ABSTRACT

SHARMA, ABHINAV. A Reconstruction based Finite Volume Method for Diffusion Equation on Unstructured Grids. (Under the direction of Dr. Hong Luo.)

A reconstructed finite volume method is developed on unstructured grids for diffusion problem. In this method, the underlying cell-average finite volume solution is first used to reconstruct a linear polynomial solution via a least-squares reconstruction scheme, and the reconstructed linear polynomial solution in each cell is then used to reconstruct a continuous linear polynomial solution on the union of two neighboring, interface-sharing cells. The resulting system of linear equations is solved using a preconditioned GMRES method. The scheme is found to be grid transparent and second order accurate for regular unstructured and distorted meshes also. It is proved to be consistent on uniform grids and is used to solve diffusion equation for a variety of meshes and for different diffusion tensors. The robustness and accuracy of the scheme is confirmed on strongly anisotropic diffusion tensors. The robustness of the scheme is further confirmed by its handling of the jumps or discontinuities in the heterogeneous diffusion tensor. A comparison study between the reconstruction scheme and Modified Average Gradient approach proves that the reconstruction scheme gives better accuracy and lower absolute errors for symmetric and non symmetric stretched meshes. Finally a convergence study is also discussed that compares the effectiveness of different preconditioner with the GMRES solver. A computationally efficient and accurate method to compute the numerical jacobian is also discussed.

by
Abhinav Sharma

A thesis submitted to the Graduate Faculty of North Carolina State University in partial fulfillment of the requirements for the degree of Master of Science

Aerospace Engineering

Raleigh, North Carolina

2011

APPROVED BY:

__________________________________________  __________________________________________
Dr. Hong Luo                                   Dr. Jack Edwards
Committee Chair

__________________________________________
Dr. Tarek Eccheki
DEDICATION

To my Parents.
BIOGRAPHY

Abhinav Sharma has received his Bachelor in Technology from VIT University, India in Mechanical Engineering. Currently he is pursuing the degree of Master of Science from North Carolina State University, USA in Aerospace Engineering.
ACKNOWLEDGMENTS

I would like to thank my advisor Dr. Hong Luo for providing me with an opportunity to carry out my thesis under his supervision and all the help and guidance in completing this work successfully. The suggestions and directions that he gave to me at the crucial times of my thesis, were of tremendous importance to me. I would also like to thank my committee members Dr. Jack Edwards and Dr. Tarek Eccheki for their valuable suggestions during the course of thesis. I would also like to thank my family and friends for continually supporting me even in the difficult times.
# TABLE OF CONTENTS

List of Tables .................................................................................................................. vii

List of Figures .................................................................................................................. ix

## Chapter 1 Introduction ............................................................................................... 1

1.1 Background and Motivation ....................................................................................... 2

1.2 Scope of Thesis ........................................................................................................... 5

1.3 Outline of Thesis ....................................................................................................... 5

## Chapter 2 Governing Equations and Numerical Methods ........................................... 7

2.1 Problem Definition ..................................................................................................... 7

2.2 Numerical Methods ................................................................................................... 7

2.2.1 Finite Volume Formulation .................................................................................... 7

2.2.2 Common Approaches for estimation of Gradients at Interfaces ......................... 8

2.2.3 Reconstruction Scheme ........................................................................................ 11

2.3 Solving System of Equations ................................................................................... 18

2.3.1 Generalized Minimum Residual (GMRES) ............................................................ 18

2.3.2 Implementation of GMRES .................................................................................. 19

2.3.3 Preconditioner ...................................................................................................... 20

2.3.4 Jacobian Evaluation .............................................................................................. 21

2.3.5 Diagonal Preconditioner ....................................................................................... 25

2.3.6 LU-SGS Preconditioner ....................................................................................... 25

2.3.7 ILU Preconditioner ............................................................................................... 26

## Chapter 3 An Analysis of Existing Viscous Flux Formulae ......................................... 28

3.1 Model Grid Topology and Formulae ........................................................................ 28

3.2 Discrete Maximum Principle .................................................................................... 29

3.3 Accuracy .................................................................................................................. 30

3.4 Consistency .............................................................................................................. 31

3.5 Common Approaches for estimation of gradients at Cell Interfaces ....................... 32

3.5.1 Modified Average Gradient Method ...................................................................... 32

3.5.2 Green Gauss Reconstruction using Diamond Path Approach ............................ 35
3.6 Reconstruction Scheme ................................................................. 39

Chapter 4 Numerical Experiments ......................................................... 43

4.1 Assessment of Condition Number .................................................. 43
4.2 Positivity Study ........................................................................... 46
4.3 Accuracy Study .......................................................................... 49
4.4 Accuracy Study on Distorted Meshes ............................................ 63
4.5 Study on Heterogeneous Diffusion Tensor ..................................... 71
4.6 Convergence Analysis .................................................................. 74
4.7 Comparison for Reconstruction scheme and Modified Gradient Average .......... 76

Conclusion ......................................................................................... 83
References ......................................................................................... 84
LIST OF TABLES

Table 2.1    Number of columns and groups formed after CPR algorithm .................. 25
Table 4.1    Condition Numbers for gradient at cell centers for Regular Triangular Meshes .. 44
Table 4.2    Condition Numbers for gradient at cell center for Unstructured Meshes ........ 44
Table 4.3    Condition Numbers for gradients at cell centers for Anisotropic Meshes ...... 44
Table 4.4    Condition Numbers for gradient at cell interfaces for Regular Triangular
              Meshes ............................................................................................................. 45
Table 4.5    Condition Numbers for gradient at cell interfaces for Anisotropic Meshes ..... 45
Table 4.6    Condition Numbers for gradient at cell interfaces for Unstructured Meshes .... 45
Table 4.7    Results for test case 1 on Regular Triangular Meshes for Accuracy Study ...... 49
Table 4.8    Results for test case 1 on Unstructured Meshes for Accuracy Study .......... 50
Table 4.9    Results for test case 1 on Anisotropic Meshes for Accuracy Study ............ 50
Table 4.10   Comparison between Reconstruction scheme and CDG Methods .................. 51
Table 4.11   Comparison between Reconstruction and Recovery methods ..................... 53
Table 4.12   Results for test case 2 on Regular Triangular Meshes for Accuracy Study ...... 54
Table 4.13   Results for test case 2 on Unstructured Meshes for Accuracy Study .......... 55
Table 4.14   Results for test case 2 on Anisotropic Meshes for Accuracy Study .......... 55
Table 4.15   Results for test case 3 on Regular Triangular Meshes for Accuracy Study ..... 58
Table 4.16   Results for test case 3 on Unstructured Meshes for Accuracy Study .......... 58
Table 4.17   Results for test case 3 on Anisotropic Meshes for Accuracy Study .......... 59
Table 4.18   Results for test case 4 on Regular Triangular Meshes for Accuracy Study ...... 60
Table 4.19   Results for test case 4 on Unstructured Meshes for Accuracy Study .......... 60
Table 4.20   Results for test case 4 on Anisotropic Meshes for Accuracy Study .......... 61
Table 4.21   Results for test case 1 for accuracy study on distorted meshes with Distortion
              Parameter = 0.6.................................................................................................. 65
Table 4.22   Results for test case 2 for accuracy study on distorted meshes with Distortion
              Parameter = 0.6.................................................................................................. 66
Table 4.23   Results for test case 3 for accuracy study on distorted meshes with Distortion
              Parameter = 0.4.................................................................................................. 68
Table 4.24  Results for test case 3 for accuracy study on distorted meshes with Distortion Parameter = 0.6…………………………………………………………………………………68
Table 4.25  Results for test case 3 for accuracy study on distorted meshes with Distortion Parameter = 0.8…………………………………………………………………………………69
Table 4.26  CPU Times and number of iterations taken by each Preconditioner ………… 74
Table 4.27  Comparison between reconstruction scheme and modified gradient approach on Regular Triangular meshes for test case 1 ………………………………………… 77
Table 4.28  Comparison between reconstruction scheme and modified gradient approach on Unstructured meshes for test case 1 ………………………………………… 78
Table 4.29  Comparison between reconstruction scheme and modified gradient approach on Anisotropic meshes for test case 1 ………………………………………… 78
Table 4.30  Comparison between Reconstruction and Modified Average Approach over set of Symmetric Stretched Meshes ………………………………………………… 80
Table 4.31  Comparison between Reconstruction and Modified Average Approach over set Of Non Symmetric Stretched Meshes ………………………………………………… 81
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Diamond Path Reconstruction. Simple Path</td>
<td>11</td>
</tr>
<tr>
<td>2.2</td>
<td>A Typical Triangular element with neighboring cells</td>
<td>12</td>
</tr>
<tr>
<td>2.3</td>
<td>Two Neighboring cells considered as a union</td>
<td>15</td>
</tr>
<tr>
<td>2.4</td>
<td>A union of two neighboring cells</td>
<td>19</td>
</tr>
<tr>
<td>3.1</td>
<td>A Uni-Directional Stretched Grid</td>
<td>28</td>
</tr>
<tr>
<td>3.2</td>
<td>Diamond Path Reconstruction Stencil</td>
<td>34</td>
</tr>
<tr>
<td>3.3</td>
<td>Simple Averaging Procedures at Subtended Vertex</td>
<td>35</td>
</tr>
<tr>
<td>3.4</td>
<td>Simple Reconstruction Diamond Path</td>
<td>36</td>
</tr>
<tr>
<td>3.5</td>
<td>Stencil for Diamond Path Reconstruction on Uni-Directional Stretched Grids</td>
<td>37</td>
</tr>
<tr>
<td>3.6</td>
<td>1D Uni-Directional Stretched Mesh</td>
<td>38</td>
</tr>
<tr>
<td>4.1</td>
<td>Sets of Meshes used for experiments of Non Negativity</td>
<td>47</td>
</tr>
<tr>
<td>4.2</td>
<td>Results of Non Negativity tests on sets of Meshes considered</td>
<td>48</td>
</tr>
<tr>
<td>4.3</td>
<td>Plots for comparison between reconstruction and CDG method</td>
<td>52</td>
</tr>
<tr>
<td>4.4</td>
<td>Plots for order of Accuracy for test case 1 and test case 2 on different meshes considered</td>
<td>56</td>
</tr>
<tr>
<td>4.5</td>
<td>Contour Plots for Numerical Solution and Exact Solution for test case 2</td>
<td>57</td>
</tr>
<tr>
<td>4.6</td>
<td>Logarithmic Plots for order of Accuracy for test case 3 and test case 4 for Different meshes</td>
<td>62</td>
</tr>
<tr>
<td>4.7</td>
<td>Contour plots for Numerical Solution and Exact solution for test case 4</td>
<td>62</td>
</tr>
<tr>
<td>4.8</td>
<td>Different types of Distorted Meshes considered</td>
<td>63</td>
</tr>
<tr>
<td>4.9</td>
<td>Refined Distorted Meshes with Distortion Parameter as 0.6</td>
<td>64</td>
</tr>
<tr>
<td>4.10</td>
<td>Logarithmic Plots for test case 1 and test case 2 for Accuracy Study on Distorted Meshes</td>
<td>67</td>
</tr>
<tr>
<td>4.11</td>
<td>Logarithmic Plots for test case 3 for Accuracy Study on Distorted Meshes with Different diffusion parameter</td>
<td>69</td>
</tr>
<tr>
<td>4.12</td>
<td>Results showing Jumps across Mesh Edges in case of an anisotropic mesh</td>
<td>71</td>
</tr>
</tbody>
</table>
Figure 4.13  Results showing Jumps across Mesh Edges in case of an anisotropic mesh

Figure 4.14  Plots of Log (Residual) and Number of Iterations

Figure 4.15  Plots of Log (Residual) with CPU times for GMRES with different Preconditioners

Figure 4.16  Mesh Topologies of Symmetric Stretched and Non Symmetric Stretched Meshes

Figure 4.17  Comparisons between Reconstruction Scheme and Modified Average Gradient Method for test case1

Figure 4.18  Plots of comparison between Reconstruction and Modified Average Gradient Method for test case2
Chapter 1

Introduction

Computational Fluid Dynamics, once a domain of researchers and scholars is fast moving to the hands of engineers and professionals who are responsible for making industrial applications of CFD possible. This rapid transformation is fuelled further with enhancement of the computing power available today. Thus many industries rely heavily on the computer simulation than experimental work. Thus CFD has established itself as a viable tool for engineering analysis and design. As more and more complex flow phenomenon are being explored for complex geometries thus even more sophisticated algorithms are developed which are not only accurate but also computationally and economically efficient. Similarly flow over complex geometries are tackled using unstructured grids instead of structured grids as unstructured grids provide more accurate representation of complex geometries in mesh form.

The mathematical form of the physical flows forms the basis of CFD techniques. These mathematical forms are called the governing equations. The three conservation equations namely, ‘conservation of mass, ‘conservation of momentum’ and ‘conservation of energy’ form the iconic Navier Stokes equations. These equations with equation of state form the basis for almost all of the physical flows. Navier Stokes equations recognize the transport of the flow quantity through convective as well as diffusive fluxes. They also consider the production of the quantity by including source terms. Solving these equations requires the discretization of governing equations determining the flow. The discretization can be said to be a process which transforms the physical equation in a form that can be understood and
solved by the computer. Classical CFD has following techniques for discretization finite difference methods, finite volume methods, finite element methods, spectral methods, finite point methods and discontinuous Galerkin Methods. Out of these methods finite volume methods today form a major chunk of industrial applications of CFD.

Discretizations using finite volume methods are most sought after as they use the integral conservation form of the governing equations. Thus they are not limited by discontinuities unlike finite difference methods. Because of their approach of using a control volume they can be easily used for unstructured and arbitrary grids too. The finite volume discretization of Navier Stokes equations requires an estimation of convective and diffusive fluxes for each element. The convective fluxes have been extensively worked with. There are schemes like upwind, Van Leer Scheme etc for convective fluxes. But the diffusive fluxes have received little attention. One primary reason for it is the estimation of gradients of the flow property at the interfaces of the elements. For structured grids the interfacial gradients are estimated simply by taking an average of the cell center gradients but for unstructured grids this method is not adopted.

1.1 Background and Motivation

There is a set of requirements to be taken care of while developing the schemes. The order of accuracy should be higher than first order even for the unstructured anisotropic and distorted meshes. It should work with heterogeneous full diffusion tensor and discontinuous tensor. It should be able to handle the jumps and discontinuities. It should not violate the discrete maximum principle. It should be grid transparent i.e. it should be independent of the type of
element used whether triangular, polygonal etc. Our scheme has been made to satisfy all these requirements [24, 36, 40]. A detailed numerical analysis is in chapter 4.

Two of the most widely known schemes for estimating gradients are given by Coirer [37, 38] and Haselbacher [1, 2, 3]. Coirer used a diamond path approach which is also used by Vassesn et al where centers of left and right neighbors are connected to each vertex of the cell element considered thus resulting in a diamond like polyhedron. Thus the gauss divergence theorem leads to summation of the fluxes at the triangular faces so formed in the diamond structure [28, 29]. The problem with Coirer’s approach is that it is not compact thus a sparse system of matrix is not formed that can be solved using implicit methods. Similarly the modified average gradient method, introduced by Haselbacher, results in odd-even decoupling thus instability is observed for fine Meshes. This approach is although grid transparent but lacks in compactness. Hermeline presented a finite volume method for discretization of 3D variable diffusion tensor with full tensor coefficients. In his work he has worked with anisotropic tensor. Here the approach suggested by Hermeline is proved to be of first order on distorted meshes [11]. Least squares reconstruction is done by Gooch et al the linear reconstruction gave a second order accuracy while cubic reconstruction gave a fourth order accuracy. But some degeneracies are also found in the reconstruction schemes because of the leading error term of first and third order [5]. Lipnikov suggested the interpolation scheme which is linear in nature. The scheme is found to be satisfying most of the requirements as given above but the extension of the scheme is not possible to Navier Stokes equation is not straightforward [23, 24]. Similarly the approaches using non linear schemes like Mixed Finite Element (MFE) and Multi Point Flux Approximation (MPFA) are also
described by Lipnikov et al. These schemes were found to be violating the Monotonicity Requirement [22]. A reconstruction based discontinuous method is explained by Luo et al for Navier Stokes equations. Least squares reconstruction is used here to extrapolate a smooth solution using inter cell reconstruction method. Here the authors use discontinuous Galerkin Method which are shown to be more efficient, highly parellizable and more accurate. The methods are similar to the finite volume methods with respect to determining the solution at the interfaces of the elements [18]. An approach of using finite volume methods with diffusion problem is demonstrated by Gassner et al. Here the exact solutions of diffusive Riemann problem are used to define the finite volume numerical scheme leading to a space time formulation for convection diffusion equations but this scheme cannot use a piecewise constant values for the diffusion problems as this leads to inconsistency thus at least a piecewise linear is to be taken. The order of convergence is found to be same as the degree of piecewise solution approximated [15].

The discretization of governing equations, leads to formation of a system of equations which needs to be solved for the unknowns. There are many linear equation solver techniques available like conjugate gradient (CG), Generalized Minimum Residual (GMRES). As the jacobian matrix in the present work is not symmetric in nature, hence we cannot use CG, thus GMRES is used here. More details of the GMRES solver are explained in the later chapters. Previous works using GMRES reported an increase in convergence rate while using preconditioned GMRES. Vassesn et al used GMRES solver with a BILU (k) preconditioner. Overall a quadratic reconstruction is used except in the vicinity of shock waves where the use of limiter and detector is compulsory to reduce the order of reconstruction. Accurate
solutions were obtained using the scheme but the computational costs and times were very high compared to the lower order schemes [29]. Another work on implicit integration using GMRES is done by Michalak et.al and Luo et al. A matrix free implementation of GMRES is used and an explicit jacobian matrix is calculated for usage as a preconditioner [6, 17]. Jacobian evaluation is reported in literature as most time taking activity in solving the system. Hence CPR algorithm is used in the present work for numerical jacobian evaluation which uses consistent partition of the columns [4]. This method is very advantageous as it reduces the number of function evaluations drastically. The need for even better convergence of GMRES leads to the use of preconditioners. Incomplete factorization of the preconditioner is found to be more efficient in this case.

1.2 Scope of Thesis

The present work focuses on the approach of gradient estimation at the cell centers and the interfaces of the cell elements. The reconstruction scheme is introduced here for discretization of diffusive fluxes. The reconstruction scheme is detailed more in next chapter. Briefly explaining, it is an interpolation scheme using Taylor’s expansion and solving the system of least squares problem. The discretization scheme leads to formation of a system of equations which is solved using GMRES (Restarted) solver. GMRES was employed with three preconditioners in this work namely diagonal, Lower Upper Symmetric Gauss Seidel (LU-SGS), ILU (Incomplete Lower Upper) preconditioner. The numerical results were compared with exact solutions and thus the scheme was found to be second order accurate.
1.3 **Outline of Thesis**

The thesis is organized in four main chapters. After this chapter of introduction, chapter 2 explains the problem statement and the methodology adopted. The scheme is extensively explained in chapter 2 and solving the system of equations is also explained. Estimation of gradients at the boundary elements and the boundary faces is also discussed. Chapter 3 analyzes theoretically the reconstruction scheme and other existing common schemes for gradient estimation. The analysis is done over uni-directional stretched grids. Chapter 4 discusses numerical experiments conducted using different analytical functions and diffusion tensors. In this chapter other than showing the scheme to be second order, other requirements like monotonicity and handling discontinuities are also discussed. Numerical comparison of the scheme is presented and existing approaches is also done in this chapter. A conclusion is made about the scheme in the light of results obtained from different experiments.
Chapter 2

Governing Equations and Numerical Methods

This chapter discusses the problem statement and the methodology adopted in the present work. The treatment of boundary conditions and the gradient calculation at boundary faces and elements are also discussed here. Finally solving the system of equations implicitly is highlighted.

2.1 Problem Definition

The problem considered in this work is the diffusion equation with Dirichlet boundary condition as follows,

\[ -\nabla \cdot (D \nabla U) = f \quad \text{in} \quad \Omega \]

\[ U = g \quad \text{on} \quad \Gamma = \partial \Omega \]

where, \( U \) is the physical property for which the governing equation is to be solved, \( \Omega \) is a bounded domain in \( \mathbb{R}^2 \) and \( \Gamma \) represents the boundary of the domain where Dirichlet boundary condition is applied. \( F \) is the source term, \( D \) is the diffusion tensor and \( g \) is the boundary value.

2.2 Numerical Methods

2.2.1 Finite Volume Formulation

In finite volume formulation the computational space is divided into a set of computational cells. The integral formulation of the governing equation can be expressed as follows,

\[ -\int_{\Omega_i} \nabla \cdot (D \nabla U) \, d\Omega = \int_{\Omega_i} f \, d\Omega \]
\( \Omega_i \) is the area of the \( i^{th} \) element. The integral of the flux term over the control volume is reduced to the surface integral over the faces or walls of control element using the Green Gauss theorem as in equation 2.3.

\[
-\oint_{\Gamma_i} (D \nabla U) \cdot \vec{n} d\Gamma = \int_{\Omega_i} f d\Omega
\]  

(2.3)

where, \( \Gamma_i \) is the boundary of \( \Omega_i \) control volume and \( \vec{n} \) is the surface normal to the face.

Considering the discretization of only flux term from equation 2.2 and thus dropping the ‘-’ sign for convenience, we have

\[
\int_{\Omega_i} \nabla \cdot (D \nabla U) \ d\Omega = \oint_{\Gamma_i} (D \nabla U) \cdot \vec{n} d\Gamma = \sum_{iface=1}^{nFace} \left( (D \nabla U) \cdot \vec{n} \right)_{iface}
\]  

(2.4)

\( nFace \), represents the number of faces of the element. Thus for carrying out the finite volume discretization of the flux terms we need to calculate the gradients at the interfaces.

The estimation of the gradients at the cell interfaces has been the focus of research in the present work.

### 2.2.2 Common Approaches used for estimating gradients at cell interfaces

Many methods have been developed in the literature for the computation of face gradient. Two most commonly adopted methods are discussed next.

**Method 1: Gradient Computation based on modified averaging**

The simplest approach of estimating gradients at the interfaces is by taking a simple average of the gradients at the cell centers of neighboring elements. Thus,
\[
\n\bar{\nabla} U \bigg|_{\text{face}} = 0.5 \left( \bar{\nabla} U_L + \bar{\nabla} U_R \right) \tag{2.5}
\]

\(\bar{\nabla} U \bigg|_{\text{face}}\) is the averaged gradient. \(\bar{\nabla} U_L\) and \(\bar{\nabla} U_R\) are the gradients of the elements left and right of the interface but this method produces checkboard instability. The solution to this problem is by adopting a gradient in the direction of the vector connecting the cell centers i.e.

\[
\frac{\partial U}{\partial \vec{l}_{LR}} \bigg|_{\text{face}} = \frac{U_R - U_L}{\vec{l}_{LR}} \tag{2.6}
\]

Finally, the average gradient is modified as

\[
\bar{\nabla} U \bigg|_{\text{face}} = \bar{\nabla} U \bigg|_{\text{face}} - \left( \bar{\nabla} U \bigg|_{\text{face}} \cdot \frac{\vec{l}_{LR}}{\vec{l}_{LR}} \right) \frac{\vec{l}_{LR}}{\vec{l}_{LR}} \tag{2.7}
\]

\(\vec{l}_{LR}\) is the vector between the cell centers of right and left cells and \(|\vec{l}_{LR}|\) denotes the distance between the cell centers. Although this method is not compact as it involves neighbors of neighbors, it is grid transparent. This method maintains positivity and it is inconsistent for stretched meshes as will be shown in the analysis in chapter 3 section 3.4. A comparative study in chapter 3 between this scheme and the reconstruction scheme show that the modified average scheme is less accurate for non uniform grids.

**Method 2: Gradient computation based on Gauss Divergence Theorem using Diamond Path**

The method introduced by Coirer is based on Gauss divergence theorem. It makes use of diamond path approach which involves forming a secondary volume by connecting the vertices of an edge with the centroid of the cells sharing that interface as shown in Figure 2.1.
The value of the physical property is evaluated at the vertices using interpolation. The interpolation is supported by a set of elements that share a particular vertex. An improvement is done by more accurate way of finding the values at the vertices. In the improved scheme the reconstruction is identical but uses a linearly preserving weighting function to find the values at the vertices. The weighting makes sure that the data provided to the reconstruction procedure is obtained linearly using the centroidal data. More study on the linearity preserving scheme is done in chapter 3. In terms of weighting the value at the vertex is given as,

\[
U_{\text{vertex}} = \frac{\sum_{n=1}^{N} \omega_n U_n}{\sum_{n=1}^{N} \omega_n}
\]

(2.8)

where, N is the number of cells that share the vertex and \( \omega_n = 1 \) for all n for a simple averaging procedure. With no linearity preserving technique, this scheme is dangerously inconsistent while for linearity preserving techniques at least local grid convergence can be achieved but consistency is not guaranteed although positivity is maintained. Thus another technique, quadratic preserving scheme is used which is consistent and more accurate but less positive.

This approach is not compact as it involves the vertex neighbors of each node. It is also not grid transparent i.e. the scheme depends on the mesh element shape like triangular or polygonal etc.
2.2.3 Reconstruction Scheme

The finite volume formulation of diffusive fluxes requires estimation of gradients at the cell interfaces, which also necessitates the estimation of gradients at the cell centers. Present work concentrates on such a gradient estimation scheme namely Reconstruction Finite Volume (RFV). Least squares reconstruction forms the basis of this scheme. There are two types of reconstruction carried out in this work namely,

a) In-Cell Reconstruction

b) Inter-Cell Reconstruction

a) In-Cell Reconstruction

The gradients at the cell centers are reconstructed using the in-cell reconstruction from the underlying finite volume formulation. Calculation of the gradients at the cell centers is done using Taylor’s expansion for the neighboring cell. Consider an element 1 with the neighboring elements as shown in figure 2.2 namely 1, 2 and 3. We use the normalized form of the Taylor’s expansion. The normalized form gives the better conditioned matrix than non
normalized form, as explained in section 4.2 of chapter 4. Thus normalized form is better conditioned and reduces the stiffness of the matrix. Equation 2.9 to 2.11 represent the normalized form of Taylor’s expansion for \(i^{th}\) cell to the neighboring cells as shown in figure 2.2.

![Figure 2.2 A typical triangular element with neighboring cells](image)

\[
U_1 = U_i + \frac{\partial U}{\partial x} \Delta x_i \frac{(x_1 - x_i)}{\Delta x_i} + \frac{\partial U}{\partial y} \Delta y_i \frac{(y_1 - y_i)}{\Delta y_i} \tag{2.9}
\]

\[
U_2 = U_i + \frac{\partial U}{\partial x} \Delta x_i \frac{(x_2 - x_i)}{\Delta x_i} + \frac{\partial U}{\partial y} \Delta y_i \frac{(y_2 - y_i)}{\Delta y_i} \tag{2.10}
\]

\[
U_3 = U_i + \frac{\partial U}{\partial x} \Delta x_i \frac{(x_3 - x_i)}{\Delta x_i} + \frac{\partial U}{\partial y} \Delta y_i \frac{(y_3 - y_i)}{\Delta y_i} \tag{2.11}
\]

where, \(\Delta x_i = 0.5(x_{\text{max}} - x_{\text{min}})\) and \(\Delta y_i = 0.5(y_{\text{max}} - y_{\text{min}})\).

\((x_{\text{max}}, y_{\text{max}})\) and \((x_{\text{min}}, y_{\text{min}})\) are maximum and minimum coordinates of the cell. Taylor’s expansions for the neighboring elements give an over determined system of equation which is solved using least squares method. Referring to the equations 2.9 to 2.11 we can write the system of equations for any element in the matrix form as,
\[
\begin{bmatrix}
(x_1 - x_i) & (y_1 - y_i) \\
\Delta x_i & \Delta y_i \\
(x_2 - x_i) & (y_2 - y_i) \\
\Delta x_i & \Delta y_i \\
(x_3 - x_i) & (y_3 - y_i) \\
\Delta x_i & \Delta y_i
\end{bmatrix}
\begin{bmatrix}
\frac{\partial U}{\partial x} \\
\partial y
\end{bmatrix}
\begin{bmatrix}
\Delta x_i \\
\Delta y_i
\end{bmatrix} =
\begin{bmatrix}
(U_1 - U_i) \\
(U_2 - U_i) \\
(U_3 - U_i)
\end{bmatrix}
\]

Such a system can be solved by using normal equation approach. The normalized form of coefficient matrix gives a better conditioned system. The condition number is calculated in the chapter 4 and a comparison is also made between normalized and non normalized system.

b) \textbf{Inter-Cell Reconstruction}

The inter-cell reconstruction reconstructs a linear polynomial solution at each interface. In this scheme a union of two neighboring cells is considered as in figure 2.3. A linear polynomial is reconstructed as shown in equation 2.13.

\[
U_{ij} = \overline{U}_{ij} + \frac{\partial U}{\partial x}
\left(x - x_{ij}\right) + \frac{\partial U}{\partial y}
\left(y - y_{ij}\right)
\]

(2.13)

where, \(\overline{U}_{ij}\) is the cell averaged value in the union of two cells, \((x_{ij}, y_{ij})\) are the coordinates of the centroid of the union. The averaged value of the flow parameter at the centroid of the union is given by conservation.

\[
\int_{\Omega_i} U_i d\Omega_i + \int_{\Omega_j} U_j d\Omega_j = \int_{\Omega_{ij}} \overline{U}_{ij} d\Omega_{ij}
\]

(2.14)

\[
\overline{U}_{ij} = \frac{U_i \Omega_i + U_j \Omega_j}{\Omega_i + \Omega_j}
\]

(2.15)
Similarly, the centroid of the union of two cells can be calculated using the area weighted average of coordinates of the cell centers.

\[
x_{ij} = \frac{x_i \Omega_i + x_j \Omega_j}{\Omega_i + \Omega_j}
\]

\[
y_{ij} = \frac{y_i \Omega_i + y_j \Omega_j}{\Omega_i + \Omega_j}
\]

(\(x_i, y_i\)) are the coordinates of the center of cell I and \(\Omega_i\) is the area of the element.

At \((x = x_i \text{ and } y = y_i)\) the function value as given by equation 2.13 is equal to the function value at the center of cell I thus \(U_{ij} = U_i\). Thus for cells \(i\) and \(j\), linear reconstructed polynomials in normalized form can be written as in equation 2.18 and 2.19.

\[
U_i = \frac{U_{ij}}{\partial x_{ij}} \Delta x_{ij} \frac{x_i - x_{ij}}{\Delta x_i} + \frac{\partial U}{\partial y_{ij}} \Delta y_{ij} \frac{y_i - y_{ij}}{\Delta y_{ij}}
\]

(2.18)

\[
U_j = \frac{U_{ij}}{\partial x_{ij}} \Delta x_{ij} \frac{x_j - x_{ij}}{\Delta x_j} + \frac{\partial U}{\partial y_{ij}} \Delta y_{ij} \frac{y_j - y_{ij}}{\Delta y_{ij}}
\]

(2.19)

where,

\[
\Delta x_{ij} = 0.5(x_{\text{max}} - x_{\text{min}}) \text{ and } \Delta y_{ij} = 0.5(y_{\text{max}} - y_{\text{min}})
\]

\(\Delta x_{ij}, \Delta y_{ij}\) are used to normalize the above equations. \((x_{\text{max}}, y_{\text{max}})\) and \((x_{\text{min}}, y_{\text{min}})\) are maximum and minimum coordinates of the union of the two cells.

\[\text{Figure 2.3 Two neighboring cells considered as a union}\]

At the cell centers the gradient is equal to the gradient of the reconstructed polynomial.

\[
\Delta x_{ij} \frac{\partial U}{\partial x}_{ij} = \Delta x_{ij} \frac{\partial U}{\partial x}_i
\]

(2.20)
\[ \Delta y_{ij} \frac{\partial U}{\partial y}_{ij} = \Delta y_{ij} \frac{\partial U}{\partial y}_i \]  
(2.21)

\[ \Delta x_{ij} \frac{\partial U}{\partial x}_{ij} = \Delta x_{ij} \frac{\partial U}{\partial x}_j \]  
(2.22)

\[ \Delta y_{ij} \frac{\partial U}{\partial y}_{ij} = \Delta y_{ij} \frac{\partial U}{\partial y}_j \]  
(2.23)

Thus over determined system of equations is formed. Using the same set of equations given in equations 2.18 to 2.23, we can write the system of linear equations in the matrix form.

\[
\begin{bmatrix}
(x_i - x_{ij}) & (y_i - y_{ij}) \\
\Delta x_{ij} & \Delta y_{ij} \\
1 & 0 \\
(x_j - y_{ij}) & (y_j - y_{ij}) \\
\Delta x_{ij} & \Delta y_{ij} \\
1 & 0
\end{bmatrix}
\begin{bmatrix}
\frac{\partial U}{\partial x}_{ij} \\
\frac{\partial U}{\partial y}_{ij}
\end{bmatrix}
= 
\begin{bmatrix}
(U_i - U_{ij}) \\
\Delta x_{ij} \\
\Delta y_{ij} \\
(U_j - U_{ij}) \\
\Delta x_{ij} \\
\Delta y_{ij}
\end{bmatrix}
\]  
(2.24)

Condition number of the system matrix of least squares problem is calculated in chapter 4 section 4.2. Condition number of the normalized form is found to be better than that of non normalized form.

**Boundary Treatment**

Dirichlet boundary conditions are applied in this work. For the experiments where the exact functions are known, the boundary conditions considered are the exact function values. In the experiments where the exact function is unknown, boundary condition applied is homogeneous boundary condition i.e. the flow property is zero at the boundaries. The values at the ghost cells are approximated using the boundary condition. Boundary face values are
considered as a simple average of values at the boundary element and the ghost element. Thus,

$$U_b = \frac{U_g + U_i}{2}$$

So,

$$U_g = 2U_b - U_i$$ (2.25)

where, $U_b$ is the boundary face value, $U_g$ is the ghost cell value, $U_i$ is the boundary cell value.

More on computing the gradients at the cell center of boundary elements and at the boundary faces is covered in next section of this chapter.

**Calculation of Gradients at the Boundary Elements**

For boundary elements, ghost element is considered as one of the neighboring element. Ghost cell values are determined by utilizing the boundary face values as in equation 2.25. Cell centers of the ghost elements are calculated by assuming that they are the mirror images of the boundary cells. The system of equation formed for the boundary elements is formed as,

$$\begin{bmatrix}
\frac{(x_1 - x_i)}{\Delta x_i} & \frac{(y_1 - y_i)}{\Delta y_i} \\
\frac{(x_2 - x_i)}{\Delta x_i} & \frac{(y_2 - y_i)}{\Delta y_i} \\
\frac{(x_g - x_i)}{\Delta x_i} & \frac{(y_g - y_i)}{\Delta y_i}
\end{bmatrix}
\begin{bmatrix}
\frac{\partial U}{\partial x_i} \\
\frac{\partial U}{\partial y_i}
\end{bmatrix}
= \begin{bmatrix}
(U_1 - U_i) \\
(U_2 - U_i) \\
(U_g - U_i)
\end{bmatrix}$$ (2.26)

where, $\Delta x_i = 0.5(x_{\text{max}} - x_{\text{min}})$ and $\Delta y_i = 0.5(y_{\text{max}} - y_{\text{min}})$. $(x_{\text{max}}, y_{\text{max}})$ is the maximum coordinate of the cell and $(x_{\text{min}}, y_{\text{min}})$ is the minimum coordinate of the cell, $x_g$ and $y_g$ are the coordinates of the cell center of the ghost cells, $U_g$ is the quantity of the flow
parameter in the ghost cell. The cell center gradients are used in the inter cell reconstruction for calculating the gradients at the cell interfaces.

**Calculation of gradients at the boundary faces**

For boundary faces, the gradient at the faces is estimated using Taylor’s expansion. The function value at the boundary face is extrapolated to the cell center of the boundary element. The Taylor’s expansion as done previously for interior faces is repeated here for boundary faces.

\[
U_i = U_b + \frac{\partial U}{\partial x_{ib}} (x_i - x_{ib}) + \frac{\partial U}{\partial y_{ib}} (y_i - y_{ib})
\]  

\[(2.27)\]

\((x_{ib}, y_{ib})\) is the coordinate of the face center of boundary face. \(U_b\) is the value of the function at the boundary given by the boundary conditions. \((\frac{\partial U}{\partial x}_{ib}, \frac{\partial U}{\partial y}_{ib})\) are the gradients at the boundary faces. Similar to the interior faces, the gradient of the linear reconstructed polynomial at the cell center is equal to the gradient at the cell center of the boundary element.

\[
(\Delta x_i) \frac{\partial U}{\partial x_{ib}} = \frac{\partial U}{\partial x_i} (\Delta x_i)
\]  

\[(2.28)\]

\[
(\Delta y_i) \frac{\partial U}{\partial y_{ib}} = \frac{\partial U}{\partial y_i} (\Delta y_i)
\]  

\[(2.29)\]

Above set of equations can be written as a system of equations \(Ax = B\), in Matrix form as,

\[
\begin{bmatrix}
\frac{x_i - x_{face}}{(\Delta x_i)} & \frac{y_i - y_{face}}{(\Delta y_i)} \\
(\Delta x_i) & (\Delta y_i)
\end{bmatrix}
\begin{bmatrix}
\frac{\partial U}{\partial x_{ib}} \\
\frac{\partial U}{\partial y_{ib}}
\end{bmatrix}
= \begin{bmatrix}
\frac{U_i - U_b}{\Delta x_i} \\
\frac{U_i - U_b}{\Delta y_i}
\end{bmatrix}
\]  

\[(2.30)\]
Above least squares problem can be solved using normal equation approach. The normal component of the gradients at the cell interfaces calculated above, are used in computing the diffusive fluxes through the interface.

2.3 Solving System of Equations

The finite volume discretization in equation 2.3 leads us to a system of equations as $Ax = B$. Here $A$ is the coefficient matrix, $x$ is the unknown vector and $B$ is the right hand side vector. In this work the Restarted Generalized Minimum Residual Method (GMRES) is used to solve the linear system as the coefficient matrix is non symmetric in nature.

2.3.1 Generalized Minimum Residual (GMRES)

The underlying concept of this methodology for solving a system of equation is that each iteration can be expressed as sum of initial guess and a linear combination of $A_i r(x^0)$ where $r(x^0)$ is the initial residual. $A_i r(x^0)$ are called Krylov space vectors. The convergence is further fastened by employing the restarted GMRES where solution after certain number of Krylov iterations forms the initial solutions for next restarted GMRES solver. For any $m$th iteration the solution is given as

$$x^m = x^0 + c_0 r^0 + c_1 Ar^0 \ldots \ldots \ldots c_{m-1}A^{m-1}r^0$$  \hspace{1cm} (2.31)

The basis of this method lies in minimizing the residual $R(x^m) = r(x^m)^T r(x^m)$. Here we do not explicitly calculate the coefficient matrix. This saves a lot of memory and computation which makes GMRES a very effective method.
2.3.2 Implementation of GMRES

GMRES solver requires the residual, given as $Ax - B$. In our problem, there is no calculation to obtain the coefficient matrix ($A$) explicitly. Hence the residual is directly calculated using the diffusive fluxes. The net diffusive fluxes through any element are the summation of the normal components of the gradients at the cell interfaces. As each face is shared by exactly two elements, thus to avoid calculating the flux for any element twice, a loop over the faces is considered. The flux is calculated over the face as the product of gradient at the face and the normal to the face. For every face the flux is added for the left neighbor and it is subtracted for right neighbor.

![Figure 2.4 A Union of Two neighboring cells](image)

Consider that our face structure is such that the element 1 is the left and 2 is the right element. Thus while writing the net flux through each element,

$$flux(1) = flux(1) + (\nabla U \cdot \vec{n})_{interface}$$

(2.32)

$$flux(2) = flux(2) - (\nabla U \cdot \vec{n})_{interface}$$

(2.33)

where, $flux(1)$ and $flux(2)$ are the fluxes through the elements 1 and 2.

The procedure is adopted for all the faces of these triangles. The net value of the fluxes for every element is obtained. The net diffusive fluxes after including the source term also, through an element $I$ is given as equation 2.34,
\[ f_{lux}(i) = \sum_{\text{iface} = 1}^{n_{\text{face}}} \left((D\vec{v}U).\vec{n}\right)_{\text{iface}} + f\Omega_i \]  

(2.34)

For the purpose of increasing the efficiency, GMRES solver is also employed along with preconditioner. Three different types of preconditioner are tested in this work namely 1) Diagonal Preconditioner, 2) LU-SGS Preconditioner and 3) ILU Preconditioner.

### 2.3.3 Preconditioner

A preconditioner is a matrix which is used to faster the convergence rate. Consider \( P \) as a preconditioner matrix for a system of equations. The matrix is chosen such that \( P^{-1}Ax = P^{-1}B \) is better conditioned than \( Ax = B \). Two of the requirements for a preconditioner desired are:

a) It should be easily invertible,

b) The preconditioner matrix should be as close to \( A \) as possible i.e. the spectral radius of \( I - P^{-1}A \) should be as small as possible where \( I \) is the identity matrix.

Broadly, the preconditioner is employed in two different ways,

a) **Left Preconditioning**

The precondition matrix is left multiplied to the system of equations as,

\[ P^{-1}Ax = P^{-1}B \]

b) **Right Preconditioning**

The precondition matrix is right multiplied to the system of equation as,

\[ AP^{-1}P = B \]
2.3.4 Jacobian Evaluation

The preconditioner in this work is obtained from the coefficient matrix $A$ of the system $Ax=B$. The matrix $A$ is obtained using the jacobian evaluation. The residual for the GMRES solver is written as follows,

$$R_1 = a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n - b$$
$$R_2 = a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n - b$$
till,

$$R_n = a_{n1}x_1 + a_{n2}x_1 + \cdots + a_{nn}x_n - b$$

where, $R_i$ is the residual for the $i^{th}$ element, $x_i$ is the unknown value for $i^{th}$ element, $a_{ij}$ is $(i,j)$ element of the coefficient matrix $A$ to be determined.

Since the coefficient matrix is not known explicitly, hence it is derived from the analytical function for residual as given above. The jacobian matrix evaluated analytically is given in equation 2.35.

$$J = \begin{bmatrix}
\frac{\partial R_1}{\partial x_1} & \frac{\partial R_1}{\partial x_2} & \frac{\partial R_1}{\partial x_3} & \cdots & \frac{\partial R_1}{\partial x_n} \\
\frac{\partial R_2}{\partial x_1} & \frac{\partial R_2}{\partial x_2} & \frac{\partial R_2}{\partial x_3} & \cdots & \frac{\partial R_2}{\partial x_n} \\
\frac{\partial R_3}{\partial x_1} & \frac{\partial R_3}{\partial x_2} & \frac{\partial R_3}{\partial x_3} & \cdots & \frac{\partial R_3}{\partial x_n} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\frac{\partial R_n}{\partial x_1} & \frac{\partial R_n}{\partial x_2} & \frac{\partial R_n}{\partial x_3} & \cdots & \frac{\partial R_n}{\partial x_n}
\end{bmatrix}$$

The determination of the above matrix is quite complicated. Hence we calculate the numerical jacobian using finite difference method. The numerical jacobian is given as in equation 2.36 where $i$ denotes $i^{th}$ component of diffusive flux vectors and $k$ denotes the $k^{th}$ component of the dependent variable. A perturbation magnitude is selected labeled as $\epsilon$. A
comfortable assumption of perturbation magnitude is the square root of machine zero which
is nothing but the smallest number a machine can compute as per Onur and Eyi [28].
Machine zero can be easily determined by the system using ifort compiler.

\[ \varepsilon = \sqrt{\omega} \]

where, \( \omega \) is the machine zero. The value of the machine zero comes out to be \( 1 \times 10^{-16} \) in
our case. The step size can be taken larger if the calculations involve too many floating point
operations. The gradients are calculated as,

\[ j_{ik} = \frac{\partial f_i}{\partial x_k}, \text{such that} \ 1 \leq i \leq n \ \text{and} \ 1 \leq k \leq n \]

\[ \frac{\partial f_i}{\partial x_k} = \frac{f_i(x_k + \varepsilon) - f_i(x_k)}{\varepsilon} \]  

(2.36)

As given in Onur and Eyi [28], another approximation for \( \varepsilon \) is

\[ \varepsilon = \frac{1}{2^m} \text{ such that } 1 + \varepsilon > 1 \]

where, \( m \) is defined in the literature, as the number of highest possible bits in the binary
representation of mantissa. The jacobian matrix evaluated as explained above represents a
very time taking procedure for a fine mesh. In this method the function evaluation \((Ax - B)\)
is done \((n + 1)\) times, where \( n \) is the number of elements. For fine meshes the number of
cell elements can be as large as millions thus the above approach becomes very complicated.

In order to make the jacobian evaluation cost effective an algorithm is adopted. This method
is called CPR algorithm after Curtis Powell and Reid [4]. The concept of consistent partition
is used for the sparse jacobian matrix. Similar approach is also followed by YPMA [35]. As
previously explained in initial chapters, the flux discretization for any cell depend on the ten
point stencil i.e. the face neighbors and the neighbors of neighbors of the cell. Thus we assume here that the sparse matrix contains non-zero elements corresponding to these nine neighbors and one for self and all other entries for any column should be zero. As largely the coefficient matrix is sparse, thus we need to solve only for non-zero elements of the matrix. Thus a matrix $S$ is formed such that,

$$s_{ij} = \begin{cases} 1 & \text{if } i \text{ is a neighbour or neighbour of neighbour to } j \\ 0 & \text{otherwise} \end{cases}$$

The above matrix has 1 at the position of non zero element in the coefficient matrix and 0 at the position of zero elements of coefficient matrix. Corresponding to the consistent partition in matrix $s$, non-zero elements of the coefficient matrix is collected in groups of columns. They are created such that two important criterions are satisfied. 1) One column should be associated with one and only one group. 2) No two or more columns in a group should have non zero entries at the same row position. Thus the group matrix consists of largely non-zero elements. This way a jacobian matrix of $n$ columns is reduced to $m$ columns of group matrix. Thus the function evaluation is reduced from ($n+1$) to ($m+1$). This greatly simplifies the jacobian evaluation and takes considerably less time. The algorithm can be explained easily by taking the following example. Consider a sparse jacobian matrix as given in equation 2.35. Consider $C_{1-6}$ as columns 1 to 6 and $R_{1-6}$ represent the rows 1 to 6 in the jacobian equation 2.37. The consistent partition as described earlier group the columns in a way that it satisfies the requirements of the consistent partition as described.
Thus the group matrix so formed is,

\[
\begin{bmatrix}
A_{11} & A_{12} & A_{13} & 0 & 0 & 0 \\
0 & A_{22} & A_{23} & A_{24} & 0 & 0 \\
0 & 0 & A_{33} & A_{34} & A_{35} & 0 \\
0 & 0 & 0 & A_{44} & A_{45} & A_{46} \\
0 & 0 & 0 & 0 & A_{55} & A_{56} \\
0 & 0 & 0 & 0 & 0 & A_{66}
\end{bmatrix}
\]

\[\text{Group } 1 = C_1 + C_4 ; \; \text{Group } 2 = C_2 + C_5 ; \; \text{Group } 3 = C_3 + C_6\]

Thus n columns of the jacobian are reduced to m groups using CPR Algorithm as shown in equation 2.38. There can be further compact reduction for meshes having sparser matrix. With the reduction in the number of columns, the subroutine call for the evaluation of function \((Ax - B)\) is reduced from \((\text{no. of Elements} + 1)\) to \((\text{no. of Groups} + 1)\). This saves a lot of computational time. For finer meshes the reduction is more prominent. Table 2.1 provides the result of CPR algorithm, showing the reduction of number of columns to number of groups.
### Table 2.1 Number of Columns and groups formed after CPR Algorithm

<table>
<thead>
<tr>
<th>Mesh</th>
<th>Number of Columns</th>
<th>Number of Groups</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 X 4</td>
<td>32</td>
<td>15</td>
</tr>
<tr>
<td>8 X 8</td>
<td>128</td>
<td>17</td>
</tr>
<tr>
<td>16 X 16</td>
<td>512</td>
<td>19</td>
</tr>
<tr>
<td>32 X 32</td>
<td>2048</td>
<td>17</td>
</tr>
</tbody>
</table>

#### 2.3.5 Diagonal Preconditioner

Diagonal preconditioner or jacobi preconditioner is the easiest one to employ. The diagonal elements of the coefficient matrix serve as the preconditioner matrix. The inverse of the diagonal matrix is used to obtain required preconditioning. Thus

\[ P = a_{ii} \]  \hspace{1cm} (2.39)

#### 2.3.6 LU-SGS Preconditioner

In Lower Upper Symmetric Gauss Seidel (LU-SGS) Preconditioner,

\[ P = (D + L)D^{-1}(D + U) \]  \hspace{1cm} (2.40)

where, D is the diagonal matrix, L is the lower triangular matrix and U is the upper triangular matrix.
2.3.7 **ILU Preconditioner**

ILU Preconditioner is the most efficient preconditioner considered here. It is also most expensive to employ. The algorithm for implementing this preconditioner is given as suggested by Kim, Yun [31].

\[ A_0 = A \]

for \( k = 1, n - 1 \)

\[ N_k = \left( n_{ij}^k \right), \text{where } n_{ij}^k = -a_{ij}^k \text{ if } (k,j) \in P \text{ for all } j \text{ such that } j > k \]

\[ n_{ij}^k = -a_{ij}^k \text{ if } (i,k) \in P \text{ for all } i \text{ such that } i < k \]

\[ n_{ij}^k = 0, \text{ Otherwise} \]

\[ \tilde{A}_k = \left( \bar{a}_{ij}^k \right) = A_{k-1} + N_k \]

\( L_k = \left( l_{ij}^k \right) \text{ is an elementory lower triangular matrix} \)

where \( \left( l_{ij}^k \right) = -\bar{a}_{ik}^k / \bar{a}_{kk}^k \)

\[ A_k = L_k \tilde{A}_k \]

\[ \bar{L} = (L_{n-1}L_{n-2} \ldots \ldots L_2L_1)^{-1} \]

\[ \bar{U} = A_{n-1} \]

\[ N = \sum_{k=1}^{n-1} N_k \]

This is the ILU decomposition of coefficient matrix A. The preconditioner matrix is given as product of \( \bar{L} \) and \( \bar{U} \).

\[ P = \bar{L}\bar{U} \]  \hspace{1cm} (2.41)
Although jacobian evaluation is required for both LU-SGS and ILU preconditioner, but still they are proved to be the fastest preconditioners considered as will be seen in chapter 4.
Chapter 3

An analysis of Existing Viscous Flux Formulae

Two existing cell centered finite volume schemes for viscous fluxes are analyzed in this chapter on uni-directional stretched grids. The existing schemes are compared with reconstruction scheme.

3.1 Model grid Topology and Formulae

The grid topology used for the analysis in this chapter is uni-directional stretched grid as shown in figure 3.1.

\[
\delta = \frac{h}{\Delta y_0}
\]

Figure 3.1 A Uni-Directional Stretched Grid

For the uni-directional stretched grid, a non-unity aspect ratio is chosen such that,
\[
\Delta y_N \Delta y_0 = \Delta y_s = \beta
\]

\(\beta\) is the stretching ratio and \(\delta\) is the aspect ratio.

The finite volume formulation of the laplace equation for an element \(I\) with volume \(\Omega_i\) give us,

\[
\int_{\Omega_i} \nabla^2 U d\Omega = \int_{\Gamma} \nabla U \cdot \hat{n} d\Gamma = \nabla U|_1 \cdot \hat{n}A_1 + \nabla U|_2 \cdot \hat{n}A_2 + \nabla U|_3 \cdot \hat{n}A_3 + \nabla U|_4 \cdot \hat{n}A_4
\] (3.1)

For the above topology considered, the laplace operator reduces to the following form.

\[
L(U_0) = \frac{1}{h} \left( U_{x,E} - U_{x,W} + \delta (U_{y,N} - U_{y,S}) \right) \] (3.2)

where, \(U_{x,E}, U_{x,W}, U_{y,N}\) and \(U_{y,S}\) are the gradients at the faces 1, 2, 3 and 4 respectively as shown in figure 3.1. The calculation of the face gradients poses a major challenge today. Two approaches, introduced in chapter 2 are also discussed and analyzed here. The analysis of the scheme includes analysis on Discrete Maximum Principle (DMP), consistency and accuracy of the scheme.

### 3.2 Discrete Maximum Principle

Discrete Maximum Principle states that maximum of the function occur at the boundaries. Locally this means that the value at a point is bounded by the solution in the neighborhood of the point. An N point stencil is arrived at for the laplace operator as given in equation 3.3.

\[
\nabla^2 U = L(U) = \sum_{n=0}^{N} \alpha_n U_n \] (3.3)
N is the total number of points. If a stencil could be found such that the coefficients $\alpha_n$ are positive, then the scheme satisfies the discrete maximum principle. Locally, above equation can be solved for $U_0$ as follows,

$$U_0 = \sum_{n=1}^{N} \omega_n U_n$$  \hspace{1cm} (3.4)

where, $\omega_n = -\frac{\alpha_n}{\alpha_0}$. To be bounded by neighbors the value $U_0$ must satisfy following condition,

$$\min(U_1, U_2, ...) < U_0 < \max(U_1, U_2, ...)$$  \hspace{1cm} (3.5)

The condition in equation 3.5 is satisfied if $\omega_n \geq 0$. Thus if in the stencil all the coefficients are positive, then it satisfies Discrete Maximum Principle.

### 3.3 Accuracy

Consider the Laplace operator again, writing the Taylor’s expansion and collecting the partial derivative terms as in equation 3.6.

$$L(U) = \sum_n (\alpha_n) + \left( \sum_n \alpha_n \zeta_n \frac{\partial U}{\partial x} \right) + \left( \sum_n \alpha_n \eta_n \frac{\partial U}{\partial y} \right) + \frac{\sum_n \alpha_n \zeta_n^2}{2} \frac{\partial^2 U}{\partial x^2} + \frac{\sum_n \alpha_n \eta_n^2}{2} \frac{\partial^2 U}{\partial y^2} + \ldots$$  \hspace{1cm} (3.6)

where, $\zeta_n = (x_n - x_0) \text{ and } \eta_n = (y_n - y_0)$.

In the above equation consider the following equations,

$$\sum_n (\alpha_n) = 0 \hspace{2cm} (3.7) \hspace{2cm} \sum_n \alpha_n \zeta_n = 0 \hspace{2cm} (3.8)$$

$$\sum_n \alpha_n \eta_n = 0 \hspace{2cm} (3.9) \hspace{2cm} \sum_n \alpha_n \zeta_n^2 = 2 \hspace{2cm} (3.10)$$
Equations 3.7 to 3.12 ensure a first order accurate scheme and equations 3.13 to 3.16 ensure second order accuracy. Thus a laplace operator can be written as,

\[ L(U) = \nabla^2 U + O(h^2) \]  

\( O(h^2) \) denotes the higher order terms. As the higher order are of the order of \( h^2 \), thus it has a second order accuracy.

### 3.4 Consistency

A stencil for laplace is consistent if it can be expressed as,

\[ L(U) = (\alpha_1 U_x + \alpha_2 U_y) + k_1 U_{xx} + k_2 U_{xy} + k_3 U_{yy} + O(h^2) \]  

Such that if \( \alpha_1 \neq 0 \) and \( \alpha_2 \neq 0 \), then the scheme is dangerously inconsistent. In such a case the grid convergence even to a wrong equation can never be achieved. A consistent scheme for laplace operator is obtained if \( k_1 = k_2 \) and \( \alpha_1 = \alpha_2 = 0 \).

### 3.5 Common Approaches for estimation of Gradients at Cell Interface

The laplace equation is solved in finite volume using the green gauss theorem. This requires us to estimate the gradients at the cell interfaces. There are two common approaches for estimating the gradients at the edges. These are discussed next and analyzed. The scheme developed in this work is also analyzed here.
3.5.1 Modified Average Gradient Method

The averaging of the neighboring cell center gradients for the gradient at the interface is the simplest approach. It leads to checkerboard instability due to odd even decoupling. If this instability is somewhat stabilized using truncation error at boundaries, it will result into non smooth solution. The solution to this problem is adopting a gradient in the direction of the vector connecting the cell centers i.e,

$$\frac{\partial U}{\partial l_{LR}}|_{face} = \frac{U_R - U_L}{l_{LR}}$$ \hspace{1cm} (3.19)

Finally, the gradient is computed as

$$\overline{\nabla U}|_{face} = \overline{\nabla U}|_{face} - \left(\overline{\nabla U}|_{face} \cdot \frac{l_{LR}}{|l_{LR}|} \right) \frac{l_{LR}}{|l_{LR}|}$$ \hspace{1cm} (3.20)

$\overline{\nabla U}|_{face}$ is the averaged gradient, given as an average of the cell center gradients of the neighboring cells. $\overline{\nabla U}|_{face} = 0.5(\overline{\nabla U}_L + \overline{\nabla U}_R)$. Here $\overline{\nabla U}_L$ and $\overline{\nabla U}_R$ are the gradients of the elements left and right of the interface.

This method is analyzed for a uni-directional stretched mesh as in figure 3.1. For a structured mesh, $\frac{l_{LR}}{|l_{LR}|}$ is a unit vector. Consider a face 1 in the figure 3.1, the gradients at this face using modified average approach is given as,

$$\left. \frac{dU}{dx} \right|_1 \hat{i} + \left. \frac{dU}{dy} \right|_1 \hat{j} = \frac{(U_1 - U_i)}{\sqrt{(x_i - x_1)^2 + (y_i - y_1)^2}} \left(\frac{x_i - x_1}{\sqrt{(x_i - x_1)^2 + (y_i - y_1)^2}} \hat{i} + \frac{y_i - y_1}{\sqrt{(x_i - x_1)^2 + (y_i - y_1)^2}} \hat{j} \right)$$ \hspace{1cm} (3.21)

Similar expressions are written for the gradients of all the interfaces of the cell. Following equations give the gradients at the cell interfaces for element I in figure 3.1.
\[
\frac{dU}{dx}_1 = \frac{(U_i - U_1)}{(x_i - x_1)} = \frac{(U_i - U_1)}{h} 
\]
(3.22)

\[
\frac{dU}{dy}_1 = 0
\]
(3.23)

\[
\frac{dU}{dx}_3 = \frac{(U_3 - U_i)}{(x_3 - x_i)} = \frac{(U_i - U_3)}{h} 
\]
(3.24)

\[
\frac{dU}{dy}_3 = 0
\]
(3.25)

The gradients at the upper and lower faces are computed as,

\[
\frac{dU}{dx}_2 = 0 
\]
(3.26)

\[
\frac{dU}{dy}_2 = \frac{(U_2 - U_i)}{(y_2 - y_i)} = \frac{2(U_2 - U_i)}{\Delta y_i (1 + \beta)}
\]
(3.27)

\[
\frac{dU}{dx}_4 = 0 
\]
(3.28)

\[
\frac{dU}{dy}_4 = \frac{(U_4 - U_i)}{(y_4 - y_i)} = \frac{2(U_2 - U_i)\beta}{\Delta y_i (1 + \beta)} 
\]
(3.29)

Substituting the expressions for the interface gradients i.e. equations 3.22 to 3.29 in the equation 3.2 for the laplace operator and writing laplace equation in discretized form by collecting coefficients as,

\[
L(U) = U_W + U_E + \alpha_0 U_0 + \alpha_N U_N + \alpha_S U_S 
\]
(3.30)

We obtain following stencil as in figure 3.2,

\[
\alpha_0 = -2(1 + \delta^2)
\]
(3.31)

\[
\alpha_N = \frac{2\delta^2}{(1 + \beta)}
\]

\[
\alpha_S = \beta \alpha_N
\]
The consistency of the scheme is checked by substituting face gradients from equation 3.22 to 3.29 into equation 3.2 and Taylor’s expressions are also substituted for neighboring cells, thus we obtain the laplace equation as,

\[ L(U) = \frac{1}{h^2} \begin{array}{ccc} 0 & \alpha_N & 0 \\ 1 & \alpha_0 & 1 \\ 0 & \alpha_S & 0 \end{array} \]

Comparing the above equation with equation 3.18 we can write \( \alpha_1 = \alpha_2 = 0 \). Thus the scheme is not dangerously inconsistent. Also \( k_1 \neq k_2 \) for \( \beta \neq 1 \) thus it shows that the scheme is inconsistent for uni-directionally stretched mesh.

### 3.5.2 Green Gauss Reconstruction using Diamond Path Approach

Another approach for gradient calculation at the interface is presented in which a diamond path is created by connecting the vertices of the face with the centroid of the two neighboring cells. An interpolation is done for estimating the values at the vertices using the centroidal values. For this a simple approach of taking the average of the values of the neighboring cell centroid is taken.
Figure 3.3 Simple Averaging Procedures at Subtended Vertex.

Thus at vertex R, \( U_R = 1/4(U_N + U_0 + U_{NE} + U_{SE}) \), and for vertex L the value is given as, \( U_L = 1/4(U_N + U_0 + U_{NW} + U_{SW}) \). An improvement in the scheme is observed by using the linearly preserving weighting to find the values at the vertices. The value of the physical property is evaluated at the vertices using interpolation. The interpolation is supported by a set of elements that share a particular vertex. The value of the flow property at the vertex \( (v_i) \) is given as,

\[
\bar{U}_{v_i} = \frac{\sum_{j=1}^{N_{v_i}} \zeta_j U_j}{\sum_{j=1}^{N_{v_i}} \zeta_j}
\]  

where \( U_j \) represents the value at the cell \( \Omega_j \), \( N_{v_i} \) is the number of neighbors surrounding vertex \( v_i \). \( \zeta_j \) are the dimensionless weights given as,

\[
\zeta_j = 1 + \lambda_x (x_j - x_{v_i}) + \lambda_y (y_j - y_{v_i})
\]

\[
\lambda_x = \frac{I_{xy} R_y - I_{yy} R_x}{I_{xy} I_{yy} - I_{xy}^2}; \quad \lambda_y = \frac{I_{xy} R_y - I_{xx} R_x}{I_{xy} I_{yy} - I_{xy}^2}
\]

And the moments \( R_x, R_y, I_{xx}, I_{xy} \) and \( I_{yy} \) are defined as follows,
The laplace operator for a single vertex is expressed as,

\[ L(U_0) = \sum_{i} \zeta_i (U_i - U_0) = 0 \]

Using the above relation, equation 3.33 can be easily obtained for computing the value at the vertex. The gradients at the faces are found using following equation,

\[ U_x = \frac{1}{\Omega} \left( \frac{(n_{1x} + n_{2x})}{2} U_r + \frac{(n_{2x} + n_{3x})}{2} U_t + \frac{(n_{3x} + n_{4x})}{2} U_b + \frac{(n_{4x} + n_{1x})}{2} U_R \right) \] (3.35)

where, \( n_{1,x}, n_{1}, n_{1,x} \) and \( n_{1,x} \) are normal as shown in figure 3.4.

**Figure 3.4 Sample Reconstruction Diamond Path**

For uniform grids, equation 3.35 reduces to a simple central differencing equation making the scheme consistent due to fortunate geometric cancellations but for uni-directionally
stretched grids it’s not so straight forward. The interfacial gradients are substituted in the
discretized laplace equation 3.2 and the laplace equation is written as in equation 3.30 by
collecting coefficients, which gives the stencil as in figure 3.5,

\[ L(U) = \frac{1}{h^2} \]

\[
\begin{array}{ccc}
0 & \alpha_N & 0 \\
1 & \alpha_0 & 1 \\
0 & \alpha_s & 0
\end{array}
\]

**Figure 3.5 Stencil for Diamond Path Reconstruction on Uni-Directional Stretched Grids**

\[
\alpha_0 = -2(1 + \delta^2)
\]

\[
\alpha_N = \frac{2\delta^2}{(1 + \beta)}
\]

\[
\alpha_s = \beta \alpha_N
\]

Substituting the above coefficients in N point stencil, we obtain the expression for laplace
operator as,

\[
L(U) = \nabla^2 U = U_{xx} + \frac{(1 + \beta)^2}{4\beta} U_{yy} + \frac{h(1 + \beta)^3(\beta - 1)}{24\delta^2 \beta} U_{yyy} + \cdots \quad (3.36)
\]

From the equation 3.36, consistency is obtained only if \( \beta = 1 \) i.e for a uniform grid. For other
values of \( \beta \), i.e. for any uni directional stretched grid the scheme is not consistent.
Comparing equation 3.36 and equation 3.18 can see that $\alpha_1 = \alpha_2 = 0$, thus the scheme is not dangerously inconsistent.

Both the approaches discussed above have shown identical behavior. They are consistent for uniform grids but for uni-directional stretched grids both the schemes display inconsistency.

The diamond path approach is not grid transparent and also not compact as it involves vertex neighbors of each element. A comparative study done in the next section, shows that the order of accuracy of the modified gradient approach is even lesser than second order for non uniform and non regular grids.

Diamond path without linear preserving, is dangerously inconsistent while with linearly preserving at least local grid convergence can be achieved but consistency is not guaranteed although positivity is maintained. Hence next discussed is a technique which is second order accurate, consistent for uniform meshes, grid transparent and doesn’t violate DMP.

### 3.6 Reconstruction Scheme

In the work presented here reconstruction scheme is used to estimate the gradients at the cell interfaces. This requires calculating the gradients at the cell centers of the neighboring cells.

An analysis is presented here of the reconstruction scheme on 1D non uniform grids.

![Figure 3.6 1D Uni-Directionally Stretched Mesh](image)

Above grid is chosen such that it is a uni-directional stretched grid. Thus for $\beta$ as the stretching ratio,
\[
\frac{\Delta x_{i+1}}{\Delta x_i} = \frac{\Delta x_i}{\Delta x_{i-1}} = \beta
\]  

(3.37)

Gradients at the cell centers are given by the extrapolation of the cell center values to the neighboring cells.

\[
U_{i+1} = U_i + \left. \frac{du}{dx} \right|_{i} (x_{i+1} - x_i)
\]  

(3.38)

\[
U_{i-1} = U_i + \left. \frac{du}{dx} \right|_{i} (x_{i-1} - x_i)
\]  

(3.39)

From Figure 3.6,

\[
(x_{i+1} - x_i) = 0.5(\Delta x_i + \Delta x_{i+1}) = \frac{(\Delta x_i / 2)(1 + \beta)}{\text{and}}
\]

\[
(x_{i-1} - x_i) = 0.5(\Delta x_{i-1} + \Delta x_i) = -\left(\frac{\Delta x_i}{2}\right)\left(\frac{1}{\beta} + 1\right)
\]

Above least squares problem is solved only for one variable thus giving the cell center gradient as,

\[
\left. \frac{dU}{dx} \right|_i = \frac{2\beta (U_{i+1} - U_i) - (U_{i-1} - U_i))}{\Delta x_i(1 + \beta)(1 + \beta^2)}
\]  

(3.40)

Writing, similar equations for other cells and computing the cell center gradients,

\[
\left. \frac{dU}{dx} \right|_{i+1} = \frac{2\beta (U_{i+2} - U_{i+1}) - (U_i - U_{i+1}))}{\Delta x_i(1 + \beta)(1 + \beta^2)}
\]  

(3.41)

\[
\left. \frac{dU}{dx} \right|_{i-1} = \frac{2\beta^2 (U_i - U_{i-1}) - (U_{i-2} - U_{i-1}))}{\Delta x_i(1 + \beta)(1 + \beta^2)}
\]  

(3.42)

Gradients at the interface are obtained by considering a union of two neighboring cells.

Using the conservation the value of the physical property can be written as,
\[
\int_{\Omega_i} U_{i+1} d\Omega_i = \int_{\Omega_i} U_i d\Omega_i + \int_{\Omega_{i+1}} U_{i+1} d\Omega_{i+1}
\]  \hspace{1cm} (3.43)

\[
U_{i+1} = \frac{U_i \Omega_i + U_{i+1} \Omega_{i+1}}{\Omega_i + \Omega_{i+1}} = \frac{U_i \Delta x_i + U_{i+1} \Delta x_{i+1}}{\Delta x_i + \Delta x_{i+1}} = \frac{U_i + U_{i+1} \beta}{1 + \beta}
\]  \hspace{1cm} (3.44)

\[
x_{i+1} = \frac{x_i \Omega_i + x_{i+1} \Omega_{i+1}}{\Omega_i + \Omega_{i+1}} = \frac{x_i \Delta x_i + x_{i+1} \Delta x_{i+1}}{\Delta x_i + \Delta x_{i+1}} = \frac{x_i + x_{i+1} \beta}{1 + \beta}
\]  \hspace{1cm} (3.45)

Extrapolation of the centroidal value of the union to the cell center values of the neighboring cells is done as following equations,

\[
U_i = U_{i+1} + \left. \frac{dU}{dx} \right|_{i+1} (x_i - x_{i+1})
\]  \hspace{1cm} (3.46)

\[
\Delta x_i \left. \frac{dU}{dx} \right|_{i+1} = \Delta x_i \left. \frac{dU}{dx} \right|_i
\]  \hspace{1cm} (3.47)

\[
U_{i+1} = U_{i+1} + \left. \frac{dU}{dx} \right|_{i+1} (x_{i+1} - x_{i+1})
\]  \hspace{1cm} (3.48)

\[
\Delta x_{i+1} \left. \frac{dU}{dx} \right|_{i+1} = \Delta x_{i+1} \left. \frac{dU}{dx} \right|_{i+1}
\]  \hspace{1cm} (3.49)
Thus the over-determined system is solved for just one variable. Thus the gradient at the interface is obtained as in equation 3.50 and equation 3.51.

\[
\frac{dU}{dx}_{i+1} = \frac{2(U_{i+1} - U_i)}{5(1 + \beta)\Delta x_i} + \frac{8\beta(U_{i+1} - U_i) - (U_{i-1} - U_i)}{5(1 + \beta)(1 + \beta^2)\Delta x_i} + \frac{8\beta^2(U_{i+2} - U_{i+1}) - (U_i - U_{i+1})}{5(1 + \beta)(1 + \beta^2)\Delta x_i} \quad (3.50)
\]

\[
\frac{dU}{dx}_{i-1} = \frac{2(U_i - U_{i-1})\beta}{5(1 + \beta)\Delta x_i} + \frac{8\beta^3(U_i - U_{i-1}) - (U_{i-1} - U_i)(1 - \beta^2)}{5(1 + \beta)(1 + \beta^2)^2\Delta x_i} + \frac{8\beta^2(U_i - U_{i-1}) - (U_{i-2} - U_{i-1})}{5(1 + \beta)(1 + \beta^2)^2\Delta x_i} \quad (3.51)
\]

Finite volume formulation of the laplace equation gives,

\[
\int_{\Omega_i} \frac{d^2U}{dx^2} \left|_i \right. = \frac{dU}{dx}_{i+1} (\Delta x_i) - \frac{dU}{dx}_{i-1} (\Delta x_i) \quad (3.52)
\]

Thus,

\[
\int_{\Omega_i} \frac{d^2U}{dx^2} \left|_i \right. = \frac{2}{5(1 + \beta)\Delta x_i}((U_{i+1} - U_i) - (U_i - U_{i-1})) + \frac{8\beta(U_{i+1} - U_i) - (U_{i-1} - U_i)(1 - \beta^2)}{5(1 + \beta)(1 + \beta^2)^2\Delta x_i} + \frac{8\beta^2(U_{i+2} - U_{i+1}) - (U_i - U_{i+1}) - \beta(U_i - U_{i-1}) + (U_{i-2} - U_{i-1})}{5(1 + \beta)(1 + \beta^2)^2\Delta x_i} \quad (3.53)
\]

Substituting following Taylor’s expansions,

\[
U_{i+1} = U_i + \frac{dU}{dx} \left|_i \right. (x_{i+1} - x_i) + \frac{d^2U}{dx^2} \left|_i \right. \frac{(x_{i+1} - x_i)^2}{2} + O(h^2) \quad (3.54)
\]

\[
U_{i-1} = U_i + \frac{dU}{dx} \left|_i \right. (x_{i-1} - x_i) + \frac{d^2U}{dx^2} \left|_i \right. \frac{(x_{i-1} - x_i)^2}{2} + O(h^2) \quad (3.55)
\]

\[
U_{i+2} = U_i + \frac{dU}{dx} \left|_i \right. (x_{i+2} - x_i) + \frac{d^2U}{dx^2} \left|_i \right. \frac{(x_{i+2} - x_i)^2}{2} + O(h^2) \quad (3.56)
\]
\[ U_{i-2} = U_i + \frac{dU}{dx} \left( x_{i-2} - x_i \right) + \frac{d^2U}{dx^2} \left( x_{i-2} - x_i \right)^2 + O(h^2) \quad (3.57) \]

Substituting the expansions from 3.54 to 3.57 in equation 3.53, we obtain,

\[
\begin{align*}
\nabla^2 U &= (0) \frac{dU}{dx} + \frac{d^2U(\Delta x_i)}{dx^2} \left[ \frac{(1 + \beta)^2}{20\beta} + \frac{(1 + \beta)(\beta^3 - 1)(1 - \beta^2)}{5\beta(1 + \beta^2)^2} \right] \\
&+ \left[ \frac{(1 + \beta)}{5\beta^2(1 + \beta^2)} \right] \left[ \beta \left( 1 + \beta^3 \right) - 1 + \frac{1}{\beta^2} \left( 1 + \frac{(1 + \beta)^2}{\beta^4} - \frac{1}{\beta^4} \right) \right] \\
\end{align*}
\]

(3.58)

Comparing equation 3.58 with equation 3.18, we can say that our scheme is not dangerously inconsistent although it is inconsistent on non uniform meshes. However our scheme is more accurate than the modified average gradient approach on the irregular and distorted meshes as shown in chapter 4 section 4.7. As the numerical analysis is done in chapter 4, this scheme gives a second order accuracy even for highly distorted meshes.
Chapter 4

Numerical Experiments

Numerical experiments were carried out in this work on various meshes. The results for assessment of accuracy with different diffusion tensors are presented here. These are followed by the accuracy study on distorted meshes. In the following subsection, the scheme is tested for its robustness on heterogeneous diffusion tensor. Further, a study is also carried out to determine the convergence of the preconditioned GMRES solver. A comparison is made between the reconstruction scheme and modified average gradient method. Four kinds of meshes are used to carry out the experiments. Figure 4.1 demonstrates the regular triangular, unstructured and anisotropic meshes while figure 4.9 shows the distorted meshes used. In the presented work some experiments are carried out with known exact solutions. Thus error estimation was done by $L_2$ norm as given in equation 4.1.

$$\|U_h - U\|_{L_2} = \sqrt{\sum_{i=1}^{N} (U_i^{Numeric\ al} - U_i^{Exact})^2} \left[\Omega_i\right] \quad (4.1)$$

where, $N$ is the number of elements. $U_h$ is the numerical solution and $U$ is the exact solution.

4.1 Assessment of Condition Number

The gradient at the cell centers of elements is calculated using the in-cell reconstruction. Thus an over determined system of equations is obtained as given in equations 2.9 to 2.11. This is solved using least squares approach. Normal equation method is used to solve this system but it makes the system ill conditioned and prone to errors. Hence the condition number of the matrix is calculated for a square matrix $A$ as given in equation 4.2.
\[ CN = \frac{1}{\|A\|^2} \] (4.2)

For better conditioning the non normalized form is used as given by equation 2.12. Similarly the normalization of the system is done for calculation of the gradients at the cell interfaces. A comparison is also made between the condition numbers of normalized and non-normalized forms of matrix. Following are the results for condition number for calculation of cell center gradients.

**Table 4.1 Condition Numbers for Gradient at Cell Centers for Regular Triangular Meshes**

<table>
<thead>
<tr>
<th>Meshes</th>
<th>4 X 4 Mesh</th>
<th>8 X 8 Mesh</th>
<th>16 X 16 Mesh</th>
<th>32 X 32 Mesh</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-Normalized</td>
<td>2.766</td>
<td>2.766</td>
<td>2.766</td>
<td>2.766</td>
</tr>
<tr>
<td>Normalized</td>
<td>1.500</td>
<td>1.500</td>
<td>1.500</td>
<td>1.500</td>
</tr>
</tbody>
</table>

**Table 4.2 Condition Numbers for Gradient at Cell Centers for Unstructured Meshes**

<table>
<thead>
<tr>
<th>Meshes</th>
<th>Coarse Mesh</th>
<th>Medium Mesh</th>
<th>Fine Mesh</th>
<th>Very Fine Mesh</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-Normalized</td>
<td>5.158</td>
<td>8.260</td>
<td>16.188</td>
<td>31.789</td>
</tr>
<tr>
<td>Normalized</td>
<td>3.254</td>
<td>3.211</td>
<td>3.319</td>
<td>3.500</td>
</tr>
</tbody>
</table>

**Table 4.3 Condition Numbers for Gradient at Cell Centers for Anisotropic Meshes**

<table>
<thead>
<tr>
<th>Meshes</th>
<th>2 X 6 Mesh</th>
<th>4 X 12 Mesh</th>
<th>8 X 24 Mesh</th>
<th>16 X 48 Mesh</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-Normalized</td>
<td>18.729</td>
<td>18.729</td>
<td>18.729</td>
<td>18.729</td>
</tr>
<tr>
<td>Normalized</td>
<td>3.00</td>
<td>3.00</td>
<td>3.00</td>
<td>3.00</td>
</tr>
</tbody>
</table>
Thus we observe that the normalized matrix gives better conditioning. Similarly the condition number is calculated for the cell interfaces by forming the least squares problem using reconstruction equations 2.18 to 2.23.

**Table 4.4 Condition Numbers for gradient at cell interfaces for Regular Triangular Meshes**

<table>
<thead>
<tr>
<th>N by N Mesh</th>
<th>4 X 4 Mesh</th>
<th>8 X 8 Mesh</th>
<th>16 X 16 Mesh</th>
<th>32 X 32 Mesh</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-Normalized</td>
<td>1.138</td>
<td>1.138</td>
<td>1.138</td>
<td>1.138</td>
</tr>
<tr>
<td>Normalized</td>
<td>1.138</td>
<td>1.138</td>
<td>1.138</td>
<td>1.138</td>
</tr>
</tbody>
</table>

**Table 4.5 Condition Numbers for gradient at cell interfaces for Anisotropic Meshes**

<table>
<thead>
<tr>
<th>Stretched Mesh</th>
<th>2 X 6 Mesh</th>
<th>4 X 12 Mesh</th>
<th>8 X 24 Mesh</th>
<th>16 X 48 Mesh</th>
</tr>
</thead>
</table>

**Table 4.6 Condition Numbers for gradient at cell interfaces for Unstructured Meshes**

<table>
<thead>
<tr>
<th>Unstructured</th>
<th>Coarse Mesh</th>
<th>Medium Mesh</th>
<th>Fine Mesh</th>
<th>Very Fine Mesh</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-Normalized</td>
<td>2.961</td>
<td>5.172</td>
<td>11.215</td>
<td>22.082</td>
</tr>
<tr>
<td>Normalized</td>
<td>2.484</td>
<td>4.789</td>
<td>9.393</td>
<td>18.477</td>
</tr>
</tbody>
</table>
The condition number shows that in case of unstructured meshes the normalized approach decreases the condition number thus it is more beneficial than using non-normalized approach.

4.2 Positivity Study

In this test case it is shown that our scheme does not violate the Discrete Maximum Principle. Different meshes are tried here with different triangulations and the positivity is found to be maintained in all the cases. The analytical solution is not known in this case. Homogeneous Dirichlet boundary condition is applied. The meshes used for the positivity study are shown in figure 4.1. The results obtained on different types of mesh are given in figure 4.2. In this study we consider diffusion problem as in equation 2.1, in the unit square \( \Omega = (0, 1)^2 \). In this study it was shown that the function attains a minimum value of zero at the boundaries. Thus the positivity is maintained throughout the domain.

4.2.1 Test Case 1

a) **Source Term**

\[
f(x, y) = \begin{cases} 
1 & \text{if } (x, y) \in [3/8, 5/8]^2 \\
0 & \text{Otherwise}
\end{cases}
\]  

(4.3)

b) **Diffusion Tensor**

\[
D = \begin{pmatrix} 
y^2 + \varepsilon x^2 \\
-(1 - \varepsilon)xy \\
-\varepsilon y^2 + x^2
\end{pmatrix}, \quad \varepsilon = 5 \times 10^{-2}
\]  

(4.4)

c) **Analytical Function**

*Not Known*

The analytical function is not known in this case. Similar test case has been done in the literature.
Figure 4.1 Sets of Meshes used for experiments of Non Negativity

Following are the results of non negativity test obtained for these meshes.
Figure 4.2 Results of Non-Negativity Test on sets of Meshes considered

(a) 32 X 32 Mesh
(b) Very Fine Mesh
(c) 16 X 48 Mesh
4.3 **Accuracy Study**

Accuracy study is carried out for different diffusion tensors and analytical functions. In all the test cases considered here, the exact solution is known thus the numerical results are compared with the exact solution. In this study we consider diffusion problem as in equation 2.1, in the unit square $\Omega = (0, 1)^2$. Dirichlet boundary condition is applied. Following three sets of meshes are considered in this work, 1) regular triangular meshes 2) unstructured meshes and 3) anisotropic meshes as shown in fig. 4.1 for all the test cases in this study.

4.3.1 **Test Case 1**

a) **Analytical Solution**

\[
U(x, y) = \exp(\alpha \sin(ax + by) + \beta \cos(cx + dy))
\]

where,

$\alpha = 0.1, \beta = 0.3, a = 5.1, b = -6.2, c = 4.3, d = 3.4$

b) **Diffusion Tensor**

\[
D = 1
\]

<table>
<thead>
<tr>
<th>Mesh</th>
<th>Grid Size</th>
<th>$L_2$ Norm Error</th>
<th>$L_\infty$ Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 X 4</td>
<td>0.25</td>
<td>2.07E-2</td>
<td>3.460E-2</td>
</tr>
<tr>
<td>8 X 8</td>
<td>0.125</td>
<td>4.861E-3</td>
<td>1.202E-2</td>
</tr>
<tr>
<td>16 X 16</td>
<td>0.0625</td>
<td>1.196E-3</td>
<td>3.072E-3</td>
</tr>
<tr>
<td>32 X 32</td>
<td>0.03125</td>
<td>3.012E-4</td>
<td>7.487E-4</td>
</tr>
</tbody>
</table>

| Rate | 2.034 | 1.855 |
Table 4.8 Results for test case 1 on Unstructured Meshes for Accuracy Study

<table>
<thead>
<tr>
<th>Mesh</th>
<th>Grid Size</th>
<th>$L_2$ Norm Error</th>
<th>$L_\infty$ Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coarse</td>
<td>0.2425</td>
<td>1.878E-2</td>
<td>4.94E-2</td>
</tr>
<tr>
<td>Medium</td>
<td>0.1212</td>
<td>3.418E-3</td>
<td>9.53E-3</td>
</tr>
<tr>
<td>Fine</td>
<td>0.0606</td>
<td>7.693E-4</td>
<td>2.419E-3</td>
</tr>
<tr>
<td>Very Fine</td>
<td>0.0303</td>
<td>1.876E-4</td>
<td>6.090E-4</td>
</tr>
<tr>
<td>Rate</td>
<td></td>
<td><strong>2.20</strong></td>
<td><strong>2.10</strong></td>
</tr>
</tbody>
</table>

Table 4.9 Results for test case 1 on Anisotropic Meshes for Accuracy Study

<table>
<thead>
<tr>
<th>Mesh</th>
<th>Grid Size</th>
<th>$L_2$ Norm Error</th>
<th>$L_\infty$ Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 X 6</td>
<td>0.288</td>
<td>8.14E-2</td>
<td>1.00E-1</td>
</tr>
<tr>
<td>4 X 12</td>
<td>0.144</td>
<td>2.20E-2</td>
<td>7.95E-2</td>
</tr>
<tr>
<td>8 X 24</td>
<td>0.072</td>
<td>5.63E-3</td>
<td>4.87E-2</td>
</tr>
<tr>
<td>16 X 48</td>
<td>0.036</td>
<td>1.45E-3</td>
<td>2.69E-2</td>
</tr>
<tr>
<td>Rate</td>
<td></td>
<td><strong>1.937</strong></td>
<td><strong>0.641</strong></td>
</tr>
</tbody>
</table>

Tables 4.7, 4.8, and 4.9 give the $L_2$ norm error for different grid sizes. It can be observed from results that in all the sets of meshes considered, the error decreases with the increase in
mesh refinement. The logarithmic values of $L_2$ norm error and the logarithmic values of grid spacing are plotted as in figure 4.4. The slope of the best fit line gives us the order of accuracy which in case of every set of mesh comes out to be second order.

Similar case is tested by Peraire et al. They have tested this case on compact discontinuous Galerkin (CDG) method for diffusion problems. A comparison is drawn between our scheme and CDG method in table 4.10. It is observed that our results match with the CDG methods. Absolute errors for our scheme are lesser for fine meshes as shown in table 4.10. The comparison is carried out on regular triangular meshes. Both the schemes are second order accurate.

**Table 4.10 Comparison between Reconstruction Scheme and CDG Method for Accuracy Study**

<table>
<thead>
<tr>
<th>Mesh</th>
<th>Grid Size</th>
<th>$L_2$ Norm Error (Reconstruction)</th>
<th>$L_2$ Norm Error (CDG Method)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 X 4</td>
<td>0.25</td>
<td>2.071E-2</td>
<td>1.52E-2</td>
</tr>
<tr>
<td>8 X 8</td>
<td>0.125</td>
<td>4.861E-3</td>
<td>4.63E-3</td>
</tr>
<tr>
<td>16 X 16</td>
<td>0.0625</td>
<td>1.196E-3</td>
<td>1.26E-3</td>
</tr>
<tr>
<td>32 X 32</td>
<td>0.03125</td>
<td>3.012E-4</td>
<td>3.27E-4</td>
</tr>
<tr>
<td>Rate</td>
<td></td>
<td><strong>2.034</strong></td>
<td><strong>1.9</strong></td>
</tr>
</tbody>
</table>
Plots are drawn for both the schemes to determine the order of accuracy. Figure 4.3 gives the plots of error against grid size for both schemes and we observe that for fine meshes the absolute errors are almost the same.

The reconstruction scheme is compared with recovery scheme on the similar test case. Table 4.11 gives the comparison between the two schemes with respect to absolute error and order of accuracy.

![Plots comparing errors vs grid size for Reconstruction and CDG schemes](image)

**Figure 4.3 Plots for comparison between Reconstruction scheme and CDG Method**
Table 4.11 Comparison between Reconstruction Scheme and Recovery Scheme for Accuracy Study

<table>
<thead>
<tr>
<th>Mesh</th>
<th>Grid Size</th>
<th>$L_2$ Norm Error (Reconstruction)</th>
<th>$L_2$ Norm Error (Recovery)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 X 4</td>
<td>0.25</td>
<td>2.071E-2</td>
<td>1.94E-002</td>
</tr>
<tr>
<td>8 X 8</td>
<td>0.125</td>
<td>4.861E-3</td>
<td>4.77E-003</td>
</tr>
<tr>
<td>16 X 16</td>
<td>0.0625</td>
<td>1.196E-3</td>
<td>1.19E-003</td>
</tr>
<tr>
<td>32 X 32</td>
<td>0.03125</td>
<td>3.012E-4</td>
<td>3.02E-004</td>
</tr>
<tr>
<td>Rate</td>
<td></td>
<td>2.034</td>
<td>1.9</td>
</tr>
</tbody>
</table>

As it can be observed from table 4.11, the absolute errors for the two schemes are almost similar and hence numerically the order of accuracy obtained is second order for both the schemes. But computationally we can say that reconstruction is better than recovery scheme. The recovery scheme involves computing integrals over the cell volumes. This makes it computationally more complex. A recovery scheme cannot use the boundary face function value for computing gradient at boundary elements, as it requires the area integral over the neighboring cells but reconstruction scheme can use it and compute the gradient at boundary element. This limitation makes the reconstruction scheme better than recovery scheme.
4.3.2 Test Case 2

a) **Analytical Solution**

\[ U(x, y) = \exp(\alpha \sin(ax + by) + \beta \cos(cx + dy)) \]  \hspace{1cm} (4.7)

where,

\[ \alpha = 0.1, \beta = 0.3, a = 5.1, b = -6.2, c = 4.3, d = 3.4 \]

b) **Diffusion Tensor**

\[ D = \begin{pmatrix} (x + 1)^2 + y^2 & -xy \\ -xy & (y + 1)^2 \end{pmatrix} \]  \hspace{1cm} (4.8)

**Table 4.12 Results for test case 2 for Regular Triangular Meshes for Accuracy Study**

<table>
<thead>
<tr>
<th>Mesh</th>
<th>Grid Size</th>
<th>(L_2) Norm Error</th>
<th>(L_\infty) Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 X 4</td>
<td>0.25</td>
<td>2.33E-2</td>
<td>3.50E-2</td>
</tr>
<tr>
<td>8 X 8</td>
<td>0.125</td>
<td>4.91E-3</td>
<td>1.17E-2</td>
</tr>
<tr>
<td>16 X 16</td>
<td>0.0625</td>
<td>1.20E-3</td>
<td>2.93E-3</td>
</tr>
<tr>
<td>32 X 32</td>
<td>0.03125</td>
<td>3.02E-4</td>
<td>7.45E-4</td>
</tr>
<tr>
<td>Rate</td>
<td></td>
<td>2.08</td>
<td>1.86</td>
</tr>
</tbody>
</table>
### Table 4.13 Results for test case 2 for Unstructured Meshes for Accuracy Study

<table>
<thead>
<tr>
<th>Mesh</th>
<th>Grid Size</th>
<th>$L_2$ Norm Error</th>
<th>$L_\infty$ Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coarse</td>
<td>0.176</td>
<td>1.66E-2</td>
<td>3.78E-2</td>
</tr>
<tr>
<td>Medium</td>
<td>0.088</td>
<td>3.34E-3</td>
<td>9.89E-3</td>
</tr>
<tr>
<td>Fine</td>
<td>0.044</td>
<td>7.92E-4</td>
<td>2.35E-3</td>
</tr>
<tr>
<td>Very Fine</td>
<td>0.022</td>
<td>1.99E-4</td>
<td>5.88E-4</td>
</tr>
<tr>
<td>Rate</td>
<td></td>
<td><strong>2.12</strong></td>
<td><strong>2.00</strong></td>
</tr>
</tbody>
</table>

### Table 4.14 Results for test case 2 for Anisotropic Meshes for Accuracy Study

<table>
<thead>
<tr>
<th>Mesh</th>
<th>Grid Size</th>
<th>$L_2$ Norm Error</th>
<th>$L_\infty$ Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 X 6</td>
<td>0.288</td>
<td>8.02E-2</td>
<td>7.78E-2</td>
</tr>
<tr>
<td>4 X 12</td>
<td>0.144</td>
<td>2.04E-2</td>
<td>5.10E-2</td>
</tr>
<tr>
<td>8 X 24</td>
<td>0.072</td>
<td>5.09E-3</td>
<td>3.49E-2</td>
</tr>
<tr>
<td>16 X 48</td>
<td>0.036</td>
<td>1.29E-3</td>
<td>1.99E-2</td>
</tr>
<tr>
<td>Rate</td>
<td></td>
<td><strong>1.987</strong></td>
<td><strong>0.644</strong></td>
</tr>
</tbody>
</table>
Logarithmic values of error are plotted against logarithmic values of grid sizes as in figure 4.4 and it is observed that the scheme is second order accurate for this test case also.

**Figure 4.4** Plots for Order of Accuracy for test case 1 and test case 2 on different meshes considered
For comparative purposes, figure 4.5 gives the contour plot of the exact solution and numerical solution for a fine unstructured mesh.

![Contour plots for numerical solution and exact solution for test case 2](image)

**Figure 4.5 Contour plots for numerical solution and exact solution for test case 2**

4.3.3 **Test Case 3**

a) **Analytical Solution**

\[ U(x, y) = 2 \cos(\pi x) \sin(2\pi y) + 2 \]  

(4.9)

b) **Diffusion Tensor**

\[ D = 1 \]  

(4.10)

In this case a unit diffusion tensor is used as done previously also.
Table 4.15 Results for test case 3 for Regular Triangular Mesh for Accuracy Study

<table>
<thead>
<tr>
<th>Mesh</th>
<th>Grid Size</th>
<th>$L_2$ Norm Error</th>
<th>$L_\infty$ Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 X 4</td>
<td>0.25</td>
<td>3.39E-1</td>
<td>2.20E-1</td>
</tr>
<tr>
<td>8 X 8</td>
<td>0.125</td>
<td>6.63E-2</td>
<td>3.82E-2</td>
</tr>
<tr>
<td>16 X 16</td>
<td>0.0625</td>
<td>1.57E-2</td>
<td>9.06E-3</td>
</tr>
<tr>
<td>32 X 32</td>
<td>0.03125</td>
<td>3.88E-3</td>
<td>2.22E-3</td>
</tr>
<tr>
<td>Rate</td>
<td></td>
<td>2.141</td>
<td>1.895</td>
</tr>
</tbody>
</table>

Table 4.16 Results for test case 3 for Unstructured Mesh for Accuracy Study

<table>
<thead>
<tr>
<th>Mesh</th>
<th>Grid Size</th>
<th>$L_2$ Norm Error</th>
<th>$L_\infty$ Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coarse</td>
<td>0.176</td>
<td>1.25E-1</td>
<td>2.69E-1</td>
</tr>
<tr>
<td>Medium</td>
<td>0.088</td>
<td>2.23E-2</td>
<td>5.86E-2</td>
</tr>
<tr>
<td>Fine</td>
<td>0.044</td>
<td>5.32E-3</td>
<td>1.60E-2</td>
</tr>
<tr>
<td>Very Fine</td>
<td>0.022</td>
<td>1.35E-3</td>
<td>-4.73E-3</td>
</tr>
<tr>
<td>Rate</td>
<td></td>
<td>2.165</td>
<td>1.937</td>
</tr>
</tbody>
</table>
Table 4.17 Results for test case 3 for Anisotropic Mesh for Accuracy Study

<table>
<thead>
<tr>
<th>Mesh</th>
<th>Grid Size</th>
<th>$L_2$ Norm Error</th>
<th>$L_\infty$ Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 X 6</td>
<td>0.288</td>
<td>4.85E-1</td>
<td>2.26E-1</td>
</tr>
<tr>
<td>4 X 12</td>
<td>0.144</td>
<td>1.08E-1</td>
<td>4.66E-2</td>
</tr>
<tr>
<td>8 X 24</td>
<td>0.072</td>
<td>2.54E-2</td>
<td>1.29E-2</td>
</tr>
<tr>
<td>16 X 48</td>
<td>0.036</td>
<td>6.22E-3</td>
<td>3.27E-3</td>
</tr>
<tr>
<td>Rate</td>
<td></td>
<td><strong>2.094</strong></td>
<td><strong>2.018</strong></td>
</tr>
</tbody>
</table>

4.3.3 Test Case 4

a) **Analytical Solution**

\[ U(x, y) = 2 \cos(\pi x) \sin(2\pi y) + 2 \]  \hspace{1cm} (4.11)

b) **Diffusion Tensor**

\[ D = \begin{pmatrix} \cos \theta & \sin \theta \\ -\cos \theta & \sin \theta \end{pmatrix} \begin{pmatrix} k_1 & 0 \\ 0 & k_2 \end{pmatrix} \begin{pmatrix} \cos \theta & -\sin \theta \\ \cos \theta & \sin \theta \end{pmatrix} \]  \hspace{1cm} (4.12)

where, $k_1 = 1000$, $k_2 = 0$ and $\theta = \pi/6$. 
Table 4.18 Results for test case 4 for Regular Triangular Meshes for Accuracy Study

<table>
<thead>
<tr>
<th>Mesh</th>
<th>Grid size</th>
<th>$L_2$ Norm Error</th>
<th>$L_\infty$ Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 X 4</td>
<td>0.25</td>
<td>4.90E-1</td>
<td>0.911</td>
</tr>
<tr>
<td>8 X 8</td>
<td>0.125</td>
<td>1.13E-1</td>
<td>0.289</td>
</tr>
<tr>
<td>16 X 16</td>
<td>0.0625</td>
<td>2.82E-2</td>
<td>7.11E-2</td>
</tr>
<tr>
<td>32 X 32</td>
<td>0.03125</td>
<td>7.05E-3</td>
<td>1.76E-2</td>
</tr>
<tr>
<td>Rate</td>
<td></td>
<td>2.036</td>
<td>1.995</td>
</tr>
</tbody>
</table>

Table 4.19 Results for test case 4 for Unstructured Meshes for Accuracy Study

<table>
<thead>
<tr>
<th>Mesh</th>
<th>Grid Size</th>
<th>$L_2$ Norm Error</th>
<th>$L_\infty$ Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coarse</td>
<td>0.176</td>
<td>3.15E-1</td>
<td>5.58E-1</td>
</tr>
<tr>
<td>Medium</td>
<td>0.088</td>
<td>8.30E-2</td>
<td>2.01E-1</td>
</tr>
<tr>
<td>Fine</td>
<td>0.044</td>
<td>2.33E-2</td>
<td>5.85E-2</td>
</tr>
<tr>
<td>Very Fine</td>
<td>0.022</td>
<td>6.65E-3</td>
<td>1.91E-2</td>
</tr>
<tr>
<td>Rate</td>
<td></td>
<td>1.853</td>
<td>1.639</td>
</tr>
</tbody>
</table>
Table 4.20 Results for test case 4 for Anisotropic Meshes for Accuracy Study

<table>
<thead>
<tr>
<th>Mesh</th>
<th>Grid Size</th>
<th>$L_2$ Norm Error</th>
<th>$L_\infty$ Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 X 6</td>
<td>0.288</td>
<td>4.50E-1</td>
<td>6.58E-1</td>
</tr>
<tr>
<td>4 X 12</td>
<td>0.144</td>
<td>9.39E-2</td>
<td>1.41E-1</td>
</tr>
<tr>
<td>8 X 24</td>
<td>0.072</td>
<td>2.20E-2</td>
<td>3.39E-2</td>
</tr>
<tr>
<td>16 X 48</td>
<td>0.036</td>
<td>5.42E-3</td>
<td>8.40E-3</td>
</tr>
<tr>
<td>Rate</td>
<td></td>
<td>2.121</td>
<td>2.093</td>
</tr>
</tbody>
</table>

Figure 4.6 gives the logarithmic plots for both test case 1 and test case 2 for all the three sets of meshes considered. The scheme is again proved to be second order accurate.
Figure 4.6 Logarithmic plots for test case 3 and test case 4 for different meshes

Figure 4.7 compares the contour plots of numerical solutions to the exact solutions. Thus for both the cases the scheme is shown to be second order accurate.

Figure 4.7 Contour plots for Numerical Solution and Exact Solution for test case 4
4.4 **Accuracy study on Distorted Meshes**

The computation is carried out for distorted meshes, in order to test our scheme in rigorous conditions. These meshes are produced by perturbing the coordinates of the mesh points by a small random number. The coordinates are perturbed as follows.

\[
\begin{align*}
    x' &= x + \alpha h \\
    y' &= y + \alpha h
\end{align*}
\]

where,

\( \alpha = \) Distortion Parameter

\( h = \) Grid Spacing

<table>
<thead>
<tr>
<th>Mesh with Distortion Parameter = 0.4</th>
<th>Mesh with Distortion Parameter = 0.6</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Mesh 0.4" /></td>
<td><img src="image2.png" alt="Mesh 0.6" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Mesh with Distortion Parameter = 0.8</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image3.png" alt="Mesh 0.8" /></td>
</tr>
</tbody>
</table>

*Figure 4.8 Different types of Distorted Meshes considered*
Three sets of meshes are produced each for distortion parameter as 0.4, 0.6 and 0.8. Three test cases are considered in this study. Figure 4.8 provides the different types of meshes considered for different distortion parameters. Figure 4.9 provides the four refined levels of a
set of meshes with distortion parameter = 0.6. We consider problem as in equation 2.1, in the unit square such that the computational domain is given as \( \Omega = (0, 1)^2 \). In all the cases considered here Dirichlet boundary condition is applied.

4.4.1 Test Case 1

a) Analytical Solution

\[ U(x, y) = \exp(\alpha \sin(ax + by) + \beta \cos(cx + dy)) \]

where,
\[ \alpha = 0.1, \beta = 0.3, a = 5.1, b = -6.2, c = 4.3, d = 3.4 \]

b) Diffusion Tensor

\[ D = 1 \]

The set of grids considered in this case is distorted mesh with distortion parameter = 0.6 as in figure 4.8.

Table 4.21 Results for test case 1 for accuracy study on distorted meshes with Distortion Parameter = 0.6

<table>
<thead>
<tr>
<th>Mesh</th>
<th>Grid Size</th>
<th>( L_2 ) Norm Error</th>
<th>( L_\infty ) Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coarse</td>
<td>0.176</td>
<td>1.49E-2</td>
<td>2.86E-2</td>
</tr>
<tr>
<td>Medium</td>
<td>0.088</td>
<td>1.92E-3</td>
<td>4.61E-3</td>
</tr>
<tr>
<td>Fine</td>
<td>0.044</td>
<td>4.69E-4</td>
<td>2.21E-3</td>
</tr>
<tr>
<td>Very Fine</td>
<td>0.022</td>
<td>1.13E-4</td>
<td>5.12E-4</td>
</tr>
<tr>
<td>Rate</td>
<td></td>
<td>2.237</td>
<td>1.871</td>
</tr>
</tbody>
</table>
4.4.2 Test Case 2

a) Analytical Solution

\[ U(x,y) = \exp(\alpha \sin(ax + by) + \beta \cos(cx + dy)) \]  

where,
\[ \alpha = 0.1, \beta = 0.3, a = 5.1, b = -6.2, c = 4.3, d = 3.4 \]

b) Diffusion Tensor

\[ D = \begin{pmatrix} (x + 1)^2 + y^2 & -xy \\ -xy & (y + 1)^2 \end{pmatrix} \]

The set of grids considered in this case is distorted mesh with distortion parameter = 0.6 as shown in figure 4.8.

Table 4.22 Results for test case 2 of accuracy study on distorted meshes with Distortion Parameter = 0.6

<table>
<thead>
<tr>
<th>Mesh</th>
<th>Grid Size</th>
<th>( L_2 ) Norm Error</th>
<th>( L_{\infty} ) Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.176</td>
<td>1.38E-2</td>
<td>4.12E-2</td>
</tr>
<tr>
<td>2</td>
<td>0.088</td>
<td>2.64E-3</td>
<td>8.31E-3</td>
</tr>
<tr>
<td>3</td>
<td>0.044</td>
<td>8.05E-4</td>
<td>3.16E-3</td>
</tr>
<tr>
<td>4</td>
<td>0.022</td>
<td>2.49E-4</td>
<td>9.15E-4</td>
</tr>
<tr>
<td>Rate</td>
<td></td>
<td>1.902</td>
<td>1.787</td>
</tr>
</tbody>
</table>

It can be observed from table 4.21 and table 4.22, for the distorted meshes also in above cases the scheme is proved to be second order.
4.4.3 Test Case 3

a) Analytical Solution

\[ U(x, y) = 2 \cos(\pi x) \sin(2\pi y) + 2 \]  \hspace{1cm} (4.17)

b) Diffusion Tensor

\[ D = 1 \]  \hspace{1cm} (4.18)

Three sets of grids are considered in this case 1) distorted mesh with distortion parameter = 0.4 2) distorted mesh with distortion parameter = 0.6 and 3) distorted mesh with distortion parameter = 0.8 as shown in fig.4.8.
Table 4.23 Results for test case 3 for Accuracy Study on Distorted Mesh with Distortion Parameter as 0.4

<table>
<thead>
<tr>
<th>Mesh</th>
<th>Grid Size</th>
<th>$L_2$ Norm Error</th>
<th>$L_\infty$ Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coarse</td>
<td>0.176</td>
<td>2.25E-1</td>
<td>4.89E-1</td>
</tr>
<tr>
<td>Medium</td>
<td>0.088</td>
<td>4.51E-2</td>
<td>1.42E-1</td>
</tr>
<tr>
<td>Fine</td>
<td>0.044</td>
<td>1.01E-2</td>
<td>2.98E-2</td>
</tr>
<tr>
<td>Very Fine</td>
<td>0.022</td>
<td>2.41E-3</td>
<td>7.21E-3</td>
</tr>
<tr>
<td>Rate</td>
<td></td>
<td>2.181</td>
<td>2.051</td>
</tr>
</tbody>
</table>

Table 4.24 Results for test case 3 for Accuracy Study on Distorted Mesh with Distortion Parameter as 0.6

<table>
<thead>
<tr>
<th>Mesh</th>
<th>Grid Size</th>
<th>$L_2$ Norm Error</th>
<th>$L_\infty$ Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coarse</td>
<td>0.176</td>
<td>2.84E-1</td>
<td>5.75E-1</td>
</tr>
<tr>
<td>Medium</td>
<td>0.088</td>
<td>5.67E-2</td>
<td>1.82E-1</td>
</tr>
<tr>
<td>Fine</td>
<td>0.044</td>
<td>1.25E-2</td>
<td>3.90E-2</td>
</tr>
<tr>
<td>Very Fine</td>
<td>0.022</td>
<td>2.96E-3</td>
<td>9.45E-3</td>
</tr>
<tr>
<td>Rate</td>
<td></td>
<td>2.192</td>
<td>2.001</td>
</tr>
</tbody>
</table>
Table 4.25 Results for test case 3 for accuracy study on distorted mesh with

**Distortion Parameter as 0.8**

<table>
<thead>
<tr>
<th>Mesh</th>
<th>Grid Size</th>
<th>( L_2 ) Norm Error</th>
<th>( L_\infty ) Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coarse</td>
<td>0.176</td>
<td>3.39E-1</td>
<td>7.07E-1</td>
</tr>
<tr>
<td>Medium</td>
<td>0.088</td>
<td>6.73E-2</td>
<td>2.19E-1</td>
</tr>
<tr>
<td>Fine</td>
<td>0.044</td>
<td>1.46E-2</td>
<td>4.76E-2</td>
</tr>
<tr>
<td>Very Fine</td>
<td>0.022</td>
<td>3.44E-3</td>
<td>1.19E-2</td>
</tr>
<tr>
<td><strong>Rate</strong></td>
<td><strong>2.207</strong></td>
<td><strong>1.986</strong></td>
<td></td>
</tr>
</tbody>
</table>

Thus we obtain the result that the scheme is still second order accurate even for distorted mesh.

*Figure 4.11 Logarithmic Plots for test case3 for accuracy study on distorted meshes with different distortion parameter*
4.5 Study on Heterogeneous Diffusion Tensor

In this study the scheme is analyzed to show that it can handle the jumps of heterogeneous diffusion tensor. The study is carried out on two test cases. These cases have different distribution of the diffusion tensor. The source term and boundary conditions considered are same as done by Lipnikov et.al [14]. The computational domain is partitioned in four sub domains and different diffusion tensors are given in different sub domains by varying certain parameters in the diffusion tensor. This introduces the discontinuities at the edges of the computational sub domain. Homogeneous Dirichlet boundary condition is applied. Analytical function is not known in both test cases. The meshes considered are the unstructured meshes as shown in fig 4.1(b).

4.5.1 Test Case 1

a) Source Term

\[
f(x) = \begin{cases} 
\frac{1}{|\omega|} & \text{if } x \in \omega, \text{ where } \omega = [7/18, 11/18]^2 \\
0 & \text{Otherwise}, 
\end{cases}
\]  

(4.19)

b) Diffusion Tensor

\[
D = \begin{pmatrix} 
\cos \theta & \sin \theta \\
-\cos \theta & \sin \theta 
\end{pmatrix} \begin{pmatrix} 
k_1 & 0 \\
k_2 & 0 
\end{pmatrix} \begin{pmatrix} 
\cos \theta & -\sin \theta \\
\sin \theta & \cos \theta 
\end{pmatrix}
\]  

(4.20)

In the diffusion tensor as given in equation 4.20, \(k_1\) and \(k_2\) are the eigen values. The values of \(k_1\), \(k_2\) and \(\theta\) are varied as demonstrated in figure 4.12(a). In this case \(k_1\) and \(k_2\) are fixed as 1000 and 1 respectively and the orientation are changed for the sub domains. Thus we obtain different diffusion tensors in different sub domains.
Figure 4.12 Results showing jumps across mesh edges in case of Anisotropic meshes
4.5.2 Test Case 2

a) **Source Term**

\[
f(x) = \begin{cases} 
\frac{1}{|\omega|} & \text{if } x \in \omega, \text{ where } \omega = [7/18, 11/18]^2 \\
0 & \text{Otherwise},
\end{cases}
\]  

(4.21)

b) **Diffusion Tensor**

\[
D = \begin{pmatrix} 
\cos\theta & \sin\theta \\
-\cos\theta & \sin\theta 
\end{pmatrix} \begin{pmatrix} 
k_1 & 0 \\
0 & k_2
\end{pmatrix} \begin{pmatrix} 
\cos\theta & -\sin\theta \\
\sin\theta & \cos\theta
\end{pmatrix}
\]  

(4.22)

In this test case a chess board distribution of diffusion tensor is used for \( k_1 \) and \( k_2 \) as shown in figure 4.13(a).

Figure 4.13 Results showing jumps across mesh edges in case of Anisotropic meshes
4.6 Convergence Study

The GMRES solver is employed with left preconditioning as explained in chapter 2. Thus computation is carried out for a comparison of the three preconditioners used in this work namely Diagonal, LU-SGS and ILU. The logarithmic values of residuals are plotted with the number of iterations, using each preconditioner. The effect of preconditioner is easily established as can be seen in the plots. The number of iterations required for convergence drastically decreases with each preconditioner. It is easily observed that ILU preconditioner is the best preconditioner for the given problem with least number of iterations required. The CPU time taken is also plotted here as can be seen again that minimum time is taken by ILU preconditioner and maximum time is taken by diagonal preconditioner. A study is conducted here showing the convergence of GMRES with three different preconditioners. All the plots are obtained by carrying out this study on regular triangular mesh with grid size 0.25. The relative tolerance is set as 1E-08 for convergence.

4.6.1 Test Case

a) Analytical Solution

\[ U(x, y) = \exp(\alpha \sin(ax + by) + \beta \cos(cx + dy)) \]  

(4.23)

where,

\[ \alpha = 0.1, \beta = 0.3, a = 5.1, b = -6.2, c = 4.3, d = 3.4 \]

b) Diffusion Tensor

\[ D = 1 \]  

(4.24)

As previously done, unit diffusion tensor is applied.
As it can be seen from the plot in figure 4.14, the rate of convergence is increased with use of preconditioner. Also the rate of convergence is higher for the case when ILU preconditioner is used than the case when diagonal and LU-SGS preconditioners are used. Table 4.26 shows the CPU time taken and number of iterations taken for every preconditioner for convergence.

**Table 4.26 CPU Times and number of iterations taken by each Preconditioner**

<table>
<thead>
<tr>
<th>Method</th>
<th>CPU Time (Seconds)</th>
<th>Number of Iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>GMRES</td>
<td>0.4599</td>
<td>780</td>
</tr>
<tr>
<td>GMRES + Diagonal</td>
<td>0.20096</td>
<td>310</td>
</tr>
<tr>
<td>GMRES + LU-SGS</td>
<td>1.299E-002</td>
<td>19</td>
</tr>
<tr>
<td>GMRES + ILU</td>
<td>4.998E-003</td>
<td>9</td>
</tr>
</tbody>
</table>
Thus it is easily observed that without preconditioner, GMRES takes more number of iterations than the case when preconditioner is used. Also the number of iterations taken by ILU preconditioner is the smallest. Figure 4.15 depicts the variation of logarithmic value of residual with CPU time. So it is observed that using GMRES with ILU Preconditioner results in fastest drop in residual values. Although the results plotted here are only for unstructured triangular elements but similar improvement in convergence and CPU times were recorded for regular triangular meshes and anisotropic meshes.

4.7 Comparison for Reconstruction scheme and Modified Gradient Approach

In this section the reconstruction scheme is compared with modified average gradient Approach. The consistency is maintained for both the schemes for 1D uniform grid. The order of accuracy is second order for both schemes in the case of uniform grids. In test case 1 regular triangular unstructured grids, unstructured triangular meshes and anisotropic meshes
as shown in fig 4.1 are chosen for numerical analysis. In test case 2 symmetric stretched grids and non symmetric stretched grids as shown in fig 4.16 are used for study. In these meshes a stretching parameter is used for y-direction as given in equation 4.25.

\[ \beta = \frac{\Delta y_i}{\Delta y_{i-1}} = \frac{\Delta y_{i+1}}{\Delta y_i} \]  \hspace{1cm} (4.25)

where, \( \beta \) is the stretching ratio and \( \Delta y_i \) is the length of \( i^{th} \) cell in y-direction. For our case we have considered a stretching ratio of 1.2.

In this study two test cases are considered. Dirichlet boundary conditions are used in both the cases.

![Symmetric Stretched Mesh](image1) ![Non Symmetric Stretched Mesh](image2)

**Figure 4.16 Mesh Topologies of Symmetric Stretched and Non Symmetric Stretched Meshes**
4.7.1 Test Case 1

a) Analytical Solution

\[ U(x, y) = \exp(\alpha \sin(ax + by) + \beta \cos(cx + dy)) \] \hspace{1cm} (4.27)

where,

\[ \alpha = 0.1, \beta = 0.3, a = 5.1, b = -6.2, c = 4.3, d = 3.4 \]

b) Diffusion Tensor

\[ D = \begin{pmatrix} (x + 1)^2 + y^2 & -xy \\ -xy & (y + 1)^2 \end{pmatrix} \] \hspace{1cm} (4.28)

Table 4.27 Comparison between reconstruction scheme and modified gradient approach on regular triangular meshes for test case 1

<table>
<thead>
<tr>
<th>Mesh</th>
<th>Grid Size</th>
<th>L₂ Norm Error (Reconstruction)</th>
<th>L₂ Norm Error (Modified Gradient)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 by 4</td>
<td>0.2500</td>
<td>2.33E-2</td>
<td>4.26E-2</td>
</tr>
<tr>
<td>8 by 8</td>
<td>0.1250</td>
<td>4.90E-3</td>
<td>1.17E-2</td>
</tr>
<tr>
<td>16 by 16</td>
<td>0.0625</td>
<td>1.20E-3</td>
<td>3.75E-3</td>
</tr>
<tr>
<td>32 by 32</td>
<td>0.03125</td>
<td>3.02E-4</td>
<td>7.73E-4</td>
</tr>
<tr>
<td>Rate</td>
<td></td>
<td>2.08</td>
<td>1.93</td>
</tr>
</tbody>
</table>
Table 4.28 Comparison between reconstruction scheme and modified gradient approach on unstructured meshes for test case 1

<table>
<thead>
<tr>
<th>Mesh</th>
<th>Grid Size</th>
<th>$L_2$ Norm Error (Reconstruction)</th>
<th>$L_2$ Norm Error (Modified Gradient)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coarse</td>
<td>0.2425</td>
<td>1.66E-2</td>
<td>6.47E-2</td>
</tr>
<tr>
<td>Medium</td>
<td>0.1212</td>
<td>3.34E-3</td>
<td>2.12E-2</td>
</tr>
<tr>
<td>Fine</td>
<td>0.0606</td>
<td>7.92E-4</td>
<td>5.73E-3</td>
</tr>
<tr>
<td>Very Fine</td>
<td>0.0303</td>
<td>1.99E-4</td>
<td>2.04E-3</td>
</tr>
<tr>
<td>Rate</td>
<td></td>
<td>2.12</td>
<td>1.63</td>
</tr>
</tbody>
</table>

Table 4.29 Comparison between reconstruction scheme and modified gradient approach on anisotropic meshes for test case 1

<table>
<thead>
<tr>
<th>Mesh</th>
<th>Grid Size</th>
<th>$L_2$ Norm Error (Reconstruction)</th>
<th>$L_2$ Norm Error (Modified Gradient)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 X 6</td>
<td>0.288</td>
<td>8.029E-2</td>
<td>1.284E-1</td>
</tr>
<tr>
<td>4 X 12</td>
<td>0.144</td>
<td>2.047E-2</td>
<td>5.464E-2</td>
</tr>
<tr>
<td>8 X 24</td>
<td>0.0721</td>
<td>5.092E-3</td>
<td>1.766E-2</td>
</tr>
<tr>
<td>16 X 48</td>
<td>0.0360</td>
<td>1.293E-3</td>
<td>4.643E-3</td>
</tr>
<tr>
<td>Rate</td>
<td></td>
<td>1.987</td>
<td>1.599</td>
</tr>
</tbody>
</table>
Figure 4.17 shows the plots between logarithmic values of $L_2$ norm error against grid sizes. The order of accuracy of the two schemes is determined by comparing the slopes of the linear plots between log (error) vs. log (grid size). It is observed from table 4.28 for uniform triangular grids both the schemes showed second order accuracy but as shown in table 4.29 and 4.30 for unstructured and anisotropic meshes the reconstruction scheme is close to second order accuracy while the modified average is less than second order. The absolute errors for the reconstruction scheme are found to be lesser than modified average gradient method. Hence with this study we can conclude that our scheme is more accurate than modified average gradient method even on unstructured and anisotropic grids.

Figure 4.17 Comparisons between Reconstruction Schemes and Modified Gradient Approach for test case 1
4.7.2 Test Case 2

a) Analytical Function

Following logarithmic function is used in this case.

\[ U(x, y) = \log(y + 0.1) \]  

\( (4.29) \)

b) Diffusion Tensor

\[ D = 1 \]  

\( (4.30) \)

Table 4.30 Comparison between Reconstruction and Modified Average Approach over set of Symmetric Stretched Meshes

<table>
<thead>
<tr>
<th>Mesh</th>
<th>Grid Size</th>
<th>L2 Norm Error (Reconstruction)</th>
<th>L2 Norm Error (Modified Gradient)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 X 4</td>
<td>0.25</td>
<td>1.495E-1</td>
<td>1.284E-1</td>
</tr>
<tr>
<td>8 X 8</td>
<td>0.125</td>
<td>2.671E-2</td>
<td>2.465E-2</td>
</tr>
<tr>
<td>16 X 16</td>
<td>0.0625</td>
<td>2.862E-3</td>
<td>3.096E-3</td>
</tr>
<tr>
<td>32 X 32</td>
<td>0.03125</td>
<td>2.573E-3</td>
<td>2.767E-3</td>
</tr>
<tr>
<td>Rate</td>
<td></td>
<td>1.958</td>
<td>1.958</td>
</tr>
</tbody>
</table>
Table 4.31 Comparison between Reconstruction and Modified Average Approach over set of Non Symmetric Stretched Meshes

<table>
<thead>
<tr>
<th>Mesh</th>
<th>Grid Size</th>
<th>$L_2$ Norm Error (Reconstruction)</th>
<th>$L_2$ Norm Error (Modified Gradient)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 X 4</td>
<td>0.25</td>
<td>1.52E-1</td>
<td>3.07E-1</td>
</tr>
<tr>
<td>8 X 8</td>
<td>0.125</td>
<td>2.71E-2</td>
<td>4.59E-2</td>
</tr>
<tr>
<td>16 X 16</td>
<td>0.0625</td>
<td>2.78E-3</td>
<td>1.35E-2</td>
</tr>
<tr>
<td>32 X 32</td>
<td>0.03125</td>
<td>2.55E-3</td>
<td>1.30E-2</td>
</tr>
<tr>
<td>Rate</td>
<td></td>
<td>1.958</td>
<td>1.54</td>
</tr>
</tbody>
</table>

Figure 4.18 Plots of Comparison between Reconstruction and Modified Average for test case 2
Thus from table 4.30 and 4.31, it is observed that for symmetric stretched meshes the order of accuracy for both the schemes is similar but for non symmetric stretched meshes the order of accuracy is more and closer to second order for reconstruction scheme. Absolute error is observed to be lesser for reconstruction scheme than modified approach for both types of meshes.
Conclusion

A reconstruction based finite volume scheme is developed and analyzed in this work for estimation of gradients. An accuracy study on regular and distorted meshes showed that our scheme is second order accurate. It does not violate the Discrete Maximum Principle (DMP). Our scheme is shown to be robust and able to handle the discontinuities in the heterogeneous diffusion tensor. Reconstruction scheme is analyzed and compared with modified average gradient method and it was found that both the schemes are consistent on uniform grids and inconsistent on non uniform grids, but our scheme is found to be better as it is grid transparent and gives second order accuracy numerically even for symmetric and non-symmetric stretched grids. An important advantage of the reconstruction scheme is that its extension to the Navier Stokes equation is straight forward for the discretization of the viscous and heat fluxes. Further a convergence study is conducted for the GMRES solver using three different preconditioners. ILU preconditioner is found to be most effective as the CPU time taken and number of iterations were least recorded for ILU preconditioner. Evaluation of the numerical jacobian is very complicated thus CPR algorithm is tried which reduces significantly the function evaluations for jacobian matrix computation.

Thus we can conclude that compared to other common approaches, the reconstruction scheme is an effective and more accurate way of estimating the gradients and it can be easily extended for Navier Stokes equations where the viscous fluxes require the gradients at the interfaces to be evaluated.
References


