Advection Upscaling for Heat Transport in Fractured Geothermal Reservoir

Master of Science Thesis
in Applied and Computational Mathematics

Marija Ninic
Departament of Mathematics
University of Bergen

June 1, 2018

Acknowledgements

I would like to thank my supervisors Inga Berre and Eirik Keilegavlen. I do much appreciate their time spent to advise me and give suggestions for this thesis. I would also like to thank Anna Nissen. Her paper "Heterogeneity Preserving Upscaling for Heat Transport in Fractured Geothermal Reservoirs" was inspiration for my thesis. I would like to thank to my husband and son for their love and support.

Abstract

The main objective in this thesis is to give numerical solution for advection term of heat transport equation in fractured geothermal reservoir. To get better insight in the fracture influence on the processes we will present all numerical results for domain with and without fracture field. Fracture field is modeled explicitly using discrete fracture model. Advection term of heat transport equation is discretized in space using upwind scheme for time discretization we use implicit method: Euler's backward scheme. For advection term upscaling we use the known flux values that we get from numerical solution of pressure equation. To get comutationally more efficient transport solver we will use upscaling and upgridding. We will use flow-based indicators for upgridding fine scale grid. In the case of fractured domain beside standard flow-based indicators: permeability, velocity and time of flight we will use distance to the nearest fracture and the combination of distance and time of flight as indicators. At the end we will compare results on different coarse grids with result on fine scale grid.

Contents

Chapter 1

Introduction

In this thesis we will present a numerical solution method for fluid flow and heat transport model in fractured geothermal reservoirs. The reason we are interested in studying these processes is related to the extraction of the geothermal energy. Geothermal reservoirs are renewable energy resource with wide potential. Most of the geothermal reservoirs occurs in fractured porous rocks at high temperatures [4]. Thus, it is important to consider the fractures when modeling a geothermal reservoir. The characteristic of the physical processes are highly dominated by the occurrence of the fractured field [22].

When we do reservoir simulations geothermal reservoir is modeled as a $3D$ grid with cells that contains geological properties of the reservoir while fractures are represented explicitly using discrete fracture model. Simulations of these structurally dominated processes are very large with a lot of details to be resolved in the discrete model [22]. We need to develop effective discretization strategies to limit computational effort and on the other hand to maintain good accuracy.

We will examine both fluid flow through the reservoir and the heat transport process. The fluid flow is modeled using the pressure equation. The heat transport is modeled using an advection-conduction equation for temperature and in this thesis, we are focused just on the advection term of the equation. Pressure equation is solved once on the fine scale grid to obtain pressure and flux values and apply these parameters to solve advection-conduction equation [27]. Advection term is solved both on the fine scale grid and on the coarse scale grid. For both equations we will consider case without fractures and with presence of fractures. Coarse scale grid was constructed using amalgamation of fine grid cells and using different coarse scale partitions in order to obtain most efficient computational grid. We will analyze which grid is most efficient for our test cases.

1.1 Geothermal Reservoirs

Geothermal energy consists of the thermal energy contained in the Earth subsurface. Geothermal energy is important because it can provide both electrical power and heat. The main advantages of geothermal energy are its availability, continuity and renew-ability. By availability we considered at geothermal energy is independent of weather conditions [21]. Continuity means that it does not need any backup source [21]. In 2016 world geothermal power production was 13.8 GW and it is in constant growth in 2021 production is expected to be 23 GW [9], [7]. Overall the geothermal share in global power generation is about 0.3%, however in some countries it is significant energy source such as Kenya with (44%) and Iceland with (27%) of power [9].

A naturally occurring geothermal system is known as hydrothermal system. Hydrothermal resources usage depending on the temperature and how deep they are [11]. An enhanced geothermal system (EGS) occurs in reservoirs with high temperatures but with lack of fractures, pores and water [21], [8]. In (EGS) cold water is injected into the hot rock and fracturing rock sufficiently to enable water to flow [4], [8], [2]. The water flow through the permeable pathways and transport heat via production well [27], [8], [2]. While flow through these permeable pathways water absorbed heat form the surrounding hot rock in the process known as conduction [4], [8]. Process of heat transport with water flow is convection. At the surface the fluid passes through the power plant where the electricity is generated (Figure 1.1). Geothermal reservoirs are typically situated in igneous rock where the permeability of the reservoir is dominated by the presence of the discrete fractures [22]. Flow and transport processes are highly dominated by these structural heterogeneities. In the following chapters the processes of fluid flow and heat transfer through the porous material will be present in more detail.

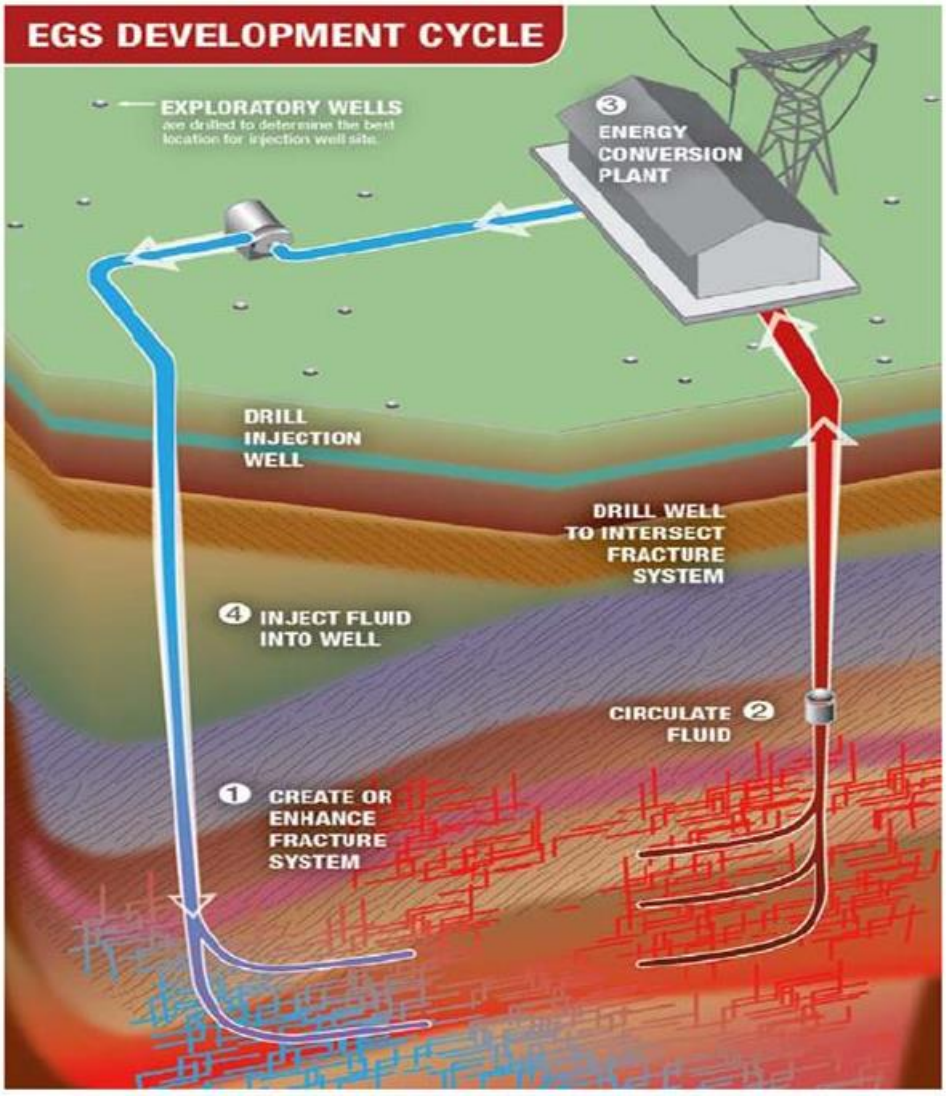


Figure 1.1: Enhanced geothermal system [3]

1.2 Outline

Our goal is to present $3D$ simulations of advection term of heat transport equation in fractured geothermal reservoir and to give a closer look at process within fractures. The course we have chosen is as follows:

In chapter 2 we define fluid and rock parameters together with the equations which describe fluid flow in porous media. This chapter will give us a brief introduction in the fracture flow in porous media.

In chapter 3 we present heat transfer in porous media. The chapter contains the governing equations for advection-conduction equation which we use later on in numerical computations.

In chapter 4 we will present numerical methods we used to solve partial differential equations from previous two chapters. Pressure equation will be discretized in space while advection part of energy equation will be discretized both in space and time. $TPFA$ method used for space discretization will be explained. Advection equation will be discretized in space with upwind scheme and in time using implicit scheme. We will give explanation of fractures discretization. The implementation of source term in numerical schemes are given here.

In chapter 5 we will construct coarse grid for our simulations. Upscaling techniques by standard arithmetic mean for rock properties is described. The coarse scale advection term is upscaled using the sums of fine scale fluxes [22]. We will discuss upgridding procedures to best represent that we will further use in the simulation of advection term of heat transport equation. For coarse grid construction we will use flow based nonuniform coarsening algorithm with different flow indicators for grid construction such as time of flight, velocity and permeability. To examine fractures influence on heat transport we will use the distance to the nearby fractures as coarsening indicator. Fine scale cells in the matrix are classified by their Euclidean distance to the nearest fracture [22].

In chapter 6 we will present and analyze numerical experiments. We will implement pressure equation on homogeneous media on fine grid scale with and without fractures.

Numerical experiments for advection term of heat transport equation for homogeneous media are simulated on different grids such as fine scale grid and different coarse scale grids all these experiments are performed for case without and with fractures.

Chapter 2

Fluid Flow in Fractured Porous Media

The focus in this chaper is on the fluid flow in fractured porus media. The presentation in this chapter follows [23], [27], [18], [26] and [13].

2.1 Porous Media

A *porous medium* is material consists of a solid part usually called matrix and a pore space allows for fluid to flow through the medium. Fluid flow through the porous medium is considered as *single-phase flow* if the pore space is filled with just one fluid. In this thesis only, single-phase flow will be considered and below we are going to describe porous media parameters and governing equations related to it.

The geometry of porous medium is highly complex and cannot be described in mathematically manner directly. The microscopical point of a porous medium is either in the matrix or in the pore space hence properties such as porosity cannot in general be defined point-wise [18], [21]. To cope with this issue the concept of *representative elementary volume*, *REV* first introduced by Bear [13] is used. The *REV* concept underlines continuum approach where the passage from microscopic level to the macroscopic is obtained by averaging the microscopic values of the porosity over a certain volume of porous medium [12]. The *REV* is particular length scale where the parameter of interest in our case porosity does not change significantly, which allows us to take its average value over the *REV* scale [13].

We will use the porosity as one of the basic macroscopic matrix properties

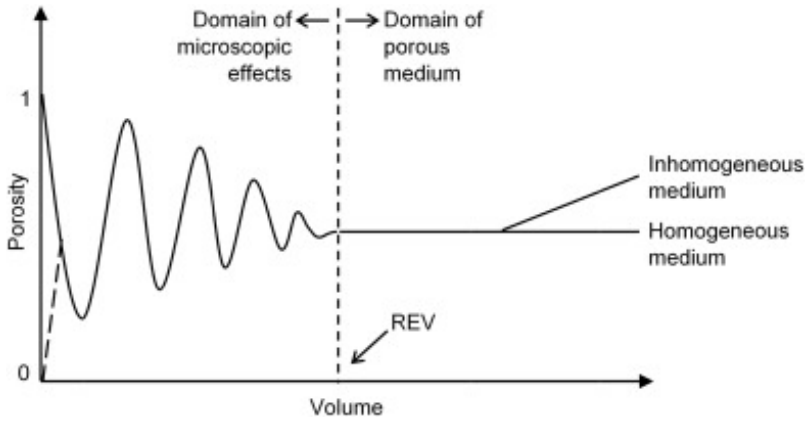


Figure 2.1: Representative elementary volume (from [28])

to illustrate the *REV* concept 2.1. Porosity is denoted by ϕ is:

$$\phi = \frac{V_p}{V_t}, \tag{2.1}$$

where V_p is the volume of the pore space in *REV* and V_t is total volume of *REV* [23]. Figure 2.1 give as visualization of the *REV* concept. A displacement on the microscopic scale (left) lead to huge value changes of the porosity, values changes from one to zero if the movement happened from the rock to the small-scale pore [26]. On the *REV* scale (middle) these heterogeneities are averaged. If the property varies depending on position in domain the medium is called heterogeneous (right), otherwise the medium is called homogeneous [21].

2.2 Darcy's Law

The law was stated by Henry Darcy French hydraulic engineer in 1856 as a result of a series of experiments [13],[23]. In this thesis we will use the generalized Darcy's law obtained from Navier-Stokes equation and express the conservation of momentum [27],[19]. The equation

$$\vec{v} = -\frac{\mathbf{K}}{\mu}(\nabla p - \rho\vec{g}), \tag{2.2}$$

generalizes Darcy's law, where $\vec{v}$ is volumetric velocity flux called *Darcy's velocity*, $\mathbf{K}$ is permeability tensor, μ is viscosity of the fluid, ρ is fluid density, p is pressure and $\vec{g}$ is gravitational acceleration vector.

As it is stated above Darcy velocity is not actual velocity but rather averaged volume flux density over the entire domain including both fluid and matrix [18]. The averaged fluid velocity V called *intrinsic velocity* is flow only through cross section of fluid [18]. Relation between these two velocities referred to as:

$$\vec{V} = \frac{\vec{v}}{\phi}, \tag{2.3}$$

where ϕ is porosity.

Permeability denoted by K can be defined as a porous medium ability to transmit fluid through it. The SI unit for permeability is m^2 or derived units Darcy $1D \approx 0.987 \times 10^{-12} m^2$ or more common milliDarcy mD [23]. The permeability in single point of medium can varie with respect to direction in which case the medium is referred to as *anisotropic* and permeability is represent using second order tensor [23]. Medium is referred to as *isotropic* if the property values are independent of directions. Using hydraulic conductivity tensor κ permeability can be defined as follows:

$$\mathbf{K} = \kappa \frac{\mu}{\rho g}. \tag{2.4}$$

Permeability values for fractured ingenious and metamorphic rock is in range from 10^{-15} m^2 to 10^{-11} m^2 [23]. There is difference in permeability between matrix and fractures. In this thesis we consider high permeable fractures which dictate fluid flow.

2.3 Conservation of Mass

Mass conservation law states that any change of mass in the arbitrary volume Ω is equal to the net mass flow into the volume through the boundaries plus mass added to the volume through the source Q [23].
Mass conservation equation referred to as:

$$\int_\Omega \frac{\partial}{\partial t}(\phi\rho) d\Omega + \int_{\partial\Omega} (\rho\vec{v}) \cdot \vec{n} dS = \int_\Omega Q d\Omega, \tag{2.5}$$

where Ω is domain with boundary $\partial\Omega$, $\vec{n}$ is outward normal unit vector and Q is possible source or sink. After applying divergence theorem for an arbitrary domain Ω we get [18]

$$\int_\Omega (\frac{\partial(\phi\rho)}{\partial t} + \nabla \cdot (\rho\vec{v}) - Q) dV = 0, \tag{2.6}$$

and in differential form

$$\frac{\partial(\phi\rho)}{\partial t} + \nabla \cdot (\rho\vec{v}) = Q. \tag{2.7}$$

Mass conservation equation has velocity vector components as unknowns and in addition either density or porosity could be unknowns if either of them changes over time. This leads to conclusion that mass conservation equation cannot be solved without combining with other constitutive equations such as Darcy's law 2.2 for example [27].

In this thesis we consider a single phase flow for incompressible fluid without gravity, under these assumptions mass conservation equation 2.5 stated as follows [22]:

$$\nabla \cdot \vec{v} = Q. \tag{2.8}$$

This equation in combination with Darcy's law 2.2 give us an elliptic partial differential equation for pressure [22]:

$$-\nabla \cdot (\frac{\mathbf{K}}{\mu}\nabla p) = Q, \tag{2.9}$$

where $\mathbf{K}$ is permeability tensor, μ is constant viscosity, p is pressure and Q contains any sources or sink.

2.4 Time of Flight

In order to give explanation for time of flight we use the following sources [24], [27], [19]. The time of flight $\tau(x)$ is time that passive fluid particle travels from injector to a given point x [20]. To define $\tau(x)$ we will use streamlines. Streamlines Ψ are family of curves that is identical to the path traced out by a fluid particle under the assumption of steady velocity (considered herein) [20]. Particle moves with fluid velocity field and we can assume that $\vec{v} = \vec{v}(x)$. Velocity field vector is tangential to the streamline Ψ at every point x thus the streamline is an integral curve of the velocity field [20],[27].
Time of flight along the streamline can be mathematically expressed as differential equation:

$$\vec{v} \cdot \bigtriangledown\tau = \phi, \tag{2.10}$$

or equivalently as parametric equation:

$$\tau(r) = \int_0^r \frac{\phi(\vec{x}(s))}{|\vec{v}(\vec{x}(s))|} ds, \tag{2.11}$$

where r is a distance fluid particle makes along a streamline $\vec{v}$ is Darcy's velocity and ϕ is porosity.

2.5 Fracture Flow

This section is based on the following sources [14], [15], [25], [18] and [22]. Most geothermal systems are found in fracture rock. The size of fractures in fractured geothermal reservoir typically follow a power low distribution [14], [22]. The main characteristics of fractal geometry is the lack of homogenization scale or representative elementary volume [14]. Fractures can be distinguished as large-scale fractures and small-scale fractures [14]. Large-scale fractures tend to dictate fluid flow through the reservoir and is of particular interest in this thesis [22].

The fracture network possesses various properties that needed to be defined to fully characterize fractures, most common are fracture aperture and fracture density [18]. A fracture aperture is defined as a perpendicular distance between fracture walls [15]. The fracture aperture has essential influence on fluid flow and transport through fractures making aperture crucial parameter to determine [18], [15], [14]. However, the determination is not trivial because in nature this value is not constant but rather varies throughout the fracture from a few micrometers up to centimeters [14], [15].
Fracture density is the ratio of fracture space in the medium [18].
The fluid flow through fractures can be explained by considering it as laminar between two parallel plates with constant aperture a_f see 2.2, [18]. To apply parallel concept we assume that the fracture length is significantly larger then fracture aperture [15]. The model is developed from continuity equation and Navier-Stokes equation for incompressible flow referred to as:

$$\rho(\frac{\partial \vec{v}}{\partial t} + \vec{v}\nabla\vec{v}) = -\nabla p + \mu\nabla\vec{v} \tag{2.12}$$

Boundary conditions assume constant pressures p_1 and p_2 at the inlet and outlet boundaries respectively where $p_1 \geq p_2$ [18]. Under this boundary conditions 2.12 reduces to:

$$\frac{p_1 - p_2}{l} + \mu\frac{\partial^2 v_y}{\partial z^2} = 0 \tag{2.13}$$

Flow takes place only in y direction, in other directions is equal to zero. Here l denotes fracture length. After 2.12 is integrated twice and after boundary conditions are implemented we obtained the following velocity field:

$$v_y(z) = \frac{1}{2\mu}\frac{p_1 - p_2}{l}[(a_f^2 - z^2)], \tag{2.14}$$

using 2.14 velocity is equal to

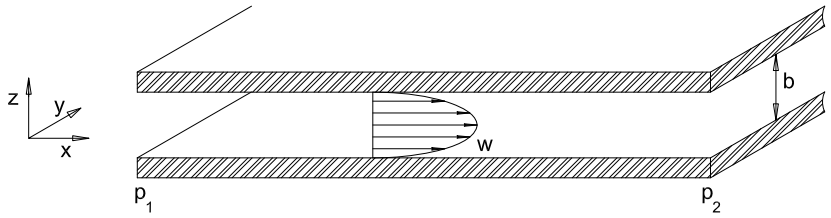


Figure 2.2: Laminar flow between two parallel plates

$$v_y(z) = \frac{1}{wb} \int_{-a_f/2}^{a_f/2} \int_0^w v_y(z) dy dz. \tag{2.15}$$

Using equation 2.15 the average velocity is [18]:

$$\begin{aligned} \overline{v} &= \frac{1}{wb} \int_{-a_f/2}^{a_f/2} \int_0^w v_y(z) dy dz \\ &= \frac{1}{2\mu b} \frac{p_1 - p_2}{l} \int_{-a_f/2}^{a_f/2} [(a_f/2)^2 - (z)^2] dz \\ &= \frac{a_f^2}{12\mu} \frac{p_1 - p_2}{l}. \end{aligned} \tag{2.16}$$

This average velocity can be seen as Darcy's velocity [18].

Fracture permeability is equal to:

$$\mathbf{K} = \frac{a_f^2}{12}. \tag{2.17}$$

The product of fracture permeability and the cross sectional area is the variable known as transmissivity:

$$T = \mathbf{K}A = \frac{w a_f^3}{12} \tag{2.18}$$

Volumetric flow rate is

$$q_v = \frac{w a_f^3}{12\mu} \frac{p_1 - p_2}{l}. \tag{2.19}$$

Equation 2.19 is known as *cubic law*.

Chapter 3

Heat Transfer

In this chapter we will discuss heat transfer in porous media. There are three different processes of heat transfer: conduction, convection, radiation [18]. The aim of this chapter is to arrive to advection-conduction equation, in this purpose we will explain heat transfer due to conduction and convection, the heat transfer due to radiation is neglected. For presentation of heat transport and governing equations in this chaper we use the following sources [27], [18], [23].

3.1 Conduction

Conduction or *diffusion* is heat transfer that occurs at a molecular level, where energy is transferred from more energetic particles to less energetic particles. At the molecular level, heat flow can be explained via collisions of molecules for fluids, and via the vibrations of particles and free electrons for solids. In geothermal reservoir conduction is the way that heat reaches and leaves the surface of the matrix [27].

At the macroscopic level heat is transferred form the object with high temperature to the object with low temperature this is also known as a *Fourier's law* or a thermodynamics analogue of Darcy's law [18]. Mathematically conductive heat flux is described by Fourier's law [27]:

$$\vec{q}_{cond} = -\mathbf{k}\nabla T \tag{3.1}$$

where $\vec{q}_{cond}$ is conductive heat flux, $\mathbf{k}$ is the thermal conductivity while T is temperature [27]. The *thermal conductivity* $\mathbf{k}$ is material ability to conduct heat, it is a second order tensor and depends on pressure and temperature [27]. In this paper we will assume that thermal conductivity is equal to zero.

3.2 Convection

Convection is a heat transfer that occurs by the molecular movement of a fluids. It consists of two separate mechanisms: *diffusion* and *advection*. Diffusion occurs due to random motion and interaction of the fluid molecules, while advection represents collectively fluid moving. Considering the cause of convection, it can be classified as either natural or forced convection. Natural convection occurs due to buoyancy forces (warming or cooling) at the other hand the forced convection occurs due to some external source or mechanism (pump, fan). The heat flux due to advection can be found as:

$$\vec{q}_{ad} = e\vec{V}, \tag{3.2}$$

where e is the thermal energy density and $\vec{V}$ is fluid velocity (intrinsic volume flux) [27].

Advective heat flux can be defined in terms of heat capacity c_p as:

$$\vec{q}_{ad} = \rho c_p \vec{V} T \tag{3.3}$$

where $\vec{q}_{ad}$ is advective heat flux, ρ is fluid density, $\vec{V}$ fluid velocity, T is temperature and c_p is heat capacity under constant pressure [27].

Newton's law of cooling is the experimental law which states that convective heat transfer between the fluid and the solid is proportional to differences in temperature between fluid and solid [18]. Mathematically this can be written as:

$$\vec{q} = ah(T_s - T_f) \tag{3.4}$$

where $\vec{q}$ is heat flux due to diffusion, energy is transferred from solid to fluid, the temperatures T_s and T_f are the temperatures of solid and fluid, respectively, a is a surface area, h is a heat transfer coefficient [27]. Due to the fact that determining the heat transfer coefficient h is one of the most difficult part in describing the system and that Newton's law of cooling is valid only for some idealized cases, we assume the local thermal equilibrium [18]. Local thermal equilibrium LTE is achieved if the temperature difference in REV is smaller than temperature difference in system[27].Local thermal equilibrium in practice mean that $T_s = T_f$.

3.3 The Conservation of Energy

The Conservation of Energy law also known as *The First Law of Thermodynamics* states that total energy of an isolated system is conserve, i.e. the

change in energy of an arbitrary system is equal to the energy added or removed to the system over the boundaries or through the sources or sinks. The total energy of the system is equal to the sum of the mechanical energy and internal energy. For the purposes of this thesis we will consider only the last term, due to relatively slow fluid velocities and under the assumption of no gravity, term mechanical energy can be neglected. The internal energy may consists of different types of energy however, in this thesis we will assume that the internal energy is equal to thermal energy of the system. From the first law of thermodynamics the internal energy is equal to the sum of heat fluxes (advective and conductive) and heat flux through sources or sinks. The energy conservation equation for fluid phase is [27]

$$\underbrace{\phi\frac{d}{dt}\int_{\Omega} e_f dx}_{\text{change in thermal energy}} = \underbrace{-\phi\int_{\partial\Omega}\vec{q}_{ad}\cdot\vec{n}dS}_{\text{advective term}} - \underbrace{\phi\int_{\partial\Omega}\vec{q}_{cond}\cdot\vec{n}dS}_{\text{conductive term}} + \underbrace{\int_{\Omega} Q_f dx}_{\text{source term}}, \quad (3.5)$$

where subscript f states for fluid phase, ϕ is porosity, e energy density , $\vec{q}_{ad}$ is advection heat flux, $\vec{q}_{cond}$ conductive heat flux, and Q_f is the source term resulting from internal heat sources and convection from the matrix. Source term Q_f can be divided into two term:
$Q_f = Q_{cond} + \phi Q_{source} = ah(T_s - T_f) + \phi Q_{source}$.

Further, we insert equations 3.1 and 3.2 in 3.5 and in terms of LTE $Q_f = \phi Q_{source}$ we get [27]:

$$\phi\frac{de_f}{dt} + \phi\nabla\cdot(e_f\vec{V}) - \phi\nabla\cdot(\mathbf{k}_f\nabla T_f) = Q_f, \quad (3.6)$$

and for solid phase:

$$(1-\phi)\frac{de_s}{dt} - (1-\phi)\nabla\cdot(\mathbf{k}_s\nabla T_s) = Q_s. \quad (3.7)$$

The equation for solid phase differ from the equation for fluid phase because it does not include advection term and it is multiplied by $(1-\phi)$ and source term for solid phase is $Q_s = (1-\phi)\ Q_{source}$ [27]. After we sum these two equations for solid and fluid phase we will apply divergence theorem and use 3.3 and arrive to the expression for *advection-conduction equation* for temperature in differential form [27]:

$$(\rho c_p)_{eff}\frac{\partial T}{\partial t} + \underbrace{(\rho c_p)_f\vec{v}\cdot\nabla T}_{\text{advective term}} - \underbrace{\nabla\cdot(\mathbf{k}_{eff}\nabla T)}_{\text{conductive term}} = Q_{eff}, \quad (3.8)$$

where $\mathbf{k}_{eff}$ is effective thermal conductivity and Q_{eff} is effective source term:

$$\mathbf{k}_{eff} = (1-\phi)\mathbf{k}_s + \phi k_f,$$
$$Q_{eff} = Q_s + Q_f,$$
$$(\rho c_p)_{eff} = (1-\phi)(\rho c_p)_s + \phi(\rho c_p)_f \ .$$

The system of equations 2.9 and 3.8 is decoupled system of equations that together with source term and initial conditions we will solve sequentially in numerical simulations in the following chapter [22].

Chapter 4

Numerical Methods

In this chapter we will use the following sources [27], [19], [18], [17], [22] and [26]. In this chapter we will describe numerical solution of mathematical model that simulate the physical processes in the reservoir.

Mathematical model describes processes of fluid flow and heat transport from the source throughout geothermal reservoir, it is given as a set of *partial differential equations*, PDE derived in chapter 2 and 3 [19]. Geothermal reservoir is modeled as a $3D$ grid where grid cells contains rock properties porosity and permeability. Furthermore, we will describe modeling approach that we are going to use for fractured geothermal reservoir.

In this chapter we will diverse solution methods for pressure equation 2.9 and advection term of heat transport equation 3.8 that we will solve computationally in the chapter 6 . Equations will be solved sequentially first pressure equation and then advection term of heat transport equation. These equations usually cannot be solved analytically except for some simplified cases [27]. Various numerical methods have been developed in order to computationally solve PDE. Since pressure equation is elliptic while advection term of heat transport equation is hyperbolic PDE we will use different discretization techniques for them. This chapter is organized in the way that the section on the discretization in space will be followed with the section on fine scale discretization and discretization in time. Discretization in space will give us grid construction in more details and motivation for using *discrete fracture matrix* DFM model and the construction of the model. We will describe two different discretization methods for mathematical model of fluid flow and heat transport processes. Finally, time discretization explains temporal discretization of advection term of heat transport equation.

4.1 Discretization in Space

Numerical methods for solving porous media problems are run on some spatial domain which represent geothermal reservoir. Due to its complexity geothermal reservoir is modeled using gridding techniques, while fracture network is modeled explicitly using discrete matrix model. Some of the most used numerical methods include the finite difference method (FDM), the finite element method (FEM) and finite volume method (FVM) [18]. Each of the methods stated above have its advantages and constraints and the choice of the method is related to the particular problem [18]. In this thesis we will use the (FVM) method.

4.1.1 Gridding

Geothermal reservoirs are characterized by their geometry and by their rock properties and can be modeled using different gird types [17]. The simplest grid type is structured gird with regular grid cells. In this thesis we will consider a $3D$ Cartesian grid as a simplest model of geothermal reservoir without fractures. In figure 4.1 Cartesian $3D$ grid is shown on the left-hand side and on the right-hand side we can see grid cell with its basic geometry characteristics. The advantages of this grid included that they are relatively easy to construct and can be easily partitioning into coarse grid if need see chapter 5 [17]. In addition, they are all orthogonal which allows us to apply simple numerical discretization schemes for instance two-point flux approximation that in turn produces simpler systems to solve [17]. The main disadvantage of Cartesian $3D$ grid is that it is not flexible for modeling of geological features [17]. In 3 D Cartesian grid cell is a cube associated with a certain number of parameters to be determined when the system is solved 4.1 [19]. Each of the grid cell represents representative elementary volume REV [17]. The geometry of the grid cell is defined by a set of vertexes, a set of edges that connect pair of vertexes and a set of faces [19]. Faces are surfaces delimited by a subset of the edges, that define the interface between two cells [19]. In three dimensions face is the plate between two neighboring cells. We say that two cells are connected if they have a common face [19]. In our grids fracutres are plates papallel to one of the axis.

4.1.2 Fractures and Discrete Fracture Matrix (DFM) Methods

This section is mainly based on the sources [26], [22] and [18]. In some cases, fractures may not satisfy the conditions allowing us to use the averaging by

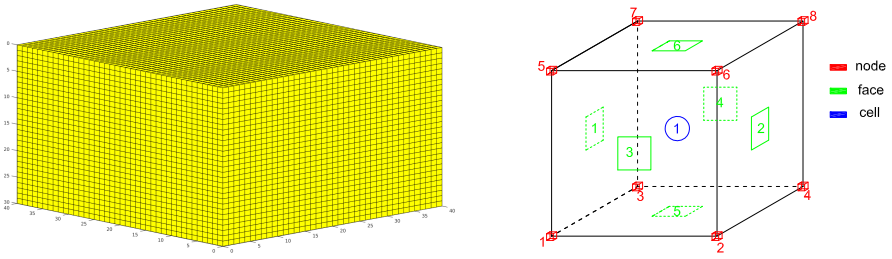


Figure 4.1: 3 D Cartesian grid and a single grid cell with belonging information

REV and must be accounted for explicitly [26].

Their properties may differ dramatically from the properties of the rest of the rock. Consequently fluid and transport properties are strongly dominated by the presence of the fractures. Although we do not know scale separations for realistic scenarios we will assume that we can determine which fractures are significant one to be resolved explicitly [26].

Modeling approaches for fractured reservoirs can be classified into three main categories according to their spatial representation in the reservoir: single-continuum models, multi continuum models and discrete fracture models [22]. Single continuum models assume that the fractures effects on the fluid flow is within some limit and that it could be upscaled and included in the modified permeability [26]. The concept of REV is valid for this model. The advantage of this model is that the standard reservoir simulations can be applied.

In multi-continuum models fractures and matrix are modeled by a representative continuum with associated flow characteristics. The main advantage of this model comparing to the single model is that it allows us to better capture the integral transport behavior[22].

Discrete fracture network (DFN) models assume that the flow takes place in the fractures neglected the surrounding pores medium [26], [22].

Discrete fracture-matrix (DFM) models represents the combination of the multi-continuum models and (DFN) models. The main characteristics of the discrete fracture model is that large-scale fractures are represented explicitly while the effects of the small-scale fractures are included into the permeability of the porous matrix [22]. Fractures are very thin compared to their length, this characteristic allows us to model them in co-dimension [19]. In DFM model fractures are constructed by converting faces into a thin cells called *hybrid cells* with a volume equal to the product of aperture and a hybrid face area. Hybrid cells are lower dimensional objects i.e. plates in our case and consists of hybrid faces i.e. lines [6]. Hybrid cells are connected

both with other hybrid cells and matrix cells, except in the case of isolated fracture and matrix cells [24]. These connections will be furthered analyzed in section 4.2.1.

A major challenge in discrete fracture model is large computational cost due to individual representation of fractures in large-scale simulations. At the other hand regarding accuracy, results obtained by DFM model gives the most accurate impact of the fracture network on the fluid flow.

4.2 Fine Scale Discretization

We consider that Ω is reservoir domain represented by a discrete model i.e grid that consists of grid cells denoted Ω_i referred to as *control volumes* . The finite volume method (FMV) is based on integration of the PDE over each control volume Ω_i [17]. In the next step volume integral will be converted using divergence theorem into the flux over the surface $\partial\Omega_i$. This principle leads to a method with the mass conservation law within each control volume [18]. To illustrate the method, we will use pressure equation with source term 2.9. Under the assumptions of constant density and viscosity without gravity condition equation becomes:

$$-\nabla \cdot (\mathbf{K}\nabla p) = Q \tag{4.1}$$

Let Ω be a fine grid domain and Ω_i set of fine grid cells i.e. *control volumes* such that $\Omega = \cup\Omega_i$, where $i = 1, ..., N_f$, N_f is the total number of fine grid cells. The equation 4.1 is integrated over each grid cell Ω_i and after applying divergence theorem it becomes [18]:

$$\int_{\partial\Omega_i} \vec{v} \cdot \vec{n} dS = \int_{\Omega_i} Q dV, i = 1, ...N_f \tag{4.2}$$

where $\partial\Omega_i$ is the boundary of the grid cell Ω_i, and $\vec{n}$ is the outward normal vector of the $\partial\Omega_i$. Here $\vec{v} = \mathbf{K}\nabla p$ is the velocity vector.

The system of equations 4.2 depend on the finite set of unknowns that can be solved over each control volume Ω_i [27]. We suppose that the PDE 2.9 is solved over the domain Ω if the corresponding system of discretized equations are solved over each control volume Ω_i [17]. The discretization of equation 2.9 rises to a linear system:

$$\mathbf{A}\mathbf{p}_f = \mathbf{Q}, \tag{4.3}$$

where vector $\mathbf{p}_f$ contains cell-centred pressure values.

4.2.1 Two Point Flux Approximation

A two-point flux approximation $TPFA$ is used to discretized pressure equation 2.9. $TPFA$ is the simplest of FVM discretization. It is widely used due to its advantages as simplicity, computational efficiency and robustness.

The equation 4.2 is the mass conservation equation for each grid cell and using Darcy's law the flux over each side can be approximated by the pressures from the neighboring sides [27]. The idea is to express the pressure gradient as the difference between the pressure at the face centroid and the

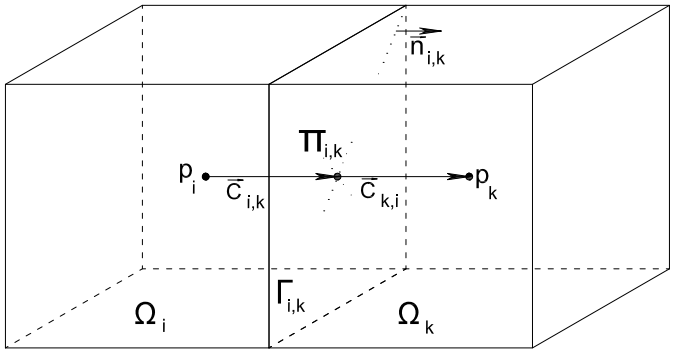


Figure 4.2: Two-point flux approximation for grid cells Ω_i and Ω_k in $3D$ Cartesian grid

pressure at the cell centroid [19]. The flux from cell Ω_i to cell Ω_k is approximated by [19]

$$v_{i,k} \approx A_{i,k}\mathbf{K}_i \frac{(p_i - \pi_{i,k})\vec{c}_{i,k}}{\mid \vec{c}_{i,k} \mid^2} \cdot \vec{n}_{i,k} = T_{i,k}(p_i - \pi_{i,k}), \tag{4.4}$$

where $A_{i,k}$ is the area of shared edge, $\mathbf{K}_i$ is the permeability in cell Ω_i, $\vec{c}_{i,k}$ is the distance vector from centroid of the cell to the face center, p_i is the pressure at the centroid in the cell Ω_i, $\pi_{i,k}$ is the pressure in the face center $\Gamma_{i,k}$ and $T_{i,k}$ is a half transmissibility (Figure 4.2). Considering the continuity of fluxes and the continuity of face pressures, we get the formula for the flux across the interface Γ_{ik} between the cells Ω_i and Ω_k

$$T_{i,k}^{-1} v_{ik} = p_i - \pi_{ik}, \tag{4.5}$$

$$-T_{k,i}^{-1} v_{ik} = p_k - \pi_{ik}, \tag{4.6}$$

After we eliminate the interface pressure π_{ik} we obtain the following $TPFA$ system of equations:

$$\Sigma_{k=1}^{n_f} T_{ik}(p_i - p_k) = q_i \tag{4.7}$$

for each cell Ω_i, where n_f is the face number for the cell Ω_i. Face transmissibility denoted by T_{ik} is equal to the harmonic average of the two half

transmissibility associated to the common face for cells i and k [19], [27]. Half transmissibility is equlal to:

$$\alpha_{i,k} = \frac{A_{i,k}\vec{n}_{i,k}K_i}{\vec{c}_{i,k}\cdot\vec{c}_{i,k}}\vec{c}_{i,k} \tag{4.8}$$

A_i is area, $\vec{n}_i$ is a unit normal vector, K_i is permeability assigned to the cell i and $\vec{c}_i$ is distance vector from cell center to the face centroid. The system 4.7 is made symmetric and positive definite by adding the positive constant $p_1 = 0$ to the first diagonal of the matrix $A = [a_{ij}]$ where

$$a_{ij} = \begin{cases} \Sigma_k T_{ik}, & \text{if } i = j \\ -T_{ij}, & \text{if } i \neq j \end{cases}.$$

The matrix A is sparse and will have a *heptadiagonal* structure for $3D$ grid [19].

Now we will take into account the fractures and we will implement $TPFA$ method for them. The unknowns associated with the fracture cells remains after the completing the unknown's pressures at the centroids of the matrix cells [26]. As lower dimensional objects fracture faces are converted into fracture cells by multiplying the fracture apertures with original face area [6]. This way the volume of the fracture cell is obtained. Hybrid faces are defined as faces parallel to the fracture faces displaced half an aperture to either side [6]. Half transmissibility is corresponding to each hybrid face [6]. Since fracture cell may have common face with matrix cell or with other fracture cell, we will distinguish matrix-fracture connections and fracture-fracture connections. For common face between fracture cell and matrix cell i.e. matrix fracure connection, half transmissibility is obtained the same way as for the matrix-matrix connections with adjustment to the distance vectors [24], [26]. Two different vectors are used on the matrix and on the fracture side of the face [24]:

$$\vec{c}_f = \frac{a_f}{2}\vec{n}_f, \tag{4.9}$$

and

$$\vec{c}_m = \vec{c}_m - \vec{c}_f, \tag{4.10}$$

where $\vec{c}_f$ and $\vec{c}_m$ represents the distance from fracture and from matrix while a_f is fracture aperture (Figure 4.3). For common face between two fracture cells i.e.fracture-fracture connections and under the assumption that fracture length is bigger than fracture aperture, half transmissibility is [26]

$$T_{ij} = \frac{\alpha_i\alpha_j}{\Sigma_{k=1}^{n}\alpha_k} \tag{4.11}$$

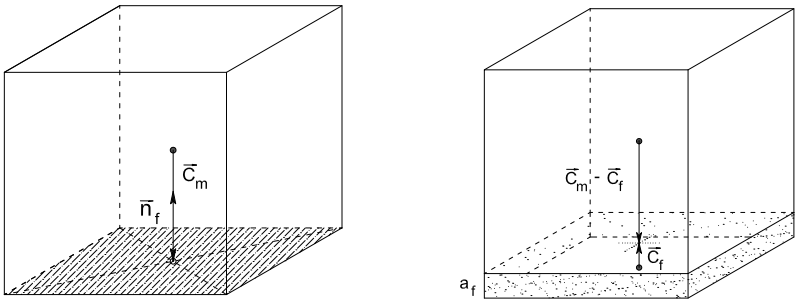


Figure 4.3: Grid domain and computational domain for matrix-fracture connections

$\alpha_i, \alpha_j, \alpha_k$ are half transmissibility of fracture cells while n is the number of fracture cells meeting intersection.

We will close this section with discussion of convergence of $TPFA$ method. $TPFA$ method converge if the grid is $K-orthogonal$. The $K-orthogonality$ condition means that vectors $\mathbf{K}\vec{n}_{i,k}$ and $\vec{c}_{i,k}$ are parallel [19]. In the case that $K-orthogonality$ condition is not fulfilled $TPFA$ discretization is not consistent and numerical solution may not converge to the true solution [18].

4.2.2 Upwind discretization

Advection term of heat transport equation is discretized using the upwind method. The reason for this type of discretization is that the flux direction is known to us. The advantage of this method is that it distinguishes inflow and outflow through the edge of the cell and based on that gives the positive or negative sign to the flux vector [27]. The advection term referred to as:

$$\nabla \cdot (T\vec{v}) = Q \tag{4.12}$$

After integrating the equation over each grid cell and applying divergence theorem we get [27]

$$\int_{\partial\Omega_i} (T\vec{v}) \cdot \vec{n} dS = \int_{\Omega_i} Q d\Omega_i. \tag{4.13}$$

We will explore the flux flow through the internal edges. Let Γ_{ij} be the common face between two cells Ω_i and Ω_j. The flux over the edge Γ_{ij} in the direction of the unit normal vector $\vec{n}_{ij}$ is the scalar product of the vectors

$\vec{n}_{ij}$ and $\vec{v}$ [27]. The heat flux through the edge Γ_{ij} is

$$\int_{\partial\Omega_i} (T\vec{v}) \cdot \vec{n} dS \approx \begin{cases} T_i \int_{\Gamma_{ij}} \vec{v}_{ij} \cdot \vec{n}_{ij} dS, & \text{if } \vec{v}_{ij} \cdot \vec{n}_{ij} \geq 0 \\ -T_j \int_{\Gamma_{ij}} \vec{v}_{ij} \cdot \vec{n}_{ij} dS, & \text{if } \vec{v}_{ij} \cdot \vec{n}_{ij} < 0 \end{cases}$$

.

4.3 Discretization in Time

The pressure equation 2.9 for incompressible fluid flow with constant porosity and viscosity becomes elliptic partial differential equation not dependent on time [27]. This equation is solved once using the $TPFA$ method. At the other hand advection-conduction heat transport equation 3.8 consists of two terms advection and conduction and needs temporal discretization. The coupled system of equations is of the form [18]:

$$\frac{\partial}{\partial t} \int_{\Omega_i} T d\Omega = \int_{\Omega_i} F(T) d\Omega, \tag{4.14}$$

where F is operator depends on fine scale temperature T, time t, and Darcy's velocity $\vec{v}$ and Ω_i is grid cell. After multiplying with reciprocal domain we get:

$$\frac{\partial}{\partial t} (\frac{1}{\mid \Omega_i \mid} \int_{\Omega_i} T d\Omega) = \frac{1}{\mid \Omega_i \mid} \int_{\Omega_i} F(T) d\Omega. \tag{4.15}$$

The product on the left-hand side is the average of the integrand in cell Ω_i while the right-hand hide is the function average in cell Ω_i [18]. If we list values and functions in all cells of the grid as a vector we get the followed simplified system of equations [18]:

$$\frac{dT_i}{dt} = F_i(T), \tag{4.16}$$

where T is temperature vector and F is operator containing discretization of advection-conduction equation

$$T = \begin{bmatrix} T_1 \\ \vdots \\ T_N \end{bmatrix} F = \begin{bmatrix} F_1 \\ \vdots \\ F_N \end{bmatrix}$$

.

Time scale is discretized as $t_0, t_1, ..., t_N$ with time step $\triangle t_n = t_{n+1} - t_n$. After applying *theta method* for $\Theta \in [0, 1)$ we get:

$$\frac{T_i^{n+1} - T_i^n}{\triangle t_n} = \Theta F_i(T_i^{n+1}) - (\Theta - 1) F_i(T_i^n). \tag{4.17}$$

Advection-conduction heat transport equation is discretized using $TPFA$ method for conduction term and upwind scheme for advection term. Since in this thesis we are focused on advection part of the equation while assume that the conduction part is equal to zero, we will assume that $\Theta = 1$ and obtain *Euler's backward method*. The Euler's backward scheme is:

$$T_i^{n+1} = T_i^n + \triangle t F_i(t_n + \triangle t, T_i^{n+1}). \tag{4.18}$$

Advection part contains heat capacity c_p which usually depends on temperature thus the system 4.18 is not linear and we need to use iterative Newton Raphson method to approximate T^{n+1} [27]. For simplicity, we will in this thesis assume, that heat capacity is constant which give us that operator F is linear. We have to solve the system:

$$F_i(T_i) = \mathbf{A}T_i. \tag{4.19}$$

In $MATLAB$ linear system of equations is solved using LU solver [19]. This mean that the matrix $\mathbf{A}$ is decomposed into lower and upper triangular matrices respectively $\mathbf{L}$ and $\mathbf{U}$ where $\mathbf{A}$=$\mathbf{LU}$[27]. Euler's backward method as an implicit scheme is time stable [26].

Chapter 5

Coarse Scale Discretization

The presentation in this chapter follows [16], [17] and [27]. For section Error Computaion we use [26] for source. Fine scale reservoirs models are very detailed which leads to large matrices in transport solvers. Sufficient computational capacities are needed to process these detailed models in short time. In this thesis we will present two techniques to increase computational efficiency: upscaling and uppgidding. Upgridding is a process of creating a coarse grid while upscaling bring reservoir properties from the fine scale grid to coarse scale grid. Typically, upscaling of the reservoir properties is carried out in combination with upgridding.

5.1 Upscaling

There are numerous techniques developed to upscale reservoir parameters. The choice of the adequate upscaling procedure depends on the problem we are faced with. Here we will explain upscaling using standard arithmetic mean. Reservoirs properties included in heat transport equation need to be upscaled. Fluid porosity on coarse scale can be upscaled through the simple volumetric average:

$$\phi^* = \frac{1}{V_{\Omega_l^c}} \int_{V_{\Omega_l^c}} \phi dV \approx (\Sigma_{i \in \Omega_c^l} \phi_i V_i / V_{\Omega_l^c}) \tag{5.1}$$

where Ω_l^c denotes a coarse block with a bulk volume denoted as $V_{\Omega_l^c}$. Coarse block Ω_l^c consists of fine grid cells denoted with i, porosity of a fine cell i is equal to ψ_i and volume is equal to V_i.

Since in this thesis we consider fine scale grid with homogeneous medium, the only rock parameter that we are going to upscale is porosity. In the case of single phase flow with constant porosity, such is in this thesis, the resulting

porosity in each coarse grid blocks will be constant value the same as for fine gird cells. Permeability is used only once when solving the pressure equation so we will not upscale it.

Coarse scale flow rate q^*_{vol} across the interface l, between two grid blocks, is equal to the sum of the flow rates corresponding to fine scale faces which gain the interface l i.e [27]:

$$q^*_{vol} = \Sigma_{i=1}^{N_l^f} q_i \tag{5.2}$$

$$q_i = A_i \vec{v}_i \cdot \vec{n}_i \tag{5.3}$$

where N_l^f is the total number of fine scale faces contains in coarse face l. For fine scale face i which belong to coarse interface l we denote A_i as face area, $\vec{n}_i$ as normal outward vector and $\vec{v}_i$ as velocity vector. Normal on a coarse face is equal to sum of normal vectors on fine scale faces which belong to the coarse face [27].

5.1.1 Coarse Scale Advection Term

The advection term of heat transport equation on the coarse scale referred to as:

$$\int_{\partial\Omega_l^c} (\vec{v} \cdot \vec{n}) dS = \int_{\Omega_i^c} Q d\Omega \tag{5.4}$$

The sub-script c indicating that integration is done on the coarse grid block Ω_i^c. The flux over the coarse scale interface γ_{ij} is equal to the sum of fluxes over the corresponding fine scale interfaces Γ_{ij}. We will find all fine scale faces that lie on the interface between the coarse grid blocks. The fine-scale fluxes are obtained after we solved pressure equation using $TPFA$ method. The influx on fine scale face has sign 1, while out flux from the fine scale face has a sign -1 [19]. To get flux values on the coarse grid we need to multiply fine scale flues with sign before summing them along coarse scale interface to obtain coarse net flux [5]. We assume that temperature is constant over each grid block Ω_i^c. Temperature in the coarse grid block is equal to upscaled fine scale temperature [17]:

$$T_l = \frac{1}{\mid \Omega_l^c \mid} \int_{\Omega_l^c} T d\Omega \approx \Sigma_{i \in \Omega_l^c} T_i \frac{\mid \Omega_i^c \mid}{\mid \Omega_l^c \mid} \tag{5.5}$$

Then conservative coarse scale discretization of advection term is obtained by summing a single point discretization for all fine grid cells in coarse grid block [17].

$$\int_{\partial\Omega_l^c} (\vec{v}T) \cdot \vec{n} dS \approx \Sigma_{k \neq l} max(T_l \Sigma_{\Gamma_{ij} \in \gamma_{kl}} \vec{v}_{ij} - T_k \Sigma_{\Gamma_{ij} \in \gamma_{kl}} \vec{v}_{ij}), \tag{5.6}$$

From the formula for advection term discretization on coarse grid we can notice that this is a single point upwind scheme on the coarse grid [17].

5.2 Upgridding

Coarse grid is constructed by grouping fine grid cells into blocks Ω_l^c where $l = 1, ..., N_c$, N_c is the total number of coarse grid blocks [16]. We use a partitioning vector to represent coarse grid. Partitioning vector consists of N_f elements, where N_f is the number of fine grid cells. If fine grid cell i belongs to coarse block Ω_l^c partitioning vector has value l on the $i - th$ position. The relationship between coarse grid blocks and belonging fine grid cells that make each of the blocks can be seen in figure 5.1.

In this paper we present the flow-based non-uniform partitioning of the fine scale grid using amalgamation algorithm see [1]. We consider coarsening of fine scale grid with fracture filed and fine scale grid without fractures. Since fracture cells are low dimensional objects, coarsening for fine grid with fractures cannot be done directly. Special emphasis is put on how to treat fine scale fracture cells. The approach we use is to add the set of the fine scale fracture cells to the set of fine scale matrix cells thus, fracture cells and matrix cells will be treat equally.

For upgridding algorithm based on amalgamation we need to define indicators based on flow parameters: time-of-flight, permeability and velocity. Since velocity is vector we will rather use magnitude of velocity instead. We ideally, would like that resulting coarse grid for two different idicators: time of flight and prermeability, have small grid blocks both for a case of low values of time-of-flight and for case with high values of permeability. This leads to scaling of flow indicators using indicator function. For a chosen flow parameter, we define indicator function as [27]:

$$g(y) = log|y| - min(log|y|) + 1. \tag{5.7}$$

Flow parameters are absolute permeability, magnitude of velocity and time-of-flight. For upgridding of fractured fine scale gird, we will introduce two new coarsening parameters: square root to the nearest fracture and combination of the time-of-flight and square root to the nearest fracture. The motivation for introducing these parameters is that we are focused on coarse scale discretization that give us a good representation of fractured network. In that context the choice of distance parameter seems as a natural choice, the main reason for using square root instead of distance to the nearest fracture is its smaller scale and smoothness. The combination of two parameters

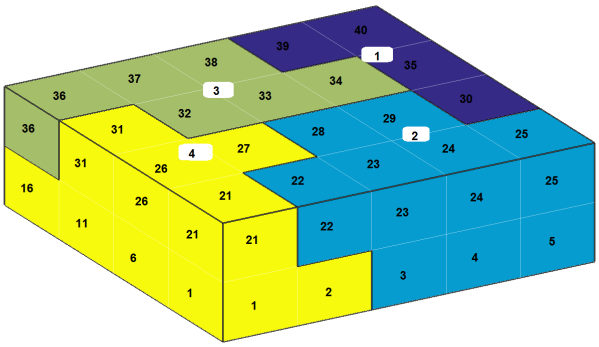


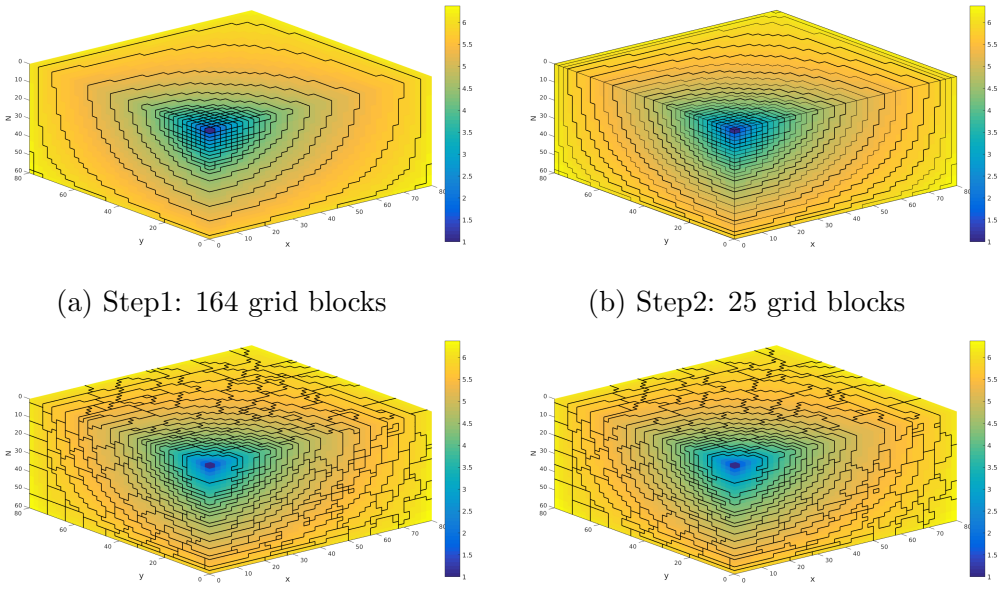
Figure 5.1: Flow based coarse grid with coarse grid blocks consist of fine grid cells

flow parameter time-of-flight and distance parameter square root to the nearest fracture is the result of our wish to construct coarse grid that works fine for transport problems dominated by fracture flow. For square root distance as indicator we will not use indicator function from 5.7 but rather indicator itself, in the case of combination of two indicators we will define indicator function as a sum of scaled time-of-flight using 5.7 and square root distance.

Flow-based non-uniform coarsening method consists of four steps as follows: [16]:

1. Compute an initial partitioning using the flow indicator function
2. Merging of small grid blocks with volumes under the lower volume bound into the larger blocks [21]
3. Refining grid blocks with total volume that exceeds defined upper bound
4. Repeat step 2.

In order to start the algorithm we need to define N as number of bins which is utilized in the first step, N_L as lower volume bound, N_U as upper bound of total amount of flow through each grid block. The indicator function defined with 5.7 is segmented into N uniform blocks after the first step. A lower bound N_L prevented algorithm from generating to small grid blocks [17]. If the condition $\mid \Omega_l^c \mid < \frac{N_L}{N} \mid \Omega \mid$ is fulfilled for coarse gird block Ω_l^c obtained after first step this block is merged with neighboring block that has the closest value of indicator function [17]. The upper bound N_U prevented



(a) Step1: 164 grid blocks

(b) Step2: 25 grid blocks

(c) Step 3: 1345 grid blocks

(d) Step 4: 805 grid blocks

Figure 5.2: Four steps of the coarsening algorithm described in 5.2 based on time-of-flight flow indicator

algorithm from generating too large gird blocks [16], [19]. This value is used to refine coarse grid blocks with volume grater than N_U. Refining algorithm starts from the fine grid cell which is furthered away from coarse grid block center and letting it add the neighboring fine grid cells until the upper bound is exceeded [21]. The algorithm is repeated if the upper bound in the remaining fine grid cells inside the block violate the upper bound. After refining grid blocks we repeat step 2 because there might remain grid blocks that violate the lower bound [16]. Predefined values: N, N_L and N_U have significant impact on the number of grid blocks and on the quality of the grid see [19],[16]. From [16] we can notice that smaller number of bins lead to a larger grid blocks, also small values of lower and upper bound lead to smaller coarse blocks in regions with low flow. The coarse grid blocks constructed this way are connected components. In figure 5.2 we can see four steps of coarsening algorithm described in 5.2. From figure 5.2 we can see that coarse blocks after first step are very large especially in the regions with low time-of-flight and some of them go all around the domain. After refining steps *step*3 and *step*4 resulting grid is with higher number of grid blocks than after *step*1 but we achieved better distribution of grid blocks specially within regions of low time-of-flight.

5.3 Error Computation

In this thesis we will use different indicators to construct coarse girds and we are interesting in the quality of the results obtained on each of these coarse grids. One of the most common measure of quality is to calculate errors of the solution on the coarse grid relative to the reference solution. Errors could be calculated for the parameters of interests such as pressure, time of flight and temperature. In order to calculate errors, parameters have to be defined in each grid cell. Since we will solve pressure equation only once on the fine scale we will calculate values for pressure and time of flight only on the fine scale grid so we cannot use this parameters as a measure of quality of coarse grid. We will use temperature as quality measure. The error we are going to use is defined using L^1 norm where as a reference solution is taken fine scale solution:

$$E_\Omega = \frac{\Sigma_{i=1}^{N_f} \mid T_{coarse} - T_{fine} \mid}{\Sigma_{i=1}^{N_f} T_{fine}} \tag{5.8}$$

where Ω is domain, N_f is total number of fine grid cells, while T_{coarse} and T_{fine} are temperatures in fine grid cell i respectively.

Chapter 6

Numerical Experiments

The main objective of this chapter is to discuss the effects of fractures on fluid flow and heat transport processes and demonstrate coarsening methods described in previous chapter. For this purpose, we will run our numerical models described in detail in chapter 4 on the fine scale grids without fractures and with fracture field. Geothermal reservoir is modeled by different grids fine scale grid and coarse scale grids obtained by coarsening with different indicator function. We will investigate which coarse grid is best approximation of geothermal reservoir i.e. which coarse grid model give us the smallest error for the advection term in heat transport equation. In this chapter we will present simulations of mathematical models for fluid flow described in chapter 2 and advection part of the heat transport equation chapter 3. Numerical methods that have been used for these simulations are described in detail in chapter 4. For code writting in this thesis we use $Matlab$ Reservoir Simulation Toolbox ($MRST$) developed by ($SINTEF$) [10]. $MRST$ is an open source sofware in use for reservoir simulation and modelling which contains core funcionality and different ad-ons modules [10]. The basic pressure and transport solvers that implement $TPFA$ model is consist in the core part of $MRST$ [18], [10]. In this thesis we will use ($MRST$) add-ons modules such as: *agglom* and DFM. We used a coarse by amalgamation module *agglom* with main characteristics that the coarse blocks are generated by amalgamating cells of the original fine grid [1], [27]. For generating fractures in this paper we use a DFM module as described above in section 4.1.2. DFM module has $TPFA$ solver and is compatible for unstructured grids[6].

$MRST$ does not have any heat transfer solver for linear advection-conduction equation thus we use the solvers created in Reservoir group of the University of Bergen. The transport solvers for advection term of heat transport equation for the fractured $3D$ grid was developed by Ivar Stefansson as a part of his master thesis see [26]. In this thesis we will use solver for implicit advec-

tion term of energy conservation equation and modeled it for a coarse grid. The solvers can simulate the advection part of the heat transport equation and return temperatures in every grid cells as a result [26].

6.1 Numerical Experiments on the Fine Scale

In this section we consider two different test cases in order to illustrate the applicability of numerical methods described in chapter 4 for pressure equation 2.9 and advection part of heat transport equation 3.8. The section is divided into three subsections: case set up, numerical results on the fine scale and discussion of numerical results. In the first subsection we will describe our two test cases with parameters that described them. In the second subsection we will give numerical results for fluid flow model and advection term of heat transport on the fine scale for test cases. In the third subsection we will give the further explanation of numerical results.

6.1.1 Case setup

Here we will describe test cases we are going to use for numerical simulation. We will construct two test cases. These test cases are very simple but at the other hand illustrative to see the difference between the processes that are run on the fractured grid and on the grid without fractures. We will further use them in the next section 6.2.

Case 1 assumes Cartesian grid with dimensions $40 \times 40 \times 30$ and with physical dimensions $80m \times 80m \times 60m$ with rock and fluid properties stated in 6.1. In further text we will also call this grid: *fine scale grid without fractures*. We use source term to implement fluid flow into and out the interior points of domain [19].

Number of grid cells	48000
Porosity	$\phi = 0.01$
Permeability	$\mathbf{K} = [1\ 1\ 1]\ mD$
Density	$\rho = 1014\ kg\ /\ m^3$
Viscosity	$\mu = 1\ centiPoise$

Table 6.1: Fine grid fluid and rock parameters. We list number of grid cells and rock porosity, isotropic permeability and fluid parameters density and viscosity.

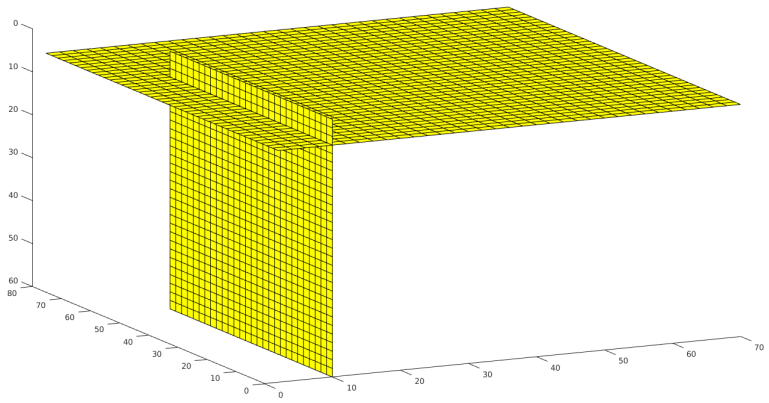


Figure 6.1: Fracture field

Case 2 is grid with the same dimensions as in Case 1, with fracture filed as in figure 6.1, where matrix and fractures properties are described in table 6.2. From 6.2 we can notice that dispute the same dimensions this grid has more cells due to the presence of fracture filed, it can be also seen that fractures are high permeable comparing to the surrounding rock. Fractures in our simulations are constructed in $MRST$ module DFM see 4.3. In DFM module fracture cells are distinguished from matrix cells by mark them as a hybrid cells [6]. DFM has predefined functions for computed transmissibility with a two-point flux approximation 4.3 modified from the original functions in core $MRST$ to include hybrid cells [6]. Here we will define initial conditions for our test cases. Source is set in the first grid cell in the left upper corner and at the last grid cell with flux rate respectively of $1m^3/Day$ and $-1m^3/Day$. We assume zero fluid flow through boundaries of domain. In the case of incompressible fluid with no boundary condition considered here flux must sum to zero to have a well posed model [19]. Let us set initial temperature of the reservoir at $1°C$ and the injection cold water temperature at $0°C$. We inject water at the upper left corner and pump the warm water at the right down corner. We assume zero heat flow through the boundaries. Next, we specify thermal conductivity to be equal to zero, thus we obtain only advection term of the equation.

Number of grid cells	49980
Porosity	$\phi = 0.01$
Matrix permeability	$\mathbf{K}_m = [1\ 1\ 1]\ mD$
Density	$\rho = 1014\ kg\ /\ m^3$
Viscosity	$\mu = 1\ centiPoise$
Fracture permeability	$\mathbf{K}_f = 10000 \times [1\ 1\ 1] mD$
Fracture aperture	$a_f = 0.001$

Table 6.2: Fractured grid. We list number of grid cells and rock porosity, isotropic permeability and fluid parameters density and viscosity, finally we list fracture properties fracture permeability and fracture aperture.

6.1.2 Numerical Results on the Fine Scale

Here we will represent numerical solutions for pressure equation and for advection term of heat transport equation for our two test cases. Numerical methods for both equations are described in detail in chapter 4. As it is described in chapter 4 we will use two different numerical methods for equations and solve them consequently. We begin with numerically solving pressure equation for the first test case. Considering that we have source term and no boundary conditions the fluid flow from the left upper corner to the right down corner. $TPFA$ method 4.2.1 is used to numerically solve pressure equation. In figure 6.2 we can see solution of pressure equation 2.9 on the fine grid. Pressure values are highest near the upper left corner and lowest near the right down corner. Time of flight is calculated using formula 2.10 these values are lowest near the upper left corner and highest on the left and the right down edges opposite to the upper left source term. We follow with simulations for Case 2.

Pressure equation 2.9 is numerically solved using predefined function for $TPFA$ method in DFM module. We compute time-of-flight in DFM module using fluxes from $TPFA$ in DFM and predefined function to account for hybrid grid cells. In figure 6.3 we can see pressure distribution and time-of-flight on fine scale fracture grid.

We will follow with numerical solution of advection part of heat transport equation. Fluxes obtained using $TPFA$ method is used in upwind discretization 4.2.2. To numerically solve advection term of heat transport equation we use *Euler's backward scheme* 4.18. As an implicit method *Euler's backward scheme* has no requirements on time step size, but it needs large computational time. We will set number of time steps to be equal to 10. In figure 6.4 we can see how the heat is transferred through the reservoir due to fluid motion at time $14e^8$ $[s]$ which is approximately 41 years. Cold water is transferred from upper left corner to the right. Temperature due to advection is passed by the motion of the fluid. Figures 6.5a and 6.5 give us temperature distribution for Case 2. Fracture field 6.5a is noticeable as two darker lines comparing to the rest of the grid, due to its lower temperature.

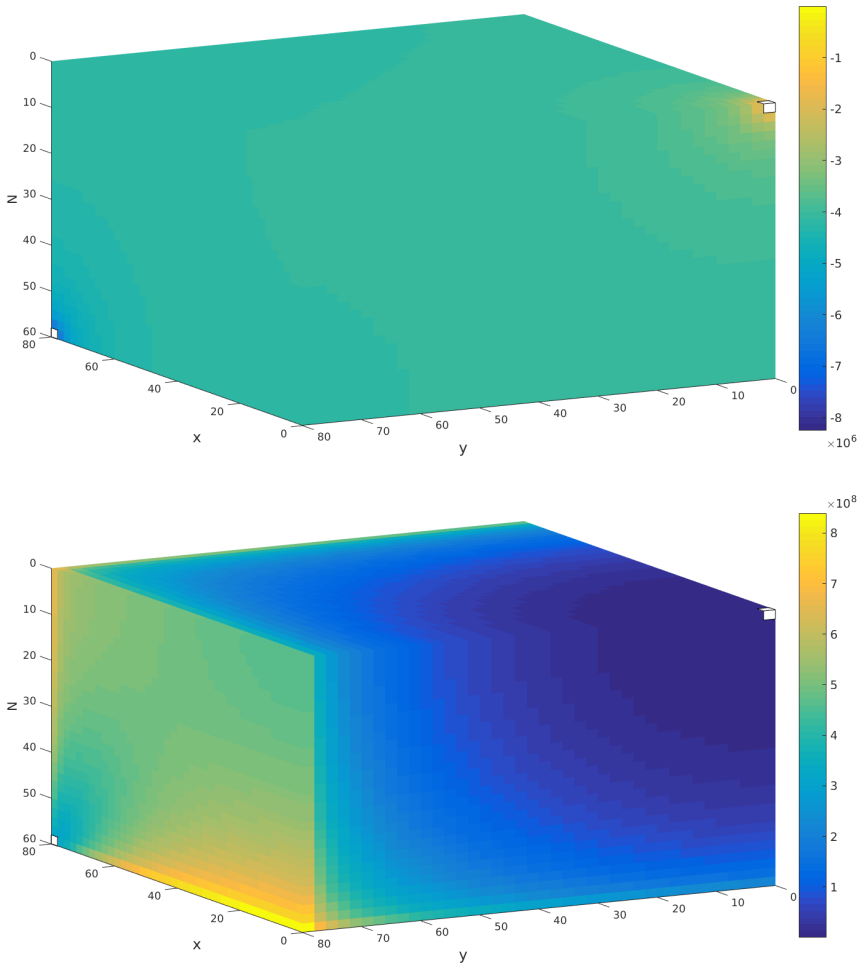


Figure 6.2: Pressure and time-of-flight distribution for Case 1, grid is rotated from left to right so we can see the back side of the grid, white cells represent source term.

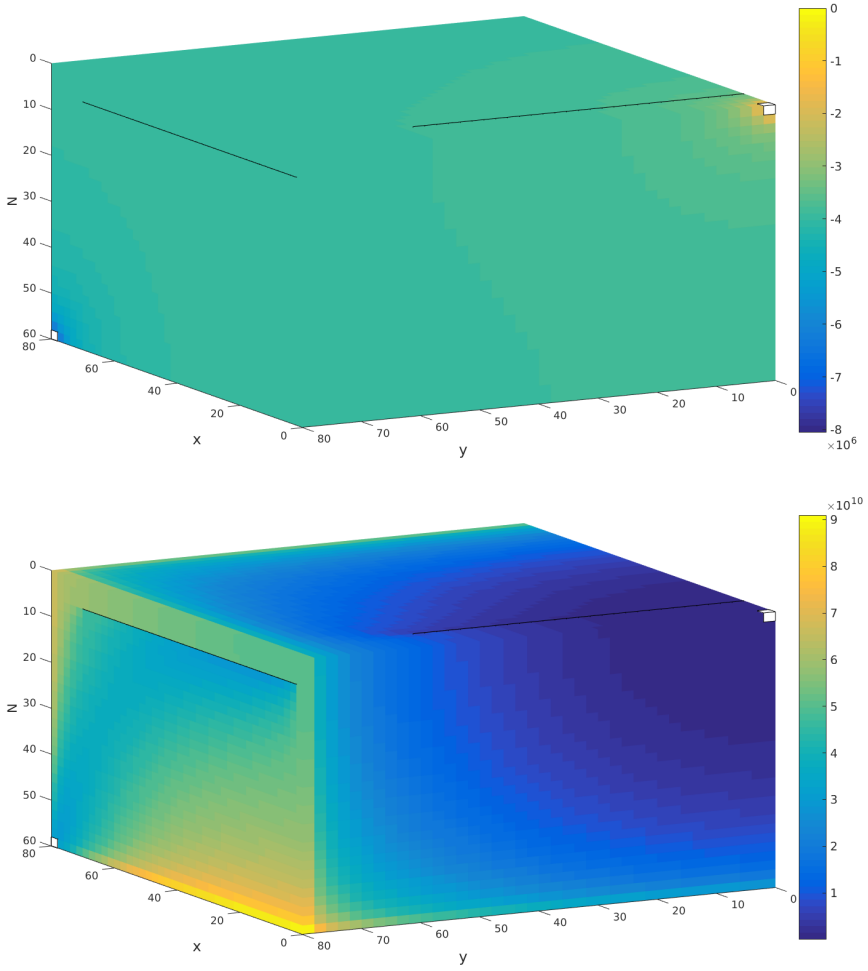


Figure 6.3: Pressure and time-of-flight distribution for Case 2, grid is rotated from left to right so we can see the back side of the grid, white cells represent source term.

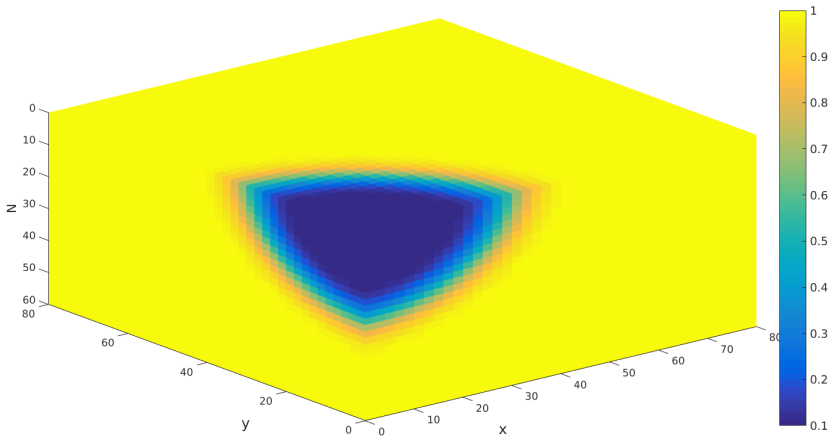
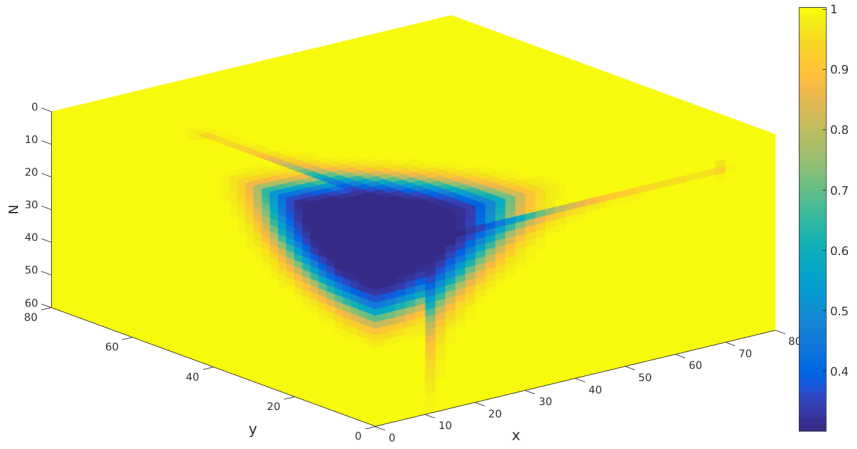
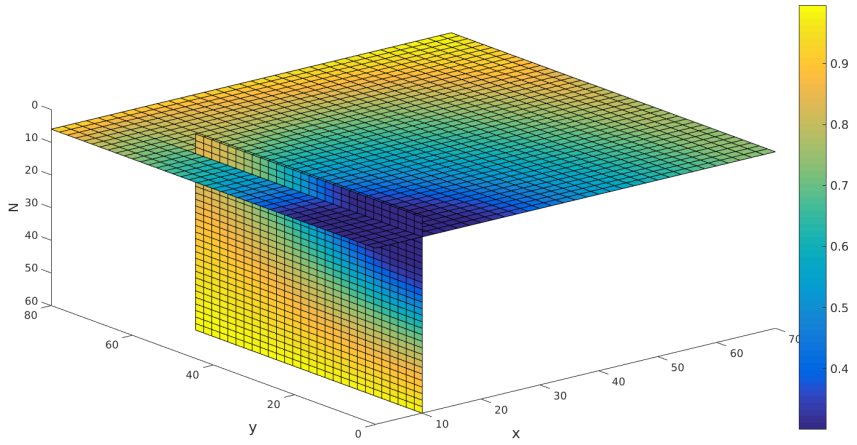


Figure 6.4: Temperature distribution for Case 1 at time $14e^8$ $[s]$.



(a) Temperature distribution for Case 2.



(b) Temperature distribution within fracture filed

Figure 6.5: Temperature distribution for Case 2 and temperature distribution within fracture field at time $14e^8$.

6.1.3 Discussion of the Numerical Results on Fine Scale

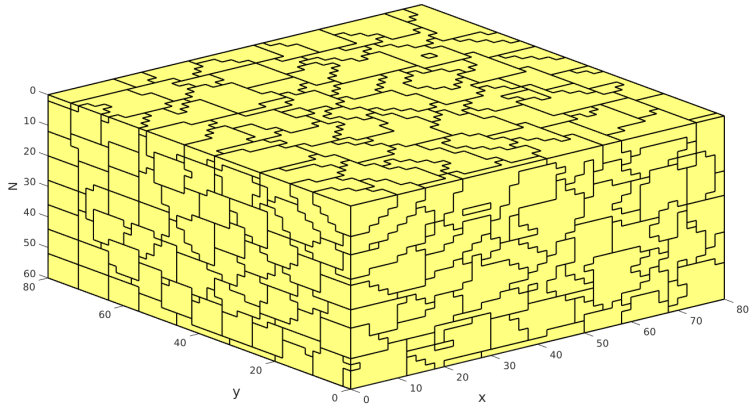
Comparing fluid flow solutions for test cases it can be noticed how time-of-flight depend on the presence of fractures. Fractures are much more permeable than the matrix, so it needs less time for fluid to flow through them and time of flight values are lower within the fracture field. In the case of fractures parallel to the $x - z$ plane, we can see that the time-of-flight values decreases under the fractures. Comparing pressure distribution for our two test cases we can see that in Case 2, pressure values are higher and that difference in pressure distribution is not that visible as in the case of time-of-flight. From figures 6.9 and 6.5a we can see that temperature distribution due to advection depends on the presence of the fractures. Fracture field is clearly visible in the second figure and temperature distribution is not symmetrical regarding diagonal from the left upper to the right down corner as in the first test case but rather translated to the fracture filed. Temperature within the fractures are lower than in the rest of the grid this is because fractures are high permeable cold water flow easier through the fractures than through the neighboring rock and heat is transported with motion of the fluid. From figures for fluid flow and heat transport for the test case without fractures and with fracture field we can see that fluid flow and heat transport depending on the presence of the high permeable fractures which makes fluid flow parameters as natural choice for coarsening indicator.

6.2 Numerical Experiments on the Coarse Scale

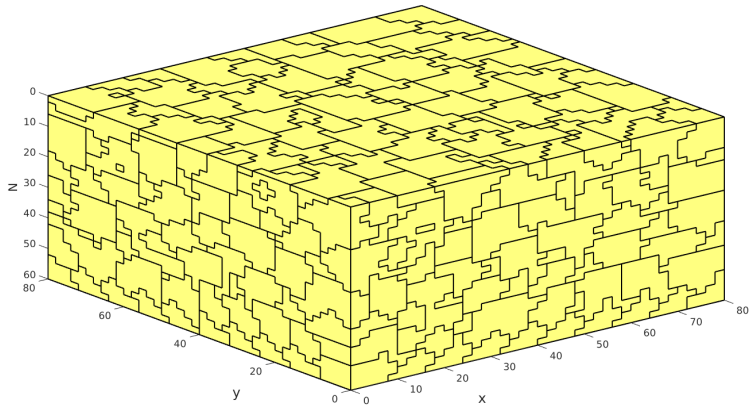
In this section we will run numerical experiments for advection term for each of the test cases defined in section 6.1 on the different coarse grids. We will close this section with discussion about the quality of the solution. The quality of the solution is measured by calculating errors using equation 5.8. In addition, computational times for different coarse grids is presented, we will compare these times with time obtained on the fine grid.

6.2.1 Coarse Grid Construction

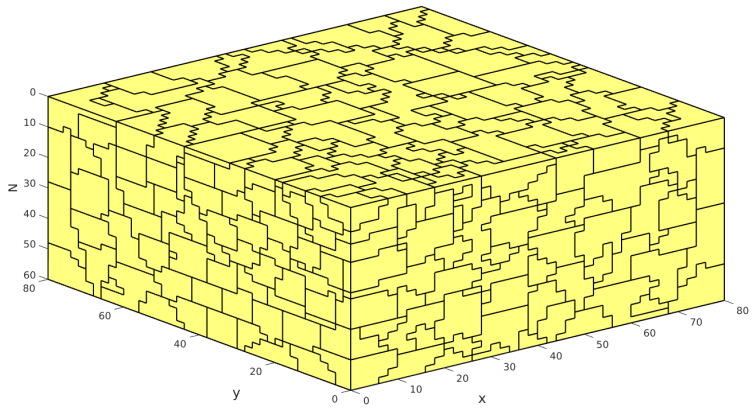
Coarse grid for Case 1 To construct three different girds for Case 1, we will use the following cell-wise indicators: time-of-flight, the, absolute permeability, the magnitude of velocity at cell center. For all indicators we use logarithmic scaling form equation 5.7. We will use $MATLAB$ module *agglom* for coarse grids construction. The resulting algorithm for non-uniform coarsening in module *agglom* is a special case of a more general algorithm described in 5.2 [1]. The main improvement in *agglom* is that the last two steps are repeated few times to get better grids [1]. Term better grid in this context mean that we will repeat steps so we get small number of the grid blocks that volatile the predefined upper and lower bounds and that resuliting grid blocks are connected. We set number of bins value as $N = 10$ and segment the indicator value into ten bins after the first step in coarsening algorithm. We need to define the lower volume bound as $N_L = 20$ and upper volume bound as $N_U = 60$. The non-uniform coarse algorithm needs to know all the neighbors of a given cell on the boundary of the block that we are going to merge in step two using the lower bound to choose which fine cells it should add to the growing block [16]. After merging the grid blocks we will use $refineGreedy2$ function for refinement of blocks in which the indicator function volatile the upper bound.



(a) Time-of-flight, 796 grid coarse grid blocks



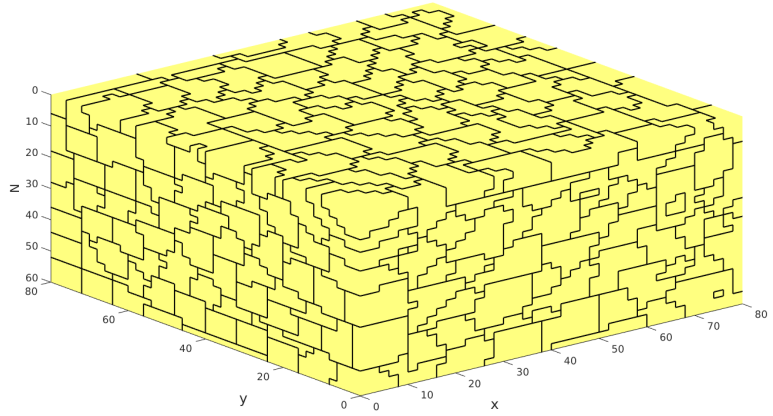
(b) Permeability, 785 coarse grid blocks



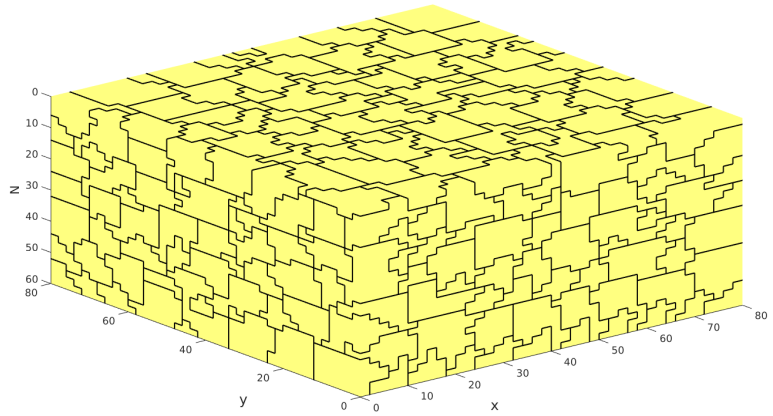
(c) Velocity, 798 coarse grid blocks

Figure 6.6: The resulting coarse scale grids for different flow parameters for Case 1

Coarse grids for Case 2. We will follow with coarse girds construction for the Case 2, since this case has fracture field, before we start with grid coarsening, we need to redefine fracture grid cells into ordinary fine grid cells. This allows us to use the same coarsening algorithm as for the Case 1. We will use the same values for N, N_L and N_U as for coarsening of fine grid without fractures. Flow indicators for Case 2 are: time-of-flight and absolute permeability. The indicators will be scaled the same way as for the Case 1 using 5.7. In addition, two new indicators will be used for Case 2 there are: square root distance and combination of square root distance and time-of-flight. By the distance we consider minimum distance to the nearest fracture. The combination of two indicators is the sum of the logarithmic scaled time-of-flight and the square root distance.

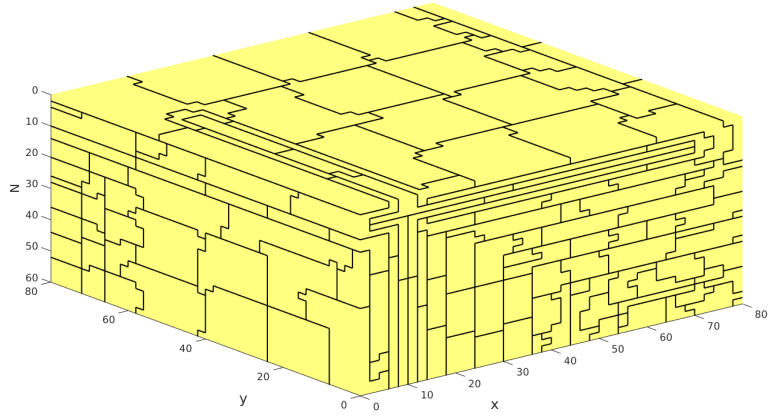


(a) Time-of-flight, 1585 grid coarse grid blocks

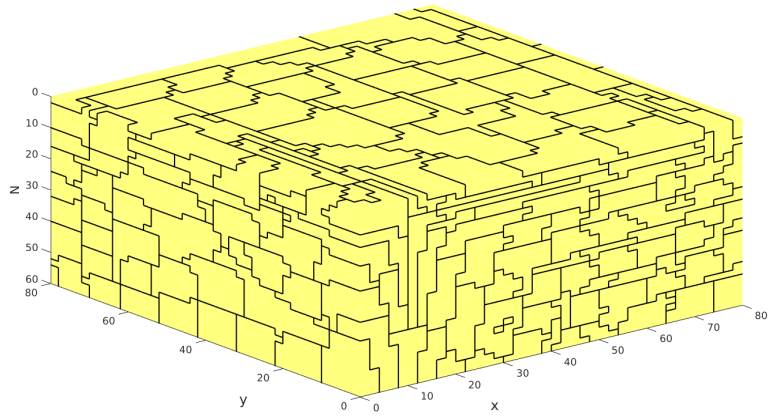


(b) Permeability, 1060 grid coarse grid blocks

Figure 6.7: Coarsening of Case 2 using different indicators respectively permeability, time-of-flight, distance to the nearest fracture, combination of the time-of-flight and distance to the nearest fracture.



(a) Square root distance, 1630 grid coarse grid blocks



(b) Combination of time-of-flight and square root distance, 2519 coarse grid blocks

Figure 6.8: Coarsening of Case 2 using square root distance and combination of time-of-flight and distance.

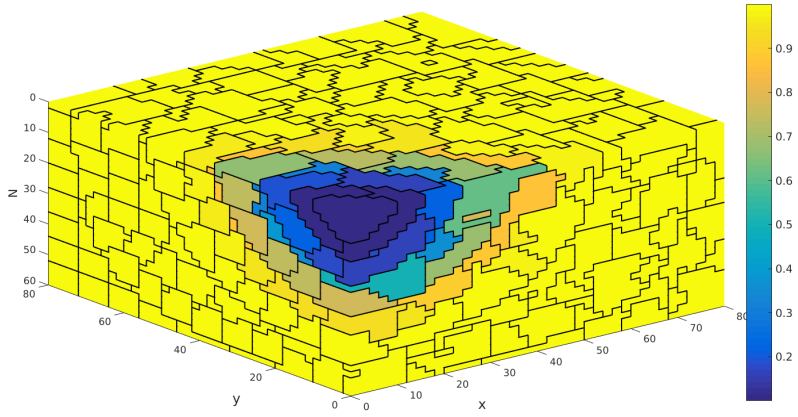
6.2.2 Numerical Results on the Coarse Grid

After a coarse grid construction for our two cases we will numerically solve advection term of heat transport equation on each of the coarse grids described in the previous subsection. As we described in chapter 4 to solve advection term of heat transport equation we need to numerically solve the pressure equation on the fine scale to get the flux values. The fluxes for Case 1 are obtained from $TPFA$ discretization of pressure equation.

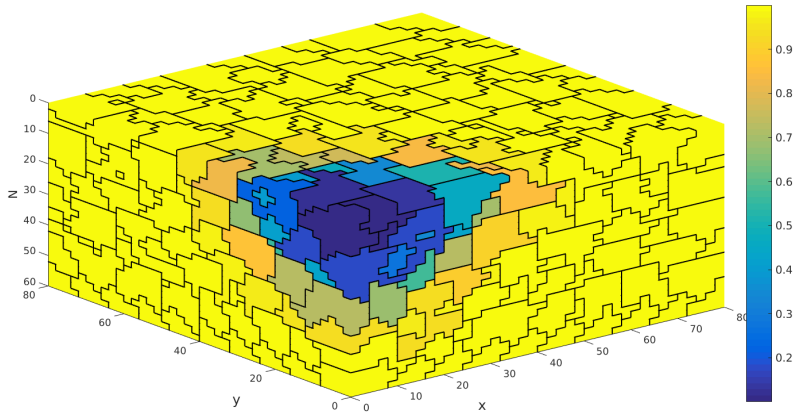
For Case 2 we will use coarse grids as described in the previous section. We will keep flux values for fracture cell to fracture cell connections after we redefined fracture cells into ordinary grid cells. Fluxes are furthered used for upscaling of advection term of heat transport equation as described in 5. Since advection term of heat transport equation is temporally depending we will use the implicit scheme *Euler's backward scheme*, as in our simulations on the fine scale, thus as a result we will obtain temperature distribution at a given time. In figure 6.9 advection term solution on the coarse grids for Case 1 are shown.

In figures 6.10 and 6.8 we can see the temperature distribution for all four coarse grids for Case 2.

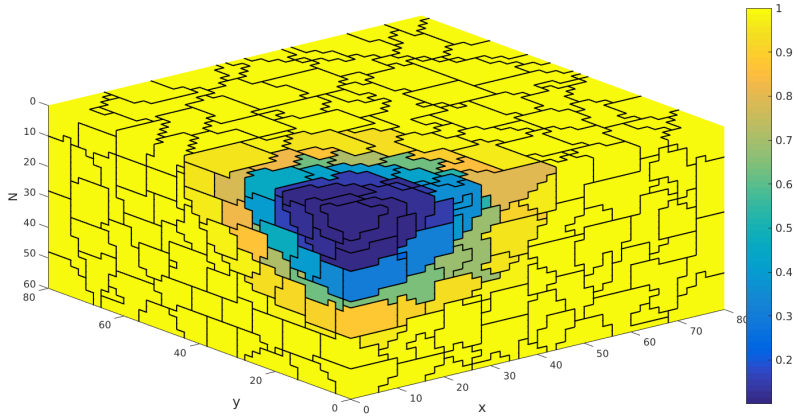
In addition we are interested in the quality of numerical results for advection term of heat transport on the coarse scale. We will calculate errors from equation 5.8. In figure 6.12a errors for three coarse grids for Case 1 are shown while in figure 6.12b we can see errors for Case 2. In table 6.3 the CPU time for different grid types for Case 1 are listed. In table 6.4 computational times for different grids for Case 2 are presented. The computations were done on the with operation system Ubuntu and with following features memory $16GiB$, processor: $Intel\ Corei5-6300U\ CPU@2.40GHz$ 4 and graphics: *Intel HD Graphics* 520. These computational times can differ depending on the computer capacity.



(a) Time-of-flight

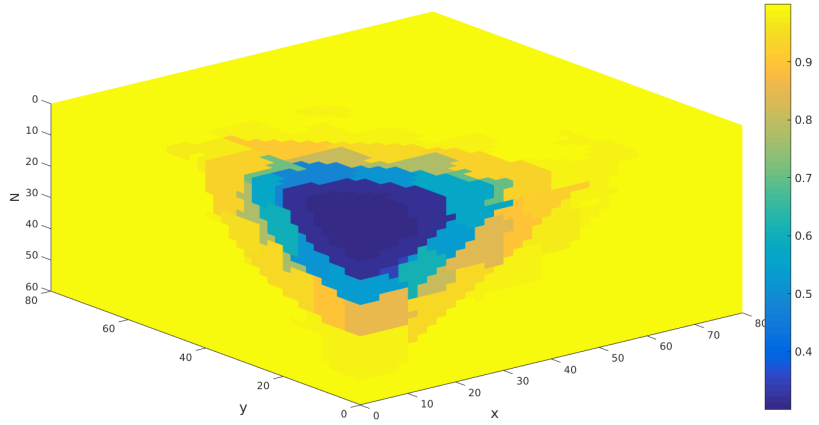


(b) Permeability

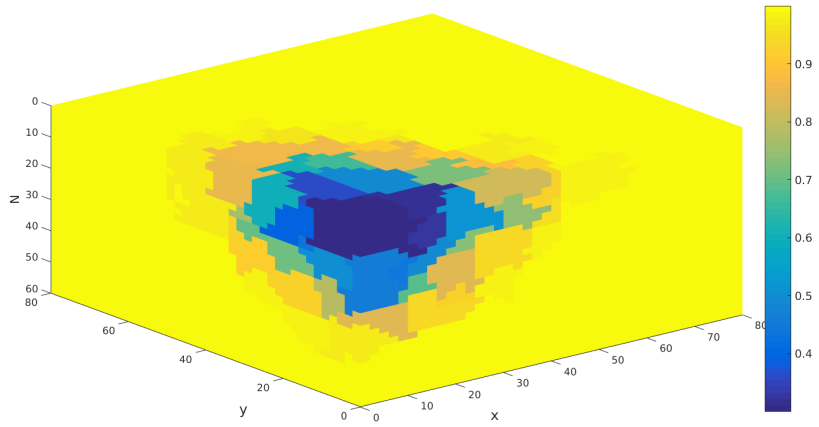


(c) Velocity

Figure 6.9: Temperature distribution for Case 1 using different coarsening indicators at time $14e8$ $[s]$

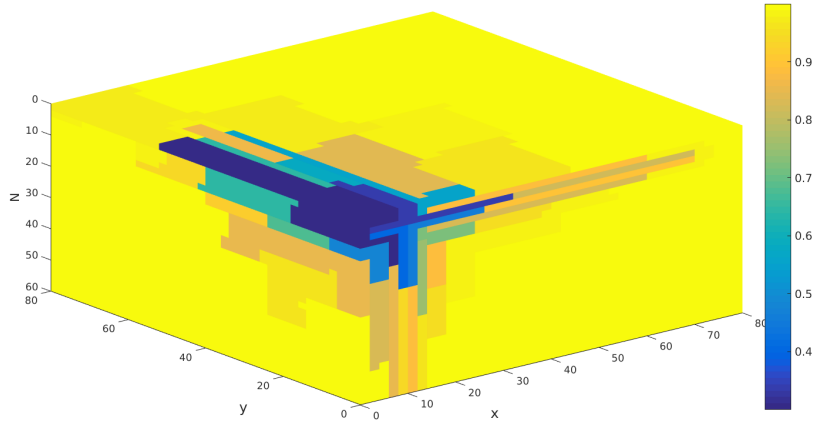


(a) Time-of-flight

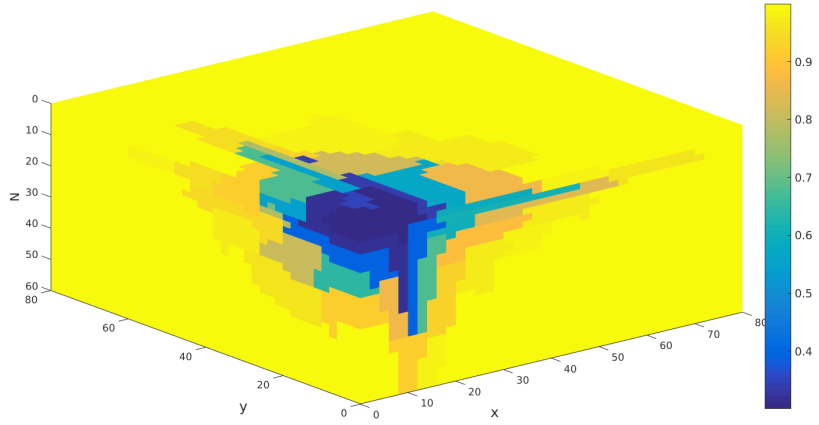


(b) Permeability

Figure 6.10: Temperature distribution for Case 2 with flow based indicators respectively time-of-flight and permeability at time $14e8$ $[s]$



(a) Square root distance



(b) Combination of time-of-flight and square root distance

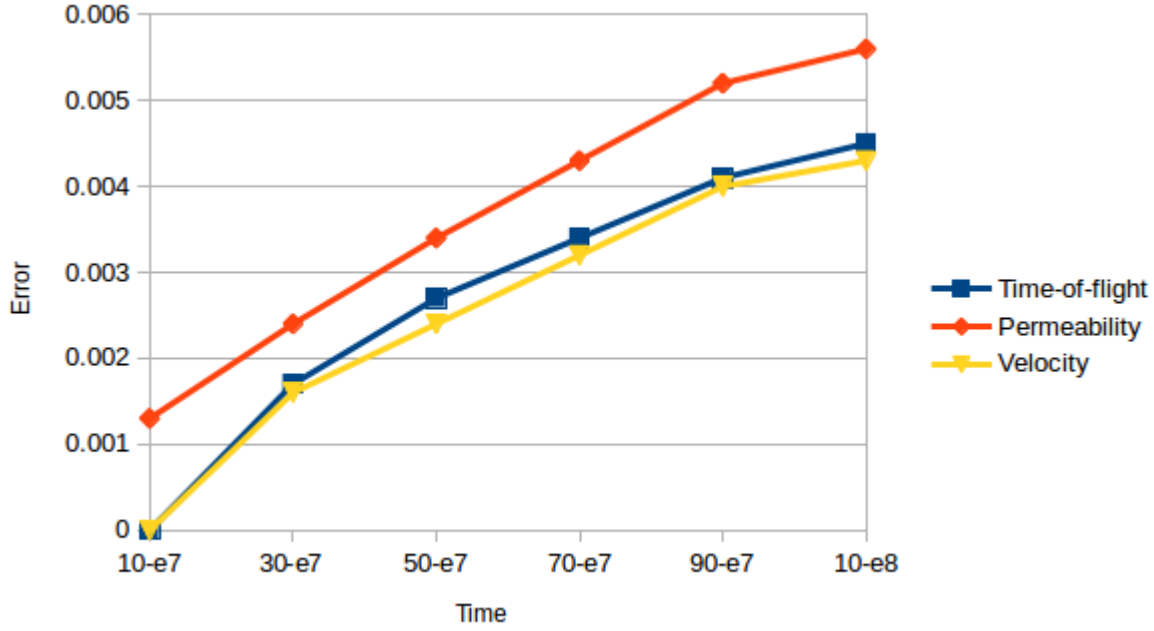
Figure 6.11: Temperature distribution on the fractured coarse grid with square root distance and combination of time-of-flight and square root distance at time $14e8$ $[s]$

Type of grid	CPU Time [s]
Fine scale	22.279658
Coarse scale time-of-flight as indicator	2.119540
Coarse scale permeability as indicator	1.679754
Coarse scale velocity as indicator	1.665713

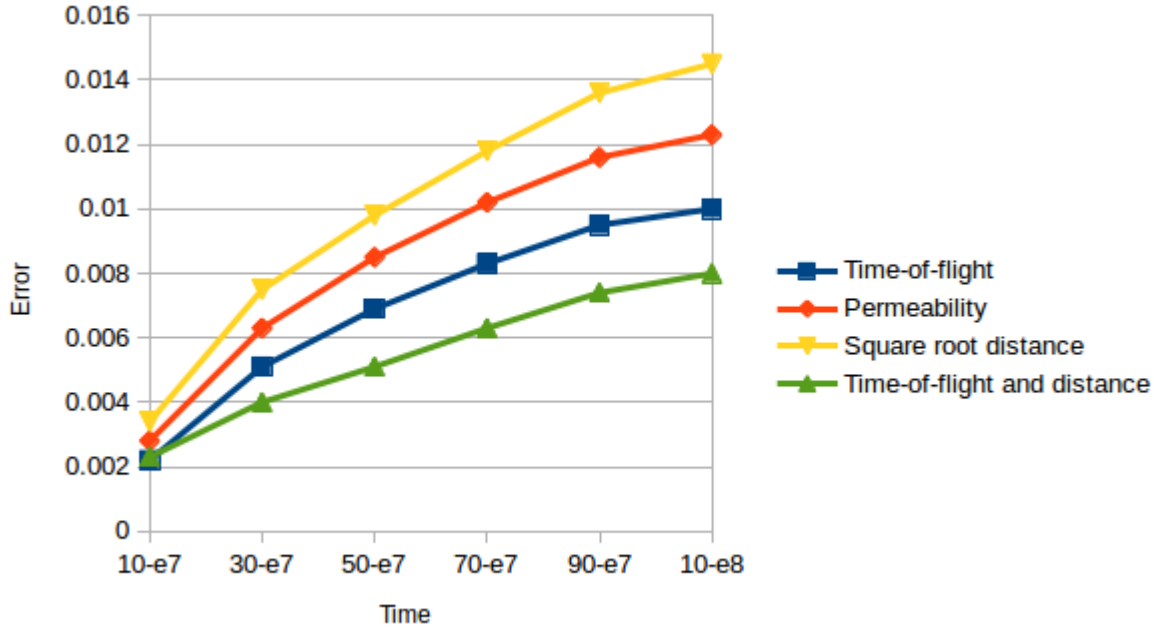
Table 6.3: Computational time on different grids for Case 1 for advection term of heat transport

Type of grid	CPU Time [s]
Fine scale grid with fractures	26.011763
Coarse scale time-of-flight as indicator	5.124017
Coarse scale permeability as indicator	5.683667
Coarse scale square root distance as indicator	5.720738
Coarse scale combination of indicators	8.083078

Table 6.4: Computational time on different grids for Case 2 for advection term of heat transport



(a) The error in L1 norm for different flow indicators for Case 1 , x-axis represents time in seconds, y-axis represents error



(b) The error in L1 norm for various coarsening indicators for Case 2, x-axis represents time in seconds, y-axis represents error

6.2.3 Discussion of Numerical Results on Coarse Scale

The aim of this subsection is to compare numerical results on the different coarse grids for Case 1 and Case 2. We will begin with comparing different coarse grids. Coarse grids are compared according the following criteria: number of grid cells, adaption to the fluid flow pattern, visibility of the fracture filed for Case 2, shape, size and distribution of gird cells.

Figure 6.6 give us the resulting coarse grids for flow indicators. The highest number of grid cells is obtained for magnitude velocity as indicator 798, than comes time-of-flight with 796 and at the end permeability has 785 blocks. Magnitude of velocity and time-of-flight comparing to permeability as indicator give coarse grids that are better adopt to flow pattern. This is since permeability is isotropic and defined values N, N_L and N_U defined the grid look. Looking closer at permeability upgridding we can see that the coarse blocks are of the roughly same size and shape, this grid remind us on the uniform grids. From figure 6.6 we can see that coarse grids have approximately the same number of blocks. From table 6.12a we can see that the samllest error has velocity with the highest number of grid cells, while the highest error has permeability with smallest number of grid cells. Since we have small differences in number of grid blocks it is not clearly if the smallest error is related to the highest number of grid bloks. In paper see [16], we can fine more about how grid quality is influenced by values N, N_L, N_U. This give us the idea that grid quality in our examples can be better if we increase number of bins, and decrease upper bound. However, this can lead to very small grid blocks in the regions of low flow [16].

In figures 6.7 and 6.8 we can see four different coarse grids for Case 2. Number of grid cells are highest for combination of square root distance and time of flight 2519, followed by square root distance with 1630 and time-of-flight with 1585 grid blocks at the end comes permeability with smallest number of grid blocks equal to 1060. Since we treat equally fracture cells with other gird cells during the grid construction we can notice that the neighboring fracture cells belong to different coarse blocks. Grid constructed with time-of-flight as indicator is well adopted to the flow pattern, we can see that it forms ring shape around the source, however the fracture field is not clearly distinguished. Permeability as indicator give us a coarse grid with grid cells almost uniformly distributed. Since we have only two values for permeability one for the rock and the other significantly higher value for fracture field we could expect that the resulting gird will give us better representation of fracture field. Unfortunately, this is not the case, the values for N, N_L and N_U, similar as for Case 1, dictate the size and distribution of gird blocks.

Square root distance indicator gives grid with clearly presented fracture field, gird cells are smaller around the fractures and bigger as we go further form them. We can also notice that some grid blocks for example grid blocks near the left-hand edge are surprisingly smaller although they are further from fracture this can be due to the effect of the upper and lower bound which tend to create smaller grid blocks in the regions with low flow volume. The main disadvantage of coarse grid constructed this way is that it is not good adopted to flow field.

Combination of two indicators gives good representation of fracture field and ring distributed coarse blocks around the upper left corner. Indicators are scaled in the way that values for logarithm scaled time-of flight goes from 1 $[s]$ to approximately 5.6 $[s]$, while the values for square root distance goes from zero m for the fractures to 7.3 m. The values for two indicators are in the same range which contribute that both fracture field and fluid flow are good represented in the coarse gird, thus combination of indicators works fine for the Case 2. Quality of indicator defined this way needs further investigation, since we do not know how it can behave if the two values are in disbalance.

In addition, we will compare temperature distribution on the different coarse grids for Case 1 and Case 2 In this purpose, we will use $L1$ norm error from 5.8. The solution in 6.4 is considered as a true solution for Case 1, while the solution 6.5a is assumed as true solution for Case 2.

From figure 6.12a for Case 1 we can see that the smallest error is obtained for coarse grid with time-of-flight as flow indicator, then for the magnitude velocity, while permeability gives the biggest error. This very similar error for time-of-flight and magnitude velocity is related to the very similar grid blocks shape, size and number, distribution of grid blocks especially around the left upper source term is also very similar. Permeability upridding give us higher error which we related to the relatively huge grid blocks around the source term. We can also notice that with time propagation the errors are bigger.

In figure 6.10 temperature distribution for Case 2 using flow indicators: time-of-flight and permeability are shown. Comparing temperature distribution with true solution we can notice that, since fracture filed is not distinguished from the rock field, in the process of grid coarsening the temperature distribution within fracture field is not good represented. Furthermore, form the resulting temperature distribution it is not possible to notice the presence of fracture field.

When we look at temperature field using square root distance as indicator in figure 6.8 we can notice that it is strongly dominated by the presence of the fractures. Fractures are clearly distinguished from the surrounding rock,

large grid blocks around the source and on the top of the grid lead to the temperature filed with lower temperatures in the rock comparing to true solution. Combination of time-of-flight and square root distance as indicator give us the temperature distribution that looks similar to temperature distribution for the true solution. This indicator is good adopted to the flow filed thanks to the time-of-flight part and give us clearly visible fractures thanks to the distance part. Since grid blocks are bigger comparing to the fine scale cells the resulting temperature distribution give us lover temperatures comparing to the true solution.

In figure 6.12b we can see that the combination of time-of-flight and square root distance give us the smallest error, time-of-flight is also very effective indicator while the square root distance alone is not good indicator. Permeability give us large error and should be avoid as coarsening indicator in the case of isotropic permeability. We can also notice that errors getting higher as times go. Comparing errors for different grids and their number of blocks we can not relate the grid quality with number of blocks for example permeability give us smaller error than square root distance, although it is grid with smaller number of grid blocks. Considering [16] we can improve grid quality by increasing the number of grid blocks.

One of the benefits of using coarse grids in our computations is faster computational time in comparison to the computational time on fine scale grid with same properties. From tables 6.3 and 6.4 we can see that computation time for coarse grids is much lower comparing to the computational time on the fine grid. This smaller computation time in combination with small errors make coarse grids as very useful for fluid flow and heat transport computations.

Chapter 7

Summary and Further Work

We have upscaled advection term of the heat transport equation and solved it for homogeneous grid without and with fracture field. In order to give better illustration of the heat transport due to advection we constructed two test cases. These cases are very simply and we use them to see the difference between fluid flow and advection term on the fine scale grid without fractures and with fracture field. As it can be seen from experiments on the fine scale, fluid flow faster through high permeable fractures and the temperature is values are lower in the fracture and its surroundings than in the rest of the rock. Further we will use these two test cases to construct coarse grids with different indicators and solve advection term of heat transport equation on each of the grids. Flow based indicators such as time-of-flight, velocity and permeability were used for upgridding of the fine scale grid without fractures. As it can be noticed time-of-flight and velocity give us a small errors for temperature distribution and a good approximation of the fine grid. Permeability is not such a good indicator thus can be because we use homogeneous medium with isotropic permeability.

Next we were focused on the construction of coarse grid upgridding the fractured fine grid. We decided to start with grid construction using flow based indicators this time time-of-flight and permeability, we decided not to use velocity because from the case of the coarse grid without fractures we see that it give us almost the same temperature distribution as time-of-flight. The main issue with upgridding fractured fine scale grid is how to threat the fracture cells. In this thesis we choose the approach to redefine fracture cells into ordinary grid cells but to keep their flux values. These flux values are used in upscaling of the advection term. However, in the grid coarsening fracture cells are add to the total number of cells and treated equally as matrix cells. Since we are interested to best represent fracture filed in our simulations we introduced square root to the nearest fracture and combination of

this distance and time of flight as new coarsening indicators. Square root distance give us good approximation of temperature distribution within the fractures but not at the rest of the grid. The grid coarsening using combination of time-of-flight and square root distance give us the best approximation in the sense of errors and in the sense of the temperature distribution in the matrix and within the fracture field. Results on the different coarse grids were compared using $L1$ norm error and computational time. Since amalgamation algorithm a wide range of choice, we can improve grid quality with different combinations of values for number of bins, upper and lower bound, and with different choices of scaling of indicators [16]. In this thesis we used very simple fracture field with only two fractures with intersection, this work can be tested on the more complex grids i.e. on the grids with more complex fracture filed. Further work can also include the effect on the conduction term on the heat transport in the fractured geothermal reservoir.

List of Symbols

Simbol	Explanation	Units
Ω	domain	
$\Gamma, \partial\Omega$	boundary of domain	
$\vec{q}_{ad}$	advective heat flux	W/m^2
$\vec{q}_{cond}$	conductive heat flux	W/m^2
$\vec{q}$	heat flux	W/m^2
κ	hydraulic conductivity	m/s
Q	source	m^3/Day
p	pressure	Pa
μ	viscosity	$Pa \cdot s$
ρ	density	kg/m^3
$\vec{g}$	gravitational acceleration	m/s^2
$\vec{v}$	Darcy's velocity	m/s
$\vec{V}$	intrinsic velocity	m/s
a_f	fracture aperture	m
l, L	length	m
e	energy desity	J/m^3
K	permeability	m^2 or mD
m	mass	kg
τ	time of flight	s
T	transmissibility	$m/(s \cdot Pa)$
T	temperature	C^0
α	half-transmissibility	$m/(s \cdot Pa)$
q_{vol}	volumetric flow rate	m^3/s
A	cross sectional area	m^2
c	specific heat capacity	$J/(kg \cdot C^0)$
V	volume	m^3
$\mathbf{k}$	thermal conductivity	$W/(m \cdot C^0)$

List of Figures

List of Tables

Bibliography

[1] Agglom: Flow and property-based coarse-grid generation — The Matlab Reservoir Simulation Toolbox 2017b documentation. [Online; accessed 26. Apr. 2018].

[2] Enhanced Geothermal System (EGS) Fact Sheet - egs_basics.pdf, Sep 2012. [Online; accessed 30. May 2018].

[3] Development of Advanced Thermal-Hydrological-Mechanical-Chemical (THMC) Modeling Capabilities for Enhanced Geothermal Systems - chemistry_wu_thmc_modeling.pdf, Feb 2014. [Online; accessed 13. Apr. 2018].

[4] Microsoft PowerPoint - Garg GRC_ReservoirEngineering_Sept2014 - Garg GRC_ReservoirEngineering_Sept2014.pdf, Nov 2014. [Online; accessed 30. May 2018].

[5] coarsegrid: Generation of coarse grids — The Matlab Reservoir Simulation Toolbox 2017b documentation, Dec 2017. [Online; accessed 25. Mar. 2018].

[6] dfm: Discrete fracture matrix implementation — The Matlab Reservoir Simulation Toolbox 2017b documentation, Dec 2017. [Online; accessed 24. Mar. 2018].

[7] 2017 Outlook: Geothermal Is Trending Upwards, May 2018. [Online; accessed 30. May 2018].

[8] Enhanced Geothermal Systems Technologies | Department of Energy, May 2018. [Online; accessed 30. May 2018].

[9] Geothermal power | Renewable energy | Statistical Review of World Energy | Energy economics | BP, Mar 2018. [Online; accessed 25. Mar. 2018].

[10] MRST, Apr 2018. [Online; accessed 26. Apr. 2018].

[11] What is a geothermal reservoir? Types of geothermal reservoirs. Institut Cartogràfic i Geològic de Catalunya, May 2018. [Online; accessed 30. May 2018].

[12] Yehuda Bachmat and Jacob Bear. On the Concept and Size of a Representative Elementary Volume (Rev). *SpringerLink*, pages 3–20, 1987.

[13] Jacob Bear. *Dynamics of fluids in porous media*. Courier Corporation, 2013.

[14] Eric Bonnet, Olivier Bour, Noelle E Odling, Philippe Davy, Ian Main, Patience Cowie, and Brian Berkowitz. Scaling of fracture systems in geological media. *Reviews of geophysics*, 39(3):347–383, 2001.

[15] Peter Dietrich, Rainer Helmig, Martin Sauter, Heinz Hötzl, Jürgen Köngeter, and Georg Teutsch. *Flow and transport in fractured porous media*. Springer Science & Business Media, 2005.

[16] Vera Louise Hauge. Multiscale methods and flow-based gridding for flow and transport in porous media. 2010.

[17] Vera Louise Hauge, Knut-Andreas Lie, and Jostein R Natvig. Flow-based coarsening for multiscale simulation of transport in porous media. *Computational Geosciences*, 16(2):391–408, 2012.

[18] Victor Lampe. Modelling fluid flow and heat transport in fractured porous media. Master's thesis, Department of Mathematics University of Bergen, 2013.

[19] Knut-Andreas Lie. An introduction to reservoir simulation using matlab: user guide for the matlab reservoir simulation toolbox (mrst). sintef ict, 2014.

[20] Jostein R Natvig, Knut-Andreas Lie, Birgitte Eikemo, and Inga Berre. An efficient discontinuous galerkin method for advective transport in porous media. *Advances in water resources*, 30(12):2424–2438, 2007.

[21] Anette Cathrine Nedreli. Transport upscaling in porous media. Master's thesis, The University of Bergen, 2014.

[22] Anna Nissen, Eirik Keilegavlen, Tor Harald Sandve, Inga Berre, and Jan Martin Nordbotten. Heterogeneity preserving upscaling for heat transport in fractured geothermal reservoirs. *arXiv preprint arXiv:1702.06074*, 2017.

[23] Jan Martin Nordbotten and Michael A Celia. *Geological storage of CO2: modeling approaches for large-scale simulation.* John Wiley & Sons, 2011.

[24] TH Sandve, I Berre, and JM Nordbotten. An efficient multi-point flux approximation method for discrete fracture–matrix simulations. *Journal of Computational Physics*, 9(231):3784–3800, 2012.

[25] Sudipta Sarkar, M Nafi Toksoz, and Daniel R Burns. Fluid flow simulation in fractured reservoirs. Technical report, Massachusetts Institute of Technology. Earth Resources Laboratory, 2002.

[26] Ivar Stefansson. A comparison of two numerical models for flow in fractured porous media and the impact of fracture intersection cell removal. Master's thesis, Department of Mathematics University of Bergen, 2016.

[27] Anna Varzina. Heat transfer upscaling in geothermal reservoirs. Master's thesis, Department of Mathematics University of Bergen, 2015.

[28] M. H. N. Yio, H. S. Wong, and N. R. Buenfeld. Representative elementary volume (REV) of cementitious materials from three-dimensional pore structure analysis. *Cem. Concr. Res.*, 102:187–202, Dec 2017.