Numerical Tools for Multicomponent, Multiphase, Reactive Processes: Flow of CO_2 in Porous Medium

Sanjay Kumar Khattri

A Dissertation in Applied Mathematics
Department of Mathematics
University of Bergen
Norway

2006

To The love of
My wife“Ritu Khattri”
and
My daughter “Avni Sanjay”

ACKNOWLEDGMENTS

This work would have never been completed without the support and encouragement from my advisors Gunnar Fladmark and Ivar Aavatsmark during the course of PhD studies. Their guidance and advise made me work hard and progress in the field of scientific computation. In life and mathematics direction is the most important thing, I thanks my advisors for showing me the right path. I would also like to take this opportunity to thank professors Magne Espedal and Helge Dahle whose help and encouragement was of great help in understanding many mathematical problems. I greatly appreciate their concern, remarks and advice. I feel lucky and thankful for the progressive research environment at the department of mathematics. I am thankful to the graduate students and faculty members at the department.

I am indebted to my mother, Asha Khattri, for everything. I thanks my father in-law, Kamlesh Kumar Mehrotra, and mother in-law, Shell Mehrotra, for their help. Finally this work is devoted to the three most important persons in my life my mother, my wife and my daughter.

This work is supported by the Norwegian Research Council and Norks Hydro under the grants 151400/210.

PREFACE

This work is part of the author's PhD thesis done at the Department of Mathematics, University of Bergen, under the supervision of Professor Gunnar Fladmark and Professor Ivar Aavatsmark. The thesis is concerned with numerically simulating multicomponent, multiphase, reactive transport in heterogenous porous medium. Such processes are ubiquitous, for example, deposition of green house gases, flow of hydrocarbons and groundwater remediation. Understanding such processes are important from social and economic point of view. For the success of geological sequestration, an accurate estimation of migration patterns of green-house gases is essential. Due to an ever increasing computer power, computational mathematics has become an important tool for predicting dynamics of porous media fluids. Numerical and mathematical modelling of processes in a domain requires grid generation in the domain, discretization of the continuum equations on the generated grid, solution of the formed linear or nonlinear system of discrete equations and finally visualization of the results. During the course of PhD studies, the author got the opportunity of working with all of these topics.

The thesis is composed of three chapters and eight papers. The rest of the thesis is organised as follows.

The Chapter 2 presents two techniques for generating structured quadrilateral and hexahedral meshes. These techniques are called algebraic and elliptic methods. Algebraic techniques are by far the most simple and computationally efficient method for grid generation. Transfinite interpolation operators are a kind of algebraic grid generation technique. In this chapter, many transfinite interpolation operators for grid generation are derived from 1D projection operators. The author has developed a block structured hexahedral grid generator named RMS-GG. The RMS-GG is based on algebraic techniques. RMS-GG can import geometry from IRAP-RMS and Eclipse softwares into the inhouse flow simulator named Athena. In this chapter, some important properties of hexahedral elements are also mentioned. These properties are useful in discretization of partial differential equations on hexahedral mesh, improving quality of the hexahedral mesh, mesh generation and visualization.

The Chapter 3 is about CO_2 flow in porous media. In this chapter, we present the mathematical models and their discretization for capturing major physical processes associated with CO_2 deposition in geological formations. Some important simulations of practical applications in 2D and 3D are presented. The author has done the verification of the existing software package named Athena for understanding CO_2 deposition.

The Chapter 4 presents Control Volume discretization on adaptive meshes. In this chapter, criteria for adaptive refinement and an adaptive algorithm is presented. Numerical examples of single phase flow in heterogenous media are also reported.

The following papers are included in the Part II

Paper A: **A New Smoothing Algorithm for Quadrilateral and Hexahedral Meshes**
This paper presents an alternative to the Laplacian smoothing. The new smoothing is called the parallelogram smoothing. Parallelogram smoothing tries to fit a domain with the best possible parallelograms or parallelepipeds. Since many numerical

methods in porous media flow such as the well known MPFA produces a symmetric system on parallelogram meshes (cf. [2]). So, the parallelogram smoothing can be useful for porous media flow simulations. Error of streamline methods on parallelogram and parallelopiped mesh is minimum (see the Figures 7, 11 and 15 in [68]). So, it can be useful in streamline simulation. This paper has been submitted in the Journal of Computing Letters. A smaller version of the paper has been published in the proceedings of the International Conference of Numerical Analysis and Applied Mathematics (ICNAAM'05). The paper has been subjected to referee procedure.

Paper B: **Hexahedral Mesh by Area Functional**
We review the Area functional for generating hexahedral meshes. An algorithm for optimization of the area functional is presented. Since a global optimization can be computationally expensive, it is shown that such an optimization can be applied locally. The paper has been published in the proceedings of the International Conference of Numerical Analysis and Applied Mathematics (ICNAAM'05). The paper has been subjected to referee procedure.

Paper C: **An Effective Quadrilateral Mesh Adaptation**
Paper is about generating adaptive quadrilateral meshes. We present an extension of the Area functional for generating adaptive meshes. Several numerical examples are reported for showing effectiveness of the functional. Generally for quadrilateral mesh adaptation, we solve a coupled system of non-linear partial differential equations such as the well known non-linear elliptic system. Presented new idea is simple and computationally efficient. The other big plus of the method is that even after generating the solution adapted grid, the cells remain convex. The paper has been submitted in the Journal of Applied Mathematics Letters. A smaller version of the paper has been published in the proceedings of the Third MIT Conference, MIT, USA. The paper was subjected to referee procedure. For this work, the author was awarded the Young Researcher Fellowship by the MIT.

Paper D: **Deposition of Green House Gases by Compositional Simulator: Long Term Reactive Transport of CO2 in the Sand of Utsira**
In this work, we present the mathematical models and their discretization for capturing major physical processes associated with CO_2 sequestration/deposition in a porous medium. We verify our simulator by comparing our results against available results.

We are also simulating two scenarios with and without regional flow. We analyze impact of fluid movement on long term CO_2 migration at the Utsira. Here, we analyze how flow of medium fluids affects important parameters such as the pH and evolution of CO_2 saturation. The input data for the simulations has similarity with the CO_2 storage facility at the Sleipner Vest field in the Norwegian sector of the North sea. For example, the injection rate, injection period, properties of sand and shale layers. My contributions to the paper include mesh generation in the Utsira formation, development of geometrical and lithological models, coupling of the Accrete and Athena code. The author has also written the article. The work is done under the guidance of Gunnar Fladmark. The article has been submitted in the Journal of Transport in Porous Media.

Paper E: **Control Volume Finite Difference On Adaptive Meshes**

It is shown that discrete system formed on the adaptive meshes is not only more accurate but are also well conditioned compared to the one formed on the uniform meshes. Paper has been presented and submitted at the 16th International Conference on Domain Decomposition Methods. The author has performed the numerical work reported in the article. It will be subjected to referee procedure. The article is also available at "http://www.cims.nyu.edu/dd16/proceedings/khattri_contrib.pdf".

Paper F: **Grid Generation and Adaptation by Functionals**

Paper reviews various functionals for grid generation and adaptation. It is a well known fact that accuracy of a numerical simulation and quality of the grid are strongly related. Here quality means orthogonality at the boundaries and quasi-orthogonality within the critical regions, smoothness, bounded aspect ratios, solution adaptive behaviour, etc. In this article, we review various functionals for generating high quality structured quadrilateral meshes in two dimensional domains. Analysis of Winslow and Modified Liao functionals are presented. Numerical experiments are also reported to support our theoretical analysis. We demonstrate use of the Area functional for generating adaptive quadrilateral meshes. A short version of this paper has been published in the proceedings of the International Conference of Numerical Analysis and Applied Mathematics (ICNAAM'05). The paper was subjected to referee procedure. This extended version has been submitted in the ICNAAM Journal named "Mathematical Methods in the Applied Sciences (Wiley & Sons) ".

Paper G: **CO_2 storage in the Utsira Formation-ATHENA 3D reactive transport simulations**

Article presents 3D simulation of CO_2 sequestration/deposition at the Utsira formation. Our model consists of fourteen chemical and sixteen mineral species. We present 1000 years simulation of CO_2 deposition.

In this work, the author prepared the geometrical model of the Utsira formation. He coupled the ACCRETE geochemistry module and the Athena simulator. The author performed all the simulations. For this work, the author was guided by Gunnar Fladmark. The paper has been submitted in the Journal of Basin Modelling.

Paper H: **Numerical convergence on adaptive grids for control volume methods**

The article presents convergence of the Finite Volume Method on uniform and adaptive meshes. We also analyse convergence of the method in various norms (L_2 convergence for pressure and Darcy velocity and L_∞ convergence for pressure) on uniform meshes for problems with regularity $\mathbf{H}^{1+\gamma}$, for $\gamma = 0.1, 0.2, \ldots, 0.9$. For numerical work, we consider mixed boundary (Neumann + Dirichlet). The author did the numerical work. For implementing Multi Point Flux on adaptive meshes, the author was guided by Ivar Aavatsmark. The article has been submitted in the Journal of Numerical Methods for Partial Differential Equations.

TABLE OF CONTENTS

I Background Material 1

1 Introduction . . . 3

2 Grid Generation in Geological Formations . . . 5

2.1 Algebraic Method of Grid Generation . . . 5
2.2 One Dimensional Projection Operators . . . 5
2.3 Properties of Projection Operators . . . 8
2.3.1 Tensor Product . . . 8
2.3.2 Boolean Sum . . . 9
2.3.3 Transfinite Interpolation . . . 10
2.3.4 RMSGG: A multi-block algebraic grid generator . . . 10
2.3.5 Example 1: Algebraic Grid Generation . . . 11
2.3.6 Example 2: Lithology Implementation . . . 11
2.4 A Note About Trilinear Mapping . . . 13
2.4.1 Properties of Trilinear Map . . . 14
2.4.1.1 Covariant and Contravariant Vectors . . . 14
2.4.1.2 Jacobian . . . 15
2.4.2 ***G***-Tensor . . . 16
2.5 Elliptic Grid Generation . . . 16
2.6 C^{++} Implementation . . . 18
2.6.1 Numerical Examples : Elliptic Grid Generation . . . 19
2.6.1.1 Example 1 . . . 19
2.6.1.2 Example 2 . . . 20
2.6.1.3 Example 3 . . . 20
2.6.1.4 Example 4 . . . 20
2.7 Grid Generation and Simulations in Geological Formations . . . 22

3 Flow of CO_2 in Porous Media . . . 23

3.1 The Athena . . . 24
3.2 Continuum Equations of CO_2 Flow in Porous Medium . . . 24
3.2.1 Mass Conservation Equation . . . 25
3.2.2 Water Pressure Equation . . . 26
3.2.3 Temperature Equation . . . 26
3.3 Discretization of the Continuum Equations . . . 27
3.3.1 Discretization of Molar Mass Equation . . . 27
3.3.2 Discretization of Temperature Equation . . . 29
3.3.3 Discretization of the Water Pressure Equation . . . 30
3.4 Thermodynamics of Phase Equilibria In Athena . . . 32
3.4.1 Single Phase System . . . 33

3.4.2 Two Phase System 33
3.5 Simulator Verification 34
3.6 Simulation on the Utsira Formation 35
3.6.1 Simulation when the shale layers are flat. 37
3.7 Geochemical Transport with Athena 37
3.8 Modelling of Hydrate Processes with Athena (Non-Volume Method) 42

4 Finite Volume Method on Adaptive Meshes 45

4.1 Two Point Finite Volume Discretization 45
4.1.1 Implementation of Boundary Conditions 47
4.2 Adaptive Criteria and Adaptive Algorithm 48
4.3 Numerical Examples 51
4.3.1 Robustness of the adaptivity index η 54
4.3.2 The δ Effect 54
4.4 Which Meshes are Better Conditioned; Adaptive, Uniform, Locally Refined or Locally Adjusted ? 55
4.5 Converting Athena into an Adaptive Porous Media Flow Simulator 58

Bibliography 70

II Published and Submitted Work 71

A A New Smoothing Algorithm for Quadrilateral and Hexahedral Meshes . . .

B Hexahedral Mesh By Area Functional . . . .

C An Effective Quadrilateral Mesh Adaptation . . . .

D Deposition of Green House Gases by Compositional Simulator: Long Term Reactive Transport of CO_2 in the Sand of Utsira . . . .

E Control Volume Finite Difference on Adaptive Meshes . . . .

F Grid Generation and Adaptation by Functionals . . . .

G CO_2 storage in the Utsira Formation-ATHENA 3-D reactive transport simulations . . . .

H Numerical convergence on adaptive grids for control volume methods . . . .

Part I

Background Material

Chapter 1
INTRODUCTION

Problems involving multiphase flow, multicomponent transport and heat transfer in heterogeneous porous media have wider practical applications. Examples are deposition/sequestration of green-house gases, hydrocarbon flow, ground-water contamination and remediation. Such problems can have singular or localised solutions. Although contemporary computers are growing ever more powerful and capable of describing flows with increasing precision, it is still very difficult to capture problems with singular behaviour. Due to computational efficiency and accuracy, special solution techniques, such as adaptive, locally refined and locally adjusted meshes are preferred over uniform meshes for capturing singular or localised solutions. Chapter 4 and Articles E, H extend Finite Volume Method to adaptive meshes. The presented technique can be implemented in existing simulators. Such an implementation can improve simulation capabilities of the simulator. This work is concerned with developing new numerical tools or extending existing simulation tools for understanding reactive transport in porous media. We study deposition of CO_2 in deep saline aquifers. See Chapters 3 and Articles D, G. Largescale injection of CO_2 into aquifers induces a variety of coupled physical and chemical processes. It involves two phases (CO_2 and brine), and there exists mass transfer inbetween the phases. CO_2 deposition is a multicomponent transport with mineral reactions. Mineral reactions can change the medium properties by precipitation and dissolution in porous rocks, and it can significantly modify the physical (porosity) and chemical properties (pH) of the porous medium. Deposition of CO_2 can be described by a coupled system of partial differential equations (PDE). See Chapter 3 and Article D. Roughly speaking, numerically solving these PDEs in a domain such as the Utsira formation is a three step process. The first important step is dividing the given domain into smaller elements. This dividing process is called mesh generation in the domain. See Chapter 1 and Articles A, B, C, F. The second step is the discretization of the PDE (by methods such as Finite Volume and Finite Element) on the generated mesh. See Chapters 3 and 4. The final step is solving the formed system of equations. See Chapter 4 and Article H.

The main contributions of this research work are the following

1: - A software package is developed for multi-block grid generation. It is called RMS-GG. RMS-GG contains algorithms for accurate description of geometrical and lithological features of geological formations. The package can also model different boundary conditions. It can import geometrical and lithological models from the commercial packages such as the IRAP-RMS and Eclipse into the in-house research simulator Athena. RMS-GG is used for mesh generation in the Utsira formation. See Papers A, B, C, D, F, G and Chapters 2, 3.

2: - The existing simulator Athena is extended for simulating reactive transport. 3D simulations on the Utsira formation is performed by taking into account mineral and chemical reactions. See Papers D, G and Chapter 3.

3: - Software verification is an important and difficult problem. The author verified the Athena simulation tool for simulating CO_2 deposition. See Paper D and Chapter 3.

4: - The author extended the Two-Point Finite Volume Method to adaptive meshes. For this purpose, the author contributed in the form of an adaptive algorithm and an adaptivity criterion. This scheme can be implemented in an existing simulator. See Chapter 4 and Papers E, H.

Chapter 2
GRID GENERATION IN GEOLOGICAL FORMATIONS

Discretization of partial differential equations (PDEs) by numerical methods such as Finite Elements, Finite Volumes or Finite Differences requires discretization of the domain of interest into suitable (convex, linear) smaller elements. Depending on the method of discretization, elements are called finite volumes or finite elements. These elements in 3D can be hexahedras, tetrahedras, prisms and pyramids where as in 2D these elements can be triangles and quadrilaterals. Process of dividing a domain into smaller elements is called mesh generation. In this chapter, algebraic and elliptic grid generation methods are presented. Both methods produce structured grids. In structured meshes, every interior node is connected to the same number of neighbouring nodes.

2.1 Algebraic Method of Grid Generation

In the algebraic method of grid generation, we seek an algebraic mapping from a cube in computational or reference space to a physical space with the corresponding boundary surfaces [78; 81]. Transfinite interpolation (TFI) is such an algebraic mapping. TFI is also referred to as multivariate interpolation or Coons Patch. Figure 2.1 shows a mapping from a unit cube in the reference space onto a physical domain. Let the reference or computational space be defined by ξ, η and κ coordinates, and the physical space be defined by x, y and z coordinates. Suppose there exists a transformation or mapping, $\mathbf{r} = \mathbf{r}(\xi, \eta, \kappa)$, which maps the unit cube onto the interior of the physical domain, and this mapping maps the boundary surfaces of the cube to the corresponding boundary surfaces of the physical domain. Thus, $\eta = 1$ surface of the cube is mapped to the $\mathbf{r}(1, \eta, \kappa)$ boundary surface of the physical domain. Transfinite interpolation is the boolean sum of univariate interpolations in each of the computational coordinates. Univariate interpolations are also referred to as one dimensional projection operators or projectors. Boolean sum of the projection operators are defined below. A univariate interpolation is an operator that vary only in one dimension or roughly speaking it is a function of only one reference coordinate. A univariate interpolation can be linear, quadratic and cubic. Any univariate interpolation can be applied in a coordinate direction. Generally a higher order interpolation operator is desired in flow direction. TFI is composed of 1D projection operators, let us first define some one dimensional projection operators.

2.2 One Dimensional Projection Operators

A 1D projection operator or projector can be defined in many ways depending upon the available information. For example, a linear projector can be formed from two surfaces; a Hermite projector can be formed from two surfaces and directional derivatives at these surfaces; a Lagrangian projector can be defined from two boundary surfaces and internal surfaces.

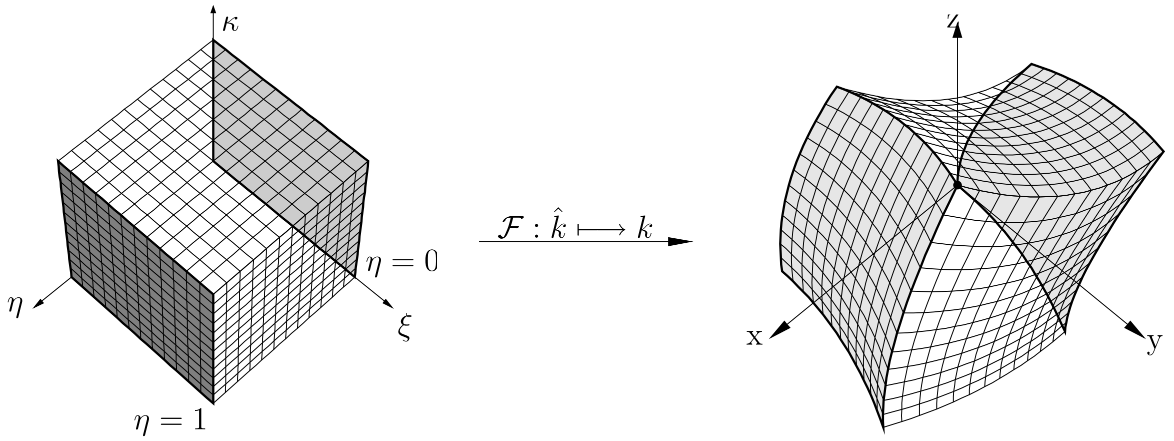


Figure 2.1: Mapping a unit cube onto a physical domain.

Let the reference space be defined by ξ, η and κ coordinates ($\xi \in [0,1]$, $\eta \in [0,1]$ and $\kappa \in [0,1]$). Suppose there exists a transformation $\mathbf{r}(\xi,\eta,\kappa)$ from a unit cube in the reference space onto a physical domain. That is $\mathbf{r}\colon \hat{k} \longmapsto k$. Let the physical space be defined by six boundary surfaces. A ξ surface in the physical space is a surface on which value of ξ is constant. Thus, two ξ boundary surfaces are $\mathbf{r}(0,\eta,\kappa)$ and $\mathbf{r}(1,\eta,\kappa)$. Similarly, two η and κ boundary surfaces are given as $\mathbf{r}(\xi,0,\kappa)$, $\mathbf{r}(\xi,1,\kappa)$ and $\mathbf{r}(\xi,\eta,0)$, $\mathbf{r}(\xi,\eta,1)$, respectively. From these six boundary surfaces, the following 1D projection operators are defined

$$\mathbf{P}_\xi \overset{\textbf{def}}{=} (1-\xi)\,\mathbf{r}(0,\eta,\kappa) + \xi\,\mathbf{r}(1,\eta,\kappa) \ , \tag{2.1}$$

$$\mathbf{P}_\eta \overset{\textbf{def}}{=} (1-\eta)\,\mathbf{r}(\xi,0,\kappa) + \eta\,\mathbf{r}(\xi,1,\kappa) \ , \tag{2.2}$$

$$\mathbf{P}_\kappa \overset{\textbf{def}}{=} (1-\kappa)\,\mathbf{r}(\xi,\eta,0) + \kappa\,\mathbf{r}(\xi,\eta,1) \ . \tag{2.3}$$

The projectors $\mathbf{P}_\xi$, $\mathbf{P}_\eta$ and $\mathbf{P}_\kappa$ are 1D projection operators and they are functions of the coordinates (ξ,η,κ). The projection operators defined by equations (2.1), (2.2) and (2.3) are linear in ξ, η and κ coordinates. It can be notice that the operators are defined from two surfaces of a particular kind. For example, $\mathbf{P}_\xi$ is defined from two ξ boundary surfaces in the physical space $\mathbf{r}(0,\eta,\kappa)$ and $\mathbf{r}(1,\eta,\kappa)$.

If in addition to the boundary surfaces we also know the internal surfaces of a domain then a projection operator can also be defined from more than two surfaces of a kind. For example, if there are $n+1$ surfaces of ξ type ($n-1$ internal curves and 2 boundary surfaces) then $\mathbf{P}_\xi$ projection operator can be defined as

$$\mathbf{P}_\xi \overset{\textbf{def}}{=} \sum_{j=0}^{n} \beta_j(\xi)\,\mathbf{r}(\xi_j,\eta,\kappa) \ , \tag{2.4}$$

[14; 72]. Here, $j = 0$ and $j = n$ are the boundary surfaces while $j = 1, \ldots, n-1$ are the internal surfaces, and β_j is the Lagrangian weighting factor. The Lagrangian weighting factor is given as follows

$$\beta_j(\xi) = \prod_{i=0,\, i\neq j}^{n} \left(\frac{\xi - \xi_i}{\xi_j - \xi_i} \right) \ . \tag{2.5}$$

It can be notice that the weighting factor $\beta_j(\xi)$ is an order n polynomial having zeros at all of the surfaces except the jth surface. The Lagrangian weighting factor satisfies the following

$$\beta_j(\xi_i) = \begin{cases} 1 & \text{if } i = j \ , \\ 0 & \text{if } i \neq j \ , \end{cases} \quad \text{and} \quad \sum_{j=0}^{n} \beta_j = 1.0 \ . \tag{2.6}$$

Now let us express the Lagrangian projection operator in another form. The numerator in the Lagrange weighting factor (2.5) can be written as

$$\frac{[(\xi - \xi_0)\,(\xi - \xi_1)\, \cdots \,(\xi - \xi_n)]}{(\xi - \xi_j)} = \frac{\Omega}{(\xi - \xi_j)} \ , \tag{2.7}$$

[see 14]. Let us define the barycentric weights as [14]

$$\omega_j = \frac{1}{\prod_{i=0,\, i\neq j}^{n} (\xi_j - \xi_i)} \ . \tag{2.8}$$

Using equations (2.7) and (2.8), the Lagrangian projection operator (2.4) can also be written as [14]

$$\mathbf{P}_\xi \overset{\textbf{def}}{=} \Omega \sum_{j=0}^{n} \frac{\omega_j}{\xi - \xi_j}\, \mathbf{r}(\xi_j, \eta) \ . \tag{2.9}$$

In the grid generation literature, the equation (2.4) is used but the new form (2.9) is computationally more efficient [cf. 14]. Similarly, if in addition to the boundary surfaces we are also given the derivatives (direction vectors) on these boundary surfaces then we can define the Hermite interpolation operators. For example, if we are given two ξ surfaces: $\mathbf{r}(0, \eta, \kappa)$ and $\mathbf{r}(1, \eta, \kappa)$, and let the direction vectors on these surfaces be $\mathbf{r}'(0, \eta, \kappa)$ and $\mathbf{r}'(0, \eta, \kappa)$, respectively. Then, the 1D Hermite projection operator can be defined as

$$\begin{aligned} \mathbf{P}_\xi \overset{\textbf{def}}{=} (2\,\xi^3 - 3\,\xi^2 + 1)\,\mathbf{r}(0, \eta, \kappa) + (-2\,\xi^3 + 3\,\xi^2)\,\mathbf{r}(1, \eta, \kappa) \\ + (\xi^3 - 2\,\xi^2 + \xi)\,\mathbf{r}'(0, \eta, \kappa) + (\xi^3 - \xi^2)\,\mathbf{r}'(1, \eta, \kappa) \ . \end{aligned} \tag{2.10}$$

Hermite projectors are easy to implement and are powerful tools for grid generation. Grid lines can be made orthogonal by the proper choice of direction vectors. This may help in accurate modelling of boundary conditions.

Figure 2.2 shows a physical domain containing three ξ, three η and two κ surfaces. Since the domain contains three ξ surfaces, three η surfaces and two κ surfaces thus we

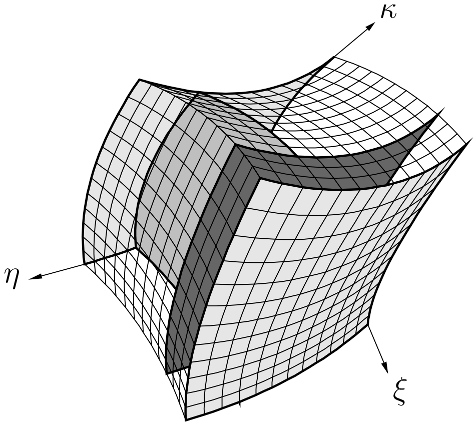


Figure 2.2: A 3D physical domain containing 3 ξ, 3 η and 2 κ surfaces.

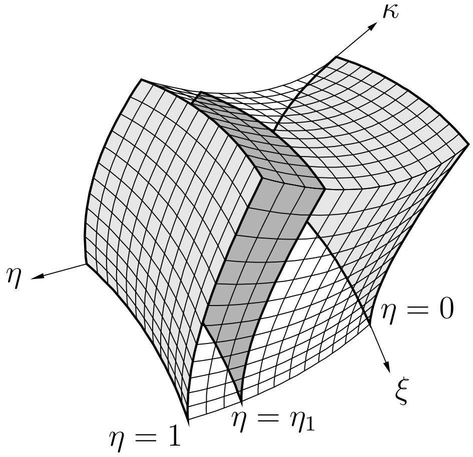


Figure 2.3: A 3D physical domain containing 2 ξ, 3 η and 2 κ surfaces.

can define a Lagrangian $\mathbf{P}_\xi$ operator, a Lagrangian $\mathbf{P}_\eta$ operator and a linear $\mathbf{P}_\kappa$ operator. Figure 2.3 shows another physical domain with two ξ, three η ($\mathbf{r}(\xi, 0, \kappa)$, $\mathbf{r}(\xi, \eta_1, \kappa)$ and $\mathbf{r}(\xi, 1, \kappa)$) and two κ surfaces. For this domain, a linear $\mathbf{P}_\xi$, a Lagrangian $\mathbf{P}_\eta$ and a linear $\mathbf{P}_\kappa$ operators can be defined. For this domain, the Lagrangian $\mathbf{P}_\eta$ operator is given as

$$\mathbf{P}_\eta = \Omega \left[\left(\frac{\omega_0}{\eta - 0} \right) \mathbf{r}(\xi, 0, \kappa) + \left(\frac{\omega_1}{\eta - \eta_1} \right) \mathbf{r}(\xi, \eta_1, \kappa) + \left(\frac{\omega_2}{\eta - 1} \right) \mathbf{r}(\xi, 1, \kappa) \right] \quad , \tag{2.11}$$

where Ω is given as,

$$\Omega = \eta \, (\eta - \eta_1) \, (\xi - \xi_3) \enspace ,$$

and ω_0, ω_1, ω_2 and ω_3 are given as,

$$\omega_0 = \frac{1}{\eta_1} \enspace , \qquad \omega_1 = \frac{1}{(-\eta_1)\,(1 - \eta_1)} \enspace , \qquad \omega_2 = \frac{1}{(-1)\,(\eta_1 - 1)} \enspace . \tag{2.12}$$

Now let us study two important and useful properties of projection operators. These properties are called tensor product and boolean sum of projection operators.

2.3 Properties of Projection Operators

This section presents two important properties of projection operators.

2.3.1 Tensor Product

Tensor product $\mathbf{P}_{\xi \circ \eta}$ of the projection operators $\mathbf{P}_\xi$ and $\mathbf{P}_\eta$ is defined as follows

$$\mathbf{P}_{\xi \circ \eta} \overset{\mathbf{def}}{=} \mathbf{P}_\xi \circ \mathbf{P}_\eta = (1 - \xi) \, [\mathbf{P}_\eta]_{\xi = 0} + \xi \, [\mathbf{P}_\eta]_{\xi = 1} \enspace . \tag{2.13}$$

Here, $\mathbf{P}_\xi$ is assumed to be linear projection operator as defined by the equation (2.1). It is clear from equation (2.13) that $\mathbf{P}_\xi$ is a projection operator. That is $\mathbf{P}_{\xi\circ\xi} = \mathbf{P}_\xi$. If $\mathbf{P}_\xi$ is Lagrangian projection operator then the tensor product is defined as

$$\mathbf{P}_{\xi\circ\eta} \overset{\mathbf{def}}{=} \mathbf{P}_\xi \circ \mathbf{P}_\eta = \sum_{j=0}^{n} \beta_j(\xi)\ [\mathbf{P}_\eta]_{\xi=\xi_j} \quad . \tag{2.14}$$

Tensor product of two projection operators is also a projection operator ($\mathbf{P}_{\xi\circ\eta}$ is a projection operator). Since tensor product is also a projection operator, it is commutative in nature. That is $\mathbf{P}_{\xi\circ\eta} = \mathbf{P}_{\eta\circ\xi}$. Similarly tensor products can be defined for an arbitrary number of projection operators. For example, the tensor product of three projection operators is defined as follows

$$\mathbf{P}_{\xi\circ\eta\circ\kappa} \overset{\mathbf{def}}{=} \mathbf{P}_\xi \circ (\mathbf{P}_\eta \circ \mathbf{P}_\kappa) = (1-\xi)\,[\mathbf{P}_{\eta\circ\kappa}]_{\xi=0} + \xi\,[\mathbf{P}_{\eta\circ\kappa}]_{\xi=1} \quad . \tag{2.15}$$

In the above equation, the projection operator $\mathbf{P}_\xi$ is linear.

2.3.2 Boolean Sum

Boolean sum of two projection operators is a also a projection operator and it is defined as follows

$$\mathbf{P}_{\xi\oplus\eta} \overset{\mathbf{def}}{=} \mathbf{P}_\xi \oplus \mathbf{P}_\eta = \mathbf{P}_\xi + \mathbf{P}_\eta - \mathbf{P}_{\xi\circ\eta} \quad . \tag{2.16}$$

Here, $\mathbf{P}_{\xi\circ\eta}$ is the tensor product of the $\mathbf{P}_\xi$ and $\mathbf{P}_\eta$ projection operators. Boolean sum is commutative in nature. That is $\mathbf{P}_\xi \oplus \mathbf{P}_\eta = \mathbf{P}_\eta \oplus \mathbf{P}_\xi$. Since boolean sum is also a projection operator thus it follows the projection property. That is $\mathbf{P}_{\xi\oplus\xi} = \mathbf{P}_\xi$. Similarly, the boolean sum can also be defined for an arbitrary number of projection operators. The boolean sum of three projectors is defined by using the fact that boolean sum and tensor product of two projection operators are also projection operators. Thus, the boolean sum of $\mathbf{P}_\xi$, $\mathbf{P}_\eta$ and $\mathbf{P}_\kappa$ operators is given as

$$\begin{aligned}
\mathbf{P}_{\xi\oplus\eta\oplus\kappa} &= \mathbf{P}_\xi \oplus \mathbf{P}_\eta \oplus \mathbf{P}_\kappa \quad , \\
&= \mathbf{P}_\xi \oplus (\mathbf{P}_\eta \oplus \mathbf{P}_\kappa) \quad , \\
&= \mathbf{P}_\xi \oplus (\mathbf{P}_\eta + \mathbf{P}_\kappa - \mathbf{P}_{\eta\circ\kappa}) \quad , \\
&= \mathbf{P}_\xi \oplus \mathbf{P}_\eta + \mathbf{P}_\xi \oplus \mathbf{P}_\kappa - \mathbf{P}_\xi \oplus \mathbf{P}_{\eta\circ\kappa} \quad , \\
&= \mathbf{P}_\xi + \mathbf{P}_\eta - \mathbf{P}_{\xi\circ\eta} + \mathbf{P}_\xi + \mathbf{P}_\kappa - \mathbf{P}_{\xi\circ\kappa} - \mathbf{P}_\xi - \mathbf{P}_{\eta\circ\kappa} + \mathbf{P}_{\xi\circ\eta\circ\kappa} \quad .
\end{aligned}$$

Thus,

$$\mathbf{P}_{\xi\oplus\eta\oplus\kappa} = \mathbf{P}_\xi + \mathbf{P}_\eta + \mathbf{P}_\kappa - \mathbf{P}_{\xi\circ\eta} - \mathbf{P}_{\xi\circ\kappa} - \mathbf{P}_{\eta\circ\kappa} + \mathbf{P}_{\xi\circ\eta\circ\kappa} \quad . \tag{2.17}$$

Here, $\mathbf{P}_{\xi\circ\eta}$ denotes the tensor product of $\mathbf{P}_\xi$ and $\mathbf{P}_\eta$ projection operators and $\mathbf{P}_{\xi\circ\eta\circ\kappa}$ denotes the tensor product of $\mathbf{P}_\xi$, $\mathbf{P}_\eta$ and $\mathbf{P}_\kappa$ projection operators.

2.3.3 Transfinite Interpolation

Boolean sum of projection operators is the basis for Transfinite Interpolation. TFI are used extensible for algebraic grid generation. 1D projection operators ($\mathbf{P}_\xi$, $\mathbf{P}_\eta$, $\mathbf{P}_\kappa$) comes in many flavours such as the Lagrangian and the Hermite, Transfinite Interpolation can be defined by many different expressions depending upon how the 1D projection operators are defined. Linear Transfinite Interpolation creates a grid in 3D or 2D using surfaces or curves that define the boundaries. The quality of the generated grid strongly depends on the parametrizations of the boundary curves. In its simplest form this mapping blends two given surfaces or curves to create a grid in the region bounded by the surfaces or curves. Linear Transfinite Interpolation mapping is defined from 4 curves in 2D or 6 surfaces in 3D. Transfinite Interpolation mapping will only give a reasonable grid if

i: - The surfaces that define the boundary match at the edges.

ii: - The surfaces are parametrized in the same direction. Otherwise grid lines could cross each other.

We are using the expression given by the equation (2.17) for mesh generation. The position vector in the physical space is given as

$$\mathbf{r}(\xi, \eta, \kappa) = \mathbf{P}_{\xi+\eta+\kappa} = \mathbf{P}_\xi \oplus \mathbf{P}_\eta \oplus \mathbf{P}_\kappa \tag{2.18}$$

It should be noted here that $\mathbf{P}_\xi$, $\mathbf{P}_\eta$ and $\mathbf{P}_\kappa$ are 1D projectors and they can be in the linear, Lagrangian or in the Hermite form. We have implemented the above form of TFI for algebraic grid generation in our package RMS-GG. The program can adaptively use various kind of 1D projectors. Since the boolean sum is a projection operator and TFI is also a boolean sum, TFI is also a projection operator.

If the surfaces of the geological formation are not smooth, then the grid generated with transfinite interpolation will not be smooth since Transfinite Interpolation propagate the surface smoothness property inside the domain. It is a well known fact that quality of grid (orthogonality of grid lines and convexity of cells) directly influences conditioning of discrete system and accuracy of the discrete solution [1; 57; 84; 85; 86]. There are various methods for improving quality of the generated meshes like the Jacobian conditioning and the Laplacian smoothing. The Jacobian conditioning [57; 78; 83; 84; 85; 86, and references therein] is robust but is computationally very expensive. The Laplacian smoothing technique locate nonsmooth locations to the average locations. Recently authors proposed an alternative to the Laplacian smoothing known as the parallelogram smoothing. It is presented in the paper A. The Parallelogram technique tries to fit the domain with the best possible parallelograms in 2-D and 3-D. Multi Point Flux Approximations (MPFA) are used for numerically capturing porous media flow. Since MPFA discretization on parallelogram grid results in a symmetric positive definite discrete system. The author believe that the parallelogram smoothing is especially applicable for porous media flow.

2.3.4 RMSGG: A multi-block algebraic grid generator

RMSGG is a multiblock grid generation package. It is written in the C^{++} language. Discretizing a complicated 3D domain into good quality hexahedral elements is a difficult work. There is no algorithm that guarantees generating good quality hexahedral elements

[29; 97]. Many geological formations and reservoirs of interest can be divided into layers based on the geological characteristics such as faults, pinchouts or the lithological properties such as shale and sandstone. Each of these layers can be meshed into hexahedrals by the algebraic methods independent of other layers. In this way grid distribution and quality of mesh can be improved and controlled in each of the layers separately. This technique is called the multilayer or the multiblock approach. This approach is similar to the concept of domain decomposition (DD) [111; 114] (divide a big problem into smaller ones and solve smaller problems on separate computers and glue the solutions together). The main reasons why we choose multiblock/multilayer structured grid generation approach for geological formations are the following

I: - Many geological formations can be handle by this concept.

II: - It makes parallelization of a single phase problem straight forward. The multiblock/multilayer concept used as a domain decomposition approach allows the direct parallelization of both grid generation and flow codes on massively parallel systems. The author wants to point out that parallelization of multiphase problems is very difficult compared to single phase problems.

III: - Grid density, distribution and quality can be controlled easily. It is desirable that in the areas of expected great nonlinear changes of solutions (around wells and material discontinuity) mesh should be refined.

IV: - Controllability over the simulation. For example, the implementation of lithology and local optimization of mesh.

V: - Though at the global level multilayer grids are unstructured in nature. Still at local level mesh can be expressed by logical numbering. Optimization of the quality of structured grids is easier. Instead of performing global mesh optimization, mesh can be optimized around critical locations such as wells. Structured grids can easily be made orthogonal at the boundaries and also almost orthogonal within the solution domain thus facilitating implementation of boundary conditions and also increase numerical accuracy. Discretization of PDEs on structured meshes is easier than on unstructured meshes.

VI: - A structured grid produces a structured matrix and thus makes it easier to use sophisticated linear solvers.

2.3.5 Example 1: Algebraic Grid Generation

We are using linear Transfinite Interpolation for grid generation. The geological formation shown in Figure 2.4 is divided into 9 layers based on the property of the medium. Four of these nine layers are are densly meshed.

2.3.6 Example 2: Lithology Implementation

For Finite Volume computation, lithology of the medium is associated with each finite volume. We have implemented algorithms in the RMSGG package for associating lithology with the cells. Figure 2.5 is an output of this package.

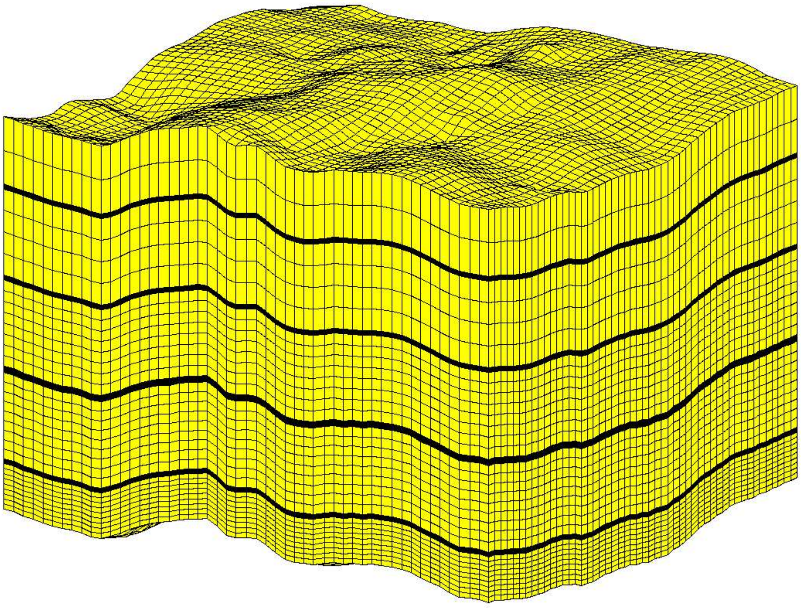


Figure 2.4: Multiblock grid generation in a layered geological formation.

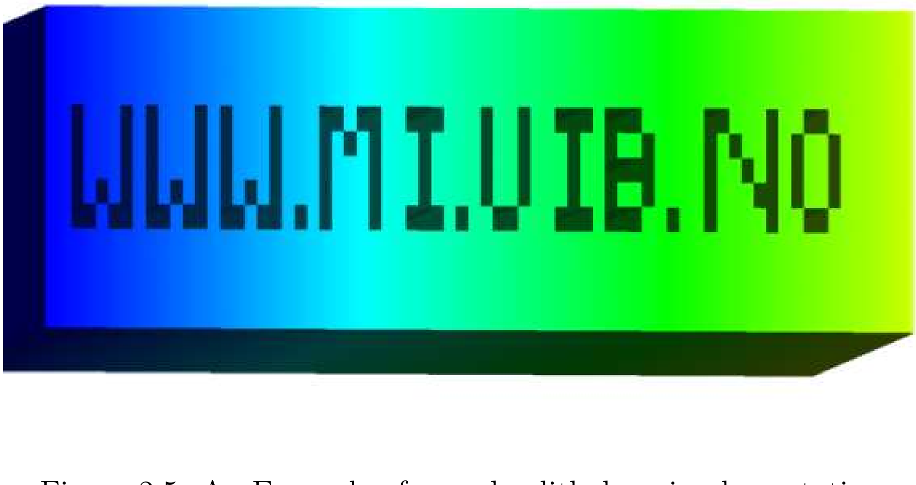


Figure 2.5: An Example of complex lithology implementation.

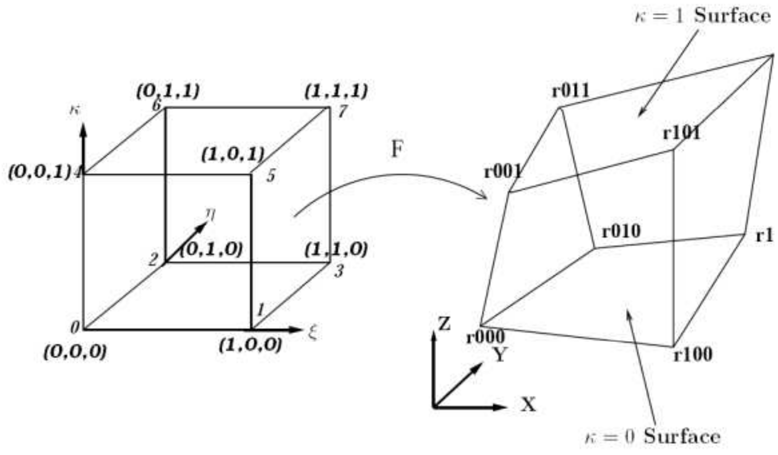


Figure 2.6: Trilinear mapping F_K from a reference unit cube ($\hat{K}$) on the left to the hexahedra (K) in the physical space.

2.4 A Note About Trilinear Mapping

A hexahedral element in the physical space can be expressed by a trilinear mapping from a unit cube in reference space as shown in the Figure 2.6. Trilinear mapping is an algebraic expression which expresses the position vectors in the physical space as a function of position vector in the reference space ($F_K(\xi, \eta, \kappa) \colon \hat{K} \longrightarrow K$). Trilinear mapping can be formulated from 1D projection operators.

Trilinear mapping is the tensor product of three one dimensional projectors given by equations (2.1), (2.2) and (2.3). Thus trilinear mapping is given as

$$\mathbf{P}_{\xi\eta\kappa} := \mathbf{P}_{\xi} \circ \mathbf{P}_{\eta} \circ \mathbf{P}_{\kappa} = (1-\xi)[\mathbf{P}_{\eta\kappa}]_{\xi=0} + \xi[\mathbf{P}_{\eta\kappa}]_{\xi=1}, \tag{2.19}$$

or

$$\begin{aligned}\mathbf{P}_{\xi\eta\kappa} = (1-\xi)(1-\eta)(1-\kappa)\,\mathbf{r}(0,0,0) + \xi\,(1-\eta)(1-\kappa)\,\mathbf{r}(1,0,0) \\ + (1-\xi)\,\eta\,(1-\kappa)\,\mathbf{r}(0,1,0) + (1-\xi)(1-\eta)\,\kappa\,\mathbf{r}(0,0,1) \\ + \xi\,\eta\,(1-\kappa)\,\mathbf{r}(1,1,0) + (1-\xi)\,\eta\,\kappa\,\mathbf{r}(0,1,1) \\ + \xi\,(1-\eta)\,\kappa\,\mathbf{r}(1,0,1) + \xi\,\eta\,\kappa\,\mathbf{r}(1,1,1),\end{aligned} \tag{2.20}$$

Let $F_K = \mathbf{P}_{\xi\eta\kappa}$. It is clear that $F_K \in Q_1^3$. If K is a convex hexahedra, there exists a unique invertible mapping $F_K \in Q_1^3$ that maps $\hat{K}$ into K. If F_K is one to one map then $K := F_K(\hat{K})$. The edges of a hexahedra are always straight segments, but each face is planar if and only if its four vertices lie in the same plane. Trilinear map can also expresses non planar hexahedra [98]. Replacing terms $\mathbf{r}(i,j,k)$ by $\mathbf{r}_{ijk}$ in the equation (2.19) reads as follows

$$\mathbf{r} = \mathbf{r}_{000} + \xi\,(\mathbf{r}_{100} - \mathbf{r}_{000}) + \eta\,(\mathbf{r}_{010} - \mathbf{r}_{000}) + \kappa\,(\mathbf{r}_{001} - \mathbf{r}_{000})$$

$$
\begin{aligned}
&+ \xi\,\eta\,(\mathbf{r}_{000} + \mathbf{r}_{110} - \mathbf{r}_{100} - \mathbf{r}_{010}) + \xi\,\kappa\,(\mathbf{r}_{000} + \mathbf{r}_{101} - \mathbf{r}_{100} - \mathbf{r}_{001}) \\
&\qquad + \eta\,\kappa\,(\mathbf{r}_{000} + \mathbf{r}_{011} - \mathbf{r}_{001} - \mathbf{r}_{010}) \\
&\quad + \xi\,\eta\,\kappa\,(\mathbf{r}_{100} + \mathbf{r}_{010} + \mathbf{r}_{001} + \mathbf{r}_{111} - \mathbf{r}_{000} - \mathbf{r}_{110} - \mathbf{r}_{101} - \mathbf{r}_{011}),
\end{aligned} \tag{2.21}
$$

or

$$
\mathbf{r} = \mathbf{c}_0 + \xi\,\mathbf{c}_\xi + \eta\,\mathbf{c}_\eta + \kappa\,\mathbf{c}_\kappa + \xi\,\eta\,\mathbf{c}_{\xi\eta} + \xi\,\kappa\,\mathbf{c}_{\xi\kappa} + \eta\,\kappa\,\mathbf{c}_{\eta\kappa} + \xi\,\eta\,\kappa\,\mathbf{c}_{\xi\eta\kappa}. \tag{2.22}
$$

Here:

$$
\begin{cases}
\mathbf{c}_{\xi\eta}: & \text{bottom surface;} \\
\mathbf{c}_{\xi\kappa}: & \text{front surface;} \\
\mathbf{c}_{\eta\kappa}: & \text{left surface.}
\end{cases}
$$

It can be easily seen that $\mathbf{c}_{\xi\eta}$ will be zero if the bottom surface of the hexahedra is a parallelogram. If any two pair (ξ or η or κ) of surfaces are parallelogram, the term $\mathbf{c}_{\xi\eta\kappa}$ will be zero.

Trilinear mapping $F_K(\hat{K})$ is widely used for discretizing PDEs [91] with methods such as Finite Volume, Finite Element and Finite Difference. One of the major problem in numerical computation is ensuring the invertibility of the mapping $F_K(\hat{K})$. Trilinear mapping $F_K(\hat{K})$ is not linear. The positivity of the Jacobian at the vertices does not ensure the global invertibility of $F_K(\hat{K})$ (or ensuring the invertibility of the mapping is computationally intensive). One encouraging fact about parallelepipeds is that the mapping $F_K(\hat{K})$ becomes linear. Thus parallelepipeds in the physical space can be expressed as: $K = C + \boldsymbol{J} \cdot \hat{K}$. Here $\boldsymbol{J}$ is the Jacobain matrix of $F_K(\hat{K})$. For a linear mapping, the Jacobian $\boldsymbol{J}$ is independent of (ξ, η, κ). Assuming that $\boldsymbol{J}$ is not singular and also that the element is not inverted (i.e., $\boldsymbol{J}$ is positive) then the mapping $F_K(\hat{K})$ is invertible everywhere in $\hat{K}$ and the inverse mapping is given as $\hat{K} = \boldsymbol{J}^{-1} \cdot K - \boldsymbol{J}^{-1} \cdot C$.

2.4.1 Properties of Trilinear Map

2.4.1.1 Covariant and Contravariant Vectors

The trilinear mapping is a function of ξ, η and κ. Differentiation of the mapping wrt. independent variables (ξ , η and κ) can be defined. Let us denote the three covariant base vectors at any point in the reference space as

$$
\mathbf{g}_1 = \frac{\partial \mathbf{r}}{\partial \xi}, \quad \mathbf{g}_2 = \frac{\partial \mathbf{r}}{\partial \eta} \quad \text{and} \quad \mathbf{g}_3 = \frac{\partial \mathbf{r}}{\partial \kappa}.
$$

It can be easily seen that

$$
\mathbf{g}_1 = \mathbf{c}_\xi + \eta\,\mathbf{c}_{\xi\eta} + \kappa\,\mathbf{c}_{\xi\kappa} + \eta\,\kappa\,\mathbf{c}_{\xi\eta\kappa}, \tag{2.23}
$$

$$
\mathbf{g}_2 = \mathbf{c}_\eta + \xi\,\mathbf{c}_{\xi\eta} + \kappa\,\mathbf{c}_{\xi\kappa} + \xi\,\kappa\,\mathbf{c}_{\xi\eta\kappa}, \tag{2.24}
$$

$$
\mathbf{g}_3 = \mathbf{c}_\kappa + \xi\,\mathbf{c}_{\eta\kappa} + \eta\,\mathbf{c}_{\eta\kappa} + \xi\,\eta\,\mathbf{c}_{\xi\eta\kappa}. \tag{2.25}
$$

The base vector $\mathbf{g}_1$ points in the direction of increasing ξ. The three covariant basis vectors $\{\mathbf{g}_1, \mathbf{g}_2, \mathbf{g}_3\}$ at any point (ξ, η, κ) form a triad. At any point (ξ, η, κ) the Jacobian (volume of the parallelepiped formed by the covariant base vectors $\mathbf{g}_i$) is given as

$$
J = \mathbf{g}_1 \cdot (\mathbf{g}_2 \times \mathbf{g}_3), \tag{2.26}
$$

Similarly at any point (ξ, η, κ) we can also define a dual basis called contravariant basis $\{\mathbf{g}^1, \mathbf{g}^2, \mathbf{g}^3\}$ as

$$\mathbf{g}^1 = \frac{1}{J}(\mathbf{g_2} \times \mathbf{g_3}), \quad \mathbf{g}^2 = \frac{1}{J}(\mathbf{g_3} \times \mathbf{g_1}), \quad \mathbf{g}^3 = \frac{1}{J}(\mathbf{g_1} \times \mathbf{g_2}). \tag{2.27}$$

The contravariant vector $\mathbf{g}^1$ is perpendicular to the plane defined by the covariant vectors $\mathbf{g}_2$ and $\mathbf{g}_3$. Finite Volume Methods require computation of fluxes through surfaces and the contra-variant vectors can be used for this. For example, $\mathbf{g}^1$ can be used for computing flux through a surface spanned by the covariant vectors $\mathbf{g}_2$ and $\mathbf{g}_3$. Covariant and contravariant vectors form a curvelinear basis. These are used for measuring distortion of hexahedral cells [16; 29; 57; 81; 85; 118; 127]. These quantities can be computed a priori. For example during grid generation.

2.4.1.2 Jacobian

Columns of the Jacobian matrix at a point are the covariant base vectors at that point. Any column of the Jacobian matrix measures sensitivity of the transformation towards a single independent variable and any row of the Jacobian matrix measures sensitivity of transformation from all of the independent variables. The Jacobian matrix of the transformation $F_K \colon \hat{K} \longrightarrow K$ is given as

$$\boldsymbol{J} := DF_K(\xi, \eta, \kappa) := \left[\frac{\partial \mathbf{r}}{\partial \xi} \quad \frac{\partial \mathbf{r}}{\partial \eta} \quad \frac{\partial \mathbf{r}}{\partial \kappa}\right], \tag{2.28}$$

or

$$\boldsymbol{J} := (DF_K)(\xi, \eta, \kappa) = \begin{pmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} & \frac{\partial x}{\partial \kappa} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} & \frac{\partial y}{\partial \kappa} \\ \frac{\partial z}{\partial \xi} & \frac{\partial z}{\partial \eta} & \frac{\partial z}{\partial \kappa} \end{pmatrix} = \begin{pmatrix} x_\xi & x_\eta & x_\kappa \\ y_\xi & y_\eta & y_\kappa \\ z_\xi & z_\eta & z_\kappa \end{pmatrix}. \tag{2.29}$$

For invertibility of the mapping F_k, the Jacobian matrix $\boldsymbol{J}$ is required to be strictly positive definite matrix (all eigenvalues and the determinant of $\boldsymbol{J}$ are greater than zero) at all points in the reference cube. The determinant of the Jacobian matrix (2.29) is given as

$$J = x_\xi\, y_\eta\, z_\kappa + y_\xi\, z_\eta\, x_\kappa + z_\xi\, x_\eta\, y_\kappa - z_\xi\, y_\eta\, x_\kappa - y_\xi\, x_\eta\, z_\kappa - x_\xi\, z_\eta\, y_\kappa. \tag{2.30}$$

It was found that quality of hexahedral cells is directly related to the condition number of the Jacobian matrix of the trilinear mapping (see [86, and references therein]). Knupp [57; 78; 79; 83; 84; 85; 86, and references therein] suggested that optimization of the condition number can remove distorted or singular elements (concave, badly shaped, invalid or folded) from a mesh. The condition number [57] of the Jacobian matrix (2.28) is given as

$$k = \frac{\left[|\mathbf{g}_1|^2 + |\mathbf{g}_2|^2 + |\mathbf{g}_3|^2\right]^{\frac{1}{2}} \left[|\mathbf{g}_1 \times \mathbf{g}_2|^2 + |\mathbf{g}_2 \times \mathbf{g}_3|^2 + |\mathbf{g}_3 \times \mathbf{g}_1|^2\right]^{\frac{1}{2}}}{|(\mathbf{g}_1 \times \mathbf{g}_2) \cdot \mathbf{g}_3|}. \tag{2.31}$$

Here $\mathbf{g}_1$, $\mathbf{g}_2$ and $\mathbf{g}_3$ are the covariant vectors.

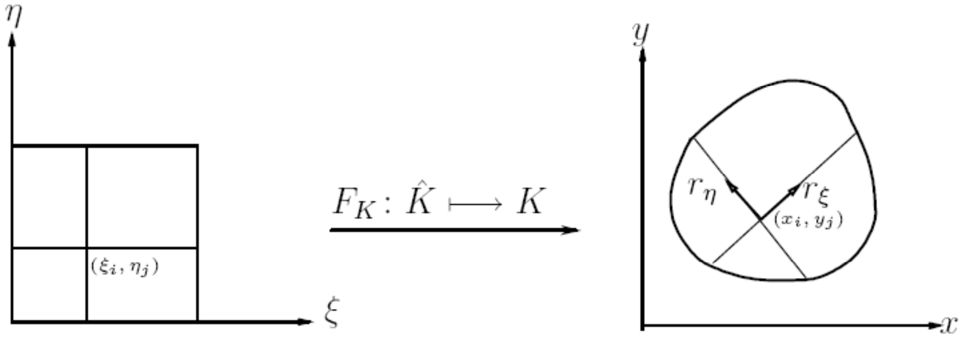


Figure 2.7: Mapping F_K from a reference unit square ($\hat{K}$) on the left to a physical domain (K).

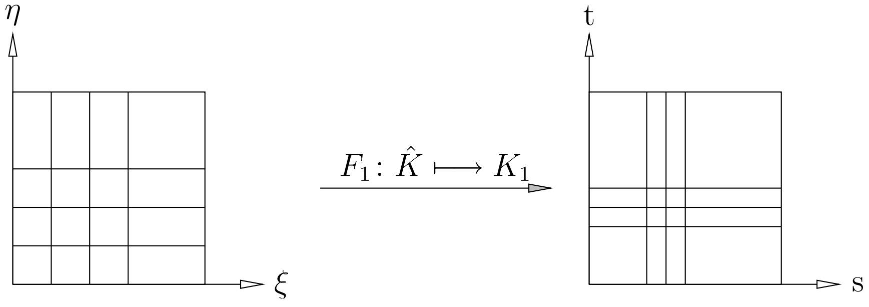


Figure 2.8: Mapping F_1 from a unit square ($\hat{K}$) in the reference space to a unit square in the parameter space (K_1).

2.4.2 *G*-Tensor

G-Tensor is defined as the product of transpose of Jacobian matrix with itself. It is always symmetric positive definite. It is strictly positive definite if the mapping F_k is invertible. Let us denote $g_{ij} = \mathbf{g_i} \cdot \mathbf{g_j}$ then the ***G***-tensor is defined as

$$\boldsymbol{G} := \boldsymbol{J}^t \boldsymbol{J} = \begin{pmatrix} g_{11} & g_{12} & g_{13} \\ g_{21} & g_{22} & g_{23} \\ g_{31} & g_{32} & g_{33} \end{pmatrix}. \tag{2.32}$$

The matrix ***G*** is diagonal at a point (ξ, η, κ) in the reference space if the co-variant vectors at this point are orthogonal to each other.

Similar properties can also be defined for any TFI mapping.

2.5 Elliptic Grid Generation

Apart from algebraic grid generation, the classical elliptic grid generation system are also extensively used [78; 78; 80; 81; 82; 84]. We present non-linear elliptic system for generating adaptive quadrilateral grids in curved domains. Elliptic grid generation system is known to produce smooth grids. For meshing a domain into non-simplex elements (quadrilaterals in 2D and hexahedrals in 3D), we generally seek a mapping from a reference square or cube to the physical domain. This mapping can be algebraic in nature like Transfinite Interpolation or it can be expressed by a system of nonlinear partial differential equations. The first study of elliptic grid generation was done by Winslow [123; 124]. For grid generation, it is required that the mapping is invertible (onto and one to one). Proving the invertibility of algebraic or differential mapping can be difficult and it depends on the boundary of the physical domain. We are looking for a vector mapping, $\mathcal{F}_K(\hat{K}) = (x, y)^t$, from a unit square in the reference space $(\hat{K})$ to the physical space K; i.e., $\mathcal{F}_K \colon \hat{K} \longmapsto K$. Figure 2.7 shows such a mapping. Mapping $\mathcal{F}_K$ gives the position of a point in the physical space corresponding to a point in the computational space (reference space). Let the physical space be given by the x and y coordinates and the computational space be given by the ξ and η coordinates. We are using the following elliptic system for defining the mapping $\mathcal{F}_K = (x, y)^t$

$$g_{22}\,\frac{\partial^2 x}{\partial \xi^2} - 2\,g_{12}\,\frac{\partial^2 x}{\partial \xi \partial \eta} + g_{11}\,\frac{\partial^2 x}{\partial \eta^2} + P\,x_\xi + Q\,x_\eta = 0, \tag{2.33}$$

$$g_{22}\,\frac{\partial^2 y}{\partial \xi^2} - 2\,g_{12}\,\frac{\partial^2 y}{\partial \xi \partial \eta} + g_{11}\,\frac{\partial^2 y}{\partial \eta^2} + P\,y_\xi + Q\,y_\eta = 0, \tag{2.34}$$

where terms P and Q are used for grid adaptation and are given as

$$P = g_{22}\,P^1_{11} - 2\,g_{12}\,P^1_{12} + g_{11}\,P^1_{22}, \tag{2.35}$$

$$Q = g_{22}\,P^2_{11} - 2\,g_{12}\,P^2_{12} + g_{11}\,P^2_{22}. \tag{2.36}$$

Equations (4.29)-(4.30) are non-linear and are coupled through the coefficients g_{ij} (coefficients of the metric tensor). Metric coefficients are given as

$$g_{11} = x_\xi^2 + y_\xi^2, \quad g_{22} = x_\eta^2 + y_\eta^2 \quad \text{and} \quad g_{12} = x_\xi\,x_\eta + y_\xi\,y_\eta. \tag{2.37}$$

For generating grids in the physical space, the elliptic system (4.29)-(4.30) is solved for the co-ordinates (x, y) on a unit square in the computational space by the method of Finite Differences. Boundary of the physical domain is specified as the Dirichlet boundary condition on the unit square in the computational space. In the Figure 2.7, $\mathbf{r}_\xi$ and $\mathbf{r}_\eta$ are the covariant base vectors at the point (x_i, y_j). Figure 2.9 shows a finite difference stencil around the point (ξ_i, η_j) in the computational space. A finite difference approximation of x_ξ and x_η is

$$x_\xi = \frac{x(i+1,j) - x(i-1,j)}{2\,\Delta\xi}, \quad x_\eta = \frac{x(i,j+1) - x(i,j-1)}{2\,\Delta\eta}.$$

Similary x_η and y_η can be defined. Here, we are assuming that the grid in the computational space is uniform. However, grid in the physical space can be compressed or stretched.

Terms P^k_{ij} ($i = 1{,}2$ and $j = 1{,}2$ and $k = 1{,}2$ and $P^k_{12} = P^k_{21}$) in the equations (4.29)-(4.30) are determined through another mapping $\mathcal{F}_1$. The mapping $\mathcal{F}_1$ is shown in the Figure

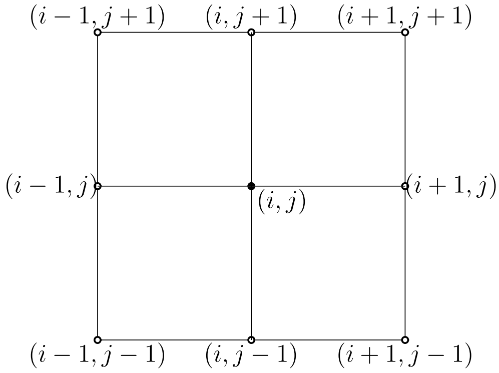


Figure 2.9: Finite difference stencil in the ξ-η computational space.

2.8. This mapping maps a unit square in the computational space to a unit square in the parameter space. For defining the mapping $\mathcal{F}_1 \colon \hat{K} \longrightarrow K_1$, the boundary and internal grid points of the parameter space are mapped to the reference space. The Jacobian matrix $\boldsymbol{T}$ (columns are the covariant vectors) of the mapping $\mathcal{F}_1$ and the vectors $\mathbf{P}_{11}$, $\mathbf{P}_{12}$ and $\mathbf{P}_{22}$ are given as follows

$$\boldsymbol{T} = \begin{pmatrix} s_\xi & s_\eta \\ t_\xi & t_\eta \end{pmatrix}, \qquad \mathbf{P}_{11} = -\boldsymbol{T}^{-1} \begin{pmatrix} s_{\xi\xi} \\ t_{\xi\xi} \end{pmatrix}, \tag{2.38}$$

$$\mathbf{P}_{22} = -\boldsymbol{T}^{-1} \begin{pmatrix} s_{\eta\eta} \\ t_{\xi\eta} \end{pmatrix}, \qquad \mathbf{P}_{12} = -\boldsymbol{T}^{-1} \begin{pmatrix} s_{\xi\eta} \\ t_{\xi\eta} \end{pmatrix}. \tag{2.39}$$

The terms P_{ij}^1 $(i,j = 1,2)$ are the first component of the vector $\mathbf{P}_{ij}$ and the terms P_{ij}^2 are the second component of the vector $\mathbf{P}_{ij}$. It should be noted that the vectors $\mathbf{P}_{11}$, $\mathbf{P}_{12}$ and $\mathbf{P}_{22}$ can be computed a priori for clustering the grid points. A second order finite difference approximation of different operators required for computing the vectors $\mathbf{P}_{11}$, $\mathbf{P}_{22}$, $\mathbf{P}_{12}$ and the Jacobian $\boldsymbol{T}$ are given in the Table 2.1. We are using the stencil shown in the Figure 2.9.

2.6 C^{++} Implementation

We have implemented the presented technique in the C^{++} language for generating adaptive grids. It is freely available at `www.mi.uib.no/∼sanjay`. The package can write meshes in the Matlab and GMV [101] formats. It consists of one Domain class. This class is used for defining the unit square in the computational space, the unit square in the parameter space and the physical domain. For clustering grids in the parameter space different functions are defined in the domain class. The algorithm consists of two loops outer loop and inner loop. The outer loop is controlled by the maximum number of SOR iterations and a given tolerance. Each iteration of an inner loop provides a new mesh by the SOR relaxation. The algorithm proceeds as follows. Generate grids in the computational and parameter spaces. Compute the matrix $\boldsymbol{T}$ and vectors $\mathbf{P}_{ij}$ for defining the mapping $\mathcal{F}_1$.

Table 2.1: Finite difference approximation of continuous operators.

$$s_\xi = \frac{s(i+1,j) - s(i-1,j)}{2\,\Delta\xi} \;, \quad s_{\xi\xi} = \frac{s(i+1,j) - 2\,s(i,j) + s(i-1,j)}{\Delta\xi^2}$$

$$t_\xi = \frac{t(i,j+1) - t(i,j-1)}{2\,\Delta\eta} \;, \quad t_{\eta\eta} = \frac{t(i,j+1) - 2\,t(i,j) + t(i,j-1)}{\Delta\eta^2}$$

$$s_{\eta\eta} = \frac{s(i,j+1) - 2\,s(i,j) + s(i,j-1)}{\Delta\eta^2} \;, \quad t_{\xi\xi} = \frac{t(i+1,j) - 2\,t(i,j) + t(i-1,j)}{\Delta\xi^2}$$

$$s_{\xi\eta} = \frac{s(i+1,j+1) + s(i-1,j-1) - s(i-1,j+1) - s(i+1,j-1)}{4\,\Delta\xi\Delta\eta}$$

$$t_{\xi\eta} = \frac{t(i+1,j+1) + t(i-1,j-1) - t(i-1,j+1) - t(i+1,j-1)}{4\,\Delta\xi\Delta\eta}$$

An initial grid, $\mathbf{r}_{\text{old}}$, on the physical region is generated (say by Transfinite Interpolation). The first step of the outer loop is to generate an initial grid, $\mathbf{r}_{old}$, in the physical space. The coupled elliptic system are linearized by the method of finite difference and the resulting system is solved by the SOR relaxation. Each iteration of SOR provide a new mesh, $\mathbf{r}_{\text{new}}$. The next iteration of outer loop starts if the difference in the old and new mesh ($\|\mathbf{r}_{\text{new}} - \mathbf{r}_{\text{old}}\|$) is greater than the tolerance or the iteration of the outer loop is less than the maximum iterations. During the next iteration of the outer loop the old mesh is replaced by the newly formed mesh; i.e., $\mathbf{r}_{\text{old}} = \mathbf{r}_{\text{new}}$. The outer loop is exited if the difference between the old and new values is less than a given tolerance.

2.6.1 Numerical Examples : Elliptic Grid Generation

2.6.1.1 Example 1

In this example, $P = 0$ and $Q = 0$ in the equations (4.29) and (4.30). The equations are still non-linear and coupled through the coefficients g_{ij}. The resulting system is called the Winslow equations. One inherent property of the elliptic grid generators is that the grid generated is always smooth; i.e., discontinuities in slope at the boundaries are not propagated into the interior of the domain. Algebraic grid generators do propagate the boundary discontinuity into the interior of the domain. In this example, we compare the algebraic grids with elliptic grids. Figure 2.11 is the grid by the elliptic system. Figure 2.10 shows the grid generated by the transfinite interpolation. It is clear from Figure 2.10 that transfinite interpolation can generate folded grid for curved boundaries. For the elliptic grid generator, we assumed initial position vector of internal nodes is zero vector.

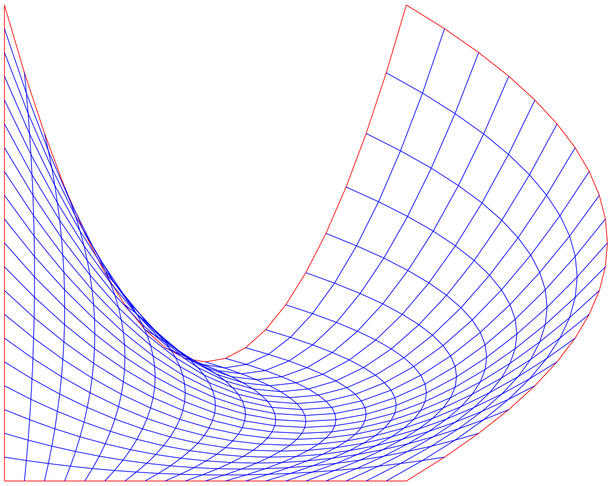


Figure 2.10: Folded Grid by Transfinite Interpolation.

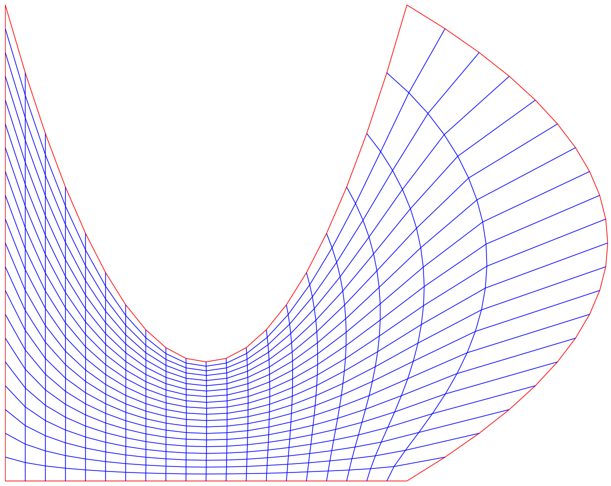


Figure 2.11: Smooth grid by elliptic system.

2.6.1.2 Example 2

Figure 2.13 shows the grid in the parameter space for concentrating grids at the center of the physical space. Parameter space is used for computing P and Q in the elliptic grid generation system. Density of the grid in the physical space is determined by the grid density in the parameter space. Figure 2.14 shows the converged grid in the physical space.

2.6.1.3 Example 3

Figure 2.12 is the initial grid in the physical space. Figure 2.15 is the grid in the parameter space for concentrating grids at the boundary of the physical domain. Adaptive grid in the physical space is shown in the Figure 2.16.

2.6.1.4 Example 4

Figure 2.12 is the initial grid in the physical space. Figure 2.17 presents grid in the parameter space. The final adaptive grid is shown in the Figure 2.18.

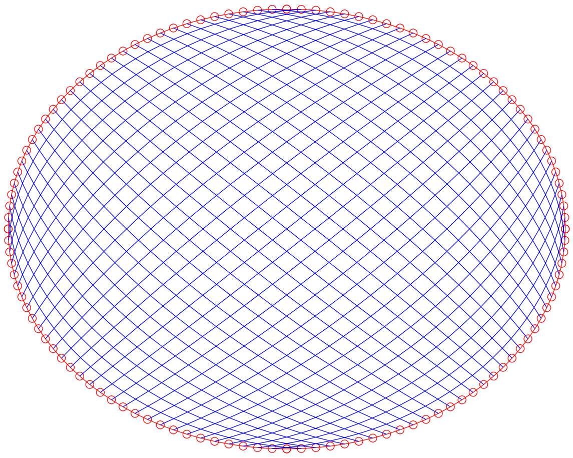


Figure 2.12: Initial grid in the physical space.

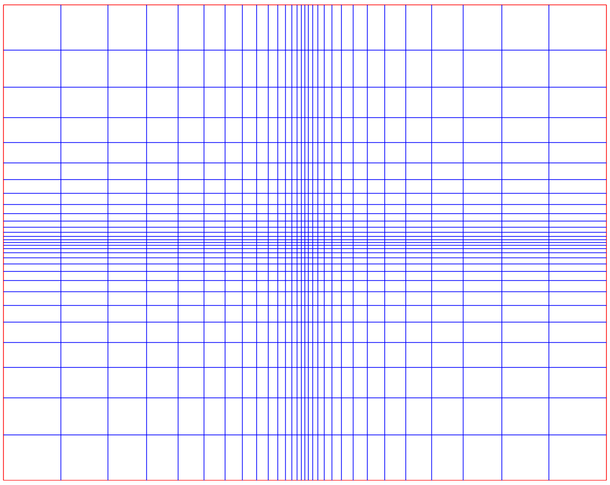


Figure 2.13: Grid in the parameter space.

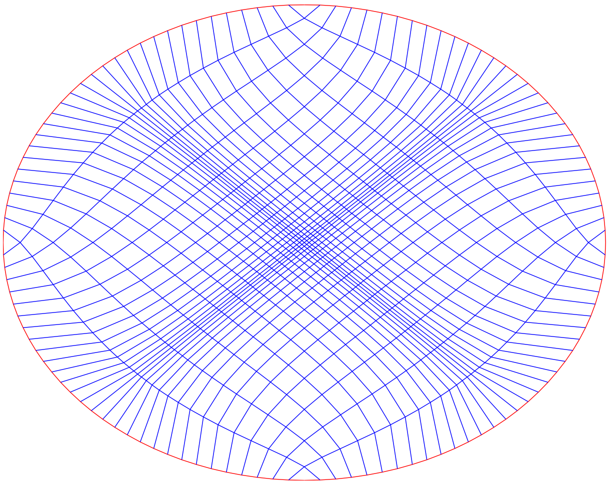


Figure 2.14: Grid by elliptic system.

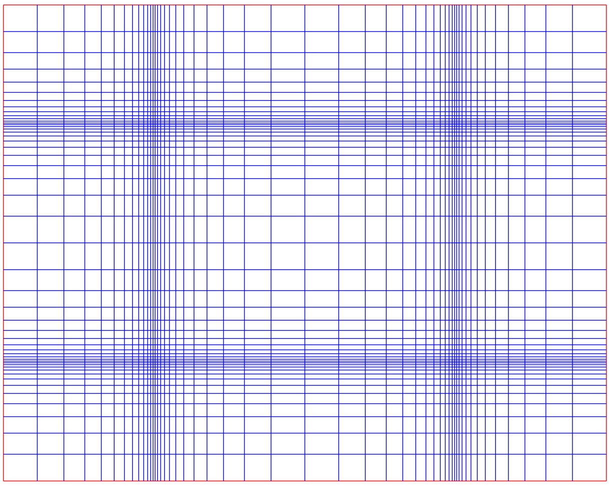


Figure 2.15: Grid in the parameter space.

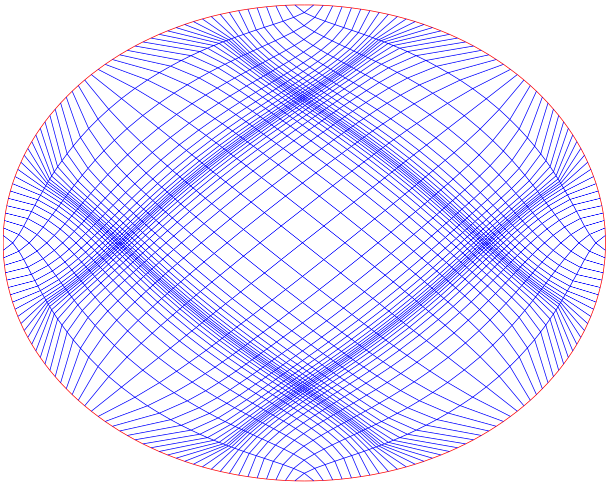


Figure 2.16: Grid by elliptic system.

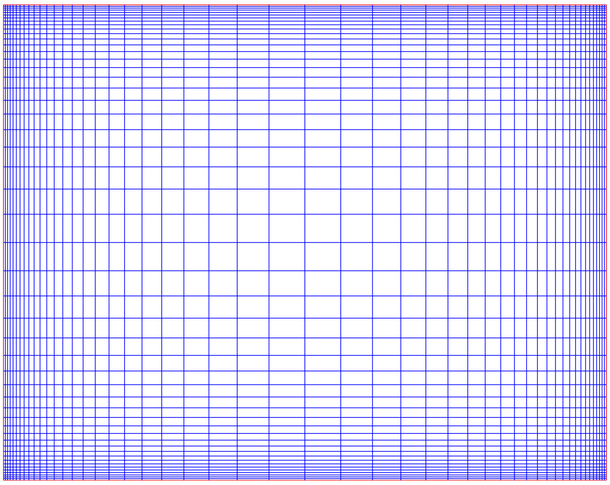


Figure 2.17: Grid in the parameter space.

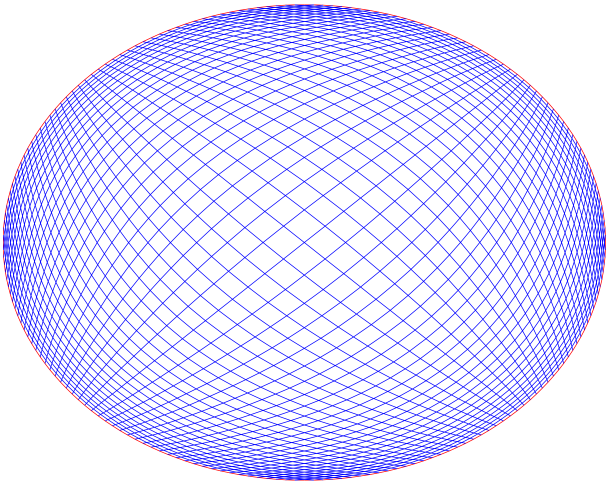


Figure 2.18: Grid by elliptic system.

The Article C presents a simple technique for generating adaptive quadrilateral meshes. Paper F reviews functionals for grid generation and adaptation.

2.7 Grid Generation and Simulations in Geological Formations

Grid generation is the first important step in performing a successful simulation. For the reasons of simplicity, let us consider the single phase pressure equation. Discretization of the single phase pressure equation $\operatorname{div}(-\boldsymbol{K}\operatorname{grad}p) = q$ on a grid results in a discrete system $\boldsymbol{A}\,\mathbf{p}_h = \mathbf{b}$. Here the operator $\boldsymbol{A}$ consists of the following information

1. Grid.
2. Medium property $\boldsymbol{K}$.
3. Dirichlet boundary conditions.

The right hand vector $\mathbf{b}$ consists the following information

1. Sources such as wells.
2. Flux and Dirichlet boundary conditions.

Ensuring monotonicity of the operator $\boldsymbol{A}$ is a big concern in porous media flow. For example, a non-monotone operator can produce un-physical results and it can also retard the convergence of the Newton's iteration.

Many discretization schemes have been proposed in the literature for improving monotone behaviour of the operator $\boldsymbol{A}$. Implementing a new discretization scheme in an existing porous media simulator is a difficult task compared to improving the quality of the underlying grid. Roughly speaking, problems such as non-monotone behaviour of a discretization scheme are caused by the underlying grid. Instead of using a new discretization technique for a new grid. Improving the quality of the underlying grid can improve the monotone characteristic of a discretization scheme.

Chapter 3
FLOW OF CO_2 IN POROUS MEDIA

Today many experts agree that emission of greenhouse gases into the atmosphere is causing the global warming. Carbon dioxide (CO_2) is one of the greenhouse gases and is considered by far mostly responsible for global warming. Instead of releasing CO_2 into the atmosphere, CO_2 can be captured and stored in underground geological formations. CO_2 can be sequestrated in depleted oil and gas fields, coal mines and deep saline aquifers as shown in Figure 3.1. In Table 3.1, estimates of storage capacities for different geological formations are presented [58].

Sequestration of CO_2 in porous media is a multi-component, multi-phase, reactive transport; moreover reactive transport of multi-component and multi-phase system in a porous medium is a mixture of many sciences. Reactive transport of multi-component and multi-phase system involves physical and chemical processes. Physical processes can be convection and diffusion of species and chemical processes can be reaction of species with each other, with the medium fluids and even with the medium such as mineral reactions. Such reactive transport can be described by coupled non-linear system of partial differential equations (PDEs). These PDEs can be solved numerically for capturing the physics and chemistry of the underlying phenomenon.

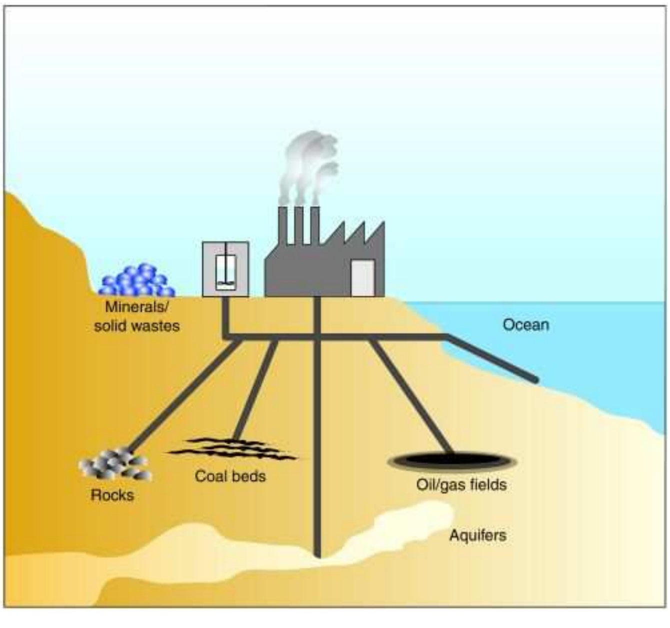


Figure 3.1: Various options of CO_2 sequestration.

Storage Option	Global Capacity (Gt. [gigatons] CO_2)
Depleted gas fields	690
Depleted oil fields/CO2-EOR	120
Deep saline aquifers	400 - 10 000
Unminable coal seams	40

Table 3.1: Estimate of Storage Capacities for Different Geological Trap Types

3.1 The Athena

In this work we are using in-house research simulator Athena for understanding flow pattern of CO_2 in porous medium. Athena was a result of a large European Union project running from January 1994 till December 1996. It has been under constant development since then and there are various versions of Athena. It has been extensively used in the past for understanding migration of hydrocarbons in reservoir formations and fractures [53; 59; 60; 107]. It was written in the C^{++} programming language, and consists of many independent modules.

Athena solves the temperature equation, the water pressure equation and n_c (number of chemical components) molar mass conservation equations. Thermodynamic computations in Athena can use both a fully generalized compositional model based on a cubic equation of state and a black oil model based on dew and bubble point curves. The temperature equation is solved first. The pressure and molar mass equations are nonlinear and depend on each other. There are two approaches for solving pressure and molar mass conservation equations:

1. Sequential implicit.
2. Implicit pressure and Explicit Composition (IMPEC)

In the Sequential implicit approach, the pressure equation is solved implicitly and the molar mass equations are solved in sequential implicit fashion using binary mixture thermodynamics. The phase calculations are done by tables for dew and bubble points. In the second approach, the pressure equation is solved implicitly and the molar mass equations are solved explicitly using CFL criteria for stability.

In the next section, continuum equations and their discretization is presented.

3.2 Continuum Equations of CO_2 Flow in Porous Medium

In this section, we will derive continuum equations for CO_2 dynamics in a porous medium. For notational convenience, the sub-index are used for denoting components and the super-index for denoting phases.

The system consists of two phases : water phase (w) and gas phase (g). Let there be total n_c components distributed in-between these two phases. These components can be water, carbon-dioxide, sodium chloride, various kinds of chemicals etc. The primary variables are temperature T, pressure P and molar masses N_p of each of the components

p. Our task in this section is to form partial differential equations for these primary variables. Secondary variables like molar mass of component p in phase l, N_p^l, can be expressed in terms of the primary variables.

The geological formation can be divided into smaller elements called finite volumes. Let us consider a finite volume with a volume V and with boundary S in space. We are assuming that the finite volume is not changing with time. The geological formation is composed of these finite volumes. Each of the volume V has unique medium properties (porosity, permeability) associated with them. In a control/finite volume all the phases can be present.

3.2.1 Mass Conservation Equation

A mass balance equation for a component p in a finite volume with volume V and surface S is written as follows:

$$\frac{\partial}{\partial t}\int_V m_p + \int_V \nabla \cdot \mathbf{m_p} = \int_V q_p, \tag{3.1}$$

In the equation (3.1), the sub-script V on a integral sign indicates that it is a volume integral. The equation (3.1) can be read as: the rate of change of a species p in a control volume V is equal to the rate at which species p being generated or disappeared minus flux of the species through the bounding surface (divergence of the species p). In the equation (3.1), m_p denotes the moles per unit volume (molar mass density) of the component p, $\mathbf{m_p}$ is the molar mass flux and q_p denotes the number of moles of p that are generating or disappearing per unit volume per unit time. Since the component p can be in any of the phases (water and gas). The molar mass velocity of component p is given as

$$\mathbf{m_p} = \sum_{l=w,g} C_p^l \, \xi^l \, \mathbf{v}^l, \tag{3.2}$$

In the equation (3.2), l denotes either the water or gas phase, C_p^l denotes the mass fraction of component p in phase l, ξ^l is the molar density of phase l (density/molecular weight) and $\mathbf{v}^l$ is the Darcy velocity of the phase l. Molar mass m_p density of the component p is given as

$$m_p = \phi_p \sum_{l=w,g} C_p^l \, S^l \, \xi^l, \tag{3.3}$$

In the equation (3.3), ϕ_p is the porosity of the finite volume V (medium property). It should be noted that the porosity can change due to mineral reactions. The total mass N_p of component p in the finite volume V is given as follows:

$$N_p = \int_V m_p. \tag{3.4}$$

The total source/sink for the component p in the finite volume V is $Q_p = \int_V q_p$.

3.2.2 Water Pressure Equation

Since water pressure is one of the primary variables an equation for water pressure is required. This equation is constructed using the method of Volume Balance [21].

Define the difference R between the pore volume V_p and volume of all phases as

$$R = V_p - \sum_{l=w,g} V^l, \tag{3.5}$$

This difference is a function of the water pressure P^w, the overburden pressure W, and molar mass of each component N_p; i.e., $R = R(P^w,\, W,\, N_p)$.

According to the Volume Balance method, the difference R between the volume of pore V_p and the volume of all phases has to be zero at all time. Linearising the difference R by expanding it through the Taylor Series in time reads as follows:

$$R(t + \Delta t) = R(t) + \frac{\partial R}{\partial t}\,\Delta t, \tag{3.6}$$

By the Volume Balance method, putting the term $R(t+\Delta t)$ equal to zero and using chain rule for $(\partial R/\partial t)$ results in:

$$\frac{\partial R}{\partial P^w}\,\frac{\partial P^w}{\partial t} + \sum_{i=1}^{n_c} \frac{\partial R}{\partial N_i}\,\frac{\partial N_i}{\partial t} = -\frac{R}{\Delta t} - \frac{\partial R}{\partial W}\,\frac{\partial W}{\partial t}. \tag{3.7}$$

This is the equation for the water pressure, and it is parabolic in nature.

The CO_2 gas phase pressure is given in terms of the water phase pressure and the capillary pressure as:

$$P^{CO_2} = P^{H_2O} + P_c^{wg} \quad (P^g = P^w + P_c^{wg}). \tag{3.8}$$

Here P_c^{wg} is the capillary pressure between water and gas phases.

3.2.3 Temperature Equation

For deriving the temperature equation, we are using the following energy conservation equation for a control volume V

$$\frac{\partial}{\partial t}\int_V \rho\, u - \int_S k\,\nabla T = -\int_S h\,\rho\,\mathbf{u} + \int_V q, \tag{3.9}$$

where $\rho\, u$ represents the capacity term, and $h\,\rho\,\mathbf{u}$ represents the convective flux term. These terms are expressed as follows:

$$\rho\, u = \sum_{l=w,g} \phi_p S^l u^l \rho^l + u_r \rho_r\,(1 - \phi_p), \tag{3.10}$$

$$h\,\rho\,\mathbf{u} = \sum_{l=w,g} h^l \rho^l \mathbf{v}^\mathbf{l}. \tag{3.11}$$

Here k is the bulk heat conductivity, ρ_r is the mass density of the rock, u_r is the internal energy of the rock, u^l is the internal energy of the phase l, and h^l is the enthalpy of the phase l.

3.3 Discretization of the Continuum Equations

3.3.1 Discretization of Molar Mass Equation

In the following section, we derive an implicit formulation of the mass transport equation (3.1) [47; 105; 107**?**]. After substituting (3.2), (3.3), (3.4) and $\int_V q_p = Q_p$ in the equation (3.1) reads

$$\frac{\partial N_p}{\partial t} + \int_V \sum_{l=w,g} \nabla \cdot \left(C_p^l \, \xi^l \, \mathbf{v}^l\right) = Q_p, \tag{3.12}$$

The total molar mass of a phase l (water or gas) is given as $N^l = V^l \, \xi^l$ (volume of phase l times the molar density of phase l). Let the specific volume of the phase l is a^l ($a^l = 1/V^l$). Substituting ξ^l and V^l in the equation (3.12)

$$\frac{\partial N_p}{\partial t} + \int_V \sum_{l=w,g} \nabla \cdot \left(C_p^l \, a^l \, N^l \, \mathbf{v}^l\right) = Q_p, \tag{3.13}$$

Applying the divergence theorem to the second term in the above equation and substituting the molar mass of component p in the phase l ($N_p^l = C_p^l \, N^l$) reads

$$\frac{\partial N_p}{\partial t} + \int_S \sum_{l=w,g} \left(N_p^l \, a^l \, \mathbf{v}^l\right) \cdot \hat{\mathbf{n}} = Q_p, \tag{3.14}$$

In the equation (3.14), $\hat{\mathbf{n}}$ is the unit normal to the surface S pointing away from the center of the control volume V. Let the boundary surface S is composed of small surfaces S_i such that $\cup S_i = S$ and $S_i \cap S_j = 0$ if $i \neq j$.

Space discretization of the equation (3.14) is done by Cell Centered Finite Difference approximation method and the continuity of the flux term is enforced on the interfaces. The surface integral in the equation (3.14) can be written in the discretized form as follows:

$$\int_S \sum_{l=w,g} \left(N_p^l \, a^l \, \mathbf{v}^l\right) \cdot \hat{\mathbf{n}} \approx \sum_{S_i} \sum_{l=w,g} \left(N_p^l \, a^l\right)_{[in]} \theta_i^l, \tag{3.15}$$

In the equation (3.15), $\theta_i^l = (\mathbf{v}^l \cdot \hat{\mathbf{n}})_i \, A_i$. Here A_i is the surface area of the interface S_i. The subindex in denotes the upstream control volume with respect to the interface S_i of the control volume V. Substituting equation (3.15) in the equation (3.14) results in

$$\frac{\partial N_p}{\partial t} + \sum_{S_i} \sum_{l=w,g} \left(N_p^l \, a^l\right)_{[in]} \theta_i^l = Q_p, \tag{3.16}$$

In the above equation, the convective part θ_i^l and the specific phase volume a^l are computed at the time t^n while N_p^l is computed at time t^{n+1}.

$$\frac{\partial N_p}{\partial t} + \sum_{S_i} \sum_{l=w,g} \left(\left[N_p^l\right]^{n+1} \left[a^l\right]^n \right)_{[in]} \left[\theta_i^l\right]^n = \left[Q_p\right]^n , \tag{3.17}$$

Time discretization in the above equation is done by the Backward Euler approximation

$$\frac{\left[N_p\right]^{n+1} - \left[N_p\right]^n}{\Delta t} + \sum_{S_i} \sum_{l=w,g} \left(\left[N_p^l\right]^{n+1} \left[a^l\right]^n \right)_{[in]} \left[\theta_i^l\right]^n = \left[Q_p\right]^n , \tag{3.18}$$

In the equation (3.18), N_p is the primary variable and N_p^l is the secondary variable. Our task is to express secondary variables in terms of primary variables. Since N_p^l is a function of the molar masses of individual components (the primary variables N_p), i.e., $N_p^l = N_p^l(N_1, N_2, ...)$. Linearising the molar mass of component p in the phase l in terms of molar masses of species by the Taylor series

$$\left[N_p^{l[k+1]}\right]^n = \left[N_p^{l[k]}\right]^n + \sum_{\mu} \left[\left(\frac{\partial N_p^l}{\partial N_\mu} \right)^{[k]} \right]^n \cdot \left[\Delta N_\mu\right]^n , \tag{3.19}$$

In the above equation, the summation is over all the components n_c, $[k]$ is the point about which Taylor series was expanded to get the value at the point $[k+1]$, and $\left[\Delta N_\mu\right]^n = \left[N_\mu^{[k+1]}\right]^n - \left[N_\mu^{[k]}\right]^n$. For simplifying the above formulation, we are ignoring the cross-derivative terms in the equation (3.19). The assumption of negligible cross-derivatives between different components eliminates the coupling and makes it possible to achieve greater computer efficiency by sequential solution [60; 60]. Thus, considering

$$\frac{\partial N_p^l}{\partial N_\mu} = 0 \quad \text{if} \quad p \neq \mu \tag{3.20}$$

in the equation (3.19), we will get

$$\left[N_p^{l[k+1]}\right]^n = \left[N_p^{l[k]}\right]^n + \left[\left(\frac{\partial N_p^l}{\partial N_p} \right)^{[k]} \right]^n \cdot \left[\Delta N_p\right]^n , \tag{3.21}$$

Substituting equation (3.21) into equation (3.18) results in the following Newton Iteration

$$\mathbf{J} \left[\Delta N_p{}^{[k]}\right]^n = \beta_p, \tag{3.22}$$

$$N_p^{k+1} = N_p^k + \Delta N_p{}^{[k]} \quad k = 0, 1, \ldots . \tag{3.23}$$

Here $N_p^0 = [N_p]^n$, the Jacobian and right hand side is given as:

$$\mathbf{J} = \left[\frac{\mathbf{I}}{\Delta t} + \left(\sum_{S_i}\sum_{l=w,g}\left(\frac{\partial N_p^l}{\partial N_p}\right)\left[a^l\right]^n [\theta_i^l]^n\right)_{[in]}\right]^k, \tag{3.24}$$

$$\beta_p = Q_p - \frac{[N_p^k]^n - [N_p]^n}{\Delta t} - \sum_{S_i}\sum_{l=w,g}\left[N_p^{l[k]}a^l\right]_{[in]}^n [\theta_i^l]^n. \tag{3.25}$$

After sufficient number of iterations, i.e., when $k \to \infty$ then $N_p^{k+1} \to [N_p]^{n+1}$.

3.3.2 Discretization of Temperature Equation

The conductive term in the heat flow equation (3.9) is approximated by Two Point Flux Approximation (TPFA) as follows

$$\int_S k\,\nabla T = \sum_m \int_{S_m} k\,\nabla T \approx \sum_m \sum_j \xi_{ij}\,(T_j - T_i), \tag{3.26}$$

where the index j means a control volume with a temperature T_j sharing the interface S_m. S_m is the m^{th} part of the surface boundary S. ξ_{ij} is the conductivity coupling coefficient of the two control volumes i and j sharing the interface S_m.

The source term in the equation (3.9) is treated explicitly for each control volume

$$\int_V q \approx Q, \tag{3.27}$$

The convective flux in the heat flow equation (3.9) is small compared to the conductive flux. So it can be neglected or can be treated explicitly. For explicit treatment the Darcy velocity at the boundary surfaces are approximated by the value at the upstream control volume. The convective flux term is discretized as

$$\int_S h\,\rho\,\mathbf{u} = \sum_m \int_{S_m} h\,\rho\,\mathbf{u} \approx \sum_m \left[A_m \sum_{l=w,g} h^l\,\rho^l\,\left(\mathbf{v}^l\right)^{UP}\right] = \Phi, \tag{3.28}$$

where S_m is denoting the m^{th} part of the surface S with a surface area A_m, and UP is denoting the upstream control volume.

The capacity term in the equation (3.9) is a function of temperature. Both the internal energy of the phase u^l and the internal energy of rock u_r in the capacity term are dependent on temperature. The capacity term is treated as:

$$\frac{\partial}{\partial t}\int_V \rho\,u \simeq \int_V \frac{\partial}{\partial t}(\rho\,u) \simeq \int_V \beta\,\frac{\partial T}{\partial t}, \tag{3.29}$$

where β is given as:

$$\beta = \sum_{l=w,g} \phi_p\,S^l\,c^l\rho^l + c_r\,\rho_r\,(1-\phi_p), \tag{3.30}$$

In the above equation, $c^l = \partial u^l / \partial T$ is the specific heat capacity of phase l and $c_r = \partial u_r / \partial T$ is the specific heat capacity of rock. For discretization β is approximated by cell center value.

After substituting the above terms, the discrete form of energy conservation equation (3.9) for a control volume i with volume V_i and surface S_i reads as

$$\beta_i \frac{\partial T_i}{\partial t} + \sum_m \xi_{ij}(T_j - T_i) = Q_i - \Phi_i, \tag{3.31}$$

Backward Euler is used for time discretization. The above equation can be expressed in residual form by matrix notation as

$$\mathbf{F}(\mathbf{T^{n+1}}) \equiv \mathbf{D}(\mathbf{T^{n+1}}) \frac{\mathbf{T}^{n+1} - \mathbf{T}^n}{\Delta t^n} + \mathbf{A}(\mathbf{T^{n+1}})\mathbf{T^{n+1}} - \mathbf{b}(\mathbf{T^n}) = 0, \tag{3.32}$$

where $\mathbf{D}(\mathbf{T^{n+1}})$ =diag(β_i), $\mathbf{A}(\mathbf{T^{n+1}}) = [\xi_{ij}]$ and $\mathbf{b}(\mathbf{T^n}) = (Q_i - \Phi_i)$. The rock temperature is almost constant, and both the convection and conduction terms have coefficients which are dominated by the rock temperatures. The Newton algorithm for the above non-linear equation can be written as

$$\mathbf{J}\, \Delta \mathbf{T}^k = -\mathbf{f^n}, \tag{3.33}$$

$$\mathbf{T^{k+1}} = \mathbf{T^k} + \Delta \mathbf{T^k} \quad k = 0, 1, \ldots. \tag{3.34}$$

Here

$$\mathbf{J} = \frac{\mathbf{D}(\mathbf{T^n})}{\Delta t^n} + \mathbf{A}(\mathbf{T^n}), \tag{3.35}$$

$$\mathbf{f^n} = \mathbf{A}(\mathbf{T^n})\, \mathbf{T^n} - \mathbf{b^n}. \tag{3.36}$$

In the limit $k \to \infty$, $\mathbf{T}^{k+1} \to \mathbf{T}^{n+1}$.

3.3.3 Discretization of the Water Pressure Equation

Space discretization of the water pressure equation (3.7) is done by cell centered volume method

$$\delta_{[i]} \frac{\partial P^w_{[i]}}{\partial t} + \sum_{p=1}^{nc} \varepsilon_{p[i]} \frac{\partial N_{p[i]}}{\partial t} = s_{[i]}, \tag{3.37}$$

where

$$\delta_{[i]} = \left(\frac{\partial R}{\partial p^w} \right)_{[i]}, \quad \varepsilon_{p[i]} = \left(\frac{\partial R}{\partial N_p} \right)_{[i]}. \tag{3.38}$$

and $s_{[i]}$ denotes the right hand side of the equation (3.7) at the center of the i^{th} cell in the mesh. In the equation (3.37), the molar mass derivative can be expressed in terms of the water pressure through the molar mass equation (3.12) as

$$\frac{\partial N_p}{\partial t} + \int_S \sum_{l=w,g} C^l_p \, \xi^l \, \mathbf{v^l} \cdot \hat{\mathbf{n}} = Q_p, \tag{3.39}$$

where $\hat{\mathbf{n}}$ is the unit normal to the boundary surface S and it points away from the center of the control volume i, Q_p is the integral of the source term for the component p. The Darcy velocity of the phase l is given as

$$\mathbf{v}^{\mathbf{l}} = -\sum_{m=g,w} \underline{K} \frac{k_r^{lm}}{\mu^m} \left(\nabla P^m - \gamma^m \nabla d \right) \tag{3.40}$$

where k_r^{lm} is the generalized relative permeability for coupled multiphase flow, $\underline{K}$ is the absolute permeability tensor, μ^m is the viscosity of the phase m, P^m is the fluid pressure of phase m, γ^m is the specific weight of the phase m, and d is the depth. The transmissibility tensors of each chemical component p are given as:

$$t_p^m = \sum_{l=w,g} C_p^l \, \xi^l \, \underline{K} \frac{k_r^{lm}}{\mu^m}, \tag{3.41}$$

The molar mass derivative can be expressed by uing equations (3.39), (3.40) and (3.41) as follows

$$\frac{\partial N_p}{\partial t} = \int_S \sum_{l=w,g} t_p^m \left(\nabla P^m - \gamma^m \nabla d \right) \cdot \hat{\mathbf{n}} + Q_p, \tag{3.42}$$

The water pressure has been chosen as a primary variable so the $\mathrm{CO_2}$ gas phase pressure is given in terms of the water pressure and the capillary pressure as

$$P^{CO_2} = P^{H_2O} + P_c^{wg} \quad (P^g = P^w + P_c^{wg}). \tag{3.43}$$

and part of the conductivity tern can be written as

$$\sum_{m=w,g} t_p^m \nabla P^m = \sum_{m=w,g} t_p^m \nabla P^w + t_p^g \nabla P_c{}^{wg}, \tag{3.44}$$

Substituting equation (3.44) into equation (3.42) results at the center of each control volume i

$$\frac{\partial N_p}{\partial t} = \sum_j t_{p[ij]} \, P_{[j]}^w + \Psi_{p[i]}, \tag{3.45}$$

where

$$\Psi_{p[i]} = \sum_j (t_{p[ij]}^g \, P_{c[j]}^{wg} - g_{p[ij] \, d_{[j]}}) + Q_{p[i]}, \tag{3.46}$$

where j denotes the surfaces of the control volume, $t_{p[ij]} = \sum_l t_{p[ij]}^l$ denotes the discrete transmissibilities, and $g_{p[ij]}$ is the equivalent discrete gravitation. Substituting the molar mass derivative approximation given by equation (3.45) into the equation (3.37) results

$$\delta_{[i]} \frac{\partial P_{[i]}^w}{\partial t} + \sum_j \alpha_{[ij]} \, P_{[i]}^w = \beta_{[i]}, \tag{3.47}$$

where

$$\alpha_{[ij]} = \sum_{p=1}^{n_c} \varepsilon_{p[i]} t_{p[ij]}, \quad \beta_{[i]} = s_{[i]} - \sum_{p=1}^{n_c} \varepsilon_{p[i]} \, \Psi_{p[i]}. \tag{3.48}$$

Using backward Euler time discretization in equation (3.47), and writing the resulting water pressure equation in the residual form

$$F(\mathbf{p^{n+1}}) = \mathbf{D}(\mathbf{P^{w[n+1]}})\frac{\mathbf{P^{w[n+1]}} - \mathbf{P^{w[n]}}}{\Delta t^n} + \mathbf{A}(\mathbf{P^{w[n+1]}})\,\mathbf{P^{w[n+1]}} - \mathbf{b}(\mathbf{P^{w[n+1]}}) = 0, \quad (3.49)$$

where $\mathbf{D} = diag(\delta_{[i]})$, $\mathbf{A} = \left[\alpha_{[ij]}\right]$ and $\mathbf{b} = (\beta_{[i]})$. The Newton iteration for the above non-linear residual equation is:

$$\mathbf{J}\,\Delta\mathbf{P^{w[k]}} = -\mathbf{f^n}[\mathbf{k}], \quad (3.50)$$

$$\mathbf{P^{w[k+1]}} = \mathbf{P^{w[n]}} + \Delta\mathbf{P^{w[k]}} \quad k = 0, 1, \ldots. \quad (3.51)$$

Here k is the Newton iteration index, and Jacobian and right hand side is given as

$$\mathbf{J} = \left(\frac{\partial F}{\partial p^w}\right) \approx \frac{\mathbf{D}}{\Delta t} + \mathbf{A}^{[n]} \quad (3.52)$$

$$\mathbf{f} = \mathbf{D}\,\frac{\mathbf{P}^{w[k]} - \mathbf{P}^{w[n]}}{\Delta t} + \mathbf{A}^{[n]}\,\mathbf{p^{w[k]}} - \mathbf{b^{[n]}} \quad (3.53)$$

In the limit $k \to \infty$, $\mathbf{P^{w[k]}} \to \mathbf{P^{w[n+1]}}$.

The mathematical models (3.22)-(3.23), (3.33)-(3.34) and (3.50)-(3.51) have been implemented in the in-house research simulator named Athena-CO_2 at the University of Bergen. Solution of these equations will provide the primary variables temperature (T), water pressure (P^w) and molar masses of components (N_p).

3.4 Thermodynamics of Phase Equilibria In Athena

Solution of the temperature, pressure, and molar mass conservation equations provide the primary variables water pressure (P_w), temperature (T) and molar masses of each component (N_ν). The secondary variables are: the molar mass of each component in each phase (N_ν^l), molar fraction of each componenet in each phase (C_ν^l), molar density of each phase (ξ^l), volume of each phase (V^l), saturation of each phase S^l.

Secondary variables should be computed using thermodynamic phase equilibrium constraints. There are two methods for computing the secondary variables. One method is to use the constraint of equal fugasities of components in different phases at equilibrium ($f_\nu^{l_i}$ = $f_\nu^{l_j}$) together with the cubic equation of state. Since such computations need to done for each grid cell in the mesh, it can computationally very expensive. Another approach is based on the binary mixture thermodynamics. All the components in the system are grouped into two groups water group (wg) and carbon group (cg). The water group contains H_2O, NaCl, CO_2, and different minerals, and carbon group contains only CO_2. We assume that the componenets in the two groups can distribute themselves between two phases (liquid and gas phase). This approximation at any fixed times t, generates for each phase l = w,g the partial derivatives.

$$\frac{\partial N_\nu^l}{\partial N_\mu} \quad \nu, \mu = wg, cg. \quad (3.54)$$

Let there be N_{wg} moles of water group and N_{cg} moles of carbon group. The derivatives given by the above equation depends on the number of phases that are present in a control volume.

3.4.1 Single Phase System

The single phase generates two cases eighter pure water phase or pure gas phase.

Assuming only water phase is present:

$$N^g = 0 \Rightarrow \quad N^w = N_{wg} + N_{cg}, \quad N^w_{wg} = N_{wg} \quad \text{and} \quad N^w_{cg} = N_{cg}$$

Assuming only gas phase is present:

$$N^w = 0 \Rightarrow \quad N^g = N_{wg} + N_{cg}, \quad N^g_{wg} = N_{wg} \quad \text{and} \quad N^g_{cg} = N_{cg}$$

Here we are assuming that there is no water group component in the gas phase, i.e, $N^g_{wg} = 0$. The Jacobian matrix assuming the condition (3.20) is the unit matrix, $\mathbf{J} = \mathbf{I}$.

3.4.2 Two Phase System

Consider a control volume where both gas and liquid phases are present. The partial derivatives may be expressed in terms of the bubble, $\beta(P^w, T)$, and dew point, $\alpha(P^w, T)$,

$$\beta = \frac{N^w_{wg}}{N^w}, \quad \alpha = \frac{N^g_{wg}}{N^g}, \quad N^w = N^w_{wg} + N^w_{cg}, \quad N^g = N^g_{wg} + N^g_{cg}, \tag{3.55}$$

here N^n_m means moles of component n in group m. We are assuming that there is no water group in the gas phase so, $N^g_{wg} = 0.0$. The molar fraction of the two groups is defined as

$$z_{wg} = \frac{N_{wg}}{N}, \quad z_{cg} = \frac{N_{cg}}{N}, \quad N = N_{cg} + N_{wg}, \tag{3.56}$$

here z_{wg} is the molar fraction of the water group, and z_{cg} is the molar fraction of the carbon group. To further simplify the calculations lets introduce the molar fractions of the water phase $z^w = N^w/N$. It can be easily shown

$$z_{wg} = \frac{N^w\beta + N^g\alpha}{N}, \quad \frac{N^w}{N} = \frac{(N^w\beta + N^g\alpha)/(N) - \alpha}{\beta - \alpha}, \quad z^w = \frac{z_{wg} - \alpha}{\beta - \alpha}, \tag{3.57}$$

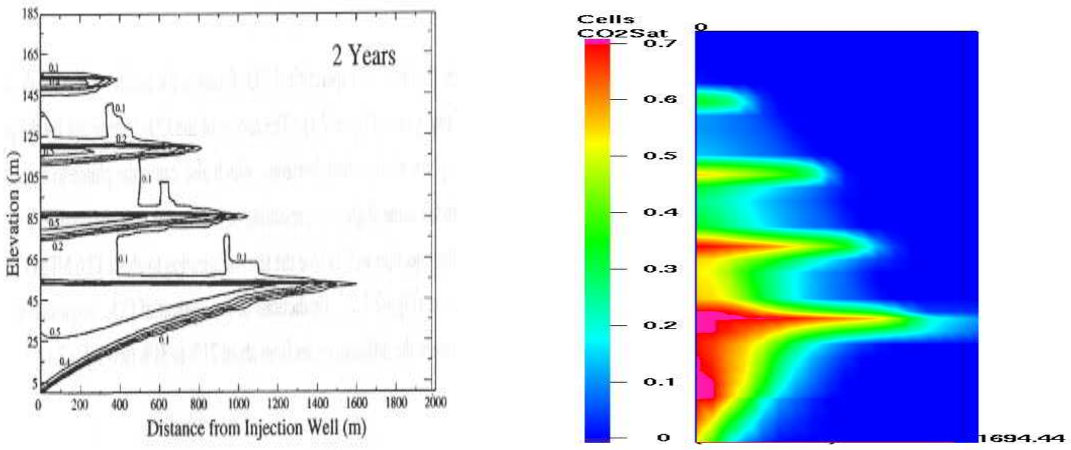
The quantities of interest can be expressed as

$$N^w_{wg} = \beta N^w, \qquad N^g_{cg} = (1-\alpha)N^g \quad = (1-\alpha)(1-z^w)N, \tag{3.58}$$

$$N^g_{wg} = \alpha N^g = \alpha(1-z^w)N, \qquad N^o_{cg} = (1-\beta)N^o \quad = (1-\beta)z^w N, \tag{3.59}$$

The partial derivative of the molar fraction of the water group with respect to the number of moles of water and gas group can be expressed as

$$\frac{\partial z_{wg}}{\partial N_{wg}} = \frac{1 - z_{wg}}{N}, \quad \frac{\partial z_{wg}}{\partial N_{cg}} = \frac{-z_{wg}}{N}, \tag{3.60}$$



(a) Migration of CO$_2$ by other Simulators.

(b) Migration of CO$_2$ by Athena.

Figure 3.2: The Left Figure is from an LBNL report [104] and right Figure report simulation result from Athena.

The Jacobian matrix is defined as

$$\left(\frac{\partial N_\nu^w}{\partial N_\mu}\right) = \begin{bmatrix} \frac{\partial N_{wg}^w}{\partial N_{wg}} & \frac{\partial N_{cg}^w}{\partial N_{wg}} \\ \frac{\partial N_{wg}^w}{\partial N_{cg}} & \frac{\partial N_{cg}^w}{\partial N_{cg}} \end{bmatrix} = \begin{bmatrix} \frac{(1-\alpha)\beta}{\beta-\alpha} & \frac{(1-\alpha)(1-\beta)}{\beta-\alpha} \\ -\frac{\alpha\beta}{\beta-\alpha} & -\frac{\alpha(1-\beta)}{\beta-\alpha} \end{bmatrix}, \tag{3.61}$$

Similarly for the gas phase

$$\left(\frac{\partial N_\nu^g}{\partial N_\mu}\right) = \begin{bmatrix} \frac{\partial N_{wg}^g}{\partial N_{wg}} & \frac{\partial N_{cg}^g}{\partial N_{wg}} \\ \frac{\partial N_{wg}^g}{\partial N_{cg}} & \frac{\partial N_{cg}^g}{\partial N_{cg}} \end{bmatrix} = \begin{bmatrix} -\frac{\alpha(1-\beta)}{\beta-\alpha} & -\frac{(1-\alpha)(1-\beta)}{\beta-\alpha} \\ \frac{\alpha\beta}{\beta-\alpha} & \frac{(1-\alpha)\beta}{\beta-\alpha} \end{bmatrix}. \tag{3.62}$$

The sum of the above two matrices is equal to an identity matrix.

3.5 Simulator Verification

We investigated the Test Problem 7 [103; 104] by the Athena simulator. This problem consists of CO$_2$ injection into a 2D layered brine formation. This problem has great similarity with the CO$_2$ injection project at the Sleipner Vest field in the Norwegian section of the North Sea, e.g., the injection rate, and properties of sand and shale layers.

The Figure 4.17 is a qualitative comparison of our results 3.7(b) against the results 3.7(a) from other simulators. It is clear from the Figure 4.17 that our simulator is capturing horizontal and vertical migration of CO$_2$ correctly.

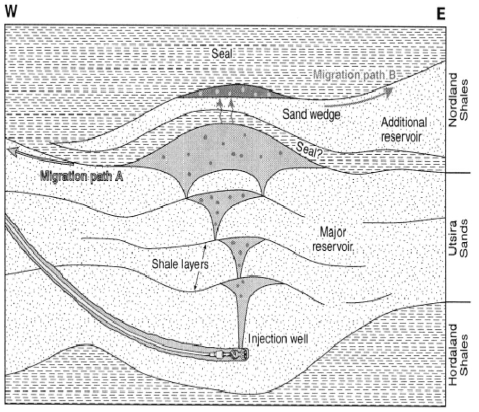


Figure 3.3: The Utsira formation. Formation consists of four shale layers. All of these four shale layers are following the topography of top surface [128].

3.6 Simulation on the Utsira Formation

Utsira formation is the first industrial-scale underground CO_2 storage site at the Sleipner Vest field in the Norwegian sector of the North sea. It is the only industrial-scale CO_2 disposal project currently in operation. Approximately 10^6 tonnes/year of CO_2 is injected into the sands of the Utsira formation. Time lapse seismic surveys shows that CO_2 movement at Utsira is strongly affected by the presence of shale layers. The formation consists four shale layers, and these layers follow the topography of the top surface as can be seen in the Figure 3.3 [128]. Figure 3.6 shows the porosity distribution and Figure 3.6 shows the permeability distribution in the formation. Figure 3.6 is the hexahedral meshing of the formation. It can be seen that the mesh is fine in the regions of high permeability.

We conducted numerical simulations for understanding affect of shale layers on the migration pattern of CO_2. Numerical results does indicate that the shale layers control the vertical migration of the CO_2. The presence of shale layers suppress vertical migration of CO_2 while helps lateral movement of CO_2. This is called hydrodynamic trapping and can be called geometrical or medium trapping.

The input data for this problem are the same as for the previous test case problem. But the depth of the formation is 250m. From the depth map we created the bottom topography parallel to the top surface at a depth of 250 m. We also created four shale layers of 3 m thickness parallel to the top surface at a depth of 30m, 70m, 120m, 200m respectable from the top surface. It is also clear from 3D seismic data that the Utsira formation contains several shale layers and which follow the topography of the top surface as can be seen in the Figure 3.3. The Figure 3.7 presents results of the Athena simulation in the presence and in the absence of shale layers (shale layers assumed to have same permeability as sand layers). As it is expected CO_2 starts forming a cloud beneath shale layers. It is interesting to see through simulations that a curved shale surface retard both

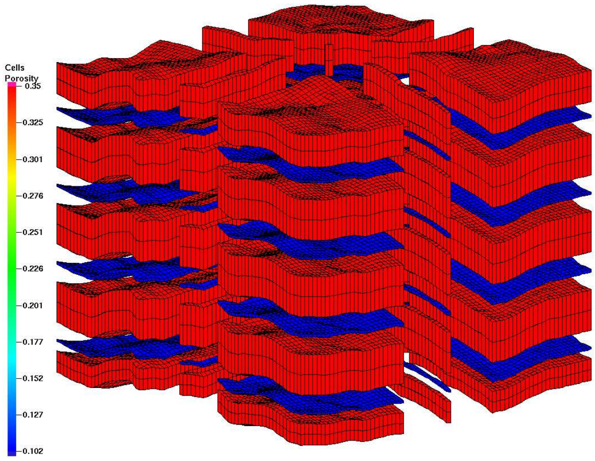


Figure 3.4: Formation consists of four shales. Porosity distribution

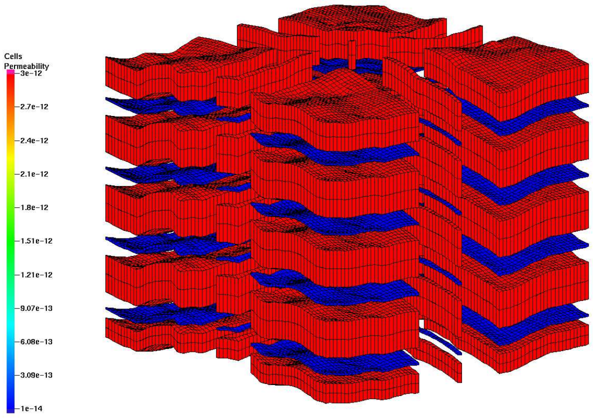


Figure 3.5: Formation consists of four shales. Permeability distribution

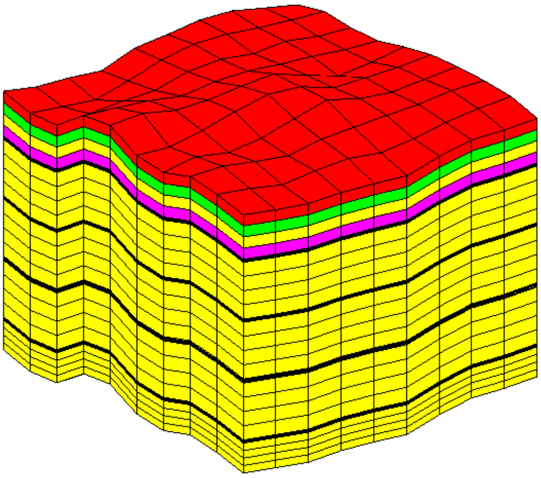


Figure 3.6: Grid Generation in the Utsira Formation. Formation Consists of Four Shale Layers. All of these four shale layers are following the topography of top surface.

horizontal and vertical migration of CO_2. It can be concluded from the Figure 3.7 that the dynamics of carbon-dioxide is strongly depends on the medium properties.

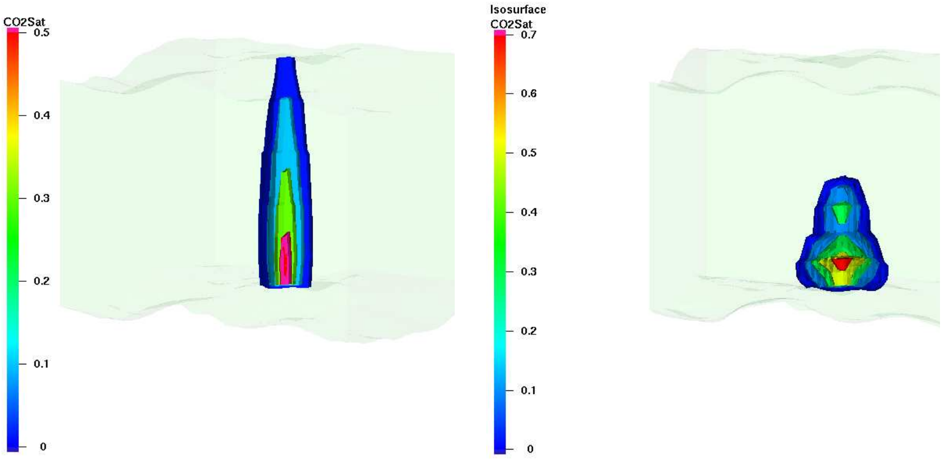
3.6.1 Simulation when the shale layers are flat.

In the previous simulations the shale layers were curved surfaces. In this section we are presenting simulations by considering the shales as flat surfaces. We are assuming that formation contains three flat shale layers. The data for the problem are the same as for the test case problem no. 7. Figure 3.8 shows the location and the porosity of the flat shale layers. Figure 3.9 presents simulation results when the formation contains three flat shale layers. CO_2 cloud form beneath curved and horizontal shale layers. It can be seen in the Figure3.9 that CO_2 migrates horizontally more beneath flat shale than curved shales.

3.7 Geochemical Transport with Athena

Let first define different mechanisms of containing CO_2 in a formation. Carbon dioxide can be trapped in geological formations in three ways:

1. Hydrodynamic trapping : CO_2 can be trapped as a supercritical liquid or gas in an aquifer because of low permeable shale layers. As it is clear from the numerical simulations that hydrodynamic trapping retard vertical migration of CO_2. CO_2 would take a long time to reach the top surface of the formation. So the concept behind hydrodynamic trapping is similar the way natural gases are trapped beneath low permeable shale layers.



(a) Migration of CO_2 in the absence of shale layers.

(b) Migration of CO_2 in the presence of four shale layers.

Figure 3.7: Effect of the medium properties on migration of CO_2.

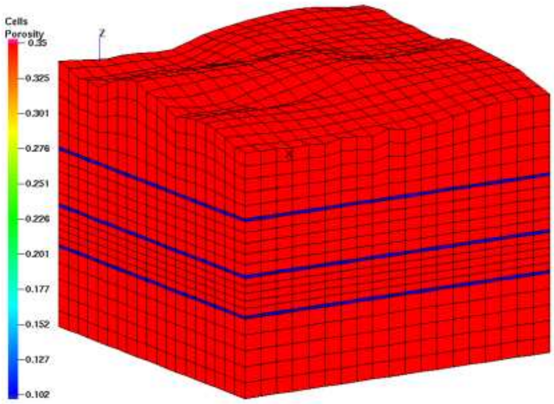


Figure 3.8: Formation consists three flat shale layers.

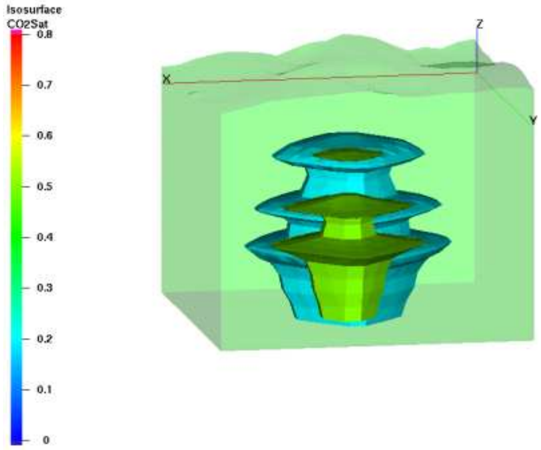


Figure 3.9: Migration of CO_2 in the presence of three flat shale layers.

2. Solubility Trapping : CO_2 can dissolve into the groundwater. The dissolution of CO_2 in water will increase the acidity of the water (lowers the pH, see the Figure 3.11) and effects the solubilities of minerals.

3. Mineral Trapping : CO_2 can react directly or indirectly with the minerals present in the medium leading to the precipitation of the secondary carbonates. It is the most desired form of trapping CO_2 because of long timescales (CO_2 can be retained for long period of time).

The mineral trapping can also be seen as the dissolved CO_2 (CO_2 stored in the form of solubility trapping) can react with the existing divalent cations and form stable mineral carbonates. Now for mineral trapping it is required to have CO_2 in the form of solubility trapping, and hydrodynamic trapping can provide enough time for the solubility trapping and thus ultimately can motivate mineral trapping.

It is clear from the numerical experimentations that curved surfaces and shale layers provide excellent opportunity for hydrodynamic trapping to occur. So it can be deduced that a formation containing shale layers in the form of curved surfaces is desirable for CO_2 sequestration. Since it will not only retard vertical (low permeable) and horizontal (curved surface) migration of CO_2 but also provide sufficient time for mineral and solubility trappings.

For modelling geochemical changes, we have implemented the Algorithm 1 in the Athena. Here, Water group := {Water, Chemicals} and CO_2 group := $\{CO_2\}$. For chemicals, please see the Table 3.2. Minerals are part of the solid phase.

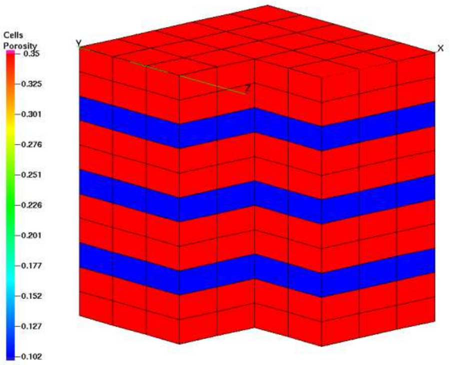


Figure 3.10: Porosity distribution.

Algorithm 1: Implementation of geochemical reactions in Athena. The Accerete solve the geochemistry decoupled from the flow/flash calculations in Athena.

```
for (time = 0 ; time ≤ $max_{time}$ ; $time^{++}$) do
    forall (Finite Volumes in the Mesh) do
        Find $H_2O$ and Chemicals in the Control Volume (available from the
        previous time step)
        Call Geochemical-Reactor ACCRETE()
    end
    forall (Finite Volumes in the Mesh) do
        forall (Components in the System) do
            Group the components in two groups
            Water group and $CO_2$ group
        end
    end
    Newton iteration for Water Group
    Newton iteration for $CO_2$ Group
    Inverse Group the Components
end
```

For doing geochemical computations we are using a geochemistry solver called Accrete. Accrete is written in Fortran programming language, and is an acronym for Athena Carbon Capture and stoRage geochEmistry modulE. Accrete is called as an external module from the Athena simulator as can be seen in the Algorithm 1 for each control volume in the mesh and for each time step. Meaning of various variables are given in the Table 3.2. A basic issue in sequestration of CO_2 is the physical and chemical behaviour in the vicinity of a CO_2 injection well. So to analyse the chemical behaviour we performed simulations. Figure 3.10 shows the grid and porosity distribution. Figure 3.11 shows the evolution of pH, and Figure 3.12 shows the concentration profile of H_2CO_3. It can be seen that injection of CO_2 increases the acidity of water.

Table 3.2: Different chemicals and minerals in the Accrete module.

Components	Meaning	Components	Meaning
Chemical[0]	H^+	mineral[0]	Calcite
Chemical[1]	Ca^{++}	mineral[1]	Albite
Chemical[2]	H2CO3	mineral[2]	Microcline
Chemical[3]	HCO3-	mineral[3]	Quartz
Chemical[4]	Na+	mineral[4]	Clinochlore-14A
Chemical[5]	Cl-	mineral[5]	Muscovite
Chemical[6]	Al+++	mineral[6]	Phlogopite
Chemical[7]	CO3—		
Chemical[8]	Mg++		
Chemical[9]	OH-		
Chemical[10]	H4SiO4		
Chemical[11]	K+		
Chemical[12]	Fe++		
Chemical[13]	Al(OH)4-		
Chemical[14]	H2O		

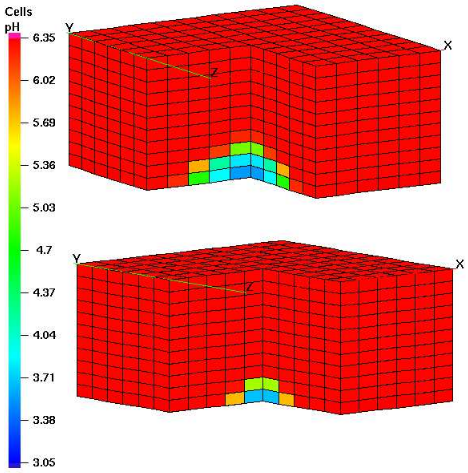


Figure 3.11: Evolution of pH near injection point.

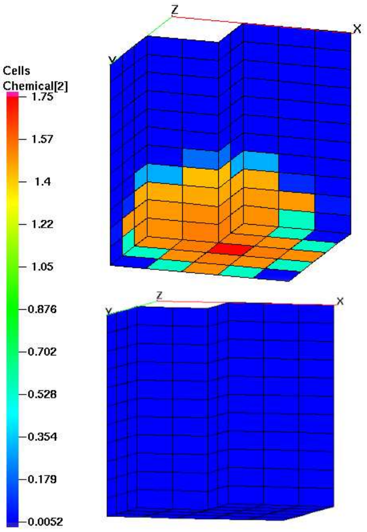


Figure 3.12: Change in the concentration of H_2CO_3.

3.8 Modelling of Hydrate Processes with Athena (Non-Volume Method)

The purpose of this section is to explain how to model hydrate processes by using the existing simulation tool Athena. It is assumed that hydrate processes occur in the form of chemical reactions at the internal surface boundaries in a thin film. So there will no continuity of flux and potential across the interfaces. The Non-Volume Method can handle discontinuity in flux and potential [53].

In the current version of Athena simulator it is possible to impose conditions on the internal surfaces. So we will exploit this possibility together with the Non Volume Method [54, and references therein]. The Figure 3.13 shows flow through an interface shared by two control volumes. It is assumed that there exists a thin fracture layer of zero volume at the interface between the cells. Imposing Robbin boundary condition at both side of the fracture shown in the Figure 3.13, i.e., $\alpha\, u + \beta\, \dfrac{\partial u}{\partial n} = \gamma$. The parameters α, β and γ may be selected arbitrary depending whether the unknown flow potential u or the flux $\frac{\partial u}{\partial n}$ normal to the boundary is specified.

The Darcy velocity of a fluid phase l is given as:

$$\mathbf{j}^l = -a^l\, K\, \nabla u^l \tag{3.63}$$

here

$$a^l = \xi^l \frac{k_{rl}}{\mu^l}$$

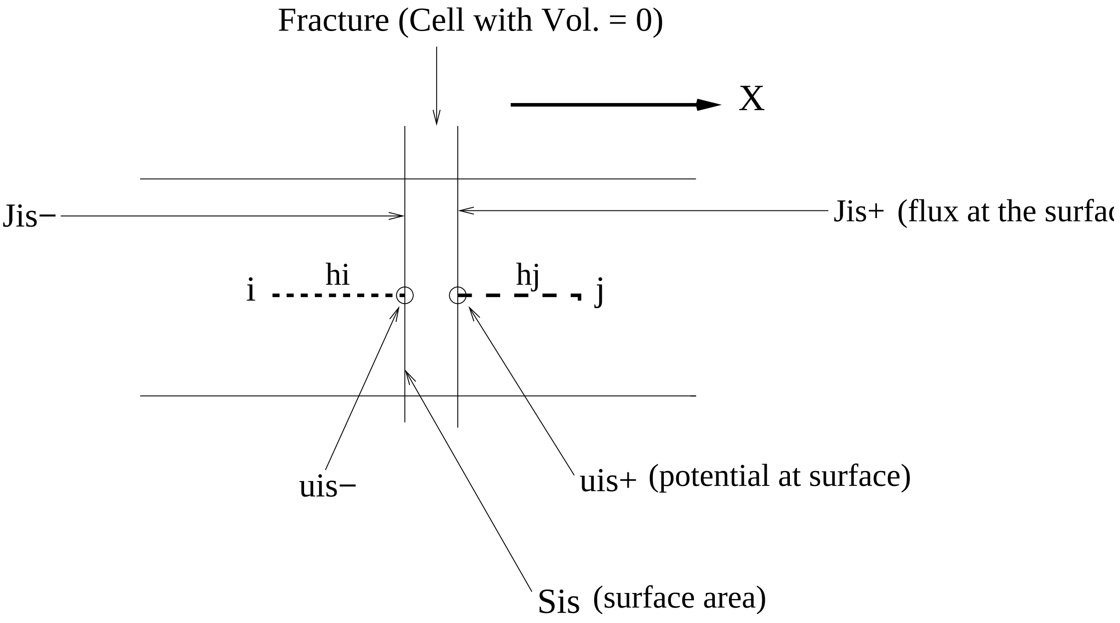


Figure 3.13: Flow across an internal face shared by two control volumes.

The flux of a component ν is expressed as

$$\mathbf{j}_\nu = \sum_l C^l_\nu \, \mathbf{j}^l \tag{3.64}$$

Thus the total flux leaving the control volume CV_i normal to the boundary CS_i is read as follows:

$$\mathbf{J_{is}} = -a^l_i \int_{CS_{is}} K \, \nabla u^l \cdot d\mathbf{S} \tag{3.65}$$

Selecting $\beta = -a^l_i$, in the Robbin boundary condition, we have the normal flux on the boundary

$$j^l_n \, S_{is} = a^l_i \, (\alpha^l \, u^l_{is} - \gamma^l) \, S_{is} \tag{3.66}$$

The flux leaving the control volume normal to the boundary and the flux at the boundary should be equal, i.e.,

$$\mathbf{J_{is}}^l = j^l_n \, S_{is}$$

By Two Point Flux Approximation :

$$u^l_{is} = \frac{1}{\alpha^l_{is} + \lambda_i} \, (\gamma^l_{is} + \lambda_i \, u^l_i) \tag{3.67}$$

here

$$\lambda_i = \frac{K_i}{h_i}$$

Since γ^l_{is} is an arbitrary parameter representing the potential. Let select $\gamma^l_{is} = \alpha^l_{is} \, u^l_{is}$.

Substituting this value into equation (3.66) and using the condition ${\mathbf{J_{is}}}^l = j_n^l\, S_{is}$, we get the expression for the flux through the boundary C_{is}

$$j_{is}^l = a_i^l\, \lambda_i\, \overline{\alpha}_{is}^l\, (u_i^l - u_{is}^l) \tag{3.68}$$

where

$$\overline{\alpha}_{is}^l = \frac{\alpha_{is}^l}{\alpha_{is}^l + \lambda_i}$$

It can be seen that $\overline{\alpha}_{is}^l$ is non-dimensional and due to physical reasons $\overline{\alpha}_{is}^l \in [0, 1]$ (please see[53]). It may represent properties at the boundary that may depend on the phases.

We are now considering two boundaries is^- and is^+ at the boundary CS$_i$, see Figure 3.13. The distance between these two boundaries is "zero" compared to the mesh dimension. This is the reason for the name of the method, The Non-Volume Method.

The flux from CV_i crossing the boundary is^- is:

$$J_{is^-}^l = a_i^l\, \lambda_i\, \overline{\alpha}_{is^-}^l\, (u_i^l - u_{is^-}^l)\, S_{is} \tag{3.69}$$

Similarly the flux crossing the boundary is^+ is

$$J_{is^+}^l = a_j^l\, \lambda_j\, \overline{\alpha}_{is^+}^l\, (u_{is^+}^l - u_j^l)\, S_{is} \tag{3.70}$$

The "Non-Volume" grid may represent a part of a fracture system with it's own flow model, or it may represent a grid undergoing it's own processes like geochemical reactions.

Neither the potential nor the fluxes need to be equal. A discontinuity may exist. Considering chemical reactions the potentials and the fluxes may change. Thus we may write:

$$J_{is^-}^l - J_{is^+}^l = Q_{is}^l \tag{3.71a}$$

$$u_{is^-}^l - u_{is^+}^l = \Delta {u_{is}}^l \tag{3.71b}$$

here Q_{is}^l and $\Delta u_i s^l$ denote the changes due to chemical reactions. The fluxes $J_{is^-}^l$ and $J_{is^+}^l$ may now be found by using the equations (3.71a) and (3.71b) for eliminating the potentials at the boundaries:

$$J_{is^-}^l = t_{i,j}^l\, (u_i^l - u_j^l) - t_{i,j}\, \Delta u_{is}^l + Q_{is}^l \tag{3.72}$$

here

$$t_{i,j}^l = \frac{k_{is^-}^l\, k_{is^+}^l}{k_{is^-}^l + k_{is^+}^l}$$

$$k_{is^-}^l = a_i^l\, \lambda_i \overline{\alpha}_{is^-}^l\, S_{is}$$

$$k_{is^+}^l = a_j^l\, \lambda_j\, \overline{\alpha}_{is^+}^l\, S_{is}$$

The flux given by the expression (3.72) can be implemented in the Athena simulator. The Darcy flux in Athena will then be replaced by the new flux expression (3.72).

The "Non-Volume Method" may also be applied to heat flow calculations where discontinuities in the temperature and/or heat flux are due to chemical reactions. The Darcy equation is simply replaced by the Fix's law of heat conduction for heat flow.

Chapter 4
FINITE VOLUME METHOD ON ADAPTIVE MESHES

We present 2P-FVM on adaptive meshes. Through numerical work, we show that convergence of the presented adaptive technique does not depend on the regularity. An adaptive technique depends on several factors such as error indicator and adaptive algorithm. We present a simple adaptive criterion, an adaptive algorithm and several examples. We do not change the discretization method (flux across the internal edges is given by the TPFA). Thus, our method can easily be implemented in existing simulators. 2P-FVM on uniform and adaptive meshes results in symmetric positive definite systems (for single phase flow) and hence, efficient solvers such as the Conjugate Gradient can be used. Ensuring monotonicity of a discretization scheme is difficult [100] and non-monotone schemes can provide unphysical results [2; 100]. Apart from being simple and fast, one other big advantage of the 2P-FVM is that the matrix system associated with 2P-FVM discretization is always monotone (cf. [100]). Let us consider the steady state pressure equation of a single phase flowing in a porous medium Ω [2; 41; 100; 110]

$$-\operatorname{div}(\boldsymbol{K}\operatorname{grad} p) = f \qquad \text{in} \quad \Omega, \tag{4.1}$$

$$p(x,y) = p^{\mathrm{D}} \qquad \text{on} \quad \partial\Omega_{\mathrm{D}}. \tag{4.2}$$

Here, Ω is a polyhedral domain in $\mathbb{R}^2$, the source function f is assumed to be in $L^2(\Omega)$ and the diagonal tensor coefficient $\boldsymbol{K}(x,y)$ is positive definite and piecewise constant. $\boldsymbol{K}$ is allowed to be discontinuous in space. In porous media flow, the unknown function $p = p(x,y)$ represents the pressure of a single phase, $\boldsymbol{K}$ is the permeability or hydraulic conductivity of the porous medium and the velocity $\mathbf{u}$ of the phase is given by the Darcy law as: $\mathbf{u} = -\boldsymbol{K}\operatorname{grad} p$ [2; 10; 41; 100]. For convergence of 2P-FVM on uniform, locally refined and non-uniform meshes, the following works are recommended [48; 49; 56; 116; 121]. A numerical analysis of the convergence of the 2P-FVM on uniform meshes can be found in [41].

4.1 Two Point Finite Volume Discretization

For solving partial differential equations (PDEs) on a domain by numerical methods such as the 2P-FVM, the domain is divided into smaller elements (meshing of the domain) called finite volumes or cells. Integrating equation (4.29) over one of the finite volumes V in the mesh and using the Gauss divergence theorem leads to

$$-\int_{\partial V} \boldsymbol{K}\,\nabla p \cdot \hat{\mathbf{n}}\,\mathrm{d}\sigma = \int_V f\,\mathrm{d}\tau, \tag{4.3}$$

where $\hat{\mathbf{n}}$ is the unit outward normal on the boundary ∂V of the finite volume V. Let us assume that finite volumes V are rectangular. The boundary of these finite volumes consists of 4 segments ∂V_i. The above equation can be written as

$$-\sum_{i=1}^{4} \int_{\partial V_i} \boldsymbol{K}\, \nabla p \cdot \hat{\mathbf{n}}\, \mathrm{d}\sigma = \int_V f\, \mathrm{d}\tau, \tag{4.4}$$

the term $-\int_{\partial V_i} \boldsymbol{K}\, \nabla p \cdot \hat{\mathbf{n}}\, \mathrm{d}\sigma$ is referred to as the flux through the edge ∂V_i. Denote it by $\mathcal{F}_i$. Thus, equation (4.4) can be written as

$$\sum_{i=1}^{4} \mathcal{F}_i = \int_V f\, \mathrm{d}\tau. \tag{4.5}$$

The degrees of freedom (DOF) for the 2P-FVM [2; 41] lie at the cell centers. Each finite volume in the mesh gives rise to a discrete equation (4.5). Collecting all such equations results in a discrete system $\boldsymbol{A}\, \mathbf{p}_h = \mathbf{b}$. Now let us consider computing $\mathcal{F}_i$ in equation (4.5). Two point approximation of the flux across the edge MN (see Figure 4.1(a)) is given as [2; 48; 49]

$$\mathcal{F}_{\mathrm{MN}} = \Phi_{\mathrm{MN}}\, (p_2 - p_1), \tag{4.6}$$

where the scalar Φ_{MN} is referred to as the transmissibility of the interface MN and is given as

$$\Phi_{\mathrm{MN}} = \mathrm{K}_1\, \mathrm{K}_2 \left(\frac{l}{h_1\, h_2} \right) \frac{1}{(\mathrm{K}_1/h_1 + \mathrm{K}_2/h_2)}. \tag{4.7}$$

Here, K_1 and K_2 refers to the permeability of the cells 1 and 2 in Figure 4.1(a); i.e., $\boldsymbol{K}_1 = \mathrm{K}_1\, \boldsymbol{I}$, $\boldsymbol{K}_2 = \mathrm{K}_2\, \boldsymbol{I}$. h_1 is the perpendicular distance of the interface MN from the center of cell 1 and similarly h_2 is the perpendicular distance of the interface MN from the center of cell 2. l is the length of interface MN. Adaptive discretization can result in a non-matching grid as shown in Figure 4.1(b). We are using the Two Point Flux Approximation for computing flux on a non matching grid. The flux through the interfaces AO and BO on the non-matching grid (see Figure 4.1(b)) is given as follows,

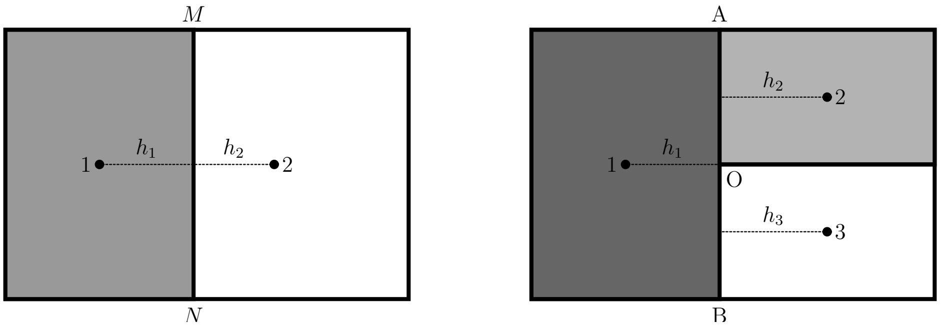
$$\mathcal{F}_{\mathrm{AO}} = \Phi_{\mathrm{AO}}(p_2 - p_1), \tag{4.8}$$

$$\mathcal{F}_{\mathrm{BO}} = \Phi_{\mathrm{BO}}(p_3 - p_1), \tag{4.9}$$

where p_1, p_2 and p_3 are the pressures of cells 1, 2 and 3. The transmissibilities Φ_{AO} and Φ_{BO} of the interfaces AO and BO are given as

$$\Phi_{\mathrm{AO}} = \mathrm{K}_1\, \mathrm{K}_2 \left(\frac{l_1}{h_1\, h_2} \right) \frac{1}{(\mathrm{K}_1/h_1 + \mathrm{K}_2/h_2)}, \tag{4.10}$$

$$\Phi_{\mathrm{BO}} = \mathrm{K}_1\, \mathrm{K}_3 \left(\frac{l_2}{h_1\, h_3} \right) \frac{1}{(\mathrm{K}_1/h_1 + \mathrm{K}_3/h_3)}. \tag{4.11}$$



(a) Flux across the interface MN shared by two cells 1 and 2.

(b) Flux across the interfaces OA and OB.

Figure 4.1: Computation of flux across an edge.

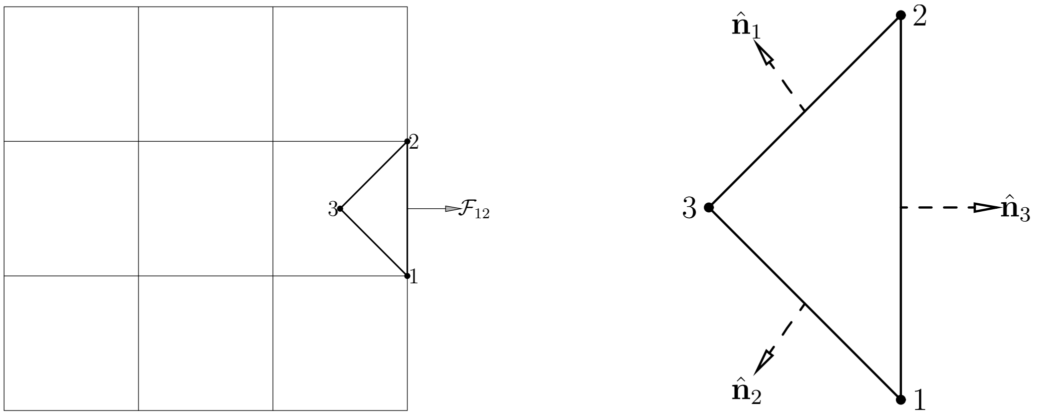
Here, l_1 is the length of interface AO and l_2 is the length of interface OB. F_{AO} and F_{BO} are the fluxes through edges AO and BO, respectively. K_1, K_2 and K_3 refers to the permeabilities of cells 1, 2 and 3 (see Figure 4.1(b)); i.e., $\boldsymbol{K}_1 = \mathrm{K}_1 \boldsymbol{I}$, $\boldsymbol{K}_2 = \mathrm{K}_2 \boldsymbol{I}$ and $\boldsymbol{K}_3 = \mathrm{K}_3 \boldsymbol{I}$. The total flux through edge AB is given as the sum of the fluxes through the edges OA and OB, i.e., $\mathcal{F}_{\mathrm{AB}} = \mathcal{F}_{\mathrm{OA}} + \mathcal{F}_{\mathrm{OB}}$.

4.1.1 Implementation of Boundary Conditions

In the case of Finite Volume discretization, every finite volume in the mesh will result in a discrete equation like (4.5). Thus, for handling boundary cells, boundary conditions are converted into an equivalent flux expression. Since Flux $= -\boldsymbol{K}\, \nabla p \cdot \hat{\mathbf{n}}$, the computation of the flux across an edge requires computation of the pressure gradient. Let us write an expression for the gradient of the scalar pressure p. Let the pressure at the three vertices of the triangle 4.2(b) be p_1, p_2 and p_3. Assuming that the pressure is varying linearly inside the triangle, the constant gradient of the pressure (∇p) in the triangle can be expressed by (see [2])

$$\nabla p = \frac{-1}{2\,|\Omega|} \sum_{i=1}^{3} p_i \, \hat{\mathbf{n}}_i. \tag{4.12}$$

Here, $|\Omega|$ is the area of the triangle and $\hat{\mathbf{n}}_i$ is the outward normal vector on the edge opposite to the vertex i. The magnitude of the vector $\hat{\mathbf{n}}_i$ is equal to the length of the edge. Figure 4.2(a) shows a 3×3 mesh. Let the pressure be specified at the boundary points 1 and 2. Applying the conservation principle (sum of the fluxes through the boundaries of the cell equal to the source inside the cell; i.e. equation (4.5)) to the boundary of cell 3, we have to compute the flux ($\mathcal{F}_{12}$) through the boundary edge 12. For computing the flux, let us form a boundary triangle 123 as shown in the Figure 4.2(b). Let the unknown pressure at the center of the boundary cell 3 be p_3. The pressure gradient inside the boundary triangle can be approximated by the expression (4.12). Thus, the flux through the boundary edge 12 is $F_{12} = -(\boldsymbol{K}\nabla p) \cdot \hat{\mathbf{n}}_3$. Let the outward normal vector on the edge



(a) A 3×3 mesh. Pressure is specified at the boundary points 1 and 2. Flux ($\mathcal{F}_{12}$) through the edge 12 is expressed as a linear combination of the pressures at the locations 1, 2 and 3. See the equation (4.13).

(b) Boundary triangle. Here, $\hat{\mathbf{n}}_i$ for $i = 1 \ldots 3$ are the normal vectors on the edges.

Figure 4.2: Implementation of the Dirichlet boundary condition.

(see the triangle 4.2(b)) opposite to the vertex i be $\hat{\mathbf{n}}_i = (nx_i, ny_i)^t$. Let the permeability of the boundary cell 3 be $\boldsymbol{K} = \text{diag}(kx, ky)$. Substituting the values of $\boldsymbol{K}$ and ∇p (given by the equation (4.12)) in the equation $\mathcal{F}_{12} = -(\boldsymbol{K}\,\nabla p) \cdot \hat{\mathbf{n}}_3$ results in

$$\mathcal{F}_{12} = -\frac{1}{2\,|\Omega|}\left[kx\left(\sum_{i=1}^{3} p_i\,nx_i\right)nx_3 + ky\left(\sum_{i=1}^{3} p_i\,ny_i\right)ny_3\right]. \tag{4.13}$$

Here, $|\Omega|$ is the area of the boundary triangle 123. Implementation of Neumann or flux boundary condition is even simpler. Flux across a boundary edge will go on the right hand side vector $\mathbf{b}$ of the discrete system $\boldsymbol{A}\,\mathbf{p}_h = \mathbf{b}$.

4.2 Adaptive Criteria and Adaptive Algorithm

The presented criteria has been extensively used in the Finite Element community (see the References [22; 65; 92; 109; 119, and references therein]). Let a mesh consist of N finite volumes/cells Ω_i, $i = 1, 2, \ldots, N$ and let p_h be the Control Volume solution on this mesh. Let p be the exact solution. The residual r can be defined as

$$r = f - \nabla \cdot (-\boldsymbol{K}\,\nabla p_h). \tag{4.14}$$

Intergrating and using the Gauss divergence theorem results in

$$\int_{\Omega} r\,dx = \int_{\Omega} f\,dx + \int_{\partial\Omega} (\boldsymbol{K}\,\nabla p_h)\cdot\hat{\mathbf{n}}\,dx, \tag{4.15}$$

and taking the modulus of both sides and applying the triangle inequality to the right hand side,

$$\left|\int_{\Omega} r\,dx\right| \leq \left|\int_{\Omega} f\,dx\right| + \left|\int_{\partial\Omega} (\boldsymbol{K}\,\nabla p_h)\cdot\hat{\mathbf{n}}\,dx\right|, \tag{4.16}$$

$$\left|\int_{\Omega} r\,dx\right| \leq \left|\sum_{i=1}^{N}\int_{\Omega_i} f\,dx\right| + \left|\sum_{i=1}^{N}\int_{\partial\Omega_i} (\boldsymbol{K}\,\nabla p_h)\cdot\hat{\mathbf{n}}\,dx\right|, \tag{4.17}$$

$$\left|\int_{\Omega} r\,dx\right| \leq \sum_{i=1}^{N}\left|\int_{\Omega_i} f\,dx\right| + \sum_{i=1}^{N}\left|\int_{\partial\Omega_i} (\boldsymbol{K}\,\nabla p_h)\cdot\hat{\mathbf{n}}\,dx\right|. \tag{4.18}$$

Using the Cauchy-Schwarz inequality; i.e., $|\int f| \leq \|f\|_{L_2}\,\|1\|_{L_2}$,

$$\left|\int_{\Omega} r\,dx\right| \leq \sum_{i=1}^{N}\left[\|f\|_{L^2(\Omega_i)}\,|\Omega_i|^{\frac{1}{2}}\right] + \sum_{i=1}^{N}\left[\|(\boldsymbol{K}\,\nabla p_h)\cdot\hat{\mathbf{n}}\|_{L^2(\partial\Omega_i)}\,|\partial\Omega_i|^{\frac{1}{2}}\right], \tag{4.19}$$

$$\left|\int_{\Omega} r\,dx\right| \leq \sum_{i=1}^{N}\left[\|f\|_{L^2(\Omega_i)}\,|\Omega_i|^{\frac{1}{2}} + \|(\boldsymbol{K}\,\nabla p_h)\cdot\hat{\mathbf{n}}\|_{L^2(\partial\Omega_i)}\,|\partial\Omega_i|^{1/2}\right], \tag{4.20}$$

$$\left|\int_{\Omega} r\,dx\right| \leq \sum_{i=1}^{N} r_i. \tag{4.21}$$

We are using the following expression for computing the error from cell i in a mesh,

$$\epsilon_i \overset{\text{def}}{=} \left[\|f\|_{L^2(\Omega_i)}|\Omega_i|^{\frac{1}{2}} + \|(\boldsymbol{K}\,\nabla p_h)\cdot\hat{\mathbf{n}}\|_{L^2(\partial\Omega_i)}|\partial\Omega_i|^{1/2}\right]. \tag{4.22}$$

Here, $|\Omega_i|$ is the area of the finite volume, $|\partial\Omega_i|$ is the circumference of the finite volume and $\hat{\mathbf{n}}$ is the unit outward normal. The quantity $\|(\boldsymbol{K}\,\nabla p_h)\hat{\mathbf{n}}\|_{L^2(\partial\Omega_i)}|\partial\Omega_i|^{1/2}$ is the flux associated with cell i. Let us further define a quantity named adaptivity index for cell i in a mesh,

$$\eta_i \overset{\text{def}}{=} \left[\frac{\epsilon_i}{\max\limits_{j\in\text{cells}} \epsilon_j}\right]. \tag{4.23}$$

It is clear from the above definition of the adaptivity index that η_i will be in the range $[0, 1]$. We are doing adaptive refinement by the Algorithm 2. The adaptivity index (4.23) is used for marking cells for further refinement. When a finite volume is selected for further refinement based on the value of the adaptivity index (4.23), this finite volume is divided into four equal finite volumes (see Figure 4.3). During the adaptive refinement process, all finite volumes Ω_i in a mesh for which the adaptivity index η_i is greater than a given tolerance δ, are refined. The tolerance δ lies between the values 0 and 1. δ equal to 0 means uniform refinement (refine all finite volumes) and δ equal to 1 means that the adaptive algorithm will refine a single finite volume per iteration step which can be costly. Both of these values can be computationally expensive and may not be optimal. A small δ will refine many finite volumes and thus introduce many new cells per iteration step of the adaptive algorithm. On the other hand, a large value of δ will refine fewer cells and thus introduce fewer new finite volumes per iteration step. It should be kept in mind that during each iteration step of the adaptive algorithm a discrete system needs to be solved. Typically a value of $\delta = 0.5$ is used [108].

It can be seen in the Algorithm 2 that the driving force for the Algorithm is the adaptivity index η. The adaptivity index (4.23) drives the Algorithm 2 by selecting some finite volumes for further refinement. Apart from the driving force, another important aspect of an algorithm is its stopping criteria. We are using three stopping criteria. The first two criteria are quite obvious.

Our first criterion "DOF $\leq$ DOF$_{\max}$" is the maximum allowed degrees of freedom (DOF$_{\max}$) or the maximum allowed mesh refinement. The second criterion "Iter $\leq$ Iter$_{\max}$" is the maximum allowed adaptive iteration steps.

The third criterion "$\xi_k/\xi_0 \leq tol$" is the error reduction after k iteration steps of the adaptive algorithm. Here, ξ_k denotes the maximum error (maximum value of ϵ_i on a mesh) on an adaptively refined mesh after k iteration steps of the adaptive Algorithm 2. The quantity ξ_k/ξ_0, which measures the reduction of the a posteriori error estimate, provides some information of the relative error reduction. Thus, ξ_k/ξ_0 can be used as a stopping criterion apart from the maximum number of degrees of freedom.

Algorithm 2: Adaptive Algorithm.

```
Mesh the domain;
Set Iteration Counter Iter = 0;
while $DOF \leq DOF_{\max}$ or $Iter \leq Iter_{\max}$ or $[\xi_k/\xi_0] \leq tol$ do
    Discretize the PDE over the mesh by FVM;
    Solve the discrete system with a given tolerance;
    forall (Finite Volumes j in the Mesh) do
        if ($\eta_j \geq \delta$) then
            Refine the Finite Volume j in the mesh;
        end
    end
    Form a new mesh;
    Iter$^{++}$;
end
```

Measuring the effectiveness of the adaptivity index (4.23) in selecting the cells with maximum error, we use the relation

$$\Gamma := \frac{\text{Cell number with } \eta = 1.0}{\text{Cell number with maximum point-wise error } |p - p_h|}. \tag{4.24}$$

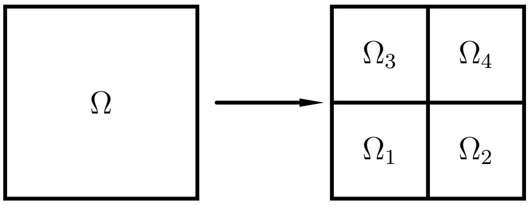


Figure 4.3: Refinement of the cell Ω into four cells Ω_i, $i = 1 \ldots 4$.

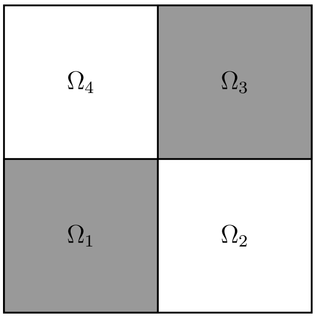


Figure 4.4: Domain Ω is divided into four sub-domains Ω_i, $i = 1 \ldots 4$ according to the permeability. Permeability in the sub-domain Ω_i is $\boldsymbol{K}_i$. We are assuming $\boldsymbol{K}_1 = \boldsymbol{K}_3 = R\,\boldsymbol{I}$ and $\boldsymbol{K}_2 = \boldsymbol{K}_4 = \boldsymbol{I}$.

Here, Γ is the robustness of the indicator η. If Γ is close to 1, the cells with the maximum point-wise error and the cells with the maximum error given by the error indicator (4.22) are the same. We compute the robustness quantity Γ of the adaptive index during each iteration step of the adaptive Algorithm 2. Figure 4.9 is a plot of the robustness quantity Γ against the iteration steps of the algorithm.

4.3 Numerical Examples

For all numerical experiments, the exact solution is given by an analytical form. The solution is enforced inside the domain by the Dirichlet boundary condition and the source term. For solving the discrete systems of equations formed on the sequence of adaptive and uniform meshes, we are using the ILU preconditioned Conjugate Gradient (CG) iterative solver unless mentioned otherwise. Let the domain be $\Omega = [-1, 1] \times [-1, 1]$. The domain is divided into four sub-domains according to the permeability $\boldsymbol{K}$ (see the Figure 4.4). Let the permeability in the sub-domain Ω_i be $\boldsymbol{K}_i$. It is assumed that the permeability in the sub-domain Ω_1 is equal to the permeability in the sub-domain Ω_3 and the permeability in the sub-domain Ω_2 is equal to the permeability in the sub-domain Ω_4; i.e., $\boldsymbol{K}_1 = \boldsymbol{K}_3$ and $\boldsymbol{K}_2 = \boldsymbol{K}_4$. Let us further assume that $\boldsymbol{K}_1 = \boldsymbol{K}_3 = R\,\boldsymbol{I}$ and $\boldsymbol{K}_2 = \boldsymbol{K}_4 = \boldsymbol{I}$. The parameter R is defined below. Let the exact solution in polar form be (analytical solution presented in Article H is almost the same as this solution)

$$p(r,\theta) = r^{\gamma}\,\eta(\theta), \tag{4.25}$$

(see [22; 92]). The parameter γ denotes the singularity in the solution [22] and it depends on the permeability distribution in the domain (see Figures 4.5(a) and 4.5(b) for the permeability distribution for the singularities $\gamma = 0.1$ and $\gamma \approx 0.13$). $\eta(\theta)$ is given as

$$\eta(\theta) = \begin{cases} \cos\left[(\pi/2-\sigma)\gamma\right]\,\cos\left[(\theta-\pi/2+\rho)\gamma\right], & \theta \in [0, \frac{\pi}{2}], \\ \cos(\rho\gamma)\,\cos\left[(\theta-\pi+\sigma)\gamma\right], & \theta \in [\frac{\pi}{2}, \pi], \\ \cos(\sigma\gamma)\,\cos\left[(\theta-\pi-\rho)\gamma\right], & \theta \in [\pi, \frac{3\pi}{2}], \\ \cos\left[(\pi/2-\rho)\gamma\right]\,\cos\left[(\theta-3\pi/2-\sigma)\gamma\right], & \theta \in [\frac{3\pi}{2}, 2\pi], \end{cases} \tag{4.26}$$

and the parameters R, γ, ρ and σ satisfy the following nonlinear equations

$$\begin{cases} R & = -\tan\left[(\pi-\sigma)\,\gamma\right]\,\cot(\rho\gamma), \\ 1/R & = -\tan(\rho\gamma)\,\cot(\sigma\gamma), \\ R & = -\tan(\sigma\gamma)\,\cot\left[(\pi/2-\rho)\gamma\right], \end{cases} \tag{4.27}$$

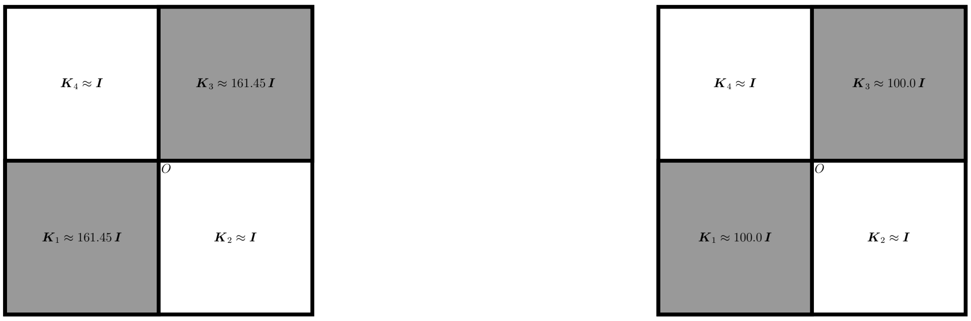
under the following nonlinear constraints

$$\begin{array}{rl} 0 < & \gamma < 2, \\ \max\{0, \pi\gamma - \pi\} < & 2\gamma\rho < \min\{\pi\gamma, \pi\}, \\ \max\{0, \pi - \pi\gamma\} < & -2\gamma\rho < \min\{\pi, 2\pi - \pi\gamma\}. \end{array} \tag{4.28}$$

The constrained nonlinear equations (4.27) can be solved for the parameters R, σ, and ρ by Newton's iteration algorithm for different degrees of singularity γ. The analytical solution $p(r,\theta)$ satisfies the usual interface conditions: p and $\mathbf{K}\,\frac{\partial p}{\partial \mathbf{n}}$ are continuous across the interfaces. It can be shown that the solution p belongs in the fractional Sobolev space $\mathbf{H}^{1+\kappa}(\Omega)$ $(\kappa < \gamma)$ [115]. Let the singularity be $\gamma = 0.1$. The parameters which satisfy the relations (4.27) under the constraints (4.28) are

$$R \approx 161.4476, \quad \rho \approx 0.7854 \quad \text{and} \quad \sigma \approx -14.9225.$$

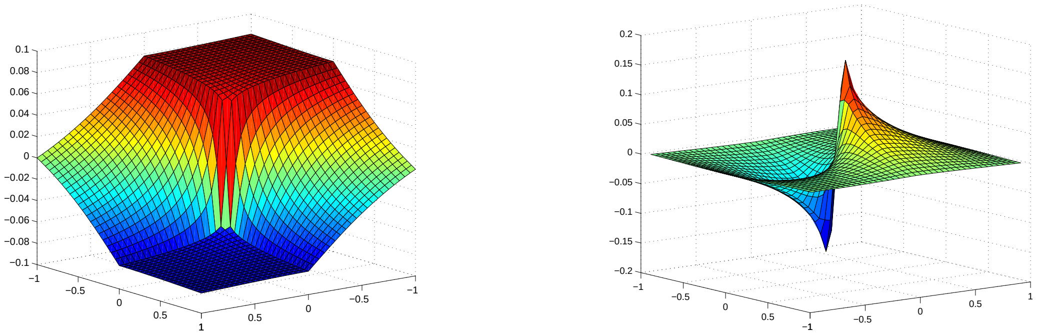
The permeability distribution is shown in Figure 4.5(a). The exact solution belongs to the fractional Sobolev space $\mathbf{H}^{1+k}$ $(k < 0.1)$. We have solved this problem on adaptive and uniform meshes. The outcome of our numerical work is reported in Figures 4.6, 4.7 and 4.8. Figure 4.6(a) is a surface plot of the exact solution. The solution is singular at the origin. Figure 4.6(b) presents a surface plot of the error. It can be seen that the error is maximum at the singularity. Figure 4.7 compares the convergence behaviour on adaptive and uniform meshes in the L_∞ norm. We did not notice any convergence in the L_∞ norm on uniform meshes till one million degrees of freedom. A similar behaviour was also observed in [41] on uniform meshes for singular problems. It was suggested in [41] (see page no. 1094) that adaptive meshes may be ideal for such solutions. On adaptive meshes, we are getting $\|\mathbf{p} - \mathbf{p}_h\|_{L_\infty} \approx \text{DOF}^{-P/2}$ with the convergence $P \approx 1$ (see Figure



(a) $\gamma = 0.1$.

(b) $\gamma = 0.126902$.

Figure 4.5: Permeability distribution for the singularities $\gamma = 0.1$ and $\gamma = 0.1269$. The solution is singular at $O = (0, 0)$.



(a) Exact solution given by the equation (4.25) for $\gamma = 0.1$.

(b) Surface plot of the error $(u - u_h)/\|u\|_{L_\infty}$ for $\gamma = 0.1$.

Figure 4.6: Surface plots of the exact solution and error for the singularity parameter $\gamma = 0.1$.

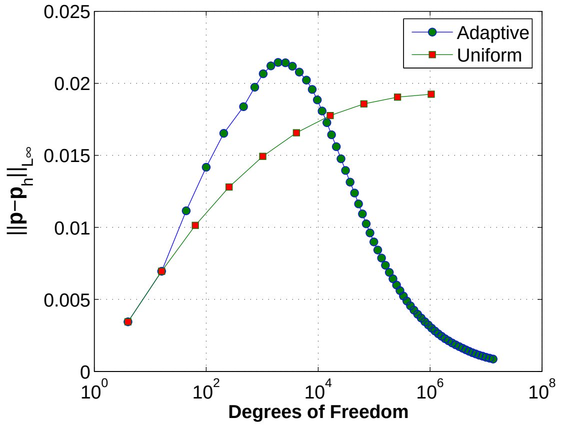


Figure 4.7: On adaptive meshes $\|\mathbf{p}-\mathbf{p}_h\|_{L_\infty} \approx \mathrm{DOF}^{-1.0/2}$. On uniform meshes there is no convergence till 1 million degrees of freedom. The solution is in $\mathbf{H}^{1+k}$ ($k < 0.1$).

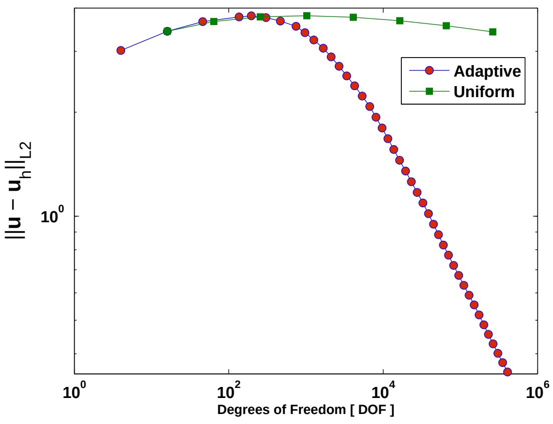


Figure 4.8: On adaptive meshes $\|\mathbf{u}-\mathbf{u}_h\|_{L_\infty} \approx \mathrm{DOF}^{-1.0/2}$. On uniform meshes there is no convergence till 1 million degrees of freedom. The solution is in $\mathbf{H}^{1+k}$ ($k < 0.1$).

4.7). Because of the regularity of the solution, this convergence is quasi optimal [22; 92]. Convergence of the Darcy flux on adaptive and uniform meshes is reported in Figure 4.8. On uniform meshes, the convergence of the Darcy flux is approximately equal to the regularity of the solution. On the other hand, the method is converging with a rate approximately equal to 1 on the adaptive meshes.

4.3.1 Robustness of the adaptivity index η

Figure 4.9 is a plot of the robustness against the iteration steps of the adaptive algorithm. It can be seen in Figure 4.9 that the robustness is almost always equal to 1 for all of the adaptive iteration steps. Hence, the cells with the maximum point-wise error and the cells with the maximum value of the error indicator (given by the equation (4.22)) are the same.

4.3.2 The δ Effect

The adaptive Algorithm 2 depends on many parameters such as the adaptivity index η and the parameter δ. Most of these parameters also depend on each other. Thus, finding an optimal mesh is a non-linear problem. The parameter δ is an important ingredient of the Algorithm 2. Based on the value of δ, the algorithm refines the finite volumes. A higher value of δ means that the algorithm will refine fewer cells, while for a lower value of δ, the algorithm will refine many cells. For example, if $\delta = 0$ the algorithm will refine all the cells in the mesh, and if $\delta = 1$, the algorithm will refine only one cell. Both of these situations can be costly. For the singularity parameter $\gamma = 0.1$, we run the algorithm for three values of δ $(0.5, 0.6, 0.7)$. Figure 4.10 reports the outcome of our work. It can be seen in Figure 4.10 that for a given degree of freedom, $\delta = 0.5$ is a better choice. It

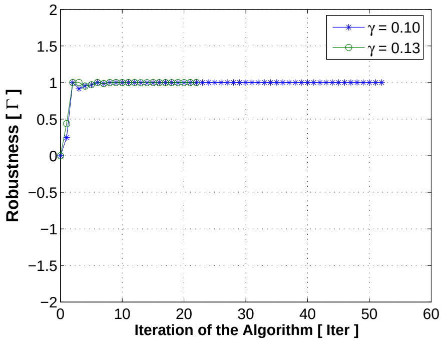
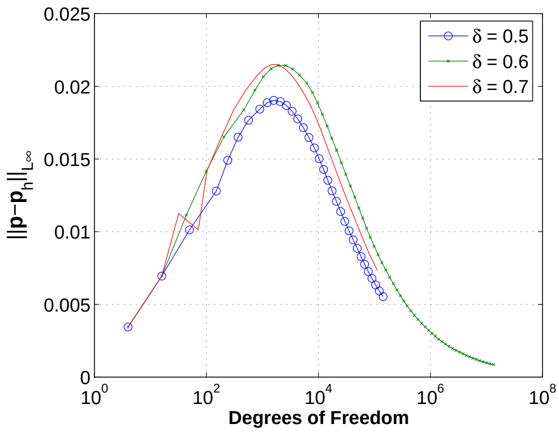


Figure 4.9: Robustness (defined by the equation (4.24)) of the adaptivity index for finding cells with most error. Solutions are in the spaces $\mathbf{H}^{1.126902}$ and $\mathbf{H}^{1.1}$.



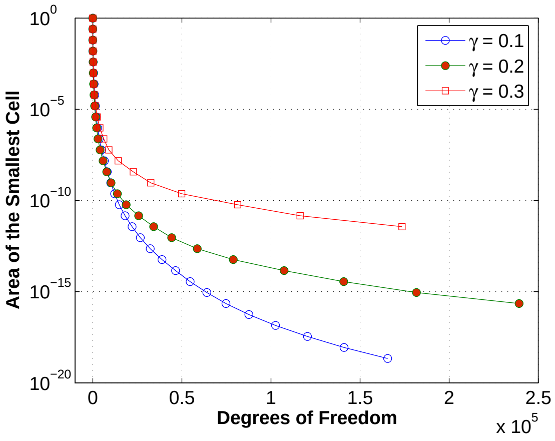


Figure 4.10: Effect of δ. See the Algorithm 2 for δ.

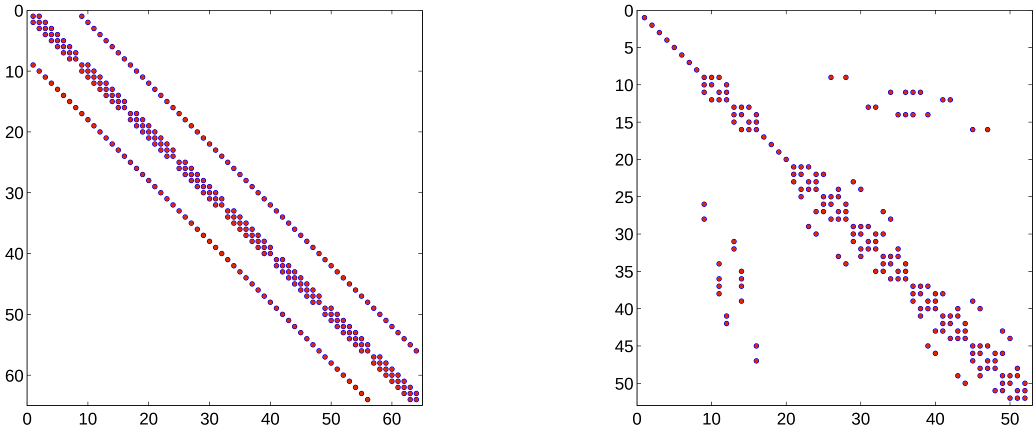
Figure 4.11: Area of the smallest cell in an adaptive mesh vs the number of cells.

should be noted that the number of iteration steps of the adaptive algorithm increases with increasing value of δ. Thus, the computational work required for $\delta = 0.5$ is lower than the work required for $\delta = 0.6$ and $\delta = 0.7$. Of these three values of δ, $\delta = 0.5$ is optimal. It requires the least amount of computational work for a given tolerance.

Figure 4.12 shows the sparsity pattern of the matrices associated with uniform and adaptive meshes. Figure 4.12(a) presents the sparsity pattern of a matrix formed on a 8×8 uniform mesh (DOF = 64). Figure 4.12(b) shows the sparsity pattern of an adaptive discrete system formed on an adaptive mesh with DOF = 52.

4.4 Which Meshes are Better Conditioned; Adaptive, Uniform, Locally Refined or Locally Adjusted ?

Adaptive, locally refined and locally adjusted meshes are preferred over uniform meshes for capturing singular or localised solutions. For local refinements applied to porous media



(a) Sparsity pattern of a discrete system formed on a uniform mesh with DOF = 64. Number of non-zeros are 288.

(b) Sparsity pattern of a discrete system formed on an adaptive mesh with DOF = 52. Number of non-zeros are 196.

Figure 4.12: Sparsity pattern of the discrete systems formed on adaptive and uniform meshes.

flow, see the References [7; 15; 26; 48; 49; 50; 51]. Roughly speaking, for a given degree of freedom, a solution associated with adaptive, locally refined and locally adjusted meshes is more accurate than the solution given by uniform meshes. In this work, we discuss which meshes are better conditioned. We found, for approximately the same degree of freedom (same size of matrix), that it is easier to solve a system of equations associated with an adaptive mesh. Uniform, locally adjusted, adaptive and locally refined meshes are shown in Figures 4.13, 4.14, 4.15 and 4.16, respectively. Here, for each mesh the number of cells (or degrees of freedom) are approximately 1024 ($2^5 \times 2^5$). Let us consider the steady state pressure equation of a single phase flow in a porous medium Ω [2]

$$-\mathrm{div}\,(\mathrm{K}\,\mathrm{grad}\,p) = f \qquad \text{in} \quad \Omega, \tag{4.29}$$

$$p(x,y) = p^{\mathrm{D}} \qquad \text{on} \quad \partial\Omega_{\mathrm{D}}. \tag{4.30}$$

We solve the problem (4.29) on the four meshes. Let the exact solution be given by the equation (4.25) and the singularity parameter be $\gamma = 0.1$. We enforce the solution inside the domain by the Dirichlet boundary condition and the source term. For solving the discrete system of equations formed on the meshes, we use the Conjugate Gradient (CG) solver (see [120, chap. 5], [64]). Table 4.1 presents eigenvalues and condition numbers of the matrix systems associated with the different meshes. Note that in this table, the largest eigenvalue on all four meshes is approximately same. However, the smallest eigenvalue associated with the adaptive mesh is greater than the smallest eigenvalues associated with other three meshes. When solving the Symmetric Positive Definite (SPD) linear system $\boldsymbol{A}\,\mathbf{p}_h = \mathbf{b}$ with the CG method, the smallest eigenvalues in the matrix $\boldsymbol{A}$ often slows down the convergence (cf. [64; 120]). Several techniques have been proposed in

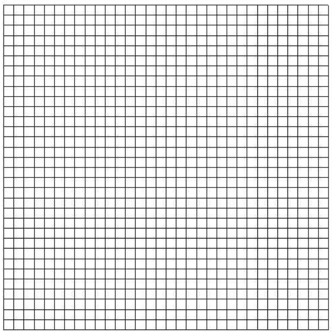


Figure 4.13: Uniform mesh containing 1024 cells (32×32).

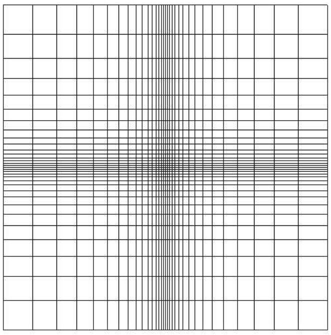


Figure 4.14: Localised mesh containing 1024 cells (32×32).

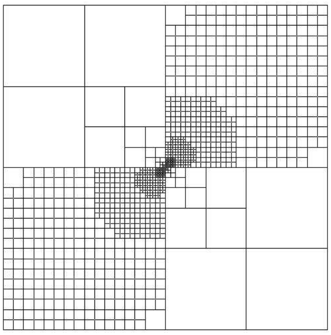


Figure 4.15: Adaptive mesh containing 1150 cells.

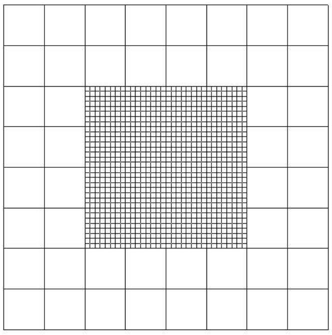


Figure 4.16: Locally refined mesh containing 1074 cells.

the literature to remove bad effect of the smallest eigenvalues (see [19; 64, and references therein]). Convergence of the CG solver for these four systems are shown in Figure 4.17.

It can be seen in Table 4.1 and Figure 4.17 that it is easier to solve a matrix system associated with an adaptive mesh than to solve systems associated with uniform, localised and locally refined meshes. The adaptive mesh is generated by equal distribution of the fluxes over all the cells in the mesh. Why do equal distribution of fluxes create meshes which are better conditioned ? Or, why do equal distribution of fluxes remove bad effect of small eigenvalue ? The answers to these questions can help in designing new preconditioners or improving existing preconditioners.

All the reported numerical work is done on uniform and adaptive meshes containing only square cells. In the work [7; 33], 2P-FVM discretization on non matching rectangular meshes was found to be nonconverging. Articles E and H present finite volume method on adaptive meshes. In the article E, flux is approximated by the Two Point Method while in the article H, the flux is approximated by the Multi Point Method.

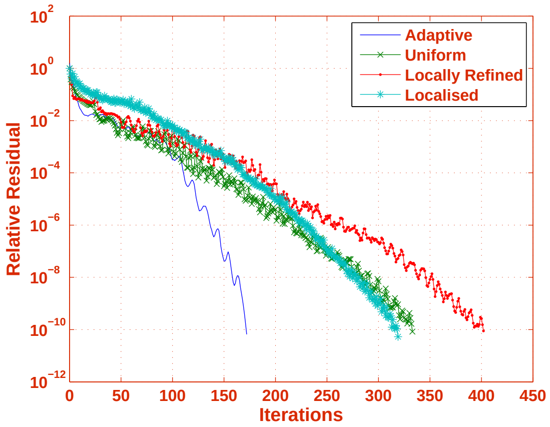


Figure 4.17: Convergence curves for the matrix systems formed on the meshes given in the Figures 4.13, 4.14, 4.15 and 4.16.

Table 4.1: Eigenvalues and condition numbers of the matrix systems formed on the four meshes shown in the Figures 4.13, 4.14, 4.15 and 4.16.

Mesh	Smallest eig.	Largest eig.	Condition Number
Adaptive	4.150×10^{-1}	1.2878×10^{3}	3.1029×10^{3}
Localised	5.50×10^{-2}	0.7821×10^{3}	1.4216×10^{4}
Uniform	7.62×10^{-2}	1.2885×10^{3}	1.6916×10^{4}
Locally Refined	3.94×10^{-2}	1.2793×10^{3}	3.2495×10^{4}

4.5 Converting Athena into an Adaptive Porous Media Flow Simulator

The work presented shows that adaptivity is not a luxury but a necessity. The presented ideas can easily be incorporated in the existing porous media flow simulator the Athena. Athena reads geometry from the geom_00.dat file. This file has information about the grid connection, number of cells, number of nodes and volume of cells. Athena computes transmissibilities through all the internal surfaces of a mesh. The author's idea is to generate this file adaptively by repeatedly calling an adaptive grid generator from Athena. The scheme is presented in Figure 4.18.

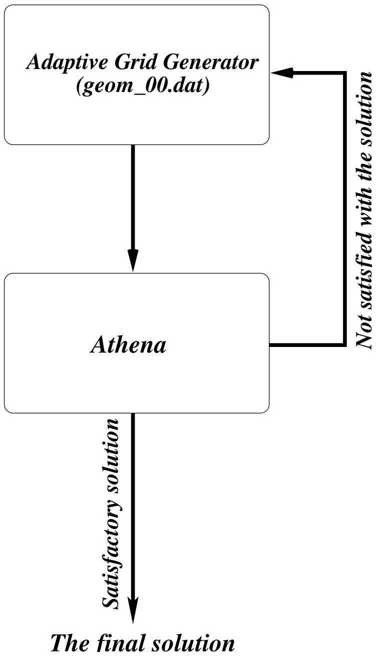


Figure 4.18: An adaptive porous media flow simulator.

BIBLIOGRAPHY

[1] J. Aarnes and M. S. Espedal. A new approach to upscaling for two-phase flow in heterogeneous porous media. In *Fluid flow and transport in porous media: mathematical and numerical treatment (South Hadley, MA, 2001)*, volume 295 of *Contemp. Math.*, pages 1–11. Amer. Math. Soc., Providence, RI, 2002.

[2] I. Aavatsmark. An introduction to multipoint flux approximations for quadrilateral grids. *Comput. Geosci.*, 6(3-4):405–432, 2002.

[3] I. Aavatsmark, T. Barkve, Ø Bœ, and T. Mannseth. Discretization on non-orthogonal, quadrilateral grids for inhomogeneous, anisotropic media. *Journal of Computational Physics*, 127:2–14, 1996.

[4] I. Aavatsmark, T. Barkve, Ø. Bøe, and T. Mannseth. Discretization on unstructured grids for inhomogeneous, anisotropic media. I. Derivation of the methods. *SIAM J. Sci. Comput.*, 19(5):1700–1716 (electronic), 1998.

[5] I. Aavatsmark, T. Barkve, Ø. Bøe, and T. Mannseth. Discretization on unstructured grids for inhomogeneous, anisotropic media. II. Discussion and numerical results. *SIAM J. Sci. Comput.*, 19(5):1717–1736 (electronic), 1998.

[6] I. Aavatsmark, T. Barkve, and T. Mannseth. Control-volume discretization methods for 3d quadrilateral grids in inhomogeneous, anisotropic reservoirs. *SPE 38000*, 1997.

[7] I. Aavatsmark, E. Reiso, and R. Teigland. Control-volume discretization method for quadrilateral grids with faults and local refinements. *Comput. Geosci.*, 5(1):1–23, 2001.

[8] M. Afif and B. Amaziane. On convergence of finite volume schemes for one-dimensional two-phase flow in porous media. *J. Comput. Appl. Math.*, 145(1):31–48, 2002.

[9] T. Apel and N. Düvelmeyer. Transformation of hexahedral finite element meshes into tetrahedral meshes according to quality criteria. *Computing*, 71(4):293–304, 2003.

[10] K. Aziz and A. Settari. *Petroleum Reservoir Simulation.* Applied Science Publishers, New York, 1979.

[11] I. Babuška and A. Miller. A feedback finite element method with a posteriori error estimation. I. The finite element method and some basic properties of the a posteriori error estimator. *Comput. Methods Appl. Mech. Engrg.*, 61(1):1–40, 1987.

[12] I. Babuška and W. C. Rheinboldt. Adaptive finite element processes in structural mechanics. In *Elliptic problem solvers, II (Monterey, Calif., 1983)*, pages 345–377. Academic Press, Orlando, FL, 1984.

[13] R. E. Bank and D. J. Rose. Some errors estimates for the box method. *SIAM J. Numer. Anal.*, 24(4):777–787, 1987.

[14] J. P. Berrut and L. N. Trefethen. Barycentric Lagrange interpolation. *SIAM Rev.*, 46(3):501–517 (electronic), 2004.

[15] J. H. Bramble, R. E. Ewing, J. E. Pasciak, and A. H. Schatz. A preconditioning technique for the efficient solution of problems with local grid refinement. *Comput. Methods Appl. Mech. Eng.*, 67(2):149–159, 1988.

[16] L. V. Branets and V. A. Garanzha. Distortion measure of trilinear mapping. Application to 3-D grid generation. *Numer. Linear Algebra Appl.*, 9(6-7):511–526, 2002. Preconditioned robust iterative solution methods, PRISM '01 (Nijmegen).

[17] Z. Cai, J. Jr. Douglas, and P. Moongyu. Development and analysis of higher order finite volume methods over rectangles for elliptic equations. *Advances in Computational Mathematics.*, 2003.

[18] H. Cao. Development of Techniques for General Purpose Simulators. *PhD Thesis, The Stanford University*, 2002.

[19] B. Carpentieri, I. S. Duff, and L. Giraud. A Class of Spectral Two-Level Preconditioners. *SIAM Journal on Scientific Computing*, 25(2):749–765, 2003.

[20] A. Castellini. Flow based grid for reservoir simulation. *Master Thesis, The Stanford University*, 2001.

[21] Z. Chen. Formulations and numerical methods of the black oil model in porous media. *SIAM J. Numer. Anal.*, 38(2):489–514, 2000.

[22] Z. Chen and S. Dai. On the efficiency of adaptive finite element methods for elliptic problems with discontinuous coefficients. *SIAM J. Sci. Comput.*, 24(2):443–462 (electronic), 2002.

[23] Z. Chen and R. Ewing. Mathematical analysis for reservoir models. *SIAM J. Math. Anal.*, 30(2):431–453, 1999.

[24] Z. Chen and R. E. Ewing. Fully discrete finite element analysis of multiphase flow in groundwater hydrology. *SIAM J. Numer. Anal.*, 34(6):2228–2253, 1997.

[25] Z. Chen and R. E. Ewing. Stability and convergence of a finite element method for reactive transport in ground water. *SIAM J. Numer. Anal.*, 34(3):881–904, 1997.

[26] Z. Chen and R. E. Ewing. Degenerate two-phase incompressible flow iv: Local refinement and domain decomposition. *J. Sci. Comput.*, 18(3):329–360, 2003.

[27] P. G. Ciarlet. *Finite Element Method for Elliptic Problems.* Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2002.

[28] H. K. Dahle, M. S. Espedal, R. E. Ewing, and O. Sævareid. Characteristic adaptive subdomain methods for reservoir flow problems. *Numer. Methods Partial Differential Equations*, 6(4):279–309, 1990.

[29] M. Delanaye, Ch. Hirsch, and K. Kovalev. Untangling and optimization of unstructured hexahedral meshes. *Zh. Vychisl. Mat. Mat. Fiz.*, 43(6):845–853, 2003.

[30] S. Deng, K. Ito, and Z. Li. Three-dimensional elliptic solvers for interface problems and applications. *J. Comput. Phys.*, 184(1):215–243, 2003.

[31] J. K. Dukowicz. Efficient volume computation for three-dimensional hexahedral cells. *J. Comput. Phys.*, 74(2):493–496, 1988.

[32] Eclipse. Eclipse 100 Technical Description, Schlumberger Geoquest. *Report*, 1997.

[33] M. G. Edwards. Elimination of adaptive grid interface errors in the discrete cell centered pressure equation. *J. Comput. Phys.*, 126(2):356–372, 1996.

[34] M. G. Edwards. M-matrix flux splitting for general full tensor discretization operators on structured and unstructured grids. *J. Comput. Phys.*, 160(1):1–28, 2000.

[35] M. G. Edwards. Unstructured, control-volume distributed, full-tensor finite-volume schemes with flow based grids. *Comput. Geosci.*, 6(3-4):433–452, 2002. Locally conservative numerical methods for flow in porous media.

[36] M. G. Edwards, R. Agut, and K. Aziz. Quasi K-orthogonal streamline grids: gridding and discretization. In *SPE 49072. Proceedings, SPE Annual Technical Conference and Exhibition, New Orleans, Louisiana, September 27-30*, 1998.

[37] M. G. Edwards, R. D. Lazarov, and I. Yotov, editors. *Locally conservative numerical methods for flow in porous media.* Kluwer Academic Publishers, Dordrecht, 2002. Comput. Geosci. **6** (2002), no. 3-4.

[38] M. G. Edwards and C. F. Rogers. Finite volume discretization with imposed flux continuity for the general tensor pressure equation. *Comput. Geosci.*, 2(4):259–290 (1999), 1998.

[39] G. T. Eigestad. Reservoir simulation on heterogenous and anisotropic media with imposed flux continuity conditions for general geometries, and the inclusion of hysteresis in forward modeling. *PhD Thesis, The University of Bergen, Norway*, 2003.

[40] G. T. Eigestad, I. Aavatsmark, and M. Espedal. Symmetry and M-matrix issues for the O-method on an unstructured grid. *Comput. Geosci.*, 6(3-4):381–404, 2002. Locally conservative numerical methods for flow in porous media.

[41] G. T. Eigestad and R. A. Klausen. On the convergence of the multi-point flux approximation O-method: Numerical experiments for discontinuous permeability. *Numerical Methods for Partial Differential Equations*, 21(6):1079–1098, 2005.

[42] M. S. Eikemo and M. S. Espedal. Domain decomposition methods for a three dimensional extrusion model. In *Domain decomposition methods in sciences and engineering (Beijing, 1995)*, pages 469–476. Wiley, Chichester, 1997.

[43] D. Eppstein. Linear complexity hexahedral mesh generation. *Comput. Geom.*, 12(1-2):3–16, 1999. 12th ACM Symposium on Computational Geometry (Philadelphia, PA, 1996).

[44] B. G. Ersland and M. S. Espedal. A domain decomposition method for heterogeneous reservoir flow. In *Domain decomposition methods in sciences and engineering (Beijing, 1995)*, pages 477–484. Wiley, Chichester, 1997.

[45] B. G. Ersland, M. S. Espedal, and R. Nybø. Numerical methods for flow in a porous medium with internal boundaries. *Comput. Geosci.*, 2(3):217–240, 1998.

[46] M. S. Espedal, K. J. Hersvik, and B. G. Ersland. Domain decomposition methods for flow in heterogeneous porous media. In *Domain decomposition methods, 10 (Boulder, CO, 1997)*, volume 218 of *Contemp. Math.*, pages 104–120. Amer. Math. Soc., Providence, RI, 1998.

[47] M. S. Espedal and K. H. Karlsen. Numerical solution of reservoir flow models based on large time step operator splitting algorithms. In *Filtration in porous media and industrial application (Cetraro, 1998)*, volume 1734 of *Lecture Notes in Math.*, pages 9–77. Springer, Berlin, 2000.

[48] R. E. Ewing, R. D. Lazarov, and P. S. Vassilevski. Local refinement techniques for elliptic problems on cell-centered grids. I. Error analysis. *Math. Comp.*, 56(194):437–461, 1991.

[49] R. E. Ewing, R. D. Lazarov, and P. S. Vassilevski. Local refinement techniques for elliptic problems on cell-centered grids. III. Algebraic multilevel BEPS preconditioners. *Numer. Math.*, 59(5):431–452, 1991.

[50] R. E. Ewing, R. D. Lazarov, and P. S. Vassilevski. Local refinement techniques for elliptic problems on cell-centered grids. II. Optimal order two-grid iterative methods. *Numer. Linear Algebra Appl.*, 1(4):337–368, 1994.

[51] R. E. Ewing, J. Shen, and J. Wang. Point-distributed algorithms on locally refined grids for second order elliptic equations. pages 37–51, 2001.

[52] C. J. Fitzsimons, J. J. H. Miller, S. Wang, and C. H. Wu. Hexahedral finite elements for the stationary semiconductor device equations. *Comput. Methods Appl. Mech. Engrg.*, 84(1):43–57, 1990.

[53] G. E. Fladmark. Secondary Oil Migration. Mathematical and Numerical Modelling in SOM Simulator. *Tech. Rep. , Norks Hydro ASA, E & P Research Center, Bergen, Norway*, 1997.

[54] G. E. Fladmark. Athena as a Thermal Mass Flow and Solid Mechanics Simulator. *Lecture Notes, The University of Bergen, Norway*, May, 2005.

[55] P. A. Forsyth, Jr. and P. H. Sammon. Practical considerations for adaptive implicit methods in reservoir simulation. *J. Comput. Phys.*, 62(2):265–281, 1986.

[56] P. A. Forsyth, Jr. and P. H. Sammon. Quadratic convergence for cell-centered grids. *Appl. Numer. Math.*, 4(5):377–394, 1988.

[57] L. A. Freitag and P. M. Knupp. Tetrahedral mesh improvement via optimization of the element condition number. *Internat. J. Numer. Methods Engrg.*, 53(6):1377–1391, 2002.

[58] J. Gale. Overview of CO_2 emission sources, potential, transport and geological distribution of storage possibilities. *IPCC Workshop on Carbon Dioxide Capture and Storage*, 1, 1990.

[59] I. Garrido, G. E. Fladmark, and M. Espedal. An improved numerical simulator for multiphase flow in porous media. *Internat. J. Numer. Methods Fluids*, 44(4):447–461, 2004.

[60] I. Garrido, E. Øian, M. Chaib, G. E. Fladmark, and M. S. Espedal. Implicit treatment of compositional flow. *Comput. Geosci.*, 8(1):1–19, 2004.

[61] F. Gibou and R. Fedkiw. A fourth order accurate discretization for the laplace and heat equations on arbitrary domains, with applications to the stefan problem. *J. Comput. Phys.*, 202(2):577–601, 2005.

[62] F. Gibou, R. P. Fedkiw, L. T. Cheng, and M. Kang. A second-order-accurate symmetric discretization of the poisson equation on irregular domains. *J. Comput. Phys.*, 176(1):205–227, 2002.

[63] V. E. Ginting. *Computational upscaled modeling of heterogeneous porous media flow utilizing finite volume method.* PhD thesis, College Station, TX, USA, 2004. Chair-Raytcho Lazarov and Chair-Yalchin Efendiev.

[64] L. Giraud, D. Ruiz, and A. Touhami. A comparative study of iterative solvers exploiting spectral information for SPD systems. Technical Report TR/PA/04/40, CERFACS, Toulouse, France, 2004.

[65] T. Grätsch and K. J. Bathe. A posteriori error estimation techniques in practical finite element analysis. *Computer & Structures, in press*, 1, 2005.

[66] W. Hackbusch. On first and second order box schemes. *Computing*, 41(4):277–296, 1989.

[67] D. Hacon and C. Tomei. Tetrahedral decompositions of hexahedral meshes. *European J. Combin.*, 10(5):435–443, 1989.

[68] H. Hægland, H. K. Dahle, G. T. Eigestad, K. A. Lie, and I. Aavatsmark. Improved streamlines and time of flight for streamline simulation on irregular grids. *Submitted in Journal. Available on line at http://heim.ifi.uio.no/k̃alie/papers/cvi-sl.pdf. October 2005.*

[69] R. Hansen, M. S. Espedal, and O. Nygaard. An operator splitting technique for two-phase immiscible flow dominated by gravity and capillary forces. In *Computational methods in water resources, IX, Vol. 1 (Denver, CO, 1992)*, pages 59–67. Comput. Mech., Southampton, 1992.

[70] C. He, M. G. Edwards, and L. J. Durlofsky. Numerical calculation of equivalent cell permeability tensors for general quadrilateral control volumes. *Comput. Geosci.*, 6(1):29–47, 2002.

[71] K. J. Hersvik and M. S. Espedal. Adaptive hierarchical upscaling of flow in heterogeneous reservoirs based on an a posteriori error estimate. *Comput. Geosci.*, 2(4):311–336 (1999), 1998.

[72] N. J. Higham. The numerical stability of barycentric Lagrange interpolation. *IMA J. Numer. Anal.*, 24(4):547–556, 2004.

[73] R. C. Hornung and J. A. Trangenstein. Adaptive mesh refinement and multilevel iteration for flow in porous media. *J. Comput. Phys.*, 136(2):522–545, 1997.

[74] S. Hou and X. Liu. A numerical method for solving variable coefficient elliptic equation with interfaces. *J. Comput. Phys.*, 202(2):411–445, 2005.

[75] J. M. Hyman, S. Li, P. M. Knupp, and M. Shashkov. An algorithm for aligning a quadrilateral grid with internal boundaries. *J. Comput. Phys.*, 163(1):133–149, 2000.

[76] M. Shashkov J. Hyman, J. Morel and S. Steinberg. Mimetic finte difference methods for diffusion equations. *Comput. Geosci.*, 2002.

[77] P. Knupp. Mesh generation using vector-fields. *J. Comput. Phys.*, 119(1):142–148, 1995.

[78] P. Knupp and S. Steinberg. *Fundamentals of grid generation.* CRC Press, Boca Raton, FL, 1994. With 1 IBM-PC floppy disk (3.5 inch; HD).

[79] P. M. Knupp. On the invertibility of the isoparametric map. *Comput. Methods Appl. Mech. Engrg.*, 78(3):313–329, 1990.

[80] P. M. Knupp. The direct variational grid generation method extended to curves. *Appl. Math. Comput.*, 43(1, part I):65–78, 1991.

[81] P. M. Knupp. Intrinsic algebraic grid generation. In *Mathematical aspects of numerical grid generation*, volume 8 of *Frontiers Appl. Math.*, pages 75–97. SIAM, Philadelphia, PA, 1991.

[82] P. M. Knupp. A robust elliptic grid generator. *J. Comput. Phys.*, 100(2):409–418, 1992.

[83] P. M. Knupp. Surface grid generation in the tangent plane. *Appl. Math. Comput.*, 59(1):41–62, 1993.

[84] P. M. Knupp. Jacobian-weighted elliptic grid generation. *SIAM J. Sci. Comput.*, 17(6):1475–1490, 1996.

[85] P. M. Knupp. Algebraic mesh quality metrics. *SIAM J. Sci. Comput.*, 23(1):193–218 (electronic), 2001.

[86] P. M. Knupp and N. Robidoux. A framework for variational grid generation: conditioning the Jacobian matrix with matrix norms. *SIAM J. Sci. Comput.*, 21(6):2029–2047 (electronic), 2000.

[87] S. H. Lee, H. Tchelepski, and L. F. De Chant. Implementation of a flux continuous finite difference method for stratigraphic, hexahedron grids. In *Proceedings of the SPE Reservoir Simulation Symposium (Houston, Tex.,)*, 1999.

[88] Z. Li. A fast iterative algorithm for elliptic interface problems. *SIAM J. Numer. Anal.*, 35(1):230–254, 1998.

[89] E. E. Madsen and G. E. Fladmark. Some finite difference methods for solution of heat conduction problems. In *Numerical solution of partial differential equations (Proc. NATO Advanced Study Inst., Kjeller, 1973)*, pages 223–240. NATO Adv. Study Inst. Ser., Ser. C: Math. and Phys. Sci., 2. Reidel, Dordrecht, 1973.

[90] T. A. Manteuffel and Jr. A. B. White. The numerical solution of second-order boundary value problems on nonuniform meshes. *Math. Comput.*, 47(176):511–535, 1986.

[91] G. Matthies. Mapped finite elements on hexahedra. Necessary and sufficient conditions for optimal interpolation errors. *Numer. Algorithms*, 27(4):317–327, 2001.

[92] P. Morin, R. H. Nochetto, and K. G. Siebert. Data oscillation and convergence of adaptive FEM. *SIAM J. Numer. Anal.*, 38(2):466–488 (electronic), 2000.

[93] P. Morin, R. H. Nochetto, and K. G. Siebert. Convergence of adaptive finite element methods. *SIAM Rev.*, 44(4):631–658 (electronic) (2003), 2002. Revised reprint of "Data oscillation and convergence of adaptive FEM" [SIAM J. Numer. Anal. **38** (2000), no. 2, 466–488 (electronic); MR1770058 (2001g:65157)].

[94] P. Morin, R. H. Nochetto, and K. G. Siebert. Local problems on stars: a posteriori error estimators, convergence, and performance. *Math. Comp.*, 72(243):1067–1097 (electronic), 2003.

[95] M. Müller-Hannemann. Improving the surface cycle structure for hexahedral mesh generation. In *Proceedings of the Sixteenth Annual Symposium on Computational Geometry (Hong Kong, 2000)*, pages 19–28 (electronic), New York, 2000. ACM.

[96] M. Müller-Hannemann. Shelling hexahedral complexes for mesh generation. *J. Graph Algorithms Appl.*, 5:no. 5, 59–91 (electronic), 2001.

[97] M. Müller-Hannemann. Quadrilateral surface meshes without self-intersecting dual cycles for hexahedral mesh generation. *Comput. Geom.*, 22(1-3):75–97, 2002. 16th ACM Symposium on Computational Geometry (Hong Kong, 2000).

[98] R. L. Naff, T. F. Russell, and J. D. Wilson. Shape functions for velocity interpolation in general hexahedral cells. *Comput. Geosci.*, 6(3-4):285–314, 2002. Locally conservative numerical methods for flow in porous media.

[99] J. M. Nordbotten and I. Aavatsmark. Monotonicity conditions for control volume methods on uniform parallelogram grids in homogeneous media. *Comput. Geosci.*, 9(1):61–72, 2005.

[100] J. M. Nordbotten and G. T. Eigestad. Discretization on quadrilateral grids with improved monotonicity properties. *J. Comput. Phys.*, 203(2):744–760, 2005.

[101] F. Ortega. The general mesh viewer, gmv, is an easy to use 3d scientific visualization tool designed to view simulation data from any type of structured or unstructured mesh. *Freely available at http://laws.lanl.gov/XCM/gmv/GMVHome.html*, 2005.

[102] R. B. Pember, J. B. Bell, P. Colella, W. Y. Crutchfield, and M. L. Welcome. An adaptive cartesian grid method for unsteady compressible flow in irregular regions. *J. Comput. Phys.*, 120(2):278–304, 1995.

[103] K. Pruess and J. Garcia. Solutions of Test Problems for Disposal of CO_2 in Saline aquifers. *LBNL-51812*, 2002.

[104] K. Pruess, J. Garcia, T. Kovscek, C. Oldenburg, J. Rutqvist, Steefel C., and T. Xu. Intercomparison of Numerical Simulation Codes for Geologic Disposal of CO_2. *LBNL-51813.*, December, 2002.

[105] G. Qin, H. Wang, R. E. Ewing, and M. S. Espedal. Numerical simulation of compositional fluid flow in porous media. In *Numerical treatment of multiphase flows in porous media (Beijing, 1999)*, volume 552 of *Lecture Notes in Phys.*, pages 232–243. Springer, Berlin, 2000.

[106] H. Reme, M. S. Espedal, and G. E. Fladmark. A preconditioning technique as an upscaling procedure. In *Resource recovery, confinement, and remediation of environmental hazards (Minneapolis, MN, 2000)*, volume 131 of *IMA Vol. Math. Appl.*, pages 283–296. Springer, New York, 2002.

[107] H. Reme, G. Å. Øye, M. S. Espedal, and G. E. Fladmark. Parallelization of a compositional reservoir simulator. In *Numerical treatment of multiphase flows in porous media (Beijing, 1999)*, volume 552 of *Lecture Notes in Phys.*, pages 244–266. Springer, Berlin, 2000.

[108] B. Rivière. Discontinuous galerkin fnite element methods for solving the miscible displacement problem in porous media. *PhD Thesis, The University of Texas at Austin*, 2000.

[109] B. Rivière and M. F. Wheeler. A posteriori error estimates for a discontinuous Galerkin method applied to elliptic problems. Log number: R74. *Comput. Math. Appl.*, 46(1):141–163, 2003. p-FEM2000: p and hp finite element methods—mathematics and engineering practice (St. Louis, MO).

[110] B. Rivière, M. F. Wheeler, and K. Banas. Part II. Discontinuous Galerkin Method Applied to Single Phase Flow in Porous Media. *Comput. Geosci.*, 4:337–341, 2000.

[111] O. Sævareid, H. K. Dahle, and M. S. Espedal. Domain decomposition for reservoir flow problems. In *Third International Symposium on Domain Decomposition Methods for Partial Differential Equations (Houston, TX, 1989)*, pages 481–491. SIAM, Philadelphia, PA, 1990.

[112] P. H. Sammon. Numerical approximations for a miscible displacement process in porous media. *SIAM J. Numer. Anal.*, 23(3):508–542, 1986.

[113] R. Schneiders. Octree-based hexahedral mesh generation. *Internat. J. Comput. Geom. Appl.*, 10(4):383–398, 2000. Selected papers from the Sixth International Meshing Roundtable, Part II (Park City, UT, 1997).

[114] B. Smith, P. Bjørstad, and W. Gropp. *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations.* Cambridge University Press, New York, USA, 1996.

[115] G. Strang and G. J. Fix. An analysis of the finite element method. *Wiley New York*, 1973.

[116] E. Süli. Convergence of finite volume schemes for Poisson's equation on nonuniform meshes. *SIAM J. Numer. Anal.*, 28(5):1419–1430, 1991.

[117] R. Teigland and G. E. Fladmark. Cell-centered multigrid methods in porous media flow. In *Multigrid methods, III (Bonn, 1990)*, volume 98 of *Internat. Ser. Numer. Math.*, pages 365–376. Birkhäuser, Basel, 1991.

[118] O. V. Ushakova. Conditions of nondegeneracy of three-dimensional cells. A formula of a volume of cells. *SIAM J. Sci. Comput.*, 23(4):1274–1290 (electronic), 2001.

[119] R. Verfürth. A posteriori error estimation and adaptive mesh-refinement techniques. In *Proceedings of the Fifth International Congress on Computational and Applied Mathematics (Leuven, 1992)*, volume 50, pages 67–83, 1994.

[120] H. A. van der Vorst. *Iterative Krylov Methods for Large Linear Systems.* Cambridge monographs on applied and computational mathematics. Cambridge University Press, New York, 2003.

[121] A. Weiser and M. F. Wheeler. On convergence of block-centered finite differences for elliptic-problems. *SIAM J. Numer. Anal.*, 25(2):351–375, 1988.

[122] J. A. Wheeler, M. F. Wheeler, and I. Yotov. Enhanced velocity mixed finite element methods for flow in multiblock domains. *Comput. Geosci.*, 6(3-4):315–332, 2002. Locally conservative numerical methods for flow in porous media.

[123] A. Winslow. Equipotential zoning of two dimensional meshes. *Technical Report UCRL-7312, Lawrence Livermore National Laboratory*, 1963.

[124] A. Winslow. Numerical solution of the quasilinear poisson equation in a nonuniform triangle mesh. *J. Comput. Phys.*, 1:149–172, 1967.

[125] X. H. Wu and R. R. Parashkevov. Effect of Grid Deviation on Flow Solutions. *SPE 92868*, 2005.

[126] S. Yamakawa and K. Shimada. Fully-automated hex-dominant mesh generation with directionality control via packing rectangular solid cells. *Internat. J. Numer. Methods Engrg.*, 57(15):2099–2129, 2003.

[127] K. Y. Yuan, Y. S. Huang, H. T. Yang, and T. H. H. Pian. The inverse mapping and distortion measures for 8-node hexahedral isoparametric elements. *Comput. Mech.*, 14(2):189–199, 1994.

[128] P. Zweigel, M. Hamborg, R. Arts, A. Lothe, and A. Tømmerås. Prediction of migration of CO_2 injected into an underground depository: Reservoir geology and migration modelling in the Sleipner case (North Sea). *5th International Conference on Greenhouse Gas Control Technologies, Cairns (Australia)*, August, 2000.