# Non-Linear Finite Volume discretization for Subsurface Flow and Mechanics problem

by

Tripuraneni Sree Rama Teja

to obtain the degree of Master of Science

at the Delft University of Technology,

to be defended publicly on Friday July 23rd, 2021 at 12:00 AM.

| | | |
|---|---|---|
| Student number: | 5012279 | |
| Project duration: | November 1, 2020 – July 20, 2021 | |
| Supervisors: | Prof. Dr. D.V. Voskov, | TU Delft |
| | A. Novikov, | TU Delft |
| Committee Members: | Prof. Dr. D.V. Voskov, | TU Delft |
| | Prof. Dr. H. Hajibeygi, | TU Delft |
| | Prof. Dr. M. Möller, | TU Delft |
| | A. Novikov, | TU Delft |




Delft University of Technology

# Abstract

Energy transition extends the range of geological settings and physical processes to be taken into account in subsurface reservoir modelling. Many of these applications consider essentially anisotropic reservoir or require advanced gridding that can not be resolved consistently by conventionally used Two Point Flux Approximation (TPFA).

In this project we present a Nonlinear Two Point Flux Approximation (NTPFA) based on gradient reconstruction and homogenization function. The approximation provides consistent solution for full permeability tensor on various grids. The approach combines flux guesses in a nonlinear way such that the obtained approximation is essentially monotone that guarantees the positivity of solution. We demonstrate the consistency of approach on several examples. We also use the multi-physics capabilities to test the simulator on saturation transport of dead oil when displaced with water. The developed approximation was implemented within Delft Advanced Research Terra Simulator (DARTS).

Discrete fracture network (DFN) is also incorporated into this framework, which can be used to simulate fluid flow through lower dimensional elements in our mesh. The main requirement for DFN is the mesh is aligned with the fracture so that we can treat these interfaces on fractures as fractures themselves. Permeability in fractures is considered as scalar (along the fracture) unlike the full tensor in the matrix. We also use linear TPFA approximation for fracture-fracture and matrix-fracture connections.

Next we propose a new Nonlinear Two Point Stress Approximation technique which follows the collocated finite volume scheme for mechanical problem. In this section we try to discretize the linear elasticity equation by using nonlinear traction flux at interfaces similar to the setup used in fluid flow problem. This is done by balancing each component of traction individually and using the weighting scheme suggested in flux approximations. The gradient reconstruction is also different from the one used in flux approximation as we have to reconstruct 3 gradients of displacements simultaneously such that we obtain $3 \times 3$ matrix for transmissibilities in front of displacement vectors involved in our stencil instead of scalar values.

# Acknowledgements

I am very thankful to my professor and supervisor Dr. Denis Voskov, for giving me the opportunity to be a member of the DARTS development group. Over the course of this master's degree, I have not only learned a great deal about reservoir simulation but also how to tackle a problem from a researcher's point of view. After stepping outside of my home country for the first time, I had a nervous start to this master program, despite that you trusted in my capabilities and handed me an opportunity to start the honor's project. Your persistence was crucial to bring this project to reality.

I am profoundly grateful to Dr. Hadi Hajibeygi and Dr. Matthias Möller for providing me with valuable feedback on this project. Dr. Hadi's lectures on reservoir simulation undeniably increased my curiosity in this domain and helped me to reinforce my interest in physics.

Moreover, this project would not have been nearly as successful without the help of Aleks Novikov. The technical understanding you possess came in clutch during the whole period of the project and interacting with you almost every day kept me motivated during these tough times. Your professionalism as a Ph.D. candidate set a high bar for me when I pursue the same. Thank you for everything.

Finally, I would like to thank my mother (Sridevi), father (Rama Rao), sister (Ramya), and all my friends back at home for guiding me even though we are several miles apart. This project is dedicated to all of you.

**Thank you all for your support, guidance, and valuable feedback!**

# Contents

# Nomenclature

| | |
|---|---|
| $\mathbb{S}$ | Material compliance Tensor |
| $\alpha, \beta, \gamma$ | Boundary coefficient |
| $\mathbf{f}$ | Source/Sink term |
| $\mathbf{u}_\alpha$ | Phase Darcy velocity |
| $\boldsymbol{x}$ | Cell Collocation point |
| $\boldsymbol{x}_\sigma$ | Interface Collocation point |
| $\mathbb{A}, \mathbb{B}$ | Stiffness sub-matrices |
| $\mathbb{I}$ | $3 \times 3$ identity matrix |
| $\mathbb{K}$ | Permeability Tensor |
| $\mathbb{R}_{xyz}$ | Rotation matrix |
| $\mathbb{T}_1$ | Transmissibility coefficient |
| $\mathcal{H}_{1,i}^{\Sigma}$ | Homogenization Function |
| $\mu_i$ | Convex combination weight |
| $\mu_o$ | Oil viscosity |
| $\mu_w$ | Water viscosity |
| $\nabla p_i$ | Pressure gradient in cell |
| $\nabla u, \nabla v, \nabla w$ | Displacement gradients |
| $\phi$ | Rock porosity |
| $\rho_o$ | Oil density |
| $\rho_w$ | Water density |
| $\sigma$ | Stress Tensor |

$A_i$ Area of interface

$cr$ Convergence rate

$E$ $6 \times 6$ material stiffness tensor

$e_p$ $L^2$ norm of pressure error

$e_u$ $L^2$ norm of displacement error

$k_{fi}$ Scalar permeability

$n_\sigma$ Normal vector of interface

$p$ Pore Pressure

$q^\sigma$ Total flux across interface

$q_i^\sigma$ Semi flux across interface

$r$ Perpendicular distance cell to interface

$S_\alpha$ Phase saturation

$u, v, w$ Displacement vector

$V_i$ Volume of cell

# Chapter 1

# Introduction

## 1.1 Background and Motivation

The Finite Volume Method (FVM) provides a convenient framework in various engineering applications where advection, diffusion, and other types of physics drive nature. The conservation laws involved within the domain are represented by a partial differential equation (PDE) which are approximated by the FVM to obtain the solution of an unknown variable for any point in space and time. The main advantage of FVM is its local conservation property i.e. in the field of fluid mechanics, we can balance the fluxes across interfaces of the control volumes (CV) which then helps us conserve the mass of fluid moving in and out of control volume locally. This specific property is not natural in Finite Element Methods. We can also take advantage of an arbitrary grid on which we can find the response to our problem. (Kolditz, 2002)

In this thesis, we start from using the FVM framework for subsurface reservoir modelling. In reservoir simulation, FVMs have been used for many years to estimate how oil, gas, or water flows through the underground porous media and what kind of flow rates we can expect when we inject or produce (K. Aziz, 1979). Reservoir simulation pertaining to industrial standard uses a Two-Point Flux Approximation (TPFA) where flux across an interface between two control volumes is usually approximated by the pressure difference across these two cells (K. Aziz, 1979). But this approximation is not always accurate which in turn gives rise to a solution that does not make sense physically.

Especially when we look into deep subsurface oil and gas simulation, the grids we have are often highly distorted because of the presence of impermeable zones, fractures, faults, or altogether. These features not only complicate the geological model but also bring additional sensitivity. This sensitivity is related to the dynamic response of the reservoir to any perturbations in the geology. From a mathematical point of view, we can attribute these features to a heterogeneous permeability tensor which characterizes how easy fluid flows in porous media. Due to the presence of this heterogeneous and anisotropic permeability ten-

sor, the conventional TPFA will not be a representative approximation of flux and gives up convergence. To tackle this problem there have been numerous theories and developments, which help us conceptually understand how fluid flow is controlled by these complexities.

It started with the development of Multi-Point flux approximation (MPFA) schemes (Aavatsmark, 2002; I. Aavatsmark, 2008; Nordbotten, 2007) where we utilize more than two cells for approximating flux across the interface between control volumes. The MPFA method generally gives an accurate approximation of flux across the interface where all or most cells surrounding the interface are considered. However, in many cases, MPFA methods are known to be conditionally monotone according to (Kershaw, 1981) and violate the discrete maximum principle (DMP) in extreme cases. This violation often takes the form of spurious oscillations in our numerical solution across the grid. There has been extensive research associated with how we can minimize these oscillations and make our MPFA discretization more robust.

To tackle this problem, a nonlinear formulation of the discrete flux equation was developed by (Le Potier, 2005; Bertolazzi and Manzini, 2004) and then modified by various researches (Kapyrin, 2007; Danilov and Vassilevski, 2009; Nikitin et al., 2013; Gao and Wu, 2015). In this formulation, the linear elliptical equation is transformed to a nonlinear form such that the scheme becomes monotone. The idea of nonlinear Finite Volume (FV) approaches is that the flux approximations should have non-negative coefficients in front of unknowns which is pressure in our case.

This approach was good enough for cases where there was a single discontinuity surrounding a specific control volume (Lipnikov et al., 2009). When there was a fully heterogeneous domain i.e. all cells surrounding a CV having different permeability tensors, the methods in (Lipnikov et al., 2009; Vassilevski et al., 2020) which are based on finite differences and proved to be unsuccessful to approximate the flux accurately. Another kind of approach involves using vertex interpolation of pressure to construct a positive basis for gradient reconstruction (Yuan and Sheng, 2008; Gao and Wu, 2015). But considering vertices will introduce additional unknowns to the discretization equation which becomes computationally expensive. In the lines of the above research, the concept of Harmonic averaging is developed (Agelas et al., 2009) and successfully extended to multi-phase and multi-dimensional problems (Terekhov et al., 2016; Schneider et al., 2018) in the past few years.

Now after the MPFA procedure gained attention because of its ability to handle anisotropic discontinuous full permeability tensor, there have been attempts to formulate a similar type of Finite volume framework for linear elasticity problem in (Tuković et al., 2013; Nordbotten, 2014; Keilegavlen and Nordbotten, 2017; Terekhov and Tchelepi, 2020). Although the research in the field of simulation of momentum equations by Finite Elements (FE) is much mature, the advantage of using a FV framework in mechanics is the ability to resolve a coupled poro-mechanical problems by a unified FV framework. The local conservation guaranteed in the FV framework makes it more feasible to shift mechanics to FV rather than flow to FE.

This brings us to the main motivation for creating this project. In the research related to collocated FVM in mechanics, there are certain types of oscillations in traction observed on a fracture (considered to be made of no-slip interfaces). These oscillations are attributed to the violation of monotonicity in our MPSA discretization. In this study, we wanted to develop a discretization procedure that can give us a smooth solution or at least improve the MPSA framework. So, we implement a Non-linear TPFA discretization, specifically using the gradient reconstruction procedure suggested in (Terekhov et al., 2016) which utilizes a homogenization function to reconstruct pressure gradient in any arbitrary direction. This gave us insights on how to implement the same for linear elastic material using finite volume as our base. Hence we tried to introduce a nonlinear discretization framework for a purely mechanical problem which from past experiences can deal with monotonicity.

Within this project, we looked into the flow and mechanical aspects separately. We can address a coupled problem in the future.

## 1.2 Project Objectives

The project sets the following goals

- To develop a nonlinear discretization framework for flow in subsurface porous media within DARTS platform
- To introduce a novel nonlinear two-point stress approximation (NTPSA) and to demonstrate its applicability to the solving of elasticity problem
- To investigate the numerical properties of developed nonlinear approximations and to validate them in a number of test cases.

## 1.3 Thesis Structure

Chapter 2 begins with delineating the mass and momentum conservation equations and physics considered in our simulation platform. We present the equations that govern the processes which can occur in subsurface flows. Bear in mind that we will consider flow and mechanics separately. Then in chapter 3, we delineate the properties of different discretization schemes and show the complexities which can arise for industrial standard simulators by using simple approximations. In Chapter 4, we derive the equations for gradient reconstruction and nonlinear weighting for flow. Chapter 5 covers a similar derivation for the elasticity problem. This is followed by several numerical tests where we prove the $2^{nd}$ order convergence of the scheme with respect to primary unknowns and demonstrate the consistency of the scheme in a few challenging examples. Finally, we discuss the limitations we encountered in this development and what can be followed from this work.

# Chapter 2

# Problem Formulation

In this chapter, we present the type of physics involved in subsurface reservoir engineering application and relevant conservation laws which can be applied to support the physics of our domain. We also consider the momentum equation, which is used for a purely mechanical problem.

## 2.1 Flow in Porous Media

As already mentioned, we consider two main types of problems in flow namely the stationary single-phase flow which is governed by an elliptical PDE, and a multi-phase flow problem, which has a hyperbolic component. We can also consider gravity and capillarity on a condition that a relevant nonlinear discretization framework is provided for them.

### Single phase flow

In this section, we provide discretization for the elliptical flow equation which is represented in our domain by the following formula.

$$-\nabla \cdot \mathbf{u} = \mathbf{f} \quad \textbf{in} \quad \boldsymbol{\Omega}, \qquad \alpha p + \beta \mathbf{n} \cdot \mathbf{u} = \gamma \quad \textbf{on} \quad \partial\boldsymbol{\Omega}, \tag{2.1}$$

Where $\boldsymbol{\Omega}$ represents our three-dimensional domain, $\partial\boldsymbol{\Omega}$ represents the boundary of our domain, $\mathbf{u}$ is the Darcy's velocity, $\mathbf{f}$ represents source/sink and $p$ is the pressure in our domain. Two different types of boundary conditions, Dirichlet ($\alpha = 1, \beta = 0$) and Neumann ($\alpha = 0, \beta = 1$) are considered within (2.1). We can also use mixed boundary conditions when both $\alpha$ and $\beta$ are non-zero.

The equation (2.1) fundamentally represents the mass conservation of an incompressible single-phase flow problem in an incompressible porous medium. We need additional infor-

mation related to fluid flow in porous media which will be defined by Darcy's law.

$$\mathbf{u} = -\frac{\mathbb{K}}{\mu}\nabla p, \tag{2.2}$$

Where $\nabla p$ is the gradient of pressure which allows fluid to move with a velocity $\mathbf{u}$, $\mu$ is the viscosity of the fluid and $\mathbb{K}$ is the permeability tensor.

## Two phase flow

In this formulation, we consider a two-phase two-component flow which assumes immiscible phases (each phase do not dissolve into one another). Water flooding into an oil reservoir is a common case which can be represented by this model assuming the oil content is relatively high and does not liberate gas once pressure decreases.

The mass conservation equation can be represented as:

$$\frac{\partial(\phi\rho_{\mathfrak{a}}S_{\mathfrak{a}})}{\partial t} - \nabla\cdot(\rho_{\mathfrak{a}}\mathbf{u}_{\mathfrak{a}}) = f_{\mathfrak{a}} \quad \textbf{in} \quad \boldsymbol{\Omega}, \quad \mathfrak{a} = w, o, \tag{2.3}$$

where we have two phases $w, o$ as water and oil respectively. The boundary conditions are similar to equation (2.1) and $\mathbf{u}_{\mathfrak{a}}$ is the phase Darcy velocity, which will be different for both phases. In addition to the fluid flow caused by a pressure gradient, we can have gravity and capillary forces acting on the fluids which can impact the direction of fluid fronts. For that we can consider the following equations:

$$\mathbf{u}_{\mathfrak{a}} = -\lambda_{\mathfrak{a}}\mathbb{K}\nabla p_{\mathfrak{a}} + \mathbf{u}_{\mathfrak{a}g}, \quad \lambda_{\mathfrak{a}} = \frac{k_{r\mathfrak{a}}}{\mu_{\mathfrak{a}}}, \tag{2.4}$$

$$S_o + S_w = 1, \tag{2.5}$$

where

$$\mathbf{u}_{\mathfrak{a}g} = \lambda_{\mathfrak{a}}\mathbb{K}(g\nabla z), \qquad p_c = p_o - p_w. \tag{2.6}$$

Here $\lambda_{\mathfrak{a}}$ is the mobility of phase $\mathfrak{a}$ which depends on phase viscosity $\mu_{\mathfrak{a}}$, and relative phase permeability $k_{r\mathfrak{a}}$. The phase viscosities and phase densities depend on the pressure of the phase $p_{\mathfrak{a}}$ i.e. an assumption that phases are compressible is valid. The relative phase permeabilities depend on the saturation of the corresponding phase $S_{\mathfrak{a}}$ and can behave as highly nonlinear functions. For this project, we assume the phases are slightly compressible and we use Corey's relationship for phase relative permeabilities which are defined as follows:

$$\rho_{\mathfrak{a}} = \rho_{\mathfrak{a}0}\Big(1 + C_{\mathfrak{a}}(p_{\mathfrak{a}} - p^0_{\mathfrak{a}})\Big), \quad \mu_{\mathfrak{a}} = \mu_{\mathfrak{a}}(p_{\mathfrak{a}}), \tag{2.7}$$

$$k_{r\mathfrak{a}} = k^0_{r\mathfrak{a}}\frac{S_{\mathfrak{a}} - S_{\mathfrak{a}r}}{1 - S_{wr} - S_{or}}. \tag{2.8}$$

In equation (2.7), $C_{\mathfrak{a}}$ is the phase compressibility, $S_{\mathfrak{a}r}$ denotes the residual saturation of the specific phase and $k^0_{r\mathfrak{a}}$ represents the maximum relative permeability of phase $\mathfrak{a}$. Furthermore, $\mathbf{u}_{\mathfrak{a}g}$ is the component of Darcy velocity due to gravity and $p_c$ is the capillary pressure which will depend on phase saturation.

## 2.2 Mechanics

Here we consider equations and physics involved in the mechanics problem. We assume the material is elastic, which represents the mechanical properties of unsaturated subsurface formations. We do not couple it with the fluid flow. The momentum balance equation looks as follows:

$$\begin{aligned} -\nabla \cdot \sigma &= \mathbf{f} \quad in \quad \Omega, \\ \alpha \boldsymbol{u}_b + \beta P(\sigma \cdot \mathbf{n}) &= \mathbf{r} \quad in \quad \partial\Omega, \end{aligned} \tag{2.9}$$

Here $\sigma$ is the stress tensor which is a $3 \times 3$ matrix, $\mathbf{f}$ is analogous to source/sink term in equation (2.1) which corresponds to external force vector $3 \times 1$ and in the second part of equation (2.9) is applied on the boundaries and specifically used to treat Dirichlet and Neumann conditions where $\alpha$ and $\beta$ values are similar to values used in equation (2.1) , $u_b$ is the displacement at the boundary interface, $P$ is a $3 \times 3$ projection operator and $\mathbf{r}$ is similar to $\gamma$ in equation (2.1). We use Dirichlet, Neumann and Roller boundary condition which are discussed more detailed in chapter 5. $\Omega$ and $\partial\Omega$ are cell and boundary domains as considered previously.

We also need a stress-strain relationship which has the same purpose as equation (2.2), so we consider Hooke's law in a continuous elastic medium as:

$$\begin{aligned} \sigma = \mathbb{C} : \epsilon = \mathbb{C} : \frac{\nabla \mathbf{u} + \nabla \mathbf{u}^T}{2}, \\ \left[\frac{\nabla \mathbf{u} + \nabla \mathbf{u}^T}{2}\right] = \mathbb{S} : \sigma. \end{aligned} \tag{2.10}$$

Here, the first part of equation (2.10) is represented in tensorial form with $\mathbb{C}$ being $4^{th}$ order material stiffness tensor, $\epsilon$ is the strain tensor. This equation can be rearranged in the format represented in second part of (2.10) where $\nabla \mathbf{u}$ is the gradient of displacement (also $3 \times 3$ tensor) and $\mathbb{S}$ is the material compliance tensor of rank 4.

## 2.3 DARTS Framework

The current discretization framework is implemented on DARTS platform where we can model different types of subsurface applications relevant to energy transition. For example, in case of geothermal applications, we use energy and mass balance to describe the problem. The equations for geothermal considered in (Wang et al., 2020), looks as follows:

$$\begin{aligned} \frac{\partial(\phi \rho_w)}{\partial t} + \nabla.(\rho_w u_w) + \rho_w q = 0, \\ \frac{\partial(\phi \rho_w U_w + (1-\phi) U_r)}{\partial t} + \nabla.(h_w \rho_w u_w) - \nabla.(\kappa \nabla T) = 0. \end{aligned} \tag{2.11}$$

One of the distinctive features of DARTS platform is Operator based linearization (OBL), where we can parameterize the parameter space (Voskov, 2017) which is usually consists of

the primary variable dependant functions like density, viscosity mentioned in (2.4). This form of parameterization is particularly effective when we want to compute Jacobian in nonlinear loop.

## 2.4 Role of Nonlinear Discretization

As we can see above all our main conservation equations (2.1), (2.3), (2.9) have a distinctive elliptic part (the divergence of flux). A nonlinear discretization comes into play when we are treating the divergence part, as we transform the flux divergence through control volume using Stokes theorem to fluxes through the interfaces.

$$-\int_V \nabla \cdot (\mathbb{K}\nabla p)\mathbf{d}V = -\int_{dV} \mathbf{n}_\delta \cdot (\mathbb{K}\nabla p)\mathbf{d}S = \sum_{\delta \in F(V)} \int_\delta \mathbf{n}_\delta \cdot (\mathbb{K}\nabla p)\mathbf{d}S = \sum_{\delta \in F(V)} |\delta| \, (\mathbb{K}\nabla p) \cdot \mathbf{n}_\delta. \quad (2.12)$$

As formulated in equation (2.12) for the elliptical flow equation in (2.1) the fluxes corresponding to interface $\delta$ can be expressed in a nonlinear way such that we can obtain a physically meaningful approximation when we take arbitrary heterogeneity or extreme anisotropy into account. The same form of equation (2.12) will be derived for the momentum equation in chapter-5.

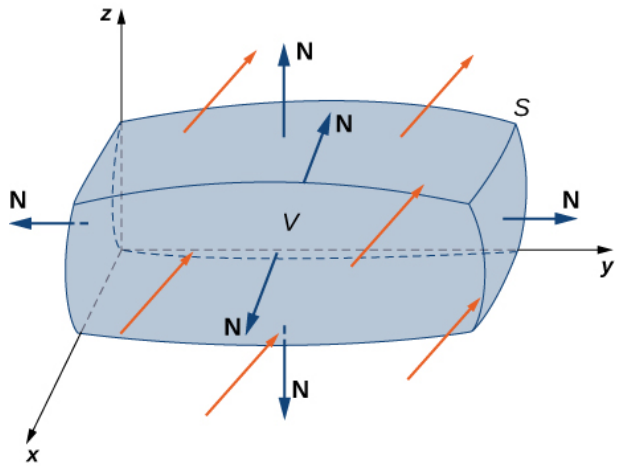



Figure 2.1: Divergence of flux (orange) through a control volume with surfaces (blue), Figure courtesy https://cnx.org (calculustoprod)

# Chapter 3

# Finite Volume Discretization

In this chapter, we discuss some details relevant to different discretization methods. But before we go into details, we first look at the solution strategy imposed by finite volume formulation. As discussed in the previous section, we use the Stokes theorem to obtain the equation in the form of summation of flux through the interfaces of the domain. We can write it in matrix form which stores the information based on cell unknowns such that we can obtain the solution for the system of equations corresponding to each cell.

In our framework, we assume cells are star-shaped in 3D with respect to the collocation point, and each interface is star-shaped with respect to the face center. The main difference between different kinds of discretization techniques comes into play when we need to describe the flux through the interface. For the most part, when we have a structured grid and the permeability tensor is aligned with the orientation of the grid, we can calculate the flux across the interface by using the pressure or concentration in two cells at either side of the interface.

## 3.1 K-Orthogonality

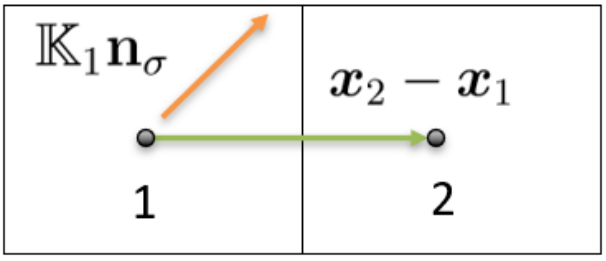



Figure 3.1: Two cell problem where different directions of $\mathbb{K} \cdot \boldsymbol{n}$ and $(\boldsymbol{x}_2 - \boldsymbol{x}_1)$ can be observed

Before we discuss any nonlinear or multi-point discretization techniques, we need to describe

what K-orthogonality means. For a generic flow problem with permeability tensor, we can represent flux in the following way (usually used in linear TPFA discretization).

$$q = \frac{\mathbb{K}\boldsymbol{n} \cdot (\boldsymbol{x}_2 - \boldsymbol{x}_1)}{|x_2 - x_1|} \frac{p_2 - p_1}{|x_2 - x_1|}. \tag{3.1}$$

As we can see in Figure 3.1, although the structured domain is used, the permeability tensor is not K-orthogonal i.e. the vector $\mathbb{K} \cdot \boldsymbol{n}$ is not aligned with $(\boldsymbol{x}_2 - \boldsymbol{x}_1)$. But in the TPFA equation (3.1) we take pressure gradient in the direction of $(\boldsymbol{x}_2 - \boldsymbol{x}_1)$ despite the $\mathbb{K}\boldsymbol{n}$ vector not being oriented in the same direction. Intuitively we can say flux will be in the direction of $\mathbb{K}\boldsymbol{n}$, so we need pressure gradient in that direction to get an accurate representation of flux. Generally all unstructured grids (wedges, hexahedrons etc.) are considered non K-orthogonal due to the arbitrary orientation of normal vector to $\mathbb{K}\boldsymbol{n}$. The presence of anisotropy or discontinuity (heterogeneity) in our domain just complicates the problem on an unstructured mesh.

## 3.2 Consistency

A numerical scheme is said to be consistent if the truncation error of our approximation decreases mesh resolution increases. If we consider the norm of this error w.r.t the characteristic size of discretization, we should see the decrease in the magnitude of this error at the same order we use to discretize the system.

$$\begin{aligned} \mathcal{F}_\sigma &= \bar{\mathcal{F}}_\sigma \pm \mathcal{O}(\Delta \boldsymbol{x}), \\ p_i &= \bar{p}_i \pm \mathcal{O}(\Delta \boldsymbol{x}^2). \end{aligned} \tag{3.2}$$

Here $\bar{p}_i$ and $\bar{\mathcal{F}}_\sigma$ represent exact pressure and flux, where as $p_i$ and $\mathcal{F}_\sigma$ are numerical. $\mathcal{O}$ represents the order of error.

So, in our case when we use a linear TPFA discretization, the simulator looses consistency on a non-K-orthogonal grids which will result in pressure not being second order convergent.

## 3.3 Monotonicity and Maximum Principles

In our study, we define monotonicity and discrete maximum principles as two different domains. A monotone $A$ matrix does not necessarily fulfil the DMP principle. More investigation into extrema preserving schemes can be found in (Cancès et al., 2013; Sheng and Yuan, 2011; Terekhov et al., 2016; Schneider et al., 2018; Vassilevski et al., 2020). For now we consider the equation (3.3) where $A$ matrix is assembled by taking all the connections which are used to define divergence of flux in all cells, $q$ is the source/sink term.

$$Ap = q, \tag{3.3}$$

Generally we can characterize monotonicity as the oscillations in our solution space. This can be written in the form of Eirik Keilegavlen (2011): assuming permeability tensor is positive definite (positive eigenvalues), the boundary is smooth and we have non-negative source/sink term in our domain $\Omega$. The solution of pressure obtained from equation (3.3) should be such that there is no local maxima or minima for $x_i \subset \Omega$

$$min\big(p(x)\big) \leq p(x_i) \leq max\big(p(x)\big), \qquad \forall x; \tag{3.4}$$

where $x$ belongs to the neighbourhood of $x_i$ such that they share an interface with $x_i$ other than $x_i$ itself.

A generic linear TPFA discretizer satisfies this condition as $A$ matrix is monotone, but MPFA formulation is not considered monotone due to the presence of an extended stencil during the flux formulation. So, we can provide a nonlinear formulation that can not only be considered as an extended stencil to represent the full pressure gradient but also be structured in such a way that the $A$ matrix becomes monotone i.e. inverse of this matrix should be positive. As we cannot make this check every time one way to ensure $A$ to be monotone is to make it an M-matrix which is a special case of a monotone matrix that guarantees positivity. The structure of M-matrix define as follows (Kuzmin, 2010): we have $A^{-1} \geq 0$ and assuming $a_{ij}$ to be an element in $A$ matrix located at $i^{th}$ row and $j^{th}$ column we should have

$$\begin{aligned} a_{ii} &> 0, \quad \forall\ i \\ a_{ij} &\leq 0, \quad \forall\ j \neq i \\ \sum_j a_{ij} &\geq 0, \quad \forall\ i. \end{aligned} \tag{3.5}$$

This means an $A$ matrix is M-matrix when it has positive diagonal and non-positive off diagonal entries. The theorem (Kuzmin, 2010) also states that the 3rd inequality is strict in (3.5) for at least 1 row.

However Discrete Maximum Principle has some additional constraints which will not be discussed in this work. These constraints are met when using the nonlinear multi-point technique which usually satisfies the DMP but at the cost of too many nonlinear iterations in complex problems as demonstrated in (Schneider, 2019). So, we adapt a method which not only provides a monotone solution but converges in a minimum number of iterations.

# Chapter 4

# Discretization of Flow Equation

Continuing from chapter 3, we will derive a nonlinear formulation to represent flux such that contributions from cells additional to the two-point neighbours will be considered.

## 4.1 Nonlinear Flux Characterization

Here we repeat the derivation of non-linear TPFA carried out in (Nikitin et al., 2013) but with the use of co-normal decomposition for flow representation (Terekhov et al., 2016). The derivations presented here follow (Vassilevski et al., 2020).

First, we provide discretization for the elliptical flow equation as discussed in chapter 3, which is represented in our domain by the following formula. We assume cells are star shaped in 3D with respect to the centroid, and each interface is star shaped with respect to face center. Next, we re-write (2.1) in the following form

$$\begin{aligned} -\nabla.(\mathbb{K}\nabla p) &= \mathbf{f} \quad \textbf{in} \quad \boldsymbol{\Omega}, \\ \alpha p + \beta \mathbf{n}.(\mathbb{K}\nabla p) &= \gamma \quad \textbf{on} \quad \partial\boldsymbol{\Omega}. \end{aligned} \tag{4.1}$$

Two different types of boundary conditions, Dirichlet ($\alpha = 1, \beta = 0$) and Neumann ($\alpha = 0, \beta = 1$) can be dealt within (4.1). We can also use mixed boundary conditions when both $\alpha$ and $\beta$ are non-zero.

The fluid flow $\mathcal{F}$ over the interface can be defined as

$$\mathcal{F} \equiv \boldsymbol{n}^T \mathbb{K}\nabla p, \tag{4.2}$$

where $\boldsymbol{n}$ is a unit normal to the interface, $\mathbb{K}$ – permeability tensor at the interface, $\nabla p$ – pressure gradient at the interface.

For a Linear TPFA, the expression for flux can be represented as follows:

$$\begin{aligned} \mathcal{F}_\sigma^{TPFA} &= -T_{12}(p_2 - p_1), \\ T_{12} = \frac{T_1 T_2}{T_1 + T_2}, \quad T_j &= \frac{d_j.(\mathbb{K}_j.\boldsymbol{n}_i)A_i}{|d_j|^2} \end{aligned} \tag{4.3}$$

where $T_j$ represents the half transmissibility from cell to interface $\sigma$ and $T_{12}$ is the harmonic mean of two half transmissibilities from the two point cells.

Whereas for MPFA, the flux equation can generally be represented as:

$$\mathcal{F}_\sigma^{MPFA} = \sum_{i=1}^{n} T_i p_i \boldsymbol{n}^T \mathbb{K} \nabla p, \tag{4.4}$$

where $T_i$ is the transmissibility contribution from $i^{th}$ cell which depends on properties of cell in the grid and permeability, $n$ is the number of cells which will contribute to $\mathcal{F}_\sigma^{MPFA}$ and will depend on type of MPFA scheme we are using i.e. O, L or D.

Now let us consider the interface $\sigma$ between 1st and 2nd cells. Then following the (Terekhov et al., 2016), we require pressure and flux continuity at a specific point $\boldsymbol{x}_\sigma$ on the interface. First pressure is assumed to be piece-wise linear function in each cell, which implies we can determine the pressure anywhere in a cell as long as we know the co-ordinates of that point and pressure at a known point. Usually the pressure at the center of cell is our primary unknown and is used as a reference. The equations are as follows

$$p_1 + (\boldsymbol{x}_\sigma - \boldsymbol{x}_1)^T \cdot \nabla p_1 = p_2 + (\boldsymbol{x}_\sigma - \boldsymbol{x}_2)^T \cdot \nabla p_2, \tag{4.5}$$

$$-(\mathbb{K}_1 \boldsymbol{n}_{\boldsymbol{\sigma}})^T \cdot \nabla p_1 = -(\mathbb{K}_2 \boldsymbol{n}_{\boldsymbol{\sigma}})^T \cdot \nabla p_2, \tag{4.6}$$

where $p_1, p_2$ – pressures in the cells, $\boldsymbol{x}_1, \boldsymbol{x}_2$ – cell centers, $\mathbb{K}_1, \mathbb{K}_2$ – permeability tensors in the cells, $\boldsymbol{n}_\sigma$ – unit normal to the interface $\sigma$.

From the above two equations we need pressure gradient information in each cell to reconstruct flux $\mathcal{F}$. Using co-normal decomposition, we can decompose the pressure gradient in a cell for a cell-face pair according to the following expressions

$$\begin{aligned} \nabla p_1 &= \frac{1}{r_1}\boldsymbol{n}_\sigma(p_h - p_1) + \Big(\mathbb{I} - \frac{1}{r_1}\boldsymbol{n}_\sigma.(\boldsymbol{x}_h - \boldsymbol{x}_1)\Big)\nabla p_\tau, \\ \nabla p_2 &= \frac{1}{r_2}\boldsymbol{n}_\sigma(p_2 - p_h) + \Big(\mathbb{I} - \frac{1}{r_2}\boldsymbol{n}_\sigma.(\boldsymbol{x}_2 - \boldsymbol{x}_h)\Big)\nabla p_\tau. \end{aligned} \tag{4.7}$$

In the 2nd part of equation (4.7), the transversal pressure gradient from both sides is equal, $r_i$ is the perpendicular distance from $i^{th}$ collocation point to interface, the normal $n$ is facing same direction in both parts of equation, $x_h$ and $p_h$ are coordinates and pressure at the continuity point on interface.

## 4.2 Harmonic Averaging Points

Now we use (4.6) and (4.7), to eliminate pressure gradients and determine the value of $p_h$

$$p_h = \frac{r_2\lambda_1 p_1 + r_1\lambda_2 p_2}{r_2\lambda_1 + r_1\lambda_2} + \left(\boldsymbol{x}_h - \frac{r_1 r_2(\mathbb{K}_1 - \mathbb{K}_2)n_\sigma + r_2\lambda_1\boldsymbol{x}_1 + r_1\lambda_2\boldsymbol{x}_2}{r_2\lambda_1 + r_1\lambda_2}\right)\nabla p_\tau. \tag{4.8}$$

Here, to eliminate $\nabla p_\tau$, we can take our continuity point in the following way

$$\boldsymbol{x}_h = \frac{r_1 r_2(\mathbb{K}_1 - \mathbb{K}_2)\boldsymbol{n}_\sigma + r_2\lambda_1\boldsymbol{x}_1 + r_1\lambda_2\boldsymbol{x}_2}{r_2\lambda_1 + r_1\lambda_2}, \tag{4.9}$$

where $\lambda_1 = \boldsymbol{n}_\sigma^T\mathbb{K}_1\boldsymbol{n}_\sigma$, and $\lambda_2 = \boldsymbol{n}_\sigma^T\mathbb{K}_2\boldsymbol{n}_\sigma$.

From (4.9) we call $\boldsymbol{x}_h$ as our harmonic averaging point on interface $\sigma$ and the corresponding $p_h$ as harmonic averaging pressure. By doing the elimination in (4.9), we can represent $p_h$ in a convenient format and it does not depend on $\nabla p_\tau$. This harmonic averaging scheme was 1st introduced in (Le Potier, 2005). Now $p_h$ will be equal to

$$p_h = \frac{r_2\lambda_1 p_1 + r_1\lambda_2 p_2}{r_2\lambda_1 + r_1\lambda_2}. \tag{4.10}$$

Similarly, if we look into a boundary face, we have face center denoted as $\boldsymbol{x}_b$, with the unit normal vector $\boldsymbol{n}_b$, and utilizing the first part of (4.7) for pressure gradient in the cell connected to boundary face, we can calculate pressure at any point on the boundary as

$$p(x) = p_h + (\boldsymbol{x} - \boldsymbol{x}_h)^T \cdot \nabla p_\tau, \tag{4.11}$$

where $\nabla p_\tau$ is the tangential projection of pressure gradient estimated on the boundary interface.

Combining equations (4.11), (4.8) (4.1) for a boundary face we obtain the following relation for $p_h$ as

$$p_h = \left(\alpha + \beta\frac{\lambda_1}{r_1}\right)^{-1}\left(\gamma + \beta\frac{\lambda_1}{r_1}p_1\right) + \\ + \left(\boldsymbol{x}_h - \left(\alpha + \beta\frac{\lambda_1}{r_1}\right)^{-1}\left(\beta\mathbb{K}_1\boldsymbol{n}_b + \beta\frac{\lambda_1}{r_1}\boldsymbol{x}_1 + \alpha\boldsymbol{x}_b\right)\right)\cdot\nabla p_\tau. \tag{4.12}$$

Following the strategy in equation (4.9), to eliminate $\nabla p_\tau$ we can derive the location of harmonic averaging point as

$$\boldsymbol{x}_h = \left(\alpha + \beta\frac{\lambda_1}{r_1}\right)^{-1}\left(\beta\mathbb{K}_1\boldsymbol{n}_b + \beta\frac{\lambda_1}{r_1}\boldsymbol{x}_1 + \alpha\boldsymbol{x}_b\right). \tag{4.13}$$

And we can show that harmonic pressure will be independent of $\nabla p_\tau$ when we use the harmonic averaging point from equation (4.13)

$$p_h = \left(\alpha + \beta\frac{\lambda_1}{r_1}\right)^{-1}\left(\gamma + \beta\frac{\lambda_1}{r_1}p_1\right). \tag{4.14}$$

As we have the harmonic points and harmonic pressures from the respective points, we can determine the pressure gradient in each cell. We use two different types of gradient reconstruction where both use the cell center information to interpolate pressure gradient i.e. we will not be using pressure at nodes to determine the pressure gradient.

## 4.3 Gradient Reconstruction

We require a positive basis for gradient reconstruction in each cell-face pair that implies individual gradient reconstruction for each interface of the cell it is connected to. Before we get into details we have to define the following terms which will make our gradient reconstruction procedure easier. For an internal interface, we can define the following terms:

$$m_h = \frac{\lambda_2 r_1}{\lambda_1 r_2 + \lambda_2 r_1}, \quad \boldsymbol{l}_h = \frac{r_1 r_2 (\mathbb{K}_1 - \mathbb{K}_2)\boldsymbol{n}_\sigma}{\lambda_1 r_2 + \lambda_2 r_1}, \tag{4.15}$$

$$p_h = (1 - m_h)p_1 + m_h p_2, \quad \boldsymbol{x}_h = (1 - m_h)\boldsymbol{x}_1 + m_h \boldsymbol{x}_2 + \boldsymbol{l}_h. \tag{4.16}$$

For boundaries, these terms look like

$$\begin{gathered} m_h = \left(\alpha + \beta \frac{\lambda_1}{r_1}\right)^{-1} \alpha, \quad \boldsymbol{l}_h = \left(\alpha + \beta \frac{\lambda_1}{r_1}\right)^{-1} \left(\beta \mathbb{K}_1 \boldsymbol{n}_b + \alpha \boldsymbol{x}_b\right), \\ g_h = \left(\alpha + \beta \frac{\lambda_1}{r_1}\right)^{-1} \gamma, \end{gathered} \tag{4.17}$$

$$p_h = (1 - m_h)p_1 + g_h, \quad \boldsymbol{x}_h = (1 - m_h)\boldsymbol{x}_1 + \boldsymbol{l}_h. \tag{4.18}$$

We use equations (4.16) and (4.18) to reconstruct pressure in all cells. We start the procedure by the system of equations when we consider pressure to be piece-wise linear function, we will have to gather at least 3 equations (i.e. 3 neighbour cells or faces) to reconstruct pressure gradient in a specific cell.

$$\begin{bmatrix} (\boldsymbol{x}_{h1} - \boldsymbol{x}_1)^T \\ (\boldsymbol{x}_{h2} - \boldsymbol{x}_1)^T \\ (\boldsymbol{x}_{h3} - \boldsymbol{x}_1)^T \end{bmatrix} \nabla p_1 = \begin{bmatrix} p_{h1} - p_1 \\ p_{h2} - p_1 \\ p_{h3} - p_1 \end{bmatrix}, \tag{4.19}$$

$$Q^{-1} = \begin{bmatrix} (\boldsymbol{x}_{h1} - \boldsymbol{x}_1)^T \\ (\boldsymbol{x}_{h2} - \boldsymbol{x}_1)^T \\ (\boldsymbol{x}_{h3} - \boldsymbol{x}_1)^T \end{bmatrix}. \tag{4.20}$$

Here the matrix $Q^{-1}$ is $3 \times 3$, $\nabla p_1$ is $3 \times 1$ vector and $(p_h - p_1)^T$ is $3 \times 1$ vector for a 3D system where the gradient is being reconstructed. Generally speaking, we will have a total of 5 interfaces for wedges and 6 interfaces for hexahedrons. So, for an over determined system by taking only 3 such vectors of $Q^{-1}$ which correspond to harmonic points on 3 such faces we can determine the pressure gradient in a cell.

Whereas for boundary interfaces we have to reconstruct the gradient at the interface. We make use of face center $x_b$ and the unknown pressure $p_b$ at the boundary and represent our reconstruction in boundary interface $\sigma_b$ as follows

$$\begin{aligned}(\boldsymbol{x}_1 - \boldsymbol{x}_b) \cdot \nabla p_b &= p_1 - p_b, \\ (\boldsymbol{x}_{hi} - \boldsymbol{x}_b) \cdot \nabla p_b &= p_{hi} - p_b,\end{aligned} \tag{4.21}$$

where the $i$ in above equation corresponds to interface other than interface $\sigma_b$, which is the specific boundary interface itself. We also replace $\boldsymbol{x}_1$ and $p_1$ equations (4.15) and (4.16) to $\boldsymbol{x}_b$ and $p_b$, and $r_1$ is replace by distance from face $i$ to boundary interface center $\boldsymbol{x}_b$.

We then evaluate the $3 \times 3$ matrix $Q$ and then we can choose our interfaces such that they satisfy the following condition in equation (4.22).

$$\mathbb{K}_1 \mathbf{n}_\sigma \cdot \nabla p_1 = \mathbb{K}_1 \mathbf{n}_\sigma \cdot Q \begin{bmatrix} p_{h_1} - p_1 \\ p_{h_2} - p_1 \\ p_{h_3} - p_1 \end{bmatrix} = \begin{bmatrix} c_1 & c_2 & c_3 \end{bmatrix} \begin{bmatrix} p_{h_1} - p_1 \\ p_{h_2} - p_1 \\ p_{h_3} - p_1 \end{bmatrix}, \tag{4.22}$$

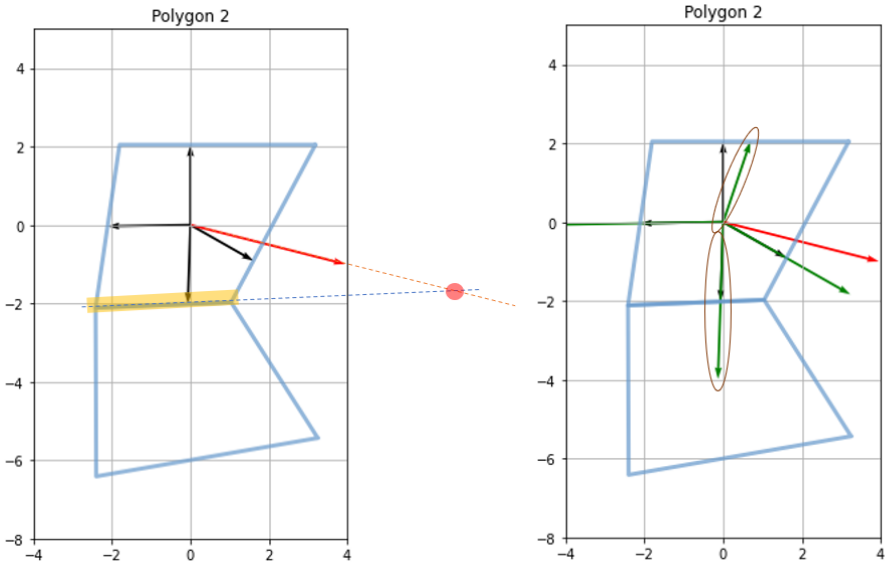
where we choose three harmonic points such that the coefficients $c_i$ in the end of equation (4.22) will be non-negative. So, naturally one of the procedure is the loop through all combinations of interfaces belonging to cell in which we are reconstructing gradient such that we obtain a non-negative basis. One of the possibilities that will always guarantee coefficients to be positive is pick the triplet of harmonic points such that $\mathbb{K}_1 \mathbf{n}_\sigma$ vector will lie in the positive basis of $Q^{-1}$ vectors.

But, in our experience, this is not a robust way to choose the gradient as we encountered some cases where even when $\mathbb{K}_1 \mathbf{n}_\sigma$ lies in the positive basis of a triplet, this specific triplet might not contain the two-point cell i.e. the other cell which shares interface $\sigma$. Even in hexahedrons we were not able to find a positive basis (with two-point cell) for cases where the permeability ratio is extreme i.e. 1000:1 and there is discontinuity in permeability tensor which gives rise to harmonic point which lies way outside the interface edges.

Here, in Figure 4.1, the $\mathbb{K}_1 \mathbf{n}_\sigma$ vector is indicated by red. For the interface denoted by yellow the harmonic point is inside the edges of the interface. Also note that this is an extruded mesh which is why we see only 4 interfaces, the interfaces at top and bottom along z-axis are parallel to each other.

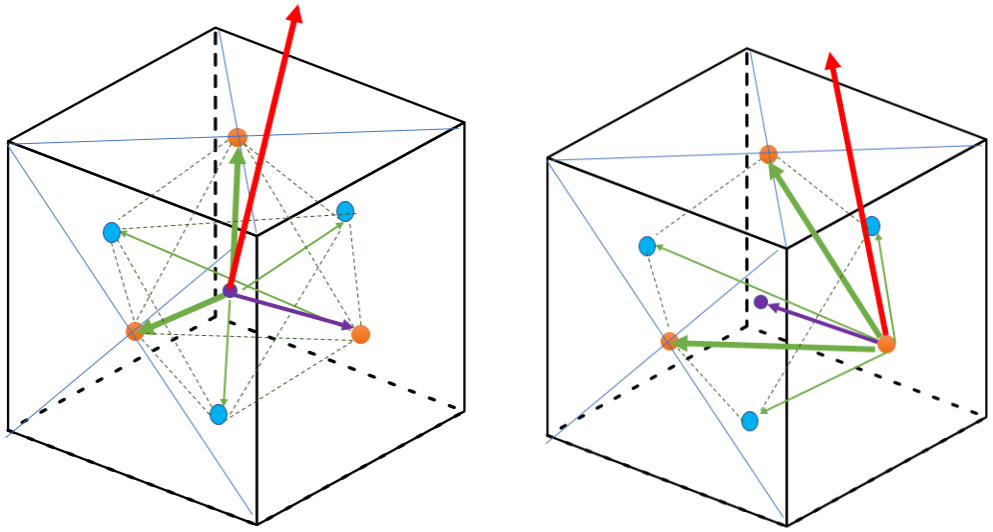
Similarly for the boundary interfaces, it is difficult for the $\mathbb{K}_1 \mathbf{n}_\sigma$ vector to lie in the basis of harmonic vectors which arise from the face center of boundary to harmonic point on other faces (except the cell itself where the vector just points towards the collocation point) that gives rise to a restrictive overall basis. In Figure 4.2 we can see in the reconstruction from boundary to cell, the $\mathbb{K}_1 \mathbf{n}_\mathbf{b}$ vector lies outside the basis formed by the harmonic vectors indicated by solid lines green and purple.

In theory, there have been some advances in eliminating the limitations when using harmonic points, which arise due to high anisotropic ratios. One such procedure is using optimization to compute transversal pressure gradient (Zhang and Al Kobaisi, 2020) and the harmonic




(a) (b)

Figure 4.1: $\mathbb{K}_1\mathbf{n}_\sigma$ vector for distorted grid (a) Harmonic vectors for each face (b) cell to cell or face center vectors



(a) (b)

Figure 4.2: $\mathbb{K}_1\mathbf{n_b}$ vector for structured grid (a) From cell to boundary (b) From boundary to cell

points can be adjusted in such a way that the basis in $\mathbb{K}_1\mathbf{n}_\sigma$ vector is not too skewed. But this will not improve our case since our harmonic point lies inside the edges of the interface and we need to adjust our $\mathbb{K}_1\mathbf{n}_\sigma$ vector itself when there is no discontinuity is present.

Two other methods were identified to overcome this issue. In (Schneider et al., 2018) the gradient reconstruction procedure is considered as an optimization problem where more than

3 faces can be present during the reconstruction process. The overall basis is considered to weaken the condition that all coefficients must be non-negative. This method coupled with extending of our basis can help to have an approximation which can still be valid for complex meshes and permeability tensors as shown in (Schneider et al., 2018).

But in this project, we wanted to adapt a more intuitive approach where a homogenization function is considered such that pressure can be interpolated across arbitrary number of faces. This will provide an opportunity to search across the whole domain i.e. cells which are no longer in the direct interface neighbourhood of the cell where we want to reconstruct the pressure gradient.

## 4.4 Homogenization Function

This function basically allows us to interpolate the pressure gradient no matter how discontinuous the permeability tensor is. We start the derivation by referring back to (4.5) and (4.6) and decompose them into normal and tangential parts as follows

$$p_1 + r_1 \nabla \mathcal{P}_{h_1} + (\boldsymbol{x}_\sigma - \boldsymbol{y}_1)^T . \nabla \mathcal{P}_\tau = p_2 - r_2 \nabla \mathcal{P}_{h_2} + (\boldsymbol{x}_\sigma - \boldsymbol{y}_2)^T . \nabla \mathcal{P}_\tau, \tag{4.23}$$

$$\lambda_1 \nabla \mathcal{P}_{h_1} + \overrightarrow{\varrho_1} . \nabla \mathcal{P}_\tau = \lambda_2 \nabla \mathcal{P}_{h_2} + \overrightarrow{\varrho_2} . \nabla \mathcal{P}_\tau, \tag{4.24}$$

where $\nabla \mathcal{P}_{h_1}$, $\nabla \mathcal{P}_{h_2}$ and $\nabla \mathcal{P}_\tau$ are defined from equation (4.7) as:

$$\nabla \mathcal{P}_{h_1} = \frac{1}{r_1} \boldsymbol{n}_\sigma (p_h - p_1), \quad \nabla \mathcal{P}_{h_2} = \frac{1}{r_1} \boldsymbol{n}_\sigma (p_h - p_2), \tag{4.25}$$

$$\nabla \mathcal{P}_\tau = \Big( \mathbb{I} - \frac{1}{r_2} \boldsymbol{n}_\sigma . (\boldsymbol{x}_2 - \boldsymbol{x}_h) \Big) \nabla p_\tau = \Big( \mathbb{I} - \frac{1}{r_1} \boldsymbol{n}_\sigma . (\boldsymbol{x}_h - \boldsymbol{x}_1) \Big) \nabla p_\tau. \tag{4.26}$$

Also, note that $\overrightarrow{\varrho} = (\mathbb{K} - \lambda \mathbb{I}) \boldsymbol{n}_\sigma$ and $y$ is the projection of collocation point on the interface.

Now we can use equation (4.24) to obtain

$$\nabla \mathcal{P}_{h_2} = \frac{\lambda_1}{\lambda_2} \nabla \mathcal{P}_{h_1} + \nabla \mathcal{P}_\tau . \frac{\overrightarrow{\varrho_1} - \overrightarrow{\varrho_2}}{\lambda_2}. \tag{4.27}$$

Using equation (4.27) in (4.23) we can derive

$$p_2 = p_1 + \nabla \mathcal{P}_{h_1} \Big( r_1 + r_2 + \frac{r_2}{\lambda_2} (\lambda_1 - \lambda_2) \Big) + \nabla \mathcal{P}_\tau . \Big( \boldsymbol{y}_2 - \boldsymbol{y}_1 + \frac{r_2}{\lambda_2} (\overrightarrow{\varrho_1} - \overrightarrow{\varrho_2}) \Big). \tag{4.28}$$

Substituting the values $\boldsymbol{x}_2 - \boldsymbol{x}_1 = (r_1 + r_2) \boldsymbol{n}_\sigma + \boldsymbol{y}_2 - \boldsymbol{y}_1$ and $(\mathbb{K}_1 - \mathbb{K}_2) n_\sigma = (\lambda_1 - \lambda_2) n_\sigma + (\overrightarrow{\varrho_1} - \overrightarrow{\varrho_2})$ in (4.28) we can group the decomposed equation again to get

$$p_2 = p_1 + \nabla p_1 \Big( \boldsymbol{x}_2 - \boldsymbol{x}_1 + \frac{r_2}{\lambda_2} (\mathbb{K}_1 - \mathbb{K}_2) \boldsymbol{n}_\sigma \Big). \tag{4.29}$$

Generalizing the equation (4.29) for any point $x \in V_2$ we get

$$p(\boldsymbol{x}) = p_1 + \nabla p_1 \Big( \big[\mathbb{I} + \frac{1}{\lambda_2}[\mathbb{K}_1 - \mathbb{K}_2]\boldsymbol{n}_\sigma \boldsymbol{n}_\sigma^T\big](\boldsymbol{x} - \boldsymbol{x}_1) - \frac{r_1}{\lambda_2}(\mathbb{K}_1 - \mathbb{K}_2)\boldsymbol{n}_\sigma \Big). \tag{4.30}$$

The resulting equation (4.30) is the generalized form of pressure in cell 2, and this forms our basis to interpolate the pressure in the next cell and cell after that. First we start by finding the derivative of $p$ w.r.t $x$ in equation (4.30). This will give us the gradient of pressure in cell 2

$$\widetilde{\nabla} p_2 = \nabla p_1 \big[\mathbb{I} + \frac{1}{\lambda_2}[\mathbb{K}_1 - \mathbb{K}_2]\boldsymbol{n}_\sigma \boldsymbol{n}_\sigma^T\big]. \tag{4.31}$$

After obtaining $\widetilde{\nabla} p_2$ and $p_2$ from equations (4.31) and (4.29) respectively, we will go back to (4.23) and (4.24), and formulate them for cells 2 and 3 which are connected together through interface $\delta$

$$p_2 + r_2^\delta \nabla \mathcal{P}_{h_2} + (\boldsymbol{x}_\delta - \boldsymbol{y}_{2_\delta})^T . \nabla \mathcal{P}_\tau = p_3 - r_3^\delta \nabla \mathcal{P}_{h_3} + (\boldsymbol{x}_\delta - \boldsymbol{y}_{3_\delta})^T . \nabla \mathcal{P}_\tau, \tag{4.32}$$

$$\lambda_2^\delta \nabla \mathcal{P}_{h_2} + \overrightarrow{\varrho_2} . \nabla \mathcal{P}_\tau = \lambda_3^\delta \nabla \mathcal{P}_{h_3} + \overrightarrow{\varrho_3} . \nabla \mathcal{P}_\tau \tag{4.33}$$

Now we follow the same procedure from equations (4.23) to (4.29) and use $\widetilde{\nabla} p_2$ and $p_2$ as suggested above to arrive on an alternate form of equation (4.30) which give rise to

$$p_3 = p_2 + \nabla p_2 \Big( \boldsymbol{x}_3 - \boldsymbol{x}_2 + \frac{r_3^\delta}{\lambda_3}(\mathbb{K}_2 - \mathbb{K}_3)\boldsymbol{n}_\delta \Big), \tag{4.34}$$

$$\begin{aligned} p_3 = p_1 + \nabla p_1 \Big( \boldsymbol{x}_2 - \boldsymbol{x}_1 + \frac{r_2}{\lambda_2}(\mathbb{K}_1 - \mathbb{K}_2) n_\sigma \Big) \\ + \nabla p_2 \Big( \boldsymbol{x}_3 - \boldsymbol{x}_1 + \frac{r_3^\delta}{\lambda_3}(\mathbb{K}_2 - \mathbb{K}_3)\boldsymbol{n}_\delta \Big) - \nabla p_2 \Big( \boldsymbol{x}_2 - \boldsymbol{x}_1 \Big), \end{aligned} \tag{4.35}$$

$$\begin{aligned} p_3 = p_1 + \nabla p_1 (\boldsymbol{x}_2 - \boldsymbol{x}_1) + \nabla p_1 \Big( \frac{r_2}{\lambda_2}(\mathbb{K}_1 - \mathbb{K}_2)\boldsymbol{n}_\sigma \Big) \\ + \nabla p_1 \big[\mathbb{I} + \frac{1}{\lambda_2}[\mathbb{K}_1 - \mathbb{K}_2]\boldsymbol{n}_\sigma \boldsymbol{n}_\sigma^T\big] \Big( \boldsymbol{x}_3 - \boldsymbol{x}_1 + \frac{r_3^\delta}{\lambda_3}(\mathbb{K}_2 - \mathbb{K}_3)\boldsymbol{n}_\delta \Big) \\ - \nabla p_1 \big[\mathbb{I} + \frac{1}{\lambda_2}[\mathbb{K}_1 - \mathbb{K}_2]\boldsymbol{n}_\sigma \boldsymbol{n}_\sigma^T\big] \Big( \boldsymbol{x}_2 - \boldsymbol{x}_1 \Big). \end{aligned} \tag{4.36}$$

In the above equation (4.36), we can substitute $\boldsymbol{n}_\sigma^T . (\boldsymbol{x}_2 - \boldsymbol{x}_1) = r_1 + r_2$ and

$$\begin{aligned} p_3 = p_1 - \nabla p_1 \Big( \frac{r_1}{\lambda_2}(\mathbb{K}_1 - \mathbb{K}_2)\boldsymbol{n}_\sigma \Big) \\ + \nabla p_1 \big[\mathbb{I} + \frac{1}{\lambda_2}[\mathbb{K}_1 - \mathbb{K}_2]\boldsymbol{n}_\sigma \boldsymbol{n}_\sigma^T\big] \Big( \boldsymbol{x}_3 - \boldsymbol{x}_1 + \frac{r_3^\delta}{\lambda_3}(\mathbb{K}_2 - \mathbb{K}_3)\boldsymbol{n}_\delta \Big). \end{aligned} \tag{4.37}$$

Here we can generalize the 2nd part of equation (4.37) in the same way as equation (4.30) for any arbitrary point in cell 3

$$p(\boldsymbol{x})_3 = p_1 - \nabla p_1\Big(\frac{r_1}{\lambda_2}(\mathbb{K}_1 - \mathbb{K}_2)\boldsymbol{n}_\sigma\Big) + \nabla p_1\Big[\mathbb{I} + \frac{1}{\lambda_2}[\mathbb{K}_1 - \mathbb{K}_2]\boldsymbol{n}_\sigma\boldsymbol{n}_\sigma^T\Big]\Big[\mathbb{I} + \frac{1}{\lambda_3}[\mathbb{K}_2 - \mathbb{K}_3]\boldsymbol{n}_\delta\boldsymbol{n}_\delta^T\Big](\boldsymbol{x} - \boldsymbol{x}_1) - \nabla p_1\Big[\mathbb{I} + \frac{1}{\lambda_2}[\mathbb{K}_1 - \mathbb{K}_2]\boldsymbol{n}_\sigma\boldsymbol{n}_\sigma^T\Big]\frac{r_1^\delta}{\lambda_3}(\mathbb{K}_2 - \mathbb{K}_3)\boldsymbol{n}_\delta, \quad (4.38)$$

where $r_1^\delta$ is the shortest distance from collocation point of cell 1 to interface $\delta$.

Once we obtain equation (4.38), we can generalize it in such a way that arbitrary number of interpolations across any direction can be represented. The pressure difference w.r.t the reference pressure in cell 1 will look like

$$p(\boldsymbol{x})_3 = p_1 - \nabla p_1\Big(\frac{r_1}{\lambda_2}(\mathbb{K}_1 - \mathbb{K}_2)\boldsymbol{n}_\sigma\Big) + \nabla p_1\Big[\mathbb{I} + \frac{1}{\lambda_2}[\mathbb{K}_1 - \mathbb{K}_2]\boldsymbol{n}_\sigma\boldsymbol{n}_\sigma^T\Big]\Big[\mathbb{I} + \frac{1}{\lambda_3}[\mathbb{K}_2 - \mathbb{K}_3]\boldsymbol{n}_\delta\boldsymbol{n}_\delta^T\Big](\boldsymbol{x} - \boldsymbol{x}_1) - \nabla p_1\Big[\mathbb{I} + \frac{1}{\lambda_2}[\mathbb{K}_1 - \mathbb{K}_2]\boldsymbol{n}_\sigma\boldsymbol{n}_\sigma^T\Big]\frac{r_1^\delta}{\lambda_3}(\mathbb{K}_2 - \mathbb{K}_3)\boldsymbol{n}_\delta. \quad (4.39)$$

Now we can express equation (4.39) in the homogenized form which will be multiplied by the pressure gradient in 1 $\nabla p_1$ as

$$\mathcal{H}_{1,3}^{\sigma,\delta}(\boldsymbol{x} - \boldsymbol{x}_1) = \Big[\mathbb{I} + \frac{1}{\lambda_2}[\mathbb{K}_1 - \mathbb{K}_2]\boldsymbol{n}_\sigma\boldsymbol{n}_\sigma^T\Big]\Big[\mathbb{I} + \frac{1}{\lambda_3}[\mathbb{K}_2 - \mathbb{K}_3]\boldsymbol{n}_\delta\boldsymbol{n}_\delta^T\Big](\boldsymbol{x} - \boldsymbol{x}_1) - \Big[\mathbb{I} + \frac{1}{\lambda_2}[\mathbb{K}_1 - \mathbb{K}_2]\boldsymbol{n}_\sigma\boldsymbol{n}_\sigma^T\Big]\frac{r_1^\delta}{\lambda_3}(\mathbb{K}_2 - \mathbb{K}_3)\boldsymbol{n}_\delta - \Big(\frac{r_1}{\lambda_2}(\mathbb{K}_1 - \mathbb{K}_2)\boldsymbol{n}_\sigma\Big). \quad (4.40)$$

We can then proceed to generalize equation (4.40) for any combination of interfaces as long as they are connected by cells. In this way, we move from one interface to another and determine coefficients which have to be multiplied to the pressure gradient in any specific cell. But as a prerequisite, we have to map cell-to-cell connections and cell-to-face connections which not only include the cells which share an interface but also other cells which are neighbours to the neighbours. So, for each cell-interface connection, we derive two sets $\Sigma = \sigma_1, \sigma_2, \sigma_3, ...\sigma_k$ and $\Theta = V_1, V_2, V_3, ...V_k, V_{k+1}$, where $\sigma_j = V_j \cap V_{j+1}$. This allows us to represent the homogenization function for $k + 1^{th}$ interface as:

$$\mathcal{H}_{1,k+1}^{\Sigma}(\boldsymbol{x} - \boldsymbol{x}_1) = \prod_{i=1}^{k}\Big[\mathbb{I} + \frac{1}{\lambda_{i+1}^i}[\mathbb{K}_i - \mathbb{K}_{i+1}]\boldsymbol{n}_i\boldsymbol{n}_i^T\Big](\boldsymbol{x} - \boldsymbol{x}_1) - \sum_{j=1}^{k}\Big(\prod_{i=1}^{j-1}\Big[\mathbb{I} + \frac{1}{\lambda_{i+1}^i}[\mathbb{K}_i - \mathbb{K}_{i+1}]\boldsymbol{n}_i\boldsymbol{n}_i^T\Big]\Big)\frac{r_1^j}{\lambda_{j+1}^j}(\mathbb{K}_j - \mathbb{K}_{j+1})\boldsymbol{n}_j, \quad (4.41)$$

where $\lambda^i_{i+1} = \boldsymbol{n}_i^T \mathbb{K}_{i+1} \boldsymbol{n}_i$ and $r_1^j$ is the shortest distance between collocation point in $V_1$ i.e. $\boldsymbol{x}_1$ and interface $j$. Also we can see that for a medium with homogeneous permeability tensor, the homogenization function reduces to

$$\mathcal{H}^{\Sigma}_{1,k+1}(\boldsymbol{x} - \boldsymbol{x}_1) = \boldsymbol{x} - \boldsymbol{x}_1. \tag{4.42}$$

After we obtain the homogenization function, we can define our auxiliary conditions which will help us to reconstruct the pressure gradient. Auxiliary conditions in our system help us to determine the pressure gradient which is basically composed of 3 unknowns for 3D problem and 2 unknowns for 2D. In this work we focus on 3D problem so we have 3 unknowns we have to solve for. As discussed previously in equation (5.14), (4.21) and (4.22), we can form similar conditions but with a homogenization function added to it which will allow us to choose 3 cells for gradient reconstruction from any number of interfaces (considering neighbours to neighbours and so on), such that a positive basis for $\mathbb{K}_1 \mathbf{n}_\sigma \cdot Q$ can be obtained.

For an interior interface or cell, the auxiliary condition can be represented as following:

$$\mathcal{H}^{\Sigma}_{1,i}(\boldsymbol{x}_i - \boldsymbol{x}_1) \nabla p_1 = p_i - p_1, \tag{4.43}$$

where $\boldsymbol{x}_i$ is the collocation point in control volume $V_i$. Although homogenization allows us to interpolate pressure at arbitrary points in the control volume, we will use collocation point information as that is readily available and we do not need to any additional unknowns or calculation to determine the location of these points. We can also form auxiliary condition for boundary interfaces, but we do it in terms of Dirichlet or Neumann

$$\begin{aligned} \mathcal{H}^{\Sigma}_{1,i}(\boldsymbol{x}_D - \boldsymbol{x}_1) \nabla p_1 = g_D - p_1, \quad \textbf{on} \quad \sigma \in V_i \cap \partial \boldsymbol{\Omega}_{\mathbf{D}}, \\ \nabla \mathcal{H}^{\Sigma}_{1,i} \mathbb{K}_i \boldsymbol{n}_\sigma \nabla p_1 = g_N, \quad \textbf{on} \quad \sigma \in V_i \cap \partial \boldsymbol{\Omega}_{\mathbf{N}}. \end{aligned} \tag{4.44}$$

Here the homogenization function gradient is represented as:

$$\nabla \mathcal{H}^{\Sigma}_{1,k+1}(\boldsymbol{x} - \boldsymbol{x}_1) = \prod_{i=1}^{k} \Big[ \mathbb{I} + \frac{1}{\lambda^i_{i+1}} [\mathbb{K}_i - \mathbb{K}_{i+1}] \boldsymbol{n}_i \boldsymbol{n}_i^T \Big], \tag{4.45}$$

We use equations (4.43) and (4.44) in the above formulation (4.22), by determining the $Q$ matrix which can satisfy the positive basis condition.

## 4.5 Stencil Selection

First, we loop through the neighboring interfaces and check through all combinations of vectors in the $Q$ matrix, which will yield all non-negative coefficients in our combination. But when we do not have such a basis, we proceed to homogenization where we determine all the interfaces and cells present in the purview of the cell of interest. We use this information

and calculate the auxiliary vectors corresponding to all such interfaces belonging to our homogenization set. Then we loop through the new group of cells (maybe 20 or 40) depending on our choice and pick three vectors in that space that produce non-negative coefficients, and the sum of the coefficients is minimal. Weights can also be assigned w.r.t the level of homogenization, and an additional condition that minimizes weights can be handy as suggested in (Danilov and Vassilevski, 2009).

One has to be careful here as there might be a chance of the selected stencil not having the two-point counterpart of the cell-interface pair i.e. cell-2 if we are homogenizing cell 1. So we have to make sure all our combinations have cell-2 such that we include this vector in our final stencil and our non-linear weighting becomes robust. As shown in Figure 4.3, the

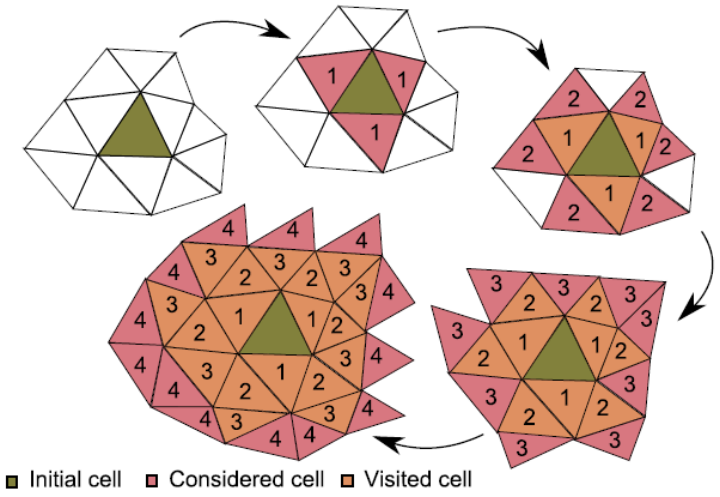



Figure 4.3: Stencil selection algorithm, where cells are considered in layers to construct non-negative basis for one sided flux approximation and gradient reconstruction (Terekhov et al., 2016)

layers are defined on how many neighbour to neighbours we are considering. In our case we consider 4 layers, but it is also possible to consider more.

## 4.6 Nonlinear Convex Combination and Weights

### Internal faces

By combining our knowledge from equations (4.2), (4.6) and (4.22), we can formulate both semi fluxes from each of the two points connections for internal interface between two cells $V_1 \cap V_2$. For convenience, we can take 3 interfaces for cells 1 and 2 such that $f_{11}, f_{12}, f_{13} \in V_1$ and $f_{21}, f_{22}, f_{23} \in V_2$ where $f_{ij}$ is the $j^{th}$ face in control volume $i$. Also to have a generalized derivation, we can take $f_{11}, f_{12}, f_{21}, f_{22} \in \mathcal{F}_I$ and $f_{13}, f_{23} \in \partial\mathbf{\Omega}$, where $\mathcal{F}_I$ is the set containing internal interfaces.

Discrete semi-fluxes can be derived as:

$$\begin{aligned} q_1^\sigma &= c_{11}(p_1 - p_2) + c_{12}(p_1 - p_3) + c_{13}(p_1 - g_1), \\ q_2^\sigma &= c_{21}(p_1 - p_2) + c_{22}(p_4 - p_2) + c_{23}(g_2 - p_2), \end{aligned} \tag{4.46}$$

where $c_{ij}$ is similar to $f_{ij}$, and we substitute the value of harmonic pressures $p_{hi}$ in the form of cell center pressures $p_i$ by using the transformation mentioned in equations (4.15), (4.17), (4.16), (4.18). As $m_H$ and $l_H$ are positive, multiplying with non-negative coefficients $c_1, c_2, c_3$ mentioned in equation (4.22) will yield non-negative coefficients $c_{ij}$ for both the semi-fluxes. It is also better for us to consider ghost cells in case of Neumann boundary condition. So, if we assume $g_1 \in \partial\mathbf{\Omega_N}$ and $g_2 \in \partial\mathbf{\Omega_D}$, we can rearrange our equations in (4.46) as follows

$$\begin{aligned} q_1^\sigma &= c_{11}(p_1 - p_2) + c_{12}(p_1 - p_3) + c_{13}(p_1 - p_{G_1}), \\ q_2^\sigma &= c_{21}(p_1 - p_2) + c_{22}(p_4 - p_2) + c_{23}(g_D - p_2). \end{aligned} \tag{4.47}$$

Here $p_{G_1}$ (ghost cell pressure) will be equal to $p_1$ and $g_D$ is the pressure at the Dirichlet boundary.

Now we will use the convex combination to combine these discrete semi-fluxes into total flux which can have unequal contribution from both these fluxes. The convex combination looks like:

$$q^\sigma = \mu_1 q_1^\sigma + \mu_2 q_2^\sigma, \tag{4.48}$$

where the two weights or parameters $\mu_1$ and $\mu_2$ must satisfy the condition $\mu_1 + \mu_2 = 1$. Combining equations (4.47) and (4.48) and taking straight forward choice for weights as $\mu_1 = \mu_2 = 0.5$, we will get an MPFA approximation which is different than MPFA-O or MPFA-L discussed in (Aavatsmark, 2002) and other literatures. We will call this as Average MPFA method as described in (Terekhov et al., 2016; Vassilevski et al., 2020)

$$\begin{aligned} q_{Avg}^\sigma = 0.5\Big[c_{11}(p_1 - p_2) + c_{12}(p_1 - p_3) + c_{13}(p_1 - p_{G_1})\Big] \\ + 0.5\Big[c_{21}(p_1 - p_2) + c_{22}(p_4 - p_2) + c_{23}(g_D - p_2)\Big]. \end{aligned} \tag{4.49}$$

But as our purpose is to find a monotone and non-negative solution, the final approximation for flux should be expressed in the following form

$$q^\sigma = \mathbb{T}_1 p_1 - \mathbb{T}_2 p_2. \tag{4.50}$$

In equation (4.50), $\mathbb{T}_2$ and $\mathbb{T}_1$ are non-negative and function of $p_3, p_4, p_{G_1}, g_D$, hence they will change according to how the pressure in the reservoir domain changes as $p_3, p_4, p_{G_1}$ are unknowns in our formulation and we have to provide the values from previous iteration which will be discussed later in numerical scheme.

Now we combine equations (4.47) and (4.48) in a different manner such that weights are variable to obtain the equation represented in (4.50)

$$\begin{aligned} q^{\sigma} = \Big(\mu_1\big[c_{11}+c_{12}+c_{13}\big]+\mu_2\big[c_{21}\big]\Big)p_1 \\ -\Big(\mu_2\big[c_{21}+c_{22}+c_{23}\big]+\mu_1\big[c_{11}\big]\Big)p_2 \\ -\mu_1\big[c_{12}p_3+c_{13}p_{G_1}\big]+\mu_2\big[c_{22}p_4+c_{23}g_D\big]. \end{aligned} \tag{4.51}$$

Hence in equation (4.51), $\mu_1\big[c_{12}p_3+c_{13}p_{G_1}\big]-\mu_2\big[c_{22}p_4+c_{23}g_D\big]$ should be equal to zero such that the final form can be reduced to equation (4.48).

By this preposition we have two conditions for weights

$$\begin{aligned} \mu_1+\mu_2 &= 1, \\ \mu_1\big[c_{12}p_3+c_{13}p_{G_1}\big]-\mu_2\big[c_{22}p_4+c_{23}g_D\big] &= 0. \end{aligned} \tag{4.52}$$

These two system of equations in (4.52) have two unknowns and we can obtain a unique solution for both the weights as follows

$$\mu_1 = \frac{\mathbb{U}_2+\varepsilon}{(\mathbb{U}_1+\mathbb{U}_2+2\varepsilon)}, \qquad \mu_2 = \frac{\mathbb{U}_1+\varepsilon}{(\mathbb{U}_1+\mathbb{U}_2+2\varepsilon)}, \tag{4.53}$$

where $\mathbb{U}_i$ is:

$$\mathbb{U}_1 = c_{12}p_3+c_{13}p_{G_1}, \qquad \mathbb{U}_2 = c_{22}p_4+c_{23}g_D, \tag{4.54}$$

and $\varepsilon$ is a small regularization parameter such that weights have finite value when both $\mathbb{U}_1$ and $\mathbb{U}_2$ become zero. So, the final form of flux can be represented as:

$$q^{\sigma} = \Big(\mu_1\big[c_{11}+c_{12}+c_{13}\big]+\mu_2\big[c_{21}\big]\Big)p_1 - \Big(\mu_2\big[c_{21}+c_{22}+c_{23}\big]+\mu_1\big[c_{11}\big]\Big)p_2. \tag{4.55}$$

### Boundary faces

Unlike the internal faces, at the boundaries we do not have $p_2$ i.e. there is only 1 cell connected with the boundary. So, for the primary cell, i.e. cell 1, we can proceed to define faces and coefficients similar to internal face as $f_{11}, f_{12}, f_{13} \in V_1$ but for the faces in other semi-flux corresponding to boundary $f_{b1}, f_{b2}, f_{b3} \in V_1$. This implies that we have to reconstruct the gradient at face center of the boundary interface. Also the prerequisites for derivation will be slightly different for internal faces as $f_{21}, f_{22} \in \mathcal{F}_I$ and $f_{11}, f_{12}, f_{13}, f_{23} \in \partial\mathbf{\Omega}$, where $f_{11} = f_{12}$.

The Discrete semi-fluxes for boundaries can be derived as:

$$\begin{aligned} q_1^{\sigma} &= c_{11}(p_1-p_b)+c_{12}(p_1-p_3)+c_{13}(p_1-g_1), \\ q_b^{\sigma} &= c_{b1}(p_1-p_b)+c_{b2}(p_4-p_b)+c_{b3}(g_2-p_b), \end{aligned} \tag{4.56}$$

where $c_{bj}$ is similar to $f_{bj}$, and we substitute the value of harmonic pressures $p_{hi}$ in the form of cell center pressures $p_i$ by using the transformation mentioned in equations (4.15), (4.17), (4.16), (4.18). Equation (4.21) is used to determine the value of $c_{bj}$. So, if we assume $p_b$ to be a general boundary i.e. either Dirichlet or Neumann, $g_1 \in \partial\mathbf{\Omega_N}$ and $g_2 \in \partial\mathbf{\Omega_D}$, we can rearrange our equations in (4.46) as follows

$$\begin{aligned} q_1^\sigma &= c_{11}(p_1 - p_b) + c_{12}(p_1 - p_3) + c_{13}(p_1 - p_{G_1}), \\ q_b^\sigma &= c_{b1}(p_1 - p_b) + c_{b2}(p_4 - p_b) + c_{b3}(g_D - p_b). \end{aligned} \tag{4.57}$$

Now using the same concept suggested in equations (4.48) and (4.52), we can have total flux represented in the form of

$$\begin{aligned} q^\sigma = \Big(\mu_1\big[c_{11} + c_{12} + c_{13}\big] + \mu_2\big[c_{b1}\big]\Big)p_1 \\ - \Big(\mu_2\big[c_{b1} + c_{b2} + c_{b3}\big] + \mu_1\big[c_{11}\big]\Big)p_b \\ - \mu_1\big[c_{12}p_3 + c_{13}p_{G_1}\big] + \mu_2\big[c_{b2}p_4 + c_{b3}g_D\big], \end{aligned} \tag{4.58}$$

and substituting the weights such that,

$$\mu_1\big[c_{12}p_3 + c_{13}p_{G_1}\big] - \mu_2\big[c_{b2}p_4 + c_{b3}g_D\big] = 0. \tag{4.59}$$

The final equation for flux through the boundary interface can be represented as

$$\begin{aligned} q^\sigma &= \mathbb{T}_1 p_1 - \mathbb{T}_b p_b, \\ q^\sigma &= \Big(\mu_1\big[c_{11} + c_{12} + c_{13}\big] + \mu_2\big[c_{b1}\big]\Big)p_1 - \Big(\mu_2\big[c_{b1} + c_{b2} + c_{b3}\big] + \mu_1\big[c_{11}\big]\Big)p_b. \end{aligned} \tag{4.60}$$

But in this equation (4.60), $p_b$ is an unknown and we can eliminate it by using

$$\begin{aligned} \alpha p_b + \beta(\mathbb{T}_b p_b - \mathbb{T}_1 p_1) &= \gamma, \\ p_b &= \frac{\beta\mathbb{T}_1 p_1 + \gamma}{\beta\mathbb{T}_b + \alpha}. \end{aligned} \tag{4.61}$$

Now after combining equations (4.60) and (4.61) we can define our new flux expression on boundary as

$$q^\sigma = \frac{\alpha\mathbb{T}_1 p_1 - \mathbb{T}_b\gamma}{\beta\mathbb{T}_b + \alpha}. \tag{4.62}$$

The $A$ matrix described in previous chapter can be assembled by transmissibility values obtained from equations (4.55) and (4.62). This will make sure that diagonal is always positive i.e. transmissibility of $p_1$ and the off-diagonal terms are always negative that have transmissibility values corresponding to $p_2$. Because of the above formulation, we can ensure the obtained $A$ matrix is M-matrix which makes our solution monotone. Also, note that this

is specific to each Picard's iteration level. But still, we may violate the global maximum or minimum.

Another question arises when we are applying this procedure for the multiphase flow problem, and we need to derive Jacobian terms. In that case, the $A$ matrix will have contributions from other cells in the stencil and the entries in off-diagonal parts may have non-negative entries. This phenomenon was not fully explored in this project as we did not have any extreme cases in our hands to test it. We have also not faced this problem as mentioned in the numerical test section.

## 4.7 Fracture Discretization

Another feature implemented in this simulation framework was the treatment of fractures. A Discrete Fracture Model (DFM) framework is used as suggested in (Kar, 2003). One of the main features of the DFM framework is that we construct a fracture network first which is then followed by generating grid and matrix elements around the fractures. After this, we identify two sub-domains in our internal domain which are matrix elements (3D objects) and fractures (2D planes). Fractures are considered as lower-dimensional features where flow is present and primary unknowns are collocated, but we only view the solution related to matrix elements in our plots.

There are three different kinds of connections identified such as matrix-matrix, matrix-fracture, and fracture-fracture. The flux across matrix-matrix connections is evaluated using the procedure suggested above for nonlinear two-point flux, but we consider the flux across fracture-fracture and fracture-matrix is defined by linear two-point flux approximation. This was mainly due to the fact that permeability is regarded as a scalar value rather than a tensor in the fracture. Also, using permeability tensor in gradient reconstruction when the width of the element (i.e. fracture) is negligible can cause problems while calculating the harmonic averaging point defined in equation (4.9).

To derive transmissibilities for fractured system, we introduce

$$T_{mi} = \frac{r_i.(\mathbb{K}_i.\boldsymbol{n}_\sigma)A_i}{|r_i|^2}, \quad T_{fi} = \frac{k_{fi}A_{fi}}{|r_{fi}|}, \tag{4.63}$$

where $T_{mi}$ is the half transmissibility of matrix cell, $T_{fi}$ is the half transmissibility corresponding to fracture element. Here $k_{fi}$ is the scalar value of permeability usually defined along the length of the fracture, $A_{fi}$ is the area of fracture along its length and $r_{fi}$ is half of the aperture of the fracture. Using these terms, we can derive matrix-fracture and fracture-fracture transmissibilities as

$$T_{mf} = \frac{T_{mi}T_{fi}}{T_{mi} + T_{fi}}, \quad T_{ff} = \frac{T_{fi}T_{fj}}{T_{fi} + T_{fj}}. \tag{4.64}$$

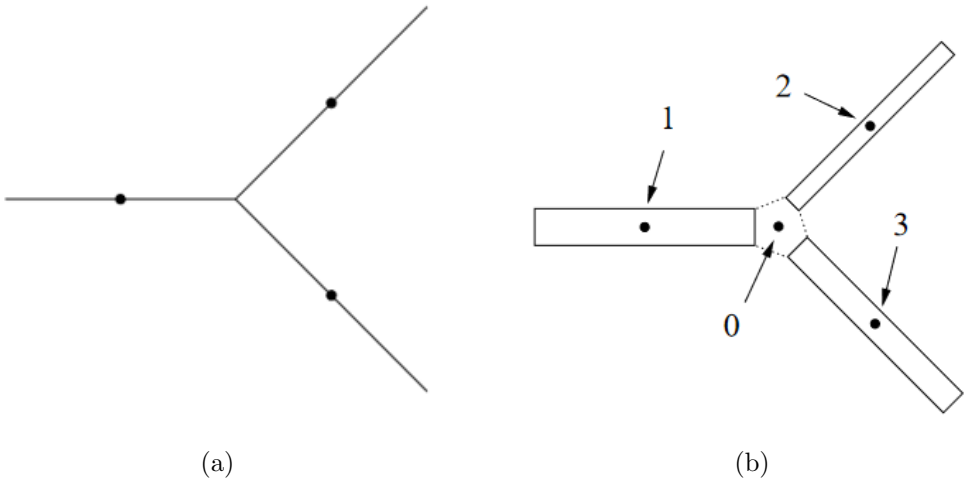



Figure 4.4: Fracture representation in (a) Grid domain (b) Computational domain

Here $T_{ff}$ considers transmissibility across two fracture elements. But we may have a case for intersecting fractures where more than 2 fractures will intersect on an edge (1D feature) in our grid. In than case, we can resolve the transmissibility across two such fractures within an intersection by using star delta combination as suggested in (Kar, 2003)

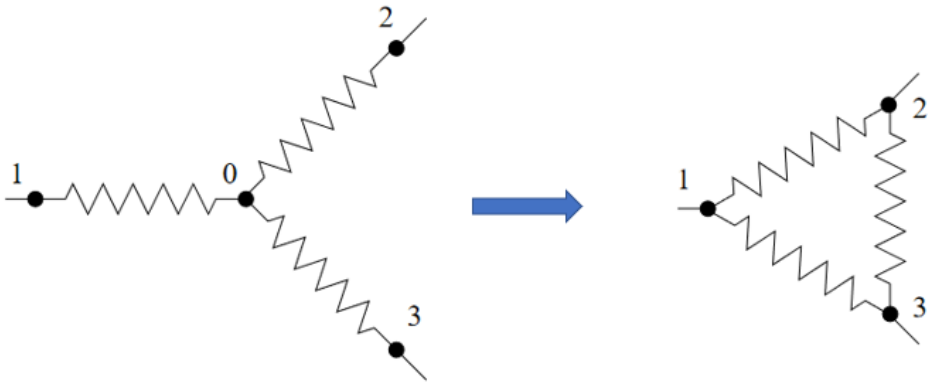



Figure 4.5: Transforming a intersection of three fractures to delta form to evaluate the transmissibilities (Kar, 2003)

$$T_{12} = \frac{T_{f1}T_{f2}}{T_{f1}+T_{f2}+T_{f3}}, \quad T_{23} = \frac{T_{f2}T_{f3}}{T_{f1}+T_{f2}+T_{f3}}, \quad T_{31} = \frac{T_{f1}T_{f3}}{T_{f1}+T_{f2}+T_{f3}}. \tag{4.65}$$

By using equation (4.65), we can calculate the effective transmissibilities in all three individual connections. This can be extended to 4 or more intersecting fractures in our framework. These transmissibilities i.e. both fracture-fracture and matrix-fracture are evaluated only once during the pre-processing stage, but the non-linear matrix-matrix transmissibilities will be evaluated in the beginning of every nonlinear iteration.

# Chapter 5

# Discretization of Momentum Equation

In this section, we provide discretization for the momentum equation. First we consider the equation for pure mechanics as discussed earlier in Chapter 2. The discretization framework will follow a similar path to NTPFA, but few problems encountered in the gradient reconstruction process will be discussed later. Some modification are made in order to achieve robust gradient reconstruction for homogeneous medium. A heterogeneous case was not addressed in the nonlinear framework as we focused mainly on oscillations in traction along non-existent fracture, and would be addressed in future works.

## 5.1 Characterizing Traction at Interface

Following the works of (Terekhov and Tchelepi, 2020), which focused on development of multi-point stress approximation in a FV framework for linear elastic problem, we proposed a new form of discretization. This form will use the FV framework as a base but will be adapted to a nonlinear approximation as discussed in Chapter 4. The equations have a following form

$$\begin{aligned} -\nabla \cdot \sigma &= \mathbf{f} \quad \textbf{in} \quad \boldsymbol{\Omega}, \\ \textbf{B.C.} \quad &\textbf{in} \quad \partial\boldsymbol{\Omega}, \end{aligned} \tag{5.1}$$

where we use Dirichlet, Neumann and Roller boundary condition which are discussed later. Here $\mathbf{f}$ is analogous to source/sink term which corresponds to external force $3 \times 1$ vector, $\Omega$ is our cell domain, $\partial\Omega$ corresponds to boundary interfaces and $\sigma$ is the stress $3 \times 3$ tensor.

The stress tensor satisfies the stress-strain relationship as follows:

$$\left[\frac{\nabla \mathbf{u} + \nabla \mathbf{u}^T}{2}\right] = \mathbb{S} : \sigma, \tag{5.2}$$

where $u$ is the displacement vector $[u, v, w]^T$ and $\mathbb{S}$ is the material compliance tensor of rank 4. In our case, we focus on isotropic material but the same formulation can be applied to

anisotropic medium. Note that we do not focus on rigid-tropic material where the $E$ matrix can be singular i.e. null eigenvalues in compliance tensor.

Using the formulation suggested in (Terekhov and Tchelepi, 2020), we denote the compliance tensor $\mathbb{S}$ in the form of Voigt notation $6 \times 6$ symmetric matrix

$$C = \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} & c_{15} & c_{16} \\ c_{12} & c_{22} & c_{23} & c_{24} & c_{25} & c_{26} \\ c_{13} & c_{23} & c_{33} & c_{34} & c_{35} & c_{36} \\ c_{14} & c_{24} & c_{34} & c_{44} & c_{45} & c_{46} \\ c_{15} & c_{25} & c_{35} & c_{45} & c_{55} & c_{56} \\ c_{16} & c_{26} & c_{36} & c_{46} & c_{56} & c_{66} \end{bmatrix}. \tag{5.3}$$

With the stress and strain tensor expressed as $6 \times 1$ vectors due to symmetry

$$\begin{aligned} \varepsilon &= \left[u_x, v_y, w_z, v_z + w_y, u_z + w_x, v_x + u_y\right]^T, \\ \xi &= \left[\sigma_{xx}, \sigma_{yy}, \sigma_{zz}, \sigma_{yz}, \sigma_{zx}, \sigma_{xy}\right]^T, \end{aligned} \tag{5.4}$$

where the individual displacement component gradients are defined as

$$\nabla u = [u_x, u_y, u_z]^T, \quad \nabla v = [v_x, v_y, v_z]^T, \quad \nabla w = [w_x, w_y, w_z]^T. \tag{5.5}$$

From (5.3) and (5.4) we can represent the stress strain relationship as $\varepsilon = C\xi$ or $\xi = E\varepsilon$ where $E$ is inverse of $C$.

When we look into finite volume framework, we take equations (5.1) and (5.2) and apply Stokes theorem to the divergence operator in the left-hand side to get the following formulation

$$\int_V \nabla \cdot \mathbb{C} : \frac{\nabla \mathbf{u} + \nabla \mathbf{u}^T}{2} \mathbf{d}V = \sum_{\delta \in F(V)} \int_\delta \mathbb{C} : \frac{\nabla \mathbf{u} + \nabla \mathbf{u}^T}{2} \mathbf{n} \mathbf{d}S = \sum_{\delta \in F(V)} |\delta| \left[ \mathbb{C} : \frac{\nabla \mathbf{u} + \nabla \mathbf{u}^T}{2} \right] \mathbf{n}_\delta, \tag{5.6}$$

where $\delta$ is the interface belonging to control volume $F(V)$, $|\delta|$ is the area of interface. Using (5.6) we can now proceed to calculating the traction along an interface.

## Traction at the interface

In the case of elastic medium we have the following expression for the traction $\mathbf{F}$ at the interface of the unit normal $\mathbf{n}$

$$\mathbf{F} = \left[ \mathbb{C} : \frac{\nabla \mathbf{u} + \nabla \mathbf{u}^T}{2} \right] \cdot \mathbf{n}, \tag{5.7}$$

where $\mathbf{u}$ represents vector of displacements, $\mathbb{C}$ is $4^{th}$ rank tensor of stiffness coefficients. Using Voigt notation, (5.7) can be rewritten through matrix-vector multiplications where $E$ will represent $6 \times 6$ symmetric matrix. Then (5.7) can be rewritten as follows (Terekhov and Tchelepi, 2020)

$$\mathbf{F} = \begin{bmatrix} \boldsymbol{n}^T\mathbb{A}_1 & \boldsymbol{n}^T\mathbb{A}_6^T & \boldsymbol{n}^T\mathbb{A}_5^T \\ \boldsymbol{n}^T\mathbb{A}_6 & \boldsymbol{n}^T\mathbb{A}_2 & \boldsymbol{n}^T\mathbb{A}_4^T \\ \boldsymbol{n}^T\mathbb{A}_5 & \boldsymbol{n}^T\mathbb{A}_4 & \boldsymbol{n}^T\mathbb{A}_3 \end{bmatrix} \begin{bmatrix} \nabla u \\ \nabla v \\ \nabla w \end{bmatrix}, \tag{5.8}$$

where $\mathbb{A}_i$ is a $3 \times 3$ matrix derived from material stiffness tensor $E$. This representation allows us to represent traction in matrix vector multiplication format. Also matrices $\mathbb{A}_i$ belong to cell 1 and $\mathbb{B}_i$ belong to cell 2 (i.e. derived from material tensor of cell 2).

Now let us consider balance of traction between 1st and 2nd cells. We assume balance of traction and displacement continuity at the interface as considered in (Terekhov and Tchelepi, 2020)

$$\mathbf{F} = \begin{bmatrix} \boldsymbol{n}^T\mathbb{A}_1 \cdot \nabla u_1 + \boldsymbol{n}^T\mathbb{A}_6^T \cdot \nabla v_1 + \boldsymbol{n}^T\mathbb{A}_5^T \cdot \nabla w_1 \\ \boldsymbol{n}^T\mathbb{A}_6 \cdot \nabla u_1 + \boldsymbol{n}^T\mathbb{A}_2 \cdot \nabla v_1 + \boldsymbol{n}^T\mathbb{A}_4^T \cdot \nabla w_1 \\ \boldsymbol{n}^T\mathbb{A}_5 \cdot \nabla u_1 + \boldsymbol{n}^T\mathbb{A}_4 \cdot \nabla v_1 + \boldsymbol{n}^T\mathbb{A}_3 \cdot \nabla w_1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{n}^T\mathbb{B}_1 \cdot \nabla u_1 + \boldsymbol{n}^T\mathbb{B}_6^T \cdot \nabla v_1 + \boldsymbol{n}^T\mathbb{B}_5^T \cdot \nabla w_1 \\ \boldsymbol{n}^T\mathbb{B}_6 \cdot \nabla u_1 + \boldsymbol{n}^T\mathbb{B}_2 \cdot \nabla v_1 + \boldsymbol{n}^T\mathbb{B}_4^T \cdot \nabla w_1 \\ \boldsymbol{n}^T\mathbb{B}_5 \cdot \nabla u_1 + \boldsymbol{n}^T\mathbb{B}_4 \cdot \nabla v_1 + \boldsymbol{n}^T\mathbb{B}_3 \cdot \nabla w_1 \end{bmatrix}, \tag{5.9}$$

$$\mathbf{u}_1 + G_1(\boldsymbol{x} - \boldsymbol{x}_1) = \mathbf{u}_2 + G_2(\boldsymbol{x} - \boldsymbol{x}_2), \tag{5.10}$$

where,

$$\mathbf{u}_i = \begin{bmatrix} u_i, v_i, w_i \end{bmatrix}, \quad G_i = \begin{bmatrix} \frac{\partial u_i}{\partial x} & \frac{\partial u_i}{\partial y} & \frac{\partial u_i}{\partial z} \\ \frac{\partial v_i}{\partial x} & \frac{\partial v_i}{\partial y} & \frac{\partial v_i}{\partial z} \\ \frac{\partial w_i}{\partial x} & \frac{\partial w_i}{\partial y} & \frac{\partial w_i}{\partial z} \end{bmatrix}. \tag{5.11}$$

Here $\mathbf{u}_i$, $G_i$ are the displacement and displacement gradient matrix in cell $i$.

Then the final equation of traction will be represented by equations (5.12) as recommended in (Terekhov and Tchelepi, 2020) where it is decomposed into harmonic and transversal parts

$$\mathbf{F} = T(\mathbf{u}_2 - \mathbf{u}_1) - \Gamma G_\tau, \quad T = T_1(r_1T_2 + r_2T_1)^{-1}T_2, \tag{5.12}$$

$$\Gamma = r_2T_1(r_1T_2 + r_2T_1)^{-1}\Gamma_2 + r_1T_2(r_1T_2 + r_2T_1)^{-1}\Gamma_1 + T[\mathbb{I} \otimes (\mathbf{y_1} - \mathbf{y_2})^T], \tag{5.13}$$

where first term in (5.12) called harmonic $\mathbf{F}_h = T(\mathbf{u}_2 - \mathbf{u}_1)$ and the second one is called transversal $\mathbf{F}_\delta = \Gamma G_\tau$. $G_\tau$ is the transversal gradient of displacement, $T_1$ and $T_2$ are $3 \times 3$ matrices derived from normal component of vectors $\boldsymbol{n}^T\mathbb{A}_i$ and $\boldsymbol{n}^T\mathbb{B}_i$ respectively from equations (5.8) and (5.9), $r_i$ is the perpendicular distance from collocation point of cell $i$ to the two point interface, $T$ is the tensorial transmissibility which is equal to $T = T_1(r_1T_2 + r_2T_1)T_2$.

## 5.2 Gradient Reconstruction

As we showed that harmonic part of traction vector can be approximated by two point stencil i.e. $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$, we proceed to apply gradient reconstruction to the transversal part of traction to estimate $G_\tau$. One can conclude to the following representation of transversal traction $\mathbf{F}_{\delta i} = \Gamma G_{\tau i}$ approximated from both sides of the interface

$$\Gamma \begin{bmatrix} \nabla u_1 \\ \nabla v_1 \\ \nabla w_1 \end{bmatrix} = \Gamma(Q)^{-1} \begin{bmatrix} \mathbf{u}_2 - \mathbf{u}_1 \\ \mathbf{u}_3 - \mathbf{u}_1 \\ r_4 - \alpha \mathbf{u}_1 \end{bmatrix}, \tag{5.14}$$

where the matrix $Q$ is formed by auxiliary conditions for gradient reconstruction which are taken from (Terekhov and Tchelepi, 2020)

$$\left( [\mathbb{I} \otimes (\boldsymbol{x}_2 - \boldsymbol{x}_1)^T] + r_2 T_2^{-1} \begin{bmatrix} \boldsymbol{n}^T(\mathbb{A}_1 - \mathbb{B}_1) & \boldsymbol{n}^T(\mathbb{A}_6 - \mathbb{B}_6)^T & \boldsymbol{n}^T(\mathbb{A}_5 - \mathbb{B}_5)^T \\ \boldsymbol{n}^T(\mathbb{A}_6 - \mathbb{B}_6) & \boldsymbol{n}^T(\mathbb{A}_2 - \mathbb{B}_2) & \boldsymbol{n}^T(\mathbb{A}_4 - \mathbb{B}_4)^T \\ \boldsymbol{n}^T(\mathbb{A}_5 - \mathbb{B}_5) & \boldsymbol{n}^T(\mathbb{A}_4 - \mathbb{B}_4) & \boldsymbol{n}^T(\mathbb{A}_3 - \mathbb{B}_3) \end{bmatrix} \right) \begin{bmatrix} \nabla u_1 \\ \nabla v_1 \\ \nabla w_1 \end{bmatrix} = u_2 - u_1, \tag{5.15}$$

$$\left( \alpha [\mathbb{I} \otimes (\boldsymbol{x}_b - \boldsymbol{x}_1)^T] + \beta P \begin{bmatrix} \boldsymbol{n}^T\mathbb{A}_1 & \boldsymbol{n}^T\mathbb{A}_6^T & \boldsymbol{n}^T\mathbb{A}_5^T \\ \boldsymbol{n}^T\mathbb{A}_6 & \boldsymbol{n}^T\mathbb{A}_2 & \boldsymbol{n}^T\mathbb{A}_4^T \\ \boldsymbol{n}^T\mathbb{A}_5 & \boldsymbol{n}^T\mathbb{A}_4 & \boldsymbol{n}^T\mathbb{A}_3 \end{bmatrix} \right) \begin{bmatrix} \nabla u_1 \\ \nabla v_1 \\ \nabla w_1 \end{bmatrix} = r - \alpha u_1, \tag{5.16}$$

$$\left( \boldsymbol{n}\boldsymbol{n}^T [\mathbb{I} \otimes (\boldsymbol{x}_b - \boldsymbol{x}_1)^T] + P \begin{bmatrix} \boldsymbol{n}^T\mathbb{A}_1 & \boldsymbol{n}^T\mathbb{A}_6^T & \boldsymbol{n}^T\mathbb{A}_5^T \\ \boldsymbol{n}^T\mathbb{A}_6 & \boldsymbol{n}^T\mathbb{A}_2 & \boldsymbol{n}^T\mathbb{A}_4^T \\ \boldsymbol{n}^T\mathbb{A}_5 & \boldsymbol{n}^T\mathbb{A}_4 & \boldsymbol{n}^T\mathbb{A}_3 \end{bmatrix} \right) \begin{bmatrix} \nabla u_1 \\ \nabla v_1 \\ \nabla w_1 \end{bmatrix} = \boldsymbol{n}\boldsymbol{n}^T \boldsymbol{u}_1, \tag{5.17}$$

where (5.15) is used for internal cells, (5.16) for Dirichlet and Neumann boundaries and (5.17) for roller boundary. Here each equation is essentially a 3x9 matrix being multiplied by 9x1 vector of gradients. Here important to note that the process of estimating $Q^{-1}$ is performed during the pre-processor stage which means we only have to compute the coefficients and stencil once per discretization.

When doing nonlinear discretization, gradient reconstruction is crucial part as we have to make sure the coefficients are non-negative and their sum is minimal. So, for that purpose, we tried three different methods for the stencil. In the end we should make sure that all the entries of $\Gamma(Q)^{-1} \quad 3 \times 9$ matrix are positive for the nonlinear framework to succeed.

### Direct reconstruction

In this method, we reconstruct the gradient by choosing 3 faces and inverting the $9 \times 9$ matrix we obtain and multiply by $\Gamma$. We only consider the faces and respective cells corresponding

to the cell where we reconstruct the gradients i.e. for wedge and hexahedrons we will have 5 and 6 neighbouring cells respectively. In this way we reconstruct all the components of displacement gradient $[\nabla u, \nabla v, \nabla w]^T$.

The downside of this reconstruction is that it is too restrictive in terms of all the gradients are determined by choosing 3 neighbours out of 5 (wedge) or 6 (hexahedron). Because of this restriction, we might not be able to find all 27 positive entries in the coefficient matrix. This is similar to restriction we face during gradient reconstruction for flow problem with harmonic averaging points when we have high anisotropy or heterogeneity in permeability tensor.

## Homogenization function

In the previous reconstruction we were limited by choosing 3 cells out of 5 or 6. But by using a homogenization function we can increase the number of choices we have by approximating the gradients across neighbours to neighbours and so on. Homogenization function was used in (Terekhov et al., 2016) for flow equation where pressure gradient was reconstructed using not only the cells which share a face with the current cell, but also additional cells which share face with all the neighbours of the current cell and their neighbours.

Here we derive the homogenization function for the case of momentum balance. Rearranging of (5.15) will give us

$$\mathbf{u_1} + G_1(\mathbf{x_2} - \mathbf{x_1}) + r_2 T_2^{-1} [C_1 - C_2] : \frac{\nabla G_1 + \nabla G_1^T}{2} \mathbf{n}_\delta = \mathbf{u_2}. \tag{5.18}$$

As suggested in (Terekhov and Tchelepi, 2020), we can generalize this equation for any point in neighbouring cell i.e. interpolating not only to collocation point of cell 2

$$\begin{aligned} \mathbf{u_1} + G_1(\mathbf{x} - \mathbf{x_1}) + T_2^{-1} [C_1 - C_2] : \frac{\nabla G_1 + \nabla G_1^T}{2} \mathbf{n}_\delta \mathbf{n}_\delta^\mathbf{T} (\mathbf{x} - \mathbf{x_1}) \\ - r_1 T_2^{-1} [C_1 - C_2] : \frac{\nabla G_1 + \nabla G_1^T}{2} \mathbf{n}_\delta = \mathbf{u}(\mathbf{x}). \end{aligned} \tag{5.19}$$

Let us denote the $3 \times 9$ difference matrix (the second term in left-hand side) in (5.15) as the following

$$\mathbb{L}_k = T_{k+1}^{-1} \begin{bmatrix} \boldsymbol{n}_k^T(\mathbb{A}_1^k - \mathbb{A}_1^{k+1}) & \boldsymbol{n}_k^T(\mathbb{A}_6^k - \mathbb{A}_6^{k+1})^T & \boldsymbol{n}_k^T(\mathbb{A}_5^k - \mathbb{A}_5^{k+1})^T \\ \boldsymbol{n}_k^T(\mathbb{A}_6^k - \mathbb{A}_6^{k+1}) & \boldsymbol{n}_k^T(\mathbb{A}_2^k - \mathbb{A}_2^{k+1}) & \boldsymbol{n}_k^T(\mathbb{A}_4^k - \mathbb{A}_4^{k+1})^T \\ \boldsymbol{n}_k^T(\mathbb{A}_5^k - \mathbb{A}_5^{k+1}) & \boldsymbol{n}_k^T(\mathbb{A}_4^k - \mathbb{A}_4^{k+1}) & \boldsymbol{n}_k^T(\mathbb{A}_3^k - \mathbb{A}_3^{k+1}) \end{bmatrix}, \tag{5.20}$$

where equation (5.20) corresponds to difference in material stiffness between two cells, where $\mathbb{A}^k$, $\mathbb{A}^{k+1}$ correspond to $\mathbb{A}$, $\mathbb{B}$ in equation (5.15) respectively. We can rearrange equation (5.19) as follows

$$\mathbf{u_1} + \left( [\mathbb{I} \otimes (\boldsymbol{x} - \boldsymbol{x}_1)^T] + \mathbb{L}_1 [\mathbf{n}_\delta . (\boldsymbol{x} - \boldsymbol{x}_1)] - r_1 \mathbb{L}_1 \right) \begin{bmatrix} \nabla u_1 \\ \nabla v_1 \\ \nabla w_1 \end{bmatrix} = \mathbf{u}(\mathbf{x}), \tag{5.21}$$

where $\mathbb{I} \otimes (\boldsymbol{x} - \boldsymbol{x}_1)^T$ is a $3 \times 9$ matrix, $\mathbb{L}_1$ is $3 \times 9$ matrix, $\mathbf{n}_\delta \cdot (\boldsymbol{x} - \boldsymbol{x}_1)$ is a scalar value and $\mathbf{u}(\mathbf{x})$ is a $3 \times 1$ vector.

Now differentiating equation (5.21) w.r.t. $x, y, z$ should give us gradient of displacement in cell 2 (R.H.S). We performed the similar type of differentiation from equation (4.30) to (4.31) in chapter 4, and now for displacement vector the equation looks like:

$$(\mathcal{I} + (\mathbb{I} \otimes \mathbf{n}_\delta)\mathbb{L}_1)\, \nabla \mathbf{u_1} = \nabla \mathbf{u_2}, \tag{5.22}$$

where $\mathcal{I}$ is a $9 \times 9$ Identity matrix which results from differentiating $\mathbb{I} \otimes (\boldsymbol{x} - \boldsymbol{x}_1)^T$ in equation (5.21), $(\mathbb{I} \otimes \mathbf{n}_\delta)$ is a $9 \times 3$ matrix and $\nabla \mathbf{u_2}$ is the gradient of displacement in cell 2.

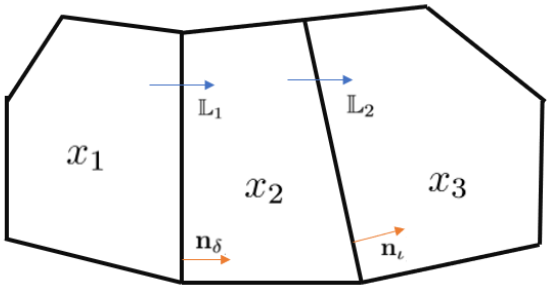



Figure 5.1: 3 cells in a 2D plane where 1 and 3 do not share interface

When we look back into equation (5.18) and (5.21), we can formulate a similar condition from cell 2 to 3 as shown in Fig. 5.1, where the normal of this interface is denoted by $\mathbf{n}_\iota$

$$\mathbf{u_2} + \left([\mathbb{I} \otimes (\boldsymbol{x}_3 - \boldsymbol{x}_2)^T] + r_3\mathbb{L}_2\right) \nabla \mathbf{u_2} = \mathbf{u_3}. \tag{5.23}$$

Here $r_3$ is the distance from center of cell 2 to the interface $\iota$. In equation (5.23) we have to replace $\mathbf{u_2}$ from equation (5.15) and $\nabla \mathbf{u_2}$ from equation (5.22)

$$\mathbf{u_1} + \nabla \mathbf{u_1}\, (\mathcal{I} + (\mathbb{I} \otimes \mathbf{n}_\delta)\mathbb{L}_1) \left([\mathbb{I} \otimes (\boldsymbol{x}_3 - \boldsymbol{x}_1)^T] + r_3\mathbb{L}_2\right) - \nabla \mathbf{u_1}\, (r_1\mathbb{L}_1) = \mathbf{u_3}. \tag{5.24}$$

Now we have to generalize the equation (5.24), same as we changed equation (5.18) to (5.21). This gives us equation in the final form.

$$\nabla \mathbf{u_1}\, [\mathcal{I} + (\mathbb{I} \otimes \mathbf{n}_\delta)\mathbb{L}_1] \left([\mathbb{I} \otimes (\boldsymbol{x} - \boldsymbol{x}_1)^T] + \mathbb{L}_2[\mathbf{n}_\iota.(\boldsymbol{x} - \boldsymbol{x}_1)] - r_1^\iota\mathbb{L}_2\right) - \nabla \mathbf{u_1}\, (r_1\mathbb{L}_1) = \mathbf{u}(\mathbf{x}) - \mathbf{u_1}, \tag{5.25}$$

where $r_1^\iota$ is the distance from cell center of 1 to the interface $\iota$. We can also write (5.25) for

arbitrary number of interpolations across interfaces which looks like

$$H_{1,k+1}(\boldsymbol{x}-\boldsymbol{x}_1) = \left[\mathbb{I}\otimes(\boldsymbol{x}-\boldsymbol{x}_1)^T + \mathbb{L}_k[\boldsymbol{n}_k^T.(\boldsymbol{x}-\boldsymbol{x}_1)]\right]\cdot \cdot\left(\prod_{i=1}^{k-1}\left[\mathcal{I}+(\mathbb{I}\otimes\boldsymbol{n}_i)\mathbb{L}_i\right]\right) - \sum_{j=1}^{k} r_1^j\mathbb{L}_j\left(\prod_{i=1}^{j-1}\left[\mathcal{I}+(\mathbb{I}\otimes\boldsymbol{n}_i)\mathbb{L}_i\right]\right), \quad (5.26)$$

where $k$ is the final interface we are interpolating to and $k+1$ is the final cell. Note that $\boldsymbol{n}_k$ is the normal to the interface between cell $k$ and $k+1$.

When the function was tested for up to 6 interpolations, we were not able to find a complete positive basis i.e. in $3\times 9$ matrix. This could be due to the fact that homogenization function is derived only for internal faces, boundary faces are not considered here. But for future work we could combine this with optimization as suggested in (Schneider et al., 2018).

## Gradient splitting

Due to the limitations of the previous two methods, we tried a third approach where we are splitting the displacement gradients and getting the corresponding coefficient matrices individually. We tried this method for a homogeneous medium for several tests. If (5.15) is considered for a homogeneous medium, then $\mathbb{L}_1$ matrix reduces to zero. This reduces the equation (5.15) in such a way that $\nabla u_1, \nabla v_1, \nabla w_1$ depend on $x, y, z$ components of $\boldsymbol{u}_2 - \boldsymbol{u}_1$ independently.

$$\begin{aligned}
\boldsymbol{m}_{11}\nabla u_1 + \boldsymbol{m}_{12}\nabla v_1 + \boldsymbol{m}_{13}\nabla w_1 &= (u_2-u_1)^x,\\
\boldsymbol{m}_{21}\nabla u_1 + \boldsymbol{m}_{22}\nabla v_1 + \boldsymbol{m}_{23}\nabla w_1 &= (u_3-u_1)^x,\\
\boldsymbol{m}_{31}\nabla u_1 + \boldsymbol{m}_{32}\nabla v_1 + \boldsymbol{m}_{33}\nabla w_1 &= (u_4-u_1)^x,
\end{aligned} \quad (5.27)$$

where $m_{12}, m_{13}, m_{22}, m_{23}, m_{32}, m_{33}$ are equal to zero, $m_{11}, m_{21}, m_{31}$ are $3\times 1$ vectors which go from collocation point 2, 3, 4 to 1 respectively.

Not only we need the above (5.27), but we also need to split the $\Gamma$ matrix from equation (5.14) into 9 different parts as shown in Figure 5.2 such that each part (which is $3\times 1$ vector) is multiplied with their respective $3\times 3$ matrix (inverted matrix from equation (5.27)). Now the resultant $3\times 1$ vector from the above matrix-vector multiplication has to have all non-negative entries. As shown in Fig. (5.2) yellow, blue and green entries are multiplied with $\nabla u_1$, $\nabla v_1$, $\nabla w_1$ respectively.

This method produces a $3\times 9$ coefficient matrix where all entries are non-negative in the meshes we used. In the later section, it is shown that the gradient splitting works consistently as the AvgMPSA scheme that uses the method is proven to converge. But the question also

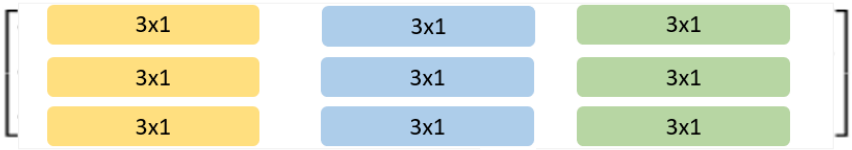



Figure 5.2: $\tau$ matrix, each of these $3 \times 1$ entries are multiplied to gradients

remains if this can be extended to heterogeneous media as we are eliminating several terms in equation (5.27). When we do not exclude these terms in equation (5.27) and use two more sets of similar equations corresponding to the y and z component of displacement, we will have nine unknowns and nine equations. And then choosing each triplet individually, for example, if we have wedges 3 out of 5 from 1st three equations, 3 out of 5 from next three, and so on such that three different triplets we choose are in the same $9 \times 9$ matrix. Inverting this matrix and multiplying with $\Gamma$ we can selectively obtain a non-negative basis, but this type of searching algorithm will be computationally expensive as we will have 1000 different combinations ($10 \times 10 \times 10$ which is 10 combinations for each gradient in polyhedron with 5 interfaces) to choose from for one cell-cell pair.

To alleviate this formulation we can try to implement an optimization problem described in (Schneider et al., 2018) and get the values of coefficients. But this will not be addressed further in our current work.

For internal faces, this procedure is carried out for both the two point cells which share this interface, and once we determine the coefficient matrix, we can proceed to compute the traction.

## 5.3 Nonlinear Weighting of Traction

The traction at the interface can be represented in a non-linear format as follows

$$\mathbf{F} = -T(\mathbf{u}_2 - \mathbf{u}_1) + \boldsymbol{\mu}_1 \mathbf{F}_{\delta 1} + \boldsymbol{\mu}_2 \mathbf{F}_{\delta 2}, \tag{5.28}$$

where the 1st part of RHS is the harmonic flux and 2nd is transversal. In the current formulation for non linear two point stress scheme $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2$ are $3 \times 1$ vectors such that $\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$. Taking $\boldsymbol{\mu}_1 = \boldsymbol{\mu}_2 = \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \end{bmatrix}$ will result in Average MPSA method.

This Average MPSA method is tested and convergence is proved, which implies the gradient reconstruction works correctly. For weight functions we need 3 pairs of weights for 3 components of tangential traction vector. Sum of each pair is equal to 1. All 3 components of transversal traction rearranged looks like this

$$\mathbf{F}_\delta = \boldsymbol{\mu}_1 \mathbf{F}_{\delta 1} + \boldsymbol{\mu}_2 \mathbf{F}_{\delta 2}, \tag{5.29}$$

$$\begin{aligned}\mathbf{F}_{\delta 1} = \boldsymbol{\mu}_1[&\mathbb{R}_2^c(\mathbf{u}_2 - \mathbf{u}_1) + \mathbb{R}_3^c(\mathbf{u}_3 - \mathbf{u}_1) + \mathbb{R}_4^c(\mathbf{u}_4 - \mathbf{u}_1) \\ &+\mathbb{R}_5^c(\boldsymbol{r}_5 - \alpha\mathbf{u}_1) + \mathbb{R}_6^c(\boldsymbol{r}_6 - \alpha\mathbf{u}_1)],\end{aligned} \tag{5.30}$$

$$\begin{aligned}\mathbf{F}_{\delta 2} = \boldsymbol{\mu}_2[&\mathbb{R}_1^d(\mathbf{u}_2 - \mathbf{u}_1) + \mathbb{R}_7^d(\mathbf{u}_2 - \mathbf{u}_7) + \mathbb{R}_8^d(\mathbf{u}_2 - \mathbf{u}_8) \\ &+\mathbb{R}_9^d(\alpha\mathbf{u}_2 - \boldsymbol{r}_9) + \mathbb{R}_{10}^d(\alpha\mathbf{u}_2 - \boldsymbol{r}_{10})],\end{aligned} \tag{5.31}$$

$$\begin{aligned}\mathbf{F}_\delta = \boldsymbol{\mu}_1[&\mathbb{R}_2^c\mathbf{u}_2 + \mathbb{R}_3^c\mathbf{u}_3 + \mathbb{R}_4^c\mathbf{u}_4 + \mathbb{R}_5^c\boldsymbol{r}_5 + \mathbb{R}_6^c\boldsymbol{r}_6 - \mathbb{R}_{11}^c\mathbf{u}_1] \\ &+ \boldsymbol{\mu}_2[\mathbb{R}_{22}^d\mathbf{u}_2 - \mathbb{R}_7^d\mathbf{u}_7 - \mathbb{R}_8^d\mathbf{u}_8 - \mathbb{R}_9^d\boldsymbol{r}_9 - \mathbb{R}_{10}^d\boldsymbol{r}_{10} - \mathbb{R}_1^d\mathbf{u}_1],\end{aligned} \tag{5.32}$$

$$\begin{aligned}\mathbf{F}_\delta = &[\boldsymbol{\mu}_1\mathbb{R}_2^c + \boldsymbol{\mu}_2\mathbb{R}_{22}^d]\mathbf{u}_2 - [\boldsymbol{\mu}_1\mathbb{R}_{11}^c + \boldsymbol{\mu}_2\mathbb{R}_1^d]\mathbf{u}_1 + \\ &\boldsymbol{\mu}_1[\mathbb{R}_3^c\mathbf{u}_3 + \mathbb{R}_4^c\mathbf{u}_4 + \mathbb{R}_5^c\boldsymbol{r}_5 + \mathbb{R}_6^c\boldsymbol{r}_6] - \boldsymbol{\mu}_2[\mathbb{R}_7^d\mathbf{u}_7 + \mathbb{R}_8^d\mathbf{u}_8 + \mathbb{R}_9^d\boldsymbol{r}_9 + \mathbb{R}_{10}^d\boldsymbol{r}_{10}],\end{aligned} \tag{5.33}$$

$$\begin{aligned}\mathbf{F}_\delta = &[\boldsymbol{\mu}_1\mathbb{R}_2^c + \boldsymbol{\mu}_2\mathbb{R}_{22}^d]\mathbf{u}_2 - [\boldsymbol{\mu}_1\mathbb{R}_{11}^c + \boldsymbol{\mu}_2\mathbb{R}_1^d]\mathbf{u}_1 + \\ &\boldsymbol{\mu}_1[\mathbb{R}_3^c\mathbf{u}_3 + \mathbb{R}_4^c\mathbf{u}_4 + \mathbb{R}_5^c\boldsymbol{r}_5 + \mathbb{R}_6^c\boldsymbol{r}_6] - (1 - \boldsymbol{\mu}_1)[\mathbb{R}_7^d\mathbf{u}_7 + \mathbb{R}_8^d\mathbf{u}_8 + \mathbb{R}_9^d\boldsymbol{r}_9 + \mathbb{R}_{10}^d\boldsymbol{r}_{10}].\end{aligned} \tag{5.34}$$

Here $\mathbf{F}_\delta$ is the total transversal tarction obtained from mixing $\mathbf{F}_{\delta 1}$ and $\mathbf{F}_{\delta 2}$, $\mathbb{R}$ is the 3x3 coefficient matrix (all non negative entries). $\mathbf{u}$ and $\boldsymbol{r}$ represent displacement vectors and boundary conditions.

Now we rearrange this equation such that $F_{\delta x}, F_{\delta y}, F_{\delta z}$ are split. In order to do that, $\mathbb{R}$ matrices should be split row-wise and multiplied with the corresponding displacement vector for each component of tangential traction. The weights mentioned below indicated by $\mu_1$ and $\mu_2$ are the x-components of weight vectors $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ respectively. For wedges we will have a maximum of 6 cells involved in a semi-flux calculation. The $\mathbb{R}$ matrices corresponding to these displacement vectors will be very sparse as not all cells contribute to each gradient reconstruction.

$$\begin{aligned}F_{\delta x} = &[\mu_1\mathbb{R}_2^{cx} + \mu_2\mathbb{R}_{22}^{dx}]\mathbf{u}_2 - [\mu_1\mathbb{R}_{11}^{cx} + \mu_2\mathbb{R}_1^{dx}]\mathbf{u}_1 \\ &+\mu_1[\mathbb{R}_3^{cx}\mathbf{u}_3 + \mathbb{R}_4^{cx}\mathbf{u}_4 + \mathbb{R}_5^{cx}r_5 + \mathbb{R}_6^{cx}r_6] \\ &-\mu_2[\mathbb{R}_7^{dx}\mathbf{u}_7 + \mathbb{R}_8^{dx}\mathbf{u}_8 + \mathbb{R}_9^{dx}r_9 + \mathbb{R}_{10}^{dx}r_{10}],\end{aligned} \tag{5.35}$$

where the weight equations are

$$\mu_1 + \mu_2 = 1, \quad \mu_1\mathbb{R}_t^{cx} - \mu_2\mathbb{R}_t^{dx} = 0. \tag{5.36}$$

Here $\mathbb{R}_t^{cx} = [\mathbb{R}_3^{cx}u_3 + \mathbb{R}_4^{cx}u_4 + \mathbb{R}_5^{cx}r_5 + \mathbb{R}_6^{cx}r_6]$ and $\mathbb{R}_t^{dx} = [\mathbb{R}_7^{dx}u_7 + \mathbb{R}_8^{dx}u_8 + \mathbb{R}_9^{dx}r_9 + \mathbb{R}_{10}^{dx}r_{10}]$.

Note that in (5.32) the weights are 3x1 vectors and in (5.35) and (5.36) they are single valued as we split the (5.32) into 3 components. Using these two equations weights can be computed as

$$\mu_1 = \frac{\mathbb{R}_t^{dx}}{\mathbb{R}_t^{dx} + \mathbb{R}_t^{cx}}, \quad \mu_2 = \frac{\mathbb{R}_t^{cx}}{\mathbb{R}_t^{dx} + \mathbb{R}_t^{cx}}. \tag{5.37}$$

Now while computing the total traction we add the harmonic part mentioned in (5.28) and transversal gradient also reduces such that only the two point parts i.e. $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ remain

$$\mathbf{F} = \mathbb{T}_1\mathbf{u}_1 - \mathbb{T}_2\mathbf{u}_2, \tag{5.38}$$

where $\mathbb{T}_1$ and $\mathbb{T}_2$ compose of both harmonic and transversal parts and depend on displacement in other cells in the stencil like $\boldsymbol{u}_3, \boldsymbol{u}_4$ etc.

Domain shifting is performed and tested in both AvgMPSA and non-linear TPSA which ensures $\boldsymbol{u}$ is always positive. We add a large value to the initial displacement for all cells in our domain such that displacement solution is never negative when we solve the equation with any specified boundary conditions.

A second form of weights can also be considered as follows.

## Nine pairs of weights

Taking 3 pairs of weights might be restricting for residual equation, and when we used (5.33) and (5.38) to construct residual it takes too many non-linear iterations before converging.

In an effort to resolve this issue, a different type of convex combination is considered where we use 9 pairs of weights. In this formulation, the $3 \times 3$ matrix of weights is used such that we multiply it component wise and balance the weights individually, which results in 9 separate conditions for each of the two equations in (5.41)

$$\mathbf{F}_\delta = \boldsymbol{\mu}_1 \circ \mathbf{F}_{\delta 1} + \boldsymbol{\mu}_2 \circ \mathbf{F}_{\delta 2}, \tag{5.39}$$

where $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ are $3 \times 3$ matrices whose entries form the 9 pairs of weights mentioned above. The second part of equation (5.34) will look like:

$$\begin{bmatrix} \mu_{11} & \mu_{12} & \mu_{13} \\ \mu_{14} & \mu_{15} & \mu_{16} \\ \mu_{17} & \mu_{18} & \mu_{19} \end{bmatrix} \circ \left( \begin{bmatrix} R_{31} & R_{32} & R_{33} \\ R_{34} & R_{35} & R_{36} \\ R_{37} & R_{38} & R_{39} \end{bmatrix} \begin{bmatrix} u_3^x \\ u_3^y \\ u_3^z \end{bmatrix} + ... \right)$$
$$- \begin{bmatrix} \mu_{21} & \mu_{22} & \mu_{23} \\ \mu_{24} & \mu_{25} & \mu_{26} \\ \mu_{27} & \mu_{28} & \mu_{29} \end{bmatrix} \circ \left( \begin{bmatrix} R_{71} & R_{72} & R_{73} \\ R_{74} & R_{75} & R_{76} \\ R_{77} & R_{78} & R_{79} \end{bmatrix} \begin{bmatrix} u_7^x \\ u_7^y \\ u_7^z \end{bmatrix} + ... \right) = 0, \tag{5.40}$$

$$\mu_{11} + \mu_{21} = 1, \quad \mu_{11}(R_{31}u_3^x) - \mu_{21}(R_{71}u_7^x) = 0, \tag{5.41}$$

$$\mu_{11} = \frac{R_{71}u_7^x}{R_{31}u_3^x + R_{71}u_7^x}, \quad \mu_{21} = \frac{R_{31}u_3^x}{R_{31}u_3^x + R_{71}u_7^x}, \tag{5.42}$$

where we can selectively eliminate the weights such that summation of 1st term of $\mu_1 \times R_3 \times u_3^x$ and 1st terms of $\mu_2 \times R_7 \times u_7^x + ...$ is zero. This means that coefficients corresponding to $u^x, u^y$ and $u^z$ will be eliminated separately such that we will have 9 additional equations. The original 9 equations were formed by $\mu_1 + \mu_2 = 1$ .

Both the weight approximations used in (5.36) and (5.42) are tested, and we did not observe any significant improvement in number of iterations.

## 5.4 Linear Homogenized Traction

The previous three sections helped us to build our non-linear two-point stress approximation framework. In this section, we will provide a modification to the multi-point stress approximation used in (Terekhov and Tchelepi, 2020). During the least-squares gradient reconstruction process, we can take an extended stencil that has coefficients corresponding to the homogenized value as seen in equation (5.26). This means utilizing more cells instead of just five adjacent interfaces for wedges or six for hexahedrons. We can consider the neighbor interfaces such that more number of cells are included. This can affect the traction profile which will be later discussed in Chapter-6.

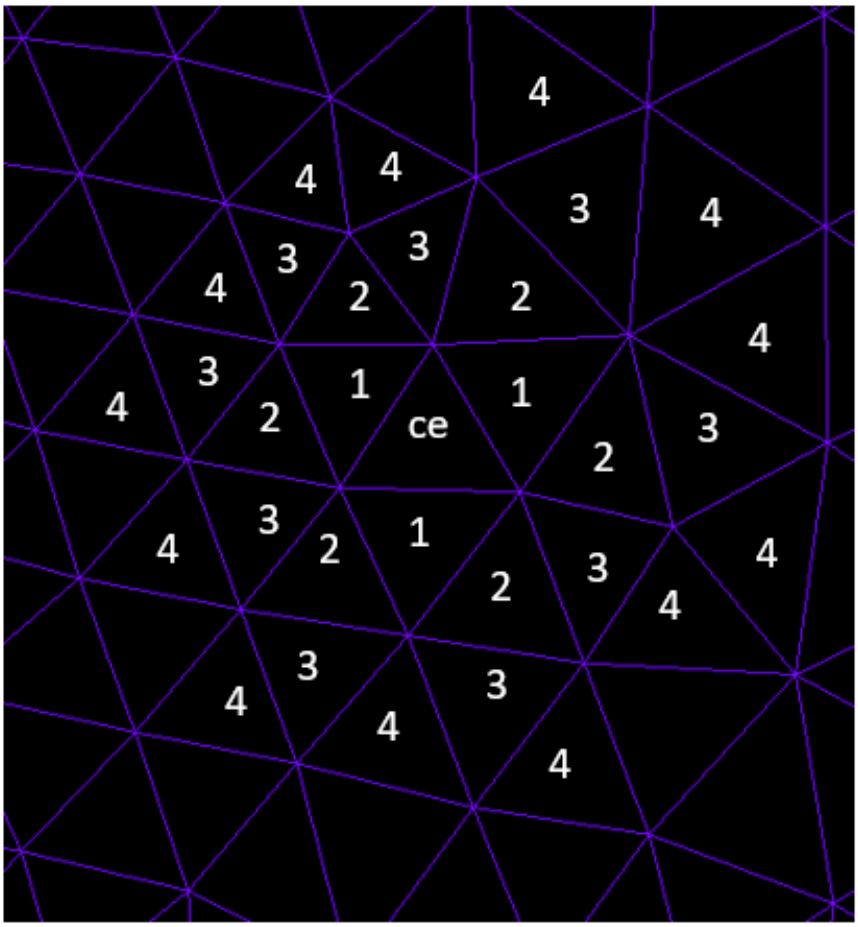



Figure 5.3: For any interface $\sigma$ in cell 'ce' we can take an extended stencil

As shown in Figure 5.3, our cell is denoted by 'ce'. In the MPSA formulation presented in (Terekhov and Tchelepi, 2020), only its neighboring cells denoted by to '1' are accounted in gradient reconstruction. We will refer to these cells as level 1. But with the help of homogenized function we can consider cells in levels 2, 3, or more such that the one-sided traction approximation in equation (5.30) and (5.31) contain information related to these additional cells.

## 5.5 Jacobian Terms

Once we determine the traction flux across the interface $\sigma$ using the equation (5.38), we have to derive the jacobian terms w.r.t all the displacement vectors involved in the stencil. Jacobian terms are required as we solve this nonlinear problem by using Newton method. Once we derive these terms, we assemble our Jacobian ($J$) matrix and residual array, and then invert the $J$ matrix to evaluate the change in displacement. These values are used to update the displacement from previous iteration and this process is repeated until we converge i.e. $L^2$ norm of residual is below the tolerance level.

# Chapter 6

# Numerical Experiments

In this chapter, we will test our discretization framework developed for both flow and mechanics on some general test problems discussed in previous works.

## 6.1 Fluid Flow in Porous Media

The following experiments are performed on our MPFA and NTPFA discretization within the DARTS framework. Some cases are specifically designed to observe the difference between the linear and nonlinear methods. In other cases, we evaluate the performance of the NTPFA framework in terms of accuracy and standardized test cases.

### 6.1.1 DMP Test

In the first test, we investigate the monotonicity and violation of DMP of our flow model. Note that this numerical experiment is done on a single-phase stationary fluid flow in order to limit the equations to purely elliptical. The basic idea is to show that the gradient reconstruction and nonlinear weighting will seldom preserve the positivity of the solution. We can perform this experiment on a structured mesh but, to introduce a stronger non-K-orthogonality, we take an unstructured mesh and extrude it in the z-axis and create a square-shaped hole in the middle of the domain. The square hole will act as our inner boundary, and we will impose Dirichlet boundary conditions on both the outer and inner boundaries of the current domain.

The following permeability tensor is taken for the test:

$$\mathbb{K} = \mathbb{R}_{xyz} \begin{bmatrix} 1000 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbb{R}_{xyz}^{T}, \tag{6.1}$$

$$\mathbb{R}_{xyz} = \mathbb{R}_x(0)\mathbb{R}_y(0)\mathbb{R}_z(\pi/6),$$

where $\mathbb{R}$ is the rotation matrix, and in our problem, we rotate the permeability tensor to $30^o$ degree along the $z$-axis, in $\overline{xy}$ plane. The domain spans $[0, 1]^2$ in $\overline{xy}$ plane and extruded to thickness of 0.05 m in $z$-direction. We also take the following mesh presented in Figure 6.1, which consists of adaptive hexahedrons (smaller on the inner boundary and slightly larger on the outer boundary). The initial and boundary conditions are given in Tables 6.1 and 6.2.

Table 6.1: Initial and boundary pressure conditions for $1^{st}$ DMP test

| **Pressure** | **Value** |
|---|---|
| Initial | 0.5 |
| Inner bound | 1.0 |
| Outer Bound | 0.0 |

Table 6.2: Initial and boundary pressure conditions for $2^{nd}$ DMP test

| **Pressure** | **Value** |
|---|---|
| Initial | 10.5 |
| Inner bound | 11.0 |
| Outer Bound | 10.0 |

We get the following results for average MPFA and NTPFA in Figures 6.2 and 6.3 respectively. We see the NTPFA solution to be more physical as it preserves the positivity of the solution. Whereas the MPFA solution goes below 0 (indicated by purple). This is usually seen in extreme cases of anisotropy, and for our case, we did take 1000:1 ratio in x and y directions. We can attribute these non-physical oscillations observed in the MPFA method to the method being non-monotone as we will not have any control over structuring our $A$ matrix in pressure solver as $M$ matrix. But in the NTPFA case, due to the preservation of monotonicity, we do not see any non-physical oscillations.

Now we analyze the results from $2^{nd}$ test which has input conditions mentioned in Table 6.2. As we perform the test for a pressure range that is not close to zero, we might observe if the NTPFA solution actually violates maximum or minimum. As suggested in papers (Vassilevski et al., 2020) and (Schneider et al., 2018), NTPFA can violate the maximum or minimum. As we can observe in Figure 6.4, average MPFA violates both maximum and minimum (black and purple respectively), but NTPFA only violates the minimum. If we invert the boundary conditions i.e., higher on the outside and lower on the inside, we see NTPFA violates the maximum instead of minimum. This means NTPFA can violate either maximum or minimum in our case, which is similar to the finding in (Vassilevski et al., 2020) and (Schneider et al., 2018). Also, MPFA will violate both maximum and minimum, and the intensity of violation is larger than NTPFA i.e., a higher negative value in MPFA case. The solution of the MPFA-O method, which is one of the first general MPFA methods proposed

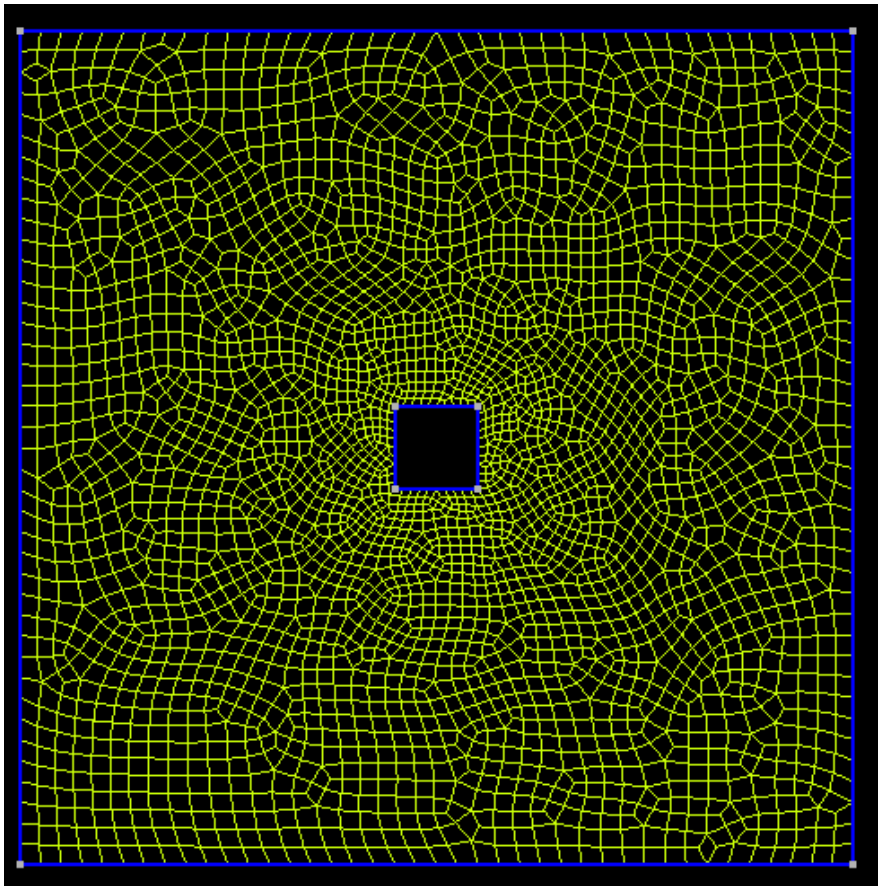


Figure 6.1: Unstructured mesh with adaptive hexahedrons from inner to outer boundaries

in (Aavatsmark, 2002) is prone to non-physical oscillations as proved in (Vassilevski et al., 2020) for a similar test case of using Dirichlet boundaries.

In Table 6.3, we can see how the violations zones vary for each test and case. Overall, NTPFA produced a fewer number of violations in cells in Test 2. But this can change when we define a different set of boundaries or increase the difference between the maximum and minimum Dirichlet boundaries. Also, implementing and analyzing a Nonlinear MPFA can provide us more insights as that discretization technique is known to preserve both maximum and minimum in boundary conditions assumed in Test 2.

Table 6.3: Statistics of DMP test. Violation of DMP for may be found in comparison to boundary values from Tables 6.1, 6.2.

| **Test Type** | **Test 1** | | | **Test 2** | | |
|---|---|---|---|---|---|---|
| | Minimum | Maximum | $\%cells$ | Minimum | Maximum | $\%cells$ |
| Average MPFA | -0.02486 | 0.99277 | 35 | 9.9733 | 11.1061 | 35 |
| NTPFA | 1.64e-9 | 0.99457 | 0 | 9.9813 | 10.9937 | 34 |

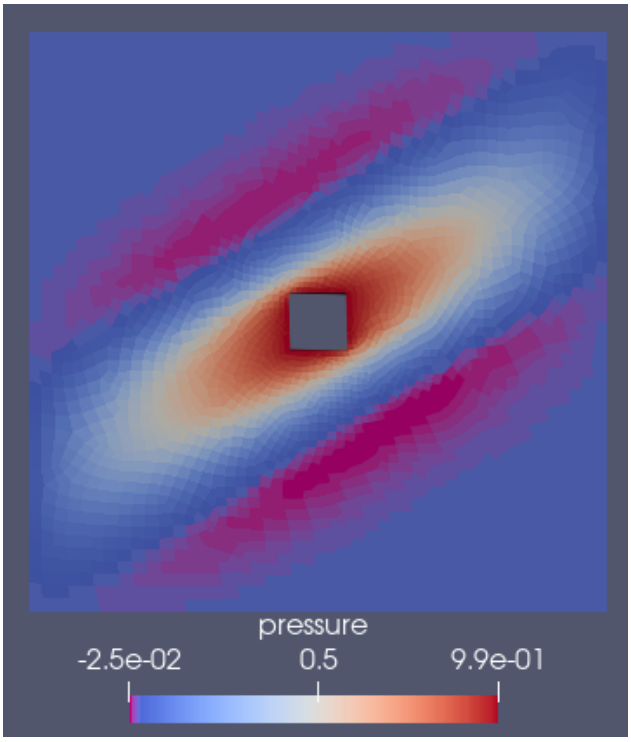



Figure 6.2: Pressure solution for Average MPFA discretization framework for Dirichlet boundaries

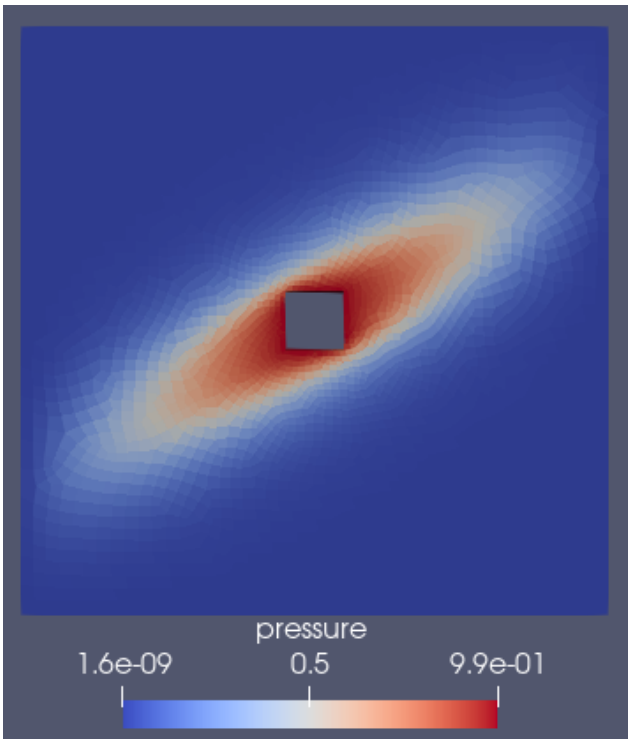
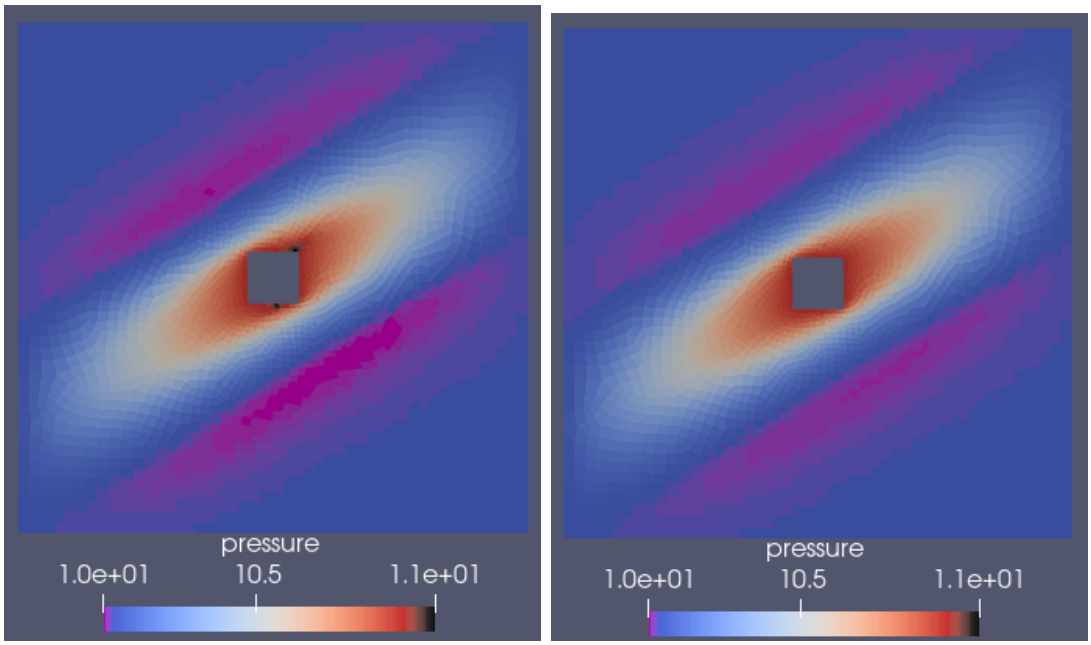



Figure 6.3: Pressure solution for NTPFA discretization framework for Dirichlet boundaries




(a) (b)

Figure 6.4: Pressure solution for (a) Avg MPFA (b) NTPFA

### 6.1.2 Convergence Study

In this section, we perform two convergence studies picked up from (Schneider et al., 2018), which are used to prove that MPFA or NTPFA is convergent in the presence of anisotropy as linear TPFA will not have a converging solution for these permeability test cases. We will try to prove that pressure is $2^{nd}$ order converging, which is true for the elliptical flow equation. The procedure for convergence test will involve computing the numerical error associated with the difference between analytical and numerical solutions. We consider the $L^2$ - norm for pressure as suggested in (Schneider et al., 2018)

$$e_p = \left( \frac{\Sigma_i |V_i| (p_{ex}(x_i) - p_i)^2}{\Sigma_i |V_i| p_{ex}(x_i)^2} \right)^{\frac{1}{2}}, \tag{6.2}$$

where $p_{ex}(x_i)$ is the value of analytical pressure at collocation point $x_i$ and $|V_i|$ is the volume of individual cell. We can calculate the convergence rate based on

$$cr(j) = -n_{dims} \frac{\log_{10}\left(e(j)/e(j-1)\right)}{\log_{10}\left(n_{dof}(j)/n_{dof}(j-1)\right)}, \tag{6.3}$$

where $n_{dof}$ is the number of degrees of freedom which is equivalent to the number of cells for the elliptical problem, $n_{dims}$ is the spatial dimension of the problem, which is 3, and $j$ denotes the level of grid resolution we use.

### Grids

We use five levels of grid resolutions in our analysis, from coarse to fine, for both structured and unstructured meshes. We impose our boundary conditions on all these meshes and solve for pressure. In essence, we can use even more complicated meshes (eg. checkerboard or non-convex). But, in order to keep a smooth boundary and impose equivalent boundary conditions for all meshes we just wanted to show it for structured and unstructured hexahedrons. The spatial domain is defined in $\Omega = [0, 1]^3$. We consider 5 meshes for each of the

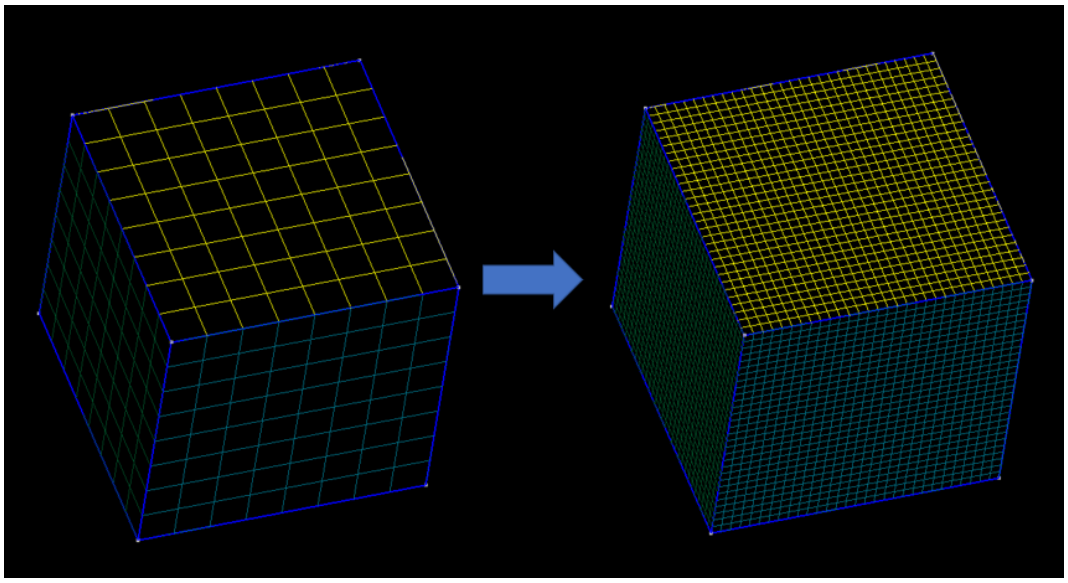


Figure 6.5: Structured meshes used in the range of left (512 cells) to right (27000 cells) mesh

case shown in Figures 6.5 and 6.6 i.e. 10 meshes in total. In this way a smooth boundary is imposed in every study.

### Analytical Benchmark: Mild Anisotropy

Usually, in a convergence test, we have the solution for the problem known to us i.e. an equation describing how pressure looks in the domain. Then using this information in equation (4.1), we can derive the source/sink term $f$. Then, we can use this value of $f$ in every cell and the boundary conditions in Dirichlet type defined by our pressure equation (6.5), and we can run our simulator to obtain the numerical solution. The Permeability tensor and equation for pressure looks like

$$\mathbb{K} = \begin{bmatrix} 1 & 0.5 & 0 \\ 0.5 & 1 & 0.5 \\ 0 & 0.5 & 1 \end{bmatrix} \tag{6.4}$$

$$p_{ex}(x, y, z) = \sin(\pi x) \sin\left(\pi\left(y + \frac{1}{2}\right)\right) \sin\left(\pi\left(z + \frac{1}{3}\right)\right) + 1. \tag{6.5}$$

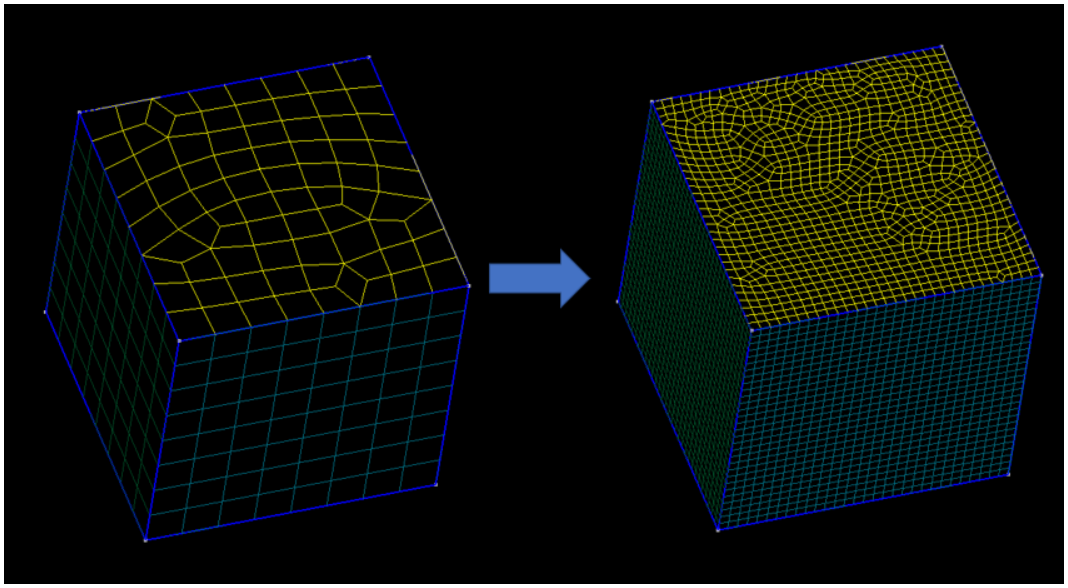


Figure 6.6: Unstructured meshes used in the range of left (624 cells) to right (31350 cells) mesh

The Analytical solutions is shown in Figure 6.7, where we can see how pressure behaves on the boundaries of the domain.

### Analytical Benchmark: Strong Anisotropy

In this case, we use the same procedure as above, but the permeability tensor involved in the discretization has stronger anisotropy (1000:1) which makes the problem highly non-K-orthogonal. The pressure solution for this setup is also different from the mild case and is represented by equation (6.7).

$$\mathbb{K} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 10^3 \end{bmatrix} \tag{6.6}$$

$$p_{ex}(x, y, z) = \sin(2\pi x)\ \sin(2\pi y)\ \sin(2\pi z) + 1, \tag{6.7}$$

### Results

In the Figures 6.9 and 6.10, we can see that w.r.t a reference slope of 2, our solutions look similar i.e. the slope is approximately 2. Even in the unstructured case, there is not much difference. The $x$ - axis represents the number of degrees of freedom which is equal to the number of cells and, the $y$ - axis represents the $log_{10}$ of error norm shown in equation (6.2). The $x$ axis h has also a log scale.

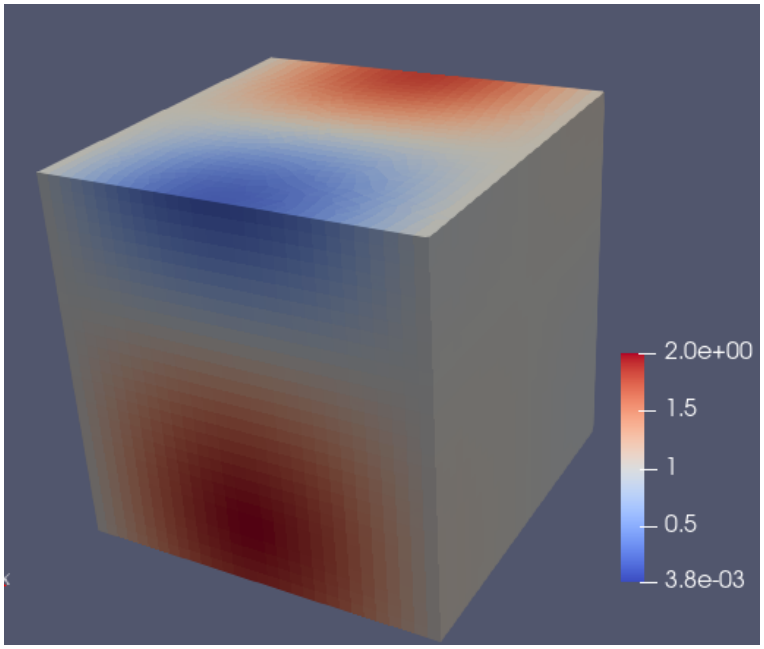



Figure 6.7: Pressure profile for Mild anisotropy test.

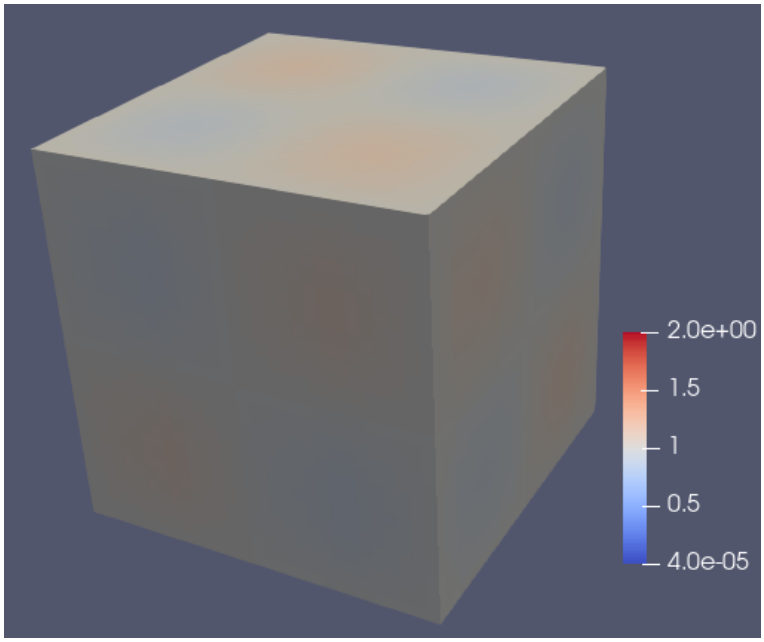



Figure 6.8: Pressure profile for Strong anisotropy test

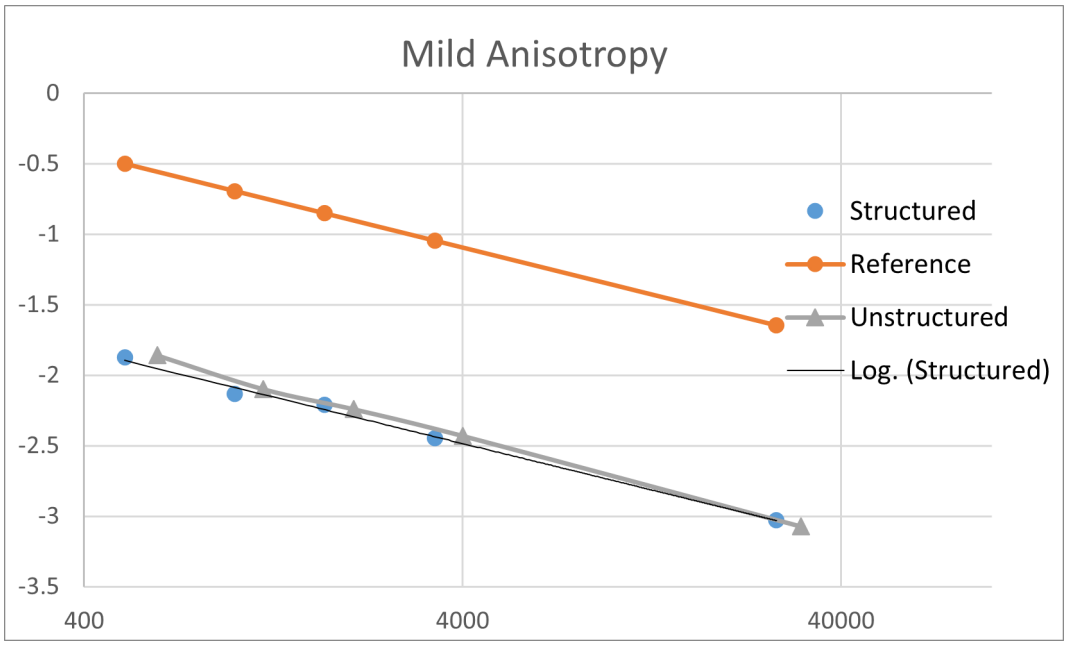



Figure 6.9: Convergence rate for Mild anisotropy test.

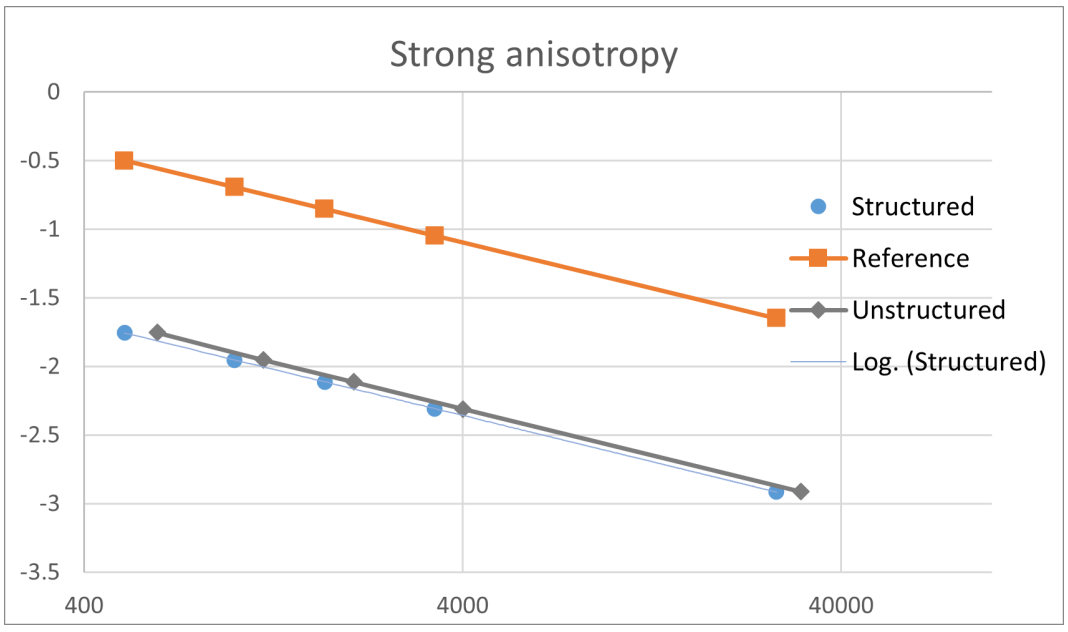



Figure 6.10: Convergence rate for Strong anisotropy test.

Table 6.4: Input parameters for multi-phase simulation

| **Properties** | **Oil** | **Water** |
|---|---|---|
| Density ($kg/m^3$) | 848.9 $kg/m^3$ | 1025.2 |
| Viscosity ($cp$) | 2.85 | 0.3 |
| Compressibility ($bar^{-1}$) | - | 4.5E-05 |
| $B_{oil}$ | 1.05 | - |

### 6.1.3 Multi-phase Flow Problem

We will discuss the solutions related to saturation transport and the two-phase flow problem in this part. Our underlying assumptions are briefly discussed in Chapter 2. Water will displace oil, and both the phases, including the rock, are considered compressible. All the input parameters are delineated in Table 6.4.

For this problem we consider several meshes spanning from coarsely structured and unstructured to finer adaptive hexahedron and wedge meshes. The main goal of this exercise is to understand the flow path of the saturation front when a discontinuous anisotropic permeability tensor is introduced in the domain $\Omega$.

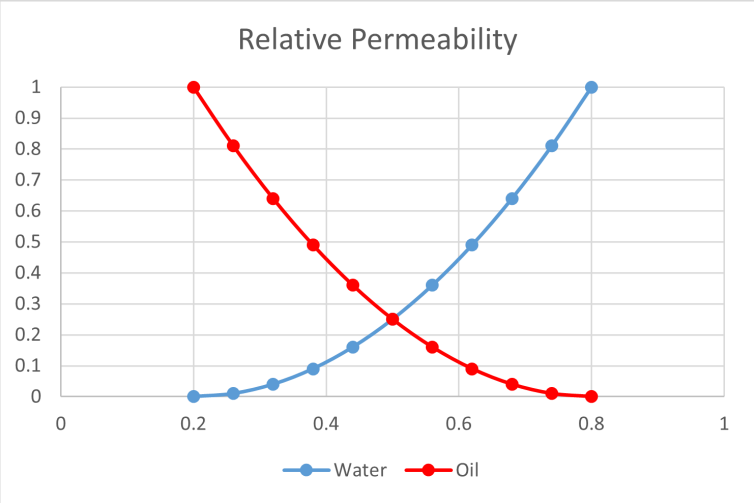



Figure 6.11: Relative permeability w.r.t. water saturation for both phases

For this case, a Linear TPFA framework will not give an accurate result due to the presence of strong anisotropy and heterogeneity. For the same case, it was performed in the literature studies by Nikitin et al. (2013) and Vassilevski et al. (2020). We tested it on a 2D mesh which gives us more or less similar results shown in Figure 6.13. Wells are assumed to be located just inside the domain and do not contact the boundaries.

Now, if we look at the NTPFA solution, we can see the saturation front follows a sharp path, unlike the TPFA solution which is more spread out. This is due to the fact that our pressure

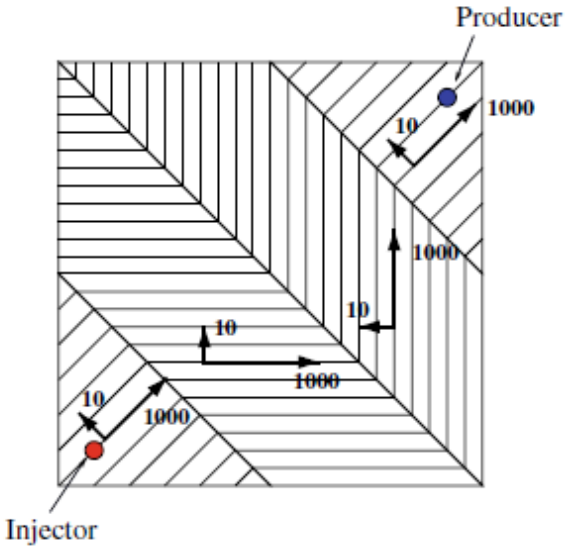
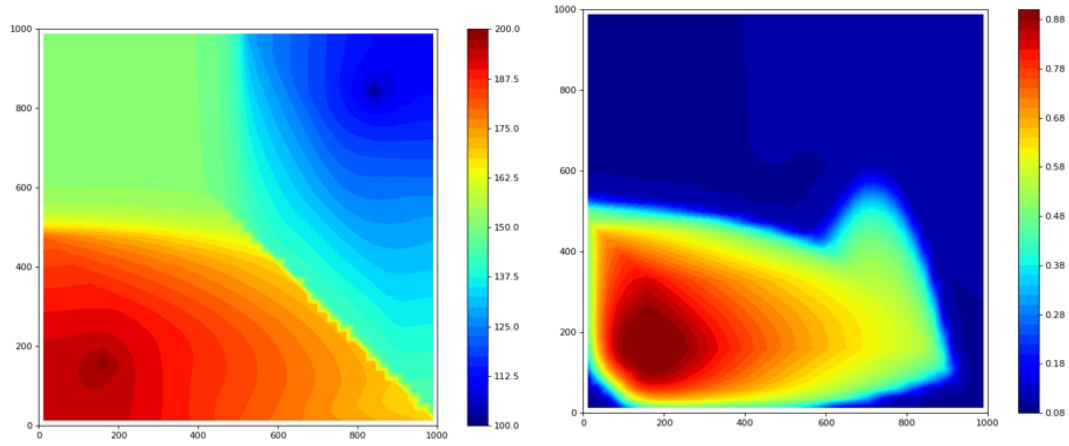



Figure 6.12: Setup depicting distribution of permeability tensor in our domain (Vassilevski et al., 2020)



(a) Pressure (b) Saturation

Figure 6.13: Saturation Transport solution by Linear TPFA simulator showing distribution of (a) Pressure (b) Saturation, in our domain

solution is different in both cases. The NTPFA pressure solution is demonstrated in Figure. 6.14. The coarse mesh used in this test case may limit the accuracy of solution presented.

On an adaptive grid with the same discontinuous tensor, we will have solutions which are illustrated in Figure 6.15 and Figure 6.16. The simulation is performed by using $dt = 200$ days for a total time of 12000 days i.e. 60-time steps. The statistics are reported in Table 6.5.

As the time-step was very large (200 days), it was expected to have a higher number of

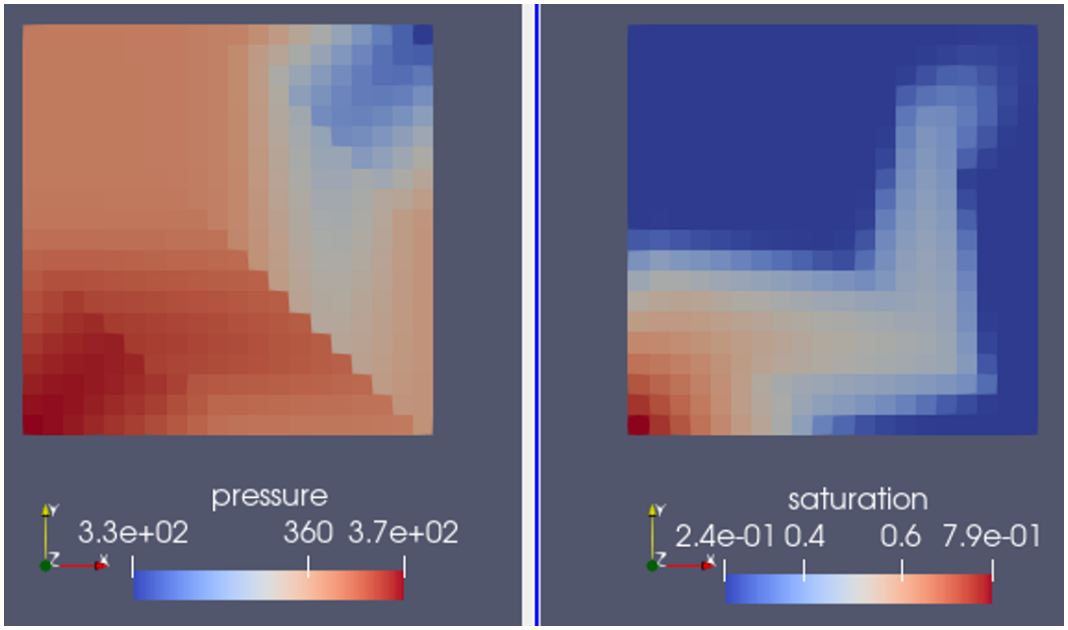
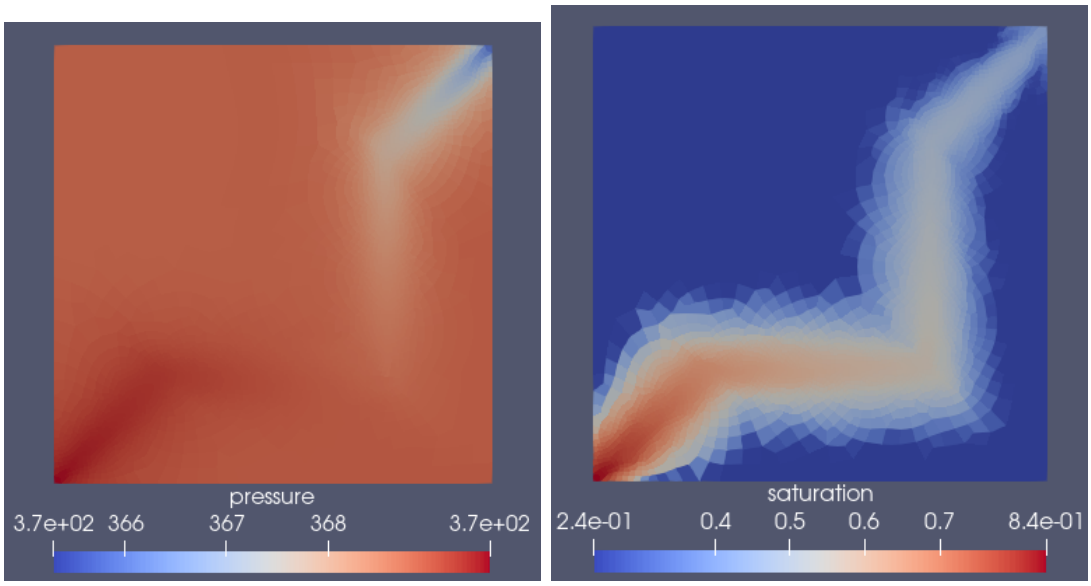



Figure 6.14: Solution for pressure (left) and saturation (right) calculated by Nonlinear TPFA simulator on a structured mesh.

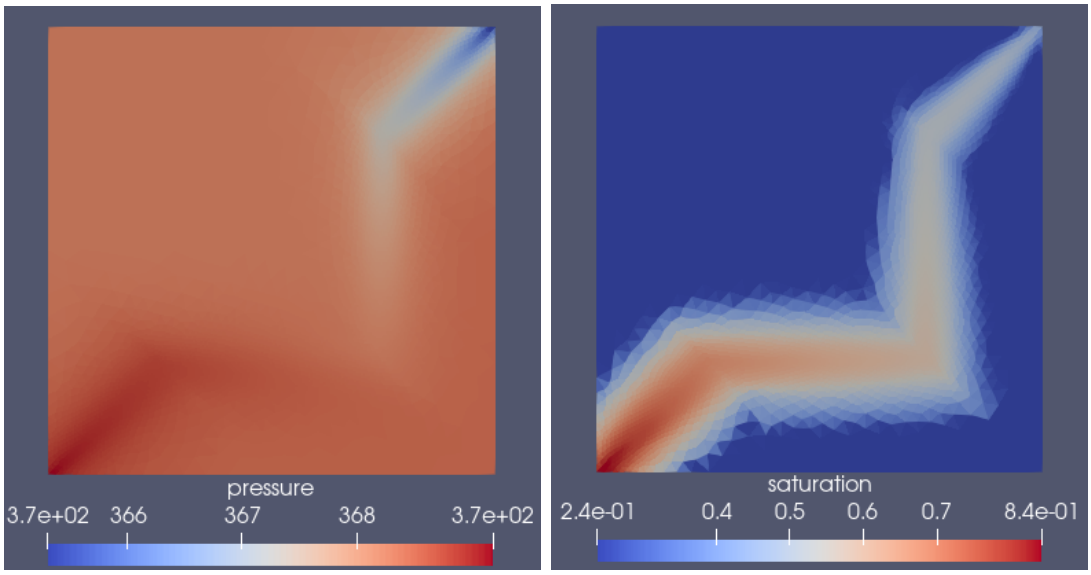
Table 6.5: Multi-phase flow test in different meshes

| **Mesh Type** | $\#cells$ | $NI$ | $NI/dt$ |
|---|---|---|---|
| Coarse Transfinite | 576 | 176 | 2.93 |
| Coarse Hexahedrons | 669 | 195 | 3.25 |
| Adaptive Hexahedrons | 3277 | 220 | 3.67 |
| Adaptive Wedges | 6702 | 244 | 4.07 |
| Fine Transfinite | 6561 | 248 | 4.13 |
| Fine Wedges | 7064 | 249 | 4.15 |




(a) Pressure (b) Saturation

Figure 6.15: Solution for pressure (left) and saturation (right) calculated by Nonlinear TPFA simulator using unstructured adaptive hexahedral mesh.




(a) Pressure (b) Saturation

Figure 6.16: Solution for pressure (left) and saturation (right) calculated by Nonlinear TPFA simulator using unstructured adaptive wedge mesh.

nonlinear iterations (between 3 to 4). The adaptive grids were finer along the path taken by saturation front. The solution on the structured fine grid looks to be more accurate at the shock front location. We also have a different well index for each case as it depends on the size of the cell, so for now, this analysis neglects this difference.

### 6.1.4 Discrete Fracture Network

In this section, we try to test our discretizer when combined with the fracture network. We say fracture network, but the examples mainly consider flow in subsurface domain with matrix anisotropy and the set of single fracture, intersecting fractures and barrier. The flow will be driven by wells (we use the Peaceman well model). All details are indicated in Table 6.6. The following permeability tensor is considered in Table 6.6.

Table 6.6: DFN + NTPFA for different scenarios

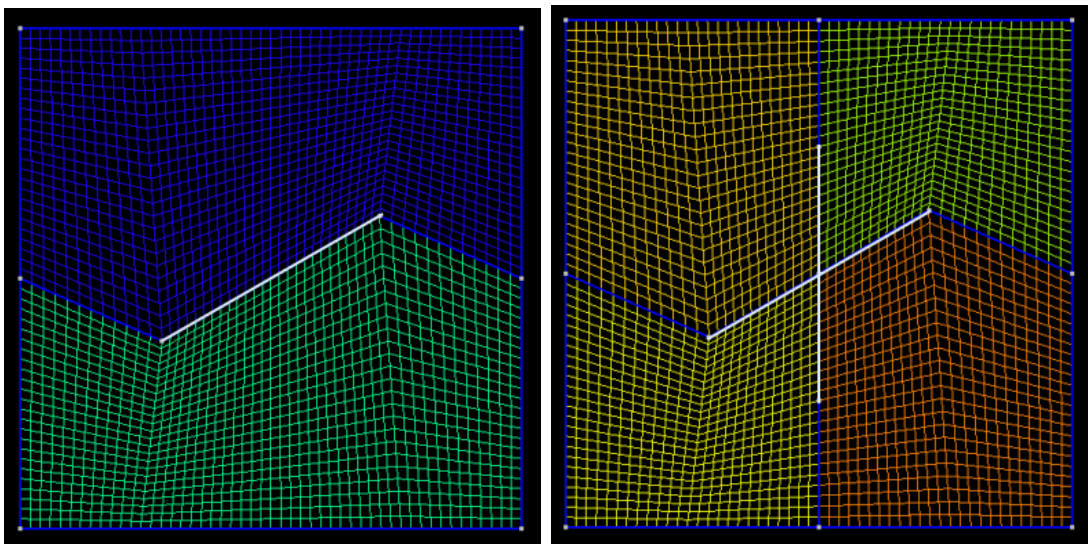
| **Case** | **Permeability** $(mD)$ | **Well location** | **Aperture** $mm$ | **t** $days$ | $NI$ | $\frac{NI}{dt}$ |
|---|---|---|---|---|---|---|
| Single Fracture | $\mathbb{K}_1$ | $\mathcal{I}$ $(0,0)$<br>$\mathcal{P}$ $(50,100)$<br>$\mathcal{P}$ $(100,0)$ | 1 | 300 | 137 | 4.5 |
| Single Fracture | $\mathbb{K}_2$ | $\mathcal{I}$ $(0,0)$<br>$\mathcal{P}$ $(50,100)$<br>$\mathcal{P}$ $(100,0)$ | 1 | 300 | 136 | 4.5 |
| Barrier | $\mathbb{K}_2$ | $\mathcal{I}$ $(100,0)$<br>$\mathcal{P}$ $(0,100)$ | $10^{-12}$ | 600 | 208 | 3.4 |
| Intersecting Fracture | $\mathbb{K}_1$ | $\mathcal{P}$ $(100,50)$<br>P $(50,100)$<br>$\mathcal{I}$ $(0,0)$ | 1 | 300 | 143 | 4.7 |
| Intersecting Fracture | $\mathbb{K}_3$ | $\mathcal{I}$ $(0,0)$<br>$\mathcal{P}$ $(50,100)$<br>$\mathcal{P}$ $(100,50)$ | 1 | 300 | 145 | 4.8 |

$$\mathbb{K}_1 = \mathbb{R}_{+85}\begin{bmatrix}1000 & 0 & 0\\ 0 & 10 & 0\\ 0 & 0 & 10\end{bmatrix}\mathbb{R}^T_{+85}, \quad \mathbb{K}_2 = \mathbb{R}_{-45}\begin{bmatrix}500 & 0 & 0\\ 0 & 10 & 0\\ 0 & 0 & 10\end{bmatrix}\mathbb{R}^T_{-45}, \quad \mathbb{K}_3 = \mathbb{R}_{-5}\begin{bmatrix}1000 & 0 & 0\\ 0 & 10 & 0\\ 0 & 0 & 10\end{bmatrix}\mathbb{R}^T_{-5}, \tag{6.8}$$

where the rotation matrices are

$$\mathbb{R}_{+85} = \mathbb{R}_x(0)\mathbb{R}_y(0)\mathbb{R}_z(+85^o), \quad \mathbb{R}_{-45} = \mathbb{R}_x(0)\mathbb{R}_y(0)\mathbb{R}_z(-45^o), \quad \mathbb{R}_{-5} = \mathbb{R}_x(0)\mathbb{R}_y(0)\mathbb{R}_z(-5^o), \tag{6.9}$$

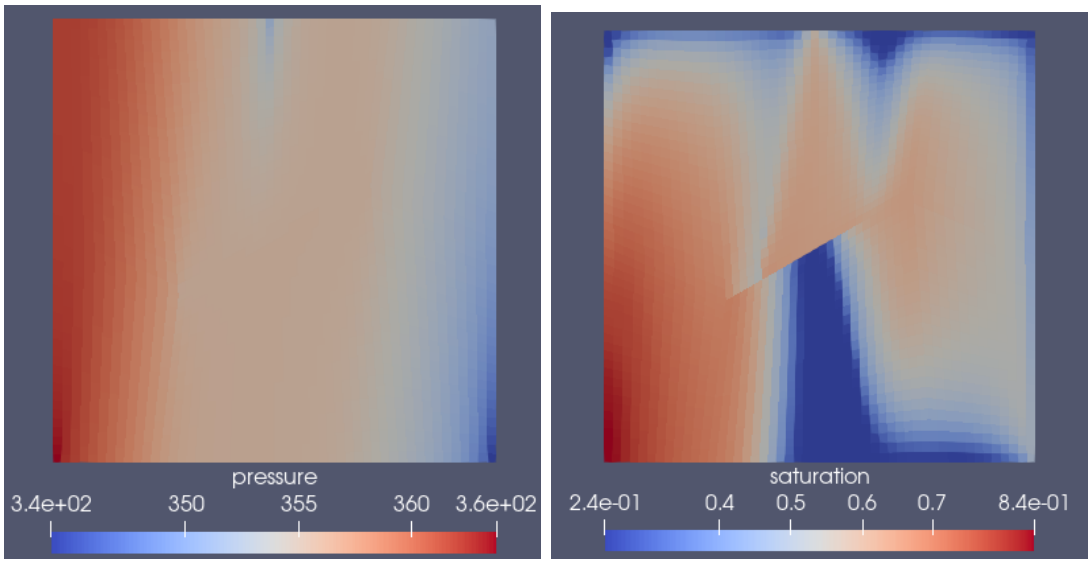
And $\mathcal{I}$ and $\mathcal{P}$ denote injector and producer in our domain, which spans $[0,100]^2$ and extruded in the z-direction. The total number of cells for both meshes, as shown in Figure 6.17 is 2500, and the number of fracture elements is 26 and 52 respectively. The following solutions in Figures 6.18, 6.19, 6.20, 6.21 and 6.22 are obtained from the tests carried out as mentioned above. Note that results are displayed after total simulation time of 300 days for all.



(a) Single (b) Intersect

Figure 6.17: Meshes used for (a) Single Fracture/Barrier (b) Intersecting Fracture, with unstructured hexahedrons and fractures are indicated by 'white' lines

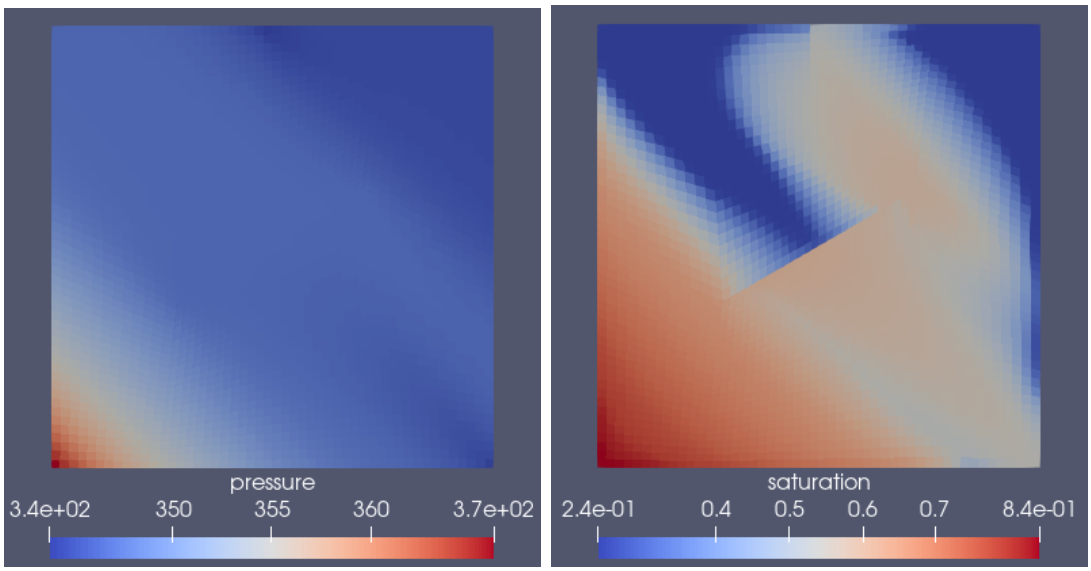
### Single Fracture




(a) Pressure (b) Saturation

Figure 6.18: Single fracture case-1 (a) Pressure (b) Saturation, after 300 days

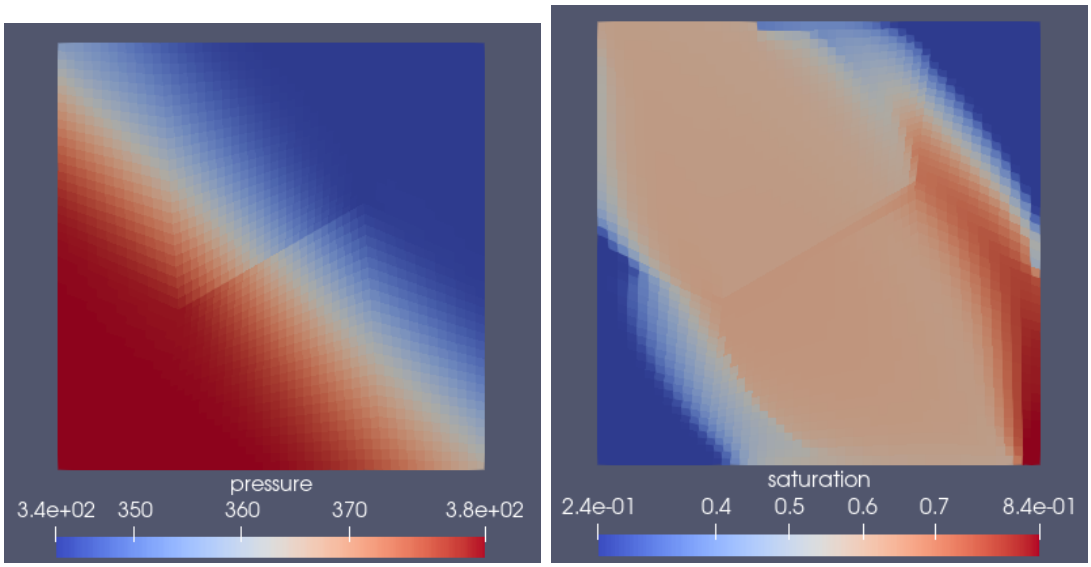
We can consider the solutions obtained in 6.18 and 6.19 to be physically plausible as saturation front initially travels in the direction of anisotropy, but as soon as it reaches the fracture, the front is influenced by it and communication of matrix cells in start and end of fracture is disrupted as the flow will only go through the fracture.




(a) Pressure (b) Saturation

Figure 6.19: Single fracture case-2 (a) Pressure (b) Saturation, after 300 days

### Barrier

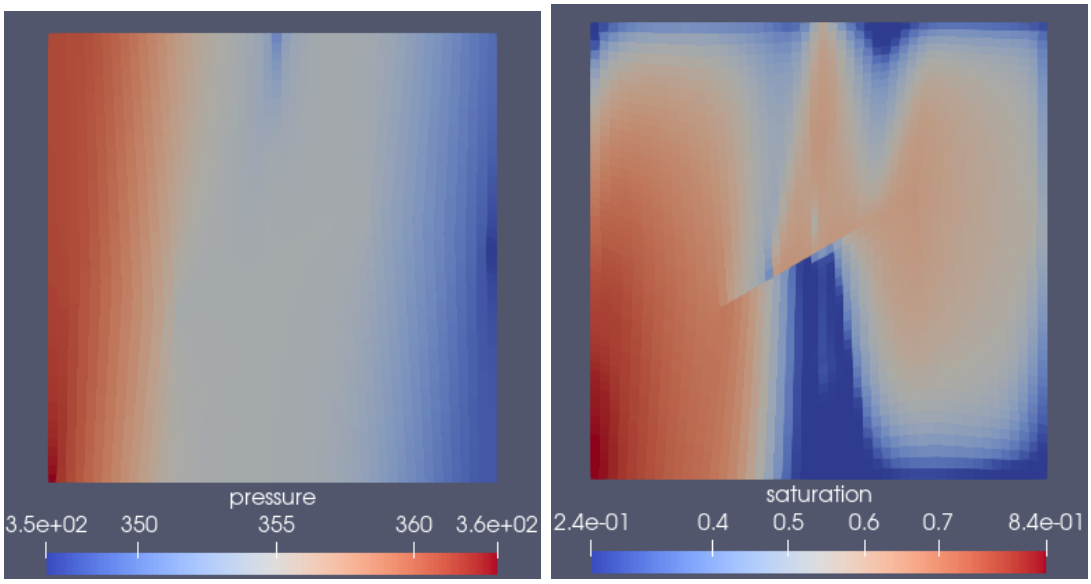



(a) Pressure (b) Saturation

Figure 6.20: Barrier case (a) Pressure (b) Saturation, after 600 days

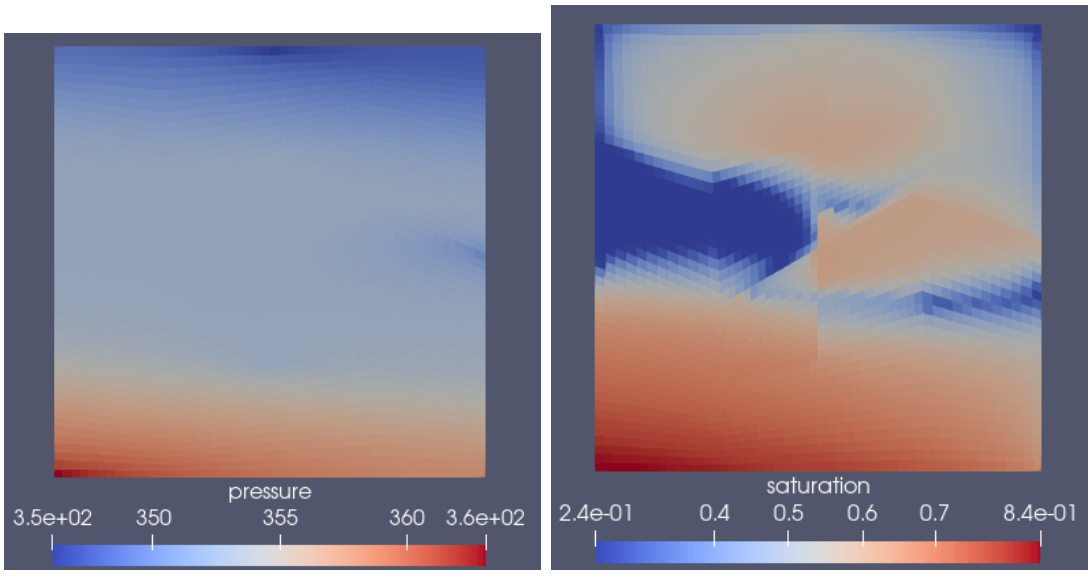
For the barrier case, we see a strange profile, as our fracture and direction of highest anisotropy (after rotation of permeability tensor) are not exactly perpendicular. Hence the saturation front finds the communication path from the right, which is closer to the injector than the left end of the barrier.

### Intersecting Fractures




(a) Pressure (b) Saturation

Figure 6.21: Intersecting fracture case-1 (a) Pressure (b) Saturation, after 300 days




(a) Pressure (b) Saturation

Figure 6.22: Intersecting fracture case-2 (a) Pressure (b) Saturation, after 300 days

There are interesting profiles for intersecting fractures too. After fluid reaches the start of one fracture, the flow is subsequently transferred through the other fracture and hence reaches both the wells.

## 6.2 Mechanical Deformation

In this section, we develop and evaluate the performance of the NTPSA discretization technique. Apart from this, we also use the homogenization function in conjunction with the Linear MPSA method to check if we have any improvements to the original solutions.

As previously discussed in the Introduction chapter, our main motivation to develop a nonlinear scheme for the mechanical problem is to reduce or even eliminate the oscillations observed in the Linear MPSA method, which uses least-squares during the gradient reconstruction method. But first, let us look at some tests related to the convergence of the problem.

### 6.2.1 Convergence Study

In this part we tried to prove the convergence of our discretization technique by using the homogeneous anisotropic tensor test suggested in (Terekhov and Tchelepi, 2020). The reference solution of the problem in cubic domain $\Omega = [0,1]^3$ is

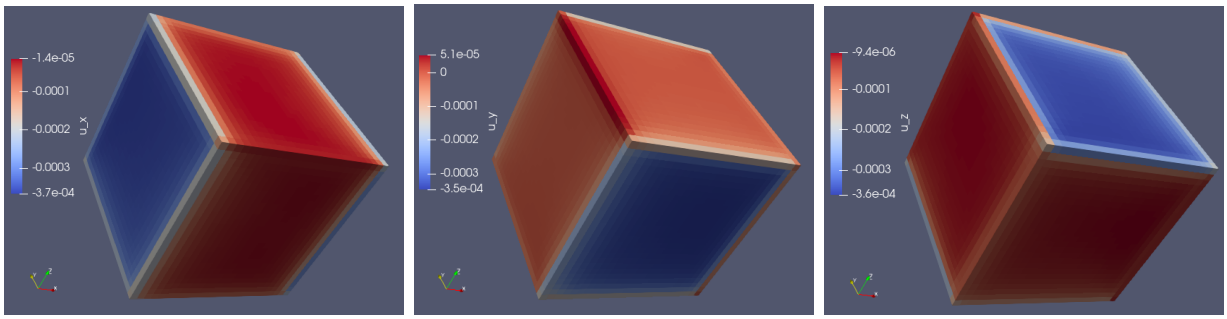
$$\begin{aligned} u(x,y,z) &= \left(x - \frac{1}{2}\right)^2 - y - z, \\ v(x,y,z) &= \left(y - \frac{1}{2}\right)^2 - x - z, \\ w(x,y,z) &= \left(z - \frac{1}{2}\right)^2 - x - y, \end{aligned} \tag{6.10}$$

and the $6 \times 6$ stiffness tensor is represented as

$$E = \begin{bmatrix} 1.323 & 0.0726 & 0.263 & 0.108 & -0.08 & -0.239 \\ 0.0726 & 1.276 & -0.318 & 0.383 & 0.108 & 0.501 \\ 0.263 & -0.318 & 0.943 & -0.183 & 0.146 & 0.182 \\ 0.108 & 0.383 & -0.183 & 1.517 & -0.0127 & -0.304 \\ -0.08 & 0.108 & 0.146 & -0.0127 & 1.209 & -0.326 \\ -0.239 & 0.501 & 0.182 & -0.304 & -0.326 & 1.373 \end{bmatrix}. \tag{6.11}$$

As discussed in the convergence study in fluid flow, we derive the right-hand side of the problem, which contains source/sink terms, by using the formula $f = -\nabla.E\epsilon$, and use Dirichlet boundary to calculate the displacement at the boundaries of the problem. The solution of all three components of displacement can be seen in Figure 6.23.

We perform this numerical solution on 6 such cubic meshes, which had $3^3, 5^3, 10^3, 14^3, 20^3, 26^3$ number of cells respectively. In Figure 6.24 we can see the convergence rate is close to 2,




(a) $u_x$ (b) $u_y$ (c) $u_z$

Figure 6.23: Analytical solution of displacements (a) $u_x$ (b) $u_y$ (c) $u_z$

Table 6.7: Convergence study for Average MPSA

| **Cells** | **Error** | **Log Error** | Convergence Rate |
|---|---|---|---|
| 27 | 0.02066 | -1.685 | 0 |
| 125 | 0.00764 | -2.117 | 1.949 |
| 1000 | 0.00193 | -2.714 | 1.982 |
| 2744 | 0.00099 | -3.006 | 1.999 |
| 8000 | 0.00048 | -3.316 | 2.001 |
| 17576 | 0.00029 | -3.544 | 2.002 |

as also calculated in Table 6.7. However, we are not able to reproduce the same for the NTPSA discretizer, which is something we have to investigate in our future work. It can be related to how the NTPSA we formulated generally has a slow convergence in other cases, which we will address below.

### 6.2.2 Compression and Shear

In this part, we take the following problem previously discussed in (A. Novikov, 2021). For pure mechanics, as we are not considering any fracture formation and slip along the fracture, we will analyze the traction profile on a specific line or feature in our grid domain. The setup of the problem is formulated in Figure 6.25. For the above problem, we consider 4 different types of grids namely structured hexahedrons, coarse adaptive wedges, coarse adaptive wedges with refined mesh at fracture, and fine adaptive wedges. But for the analysis in Figures 6.26 and 6.27, we consider coarse adaptive wedges with 1274 control volumes.

Here we see that normal traction has a higher deviation and is noisier than tangential traction. Overall the nonlinear scheme produced a higher deviation, one of the reasons could be that we stop the process when the value of residual reaches $10^{-10}$ whereas the residual for

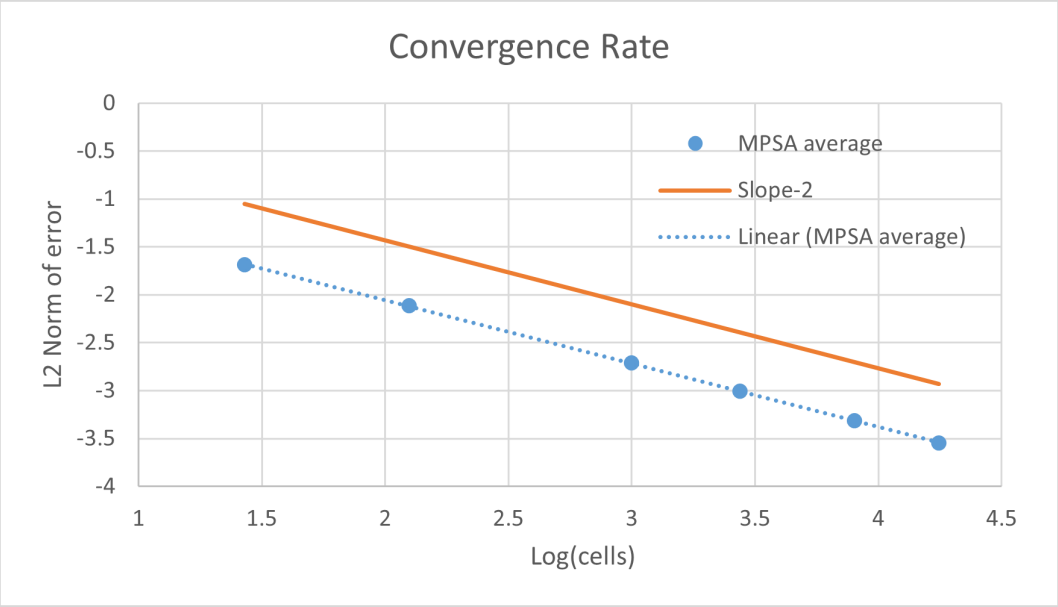



Figure 6.24: Convergence of Average MPSA method on structured meshes with homogeneous anisotropic stiffness tensor

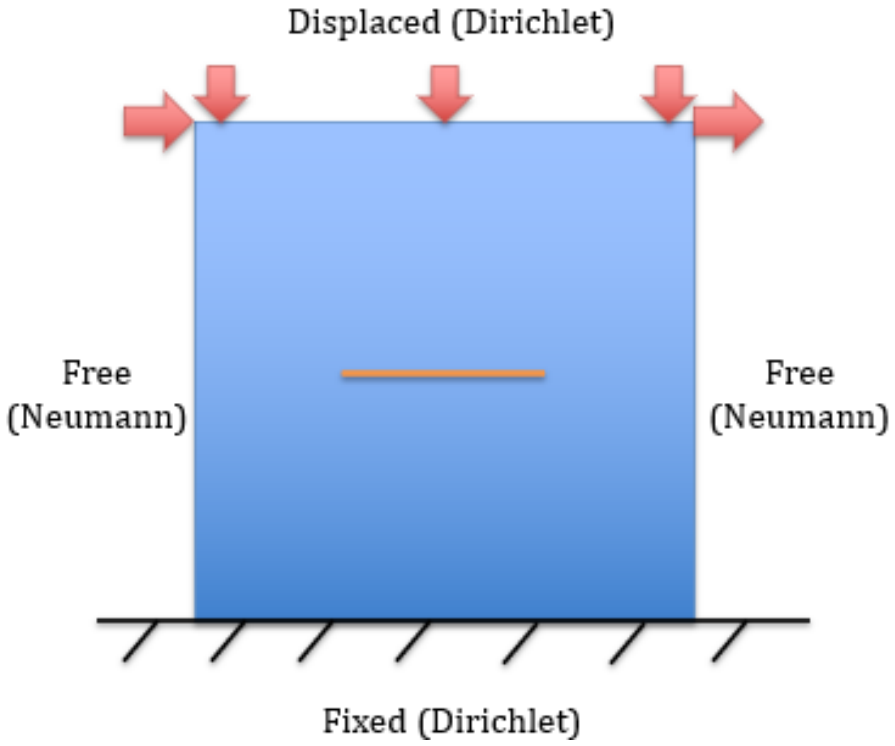
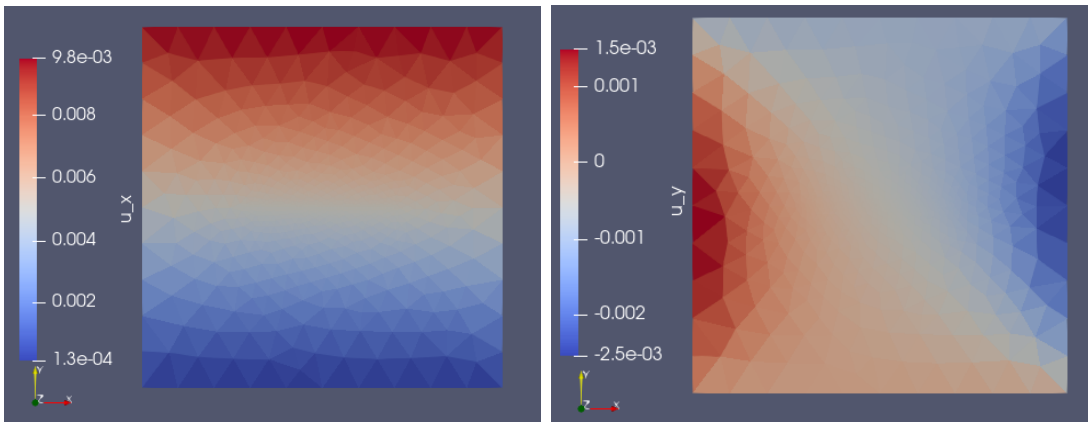
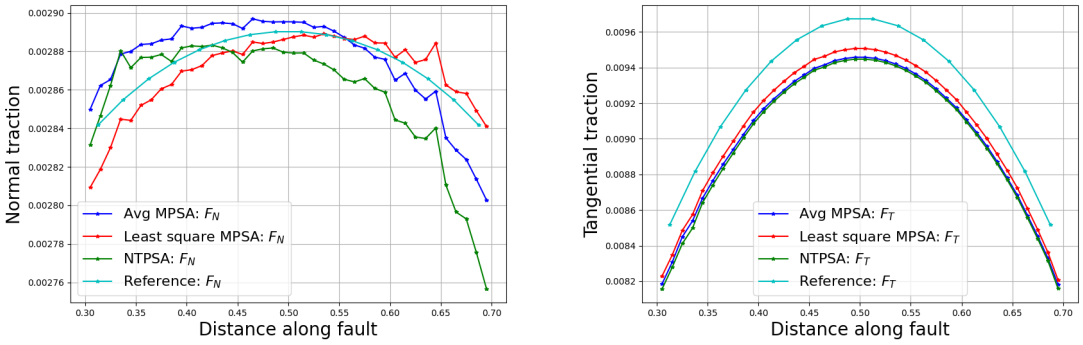



Figure 6.25: 3D domain (extruded in z-axis) with the above mentioned boundary conditions, analyzed plane is indicated by orange.




(a) $u_x$ (b) $u_y$

Figure 6.26: Compression and shear displacements (a) $u_x$ (b) $u_y$, on coarse adaptive wedges




(a) $F_n$ (b) $F_t$

Figure 6.27: Compression and shear (a) Normal (b) Tangential, components of traction along the fracture

linear problem drops to $10^{-17}$. The number of nonlinear iterations restricted us to converge the problem to this residual ($10^{-17}$). To reach the residual value of $10^{-10}$, NTPSA for this case with 1274 cells took more than 17,000 iterations. The residual drop can be seen in Figure 6.28.

### 6.2.3 Residual Maps

As we analyzed in the previous section, NTPSA takes too many iterations and seems impractical when we have a large enough domain. Both numerical and analytical Jacobians are considered, and both gives us the same response i.e. too many iterations to converge. To understand what might be the cause for poor convergence, we proceed to a plot the residual map to check whether the drop of residual is smooth or not in our solution space. To assess

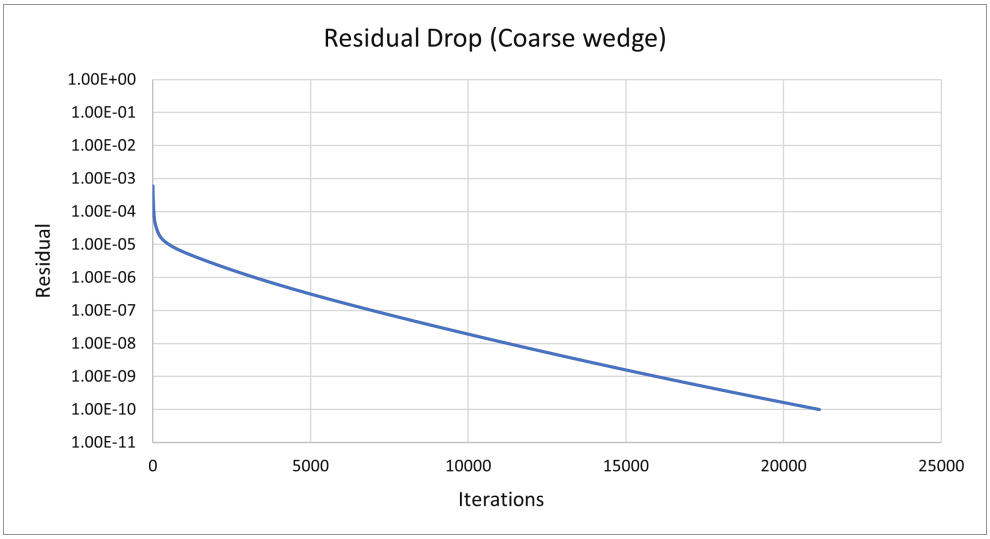
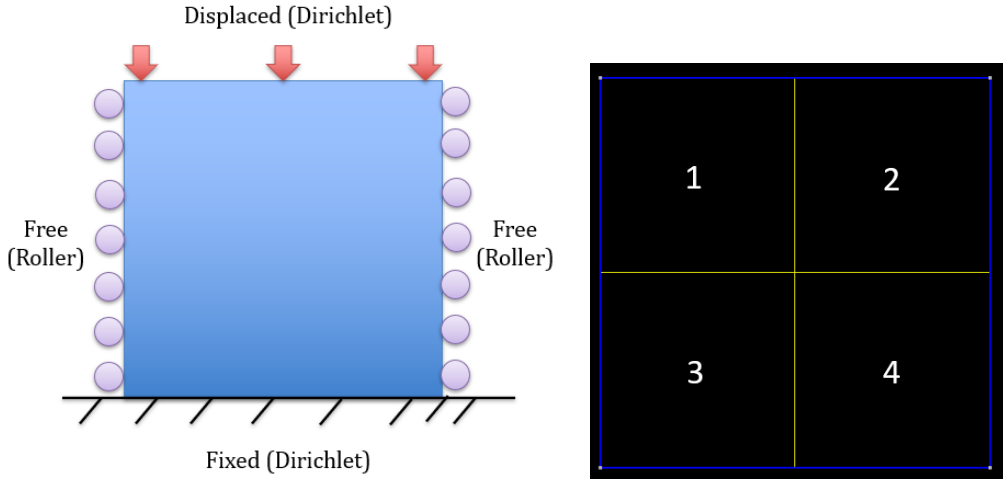



Figure 6.28: Magnitude of Residual with each iteration for coarse wedge problem

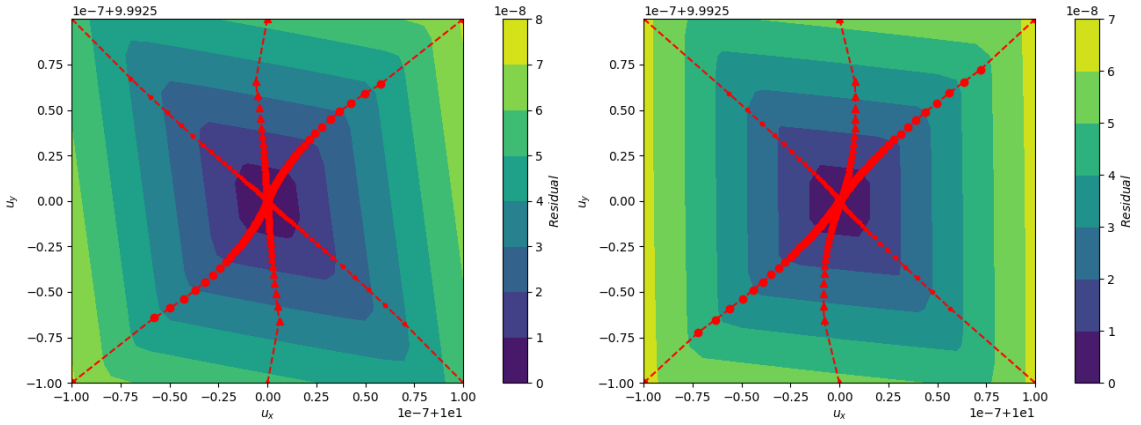
the residual map, we take a 4 cell problem (such that we have 4 internal interfaces which are discretized by NTPSA) and use linear compaction from the top. It is an extruded problem with 4 roller boundaries (x and z planes), so we obtain the solution for $u_y$ as $u_x$ and $u_z$ remain constant.




(a) Problem (b) Mesh

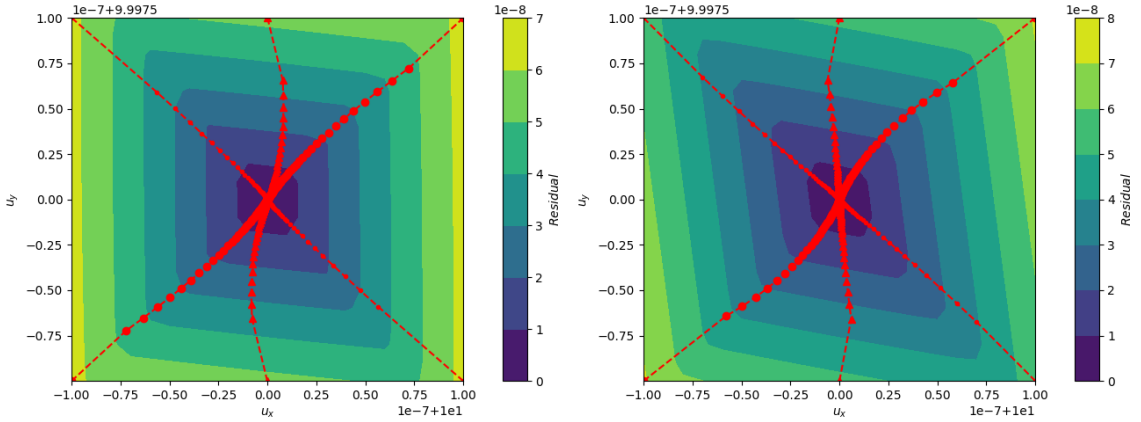
Figure 6.29: Residual map formulation (a) Problem setup (b) Used Mesh, after 300 days

We extract the analytical solution of this problem and perturb this condition to observe the magnitude of change in residual. As indicated in Figure 6.29(b), we plot residual maps for all 4 cells in the order of their numbering in Figures 6.30 and 6.31. As indicated in Figures 6.30 and 6.31, the residual map is smooth and there are no sharp perturbations in the path of residual drop (indicated by red).




(a) Cell 1 (b) Cell 2

Figure 6.30: Residual Maps for (a) cell 1 (b) cell 2, trajectories of 6 initial perturbations is indicated by red lines




(a) Cell 3 (b) Cell 4

Figure 6.31: Residual Maps for (a) cell 3 (b) cell 4, trajectories of 6 initial perturbations is indicated by red lines

We also used several resolutions of grids and see how the number of iterations changes with grid size and the behavior of weight function with the number of iterations. Two types of problems are considered:

- Linear problem with one displacement, Dirichlet on y-planes and Roller on x-planes, and
- Linear problem with two displacements, Dirichlet on y-planes and Neumann on x-planes.

Meshes are chosen such that the number of cells ranged from $2^2$ to $10^2$. In the problem with roller boundaries on the x-plane, only 1 component of displacement have been changed, that

is $u_y$, as we have load in the form of Dirichlet boundary here and both the x-bounding planes are constrained such that there is no normal displacement. Whereas in the problem with Neumann boundaries, we see displacement in the $u_x$ direction due to the Poisson's ratio i.e. the material is free to move in the x-direction. The results are displayed in Figure 6.32.

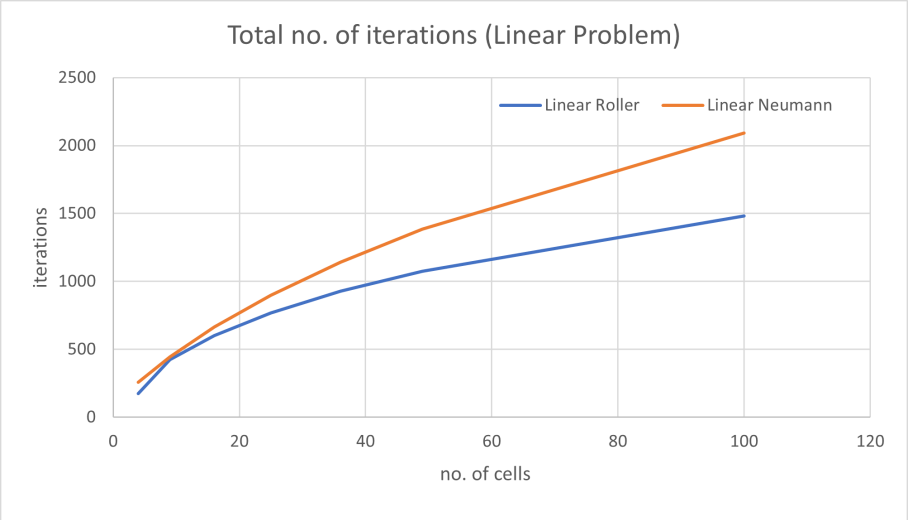
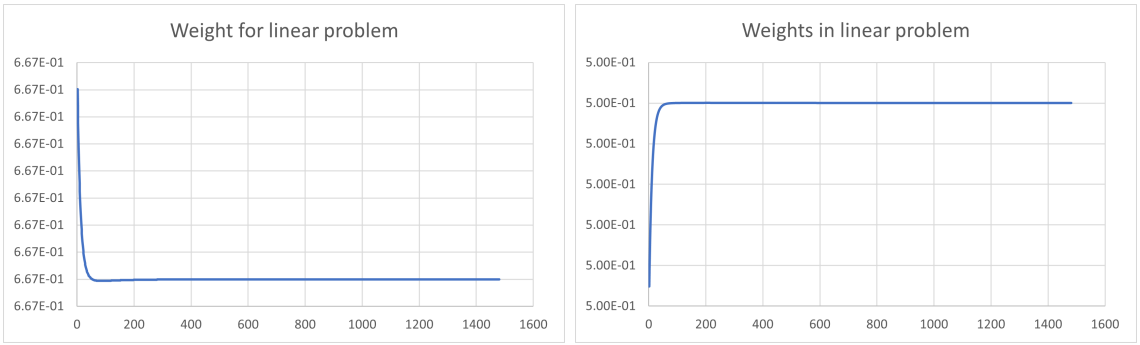



Figure 6.32: No. of iteration vs total number of cell for both linear problems

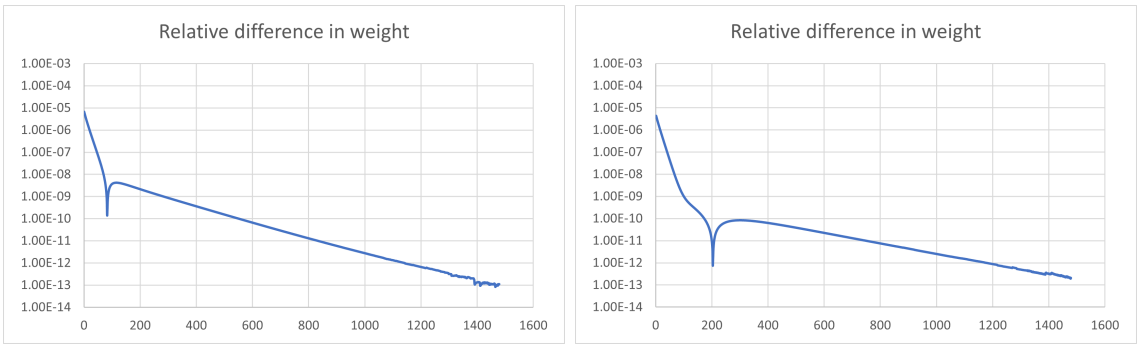
Weights are also plotted against number of iterations for a specific cell in the $10^2$ domain. Both weights related to $F_x$ and $F_y$ are plotted against the number of iterations. In Figure 6.33, we observe the value of weight on every iteration, for better understanding the relative difference between the weights, i.e. $|\mu_i - \mu_{i-1}|$ have been plotted with number of iterations in Figure 6.34. Just like the drop of residual, the rate of change of weights is almost linear. The sharp kink in Figure 6.34 indicates the moment when weights start to increase from when it was in decline or vice-versa. So, this raised a concern that NTPSA




(a) $F_x$ (b) $F_y$

Figure 6.33: Value of weights for (a) $F_x$ (b) $F_y$, components of traction in top cell

might take many iterations even though the problem is fairly linear. More investigation in the form of nonlinear convex combination is shown in the next section.

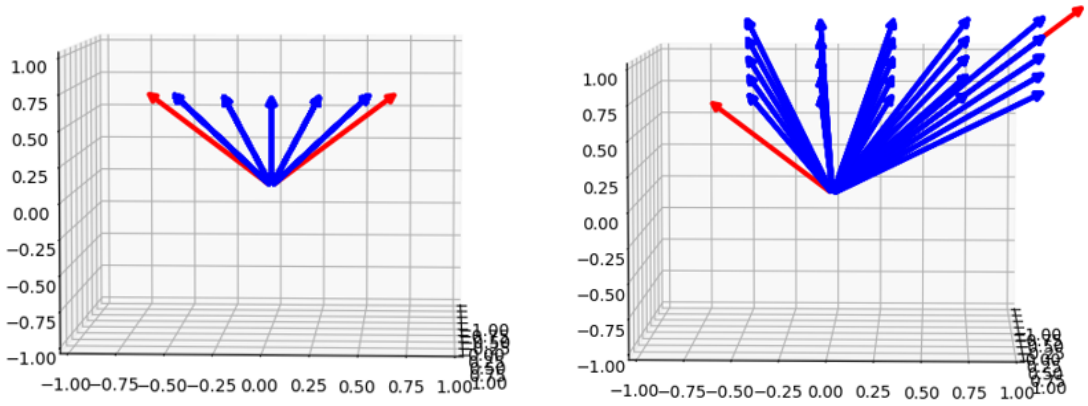



(a) $F_x$ (b) $F_y$

Figure 6.34: relative difference of weights for (a) $F_x$ (b) $F_y$, components of traction in top cell (for convenience)

### 6.2.4 Convex Combination of Vectors

During the convex combination described in chapter-5, we assigned 3 pairs of weights, one for each component of traction. The plots are indicated in Figures 6.34 and 6.33 for a linear problem with roller condition. In nonlinear flux approximation, we use the convex combination to find the intermediate flux that is the magnitude of flux $q^\sigma$ which lies in between flux from both positive and negative cells i.e. $q_1^\sigma$ and $q_2^\sigma$ across the interface. But because we are performing this procedure on each component of traction individually, we may not preserve the magnitude of total traction on the interface. In order to fully understand this phenomenon, we plot two traction vectors which are analogous to $\mathbf{F}_{\delta 1}$ and $\mathbf{F}_{\delta 2}$ with several combinations of weights for all three components.




(a) Equal Magnitude (b) Unequal Magnitude

Figure 6.35: Tangential traction vectors (a) Equal (b) Unequal, magnitudes with several combinations of weights for each component

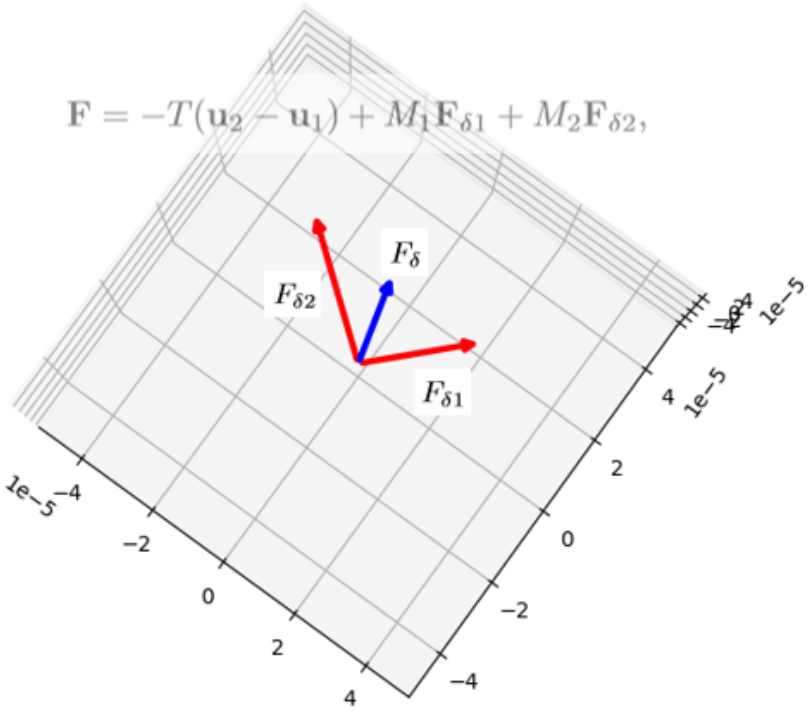



Figure 6.36: An actual convex combination observed during one of the iterations of linear problem

### 6.2.5 Homogenization for Linear MPSA

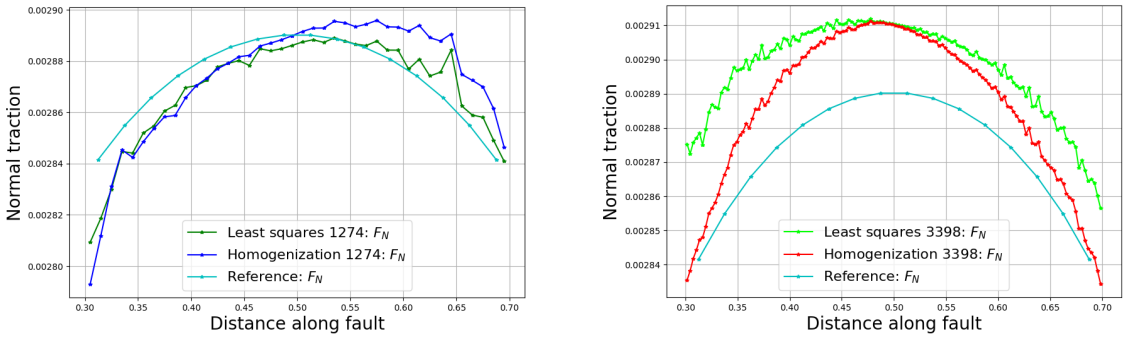
Now, as the NTPSA results did not smooth the traction profile, an alternative option is explored where we use the homogenization function directly in the least-squares reconstruction. This means, when we are performing gradient reconstruction using least-squares, we take an extended stencil that not only include immediate neighbors but also their neighbors. To make this study grid independent, we proceed to try the same three types of grids (exclude hexahedrons) as discussed in section 6.2.2. Here in Table 6.8, the parameters $L_c$ and $L_c\,frac$

Table 6.8: Convergence study for Average MPSA

| **Mesh type** | $L_c$ | $L_c\,frac$ | **Cells** | **Least squares** | **Homogenization** |
|---|---|---|---|---|---|
| Coarse adaptive | 0.1 | 0.01 | 1274 | Noisy | Noisy |
| Medium adaptive | 0.1 | 0.003 | 3398 | Noisy | Improved |
| Fine adaptive | 0.03 | 0.003 | 12874 | Improved | Improved |

are used to construct the mesh. The lower value of $L_c$ indicates a higher general resolution of the whole mesh and vice-versa. The lower value of $L_c\ frac$ indicates higher resolution only at the vicinity of fracture and so more adaptivity.

Note that the reference solution in Figure 6.37, is obtained from a structured mesh with 1600 structured hexahedrons (cubes). So, we refined the mesh even further such that both exterior and interior are fine. As shown in Figure 6.38, the solution is more closer to the




(a) 1274 cells (b) 3398 cells

Figure 6.37: Magnitude of normal traction for mesh with (a) 1274 (b) 3398 cells and comparison with reference

value of reference generated from structured mesh. We also observe the tangential traction profile is smooth, but shifts from the reference profile i.e the magnitude is shifted.

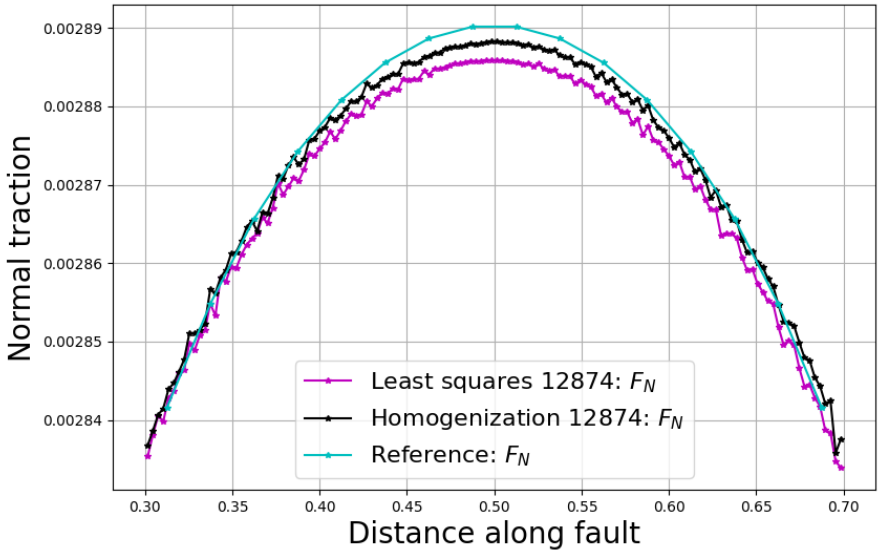



Figure 6.38: Magnitude of normal traction for mesh with 12874 cells

Next, we analyze the extent of homogenization, which will show how increasing the level of homogenization as discussed in chapter-4 and 5 will impact the solution profile. We consider 4 levels of homogenization after the least-squares stencil. Figure 6.39 shows the response we get as we increase the level of homogenization. We can see for the mesh which has 3398 cells, the initial least-square solution is quite noisy, but increasing the number of cells in the stencil gives us a better profile, although it never actually comes as close as the refined

solution in Figure 6.38.

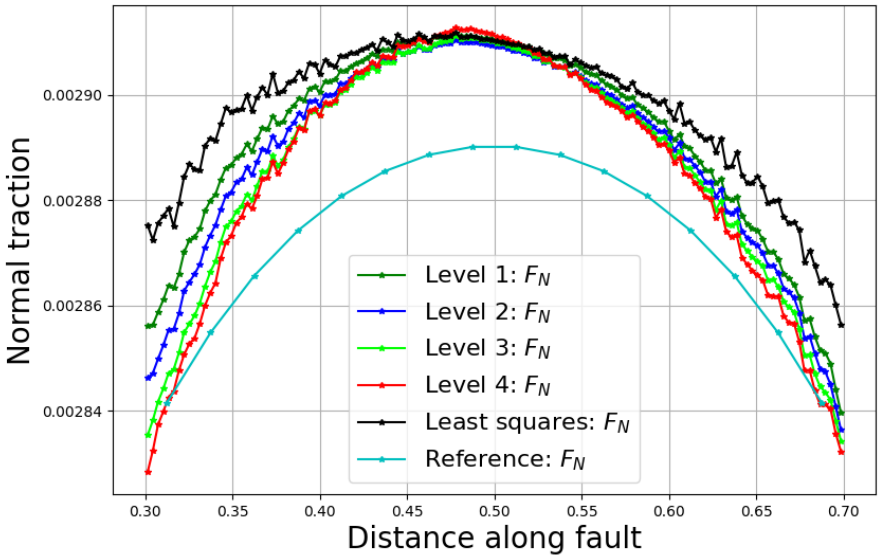



Figure 6.39: Magnitude of normal traction for different levels of homogenization

Overall, homogenization behaves as expected i.e. considering several other connections in our vicinity apart from neighbors will give us a better understanding of pressure gradient on specific cell-face pairs.

# Chapter 7

# Conclusions and Recommendations

In this chapter, we discuss our results in some details and give recommendations on how these results can be improved.

## 7.1 Conclusion

Primarily, we implemented the following schemes in this project:

- Fluid Flow
  a) Development of nonlinear TPFA discretization on subsurface fluid flow
  b) Integrating the nonlinear discretization in matrix (or reservoir blocks) to fractures in the porous media.
  c) Proving that our NTPFA scheme is monotone and positivity preserving.
  d) Consistency and convergence of the NTPFA method is also demonstrated in mild and high anisotropic cases, even when our grid is unstructured.
  e) Different gradient reconstruction techniques for NTPFA is also explored.
- Elastic Mechanics
  a) Development of nonlinear Two Point Stress Approximation in Finite Volume framework, following the earlier works in Multi-Point Stress Approximation (MPSA) methods.
  b) Developing the Homogenization function and using it for MPSA such that we can include more number of cells for computing traction along interface.
  c) Consistency and convergence of Avg MPSA scheme are proven.

d) Residual maps for NTPSA are plotted to help us investigate the nonlinear convergence.

Nonlinear discretization methods in subsurface flow simulation are known for their robustness by not only approximating an accurate form of flux (taking anisotropy and heterogeneity) across the interface, but also giving a monotone solution with no oscillations which are usually observed in MPFA. The flux approximation specifically developed in this project can reconstruct the pressure gradient in most distorted meshes with severe anisotropy by using the homogenization technique, which helps us to find a positive basis considering the minimum number of cells in the process.

Usually, in most of the reservoir simulation problems, we do not have extreme cases of anisotropy or even a Corner Point Geometry (CPG) grid is constructed in such a way that anisotropy is incorporated as heterogeneities such as, a fluvial channel or sand lenses. In our perspective, NTPFA would be most effective in subsurface reservoir simulation when we upscale a CPG grid that has a large number of discrete fractures (length order of meters) which usually give rise to extreme cases of anisotropy. In this case, an NTPFA simulator will remain monotone and can accurately approximate pressure, which is then used in saturation transport or compositional changes in the porous media.

We have also incorporated Discret Fracture Model (DFM) in our discretizer, which helpa us simulate a fractured domain. But the main reason for incorporating a DFM framework into NTPFA is to represent multi-scale heterogeneities since we expect the small-scale fractures to be up-scaled and represented as anisotropy in our domain.

After understanding and developing a general NTPFA discretizer, we used that knowledge to investigate a nonlinear discretization method for linear elastic mechanics problems based on the FV framework. This will provide us with a good understanding of how this method works in mechanics such that we can move forward to create a coupled poro-mechanical problem that we can solve in a fully coupled manner.

Nonetheless, in the analysis we performed on the mechanical problem, the NTPSA discretizer was not only less efficient with the number of nonlinear iterations, but also did not give a smooth solution for traction profile which was hypothesized during our motivation. After much investigation into residual maps and weight functions, the only reason we have for this poor performance is that traction flux might not be approximated in an accurate way as we are dealing with two vectors ($\mathbf{F}_{\delta 1}, \mathbf{F}_{\delta 2}$) now instead of scalars where magnitude is weighted. Residual maps look fairly normal, and trajectory taken by perturbation in the specific cell reaches the convergence smoothly as shown in Figure 6.30 and 6.31.

## 7.2 Recommendations

For the nonlinear flux approximation, we have to make sure that our framework performs well with minimal nonlinear iterations when using different forms of physics such as geothermal or compositional, which are in an environment with an immense amount of anisotropy or heterogeneity. We were focused on developing the technique for the mechanical problem and leave this part for future analysis. Also, an interesting study would be to take a CPG grid that has discrete fractures incorporated into it for geothermal or $\mathbf{CO_2}$ storage application, upscale it, and then use linear TPFA, MPFA, and NTPFA to observe the difference in the shape of the fluid front, breakthrough time and total energy extracted/stored during the simulation. This might give us more insights on how monotonicity and consistency of a simulator are important factors, and how adversely our predictions can be influenced by ignoring them.

For the mechanics part, it would be fair to say the development is at a nascent stage. Although we might have not produced an efficient way of solving the problem, it provides us with the knowledge on what can be the possible cause of inefficiency - the convex combination discussed by the end of Section 6.2. It might also be true that we need a different form of convex combination to approximate an accurate traction flux across an interface.

Also, it is safe to assume that the final goal in research dedicated to reservoir/subsurface simulation is to have a fully coupled framework that can take several types of grid/structures with complex permeability distribution and simulate different phenomena like multiphase, thermal and chemical flow with mechanical effects. The present work takes an initial step into developing of such framework to the fully coupled problem and solve it in a more robust that the solution remains monotone.

# Bibliography

(2003). *An Efficient Discrete Fracture Model Applicable for General Purpose Reservoir Simulators*, volume All Days of *SPE Reservoir Simulation Conference*. SPE-79699-MS.

A. Novikov, D.V. Voskov, M. K. H. H. J. J. (2021). A collocated finite volume scheme for high-performance simulation of induced seismicity in geo-energy applications. *SPE*.

Aavatsmark, I. (2002). An introduction to multipoint flux approximations for quadrilateral grids. *Computational Geosciences*, 6:405–432.

Agelas, L., Eymard, R., and Herbin, R. (2009). A nine-point finite volume scheme for the simulation of diffusion in heterogeneous media. *Comptes Rendus Mathematique*, 347(11):673–676.

Bertolazzi, E. and Manzini, G. (2004). A cell-centered second-order accurate finite volume method for convection–diffusion problems on unstructured meshes. *Mathematical Models and Methods in Applied Sciences*, 14(08):1235–1260.

Cancès, C., Cathala, M., and Le Potier, C. (2013). Monotone corrections for generic cell-centered finite volume approximations of anisotropic diffusion equations. *Numer. Math.*, 125(3):387–417.

Danilov, A. A. and Vassilevski, Y. V. (2009). A monotone nonlinear finite volume method for diffusion equations on conformal polyhedral meshes. 24(3):207–227.

Eirik Keilegavlen, I. A. (2011). Monotonicity for mpfa methods on triangular grids. *Comput Geosci*, 15:3–16.

Gao, Z.-M. and Wu, J. (2015). A second-order positivity-preserving finite volume scheme for diffusion equations on general meshes. *SIAM Journal on Scientific Computing*, 37:420–438.

I. Aavatsmark, G. T. Eigestad, B. T. M. J. M. N. (2008). A compact multipoint flux approximation method with improved robustness. *Numerical Methods for Partial Differential Equations*, pages 1329–1360.

K. Aziz, A. S. (1979). *Petroleum Reservoir Simulation*. Applied Science Publishers LTD.

Kapyrin, I. (2007). A family of monotone methods for the numerical solution of three-dimensional diffusion problems on unstructured tetrahedral meshes. *Doklady Mathematics*, 76(2):734–738.

Keilegavlen, E. and Nordbotten, J. M. (2017). Finite volume methods for elasticity with weak symmetry. *International Journal for Numerical Methods in Engineering*, 112(8):939–962.

Kershaw, D. S. (1981). Differencing of the diffusion equation in lagrangian hydrodynamic codes. *Journal of Computational Physics*, 39(2):375–395.

Kolditz, O. (2002). Finite volume method. *Computational Methods in Environmental Fluid Mechanics*.

Kuzmin, D. (2010). *A Guide to Numerical Methods for Transport Equations*.

Le Potier, C. (2005). Schéma volumes finis monotone pour des opérateurs de diffusion fortement anisotropes sur des maillages de triangles non structurés. *Comptes Rendus Mathematique*, 341(12):787–792.

Lipnikov, K., Svyatskiy, D., and Vassilevski, Y. (2009). Interpolation-free monotone finite volume method for diffusion equations on polygonal meshes. *Journal of Computational Physics*, 228(3):703–716.

Nikitin, K., Terekhov, K., and Vassilevski, Y. (2013). A monotone nonlinear finite volume method for diffusion equations and multiphase flows. *Computational Geosciences*, 18:311–324.

Nordbotten, J.M., A. I. . E. G. (2007). Monotonicity of control volume methods. *Numer. Math.*, 106:255–288.

Nordbotten, J. M. (2014). Cell-centered finite volume discretizations for deformable porous media. *International Journal for Numerical Methods in Engineering*, 100(6):399–418.

Schneider, M. (2019). *Nonlinear finite volume schemes for complex flow processes and challenging grids*.

Schneider, M., Flemisch, B., Helmig, R., Terekhov, K., and Tchelepi, H. (2018). Monotone nonlinear finite-volume method for challenging grids. *Computational Geosciences*, 22.

Sheng, Z. and Yuan, G. (2011). The finite volume scheme preserving extremum principle for diffusion equations on polygonal meshes. *J. Comput. Phys.*, 230(7):2588–2604.

Terekhov, K., Mallison, B., and Tchelepi, H. (2016). Cell-centered nonlinear finite-volume methods for the heterogeneous anisotropic diffusion problem. *Journal of Computational Physics*, 330.

Terekhov, K. M. and Tchelepi, H. A. (2020). Cell-centered finite-volume method for elastic deformation of heterogeneous media with full-tensor properties. *Journal of Computational and Applied Mathematics*, 364:112331.

Tuković, Z., Ivanković, A., and Karač, A. (2013). Finite-volume stress analysis in multi-material linear elastic body. *International Journal for Numerical Methods in Engineering*, 93(4):400–419.

Vassilevski, Y., Terekhov, K., Nikitin, K., and Kapyrin, I. (2020). *Parallel Finite Volume Computation on General Meshes*.

Voskov, D. (2017). Operator-based linearization approach for modeling of multiphase multi-component flow in porous media. *Journal of Computational Physics*, 337.

Wang, Y., Voskov, D., Khait, M., and Bruhn, D. (2020). An efficient numerical simulator for geothermal simulation: A benchmark study. *Applied Energy*, 264:114693.

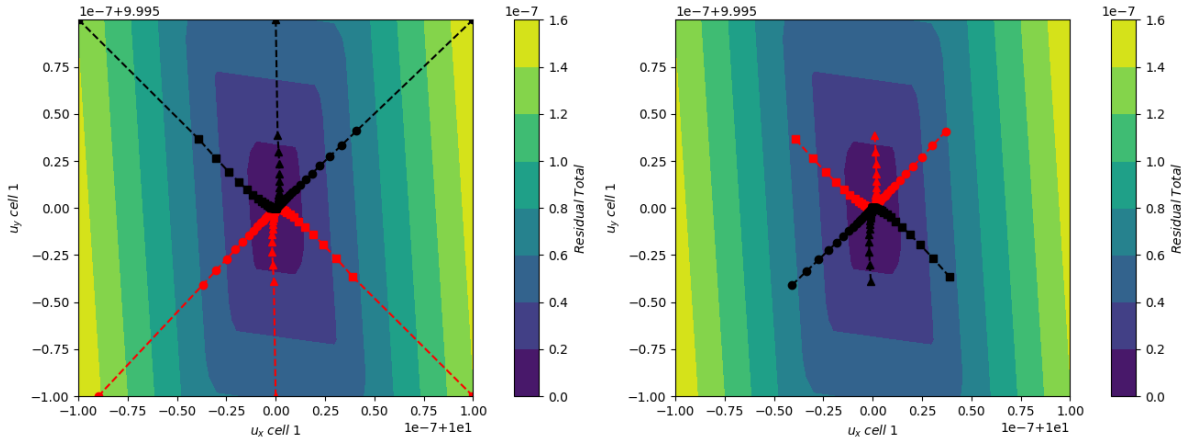
Yuan, G. and Sheng, Z. (2008). Monotone finite volume schemes for diffusion equations on polygonal meshes. *Journal of Computational Physics*, 227(12):6288–6312.

Zhang, W. and Al Kobaisi, M. (2020). Cell-Centered Nonlinear Finite-Volume Methods With Improved Robustness. *SPE Journal*, 25(01):288–309.

# Appendix A

# Residual Test

Here residual in a two-cell problem is plotted, with a trace path of 6 initial conditions. At the beginning of the simulation, we assume an analytical solution in both cells. We only Perturb the value of $u_x$ and $u_y$ in cell 1, but plot the trace path of $u_x$ and $u_y$ values in both cell 1 and 2 as shown in Figure A.1.




(a) cell 1

(b) cell 2

Figure A.1: Residual Map for two cell problem with perturbation in cell 1 (a) trace path of $u_x$ and $u_y$ in cell 1 (b) trace path of $u_x$ and $u_y$ in cell 2

As we perturb cell-1, we can see the change propagated to cell-2 from its initial condition. And we can also see the perturbation is inverted in cell 2 (follow red square in both plots for clear understanding). This would happen in the nonlinear convex combination stage, especially when we compute weights to derive the equation for traction. Due to this, our overall residual will not drop as expected.