\phi_g w_{G_g} \cdot \mathbf{S}_f = \int_{CV} \varphi_\phi dV, \quad (3.4)$$

where  $\mathcal{F}(P)$  is the set of faces of cell  $P$ ,  $\mathcal{G}(f)$  is the set of Gauss points of the face  $f$ ,  $\mathbf{S}_f$  is the face normal vector and  $w_G$  is the weight of Gauss-Legendre Quadrature. The integration of the source term for a 2D domain will be further explained.

### 3.2.1 Polynomial Reconstructions

To obtain the solution and its gradients at the integration points, a reconstruction of the unknown primitive variable is performed to form a polynomial expansion at the control volumes face to the desired accuracy.

The polynomial reconstruction is performed taking into account the order of the numerical scheme and this is achieved by an expansion of a  $p^{th}$  complete polynomial, centered on the face where the reconstruction is made:

$$\begin{aligned} \phi_f^R(x, y) = & C_1 + C_2(x - x_f) + C_3(y - y_f) + \\ & + C_4(x - x_f)^2 + C_5(x - x_f)(y - y_f) + C_6(y - y_f)^2 + \dots \end{aligned} \quad (3.5)$$

Expression (3.5) can be written in a more compact form, a vectorial one, as:

$$\phi_f^R(\mathbf{x}) = \mathbf{d}_f(\mathbf{x}) \mathbf{c}_f. \quad (3.6)$$

where the subscript  $f$  refers that the reconstruction is made at the face  $f$  centroid and  $\mathbf{d}_f(\mathbf{x}) = [1, (x - x_f), (y - y_f), (x - x_f)^2, (x - x_f)(y - y_f), (y - y_f)^2, \dots]$ ,  $\mathbf{x}_f = (x_f, y_f)$  is the face centroid coordinates vector,  $\mathbf{x} = (x, y)$  is the coordinates vector of a point used for the reconstruction and  $\mathbf{c}_f = [C_1, C_2, C_3, C_4, C_5, C_6, \dots]^T$  are the reconstruction constants. The above expressions are valid for a  $p^{th}$  order polynomial and a reconstruction will be made for every face of the domain. This reconstruction will be called of Face Least Squares (FLS) approach. The number of terms of the expansion is given by  $(p + 1)(p + 2)/2$ , according with Chassaing et al., [26]. Table 3.1 lists the number of terms of the expansion for each polynomial used in this work, up to  $7^{th}$  polynomial order.

Table 3.1: Number of terms of the Taylor expansion required for a  $p^{th}$  order polynomial at two dimensional cases.

$p$	1	3	5	7
Terms	3	10	21	36

The order of accuracy of the numerical scheme is  $p + 1$  and consequently the linear reconstruction will be  $2^{nd}$  order accurate, the cubic reconstruction will be  $4^{th}$  order accurate, the  $5^{th}$  polynomial will have  $6^{th}$  order accurate and finally the  $7^{th}$  polynomial will be  $8^{th}$  order accurate. The numerical schemes will be called of FLS( $p+1$ ) according to the global order of the implemented method.

### 3.2.2 General Approach

The Weighted Least Squares (WLS) method is a technique used for overdetermined problems, where there are more independent equations than unknowns and, consequently the problem does not has an exact solution.

Equation (3.6) results in a system of linear equations, which the form as:

$$\mathbf{D}_f \mathbf{c}_f = \boldsymbol{\phi}_s, \quad (3.7)$$

where  $\mathbf{D}_f$  is a combination of  $\mathbf{d}_f(\mathbf{x})$  for every point of the reconstruction resulting in a matrix with  $n_s \times n_{coefs}$  entries. The  $\mathbf{c}_f$  is a column vector with  $n_{coefs}$  entries and  $\boldsymbol{\phi}_s$  is a column vector with  $n_s$  entries, being  $n_{coefs}$  is the number of constants of the  $p^{th}$  polynomial (unknowns), listed in Table 3.1, and  $n_s$  the size of the stencil (number of reconstruction points that will be explained in detail in next sections). This system of equations can be written in an extended way as:

$$\begin{bmatrix} 1 & (x_1 - x_f) & (y_1 - y_f) & (x_1 - x_f)^2 & (y_1 - y_f)^2 & \dots \\ 1 & (x_2 - x_f) & (y_2 - y_f) & (x_2 - x_f)^2 & (y_2 - y_f)^2 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \\ 1 & (x_{n_s} - x_f) & (y_{n_s} - y_f) & (x_{n_s} - x_f)^2 & (y_{n_s} - y_f)^2 & \dots \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_{n_{coefs}} \end{bmatrix} = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_{n_s} \end{bmatrix}. \quad (3.8)$$

Since  $n_s > n_{coefs}$ , the problem is overdetermined and the weighted residual,  $r$ , will be not null and so the WLS technique is used in order to minimize the weighted residual of the problem, which is defined

as:

$$\min (\mathbf{r}^T \mathbf{W}_{\text{LS}} f \mathbf{r}) = \min \left( (\boldsymbol{\Phi}_s - \mathbf{D}_f \mathbf{c}_f^T)^T \mathbf{W}_{\text{LS}} f (\boldsymbol{\Phi}_s - \mathbf{D}_f \mathbf{c}_f^T) \right), \quad (3.9)$$

where  $\mathbf{r} = \boldsymbol{\Phi}_s - \mathbf{D}_f \mathbf{c}_f^T$  is the residual of the interpolation and  $\mathbf{W}_{\text{LS}} f = \text{diag}(w_{LS1}, \dots, w_{LSn_s})$  is the weight function of the LS problem that will be further analysed.

The minimum of the weighted residual can be obtained differentiating (3.9) with respect to  $\mathbf{r}^T \mathbf{W}_{\text{LS}} f \mathbf{r}$  and setting equal to zero yields:

$$\mathbf{D}_f^T \mathbf{W}_{\text{LS}} f \mathbf{D}_f \mathbf{c}_f = \mathbf{D}_f^T \mathbf{W}_{\text{LS}} f \boldsymbol{\Phi}_s. \quad (3.10)$$

For clarity the matrix product will be defined as  $\mathbf{D}_{\text{w}_f} = \mathbf{W}_{\text{LS}} f \mathbf{D}_f$  and the equation (3.10) is written as:

$$\mathbf{D}_{\text{w}_f}^T \mathbf{D}_f \mathbf{c}_f = \mathbf{D}_{\text{w}_f}^T \boldsymbol{\Phi}_s. \quad (3.11)$$

Resulting in a matrix with the same entries of matrix  $\mathbf{D}_f$ ,  $(n_s \times n_{coefs})$ , with the vector  $\mathbf{c}_f$  as unknown of the least-squares problem. It is fundamental to ensure that the columns of  $\mathbf{D}$  are linearly independent. The WLS solution for face  $f$  can be expressed as:

$$\mathbf{c}_f = \left( \mathbf{D}_{\text{w}_f}^T \mathbf{D}_f \right)^{-1} \mathbf{D}_{\text{w}_f}^T \boldsymbol{\Phi}_s, \quad (3.12)$$

with the pseudo inverse matrix defined as:

$$\mathbf{P}_f = \left( \mathbf{D}_{\text{w}_f}^T \mathbf{D}_f \right)^{-1} \mathbf{D}_{\text{w}_f}^T, \quad (3.13)$$

and the constants of the polynomial reconstruction are computed by:

$$\mathbf{c}_f = \mathbf{P}_f \boldsymbol{\Phi}_s. \quad (3.14)$$

$$\begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ \vdots \\ C_{n_{coefs}} \end{bmatrix} = \begin{bmatrix} P_{1,1} & P_{1,2} & P_{1,3} & \cdots & P_{1,n_s} \\ P_{2,1} & P_{2,2} & P_{2,3} & \cdots & P_{2,n_s} \\ P_{3,1} & P_{3,2} & P_{3,3} & \cdots & P_{3,n_s} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ P_{n_{coefs},1} & P_{n_{coefs},2} & P_{n_{coefs},3} & \cdots & P_{n_{coefs},n_s} \end{bmatrix} \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \vdots \\ \phi_{n_s} \end{bmatrix}. \quad (3.15)$$

After the  $\mathbf{P}$  matrix has been determined for every face of the domain it is possible to proceed with the construction of a linear system of equations for the Poisson problem because the constants of the reconstruction will only depend of the  $\boldsymbol{\Phi}_s$  values.

Equation (3.4) uses the fluxes through all the faces of the CV given by the divergence of the reconstruction and obtained directly by taking the derivatives with respect to the coordinates directions, resulting in:

$$\nabla \phi_f^R(\mathbf{x}) = (\nabla \mathbf{d}_f(\mathbf{x})) \mathbf{c}_f, \quad (3.16)$$

combining the above equation with equation (3.14) yields:

$$\nabla \phi_f^R(\mathbf{x}) = (\nabla \mathbf{d}_f(\mathbf{x})) (\mathbf{P}_f \boldsymbol{\Phi}_s), \quad (3.17)$$

where  $\nabla \mathbf{d}$  in vector form:

$$\nabla \mathbf{d}_f(x, y) = \begin{bmatrix} 0 & 1 & 0 & 2(x - x_f) & (y - y_f) & 0 & \dots \\ 0 & 0 & 1 & 0 & (x - x_f) & 2(y - y_f) & \dots \end{bmatrix}, \quad (3.18)$$

introducing the matrix  $\mathbf{T}$  as:

$$\mathbf{T}_f(\mathbf{x}) = (\nabla \mathbf{d}_f(\mathbf{x})) \mathbf{P}_f \quad (3.19)$$

with  $(2 \times n_s)$  entries, equation (3.17) can be expressed as  $\nabla \phi_f^R(\mathbf{x}) = \mathbf{T}_f(\mathbf{x}) \boldsymbol{\Phi}_s$  and equation (3.17) appears as:

$$\begin{bmatrix} \frac{\partial \phi_f^R}{\partial x} \\ \frac{\partial \phi_f^R}{\partial y} \end{bmatrix} = \begin{bmatrix} T_{x,1} & T_{x,2} & T_{x,3} & \dots & T_{x,n_s} \\ T_{y,1} & T_{y,2} & T_{y,3} & \dots & T_{y,n_s} \end{bmatrix} \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \end{bmatrix}. \quad (3.20)$$

Equation (3.4) is valid for a generic CV, and applying it over all CV's results in a linear system of equations with the form  $\mathbf{A}\boldsymbol{\phi} = \mathbf{b}$ , where  $\mathbf{A}$  is the Global Matrix with  $(n_{cells} \times n_{cells})$  entries and  $\mathbf{b}$  is the RHS column vector with  $n_{cells}$  entries, where  $n_{cells}$  is the number of cells of the computational grid.

Every line of  $\mathbf{A}$  corresponds to a cell of the domain and each column corresponds to the contribution of that neighbouring cell for the calculation of the values of  $\phi$ . The integration of the fluxes through the faces of the CV in equation (3.4) requires to use a Gauss Legendre Quadrature at least with the order of reconstruction,  $p$ , to maintain the reconstruction accuracy order, see Pina, [76]. Consequently, the left side of expression for the Poisson equation is given by:

$$\sum_{f \in \mathcal{F}(P)} \nabla \phi_f \cdot \mathbf{S}_f = \sum_{f \in \mathcal{F}(P)} \left( \sum_{g \in \mathcal{G}(f)} (w_{Gg} \mathbf{T}_f(\mathbf{x}_g) \cdot \boldsymbol{\Phi}_s) \mathbf{S}_f \right), \quad (3.21)$$

where  $w_G$  is the weight of Gauss-Legendre Quadrature and  $\mathbf{x}_g$  are the coordinates of the Gauss-Legendre Point of the face  $f$ . From the above expression the information for the construction of the global matrix is available, and each entry of the matrix  $\mathbf{A}$  can be written in the following form:

$$A_{ij} = \sum_{f \in \mathcal{F}(i)} \left( \sum_{g \in \mathcal{G}(f)} (\mathbf{S}_f \cdot \mathbf{t}_{fj}(\mathbf{x}_g)) w_{Gg} \right), \quad (3.22)$$

where  $i$  is the CV where the discretization is applied,  $j$  is the neighbour cell that is used for the face  $f$  reconstruction and the column vector  $\mathbf{t}_{fj}$  is the column relative to the neighbour cell  $j$  of the face matrix  $\mathbf{T}_f$  presented in expression (3.19). The right side of the equation (3.4) will be further explained.

## Pseudo Inverse Matrix Computation

In order to obtain the pseudo inverse matrix, given by the equation (3.13), it is required to implement some basic matrix operations, such as: the transpose matrix, matrix product and the inverse matrix. In this subsection it will be provided a brief explanation of these operations to make possible the calculation of the  $\mathbf{P}$  matrix.

The transpose matrix,  $\mathbf{B}^T$ , of a generic matrix,  $\mathbf{B}$ , is an operation easy to implement, since it can be obtained by:

$$(\mathbf{B}_{ij})^T = \mathbf{B}_{ji}. \quad (3.23)$$

For the matrix product, it is required to take into account the matrix dimensions, since it is mandatory to obey certain rules, namely the number of columns of the first matrix must be equal to the number of rows of the second one. Considerer two generic matrices: the matrix  $\mathbf{B}$ , with  $(n \times m)$  entries, and the matrix  $\mathbf{G}$ , with  $(m \times p)$  entries, the product of these matrices will result in the matrix  $\mathbf{L}$  with  $(n \times p)$  entries and each matrix entry can be determine by:

$$L_{ij} = \sum_{k=1}^m B_{ik} G_{kj}. \quad (3.24)$$

The calculation of the inverse matrix is only possible when the matrix is nonsingular and is a square one, otherwise is noninvertible, (see Burden, [85]). Let  $\mathbf{B}$  a generic  $(n \times n)$  nonsingular matrix, which means that is invertible. It is possible to obtain the inverse matrix from the following system of equations:

$$\mathbf{BX} = \mathbf{I}, \quad (3.25)$$

where the  $\mathbf{I}$  is the identity matrix and the  $\mathbf{X}$  is the matrix that will result in the inverse of the matrix  $\mathbf{B}$ , so the above system of equations can be given by:

$$\mathbf{BB}^{-1} = \mathbf{I}. \quad (3.26)$$

To obtain the solution of that system of equations a Gauss elimination method is used, where at the beginning of the problem the following matrices are used  $[ \mathbf{B} | \mathbf{I} ]$  and at the end of the process it would be obtained  $[ \mathbf{I} | \mathbf{B}^{-1} ]$ . The Gauss elimination method is an algorithm to solve linear system of equations and consists in a transformation, through successive steps, of a generic matrix into a lower triangular one. The first phase is the elimination one and is follow by the regressive substitution. This method has efficiency problems due the number of required steps to achieved the solution, for a better understanding by the reader, see Pina, [76], and Burden, [85] have details about this method. To obtain the inverse matrix by this process it is required to perform twice the Gauss elimination method, one to obtain a lower triangular matrix and another one to obtain an upper triangular matrix, which in this case will lead to the identity matrix. Through the analysis of the required number of operations in order to obtain the inverse matrix allows to conclude that this process is slow, reason why the inverse matrix calculation should be performed only when it is mandatory, see Pina, [76].

### 3.2.3 Interior Faces

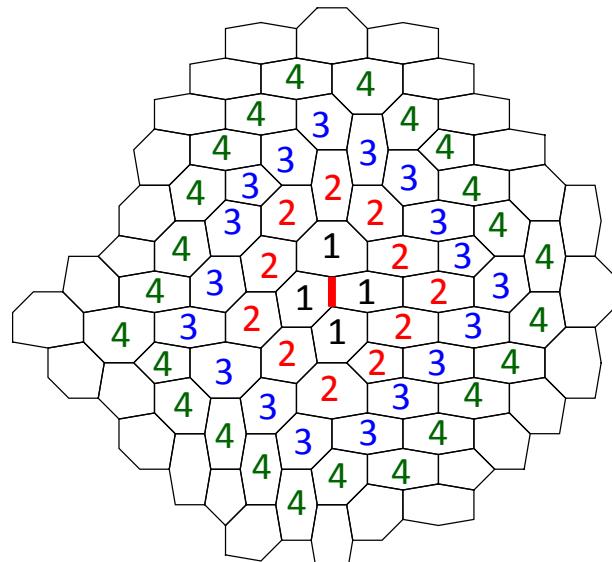
A Stencil is the set of points used for the polynomial reconstruction and its choice is very important. The interior faces, which are free from the influence of the boundaries are described separately from the faces influenced by the domain boundaries.

The polynomial reconstruction is face centered and every face has an associated stencil, represented by  $\mathcal{N}(f)$ . The stencil should be as compact as possible and should have enough points in each direction to maintain the required order. In addition the stencil size,  $n_s$  takes into account the number of terms of the polynomial reconstruction,  $n_{coefs}$  listed in table 3.1. If  $n_{coefs} > n_s$  the matrix  $\mathbf{P}$  will be singular and the WLS problem will be ill conditioned and consequently, without an accurate solution [21]. Consequently to avoid problems in the numerical scheme  $n_{coefs} < n_s$  is a necessary condition to have an overdetermined WLS problem [15, 21].

To have the most compact stencil, the cells that have at least one common vertex with face  $f$  are classified as first vertex neighbours of that face. The second order vertex neighbours are the cells that share at least one vertex with a cell that is first order vertex neighbours, and so on for the further vertex neighbours. Figure 3.6 shows two examples of the vertex neighbours that are used in the reconstructions of a face for the cases of a Cartesian grid and a Polyhedral irregular one, respectively.

4	4	4	4	4	4	4	4	4	
4	3	3	3	3	3	3	3	4	
4	3	2	2	2	2	2	3	4	
4	3	2	1	1	2	3	4		
4	3	2	1	1	2	3	4		
4	3	2	1	1	2	3	4		
4	3	2	2	2	2	2	3	4	
4	3	3	3	3	3	3	3	4	
4	4	4	4	4	4	4	4	4	

(a) Cartesian grid.



(b) Polyhedral irregular grid.

Figure 3.6: Example of the different orders of vertex neighbours of a face, marked in red, where the numbers inside of each cell represents the vertex neighbouring order of that cell relatively to the marked face.

Table 3.2: Order of vertex neighbours required for a  $p^{th}$  order polynomial.

$p$	1	3	5	7
Neighbours Order	1	2	3	4

Table 3.2 lists the order of vertex neighbours necessary to built the stencil for each of the polynomials

used in this work. The fourth vertex neighbours are required in the reconstruction of the 7<sup>th</sup> order polynomial.

Tests performed to verify if a  $p^{th}$  polynomial reconstruction actually rebuilds a  $p^{th}$  complete polynomial (the results of these tests are not included for consiness) have shown that in order to reconstruct the high-order terms, it is necessary to have in each axis direction at least  $p + 1$  cells. For the faces far enough from the boundaries this requirement was always achieved with the proposed methodology, see figure 3.6.

### 3.2.4 Boundary Treatment

The treatment given to the boundaries depends if the boundary type is Dirichlet or Neumann. For a Neumann boundary type, the flux through the face is known and it is not necessary to perform the face reconstruction because the required value in equation (3.4) can be directly applied, taking into account the Gauss points for the flux integration.

For a Dirichlet boundary the value of the transportable variable is known and the face is added to the reconstruction stencil. As a consequence the stencils of these faces will have included both cells and boundary faces, this also applies for interior faces near boundary faces. In addition, it is necessary to take into account the faces where the regression is being applied near the boundary (or on a boundary face) and does not satisfy the requirements mentioned previously, for example when  $n_s < n_{coefs}$ ,  $n_x^s < p + 1$  or  $n_y^s < p + 1$  happens, where  $n_x^s$  and  $n_y^s$  are the number of points of the stencil in each axis direction, XX and YY respectively. For these situations it is necessary to perform a stencil extension to accomplish the referred requirements.

Once the initial stencil is constructed the following expression allows to know its size:

$$n_x^s = \frac{L_x^s}{h_{ref}^s} + 1.6 = \frac{L_x^s}{h_{ref}^s} + 1.0 + 0.5 + 0.1, \quad (3.27)$$

where  $L_x^s = x_{\max}^s - x_{\min}^s$  is the length in XX direction of the stencil and centroid coordinates of the elements of  $\mathcal{N}_f$  are used in this calculation. The 1.6 factor takes into account the fact of the boundary face is added to the stencil, which contribution is 0.5, the contribution 1.0 is due the fact of the centroid coordinates of the elements of the stencil are used to calculate the stencil's length finally, 0.1 is a safety factor. The mean reference length of the stencil,  $L_{ref}^s$ , is given by:

$$h_{ref}^s = \frac{1}{n_{cells}^s} \sum_{i \in \mathcal{N}_c(f)} h_{ref}^i, \quad (3.28)$$

where  $n_{cells}^s$  is the number of cells in the stencil,  $\mathcal{N}_c(f)$  is the set of cells of the stencil corresponding to the face  $f$  and  $h_{ref}^i$  is the reference length of cell  $i$ . The calculation in other axis direction is performed similarly and equation (3.27) is valid for both types of boundary conditions and any scheme's order. When  $n_x^s$ ,  $n_y^s$  or both is lower than  $p + 1$ , the stencil will be extended in the respective direction. This operation is done for any face and repeated after an extension, in order to verified if it was sufficient.

Figure 3.7 shows an example of a stencil extension for a cubic polynomial reconstruction at the

marked boundary face, performed on an elliptic hybrid grid with Dirichlet boundary conditions. The initial stencil includes the first and second order vertex neighbours, numbered with the black color, and the boundary faces are marked with the blue one. In the YY direction the current stencil is long enough and will not grow in that direction, but in the XX direction it is necessary to perform a stencil expansion. To avoid cells in the YY direction a lower and upper YY bounds of the stencil are computed according with the centroid coordinates of the elements of the stencil and defined by  $y_{\min}^s$  and  $y_{\max}^s$ , respectively, which are represented by the dotted lines in the figure 3.7. The new cells added to the stencil after the extension are the third order vertex neighbours represented in figure 3.7 with the green color and numbered by 3. Finally the extension can be in one or both directions according with the requirements mentioned above.

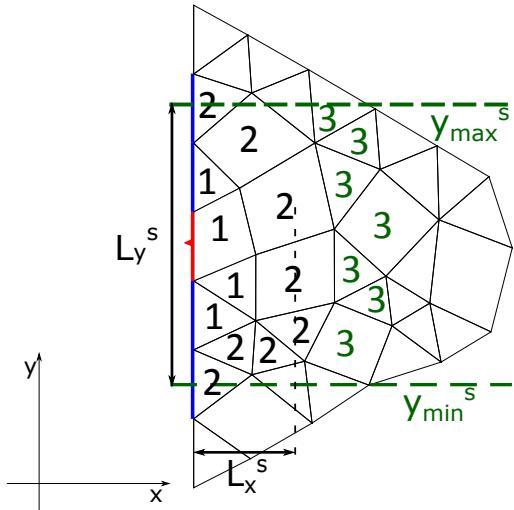


Figure 3.7: Example of a stencil extension in an elliptic hybrid grid for a cubic polynomial reconstruction for the face represented with the red color and the  $\triangleleft$  mark, the blue ones are the other boundary faces included in the regression.

For Neumann boundary faces the diffusive flux is known and their contribution to the WLS problem is slightly different of other faces. The line in the  $\mathbf{D}_f$  matrix referent to the boundary face,  $b$ , is included in the form of  $\nabla \mathbf{d}_f(\mathbf{x}_b) \mathbf{n}_b$ , instead of  $\mathbf{d}_f(\mathbf{x}_b)$ , where  $\mathbf{n}_b = (n_{b_x}, n_{b_y})$  is the boundary face outward unitary normal vector and the entry for the boundary face in the vector  $\phi_f$  will be  $\nabla \phi_b \hat{\mathbf{n}}_b$  instead of  $\phi_b$ . The system of linear equations given by equation (3.8), it will be given for this case as:

$$\begin{bmatrix} 1 & (x_1 - x_f) & (y_1 - y_f) & (x_1 - x_f)^2 & (y_1 - y_f)^2 & \dots \\ 1 & (x_2 - x_f) & (y_2 - y_f) & (x_2 - x_f)^2 & (y_2 - y_f)^2 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \\ 0 & 1n_{b_x} & 1n_{b_y} & 2(x_b - x_f)n_{b_x} & 2(y_b - y_f)n_{b_y} & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_{n_{coefs}} \end{bmatrix} = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \nabla \phi_b \hat{\mathbf{n}}_b \\ \vdots \end{bmatrix}. \quad (3.29)$$

### 3.3 Source Term

A high-order Gauss-Legendre Quadrature (GLQ) for the integral of the source term of the Poisson equation,  $\varphi_{\phi P}$  is also required. For two dimensional space the GLQ can be directly applied for quadrilateral and triangular 2D forms and tetrahedral 3D forms. Any cell can be divided into triangular or tetrahedral sub-cells for 2D or 3D respectively. This work is related with 2D unstructured grids, so it becomes necessary to divide the non triangular cells into triangular sub-cells, like the example of figure 3.8(a), where every triangular element has a common vertex, the cell centroid.

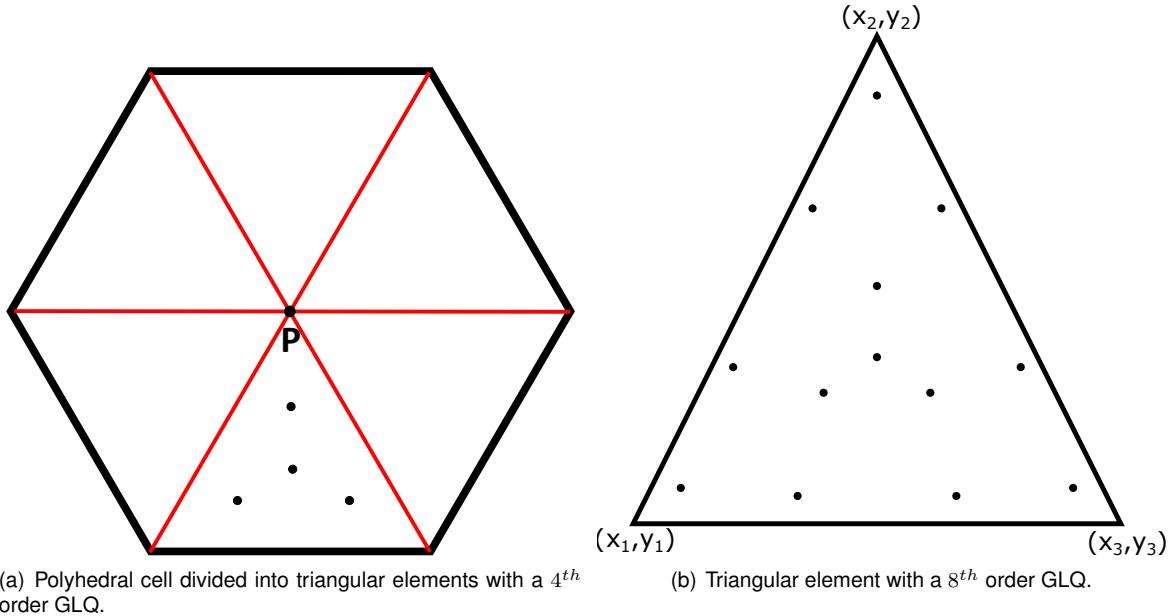


Figure 3.8: Example of a division of a polyhedral cell in triangles in order to perform the numerical integration of source term at cell  $P$ , in red is the face's sub-triangles. The dots represent the Gauss points coordinates of the GLQ

Simplex coordinates,  $(\zeta_1, \zeta_2, \zeta_3, \dots)$  are used to determine the Gauss integration points coordinates for each triangular element. A simplex region uses the number of points given by the dimension space plus one, so for the 2D space will be necessary 3 points and the area coordinates are obtained according to Akin, [74].

The integration of a triangular element is obtained by:

$$I_{\Delta}(f) = S_{\Delta} \sum_{i \in \mathcal{G}(\Delta)} w_{G_i} f(\zeta_1, \zeta_2, \zeta_3), \quad (3.30)$$

where  $\zeta_i$  denote the triangular coordinates, symplex region, of the triangular element, the respective area is computed by  $S_{\Delta} = ((x_1 - x_3) \times (x_2 - x_3)) / 2$  and  $\mathcal{G}(\Delta)$  is the set of Gauss points of the triangle  $\Delta$ . The final expression for the integration of the source term is given by:

$$\int_{CV} \varphi_{\phi} dV = \sum_{i \in \mathcal{T}(P)} \left( S_{\Delta_i} \sum_{j \in \mathcal{G}_i(\Delta_i)} w_{G_j} \varphi_{\phi}(\mathbf{x}_j) \right), \quad (3.31)$$

where  $\mathcal{T}(P)$  is the set of triangular elements resultant of the cell  $P$  division. The Gauss points coordinates of each triangular element are expressed as:

$$\mathbf{x}_j = \zeta \mathbf{x}_v = \begin{bmatrix} \zeta_{1_j} & \zeta_{2_j} & \zeta_{3_j} \end{bmatrix} \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \end{bmatrix} \quad (3.32)$$

where the  $\mathbf{x}_v$  is the vertex coordinates of the triangular element. Table 3.3 lists for a triangular element the number of required Gauss points and their coordinates with the respective Gauss weight, for each reconstruction order.

Table 3.3: Simplex coordinates and Gauss weight for a triangular element for a  $p^{th}$  order integration, from Akin, [74].

$p$	$n_G$	$\zeta_1$	$\zeta_2$	$\zeta_3$	$M$	$w_{G_i}$
1	1	0.3333333333333333	0.3333333333333333	0.3333333333333333	1	1
3	4	0.3333333333333333 0.6	0.3333333333333333 0.2	0.3333333333333333 0.2	1 3	-0.5625 0.5208333333333333
5	7	0.3333333333333333 0.797426985353087 0.059715871789770	0.3333333333333333 0.101286577323456 0.470142064105115	0.3333333333333333 0.101286577323456 0.470142064105115	1 3 3	0.225 0.125939180544827 0.132394152788506
7	13	0.3333333333333333 0.479308067841923 0.869739794195568 0.638444188569809	0.3333333333333333 0.260345966079038 0.065130102902216 0.312865496004875	0.3333333333333333 0.260345966079038 0.065130102902216 0.048690315425316	1 3 3 6	-0.14957004446767 0.175615257433204 0.053347235608839 0.077113760890257



# Chapter 4

## Results and Discussion for Unstructured Grids

In this chapter several numerical tests are presented to investigate the accuracy in respect to the weight function of WLS. This will be followed by several numerical tests under different analytical solutions to investigate the influence of several aspects of the simulation, such as: grid type, boundary conditions and the grid quality parameters. Finally an efficiency study based on the computational cost and computing time was performed in order to understand the advantages of the very high-order schemes and polyhedral unstructured grids.

### 4.1 Selection of Weight Function

In this section several numerical tests are presented to investigate the weight function of WLS method and its performance is compared for two different grid types: the Cartesian and the hybrid, which an example of each grid type is presented in figures 3.2(a) and 3.2(d), respectively. The weight function and shape factor can influence the properties of the matrix  $\mathbf{A}$ , accuracy order and also the magnitude of the schemes, according to Clain et al., [29] and Chenoweth et al., [86].

In the present work three weight functions,  $w_{LS_1}$ ,  $w_{LS_2}$  and  $w_{LS_3}$ , are compared. The first one is given by:

$$w_{LS_1}(\mathbf{x}_P) = \frac{1}{(d_{fP})^k}, \quad (4.1)$$

where  $d_{fP} = \sqrt{(x_P - x_f)^2 + (y_P - y_f)^2}$  is the distance between the face  $f$  and the cell centroid  $P$  and  $k$  is the shape factor, typically is equal 2, this function was based on Ollivier et al., [15], and Cueto-Felgueroso et al. [22]. The second one is given by:

$$w_{LS_2}(\mathbf{x}_P) = \frac{\exp\left(-\left(\frac{kd_{fP}}{2d_{f\max}}\right)^2\right) - \exp(-k^2)}{1 - \exp(-k^2)}, \quad (4.2)$$

where  $d_{f_{\max}} = \max(d_{fP})$  with  $P = 1 \dots n_s$ , can be found in [23], and finally:

$$w_{LS_3}(\mathbf{x}_P) = \frac{\exp\left(-\left(\frac{kd_{fP}}{d_{f_{\max}}}\right)^2\right) - \exp(-k^2)}{1 - \exp(-k^2)}, \quad (4.3)$$

that is based on the previous one, where is used the stencil radius,  $d_{f_{\max}}$  instead of the stencil diameter,  $2d_{f_{\max}}$ .

Lets consider an analytical solution of the Poisson equation given by:

$$\phi(x, y) = \exp\left(-\frac{(x - 0.5)^2 + (y - 0.5)^2}{0.0175}\right), \quad (4.4)$$

and the source term  $\varphi_\phi$  according to  $\nabla^2\phi$ , which is given by:

$$\varphi_\phi(x, y) = \nabla^2\phi(x, y) = -\frac{4}{0.0175} \left(1 - \frac{(x - 0.5)^2 + (y - 0.5)^2}{0.0175}\right) \phi(x, y), \quad (4.5)$$

#### 4.1.1 Selection by Properties of Matrix A

According to Clain et al., [29], the global matrix should satisfy another requirements, such as: the main diagonal should be negative or the inverse of the global matrix could has all entries with the same signal, those requirements were also successfully verified for all weight function and shape factors considered in this study.

The matrix condition number,  $\text{cond}(\mathbf{A})$ , is the ratio between the largest and smallest matrix's eigenvalues. When the condition number is very high, it means that there is an eigenvalue close to zero or infinite and the matrix is close to be singular and will be ill conditioned and as a consequence there is not an unique solution to the system of equations.

An estimation to the matrix condition number that results from the discretization of the Poisson equation in an uniform grid is given according with Demmel, [87] and Baty, [56], by:

$$\text{cond}(\mathbf{A}) = \frac{4}{\pi^2 h^2}. \quad (4.6)$$

The above expression only depends on the grid size,  $h$ , and was deduced for a 2<sup>nd</sup> order finite difference method. For a Cartesian grid with 1600 cells,  $h_{ref} = 2.5E - 02$ , and for a hybrid grid with 2969 cells,  $h_{ref} = 1.60E - 02$ , it gives  $\text{cond}(\mathbf{A}) = 6.48E + 02$  and  $\text{cond}(\mathbf{A}) = 1.59E + 03$ , respectively. These values are taken for reference purposes, and to be compared with the effective matrix condition numbers obtained with the different weight functions. Tables 4.1, 4.2 and 4.3 list the obtained condition number for the three weight functions at study with different shape factors,  $k$ , for all schemes at study.

Tables 4.1 to 4.3 show that  $k$  has negligible influence in the matrix condition number for Cartesian grids. For the hybrid grids it is possible to verify that the shape factor influences the matrix condition number. The results are, in general, consistent with the reference value, which is obtained from equation (4.6).

Table 4.1: The matrix condition number obtained with  $w_{LS_1}$  for the Cartesian grid (left) and hybrid grid (right).

$k$	Cartesian				Hybrid			
	FLS2	FLS4	FLS6	FLS8	FLS2	FLS4	FLS6	FLS8
0	2.80E+03	7.98E+03	4.22E+04	7.23E+04	4.69E+05	1.54E+05	1.11E+06	4.34E+06
2	6.74E+02	9.69E+02	1.33E+03	1.73E+03	1.33E+03	2.98E+03	4.17E+03	4.31E+03
4	8.73E+02	1.34E+03	1.52E+03	1.84E+03	2.11E+03	3.50E+03	4.26E+03	5.45E+03
6	9.28E+02	1.41E+03	1.53E+03	1.83E+03	2.48E+03	8.79E+04	4.36E+03	6.22E+03
8	9.40E+02	1.37E+03	1.57E+03	1.83E+03	2.69E+03	6.00E+03	5.76E+03	6.82E+03

Table 4.2: The matrix condition number obtained with  $w_{LS_2}$  for the Cartesian grid (left) and hybrid grid (right).

$k$	Cartesian				Hybrid			
	FLS2	FLS4	FLS6	FLS8	FLS2	FLS4	FLS6	FLS8
3	7.09E+02	2.24E+03	3.93E+05	4.12E+05	5.31E+04	4.45E+05	5.08E+06	5.37E+06
4	8.72E+02	9.63E+02	6.75E+04	2.19E+05	1.73E+03	3.01E+05	1.40E+07	3.00E+08
5	9.30E+02	1.01E+03	1.47E+03	1.92E+05	1.96E+03	2.43E+04	6.61E+06	2.99E+06

Table 4.3: The matrix condition number obtained with  $w_{LS_3}$  for the Cartesian grid (left) and hybrid grid (right).

$k$	Cartesian				Hybrid			
	FLS2	FLS4	FLS6	FLS8	FLS2	FLS4	FLS6	FLS8
3	NaN	1.07E+03	1.48E+03	2.38E+03	2.22E+03	3.53E+03	7.07E+05	9.15E+07
4	NaN	1.31E+03	1.48E+03	1.85E+03	3.45E+03	5.13E+05	5.12E+03	5.05E+04
5	NaN	1.40E+03	1.66E+03	1.92E+03	4.24E+03	9.88E+03	8.11E+03	1.11E+04
6	NaN	1.43E+03	1.75E+03	2.08E+03	4.46E+03	1.90E+04	1.20E+04	2.36E+04
8	NaN	1.47E+03	1.21E+04	3.40E+04	NaN	1.92E+05	1.66E+06	1.34E+05

Table 4.1 lists the matrix condition number for the weight function  $w_{LS_1}$ . For  $k = 0$  (an unweighted LS problem) the condition number is one or two orders of magnitude higher than the reference values for both grids and consequently this shape factor will not be considered any more for the  $w_{LS_1}$  function. Other values of  $k$  give condition numbers with similar orders of magnitude of equation (4.6) and still being considered.

Table 4.2 lists the condition number for weight function  $w_{LS_2}$ , which have the highest condition numbers, specifically three orders of magnitude greater than the reference values for FLS6 and FLS8, and consequently will not be considered. The  $w_{LS_2}$  was used by Nogueira, [23–25, 27], for a fourth-order scheme on both quadrilateral and triangular grids. The present results show that for the FLS6 and FLS8 schemes with the hybrid grid, this weight function produces large matrix condition numbers, which will lead to a ill-posed problem. Due to these facts, the  $w_{LS_2}$  function will not be considered in the next sections.

Table 4.3 lists the condition number for the weight function  $w_{LS_3}$ . For a Cartesian grid this function can not be used for FLS2 scheme because there are some entries in the main diagonal of the global matrix that are zero and it is impossible to achieve a solution. For the other schemes the condition number is of the same order of magnitude of the reference value and it is very similar to the one obtained with the function  $w_{LS_1}$ . On the hybrid grid the condition number for some  $k$ 's are two orders of magnitude greater than the reference value and consequently, they will be discarded from this study, namely  $k = 8$  for all schemes,  $k = 4$  for the FLS4 scheme and  $k = 3$  for both FLS6 and FLS8 schemes.

The  $w_{LS_3}$  will be considered as a possible weight function, although it does not work with the FLS2 scheme but, since our interest is mainly in high-order schemes.

### 4.1.2 Selection by Schemes Convergence Order

In this section the convergence will be analysed for the case of the  $w_{LS_1}$  and  $w_{LS_3}$  weighted function, given by equations (4.1) and (4.3), respectively.

This work uses the error norm-1 and the error norm- $\infty$ , given by the expressions (2.18) and (2.20), respectively, to determine the convergence order of each proposed scheme, expression (2.21) was used.

Table 4.4: The  $\mathcal{O}_1$  of the schemes for Cartesian and hybrid grids with the weight function  $w_{LS_1}$ , where the (–) indicates that a converged solution was not achieved.

$k$	Cartesian				Hybrid			
	FLS2	FLS4	FLS6	FLS8	FLS2	FLS4	FLS6	FLS8
2	2.00	3.92	5.94	7.75	2.03	4.03	6.03	7.50
4	2.00	4.01	6.03	8.25	2.05	4.08	5.94	7.72
6	2.00	3.98	6.06	8.36	2.05	—	6.01	7.51
8	2.00	3.99	6.12	8.09	2.05	—	5.96	7.58

Table 4.4 lists the average convergence order for all schemes with the weight function  $w_{LS_1}$ . For the Cartesian grid, the results show that for every  $k$  the theoretical order of convergence is achieved. The hybrid grid displays cases where the order of accuracy is slightly below to the theoretical values, but

Table 4.5: The  $\mathcal{O}_1$  of the schemes for Cartesian and hybrid grids with the weight function  $w_{LS_3}$ , where (–) indicates that a converged solution was not achieved and (\*) the discarded shape factors from the previous section.

k	Cartesian				Hybrid			
	FLS2	FLS4	FLS6	FLS8	FLS2	FLS4	FLS6	FLS8
3	–	3.97	*	*	–	3.88	*	*
4	–	*	6.02	7.96	–	*	5.73	–
5	–	3.99	6.02	8.10	–	2.53	5.86	6.70
6	–	3.99	6.02	7.25	–	–	4.73	6.90

close. It is worth mentioning that the hybrid grid may be considered as a strong robustness test to the schemes and suggests that the schemes are robust. Note that with  $k$  equal 6 and 8 the solver does not achieve a converged solution for FLS4 scheme with the hybrid grid. This result is expected since table 4.1 shows higher condition numbers for these cases, which prompts the lack of convergence and so these values of  $k$  will be discarded.

Table 4.5 lists the average convergence order for all schemes with the weight function  $w_{LS_3}$ . For the Cartesian grid the theoretical order of convergence is almost achieved for all the shape factors. However for the hybrid grid this weight function never achieved the theoretical order of convergence for the FLS8 scheme and the overall results are not as consistent as in the case of the  $w_{LS_1}$ . Failing in the robustness test here performed and as a consequence the  $w_{LS_3}$  will not be selected as a weight function. From the analysis of tables 4.4 and 4.5 the best choice for a weight function is  $w_{LS_1}$ .

### 4.1.3 Selection by Numerical Error Magnitude

The previous sections have proposed that the  $w_{LS_1}$  is the best choice to be the WLS weight function, but the best shape factor,  $k$  was not selected yet. To achieve this, a study of the shape factor,  $k$ , influence in the values of the numerical error will be conducted.

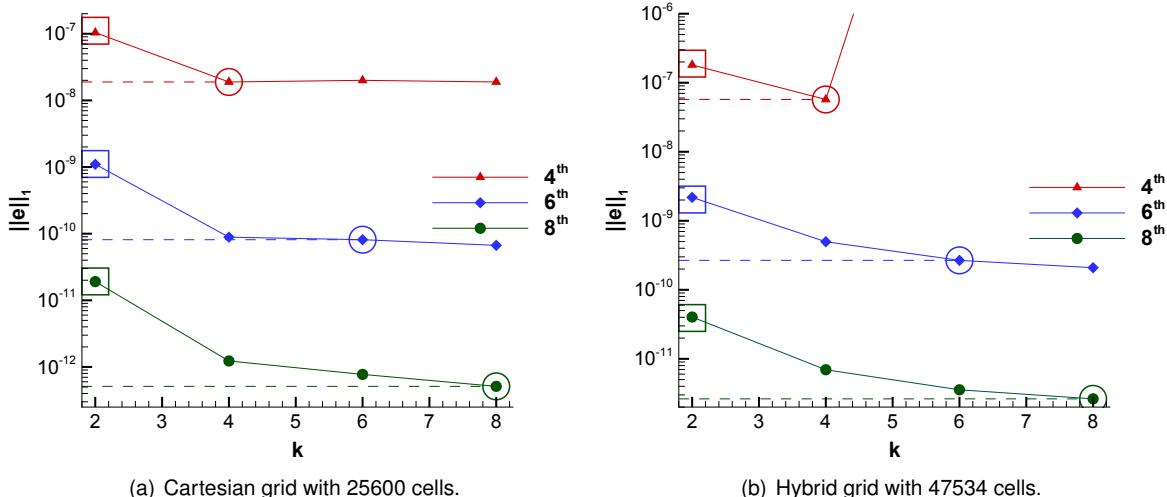


Figure 4.1: The  $\|e\|_1$  for all high-order schemes, where the  $\circ$  is the proposed shape factor and the  $\square$  is the shape factor used by most of the other authors.

Table 4.6: The  $\|e\|_1$  for a Cartesian grid with 25600 cells and a hybrid grid with 47534 cells for  $k = 2$  and  $k = q$ , where in the last line is the ratio between the errors of the two shape factors.

$k$	Cartesian Grid (25600 cells)			Hybrid Grid (47534 cells)		
	FLS4	FLS6	FLS8	FLS4	FLS6	FLS8
2	1.04E-07	1.10E-09	1.91E-11	1.81E-07	2.18E-09	4.03E-11
$q$	1.89E-08	8.12E-11	5.11E-13	5.72E-08	2.67E-10	2.63E-12
Ratio	5.49	13.48	37.45	3.16	8.16	15.33

The  $w_{LS_1}$  with a shape factor,  $k$ , equal to two is a very popular choice, see e.g. Gooch et al. [15, 18] and Cuetto-Felgueroso et al. [22].

Figure 4.1 shows the accuracy of the three high-order schemes for several shape factors, with a Cartesian grid with 25600 cells and a hybrid grid with 47345 cells. There is a significant difference between  $k = 2$  and the others shape factors. From figure 4.1(a) the best shape factor for the Cartesian grid is  $k = 8$ , however for the FLS4 scheme the best result appears with  $k = 4$  for the hybrid grid, see figure 4.1(b). Consequently, one may conclude that there is a relation between the order of convergence of the scheme and the shape factor. Due to this, the shape factor will be equal to the order of convergence of scheme,  $k = q$ , where  $q = p + 1$  is the theoretical order of convergence of the scheme.

Table 4.6 and figure 4.1 compare the error  $\|e\|_1$  for  $k = 2$  and  $k = q$ , when  $k$  takes into account the order of the scheme the solution accuracy is improved, specially for the FLS8. Generally the accuracy increases for a  $k$  value equal to the order of the scheme and the error could be reduced substantially from 3 to 37 times the error obtained with  $k = 2$ , depending on the considered scheme and grid. The dotted lines in figure 4.1 shows that difference between  $k$ 's. Similar conclusions presented in table 4.6 can be derived from the maximum error,  $\|e\|_\infty$ . From now on the weight function will be defined by:

$$w_{LS}(\mathbf{x}_P) = \frac{1}{(d_{fP})^q} = \frac{1}{(d_{fP})^{p+1}}, \quad (4.7)$$

which is the selected weight function for the proposed schemes.

## 4.2 Influence of Grid Type

To analyse the influence of the grid type on the proposed schemes, numerical experiments were performed with four different grid types: Cartesian; triangular; polyhedral and hybrid, which an example of each grid type is presented in figure 3.2.

The numerical verification was performed with an analytical solution of the Poisson equation with Dirichlet boundary in a square domain  $[0, 1]$ , which is given by:

$$\phi(x, y) = \psi_1 - \psi_2 - \psi_3 + \psi_4, \quad (4.8)$$

where  $\psi_i$  is an auxiliary exponential function, which is given by:

$$\psi_i = \exp\left(-120\left((x - x_i)^2 + (y - y_i)^2\right)\right), \quad (4.9)$$

where the peaks have the following coordinates  $(x_1, y_1) = (\frac{1}{3}, \frac{1}{3})$ ,  $(x_2, y_2) = (\frac{1}{3}, \frac{2}{3})$ ,  $(x_3, y_3) = (\frac{2}{3}, \frac{1}{3})$  and  $(x_4, y_4) = (\frac{2}{3}, \frac{2}{3})$  and the source term,  $\varphi_P$ , is computed according to  $\nabla^2\phi$ .

#### 4.2.1 Accuracy Order Verification

Figure 4.2(a) and 4.2(b) show the convergence curves of the schemes with the Cartesian grid for the mean and maximum error, respectively. For the finest grid the error of the FLS8 scheme is close to the machine truncation error. Tables 4.7 and 4.8 list the errors 1-norm and  $\infty$ -norm, respectively. The effective convergence order that is virtually equal to the theoretical value for the 1-norm and in close agreement for the  $\infty$ -norm. In addition the error norms for FLS8 are almost nine orders of magnitude lower than those obtained with the second-order scheme. Figures 4.2(c) and 4.2(d) and tables 4.9 and 4.10 list similar values of the error norms and its effective convergence order for the finest triangular grids, where the results are the expected for both norms, being super convergent for the FLS6 and FLS8 scheme.

The results obtained for irregular polyhedral grids (see figure 4.2(e) and 4.2(f), and tables 4.11 and 4.12) also display the expected order of accuracy, it is possible to verify that the higher-order schemes are slightly super convergent because the cells have more faces and the stencil is more compact. Note that once again the FLS8 reaches the machine truncation error. A regular polyhedral grid was also tested, the conclusions about the schemes are identical to the irregular polyhedral one but the values of numerical error was one order of magnitude lower.

Tests with an elliptic hybrid grid were also performed and presented in tables 4.13 and 4.14, and figures 4.2(g) and 4.2(h). The results show that the proposed method is very robust and with a convergence order close to the theoretical one.

Table 4.7: Obtained  $\|e\|_1$  and  $\mathcal{O}_1$  for Cartesian grid.

Cartesian		FLS2		FLS4		FLS6		FLS8	
$n_{cells}$	$h_{ref}$	$\ e\ _1$	$\mathcal{O}_1$						
25600	6.25E-03	1.29E-04	2.00	1.50E-07	3.99	1.34E-09	5.97	1.62E-11	8.05
57600	4.17E-03	5.72E-05	2.00	2.96E-08	4.00	1.18E-10	5.99	6.17E-13	8.07
102400	3.13E-03	3.22E-05	2.00	9.36E-09	4.00	2.10E-11	6.00	6.45E-14	7.85

Table 4.8: Obtained  $\|e\|_\infty$  and  $\mathcal{O}_\infty$  for Cartesian grid.

Cartesian		FLS2		FLS4		FLS6		FLS8	
$n_{cells}$	$h_{ref}$	$\ e\ _\infty$	$\mathcal{O}_\infty$						
25600	6.25E-03	1.87E-03	2.01	1.96E-06	3.98	8.10E-09	5.93	1.80E-10	7.92
57600	4.17E-03	8.28E-04	2.01	3.86E-07	4.01	7.22E-10	5.96	7.04E-12	7.99
102400	3.13E-03	4.66E-04	2.00	1.22E-07	4.01	1.33E-10	5.89	8.42E-13	7.38

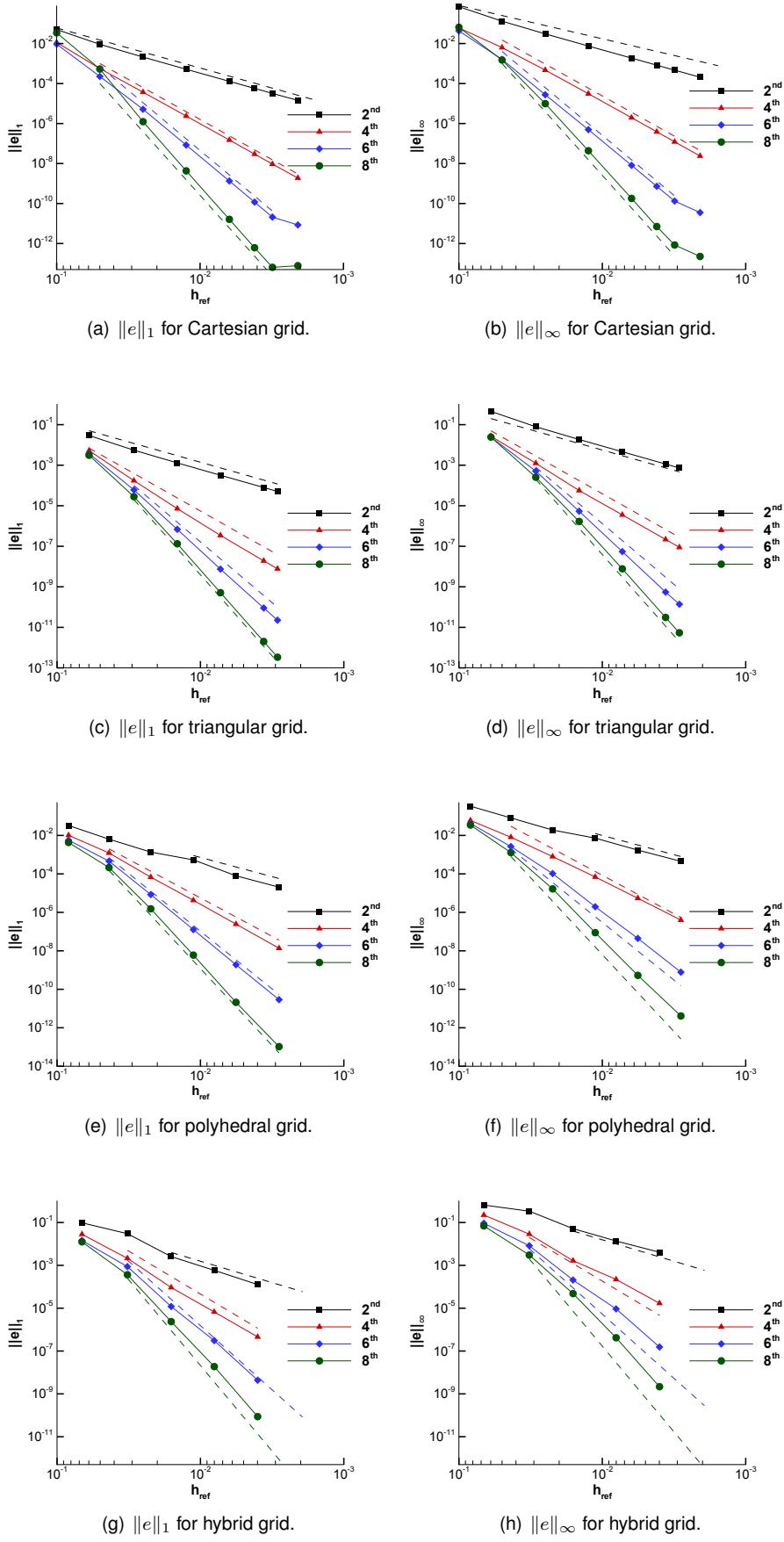


Figure 4.2: Convergence curves for all grid types for  $\|e\|_1$  and  $\|e\|_\infty$  with all schemes, where the dotted line represents the theoretical slope of each scheme.

Table 4.9: Obtained  $\|e\|_1$  and  $\mathcal{O}_1$  for triangular grid.

Triangular		FLS2		FLS4		FLS6		FLS8	
$n_{cells}$	$h_{ref}$	$\ e\ _1$	$\mathcal{O}_1$						
14632	7.25E-03	3.20E-04	2.02	3.44E-07	4.37	7.47E-09	6.50	5.15E-10	8.00
58698	3.62E-03	7.94E-05	2.01	1.91E-08	4.16	9.03E-11	6.36	2.00E-12	8.00
91772	2.90E-03	5.08E-05	2.00	7.74E-09	4.05	2.25E-11	6.22	3.35E-13	8.00

 Table 4.10: Obtained  $\|e\|_\infty$  and  $\mathcal{O}_\infty$  for triangular grid.

Triangular		FLS2		FLS4		FLS6		FLS8	
$n_{cells}$	$h_{ref}$	$\ e\ _\infty$	$\mathcal{O}_\infty$						
14632	7.25E-03	4.60E-03	2.01	3.54E-06	3.98	5.48E-08	6.60	7.70E-09	7.75
58698	3.62E-03	1.14E-03	2.01	2.17E-07	4.02	5.49E-10	6.64	3.08E-11	7.96
91772	2.90E-03	7.30E-04	2.00	8.84E-08	4.03	1.38E-10	6.19	5.38E-12	7.81

 Table 4.11: Obtained  $\|e\|_1$  and  $\mathcal{O}_1$  for polyhedral grid.

Polyhedral		FLS2		FLS4		FLS6		FLS8	
$n_{cells}$	$h_{ref}$	$\ e\ _1$	$\mathcal{O}_1$						
8321	1.12E-02	5.35E-04	1.37	4.23E-06	4.07	1.27E-07	6.15	5.97E-09	8.14
33024	5.65E-03	7.97E-05	2.77	2.45E-07	4.15	1.89E-09	6.12	2.09E-11	8.23
131585	2.83E-03	2.04E-05	1.98	1.36E-08	4.19	2.89E-11	6.06	1.04E-13	7.69

 Table 4.12: Obtained  $\|e\|_\infty$  and  $\mathcal{O}_\infty$  for polyhedral grid.

Polyhedral		FLS2		FLS4		FLS6		FLS8	
$n_{cells}$	$h_{ref}$	$\ e\ _\infty$	$\mathcal{O}_\infty$						
8321	1.12E-02	7.24E-03	1.43	6.93E-05	3.58	1.95E-06	5.83	8.80E-08	7.69
33024	5.65E-03	1.72E-03	2.09	5.39E-06	3.72	4.42E-08	5.51	5.28E-10	7.45
131585	2.83E-03	4.51E-04	1.94	3.97E-07	3.78	7.78E-10	5.86	4.11E-12	7.04

 Table 4.13: Obtained  $\|e\|_1$  and  $\mathcal{O}_1$  for hybrid grid.

Hybrid		FLS2		FLS4		FLS6		FLS8	
$n_{cells}$	$h_{ref}$	$\ e\ _1$	$\mathcal{O}_1$						
2969	1.60E-02	2.57E-03	3.50	9.23E-05	4.46	1.20E-05	6.10	2.36E-06	7.17
11827	8.01E-03	5.84E-04	2.14	6.72E-06	3.79	3.06E-07	5.31	1.88E-08	6.99
47534	3.99E-03	1.27E-04	2.19	4.00E-07	4.05	4.35E-09	6.11	8.68E-11	7.73

 Table 4.14: Obtained  $\|e\|_\infty$  and  $\mathcal{O}_\infty$  for hybrid grid.

Hybrid		FLS2		FLS4		FLS6		FLS8	
$n_{cells}$	$h_{ref}$	$\ e\ _\infty$	$\mathcal{O}_\infty$						
2969	1.60E-02	5.19E-02	2.67	1.65E-03	4.06	2.07E-04	5.24	4.80E-05	5.90
11827	8.01E-03	1.37E-02	1.92	2.20E-04	2.91	9.26E-06	4.50	4.17E-07	6.87
47534	3.99E-03	4.09E-03	1.74	1.66E-05	3.72	1.54E-07	5.88	2.16E-09	7.56

#### 4.2.2 Grid Comparison Study

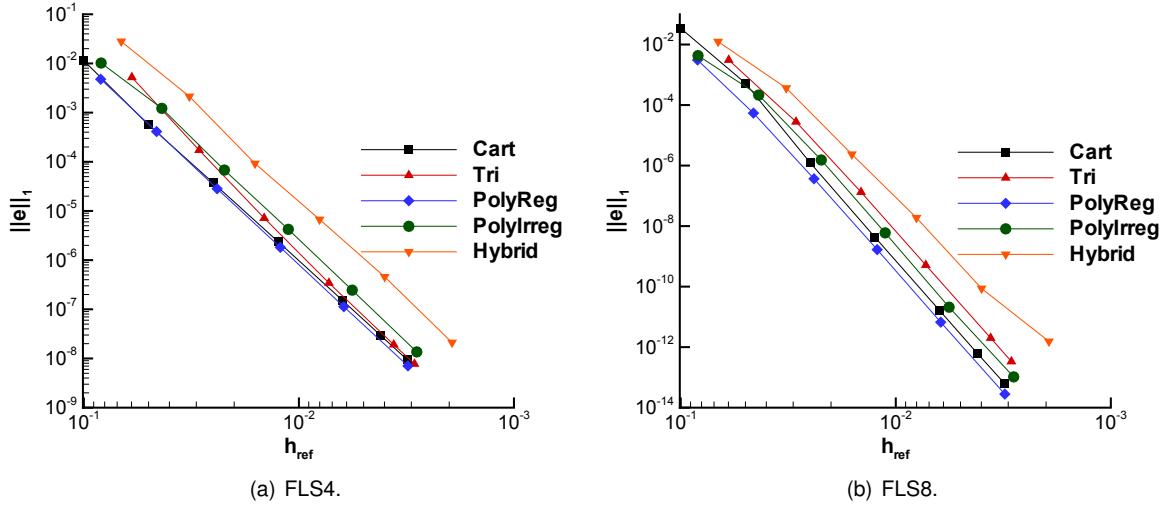


Figure 4.3: Convergence curves of the error 1-norm with the  $h_{ref}$  for the different grids with the FLS4 and FLS8.

Figure 4.3 shows a comparison of the  $\|e\|_1$  function of the reference length using different grids for FLS4 and FLS8 schemes. Figure 4.3(a) refers to the FLS4 scheme and for the same  $h_{ref}$  the regular grids displays lower numerical errors than the irregular ones, which is expected, since between cells the parameters are almost constant. For the smaller reference lengths the Cartesian, triangular and regular polyhedral grids give identical errors. The super convergent behaviour reported previously is easily to observe for the FLS4 scheme, see figure 4.3(a). After the hybrid grid, the irregular polyhedral grid is the one that presents the worse behaviour for FLS4 scheme. Note that, the hybrid grid should be not considered for this comparison because it represents only a robustness test, even it is shown on figure 4.3, and it is possible to observe that the error level is almost two orders of magnitude greater that the other grid types for both schemes.

Figure 4.3(b) refers to the FLS8 scheme and shows that the difference between grids is more evident, not existing overlap between grids. For this scheme, the triangular grid has the worse behaviour for the same grid parameter, being the error one order of magnitude worse than the irregular polyhedral grid. On the other hand the regular polyhedral grid has the best behaviour and it is detached of the Cartesian one. Note that from this figure, it is possible to observe that there is a gain of one order of magnitude between the use of a regular polyhedral grid in opposite of the use of an irregular one.

Table 4.15 lists the reference length required for an error 1-norm level of 1.00E – 10 for all schemes and grid types, where  $h_{ref}$  is estimated. Through the table, it is possible to make a comparison between schemes and grids in order to determine what is the choice that does not require a very refined grid based on grid parameter. For every scheme the regular polyhedral grid is the one that has the highest reference length and the triangular one distances itself from the others grid types with the worse behaviour. The results obtained from this table indicate that the polyhedral grid could be the best choice to reduce the computational cost of the simulation, as will be proved latter.

Table 4.15: Reference length required for a  $\|e\|_1$  of 1.00E-10 for all schemes and grid types.

Grid	FLS2		FLS4		FLS6		FLS8	
	$h_{ref}$	Ratio	$h_{ref}$	Ratio	$h_{ref}$	Ratio	$h_{ref}$	Ratio
Cart	5.51E-06	0.76	9.92E-04	0.92	4.05E-03	0.89	7.83E-03	0.91
Tri	4.11E-06	0.57	9.90E-04	0.92	3.68E-03	0.81	5.90E-03	0.68
PolyReg	7.25E-06	1.00	1.08E-03	1.00	4.54E-03	1.00	8.64E-03	1.00
PolyIrreg	5.81E-06	0.80	8.76E-04	0.81	3.48E-03	0.77	6.83E-03	0.79
Hybrid	3.35E-06	0.46	5.51E-04	0.51	2.21E-03	0.49	4.07E-03	0.47

### Final Remarks

The influence of the grid type is negligible in terms of the convergence order, being the expected one for both  $\|e\|_1$  and  $\|e\|_\infty$  error norms. The grid type only affects the accuracy of the schemes, being possible to conclude that the use of an eighth-order scheme for a determined grid parameter is advantageous. Note to the results obtained with the hybrid grid, passing in the robustness tests performed, suggesting that the proposed schemes are robust and capable to be applied with the most extreme conditions. Finally the results suggest that the regular polyhedral grid is the choice that requires a higher reference length which is an indication that this grid type reduces the computational cost of the simulation for a certain error magnitude.

## 4.3 Influence of Neumann Boundary Conditions

To verify the applicability of the proposed schemes to other boundary conditions, namely the Neumann ones, numerical tests were performed for both cases of imposed null flux and not-null flux applied to several grid types. Like in previous sections the numerical validation was performed with an analytical solution of the Poisson equation in a square domain,  $[0, 1]$ , with Neumann boundary condition at the vertical faces and Dirichlet boundary condition for the remaining ones.

For this kind of problems two different approaches were used. The first one is the generalization of the proposed method for Neumann boundary conditions, which was already explained in previous chapter. The linear system of equations to be solved with the WLS method is given by equation (3.29). Once this approach is based on the general case, it will be defined as general case (GC) approach.

The second approach which is new to the Author's knowledge, it can only be used with Neumann BC and it consists on the multiplication of each line of the  $\mathbf{D}_f$  matrix referent to Neumann boundary face,  $b$ , by the respective face area,  $S_b$ . That line will be written by  $\nabla \mathbf{d}_f(\mathbf{x}_b) S_b$ , instead of  $\nabla \mathbf{d}_f(\mathbf{x}_b) \mathbf{n}_b$  and the entry for the vector  $\Phi_f$  is given by  $\nabla \Phi_b S_b$ , in instead of  $\nabla \Phi_b \mathbf{n}_b$ . Consequently, the problem will have the

following aspect:

$$\left[ \begin{array}{cccccc} 1 & (x_1 - x_f) & (y_1 - y_f) & (x_1 - x_f)^2 & (y_1 - y_f)^2 & \dots \\ 1 & (x_2 - x_f) & (y_2 - y_f) & (x_2 - x_f)^2 & (y_2 - y_f)^2 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \\ 0 & S_{b_x} & S_{b_y} & 2(x_b - x_f)S_{b_x} & 2(y_n - y_f)S_{b_y} & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{array} \right] \left[ \begin{array}{c} C_1 \\ C_2 \\ \vdots \\ C_{n_{coefs}} \end{array} \right] = \left[ \begin{array}{c} \phi_1 \\ \phi_2 \\ \vdots \\ \nabla \Phi_b \mathbf{S}_b \\ \vdots \end{array} \right], \quad (4.10)$$

where the line in the matrix separates the contribution of the stencil cells and Dirichlet faces from the contribution of the Neumann faces included in the stencil.

The goal of this operation is to ensure that the vector  $\phi_s$  and each column of the matrix  $\mathbf{D}_f$  have equal dimensions to each other, something that does not happen with the first approach, the classic one, see equation (3.29), and thus making the WLS problem more inconsistent. Since, the dimension of each component of the WLS problem is the same, like already happens for the reconstructions without influence of Neumann boundary faces, the error that will result from the application of the WLS method to the resolution of the Poisson equation is dimensionless making the solution of the problem more consistent. This approach will be designed as dimensional correction ( $DC_N$ ) for Neumann boundary condition.

### 4.3.1 Imposed Not-Null Flux Case

Numerical tests were performed for two grid types: polyhedral and triangular. The analytical solution is given by:

$$\phi(x, y) = \sin(3\pi x) \sin(3\pi y), \quad (4.11)$$

where in the Neumann boundaries the flux through that faces will be  $\nabla \phi_b \cdot \mathbf{S}_b \neq 0$ .

Figure 4.4 shows the convergence curves to both approaches applied for all schemes with the irregular polyhedral and triangular grids, the solid line represents the  $DC_N$  approach and the dotted one represents the GC approach. The results point out that the theoretical convergence order is always achieved for both grids and indicates that the  $DC_N$  approach improves the schemes' performance, being more evident for the FLS2 and for FLS8 schemes, specially to the last one. The behaviour of the finest grids are more stable with the  $DC_N$ .

Table 4.16: Comparison between the two approaches for a problem with an imposed not-null Neumann BC for all schemes applied to the irregular polyhedral grids.

$n_{cells}$	$h_{ref}$	Polyhedral		FLS2		FLS4		FLS6		FLS8	
		$r_1$	$r_\infty$	$r_1$	$r_\infty$	$r_1$	$r_\infty$	$r_1$	$r_\infty$	$r_1$	$r_\infty$
145	8.31E-02	1.12	1.65	0.97	1.40	0.92	1.48	1.11	0.96		
545	4.34E-02	1.11	1.84	1.06	1.04	1.07	1.02	1.24	1.26		
2113	2.22E-02	1.04	1.63	1.15	1.39	1.08	1.45	1.13	2.81		
8321	1.12E-02	0.99	1.04	1.08	1.16	1.04	1.08	21.57	74.12		
33024	5.65E-03	0.98	1.10	1.04	1.12	1.02	1.04	178.69	261.82		

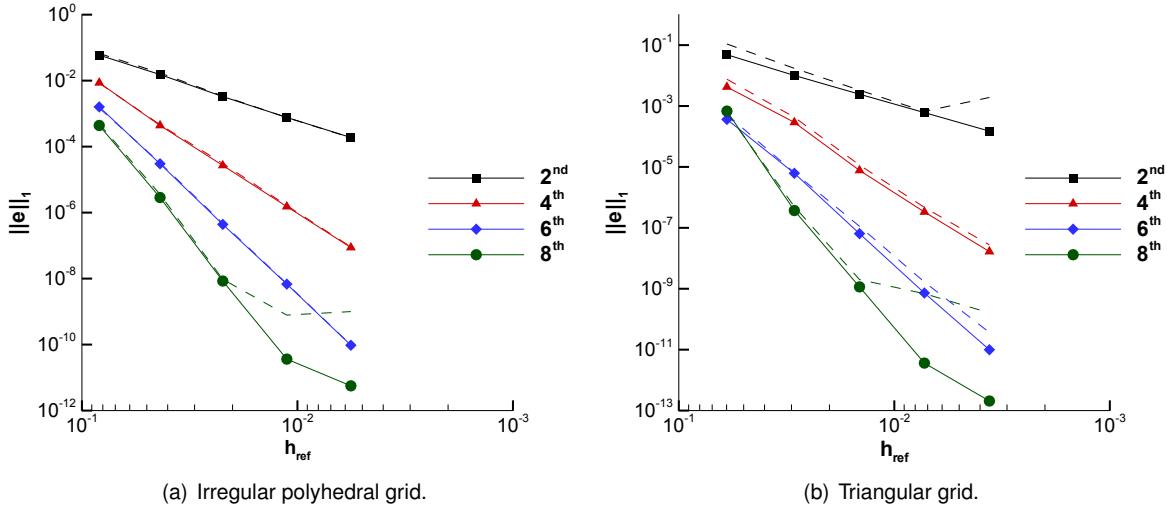


Figure 4.4: Convergence curves of the mean error for mixed boundary conditions with irregular polyhedral and triangular grids and all schemes with an imposed not-null flux. The dotted lines are the convergence curves for the GC approach and the solid ones represent the convergence curves with the DC<sub>N</sub> approach.

Table 4.17: Comparison between the two approaches for a problem with an imposed not-null Neumann BC for all schemes applied to the triangular grids.

Triangular		FLS2		FLS4		FLS6		FLS8	
$n_{cells}$	$h_{ref}$	$r_1$	$r_\infty$	$r_1$	$r_\infty$	$r_1$	$r_\infty$	$r_1$	$r_\infty$
211	5.98E-02	2.24	2.85	1.80	1.51	1.44	1.25	0.88	0.91
899	2.91E-02	1.70	4.08	1.48	1.74	1.05	0.89	1.35	0.53
3638	1.45E-02	1.36	7.47	1.49	1.94	1.73	0.94	1.74	1.98
14632	7.25E-03	1.10	7.91	1.42	1.68	2.30	0.95	190.13	297.64
58698	3.62E-03	12.73	357.14	1.65	1.73	3.70	1.44	790.73	915.16

Table 4.16 lists the comparison of the two approaches used for the Neumann BC for the irregular polyhedral grid, the comparison is made through the calculation of the ratio between both approaches used for the mean and maximum error norm, being that ratio,  $r$ , calculated by the following expression:

$$r_i = \frac{\|e\|_i^{GC}}{\|e\|_i^{DC_N}}, \quad (4.12)$$

where the  $i$  is the error norm used for the calculation. The results show that the biggest decrease of the error occurs for the maximum error. For the FLS8 scheme the error can be reduced up to 21 times.

Table 4.17 lists the comparison between the two approaches used for the Neumann BC for the triangular grid. The results obtained allow to conclude that the major decrease of the numerical error occurs for the maximum error, which can be reduced almost one order of magnitude for the second-order scheme and to half with the fourth-order scheme. For the sixth-order scheme the maximum error with this new approach is slightly worse, almost 10%, than the general approach, however in terms of mean error the gain is evident with the  $DC_N$ , where the mean error is reduced to half of the obtained with the  $GC$  approach. For the eight-order scheme, it is possible to reduce the magnitude of the error in about 3 orders of magnitude.

### 4.3.2 Imposed Null Flux Case

Identical numerical tests were performed for the imposed null flux case with the analytical solution being given by:

$$\phi(x, y) = \cos(3\pi x) \sin(2\pi y), \quad (4.13)$$

where in the Neumann boundaries the flux through that faces will be  $\nabla\phi_b \cdot \mathbf{S}_b = 0$ .

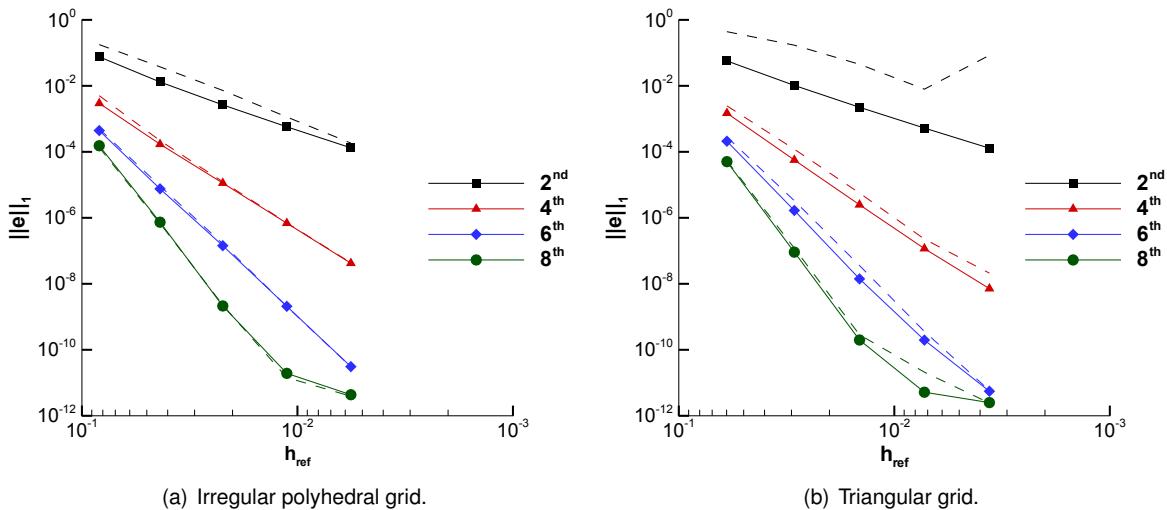


Figure 4.5: Convergence curves of the mean error for mixed boundary conditions with irregular polyhedral and triangular grids for all schemes with an imposed null flux. The dotted lines are the convergence curves for the  $GC$  approach and the solid ones represent the convergence curves with the  $DC_N$  approach.

Figure 4.5 shows the convergence curves of all schemes with the two approaches used for Neumann BC applied to polyhedral and triangular grids. Once again the solid lines correspond to the new approach proposed in this Thesis and the dotted ones are the general approach. The results allow to verify that the theoretical convergence order is always achieved for both approaches. There is an indication that DC<sub>N</sub> approach is advantageous, since the magnitude of the error decreases with this approach, being more evident in the second-order for the triangular grid, see figure 4.5(b).

Table 4.18: Comparison between the two approaches for a problem with an imposed null flux Neumann BC for all schemes applied to the irregular polyhedral grids.

Polyhedral		FLS2		FLS4		FLS6		FLS8	
$n_{cells}$	$h_{ref}$	$r_1$	$r_\infty$	$r_1$	$r_\infty$	$r_1$	$r_\infty$	$r_1$	$r_\infty$
145	8.31E-02	2.36	2.37	1.71	1.81	1.13	1.01	0.89	1.28
545	4.34E-02	2.87	3.18	1.29	1.24	1.23	1.07	0.92	1.06
2113	2.22E-02	2.74	3.86	1.10	1.29	1.14	0.89	1.06	1.00
8321	1.12E-02	1.98	4.40	1.02	1.87	0.99	0.93	0.75	1.00
33024	5.65E-03	1.39	4.57	1.03	1.99	0.99	0.87	0.90	0.74

Table 4.19: Comparison between the two approaches for a problem with an imposed null flux Neumann BC for all schemes applied to triangular grids.

Triangular		FLS2		FLS4		FLS6		FLS8	
$n_{cells}$	$h_{ref}$	$r_1$	$r_\infty$	$r_1$	$r_\infty$	$r_1$	$r_\infty$	$r_1$	$r_\infty$
211	5.98E-02	7.71	7.53	1.66	2.04	1.34	1.25	1.07	0.97
899	2.91E-02	16.56	21.23	2.13	2.07	1.99	2.17	1.32	0.97
3638	1.45E-02	20.05	41.20	2.26	2.45	2.55	2.12	1.44	1.27
14632	7.25E-03	15.09	55.96	1.94	2.48	1.83	2.33	4.05	2.04
58698	3.62E-03	655.59	3594.78	2.93	3.47	1.05	2.74	0.97	0.46

Table 4.18 lists the comparison between the two approaches used in the study of the Poisson problem with Neumann BC applied to a polyhedral grid. The results suggest that both approaches produce error norms with the same order of magnitude. For the FLS2 scheme, this new approach allows a reduction of the numerical error up two times than the one obtained with the general approach. The FLS4 and FLS6 schemes produce better results with this new approach, being the improvement about 10%. For the FLS8 scheme, the difference between both approaches are negligible.

Table 4.19 lists the comparison of the two approaches for all schemes's order with a triangular grid. With this new approach, the numerical error decreases substantially, being more evident for the FLS2 scheme where there is a decrease of the error magnitude of about one order of magnitude. For the remaining schemes the decrease of the numerical error is half of the one obtained with the general approach. It is possible to verify that this new approach is less sensitive to the influence of grid type.

Comparing this case with the one with full Dirichlet boundary conditions, it is possible to observe that the results are similar in some cases. For the FLS2 scheme, the Neumann boundary conditions are not so efficient to solve the Poisson problem, for the remaining schemes the difference between boundary conditions is around 25% of the mean error norm. The maximum error norm obtained with full Dirichlet boundary conditions is half of the one obtained with Neumann boundary conditions.

A more detailed observation of these results, for both imposed null and not-null fluxes cases, show a saturation of the solver around  $1.00E - 13$  for the sixth and eighth-order schemes, this situation was already noticed by Boullaras et al., [32], when using trigonometric functions as the analytical solution of the Poisson equation.

## Final Remarks

The results here presented allow to conclude that the new proposed approach for the Neumann boundary conditions improves the quality of the solution, reason why, it should be used with this kind of boundaries. These results are the expected ones, since in the WLS problem the dimensions of the matrices are now identical to each other, which will lead to a well posed problems.

One of the advantages of this new approach is that the solution of the problem is less sensitive to the influence of the grid type. This can be observed by figure 4.5, where is evident that for an imposed null flux the solution of the new approach is more stabilized for both grid types tested.

## 4.4 Influence of Grid Quality Parameters

In this section a grid quality study was performed based on two parameters: the non-orthogonality angle and the volume ratio, in order to find which parameter mostly affects the accuracy of the numerical solution obtained by the proposed schemes.

### 4.4.1 Influence of Non-Orthogonality Angle

The influence of the non-orthogonality angle was described by F. Juretić and D. Gosman [84], and in this section, tests with an imposed non-orthogonality angle,  $\alpha$ , of the the grid are performed. The grid type that results from the imposed  $\alpha$  has already been explained in the previous chapter. For these tests, the same analytical solution of the influence of grid type study was used, given by equation (4.8).

Table 4.20:  $\|e\|_1$  and  $\mathcal{O}_1$  of the FLS4 scheme with the twisted grids.

$h_{ref}$	$0^\circ$		$10^\circ$		$20^\circ$		$30^\circ$		$40^\circ$	
	$\ e\ _1$	$\mathcal{O}_1$	$\ e\ _1$	$\mathcal{O}_1$	$\ e\ _1$	$\mathcal{O}_1$	$\ e\ _1$	$\mathcal{O}_1$	$\ e\ _1$	$\mathcal{O}_1$
1.25E-02	2.38E-06	3.96	3.22E-06	3.96	5.36E-06	3.93	8.98E-06	3.89	1.84E-05	3.87
6.25E-03	1.50E-07	3.99	2.04E-07	3.99	3.43E-07	3.99	5.78E-07	3.99	1.19E-06	4.00
3.13E-03	9.36E-09	4.00	1.28E-08	4.01	2.15E-08	4.00	3.64E-08	4.00	7.51E-08	4.01

Table 4.21:  $\|e\|_1$  and  $\mathcal{O}_1$  of the FLS8 scheme with the twisted grids.

$h_{ref}$	$0^\circ$		$10^\circ$		$20^\circ$		$30^\circ$		$40^\circ$	
	$\ e\ _1$	$\mathcal{O}_1$	$\ e\ _1$	$\mathcal{O}_1$	$\ e\ _1$	$\mathcal{O}_1$	$\ e\ _1$	$\mathcal{O}_1$	$\ e\ _1$	$\mathcal{O}_1$
1.25E-02	4.30E-09	8.18	7.38E-09	7.74	9.34E-09	7.71	1.28E-08	7.59	5.98E-08	7.34
6.25E-03	1.62E-11	8.05	3.07E-11	7.94	3.98E-11	7.92	5.58E-11	7.90	2.77E-10	7.85
3.13E-03	6.45E-14	7.98	1.23E-13	7.97	1.62E-13	7.96	2.24E-13	7.99	1.13E-12	7.99

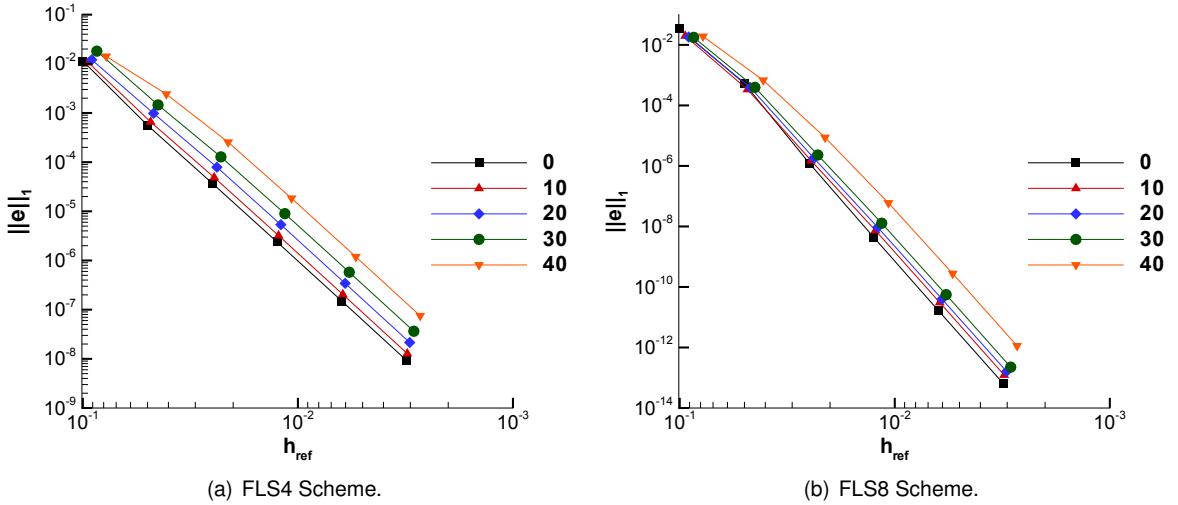


Figure 4.6: Convergence curve of the  $\|e\|_1$  for FLS4 and FLS8 schemes with the twisted grids.

From figure 4.6, it is possible to see that the effect of non-orthogonality angle is a loss of accuracy on the results. Since the convergence curves are parallel to each other, this means that the convergence order is almost constant whatever the imposed non-orthogonality angle, this suggests that the convergence order remains near to the theoretical one, since the convergence order was already been show for the Cartesian grid in figures 4.2(a) and 4.2(b). It is also possible to observe that the effect of a twisted grid for the FLS8 scheme is more pronounced for a non-orthogonality angle of  $40^\circ$ , once the results with an angle between  $10^\circ$  and  $30^\circ$  does not create many differences in the accuracy of the scheme. Tables 4.20 and 4.21 list the  $\|e\|_1$  mean error and the respective convergence order for the three finest grids used in this study for the FLS4 and FLS8 schemes and corroborate with the figure 4.6.

Table 4.22: Ratio of mean and maximum errors for all schemes between an imposed  $\alpha$  grid and the Cartesian one with 102400 cells.

$\alpha$	FLS2		FLS4		FLS6		FLS8	
	$r_1$	$r_\infty$	$r_1$	$r_\infty$	$r_1$	$r_\infty$	$r_1$	$r_\infty$
10	1.01	1.00	1.37	0.89	2.06	2.85	1.91	1.62
20	1.05	1.01	2.30	1.42	3.17	4.25	2.51	1.93
30	1.16	1.03	3.89	2.08	4.24	5.33	3.48	2.39
40	1.28	1.04	8.02	4.29	11.68	15.60	17.55	12.92

Table 4.22 lists the values of the ratios of the mean and maximum errors norms for all schemes between a grid with a imposed  $\alpha$  and the finest Cartesian one, these error ratio,  $r$ , can be calculated through the following expression:

$$r_i = \frac{\|e\|_i^\alpha}{\|e\|_i^0}, \quad (4.14)$$

where  $\alpha$  is the imposed non-orthogonality angle of the grid and  $i$  is the error norm considered.

The difference between errors could be almost of one order of magnitude for the FLS4 scheme and almost two orders of magnitude for FLS8 scheme when high non-orthogonality angle is applied on the grid. The table allows to conclude that with an imposed  $\alpha$  bellow  $30^\circ$  does not have a significant effect

in the numerical solution accuracy.

Figure 4.7 shows the error distribution for the FLS8 scheme with a non-orthogonality angle of  $0^\circ$  and  $40^\circ$ . It is possible to observe that the error distribution takes the form of the grid, which means that the error distribution rotates an angle identical to the non-orthogonality one.

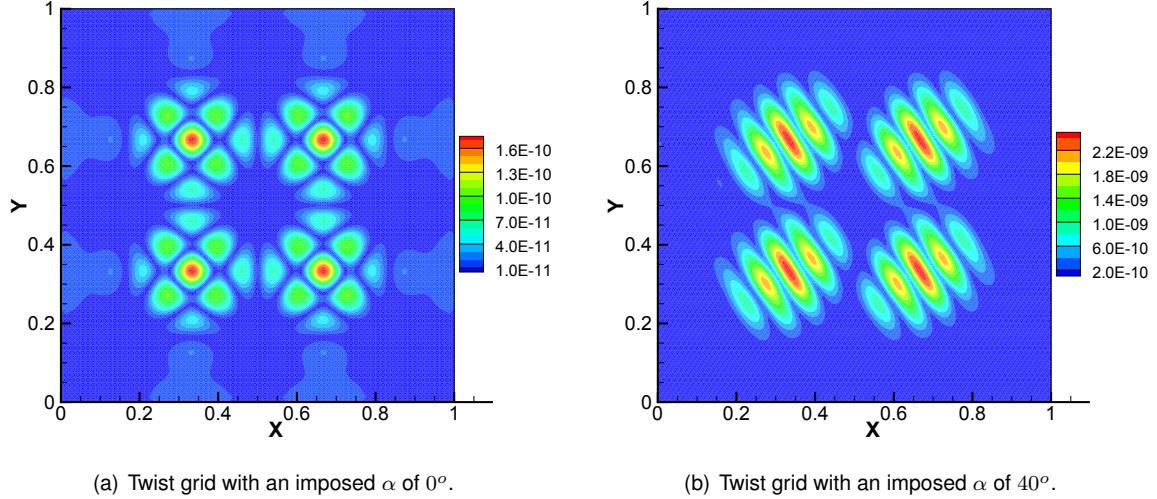


Figure 4.7: Error distribution for FLS8 scheme for two different  $\alpha$ .

#### 4.4.2 Influence of Volume Ratio

In this section, it was performed numerical tests in grids with an imposed volume ratio. The same analytical solution of the selection of the weight function section was chosen, which is given by equation (4.4). This grid type was already explained in the previous chapter, being known as perturbed grid.

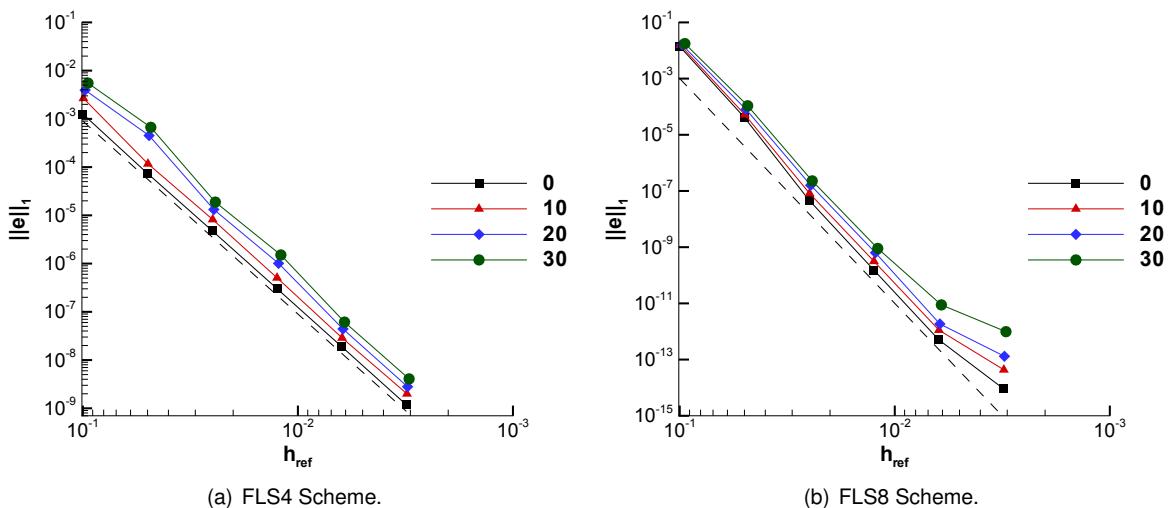


Figure 4.8: Convergence curves for different imposed perturbed grids with FLS4 and FLS8 schemes.

Table 4.23:  $\|e\|_1$  and  $\mathcal{O}_1$  for FLS4 scheme with perturbed grids.

$h_{ref}$	0%		10%		20%		30%	
	$\ e\ _1$	$\mathcal{O}_1$						
1.25E-02	3.02E-07	3.98	5.07E-07	4.02	1.01E-06	3.70	1.51E-06	3.61
6.25E-03	1.89E-08	4.00	2.88E-08	4.14	4.41E-08	4.54	6.13E-08	4.68
3.13E-03	1.18E-09	4.00	2.00E-09	3.84	2.80E-09	3.98	4.07E-09	3.92

Table 4.24:  $\|e\|_1$  and  $\mathcal{O}_1$  for FLS8 scheme with perturbed grids.

$h_{ref}$	0%		10%		20%		30%	
	$\ e\ _1$	$\mathcal{O}_1$						
2.50E-02	4.66E-08	9.78	8.19E-08	9.33	1.60E-07	8.99	2.29E-07	8.91
1.25E-02	1.53E-10	8.25	3.15E-10	8.02	6.34E-10	7.95	9.02E-10	7.93
6.25E-03	5.11E-13	8.22	1.12E-12	8.13	1.85E-12	8.46	8.84E-12	6.75

Tables 4.23 and 4.24 lists the values of the mean error norms and the respective convergence order for perturbed grids with the FLS4 and FLS8 schemes, respectively. It is possible to verify that these grids produce inconstant results when compared with a Cartesian one. The behaviour of the maximum error norm is more sensitive to the imposed volume ratio than the mean error norm, although the results for maximum error norm are not shown in this work.

Figure 4.8 shows the convergence curves for this grid type with the FLS4 and FLS8 schemes. It is possible to observe, in this figure, that the theoretical convergence orders of the schemes are achieved for every perturbed grid, however with the FLS8 scheme there is a case that the asymptotic convergence is difficult to be obtained, which is also listed in table 4.24.

Table 4.25: Ratio of mean and maximum error norms for all schemes between grids with an imposed displacement and a Cartesian one with 25600 cells.

$\gamma\%$	FLS2		FLS4		FLS6		FLS8	
	$r_1$	$r_\infty$	$r_1$	$r_\infty$	$r_1$	$r_\infty$	$r_1$	$r_\infty$
10	1.01	1.54	1.52	4.44	1.86	10.35	2.20	8.19
20	1.09	1.55	2.33	4.72	2.81	15.15	3.62	11.05
30	1.22	1.78	3.24	7.53	3.89	22.30	17.28	30.61

Table 4.25 lists the values of the ratios,  $r$ , between a grid with an imposed displacement and a regular one for the mean and maximum error norms of the finest grid. This ratio can be calculated with the equation (4.14) where  $\alpha$  should be replaced by  $\gamma$ . The difference between the volume ratio is more pronounced for the maximum error norm. Particularly for the FLS6 and FLS8 schemes, the error could be one order of magnitude greater than the obtained with the Cartesian grid. It is also shown that an imposed displacement up to 20% has a low numerical error penalization.

Figure 4.9 shows the error distribution for the FLS8 scheme with a grid perturbation ranging from 0% to 30%. It is shown that the error distribution is severely changed by the imposed perturbation on the grid, see figure 4.9(b), which means that this parameter has more influence in the results than to the non-orthogonality angle.

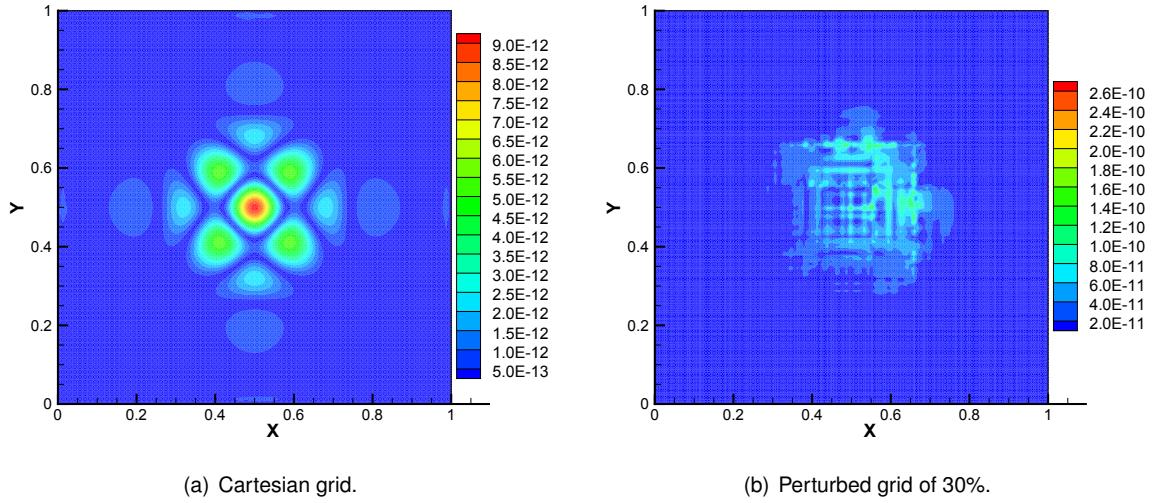


Figure 4.9: Error distribution for FLS8 scheme for two different  $\gamma$  with a grid with 25600 cells.

## Final Remarks

The results obtained with this two grid parameters show that the convergence order of the schemes are independent form both imposed non-orthogonality angle and volume ratio, despite the increasing numerical error for each successive higher  $\alpha$  or  $\gamma$ . It is also shown that the scheme's accuracy is more sensitive to volume ratios than to the non-orthogonality angle, this can be observed through tables 4.22 and 4.25. Finally for an imposed non-orthogonality angle lower than  $30^0$  and an imposed displacement lower than 20% the solution's accuracy is not too penalized, specially for the FLS8 scheme.

## 4.5 Efficiency Remarks

In this section will be presented the results of the efficiency study performed to the proposed schemes. As analytical solution, the equation given by (4.8) was used. This study can be divided into two parts, the first one relates the error level as function of the non-zero entries of the global matrix, see Lipnikov and Manzini, [13], and the second one relates the error level with respect to the solver-run time, see Gooch, [88].

#### 4.5.1 Number of Non Zeros (NNZ)

A considerable reduction of the error norms was obtained for the very high-order scheme, however the stencil is large and increases the required computational cost consequently, the efficiency is a prime concern of the high-order numerical schemes. Denoting by  $N_z$  the number of non zero entries in the global matrix, Lipnikov and Manzini, [13], have reported that the error norm will decay as  $\mathcal{O}(h^q) \sim \mathcal{O}(N_z^{q/D})$ , where  $D$  is the space dimension. So for a two dimensional space, the error norm should

decay  $q/2$  with  $N_z$ , and the order order of convergence can be calculated by:

$$\mathcal{O}_n^z = \frac{\log_{10} \|e\|_{n_1} - \log_{10} \|e\|_{n_2}}{\log_{10} N_{z_1} - \log_{10} N_{z_2}}, \quad (4.15)$$

where the superscript  $z$  means that the convergence order is related with  $N_z$ , which allows to estimate the amount of memory required for a given error level.

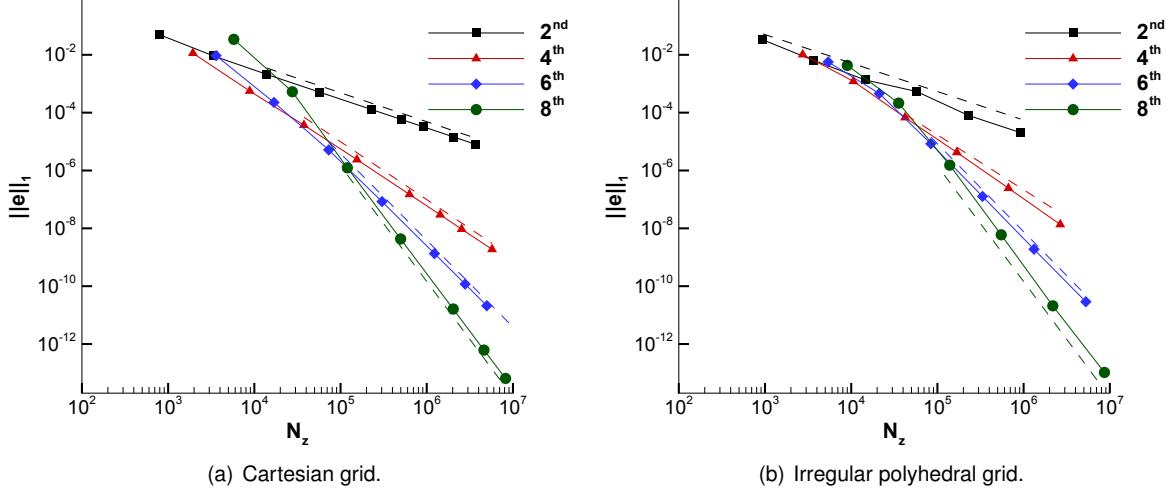


Figure 4.10: Convergence curves of error 1-norm with respect to  $N_z$  with the Cartesian and polyhedral grids for all schemes, where the dotted line represents the theoretical slope of each scheme.

Table 4.26:  $\mathcal{O}_1^z$  and  $\|e\|_1$  for Cartesian grid for FLS2, FLS4 and FLS8.

$n_{cells}$	FLS2			FLS4			FLS8		
	$N_z$	$\ e\ _1$	$\mathcal{O}_1^z$	$N_z$	$\ e\ _1$	$\mathcal{O}_1^z$	$N_z$	$\ e\ _1$	$\mathcal{O}_1^z$
25600	2.28E+05	1.29E-04	1.00	6.30E+05	1.50E-07	1.97	2.03E+06	1.62E-11	3.97
57600	5.16E+05	5.72E-05	1.00	1.43E+06	2.96E-08	1.99	4.61E+06	6.17E-13	4.00
102400	9.18E+05	3.22E-05	1.00	2.54E+06	9.36E-09	1.99	8.21E+06	6.45E-14	3.90

Table 4.27:  $\mathcal{O}_1^z$  and  $\|e\|_1$  for irregular polyhedral grid for FLS2, FLS4 and FLS8.

$n_{cells}$	FLS2			FLS4			FLS8		
	$N_z$	$\ e\ _1$	$\mathcal{O}_1^z$	$N_z$	$\ e\ _1$	$\mathcal{O}_1^z$	$N_z$	$\ e\ _1$	$\mathcal{O}_1^z$
8321	5.77E+04	5.35E-04	0.68	1.68E+05	4.23E-06	2.01	5.50E+05	5.97E-09	4.04
33024	2.30E+05	7.97E-05	1.38	6.67E+05	2.45E-07	2.07	2.18E+06	2.09E-11	4.11
131585	9.19E+05	2.04E-05	0.98	2.66E+06	1.36E-08	2.09	8.66E+06	1.04E-13	3.84

Table 4.26 lists the error  $\|e\|_1$  decay for different  $N_z$  values and the respective convergence order,  $\mathcal{O}_1^z$ , for the second, fourth and eighth-order schemes on Cartesian grids. Figure 4.10(a) shows the results of  $\|e\|_1$  with respect to  $N_z$  for all schemes with the respective theoretical convergence order being achieved for every scheme.

Table 4.27 lists the error 1-norm as a function of  $N_z$  and  $\mathcal{O}_1^z$  for FLS2, FLS4 and FLS8 schemes in the last three consecutive refinements of the irregular polyhedral grid. Figure 4.10(b) shows that the

effective values for the convergence order are in close agreement with the theoretical values.

These results support that for the same level of error a high-order method can be more efficient than second-order one, because it would require less computational resources for the same accuracy level.

Table 4.28: Comparison of the required computational resources for all orders with a 1-norm level error of 4.30E-09 for Cartesian grid and 5.97E-09 for irregular polyhedral one, where 1Gb = 1024Mb.

Scheme	Cartesian Grid $\ e\ _1 = 4.03E - 09$				Irregular Polyhedral Grid $\ e\ _1 = 5.97E - 09$			
	$N_z$	$n_{cells}$	size (A)	Ratio	$N_z$	$n_{cells}$	size (A)	Ratio
FLS8	4.98E+05	6.40E+03	5.70Mb	1.00	5.50E+05	8.32E+03	6.30Mb	1.00
FLS6	8.30E+05	1.76E+04	9.49Mb	1.66	9.08E+05	2.27E+04	10.40Mb	1.65
FLS4	3.77E+06	1.52E+05	43.17Mb	7.57	3.94E+06	1.95E+05	45.10Mb	7.16
FLS2	6.95E+09	7.74E+08	77.63Gb	13946.16	3.59E+09	5.13E+08	40.07Gb	6512.97

To compare the performance of the proposed schemes, a Cartesian grid with 1600 cells and a polyhedral one with 8321 cells were chosen. The value of the error 1-norm of the FLS8 scheme for both grids was taken as the reference value. For the other schemes, the required memory was estimated by extrapolating the  $N_z$  from the previous results. Table 4.28 lists the required computational resources to obtain a determined error level. The size of the global matrix is estimated by multiplying  $N_z$  with 12 bytes, which 8 bytes are necessary to allocate the value and the remaining 4 bytes are to allocate the position in the global matrix. The ratio is obtained by dividing the size of the matrix of the considered scheme by the size of the matrix for FLS8 scheme. The results show, for example, that to obtain the intended accuracy the FLS8 scheme uses three to four order of magnitude less computational resources than the FLS2, 7 times less than the FLS4 and almost half of the space that is needed with the FLS6. These results allow to conclude that the very high-order schemes will allow to obtain numerical solutions with higher accuracy than the second and fourth-order methods, for the same computational resources.

### 4.5.2 Solver-Run Time (SRT)

Another way to analyse the method efficiency is through the computing time that the solver requires to achieve a converged solution, see e.g. Nejat and Gooch, [88]. For this study the BICGSTAB solver was used, as the analytical solution equation (4.8) was used. Figure 4.11(a) and 4.11(b) show the SRT for the different schemes with the Cartesian and irregular polyhedral grids, respectively. Figure 4.11 shows the mean error norm with respect to the computing time required for convergence, several grids were used and enumerated. For a certain level of error the FLS8 scheme displays the fastest time, being the fastest scheme for both grid types.

Table 4.29:  $\|e\|_1$  and  $\mathcal{O}_1^{SRT}$  for Cartesian grid for FLS2, FLS4 and FLS8.

$n_{cells}$	FLS2			FLS4			FLS8		
	SRT	$\ e\ _1$	$\mathcal{O}_1^{SRT}$	SRT	$\ e\ _1$	$\mathcal{O}_1^{SRT}$	SRT	$\ e\ _1$	$\mathcal{O}_1^{SRT}$
25600	4.45E-01	1.29E-04	0.72	9.52E-01	1.50E-07	1.39	2.63E+00	1.62E-11	2.96
57600	1.64E+00	5.72E-05	0.62	3.24E+00	2.96E-08	1.32	7.68E+00	6.17E-13	3.05
102400	3.93E+00	3.22E-05	0.66	7.21E+00	9.36E-09	1.44	1.65E+01	6.45E-14	2.96

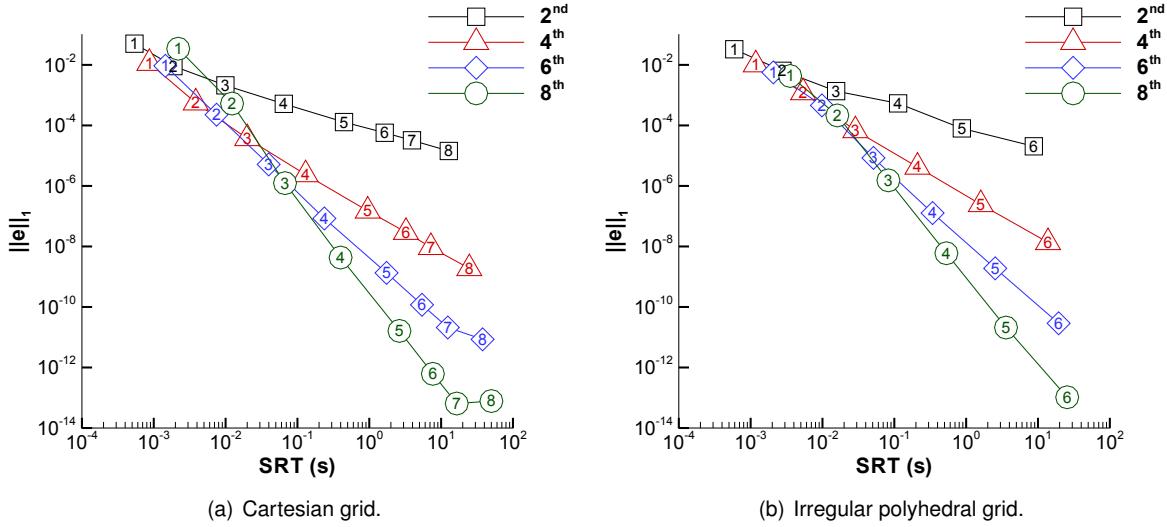


Figure 4.11: Mean error with SRT diagram for the proposed schemes for Cartesian and polyhedral grids.

Table 4.30:  $\|e\|_1$  and  $\mathcal{O}_1^{SRT}$  for irregular polyhedral grid for FLS2, FLS4 and FLS8.

$n_{cells}$	FLS2			FLS4			FLS8		
	SRT	$\ e\ _1$	$\mathcal{O}_1^{SRT}$	SRT	$\ e\ _1$	$\mathcal{O}_1^{SRT}$	SRT	$\ e\ _1$	$\mathcal{O}_1^{SRT}$
8321	1.13E-01	5.35E-04	0.47	2.10E-01	4.23E-06	1.39	5.31E-01	5.97E-09	2.98
33024	8.71E-01	7.97E-05	0.93	1.60E+00	2.45E-07	1.40	3.59E+00	2.09E-11	2.96
131585	8.75E+00	2.04E-05	0.59	1.38E+01	1.36E-08	1.34	2.55E+01	1.04E-13	2.71

Tables 4.29 and 4.30 list the results of the error and the convergence order as function of the SRT criteria with the Cartesian and polyhedral grids, respectively, and they suggest that error will decay approximately with 1/3 power law,  $\mathcal{O}(h^q) \sim \mathcal{O}((t_{solver})^{q/3})$ .

Table 4.31: Comparison of the SRT for all orders with a 1-norm level error of 4.30E-09 for Cartesian grid and 5.97E-09 for irregular polyhedral grid.

Scheme	Cartesian Grid			Irregular Polyhedral Grid		
	$n_{cells}$	$\ e\ _1 = 4.03E - 09$	Ratio	$n_{cells}$	$\ e\ _1 = 5.97E - 09$	Ratio
FLS8	6.40E+03	3.99E-01	1.00	8.32E+03	5.31E-01	1.00
FLS6	1.76E+04	9.94E-01	2.49	2.27E+04	1.47E+00	2.76
FLS4	1.52E+05	1.31E+01	32.84	1.95E+05	2.56E+01	48.17
FLS2	7.74E+08	1.38E+07	3.45E+07	5.13E+08	8.49E+06	1.60E+07

To obtain a mean error level of 5.00E – 09, the SRT for FLS8 scheme is about half, 30 times and 10 millions times less SRT than the obtained with the FLS6, FLS4 and FLS2 schemes, respectively, for both Cartesian and polyhedral grids, see table 4.31.

### 4.5.3 Efficiency Variation with Grid Type

In this section a comparison between different grid types and scheme's orders is performed, in order to conclude about what is the choice that optimizes both memory and computing time required to achieve

a converged solution. Figure 4.12 shows the behaviour of the  $\|e\|_1$  with respect to  $N_z$  and SRT for the different grid types tested with the FLS8 scheme, which has already been proved, previously, that is the more efficient scheme.

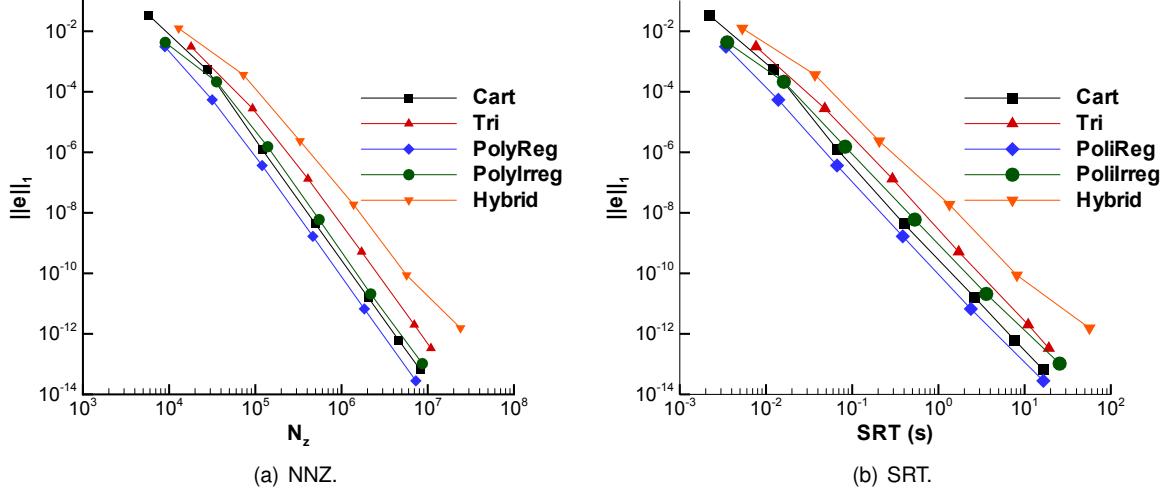


Figure 4.12: Convergence curves of the error 1-norm with respect to  $N_z$  (a) and SRT (b) for different grids types with the FLS8 scheme.

Figure 4.12(a) suggests that polyhedral grids optimize the computational resources, since the regular grid is the one that has the best behaviour and the irregular one has a similar behaviour to the Cartesian grid for the same error level, being the best regular and irregular grid type. Figure 4.12(b) refers to the SRT and it is possible to affirm that the regular polyhedral grid allows to obtain accurate solutions faster than the other grids. The triangular grid reports the worse behaviour, fact already expected due to size of the matrix.

Table 4.32: Computational Resources required for a  $\|e\|_1$  of 1.00E-10 for all schemes and grid types, where  $1\text{Tb} = 1024^2\text{Mb}$ .

Grid	FLS2			FLS4			FLS6			FLS8		
	$N_z$	size (A)	$\frac{N_z}{n_{cells}}$	$N_z$	size (A)	$\frac{N_z}{n_{cells}}$	$N_z$	size (A)	$\frac{N_z}{n_{cells}}$	$N_z$	size (A)	$\frac{N_z}{n_{cells}}$
Cart	3.00E+11	3.27Tb	8.98	2.54E+07	290.60Mb	24.88	2.95E+06	33.73Mb	48.71	1.29E+06	14.72Mb	80.48
Tri	6.16E+11	6.73Tb	12.93	2.91E+07	333.50Mb	36.67	4.08E+06	46.73Mb	72.09	2.59E+06	29.59Mb	119.27
PolyReg	1.62E+11	1.77Tb	6.32	1.89E+07	216.26Mb	17.19	2.06E+06	23.53Mb	33.53	9.37E+05	10.72Mb	55.35
Polylreg	2.29E+11	2.49Tb	6.98	2.79E+07	319.22Mb	20.19	3.49E+06	39.99Mb	40.04	1.49E+06	17.04Mb	65.83
Hybrid	9.17E+11	10.00Tb	11.98	8.45E+07	1012.36Mb	35.02	1.09E+07	125.12Mb	70.67	5.49E+06	62.78Mb	119.40

Table 4.32 lists the computational resources required for an error 1-norm level of 1.00E – 10 for all schemes and grid types, where it is estimated the  $N_z$ , the size of the global matrix and the number of non zero entries per cell on the global matrix. Through this table, it is possible to make a comparison between the different schemes' order and grids, in order to determine what is the choice that minimizes the computational resources. For every scheme the regular polyhedral grid is the one that has the lowest computational cost and the triangular one distances itself from the others grid types with the worse behaviour. The triangular grid type uses the biggest stencils, since one vertex of this grid type is connected with many cells, which is the main cause of the large stencils consequently, they require

more memory.

Table 4.33: Solver-Run time required for a  $\|e\|_1$  of 1.00E-10 for all schemes and grid types, in seconds.

Grid	FLS2		FLS4		FLS6		FLS8	
	SRT (s)	Ratio						
Cart	8.62E+09	3.80	2.35E+02	1.61	5.86E+00	1.46	1.42E+00	1.46
Tri	2.27E+09	1.00	1.46E+02	1.00	7.04E+00	1.75	2.98E+00	3.05
PolyReg	1.50E+10	6.63	1.80E+02	1.23	4.02E+00	1.00	9.77E-01	1.00
PolyIrreg	8.65E+09	3.81	5.41E+02	3.71	1.06E+01	2.63	2.12E+00	2.17
Hybrid	2.23E+10	9.81	1.28E+03	8.81	2.97E+01	7.39	7.76E+00	7.95

Table 4.33 list the estimated computing time required for an error 1-norm level of 1.00E – 10 for all schemes and grid types, it was done a study, like the one performed in table 4.32 for the required memory. This table shows that there is no standard behaviour for this parameter, and suggests that for the schemes with a order lower than the fourth the best choice is the triangular grid and for the schemes with an order higher than sixth the regular polyhedral one should be chosen. It is possible to verify that the behaviour of the triangular grid gets worse with the increase of the scheme's order and in the other hand the regular polyhedral one has the opposite behaviour.

The combination that minimizes the SRT criteria is the regular polyhedral grid applied to the FLS8 scheme, which is possible to have a solution in two thirds times less than the Cartesian one. Comparing with the triangular grid, one of the most used unstructured grid, the polyhedral grids spends one third less time than the triangular one with the FLS8 scheme.

### Final Remarks

The major conclusion of this parametric study is that the polyhedral grid type is the one that requires less required memory and SRT criteria. With regard to the proposed schemes, the FLS8 scheme is the one that minimizes both computational resources and computing time. These results are expected, because the stencils of this grid type are smaller and more compact when compared with the ones from other grid types. Another conclusion, is that the polyhedral grids are more effective in terms of computing time for the very high-order methods, higher than sixth-order, in opposite to the triangular grids which is the best choice for the second and forth-order methods.



# Chapter 5

## Conclusions

A new finite volume very high-order scheme is proposed for the solution of the Poisson equation through the use of polynomial reconstructions. A detailed description of the construction of the global matrix is included, the scheme has also a new stencil selection technique is explained, for both interior and boundary faces to reconstruct properly the high order terms. A parametric study was performed to select the weight function of the WLS reconstruction method, which to the Author's knowledge, it is a novelty in the framework of these methods.

The presented method only requires one reconstruction per face independently of the scheme's order. This is an advantage compared with other approaches that uses a reconstruction per Gauss point. The main differences between this approach and others high-order reconstruction approaches is the point value formulation and the fact of being centered at the face centroid, this can be considered as an advantage because does not require an interpolation for the face flux that is used by the cell centered reconstructions.

Tests performed in 1D framework indicate that an implicit flux reconstruction is required, in order to maintain the high-order of the scheme independently of the existent grid perturbations. A DC approach was also tested and suggests that is only possible to obtain a fine behaviour of the proposed schemes when the scheme's order of the explicit approach is the double of the one from the implicit approach.

The tests performed allow to verify the effective convergence order of the new second, fourth, sixth and eighth-order schemes flux based on reconstruction with WLS method. In the framework of unstructured finite volume it is, on the Author's knowledge, the first time that a  $8^{th}$  effective convergence order is reported for the solution of Poisson equation with any arbitrary grid.

The weight function optimization allowed to decrease the numerical error of the meyhod up to two orders of magnitude when compared to the other weight functions from the literature. A new approach for Neumann boundary conditions is proposed and its impact in the convergence order of the scheme is shown to be similar to the theoretical one. A comparison study was performed between the general approach for Neumann boundaries and this new one, and reveals that this new approach has an error reduction of one order of magnitude. The grid quality study has shown that a non-orthogonality angle bellow  $30^0$  and a volume ratio below 20% does not affect the accuracy of the solution, namely when a

eight-order scheme is being used.

Finally, it is shown that the eighth order-scheme is the one that required less computational resources for both memory and SRT criteria when compared with the other scheme's order. In terms of grid topologies, the results suggest another major conclusion, which affirms that the polyhedral grids with the FLS8 scheme is the combination that maximizes both efficiency criteria. These results are expected ones, because the stencil of this grid type is smaller and more compact than the one obtained with other grid types.

Due to the quality and novelty of the results presented in this Thesis, it was possible to submit a scientific paper titled as "A Very High-Order Finite Volume Method Based on Weighted Least-Squares for the Solution of Poisson Equation on Polyhedral Unstructured Grids" in Computer Methods in Applied Mechanics & Engineering journal. Another one is in preparation and will be focus on stability analysis performed on the proposed high-order schemes, which will be titled as "Stability Analysis of Unsteady and Very High-Order Diffusive Schemes with the Finite Volume Method" to be submitted in Numerical Heat Transfer, Part B - Fundamentals.

## 5.1 Future Work

The very high-order diffusive schemes developed in this Thesis have great potential, since there exists several applications where they can be used. The steady state heat transfer problem and the pressure correction of CFD codes are some examples, where the use these operators are mandatory. This work can be seen as the first step to create a high order schemes research field, opening the door to several new subjects for future development in the SOL code, with the huge advantage of those options being independent of each other. In this section the Author just want to elucidate the reader about the great potential of these schemes exposing a few examples of possible developments to do in the near future.

The most obvious improvement is the extension of the proposed schemes for a 3D space domain, since the mathematical principles are exactly the same and consequently, only some modifications are needed. It would be required to compute more coefficients, and an extension for 3D of Gauss quadrature is mandatory for the source term integration.

An extension for curved boundaries is also possible, since several authors have proposed these extensions for their own methods, see e.g. Gooch et al., [15] and Boularas et al., [32]. An extension for the Author's point value formulation could be studied and compared with other ones from the literature.

A high-order discretization of the diffusion-reaction equation is another valid development, where to the Author's knowledge, it would be the first high-order space discretization of the this equation in the FV framework, since there are some approaches of fourth-order in FEM framework, see Kus, [89]. The proposed diffusive schemes can be applied to the diffusion-reaction case if an algorithm for the integration of the source term, which depends on the reconstructed  $\phi_i^R$  quantity, is added to the SOL code.

An extension for convective schemes is possible through the use of the same principles already implemented, where the convective flux can be directly obtained from the  $\phi^R$ . Once the convective

schemes are implemented, it would be possible to use this methodology for the resolution of Navier-Stokes equations, in order to simulate real CFD problems.

The study of the unsteady diffusive equation for unstructured grids is another valid option, where the starting point would be the verification of the stability analysis, already done for 1D cases. The time integration formulation at each cell would be required in order to advance the simulation. This would be a powerful tool, since the majority of engineering applications are unsteady.

The last suggestion of the Author is the development of a p-adaptive code that can select the local order based on the grid size and target error at each face. This is an interesting research field, since the high-order methods allows to compute accurate solutions with coarser grids. The advantage of the p-adaptivity is that a coarser grid can be used where the gradients are small and in opposite at strong gradients a high-order method is advisable, this way the same or similar error magnitude can be obtained over all domain.



# Bibliography

- [1] J. Muylaert, L. Walpot, H. Ottens, and F. Cipollini. Aerothermodynamic Reentry Flight Experiments-EXPERT. Technical report, European Space Agency Noordwijk (Netherlands), October 2005. Research paper with briefing charts.
- [2] Antony Jameson and Kui Ou. 50 years of transonic aircraft design. *Progress in Aerospace Sciences*, 47(5):308 – 318, 2011.
- [3] Antony Jameson, T. J. Baker, and N. P. Weatherill. Calculation of Inviscid Transonic Flow over a Complete Aircraft. In *24th AIAA Aerospace Sciences Meeting*. American Institute of Aeronautics and Astronautics, jan 1986.
- [4] Antony Jameson. Successes and challenges in computational aerodynamics. In *8th AIAA Computational Fluid Dynamics Conference*, Fluid Dynamics and Co-located Conferences. American Institute of Aeronautics and Astronautics, jun 1987.
- [5] Forrester T. Johnson, Edward N. Tinoco, and N. Jong Yu. Thirty years of development and application of {CFD} at Boeing Commercial Airplanes, Seattle. *Computers & Fluids*, 34(10):1115 – 1151, 2005.
- [6] Dimitri Mavriplis. CFD in aerospace in the new millenium. *Canadian Aeronautics and Space Journal*, 46:167 – 176, 2000.
- [7] V. Venkatakrishnan. Perspective on unstructured grid flow solvers. *AIAA Journal*, 34(3):533 – 547, Mar 1996.
- [8] F. Ronald Bailey. High-end computing challenges in aerospace design and engineering. In *Computational Fluid Dynamics 2004: Proceedings of the Third International Conference on Computational Fluid Dynamics*, pages 13 – 26. Springer Berlin Heidelberg, 2006.
- [9] Philippe R. Spalart. *Topics in Detached-Eddy Simulation*, pages 3 – 12. Springer Berlin Heidelberg, 2006.
- [10] T. H. Lê, J. M. Le Gouez, and E. Garnier. High accuracy flow simulations: Advances and challenges for future needs in aeronautics. *Computers & Fluids*, 43(1):90 – 97, 2011.
- [11] Dimitris Drikakis, Dochan Kwak, and Cetin C. Kiris. Computational aerodynamics: Advances and challenges. *The Aeronautical Journal*, 120:13–36, 2016.

- [12] D. W. Zingg, S. De Rango, M. Nemec, and T. H. Pulliam. Comparison of Several Spatial Discretizations for the Navier-Stokes Equations. *Journal of Computational Physics*, 160:683 – 704, 2000.
- [13] Konstantin Lipnikov and G. Manzini. A high-order mimetic method on unstructured polyhedral meshes for the diffusion equation. *Journal of Computational Physics*, 272:360 – 385, 2014.
- [14] Timothy Barth and Paul Frederickson. Higher order solution of the Euler equations on unstructured grids using quadratic reconstruction. In *28th Aerospace Sciences Meeting*, Aerospace Sciences Meetings. American Institute of Aeronautics and Astronautics, 1990.
- [15] Carl Ollivier-Gooch and Michael Van Altena. A High-Order-Accurate Unstructured Mesh Finite-Volume Scheme for the Advection-Diffusion Equation. *Journal of Computational Physics*, 181:729 – 752, 2002.
- [16] Amir Nejat and Carl Ollivier-Gooch. A high-order accurate unstructured finite volume Newton-Krylov algorithm for inviscid compressible flows. *Journal of Computational Physics*, 227(4):2582 – 2609, 2008.
- [17] Chris Michalak and Carl Ollivier-Gooch. Unstructured High-Order Accurate Finite-Volume Solutions of the Navier-Stokes Equations. In *47th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition*, Aerospace Sciences Meetings. American Institute of Aeronautics and Astronautics, Jan 2009.
- [18] Carl Ollivier-Gooch and Alireza Jalali. Higher-Order Finite Volume Solution Reconstruction on Highly Anisotropic Meshes. In *21st AIAA Computational Fluid Dynamics Conference*, Fluid Dynamics and Co-located Conferences. American Institute of Aeronautics and Astronautics, jun 2013.
- [19] Alireza Jalali and Carl Ollivier-Gooch. Higher-order unstructured finite volume RANS solution of turbulent compressible flows. *Computers & Fluids*, 143:32 – 47, 2017.
- [20] Luis Cueto-Felgueroso, Ignasi Colominas, J. Fe, Fermín Navarrina, and Manuel Casteleiro. High-order finite volume schemes on unstructured grids using moving least-squares reconstruction. Application to shallow water dynamics. *International Journal for Numerical Methods in Engineering*, 65:295 – 331, 2006.
- [21] Luis Cueto-Felgueroso, Ignasi Colominas, Xesús Nogueira, Fermín Navarrina, and Manuel Casteleiro. Finite volume solvers and Moving Least-Squares approximations for the compressible Navier-Stokes equations on unstructured grids. *Computer Methods in Applied Mechanics & Engineering*, 196:4712 – 4736, 2007.
- [22] Luis Cueto-Felgueroso and Ignasi Colominas. High-order Finite Volume Methods and Multiresolution Reproducing Kernels. *Archives of Computational Methods in Engineering*, 15:185 – 228, 2008.

- [23] Xesús Nogueira, Ignasi Colominas, Luis Cueto-Felgueroso, and Sofiane Khelladi. On the simulation of wave propagation with a higher-order finite volume scheme based on Reproducing Kernel Methods. *Computer Methods in Applied Mechanics & Engineering*, 199:1471 – 1490, 2010.
- [24] Xesús Nogueira, Luis Cueto-Felgueroso, Ignasi Colominas, and Gómez H. Implicit Large Eddy Simulation of non-wall-bounded turbulent flows based on the multiscale properties of a high-order finite volume method. *Computer Methods in Applied Mechanics & Engineering*, 199:615 – 624, 2010.
- [25] Xesús Nogueira, Luis Cueto-Felgueroso, Ignasi Colominas, Fermín Navarrina, and Manuel Casteleiro. A new shock-capturing technique based on Moving Least Squares for higher-order numerical schemes on unstructured grids. *Computer Methods in Applied Mechanics & Engineering*, 199:2544 – 2558, 2010.
- [26] Jean-Camille Chassaing, Sofiane Khelladi, and Xesús Nogueira. Accuracy assessment of a high-order moving least squares finite volume method for compressible flows. *Computers & Fluids*, 71:41 – 53, 2013.
- [27] Luis Ramírez, Xesús Nogueira, Sofiane Khelladi, Jean-Camille Chassaing, and Ignasi Colominas. A new high-order finite volume method based on Moving Least Squares for the resolution of the incompressible Navier-Stokes equations on unstructured grids. *Computer Methods in Applied Mechanics & Engineering*, 278:883 – 901, 2014.
- [28] Stéphane Louis Clain, Gaspar J. Machado, and Rui M. S. Pereira. A new very high-order finite volume method for the 2D convection diffusion problem on unstructured meshes. In *IV Conferência Nacional em Mecânica dos Fluidos, Termodinâmica e Energia*. Laboratório Nacional de Engenharia Civil, may 2012.
- [29] Stéphane Louis Clain, Gaspar J. Machado, J. M. Nóbrega, and Rui M. S. Pereira. A sixth-order finite volume method for multidomain convection-diffusion problem with discontinuous coefficients. *Computer Methods in Applied Mechanics & Engineering*, 267:43 – 64, 2013.
- [30] Ricardo Costa, Stéphane Clain, and Gaspar José Machado. *Finite Volume Scheme Based on Cell-Vertex Reconstructions for Anisotropic Diffusion Problems with Discontinuous Coefficients*, pages 87 – 102. Springer International Publishing, 2014.
- [31] S. Clain and G.J. Machado. A very high-order finite volume method for the time-dependent convection diffusion problem with Butcher Tableau extension. *Computers & Mathematics with Applications*, 68(10):1292 – 1311, 2014.
- [32] A. Boullaras, Stéphane Louis Clain, and F. Baudoin. A sixth-order finite volume method for diffusion problem with curved boundaries. *Applied Mathematical Modelling*, 42:401 – 422, 2017.
- [33] Miguel R Visbal and Datta V Gaitonde. On the Use of Higher-Order Finite-Difference Schemes on Curvilinear and Deforming Meshes. *Journal of Computational Physics*, 181(1):155 – 185, 2002.

- [34] Xiaogang Deng, Meiliang Mao, Guohua Tu, Hanxin Zhang, and Yifeng Zhang. High-order and high accurate CFD methods and their applications for complex grid problems. *Communications in Computational Physics*, 11(04):1081 – 1102, 2012.
- [35] J.M.C. Pereira, M.H. Kobayashi, and J.C.F. Pereira. A Fourth-Order-Accurate Finite Volume Compact Method for the Incompressible Navier Stokes Solutions. *Journal of Computational Physics*, 167(1):217 – 243, 2001.
- [36] Carl Ollivier-Gooch. Higher-Order ENO Schemes for Unstructured Meshes Based on Least-Squares Reconstruction. In *35th Aerospace Sciences Meeting and Exhibit*, Aerospace Sciences Meetings. American Institute of Aeronautics & Astronautics, jan 1997.
- [37] Lucian Ivan and Clinton P. T. Groth. High-order solution-adaptive central essentially non-oscillatory (CENO) method for viscous flows. *Journal of Computational Physics*, 257:830 – 862, 2014.
- [38] Changqing Hu and Chi-Wang Shu. Weighted Essentially Non-oscillatory Schemes on Triangular Meshes. *Journal of Computational Physics*, 150:97 – 127, 1999.
- [39] Oliver Friedrich. Weighted Essentially Non-Oscillatory Schemes for the Interpolation of Mean Values on Unstructured Grids. *Journal of Computational Physics*, 144:194 – 212, 1998.
- [40] Vitaliy Gyrya and Konstantin Lipnikov. High-order mimetic finite difference method for diffusion problems on polygonal meshes. *Journal of Computational Physics*, 227:8841 – 8854, 2008.
- [41] Yen Liu, Marcel Vinokur, and Z. J. Wang. Spectral difference method for unstructured grids I: Basic formulation. *Journal of Computational Physics*, 216:780 – 801, 2006.
- [42] Z. J. Wang, Yen Liu, Georg May, and Antony Jameson. Spectral Difference Method for Unstructured Grids II: Extension to the Euler Equations. *Journal of Scientific Computing*, 32:45 – 71, 2007.
- [43] Felipe Pamplona Mariano, L. d Q. Moreira, A. d Silveira-Neto, Carlos Bettencourt da Silva, and J.C.F. Pereira. A new incompressible Navier-Stokes solver combining Fourier pseudo-spectral and immersed boundary methods. *Computer Modeling in Engineering & Sciences(CMES)*, 59(2):181 – 216, 2010.
- [44] F. Bassi and S. Rebay. High-order accurate discontinuous finite element solution of the 2d euler equations. *Journal of Computational Physics*, 138(2):251 – 285, 1997.
- [45] Alexandre Noll Marques, Jean-Christophe Nave, and Rodolfo Ruben Rosales. High order solution of Poisson problems with piecewise constant coefficients and interface jumps. *Journal of Computational Physics*, 335:497 – 515, 2017.
- [46] Hadrien Bériot, Albert Prinn, and Gwénael Gabard. Efficient implementation of high-order finite elements for Helmholtz problems. *International Journal for Numerical Methods in Engineering*, 106(3):213 – 240, 2016.

- [47] Bernardo Cockburn and Chi-Wang Shu. The Local Discontinuous Galerkin Method for Time-Dependent Convection-Diffusion Systems. *SIAM Journal on Numerical Analysis*, 35(6):2440 – 2463, 1998.
- [48] Jue Yan and Chi-Wang Shu. Local Discontinuous Galerkin Methods for Partial Differential Equations with Higher Order Derivatives. *Journal of Scientific Computing*, 17(1):27 – 47, 2002.
- [49] Douglas N. Arnold, Franco Brezzi, Bernardo Cockburn, and L. Donatella Marini. Unified Analysis of Discontinuous Galerkin Methods for Elliptic Problems. *Journal on Numerical Analysis*, 39(5):1749 – 1779, 2002.
- [50] Michael Dumbser, Martin Käser, and Eleuterio F. Toro. An arbitrary high-order Discontinuous Galerkin method for elastic waves on unstructured meshes V. Local time stepping and p-adaptivity. *Geophysical Journal International*, 171(2):695 – 717, 2007.
- [51] X. Nogueira, L. Cueto-Felgueroso, I. Colominas, H. Gómez, F. Navarrina, and M. Casteleiro. On the accuracy of finite volume and discontinuous Galerkin discretizations for compressible flow on unstructured grids. *International Journal for Numerical Methods in Engineering*, 78(13):1553 – 1584, 2009.
- [52] J. Jaśkowiec, P. Pluciński, and A. Stankiewicz. Discontinuous Galerkin method with arbitrary polygonal finite elements. *Finite Elements in Analysis and Design*, 120:1 – 17, 2016.
- [53] V. Etienne, E. Chaljub, J. Virieux, and N. Glinsky. An hp-adaptive discontinuous Galerkin finite-element method for 3-D elastic wave modelling. *Geophysical Journal International*, 183(2):941 – 962, 2010.
- [54] E. Bertolazzi and G. Manzini. A unified treatment of boundary conditions in least-square based finite-volume methods. *Computers & Mathematics with Applications*, 49(11):1755 – 1765, 2005.
- [55] Jerome Droniou. Finite volume schemes for diffusion equations: Introduction to and review of modern methods. *Mathematical Models and Methods in Applied Sciences*, 24(08):1575 – 1619, 2014.
- [56] Christopher Batty. A cell-centred finite volume method for the Poisson problem on non-graded quadtrees with second order accurate gradients. *Journal of Computational Physics*, 331:49 – 72, 2017.
- [57] S. Chantasiriwan. Methods of fundamental solutions for time-dependent heat conduction problems. *International Journal for Numerical Methods in Engineering*, 66(1):147 – 165, 2006.
- [58] Shuying Zhai, Zhifeng Weng, and Xinlong Feng. An adaptive local grid refinement method for 2D diffusion equation with variable coefficients based on block-centered finite differences. *Applied Mathematics and Computation*, 268:284 – 294, 2015.
- [59] Alireza Jalali, Mahkame Sharbatdar, and Carl Ollivier-Gooch. Accuracy analysis of unstructured finite volume discretization schemes for diffusive fluxes. *Computers & Fluids*, 101:220 – 232, 2014.

- [60] Chandan B. Sejekan and Carl F. Ollivier-Gooch. Improving finite-volume diffusive fluxes through better reconstruction. *Computers & Fluids*, 139:216 – 232, 2016.
- [61] Eleuterio F. Toro and Arturo Hidalgo. ADER finite volume schemes for nonlinear reaction-diffusion equations. *Applied Numerical Mathematics*, 59(1):73 – 100, 2009.
- [62] Luis Cea, Jerónimo Puertas, and María-Elena Vázquez-Cendón. Depth Averaged Modelling of Turbulent Shallow Water Flow with Wet-Dry Fronts. *Archives of Computational Methods in Engineering*, 14(3):303 – 341, 2007.
- [63] J. L. Guermond, P. Minev, and Jie Shen. An overview of projection methods for incompressible flows. *Computer Methods in Applied Mechanics & Engineering*, 195:6011 – 6045, 2006.
- [64] Geyang Guo and Shujuan Lü. Unconditional stability of alternating difference schemes with intrinsic parallelism for two-dimensional fourth-order diffusion equation. *Computers & Mathematics with Applications*, 71(10):1944 – 1959, 2016.
- [65] J. P. Magalhães. *An adaptive framework for the numerical simulation of environmental flows in complex geometries*. PhD thesis, Instituto Superior Técnico, Universidade Técnica de Lisboa, 2011.
- [66] Duarte M. S. Albuquerque. *Numerical Computation of Incompressible Flows on Adaptive and Unstructured Grids*. PhD thesis, Instituto Superior Técnico, Universidade Técnica de Lisboa, 2013.
- [67] Jonas A. S. Pereira. A 2-D Immersed Boundary Method on Low Reynolds Moving Body. Master's thesis, Instituto Superior Técnico, Universidade de Lisboa, 2014.
- [68] Diogo M. C. Martins. On the suppression of supirous pressure oscillations in immersed boundary methods with unstructured grids. Master's thesis, Instituto Superior Técnico, Universidade de Lisboa, 2016.
- [69] André L. A. Leite. Structured and unstructured finite volume calculations of a laminar and turbulent axisymmetric jet flow. Master's thesis, Instituto Superior Técnico, Universidade de Lisboa, 2014.
- [70] R. Reis. Cálculo paralelo das equações de Euler utilizando métodos de Newton-Krilov. Master's thesis, Instituto Superior Técnico, Universidade Técnica de Lisboa, 2005.
- [71] H. K. Versteeg and W. Malalasekera. *An Introduction to Computational Fluid Dynamics*. Pearson Education Limited, 2<sup>nd</sup> edition, 2007.
- [72] C. Hirsch. *Numerical Computation of Internal and External Flows, Fundamentals of Numerical Discretization*. Numerical Computation of Internal and External Flows. John Wiley & Sons, Ltd., 1<sup>st</sup> edition, 1991.
- [73] Parviz Moin. *Fundamentals of Engineering Numerical Analysis*. Cambridge University Press, 2<sup>nd</sup> edition, 2010.

- [74] J. E. Akin. *Finite Element Analysis with Error Estimators*. Elsevier, 1<sup>st</sup> edition, 2005.
- [75] J. N. Reddy. *An Introduction to the Finite Element Method*. McGraw-Hill Education, 3<sup>rd</sup> edition, 2006.
- [76] Heitor Pina. *Métodos Númericos*. McGraw-Hill de Portugal, 1<sup>st</sup> edition, 1995.
- [77] Eric Weisstein. Legendre-Gauss Quadrature, 2017.
- [78] Hrvoje Jasak. *Error Analysis and Estimation for the Finite Volume Method with Applications to Fluid Flows*. PhD thesis, Imperial College of Science, Technology and Medicine, 1996.
- [79] O.C. Zienkiewicz and R.L. Taylor. *The Finite Element Method: Basic formulation and linear problems*. McGraw-Hill Education, 4<sup>th</sup> edition, 1989.
- [80] C.B. Laney. *Computational Gasdynamics*. Cambridge University Press, 1998.
- [81] J.H. Ferziger and M. Peric. *Computational Methods for Fluid Dynamics*. Springer Berlin Heidelberg, 2001.
- [82] P.K. Khosla and S.G. Rubin. A diagonally dominant second-order accurate implicit scheme. *Computers & Fluids*, 2(2):207 – 209, 1974.
- [83] Per-Olof Persson and Gilbert Strang. A Simple Mesh Generator in MATLAB. *SIAM Review*, 46:329 – 345, 2004.
- [84] F. Juretić and A. D. Gosman. Error Analysis of the Finite-Volume Method with Respect to Mesh Type. *Numerical Heat Transfer, Part B: Fundamentals*, 57(6):414 – 439, 2010.
- [85] R.L. Burden and J.D. Faires. *Numerical Analysis*. Cengage Learning, 9<sup>th</sup> edition, 2010.
- [86] Samuel K. M. Chenoweth, Julio Soria, and Andrew Ooi. A singularity-avoiding moving least squares scheme for two-dimensional unstructured meshes. *Journal of Computational Physics*, 228:5592 – 5619, 2009.
- [87] James W. Demmel. *Applied Numerical Linear Algebra*. SIAM, 1997.
- [88] Amir Nejat and Carl Ollivier-Gooch. Effect of discretization order on preconditioning and convergence of a high-order unstructured Newton-GMRES solver for the Euler equations. *Journal of Computational Physics*, 227(4):2366 – 2386, 2008.
- [89] Pavel Kus. Convergence and stability of higher-order finite element solution of reaction-diffusion equation with Turing instability. *Application of Mathematics*, pages 140–147, 2015.
- [90] C. Hirsch. *Numerical Computation of Internal and External Flows: The Fundamentals of Computational Fluid Dynamics*. Butterworth-Heinemann, 2<sup>nd</sup> edition, 2007.
- [91] J. G. Charney, R. Fjortoft, and J. Von Neumann. Numerical Integration of the Barotropic Vorticity Equation. *Tellus*, 2(4):237 – 254, 1950.

- [92] C. William Gear. *Numerical initial value problems in ordinary differential equations*. Prentice-Hall, 1971.
- [93] J.D. Lambert. *Computational methods in ordinary differential equations*. Introductory mathematics for scientists and engineers. John Wiley & Sons, Ltd., 1973.
- [94] J.D. Lambert. *Numerical Methods for Ordinary Differential Systems: The Initial Value Problem*. John Wiley & Sons, Ltd., 1991.
- [95] P.J. van der Houwen. *Construction of integration formulas for initial value problems*. North-Holland, 1977.
- [96] B.P. Sommeijer, P.J. van der Houwen, and J. Kok. Time integration of three-dimensional numerical transport models. *Applied Numerical Mathematics*, 16(1):201 – 225, 1994.
- [97] W. Kutta. *Beitrag zur näherungsweisen Integration totaler Differentialgleichungen*. Teubnerer, 1901.

## Appendix A

# Stability Analysis of High-Order Diffusive Schemes

To avoid that the numerical solution has a large error, it is required to study some properties such as: consistency, stability and convergence of the numerical schemes. Consistency of a numerical scheme affirms that when the time and space step tend to zero the numerical scheme must tend to the differential equation. Stability of a numerical scheme states that the errors must remain bounded when the iterative process advances. Convergence condition affirms that the numerical solution must tend to the exact solution when the time and space step tend to zero.

According with the equivalence theorem of Lax for a well-posed initial value problem and a consistent discretization scheme, stability is the necessary and sufficient condition for convergence, [90]. This property is very important since affirm that it is only required to test the numerical schemes' stability in order to conclude about the quality of the obtained numerical solution and because so this chapter only focus on the stability analysis of the numerical schemes.

To investigate about the stability of a scheme it is required to separate the influence of the boundary conditions from the main stability analysis and is widely used periodic boundary conditions, since they not require boundary treatment, [90].

Many methods had been development to analyse the stability of the numerical schemes but the most used and relevant one is the Von Neumann method. This method was developed by Von Neumann and first introduced in this context in 1950 by Charney et al., [91], where the innovation of this method was the expansion of the error into a finite Fourier series in the spatial frequency domain.

### A.1 Unsteady-State Heat Equation

One example of an unsteady-state form of the Poisson equation is the heat equation for unsteady-state, which in conservative form is given by:

$$\frac{\partial \phi}{\partial t} = \kappa \nabla^2 \phi, \quad (\text{A.1})$$

where the  $\kappa$  is the thermal conductivity. Applying the FVM on the integral form of the conservative law (A.1) over a CV and the Gauss divergence theorem will result in the following expression:

$$\int_{CV} \frac{\partial \phi}{\partial t} dV = \int_{CS} \kappa \nabla \phi dS. \quad (\text{A.2})$$

For a 1D domain the above expression can be written in the following form:

$$\frac{\partial \phi_i}{\partial t} V_i = \kappa_i (\nabla \phi_e - \nabla \phi_w), \quad (\text{A.3})$$

where the subscript  $w$  and  $e$  are the west and east face, respectively. An example of a 1D domain is presented in figure 2.1. This is a parabolic type equation and the space dependent terms have the same behaviour to the elliptic type equation, presented in chapter 2, and consequently, centered discretization for spacial terms is required. The discretizations presented in chapter 2 will be used.

## A.2 Stability Analysis by Von Neumann Method

Since periodic boundary conditions are used, it is possible to obtain the numerical solution in terms of an expansion of Fourier series in space for each time step  $n$ . Since the space step is finite, it would be necessary to use the discrete form of the Fourier series, where the final result is obtained through the sum of all wave numbers,  $k$ , that can be represented in space.

The wave number is by definition the number of waves that can be represented in an interval of  $2\pi$ , with a determined wavelength,  $\lambda$ . The minimum wavelength can be obtained through  $\lambda_{\min} = 2\Delta x$  and the wave number can be calculated by  $k = 2\pi/\lambda$ , where in the case of  $\lambda = \lambda_{\min}$  the respective wave number will be  $k_{\max} = \pi/L$ . Since the space step is determined as  $\Delta x/n_{cells}$  the wave number can be calculated by:

$$k_j = j k_{\min} = \frac{j\pi}{n_{cells} \Delta x}. \quad (\text{A.4})$$

The phase angle can be defined as the product of the space step and wave number, being expressed by the following expression:

$$\theta = k_j \Delta x_j = \frac{j\pi}{n_{cells}}, \quad (\text{A.5})$$

where for  $\theta = 0$  is the low frequency regions and in opposite the region of high frequencies is given by  $\theta = \pi$ .

### A.2.1 Fourier Series Decomposition

Since any function can be decomposed into a Fourier series as:

$$\phi_i^n = \sum_{j=-n_{cells}}^{n_{cells}} \Phi_j^n e^{jk_j x_i} \quad (\text{A.6})$$

where  $I = \sqrt{-1}$  is the imaginary number and  $\Phi$  is the wave amplitude. It is important to mention that this decomposition separates the time and space dependency of the solution, where the amplitude of the wave represents the time behaviour and the Fourier modes represents the space dependency.

Finally, it is important to define the stability condition by Von Neumann, which affirms that the amplitude of any harmonic may not grow indefinitely in time, [90]. From this stability condition it is possible to define the amplification factor of the numerical solution,  $G$ , expressed as:

$$G = \frac{\Phi^{n+1}}{\Phi^n}, \quad (\text{A.7})$$

which is function of the diffusive criteria,  $\beta$ , which will be further explained, and the phase angle,  $\theta$ , being independent of the time step. To satisfy the Von Neumann stability condition, it is required to have an amplification factor that respects the following condition:  $|G| \leq 1$ . From this stability condition, the schemes can be unconditionally stable when is always stable, conditionally stable when to be stable the scheme must accomplish certain requirements and unconditional unstable for the cases where the scheme is always unstable.

Finally, the Von Neumann method for stability analysis is an easy way to study the stability properties for linear schemes with constant coefficients and periodic boundary conditions.

### A.3 Spectral Analysis of Numerical Errors

From the stability analysis performed through the Von Neumann method, it is possible to obtain the amplification factor as function of the phase angle and the properties of the scheme, see equation (A.7).

Starting from the analytical solution given by the equation (A.1), which is a parabolic problem, it is possible to define the wave function for this kind of problems, which is given by:

$$\phi^A = \Phi^A e^{-I\tilde{\omega}t} e^{Ikx}, \quad (\text{A.8})$$

where  $\Phi$  is the wave amplitude,  $\tilde{\omega} = I\kappa k^2 = -I\beta\theta^2/\Delta t$  is the exact dispersion relation,  $\beta$  is the diffusive criteria. Another way to express the wave equation is joining the time dependent term to the amplitude one, which will result in  $\Phi = \Phi e^{-I\omega t} = \Phi e^{-\alpha k^2 t}$  and the wave equation will be written as:

$$\phi^A = \Phi^A e^{Ikx}. \quad (\text{A.9})$$

Considering the first term of the Fourier expansion given by equation (A.6) will result in:

$$\phi = \Phi e^{Ik(i\Delta x)}, \quad (\text{A.10})$$

which can be represented by:

$$\phi^n = \Phi e^{I\omega n \Delta t}, \quad (\text{A.11})$$

where the  $\omega$  is the numerical dispersion relation of the scheme.

Since the amplification factor is independent of the time step,  $\Delta t$ , it is possible to define the amplification factor as the ratio between the wave amplitude in two consecutive time steps, which is given by:

$$\tilde{G} = \frac{\Phi^A(t + \Delta t)}{\Phi^A(t)} = e^{-\alpha k^2 \Delta t}, \quad (\text{A.12})$$

where the last equivalence is only valid for an exact solution.

From the amplification factor, it is possible to analyse the nature and the spectre of frequencies of the schemes. In general, the  $\omega$  is a complex number, and the amplification factor can be decomposed in the following form:

$$G = |G|e^{-I\theta}, \quad (\text{A.13})$$

where  $|G| = \sqrt{(\text{Re}(G))^2 + (\text{Im}(G))^2}$  and  $\theta = \arctan\left(\frac{-\text{Im}(G)}{\text{Re}(G)}\right)$ . The above equation is valid for numerical solution, in order to represent the same quantity for the exact solution  $|G|$  and  $\theta$  should be replaced by  $|\tilde{G}|$  and  $\tilde{\theta}$ , respectively.

The equation (A.13) suggests that  $|G|$  influences the amplitude of the solution and  $\theta$  affects the phase of the solution. It becomes necessary to define the error topology, that can be classified as amplitude and phase error. The amplitude error is known as diffusive error or dissipation error and it is defined by the ratio between the amplitude of the amplification factor of the numerical and exact solutions, and expressed as:

$$\varepsilon_D = \frac{|G|}{|\tilde{G}|}. \quad (\text{A.14})$$

When  $\varepsilon_D = 1$  it means that does not exist dispersion error and the solution amplitude is equal to the exact amplitude. Case  $\varepsilon_D > 1$  the solution's amplitude is higher than the exact one and the scheme amplifies the solution, in the other hand when  $\varepsilon_D < 1$  the amplitude of the solution is lower than the exact one and can be said that the scheme damps the solution.

The phase error is known as dispersion error and it is defined as the difference between the numerical and exact solution's phase angle and it is expressed by:

$$\varepsilon_\theta = \theta - \tilde{\theta}. \quad (\text{A.15})$$

For this case, when  $\varepsilon_\theta = 0$  there is no phase error and the solution propagates at the same velocity than the exact solution. For the case where  $\varepsilon_\theta > 0$  the phase angle of the solution is higher than the phase angle of the exact solution and the schemes advances the propagation of the solution, for the opposite case,  $\varepsilon_\theta < 0$ , the phase angle of the solution is lower than the exact one and the schemes delays the propagation of the solution.

In this Thesis only parabolic type problems are analysed, which means that the amplification factor is real and consequently, there is no phase error, which means that will have  $\varepsilon_\theta = 0$  for all proposed schemes independently of the time discretization performed. Since, there is no phase error for this type of problems the expression for the amplitude error, given by equation (A.14), changes and takes the following form:

$$\varepsilon_D = \frac{G}{\tilde{G}}, \quad (\text{A.16})$$

the reason for this change is to take into account the situation where the amplitude of the solution is negative. Generally for parabolic type problems for low frequencies,  $\theta \sim 0$ , the amplitude error is small, however for high frequencies,  $\theta \sim \pi$ , the amplitude error increases becoming very high, being an issue for the overall accuracy of the scheme for the cases where the initial solution contains a large number of high frequencies components, [90].

## A.4 Time Integration Methods

In this section an explanation about the time discretization performed on the time dependent terms will be provided.

Equation (A.3), with some mathematical manipulation, can be expressed in the following form:

$$\frac{\partial \phi_i}{\partial t} = \frac{\alpha_i}{\Delta x} (\nabla \phi_e - \nabla \phi_w), \quad (\text{A.17})$$

in the matrix form the above equation takes the following form:

$$\frac{\partial \Phi}{\partial t} = \mathbf{A} \Phi, \quad (\text{A.18})$$

where  $\mathbf{A}$  is the global matrix obtained from the space discretization, this expression is known as the semi-discretized representation form.

The stability analysis performed on equation (A.18) would be based on the eigenvalues and eigenvector of the matrix  $\mathbf{A}$ , since it is possible to show that the exact solution is directly determine from the eigenvalues and eigenvector of the global matrix, see Hirsch, [90], for a detailed description.

$$\det |\mathbf{A} - \Omega| = 0, \quad (\text{A.19})$$

where the  $\Omega_j$  are the eigenvalues of the matrix, which only depends on the space coordinates.

From equation (A.10) it is possible to express the equation (A.18) in the following form:

$$\mathbf{A} e^{I k_j i \Delta x} = \Omega(\theta_j) e^{I k_j i \Delta x}, \quad (\text{A.20})$$

which allows to conclude that equation (A.18) can be represented as:

$$\frac{d\phi}{dt} = \Omega \phi, \quad (\text{A.21})$$

where this equation will be from now on the basis for all stability analysis and time discretizations. Comparing with the equation (A.17), it suggests that:

$$\Omega \phi_i = \frac{\alpha_i}{\Delta x_i} (\nabla \phi_e - \nabla \phi_w) = H_i, \quad (\text{A.22})$$

Specifying for the second-order case and from the face fluxes obtained with the equation (2.7) results

in:

$$\begin{aligned} \Omega\phi_i &= \frac{\alpha_i}{\Delta x} (\nabla\phi_e - \nabla\phi_w) \Leftrightarrow \Omega\phi_i = \frac{\alpha_i}{(\Delta x)^2} (\phi_{i+1} - 2\phi_i + \phi_{i-1}) \Leftrightarrow \\ &\Leftrightarrow \Omega\phi_i = \frac{\alpha_i}{(\Delta x)^2} \left( \Phi e^{I\theta(i+1)} - 2\Phi e^{I\theta i} + \Phi e^{I\theta(i-1)} \right) \Leftrightarrow \Omega\phi_i = \frac{\alpha_i}{(\Delta x)^2} (e^{I\theta} - 2 + e^{-I\theta}) \Phi e^{I\theta i} \Leftrightarrow \\ &\Leftrightarrow \Omega\phi_i = \frac{\alpha_i}{(\Delta x)^2} (e^{I\theta} - 2 + e^{-I\theta}) \phi_i \Leftrightarrow \Omega\phi_i = \frac{2\alpha_i}{(\Delta x)^2} (\cos(\theta) - 1) \phi_i \Leftrightarrow \Omega = \frac{2\alpha_i}{(\Delta x)^2} (\cos(\theta) - 1) \end{aligned}$$

Considering that the diffusive criteria is given by:

$$\beta = \frac{\kappa\Delta t}{(\Delta x)^2} \quad (\text{A.23})$$

it is possible to conclude that for the second-order in space scheme:

$$(\Omega\Delta t)_2 = 2\beta (\cos(\theta) - 1), \quad (\text{A.24})$$

in an analogous way it is possible to obtain the same expression for the fourth, sixth and eighth-order in space schemes, given by:

$$(\Omega\Delta t)_4 = \frac{\beta}{12} (-\cos(2\theta) + 28\cos(\theta) - 27), \quad (\text{A.25})$$

$$(\Omega\Delta t)_6 = \frac{\beta}{32} \left( \frac{3}{10}\cos(3\theta) - \frac{67}{15}\cos(2\theta) + \frac{475}{6}\cos(\theta) - 75 \right), \quad (\text{A.26})$$

$$(\Omega\Delta t)_8 = \frac{\beta}{512} \left( -\frac{5}{7}\cos(4\theta) + \frac{368}{35}\cos(3\theta) - \frac{1372}{15}\cos(2\theta) + \frac{3920}{3}\cos(\theta) - 1225 \right), \quad (\text{A.27})$$

#### A.4.1 Time Discretization

In this Thesis eight different time schemes are analysed. The Euler explicit and implicit, the Crank-Nicholson, Adams-Bashforth schemes. It was also studied multi step methods, namely the Runge-Kutta scheme, where four different variants are analysed.

##### Euler Explicit

The Euler explicit (EE) scheme is a first-order explicit scheme in time and it is given by:

$$\phi^{n+1} = \phi^n + \Delta t H^n. \quad (\text{A.28})$$

From the Fourier Series it is possible to conclude that:

$$\Phi^{n+1} e^{I\theta i} = (1 + \Omega\Delta t) \Phi^n e^{I\theta i} \quad (\text{A.29})$$

and consequently the amplification factor will be defined as:

$$G_{EE}(\Omega\Delta t) = 1 + \Omega\Delta t, \quad (\text{A.30})$$

where the subscript *EE* means that the amplification factor is calculated for the Euler explicit scheme.

### Backward Euler

The backward Euler (BE) scheme is a first-order implicit scheme in time and can be defined as:

$$\phi^{n+1} = \phi^n + \Delta t H^{n+1}. \quad (\text{A.31})$$

combining the above expression with the first term of the Fourier Series, the above expression becomes:

$$\Phi^{n+1} e^{I\theta i} = \frac{1}{1 - \Omega \Delta t} \Phi^n e^{I\theta i} \quad (\text{A.32})$$

and the amplification factor will be:

$$G_{BE}(\Omega \Delta t) = \frac{1}{1 - \Omega \Delta t}. \quad (\text{A.33})$$

### Crank-Nicholson

The Crank-Nicholson (CN) scheme is a second-order accurate scheme in time, being considered a semi-implicit scheme because it has both implicit and explicit contributions for the final result. It can be written as:

$$\phi^{n+1} = \phi^n + \frac{\Delta t}{2} (H^{n+1} + H^n). \quad (\text{A.34})$$

If the Fourier series has taken into account, the expression for this scheme will be:

$$\Phi^{n+1} e^{I\theta i} = \frac{2 + \Omega \Delta t}{2 - \Omega \Delta t} \Phi^n e^{I\theta i} \quad (\text{A.35})$$

this will result in the following amplification factor:

$$G_{CN}(\Omega \Delta t) = \frac{2 + \Omega \Delta t}{2 - \Omega \Delta t}. \quad (\text{A.36})$$

### Adams-Bashworth

The Adams-Bashworth (AB) scheme is an explicit second-order scheme in time and is defined as:

$$\phi^{n+1} = \phi^n + \frac{\Delta t}{2} (3H^n - H^{n-1}). \quad (\text{A.37})$$

when the transportable variable is expanded into a Fourier series with one term, the above expression results in:

$$\Phi^{n+1} e^{I\theta i} = \left(1 + \frac{3\Omega \Delta t}{2}\right) \Phi^n e^{I\theta i} - \frac{\Omega \Delta t}{2} \Phi^{n-1} e^{I\theta i}, \quad (\text{A.38})$$

which for the amplification factor is required to use the quadratic formula, which will lead to

$$G_{AB}(\Omega \Delta t) = \frac{2 + 3\Omega \Delta t}{4} \pm \frac{1}{2} \sqrt{1 + \Omega \Delta t + \frac{9}{4} (\Omega \Delta t)^2}, \quad (\text{A.39})$$

since there are two possible solutions for the expression of the amplification factor, it is required to select one. To chose the final expression for the amplification factor it is necessary to verify which one takes the value 1 when  $\Delta t$  tends to zero, which will allow the use of positive sign, and thus the amplification factor will be:

$$G_{AB}(\Omega\Delta t) = \frac{2 + 3\Omega\Delta t}{4} + \frac{1}{2}\sqrt{1 + \Omega\Delta t + \frac{9}{4}(\Omega\Delta t)^2}, \quad (\text{A.40})$$

### Runge-Kutta

The Runge-Kutta (RK) schemes are a family of explicit high-order numerical schemes for time integration, being a multi-step method. For a detailed description see Gear, [92], Lambert, [93, 94] and Van der Houwen, [95].

The idea of RK schemes is the evaluation of the  $H$  term, given by equation (A.22), for several intermediate instants of time, between  $n\Delta t$  and  $(n+1)\Delta t$ , being the stages of the RK scheme, combined in order to obtain a high-order approximation for the instant  $(n+1)\Delta t$ . The general expression for the RK scheme with  $m$  stages is given by:

$$\begin{aligned} \phi^{(1)} &= \phi^n \\ \phi^{(2)} &= \phi^n + \Delta t \sigma_2 H^{(1)} \\ \phi^{(3)} &= \phi^n + \Delta t \sigma_3 H^{(2)} \\ &\dots \\ \phi^{(m)} &= \phi^n + \Delta t \sigma_m H^{(m-1)} \\ \phi^{n+1} &= \phi^n + \Delta t \sum_{j=1}^m \tau_j H^{(j)} \end{aligned}, \quad (\text{A.41})$$

where the  $\sigma$  and  $\tau$  are the parameters that maximizes the order of the scheme, which are determine through a Butcher table and to obtain a consistent method it is required that:

$$\sum_{j=1}^m \tau_j = 1. \quad (\text{A.42})$$

The amplification factor for this type of methods is given by:

$$G_{RK}(\Omega\Delta t) = 1 + \sum_{j=1}^m g_j (\Omega\Delta t)^j, \quad (\text{A.43})$$

where  $g_j$  is given by:

$$g_1 = 0, \quad g_2 = \sigma_m, \quad g_3 = \sigma_m \sigma_{m-1}, \quad \dots, \quad g_m = \sigma_m \sigma_{m-1} \dots \sigma_2. \quad (\text{A.44})$$

A more detailed explanation about the amplification factor of this scheme is presented in Hirsch, [90].

Four different RK schemes were analysed, three with four stages and one of three stages, which parameters are presented at table A.1. The RK4 2<sup>nd</sup> order is especially appropriate for convection-

Table A.1: Coefficients of the different RK schemes analysed.

Scheme	$\sigma_2$	$\sigma_3$	$\sigma_4$
RK4 Classic	$\frac{1}{2}$	$\frac{1}{2}$	1
RK4 2 <sup>nd</sup> Order	$\frac{1}{4}$	$\frac{1}{3}$	$\frac{1}{2}$
RK4 3/8 Rule	$\frac{1}{3}$	$\frac{2}{3}$	1
RK3 Low Storage	$\frac{1}{2}$	1	-

diffusion problems, it was firstly proposed by Sommeijer et al., [96] and will be called as RK42. The RK3 Low storage will be defined as RK3 since it is third-order accurate and can be found in Pina, [76]. The RK4 Classic will be called as RK4C and the RK4 3/8 Rule will be defined as RK4R both are fourth order accurate and were introduced by Kutta, [97].

## A.5 Stability Domain

In this section, it will be presented the different stability domain for all proposed numerical schemes. Starting from the Von Neumann stability condition, which is given by:

$$|G| \leq 1, \quad (\text{A.45})$$

and solving in order to  $\beta$ , the following stability domains presented in table A.2 were obtained:

Table A.2: Stability domains for the numerical schemes for the time integration method for the different space discretization orders.

Scheme	FCS2	FCS4	FCS6	FCS8	Stability
Euler Explicit	$0 \leq \beta \leq 0.5$	$0 \leq \beta \leq 0.4286$	$0 \leq \beta \leq 0.4027$	$0 \leq \beta \leq 0.3887$	Conditionally Stable
Backward Euler	$\beta \geq 0$	$\beta \geq 0$	$\beta \geq 0$	$\beta \geq 0$	A-Stable
Crank-Nicholson	$\beta \geq 0$	$\beta \geq 0$	$\beta \geq 0$	$\beta \geq 0$	A-Stable
Adams-Bashworth	$\beta \geq 0$	$\beta \geq 0$	$\beta \geq 0$	$\beta \geq 0$	Unconditionally Stable
RK4 Classic	$0 \leq \beta \leq 0.3239$	$0 \leq \beta \leq 0.2776$	$0 \leq \beta \leq 0.2609$	$0 \leq \beta \leq 0.2518$	Conditionally Stable
RK4 2 <sup>nd</sup> Order	$0 \leq \beta \leq 0.6963$	$0 \leq \beta \leq 0.5968$	$0 \leq \beta \leq 0.5608$	$0 \leq \beta \leq 0.5413$	Conditionally Stable
RK4 3/8 Rule	$0 \leq \beta \leq 0.4609$	$0 \leq \beta \leq 0.4028$	$0 \leq \beta \leq 0.3785$	$0 \leq \beta \leq 0.3653$	Conditionally Stable
RK3 Low Storage	$0 \leq \beta \leq 0.5$	$0 \leq \beta \leq 0.4286$	$0 \leq \beta \leq 0.4067$	$0 \leq \beta \leq 0.3887$	Conditionally Stable

Table A.2 and figure A.1 show the different stability domains for all space discretizations considered applied to several time discretization methods. the table also lists the scheme's classification in terms of stability. It is verified that, with the exception of Adams-Bashworth scheme, all pure explicit scheme are conditionally stable, being the RK4 Classic the more restrictive. The remaining ones are unconditionally stable.

The influence of the space discretization is a reduction of the stability domain with the increase of the scheme's order, being the difference higher between the second and fourth order space schemes.

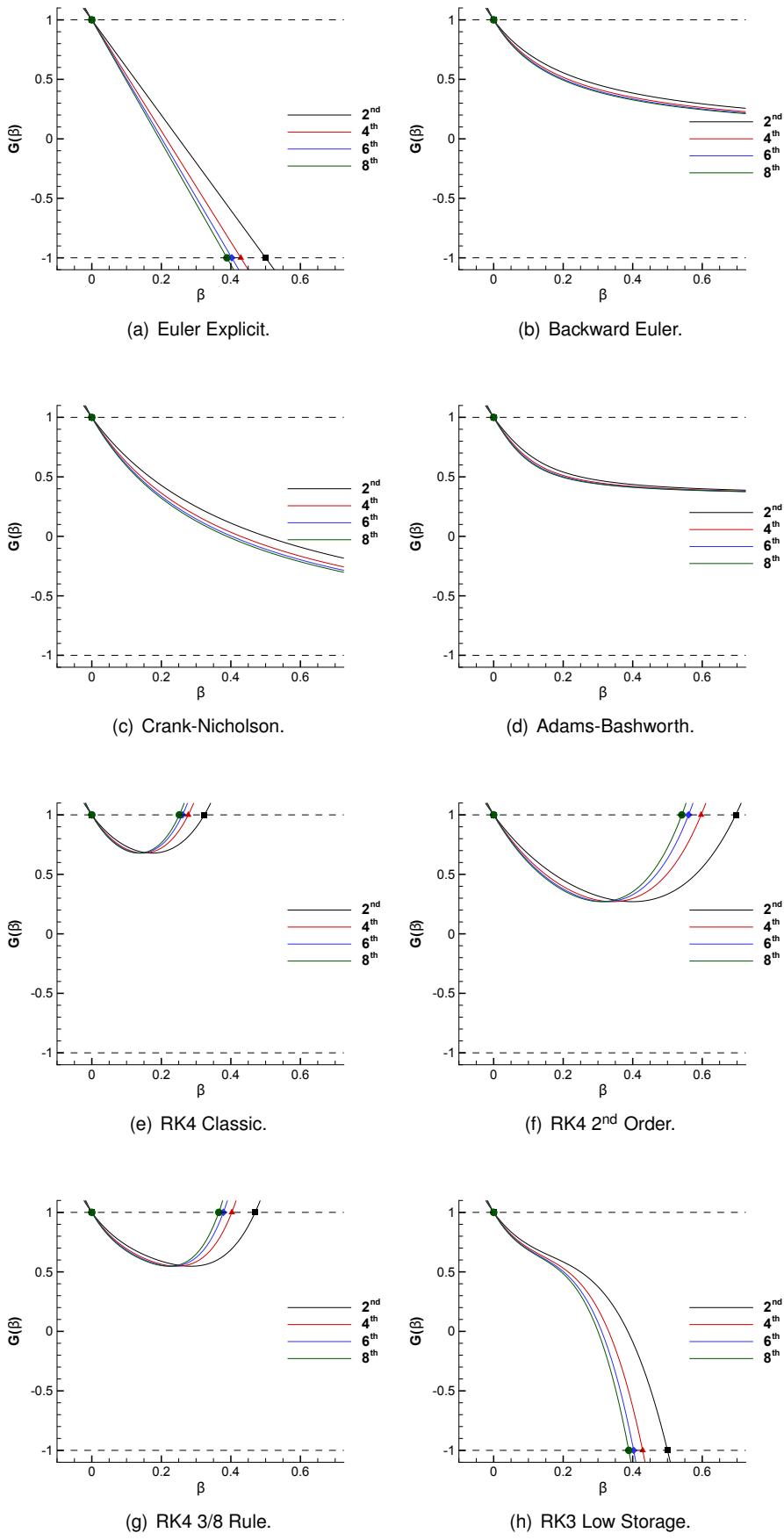


Figure A.1: Stability domains for the different time schemes for all space discretizations.

## A.6 Amplitude Error Optimization

As has already been said, the amplitude error is small for lower frequencies but it increases for the higher ones, which can be extremely high affecting the overall accuracy. So, it becomes necessary to find a way to have a low amplitude error for the spectrum of frequencies.

From equation (A.12) and knowing that  $\tilde{\omega} = I\alpha k^2 = I\beta\theta^2/\Delta t$ , it is possible to deduce that the exact amplification factor will be given by:

$$\tilde{G} = \frac{\Phi^A(t + \Delta t)}{\Phi^A(t)} = e^{-\beta\theta^2}, \quad (\text{A.46})$$

which is only function of  $\beta$  and  $\theta$ . In section A.4, it was seen that the numerical amplification factor is function of  $\Omega\Delta t$  and this one is function of  $\beta$  and  $\theta$  consequently, the numerical amplification factor will be also function of  $\beta$  and  $\theta$ . Since the amplitude error is given by equation (A.16) and defining  $\theta = \pi$  in order to represent the highest frequencies of the spectrum, the amplification factor will be only function of  $\beta$ . Thus, it is possible to find a  $\beta$  that minimizes the amplitude error for both low and high frequencies.

Table A.3:  $\beta$  that minimizes the amplitude error for high frequencies for the different time discretization methods.

Scheme	FCS2	FCS4	FCS6	FCS8
Euler Explicit	0.2221	0.1769	0.1597	0.1502
Backward Euler	-	-	-	-
Crank-Nicholson	0.4923	0.4145	0.3851	0.3688
Adams-Basworth	-	-	-	-
RK4 Classic	-	-	-	-
RK4 2 <sup>nd</sup> Order	-	-	-	-
RK4 3/8 Rule	-	-	-	-
RK3 Low Storage	0.3820	0.3248	0.3039	0.2925

Figure A.2 and table A.3 show the  $\beta$  value that minimizes the amplitude error for the different time discretizations. Note that in the figure A.2 only the cases where it is possible to obtain the optimal  $\beta$  were represented, which are the following ones: the Euler Explicit, Crank-Nicholson and RK3 schemes. The figure A.2(b) shows the behaviour of the Backward Euler scheme, where is impossible to find a  $\beta$  that minimizes the amplitude error, this behaviour is the same for the remaining schemes that do not converge for an optimal solution of  $\beta$ . It can be also concluded that the optimal diffusive criterion decreases as the space discretization order gets higher.

Figure A.3 shows the behaviour of several  $\beta$  function of  $\theta$  for all space discretization for the different time discretization, namely the ones that were possible to obtain the optimal diffusive criterion. In figure A.3(a) the dotted lines represent the behaviour of the diffusive criterion maximum that guarantee stability,  $\beta_{\max}$ , and it shows that for those cases the higher frequencies produces large amplitude errors, in opposite to the behaviour of the  $\beta_{opt}$ . The figure also suggests that to have a low amplitude error along the spectre of frequencies the use of a high-order scheme is required, specially for the cases where the space discretization is higher than one. It can be also concluded that the best time discretization is the

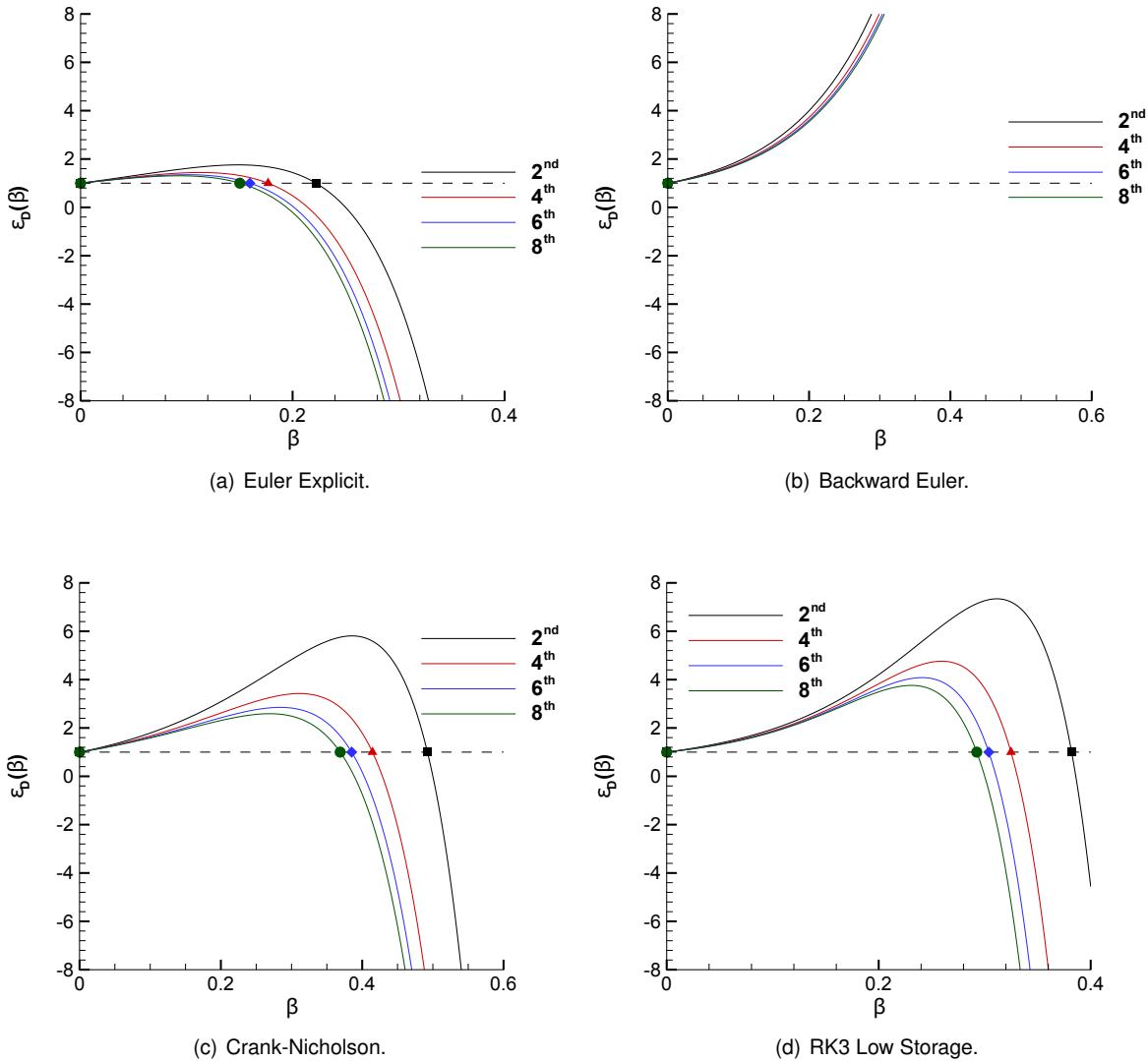


Figure A.2: Curves of the amplitude error where  $\theta = \pi$  and function of  $\beta$  for several time schemes for all space discretization orders.

CN scheme, since is the one that remains with a low amplitude error along all frequency spectre.

Finally, this study allowed to understand the behaviour of the diffusive criterion and the frequencies that the numerical solution has. These results allow to conclude that the Crank-Nicholson scheme is the best option, since it has the highest time steps for the smallest amplitude error within all frequency spectre.

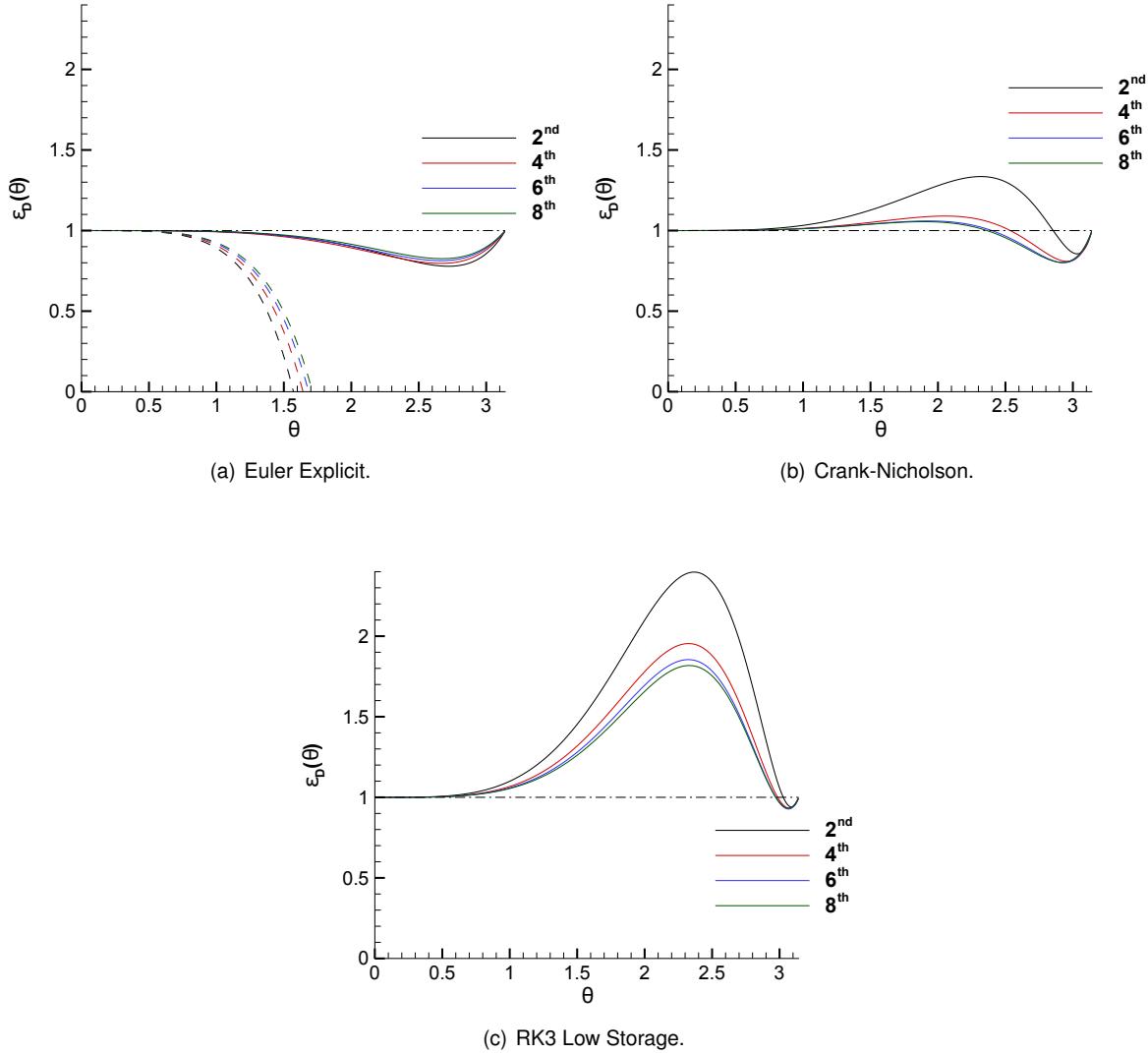


Figure A.3: Curves of the amplitude error function of  $\theta$  for the optimal  $\beta$  value for several time schemes for all space discretization orders, the dotted lines represents the  $\beta_{\max}$  (only applicable to the Euler Explicit scheme).

