POLITECNICO DI MILANO
Scuola di Ingegneria Industriale e dell'Informazione
Corso di Laurea Magistrale in Ingegneria Matematica
Dipartimento di Matematica "F. Brioschi"

3D Upscaling of Reservoir Properties using the Mixed Finite Element Method on Non-Matching Grids

Relatore: Dott. Anna SCOTTI
Correlatori: Prof. Luca FORMAGGIA
Dott. Paola PANFILI

Tesi di Laurea di:
Calogero B. RIZZO, matricola 782422

Anno Accademico 2012-2013

Contents

List of Figures

Abstract

In this thesis we have developed methodology and a software for the upscaling of a three-dimensional oil reservoir model. The upscaling technique is useful to reduce the discrete model dimensions: given a fine grid where the permeability is defined for each cell, the goal we have set is to compute geological properties such as permeability or transmissibility on a coarser grid. To do this, we use the so called *flow-based* upscaling where we need to solve some local problems at fine grid scale: we use a mixed finite element method (MFEM) for the problem approximation. We focus in particular on the cases in which the fine grid and the coarse grid are non-matching, meaning that cells faces of the coarse grid are not aligned with the fine grid. We have implemented ad hoc discrete integration techniques to minimize the approximation error. Finally we have tested the software toward the behaviours using different kind of coarse grids or different fine scale properties. We have also analysed some example of real reservoir models in conjunction with the reservoir modelling department (MOGI) of Eni spa.

Sommario

In questa tesi abbiamo sviluppato un software per l'upscaling delle proprietà di un modello tridimensionale di un giacimento petrolifero. La procedura di upscaling è utile per ridurre le dimensioni del modello discreto: data una griglia fine dove è definita la permeabilità in ogni cella e data una griglia più lasca, l'obiettivo che ci prefiggiamo è di calcolare le proprietà geologiche quali trasmissibilità e permeabilità su quest'ultima griglia. Per fare questo, utilizziamo delle tecniche chiamate flow-based dove avremo bisogno di risolvere dei problemi locali sulla scala della griglia fine: useremo il metodo degli elementi finiti misti (MFEM) per discretizzare il problema. Particolare attenzione è data al caso in cui la griglia fine e la griglia coarse sono non matching, ovvero che le facce delle celle di una non sono allineate con l'altra. Per questo motivo sono state implementate tecniche di integrazione discrete che minimizzano l'errore commesso. Infine abbiamo fatto dei test di verifica su come il metodo implementato si comporta al variare delle proprietà sulla scala fine e la geometria della griglia lasca. Sono stati analizzati anche esempi di giacimenti realistici in collaborazione con il dipartimento per la modellizzazione di giacimenti (MOGI) di Eni spa.

Introduction

The continuous development of modern processors in the last decades provides us with an incredible resource in terms of computing power. Simulation of physical processes is at present used in almost all engineering fields for feasibility studies or procedures where a direct study would be either to much expensive or impossible. However, even using the most modern computers too, there are still problems in which a direct calculation is unaffordable since they require a huge amount of data to be processed.

Since oil stocks decrease each year while there is a continuous increment of the energy demand, it is fundamental to have an optimized plan for reservoirs production. A reliable model of field properties is therefore mandatory. Since an oil reservoir could cover a big area, a full detailed model is often too expensive for practical purposes. Moreover, for feasibility studies we would need several simulations using different parameters such as wells positions or field conditions.

Many studies have been done trying to simplify a reservoir model while maintaining a good accuracy. For example, one strategy would be to solve the fluid flow problem into the reservoir domain using a coarser grid. Then the problem would be how to compute the field properties of the new mesh starting from the fine one in a way that is consistent with the physical process. This problem is the so called *upscaling* problem.

The purpose of this thesis is to implement a standalone three-dimensional upscaling software based on `C++`: starting from the works already existing in the literature we aim at generalizing the approach for generic fine and coarse grids. In particular we concentrate on the case of non-matching coarse grids, namely grids in which faces are not aligned with those of the fine grid.

The final part of the work has been done in cooperation with the Reservoir Modelling department (MOGI) of Eni (Ente nazionale idrocarburi). The upscaling technique developed in this work has indeed been applied to some real test cases and the results have been used to perform coarse scale simulations using one of the most widespread industrial reservoir simulation software.

The thesis structure is arranged as follow:

Chapter 1. In this introductory chapter we present principal techniques used in oil production and the physics governing fluid flow in porous media. Furthermore we will describe the general equations that are fundamental for reservoir simulation.

Chapter 2. We explore the state of the art of the upscaling problem, discussing the major issues. After a brief overview of various upscaling techniques, we define the procedures needed to find the field properties on a coarse grid, presenting the available features to enforce the upscaling quality.

Chapter 3. This chapter is dedicated to the analysis of the discretization scheme adopted to solve the fluid flow in porous media equations during the upscaling which is the Mixed Finite Element Method (MFEM) that ensures the local mass conservation. We present some theoretical results on the convergence for MFEM approximation. Finally we reformulate the fluid flow in porous media equations as a generic saddle-point problem and a proper MFEM approximation.

Chapter 4. Algorithms and techniques characterizing the procedure and the software developed in this work are presented in this chapter. Particular attention is given to the MFEM implementation and to the geometric operations, interpolations and quadrature needed to operate with non-matching grids.

Chapter 5. In the last chapter we test the developed software running real reservoir simulations. We study the solution behaviour varying the principal upscaling parameters presented in Chapter 2. We focus on non-matching upscaling techniques building coarse grids with various kinds of displacements with respect to the fine grid to assess the robustness of the proposed strategy.

The last section is devoted to conclusions.

Chapter 1

Reservoir Simulation

Simulation of oil and gas reservoir consists in the construction of a model that represents the behaviour of a real reservoir. A mathematical model is a set of equations that, subject to certain assumptions, describes the physical processes active in the reservoir. The approximation errors that a simulation could have are mainly of two different types: errors due to physical approximations, embedded into equations (that usually are formulated as Ordinary Differential Equations (ODE) or Partial Differential Equations (PDE) for more complex problems) and errors due to model approximations, generated by the discretization of the general equations. In this chapter we will focus to the physical approximations related to the mathematical models that rule the oil recovery process.

1.1 Oil Recovery Process

After a petroleum reservoir has been located, several oil wells are created by drilling long holes into the upper sedimentary layers with an oil rig. A steel pipe is placed in the hole, to provide structural integrity to the newly drilled well bore. Holes are then made at the base of the well to enable the oil to pass into the bore. Finally a collection of valves serves the purpose of regulating pressures and controlling flows.

Then, the oil recovery process, in which oil is extracted, starts. The whole process can be divided into three stages:

Primary recovery: this first stage happens once a well has been drilled through the underground field. Natural driving forces like gravity drainage, expansion of natural gas and the pressure inside the reservoir push the oil into the well bore. Then the oil is brought to the surface through mechanical

Figure 1.1: A pumpjack in Texas: a pumpjack is an onshore overground drive for a piston pump in an oil well.

means. The primary phase of oil recovery continues until the pressure inside the well is no longer high enough to produce oil in quantities that make it financially worthwhile. Recovery factor, that is the fraction of oil extracted from the reservoir, during the primary stage is typically 5-15%.

Secondary recovery (or *water flooding*): water flooding is the use of water injection to increase the production from oil reservoirs. During this stage there are two different kind of wells: the injection wells (where water is injected) and the production wells (where oil is recovered). This is accomplished by increasing the reservoir pressure back to its initial level and maintaining it close to that value. Another key factor that drives water flooding development and increase of use is water availability in large quantities from nearby streams, rivers or oceans. Water injection effectively makes production wells that are near the water-injection wells flow or be pumped at higher rates because of the increased reservoir pressure The water displaces oil from the pore spaces, but the efficiency of such displacement depends on many factors (e.g., oil viscosity and rock characteristics). At this stage the flow can be of two different types: two-phase immiscible flow or black-oil flow. Two-phase flow without mass transfer between the phases occurs when the reservoir pressure is above the bubble point pressure of the oil phase. In this case one phase is water and the other one is oil. On the other hand if the reservoir pressure drops below the bubble point pressure, then the hydrocarbon phase is split into a liquid phase and a gaseous phase in thermodynamic equilibrium. The water phase does not exchange mass with the other phases, but the liquid and gaseous phase exchange mass with each other. Eventually, water flooding is no longer worthwhile since a significant ratio of water flows through the production wells. After this stage, around 50% of oil will still be in the reservoir. For further informations about secondary recover we refer to three significant books written by Craig [11], Willhite [34] and Rose et al. [31] that address water flooding technology.

Tertiary recovery (or *enhanced oil recovery* - EOR): although more expensive to employ on a field, enhanced oil recovery can increase recovery rate from a well up to 75%. Although it is used after both primary and secondary recovery have been exhausted, EOR restores formation pressure and enhances oil displacement in the reservoir. There are three main types of EOR: chemical flooding, gas injection and thermal recovery. Each field must be carefully evaluated to determine which type of EOR will work best on the reservoir.

1.2 The Mathematical Model

The purpose of simulation is to find the optimal production plan to maximize the field performance (e.g., oil recovery): whereas the field can be exploited only once, at considerable expense, a mathematical model can be run many times at low expense in a short period of time. Observation and evaluation of model results representing different conditions, for example different wells location, supports the selection of the producing plan that best suits the reservoir with the consequent increment in production.

1.2.1 Definitions and considerations

In this chapter, fundamental definitions concerning the problem of interest are presented [4].

A *porous medium* body is composed by a *solid matrix* containing *void spaces* that can be filled by liquids or gas. An easy to see example of porous medium is a sponge. In particular, a reservoir is composed by porous medium and oil and water are contained into the void spaces. A *phase* is defined as a chemically homogeneous portion of a system that is separated from other such portions by a definite physical boundary. In the case of

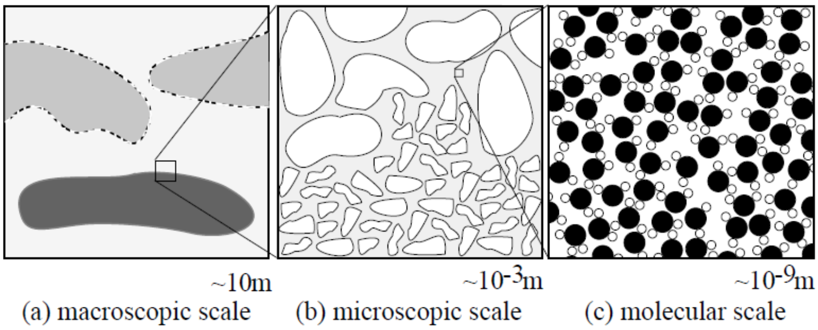


Figure 1.2: Representation of macroscopic, microscopic and molecular scales.

a single-phase system the void space of the porous medium is filled by a single fluid (e.g., water) or by several fluids completely miscible with each other (e.g., fresh water and salt water). In a multiphase system the void space is filled by two or more fluids that are immiscible with each other, in reservoir they usually are water and oil. There may only be one gaseous phase since gases are always completely miscible. Formally the solid matrix of the porous medium can also be considered as a phase called the solid phase. A *component* represents a part of a phase that is composed of an identifiable homogeneous chemical species or of an assembly of species (ions, molecules). In the case of a fresh water and salt water mixture we would have a single-phase with two components. The last physical definition is the *mean free path length* of a fluid, that is the average distance a molecule travels between successive collisions with other molecules.

In order to derive mathematical models for fluid flow in a porous medium there are some crucial properties that have to be taken into account:

P1 The void space of the porous medium is interconnected.

P2 The dimensions of the void space must be large compared to the mean free path length of the fluid molecules.

P3 The dimensions of the void space must be small enough so that the fluid flow is controlled by adhesive forces at fluid-solid interfaces and cohesive forces at fluid-fluid interfaces (multiphase systems).

The first property ensures the connection between two arbitrary points of the domain, so mass can flow through it. The second one allow us to build a continuous model into the void spaces. Finally, the third condition discards particular cases, like pipe networks, from the definition of porous medium.

A typical reservoir model is characterized by different behaviour at different scale lengths, as shown in Fig. 1.2. The *macroscopic* scale, that is on the order of 10m, catches all the ground properties due to sedimentation and related macroscopic features. Then, there is the *microscopic* scale representing all sand grains and void channels of the materials. This scale is on the order of 10^{-3}m. Finally, the *molecular* scale appear on the order of 10^{-6}. Macroscopic, microscopic and molecular scales influence the fluid flux in different ways and it is crucial to catch all the fluid dynamics. For example, all fluid properties are due to the molecular compositions while the void channels width defines the quantity of fluid that could pass through a given body.

In physics, the Navier-Stokes equations, developed by Claude-Louis Navier and George Gabriel Stokes, describe the motion of a viscous Newtonian fluid. For a constant density fluid in a domain $\Omega \subset \mathbb{R}^d$ (with $d = 2, 3$), they are:

$$\begin{cases} \dfrac{\partial \mathbf{u}}{\partial t} - \nabla \cdot \left[\nu \left(\nabla \mathbf{u} + \nabla \mathbf{u}^T\right)\right] + (\mathbf{u} \cdot \nabla)\mathbf{u} + \nabla p = \mathbf{f} & \mathbf{x} \in \Omega, t > 0 \\ \\ \nabla \cdot \mathbf{u} = 0 & \mathbf{x} \in \Omega, t > 0 \end{cases} \tag{1.1}$$

where $\mathbf{u}$ is the velocity, ρ the density, p the pressure divided by the density, μ is the dynamic viscosity, $\nu = \frac{\mu}{\rho}$ is the kinematic viscosity and $\mathbf{f}$ is the forcing term for mass unit. The Navier-Stokes equations (1.1) model the microscopic scale features of the fluid, that are represented by ρ, ν and μ.

To solve a flux problem through a porous medium, the equations (1.1) could be used, but a proper discretization is needed in order to catch both macroscopic and microscopic scales. That said, the discretization step has to be smaller than the void channels length scale, so for each axes there would be a number of intervals N on the order of the ratio between macroscopic and microscopic scales, that means $N \approx \frac{10}{10^{-3}} = 10^4$ intervals. For a generic 3D problem we would have $N^3 \approx 10^{12}$ cells that is, even with modern processors, computationally too expensive. Furthermore, in real cases the domain dimension could be of the order of some kilometres, therefore this approach is unaffordable. Even with an adequate computational power, there would still be the issue of how to model the geological variables down to the microscopical scale. All these problems motivate the development of approximated equations for flux in porous medium.

1.2.2 Fluid Mass Conservation

The main goal of a good approximation is to represent the microscopic geological features with a macroscopic continuous variable. Given a domain

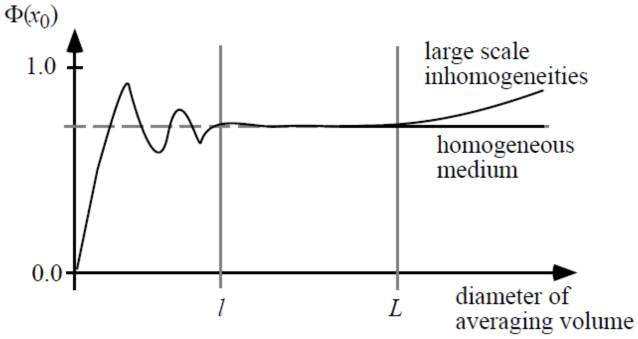


Figure 1.3: Porosity for different sizes of averaging volume.

Ω, we define the void space indicator function as:

$$\gamma(\mathbf{x}) = \begin{cases} 1 & \mathbf{x} \in \text{void space} \\ 0 & \mathbf{x} \in \text{solid matrix} \end{cases} \quad \forall \mathbf{x} \in \Omega \tag{1.2}$$

Then we define the macroscopic quantity *porosity* as:

$$\Phi(\mathbf{x}_0) = \frac{1}{|\Omega_0(\mathbf{x}_0)|} \int_{\Omega_0(\mathbf{x}_0)} \gamma(\mathbf{x}) d\mathbf{x} \tag{1.3}$$

where $\mathbf{x}_0$ is a fixed point and $\Omega_0(\mathbf{x}_0)$ is a subdomain of Ω centered in x_0. In other words, the macroscopic quantity porosity is computed as the average of microscopic quantity γ. If the diameter d of $\Omega_0(\mathbf{x}_0)$ is smaller than the mean free path length l, the discontinuous pattern of void channels induces a strong variation of $\Phi(\mathbf{x}_0)$, while for bigger values of d (e.g. larger than a given length L), macroscopic patterns such like layers of different materials lead again to a variation of $\Phi(\mathbf{x}_0)$. The averaging volume $\Omega_0(\mathbf{x}_0)$ is called a representative elementary volume (REV) if two length scales l and L can be identified as in Figure 1.3 such that the value of the averaged quantity does not depend on the size of the averaging volume. In that case we can choose the averaging volume anywhere in the range:

$$l(x_0) \ll diam(\Omega_0(\mathbf{x}_0)) \ll L(x_0) \tag{1.4}$$

More informations about averaging of microscopic properties can be found in [4, 22].

Finally, the macroscopic fluid mass conservation law can be written as:

$$\frac{\partial(\Phi\rho)}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = \tilde{m} \tag{1.5}$$

where $\tilde{m}$ is the source/sink term expressed as a mass flow rate per unit volume.

The main difference between (1.5) and the fluid mass conservation in Navier-Stokes equations (1.1) is the velocity $\mathbf{u}$: in fact, in (1.5), the $\mathbf{u}$ is an *apparent* macroscopic velocity taking in consideration the microscopic features. In particular, the mean real velocity through void channels is equal to $\frac{\mathbf{u}}{\Phi}$.

1.2.3 Darcy's Law

Darcy's law is the mathematical equation that describes the flow of a fluid through a porous medium as defined in 1.2.1. The law was firstly formulated by Henry Darcy based on the results of experiments on the flow of water through beds of sand [12]. The Darcy's law can be also derived by homogenization techniques [23] from the momentum equations in (1.1) and it can be written as:

$$\mathbf{u} = -\frac{\mathbf{K}}{\mu}(\nabla p - \rho \mathbf{g}) \tag{1.6}$$

where $\mathbf{g}$ is the gravity vector and $\mathbf{K}$ is the symmetric tensor of *absolute permeability* or *hydraulic conductivity*. It depends on the properties of the solid matrix only and may depend on the position in the case of a heterogeneous porous medium. Furthermore $\mathbf{K}$ may be anisotropic if the porous medium has a preferred flow direction. If we replace $\mathbf{u}$ in (1.5) with (1.6) we obtain an equation with p as the only unknown quantity:

$$\frac{\partial(\Phi\rho)}{\partial t} - \nabla \cdot (\rho \frac{\mathbf{K}}{\mu}(\nabla p - \rho \mathbf{g})) = \tilde{m}, \tag{1.7}$$

to be completed with proper bounding conditions. Equation (1.7) is parabolic. In case of constant density (1.7) become elliptic and it can be rewritten as

$$-\nabla \cdot (\frac{\mathbf{K}}{\mu}(\nabla p - \rho \mathbf{g})) = \tilde{q}, \tag{1.8}$$

where $\tilde{q}$ is the volumetric source term.

Unlike (1.1), in order to solve equation (1.8) we do not need fine discretization since the microscopic features are incorporated into the quantities Φ and $\mathbf{K}$ while molecular features, as in Navier-Stokes equations, are modelled by ρ and μ.

1.2.4 Two-Phase Flow Model

The equations describing two-phase flow on the fine scale can be derived by combining Darcy's law with a statement of mass conservation. In this case, in the absence of gravity, Darcy's law can be expressed as

$$\mathbf{u}_j = -\frac{k_{rj}}{\mu_j}\mathbf{K}\nabla p_j, \tag{1.9}$$

where the index j refers to the phase (e.g. $j = w$ for water and $j = 0$ for oil) and k_{rj} is the relative permeability to phase j. Mass conservation is given by

$$\frac{\partial(\phi\rho_j S_j)}{\partial t} + \nabla \cdot (\rho_j \mathbf{u}_j) = \tilde{m}_j, \tag{1.10}$$

where S_j is the saturation (volume fraction) of phase j. If we assume that $\partial\phi/\partial t = 0$, that ρ_j does not vary in time or space, and that capillary pressure (p_c) is negligible (i.e. $p_c(S_w) = p_o - p_w = 0$), we obtain:

$$\nabla \cdot \mathbf{u}_t = \tilde{q}_t, \tag{1.11}$$

where $\tilde{q}_j = \tilde{m}_j/\rho_j$ and the total volumetric source term is $\tilde{q}_t = \tilde{q}_w + \tilde{q}_o$. The total Darcy velocity $\mathbf{u}_t$ is given by:

$$\mathbf{u}_t = \mathbf{u}_w + \mathbf{u}_o = -\left(\frac{k_{rw}}{\mu_w} + \frac{k_{ro}}{\mu_o}\right)\mathbf{K}\nabla p. \tag{1.12}$$

The water velocity $\mathbf{u}_w$ can now be expressed as $\mathbf{u}_w = f(S_w)\mathbf{u}_t$ where $f(S_w)$ is the Buckley-Leverett fractional flow function (see [6]). Inserting this form for $\mathbf{u}_w$ in Equation (1.10), we can write the water saturation equation as:

$$\phi\frac{\partial S_w}{\partial t} + \nabla \cdot [\mathbf{u}_t f(S_w)] = \tilde{q}_w. \tag{1.13}$$

The corresponding pressure equation can be written by inserting Equation (1.12) into Equation (1.11):

$$-\nabla \cdot (\lambda_t(S_w)\mathbf{K}\nabla p) = \tilde{q}_t \tag{1.14}$$

where we have introduced the total mobility λ_t, defined as

$$\lambda_t = \lambda_w + \lambda_o = \frac{k_{rw}}{\mu_w} + \frac{k_{ro}}{\mu_o}. \tag{1.15}$$

The pressure and saturation equations describe the flow of two immiscible fluids. The pressure equation (1.14) is very similar to the single-phase pressure equation (1.8) except the λ_t term replaces $1/\mu$. The single-phase limit is recovered if the two phases have identical properties and do not interfere.

Chapter 2

Upscaling of Geological Models

Typical reservoir simulators are usually designed to handle a large number of computational cells, on the order of millions. The size of the problem can vary considerably depending on the type of simulation to be performed and the available computer hardware. Geological characterizations, by contrast, is typically much more detailed being described over a finer mesh. These models, which are referred to as fine grid models, geostatistical models or simply geocellular models, represent geological variation on very fine scales vertically, though their areal resolution is still relatively coarse. For example, a typical geostatistical model might contain layering of thickness of 1 meter or less, though cell sizes in the horizontal direction might vary from 15 to 30 meters. Another issue of considerable importance is the need for the assessment of risk and uncertainty in reservoir performance. Nearly every aspect of the reservoir characterization contains some degree of uncertainty, so predictions necessarily have a statistical character. The uncertainty in reservoir performance can be gauged by simulating a number of different geological realizations or scenarios. Thousands of such runs may be required to cover the range of parameter variation: therefore, it is not computationally feasible (or desirable) to perform these simulations on the fine grid model. Reliable upscaled models are required if a full assessment of project risk and uncertainty is to be accomplished. In addition, if thousands of coarse models need to be simulated, the upscaling must be highly automated. Different upscaling procedures are appropriate in different situations. The ideal procedure to use on a particular problem depends on the simulation purpose, the production mechanism, and the level of detail that can be accommodated in the coarse model. For a problem involving primary production with only oil being produced, the coarse model should correctly capture the effects of nearwell heterogeneity as well as the general large scale flow response of the reservoir. For scenarios involving displacement of oil by water or gas, it may be important to accurately capture the effects of key flow paths between injection and production wells. This may require the use of specialized gridding procedures.

2.1 Upscaling Techniques Overview

There are several different ways in which upscaling techniques can be classified. Accordingly to [16], we will classify the various methods in terms of the different kinds of parameters that are upscaled and the way in which these parameters are computed.

For single-phase flow involving a single component, the only parameters to be upscaled are porosity and the absolute permeability. In the more general case of two-phase flow, the absolute permeability and porosity as well as the relative permeability can be upscaled. However, in many cases it is possible to develop reasonably accurate coarse scale models for two-phase flow with only the upscaled absolute permeability and porosity, particularly when accurate upscaling is used in conjunction with flow-based grid generation. In models of this type, the geocellular scale relative permeabilities are used directly on the coarse scale. Thus, even for two phase flow systems, we can still generate coarse scale models with only absolute permeability and porosity upscaled in some cases: some theoretically justification for neglecting the upscaling of relative permeability is presented in [15]. We refer to this type of approach as *single-phase parameter* upscaling or *single-phase* upscaling with the understanding that the result can be used for both single and two-phase (or multiphase) flow problems.

The second type of classification is related to the way in which upscaled parameters are computed. In all cases the intent of the upscaling procedure is to replace the fine model with a coarse model. In a purely local procedure, coarse scale parameters are computed by considering only the fine scale region corresponding to the target coarse block. No additional fine scale information is included in the upscaling calculation. In a global upscaling technique, the entire fine scale model is simulated for the calculation of the coarse scale parameters. The assumption here is that the coarse scale parameters will be applicable to other flow scenarios.

There are several important variants of the purely local and global approaches. Very interesting ones are the extended local upscaling techniques. In extended local procedures, coarse scale parameters are computed by considering the region corresponding to the target coarse block plus a border region or ring around this region. Coarse scale quantities are then generally computed by averaging the fine scale solution only over the region corresponding to the target coarse block.

2.1.1 Multiscale Methods

Another important upscaling technique is the Multiscale Finite Element Method (MsFEM). These methods have the benefit that the fine scale permeability information is included into the global solution in a systematic way. So far, these procedures have mainly been applied to the pressure equation while transport calculations have generally been performed by reconstructing the fine scale velocity field. This reconstruction entails the use of the fine grid for the solution of the saturation equation. Some example of multiscale methods applied to reservoir simulation context can be found in [24, 2].

Dual grid approaches are also related to multiscale methods. In these procedures, different grids are defined and reconstruction techniques are applied to determine fine scale variables from the global coarse scale solution. These methods can provide accurate results in many cases, though, like multiscale procedures, they require the fine scale permeability information to be stored and used during the global solution. For more information we refer to [29].

2.2 Geological Quantities Upscaling

2.2.1 Analitical Upscaling of the Porosity

Porosity on the coarse scale ϕ^* is computed ensuring that the pore volume is exactly conserved between the fine and coarse scales. Since this quantity does not depend on the direction, an average over the bulk volume is enough. Specifically, ϕ^* is computed as:

$$\phi^* = \frac{1}{V_b} \int_{V_b} \phi(\mathbf{y}) \, \mathrm{d}V, \tag{2.1}$$

where V_b is the bulk volume which is usually a cell belonging to the coarse grid.

2.2.2 Analytical Upscaling of the Permeability

For now we consider steady, single-phase incompressible flow with no source terms. We introduce a conceptual two-scale model for permeability, i.e. the permeability tensor varies on two distinct scales referred to as x and y. The x scale is a slow scale, meaning that variations in x are relatively gradual. The y scale, by contrast, is a fast scale, and captures the fine scale variation of permeability. That being said, we can rewrite the pressure equation (1.7) as:

$$\nabla \cdot \mathbf{K}(x, y)\nabla p = 0. \tag{2.2}$$

Homogenization procedures allow the Equation (2.2) to be replaced with an analogous equation in which variations need to be resolved only on the x (slow) scale. The pressure equation in this case can be written as:

$$\nabla \cdot \mathbf{K}^*(x)\nabla p^c = 0. \tag{2.3}$$

where $\mathbf{K}^*(x)$ is referred to as the effective permeability tensor and p^c is the coarse scale pressure. Note that $\mathbf{K}^*$ is defined on the scale of x. Therefore y-scale variations have been homogenized or averaged. This means that, in solving problem (2.3), we do not need to resolve effects on the scale of y allowing a coarser discretization, which leads to significant computational savings.

The simplest procedures for computing grid block permeabilities are power averaging procedures introduced in [13]. These approaches do not require any numerical solutions so they are computationally very efficient. The basic approach entails computing upscaled permeability components, here denotes as k_i^*, as:

$$k_i^* = \left(\frac{1}{V_b}\int_{V_b}[k_i(\mathbf{y})]^{\omega_i}\,\mathrm{d}V\right)^{1/\omega_i}, \tag{2.4}$$

where V_b is the coarse block bulk volume and both the fine scale permeability $\mathbf{K}(\mathbf{y})$ and the upscaled permeability $\mathbf{K}^*$ are considered to be diagonal tensors, with index i indicating a diagonal component. The power averaging exponent ω_i can vary with the direction i. This type of averaging procedure can be readily applied to coarse grid cells of any shape, so it is suitable for use with irregular grids.

The power averaging exponent ω_i is constrained to lie between -1 and 1. The extremes values correspond to the suitable choices for layered systems: for flow parallel to the layers $\omega = 1$ performs the arithmetic average while for flow orthogonal to the layers $\omega = -1$ gives the harmonic average. The geometric mean corresponds to $\omega \rightarrow 0$ (in this case k_i^ω is replaced by $ln(k_i)$). Power averaging can also be applied using a combination of two different values of ω. For example, for a structured, approximately layered system in the $x - z$ coordinate system, k_x^* might be computed by harmonically

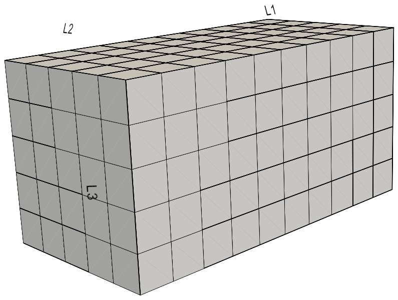


Figure 2.1: Coarse cell example.

averaging along each layer in x and then arithmetically averaging these layer averaged values. The use of this type of procedure generalize power averaging giving a higher degree of applicability. Power averaging exponents may be determined in practice by tuning against numerical upscaling results. The assumption then is that the same ω_i can be used for models with similar permeability distributions.

The fact that ω_i can vary with the direction leads to the general observation that the upscaled permeability can be anisotropic even when the underlying permeability field $\mathbf{K}$ is everywhere isotropic. For the power averaging approach described here, the upscaled permeability is still a diagonal tensor. However, for the more general numerical methods described below, the upscaled permeability is typically a full tensor.

Other analytical procedures for permeability upscaling include the renormalization approach presented in [26] and the full tensor averaging technique in [25]. Both of these approaches are very efficient and have been shown to perform well for some classes of problems. Like power averaging techniques, however, they lack the generality of the more complex procedures.

2.2.3 Flow Based Upscaling of the Permeability

The more robust and accurate procedures for computing $\mathbf{K}^*$ require the solution of the fine scale pressure equation over the target coarse block. As we will discuss in the sequel, in some cases it is beneficial to use an extended local approach in order to include the effects of neighbouring regions in the calculation of $\mathbf{K}^*$. We now consider a variety of approaches, beginning with the simplest purely local techniques.

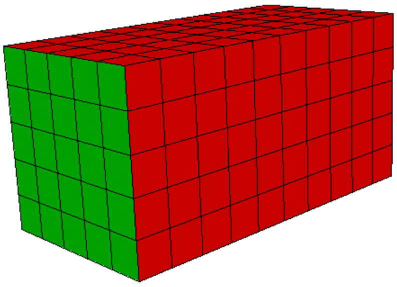


Figure 2.2: Close boundary conditions example for y_1 problem: red color represents no flux condition while green color pressure condition.

In computing equivalent grid block permeabilities, we solve the problem (2.2) over the fine scale region corresponding to the target coarse block. A significant issue in any local or extended local upscaling technique is the choice of boundary conditions to be imposed. Because the actual conditions imposed on the region during the global flow simulation are not known a priori and will in general vary, there is always some ambiguity in specifying the boundary conditions in the upscaling procedure. There is, additionally, some freedom in how the upscaled $\mathbf{K}^*$ is computed from the local fine grid solution.

We now consider the cartesian coarse cell shown in Figure 2.1. The simplest, and in many ways the most intuitive, boundary conditions for this problem consist in imposing a constant pressure on two opposite faces and no flow condition on lateral faces: we will refer to this kind of conditions as closed boundary conditions. These boundary conditions require us to solve (2.2) three times, once for each space direction. In the first solution we set

$$\begin{aligned} p(0, y_2, y_3) &= 1, \\ p(L_1, y_2, y_3) &= 0, \\ \mathbf{u}(y_1, 0, y_3) \cdot \mathbf{n} = \mathbf{u}(y_1, L_2, y_3) \cdot \mathbf{n} &= 0, \\ \mathbf{u}(y_1, y_2, 0) \cdot \mathbf{n} = \mathbf{u}(y_1, y_2, L_3) \cdot \mathbf{n} &= 0, \end{aligned} \tag{2.5}$$

where L_1, L_2, L_3 are the coarse cell sizes and $\mathbf{n}$ is the outward unit normal. The other solutions are computed respectively along y_2 and y_3 directions. From these three solutions we can compute total flow rates through the faces of the region, i.e. from the first solution, since the pressure gradient is along the y_1 axis, we obtain an average flux in the y_1 direction as

$$q_1 = \int_0^{L_2} \int_0^{L_3} \mathbf{u}(L_1, y_2, y_3) \cdot \mathbf{n} \, \mathrm{d}y_2 \, \mathrm{d}y_3. \tag{2.6}$$

Similarly, q_2 and q_3 can be computed from the second and third solutions. Then, using an averaged form of Darcy's law, we obtain the coarse permeability

$$k_1^* = \frac{q_1 L_1}{L_2 L_3 \Delta p}, \tag{2.7}$$

where $\Delta p = 1$ from boundary conditions (2.6). The quantities k_2^* and k_3^* are computed in a similar manner.

In many cases, the coarse permeability computed in this way provide reasonably accurate coarse block permeabilities. However, the boundary conditions and method for computing $\mathbf{K}^*$ from the local fine grid solution in this case preclude the calculation of the cross terms of $\mathbf{K}^*$. These terms can be significant in cases where the grid is not locally K-orthogonal (by K-orthogonality we mean that the grid geometry and permeability can be represented in terms of two-point fluxes in a finite volume discretization). Therefore, procedures for computing a full tensor $\mathbf{K}^*$ are required for more general cases.

One possible approach for generating full tensor coarse block permeabilities is to compute volume averaged velocities and pressure gradients over the entire flow domain:

$$\langle \mathbf{u} \rangle^j = \frac{1}{V} \int_V \mathbf{u}^j \, \mathrm{d}V, \tag{2.8}$$

$$\langle \nabla p \rangle^j = \frac{1}{V} \int_V (\nabla p)^j \, \mathrm{d}V, \tag{2.9}$$

where $j = I, II, III$ indicate the flow solution with the pressure difference along y_1, y_2 and y_3 respectively. Since $\langle \mathbf{u} \rangle$ and $\langle \nabla p \rangle$ have three components and we solve three flow problems, nine components of $\mathbf{K}^*$ can be calculated from these two flow solutions. Then we can write

$$\langle \mathbf{u} \rangle_i^j = -\sum_{l=1}^{3} k_{il}^* \, \langle \nabla p \rangle_l^j \,, \tag{2.10}$$

where $j = I, II, III$ is the pressure gradient direction and $i = 1, 2, 3$ is the vector component. Since the unknown quantities are the permeability tensor components, we can rearrange equations (2.10) and define the following matrix:

$$\bar{\bar{A}} = \begin{bmatrix} \langle\nabla p\rangle_1^I & \langle\nabla p\rangle_2^I & \langle\nabla p\rangle_3^I & & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & & \langle\nabla p\rangle_1^I & \langle\nabla p\rangle_2^I & \langle\nabla p\rangle_3^I \\ \langle\nabla p\rangle_1^{II} & \langle\nabla p\rangle_2^{II} & \langle\nabla p\rangle_3^{II} & & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & & \langle\nabla p\rangle_1^{II} & \langle\nabla p\rangle_2^{II} & \langle\nabla p\rangle_3^{II} \\ \langle\nabla p\rangle_1^{III} & \langle\nabla p\rangle_2^{III} & \langle\nabla p\rangle_3^{III} & & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & & \langle\nabla p\rangle_1^{III} & \langle\nabla p\rangle_2^{III} & \langle\nabla p\rangle_3^{III} \end{bmatrix} \tag{2.11}$$

and the linear system:

$$\bar{\bar{A}} \times \begin{bmatrix} k_{11}^* \\ k_{12}^* \\ k_{13}^* \\ k_{21}^* \\ k_{22}^* \\ k_{23}^* \\ k_{31}^* \\ k_{32}^* \\ k_{33}^* \end{bmatrix} = - \begin{bmatrix} \langle\mathbf{u}\rangle_1^I \\ \langle\mathbf{u}\rangle_2^I \\ \langle\mathbf{u}\rangle_3^I \\ \langle\mathbf{u}\rangle_1^{II} \\ \langle\mathbf{u}\rangle_2^{II} \\ \langle\mathbf{u}\rangle_3^{II} \\ \langle\mathbf{u}\rangle_1^{III} \\ \langle\mathbf{u}\rangle_2^{III} \\ \langle\mathbf{u}\rangle_3^{III} \end{bmatrix}, \tag{2.12}$$

which can now be solved to determine the components of $\mathbf{K}^*$.

With the closed boundary conditions, the K^* computed using (2.12) will in general not be symmetric. Various procedures can be applied to enforce symmetry; the simplest approach is to set each of the cross terms equal to $(k_{ij}^* + k_{ji}^*)/2$. A better approach is to solve a least square problem. In this case we enforce symmetry by adding three equations of the form $k_{ij}^* - k_{ji}^* = 0$

with $i \neq j$. The resulting matrix is now given by

$$\bar{\bar{A}}_{LS} = \begin{bmatrix} \langle\nabla p\rangle_1^I & \langle\nabla p\rangle_2^I & \langle\nabla p\rangle_3^I & & & & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdot & \cdot & \cdot & 0 & 0 & 0 \\ 0 & 0 & 0 & & & & \langle\nabla p\rangle_1^I & \langle\nabla p\rangle_2^I & \langle\nabla p\rangle_3^I \\ \langle\nabla p\rangle_1^{II} & \langle\nabla p\rangle_2^{II} & \langle\nabla p\rangle_3^{II} & & & & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdot & \cdot & \cdot & 0 & 0 & 0 \\ 0 & 0 & 0 & & & & \langle\nabla p\rangle_1^{II} & \langle\nabla p\rangle_2^{II} & \langle\nabla p\rangle_3^{II} \\ \langle\nabla p\rangle_1^{III} & \langle\nabla p\rangle_2^{III} & \langle\nabla p\rangle_3^{III} & & & & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdot & \cdot & \cdot & 0 & 0 & 0 \\ 0 & 0 & 0 & & & & \langle\nabla p\rangle_1^{III} & \langle\nabla p\rangle_2^{III} & \langle\nabla p\rangle_3^{III} \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}, \tag{2.13}$$

which now has 12 rows and 9 columns for 9 unknown variables. Note that last three equations should be rescaled in order to avoid numerical errors.

In addition to symmetry, we also require that $\mathbf{K}^*$ be positive definite. This requirement is generally satisfied by the methods described here. In the few cases when this is not satisfied, $\mathbf{K}^*$ can be recomputed using outlet averaging or using periodic boundary conditions (described below).

The closed boundary conditions just discussed are not the most general boundary conditions that can be used. Another alternative is to use boundary conditions that specify a linear pressure variation along the sides parallel to the direction of the pressure gradient. Since fluid is able to flow through lateral faces, this kind of conditions are named open boundary conditions and for the first flow problem along y_1 direction they can be written as:

$$\begin{aligned} p(0, y_2, y_3) &= 1, \\ p(L_1, y_2, y_3) &= 0, \\ p(y_1, 0, y_3) = p(y_1, L_2, y_3) &= 1 - y_1/L_1, \\ p(y_1, y_2, 0) = p(y_1, y_2, L_3) &= 1 - y_1/L_1. \end{aligned} \tag{2.14}$$

The boundary conditions for the other two problems are built changing the main direction with y_2 and y_3. In order to find a solution we can proceed as for the closed boundary conditions. The $\mathbf{K}^*$ computed with this strategy will in general differ from the $\mathbf{K}^*$ computed using closed boundary conditions. It is not clear from the literature which of these approaches is the more accurate, it is likely that the best choice will be case dependent. However, solutions using open boundary conditions will have, in general, a higher connectivity since more fluid can flow through the lateral faces of the coarse

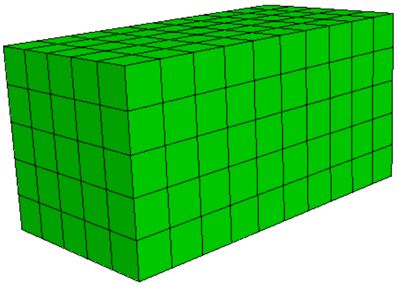


Figure 2.3: Open boundary conditions example for y_1 problem: in this case flow through lateral faces is possible.

cell increasing the average flux while closed boundary conditions define no flux condition on lateral faces.

We now describe the use of periodic boundary conditions for the calculation of $\mathbf{K}^*$. The specific form of the boundary conditions derives from the assumption that the system is replicated periodically in space. This choice eliminates some of the ambiguity of the other methods since it provides the same result for either method of post-processing of the fine grid solution. These boundary conditions again require that three local fine scale problems be solved. Periodic boundary conditions can be specified for the first problem as:

$$\begin{aligned} p(0, y_2, y_3) &= p(L_1, y_2, y_3) - L_1, \\ p(y_1, 0, y_3) &= p(y_1, L_2, y_3), \\ p(y_1, y_2, 0) &= p(y_1, y_2, L_3), \\ \mathbf{u}(0, y_2, y_3) \cdot \mathbf{n} &= -\mathbf{u}(L_1, y_2, y_3) \cdot \mathbf{n}, \\ \mathbf{u}(y_1, 0, y_3) \cdot \mathbf{n} &= -\mathbf{u}(y_1, L_2, y_3) \cdot \mathbf{n}, \\ \mathbf{u}(y_1, y_2, 0) \cdot \mathbf{n} &= -\mathbf{u}(y_1, y_2, L_3) \cdot \mathbf{n}, \end{aligned} \tag{2.15}$$

By solving this problem and the resulting problems along y_2 and y_3 directions, we can compute $\mathbf{K}^*$ using (2.12).

Periodic boundary conditions have several useful features. They guarantee that the resulting $\mathbf{K}^*$ will be symmetric and positive-definite. Thus, no post-processing of the result is necessary to ensure that these two criteria are met. However it can be proved that the differences between the various methods are slight and strongly dependent on the fine permeability structure (see [27]).

This approach could also be used in the case of the coarse cell is not a perfect parallelepiped or the faces are not aligned with the fine grid. Thus, the simulation domain is formed by a set of fine grid cells that cover the coarse cell while it is possible to choose one of the boundary condition presented above. The main difference is that all the average values are computed over the coarse cell domain. Some example can be found in [17] on in [18] where the averages are computed integrating only over the cells entirely contained in the coarse cell. A better approach for permeability upscaling over non matching grids can be found in [9].

2.2.4 Transmissibility Upscaling

The approach introduced in the previous section can also be applied to the direct calculation of upscaled transmissibility. The transmissibility is a property of the discrete equation that link the pressure with the flow rate through cells interface. Given a three dimensional cartesian grid, the flow rate $q^x_{i,i+1}$ from grid block (i,j,k) to grid block $(i+1,j,k)$ is given by:

$$q^x_{i+1/2,j,k} = (T_x)_{i+1/2,j,k}(p_{i,j,k} - p_{i+1,j,k}), \tag{2.16}$$

where the transmissibility $(T_x)_{i,i+1}$ is defined as

$$(T_x)_{i+1/2,j,k} = \frac{2(k_x)_{i+1/2,j,k}\Delta y \Delta z}{\Delta x_{i+1,j,k} - \Delta x_{i,j,k}}, \tag{2.17}$$

where $\Delta x_{i,j,k}$ denotes the size of grid block (i,j,k) and the interface permeability $(k_x)_{i+1/2,j,k}$ is given by the weighted harmonic average of $(k_x)_{i,j,k}$ and $(k_x)_{i+1,j,k}$:

$$(k_x)_{i+1/2,j,k} = \frac{(\Delta x_{i,j,k} + \Delta x_{i+1,j,k})(k_x)_{i,j,k}(k_x)_{i+1,j,k}}{\Delta x_{i,j,k}(k_x)_{i+1,j,k} + \Delta x_{i+1,j,k}(k_x)_{i,j,k}}. \tag{2.18}$$

The transmissibility along other principal directions can be computed in a similar manner. In case of full tensor permeabilities, the formation of the discretized equations is more complex (see [1]).

As we have said, coarse scale transmissibilities can also be computed from the local problems. To do this, we focus on a cell edge. In order to compute the coarse transmissibility, the local problem must contain the fine scale region corresponding to coarse blocks (i,j,k) and $(i+1,j,k)$ which are on either side of the target edge $(i+1/2,j,k)$. We solve for flow over this region using any of the boundary conditions described above and then compute the upscaled transmissibility via:

$$(T^*_x)_{i+1/2,j,k} = \frac{q^c_{i+1/2,j,k}}{\langle p \rangle_{i,j,k} - \langle p \rangle_{i+1,j,k}}, \tag{2.19}$$

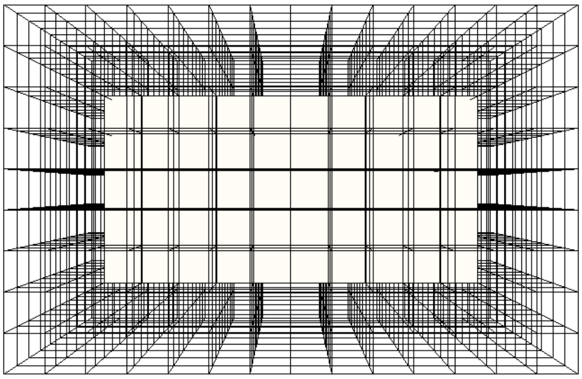


Figure 2.4: Coarse cell surrounded by a ring with $\rho = 2$.

where T_x^* designates the upscaled transmissibility in the x direction, $q_{i+1/2,j,k}^c$ is the total flow across the interface between coarse blocks (i, j, k) and $(i + 1, j, k)$ and $\langle p \rangle$ is the volume average of the fine scale pressure over a coarse block region. Similar expressions proved coarse transmissibility over y and z directions.

The transmissibility upscaling over not matching grids is also possible. Proper averages over the coarse cell domain have to be computed. For a 2D example we refer to [32] while for the 3D case we present proper integration methods in Chapter 4.

2.3 Extended Domain

It has been observed that improved accuracy in $\mathbf{K}^*$ and T^* can be achieved by extending the domain of the local problem. By including neighbouring regions in the calculation of $\mathbf{K}^*$ for a particular coarse block, the effects of large scale permeability connectivity (or lack of connectivity) can be better captured, particularly when the permeability field contains features that are not oriented with the grid. A *ring* (indicated by the letter ρ) is defined as a set of cells surrounding the local problem domain. If $\rho = 1$ then only a 1 fine cells ring over each directions is added to the fine resolution problem. Any of the boundary conditions discussed in the previous section can now be applied on the expanded domain.

For example, in the case of permeability upscaling, since we wish to compute $\mathbf{K}^*$ for only a portion of the fine scale domain, we apply the volume averaging procedure using Equations (2.8) and (2.9) computing $\langle \mathbf{u} \rangle$ and $\langle \nabla p \rangle$

only over the coarse cell domain and then solve the linear system (2.12) to compute $\mathbf{K}^*$. We can note that, even if periodic boundary conditions are applied, the $\mathbf{K}^*$ computed will not in general be symmetric. This is because the symmetry provided by periodic boundary conditions for purely local upscaling is lost when border regions are applied. However, symmetry can be approximately recovered through use of the least square technique as we have shown previously.

2.4 Global and Local-Global upscaling

All the methods described in the previous sections require the specification of boundary conditions for the local problem, which could be closed, open or periodic conditions. In global upscaling methods the intent is to solve a global flow problem and to use this solution to extract coarse scale quantities. Most of these methods apply transmissibility upscaling for the calculation of coarse grid quantities. From the fine grid solution, transmissibilities are computed by averaging over coarse block regions and then applying (2.19).

Global upscaling methods can provide very accurate results for a particular set of wells and boundary conditions. In many cases the model developed in this way can be used for other flow scenarios. However, it is also possible that the model may lack robustness with respect to other boundary conditions or well arrangements, e.g. see [33].

Another class of upscaling technique is the local-global upscaling that is presented in [10]. The main difference from global upscaling is that the global flow is computed using the coarse grid. The idea is to use such solution to estimate the boundary conditions to be used in the local calculation of the transmissibility. Then the procedure is iterated until the upscaled quantity is consistent with the global flow, i.e. the difference of the solution with respect to the previous iteration is small.

In both global and local-global upscaling, a negative transmissibility may be generated in some cell. In these cases, we should compute again the transmissibility using a local upscaling with periodic boundary condition to ensure the positivity of the upscaled quantity.

2.5 Near Well Upscaling

In local and extended local techniques, we have assumed that the flow can be locally described as approximately linear so that the large scale pressure gradient is ∇p is approximately constant. This is not assumed in the global and local-global approaches which determine local velocity and pressure from a global model solution. The assumption of a locally linear pressure field is also not applicable in the near-well region, as the steady state pressure in the well proximity varies as $\log r$, where r is the radial distance. Therefore, special techniques are required for.

The well behaviour is specified by the so called Well Index (WI) which relates the wellbore pressure to the grid cell pressure as

$$q^w_{i,j,k} = WI_{i,j,k}(p_{i,j,k} - p^w_{i,j,k}). \tag{2.20}$$

Note that it is very similar to the transmissibility definition. The WI depends of many parameters such as wellbore inclination and pipe radius.

In general, a local extended domain surrounding the well is defined and, using proper boundary conditions, a steady-state problem is solved. We then select a coarse cell that includes the well. We compute average flux on coarse grid interfaces and average pressure on coarse cells. Moreover we can compute the wellbore pressure p_w and well flow rate q_w of the steady-state solution. Finally, equivalent transmissibility and numerical WI^* are determined by

$$T^*_n = -\frac{q^c_n}{\langle p\rangle_n - \langle p\rangle_{i,j,k}} \tag{2.21}$$

$$WI^*_{i,j,k} = \frac{q_w}{\langle p\rangle_{i,j,k} - p_w} \tag{2.22}$$

where n represents a neighbour cell of the selected coarse cell (i,j,k). A in-depth analysis can be found in [14].

Chapter 3

Local Problem Discretization

In this section we discuss the discretization techniques used to solve the local problems during the Upscaling procedure. In particular, we will use the Mixed Finite Element Method (MFEM) to find the velocity and pressure fields. The computational domain of the local problem is usually very small, on the order of the coarse grid cells size. A numerical method ensuring the local mass conservation of the solution is therefore mandatory in order to limit errors due to the approximation of in the mass balance. Another reason that leads us to consider a Mixed Finite Element Method, is the possibility to reconstruct accurately the velocity field over a given set of points: this is crucial for non-matching upscaling technique, since we have to compute quantities from the final velocity field over non-matching subsets, e.g. velocity flux over the interface between a coarse cells pair. The Mixed Finite Element Method is widely used for fluid flow equations such as the Stokes problem or fluid flow through porous media.

3.1 Local Mass Conservation

Let us recall the pressure equation for a single-phase flow in case of constant density and with the absence of gravity (1.8):

$$\begin{cases} -\nabla \cdot \mathbf{K}\nabla p = f & \text{in } \Omega \subset \mathbb{R}^2 \\ p = p_0 & \text{on } \Gamma_D, \\ \dfrac{\partial p}{\partial \mathbf{n}} = p_n & \text{on } \Gamma_N. \end{cases} \tag{3.1}$$

where $\Omega \subset \mathbb{R}^3$ is a polygonal domain with a Lipshitz continuous boundary $\partial\Omega = \bar{\Gamma}_D \cup \bar{\Gamma}_N$, $\Gamma_D \cap \Gamma_N = \emptyset$, $|\Gamma_D| > 0$, $\mathbf{n}$ is the outward unit normal on $\partial\Omega$, and $\mathbf{K}$ is a symmetric, uniformly, positive definite tensor, that means there are $k_0, k_1 \in \mathbb{R}$, $0 < k_0 \leq k_1 < \infty$ such that

$$k_0 \xi^T \xi \leq \xi^T \mathbf{K}(\mathbf{x}) \xi \leq k_1 \xi^T \xi \tag{3.2}$$

for every $\mathbf{x} \in \Omega$ and $\xi \in \mathbb{R}^3$. As we have seen in Chapter 1, $\mathbf{u}$ is the Darcy velocity and $\mathbf{K}$ represents the permeability field divided by the viscosity. Since we set the fluid viscosity equal to one during the local problem resolution in the upscaling method, $\mathbf{K}$ will be the real permeability tensor. Moreover we assume that $p_0 \in H^{1/2}(\Gamma_D)$ and $p_n \in H^{1/2}(\Gamma_N)$.

In the first instance, we consider a full homogeneous Dirichlet problem, e.g. $\Gamma_D \equiv \partial\Omega$ and $p_0 = 0$. Multiplying for a test function and properly integrating by parts gives us the following weak formulation associated with (3.1):
Find $u \in V$ such that

$$a(p, \phi) = \mathcal{F}(\phi) \quad \forall \phi \in V, \tag{3.3}$$

where

$$\begin{aligned} V &= H_0^1(\Omega), \\ a(p, \phi) &= \int_\Omega \mathbf{K} \nabla p \cdot \nabla \phi \, \mathrm{d}\Omega : V \times V \to \mathbb{R}, \\ \mathcal{F}(v) &= \int_\Omega f \phi \, \mathrm{d}\Omega : V \to \mathbb{R}. \end{aligned}$$

Since $a(\cdot, \cdot)$ is a bilinear, continuous and coercive form and $F(\cdot)$ is linear and continuous, the Lax-Milgram Lemma ensures that problem (3.3) has a solution that is unique.

We can discretize the problem using a *Galerkin Finite Element Method* (GFEM). Given a geometrical regular partition $\mathcal{T}_h$ of the domain Ω into non overlapping simplexes K, let $x = (x_1, \cdots, x_d)$ and let $\mathbb{P}_r$ be the space of polynomials in the variables $x_1, \cdots, x_d$ with real coefficients and of global degree at most r over the simplex K:

$$\mathbb{P}_r(K) = \left\{ p(x) = \sum_{\substack{0 \leq i_1, \cdots, i_d \leq r \\ i_1 + \cdots + i_d \leq r}} \alpha_{i_1 \cdots i_d} x_1^{x_1} x_d^{i_d};\ \alpha_{i_1 \cdots i_d} \in \mathbb{R} \right\}.$$

We can readily verifies that $\mathbb{P}_r(K)$ is a vector space of dimension

$$\dim \mathbb{P}_r(K) = \left(\begin{array}{c} d + r \\ r \end{array} \right).$$

Then, we introduce the finite dimensional space of piecewise continuous polynomials over $\mathcal{T}_h$:

$$X_h^r(\Omega) = \{ v \in C^0(\bar{\Omega}) : v|_k \in \mathbb{P}_r(K) \forall K \in \mathcal{T}_h v|_\Gamma = 0 \}. \tag{3.4}$$

The GFEM approximation of (3.1) is:
Find $p_h \in V_h \equiv X_h^r(\Omega)$ such that

$$a(p_h, \phi_h) = \mathcal{F}(\phi_h) \quad \forall \phi_h \in V_h. \tag{3.5}$$

Again, application of the Lax-Milgram Lemma allows to conclude that (3.5) has a unique solution.

Now we consider the space of piecewise linear continuous polynomials $X_h^1(\Omega)$ for the problem (3.5). Giving a closer look at ∇p_h, it is clear that

$$\nabla p_h|_K \in (\mathbb{P}_0(K))^3 \quad \forall K \in \mathcal{T}_h$$

so the approximate vector filed is a discontinuous function over $\mathcal{T}_k$. In particular, we notice that:

1. $\forall K \in \mathcal{T}_h$, we have
$$-\text{div}\mathbf{K}\nabla p_h = 0. \tag{3.6}$$

2. $\forall e = \partial K^+ \cap \partial K^-$, K^+, K^- being two mesh elements, we have
$$\nabla p_h^+ \cdot \mathbf{n}_e^+ + \nabla p_h^- \cdot \mathbf{n}_e^- \neq 0. \tag{3.7}$$

The relation (3.6) reveals that the approximate vector field ∇p_h is not locally self equilibrated, so it does not reproduce the starting equilibrium equation. The relation (3.7) says that GFEM does not satisfy the local mass conservation properties: in other words, fictitious mass could be generated along cells interfaces without any physical meaning.

We conclude that the GFEM suffers from loss of accuracy of some physical requirements for ∇p. Moreover, this is not related to the degree r used for the finite dimensional space V_h.

3.2 Dual Mixed Formulation

In this section, we will give an alternative formulation of the pressure problem (3.1). Since we are interested in having a good approximation of the velocity field, we use the Darcy's law to explicitly define $\mathbf{u}$. Therefore we obtain the formulation named *dual mixed formulation*:
Find $(\mathbf{u}, p)$ such that

$$\begin{cases} \mathbf{u} = -\mathbf{K}\nabla p & \text{in } \Omega, \\ \nabla \cdot \mathbf{u} = f & \text{in } \Omega, \\ p = p_0 & \text{on } \Gamma_N, \\ \mathbf{u} \cdot \mathbf{n} = u_0 & \text{on } \Gamma_D, \end{cases} \tag{3.8}$$

where $\Omega \subset \mathbb{R}^3$ is a polygonal domain with a Lipshitz continuous boundary $\partial\Omega = \bar{\Gamma}_D \cup \bar{\Gamma}_N$, $\Gamma_D \cap \Gamma_N = \emptyset$, $|\Gamma_D| > 0$ and $\mathbf{n}$ is the outward unit normal on $\partial\Omega$.

We assume the source term $f \in L^2(\Omega)$ and the boundary data are $p_0 \in H^{1/2}(\Gamma_N)$ and $u_0 \in H^{1/2}(\Gamma_D)$. We can identify the first equation as the constitutive law (that is the Darcy's law), the second equation as the indefinite equilibrium condition and the remaining equations as constraint boundary conditions.

Since $\mathbf{K}$ is invertible, let us introduce a new matrix

$$\boldsymbol{\eta} = \mathbf{K}^{-1}, \tag{3.9}$$

then we can rewrite the equilibrium condition as

$$\boldsymbol{\eta}\mathbf{u} = -\nabla p.$$

We start by multiplying by a proper test functions and integrating over Ω:

$$\begin{cases} \displaystyle\int_\Omega \boldsymbol{\eta}\mathbf{u} \cdot \mathbf{v} \, \mathrm{d}\Omega + \int_\Omega \nabla p \cdot \mathbf{v} \, \mathrm{d}\Omega = 0 & \forall \mathbf{v}, \\ \displaystyle\int_\Omega (\nabla \cdot \mathbf{u} + f) q \, \mathrm{d}\Omega = 0 & \forall q. \end{cases} \tag{3.10}$$

By application of the Green's formula to the first equation we obtain

$$\int_\Omega \boldsymbol{\eta}\mathbf{u} \cdot \mathbf{v} \, \mathrm{d}\Omega - \int_\Omega p \nabla \cdot \mathbf{v} \, \mathrm{d}\Omega + \int_{\partial\Omega} p \mathbf{v} \cdot \mathbf{n} \, \mathrm{d}\sigma = 0. \tag{3.11}$$

Note that in the mixed formulation the Neumann boundary condition (open condition) is a condition on pressure. So we can substitute the pressure with p_0 in the last integral over Γ_N. We can now define the Sobolev spaces in which the solution is defined. The function space for the pressure is

$$Q = L^2(\Omega) = \left\{ q : \Omega \to \mathbb{R} : \int_\Omega q^2 \, \mathrm{d}\Omega < \infty \right\} \tag{3.12}$$

with the norm defined as

$$\|q\|_Q := \|q\|_{L^2(\Omega)} = \left(\int_\Omega q^2 \, \mathrm{d}\Omega \right)^{1/2}. \tag{3.13}$$

The natural choice for the velocity is

$$H(\mathrm{div}; \Omega) = \{ \mathbf{v} \in (L^2(\Omega))^2 : \nabla \cdot \mathbf{v} \in L^2(\Omega) \}. \tag{3.14}$$

with the norm defined as

$$\|\mathbf{v}\|_V = \|\mathbf{v}\|_{H(\mathrm{div};\Omega)} = \left(\int_\Omega \mathbf{v} \cdot \mathbf{v} \, \mathrm{d}\Omega + \int_\Omega (\nabla \cdot \mathbf{v})^2 \, \mathrm{d}\Omega \right)^{1/2}. \tag{3.15}$$

For our purposes, we set $u_0 = 0$ and $f = 0$. Since the Dirichlet boundary condition (closed condition) are enforced into the Sobolev space V in a strong way, we define V_0 as

$$V_0 = \{\mathbf{v} \in V : \mathbf{v} \cdot \mathbf{n} = 0 \text{ on } \Gamma_D\}, \tag{3.16}$$

equipped with the norm defined in (3.15). Finally, substituting the boundary data and applying all the considerations given above, we obtain the mixed variational form of the problem (3.8):
Find $(\mathbf{u}, p) \in V_0 \times Q$ such that

$$\begin{cases} a(\mathbf{u}, \mathbf{v}) + b(\mathbf{v}, p) = \mathcal{F}(\mathbf{v}) & \forall \mathbf{v} \in V_0, \\ b(\mathbf{u}, q) = 0 & \forall q \in Q, \end{cases} \tag{3.17}$$

where

$$a(\mathbf{u}, \mathbf{v}) = \int_\Omega \boldsymbol{\eta} \mathbf{u} \cdot \mathbf{v} \, \mathrm{d}\Omega, \tag{3.18}$$

$$b(\mathbf{v}, p) = -\int_\Omega p \nabla \cdot \mathbf{v} \, \mathrm{d}\Omega, \tag{3.19}$$

$$\mathcal{F}(\mathbf{v}) = -\int_{\Gamma_D} p_0 \mathbf{v} \cdot \mathbf{n} \, \mathrm{d}\sigma. \tag{3.20}$$

It is interesting to observe that the following property holds (for a proof, see [5]):

Proposition 3.1. *Let* $\mathbf{v} : \Omega \to \mathbb{R}^3$ *be such that:*

a) $\mathbf{v}|_K \in (H^1(K))^3 \quad \forall K \in \mathcal{T}_h$;

b) $\mathbf{v}^+ \cdot \mathbf{n}_e^- + \mathbf{v}^+ \cdot \mathbf{n}_e^- = 0 \quad \forall e = \partial K^+ \cap \partial K^-$.

Then, $\mathbf{v} \in H(div; \Omega)$. *Conversely, if* $\mathbf{v} \in H(div; \Omega)$ *and **a)** holds, then **b)** also holds.*

From Proposition 3.1 we immediately conclude that functions in V automatically satisfy the local mass conservation principle at cells interfaces. Finally, the relation

$$\int_\Omega (\nabla \cdot \mathbf{u} - f) q \, \mathrm{d}x = 0 \quad \forall q \in Q$$

automatically implies, due to the completeness of $L^2(\Omega)$, that $\nabla \cdot \mathbf{u} = f$ almost everywhere in Ω, which is a very good instance of the equilibrium condition.

The formulation (3.17) is a particular saddle-point problem. In the following section we will report the main stability and convergence properties for this class of problems.

3.3 The saddle-point problem

Let V and Q be Hilbert spaces equipped with norms $\|\cdot\|_V$ and $\|\cdot\|_Q$; let V' and Q' be the dual spaces associated. We introduce two bilinear forms $a(\cdot,\cdot) : V \times V \to \mathbb{R}$ and $b(\cdot,\cdot) : V \times Q \to \mathbb{R}$ that are continue, meaning that there exist $\gamma, \delta > 0$ such that:

$$|a(w,v)| \leq \gamma \|w\|_V \|v\|_V, \quad |b(w,q)| \leq \delta \|w\|_V \|q\|_Q, \tag{3.21}$$

for each $w, v \in X$ and $q \in Q$. The generic saddle-point problem can be written as:
Find $(u,p) \in V \times Q$ such that

$$\begin{cases} a(u,v) + b(v,p) = \langle l, v \rangle & \forall v \in V, \\ b(u,q) = \langle \sigma, q \rangle & \forall q \in Q, \end{cases} \tag{3.22}$$

where $l \in X'$ and $\sigma \in M'$ are two functional and $\langle \cdot,\cdot \rangle$.

It is useful to rewrite the problem (3.22) using operators. That said, we define the operators $A \in \mathcal{L}(V, V')$ and $B \in \mathcal{L}(V, M')$ defined as:

$$\begin{aligned} \langle Aw, v \rangle &= a(w,v) \quad \forall w, v \in V, \\ \langle Bv, q \rangle &= b(v,q) \quad \forall v \in V, q \in Q, \end{aligned}$$

where we have used the standard notation $\mathcal{L}(X,Y)$ for the space of linear and continuous functionals from X to Y. Let $B^T \in \mathcal{L}(Q, V')$ be the adjoint operator of B defined as:

$$\left\langle B^T q, v \right\rangle = \langle Bv, q \rangle = b(v,q) \quad \forall v \in V, q \in Q.$$

We can finally rewrite problem (3.22) as:
Find $(u,p) \in V \times Q$ such that

$$\begin{cases} Au + B^T p = l & \text{in } V', \\ Bu = \sigma & \text{in } Q'. \end{cases} \tag{3.23}$$

Then, we define the *affine variety* of the Hilbert space V as

$$V^\sigma = \{ v \in V : b(v,q) = \langle \sigma, q \rangle \, \forall q \in Q \}. \tag{3.24}$$

We note that the V^0 space identify the *Kernel* of B that is a closed subspace of V.

The following theorem ensures the existence and uniqueness of the solution for the problem (3.22):

Theorem 3.1. *The problem* (3.22) *admits one and only one solution if the following conditions are satisfied:*

1. *The bilinear form* $a(\cdot,\cdot)$ *is continuous* (3.21) *and coercive on* V^0*, i.e.* $\exists\alpha > 0$ *such that:*

$$a(v,v) \geq \alpha \|v\|_V^2 \quad \forall v \in V^0. \tag{3.25}$$

2. *The bilinear form* $b(\cdot,\cdot)$ *is continuous* (3.21)*; moreover,* $\exists\beta^* > 0$ *such that:*

$$\forall q \in Q, \exists v \in V \text{ with } v \neq 0 : b(v,q) \geq \beta^* \|v\|_V \|q\|_Q \tag{3.26}$$

Furthermore, the map $(l,\sigma) \to (u,p)$ *is an isomorphism between* $V' \times Q'$ *and* $V \times Q$*, and the following a priori estimates exist:*

$$\|u\|_V \leq \frac{1}{\alpha}\left[\|l\|_{V'} + \frac{\alpha+\gamma}{\beta^*}\|\sigma\|_{Q'}\right], \tag{3.27}$$

$$\|p\|_Q \leq \frac{1}{\beta^*}\left[\left(1+\frac{\gamma}{\alpha}\right)\|l\|_{V'} + \frac{\gamma(\alpha+\gamma)}{\alpha\beta^*}\|\sigma\|_{Q'}\right]. \tag{3.28}$$

In (3.27) and (3.28) we have used the norms of the dual spaces, defined as:

$$\|l\|_{V'} = \sup_{v\in V}\frac{\langle l,v\rangle}{\|v\|_V}, \qquad \|\sigma\|_{Q'} = \sup_{q\in Q}\frac{\langle \sigma,q\rangle}{\|q\|_Q}.$$

In order to approximate the problem (3.22), we introduce proper finite dimensional subspaces V_h and Q_h associated with V and Q.

Thus, the discrete formulation of (3.22) is:
Find $(u_h, p_h) \in (V_h \times Q_h)$ such that

$$\begin{cases} a(u_h,v_h) + b(v_h,q_h) = \langle l, v_h\rangle & \forall v_h \in V_h, \\ b(u_h,q_h) = \langle \sigma, q_h\rangle & \forall q_h \in Q_h. \end{cases} \tag{3.29}$$

The discrete saddle point problem (3.29) can be rewritten in algebraic formulation as

$$\begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix}\begin{pmatrix} \mathbf{u} \\ \mathbf{p} \end{pmatrix} = \begin{pmatrix} \mathbf{l} \\ \boldsymbol{\sigma} \end{pmatrix}, \tag{3.30}$$

that is the counterpart of the problem (3.23). In (3.30), A is a square symmetric matrix, B is a rectangular matrix, $\mathbf{u}$ and $\mathbf{p}$ are the unknown vectors, while $\mathbf{l}$ and $\boldsymbol{\sigma}$ are the right hand side vectors. System (3.30) has dimension

$$N_h = \dim(V_h) + \dim(Q_h)$$

and its coefficient matrix is symmetric but in general indefinite.

Likewise (3.24), we define the space

$$V_h^\sigma = \{v_h \in V_h : b(v_h, q_h) = \langle \sigma, q_h \rangle \, \forall q_h \in Q_h\}. \tag{3.31}$$

Then, we can formulate a new theorem, similar to Theorem 3.1, that ensures the existence and the uniqueness of the solution for the problem 3.29:

Theorem 3.2. *The problem* (3.29) *admits one and only one solution if the following conditions are satisfied:*

1. *The bilinear form* $a(\cdot,\cdot)$ *is continuous* (3.21) *and coercive on* V_h^0*, i.e.* $\exists \alpha_h > 0$ *such that:*

$$a(v_h, v_h) \geq \alpha_h \|v_h\|_V^2 \quad \forall v_h \in V_h^0. \tag{3.32}$$

2. *The bilinear form* $b(\cdot,\cdot)$ *is continuous* (3.21)*; moreover,* $\exists \beta_h > 0$ *such that:*

$$\forall q_h \in Q_h, \exists v_h \in V_h \text{ with } v_h \neq 0 : b(v_h, q_h) \geq \beta_h \|v_h\|_V \|q_h\|_Q. \tag{3.33}$$

Furthermore, the map $(l, \sigma) \to (u_h, p_h)$ *is an isomorphism between* $V' \times Q'$ *and* $V_h \times Q_h$*, and the following a priori estimates exists:*

$$\|u_h\|_V \leq \frac{1}{\alpha_h}\left[\|l\|_{V'} + \frac{\alpha_h + \gamma}{\beta_h}\|\sigma\|_{Q'}\right], \tag{3.34}$$

$$\|p_h\|_Q \leq \frac{1}{\beta_h}\left[\left(1 + \frac{\gamma}{\alpha_h}\right)\|l\|_{V'} + \frac{\gamma(\alpha_h + \gamma)}{\alpha_h \beta_h}\|\sigma\|_{Q'}\right]. \tag{3.35}$$

The relation expressed in (3.33) is widely known as the discrete *inf-sup* condition or LBB condition and it can be rewritten as:

$$\inf_{q_h \in Q_h, q_h \neq 0} \sup_{v_h \in V_h, v_h \neq 0} \frac{b(v_h, q_h)}{\|v_h\|_V \|q_h\|_Q} \geq \beta_h. \tag{3.36}$$

This condition restricts the selection of the finite subspaces V_h and Q_h. If the condition (3.36) is not satisfied, it exists a function $p_h^* \in Q_h$ such that

$p_h^* \neq 0$ and $b(v_h, p_h^*) = 0$ for each $v_h \in V_h$. Therefore, if (u_h, p_h) is a solution of the problem (3.29), then also $(u_h, p_h + \tau p_h^*)$, for every $\tau \in \mathbb{R}$, is a solution of the same problem. The p_h^* is named *spurious mode.* Since the spurious modes are transparent to the discrete model, the solution will have numeric fluctuation.

In practical cases, a direct proof of the discrete problem *inf-sup* condition may be not trivial. A useful alternative approach is to check the fulfilment of the following two conditions (see [5, 21]):

1. inclusion of the kernels

$$\mathrm{Ker} B_h \subset \mathrm{Ker} B, \tag{3.37}$$

2. existence of an operator $\Pi_h : V \to V_h$ such that

$$\begin{cases} b(v - \Pi_h v, q_h) = 0 & \forall q_h \in Q_h \\ \|\Pi_h v\|_V \leq C\|v\|_V & \forall v \in V, \end{cases} \tag{3.38}$$

where C is a positive constant independent of h.

Finally the last theorem about convergence analysis:

Theorem 3.3. *Let the conditions of Theorems 3.1 and 3.2 be true, then the solutions* (u, p) *and* (u_h, p_h) *of problems* (3.22) *and* (3.29) *respectively, satisfy the following error estimations:*

$$\|u - u_h\|_V \leq \left(1 + \frac{\gamma}{\alpha_h}\right) \inf_{v_h^* \in V_h^\sigma} \|u - v_h^*\|_V + \frac{\delta}{\alpha_h} \inf_{q_h \in Q_h} \|p - q_h\|_Q, \tag{3.39}$$

$$\|p - p_h\|_Q \leq \frac{\gamma}{\beta_h}\left(1 + \frac{\gamma}{\alpha_h}\right) \inf_{v_h^* \in V_h^\sigma} \|u - v_h^*\|_V + \left(1 + \frac{\delta}{\beta_h} + \frac{\gamma\delta}{\alpha_h\beta_h}\right) \inf_{q_h \in Q_h} \|p - q_h\|_Q. \tag{3.40}$$

Furthermore, the following estimation applies:

$$\inf_{v_h^* \in V_h^\sigma} \|u - v_h^*\|_V \leq \left(1 + \frac{\delta}{\beta_h}\right) \inf_{v_h \in V_h} \|u - v_h\|_Q. \tag{3.41}$$

A detailed proof of Theorems 3.1, 3.2 and 3.3 can be found in [28].

3.4 The Raviart-Thomas finite element

We now introduce some finite dimensional spaces that can be used in problem (3.29). For more details on possible choices of finite element spaces and their properties see [19].

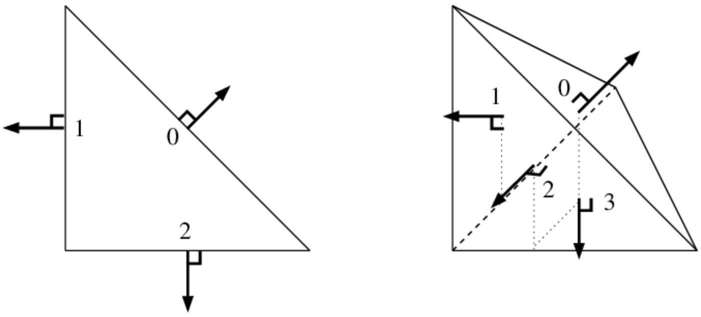


Figure 3.1: Degrees of freedom of $\mathbb{RT}_0$ *element in 2D and 3D.*

Let $\mathcal{T}_h$ be a triangulation of $\mathbb{R}^d$ and K be a simplex of $\mathcal{T}_h$. Consider the vector space of $\mathbb{R}^d$-valued polynomials

$$\mathbb{RT}_0 = [\mathbb{P}_0]^d \oplus x\mathbb{P}_0, \tag{3.42}$$

where $\mathbb{P}_0$ is the space of constant function on K. Clearly, the dimension of $\mathbb{RT}_0$ is $d+1$. For $v \in \mathbb{RT}_0$, the local degrees of freedom σ_i are chosen to be the value of the flux of the normal component of v across the faces of K, so for $0 \leq i \leq d$:

$$\sigma_i(v) = \int_{F_i} v \cdot n_i,$$

where F_i is a face of K and n_i is the relative outward unit normal.

The local base functions are:

$$\vartheta_i(x) = \frac{1}{d|K|}(x - a_i), \;\; 0 \leq i \leq d, \tag{3.43}$$

where a_i are the nodes of K. Indeed, $\vartheta \in \mathbb{RT}_0$ and $\sigma_j(\vartheta_i) = \delta_{ij}$ for $0 \leq i, j \leq d$. Note that the normal component of a local base function is constant on the corresponding face and is zero on the other faces. A conventional representation of the degrees of freedom of the Raviart-Thomas finite element is shown in Figure 3.1. An arrow denotes the flux of the normal component. These finite elements have been introduced by Raviart and Thomas [30].

The domain of the local interpolation operator can be taken to be

$$V^{\text{div}}(K) = \{v \in [L^p(K)]^d; \nabla \cdot v \in L^s(K)\},$$

for $p > 2, s \geq q, \frac{1}{q} = \frac{1}{p} + \frac{1}{d}$. The local Raviart-Thomas interpolation operator is then defined as follows:

$$\Pi_K^{RT} : V^{\text{div}}(K) \ni v \to \Pi_K^{RT} v = \sum_{i=0}^{d} \underbrace{\left(\int_{F_i} v \cdot n_i \right)}_{\sigma_i(v)} \vartheta_i \in \mathbb{RT}_0. \tag{3.44}$$

Note that Raviart-Thomas elements are d-dimensional linear functions on K.

3.5 The Mixed Finite Element Method (MFEM)

We now recall the saddle-point formulation of the fluid flow in porous media problem (3.17). Since the bilinear form (3.18) is continuous and coercive ($\boldsymbol{\eta}$ is positive definite), the bilinear form (3.19) is continuous, the linear form (3.20) is continuous, the *inf-sup* condition holds for V_0 and Q spaces, then for Theorem 3.1 problem (3.17) has a unique solution.

For a numerical approximation of (3.8), we need to choose two finite dimensional subspace of V and Q. We set

$$\begin{aligned} &Q_h = X_h^0 = \{q \in Q : q|_K \in \mathbb{P}_0(K)\ \forall K \in \mathcal{T}_h\}, \\ &V_h = \{\mathbf{v} \in V : \mathbf{v}|_K \in \mathbb{RT}_0(K) \forall K \in \mathcal{T}_h\}. \end{aligned}$$

To impose the boundary condition on the velocity we use the Nitsche's penalization method discussed in [7]. Finally the discrete approximation of problem (3.17) is:
Find $(\mathbf{u}_h, p_h) \in V_h \times Q_h$ such that

$$\begin{cases} a_h(\mathbf{u}_h, \mathbf{v}_h) + b(\mathbf{v}_h, p_h) = \mathcal{F}_h(\mathbf{v}_h) & \forall \mathbf{v}_h \in V_h, \\ b(\mathbf{u}_h, q_h) = 0 & \forall q_h \in Q_h, \end{cases} \tag{3.45}$$

where

$$a_h(\mathbf{u}_h, \mathbf{v}_h) = \int_\Omega \boldsymbol{\eta}\mathbf{u}_h \cdot \mathbf{v}_h \,\mathrm{d}\Omega + \int_{\Gamma_D} \gamma h^{-1}(\mathbf{u}_h \cdot \mathbf{n})(\mathbf{v}_h \cdot \mathbf{n}) \,\mathrm{d}\sigma, \tag{3.46}$$

$$b(\mathbf{v}_h, p_h) = -\int_\Omega p_h \nabla \cdot \mathbf{v}_h \,\mathrm{d}\Omega, \tag{3.47}$$

$$\mathcal{F}_h(\mathbf{v}_h) = -\int_{\Gamma_D} p_0 \mathbf{v}_h \cdot \mathbf{n} \,\mathrm{d}\sigma \int_{\Gamma_D} \gamma h^{-1} u_0 (\mathbf{v}_h \cdot \mathbf{n}) \,\mathrm{d}\sigma. \tag{3.48}$$

Notice that we have set $u_0 = 0$, then $\mathcal{F}_h(\mathbf{v}_h) = \mathcal{F}(\mathbf{v}_h)$.

Again we discover the typical structure of a saddle-point problem and it is possible to prove that all the required conditions for the forms (3.46), (3.47) and (3.48) hold, so the *inf-sup* condition is satisfied and a unique solution exists. Lastly, a convergence estimation for the approximate solution holds. An exaustive discussion of error analysis of each particular mixed formulation can be found in [5].

As we have done for a generic saddle-point problem in (3.30), writing velocity as a linear combination of the Raviart-Thomas base functions and pressure as a linear combination of piecewise constant base functions and properly substituting test functions we finally arrive to the algebraic formulation:

$$\begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix} \begin{pmatrix} \mathbf{u} \\ \mathbf{p} \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ \mathbf{0} \end{pmatrix} . \tag{3.49}$$

This is the linear system that has to be solved for each local problem, giving us the $\mathbb{RT}_0$ degrees of freedom for the velocity field and the $\mathbb{P}_0$ degrees of freedom for the pressure field.

Chapter 4

Implementation and Algorithms

In this chapter we present the code and the techniques developed for upscaling over non-matching grids. In the first instance, existing tools and compatibility issues are presented. We give special attention to the use of standard grid format so we will be able to analyse the upscaled field quality using commercial software. Then, we discuss in detail the code features and the workflow structure.

4.1 Compatibility with existing tools

One of the major issue is to provide a full compatibility with the most common commercial software for reservoir simulation. The most used reservoir simulator in oil and gas industry is *Eclipse* provided by *Schlumberger*. With over 30 years of continuous development, *Eclipse* simulator is a feature-rich and comprehensive reservoir simulator on the market covering the entire spectrum of reservoir models.

4.1.1 The *Corner Point* grids

The standard type of computational grid used in oil and gas industry, in particular in *Eclipse* simulator, is the *Corner Point* grid. The geometry of these grids is rather flexible and can be used to accurately describe the free surface. Moreover geological formations are usually arranged in layered structures and a *Corner Point* grid can easily follow the different layers of rock.

The *Corner Point* grids are three-dimensional *structured* grids, com-

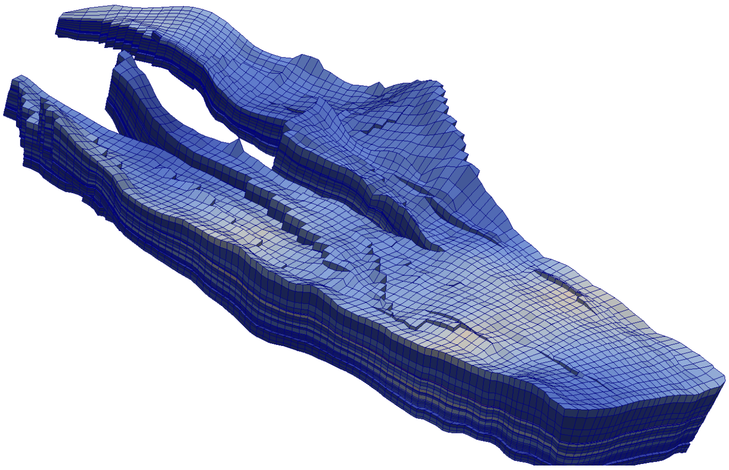


Figure 4.1: Example of Corner Point grid with fractures.

posed of hexahedral cells. A three-dimensional *structured* grid is characterized by the following features:

1. all cells are identified by eight nodes, i.e. they are topologically equivalent to a cube,
2. the cells are logically organized in a regular scheme, such that each cell position in the grid is uniquely determined by its (i, j, k)-indices, where i, j and k vary between 1 and N_x, N_y or N_z respectively.

The difference between a *Corner Point* grid and a generic *Structured* grid is that in a *Corner Point* grid the corners cannot be completely arbitrary but they lie on given lines called *Pillars*: a *Pillar* is a straight non-horizontal line defined by two points and it is identified by its (i, j)-indices, where i and j vary between 1 and $N_x + 1$ or $N_y + 1$ respectively. Usually the index i is increasing along x-axis as the index j with y-axis; for example, in a Cartesian grid (that is a particular *Corner Point* grid), the *Pillars* would be the vertical lines.

4.1.2 Eclipse Interface

Eclipse stores the grids in *Corner Point* format. The format used by *Eclipse* is defined through the two keywords `COORD` and `ZCORN`.

The `COORD` is used to define the *Pillars*, each *Pillar* is uniquely defined by two points. Since the line cannot be horizontal, the two points are denoted by 1 for the top one and 2 for bottom one. If we consider a grid with dimensions $N_x \times N_y \times N_z$, we have $(N_x + 1) \times (N_y + 1)$ *Pillars*. Thus, to define all the *Pillars* we need to provide $6 \times (N_x + 1) \times (N_y + 1)$ values. *Pillar* are given in sequence, cycling over i for every j:

COORD

$$\left.\begin{array}{lll}
x_1^{(1,1)} & y_1^{(1,1)} & z_1^{(1,1)} \\
x_2^{(1,1)} & y_2^{(1,1)} & z_2^{(1,1)} \\
\vdots & & \\
x_1^{(N_x+1,1)} & y_1^{(N_x+1,1)} & z_1^{(N_x+1,1)} \\
x_2^{(N_x+1,1)} & y_2^{(N_x+1,1)} & z_2^{(N_x+1,1)}
\end{array}\right\} 1^{st} \text{ row of pillars } (j = 1)$$

$$\begin{array}{lll}
x_1^{(1,2)} & y_1^{(1,2)} & z_1^{(1,2)} \\
x_2^{(1,2)} & y_2^{(1,2)} & z_2^{(1,2)} \\
\vdots & & \\
x_1^{(N_x+1,N_y+1)} & y_1^{(N_x+1,N_y+1)} & z_1^{(N_x+1,N_y+1)} \\
x_2^{(N_x+1,N_y+1)} & y_2^{(N_x+1,N_y+1)} & z_2^{(N_x+1,N_y+1)}
\end{array}$$

/

The depths of cells corners are defined by the keyword ZCORN. Since each cell is defined by 8 values, if we consider a grid with dimensions $N_x \times N_y \times N_z$, we will have to provide $8 \times N_x \times N_y \times N_z$ values. Corner depths are given in sequence, cycling over i fore every j and then for every k:

ZCORN

$$\left.\begin{array}{llllll}
z_1^{(1,1,1)} & z_2^{(1,1,1)} & \ldots & \ldots & z_1^{(N_x,1,1)} & z_2^{(N_x,1,1)} \\
z_3^{(1,1,1)} & z_4^{(1,1,1)} & \ldots & \ldots & z_3^{(N_x,1,1)} & z_4^{(N_x,1,1)} \\
z_1^{(1,2,1)} & z_2^{(1,2,1)} & \ldots & \ldots & z_1^{(N_x,2,1)} & z_2^{(N_x,2,1)} \\
z_3^{(1,2,1)} & z_4^{(1,2,1)} & \ldots & \ldots & z_3^{(N_x,2,1)} & z_4^{(N_x,2,1)} \\
\vdots & & & & & \\
z_1^{(1,N_y,1)} & z_2^{(1,N_y,1)} & \ldots & \ldots & z_1^{(N_x,N_y,1)} & z_2^{(N_x,N_y,1)} \\
z_3^{(1,N_y,1)} & z_4^{(1,N_y,1)} & \ldots & \ldots & z_3^{(N_x,N_y,1)} & z_4^{(N_x,N_y,1)} \\
z_5^{(1,1,1)} & z_6^{(1,1,1)} & \ldots & \ldots & z_5^{(N_x,1,1)} & z_6^{(N_x,1,1)} \\
z_7^{(1,1,1)} & z_8^{(1,1,1)} & \ldots & \ldots & z_7^{(N_x,1,1)} & z_8^{(N_x,1,1)} \\
\vdots & & & & & \\
z_5^{(1,N_y,1)} & z_6^{(1,N_y,1)} & \ldots & \ldots & z_5^{(N_x,N_y,1)} & z_6^{(N_x,N_y,1)} \\
z_7^{(1,N_y,1)} & z_8^{(1,N_y,1)} & \ldots & \ldots & z_7^{(N_x,N_y,1)} & z_8^{(N_x,N_y,1)}
\end{array}\right\} 1^{st} \text{ layer of cells } (k = 1)$$

$$\begin{array}{llllll}
z_1^{(1,1,2)} & z_2^{(1,1,2)} & \ldots & \ldots & z_1^{(N_x,1,2)} & z_2^{(N_x,1,2)} \\
\vdots & & & & & \\
z_7^{(1,N_y,N_z)} & z_8^{(1,N_y,N_z)} & \ldots & \ldots & z_7^{(N_x,N_y,N_z)} & z_8^{(N_x,N_y,N_z)}
\end{array}$$

/

4.1.3 The `CPgrid3D` class

The `CPgrid3D` custom class is the one that stores and manages the *Corner Point* grid in our `C++` environment. The data are stored in the *Eclipse* standard format, that means there are two main vectors for `COORD` and `ZCORN`. It also contains special classes to store and manage permeability and transmissibility fields. Many geometric tools have been developed with the purpose of doing the upscaling of geological fields: for example, we need tools to extract the local problem subgrid for each coarse cell problem. For this reason the function `getMeshBlock` has been developed and it can be called using different options:

```
getfem::mesh getMeshBlock(const CPcell3D &Cell,
          const UInt &ring = 0) const;
getfem::mesh getMeshBlock(const CPcell3D &firstCell,
          const CPcell3D &secondCell,
          const UInt &ring = 0) const;
```

The `CPcell3D` class is the standard container representing a generic cell and the `getfem::mesh` is a class containing the mesh information compatible with the external library `GetFEM++` where the algebraic problem is assembled: we will further present this library in the next chapter. The first call is useful when we do a permeability upscaling, where the local domain is composed by the fine cells lying into the bounding box of the coarse cell: the bounding box is defined as the set of fine grid cells which cover the coarse cells. If we do a transmissibility upscaling we need to solve the problem over the bounding box of two adjacent coarse cells. In both cases we can choose the ring size that will surround the bounding box as explained in Chapter 2.3. Since we want to solve a mixed finite element problem in each local domain, we need to build a tetrahedral grid: the easier and more efficient way is to split each hexahedron belonging to the *Corner Point* grid in six tetrahedrons. By doing this, we will avoid errors due to the interpolation of the fine permeability field on a possibly non matching fine grid.

Another important geometric tool needed for all basic manipulation of the grid is the following function:

```
CPcell3D getCell3D(const Point3D &p) const;
```

this function returns the cell containing the point `p`. In case of cartesian grid, the `getCell3D` function is relatively easy to implement, since we can take advantage of the cartesian structure; otherwise, for generic *Corner Point* grids, this problem requires special and efficient algorithms for the search.

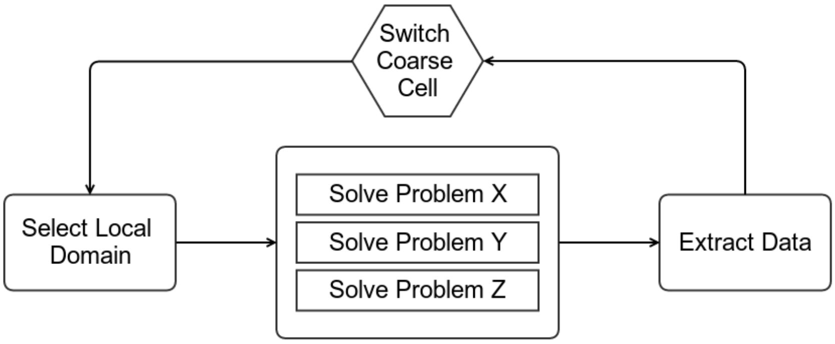


Figure 4.2: Upscaling software flowchart.

We briefly present an algorithm based on [20]: let C be a generic cell of a corner point grid, we suppose that C is topologically an hexahedron, meaning that there is a trilinear map

$$\mathbf{T}_C : \hat{C} \to C \tag{4.1}$$

operating between the reference cube $\hat{C} = [0,1]^3$ and the cell C, with the Jacobian $J_C = \frac{\partial T_C}{\partial \hat{\mathbf{x}}}$ being non singular for every $\hat{\mathbf{x}} \in \hat{C}$. We also define $\hat{\mathbf{x}} = (\hat{x}_i, \hat{x}_j, \hat{x}_k)$.

Then, to check if a given point $\mathbf{x}$ lies in C we can:

- Find $\hat{\mathbf{x}} = (\hat{x}_i, \hat{x}_j, \hat{x}_k) = \mathbf{T}_C^{-1}(\mathbf{x})$;
- Check if $0 \leq i \leq 1$, $0 \leq j \leq 1$ and $0 \leq k \leq 1$.

This algorithm can become complex for the following reasons: the faces of C are non planar moreover the map is non linear and the map is well defined only inside the reference cube $\hat{C}$. To overcome the first problem, a Newton method is employed; if C is hexahedric with planar faces, the Newton method converges in one iteration. The second problem is solved applying the Newton method only if the point lies inside the cell bounding box of C where the map is well defined.

4.2 Structure of the Software

In this chapter, we analyse in detail the structure of the software and its implementation. The code mainly performs three types of operations, that are:

1. solving the local problem,
2. extracting data from the solution,
3. looping over each coarse cell.

We can see in Figure 4.2 the interconnection among the steps; there is a fundamental difference between permeability and transmissibility upscaling: about the first one, the domain of the three problems along each direction is the same, so it can be solved within the same base structure and different bounding conditions, while the transmissibility upscaling needs three different domains in each direction, that are twice the size of the permeability ones. This means that transmissibility upscaling is, in general, more expensive than permeability upscaling.

4.2.1 Solving Local Problem

As discussed in Chapter 3, we want to solve each problem in the local domain using the Mixed Finite Element method. That said, we rely on the external library `GetFEM++`: that is a generic `C++` finite element library which aims to offer the widest range of finite element methods and elementary matrix computations for the approximation of linear or non-linear problems. It can be downloaded from http://home.gna.org/getfem/ where full documentation is provided.

The class dealing with the local problem and providing the interface with the `GetFEM++` library is the `LocalProblem3D` class. The main functions of this class are:

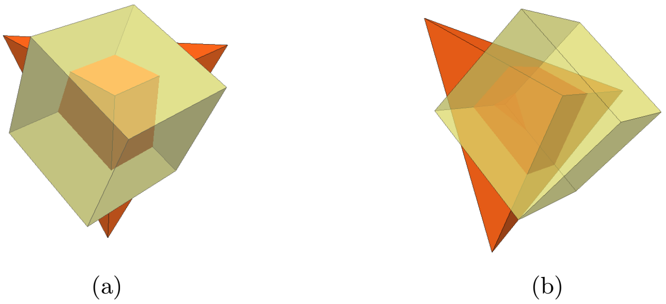
```
vector<Real> computeCoarsePermeability(
              const Geometry::CPgrid3D &fineGrid,
              const Geometry::CPcell3D &coarseCell,
              const UInt &Ring,
              const bool &Cartesian);
Real computeCoarseTransmissibility(
              const Geometry::CPgrid3D &fineGrid,
              const Geometry::CPcell3D &firstCell,
              const Geometry::CPcell3D &secondCell,
              const PbOrientation &DIR,
              const UInt &Ring,
              const bool &Cartesian);
```

The two functions above are similar: given the fine grid (`fineGrid`) and one or two coarse cells, the functions return the results of the upscaling procedure, i.e. the diagonal elements of the cell permeability tensor (all three problems are solved) or the transmissibility of the cells pair interface.

Once the local domain has been extracted from `fineGrid` variable, we have a tetrahedral mesh, represented by the `getfem::mesh` class. Then we can define the finite element spaces ($\mathbb{P}_0$ for pressure and $\mathbb{RT}_0$ for velocity) over the mesh in order to assemble the matrices and solve the problem with the mixed finite element method discussed in Chapter 3.5.

One important software feature is that we can easily choose which kind of boundary condition apply to local problems using the `BcType` flag.

4.2.2 Extract Data from the Solution


(a)

(b)

Figure 4.3: Complex intersection between a hexahedron and a tetrahedron.

Once the problem has been solved, we have the pressure field described by the $\mathbb{P}_0$ degrees of freedom and the velocity field by $\mathbb{RT}_0$ degrees of freedom. We want to compute the upscaled geological variables using equation 2.12 or 2.19. If the coarse cell coincides with a subset of fine grid cells, i.e. we are in the matching case, then it is relatively easy to compute the needed values. However many complications occur in the case of non-matching coarse cell. We have focused in developing a non-matching upscaling technique for the transmissibility field: therefore, the values we are interested in are the mean pressure over the cell:

$$\langle p \rangle_C = \frac{\int_C p \ \mathrm{d}C}{\int_C \mathrm{d}C}, \tag{4.2}$$

where C is an hexahedric coarse cell, and the flux through the interface is:

$$\mathbf{u}|_S = \int_S \mathbf{u} \cdot \mathrm{d}S, \tag{4.3}$$

where S is a coarse cell face.

We can rewrite (4.2) in a discrete form knowing that is $\mathbb{P}_0$ in each tetrahedron. Let k be the index of the k-th tetrahedron, we want to find the fraction of volume $\tilde{x}_k$ inside the hexahedron C. We start by placing each tetrahedron of the mesh into one of these group:

1. the tetrahedron k is inside the coarse cell, $\tilde{x}_k$ is equal to 1,
2. the tetrahedron k is outside the coarse cell, $\tilde{x}_k$ is equal to 0,
3. the tetrahedron k intersects one or more faces of C, $0 < \tilde{x}_k < 1$.

The average pressure can be computed as:

$$\langle p \rangle_C = \frac{\sum_k p_k V_k \tilde{x}_k}{\sum_k V_k \tilde{x}_k} \tag{4.4}$$

where p_k is the local pressure and V_k is the tetrahedron volume. The main problem is how to compute $\tilde{x}_k$ in case of intersections: indeed a generic 3D intersection between tetrahedron and hexahedron could lead to more complex polyhedrons. In order to solve this problem and to avoid the analysis of all possible cases, we have implemented the following algorithm. Let K be the current tetrahedron, we start searching for a set of points $\mathbf{P}_C$, composed by:

- vertices of C that are inside K,
- vertices of K that are inside C,
- intersections between edges of C and faces of K,
- intersections between faces of C and edges of K.

We now need the concept of convex hull: given a subset $A \in \mathbb{R}^d$ the convex hull is defined as the smallest convex set in $\mathbb{R}^d$ containing A. In particular, the convex hull of $\mathbf{P}_C$ in $\mathbb{R}^3$ is the intersection between K and C. To find such convex hull, we use another external library named `QHULL`: this library is one of the most used for convex hull computation and Delaunay triangulation. To compute the convex hull for a given set of points it uses the *Quickhull* algorithm presented in [3], returning a set of tetrahedrons covering the original polyhedron. Finally, we can compute the intersection volume $\tilde{V}$ as the convex hull volume and the fraction of volume inside the cell $\tilde{x}_T = \tilde{V}/V_K$.

We can use a similar approach to discretize the integral in (4.3). In order to avoid integration errors, we implement an integration method that is exact for $\mathbb{RT}_0$ elements. We want to find the area of the intersection between the plain interface S and the k-th tetrahedron. We start by assigning each tetrahedron to one of the following groups:

1. the tetrahedron k does not intersect the interface S,
2. the tetrahedron k intersects the interface S.

A generic 3D intersection between a finite plane and a tetrahedron could generate complex polygons, so it is not easy foreseeing how many edges they could have. Let K be the current tetrahedron, we search a new set of points $\mathbf{P}_S$ composed by:

- vertices of S that are inside K,
- intersections between edges of S and faces of K,
- intersections between S and edges of K.

Again, the convex hull of $\mathbf{P}_S$ (in this case it is 2D) is the intersection we are searching for. As we have seen in Chapter 3.4, the local base functions ϑ_i of Raviart-Thomas element are linear. Let $\tilde{S}$ be the intersection between S and K and $\mathbf{n}_S$ the S normal vector (that is constant): recalling the Raviart-Thomas interpolation function $\Pi_K^{RT}(\cdot)$ defined in Chapter 3.4, and knowing that the velocity field belongs to $\mathbb{RT}_0$ in each tetrahedron (so $\mathbf{u} = \Pi_K^{RT}(\mathbf{u})$), then the function $\tilde{f}(\mathbf{x}) = (\Pi_K^{RT}(\mathbf{u}) \cdot \mathbf{n}_S)(\mathbf{x}) : \tilde{S} \to \mathbb{R}$ is linear and using the mean value theorem we have:

$$\int_{\tilde{S}} \tilde{f}(\mathbf{x})\,\mathrm{d}\tilde{S} = \tilde{f}(\mathbf{R})|\tilde{S}|, \tag{4.5}$$

where $\mathbf{R}$ is the center of mass of $\tilde{S}$ that can be computed as:

$$\mathbf{R} = \frac{1}{|\tilde{S}|}\int_{\tilde{S}} \mathbf{x}\,\mathrm{d}\tilde{S}. \tag{4.6}$$

Moreover, the center of mass of a triangle embedded in $\mathbb{R}^3$ can be easily computed as the vertices average

$$\mathbf{R}_\triangle = \frac{1}{|\triangle|}\int_\triangle \mathbf{x}\,\mathrm{d}\triangle = \frac{\mathbf{t}_1 + \mathbf{t}_2 + \mathbf{t}_3}{3}, \tag{4.7}$$

where the $\mathbf{t}_i$ are the triangle vertices. For more complex polygons, there is no relation between center of mass and vertices average. A workaround to find the center of mass of a more complex polygon is to split it in several triangles, compute their centers of mass and lastly compute a weighted average of these points. Finally, using (4.3) and (4.5) for each tetrahedron, imposing $|\tilde{S}_k| = 0$ for tetrahedrons which does not intersect S, we obtain:

$$\mathbf{v}|_S = \sum_k \tilde{f}(\mathbf{R}_k)|\tilde{\sigma}_k| \tag{4.8}$$

which is an exact quadrature formula for plain surface integrals of $\mathbb{RT}_0$ elements.

4.2.3 The Main Loop

The core of the software is basically a for loop cycling over all the coarse cells. For each cell we need to solve the problems along each direction and extract data as explained above. There is not any correlation for the computation of the upscaled quantities among coarse cells. In fact each local problem requires only a subset of the original fine grid.

For these reason it is natural to think of a parallel version of the software. In parallel computing, an *embarrassingly parallel* workload, or *embarrassingly parallel* problem, is one for which little or no effort is required to separate the problem into a number of parallel tasks. We can think at the upscaling procedure as being part of the *embarrassingly parallel* class of problem. For example, let n_t be the number of threads, we could split the fine grid in n_t subgrids, for example along x-axis. Therefore each thread would process only one subgrid. However, we should pay attention if the ring feature has been used. Using a ring size of ρ, we should extend the subgrids dimension adding ρ cells along x-axis in both the ways resulting in n_t overlapping subgrids.

Chapter 5

Numerical Results

In this chapter, we will perform and test the upscaling of several fields with the methods and software described in the previous chapters. All flow simulations have been done using the commercial software *Eclipse*. This software uses a finite volume method with two-points flux approximation to solve material and energy balance equations modelling a subsurface petroleum reservoir. In order to solve the fluid flow problem, several input data must be set; we can distinguish them into two type: field data and external data. Some examples of external data are the injection and production well positions, which can be varied for feasibility studies over new reservoirs or to optimize the oil production in the existing ones. Field data are quantities such as porosity, permeability and transmissibility. As we have seen in Chapter 2 they characterize the reservoir. The more this quantities are matching the real physical fields, the more accurate the reservoir simulation is.

5.1 Validation Tests

5.1.1 Linear Permeability

With this first step, we want to assess the correct behaviour of the code. To this aim, we provide a case study where the solution is known analytically. Thus our first case test is shown in Figure 5.1: it is a cubical Cartesian grid made by four z-layers 1 meter thick with an isotropic permeability k_i of 2,4,6, and 8 millidarcy respectively.

In the case of permeability upscaling along the x and y axes, the flux is parallel to the grid layers, then the upscaled permeability $\mathbf{K}^*_{xx}$ and $\mathbf{K}^*_{yy}$

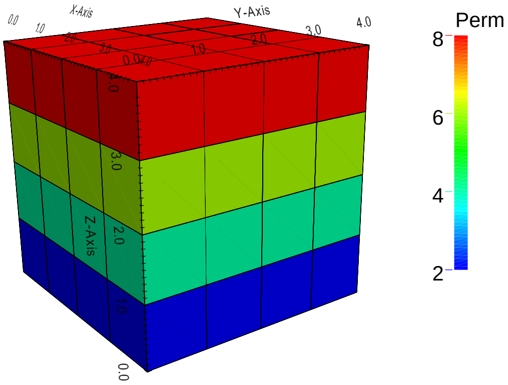


Figure 5.1: Cube with linear permeability.

should be equal to the average value of the permeability

$$\mathbf{K}^*_{xx} = \mathbf{K}^*_{yy} = \frac{1}{4}\sum_{i=1}^{4} k_i = 5. \tag{5.1}$$

On the contrary, a permeability upscaling along z axis leads to a flux orthogonal to the layers. The analytical solution for the upscaled permeability $\mathbf{K}^*_{zz}$ is equal to the harmonic average of the permeability, namely

$$\mathbf{K}^*_{zz} = \left(\frac{1}{4}\sum_{i=1}^{4}\frac{1}{k_i}\right)^{-1} = 3.84. \tag{5.2}$$

Note that it is possible to obtain an anisotropic permeability field starting from a isotropic permeability field. To better understand the equations (5.1) and (5.2), we note that the Darcy problem could be associated to an electrical circuit problem, where the permeability is associated to the conductance, the velocity to the electric intensity current and the pressure gradient to the potential difference, therefore the upscaling along x and y corresponds to a circuit with conductances in parallel (resistors in series), while along z we have conductances in series (resistors in parallel).

We test our upscaling procedure over the above cube using both open and closed boundary conditions. Using closed conditions we obtain the following tensor:

$$\mathbf{K}^*_C = \begin{bmatrix} 5 & 1.85 \cdot 10^{-16} & 2.72 \cdot 10^{-16} \\ 7.82 \cdot 10^{-16} & 5 & -4.032 \cdot 10^{-16} \\ 3.01 \cdot 10^{-16} & -1.48 \cdot 10^{-16} & 3.84 \end{bmatrix} \tag{5.3}$$

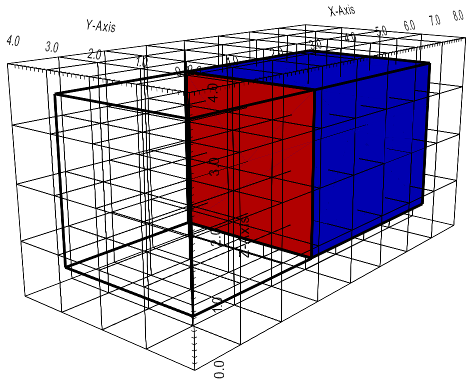


Figure 5.2: Computational grid for mismatch test.

In the case of open condition the matrix is:

$$\mathbf{K}_O^* = \begin{bmatrix} 4.9994 & -0.0006 & -0.0039 \\ -0.0006 & 4.9994 & -0.0076 \\ -0.0039 & 0.0076 & 4.5818 \end{bmatrix} \tag{5.4}$$

Note that both $\mathbf{K}_C^*$ and $\mathbf{K}_O^*$ are symmetric as a consequence of the linear least square approach used to compute the upscaled matrix (see Chapter 2.2.3).

Using closed boundary conditions we obtain an almost perfect matching with the analytical results. Instead, using open boundary conditions the permeability computed along z-axis is overestimated, plus we obtain significant values for extra-diagonal elements of the permeability tensor. However, this does not mean that upscaling using closed boundary condition is always better, because the local upscaled permeability strictly depends of the adjacent field properties: for example, if a coarse cell is surrounded by high permeability values, a method that slightly overestimate permeability would suit better, since at fine scale the global solution would flow through lateral faces encouraging the use of open boundary conditions. This is a reason that leads to the use of global or local-global methods (see 2.4).

5.1.2 Non-Matching Grids Test

Now we want to test our software using a coarse cell not aligned with the fine grid. The simulation setup shown in Figure 5.2 is now made by a Cartesian fine grid with $8 \times 4 \times 4$ cells where each of them is a perfect $1m \times 1m \times 1m$ cube. Then we define two adjacent coarse cells whose faces cut the fine grid

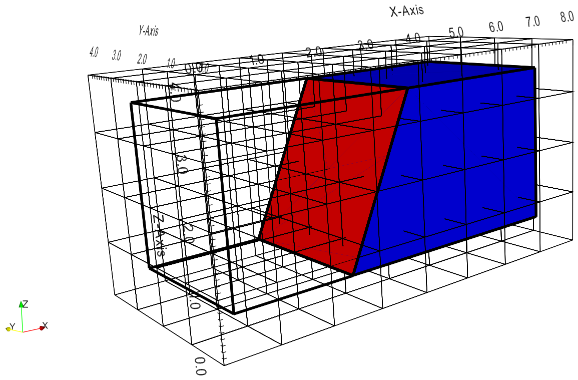


Figure 5.3: Computational grid for mismatch test with a sloped interface.

cells. We set a uniform velocity field directed along the x axis equal to $1m/s$ and we are interested in computing the flux through the coarse cells interface (the red face in Figure 5.2) to validate the algorithm described in Chapter 4.2.2. Note that the interface is crossing a lot of fine mesh tetrahedrons. Since the velocity is constant and the interface area is $9m^2$, the analytical flux is $9m^3/s$.

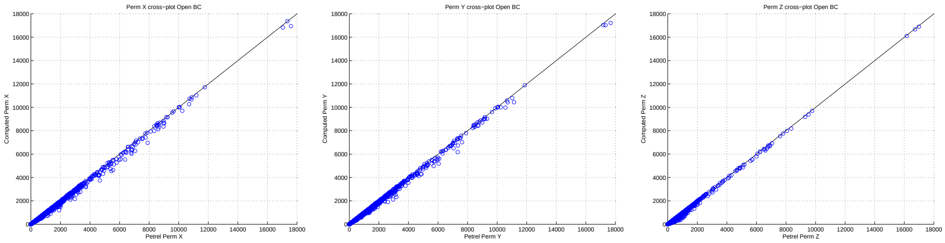
Our code is able to manage this configuration, returning a solution with negligible errors (less than 0.0001%).

For a more complex test, we use the setup shown in Figure 5.3 where we have an inclined interface. Since the velocity is still constant and parallel to the x-axis, the flux over the new interface will be unchanged. Again the software is able to extrapolate the right information from fine scale properties.

5.1.3 Existing Upscaling Software

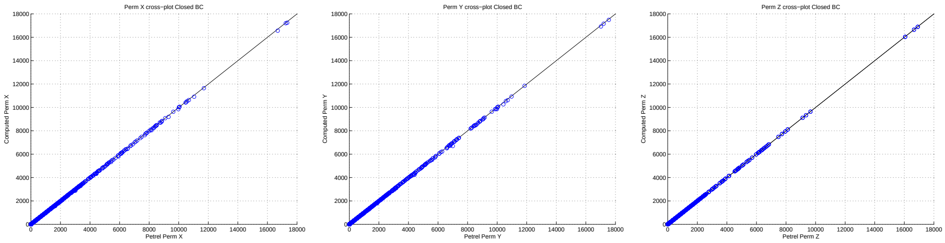
The commercial software *Petrel*, mainly used for corner point grid manipulation, is able to do a flow based upscaling of the permeability field over matching Cartesian grids. The only difference from our approach is that *Petrel* use a Finite Volume approximation to solve the local problems.

In Figures 5.4 and 5.5 are shown the cross-plot charts comparing the permeability field computed by *Petrel* and our software. To generate the upscaled permeability field, we have used a Cartesian coarse grid where each cell is double size compared to the fine scale (the coarse grid size is $55 \times 15 \times 15$). The cross-plot using closed boundary conditions does not reveal substantial differences between the two methods. However, in the case



(a) x (b) y (c) z

Figure 5.4: Cross-plot of the permeability using open BC.



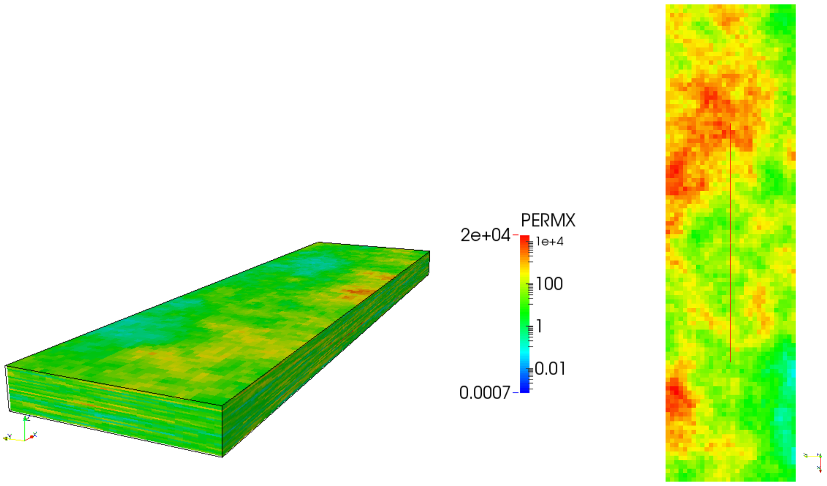
(a) x (b) y (c) z

Figure 5.5: Cross-plot of the permeability using closed BC.

of open boundary conditions, the permeability field computed using mixed finite element approximation seems to be slightly lower, leading to a less general reservoir connectivity. This fact is partially due to the higher degrees of freedom we have using a mixed finite element approach, allowing more fluid to flow through the lateral faces during the local problems resolution. In general, a lower permeability using open condition should be better since we expect an overestimation of the field connectivity.

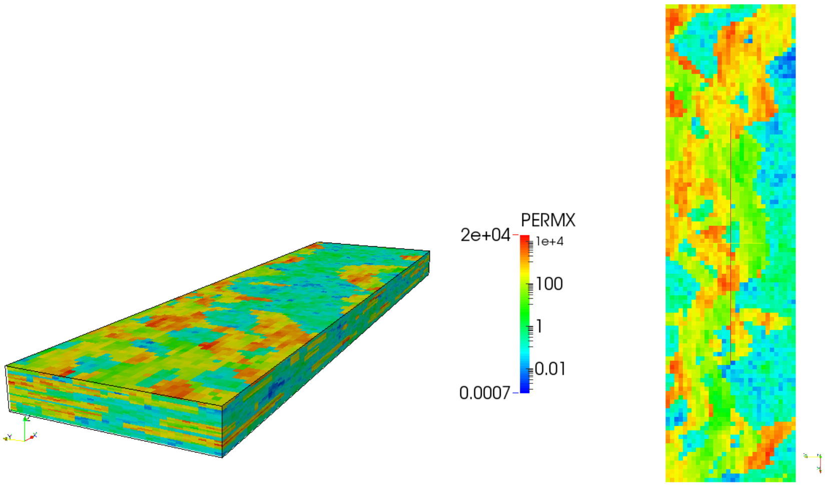
5.2 Real Reservoir Cases

The SPE Comparative Solution Project (http://www.spe.org/web/csp/) is a comparative solution projects organised by the Society of Petroleum Engineers. The purpose of the projects has been to provide benchmark datasets which can be used to compare the performance of different simulators or algorithms. In particular, the SPE Comparative Solution Project provides a reservoir model for benchmark purposes. The model consists of part of the Brent sequence. The top part of the model is the Tarbert formation, it is a representation of a prograding near shore environment. The lower part is fluvial. The model is described on a regular Cartesian grid.



(a) $\mathcal{G}_1$ (b) Sector

Figure 5.6: SPE-CSP subgrid $\mathcal{G}_1$, to the right side the middle layer seen from the top.



(a) $\mathcal{G}_2$ (b) Sector

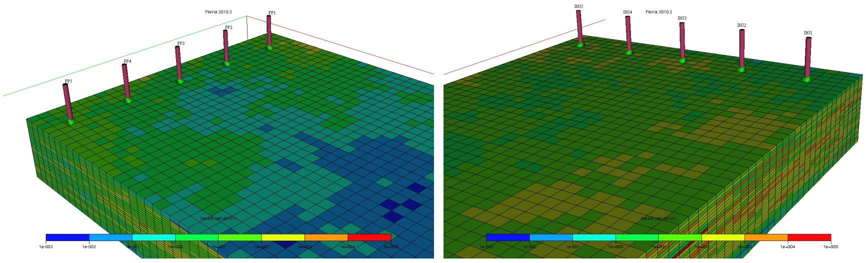
Figure 5.7: SPE-CSP subgrid $\mathcal{G}_2$, to the right side the middle layer seen from the top.

Production PP	Pos(i,j)	Injection INJ	Pos(i,j)
PP1	$(4,4)$	INJ1	$(107,27)$
PP2	$(4,10)$	INJ2	$(107,21)$
PP3	$(4,16)$	INJ3	$(107,15)$
PP4	$(4,22)$	INJ4	$(107,9)$
PP5	$(4,28)$	INJ5	$(107,3)$

Table 5.1: Wells positions for fine grid simulation.

In order to test our upscaling software, we have extracted two sub-grids from the SPE model: one from the top part, that we name $\mathcal{G}_1$, and one from the lower part, $\mathcal{G}_2$. Both grids are composed by $110 \times 30 \times 30$ (99000 cells) cells and each cell is 20ft× 10ft× 2ft (roughly $6m \times 3m \times 0.6m$). Then the grids measure 2200ft× 300ft× 60ft ($660m \times 90m \times 18m$). The permeability field of grid $\mathcal{G}_1$ shown in Figure 5.6(a) does not exhibit any particular pattern that could force the fluid flow over preferential paths. Conversely, the grid $\mathcal{G}_2$ shown in Figure 5.7(a) is characterized by high permeability channels forcing the flow. In Figures 5.6(b) and 5.7(b) we show two sectors of the grids orthogonal to the z-axes. Since the upscaling could be interpreted as an homogenization of the fine grid quantities, the upscaling of $\mathcal{G}_2$ will be more challenging.

In all our test cases we simulate the injection of water into an oil filled reservoir. Moreover the two phases are defined as incompressible and immiscible fluids with the same physical properties. The result is practically a single phase simulation that permits to track the water displacements over the fine and coarse grids easily. This type of simulation is the so called *tracer* simulation.



(a) PP position

(b) INJ position

Figure 5.8: Production and Injection wells position.

The wells setup is shown in Figure 5.8 and it is the same for all tests and in both grids. There are five aligned injection wells (IW) at one edge of the reservoir and five production wells (PW) at the opposite. All wells are 30 cells deep, covering the entire domain along z-axis. The exact wells positions are listed in Table 5.1.

The comparison between fine grid and coarse grid simulation is not trivial: we need to identify some fundamental values to look for. For this reason we introduce the *Field Water Cut* (FWCT) defined as the ratio of the water extracted from the production wells (V_{pw}) compared to the volume of total liquids produced ($V_{pt} = V_{pw} + V_{po}$ where V_{po} is the oil produced at the wells). In our simulations this quantity refers to the ratio of tracer produced compared with total production, so

$$\mathrm{FWCT} = \frac{V_{pw}}{V_{pt}}. \tag{5.5}$$

The FWCT is a dimensionless variable, thus it varies within $[0, 1]$. We could also define a similar quantity for the oil produced:

$$\mathrm{FOPR} = \frac{V_{po}}{V_{pt}}. \tag{5.6}$$

Since the initial condition sets the reservoir completely filled with oil, the FWCT will be zero until the first drop of water reaches the production wells and for this reason there will be a certain amount of time where only oil is produced (FOPR$= 1$).

Another important parameter to take into account is the Field Water Injection Rate (FWIR). Flow results are reported in terms of wells injection and production. Eclipse expresses injection rates as the water volume pumped into the reservoir in Sm^3 per day (Sm^3/Days). Sm^3 are Standard cubic meter, a derived unit of volume. It refers to a cubic meter at a specified standard temperature and pressure. In oil and gas industry standard conditions are usually 15°C for temperature and 100kPa for pressure. Since we are considering incompressible and immiscible fluids and we fix the wells pressure, the water injection rate will be constant. We can also define a specific injection rate for each well: we will refer to it as Well Water Injection Rate (WWIR).

One big issue in the comparison between fine grid and coarse grid results is the well positions: we must choose a coordinate (i, j) in which each well will be placed therefore it is impossible to place the wells at the same place in both fine and coarse grids. To overcome this problem, we use the local grid refinement (LGR) that allows enhanced grid definition in areas needing a higher level of simulation accuracy, such as near wells or in areas of complex

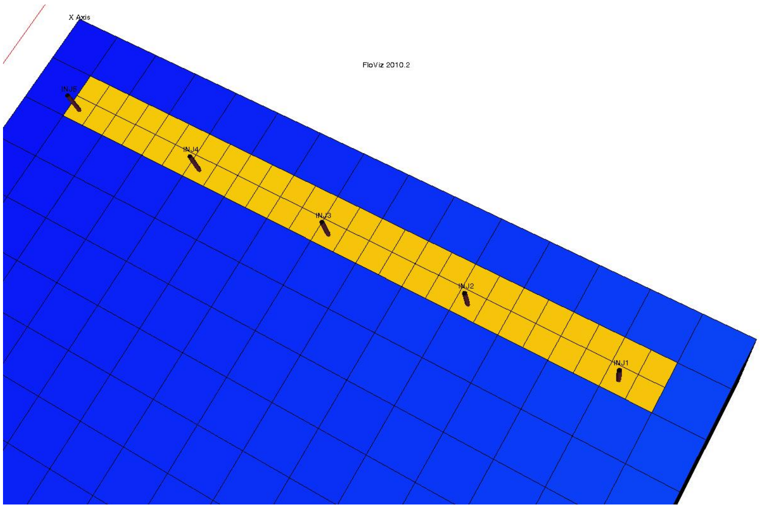


Figure 5.9: Example of local grid refinement.

geology. In our case, we use the refinement to place the wells of the coarse simulation into the same position of the fine one. An example is shown in Figure 5.9. In all the tests below a proper grid refinement has been used so the errors reported are generated by only the upscaling procedure, avoiding the component due to the near well upscaling approximation.

5.2.1 Coarse Scale Analysis

Now we want to analyse the behaviour of the upscaling procedure varying the dimension of the coarse grid cells: given a Cartesian fine grid with $n_x \times n_y \times n_z$ cells and a Cartesian coarse grid with $n_x^c \times n_y^c \times n_z^c$ cells, we define the directional scale factors as

$$s_x = \frac{n_x^c}{n_x}; \; s_y = \frac{n_y^c}{n_y}; \; s_z = \frac{n_z^c}{n_z}.$$

If $s_x = s_y = s_z$, we can define the isotropic scale factor s_{iso}. For example, if N_{fine} is the number of cells of the fine grid, if we use a scale factor $s_{iso} = 3$ the coarse grid has $N_{fine}/27$ cells.

Then, given a fine grid reservoir (in our cases $\mathcal{G}_1$), we can generate a coarse grid with a fixed scale factor. We note that using integer scale factors leads to the generation of a matching coarse grids.

In Figure 5.10 we can see the Field Water Cut as defined in (5.5) for several realizations using different s_{iso} for the coarse grid generation. We have used both open and closed conditions for boundary treatments in transmissibility upscaling. Note we have used a proper local grid refinement for each

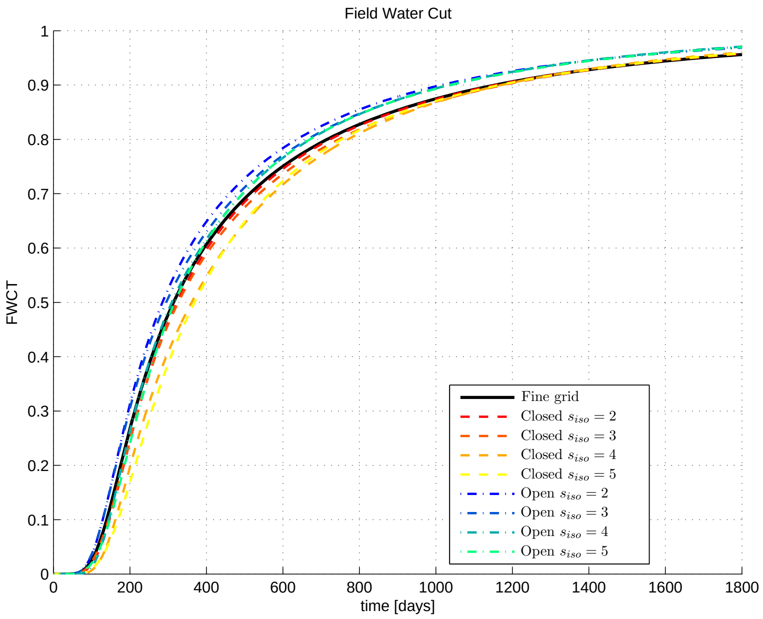


Figure 5.10: Upscale analysis: Field Water Cut for $\mathcal{G}_1$.

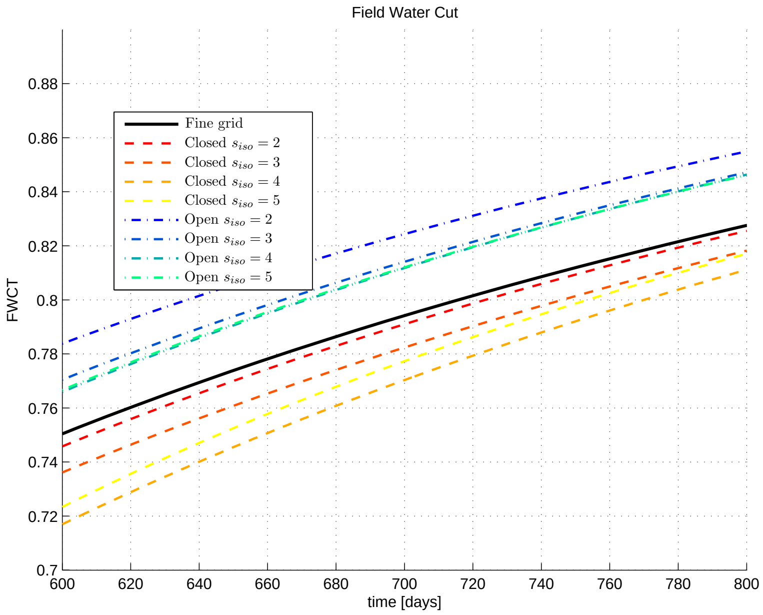
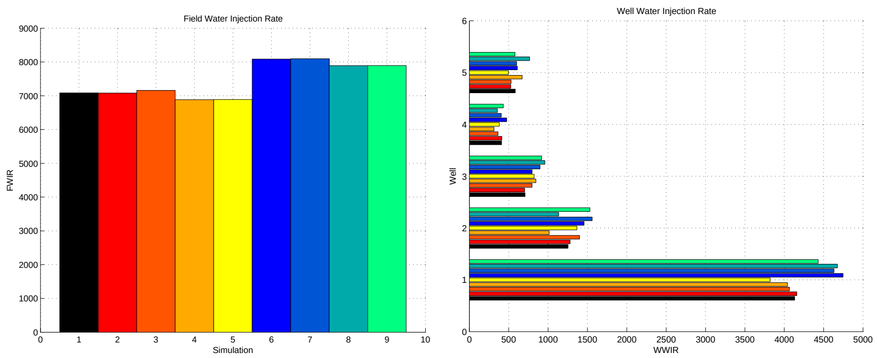


Figure 5.11: Upscale analysis: Field Water Cut zoom $\mathcal{G}_1$.

coarse grid. The first observation we can do is that the Field Water Cut of the upscaled coarse grids using open boundary conditions is significantly greater than the fine grid one. Conversely, using closed boundary conditions, the Field Water Cut is lower than the reference one. This is in agreement with the preliminary tests since simulations using open boundary condition generally provide higher values for permeability and transmissibility. Another finding is that, even when using large coarse scale, the solution is still in good agreement with the fine one if we consider that using $s_{iso} = 5$ each coarse cell replaces 125 fine cells, significantly decreasing the total degrees of freedom and speeding up the simulation.



(a) FWIR

(b) WWIR

Figure 5.12: Upscale analysis: Water Injection Rate for $\mathcal{G}_1$

In Figure 5.12(a) we represent the Field Water Injection Rate. The realizations using closed boundary conditions seem to represent more accurately the FWIR of the fine grid. However with both open and closed boundary conditions the solution is not affected by the coarse scale. In other words it seems like this kind of upscaling is particularly suitable for large coarsening levels.

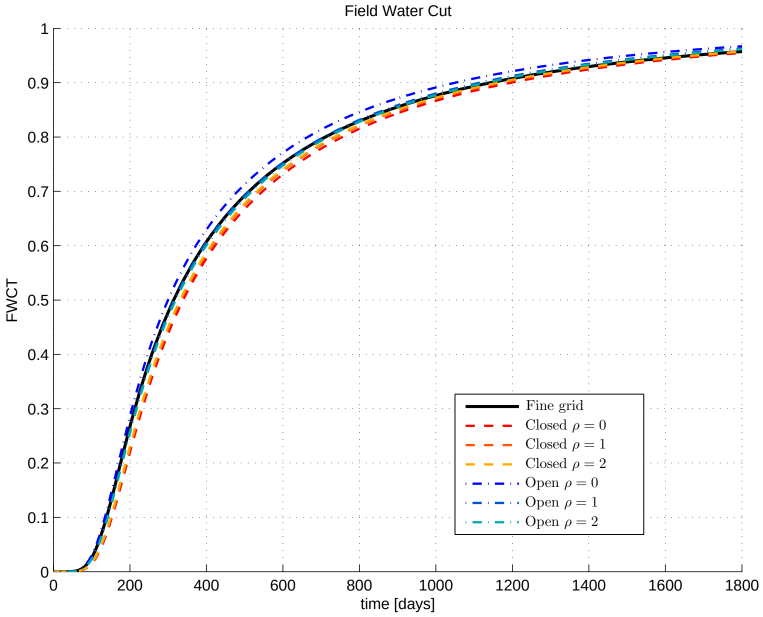


Figure 5.13: Ring analysis: Field Water Cut for $\mathcal{G}_1$.

5.2.2 Ring Analysis

Another important feature we have introduced is the possibility to extend the local problem domain. This is named ring (ρ) and it was introduced in 2.3. We now extend the local domain in each direction adding ρ rows of cells. Doing this, the influence of the boundary conditions decreases leading to a more reliable solution. However the computational cost increases for each local problem.

We use the standard setup and, starting from a $2 \times 2 \times 2$ coarse grid, we upscale the transmissibility field using different values of ρ and different boundary conditions.

We start analysing the ring influence for the upscaling of $\mathcal{G}_1$. In Figure 5.13, and in the detail shown in Figure 5.14, we can see that the Field Water Cut seems to converge to the fine grid FWCT in both open and closed boundary conditions cases. Theoretically, if the ring size tends to infinite, the transmissibility field is not affected by the boundary conditions and the solutions using open or closed conditions for the upscaling converge.

In Figure 5.15(a) the Field Water Injection Rate shows the same converging behaviour, decreasing the approximation errors due to upscaling for grater values of ρ. This is also true for each singular well as it is shown in Figure 5.15(b).

We now apply the same ring analysis for $\mathcal{G}_2$. This time, as shown in

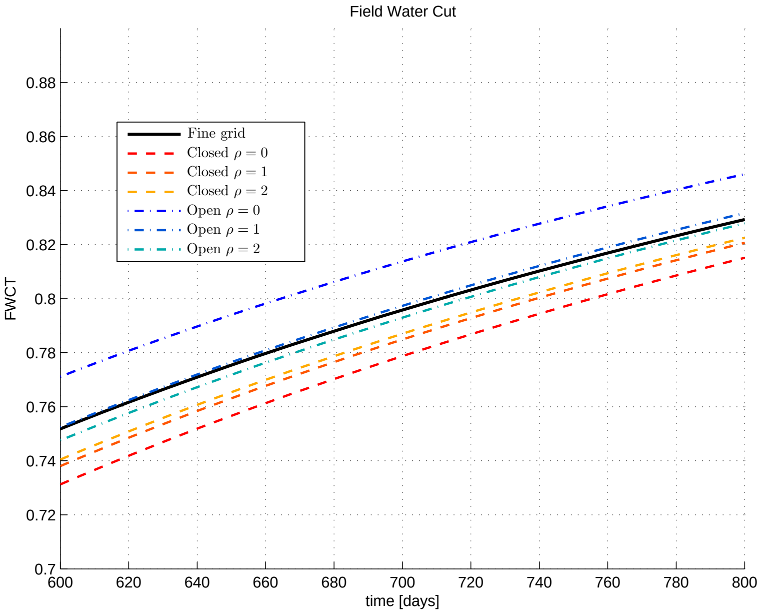
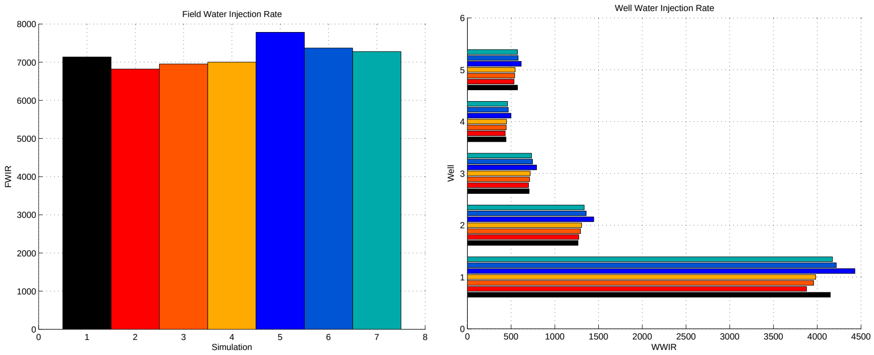


Figure 5.14: Ring analysis: Field Water Cut zoom for $\mathcal{G}_1$.



(a) FWIR

(b) WWIR

Figure 5.15: Ring analysis: Water Injection Rate for $\mathcal{G}_1$.

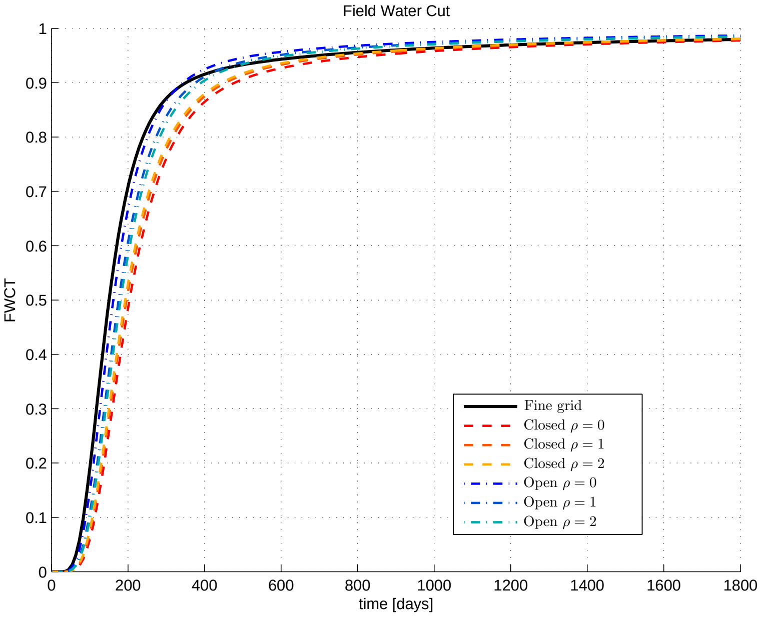


Figure 5.16: Ring analysis: Field Water Cut for $\mathcal{G}_2$.

Figure 5.16, the solutions are qualitatively less accurate with respect to the previous ones. This is in part due to an evident delay of the begin of water production. The water pumped into the injection wells arrives slightly later to the production wells, and only when this happens the water cut begin to grow. A valid explanation of these behaviours could be the spreading of high transmissibility areas of the reservoir field due to the upscaling. For example, if a coarse cell lies on the boundary of an high transmissibility channel, since the upscaling assigns an intermediate value for the coarse transmissibility, the channel will increases. Then the fluid velocity inside the channel decreases and consequently the water phase needs more time in order to cross the whole reservoir.

However, the Field Injection Rate shown in Figure 5.18(a) does not seem to suffer of systematic errors converging to the fine grid solution, though slower than $\mathcal{G}_1$, for both closed and open boundary conditions. The same behaviour for each well is shown in Figure 5.18(b).

5.2.3 Structured Displacement Analysis

Finally, we have tested our upscaling methods on non-matching grids. At first, after a 2×2×2 Cartesian coarse grid has been generated, internal points are moved along the same direction to generate a non matching coarse grid as shown in Figure 5.19. Since, to avoid errors, the well positions should be

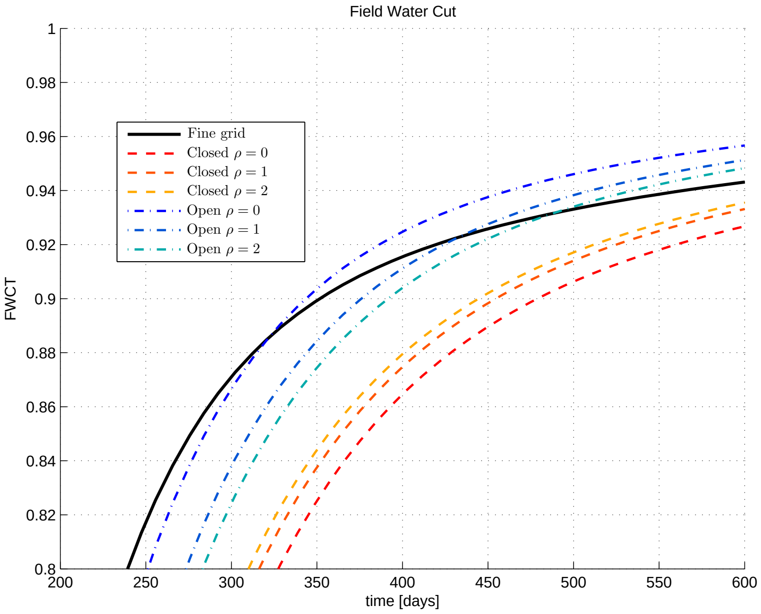
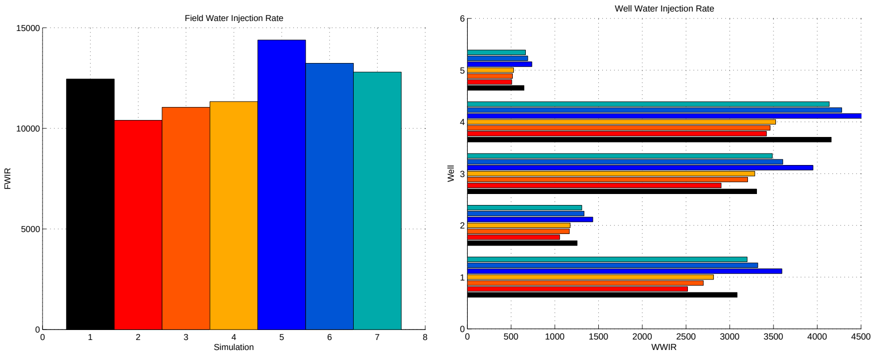


Figure 5.17: Ring analysis: Field Water Cut zoom for $\mathcal{G}_2$.



(a) FWIR

(b) WWIR

Figure 5.18: Ring analysis: Water Injection Rate for $\mathcal{G}_2$.

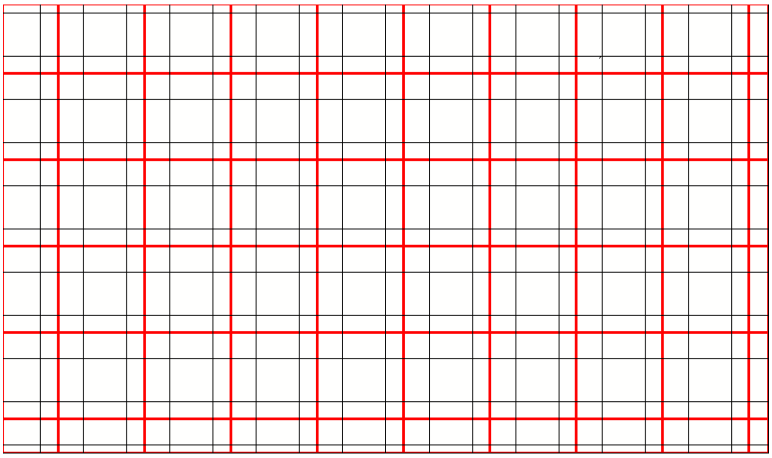


Figure 5.19: Example of structured displacement (red grid) seen from the top.

the same in the coarse and fine grids, we do not move the points that belong to the first two and last two rows of cells along the x-axis. Moreover, points belonging to the boundary are fixed too.

That said, internal points move along the direction passing through the points $(0,0,0)$ and (L_x, L_y, L_z) where L_i are the grid dimensions along each direction. Finally we define a parameter $\delta_s > 0$ indicating the displacement intensity: δ_s is the ratio between the displacement length and the cell space diagonal. Thus a displacement with $\delta_s = 1$ would move an inner cell with index (i, j, k) to the cell with index $(i+1, j+1, k+1)$.

We have generated three coarse grids with $\delta_s = 0.2/0.4/0.6$ starting from a $2 \times 2 \times 2$ upscaling of $\mathcal{G}_1$. Then we have applied transmissibility upscaling using both open and closed boundary conditions to generate the geological fields. In the case of open boundary conditions, what we notice in Figure 5.20 is that grids that have been internal displaced seem to give a better solution compared to the solution with $\delta_s = 0$. Indeed this behaviour is the fault of using the bounding box as defined in 4.1.3 to define the local problem domains. In fact, even if there is a very small displacement, the bounding box holds more fine grid cells. Thus, in the case of a $2 \times 2 \times 2$ coarse grid with $\delta_s = 0$, each local problem size would be of $2 \times 2 \times 2$ while if we apply a displacement $0 < \delta_s < 1$, the local problem size would be of $3 \times 3 \times 3$ inducing a sort of ring and then reducing errors due to the upscaling of transmissibility.

This is also noticeable in Figure 5.22(a) where the Field Water Injection Rate is shown. Again the solutions given from grids with displacement seem to approximate better the fine grid behaviour. This is true for each injection

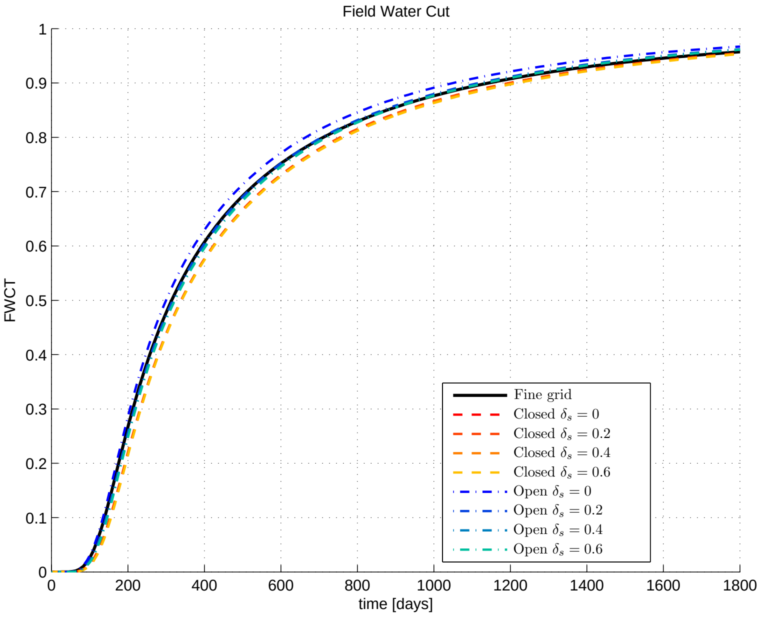


Figure 5.20: Structured displacement analysis: Field Water Cut for $\mathcal{G}_1$.

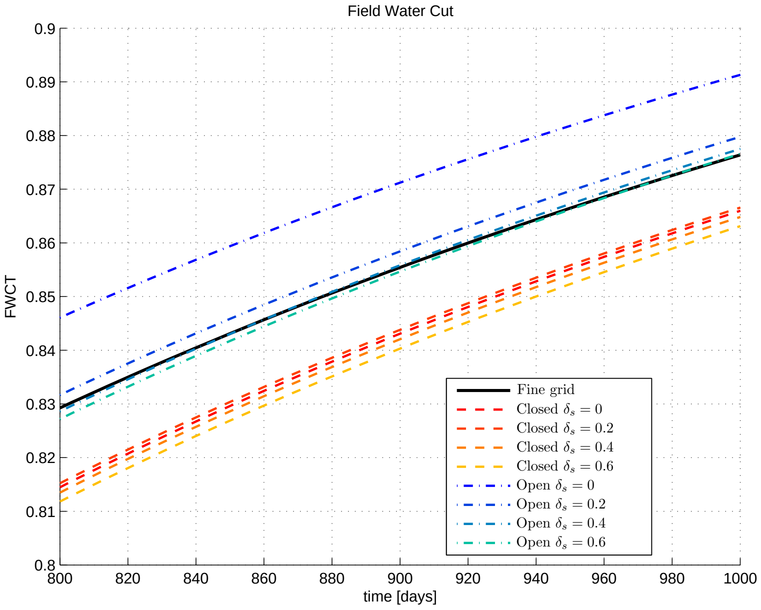
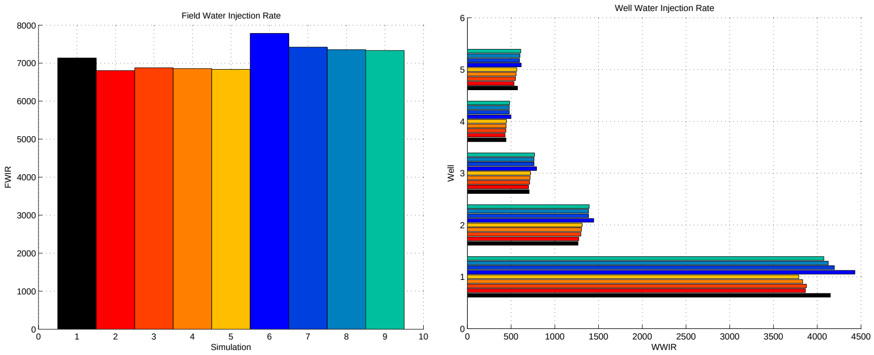


Figure 5.21: Structured displacement analysis: Field Water Cut zoom for $\mathcal{G}_1$.



(a) FWIR (b) WWIR

Figure 5.22: Structured displacement analysis: Water Injection Rate for $\mathcal{G}_1$.

well (Figure 5.22(b)).

However, the use of closed boundary conditions does not improve the solution as much as the open conditions, instead it moves away from the fine grid solution. We will try to explain this behaviour in the next section.

In short, it does not seem that structured displacements over non-matching grids can introduce systematic errors to the solutions. This validates, at least for this kind of non-matching grids, the upscaling algorithm and the extrapolation of the average pressure and average flow from the local problems. However in this case we are not taking into account the possible presence of non parallel faces between coarse and fine grids. For this reason we study another unstructured kind of grid displacements in the next chapter.

5.2.4 Unstructured Displacement Analysis

Finally, we study the behaviour of transmissibility upscaling for unstructured displacements of the coarse grid. This is achieved by a suitable movement of the pillars; first of all, we define a new parameter δ_{us}, then we divide the pillars into two groups denoted as *black* and *white*. Looking from a top view we can assign each pillar to the two groups likewise a chess board, e.g. if $i + j$ is even the pillar belongs to the *black* group otherwise if $i + j$ is odd it belongs to the *white* group. Finally we move pillars belonging to "black" groups of δ_{us} along y-axis while pillars belonging to "white" group are moved of $-\delta_{us}$ along y-axis. The resulting configuration is shown in Figure 5.23.

We start considering the first grid $\mathcal{G}_1$. In Figures 5.24(a) and 5.24(b) are shown the Field Water Cut of the solutions using different values for δ_{us} and closed boundary conditions. As we can see there is a noticeable decay

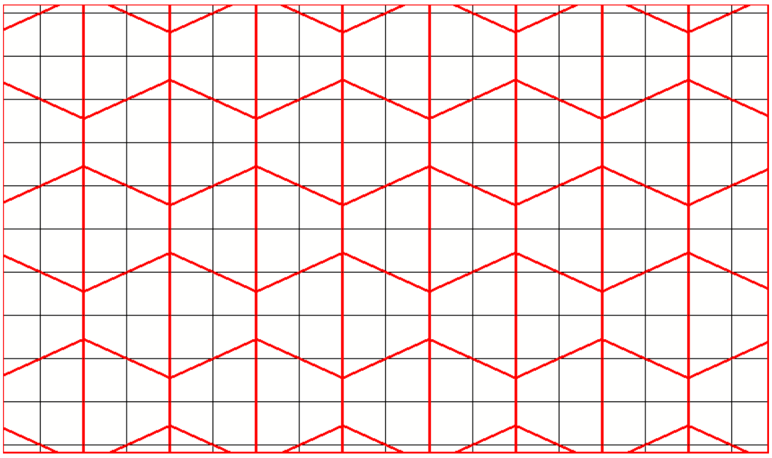
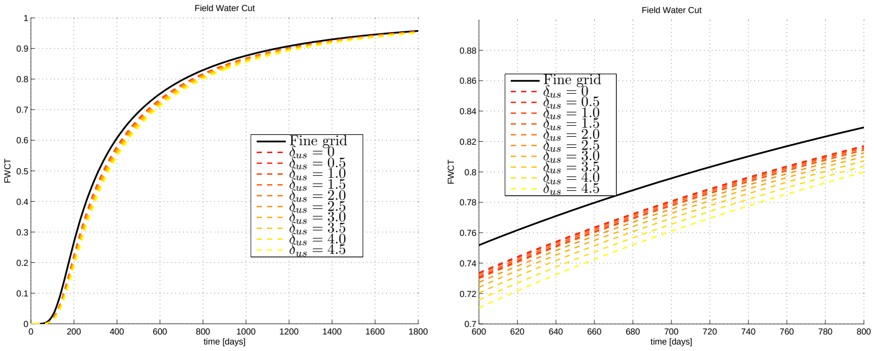


Figure 5.23: Example of unstructured displacement (red grid) seen from the top.

of the water cut growing with increasing displacement. The same behaviour is observed for the Water Injection Rate in Figures 5.25(a) and 5.25(b).

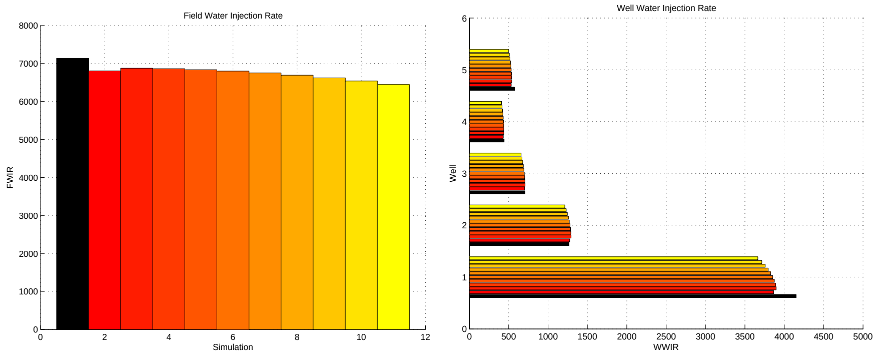


(a) FWCT

(b) FWCT zoom

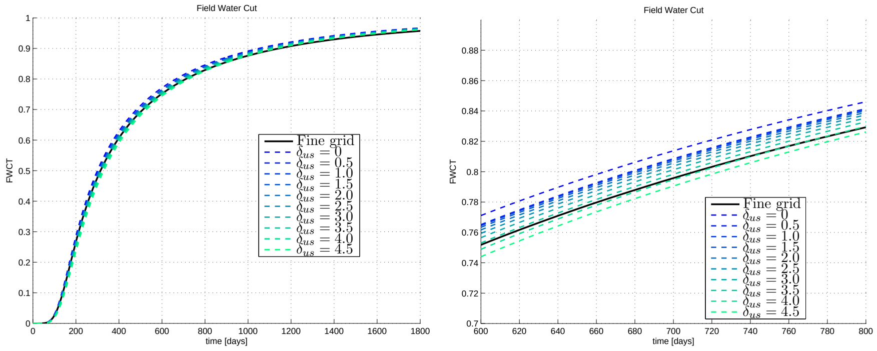
Figure 5.24: Unstructured displacement analysis: Field Water Cut for $\mathcal{G}_1$ using Closed BC.

Then, we analyse the solutions obtained using the transmissibility fields upscaled on unstructurally displaced grids and open boundary conditions. The solutions seem to decay again in both parameters FWCT (Figures 5.26(a) and 5.26(b)) and FWIR/WWIR (5.25(a) and 5.25(b)). However this time, since upscaling using open boundary conditions for transmissibility fields generally results with an higher connectivity, the solutions for high values of δ_{us} behave better. This can be in part explained recalling the fact that we are using a Finite Volume method with two-points flux over non-Cartesian grids. In fact, this method is only correct if the grid directions are aligned



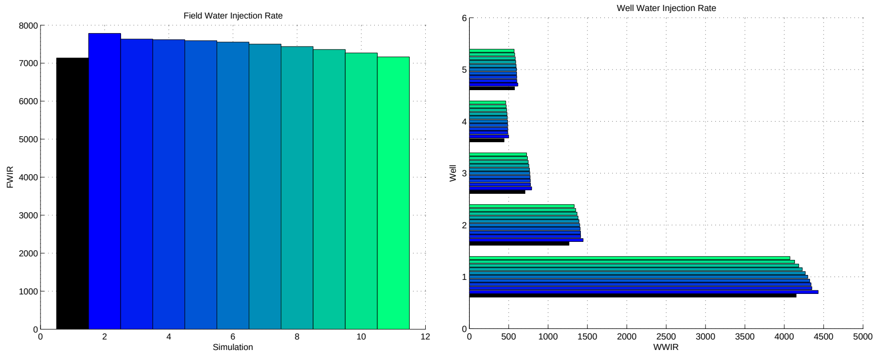
(a) FWIR (b) WWIR

Figure 5.25: Unstructured displacement analysis: Water Injection Rate for $\mathcal{G}_1$ using Closed BC.



(a) FWCT (b) FWCT zoom

Figure 5.26: Unstructured displacement analysis: Field Water Cut for $\mathcal{G}_1$ using Open BC.



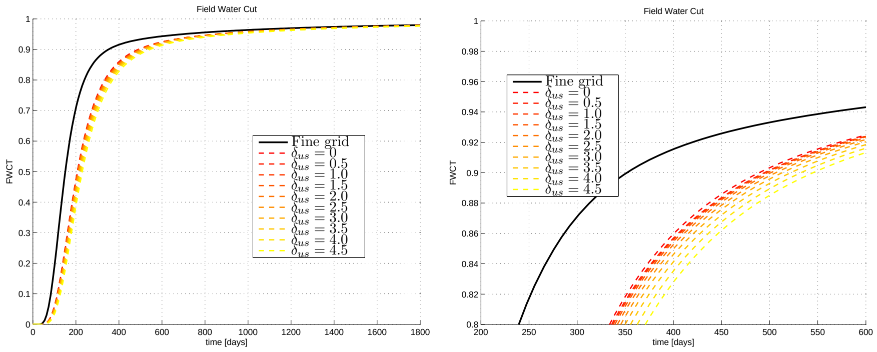
(a) FWIR (b) WWIR

Figure 5.27: Unstructured displacement analysis: Water Injection Rate $\mathcal{G}_1$ using Open BC.

with the principal directions of the permeability tensor. For general non orthogonal grids, the two-points flux scheme does not work properly. Some analysis of the two-points flux over distorted grids can be found in [1]. It would be interesting to use a multipoint approximation to avoid errors due to the adopted discretization scheme. However, this extension is beyond the purpose of the work. For a complete analysis of the finite volume multipoint flux we refer the reader to [8].

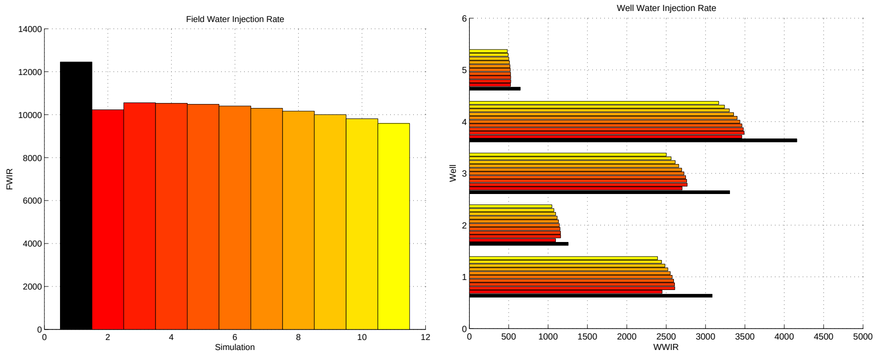
A similar behaviour is obtained using $\mathcal{G}_2$ and varying δ_{us}. With both open and closed boundary conditions there is a decreasing trend of the injected water resulting in an underestimation of the reservoir connectivity (see Figures 5.28 and 5.30).

Therefore we can conclude that upscaling over coarse grids with unstructured displacements does not produce systematic errors due to the upscaling procedure itself. The major problem in these cases is the solver used that is not optimal for this kind of grids. However for small values of the displacement the error is limited and we stress the fact that the relevant real cases are often *almost* Cartesian.



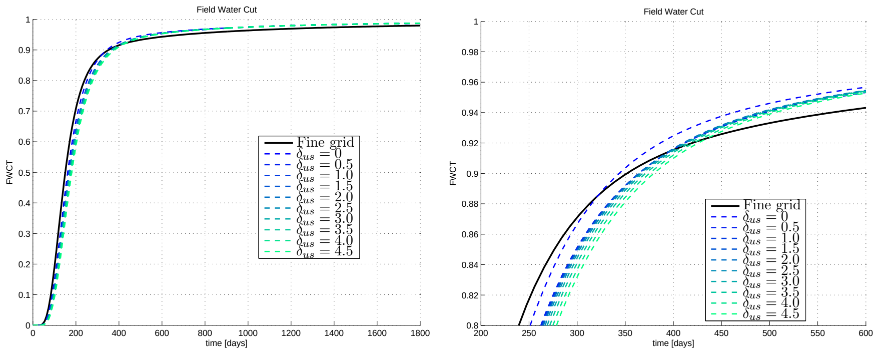
(a) FWCT (b) FWCT zoom

Figure 5.28: Unstructured displacement analysis: Field Water Cut for $\mathcal{G}_2$ using Closed BC.



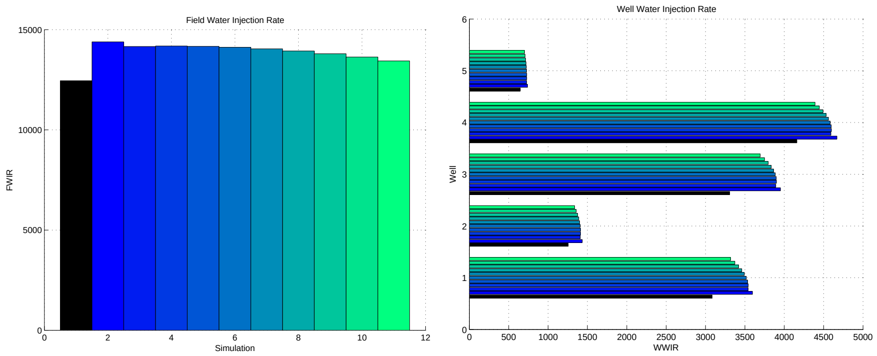
(a) FWIR (b) WWIR

Figure 5.29: Unstructured displacement analysis: Water Injection Rate for $\mathcal{G}_2$ using Closed BC.



(a) FWCT (b) FWCT zoom

Figure 5.30: Unstructured displacement analysis: Field Water Cut for $\mathcal{G}_2$ using Open BC.



(a) FWIR (b) WWIR

Figure 5.31: Unstructured displacement analysis: Water Injection Rate $\mathcal{G}_2$ using Open BC.

Conclusions

The aim of this thesis was the development of a workflow and the tools for 3D upscaling on non-matching coarse grids. We have implemented an original code in `C++` that can be interfaced with commercial reservoir simulators such as *Eclipse* using the industrial standard for grid format. Given a fine grid in which, for each cell, the permeability tensor is defined, we are able to compute upscaled properties such as permeability and transmissibility over any coarse grid with the help of different tools to improve upscaling accuracy such as variable ring size and different choices of the boundary conditions.

The integration on the intersection between non matching cells have been developed in order to manage coarse grids in which cells faces are not aligned with fine grid. We have performed several tests to assess the correct behaviour and robustness of the upscaling techniques providing comparisons with already existing software for upscaling with matching grids. The special treatments of wells placement allow us to focus on errors exclusively produced by the upscaling procedures without introducing further approximations.

We believe that a satisfactory result for various type of coarse grids has been reached, focusing on the case of non-matching coarse grids. The results show that our upscaling technique is robust with respect to the grids, i.e. it is not particularly influenced by the overlap between fine and coarse cells while the results are strongly dependent, as expected, by the properties of the global field, such as high channelised reservoir, and by the coarsening level.

We now point out some possible future development to improve the software capabilities:

- the upscaling technique could be easily parallelized using MPI or hybrid MPI/OpenMP as discussed in Chapter 4.2.3,
- the study the coarse field behaviours using a finite volume multipoint flux scheme to solve the fluid flow equations could be beneficial in the case of extremely distorted grids,

- the implementation of a local-global upscaling technique as shown in Chapter 2.4 would lead to the more precise solution,
- the implementation of a near-well upscaling scheme would eliminate errors due to the wells placement in fine/coarse grids as discussed in Chapter 2.5.

Bibliography

[1] I. Aavatsmark. Interpretation of a two-point flux stencil for skew parallelogram grids. *Computational geosciences*, 11(3):199–206, 2007.

[2] T. Arbogast, S.L. Bryant, et al. A two-scale numerical subgrid technique for waterflood simulations. *SPE Journal*, 7(04):446–457, 2002.

[3] C. Bradford Barber, D. P. Dobkin, and H. Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software (TOMS)*, 22(4):469–483, 1996.

[4] J. Bear and Y. Bachmat. *Introduction to Modeling of Transport Phenomena in Porous Media*, volume 4. Kluwer Academic Publishers, 1990.

[5] F. Brezzi and M. Fortin. *Mixed and Hybrid Finite Element Methods.* Springer, 1991.

[6] S.E. Buckley and M.C. Leverett. Mechanism of fluid displacement in sands. *Transaction of the AIME, SPE*, 146:107–116, 1941.

[7] E. Burman and P. Zunino. Numerical approximation of large contrast problems with the unfitted Nitsche method. In *Frontiers in Numerical Analysis-Durham 2010*, pages 227–282. Springer, 2012.

[8] Y. Cao, R. Helmig, and B. Wohlmuth. Geometrical interpretation of the multi-point flux approximation L-method. *International journal for numerical methods in fluids*, 60(11):1173–1199, 2009.

[9] T. Chen. *New methods for accurate upscaling with full-tensor effects.* ProQuest, 2009.

[10] Y. Chen, L.J. Durlofsky, M. Gerritsen, and X.H. Wen. A coupled local–global upscaling approach for simulating flow in highly heterogeneous formations. *Advances in Water Resources*, 26(10):1041–1060, 2003.

[11] F.F. Craig. *The reservoir engineering aspects of waterflooding.* Richardson, TX: Henry L. Doherty Memorial Fund of AIME, Society of Petroleum Engineers, 1993.

[12] H. Darcy. Les fontaines publiques de la ville de dijon. *Dalmont, Paris*, 1856.

[13] C. Deutsch. Calculating effective absolute permeability in sandstone/shale sequences. *SPE Formation Evaluation*, 4(3):343–348, 1989.

[14] Y. Ding. Upscaling on distorted gridblocks for simulation of advanced wells. *Journal of Petroleum Science and Engineering*, 43(1):87–97, 2004.

[15] L.J. Durlofsky. Coarse scale models of two phase flow in heterogeneous reservoirs: Volume averaged equations and their relationship to existing upscaling techniques. *Computational Geosciences*, 2(2):73–92, 1998.

[16] L.J. Durlofsky. Upscaling of geocellular models for reservoir flow simulation: a review of recent progress. In *7th International Forum on Reservoir Simulation Bühl/Baden-Baden, Germany*, pages 23–27. Citeseer, 2003.

[17] M.G. Edwards. Superconvergent renormalization and tensor approximation. In *5th European Conference on the Mathematics of Oil Recovery*, 1996.

[18] C.F. Eek-Jensen, I. Aavatsmark, and Ø. Bøe. Upscaling on general quadrilateral grids in 3D with application to field cases. In *10th European Symposium on Improved Oil Recovery*, 1999.

[19] A. Ern and J.L. Guermond. *Theory and Practice of Finite Elements*, volume 159. Springer, 2004.

[20] L. Formaggia. Data structures for unstructured mesh generation. In J.F. Thompson, B.K. Soni, and N.P.Weatherill, editors, *Handbook of Grid Generation*, chapter 14, pages 14/1–14/22. CRC Press, Boca Raton, Florida, 1999.

[21] M. Fortin. An analysis of the convergence of mixed finite element methods. *ESAIM: Mathematical Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique*, 11(4):341–354, 1977.

[22] R. Helmig. *Multiphase flow and transport processes in the subsurface: a contribution to the modeling of hydrosystems.* Springer, 1997.

[23] U. Hornung. *Homogenization and porous media*, volume 6. Springer, 1997.

[24] T.Y. Hou and X.H. Wu. A multiscale finite element method for elliptic problems in composite materials and porous media. *Journal of computational physics*, 134(1):169–189, 1997.

[25] E. Kasap and L.W. Lake. Calculating the effective permeability tensor of a gridblock. *SPE formation evaluation*, 5(2):192–200, 1989.

[26] M.J. King, D.G. MacDonald, S.P. Todd, and H. Leung. Application of novel upscaling approaches to the Magnus and Andrew reservoirs. In *EUROPEC'98: European petroleum conference*, 1998.

[27] G.E. Pickup, P.S. Ringrose, J.L. Jensen, and K.S. Sorbie. Permeability tensors for sedimentary structures. *Mathematical Geology*, 26(2):227–250, 1994.

[28] A. Quarteroni. *Modellistica Numerica per Problemi Differenziali*, volume 4. Springer, 5 edition, 2012.

[29] M. Rame and J.E. Killough. A new approach to the simulation of flows in highly heterogeneous porous media. *SPE Formation Evaluation*, 7:247–254, 1992.

[30] P.A. Raviart and J.M. Thomas. *A mixed finite element method for second-order elliptic problems*, volume 606. Springer, 1977.

[31] S.C. Rose, J.F. Buckwalter, and R.J. Woodhall. *The design engineering aspects of waterflooding.* Richardson, TX: Society of Petroleum Engineers, 1989.

[32] L. Turconi. Transmissibility upscaling for fluid flow in porous media on non matching grids. Master's thesis, Politecnico di Milano, 2012.

[33] C.D. White, R.N. Horne, et al. Computing absolute transmissibility in the presence of fine-scale heterogeneity. In *SPE Symposium on Reservoir Simulation.* Society of Petroleum Engineers, 1987.

[34] G.P. Willhite. *Waterflooding.* Richardson, TX: Society of Petroleum Engineers, 1986.