DEVELOPMENT OF DISCONTINUOUS GALERKIN METHOD 2 DIMENSIONAL FLOW SOLVER

A THESIS SUBMITTED TO THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES OF MIDDLE EAST TECHNICAL UNIVERSITY

 $\mathbf{B}\mathbf{Y}$

OSMAN GÜNGÖR

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE IN AEROSPACE ENGINEERING

SEPTEMBER 2019

Approval of the thesis:

DEVELOPMENT OF DISCONTINUOUS GALERKIN METHOD 2 DIMENSIONAL FLOW SOLVER

submitted by **OSMAN GÜNGÖR** in partial fulfillment of the requirements for the degree of **Master of Science in Aerospace Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar Dean, Graduate School of Natural and Applied Sciences	
Prof. Dr. İsmail Hakkı Tuncer Head of Department, Aerospace Engineering	
Prof. Dr. Serkan Özgen Supervisor, Aerospace Engineering, METU	
Examining Committee Members:	
Prof. Dr. Yusuf Özyörük Aerospace Engineering, METU	
Prof. Dr. Serkan Özgen Aerospace Engineering, METU	
Prof. Dr. Yusuf Uludağ Chemical Engineering, METU	
Assoc. Prof. Dr. Sinan Eyi Aerospace Engineering, METU	
Assist. Prof. Dr. Mustafa Kaya Aerospace Engineering, Ankara Yıldırım Beyazıt University	

Date: 09.09.2019

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Osman Güngör

Signature :

ABSTRACT

DEVELOPMENT OF DISCONTINUOUS GALERKIN METHOD 2 DIMENSIONAL FLOW SOLVER

Güngör, Osman M.S., Department of Aerospace Engineering Supervisor: Prof. Dr. Serkan Özgen

September 2019, 86 pages

In this work, 2 dimensional flow solutions of Euler equations are presented from the developed discontinuous Galerkin method finite element method (DGFEM) solver on unstructured grids. Euler equations govern the inviscid and adiabatic flows with a set of hyperbolic equations. The discretization of governing equations for DGFEM is given in detail. The DGFEM discretization provides high order solutions on an element-compact stencil hence only elements having common boundary are coupled. The required elementwise operations and mathematical operations are revisited and derivations are provided when necessary. Among the two major approaches, modal and nodal, nodal DGFEM is employed. Gaussian quadrature is utilized in the evaluation of volume and surface integrals. The flux through the cell boundaries are calculated through flux functions and several flux functions are implemented and compared. Proper boundary conditions are employed on the solution space boundaries. Several test cases in literature are used for verification and validation purposes. The high order accuracy is easily achieved in problems with smooth solutions. On the other hand, problems with shocks requires stabilization techniques which may

limit the order of accuracy or degrade solution success. The satisfactory results are obtained with comparison of experimental results which are carefully selected considering the fidelity of governing equations. Moreover, importance of curved wall boundary representations in high order methods are experienced. Furthermore, effect of grid adaptation around shocks or discontinues is pointed out.

Keywords: Computational Fluid Dynamics, Discontinuous Galerkin, High Order Accuracy, Finite Element, Euler Equations, Aerodynamics

KESİNTİLİ GALERKİN METHODU İLE 2 BOYUTLU AKIŞ ÇÖZÜCÜ GELİŞTİRİLMESİ

Güngör, Osman Yüksek Lisans, Havacılık ve Uzay Mühendisliği Bölümü Tez Yöneticisi: Prof. Dr. Serkan Özgen

Eylül 2019, 86 sayfa

Bu çalışmada 2 boyutlu Euler denklemleri ile tanımlanan akışların çalışma kapsamında geliştirilmiş olan yapılsal olmayan ağlar için kesintili Galerkin sonlu eleman yöntemini kullanan çözücü ile elde edilen sonuçları sunulmuştur. Euler denklemleri ağdasız ve ısı alışverişi olmayan akışlar için geçerli olan hiperbolik denklemlerdir. Söz konusu denklemlerin kesintili Galerkin sonlu eleman yöntemi ile ayrıştırılması detaylı olarak anlatılmıştır. Kesintili Galerkin sonlu eleman yöntemi eleman bazlı olarak yüksek dereceli çözümü sağlar, dolayısıyla sadece ortak sınıra sahip eleman birbiri ile ilişkilendirilir. Eleman bazlı ve matematiksel operasyonlar tekrar ele alınmış ve gerekli olduğunda türetilmiştir. Kesintili Galerkin sonlu eleman yönteminini iki temel yaklaşımı arasından, şekilsel ve düğümsel, düğümsel yaklaşım tercih edilmiştir. Hacim ve yüzey integrali hesaplarında Gauss tümlevi kullanılmıştır. Elemanlar arası akı, sayısal akı fonksiyonları ile hesaplanmıştır. Literatürde yaygın olan birkaç akı fonksiyonun çözümleri karşılaştırma amaçlı olarak kullanılmıştır. Çözüm alanının sınırlarında uygun sınır koşulları uygulanmıştır. Literatürde yaygın olarak kullanılmışı tır. Kesintisiz problemlerde, yüksek dereceli hassasiyete sahip çözümlere rahatlıkla ulaşılmıştır. Fakat, şok ya da kesinti içeren problemlerde istikrarlaştırma teknikleri kullanılmıştır. İstikrarlaştırma teknikleri çözüm derecesini sınırlandırabilmekte ya da çözümün başarasını düşürebilmektedir. Çözücü ile elde edilen sonuçlar ve denklemlerin geçerliliğine uygun şekilde seçilmiş olan deney sonuçları arasında tatmin edici uyum yakalanmıştır. Bükülmüş duvar koşulu uygulamasının yüksek dereceli yöntemlerdeki önemi tecrübe edilmiştir. Dahası, şoklar etrafında çözüm ağı uyarlamasının önemine dikkat çekilmiştir.

Anahtar Kelimeler: Hesaplamalı Akışkanlar Dinamiği, Kesintili Galerkin, Yüksek Dereceli Çözüm, Sonlu Eleman, Euler Denklemleri, Aerodinamik To my dearest family

ACKNOWLEDGMENTS

I wish to express my deepest gratitude to my Thesis Supervisor Prof. Dr. Serkan Özgen for giving me opportunity to work with him and providing me flexibility throughout the thesis study.

I would like to thank to the members of dissertation committee for their valuable feedback; Prof. Dr. Yusuf Özyörük, Prof. Dr. Yusuf Uludağ, Assoc. Prof. Dr. Sinan Eyi, and Asst. Prof. Dr. Mustafa Kaya.

I would like to thank my high school and college friends for their support and friendship. I always feel so lucky to have such great friends in my life and they supported me so much throughout this study.

I would like to thank my colleagues for their friendship and providing an enjoyable working environment.

Finally, I would like to thank my parents for their support, trust, understanding and patience. Without their love and support the completion of this thesis would not have been possible.

TABLE OF CONTENTS

ABSTRACT
ÖZ
ACKNOWLEDGMENTS
TABLE OF CONTENTS xi
LIST OF TABLES
LIST OF FIGURES
LIST OF ABBREVIATIONS
CHAPTERS
1 INTRODUCTION
1.1 Motivation of the Study
1.2 Limitation of the Study
1.3 Outline of the Study
2 LITERATURE REVIEW
3 METHOD
3.1 Governing Flow Equations of Aerodynamics
3.1.1Non-Dimensional Form of Governing Flow Equations of Aero- dynamics13
3.2 Basics of Discontinuous Galerkin Finite Element Method Discretiza- tion in Space

3.3	Modes, Nodes and Vandermonde Matrix					
3.4	Basis Functions					
3.5	Modes and Nodes In 2-Dimension					
3.6	.6 Discontinuous Galerkin Finite Element Method Discretization of Eu- ler Equations					
3	3.6.1 Term I and Mass Matrix					
3	3.6.2	Term II and Stiffness Matrix				
3	8.6.3	Term III and Surface Mass Matrix				
3.7	Time	Discretization				
3.8	Nume	erical Flux				
3	8.8.1	Local Lax-Friedrichs Flux				
3	3.8.2	Roe's Approximate Riemann Solver				
3	3.8.3	Harten-Lax-van Leer Approximate Riemann Solver				
3.9	Boun	dary Conditions				
3	8.9.1	Dirichlet Boundary Condition				
3	8.9.2	Inflow Boundary Condition				
	3.9.2	2.1 Subsonic Inflow				
	3.9.2	2.2 Supersonic Inflow				
3	3.9.3	Outflow Boundary Condition				
3	8.9.4	Wall Boundary Condition				
	3.9.4	4.1 Curved Walls				
3	3.9.5 Riemann Invariant Farfield Boundary Condition					
3.10 Stabilization						

3.10.1 Filter	45			
3.10.2 Slope Limiter	46			
3.11 Post Processing	47			
4 RESULTS AND DISCUSSION	49			
4.1 Isentropic Vortex	49			
4.2 Bump in a Channel	53			
4.2.1 Bump with Straight Sided Cells	56			
4.3 RAE2822 Airfoil	61			
4.3.1 RAE2822 Airfoil at Mach 0.3	61			
4.3.2 RAE2822 Airfoil at Mach 0.73	63			
4.3.2.1 RAE2822 Airfoil at Mach 0.73 with Exponential Filter .	64			
4.3.2.2 RAE2822 Airfoil at Mach 0.73 with Slope Limiter	65			
4.4 NACA0012 Airfoil	67			
4.5 Supersonic Wedge	70			
5 CONCLUSION	77			
REFERENCES				

LIST OF TABLES

TABLES

Table 1.1 Comparison of Generic Properties of Most Widespread Numeri-			
cal Schemes, \checkmark stands for success, \thickapprox represents a short-coming and (\checkmark)			
means that method is originally not capable, but with modifications it is [1]			
Table 3.1 Low Storage Fourth Order Explicit Runge-Kutta Method Coeffi-			
cients [2]	36		
Table 4.1 Non-dimensional Values of Flow Parameters	49		

LIST OF FIGURES

FIGURES

Figure 3.1	Two Representative DG Elements			
Figure 3.2	Determinant of Vandermonde Matrix for Equidistant Nodes and			
Gauss-Lobatto Points with α Symmetric Jacobi Polynomials, $P_N^{\alpha,\alpha}(r)$.				
Figure 3.3	Legendre Polynomials of Various Orders			
Figure 3.4	Schematic of Mapping of an Arbitrary Cell for Basis Function			
Evalua	ation	24		
Figure 3.5	Complete Set of Basis Functions for 4th Order Approximation .	26		
Figure 3.6	$\alpha\text{-}\mathrm{optimized}$ and Equidistant Nodes for Various Orders in Equi-			
lateral and Standard Triangle				
Figure 3.7	Interior and Exterior Element Notation	32		
Figure 3.8	Numerical Flux Calculation	37		
Figure 3.9	Visual Comparison of Straight Sided Cell and Curved Cell	43		
Figure 3.10	Contribution of Riemann Invariants to Boundary State at Farfield			
Boundary for Subsonic Inflow, Supersonic Inflow, Subsonic Outflow				
and Su	apersonic Outflow	44		
Figure 3.11	Grid Elements and Sub-Elements for a Fifth Order Solution	47		
Figure 4.1	Density Contour for Isentropic Vortex at $t = 0$	51		

Figure	4.2	Meshes Used in the Convergence Analysis for Isentropic Vortex	
Problem			
Figure	4.3	L2 Norm of Density Error	53
Figure	4.4	Inviscid Smooth Bump in a Channel Problem	54
Figure	4.5 a Chai	Meshes Used in the Convergence Analysis for Inviscid Bump in number of the convergence of	55
Figure	4.6 2, 3 ar	L2 Norm of Entropy Error History for Polynomial Orders of 1, and 4 on Coarse Grid	56
Figure	4.7 Solutio	Steady State Pressure Contours of (a) Second and (b) Fifth Order	57
Figure	4.8	Convergence Rate for Bump in Channel Problem	58
Figure	4.9 Cell S	Convergence Rate Comparison of Curved and Straight Sided olutions	59
Figure	4.10 of 2, (Cell w mial C	Entropy Error Contours, (a) Curved Cell with Polynomial Order (b) Straight Sided Cell with Polynomial Order of 2, (c) Curved with Polynomial Order of 4, (d) Straight Sided Cell with Polyno- Order of 4	60
Figure	4.11	RAE2822 Airfoil Geometry [3]	61
Figure	4.12 tion G	Solution Grid Used in RAE2822 Airfoil Simulations (a) Solurid, (b) Zoomed View on Airfoil Geometry	62
Figure	4.13 (a) De	Convergence History for RAE2822 Airfoil Solution at $M = 0.3$ nsity Residual Plot, (b) Drag Force Convergence History	63
Figure	4.14 Numb	RAE2822 Solution at M = 0.3, (a) Pressure Contour, (b) Mach er Contour	63
Figure	4.15 2.82, (RAE2822 Solution with Exponential Filter at M = 0.73 and α = (a) Density Residual, (b) Pressure Coefficient [3]	64

Figure 4.16 RAE2822 Solution with Exponential Filter at M = 0.73 and α =
2.82, (a) Pressure Contour, (b) Mach Number Contour
Figure 4.17 RAE2822 Solution with Slope Limiter at M = 0.73 and α =
2.82, (a) Density Residual, (b) Pressure Coefficient [3]
Figure 4.18 RAE2822 Solution with Slope Limiter at M = 0.73 and α =
2.82, (a) Pressure Contour, (b) Mach Number Contour
Figure 4.19 NACA0012 Airfoil Geometry 68
Figure 4.20 Solution Grid Used in NACA0012 Airfoil Simulations (a) Solu-
tion Grid, (b) Zoomed View on Airfoil Geometry
Figure 4.21 NACA0012 Solution Mach Number Contour M = 0.8 and α =
1.25, (a) DG, (b) SU2
Figure 4.22 Pressure Coefficient for NACA0012 Solutions at M = 0.8, α =
1.25 degree
Figure 4.23 Supersonic Wedge Problem Definition 72
Figure 4.24 Grids for Supersonic Wedge Problem, (a) Uniform Grid 32x32,
(b) Uniform Grid 64x64, (c) Ramp Grid Refinement 1, (d) Ramp Grid
Refinement 2
Figure 4.25 Residual Histories of Supersonic Wedge Simulations, (a) Uni-
form Grid 32x32, (b) Uniform Grid 64x64, (c) Ramp Grid Refinement
1, (d) Ramp Grid Refinement 2
Figure 4.26 Pressure Distribution Along Wedge Wall, (a) Uniform Grid 32x32,
(b) Uniform Grid 64x64, (c) Ramp Grid Refinement 1, (d) Ramp Grid
Refinement 2
Figure 4.27 Entropy Error Contours, (a) Curved Cell with Polynomial Order
of 2, (b) Straight Sided Cell with Polynomial Order of 2, (c) Curved
Cell with Polynomial Order of 4, (d) Straight Sided Cell with Polyno-
mial Order of 4

LIST OF ABBREVIATIONS

ABBREVIATIONS

CFD	Computational Fluid Dynamics
DG,DGFEM	Discontinuous Galerkin Finite Element Method
FDM	Finite Difference Method
FEM	Finite Element Method
FVM	Finite Volume Method
HLL	Harten-Lax-van Leer
LDG	Local Discontinuous Galerkin
LES	Large Eddy Simulation
LGL	Legendre-Gauss-Lobatto
PDE	Partial Differential Equation
RANS	Reynolds Averaged Navier-Stokes
RK	Runge-Kutta
RKDG	Runge-Kutta Discontinuous Galerkin
SGS	Sub-Grid Scale

CHAPTER 1

INTRODUCTION

Understanding and solving the flow problems have enormous importance in the aerospace industry. Experimental and theoretical approaches are the two historically major branches of fluid dynamics. Experimental methods help to understand and evaluate the physical phenomena in fluid dynamics however it is highly expensive to be utilized in each design process. Solving the governing equations or simplified forms is the part of the theoretical approach but analytical solutions exist for only a few simple problems. Thanks to growth in computation power over the last decades, a third approach, numerical approach known as computational fluid dynamics (CFD) has gained importance and become a standard tool for engineering and science applications. Mostly finite volume and finite element, rarely finite difference are the methods employed in the CFD solvers. The finite volume and finite difference are generally based on second order accuracy which means that as the grid spacing, h, decreases, solution error decreases in the order of h^2 . The finite element method solvers can provide high order solutions however limited to flows not involving discontinuities. It would be beneficial to give a brief review on the classical discretization methods before stating the motivation of the work and going into details of high order method.

The finite difference method employs the simplest approach to discretize the differential or divergence form of the integral equations of a conservation law. Replacing the analytical derivatives with discrete ones and obtaining a numerically soluble discrete problem is the main idea. There are numerous ways for the discrete approximations of derivatives. For the details of methodologies, Hirsch [4] and Ferziger & Peric [5] are great examples of textbooks to visit. The main advantages of the finite difference method are extreme efficiency of the scheme in terms of computational cost and ease of programming. On the other hand, finite difference applications are generally limited to structured grids hence for complex geometries, block structured grid generation which requires tremendous effort is necessary. In theory, derivation of a finite difference scheme on unstructured grids is potential however, a very complex problem, reconstruction of polynomial functions on unstructured grids is mandatory. High order finite difference scheme, in other words high order approximations of derivatives are easy to obtain by extending the number of points in the stencil. However, in order to satisfy stability requirements smooth and regular grids are required hence grid generation again needs special attention [6]. Another important subject to mention is that finite difference schemes have stability issues with problems comprising discontinuities.

Formulation of finite volume schemes are based on the integral form of the conservation laws in contrast to finite difference schemes. Fluxes through the cell boundaries have to be computed in finite volume methods rather than evaluating discrete derivatives. Because the main concern is to evaluate fluxes, there are lots of methodologies. Among those, upwind method [7], [8], [9] is the one which is very popular for convection dominated problems where characteristics of wave propagation decide the flux choice. The advantage of the finite volume methods is that method is highly suitable for unstructured grids which ease the evaluation of complex geometries. Moreover, finite volume schemes handle the discontinuities successfully. On the other hand, major drawback of the finite volume method is that high order schemes on unstructured grids require a large computation stencil. Resulting scheme with large stencil becomes exceptionally complex for programming and expansive in terms of computation cost. Furthermore, for parallel applications, large computation stencils reduce the efficiency of parallelization by requiring too much information exchange between nodes. To sum up, finite volume methods are highly suitable for evaluation of complex geometries however construction of high order schemes on unstructured grids is a fundamental problem.

The finite element discretization technique is rather different than other discretization techniques. In finite element discretization, differential equations are multiplied by an arbitrary test or weight function and integrated by parts. Linear combinations of ansatz or shape functions, generally piecewise polynomials, are used to construct discrete solution. Type of the finite element method is identified by choice of weight and shape function. For example; in Galerkin finite element methods, weight and shape function is chosen from same space or in Least Square finite element methods, each weight function is selected as the derivative of the corresponding shape function. Basically, finite element methods can be classified as continuous and discontinuous. Discontinuous methods do not seek for a continuous global solution. Discrete solution is continuous within each element however, non-conforming at the element boundaries. Since present work is about discontinuous Galerkin method, details are provided in later sections. Continuous finite element methods work well for continuous and smooth flow problems. For convection dominated flows, spurious oscillations occur resulting lack of stability. Numerous methodologies have been developed to provide additional stability to continuous finite element discretization. Artificial dissipation [10], [11]; streamline upwind Petrov-Galerkin [12], [13]; Subgrid Scale [14], [15] are examples of the methodologies. Unlike the finite volume method, a high order finite element scheme is easy to derive since the degree of polynomial weight and shape function determines the accuracy of the method. Another advantage of the finite element scheme is that method is well suited for grid as well as polynomial accuracy adaptation known as hp-adaptation thanks to compact discretization stencil. Drawback of the continuous finite element scheme is confronted when explicit time integration is employed. Explicit time integration is mandatory when unsteady simulations are required. Due to continuity requirement at element boundaries, a coupled system of equations has to be solved at each time step. However, this is not valid for discontinuous Galerkin discretization like in finite difference and finite volume methods.

Keeping in mind the aforementioned methods, a promising numerical scheme should provide geometric flexibility while supporting locally adapted resolution. As in the finite element method, capturing high-order accuracy through local approximation is highly motivating feature. However, for wave dominated problems, potential stability problems are introduced and the global Galerkin statement causes loss of the scheme locality. Nonetheless, there are advantageous features of finite volume method developed for the stated problem. The discontinuous Galerkin finite element method arouse from the combination of the basis and test functions utilizing the space as in the finite element method and a similar approach used in the finite volume method to satisfy equations. Hence, discontinuous Galerkin method houses the advantageous features of the both methodologies such as supporting complex geometries while providing high accuracy for conservation and elliptic problems.

Finally, generic properties of the most widespread numerical schemes for flow problems are compared in Table 1.1 [1] as a summary.

Table 1.1: Comparison of Generic Properties of Most Widespread Numerical Schemes, \checkmark stands for success, \bigstar represents a short-coming and (\checkmark) means that method is originally not capable, but with modifications it is [1]

			EXPLICIT		
	COMPLEX	HIGH ORDER		CONSERVATION	ELLIPTIC
	GEOMETRIES	ACCURACY	SEMI-DISCRETE FORM	LAWS	PROBLEMS
		,	,	,	,
FDM	X	√	\checkmark	\checkmark	~
FVM	\checkmark	×	\checkmark	\checkmark	(✓)
FEM	\checkmark	\checkmark	×	(✔)	1
DG-FEM	1	\checkmark	\checkmark	\checkmark	(✔)

1.1 Motivation of the Study

The traditional methods in industrial applications have at most second order accuracies and some fundamentals problems at achieving high order accurate solutions as it is aforementioned. Moreover, in industry, there are still problems that current methods do not provide satisfactory results. The majority of current CFD methods are not adequate for vortex dominated and translational flows like; rotorcraft technologies, high-lift systems and formation flights [16]. Other concerns with the low order methods are high quality grid requirement, grid generation time and time of the simulation. Fidkowski [17] provided following equation relating the total time for a solution with numerical method parameters as:

$$\log T = d(-\frac{1}{p}\log E + w\log p) - \log F + constant$$
(1.1)

T is the total time for a solution, d is the spatial dimension of the problem, p is the order of accuracy, E is the error in the solution, w is the complexity of the solution algorithm, $\frac{1}{F}$ is the time required to complete a single operation. When the desired accuracy is considerably high, i.e. E is really small and solution algorithm is moderately complex, it is expected that $\log E$ term will be more dominant than the $\log p$ term. Hence, dependency of total time required on p and d is exponential. Moreover, little change in p or w can be as significant as increasing computational power due to inverse relation between total time, T and computational speed, F.

Based on the statements made up to here, this work is motivated to develop a high order flow solver. Although there are various high order methods available in literature, for example, Flux Reconstruction, Spectral Volume, etc., Discontinuous Galerkin method is selected as the numerical scheme due to its well developed and detailed literature allowing the development of high-order accurate stable discretization for convection dominated problems. Moreover, having the element-wise compact stencil eases obtaining high-order accuracy on the unstructured grids and efficient solution strategies. Lastly, it should be noted that intention is not to develop a new numerical scheme or improvement on the method, it is an assessment of Discontinuous Galerkin method for future studies which should be able to overcome current second-order schemes used in industrial applications.

1.2 Limitation of the Study

The study of this thesis is basically limited by the fidelity of the governing fluid flow equations. The solution fidelity of the governing equations is determined by employed numerical scheme. Scope of the present work covers the two dimensional Euler equations which govern compressible inviscid fluid flow. Hence, the application of the work is limited to two dimensional inviscid flows at subsonic, transonic and supersonic flow regimes. Numerical scheme is employed on unstructured triangular grids. Implemented boundary conditions are non-reflecting Riemann far-field boundary, wall boundary, inlet boundary and outlet boundary; therefore problems to be simulated are subject to limitation of available boundary conditions.

1.3 Outline of the Study

In Chapter 2, literature review of Discontinuous Galerkin Finite Element Methods is given. Brief information on history of method is provided. Main problems and breakthroughs to their solutions are addressed and finally, up-to-date situation and applications are mentioned.

Chapter 3 provides the details of the formulation for Discontinuous Galerkin Finite Element Method and mathematical operations required to support methodology.

Chapter 4 consists of results and discussions. Problems with analytical solutions and test cases in the literature are simulated and compared.

Chapter 5 finalizes the thesis and provides conclusions to topics cover in the present work. Strategies for future work are also stated in Chapter 5.

CHAPTER 2

LITERATURE REVIEW

In 1973, Reed and Hill [18] made the first introduction of discontinuous Galerkin method for neutron transport problem on triangular meshes. One year later, LeSaint and Raviart [19] published their work on DG for linear hyperbolic problems, error estimation and proof of rates of convergence. However, for equations associated with CFD and being non-linear, developments on the high order discontinuous Galerkin method were necessary. By using DG method, Chavent and Salzano [20] were the first to solve non-linear hyperbolic problem employing Godunov's flux at cell interfaces. The breakthrough on the solution of time dependent non-linear hyperbolic equations was carried out by Cockburn and Shu in a series of publications [21], [22], [23], [24]. The work is considered as a breakthrough because finding a time discretization that would result in a stable, efficient and formally high order method was the main difficulty of development discontinuous Galerkin method at that time [25]. The Euler equations were discretized by using Time Variation Diminishing Runge-Kutta scheme in time and, Discontinuous Galerkin scheme in space. Independently from Cockburn and Shu, Allmaras and Giles [26] and Allmaras [27] developed a numerical method, now would be classified as discontinuous Galerkin method, for Euler equations of gas dynamics. Van Leer's method of moments [28] was extended for Euler equations and second order of accuracy was achieved in their work. Later, Halt [29] generalized the method for high order accuracies.

Solution of Navier-Stokes equations requires the handling of elliptic operator. In 1978 by Wheeler [30] and 1982 by Arnold [31], penalty methods, a version of discontinuous finite element methods, are introduced for pure elliptic operators. Application of interior penalty methods were presented for convection-diffusion problems

in [32], [33]. Bassi and Rebay [34] developed an alternative method by rewriting the Navier-Stokes equations as a system of first order differential equations. This method is known as first method of Bassi-Rebay. However, this scheme results in extended stencil on the contrary to the advantage of discontinuous Galerkin method. A slight modification reduced the stencil to nearest neighbor and known as second method of Bassi-Rebay [35]. Cockburn and Shu improved their method RKDG for compressible Navier-Stokes equations with the motivation from pioneering work of Bassi and Rebay. The new methodology was named as local discontinuous Galerkin method. Another technique introduced by Peraire and Persson is compact discontinuous Galerkin method [36] which is closely related to LDG and increases compactness of the stencil in higher dimensions.

Solutions with discontinuities require stabilization techniques like in classical methods in order to prevent spurious oscillations. First stabilization application on discontinuous Galerkin method was the slope limiting in the work of Cockburn and Shu as mentioned before. Another way of stabilizing the discontinuous Galerkin method around shocks or discontinuities is the implementation of artificial viscosity term. There are various strategies for application of artificial viscosity such as piece-wise constant viscosity, PDE-based viscosity, etc. Details are presented in references [37], [38], [39], [40], however, Burgess [41] suggests the use of the methodologies in [38], [36] due to robustness concerns. A popular alternative is the hp-adaptation where h stands for grid size and p represents the order of accuracy. Main strategy of the hp-adaptation is to increase the grid density and to decrease the order of solution accuracy near the discontinuities. The challenge of the approach is to determine the weights between h and p adaptation however, capability of the hp-adaptation has been shown by several studies [42], [43], [44].

The turbulent flow simulations with discontinuous Galerkin method was first accomplished by Bassi and Rebay [35]. They solved the RANS equations coupled with $k^{\neg}\omega$ closure model [45]. Spalart – Allmaras turbulence model equation was studied with compact discontinuous Galerking discretization [46] and artificial viscosity on model working variable are utilized due to negative values of model variable causing instability in numerical solution. Modifications to original model were proposed to alleviate stability problems [47]. Application of discontinuous Galerkin method to DNS first took step in the work presented by Collis [48]. Exploration of its advantageous dissipation and dispersion properties for DNS lead to increasing interest in studies of turbulent flow simulations [49], [50], [51]. Discontinuous Galerkin method has also found applications in LES due to its spectral cut-off filter like dissipation behavior, plays the role of subgrid-scale (SGS) model similar to Smagorinsky model which is a proper use in classical LES models [52], [53], [54], [55], [56].

Reaching adequate maturity level for flow simulations and increasing interest in high order methods lead to development of open-source flow solvers which use discontinuous Galerkin discretization. Beyond any doubt, one of the most popular opensource CFD solver package is Stanford University Unstructured (SU2) [57]. SU2 was initially based on finite volume and finite element methods however, latest versions includes nodal discontinuous Galerkin scheme. Implementation of discontinuous Galerkin method in another popular package, OpenFOAM [58], is presented as HopeFOAM [59]. FLEXI is a parallel high-order numerical framework for solving PDEs, with a focus on Computational Fluid Dynamics and based on the Discontinuous Galerkin Spectral Element Method (DGSEM), which allows for high-order of accuracy and fully unstructured hexahedral meshes [60]. Nektar++ is an open-source software framework designed to support the development of high-performance scalable solvers for partial differential equations using the spectral/hp element method and features discontinuous Galerkin projections [61]. FEniCS is a popular opensource computing platform for solving partial differential equations using finite element discretization and supports discontinuous Galerkin method [62]. Distributed and Unified Numerics Environment (dune-fem) is a discretization module providing an implementation of mathematical abstractions to solve PDEs on parallel computers including local grid adaptivity, dynamic load balancing, and higher order discretization schemes [63].

CHAPTER 3

METHOD

This chapter is devoted to the theories utilized in the development of 2 dimensional discontinuous Galerkin finite element method flow solver. The formulation and notation of discretization in the present study is followed from the reference [1]. Before going into details of mathematical representations, it would be beneficial to give basic ideas in simplified words. Discontinuous Galerkin method can be thought as a good combination of finite element and finite volume methods. Consider the cells with a field variable presented in Figure 3.1. It can be observed that field variable is presented as a continuous smooth function inside the elements as in the finite element method, however, it is discontinuous across element interfaces. The information between the elements passed into each other using numerical flux functions well developed for the finite volume methods. But constructing a well conditioned solution representation is not straightforward, using arbitrary functions leads to numerical instabilities hence a good orthogonal basis function set has to be selected. In this work, Legendre polynomials are utilized. Legendre polynomials are one dimensional polynomials and defined in a reference element which spans [-1, 1]. Hence, one has to consider the ways to map arbitrary elements into reference element and represent polynomials on two dimension. To sum up procedure, calculations start with mapping of mesh elements. Then, local solutions are constructed on reference element using basis function set. Next step is to evaluate semi-discrete form of the governing equations containing numerical flux and turning back to physical space. Semi-discrete form provides ordinary differential function in time, hence time marching is utilized until convergence is obtained in the solution domain.

Finally, outline of the chapter is as follows; governing equations and non-dimensional

forms are presented. Basics of the discontinuous Galerkin method discretization in one dimension is given in detail. Later, selection of modes and nodes are justified and Vandermonde matrix is introduced. After gathering all the computational tools, discretization of governing equations are carried out. Numerical flux functions are presented and provided with necessary references. Boundary conditions are detailed and implementation is discussed. As last of section of the chapter, the post-processing methodology is thought to worth be to mentioned.



Figure 3.1: Two Representative DG Elements

3.1 Governing Flow Equations of Aerodynamics

Euler equations govern the inviscid and adiabatic flows with a set of quasi-steady hyperbolic equations. In the differential form, 3 dimensional Euler equations are given below:

$$\frac{\partial U}{\partial t} + \nabla \cdot F_i(U) = 0 \tag{3.1}$$

where U is the conservative state vector:

$$U = \begin{cases} \rho \\ \rho u \\ \rho v \\ E \end{cases}$$
(3.2)

where ρ is the fluid density, (u, v) are the Cartesian components of the velocity field, E is the specific total energy which is composed of specific internal energy, e, and specific kinetic energy.

$$E = e + \frac{1}{2}(u^2 + v^2)$$
(3.3)

 F_i is the inviscid flux tensor:

$$F_i = \left\{ F_i^x, F_i^y \right\} \tag{3.4}$$

$$F_{i}^{x} = \begin{cases} \rho u \\ \rho u^{2} + P \\ \rho v u \\ \rho E u + P u \end{cases}, F_{i}^{y} = \begin{cases} \rho v \\ \rho u v \\ \rho v^{2} + P \\ \rho E v + P v \end{cases}$$
(3.5)

One more equation is required to close the system. Further relations under the thermally and calorically perfect gas assumption are given for the total enthalpy, H, as follows:

$$H = E + \frac{P}{\rho} \tag{3.6}$$

and the equation of state provides:

$$\frac{P}{\rho} = \frac{\gamma - 1}{\gamma} (H - \frac{1}{2}(u^2 + v^2))$$
(3.7)

where γ is the ratio of specific heats and assumed as constant value of 1.4.

3.1.1 Non-Dimensional Form of Governing Flow Equations of Aerodynamics

When working with dimensional variables, all variables will resolve in different order of magnitudes. This will result in ill conditioned linear systems of implicit time marching schemes, precision errors, etc. hence equations are normalized in order to catch the same order of magnitude. Non-dimensional form of the Euler equations reads:

$$\bar{t} = \frac{v_{ref}t}{L_{ref}}, \bar{\rho} = \frac{\rho}{\rho_{ref}}, \bar{P} = \frac{P}{\rho_{ref}v_{ref}}, \bar{e} = \frac{e}{v_{ref}^2}, \bar{H} = \frac{H}{v_{ref}^2},$$

$$\bar{a} = \frac{a}{v_{ref}}, \bar{u} = \frac{u}{v_{ref}}, \bar{v} = \frac{v}{v_{ref}}$$
(3.8)

The flow related reference values are taken from free-stream values, dimension related reference values are chosen according to problem to be solved. Finally,

$$\frac{\partial \bar{U}}{\partial \bar{t}} + \bar{\nabla} \cdot F_i(\bar{U}) = 0$$
(3.9)

In the solutions of the problems, non-dimensional form of the equations are used however, after this point bar over the variables in the equations are not shown.

3.2 Basics of Discontinuous Galerkin Finite Element Method Discretization in Space

Basic ideas of discontinous Galerkin discretization are presented in this section. An one dimensional general, scalar, non-linear conservation law is given below with appropriate initial conditions,

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = 0$$

$$\in \Omega = [x_l, x_r], \ u(x, 0) = u_0(x)$$
(3.10)

The discretization begins with dividing the domain, Ω , into K non-overlapping elements,

$$\Omega = \bigcup_{k=1}^{K} E^{k}, x \in E^{k} = [x_{l}^{k}, x_{r}^{k}]$$
(3.11)

The local solutions are defined as piecewise polynomials on the elements.

x

$$x \in E^{k} : u_{h}^{k}(x,t) = \sum_{n=1}^{N_{p}} \tilde{u}_{n}^{k}(t)\psi_{n}(x) = \sum_{i=1}^{N_{p}} u_{h}^{k}(x_{i}^{k},t)l_{i}^{k}(x)$$
(3.12)

where $N_p = N + 1$ and N is the order of the polynomial. The local solution is given as two different expressions. The first expression is known as modal form and second expression is nodal form. Therefore, $\tilde{u}_n^k(t)$ are modal coefficients and $u_h^k(x_i^k, t)$ are nodal coefficients. $\psi_n^k(x)$ is the polynomial basis of order N while $l_i(x)$ represents the interpolating Lagrange polynomial on the element E^k . One can imagine the modal expression as a polynomial fit for the solution on the element while nodal expression as a spline representing the solution through grid points x_i^k . Mathematically, these two expressions are equivalent, however, there are advantageous and disadvantageous properties of them for the practical use. The modal form provides ease of changing approximation order hence, very practical for *p*-multigrid applications. However, nodal form is favourable when dealing with surface integrals and boundary condition implementations. Global solution *u* is approximated as follows:

$$u = u(x,t) \simeq u_h(x,t) = \bigoplus_{k=1}^{K} u_h^k(x,t)$$
(3.13)

where the symbol \bigoplus direct sum operator hence, $u_h(x,t)$ is the approximation of global solution which is direct sum of the approximate local solution, $u_h^k(x,t)$, on element E^k .

It is a good point to introduce the test functions. Since discontinous Galerkin is a finite element method, local solutions are recovered by using test functions. Furthermore, test functions are also defined in the same space with basis functions as the definition of Galerkin approach. Define the global space of test functions,

$$\phi_h \in V_h = \bigoplus_{k=1}^K V_h^k \tag{3.14}$$

$$\phi_h^k \in V_h^k = span\psi_n(E^k)_{n=1}^{N_p}$$
(3.15)

where ϕ_h^k is locally defined test function and given as,

$$x \in E^k : \phi_h^k(x) = \sum_{n=1}^{N_p} \tilde{\phi}_n^k \psi_n(x)$$
 (3.16)

Discretization procedure is continued with forming the residual and it should vanish,

$$R_h(x,t) = \frac{\partial u_h}{\partial t} + \frac{\partial f_h(u)}{\partial x}$$
(3.17)

All test functions defined in V_h are required to be orthogonal to residual. The resulting local statement in all elements is as below,

$$\int_{E^k} R_h(x,t)\phi_n(x)dx = \int_{E^k} \left(\frac{\partial u_h}{\partial t} + \frac{\partial f_h(u)}{\partial x}\right)\phi_n(x)dx = 0, 1 \le n \le N_p \quad (3.18)$$

On each element, there are N_p unknowns and N_p equations to be solved. However, there are still a need for developments required to obtain global solution because imposing the boundary conditions and constraints on the basis and test functions are not addressed yet.

Let's continue with applying integration by parts to equation 3.18 in space. The integration holds the assumption of smooth but discontinuous test function. In other words, test function is smooth inside the elements however, constrained through element interfaces.

$$\int_{E^k} \frac{\partial u_h}{\partial t} \phi_n(x) dx - \int_{E^k} f_h(u) \frac{d\phi_n(x)}{dx} dx + \oint_{\partial E^k} \vec{n} f_h(u) \phi_n(x) dx = 0, \ 1 \le n \le N_p$$
(3.19)

where \vec{n} stands for the normal vector of the element surface. Notice that the solution has multiple valued at the element interfaces, hence evaluation of surface integral in equation 3.19 has to be further investigated. Rather than using flux values directly at the interfaces, a numerical flux, which is combination of neighbour solutions, is utilized in the element interfaces. Then, defining the numerical flux as $f_h^*(u)$, the equation becomes:

$$\int_{E^k} \frac{\partial u_h}{\partial t} \phi_n(x) dx - \int_{E^k} f_h(u) \frac{d\phi_n(x)}{dx} dx = -\oint_{\partial E^k} \vec{n} f_h^*(u) \phi_n(x) dx, \ 1 \le n \le N_p$$
(3.20)

This form is known as the weak form of the problem. For each element, E^k , there are N_p unknowns and N_p equations, hence global solution now can be achieved. Righthand side of the equation 3.20 is responsible for obtaining the global solution from local elements and employing the boundary conditions. Therefore, numerical flux has the key role and it is needed to be detailed later.

Consider the modal expansion given in equation 3.12 and select the basis and test functions from same space, that is classical Galerkin approach, and apply to equation 3.20. Then, local semi-discrete weak form is obtained:

$$\mathcal{M}^{k} \frac{d}{dt} \tilde{\boldsymbol{u}}_{h}^{k} - (\boldsymbol{\mathcal{S}}^{k})^{T} \tilde{\boldsymbol{f}}_{h}^{k} = -f^{*}(\boldsymbol{u}_{h})\boldsymbol{\psi}(\boldsymbol{x}_{r}^{k}) + f^{*}(\boldsymbol{u}_{h})\boldsymbol{\psi}(\boldsymbol{x}_{l}^{k})$$
(3.21)

where

$$\mathcal{M}_{ij}^{k} = \int_{E^{k}} \psi_{i} \psi_{j} dx, \ \mathcal{S}_{ij}^{k} = \int_{E^{k}} \psi_{i} \frac{d\psi_{j}}{dx} dx$$
(3.22)

and

$$\begin{aligned} \tilde{\boldsymbol{u}}_{\boldsymbol{h}}^{\boldsymbol{k}} &= [\tilde{u}_{1}^{k}, \tilde{u}_{2}^{k}, ..., \tilde{u}_{N_{p}}^{k}]^{T}, \\ \tilde{\boldsymbol{f}}_{\boldsymbol{h}}^{\boldsymbol{k}} &= [\tilde{f}_{1}^{k}, \tilde{f}_{2}^{k}, ..., \tilde{f}_{N_{p}}^{k}]^{T}, \\ \boldsymbol{\psi} &= [\psi_{1}(x), \psi_{2}(x), ..., \psi_{N_{p}}(x)]^{T} \end{aligned}$$
(3.23)

represent the local solution, local flux and local test function vectors.

Similarly for nodal representation of equation 3.12, local semi-discrete weak form would be:

$$\mathcal{M}^{k} \frac{d}{dt} \boldsymbol{u}_{h}^{k} - (\boldsymbol{\mathcal{S}}^{k})^{T} \boldsymbol{f}_{h}^{k} = -f^{*}(\boldsymbol{u}_{h}) \boldsymbol{l}^{k}(\boldsymbol{x}_{r}^{k}) + f^{*}(\boldsymbol{u}_{h}) \boldsymbol{l}^{k}(\boldsymbol{x}_{l}^{k})$$
(3.24)

where

$$\mathcal{M}_{ij}^{k} = \int_{E^{k}} l_{i}^{k} l_{j}^{k} dx, \quad \mathcal{S}_{ij}^{k} = \int_{E^{k}} l_{i}^{k} \frac{dl_{j}^{k}}{dx} dx \qquad (3.25)$$

and

$$\boldsymbol{u_{h}^{k}} = [u_{1}^{k}, u_{2}^{k}, ..., u_{N_{p}}^{k}]^{T},$$

$$\boldsymbol{f_{h}^{k}} = [f_{1}^{k}, f_{2}^{k}, ..., f_{N_{p}}^{k}]^{T},$$

$$\boldsymbol{l^{k}} = [l_{1}^{k}(x), l_{2}^{k}(x), ..., l_{N_{p}}^{k}(x)]^{T}$$
(3.26)

represent the nodal solution, nodal flux and nodal test function vectors. Finally, the schemes presented as equations 3.21 and 3.24 are the classical modal and nodal discontious Galerkin finite element methods given in weak form.

3.3 Modes, Nodes and Vandermonde Matrix

The global solution is represented as the direct sum of local piecewise polynomial solutions as formulated in equation 3.11 and local solution is assumed as a expansion of polynomial functions. Therefore, selection of the polynomials plays vital role in the success of the scheme. The accuracy of the scheme is directly affected by the order of the polynomials.

Consider the modal expansion for a one dimensional problem on an interval I = [-1, 1],

$$u_{h}^{k}(x,t) = \sum_{i=1}^{N_{p}} \tilde{u}_{i}^{k}(t)\psi_{i}^{k}(x)$$
(3.27)

As a simple choice, monomial basis can be used as the basis function for modal expansion,

$$\psi_n(x) = x^{n-1} \tag{3.28}$$

In order to obtain $\tilde{u}_i^k(t)$, for each N_p basis functions, L^2 projection is applied and requires that,

$$(u_i^k(x,t),\psi_n(x))_I = \sum_{i=1}^{N_p} \tilde{u}_i^k(t)(\psi_i(x),\psi_n(x))_I$$
(3.29)

The inner product on the interval I is introduced as follows:

$$(u,v)_I = \int_{-1}^1 uv dx \tag{3.30}$$

Then corresponding result would be:

$$\mathcal{M}\tilde{u}_{i}^{k}(t) = (u^{k}(x,t),\psi_{i}(x))_{I}, i = 1, 2...N_{p}$$
(3.31)

where $\mathcal{M}_{ij} = (\psi_i(x), \psi_j(x))_I$ and provides N_p unknowns with N_p equations to recover $u_i^k(x, t)$. Inner product $(\psi_i(x), \psi_j(x))_I$ gives the matrix \mathcal{M} as follows:

$$\mathcal{M}_{ij} = \frac{1}{i+j-1} (1+(-1)^{i+j})$$
(3.32)

The resultant matrix resembles a Hilbert matrix. Hilbert matrix is known to be poorly conditioned due to fact that $(i + j - 1)^{-1}$ term construct linear dependency as the (ij) increases. The accurate calculation of $u_i^k(x,t)$ is abortive hence, monomial basis is not a good choice for the polynomial expansion. The problem is overcome by the use of a suitable orthonormal basis. Utilizing an L^2 based Gram-Schmidt orthogonalization approach on the monomial basis provides[64]:

$$\psi_n(x) = \tilde{P}_{n-1}(r) = \frac{P_{n-1}(r)}{\sqrt{\gamma_n - 1}}, \ \gamma_n = \frac{2}{2n+1}$$
(3.33)

where $P_n(r)$ is the classical *n*th order Legendre polynomial which is a special kind of Jacobi polynomial and defined on the reference element $r \in I = [-1, 1]$. Hence, for an one dimensional arbitrary element, E^k an affine mapping is introduced:

$$x \in E^{k} : x(r) = x_{l}^{k} + \frac{1+r}{2}(x_{r}^{k} - x_{l}^{k})$$
(3.34)

Using Legendre polynomials as basis functions, the mass matrix, M becomes identity, therefore problem of ill conditioned matrix is solved.
Now, it is useful to reiterate that modal and nodal bases are on the same approximation space. The difference arise from the representation. The relation between two representation is easy to construct. Consider the modal basis coefficients, $\tilde{u}_i^k(t) = [\tilde{u}_1^k(t), \tilde{u}_2^k(t), ..., \tilde{u}_{N_p}^k(t)]$ and nodal values are given as follows:

$$u_i^k(x_i, t) = \sum_{j=1}^{N_p} \tilde{u}_j^k(t) \psi_j(x_i)$$
(3.35)

In matrix form,

$$u_i^k(x_i, t) = \mathcal{V}\tilde{u}_j^k(t), \ \mathcal{V}_{ij} = \psi_j(x_i)$$
(3.36)

The matrix \mathcal{V} is known as Vandermonde matrix and it provides direct relation between modal and nodal representations. For a stable conversion between modal-to-nodal and nodal-to-modal, Vandermonde matrix should be well conditioned on the nodal points x_i . Hence, next step is to determine the points to define Vandermonde matrix. Since Legendre polynomials are determined as optimal basis,

$$u_{h}^{k}(x(r),t) = \sum_{i=1}^{N_{p}} \tilde{u}_{i}^{k}(t)\tilde{P}_{n-1}(r)$$
(3.37)

and as an interpolant at grid points ξ_i ,

$$u_h^k(x(r), t) = \sum_{i=1}^{N_p} u_i^k(\xi_i, t) l_i(r)$$
(3.38)

where the interpolating Lagrange polynomial is defined with the property $l_i(r_j) = \delta_{ij}$,

$$l_{i}(r) = \prod_{\substack{j=1\\j\neq i}}^{N_{p}} \frac{r-\xi_{j}}{\xi_{i}-\xi_{j}}$$
(3.39)

As long as ξ_i 's are distinct, $l_i(r)$ is unique and exists. In order to evaluate quality of the interpolant compared to best polynomial approximation, Lebesgue constants are utilized. Lebesgue function and constant are defined respectively,

$$\lambda_n(r) = \sum_{i=1}^{N_p} |l_i(r_n)|$$
(3.40)

$$\lambda = \max_{r} \lambda_n(r) \tag{3.41}$$

It should be noted that Lebesgue function, λ_n , is constructed using cardinal functions hence it is solely dependent on the grid points, ξ_i and independent of polynomial basis. Since it is determined that,

$$\mathcal{V}^T l(r) = \tilde{P}(r) \tag{3.42}$$

the particular solution, l for minimum Lebesque constant, is given in the zeros of the following form [65],

$$f(r) = (1 - r^2)\tilde{P}'_N(r)$$
(3.43)

The points satisfying equation 3.43 are the Legendre-Guass-Lobatto quadrature points and in close relation with normalized Legendre polynomials.



Figure 3.2: Determinant of Vandermonde Matrix for Equidistant Nodes and Gauss-Lobatto Points with α Symmetric Jacobi Polynomials, $P_N^{\alpha,\alpha}(r)$

The effect of node selection on the determinant of Vandermonde matrix is presented in Figure 3.3. Moreover, other than LGL nodes, which are special case of α being zero, various Gauss-Lobatto nodes for the symmetric Jacobi polynomials are investigated. The Figure 3.3 confirms that LGL nodes provides maximum determinant value increasing with order and other α values show similar behaviour. However, equidistant nodes exhibit completely different pattern. After a certain polynomial order, equidistant nodes cannot keep the increasing profile, on the contrary start to decay.

As a summary of section, importance of selecting proper basis functions for local approximations are stated. Local approximations take the following form starting from equation 3.12,

$$u_{h}^{k}(r,t) = \sum_{n=1}^{N_{p}} \tilde{u}_{n}^{k}(t)\tilde{P}_{n-1}(r) = \sum_{i=1}^{N_{p}} u_{h}^{k}(r_{i}^{k},t)l_{i}^{k}(r)$$
(3.44)

where r_i are the Legendre-Gauss-Lobatto nodes which are proved to have significant effects on success of the scheme. Vandermonde matrix is introduced and ensured to be well-behaved by choosing orthonormal basis of Legendre functions and LGL nodes. The connection between modal and nodal representations is established through Vandermonde matrix and given as follows,

$$\mathcal{V}\tilde{\mathbf{u}}^{k} = \mathbf{u}_{h}^{k}, \, \mathcal{V}^{T}\mathbf{l}(r) = \tilde{\mathbf{P}}(r), \, \, \mathcal{V}_{ij} = \tilde{P}_{j-1}(r_{i})$$
(3.45)

3.4 Basis Functions

The approximate solution of discontinuous Galerkin discretization is presented as a series of basis functions with corresponding coefficients. The importance of selecting a proper basis functions set is provided in section 3.3. In the present work, Jacobi polynomials, which are orthonormal and hierarchic, are selected. Orthonormal property is crucial for constructing a numerically stable discretization. A hierarchical polynomial family has the property such that a Nth order polynomial contains all the N - 1th polynomials. The solution of singular Sturm-Liouville eigenvalue problem given in equation 3.46 is the classical *n*th order Jacobi polynomials, $P_n^{(\alpha,\beta)}(x)$.

$$x \in [-1,1], \frac{d}{dx}(1-x^2)w(x)\frac{d}{dx}P_n^{(\alpha,\beta)}(x) + n(n+\alpha+\beta+1)w(x)P_n^{(\alpha,\beta)}(x) = 0$$
(3.46)

where w(x) is weight function and defined as follows:

$$w(x) = (1 - x)^{\alpha} (1 + x)^{\beta}$$
(3.47)

Jacobi polynomials has an important property given as [64]

$$\frac{d}{dx}P_{n}^{(\alpha,\beta)}(x) = \sqrt{n(n+\alpha+\beta+1)}P_{n-1}^{(\alpha+1,\beta+1)}(x)$$
(3.48)

The Jacobi polynomials has no noted formula, however equation 3.46 and 3.48 provides a recurrence relation to evaluate:

$$xP_{n}^{(\alpha,\beta)}(x) = a_{n}P_{n-1}^{(\alpha,\beta)}(x) + b_{n}P_{n}^{(\alpha,\beta)}(x) + a_{n+1}P_{n+1}^{(\alpha,\beta)}(x)$$
(3.49)

where a_n is given by:

$$a_n = \frac{2}{2n+\alpha+\beta} \sqrt{\frac{n(n+\alpha+\beta)(n+\alpha)(n+\beta)}{(2n+\alpha+\beta-1)(2n+\alpha+\beta+1)}}$$
(3.50)

and b_n is given by:

$$b_n = -\frac{\alpha^2 - \beta^2}{(2n + \alpha + \beta)(2n + \alpha + \beta + 2)}$$
(3.51)

The recurrence relation has the n and n-1 terms for n+1th order polynomial hence $P_0^{(\alpha,\beta)}(x)$ and $P_1^{(\alpha,\beta)}(x)$ are needed to start recurrence.

$$P_0^{(\alpha,\beta)}(x) = \sqrt{2^{-\alpha-\beta-1} \frac{\Gamma(\alpha+\beta+2)}{\Gamma(\alpha+1)\Gamma(\beta+1)}}$$
(3.52)

$$P_1^{(\alpha,\beta)}(x) = \frac{1}{2} P_0^{(\alpha,\beta)}(x) \sqrt{\frac{\alpha+\beta+3}{(\alpha+1)(\beta+1)}} ((\alpha+\beta+2)x + (\alpha-\beta))$$
(3.53)

where $\Gamma(x)$ is the classical Gamma function. Moreover, a special case of $\alpha = 0, \beta = 0$ gives the Legendre functions.



Figure 3.3: Legendre Polynomials of Various Orders

3.5 Modes and Nodes In 2-Dimension

The approximate solution of discontinuous Galerkin discretization is presented as series of polynomial expansion in previous sections, however formulations are straightforward for one dimensional problems. Since, solution of 2-dimensional Euler equations are main objective of present work; extension to 2-dimension is required. Moreover, as it is provided in section 3.4 that basis functions span [-1, 1], a mapping from an arbitrary elements to standard element is required.

Extra geometric variation of the domain increases the complexity for extension of polynomials to 2-dimensional case. As in the 1-dimensional case, polynomials are required to be orthonormal to ensure numerical stability. The hassle of the problem

has been overcome by several studies [1, 66]. The 2-dimensional standard triangle is defined as:

$$T^{2} = [(r,s)|r,s \ge -1; r+s \le 0]$$
(3.54)

For a Nth order orthonormal basis, formulation is given as:

$$(a,b) \in [-1,1]^2; \forall (i,j) \ge 0; i+j \le N : \psi_{i,j}(r,s) = \sqrt{2}P_i^{(0,0)}(a)P_j^{2i+1,0}(b)(1-b)^i$$
(3.55)

(a, b) is a new coordinate system and the relation between (r, s) is given as below:

$$a = 2\frac{1+r}{1-s} - 1, \qquad b = s$$
 (3.56)

The equation 3.55 provides exactly $\frac{1}{2}(N+1)(N+2)$ terms for Nth order polynomial basis.



Figure 3.4: Schematic of Mapping of an Arbitrary Cell for Basis Function Evaluation

The evaluation of basis function for standard triangle is defined however, solution domain is triangulated using arbitrary elements, E^k . Hence, a mapping between standard triangle and an arbitrary triangle is required. The schematic of the procedure is provided in Figure 3.4. Firstly, arbitrary triangle is transformed to standard triangle, then transformed to basis function space and basis function is evaluated, finally, transformed back to physical space.

The arbitrary triangle is defined by counter clockwise positioned three corner points $(\mathbf{p^1}, \mathbf{p^2}, \mathbf{p^3})$. The mapping between arbitrary and standard triangle is constructed using barycentric coordinates, $(\lambda^1, \lambda^2, \lambda^3)$. The barycentric coordinates has the following property[67]:

$$0 \le \lambda^i \le 1, \quad \lambda^1 + \lambda^2 + \lambda^3 = 1 \tag{3.57}$$

The barycentric coordinates allows any point in the triangle to be defined by three corner points of the triangle:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \lambda^1 \begin{pmatrix} p_x^1 \\ p_y^1 \end{pmatrix} + \lambda^2 \begin{pmatrix} p_x^2 \\ p_y^2 \end{pmatrix} + \lambda^3 \begin{pmatrix} p_x^3 \\ p_y^3 \end{pmatrix}$$
(3.58)

Similarly in standard triangle:

$$\begin{pmatrix} r\\s \end{pmatrix} = \lambda^1 \begin{pmatrix} -1\\-1 \end{pmatrix} + \lambda^2 \begin{pmatrix} 1\\-1 \end{pmatrix} + \lambda^3 \begin{pmatrix} -1\\1 \end{pmatrix}$$
(3.59)

The equation 3.57,3.58 and 3.59 provides the direct mapping between standard triangle and arbitrary triangle:

$$\Phi(\mathbf{r}) = \begin{pmatrix} x\\ y \end{pmatrix} = -\frac{r+s}{2} \begin{pmatrix} p_x^1\\ p_y^1 \end{pmatrix} + \frac{r+1}{2} \begin{pmatrix} p_x^2\\ p_y^2 \end{pmatrix} + \frac{s+1}{2} \begin{pmatrix} p_x^3\\ p_y^3 \end{pmatrix}$$
(3.60)

The relation between standard triangle and arbitrary triangle is constructed hence it is now easy to obtain transformation metrics and Jacobian. It is important to note that the mapping provided in equation 3.60 is linear in **r** so transformation Jacobian is constant. The chain rule provides:

$$\frac{\partial \mathbf{x}}{\partial \mathbf{r}} \frac{\partial \mathbf{r}}{\partial \mathbf{x}} = \begin{bmatrix} x_r & x_s \\ y_r & y_s \end{bmatrix} \begin{bmatrix} r_x & r_y \\ s_x & s_y \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$
(3.61)

The metrics are obtained differentiating the equation 3.60 with respect to r:

$$\begin{pmatrix} x_r \\ y_r \end{pmatrix} = -\frac{1}{2} \begin{pmatrix} p_x^1 \\ p_y^1 \end{pmatrix} + \frac{1}{2} \begin{pmatrix} p_x^2 \\ p_y^2 \end{pmatrix}, \qquad \begin{pmatrix} x_s \\ y_s \end{pmatrix} = -\frac{1}{2} \begin{pmatrix} p_x^1 \\ p_y^1 \end{pmatrix} + \frac{1}{2} \begin{pmatrix} p_x^3 \\ p_y^3 \end{pmatrix} (3.62)$$

Moreover, due to 3.61:

$$\begin{pmatrix} r_x \\ r_y \end{pmatrix} = \frac{1}{J} \begin{pmatrix} y_s \\ -x_s \end{pmatrix}, \qquad \begin{pmatrix} s_x \\ s_y \end{pmatrix} = \frac{1}{J} \begin{pmatrix} -y_r \\ x_r \end{pmatrix}$$
(3.63)

where transformation Jacobian, J, is calculated as follows:

$$J = x_r y_s - x_s y_r \tag{3.64}$$

The evaluation and identification of an orthonormal basis function set on an arbitrary triangle is now complete. Behaviour of the basis function set is provided in Figure 3.5 for standard triangle up to 4th order.



Figure 3.5: Complete Set of Basis Functions for 4th Order Approximation

The another important subject for developing 2-dimensional scheme is the identification of families of nodal points that would provide healthy interpolation. In section 3.3, it is shown that Legendre-Gauss-Lobatta points are one of the suitable families. However, defining nodal points for 2-dimensional elements is not straightforward. There are several approaches to obtain a suitable nodal set families such as Fekete, LGL, Warp&Blend etc. are studied comparatively in [68] for a well-behaved interpolation. In the present work, nodal points on a triangular element are studied through Warp&Blend approach as presented in [69]. LGL nodes are mapped to edges using warp functions and interior nodes are constructed using blend functions.

The initial attempt to obtain LGL points on triangle would be tensor product of 1dimensional points, however, resulting family would have $(N + 1)^2$ points accumulated at one corner point and poor interpolation quality. Hence, a smarter approach is required. Building LGL nodes on triangle starts from the one dimensional equidistant, r^e , and LGL nodes, r^{LGL} . A connection between equidistant and LGL nodes can be constructed as follows:

$$w(r) = \sum_{i=1}^{N_p} (r_i^{LGL} - r_i^e) l_i^e(r)$$
(3.65)

 $l_i^e(r)$ are interpolating Lagrangian polynomials defined on equidistant nodes, r^e . Hence, w(r) becomes an approximation of a mapping function between equidistant and LGL nodes. In other words, it converts the equidistant nodes to healthy LGL nodes. For a triangle, one dimensional mapping approximation w(r) is used to blend the edge into the triangle. The application of w on triangle uses the barycentric coordinates. The equidistant nodes of equilateral triangle on barycentric coordinates are defined as:

$$(i,j) \ge 0, i+j \le N : (\lambda^1, \lambda^3) = (\frac{i}{N}, \frac{i}{N}), \quad \lambda^2 = 1 - \lambda^1 - \lambda^3$$
 (3.66)

The equilateral triangle is defined as:

$$T_E^2 = [(x,y)|x \ge -1; x - \frac{\sqrt{3}}{2}y \ge -1; x + \frac{\sqrt{3}}{2}y \le 1]$$
(3.67)

The equidistant nodes on the barycentric coordinates are blended along normal to the edge. Definition of warping function for first edge is given below:

$$\mathbf{w}^{1}(\lambda^{1}, \lambda^{2}, \lambda^{3}) = w(\lambda^{3} - \lambda^{2}) \begin{pmatrix} 1\\ 0 \end{pmatrix}$$
(3.68)

and corresponding blending function:

$$b^{1}(\lambda^{1},\lambda^{2},\lambda^{3}) = \frac{2\lambda^{3}}{2\lambda^{3}+\lambda^{1}}\frac{2\lambda^{2}}{2\lambda^{2}+\lambda^{1}}$$
(3.69)

In the equation 3.69, the terms in the denominators have the singularities in corner points of the corresponding edge, $\lambda^3 = \lambda^1 = 0$ and $\lambda^2 = \lambda^1 = 0$. Hence warp function, w(r) is revised to overcome singularity problem with following form:

$$\tilde{w}(r) = \frac{w(r)}{1 - r^2}$$
 (3.70)

Equations 3.68 and 3.69 have the new forms:

$$\mathbf{w}^{1}(\lambda^{1}, \lambda^{2}, \lambda^{3}) = \tilde{w}(\lambda^{3} - \lambda^{2}) \begin{pmatrix} 1\\ 0 \end{pmatrix}$$
(3.71)

and

$$b^1(\lambda^1, \lambda^2, \lambda^3) = 4\lambda^2 \lambda^3 \tag{3.72}$$

The warp and blending functions for the remaining edges are given as:

$$\mathbf{w}^{2}(\lambda^{1},\lambda^{2},\lambda^{3}) = \tilde{w}(\lambda^{1}-\lambda^{3})\frac{1}{2}\begin{pmatrix}-1\\\sqrt{3}\end{pmatrix}$$
(3.73)

$$b^{2}(\lambda^{1}, \lambda^{2}, \lambda^{3}) = 4\lambda^{1}\lambda^{3}$$
(3.74)

$$\mathbf{w}^{\mathbf{3}}(\lambda^{1},\lambda^{2},\lambda^{3}) = \tilde{w}(\lambda^{2}-\lambda^{1})\frac{1}{2}\begin{pmatrix}-1\\-\sqrt{3}\end{pmatrix}$$
(3.75)

$$b^{3}(\lambda^{1},\lambda^{2},\lambda^{3}) = 4\lambda^{1}\lambda^{2}$$
(3.76)

The w in two dimensions is written as follows:

$$\mathbf{w}(\lambda^1, \lambda^2, \lambda^3) = b^1 \mathbf{w^1} + b^2 \mathbf{w^2} + b^3 \mathbf{w^3}$$
(3.77)

A general form for the equation 3.77 is proposed

$$\mathbf{w}(\lambda^{1}, \lambda^{2}, \lambda^{3}) = (1 + (\alpha \lambda^{1})^{2})b^{1}\mathbf{w}^{1} + (1 + (\alpha \lambda^{2})^{2})b^{2}\mathbf{w}^{2} + (1 + (\alpha \lambda^{3})^{2})b^{3}\mathbf{w}^{3}$$
(3.78)

In section 3.3 Lebesque constant is introduced to measure the quality of interpolation. Similarly in two dimension, distribution of the nodes can be optimized using α in equation 3.78 minimizing Lebesque constant. The Lebesque constant for α -optimized set is tabulated in appendix.

The nodes are obtained in the equilateral triangle however, basis functions are defined on standard triangle. Hence, nodes are needed to be transformed to standard triangles. Combining the equations 3.57, 3.58 and 3.59 for equilateral and standard triangle mapping between the two is given as equation 3.79. The construction of the nodal set in a triangle is now complete. The distribution of the α -optimized nodes in the equilateral and standard triangle for various orders is presented in Figure 3.6 comparing with equidistant nodes.

$$\begin{pmatrix} r\\s \end{pmatrix} = \begin{bmatrix} 1 & \frac{-\sqrt{3}}{3}\\ 0 & \frac{2\sqrt{3}}{3} \end{bmatrix} \begin{pmatrix} x\\y \end{pmatrix} - \frac{1}{3} \begin{pmatrix} 1\\1 \end{pmatrix}$$
(3.79)



Figure 3.6: α -optimized and Equidistant Nodes for Various Orders in Equilateral and Standard Triangle

In the present section, an orthonormal basis function set is identified for two dimensional applications. Nodal sets on the triangle is constructed while acquiring a well behaved interpolation properties. Affine mapping between arbitrary, equilateral and standard triangles is derived to have the ability of working on unstructured meshes. All the efforts are spent to construct a well conditioned Vandermonde matrix to have a stable connection between modal and nodal representations in two dimensions. Similar to one dimensional case, local approximations in two dimension would be in the following form:

$$u_{h}^{k}(\mathbf{r},t) = \sum_{n=1}^{N_{p}} \tilde{u}_{n}^{k}(t)\psi_{n}(\mathbf{r}) = \sum_{i=1}^{N_{p}} u_{h}^{k}(\mathbf{r}_{i}^{k},t)l_{i}^{k}(\mathbf{r})$$
(3.80)

where \mathbf{r}_i and $\psi_n(\mathbf{r})$ are nodal set and orthonormal basis function set in two dimensions. The number of terms, N_p , in the local expression is related to order of the basis functions, N. In two dimensions, N_p is given below as shown before:

$$N_p = \frac{(N+1)(N+2)}{2} \tag{3.81}$$

Using the well conditioned and stable Vandermonde matrix, \mathcal{V} , direct transformation between modal and nodal representation is established. Rewriting the equation 3.44 for two dimensional case:

$$\mathcal{V}\tilde{\mathbf{u}}^{k} = \mathbf{u}_{h}^{k}, \, \mathcal{V}^{T}\mathbf{l}(\mathbf{r}) = \psi(\mathbf{r}), \, \, \mathcal{V}_{ij} = \psi_{j}(r_{i})$$
(3.82)

This relation offers a way to evaluate Lagrange polynomials in two dimensions which do not have a known explicit formulation.

3.6 Discontinuous Galerkin Finite Element Method Discretization of Euler Equations

Discontinuous Galerkin discretization of Euler equations are given in this section. DG method is applied on only in space discretization while temporal discretization is obtained through classical explicit methods. Rewriting the Euler equations given in section 3.1 in vector form for two dimensional flows provides:

$$\frac{\partial \mathbf{q}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} = 0 \tag{3.83}$$

where q is the state vector, F and G are the non-linear flux vectors and given as:

$$\mathbf{q} = \begin{cases} \rho \\ \rho u \\ \rho v \\ \rho E \end{cases}, \qquad \mathbf{F} = \begin{cases} \rho u \\ \rho u^2 + P \\ \rho v u \\ \rho E u + P u \end{cases}, \qquad \mathbf{G} = \begin{cases} \rho v \\ \rho u v \\ \rho u v \\ \rho v^2 + P \\ \rho E v + P v \end{cases}$$
(3.84)

In discretization process firstly, solution domain is divided into K number of nonoverlapping triangular elements, E.

$$\Omega = \bigcup_{k=1}^{K} E^k \tag{3.85}$$

In each element, global solution q is approximated as follows:

$$\mathbf{q} = \mathbf{q}(\vec{\mathbf{x}}, t) \simeq \mathbf{q}_h(\vec{\mathbf{x}}, t) = \bigoplus_{k=1}^K \mathbf{q}_h^k(\vec{\mathbf{x}}, t)$$
(3.86)

where $\mathbf{q}_h(\vec{\mathbf{x}}, t)$ is the approximation of global solution and $\mathbf{q}_h^k(\vec{\mathbf{x}}, t)$ is the approximate local solution on element E^k . The local solution is expressed as Nth order polynomial as discussed in the previous section 3.2:

$$\vec{\mathbf{x}} \in E^{k} : \mathbf{q}_{h}^{k}(\vec{\mathbf{x}}, t) = \sum_{i=1}^{N_{p}} \mathbf{q}_{i}^{k}(t)\psi_{i}^{k}(\vec{\mathbf{x}}) = \sum_{i=1}^{N_{p}} \mathbf{q}_{h}^{k}(\vec{x_{i}}^{k}, t)l_{i}^{k}(\vec{x})$$
(3.87)

The number of terms, N_p , in the local expression is related to order of the basis functions, N. As shown in section 3.5, for two dimensional problems, N_p is given as follows:

$$N_p = \frac{(N+1)(N+2)}{2} \tag{3.88}$$

Following the definition of local expression for state vector, q_h , discontinuous Galerkin statement has to be satisfied for all test functions $\phi_h \in V_h$ as below:

$$\int_{E^k} \left(\frac{\partial \mathbf{q_h}}{\partial t} + \frac{\partial \mathbf{F_h}}{\partial x} + \frac{\partial \mathbf{G_h}}{\partial y} \right) \phi_h d\Omega = 0$$
(3.89)

As in the one dimensional case, this is the weighted residual form. Performing integration by parts on the advection terms, weak form of the Euler equations, which is the basic form of discontinuous Galerkin discretization, is obtained:

$$\underbrace{\int_{E^{k}} \frac{\partial \mathbf{q_{h}}}{\partial t} \phi_{h} d\Omega}_{Term \ I} - \underbrace{\int_{E^{k}} \left(\mathbf{F_{h}} \frac{\partial \phi_{h}}{\partial x} + \mathbf{G_{h}} \frac{\partial \phi_{h}}{\partial y} \right) d\Omega}_{Term \ II} + \underbrace{\oint_{\partial E^{k} / \partial \Omega} \mathcal{H}_{h}(\mathbf{q_{h}^{-}, q_{h}^{+}, \vec{n}}) \phi_{h} d\Gamma}_{Term \ III} + \underbrace{\oint_{\partial E^{k} \cap \partial \Omega} \mathcal{H}_{h}(\mathbf{q_{h}^{-}, q_{h}^{b}, \vec{n}}) \phi_{h} d\Gamma}_{Term \ III} = 0$$

$$(3.90)$$

where $\mathcal{H}_h(\mathbf{q}_h^-, \mathbf{q}_h^+, \vec{\mathbf{n}})$ and $\mathcal{H}_h(\mathbf{q}_h^-, \mathbf{q}_h^\mathbf{b}, \vec{\mathbf{n}})$ are the numerical flux functions for interior and boundary element faces, respectively as depicted in Figure 3.7. The notation $(\cdot)^$ represents the interior element, likely, $(\cdot)^+$ stands for exterior elements. The boundary conditions are implemented through boundary state $\mathbf{q}_h^\mathbf{b}$.



Figure 3.7: Interior and Exterior Element Notation

At this point, decision of nodal and modal approach has to be made. In the present work, nodal DG approach is applied. The equation 3.90 is divided into three terms and nodal DG approach for each term is provided individually.

3.6.1 Term I and Mass Matrix

The Term I can be investigated similarly in section 3.2.

$$\mathcal{M}^{k} \frac{d}{dt} \boldsymbol{q}_{h}^{k} = \int_{E^{k}} \frac{\partial \mathbf{q}_{h}}{\partial t} \phi_{h} d\Omega$$
(3.91)

where q_h^k is the nodal values of conserved variables and:

$$\mathcal{M}_{ij}^{k} = \int_{E^{k}} l_{i}^{k}(\mathbf{x}) l_{j}^{k}(\mathbf{x}) d\mathbf{x} = J^{k} \int_{I} l_{i}^{k}(\mathbf{r}) l_{j}^{k}(\mathbf{r}) d\mathbf{r}$$
(3.92)

The J^k is the transformation Jacobian between arbitrary and standard triangle as presented in section 3.5. The J^k is constant and positive if E^k is rectilinear triangle. Using the property of equation 3.82, Lanrange polynomials can be written:

$$l_i^k(\mathbf{r}) = \sum_{n=1}^{N_p} (\mathcal{V}^T)_{in}^{-1} \psi_n(r_i)$$
(3.93)

Rewriting equation 3.92 using equation 3.93:

$$\mathcal{M}_{ij}^{k} = J^{k} \int_{I} \sum_{n=1}^{N_{p}} (\mathcal{V}^{T})_{in}^{-1} \psi_{n}(r_{i}) \sum_{m=1}^{N_{p}} (\mathcal{V}^{T})_{jm}^{-1} \psi_{m}(r_{j}) d\mathbf{r}$$

$$= J^{k} \sum_{n=1}^{N_{p}} \sum_{m=1}^{N_{p}} (\mathcal{V}^{T})_{in}^{-1} (\mathcal{V}^{T})_{jm}^{-1} \int_{I} \psi_{n}(r_{i}) \psi_{m}(r_{j}) d\mathbf{r}$$
(3.94)

Since basis functions are orthonormal Legendre polynomials,

$$\int_{I} \psi_n(r_i) \psi_m(r_j) d\mathbf{r} = \begin{cases} 0, & n \neq m \\ 1, & n = m \end{cases}$$

equation 3.21 takes the final form:

$$\mathcal{M}^{k} = J^{k} \sum_{n=1}^{N_{p}} (\mathcal{V}^{T})_{in}^{-1} (\mathcal{V}^{T})_{jn}^{-1} = J^{k} (\mathcal{V}^{T})^{-1} (\mathcal{V})^{-1} = J^{k} (\mathcal{V}\mathcal{V}^{T})^{-1}$$
(3.95)

3.6.2 Term II and Stiffness Matrix

The derivation of Term II is not straightforward as Term I in two dimensions. The similar approach defined in 3.2 for one dimensional case is followed. In order to obtain stiffness matrix, derivative of the test functions in physical plane has to be handled.

$$\int_{E^k} \left(\mathbf{F_h} \frac{\partial \phi_{\mathbf{h}}}{\partial x} + \mathbf{G_h} \frac{\partial \phi_{\mathbf{h}}}{\partial y} \right) d\Omega = \int_{E^k} \left(\mathbf{F_h^k} l_i^k(\mathbf{x}) \frac{\partial l_j^k(\mathbf{x})}{\partial x} + \mathbf{G_h^k} l_i^k(\mathbf{x}) \frac{\partial l_j^k(\mathbf{x})}{\partial y} \right) d\mathbf{x} \quad (3.96)$$

Chain rule is used to get derivatives:

$$\frac{\partial l_j^k(\mathbf{x})}{\partial x} = \frac{\partial r}{\partial x} \mathcal{D}_r + \frac{\partial s}{\partial x} \mathcal{D}_s, \ \frac{\partial l_j^k(\mathbf{x})}{\partial y} = \frac{\partial r}{\partial y} \mathcal{D}_r + \frac{\partial s}{\partial y} \mathcal{D}_s, \tag{3.97}$$

The transformation metrics are defined in equation 3.63. The differentiation matrices, D_r and D_s , are defined as:

$$\mathcal{D}_{r,ij} = \frac{\partial l_j^k(\mathbf{r})}{\partial r} \bigg|_{\mathbf{r}_i}, \ \mathcal{D}_{s,ij} = \frac{\partial l_j^k(\mathbf{r})}{\partial s} \bigg|_{\mathbf{r}_i}$$
(3.98)

The differentiation matrices transforms the nodal values to the derivatives of them at nodes. They are obtained using property of equation 3.82.

$$\mathcal{V}^T \mathcal{D}_r = \frac{\partial \psi_j}{\partial r} \bigg|_{\mathbf{r}_i}, \ \mathcal{V}^T \mathcal{D}_s = \frac{\partial \psi_j}{\partial s} \bigg|_{\mathbf{r}_i}$$
(3.99)

The orthonormal basis is defined in section 3.5 as:

$$n \in [1, ..., N_p]; \forall (i, j) \ge 0; i + j \le n : \psi_n(\mathbf{r}) = \sqrt{2} P_i^{(0,0)}(a) P_j^{2i+1,0}(b) (1-b)^i$$
(3.100)

The relation between coordinates (a, b) and (r, s) is provided in equation 3.56. Chain rule enables the evaluation of derivatives:

$$\frac{\partial \psi_j}{\partial r} = \frac{\partial a}{\partial r} \frac{\partial \psi_j}{\partial a}, \quad \frac{\partial \psi_j}{\partial s} = \frac{\partial a}{\partial s} \frac{\partial \psi_j}{\partial a} + \frac{\partial \psi_j}{\partial b}$$
(3.101)

The differentiation matrix now can be calculated at nodal points. The equation 3.96 is rewritten in stiffness matrix form as in one dimensional case.

$$\tilde{\mathbf{F}}_{\mathbf{h}}^{\mathbf{k}} \mathcal{S}_{x} + \tilde{\mathbf{G}}_{\mathbf{h}}^{\mathbf{k}} \mathcal{S}_{y} = \tilde{\mathbf{F}}_{\mathbf{h}}^{\mathbf{k}} (\mathcal{S}_{r} \frac{\partial r}{\partial x} + \mathcal{S}_{s} \frac{\partial s}{\partial x}) + \tilde{\mathbf{G}}_{\mathbf{h}}^{\mathbf{k}} (\mathcal{S}_{r} \frac{\partial r}{\partial y} + \mathcal{S}_{s} \frac{\partial s}{\partial y})$$
(3.102)

where $\tilde{\mathbf{F}}_{\mathbf{h}}^{\mathbf{k}}$ and $\tilde{\mathbf{G}}_{\mathbf{h}}^{\mathbf{k}}$ are nodal flux vectors and calculated directly from nodal solution. In order to calculate Term *II*, evaluation of stiffness matrices S_r and S_s is mandatory. Consider the product \mathcal{MD}_r :

$$(\mathcal{MD}_r)_{(i,j)} = \sum_{n=1}^{N_p} \mathcal{M}_{in} \mathcal{D}_{r,nj} = \sum_{n=1}^{N_p} \int_{-1}^{1} l_i(r) l_n(r) \frac{\partial l_j}{\partial r} \Big|_{r_n} dr$$
(3.103)

Changing the order between summation and integral and using orthonormal property of Lagrangian interpolation, the result is exactly stiffness matrix.

$$\int_{-1}^{1} l_i(r) \sum_{n=1}^{N_p} l_n(r) \frac{\partial l_j}{\partial r} \Big|_{r_n} dr = \int_{-1}^{1} l_i(r) \frac{\partial l_j}{\partial r} dr = \mathcal{S}_{r,ij}$$
(3.104)

Similarly for \mathcal{MD}_s :

$$(\mathcal{MD}_s)_{(i,j)} = \mathcal{S}_{s,ij} \tag{3.105}$$

3.6.3 Term III and Surface Mass Matrix

The discretization of Euler equations is completed with the evaluation of surface integral in following form:

$$\oint_{\partial E^k} \mathcal{H}_h(\mathbf{q}_h^-, \mathbf{q}_h^+, \vec{\mathbf{n}}) \phi_h d\Gamma = \oint_{\partial E^k} \mathcal{H}_h \vec{\mathbf{n}} l_i^k(\mathbf{x}) d\Gamma$$
(3.106)

where \mathcal{H}_h is defined as polynomial trace function constructed using numerical flux. The evaluation of the surface integral is straightforward in one dimensional case. For two dimensional case, the operation is more complex. The integral is divided into three separate edge components.

$$\forall E^k, e \ge 1; e \le 3: \quad \oint_{\partial E^k_e} \mathcal{H}_h, \vec{\mathbf{n}} l^k_i(\mathbf{x}) d\Gamma = \sum_{j=1}^{N+1} \mathcal{H}_j \int_{\partial E^k_e} \vec{\mathbf{n}} l^k_j(\mathbf{x}) l^k_j(\mathbf{x}) d\mathbf{x} \quad (3.107)$$

.....

where $\mathbf{x} = (x, y)$ is assumed to be located along the edge where N + 1 nodal points is present. Hence, following mass matrix is required to be computed:

$$\mathcal{M}_{ij}^{k,e} = \int_{\partial E_e^k} l_j^k(\mathbf{x}) l_j^k(\mathbf{x}) d\mathbf{x}$$
(3.108)

The important point in the equation 3.108 is that $l_i^k(\mathbf{x})$ has non-zero value only if \mathbf{x}_i lies on the edge. Hence, $\mathcal{M}_{ij}^{k,e}$ has non-zero values in the rows where \mathbf{x}_i lies on the edge. Therefore a one dimensional Vandermonde matrix, \mathcal{V}^{1D} , along the edge can be used to calculate edge mass matrix:

$$\mathcal{M}_{ii}^{k,e} = J^e (\mathcal{V}^{1D} (\mathcal{V}^{1D})^T)^{-1}$$
(3.109)

where J^e is the transformation Jacobian of the edge e which is the ratio of the edge length in physical coordinates to the one in standard triangle coordinates.

The calculation of surface integral using only nodes along the edges is distinct advantage of the nodal approach. On the other hand, modal approach would require the all information to calculate the solution pointwise.

3.7 Time Discretization

The spatial discretization of the Euler equations is completed in section 3.6. The resulting equation is a first order ordinary differential equation in time as given in equation 3.110. In order to advance in time, low-storage explicit fourth order Runge-Kutta method is selected [2]. The classical fourth order Runge-Kutta method requires memory enough to store 4 stages while low-storage method requires only 2 stages to be stored with expense of additional stage evaluation and has improved numerical stability.

$$\frac{d\mathbf{q}_{\mathbf{h}}}{dt} = \mathcal{R}_{(\mathbf{q}_{\mathbf{h}}, t)}$$
(3.110)

where q_h is the unknown vector. The low storage explicit Runge-Kutta discretization is given as below:

$$\mathbf{dq}_{i} = a_{i}\mathbf{dq}_{i-1} + \delta t \mathcal{R}_{(\mathbf{q}_{i}, t^{n} + c_{i}\delta t), i \in [1, 5]}$$

$$\mathbf{q}_{i} = \mathbf{q}_{i-1} + b_{i}\mathbf{dq}_{i}, i \in [1, 5]$$

(3.111)

where dq is the temporary stage. Coefficients, a_i, b_i and c_i , are provided in 3.1.

Table 3.1: Low Storage Fourth Order Explicit Runge-Kutta Method Coefficients [2]

STORAGE LEVEL,i	a_i	b_i	c_i
1	0	$\frac{1432997174477}{9575080441755}$	0
2	$-\frac{567301805773}{1357537059087}$	$\frac{5161836677717}{13612068292357}$	$\frac{1432997174477}{9575080441755}$
3	$-\tfrac{2404267990393}{2016746695238}$	$\frac{1720146321549}{2090206949498}$	$\frac{2526269341429}{6820363962896}$
4	$-\frac{3550918686646}{2091501179385}$	$\frac{3134564353537}{4481467310338}$	$\frac{2006345519317}{3224310063776}$
5	$-\tfrac{1275806237668}{842570457699}$	$\frac{2277821191437}{14882151754819}$	$\frac{2802321613138}{2924317926251}$

Another important subject of time discretization is choice of time step size. The smaller time step increases the accuracy however also computation time. On the other hand, increasing time step beyond the stability boundaries fails the computation. Hence, establishing a balance between stability and accuracy is vital. Hesthaven[1] studied time step choice in detail and provided a stability condition for triangular domains.

$$\Delta t \le C \left(\frac{2}{3} \min(\Delta r_i)\right) \min_{\Omega} \left(\frac{r_D}{|\mathbf{v}|}\right) \tag{3.112}$$

where Δr_i spacing between grid points. r_D is the characteristic length of triangle and computed as:

$$r_D = \frac{A}{s} \tag{3.113}$$

A is the area and s is half of perimeter of triangle.

3.8 Numerical Flux

Numerical flux is at the heart of the discontinuous Galerkin method establishing link between cells and boundary conditions. There are various well-developed numerical flux schemes for finite volume methods. In the present work, local Lax-Friedrichs Flux, Roe's approximate Riemann solver and Harten-Lax-van Leer approximate Riemann solver are utilized. In nodal discontinuous Galerkin method, numerical flux is calculated between each face nodes as presented in Figure 3.8. The coordinate system used in the numerical flux calculations are face normal-tangent coordinate system and all the flow variables are written respect to normal-tangent coordinate system through the section.



Figure 3.8: Numerical Flux Calculation

3.8.1 Local Lax-Friedrichs Flux

The Lax-Friedrichs flux is a suitable choice for subsonic and low supersonic flows. However, for high supersonic flows, it is not a recommended method due to the fact that dissipative behaviour of the scheme causes shocks to diffuse which results in loss of accuracy. The flux is defined as follows [1]

$$\mathcal{H}_h(\mathbf{q}_h^-, \mathbf{q}_h^+, \vec{\mathbf{n}}) = \frac{1}{2}(\mathbf{F}_h^- + \mathbf{F}_h^+) + \frac{\lambda}{2}(\mathbf{q}_h^- - \mathbf{q}_h^+)$$
(3.114)

where λ is maximum local acoustic wave speed. The equation for λ is provided below:

$$\lambda = \max_{s \in [q_h^-, q_h^+]} \left(|\mathbf{u}(s)| + \sqrt{\left| \frac{\gamma p(s)}{\rho(s)} \right|} \right)$$
(3.115)

3.8.2 Roe's Approximate Riemann Solver

Euler equations are non-linear and solutions may have discontinuities at supersonic or high subsonic flows. Hence, the calculation of the waves, used to compute numerical flux, are not straightforward as in the linear problems. The analytic solution of waves in non-linear conservation laws is known as Riemann problem. The numerical solution of Riemann problem increases the computational cost significantly. Roe[7] proposed a approximate solver assuming the problem is dominated by one strong wave. The numerical flux is given as follows:

$$\mathcal{H}_h(\mathbf{q}_h^-, \mathbf{q}_h^+, \vec{\mathbf{n}}) = \frac{1}{2} (\mathbf{F}_h^- + \mathbf{F}_h^+) - \frac{1}{2} \sum_{i=1}^4 \alpha_i |\lambda_i| \mathbf{K}^i$$
(3.116)

where α is the wave strength. λ and **K** are eigenvalues and eigenvectors of flux Jacobian matrix. The eigenvalues and eigenvectors are calculated from Roe averaged primitive variables given in equation 3.117.

$$\widetilde{u} = \frac{\sqrt{\rho^{+}u^{+} + \sqrt{\rho^{-}u^{-}}}}{\sqrt{\rho^{+} + \sqrt{\rho^{-}}}} \\
\widetilde{v} = \frac{\sqrt{\rho^{+}v^{+} + \sqrt{\rho^{-}}v^{-}}}{\sqrt{\rho^{+} + \sqrt{\rho^{-}}}} \\
\widetilde{H} = \frac{\sqrt{\rho^{+}H^{+} + \sqrt{\rho^{-}}H^{-}}}{\sqrt{\rho^{+} + \sqrt{\rho^{-}}}} \\
\widetilde{c} = \sqrt{(\gamma - 1)(\widetilde{H} - \frac{1}{2}(\widetilde{u}^{2} + \widetilde{v}^{2}))}$$
(3.117)

The eigenvalues and eigenvectors of flux Jacobian matrix are given as belows:

$$\lambda_1 = \tilde{u} - \tilde{c}, \quad \lambda_2 = \lambda_3 = \tilde{u}, \quad \lambda_4 = \tilde{u} + \tilde{c} \tag{3.118}$$

$$K_{1} = \begin{cases} 1\\ \tilde{u} - \tilde{c}\\ \tilde{v}\\ \tilde{H} - \tilde{u}\tilde{c} \end{cases}, \quad K_{2} = \begin{cases} 1\\ \tilde{u}\\ \tilde{v}\\ \frac{1}{2}(\tilde{u}^{2} + \tilde{v}^{2}) \end{cases},$$

$$K_{3} = \begin{cases} 0\\ 0\\ 1\\ \tilde{v} \end{cases}, \quad K_{4} = \begin{cases} 1\\ \tilde{u} + \tilde{c}\\ \tilde{v}\\ \tilde{H} + \tilde{u}\tilde{c} \end{cases}$$

$$(3.119)$$

Finally, α is provided in following form:

$$\alpha_{1} = \frac{1}{2\tilde{c}} (\Delta q_{1}(\tilde{u} + \tilde{c}) - \Delta q_{2} - \tilde{c}\alpha_{2})$$

$$\alpha_{2} = \frac{\gamma - 1}{\tilde{c}^{2}} (\Delta q_{1}(\tilde{H} - \tilde{u}^{2}) + \Delta q_{2}\tilde{u} - \Delta p + (\Delta q_{3} - \tilde{v}\Delta q_{1})\tilde{v})$$

$$\alpha_{3} = \Delta q_{3} - \Delta q_{1}\tilde{v}$$

$$\alpha_{4} = \Delta q_{1} - (\alpha_{1} + \alpha_{2})$$
(3.120)

where operator Δ is the jump between two states.

$$\Delta q_i = q_i^+ - q_i^- \tag{3.121}$$

3.8.3 Harten-Lax-van Leer Approximate Riemann Solver

For many problems, Roe flux is an appropriate choice however, for some cases, it produces spurious results due to violation of its basic assumption being domination of one strong wave. Harten-Lax-van Leer (HLL) approximate Riemann solver overcomes the issue by accommodating two waves and assuming three Riemann states. The HLL flux is given by following formula [70]

$$\mathcal{H}_{h}(\mathbf{q_{h}^{-}, q_{h}^{+}, \vec{n}}) = \begin{cases} \mathbf{F}^{-}, & 0 \le S^{-} \\ \frac{S^{+}\mathbf{F}^{-} - S^{-}\mathbf{F}^{+} + S^{-}S^{+}(\mathbf{q}^{+} - \mathbf{q}^{-})}{S^{+} - S^{-}}, & S^{-} \le 0 \le S^{+} \\ \mathbf{F}^{+}, & 0 \ge S^{+} \end{cases}$$
(3.122)

where S^- and S^+ are the fastest wave speeds with relation $S^+ > S^-$. The estimation of these two speeds are given in equation 3.123.

$$s^{-} = min(\lambda^{-}, \tilde{\lambda}), \quad s^{+} = max(\lambda^{+}, \tilde{\lambda})$$
 (3.123)

where $\tilde{\lambda}$ are eigenvalues of the Roe averaged state.

3.9 Boundary Conditions

3.9.1 Dirichlet Boundary Condition

Dirichlet boundary condition is used in the domain boundaries where values of flow variables are available. Hence, boundary states are directly provided.

$$\mathbf{q}_{h}^{b} = f(x,t) \tag{3.124}$$

3.9.2 Inflow Boundary Condition

The inflow boundary condition is defined using total temperature, total pressure and flow direction for subsonic flow whereas for supersonic flow, all flow parameters are required to be defined because all the information is transmitted outside the domain. The inflow boundary is not the best choice for external flows, rather it is recommended for channel flows.

3.9.2.1 Subsonic Inflow

The boundary state for subsonic inflow is constructed from outward propagating Riemann invariant and total enthalpy. The basic assumption is that flow is adiabatic and isentropic. The total enthalpy in the inner state is written as:

$$H_t = \frac{p^-}{\rho^-} \frac{\gamma}{\gamma - 1} + \frac{1}{2} ((u^-)^2 + (v^-)^2)$$
(3.125)

Riemann invariant propagating outside is given in equation 3.126.

$$R^{-} = \vec{\mathbf{u}} \cdot \vec{n} - \frac{2c^{-}}{\gamma - 1}$$
(3.126)

The boundary state is extrapolated from inner state, hence equation 3.125 and 3.126 are rewritten for boundary state:

$$H_t = \frac{\gamma(c^b)^2}{\gamma - 1} + \frac{1}{2} (\vec{\mathbf{u}}^b \cdot \vec{n})^2$$
(3.127)

$$R^{-} = \vec{\mathbf{u}}^{b} \cdot \vec{n} - \frac{2c^{b}}{\gamma - 1}$$
(3.128)

Combining equation 3.127 and 3.128, and solving for speed of sound at boundary state provides following equation:

$$(1 + \frac{2}{\gamma - 1})(c^b)^2 + 2R^-c^b + (\gamma - 1)(\frac{(R^-)^2}{2} - H_t) = 0$$
(3.129)

The equation 3.129 is a quadratic equation which has two roots. The larger of the roots provides the physically consistent result. The boundary velocity is calculated using Riemann invariant and speed of sound.

$$\vec{\mathbf{u}}^b \cdot \vec{n} = R^- + \frac{2c^b}{\gamma - 1}$$
 (3.130)

The boundary velocity and speed of sound enables the evaluation of Mach number of boundary state.

$$M^b = \frac{\vec{\mathbf{u}}^b \cdot \vec{n}}{c^b} \tag{3.131}$$

Using the isentropic relations, static pressure and temperature are calculated.

$$p^{b} = p_{t} \left(1 + \frac{\gamma - 1}{2} M_{b}^{2}\right)^{\frac{\gamma}{\gamma - 1}} \quad T^{b} = T_{t} \left(\frac{p^{b}}{p_{t}}\right)^{\frac{\gamma - 1}{\gamma}}$$
(3.132)

The boundary density is calculated using equation of state.

$$\rho^b = \frac{p^b}{T^b R} \tag{3.133}$$

The boundary state is constructed using the flow variables obtained at boundary. The velocity components are determined from the boundary velocity and specified flow direction.

$$\mathbf{q}_{h}^{b} = \begin{cases} \rho^{b} \\ \rho^{b} u^{b} \\ \rho^{b} v^{b} \\ \frac{p^{b}}{\gamma - 1} + \frac{1}{2} \rho^{b} ((u^{b})^{2} + (v^{b})^{2}) \end{cases}$$
(3.134)

3.9.2.2 Supersonic Inflow

Pressure, density and velocity is defined at supersonic inflow boundary. The boundary state is directly constructed from the boundary conditions.

$$\mathbf{q}_{h}^{b} = \begin{cases} \rho^{b} \\ \rho^{b} u^{b} \\ \rho^{b} v^{b} \\ \frac{p^{b}}{\gamma - 1} + \frac{1}{2} \rho^{b} ((u^{b})^{2} + (v^{b})^{2}) \end{cases}$$
(3.135)

3.9.3 Outflow Boundary Condition

The static pressure is set at the outflow boundary. Adiabatic and isentropic flow is assumed at the boundary. Extrapolated velocity and temperature is used with static pressure to compute density. In the case of supersonic flow at the boundary, all the flow variables are extrapolated. The boundary pressure is computed as:

$$p^{b} = \begin{cases} p^{b}, & M^{-} < 1\\ p^{-}, & M^{-} \ge 1 \end{cases}$$
(3.136)

Boundary density is calculate from boundary pressure and temperature:

$$\rho^b = \frac{\gamma p^b}{T^b}, \quad T^b = \frac{\gamma p^-}{\rho^-} \tag{3.137}$$

Then boundary state is given as:

$$\mathbf{q}_{h}^{b} = \begin{cases} \rho^{b} & \\ \rho^{b}u^{-} & \\ \rho^{b}v^{-} & \\ \frac{p^{b}}{\gamma-1} + \frac{1}{2}\rho^{b}((u^{-})^{2} + (v^{-})^{2}) \end{cases}$$
(3.138)

3.9.4 Wall Boundary Condition

Slip wall boundary requires that velocity normal to wall should diminish while velocity tangent to wall should be preserved.

$$\vec{\mathbf{u}}^b \cdot \vec{n} = 0, \quad \vec{\mathbf{u}}^b \cdot \vec{t} = \vec{\mathbf{u}}^- \cdot \vec{t} \tag{3.139}$$

Solving for boundary velocity provides the boundary state.

$$\mathbf{q}_{h}^{b} = \begin{cases} \rho^{-} & \\ \rho^{-} u^{b} & \\ \rho^{-} v^{b} & \\ \frac{p^{-}}{\gamma^{-1}} + \frac{1}{2} \rho^{-} ((u^{b})^{2} + (v^{b})^{2}) \end{cases}$$
(3.140)

3.9.4.1 Curved Walls

The discontinuous Galerkin method requires curved elements when a boundary is not straight. Bassi and Rebay [71] worked with two dimensional Euler equations to show the importance of curved elements for achieving high order accuracy. Hence, construction of curved elements are necessary in order to solve problems with curved walls. Transformation of a straight sided boundary cell into curved boundary cell is presented in Figure 3.9. The face nodes lying on the wall boundary are projected onto the curved boundary and deformations between straight and curved nodes are calculated. The work of Gordon and Hall [72] is used to blend deformation onto remaining nodes. Finally, face normals and transformation Jacobian are updated. The presented method is quick and easy to implement however it requires analytical representation of walls.



Figure 3.9: Visual Comparison of Straight Sided Cell and Curved Cell

3.9.5 Riemann Invariant Farfield Boundary Condition

Riemann invariant boundary condition utilizes incoming and outgoing characteristic waves at farfield boundary to determine local normal velocity and speed of sound. Pressure and density of boundary state is solved using entropy and speed of sound. The Mach number and flow direction is specified. Calculation of Riemann invariants to construct boundary state depends on the flow regime and direction at boundary as depicted in Figure 3.10.



Figure 3.10: Contribution of Riemann Invariants to Boundary State at Farfield Boundary for Subsonic Inflow, Supersonic Inflow, Subsonic Outflow and Supersonic Outflow

For subsonic inflow and outflow Riemann invariants are as follows:

$$R^{-} = \vec{\mathbf{u}}^{-} \cdot \vec{n} + \frac{2c^{-}}{\gamma - 1} \quad R^{+} = \vec{\mathbf{u}}^{\infty} \cdot \vec{n} - \frac{2c^{\infty}}{\gamma - 1}$$
(3.141)

For supersonic inflow, all information used to construct boundary state is taken from free stream:

$$R^{-} = \vec{\mathbf{u}}^{\infty} \cdot \vec{n} + \frac{2c^{\infty}}{\gamma - 1} \quad R^{+} = \vec{\mathbf{u}}^{\infty} \cdot \vec{n} - \frac{2c^{\infty}}{\gamma - 1}$$
(3.142)

For supersonic outflow, interior state is used to construct boundary state:

$$R^{-} = \vec{\mathbf{u}}^{-} \cdot \vec{n} + \frac{2c^{-}}{\gamma - 1} \quad R^{+} = \vec{\mathbf{u}}^{-} \cdot \vec{n} - \frac{2c^{-}}{\gamma - 1}$$
(3.143)

where

$$c^{-} = \sqrt{\frac{\gamma p^{-}}{\rho^{-}}} \quad c^{\infty} = \sqrt{\frac{\gamma p^{\infty}}{\rho^{\infty}}} \tag{3.144}$$

Velocity normal to boundary and speed of sound at boundary state is calculated from incoming and outgoing Riemann invariants.

$$\vec{\mathbf{u}}^b \cdot \vec{n} = \frac{1}{2}(R^- + R^+) \quad c^b = 4(\gamma - 1)(R^- - R^+)$$
 (3.145)

The sign of the normal velocity at boundary decides the direction of flow. If the flow is incoming to domain then tangential velocity and entropy is constructed from farfield conditions. Similarly, if the flow is outgoing, interior state is used.

$$\vec{\mathbf{u}}^{b} \cdot \vec{t} = \begin{cases} \vec{\mathbf{u}}^{\infty} \cdot \vec{t}, & \vec{\mathbf{u}}^{b} \cdot \vec{n} < 0\\ \vec{\mathbf{u}}^{-} \cdot \vec{t}, & \vec{\mathbf{u}}^{b} \cdot \vec{n} > 0 \end{cases}, s^{b} = \begin{cases} \frac{(c^{\infty})^{2}}{\gamma(\rho^{\infty})^{\gamma-1}}, & \vec{\mathbf{u}}^{b} \cdot \vec{n} < 0\\ \frac{(c^{-})^{2}}{\gamma(\rho^{-})^{\gamma-1}}, & \vec{\mathbf{u}}^{b} \cdot \vec{n} > 0 \end{cases}$$
(3.146)

Using entropy and speed of sound, density and pressure are computed as follows.

$$\rho^{b} = \left(\frac{(c^{b})^{2}}{\gamma s^{b}}\right)^{\frac{1}{\gamma-1}}, \quad p^{b} = \frac{\rho^{b}(c^{b})^{2}}{\gamma}$$
(3.147)

Rotating boundary velocity to Cartesian coordinate system provides the boundary state.

$$\mathbf{q}_{h}^{b} = \begin{cases} \rho^{b} \\ \rho^{b} u^{b} \\ \rho^{b} v^{b} \\ \frac{p^{b}}{\gamma - 1} + \frac{1}{2} \rho^{b} ((u^{b})^{2} + (v^{b})^{2}) \end{cases}$$
(3.148)

3.10 Stabilization

Euler equations obeying hyperbolic conservation laws contains discontinues named shock in solution field. The shocks leads to non-physical oscillations that may even lead to divergence of numerical method. However, basic form of numerical methods generally are not capable of handling the shocks inherently. The stabilization techniques are employed to achieve solution convergence. Among those methods, two approaches, filtering and slope limiting, are utilized in the present work.

3.10.1 Filter

Filtering approach is a kind of artificial viscosity application. The importance is that filtering modifies the expansion coefficients while preserving order of accuracy. Several variations of filtering methods applied to conservation laws on triangular elements can be found in the work of Meister et. al. [73]. Exponential filter, an extensively used filtering method in literature, is employed present work.

$$\sigma(\eta) = \begin{cases} 1, & 0 \le \eta \le \eta_c \\ e^{-\alpha((\eta - \eta_c)/(1 - \eta_c))^s}, & \eta_c < \eta \le 1 \end{cases}$$
(3.149)

where $\eta_c = \frac{N_c}{N}$. N_c is the cutoff value of untouched modes. In other words, modes having lower order than N_c are not filtered.

The filtering of solution coefficients is enabled through the filter matrix. The filter matrix is given as:

$$\mathcal{F} = \mathcal{V}\Lambda\mathcal{V}^{-1} \tag{3.150}$$

where Λ is a diagonal matrix.

$$\Lambda_{ii} = \sigma\left(\frac{i-1}{N}\right), \quad i = 1, ..., N_p \tag{3.151}$$

The parameters α and s of filter function are arbitrary. The α is generally chosen as negative logarithm of machine precision. Dissipation strength of filter function increases with increasing value of parameter α and decreasing the value of parameter s.

3.10.2 Slope Limiter

Slope limiting methodology developed for DG method by Tu and Aliabadi [74] is utilized in the present work. The presented limiter is van Albada type and claimed to not stall convergence and degrade the solution accuracy at smooth regions. The limiting procedure works with primitive variables and contains 5 steps. Before starting to limiting steps, a patch neighbour elements is constructed. As a first step, cell average values of primitive variables at centroids and element vertices are computed. The gradients at element faces are computed with Taylor series approach using average centroid and vertex values in second step. The third step is the calculation of gradients at element center using area averaging with gradients at element faces. Next step is to limit element gradients. Final step is to construct conservative variables using limited gradients.

3.11 Post Processing

Post processing of CFD data requires exporting the flow variables at physical coordinates. The state of the art finite volume solvers deliver flow variables at cell centroids or nodes. However, discontinuous Galerkin method enables more resolution inside a cell by utilizing nodal points. In the present work, nodal points are used in post processing as provided in Figure 3.11 for a fifth order solution having 15 nodes. The grid nodes are connected with solid lines while sub-element nodes are connected with dashed lines. The representation of flow data at sub-element nodes is more accurate approach however, double values show up at element boundaries. The post processors are unable to process double valued points hence it is expected to have discontinuous contour lines unless solution is highly smooth.



Figure 3.11: Grid Elements and Sub-Elements for a Fifth Order Solution

CHAPTER 4

RESULTS AND DISCUSSION

Focus of this chapter is solutions and discussions of selected test cases which are widely used in the literature. The first problem is translating isentropic vortex with analytical solution. The aim of using isentropic vortex problem is to test high order solution capability of present work. The second problem involves a smooth bump in channel with subsonic flow conditions. The importance of curved walls in high order methods is selected as the main subject of bump in a channel problem. The third and fourth problems is a validation case for a wing section. Moreover, in the third case, importance of stabilization in high order methods is tested since the problem involves shock formation. Fourth case addresses the comparison with a well known open source CFD software. Finally, last case is a supersonic flow with a oblique shock. The effect of grid refinement around the shock in high order methods is questioned.

The all test results presented in section, except isentropic vortex, obtained with nondimensional flow parameters and results are also in non-dimensional form. The flow parameters are given Table 4.1.

Table 4.1: Non-dimensional Values of Flow Parameters

$$p = 1.0 \quad \rho = \gamma \qquad T = 1.0$$
$$R = \frac{1}{\gamma} \qquad c = 1.0 \quad \gamma = 1.4$$

4.1 Isentropic Vortex

The isentropic vortex problem is widely used to validate high order methods in literature [75]. The analytical solution of the problem enables the evaluation of solver error directly at a given time. The another important feature of the problem is that it gives the ability to test dissipation characteristic of the numerical method which is crucial in vorticity dominated flow and high fidelity turbulent flow simulations.

The analytical formulation of the problem is provided in equation 4.1 [75].

$$u = 1 - \beta e^{1 - r^2} \frac{y - y_0}{2\pi},$$

$$v = \beta e^{1 - r^2} \frac{x - x_0}{2\pi},$$

$$\rho = \left(1 - \left(\frac{\gamma - 1}{16\gamma \pi^2}\right) \beta^2 e^{2(1 - r^2)}\right)^{\frac{1}{\gamma - 1}},$$

$$p = \rho^{\gamma},$$

$$r = \sqrt{(x - t - x_0)^2 + (y - y_0)^2}$$

(4.1)

where $\beta = 5$, $\gamma = 1.4$ and vortex center coordinates $x_0 = 5$ and $y_0 = 0$. The vortex is formulated such that velocity perturbations do not change entropy in the flow. Hence, numerical scheme employed to solve the problem is expected to have minimum entropy production. The initial density profile of the isentropic vortex problem is presented in Figure 4.1.



Figure 4.1: Density Contour for Isentropic Vortex at t = 0

A square computational domain is used. The vortex is located at the center of the computational domain. Domain boundaries have equivalent distance to the vortex center. Computational domain extends from 0 to 10 in x-direction and from -5 to 5 in y-direction. In literature, grid refinement methodology is used to verify the numerical accuracy as the main purpose of problem. The series of grids are generated in the computational domain having 16, 32 and 64 nodes on the boundaries. Initially, grids are generated in structured grid fashion using boundary elements. The structured domain is diagonalized to obtain unstructured triangular grid. The triangular grids used in convergence study are shown in Figure 4.2. Simulations are conducted with polynomial orders from 1 to 5 for each grids which results in total of 15 simulations. Dirichlet boundary conditions are employed at the domain boundaries which means that analytical solution is enforced at boundary elements at each time step. Local Lax-Friedrichs flux method is utilized due to its convenient nature to low subsonic





Figure 4.2: Meshes Used in the Convergence Analysis for Isentropic Vortex Problem

The L2-norm of the density error are plotted in Figure 4.3 for each polynomial order in order to show the accuracy of the scheme. The x-axis is the grid size while y-axis presents the density error in logarithmic scale. The convergence rate for each polynomial order computed using change of error with change of grid size. Convergence rate for polynomial order of 1, 2, 3, 4 and 5 is 1.52, 2.61, 3.17, 3.95 and 4.92 respectively. Theoretically, optimal convergence rate is $\mathcal{O}(h^{N+1})$, however convergence rate is observed to be around $\mathcal{O}(h^{N+\frac{1}{2}})$ which is suboptimal convergence rate.



Figure 4.3: L2 Norm of Density Error

4.2 Bump in a Channel

Inviscid smooth bump in a channel problem is a popular test case for high order CFD methods. Recently at 5th High Order CFD Workshop 2018, it is assigned to participants to test their solvers. The smooth bump test case aims to test high order CFD methods with curved boundary representation for the computation of internal flows. The flow through channel is subsonic with a Mach number of 0.5. The bump in the lower wall given by the equation 4.3 has smooth variation. The analytical solution of the problem is unknown however; since the flow is subsonic and inviscid, entropy should be constant in the channel. Hence, L2 norm of entropy error given in equation 4.2 can be used as the indication of accuracy.

$$Err_{L2} = \sqrt{\frac{\int_{\Omega} \left(\frac{p}{p_{\infty}}\left(\frac{\rho_{\infty}}{\rho}\right) - 1\right)^2 d\Omega}{\Omega}}$$
(4.2)

$$y = 0.0625e^{-25x^2} \tag{4.3}$$



Figure 4.4: Inviscid Smooth Bump in a Channel Problem

The computational domain presented in Figure 4.4 is bounded between x = -1.5and x = 1.5, and between y = 0 and y = 0.8. The inlet boundary is defined at x = -1.5 while outlet is defined at x = 1.5. The bump geometry is placed at the point [0,0] which is at equivalent distance to inlet and outlet boundaries. The subsonic inlet boundary condition is defined with total pressure, total temperature and Mach number. The static pressure is set at outflow boundary. Upper and lower boundaries are slip wall boundaries and bump geometry is represented with curved wall boundary approach. The flow is started from uniform M = 0.5 flow. HLL approximate Riemann solver is selected as numerical flux function. Time integration is carried out by fourth order explicit Runge-Kutta method.

Three meshes different in element sizes are generated to use in accuracy analysis. Coarse mesh has 667 elements. 1226 elements are utilized in medium mesh while fine mesh has 2444 elements. Coarse, medium and fine meshes are presented in Figure 4.5. The series of simulations are conducted with three meshes for polynomials order from 1 to 4 until convergence in entropy error achieved. Entropy error histories of coarse mesh solutions are presented in Figure 4.6 where x-axis is iteration number and y-axis is log scale of entropy error. The entropy error for all polynomial orders settled to a certain level. The low order simulations quickly converged however, error


Figure 4.5: Meshes Used in the Convergence Analysis for Inviscid Bump in a Channel Problem

reduction capabilities are limited. The high order simulations took longer time to converge with oscillatory behaviour.

The pressure contours from two different polynomial order on coarse mesh are presented in Figure 4.7. The pressure contour is ranged between 0.82 and 1.04. The outlet boundary condition is assumed to be undisturbed flow and has the pressure value of 1.0. Hence, pressure values lower than 1.0 represents suction regions. The solution presented in Figure 4.7.(a) uses elements with polynomial order of 1 while Figure 4.7.(b) presents the same solution for polynomial order of 4. The pressure field in Figure 4.7.(a) has discontinuities at regions high pressure gradient presents. Moreover, suction region at the bump peak is poorly captured. On the other hand, fourth order polynomials provide smooth well developed solution. Suction and high pressure regions are well captured. It can be summed up that high order solution can provide smooth solution on even coarse mesh while low order solution has disturbed regions.



Figure 4.6: L2 Norm of Entropy Error History for Polynomial Orders of 1, 2, 3 and 4 on Coarse Grid

The entropy error of each simulation is plotted with grid size, $h = 1/\sqrt{N_{DOF}}$, in Figure 4.8. The convergence rates are observed between $\mathcal{O}(h^{N+\frac{1}{2}})$ and $\mathcal{O}(h^{N+1})$. The expected convergence rate in High Order CFD Workshop is stated as $\mathcal{O}(h^{N+1})$ hence it can be concluded that consistent results are obtained.

4.2.1 Bump with Straight Sided Cells

In Chapter 3, the use of curved wall boundary representation is stated as mandatory to achieve high order accurate solution. In order to experience importance of curved wall



(b) Fifth Order DG Scheme, N=4

Figure 4.7: Steady State Pressure Contours of (a) Second and (b) Fifth Order Solutions

boundary condition, the bump in a channel solutions are computed with polynomial orders of 2 and 4 using straight sided cells on wall boundaries. The Figure 4.8 is re-drawn to compare curved and straight sided cells solution on Figure 4.9. As it is aforementioned, convergence rate is expected to be $O(h^{N+1})$. For the third order accurate solution, convergence rate drops from 3.08 to 1.69 and for the fifth order solution, convergence rates are 4.88 and 1.80. The convergence rate of straight sided cell simulations is around 2 regardless of polynomial order. Other than first order solutions, geometry representation of straight sided cells is identical to second order boundary representation of curved cells. Hence, it can be said that order of accuracy is limited by representation of wall boundaries.

In order to further investigate the accuracy loss, entropy generation in solution domain is calculated for each simulation. The maximum entropy generation in each simulation occurred around the bump however, away from the bump geometry in



Figure 4.8: Convergence Rate for Bump in Channel Problem

solution domain, generation of entropy is vanished. The entropy contours of simulations with curved and straight sided wall are compared in Figure 4.10 zooming at bump geometry. The maximum and minimum values of each contour plot is set equivalent. The grid and Legendre-Gauss-Lobatto nodes are also activated. The grid nodes are connected with solid lines while LGL nodes are connected dashed lines to show deformation of cell edges and node blending. In each simulation, maximum entropy generation occurs at the bump peak; however, simulations with straight sided cells have greater entropy generation. Another important observation is that, entropy error of curved cell simulations have smooth variation and diminish away from bump. However, in straight sided cell simulations, entropy error generated at bump geometry convected downstream and disturbed solution field.



Figure 4.9: Convergence Rate Comparison of Curved and Straight Sided Cell Solutions



Figure 4.10: Entropy Error Contours, (a) Curved Cell with Polynomial Order of 2, (b) Straight Sided Cell with Polynomial Order of 2, (c) Curved Cell with Polynomial Order of 4, (d) Straight Sided Cell with Polynomial Order of 4

4.3 RAE2822 Airfoil

The RAE2822 airfoil is a transonic airfoil which has a maximum camber of %2 positioned at %80 chord length and %22 maximum thickness to chord length ratio. The RAE2822 airfoil has a well documented test campaign which made it popular test case in computational fluid dynamics literature. The upper and lower surface coordinates of the airfoil and experimental results are obtained from test document [3]. The airfoil geometry is plotted in Figure 4.11.



Figure 4.11: RAE2822 Airfoil Geometry [3]

The solution domain for RAE2822 is generated between airfoil geometry and farfield boundary. Leading edge of the airfoil is located at [0, 0]. Curved wall boundary is applied to airfoil geometry. Farfield is generated as circular geometry with radius equal to 40 chord length and center at [0, 0]. The number of elements placed on the airfoil geometry is 120 while farfield boundary is divided into 40 equal elements. The domain between airfoil and farfield boundaries contains 1846 triangular elements. The generated grid is presented in Figure 4.12.(a). The zoomed viewed of grid around airfoil geometry is plotted on Figure 4.12.(b). HLL approximate Riemann solver is employed as numerical flux function.

4.3.1 RAE2822 Airfoil at Mach 0.3

The RAE2822 test campaign [3] does not include low Mach number flows. However, an inviscid shock-free flow around RAE2822 airfoil at zero angle of attack should



Figure 4.12: Solution Grid Used in RAE2822 Airfoil Simulations (a) Solution Grid,(b) Zoomed View on Airfoil Geometry

produce no drag force due to fact that pressure forces cancel in stream-wise direction. Hence, verification study can be carried out.

Verification analyses are conducted with polynomial order of 3. Simulation is run until convergence achieved. The density residual and drag force history is plotted in Figure 4.13. Iteration number is used as x-axis while density residual and drag force are plotted on y-axis. Logarithmic scale is applied to the y-axis of density residual plot. The density residual is reduced to around order of -11. Similarly, drag force initially oscillates and converges to value of 1.18E - 05 which is almost zero.

The pressure contour of converged solution is presented in Figure 4.14.(a). Pressure contour is ranged between 0.97 and 1.06 and spaced equally 21 lines. Figure 4.14.(b) shows Mach number contours which is also equally spaced 20 lines between values of 0.0 and 0.38. The contour plots show smooth variation of flow variables in solution domain. Moreover, drag force is obtained near zero as it is aimed. It can be concluded that numerical approach is verified.



Figure 4.13: Convergence History for RAE2822 Airfoil Solution at M = 0.3 (a) Density Residual Plot, (b) Drag Force Convergence History



Figure 4.14: RAE2822 Solution at M = 0.3, (a) Pressure Contour, (b) Mach Number Contour

4.3.2 RAE2822 Airfoil at Mach 0.73

The 2 transonic flow test cases of RAE2822, Case9 and Case10 [3], has actually become a standard test case for turbulence modelling. Case9 is run at Mach number of 0.73 and 2.8 degree angle of attack. The case9 is evaluated as subcritical flow condition where little to no separation occurs due to shock. However, Case10 is supercritical flow condition where massive separation is observed downstream of shock. Since the fidelity of present work is limited by governing equations, Case9 can be used as validation case.

The same grid and numerical flux used in low Mach number verification case is utilized. However, since there is a presence of shock, high gradient values in the solution field are expected. Therefore, stabilization technique is needed to be utilized.

4.3.2.1 RAE2822 Airfoil at Mach 0.73 with Exponential Filter

The transonic test case simulations are conducted with polynomial orders of 1. Exponential filter is applied to stabilize the simulation. The density residual history is plotted in Figure 4.15.(a). Pressure values on the airfoil geometry are extracted from converged solution. The pressure values are converted to pressure coefficient and compared with experimental results in Figure 4.15.(b). The simulation and experimental results match quite well in the lower surface. On the upper surface, significant deviations are observed up to shock position. Downstream of the shock, results are satisfactory.



Figure 4.15: RAE2822 Solution with Exponential Filter at M = 0.73 and α = 2.82, (a) Density Residual, (b) Pressure Coefficient [3]

The pressure and Mach number contours of the converged solution around the air-

foil geometry are presented in Figure 4.18.(a) and Figure 4.18.(b), respectively. The pressure contour is divided into equal 20 levels between 0.6 and 1.4. Mach number contour is consists of 25 levels with maximum value of 1.2 and minimum value of 0.0. The air accelerates on the upper surface resulting in shock formation. However, lower surface has smooth variation in flow field. Combining solution field contour plots and pressure coefficient plot, it is observed that exponential filter works well with smooth regions however fails at the high gradient regions and shows severe dissipative behaviour. Moreover, parameters of filter function are arbitrary, hence solution accuracy is dependent on the user.



(a) Pressure Contour

(b) Mach Number Contour

Figure 4.16: RAE2822 Solution with Exponential Filter at M = 0.73 and α = 2.82, (a) Pressure Contour, (b) Mach Number Contour

4.3.2.2 RAE2822 Airfoil at Mach 0.73 with Slope Limiter

The simulations conducted in section 4.3.2.1 are repeated with slope limiter. The density residual and pressure coefficient comparison are provided in Figure 4.17.(a) and Figure 4.17.(b), respectively. The lower surface pressure coefficient results are consistent with experimental results as in exponential filter approach. However, slope limiter approach is significantly more successful capturing the upper surface vales. Nonetheless, comparison results are not perfect. Leading edge region of upper surface shows notable discrepancy which may appear due to low resolution of surface



Figure 4.17: RAE2822 Solution with Slope Limiter at M = 0.73 and α = 2.82, (a) Density Residual, (b) Pressure Coefficient [3]

Using the same settings of flow field contours presented in Figure 4.18, Figure 4.18 is generated. The dissipation in filter approach is not observed in slope limiter approach. The flow field around the airfoil changes smoothly. The resolution of shock formation is considerably better. Hence, it is concluded that slope limiter approach is superior to filtering approach. However, it should be noted that slope limiting procedure reduces the maximum order of accuracy to 2 which is a contradiction to use of high order schemes.



Figure 4.18: RAE2822 Solution with Slope Limiter at M = 0.73 and α = 2.82, (a) Pressure Contour, (b) Mach Number Contour

4.4 NACA0012 Airfoil

The NACA0012 airfoil is a symmetric airfoil which has %12 maximum thickness to chord length ratio. The NACA0012 airfoil is probably the most popular airfoil in aeronautics. It is used in from helicopter rotors to airplane wing sections. Hence, it has become basic geometry for experimental studies and a standard test case for numerical studies. The upper and lower surface coordinates of the airfoil are computed with exact formulation of NACA. The Equation 4.4 used to generate airfoil coordinates and plotted in Figure 4.19.

$$y = \pm 0.6(0.2969\sqrt{x} - 0.1260x - 0.3516x^2 + 0.2843x^3 + 0.1015x^4)$$
(4.4)

NACA0012 airfoil simulations are carried out at Mach number of 0.8 and angle of attack of 1.25 degree condition, a popular transonic test case. The grid generated for NACA0012 airfoil simulations are presented in Figure 4.20.(a) full view and in Figure 4.20.(b) zoomed view around airfoil geometry. The airfoil geometry has 120 elements along its surface and leading edge is placed at [0, 0]. The Riemann farfield boundary is circular and has a radius of 30 chord length. The grid consists of 1126

elements. The Roe approximate Riemann solver is used as numerical flux. Slope limiter is also used to stabilize numerical model. Due to slope limiter, polynomial order of 1 is used in simulations.

The solutions obtained with discontinuous Galerkin method are compared with SU2 open source CFD software utilizing finite volume method [76]. Explicit second order Euler solver of SU2 is employed to solve same grid provided in Figure 4.20 with Venkatakrishnan slope limiter and Roe approximate Riemann solver.

The Mach number contour of two results are plotted in Figure 4.21. There are two shock formations observed on the airfoil surface. The air accelerates on the upper surface due to thickness of airfoil and angle of attack. Hence, the upper surface shock is stronger than the lower surface shock which is only due to airfoil thickness. The comparison shows that DG solver is able to capture shocks with similar resolution to SU2 solver. The pressure coefficient plot provided in Figure 4.22 shows that results are almost identical. Shock positions and strength are well correlated while overall pressure distribution is also matching. It can be concluded that methodology of the present work is verified.

It is also found worth to point out the effect of post processing approach mentioned in Chapter 3. The Mach contour of SU2 solution is plotted with classical approach of finite volume methods. Although pressure distributions are almost identical, Mach contour of DG solver visualize a sharper shock which is more accurate presentation as it was claimed.



Figure 4.19: NACA0012 Airfoil Geometry



Figure 4.20: Solution Grid Used in NACA0012 Airfoil Simulations (a) Solution Grid, (b) Zoomed View on Airfoil Geometry



Figure 4.21: NACA0012 Solution Mach Number Contour M = 0.8 and α = 1.25, (a) DG, (b) SU2



Figure 4.22: Pressure Coefficient for NACA0012 Solutions at M = 0.8, $\alpha = 1.25$ degree

4.5 Supersonic Wedge

Supersonic wedge problem is a numerical study of inviscid supersonic flow field on wedge with 15 degrees half angle. Freestream flows through the domain with Mach number of 2.5. Due to presence of wedge, flow is upturned 15 degrees resulting in a oblique shock formation.

The numerical model of supersonic wedge problem is presented in Figure 4.23. The leading edge of wedge is located at [0, 0]. Half wedge angle of 15 degrees is measured from the x-axis. The wedge geometry is extended up to outlet boundary located at x = 3.5 line. The inlet boundary is placed upstream of the wedge at coordinate of [-1.5, 0]. Upper boundary of the domain is also modelled as Euler wall and placed far

away enough to prevent shock reflection. The polynomial order of elements is set to 1 since slope limiter has to be employed due to presence of shock. Roe's approximate Riemann solver is used to calculate numerical flux.

Analytical solution of the problem is well known through theta-beta-Mach relations [77]. Analytical solution provides the change of flow properties across the shock. For inviscid flow, flow properties are uniform upstream and downstream of the shock.

Success of simulation is dependent on accurate resolution of shock which is severely affected by grid resolution. Hence, four different grids are generated for supersonic wedge problem. The grids are presented in Figure 4.24. The first and second grid are generated with structured fashion and triangulated. First grid contains 32 elements on all boundaries. Second grid is obtained by refining first grid to 64 elements on boundaries. Third grid is generated by triangulation of domain while keeping 32 elements on boundaries similar to first grid. The significance of third grid is that a refinement region around the shock is applied. The last grid utilizes the refinement exactly at the theoretical shock position.

The simulations are conducted for each grid until convergence achieved. The residual histories are presented in Figure 4.25 for all grids. The significant difference in convergence of grid with refinement at theoretical shock position is observed. Reaching convergence takes almost three times longer than the other grids. The results of the simulations are compared with analytical solution in Figure 4.26 which shows pressure distribution along lower wall. Moreover, Mach number contours are also presented in Figure 4.27. The coarse uniform grid poorly captures the shock and pressure distribution along the wedge. The fine uniform grid resolves the shock quite well. Pressure distribution of fine grid shows small deviation and oscillation just after the leading edge. The pressure distribution of two grids with refinement are almost identical to analytical solution. However, Mach number contours differs around the shock. The grid with refinement at analytical shock position provides a very sharp shock. On the other hand, shock resolution of grid with refinement around shock is weaker than fine uniform grid. Combining all the results, it can be concluded that numerical method is verified for supersonic flows and grid refinement around shock waves significantly improves the results. Moreover, other than refinement, alignment

of elements with respect to shock is also important parameter.



Figure 4.23: Supersonic Wedge Problem Definition



Figure 4.24: Grids for Supersonic Wedge Problem, (a) Uniform Grid 32x32, (b) Uniform Grid 64x64, (c) Ramp Grid Refinement 1, (d) Ramp Grid Refinement 2



Figure 4.25: Residual Histories of Supersonic Wedge Simulations, (a) Uniform Grid 32x32, (b) Uniform Grid 64x64, (c) Ramp Grid Refinement 1, (d) Ramp Grid Refinement 2



Figure 4.26: Pressure Distribution Along Wedge Wall, (a) Uniform Grid 32x32, (b) Uniform Grid 64x64, (c) Ramp Grid Refinement 1, (d) Ramp Grid Refinement 2



Figure 4.27: Entropy Error Contours, (a) Curved Cell with Polynomial Order of 2, (b) Straight Sided Cell with Polynomial Order of 2, (c) Curved Cell with Polynomial Order of 4, (d) Straight Sided Cell with Polynomial Order of 4

CHAPTER 5

CONCLUSION

In this thesis, solutions of Euler equations are studied using Runge-Kutta discontinuous Galerkin finite element method on triangular grids in an effort to explore a technique which can provide high order accuracy. Test cases widely used in CFD community are used for verification and validation purposes. Moreover, comparison study with an open-source CFD software is also conducted.

The explicit time integration strategy in discontinuous Galerkin method suffers severely from stability concerns. Hence, very strict limitations are applied to time step choice that resulted in enormous number of iterations to reach convergence. Therefore, focusing on implicit time integration methods should be number one topic in table for future studies.

The high order accuracy is easily achieved in smooth problems where any discontinuity or shock is not present. The density error of isentropic vortex problem having analytical solution and entropy error of bump in a channel problem were output of interest while monitoring order of accuracy. Moreover, drag force prediction of airfoil at low subsonic flow is also observed to be improved with increasing order of solution. However, stabilization techniques are noted to employed due to fluctuations at high gradient regions or around shocks leads to non-physical values or even failure of simulation. Among the two techniques studied in present work, filtering approach is found to be resulting in too much dissipation if the filter parameters are not correctly set up. Hence filtering approach for stabilization of solution is highly user dependent and increases the workload while finding optimum filter parameters. On the other hand, slope limiting approach provides more accurate solutions that can be confirmed with validation results. However, slope limiting procedure limits the order of accuracy to two which is a contradiction with purpose of high order methods. Thus, stabilization techniques to be worked in future studies should not result in too much dissipation while preserving high order of accuracy.

Another important point, also related with grid generation, is representation of wall boundaries. The solutions of bump in a channel problem with curved and standard wall boundary approach clearly shows that curved wall boundary application is mandatory in order to go beyond second order of accuracy. Therefore, high order grid generation or smart algorithms to capture wall curvature should be included in future researches.

The effect of grid refinement around shock showed significant improvement in solution results of supersonic wedge problem. Grid refinement algorithms that would also require shock detection methodologies can be another topic for future studies.

REFERENCES

- J. S. Hesthaven and T. Warburton. Nodal Discontinuous Galerkin Methods. Springer, 2008.
- [2] M. H. Carpenter and A. Kennedy. Fourth-Order 2N-Storage Runge-Kutta Schemes. Technical report, 1994.
- [3] P.H. Cook, M.A. McDonald, and M.C.P Firmin. *Aerofoil RAE 2822 Pressure Distributions, and Boundary Layer and Wake Measurements.* 1979.
- [4] C. Hirsch. Numerical Computation of Internal and External Flows: The Fundamentals of Computational Fluid Dynamics. 2007.
- [5] J. H. Ferziger and M. Peric. *Computational Methods for Fluid Dynamics*, volume 46. 2003.
- [6] D. V. Gaitonde and M. R. Visbal. High-Order Schemes for Navier-Stokes Equations: Algorithm and Implementation into FDL3DI. *Air Vehicles Directorate*, page 50, 1998.
- [7] P. L. Roe. Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes, 1981.
- [8] B. van Leer. Upwind-Difference Methods for Aerodynamic Problems Governed by the Euler Equations. In *Large-Scale Computations in Fluid Mechanics; Proceedings of the Fifteenth Summer Seminar on Applied Mathematics*, pages 327– 336, 1985.
- [9] A. Jameson, W. Schmidt, and E. Turkel. Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time-Stepping Schemes. In AIAA 14th Fluid and Plasma Dynamic Conference, volume M, pages 1981– 1259, 1981.
- [10] S. Patankar. Numerical Heat Transfer and Fluid Flow. 1980.

- [11] O. C. Zienkiewicz and R. L. Taylor. The Finite Element Method Vol. 1. 1994.
- [12] R. Codina. Comparison of Some Finite Element Methods for Solving the Diffusion-Convection-Reaction Equation. *Comput. Methods Appl. Mech. En*grg, 156(97):185–210, 1998.
- [13] I. Christie, D. F. Griffiths, A. R. Mitchell, and O. C. Zienkiewicz. Finite Element Methods for Second Order Differential Equations with Significant First Derivatives. *International Journal for Numerical Methods in Engineering*, 10(6):1389–1396, 1976.
- [14] R. Codina. Finite Element Approximation of the Convection-Diffusion Equation: Subgrid-Scale Spaces, Local Instabilities and Anisotropic Space-Time Discretizations. In *Lecture Notes in Computational Science and Engineering*, volume 81 LNCSE, pages 85–97, 2011.
- [15] T. J.R. Hughes. Multiscale Phenomena: Green's functions, the Dirichlet-to-Neumann formulation, Subgrid Scale Models, Bubbles and the Origins of Stabilized Methods. *Computer Methods in Applied Mechanics and Engineering*, 127(1-4):387–401, 1995.
- [16] A. Jameson. Computational Fluid Dynamics Past, Present and Future, 2012.
- [17] Krzysztof J. F. A High-Order Discontinuous Galerkin Multigrid Solver for Aerodynamic Applications. PhD thesis, MASSACHUSETTS INSTITUTE OF TECHNOLOGY, 2004.
- [18] W. H. Reed and T. R. Hill. Triangular Mesh Methods for the Neutron Transport Equation. *Triangular mesh methods for the neutron transport equation*, Technical(LA-UR-73-479):1–23, 1973.
- [19] P. Lesaint and P.-a. Raviart. On a Finite Element Method for Solving the Neutron Transport Equation. *Mathematical Aspects of Finite Elements in Partial Differential Equations*, pages 89–145, 1974.
- [20] G. Chavent and G. Salzano. A Finite-Element Method for the 1-D Water Flooding Problem with Gravity. *Journal of Computational Physics*, 45(3):307–344, 1982.

- [21] B. Cockburn and C.-W. Shu. The Runge-Kutta Local Projection P1- Discontinuous Galerkin Method for Scalar Conservation Laws, 1991.
- [22] B. Cockburn and C.-W. Shu. TVB Runge-Kutta Local Projection Discontinuous Galerkin Finite Element Method for Conservation Laws II: General Framework. *Mathematics of Computation*, 52(186):411, 1989.
- [23] B. Cockburn, S. Y. Lin, and C. W. Shu. TVB Runge-Kutta Local Projection Discontinuous Galerkin Finite Element Method for Conservation Laws III: One-Dimensional Systems. *Journal of Computational Physics*, 84(1):90–113, 1989.
- [24] B. Cockburn, S. Hou, and C.-W. Shu. The Runge-Kutta Local Projection Discontinuous Galerkin Finite Element Method for Conservation Laws. IV. The Multidimensional Case. *Mathematics of Computation*, 54(190):545–545, 1990.
- [25] B. Cockburn, G. E. Karniadakis, and C.-W. Shu. The Development of Discontinuous Galerkin Methods. In *Discontinuous Galerkin Methods*, volume 11, pages 3–50. 2000.
- [26] S. R. Allmaras and M. Giles. A Second Order Flux Split Scheme for the Unsteady 2-D Euler Equations on Arbitrary Meshes. In 8TH COMPUTATIONAL FLUID DYNAMICS CONFERENCE, Honolulu, HI, U.S.A., 1987.
- [27] S. R. Allmaras. A coupled Euler/Navier-Stokes Algorithm for 2-D Unsteady Transonic Shock/Boundary-Layer Interaction. Ph. d. thesis, Massachusetts Institute of Technology, 1989.
- [28] B. Van Leer. Towards the Ultimate Conservative Difference Scheme. IV. A New Approach to Numerical Convection. *Journal of Computational Physics*, 23(3):276–299, 1977.
- [29] D. W. Halt. A Compact Higher-Order Euler Solver for Unstructured Grids. Ph. d. thesis, Washington University, 1992.
- [30] M. F. Wheeler. An Elliptic Collocation-Finite Element Method with Interior Penalties. SIAM Journal on Numerical Analysis, 15(1):152–161, 1978.
- [31] D. N. Arnold. An Interior Penalty Finite Element Method with Discontinuous Elements. SIAM Journal on Numerical Analysis, 19(4):742–760, 1982.

- [32] I. Babuška, C. E. Baumann, and J. T. Oden. Discontinuous hp Finite Element Method for Diffusion Problems: 1-D Analysis. *Computers and Mathematics with Applications*, 37(9):103–122, 1999.
- [33] C. E. Baumann and J. Tinsley Oden. A Discontinuous hp Finite Element Method for Convection-Diffusion Problems. *Computer Methods in Applied Mechanics and Engineering*, 175(3-4):311–341, 1999.
- [34] F. Bassi and S. Rebay. A High-Order Accurate Discontinuous Finite Element Method for the Numerical Solution of the Compressible Navier-Stokes Equations. *Journal of Computational Physics*, 131(2):267–279, 1997.
- [35] F. Bassi, A. Crivellini, S. Rebay, and M. Savini. Discontinuous Galerkin solution of the Reynolds-averaged Navier-Stokes and k-ω turbulence model equations. *Computers and Fluids*, 34(4-5 SPEC.ISS.):507–540, 2005.
- [36] J. Peraire and P.-O. Persson. The Compact Discontinuous Galerkin (CDG) Method for Elliptic Problems. SIAM Journal on Scientific Computing, 30(4):1806–1824, 2008.
- [37] S. Premasuthan, C. Liang, and A. Jameson. Computation of Flows with Shocks Using Spectral Difference Scheme with Artificial Viscosity. 48th Aerospace Sciences Meeting, (January):2010, 2011.
- [38] G. E. Barter and D. L. Darmofal. Shock Capturing with PDE-Based Artificial Viscosity for DGFEM: Part I. Formulation. *Journal of Computational Physics*, 229(5):1810–1827, 2010.
- [39] P.-O. Persson and J. Peraire. Sub-Cell Shock Capturing for Discontinuous Galerkin Methods. In 44th AIAA Aerospace Sciences Meeting and Exhibit, 2006.
- [40] F. Bassi, A. Crivellini, A. Ghidoni, and S. Rebay. High-order Discontinuous Galerkin Discretization of Transonic Turbulent Flows. In 47th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, 2009.
- [41] N. K. Burgess. An Adaptive Discontinuous Galerkin Solver for Aerodynamic Flows. Ph. d. thesis, University of Wyoming, 2011.

- [42] N. K. Burgess and D. J. Mavriplis. An hp-Adaptive Discontinuous Galerkin Solver for Aerodynamic flows on Mixed-Element Meshes. In 49th AIAA Aerospace Sciences Meeting, number January, pages 1–18, 2011.
- [43] P. Houston and E. Süli. A Note on the Design of hp-Adaptive Finite Element Methods for Elliptic Partial Differential Equations. *Computer Methods in Applied Mechanics and Engineering*, 194(2-5 SPEC. ISS.):229–243, 2005.
- [44] R. Hartmann. Adaptive Discontinuous Galerkin Methods with Shock Capturing for the Compressible Navier–Stokes Equations. *International Journal for Numerical Methods in Fluids*, 51(9-10):1131–1156, 2006.
- [45] D. C. Wilcox. Formulation of the k-ω Turbulence Model Revisited. AIAA Journal, 46(11):2823–2838, 2008.
- [46] N. Nguyen, P.-O. Persson, and J. Peraire. RANS Solutions Using High Order Discontinuous Galerkin Methods. In 45th AIAA Aerospace Sciences Meeting and Exhibit, 2007.
- [47] T. A. Oliver. A High-Order, Adaptive, Discontinuous Galerkin Finite Element Method for the Reynolds-Averaged Navier-Stokes Equations. Ph. d. thesis, Massachusetts Institute of Technology, 2008.
- [48] B. S. S. Collis. Discontinuous Galerkin Methods for Turbulence Simulation.
 Proceedings of the 2002 Center for Turbulence Research Summer Program, (1):155–167, 2002.
- [49] A. Garai, L. Diosady, S. Murman, and N. Madavan. DNS of Flow In a Low-Pressure Turbine Cascade Using a Discontinuous-Galerkin Spectral-Element Method. In *Proceedings of ASME Turbo Expo*, pages 1–10, 2015.
- [50] L. Wei and A. Pollard. Direct Numerical Simulation of Compressible Turbulent Channel Flows Using the Discontinuous Galerkin Method. *Computers & Fluids*, 47(1):85–100, 2011.
- [51] J. B. Chapelier, M. de la Llave Plata, F. Renac, and E. Lamballais. Evaluation of a High-Order Discontinuous Galerkin Method for the DNS of Turbulent Flows. *Computers and Fluids*, 95:210–226, 2014.

- [52] G. J. Gassner and A. D. Beck. On the Accuracy of High-Order Discretizations for Underresolved Turbulence Simulations. In *Theoretical and Computational Fluid Dynamics*, volume 27, pages 221–237, 2013.
- [53] C. C. de Wiart, K. Hillewaert, L. Bricteux, and G. Winckelmans. Implicit LES of Free and Wall-Bounded Turbulent Flows Based on the Discontinuous Galerkin/Symmetric Interior Penalty Method. *International Journal for Numerical Methods in Fluids*, 78(6):335–354, 2015.
- [54] A. Abbà, L. Bonaventura, M. Nini, and M. Restelli. Dynamic Models for Large Eddy Simulation of Compressible Flows with a High Order DG Method. *Computers & Fluids*, 122:209–222, 2015.
- [55] A. Uranga, P.-O. Persson, M. Drela, and J. Peraire. Implicit Large Eddy Simulation of Transition to Turbulence at Low Reynolds Numbers Using a Discontinuous Galerkin Method. *International Journal for Numerical Methods in Engineering*, 87(1-5):232–261, 2011.
- [56] F. Bassi, L. Botti, A. Colombo, A. Crivellini, A. Ghidoni, and F. Massa. On the Development of an Implicit High-Order Discontinuous Galerkin Method for DNS and Implicit LES of Turbulent Flows. *European Journal of Mechanics*, *B/Fluids*, 55:367–379, 2016.
- [57] T. D. Economon, F. Palacios, S. R. Copeland, T. W. Lukaczyk, and J. J. Alonso. SU2: An Open-Source Suite for Multiphysics Simulation and Design. *AIAA Journal*, 54(3):828–846, 2016.
- [58] H. G. Weller, G. Tabor, H. Jasak, and C. Fureby. A Tensorial Approach to Computational Continuum Mechanics Using Object-Oriented Techniques. *Computers in Physics*, 12(6):620, 1998.
- [59] Xiao-Guang Ren Li-Yang Xu, Shuai Ye, Yong-Quan Feng, Hao Li, Xin-Hai Xu. High Order Accurate Curved-Element Implementation in OpenFOAM for Discontinuous Galerkin Method. Technical report, Advanced Science and Technology Letters, 2016.
- [60] F. Hindenlang, G. J. Gassner, C. Altmann, A. Beck, M. Staudenmaier, and C. D.

Munz. Explicit Discontinuous Galerkin Methods for Unsteady Problems. *Computers and Fluids*, 61:86–93, 2012.

- [61] C. D. Cantwell, D. Moxey, A. Comerford, A. Bolis, G. Rocco, G. Mengaldo, D. De Grazia, S. Yakovlev, J. E. Lombard, D. Ekelschot, B. Jordi, H. Xu, Y. Mohamied, C. Eskilsson, B. Nelson, P. Vos, C. Biotto, R. M. Kirby, and S. J. Sherwin. Nektar++: An Open-Source Spectral/hp Element Framework. *Computer Physics Communications*, 192:205–219, 2015.
- [62] A. Logg, K. Mardal, and G. Wells. Fenics: Automated Solution of Differential Equations by the Finite Element Method. In *Ieeexplore.Ieee.Org*, pages 337 – 359. 2011.
- [63] A. Dedner, R. Klöfkorn, M. Nolte, and M. Ohlberger. A Generic Interface for Parallel and Adaptive Discretization Schemes: Abstraction Principles and the Dune-FEM Module. *Computing (Vienna/New York)*, 90(3-4):165–196, 2010.
- [64] G. Szegő. Orthogonal Polynomials, volume 23. 1939.
- [65] J. S. Hesthaven. From Electrostatics to Almost Optimal Nodal Sets for Polynomial Interpolation in a Simplex. SIAM Journal on Numerical Analysis, 35(2):655–676, 1998.
- [66] M. Dubiner. Spectral Methods on Triangles and Other Domains. Journal of Scientific Computing, 1991.
- [67] M. S. Floater. Mean Value Coordinates. *Computer Aided Geometric Design*, 2003.
- [68] A. Karakus. GPU Accelerated High-Order Discontinuous Galerkin Level Set Methods for Incompressible Multiphase Flows. Phd. thesis, Middle East Technical University, 2015.
- [69] T. Warburton. An Explicit Construction of Interpolation Nodes on the Simplex. Journal of Engineering Mathematics, 2006.
- [70] E. F. Toro. Riemann Solvers and Numerical Methods for Fluid Dynamics. 2009.
- [71] F. Bassi and S. Rebay. High-Order Accurate Discontinuous Finite Element Solution of the 2D Euler Equations. *Journal of Computational Physics*, 1997.

- [72] W. J. Gordon and C. A. Hall. Transfinite Element Methods: Blending-Function Interpolation Over Arbitrary Curved Element Domains. *Numerische Mathematik*, 1973.
- [73] A. Meister, S. Ortleb, and Th Sonar. Application of Spectral Filtering to Discontinuous Galerkin Methods on Triangulations. *Numerical Methods for Partial Differential Equations*, 2012.
- [74] S. Tu and S. Aliabadi. A Slope Limiting Procedure in Discontinuous Galerkin Finite Element Method for Gas Dynamics Applications. *International Journal* of Numerical Analysis and Modeling, 2005.
- [75] S. C. Spiegel, H. T. Huynh, and J. R. DeBonis. A Survey of the Isentropic Euler Vortex Problem Using High-Order Methods. In 22nd AIAA Computational Fluid Dynamics Conference, 2015.
- [76] T. D. Economon, F. Palacios, S. R. Copeland, T. W. Lukaczyk, and J. J. Alonso. SU2: An Open-Source Suite for Multiphysics Simulation and Design. *AIAA Journal*, 2016.
- [77] J. D. Anderson. Fundamentals of Aerodynamics. McGraw-Hill, 2007.