Pontifícia Universidade Católica
do Rio de Janeiro

André Paoliello Modenesi

Gridding and scaling strategies for unstructured reservoir flow simulation

Dissertação de Mestrado

Dissertation presented to the Programa de Pós-Graduação em Engenharia Mecânica of PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Engenharia Mecânica.

Advisor: Prof. Ivan Fábio Mota de Menezes
Co-advisor: Dr. Leonardo Seperuelo Duarte

Rio de Janeiro
December 2019

André Paoliello Modenesi

Gridding and scaling strategies for unstructured reservoir flow simulation

Dissertation presented to the Programa de Pós-Graduação em Engenharia Mecânica of PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Engenharia Mecânica. Approved by the Examination Committee.

Prof. Ivan Fábio Mota de Menezes
Advisor
Departamento de Engenharia Mecânica – PUC-Rio

Leonardo Seperuelo Duarte
Co-advisor
Instituto TecGraf – PUC-Rio

Prof. Waldemar Celes
Departamento de Informática – PUC-Rio

Ricardo Alexandre Passos Chaves
Petróleo Brasileiro S.A.

Rio de Janeiro, December 16th, 2019

André Paoliello Modenesi

The author graduated from Federal University of Minas Gerais (UFMG) in 2008 with a major in Electrical Engineering, with specialization in Telecommunications. He entered Petrobras in 2009, as a Petroleum Engineer. He concluded a latu sensu graduate degree in Petroleum Engineering from Petrobras University in 2010.

Bibliographic data

Modenesi, André Paoliello

Gridding and scaling strategies for unstructured reservoir flow simulation / André Paoliello Modenesi ; advisor: Ivan Fábio Mota de Menezes ; co-advisor: Leonardo Seperuelo Duarte. – 2019.
124 f. : il. color. ; 30 cm

Dissertação (mestrado)–Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Mecânica, 2019.
Inclui bibliografia

1. Engenharia Mecânica – Teses. 2. Simulação de reservatórios. 3. Malhas não-estruturadas. 4. Malhas de Voronoi. 5. Transferência de escala. I. Menezes, Ivan Fábio Mota de. II. Duarte, Leonardo Seperuelo. III. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Mecânica. IV. Título.

CDD: 621

To my wife, Letícia,
and to my parents Paulo e Giovana.

Acknowledgements

I would first like to thank my advisor Ivan Menezes, for the opportunity of working in this project. His professionalism and enthusiasm were unfaltering for the whole period.

I am also grateful to my co-advisor Leonardo Duarte, who developed the reservoir simulator used for this research. His support with several issues was indispensable for the developed code to work.

I am grateful to Pontifical Catholic University of Rio de Janeiro – PUC-Rio, for giving me the opportunity of enrolling in this master's program. In addition, this study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001, to whom I also thank.

My sincere thanks to the company for which I work, Petrobras, for conceding time for me to dedicate to this master's program. The advice and guidance offered by several of my colleagues, in particular Daniel Miranda and Marcos Barbosa, was essential to help to shape this dissertation.

Finally, I'd like to acknowledge my wife and parents for their support and inspiration while working on this dissertation.

Abstract

Modenesi, André Paoliello; Menezes, Ivan Fábio Mota de (Advisor); Duarte, Leonardo Seperuelo (Co-advisor). **Gridding and scaling strategies for unstructured reservoir flow simulation.** Rio de Janeiro, 2019. 124p. Dissertação de Mestrado – Departamento de Engenharia Mecânica, Pontifícia Universidade Católica do Rio de Janeiro.

Numerical simulation represents an essential tool for modern reservoir engineering, especially for the development of offshore oil fields. Most reservoir simulations are performed on three-dimensional structured grids, with a size ranging from a few thousands to tens of millions of cells. Some simulations can have a high computational cost that hinders the field development studies, even using the processing power available nowadays. Unstructured meshes are an effective alternative to reduce the size of reservoir models (and, consequently, the overall simulation time) without sacrificing the quality of the results. In this work, we adopt Voronoi meshes, also known as perpendicular bisector grids, since their properties simplify the discretized flow equations in reservoir simulations when compared to other types of unstructured meshes. Two main steps are critical to creating an unstructured reservoir model from a refined geological model: grid generation and upscaling of the reservoir properties. Most methods employed for both steps rely on information obtained from simulations using fine-scale meshes. Although this approach yields good results, it can be time-consuming and may be optimal only for the specified set of flow conditions. This work discusses the generation of unstructured grids and upscaling techniques that do not require any previous simulations. Instead, they are based only on reservoir property distributions and the location of discrete features such as wells and faults. The proposed grid generation strategy starts from a regular set of points and then redistributes them according to a previously defined spacing map. Two iterative redistribution algorithms based on physical models are presented, and several criteria for spacing maps are also investigated. Two upscaling algorithms for unstructured grids are proposed, based on the Cardwell & Parsons and renormalization techniques for structured meshes. Finally, representative examples are presented to demonstrate the capabilities and effectiveness of the proposed strategies.

Keywords

Reservoir simulation; unstructured grids; Voronoi grids; upscaling.

Resumo

Modenesi, André Paoliello; Menezes, Ivan Fábio Mota de (Orientador); Duarte, Leonardo Seperuelo (Coorientador). **Estratégias de geração de malhas não-estruturadas e transferência de escala para simulação de escoamento em reservatórios.** Rio de Janeiro, 2019. 124p. Dissertação de Mestrado – Departamento de Engenharia Mecânica, Pontifícia Universidade Católica do Rio de Janeiro.

A simulação numérica é uma ferramenta essencial para a engenharia de reservatórios moderna, em particular no desenvolvimento de campos de óleo marítimos. A maioria das simulações de reservatórios utilizam malhas estruturadas em três dimensões, com tamanho variando de alguns milhares a dezenas de milhões de células. Algumas simulações apresentam um alto custo computacional que pode dificultar os estudos de desenvolvimento de um campo, mesmo com a alta capacidade computacional disponível hoje. Malhas de simulação não-estruturadas são uma alternativa para reduzir o tamanho dos modelos de reservatórios (e, consequentemente, o tempo de execução das simulações), sem sacrificar a qualidade dos resultados. Este trabalho utiliza malhas de Voronoi, também conhecidas como malhas de bissetores perpendiculares, uma vez que suas propriedades permitem simplificar as equações discretizadas do escoamento em comparação com outros tipos de malhas não-estruturadas. Dois passos são críticos para a criação de um modelo não-estruturado de reservatórios a partir de um modelo geológico refinado: geração da malha e transferência de escala das propriedades. A maioria dos métodos propostos para ambas as tarefas utilizam informações de simulações na malha refinada. Embora essa abordagem apresente bons resultados, pode ser muito custosa e precisa ser refeita caso haja alterações significativas nas condições de escoamento. Este trabalho discute técnicas para geração de malha e transferência de escala que não dependam de simulações na escala fina. As técnicas utilizam apenas a distribuição de propriedades de reservatórios e o posicionamento de poços, falhas e outras feições discretas. A abordagem adotada para geração da malha parte de uma disposição regular de pontos que são redistribuídos de acordo com um mapa de espaçamento previamente definido. Dois algoritmos iterativos para redistribuição desses pontos baseados em modelos físicos são propostos. Diversos critérios de espaçamento também são investigados. Dois algoritmos de

transferência de escala em malhas não-estruturadas são propostos. Estes métodos se baseiam nas técnicas de Cardwell & Parsons e de renormalização para transferência de escala em malhas estruturadas. Por fim, exemplos representativos são utilizados para demonstrar as potencialidades e eficácia das estratégias propostas.

Palavras-chave

Simulação de reservatórios; malhas não-estruturadas; malhas de Voronoi; transferência de escala.

Contents

List of Figures

List of Tables

Acronyms

AIM – Adaptive Implicit Method

CVFD - Control volume finite differences discretization method

CVFE - Control volume finite elements dicretization method

EMPower - Exxon Mobil proprietary reservoir simulator

FD - Finite differences discretization method

FE - Finite elements discretization method

FVF - Fluid formation volume factor

GERESIM - Reservoir Pre and post processing software suite

GOCAD - Geological modeling software

GSim - Academic reservoir simulator under development

IMEX - Black oil reservoir simulator

IMPES – Implicit pressure explicit saturation method

MATLAB - Numerical computing software

PEBI - Perpendicular bisector grid (Voronoi grid)

PETREL - Geological modeling software

1
Introduction

This introduction presents the motivation behind this dissertation and the objectives it intends to achieve. The organization of the reminder of the dissertation and the software involved in the implementation are discussed.

1.1.
Motivation

Numerical simulation is an essential tool for modern reservoir engineering. It is particularly useful while developing offshore oil fields, where regular well patterns (such as five spot) are not a viable development alternative, due to high costs for well drilling and infrastructure restrictions. Higher spacing between wells means heterogeneities in the reservoir and the presence of faults and fractures are much more relevant to overall production. In this context, the position of each well must be planned carefully in order to guarantee an expected recovery, and take into account subsurface uncertainties.

The first commercial software for numerical simulation appeared in the nineteen seventies, and evolved greatly in the following decades with the increase in available computational power. While the first reservoir simulations were limited to two-dimensional (areal) models, almost all grids are now three-dimensional. The need to better represent the behavior of real reservoirs has led to a constant increase in model sizes, with many currently surpassing one million grid cells, and a few even surpassing ten million.

Simulation models have also become more sophisticated in other ways. Fluid behavior for different types of reservoirs can be better represented with compositional or thermal models than with black oil models. Dual media grids are capable of representing reservoirs where both rock matrix and fractures contribute to production. Other types of physical phenomena, such as rock geomechanics or topside flow behavior, are occasionally coupled with reservoir flow simulation.

The combination of those resources requires high computational power. Even with the evolution of processing power of the last decades, some individual simulations can still take over 24 hours to end, or even worse, abort before conclusion due to numerical difficulties. This is particularly troublesome when the reservoir engineer is evaluating multiple scenarios, for instance to determine the best development strategy or to evaluate the impact of geological uncertainties.

Unstructured simulation grids are an alternative to reduce the size of reservoir models without sacrificing the quality of the results. The basic concept is that regions that have high impact on flow results should be better refined, while regions that have little impact on the results can be represented by coarse cells, reducing the overall number of cells.

Performing reservoir flow simulation on unstructured grids is not a new concept. The first academic propositions can be traced back to the early nineties (Heinemann et al., 1989; Palagi et al., 1994). However, the commercial availability of such methods was restricted for a long time, and focused mostly on well testing. Some oil companies developed their own internal unstructured simulators tailored to their needs (Khan & Dawson, 2004; Usadi et al., 2007). Only within the last decade, with the development of the latest generation of simulators, have unstructured grids became more widely available for reservoir flow simulation.

Even so, the fast generation of appropriate unstructured models is still a complex task. Most commercial geological modeling still focus on the generation of structured grids (Manckuk & Deutsch, 2009). Coarse unstructured models can be derived from a refined structured model, but permeability upscaling can become a problem. Most academic and practical solutions rely on flow results from the refined model to determine the unstructured grid geometry and the value of upscaled properties.

1.2. Objectives

The objective of this dissertation is to propose a framework to generate and simulate unstructured reservoir grids, employing the academic flow simulator that is being developed by PUC-Tecgraf in partnership with Petrobras (GSim).

Although it is not yet possible to use this environment for real full field projects, it is ideal for quickly developing and validating new concepts.

Two main steps are critical to creating an unstructured reservoir model from a refined geological model: grid generation and upscaling of the reservoir properties. The most common approaches in literature for unstructured grid generation and upscaling are flow-based – that is, fine scale simulations are employed to place cells and/or to compute properties. Although this approach yields good results, it can be time-consuming and may be optimal only for a specific set of flow conditions. Therefore, the main focus of this work is to propose grid generation and upscaling techniques that rely only on property values and geological features, without the need for fine scale simulations.

1.3. Dissertation organization

This dissertation is divided into six chapters. The first one is this introduction. Chapter 2 briefly discusses mathematical concepts behind reservoir simulation and grid discretization alternatives (both structured and unstructured). The geometry of the grid can impact the validity of the discretization scheme, resulting in approximation errors or the need for more complex equations.

Chapter 3 discusses grid generation for unstructured models. Grid generation involves defining the type of cells that will be used, which determines what sort of discretization can be used to solve the flow problem, and spatially distributing those cells, which allows the better representation of features of interest in detriment of others. Ultimately, this should lead to a better tradeoff between fidelity and computational cost than with structured reservoirs.

Chapter 4 discusses permeability upscaling methods. The upscaling of properties for structured grids is a frequently discussed problem in reservoir simulation (see King, 1989; Rubin & Gómez-Hernandez, 1990; Romeu, 1994; Renard & De Marsily, 1997 and many others). It arises from the need of the engineer to work with a model with fewer cells than were initially modeled by the geoscientists, mostly due to computational restrictions. Upscaling may be used to reduce the number of layers, increase the areal size of cells, or both. For some properties, such as porosity, upscaling is very straightforward, involving simply a

volume weighted average. The difficulty arises when upscaling properties that have a directional effect, such as permeability, and that should, in the most general form, be described as a second order tensor. With unstructured grids, the problem becomes more complex, because the connections between cells can happen at arbitrary angles, distinct from the permeability tensor principal directions.

Chapter 5 presents unstructured simulation models built for several scenarios, using the techniques discussed in the previous chapters. The simulation results are compared with those from structured models, to evaluate their fidelity. The grid generation and upscaling methods can also be compared among themselves.

The last chapter presents our main conclusions regarding the developed techniques and their potential. Many issues that require further investigation are also highlighted.

1.4. Proposed workflow and software involved

GSim is a multiphase reservoir simulator being developed at Pontifical Catholic University of Rio de Janeiro (PUC-Rio) in a joint project with Petrobras. Although it was originally conceived for simulation with structured grids, the underlying framework has support for different types of elements. GSim has a modular structure that simplifies changing specific features. This property was previously employed for accessing the efficiency of different types of solvers (Piazza, 2019).

A refined structured reservoir model is created with a geological modeling software. Unstructured grid generation and upscaling are performed in MATLAB, since it has a large mathematical library and visualization alternatives available, using the refined model as input data. The unstructured model is then transferred to GSim for flow simulation. Time series results from GSim can be viewed and compared to other simulators with the GERESIM visualization suite. However, since GERESIM does not support unstructured grid visualization yet, MATLAB was employed to view and compare grid results. An overview of the adopted workflow is represented in Figure 1.

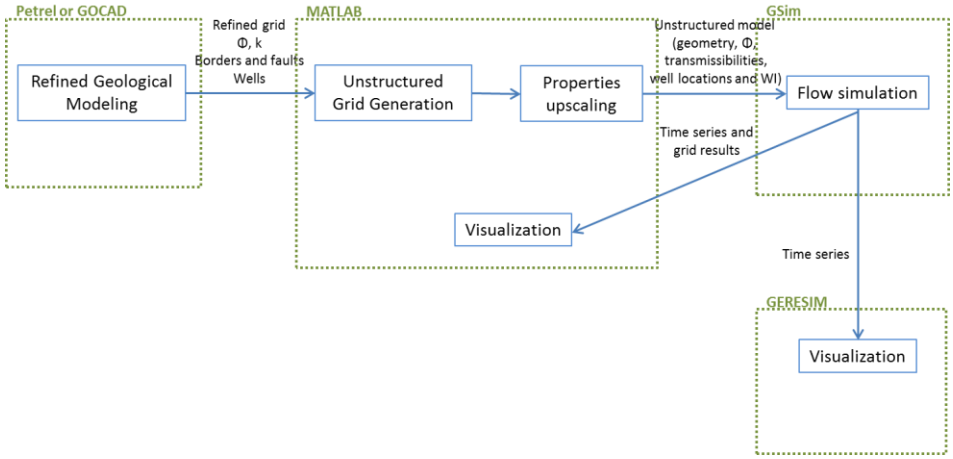

Figure 1 – General workflow adopted in this work (ϕ refers to porosity distribution and k to permeability distribution)

Although unstructured grids can be generated in three dimensions, this adds considerable complexity to some of the algorithms. Since many reservoirs have characteristics strongly influenced by deposition layers, a reasonable approach is a 2.5D unstructured model, where a horizontal two dimensional grid is extended in the vertical direction, and subdivided in horizontal layers. Thus, although the model is unstructured in the areal direction, the layers are structured, with constant or proportional height. As long as the reservoir does not have a significant dip or inclined faults, this is an acceptable representation.

2
Reservoir flow simulation

This chapter discusses reservoir flow simulation, and how it can be performed on unstructured grids. The first section presents a brief summary of the mathematical principles involved. The following sections discuss, respectively, structured and unstructured grids, and their impact on the mathematical formulation.

2.1.
Numerical reservoir simulation

2.1.1.
Porous media flow equations

Numerical flow simulation attempts to numerically represent the behavior of fluids in a porous media, in order to evaluate the production of real reservoirs. The physical phenomena involved can be mathematically described by a fluid transport law, combined with mass conservation and fluid state equation. Additional equations may be necessary to describe rock and fluid properties.

Darcy's law (eq. (1**Erro! Fonte de referência não encontrada.**)) describes f luid transport on a porous media for a slow (apparent) velocity $\vec{v}$. Rock permeability is represented in the most generic form as a symmetric second order tensor $\bar{\bar{k}}$. A common assumption is that the coordinate system is aligned to the principal axis, so that the tensor becomes diagonal. The flow potential function Φ is given by eq. (2), where p, μ and γ are, respectively, fluid pressure, viscosity and specific weight. In some situations, such as gas flow near wells or in the presence of fractures, flow rates within the rock can be high enough for inertial terms to become relevant. In those cases, it becomes necessary to adopt a variation of Darcy's law, known as Darcy-Forchheimer law.

$$\vec{v} = -\frac{\bar{\bar{k}}}{\mu}\nabla\Phi \tag{1}$$

$$\nabla\Phi = \nabla \mathrm{p} - \gamma\nabla \mathrm{z} \tag{2}$$

Mass conservation states that the net fluid mass entering a control volume (flow term) must equal the mass variation within the volume (accumulation term). It can be expressed by eq. (3), where ρ is the fluid specific density and ϕ is rock porosity.

$$\nabla.(\rho\vec{v}) = -\frac{\partial(\phi\rho)}{\partial t} \tag{3}$$

A fluid equation of state (EOS) describes the relationship between fluid properties (such as density, compressibility and viscosity), pressure and temperature conditions. Another important relationship is the effect of rock compressibility on porosity.

Darcy's law and mass conservation can be combined into what is known as the diffusivity equation. For single-phase flow, isotropic homogeneous media with permeability k and small compressibilities and pressure gradients, it is possible to simplify the diffusivity equation into a form (eq. (4)) that has analytical solutions for some initial and boundary conditions. Total compressibility of this system (rock and fluid) is represented by c_t. The principle of superposition allows this equation to also be used for situations with more than one well or with variable production rates.

$$\nabla^2 \mathrm{p} = \frac{\phi\rho c_t}{k}\frac{\partial p}{\partial t} \tag{4}$$

For multiphase flow, mass conservation must be stated for each component of the model. A black-oil model considers three components, while a compositional model may have many more, representing different ranges of hydrocarbon components. This results in a system of equations similar to eq. (5)**Erro! Fonte de referência não encontrada.**, one for each component c, where S_c is the saturation and B_c the formation volume factor. The gas equation has additional terms due to the presence of dissolved gas on the oil phase in reservoir conditions.

$$\nabla.\left(\frac{\rho_c}{\mu_c}k_{rc}\bar{\bar{k}}\nabla\Phi\right) = -\frac{\partial}{\partial t}\left(\frac{\phi S_c}{B_c}\right) \tag{5}$$

The effective permeability for each phase is smaller than the absolute permeability k for single-phase flow. This relationship is usually expressed by the

concept of a relative permeability k_{rc} for each component (eq. (6)). Relative permeabilities are a function of fluid saturations.

$$k_c = k_{rc} k \tag{6}$$

An additional relationship necessary for the multiphase solution is to assume that porous space is entirely filled by the modeled components (eq. (7)).

$$\sum_c S_c = 1 \tag{7}$$

2.1.2. Multiphase flow equations discretization

The resulting multiphase flow equation system must be discretized to be solved by a computer program. This means properties will be sampled at a finite set of points instead of the entire continuous domain, and the derivatives approximated. The objective is to determine pressure and fluid saturations at those points, and fluid production at the wells, at assigned dates.

Discretization methods can be divided into three categories: finite differences (FD), finite elements (FE) and finite volumes (FV). Finite differences employs a truncated Taylor series expansion to discretize the differential form of the flow equations. A stencil determines neighboring nodes necessary for the calculation at a specific node – thus the method is restricted to structured grids. FD can be easily implemented for Cartesian grids. Other structured grids require a transformation from physical to computational space, reflected in the Jacobian calculation (Blazek, 2015).

Finite elements divides the problem into triangular or tetrahedral subdomains, and employs shape functions to approximate the solution at each of those elements, while using an integral form of the flow equations. Its mathematical development is more sophisticated than FD and the associated computational cost is higher.

Finite volume methods begin by dividing the domain into polyhedral control volumes. Those control volumes can be identical to the grid cells, or be centered on vertices or faces of the grid. The divergence term of the flow equations are converted into surface integrals over the faces of each control volume. Several different forms of development are possible. Using Taylor series results in a control volume finite differences (CVFD) method, while other assumptions lead to a

control volume finite elements (CVFE) method. In general, FV methods are more flexible than finite differences, and less complex than finite elements.

Commercial software for reservoir simulation usually adopts a FD solution of the flow equations. The equations can also be written in a CVFD method, as exemplified by eq. (8) (based on Aziz, 1993), for a block i and component c at time step $n+1$. The left side of the equation represents the flow between connected cells, where T is the transmissibility between cells i and j for a component and ϕ is the potential at a cell. The right side of the equation corresponds the accumulation term, where V is cell volume, M the specific mass of component c, and q is the flow from external sources (wells). This equation reduces to standard FD for a Cartesian grid.

$$\sum_{j}^{j\ connected\ to\ i} T_{c,i,j}\left(\phi_j - \phi_i\right) = \frac{V_i}{\Delta t}\left(M_{c,i}^{n+1} - M_{c,i}^{n}\right) + q_{c,i} \qquad (8)$$

2.2. Structured grids

Grids for simulation (both structured and unstructured) can be defined by two general approaches (Settari et al., 1972). In the block-centered approach, the block limits are defined initially, and block points correspond to the barycenter of those blocks. In point distributed grids, the block points are defined first, and block limits derived by some rule, usually corresponding to the midway between two points. According to Aziz (1993), the block-centered approach is more accurate for calculating the accumulation terms, while point-distributed is superior for the flow terms. Commercial simulators usually adopt block-centered grids.

Structured 3D grids are those whose cells can be indexed on a lattice, so that the number of neighbors and their indexes can be determined by a simple rule (Figure 2). This property is crucial when using a finite differences discretization.

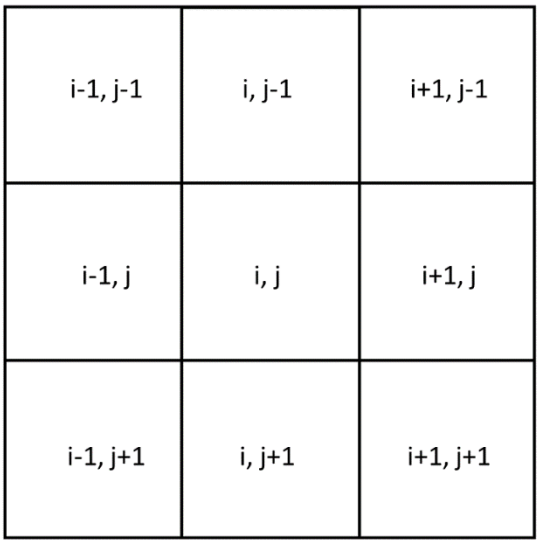

Figure 2 – Example of structured 2D grid indexing

The simplest possible 3D grid is a Cartesian grid, consisting of regularly spaced grid blocks (Figure 3.a). Each block is a rectangular cuboid with identical dimensions. With the Cartesian grid, the point-distributed and block-centered approaches are equivalent.

In order to better represent some features of the reservoir, a simple approach is to define a variable spacing, so that all blocks are still rectangular cuboids, but with different dimensions (Figure 3.b,c). An undesired effect of this sort of grid is that it may also refine regions that are not of interest, for certain features, resulting in a higher than ideal number of cells.

Rectangular grids can be refined without this problem by using local refinements (Figure 3.d). The cells that must be better represented are partitioned into smaller subcells. The abrupt contrast between cell sizes can become an issue for flow simulation with this sort of grid.

A cylindrical grid, with cells specified on the horizontal plane by radius and angle instead of x and y coordinates, can be used to describe the reservoir around horizontal wells (Figure 3.e). This is useful when interested in the detailed well behavior, for instance, while simulating the results of well tests.

More flexible grids can be described using a general corner point approach (Figure 3.f), with arbitrary hexahedral cells. This sort of approach allows the grid to better represent geological features. However, since flow is no longer orthogonal to cell limits, there is an additional error associated with the discretization. This error becomes higher when the model has highly deformed cells.

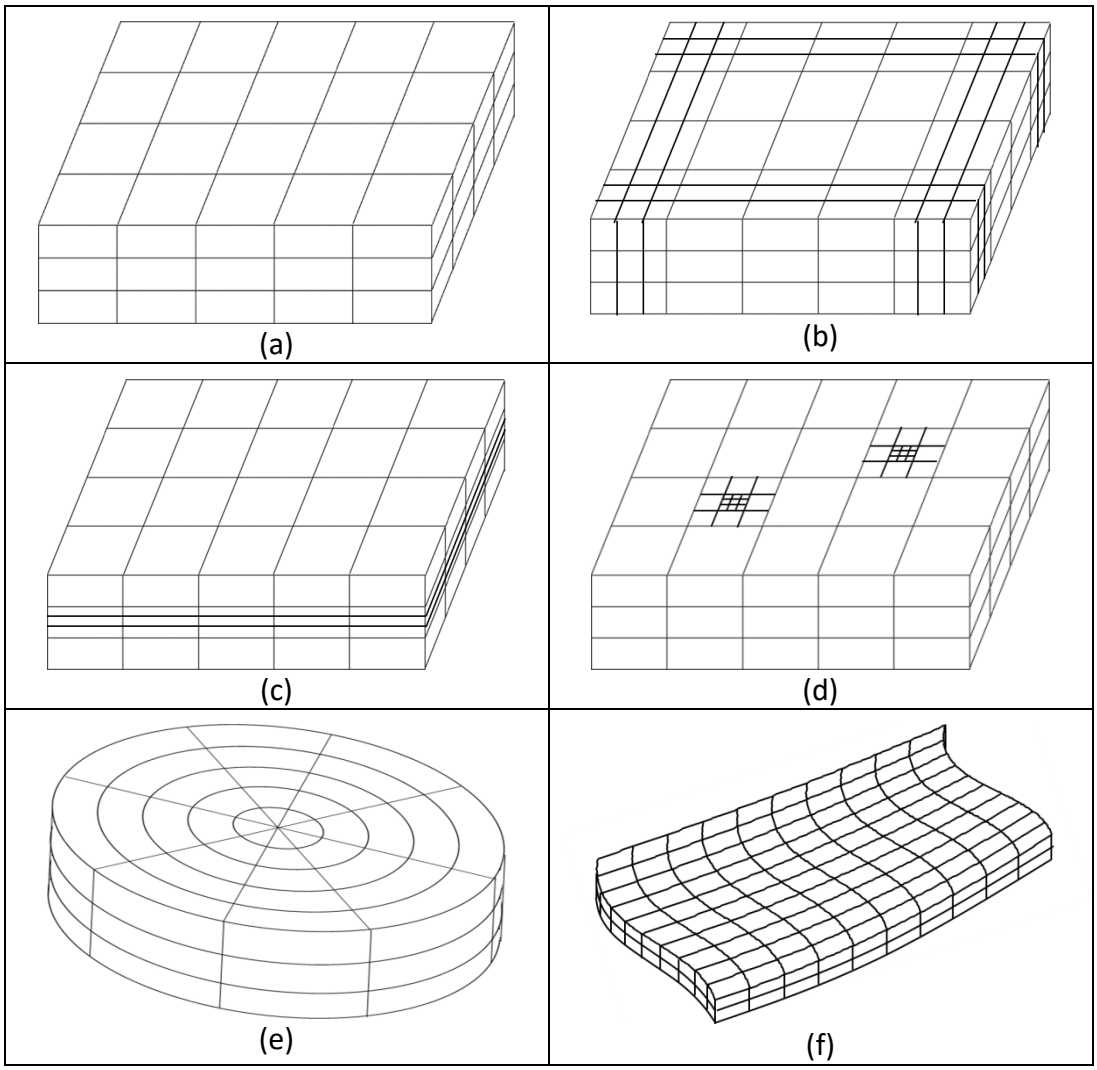

Figure 3 – Examples of 3D structured grids: (a) Cartesian, (b,c) rectilinear, (d) local refinement, (e) cylindrical, (f) corner point

2.3. Unstructured grids

One of the first types of grid used for unstructured reservoir flow simulation is called the perpendicular bisector (PEBI) grid. It corresponds to a Voronoi partition of the domain. All points of the domain that are closer to a specific site than to any other are attributed to a common cell. It is a useful sort of grid for flow simulation because the boundary between two cells is always orthogonal to the connection between their representative points. Thus, a simple two-point finite difference discretization can be applied to solve the flow problem (Aziz, 1993). Another advantage of the Voronoi grid is that cells are generally well shaped, while

triangulations can result in highly deformed cells called slivers (Hale, 2002). Voronoi grids will be better discussed in section 3.6.

Other types of unstructured grids use triangular (in 2D) or tetrahedral (in 3D) cells. In a mixed grid, other sorts of elements may be used to represent the boundary layers (Blazek, 2015). Those sorts of grids require a more sophisticated discretization of flow equations, such as a CVFE method, because flow is not necessarily orthogonal to cell limits. The number of triangles or tetrahedral for a specific node set is also larger than the Voronoi cells for the same set, resulting in higher memory and computational costs. Figure 4 presents a comparative example of a Voronoi grid and a triangular grid for the same node set.

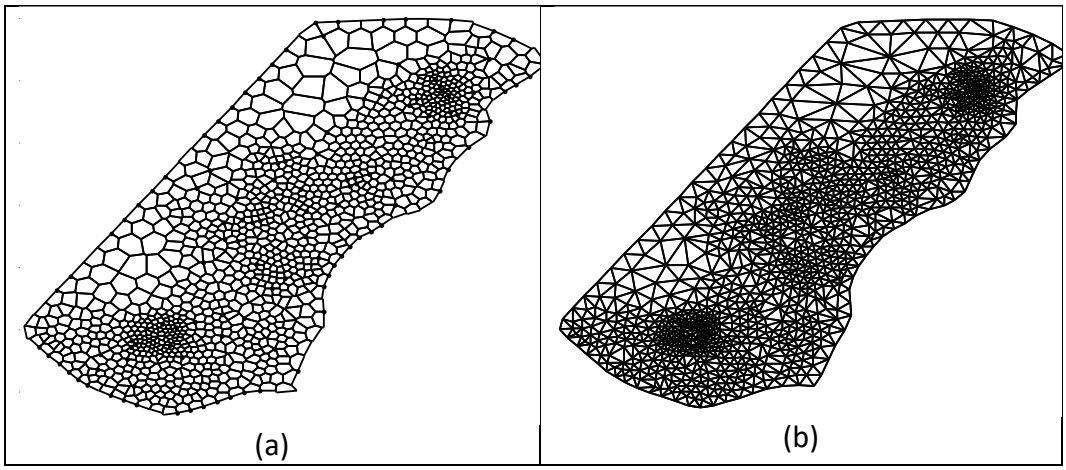

Figure 4 – Unstructured grids: (a) Voronoi diagram; (b) Delaunay triangulation

3
Unstructured Grid Generation

This chapter begins with a survey of previously used strategies for generating unstructured grids for reservoir simulation. The following sections describe our proposed strategy, starting from a refined (geological) model, going on to define the grid point distribution and ending with the Voronoi algorithms used to partition the reservoir into coarse cells.

3.1.
Grid generation survey

Voronoi grids can be generated for any arbitrary set of points. This does not mean, however, that any distribution of points is equally desirable. Cells with too many neighbors or highly irregular sides can greatly hinder the numerical solver. Ideally, the grid should be as regular as possible, with equally spaced cells, except where specific features must be represented. Those features may come from the geological model, such as the permeability distribution or faults, or be related to knowledge of flow conditions, such as well locations or streamline distribution.

A disadvantage of Voronoi grids and other point-distributed meshes, in opposition to corner point meshes, is that grid topology is determined indirectly from the grid point distribution, on an operation that can becomes costly for big models. Thus, distorted cells – with acute angles or high aspect ratios – can only be identified for certain at the end of the process (see Mlacnik et al., 2006). Modifications to the grid points in order to correct those problems require that the whole topology be recalculated.

Heinemann (1989) proposes starting with a regular hexagonal grid. Points near wells and boundaries are displaced from their initial position and aligned along those features. Finally, the location of the remaining points is smoothed to reduce irregular spacing.

Palagi et al. (1994) uses modules or templates to populate the grid. Regular hexagonal templates are used where there are no other dominant features.

Cylindrical templates are centered on vertical wells, while Cartesian or hexagonal templates can be rotated to conform to horizontal wells. He suggests using irregular templates for heterogeneities, but does not detail how. The transition between templates is automatically handled by Voronoi gridding.

Those early approaches are only capable of representing discrete attributes, such as wells and boundaries. More refined techniques are required to handle heterogeneities and other distributed features. Those algorithms usually start by determining a point density or spacing map based on relevant features, and then generate the grid respecting this map.

Mlacnik et al. (2006) performs a streamline simulation on the refined model, and uses the points defined by the intersection of streamlines and equipotential lines to generate the unstructured grid points in 2D. The segmentation of the domain into patches allows the use of distinct streamlines and equipotential lines on different regions. The cell limits are defined from the point set by a Voronoi / PEBI rule. The resulting grid has an almost rectangular structure within a region, where each cell has two neighbors along the same streamline, two along the same equipotential line, and four other possible “cross connections” along the diagonals (Figure 5). This structure means the grid can be indexed by a pair of values, and unit increments on the indexes indicate neighboring cells, even though the direction of the neighbor is variable. Laplacian smoothing and gradient based optimization are applied to minimize cross connections and to avoid severe aspect ratios.

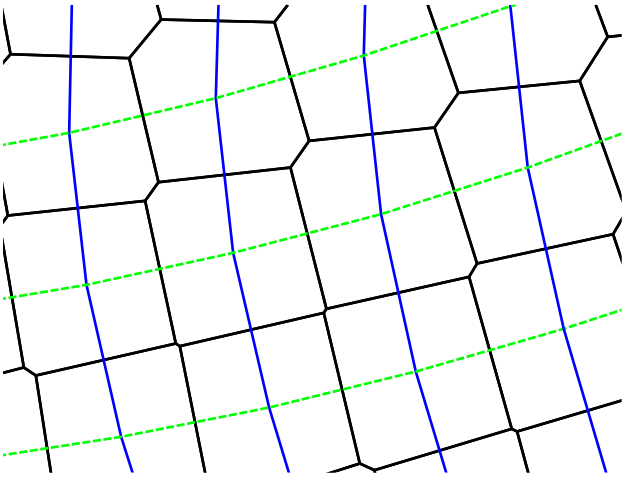

Figure 5 – Voronoi grid generated from streamlines (in blue) and equipotential lines (in green)

Evazi et al. (2009) adopts an advancing front technique with a background spacing grid in order to handle distributed features. The spacing grid is defined by a Poisson equation with sources related to relevant features, and determines the density distribution of points. Those sources can include both static and flow information. In the advancing front approach, new points are inserted sequentially, starting from the external border. Each inserted point forms a triangle with previous points, so that the length of the edges conforms to that defined by the background map.

Another technique class for unstructured grid generation is domain decomposition. It recursively subdivides cells where a higher density is needed. The base grid is usually a rectangular or triangular grid, and the final result similar to a structured locally refined grid (Figure 6). It also suffers from the same disadvantages of locally refined grids. The cell centers from this grid can be used as generating points for a Voronoi diagram (Figure 7) or Delaunay triangulation. Sahimi et al. (2010) adopt a similar strategy in order to represent rock matrix and fractures in the same computational grid.

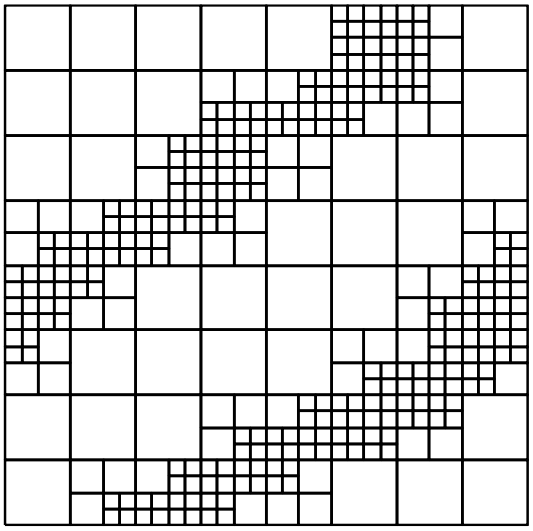

Figure 6 – Rectangular Domain decomposition grid

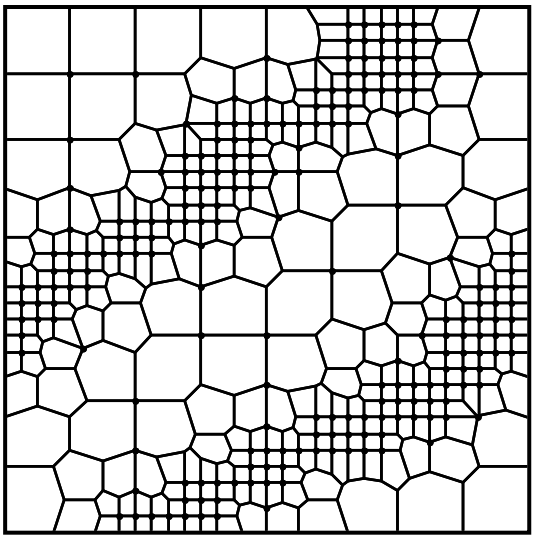

Figure 7 – Voronoi grid based on domain decomposition

3.2. Refined (Geological) Model

The starting point for building a reservoir flow simulation model is the geological model. This geological model describes the geometry of the reservoir, relevant features, such as faults, and the spatial distribution of several properties (porosity, permeability, facies, etc.). Usually those models are much more refined than is viable for flow simulation, both on horizontal and vertical directions. To the author's best knowledge, most commercial software for geological modelling can only generate structured gridding (Manckuk & Deutsch, 2009), using corner point meshes to provide some flexibility. Those corner point grids can occasionally have degenerate cells that affect flow simulation convergence.

The workflow proposed on this thesis starts with a refined geological model defined on a corner point grid with vertical pillars, using a geological modeling software such as PETREL or GOCAD. Although the grid has corner point representation, it is forced to be as regular as possible, so that it can be interpreted as a Cartesian grid for the rest of the proposed workflow. A true corner point grid (with vertical pillars) would be compatible with the unstructured grid generation steps described later, but could slow down some algorithms and require a more elaborate description of the refined model on MATLAB. The boundary of the reservoir and identified faults are described by line segments, and are considered to also have vertical pillars. A single grid layer is defined, in order to simplify the rest

of the workflow. This is reasonable for representing thin reservoirs. Multiple layers could be adopted with some minor modifications.

The geometry of the grid is exported from the geomodeling software as COORD and ZCORN keywords, recognizable by several flow simulation software. This format is more compact than the alternative CORNERS keyword (Table 1), requiring around a third of the number of values for a model with a high number of cells. And since a single layer is being exported, only COORD has to be read by MATLAB, avoiding the bigger file generated (ZCORN). The same procedure based only on COORD could be adopted with more layers, as long as their thickness was proportional to the total thickness and all pillars vertical. Reading (including properties) and preparing a single layer model with 361 x 110 cells took less than five minutes in MATLAB.

Table 1 - Values needed to represent grid geometry per keyword

Keyword	Number of values
COORD	6 * (NX + 1) * (NY + 1)
ZCORN	8 * NX * NY * NZ
CORNERS	24 * NX * NY * NZ

3.3. Discrete features representation (borders, wells and faults)

An initial regular point distribution can be used throughout the model, similar to what is done by Palagi (1994). A hexagonal distribution seems to be the best choice for this initial distribution (Figure 8.a). It allows flow in more directions than a Cartesian distribution, and is more robust to perturbations to the location of the points. With an initial Cartesian grid, even small perturbations create connections along diagonals (Mlacnik, et al., 2006 – see also Figure 5), with small transmissibility values.

Border and fault points are then generated, and initial points within the desired spacing distance of those (or outside the border) are erased (Figure 8.b). The position of the initial points can be smoothed by the techniques presented in section 3.5 (Figure 8.c) so that the spacing becomes more uniform. The position of points associated with the border and faults is preserved during this process.

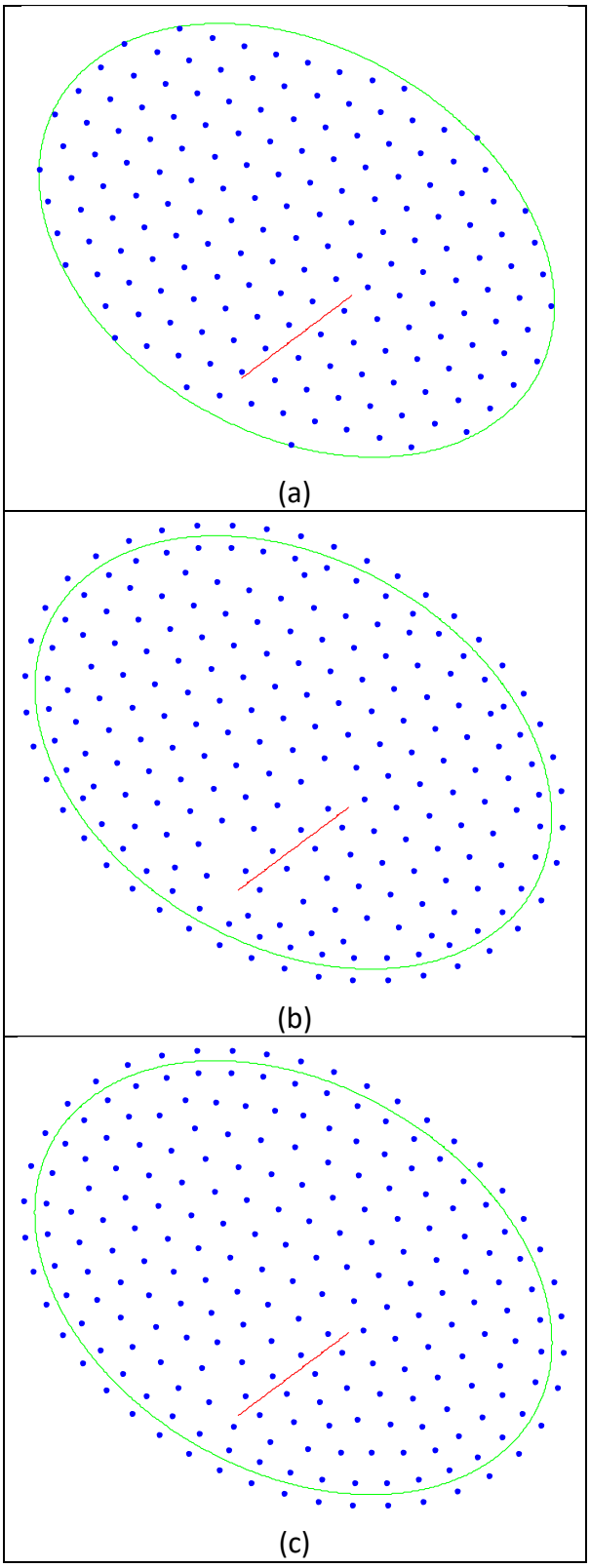

Figure 8 – Grid generation steps: (a) regular spacing; (b) border and fault point insertion; (c) smoothing. Grid points are in blue, border boundary in green and fault trace in red.

The inserted points can be placed directly along the line segments defining border and faults (Figure 9.a); however, this is not the best choice in most situations. For borders, cells created this way have edges external to the reservoir border that must be eliminated and replaced by a new edge created along the border to avoid overestimating the porous volume. Another issue is that the representative point of the cell will be directly on this new edge, far from the cell barycenter. This is only adequate when representing a boundary where pressure is specified, such as the interface with an infinite aquifer.

Inserting points along its trace is useful when representing conductive faults, since transmissibility in the direction of the fault can be easily calibrated by multiplying the transmissibility between the inserted cells. However, this representation is inadequate for sealing faults, since the cells would be located simultaneously on both sides of the fault.

A better representation for most reservoir borders and for sealing faults is to create equally distanced points to both sides of the line segments (Figure 9.b), so that the edge between cells determined by the Voronoi diagram falls approximately over the line segment. This way, the volume of the reservoir can be better represented, and transmissibility between sides of a fault set to the appropriate value.

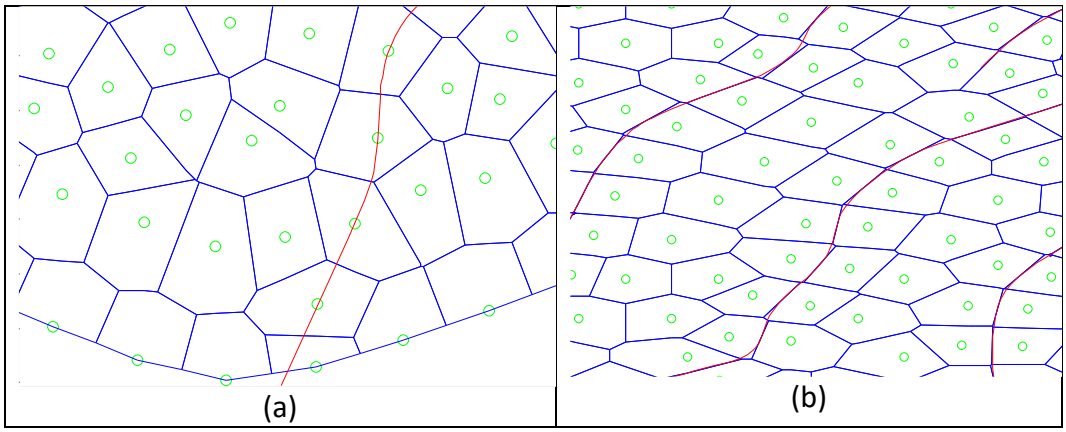

Figure 9 – Border and faults representations: (a) single layer; (b) parallel layers

Points representing wells can be inserted in a similar fashion, with a single vertical column of nodes representing vertical wells, and several points along a single layer for horizontal wells. Directional wells cannot be accurately represented along cell centers on a multiple layer 2.5D grid, since the nodes from each column

are placed on the same horizontal position. It becomes necessary to apply factors for each cell completion to represent the well deviation from cell center, similar to what happens on a structured grid, or to adopt a fully three-dimensional grid.

3.3.1. Common issues

The parallel double layer strategy can generate imperfect domain borders. Near corners with acute angles, two nodes may end up too close together or too distant apart. In regions with low cell density, an internal node may end between the double border layer and break the border's symmetry. More elaborate algorithms could be developed to handle those cases correctly. However, in general, the imperfections are small enough to not affect the flow simulation results.

A similar issue arises from intersecting faults (or the intersection of any two discrete features). Nodes generated independently for each of the faults can end up too close together and break the double layer symmetry near the intersection. Since the faults are assumed to not necessarily be straight and represented by multiple segments, a line segment intersection algorithm would be necessary to determine the exact crossing point. The intersection could then be represented by placing four equidistant nodes around this region, dividing the reservoir into four quadrants. Currently, intersecting faults are being treated by manually eliminating nodes that break symmetry.

3.4. Spacing Criteria

One of main advantages of unstructured models is the possibility of distributing cells in a flexible way, refining and coarsening the grid as needed to minimize the total number of cells (and thus also computational cost) while still representing the flow problem accurately. The position of the nodes can be influenced by discrete features, such as borders faults or wells, by properties from the refined model, or even by results from a refined scale flow simulation.

A background spacing map can be used as input during node placement or redistribution to represent the combination of those factors. Authors give different names for this map (Evazi et al., 2009; Hale, 2002; Bahrainian & Dezfuli, 2014).

For this work, the spacing map values are defined on the refined grid, so that any location within a specific refined cell is attributed the same spacing value.

3.4.1. Previously used spacing criteria

The most common strategy is to refine the model where high flow is expected. This improves the representation of pressure variations and fluid advancing fronts on those regions. However, a reliable representation of where high flow is expected usually demands some sort of simulation on the refined grid, even if single phase and incompressible.

Evazi compares three criteria for defining a spacing map. The permeability gradient on the horizontal plane is proposed as a measure of heterogeneity without requiring flow simulations, so that regions where complex flow is expected can be better refined. By running a single-phase flow simulation on the refined grid, a horizontal velocity magnitude criterion for spacing can be determined. A vorticity criterion is proposed as a better measure of heterogeneities and their effects than the previous two (while also requiring flow simulation).

Hale adopts a binary spacing map derived from faults and horizons interpreted from a seismic cube. The resulting effect is that cell edges end up aligned with those features. Bahrainian & Dezfuli define their spacing map based on the distance from fractures and fracture/matrix permeability contrast, so that regions near fractures become more refined.

Streamlines and equipotential lines, as employed by Mlacnik et al. (2006), can also be considered to form a spacing map. Streamlines will be closer to each other in high flow regions, resulting in smaller cells on those regions. Large pressure drops, which can happen near wells or where flow is restricted, cluster equipotential lines together and therefore also reduce cell size. The combination of those two factors can shape cells with a high aspect ratio.

3.4.2. Proposed spacing criteria

This work proposes to generate unstructured grids that do not rely on refined scale flow simulation to determine the cell size distribution. Although flow

simulation might be the most effective method to optimize this distribution, it has an associated computational cost and remains valid only as long as well configuration does not change much.

Several features of the reservoir model can be associated with flow condition and are thus useful in defining cell size distribution. All proposed methods are heuristic, since no closed equation defining an optimal cell size could be determined. Mean cell spacing $\bar{L}$ or minimum (L^-) and maximum (L^+) spacing are used as parameters to ensure the spacing remains within a desired range. The spacing criteria for different features be combined into a single, more elaborate spacing map.

Absolute permeability is the most immediate property one can associate with high flow rates (fluid mobility changes during simulation and so is disregarded). It can be used as an inverse parameter to determine spacing, so that regions where higher flow is expected will be better refined (eq. (9) or (10)). An alternative is to use the permeability gradient instead of permeability, as done by Evazi, so that refinement will occur at the most heterogeneous regions (eq. (11)). When permeability varies over several order of magnitude, a logarithmic transformation can be applied to the property before determining the spacing map to avoid excessive variations. Figure 10 exemplifies two dimensional permeability based spacing maps.

$$L(x) = L^+ - (L^+ - L^-)(k(x) - k^-)/(k^+ - k^-) \quad (9)$$

$$L(x) = \bar{L}\,\bar{k}/_{k(x)} \quad (10)$$

$$L(x) = L^+ - (L^+ - L^-)(|\nabla k| - |\nabla k|^-)/(|\nabla k|^+ - k|\nabla k|^-) \quad (11)$$

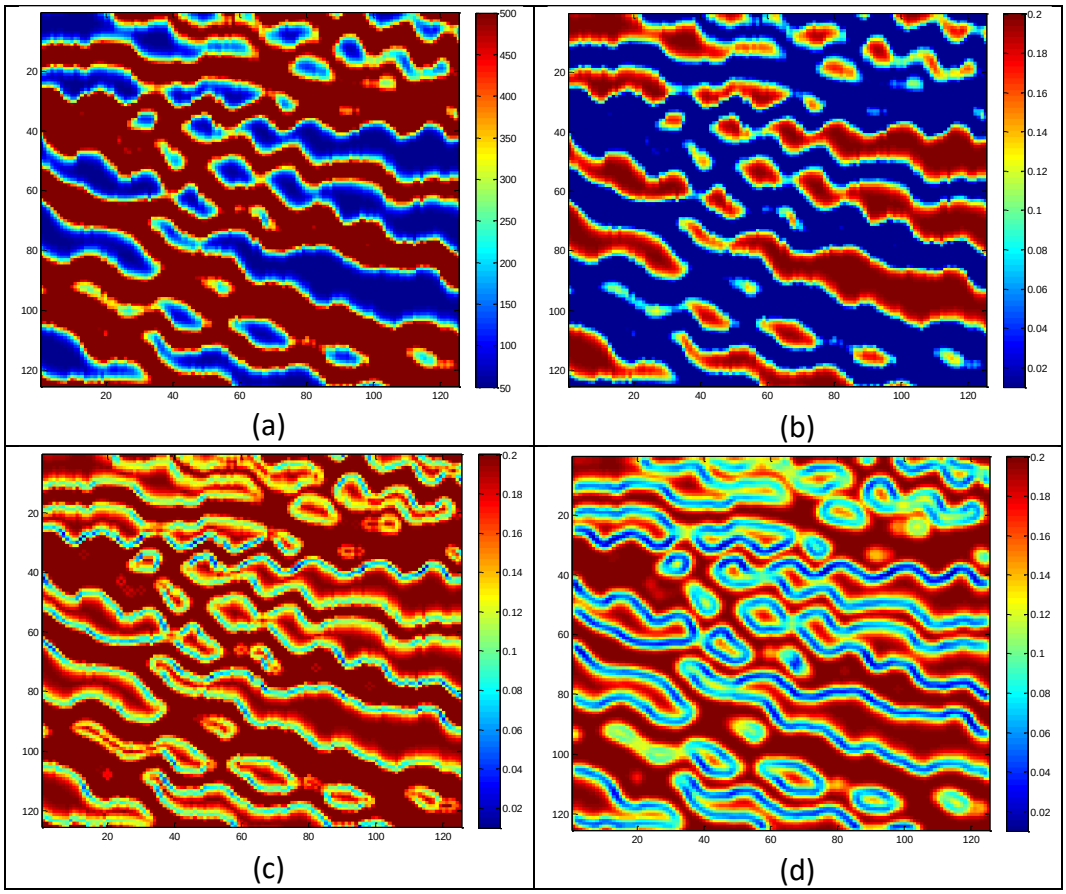

Figure 10 – (a) Permeability map; (b) Permeability based spacing map; (c) Permeability gradient based spacing map; (d) smoothed version of the previous spacing map.

Well configuration is also a determinant factor for flow distribution. Therefore, another useful heuristic criterion for spacing is to refine the grid near well locations. Without relying on flow simulations, this can be done by assuming a radial flow from the wells, so that spacing varies only with distance (eq. (12)). An influence parameter λ_i can be employed to change the size of the affected region for each well (those with higher rate and located on more permeable regions should have a larger influence). Parameter r determines the ratio between maximum and minimum spacing. A grid generated with such a criterion will no longer be valid if the well locations are changed.

$$L(x) = L^{+} \prod_{i}^{wells} \frac{1 + (r-1)\min\left[\left({}^{d(x)_i}/_{\lambda_i}\right)^2, 1\right]}{r} \qquad (12)$$

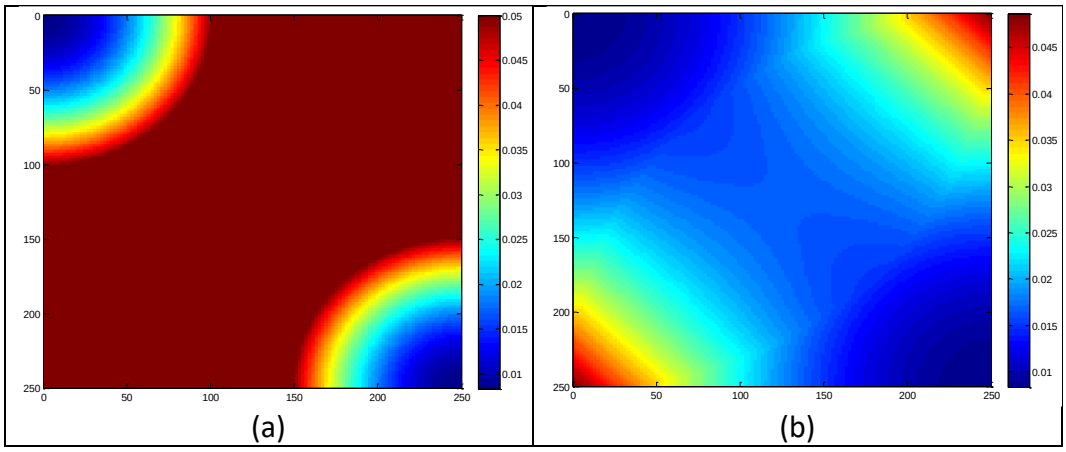

Figure 11 – Spacing maps for two wells at opposite corners: (a) influence = 50 m; (b) influence = 250 m.

Regions near faults and fractures are also candidates for refinement. Permeability channels and flow barriers caused by fault throw can cause abrupt changes on flow within a damage zone along faults. A spacing equation (eq. (13)) similar to that proposed for wells can be applied in those situations, with refinement increasing near the faults.

$$L(x) = L^{+} \prod_{i}^{faults} \frac{1 + (r-1)\min\left[\left({d(x)_i}/{\lambda_i}\right)^2, 1\right]}{r} \tag{13}$$

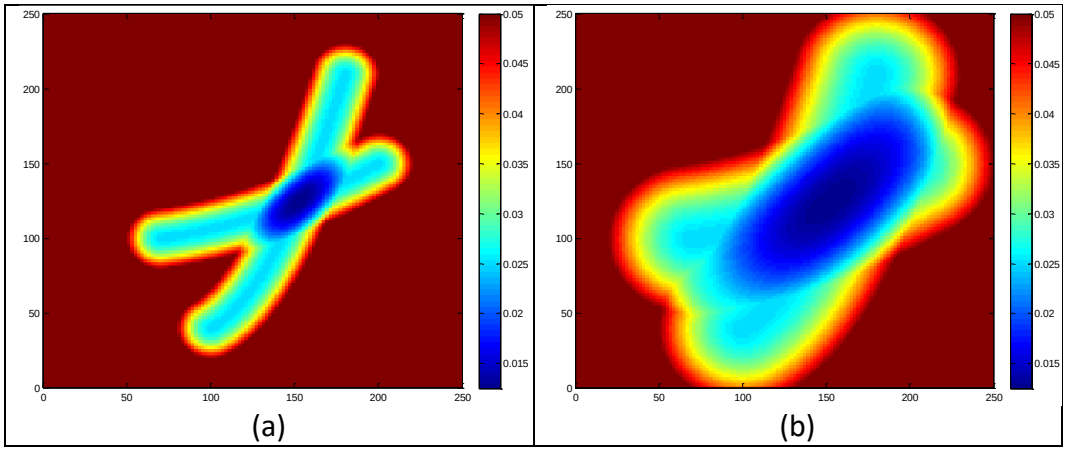

Figure 12 – Spacing maps for two overlapping faults: (a) influence = 20 m; (b) influence = 50 m.

Abrupt variations should be avoided in the spacing map, since they can cause convergence problems for grid point generation algorithms and will not be adequately represented on a coarse grid anyway. Permeability gradient in particular is prone to this sort of issue. Image smoothing techniques can be applied to the

spacing to mitigate this problem, although care should be taken not too smooth out the main features. A Gaussian filter can be used for this purpose, as exemplified on Figure 10.d.

An attempt was made to develop a spacing criterion that maximizes the admissible time step, so that simulations can run faster. This should be possible by minimizing the maximum pressure and saturation temporal variations on the grid. Pressure variations are not expected to be high in a model with pressure maintenance. Saturation variations above bubble point pressure happen mostly along the advancing water front, whose position changes during simulation. The front will advance with higher velocity on high permeability paths, which suggests those cells should be bigger (opposite to the previously suggested spacing strategy). The solution to this problem demands generating cells with a direction dependent spacing, or even an adaptive grid. Both of those alternatives fall outside the scope of this study, and thus this approach to grid spacing was abandoned.

3.5. Node Smoothing and Redistribution

With an algorithm such as advancing front, the nodes should automatically conform to the predefined spacing criteria. At most, a smoothing technique might be necessary because of points inserted to represent faults and wells. However, since this workflow starts with a regular template, the conformance to a spacing criterion must be enforced at a distinct stage. The same physical models used for smoothing can be applied for this, using the spacing criteria as weighting factor for the forces involved.

Even if the desired spacing is constant in the entire reservoir, some sort of smoothing is useful in order to minimize the effects of node insertion and removal near the border and faults. This can be achieved with a simple Laplacian smoothing algorithm, or with attraction/repulsion models based on some physical model.

Disadvantages related to those sorts of methods are that they require several iterations, one or more parameter values must be calibrated, and the definition of a stopping criteria. Some combinations of those values might be unstable, leading to worse results than the initial node distribution.

3.5.1.
Laplacian Smoothing

Laplacian smoothing moves each node towards the barycenter of its neighbor's locations $\boldsymbol{b}$ (Mlacnik et al., 2006) on each interaction:

$$\boldsymbol{x}^{k+1} = (1-w)\boldsymbol{x}^{k} + w \sum_{j=1}^{n_{neighbors}} \frac{\boldsymbol{b}_j^k}{n_{neighbors}} \tag{14}$$

The parameter w determines the size of the movement on each iteration k. With a rectangular or hexagonal lattice, the structure can define the neighbors (as done by Mlacnik). However, with a truly unstructured grid, the neighbors must be defined as an arbitrary number of closest nodes. In those cases, and especially if you have a variable density of nodes, using laplacian smoothing can actually result in node clusters and empty regions on the domain (Figure 13.b).

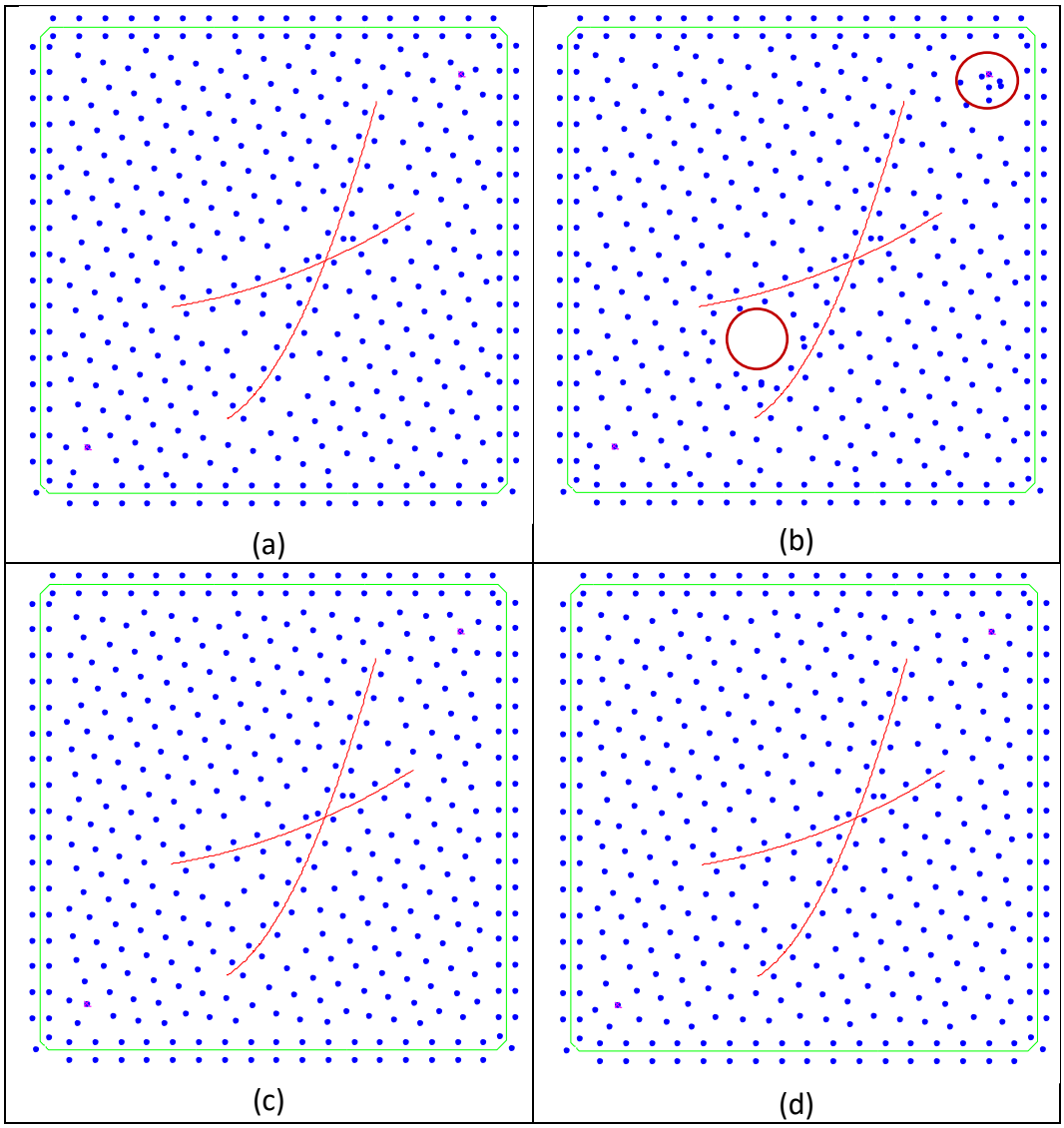

Figure 13 – Grid nodes smoothing: (a) Initial node set with hexagonal template; (b) Laplacian smoothing with issues highlighted by red circles; (c) Electrostatic repulsion; (d) Atomic forces.

3.5.2. Electrostatic Repulsion

A physical model that can be used for smoothing the node distribution is to emulate an electrostatic repulsion between nodes. All non-fixed nodes are displaced from their initial position by a small step in the direction of the resulting force, considering only the nearest neighbors (eq. (15)). This process is repeated several times. The repulsion diminishes with distance, so that theoretically the system can reach equilibrium after a sufficiently high number of iterations. The fixed nodes at the reservoir boundary act as a barrier that avoids (most) nodes from being ejected

from the domain. The factor w includes the effect from particle charge, which can be the same everywhere for a constant spacing map. A variable spacing map can be integrated as a variable charge for the particles (hence affecting w), so that repulsion becomes higher where the expected node density should be lower.

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^{k} + w \sum_{j=1}^{n_{neighbors}} \frac{\left(\boldsymbol{x}^{k} - \boldsymbol{b}_j^{k}\right)}{\left\|\boldsymbol{x}^{k} - \boldsymbol{b}_j^{k}\right\|^{3}} \tag{15}$$

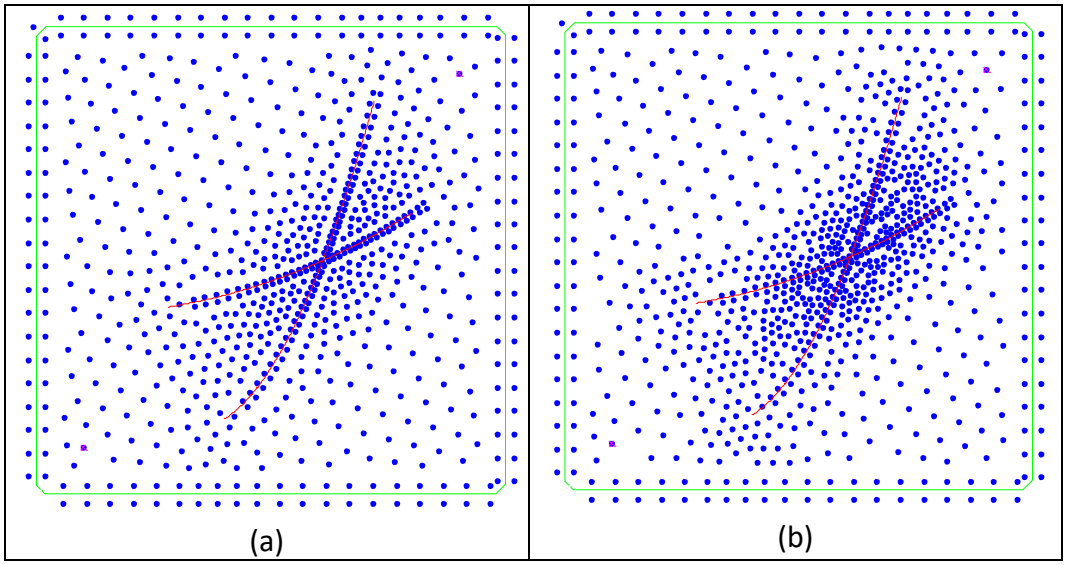

Figure 14 – Grid node redistribution for the spacing map from Figure 12.b: (a) Electrostatic repulsion; (b) Atomic forces.

3.5.3. Atomic forces equilibrium

Both Laplacian smoothing and electrostatic repulsion depend on an appropriate choice of parameters and number of iterations, especially with a variable spacing map. A model that was found to be more robust to calibration effects is to emulate inter-atomic forces (Hale, 2002). Two nodes whose distance corresponds to the local spacing length are in equilibrium. If they are closer, a repulsion force appears, while if they are further apart there is an attraction force. Hale uses a generic optimization algorithm to determine a node distribution that minimizes the global atomic potential (eq. (16**Erro! Fonte de referência não encontrada.**)) defined by the interaction between all nodes. In this work, all nodes are moved instead from their initial position in the direction determined by the

resulting atomic force in several small steps (eq. (17) and (18)). It is assumed that the spacing map varies slowly, so that its contribution to the potential gradient can be disregarded.

$$\Phi(u) = \begin{cases} \frac{153}{256} - \frac{9}{8}u + \frac{19}{24}u^3 - \frac{5}{15}u^4, & 0 \le u < {}^3/_2 \\ 0, & u \ge {}^3/_2 \end{cases} \quad (16)$$

$$\frac{d\Phi}{du} = \begin{cases} -\frac{9}{8} + \frac{19}{8}u^2 - \frac{5}{4}u^3, & 0 \le u < {}^3/_2 \\ 0, & u \ge {}^3/_2 \end{cases} \quad (17)$$

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k - w \sum_{j=1}^{n_{neighbors}} \nabla\Phi\left(\frac{\|\boldsymbol{x}^k - \boldsymbol{b}_j^k\|}{L(\boldsymbol{x}^k)}\right) \quad (18)$$

When used with a spacing map, electrostatic repulsion results in a smoother node distribution than atomic forces (Figure 14). However, atomic forces can better represent high variations on the node density distribution.

3.5.4. Related issues

When the spacing map values vary over a large range, using those algorithms may result in regions without any points. This happens because regions with small spacing act as attractors, and starting with a regularly spaced distribution does not supply enough nodes to fill those regions. The strategy adopted to minimize these effects was to generate additional nodes after every few iterations of the redistribution algorithm, using a bubble space filling algorithm. A new site is inserted whenever the site nearest to its proposed location is more distant than the local spacing parameter (so that a circle or sphere centered at the new node with radius equivalent to the spacing parameter does not contain any other node). After all possible sites have been inserted, the redistribution algorithm can run again.

Distorted cells, with high aspect ratio, acute angles or too many neighbors can cause convergence issues during flow simulation and should be avoided. Although the smoothing techniques mentioned help avoiding those issues, a more direct property would be to optimize directly those properties when adjusting node position. However, determining those aspects requires generating the Voronoi diagram beforehand, which is a costly operation with many cells and that would have to be repeated after any adjustments. With other grid types, those can be a useful alternative.

3.6.
Voronoi Gridding

A Voronoi diagram is the partitioning of a 2D or a 3D space into the regions or cells nearest to each point (usually called sites or nodes) in a set. Although different metrics are also possible, such as Manhattan or Mahalanobis distances (Figure 15), in this work only the most commonly used Euclidian distances (L_2 norm) are adopted. Edges from a Voronoi diagram are equidistant to the nodes from the two cells they divide, while vertices from the diagram are equidistant to three or more nodes.

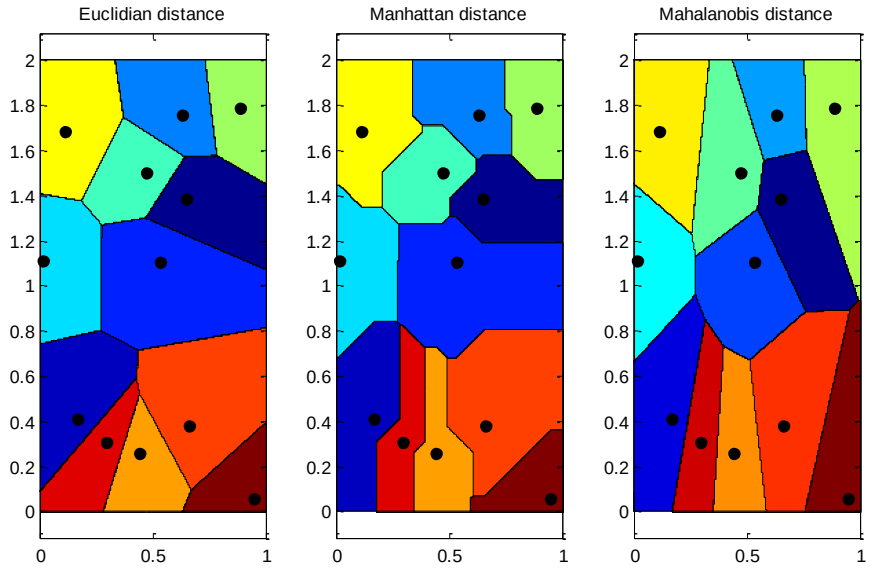

Figure 15 – Voronoi diagrams for different distance metrics

The simplicity and flexibility of Voronoi diagrams led to applications in a wide variety of fields. In crystallography, it is used to describe the structure of metals. In telecommunications, it defines the coverage of cellular telephone coverage. The Voronoi diagram also corresponds to a nearest neighbor classifier. One of the first and most notorious applications of the Voronoi diagram was John Snow's study on the London cholera epidemic in 1854. The doctor delimited the regions closest to each water pump in a city map, and correlated the deaths to one of those regions. The epidemic was subdued after the government shut down this pump.

Although a Voronoi grid can be defined in two or three (or even more) dimensions, there is a significant increase in cost to determine a three dimensional

diagram. It also eliminates the layer structure of the grid that is very useful for characterizing most reservoirs (fluid flow in the vertical direction is usually more restricted than horizontally). A more suitable solution is to use a grid that is unstructured in the horizontal direction and structured in the vertical direction. A 2.5D grid can be defined by applying a single areal Voronoi diagram, extending the cells in the vertical direction, and subdividing in layers as needed.

3.6.1. PEBI grids

Heinrich (1987) was the first to call attention to the useful properties of a Voronoi diagram (with Euclidian metrics), called by him as the perpendicular bisection grid (PEBI), in discretizing and solving various differential equations. This name refers to the fact that the path between two nodes is orthogonal to the edge of the Voronoi diagram separating their cells.

Heinemann et al. (1989) develops the discretization equations for porous flow media following this scheme, and shows that this property greatly simplifies the equations. The CVFD discretization for the mass flow rate of a component c between cells i and j can be described by eq. (19), where $A_{i,j}$ is the interface area between the cells, $\lambda_p = k_{rp}/\mu_p$ is the phase mobility and $\vec{n}$ is the interface normal vector.

$$Q_{i,j,c} = A_{i,j} \sum_p \rho_{cp} \lambda_p (\bar{\bar{k}} \nabla \Phi).\vec{n} \tag{19}$$

For an isotropic permeability field and an interface perpendicular to the path between nodes, a two point flow approximation (TPFA) is possible, as exemplified by eq. (20). Because of this property, Voronoi grids were the most common grid in early proposals for unstructured grid reservoir flow simulation (Palagi et al., 1994).

$$\left(\bar{\bar{k}} \nabla \Phi\right).\vec{n} = k(\nabla \Phi.\vec{n}) = k\left(\frac{\partial \Phi}{\partial \vec{n}}\right) \cong k\frac{\Delta \Phi_{i,j}}{l_{i,j}} \tag{20}$$

For anisotropic permeability fields, a similar simplification is only valid if the principal directions of $\bar{\bar{k}}$ are perpendicular to the cell interfaces. Possible solutions are to use a multiple point flow approximation (Chen et al., 2007), or to propose a different gridding strategy. Gunasekera et al. (1997) transform an anisotropic

reservoir into an isotropic computational space before computing the PEBI grid, so that the perpendicular flow property is preserved.

3.6.2. Voronoi grid generation

The Voronoi diagram is the dual of a Delaunay triangulation, meaning there is a unique triangulation for any Voronoi diagram, and vice versa. The Delaunay triangulation forms triangles by connecting the nodes in a way that no other node is inside the circle formed by its vertices. In 3D, the basic shape becomes a tetrahedron, and the Delaunay condition is verified inside the sphere formed by all four vertices. A Delaunay triangulation can be obtained from a Voronoi diagram by connecting the nodes of neighboring cells. A Voronoi diagram can be obtained by bisecting the edges of a Delaunay triangulation.

Implementing a Delaunay triangulation has the advantage that the number of vertices and neighbors for each element is constant, while a Voronoi cell can have an arbitrary number of neighbors (Ledoux, 2007). Because of this, it is common to generate a Delaunay triangulation rather than directly determine the Voronoi diagram. Other algorithms for determining a Voronoi diagram can be divided into three categories:

I. Divide and Conquer – the domain is iteratively split in subspaces containing part of the nodes.
II. Incremental – Nodes are added one at a time to the diagram, and the edges are recalculated each time.
III. Sweep – An imaginary line (or plane) crosses the domain. Edges are determined by the collapse or creation of influence regions of the nodes already swept by.

In general, naïve algorithms of the first two categories have $O(n^2)$ computational cost , while an optimized algorithm, such as Fortune (a sort of sweep algorithm) has a $O(n * \log n)$ (Okabe at al., 2009). Efficient algorithms for Delaunay Triangulation, such as Bowyer-Watson, also have $O(n * \log n)$ cost (Rebay, 1993).

In this work, both an incremental method and Fortune's algorithm were implemented. For the number of cells expected for a typical reservoir model (over 1000, even with a single layer), Fortune was noticeably faster. The incremental method does have one significant advantage – new nodes can be included without

the need to recalculate the entire grid, as long as they do not become part of the external border. With an appropriate data structure, it is possible to determine the initial Voronoi Diagram with the Fortune algorithm, and include additional nodes with the incremental method.

Degenerate cases, such as collinear nodes or more than three nodes on the same circle have to be handled by the algorithms and can multiply the size of the code (Ledoux, 2007). In order to avoid overcomplicating the code, a small random perturbation was included on the nodes location, when starting from a rectangular or hexagonal lattice, so that no nodes are perfectly aligned. Another issue is that several analytical geometry verifications are not straightforward with floating point algebra. Verifying if a value is null or if two values are equal (necessary, for instance, to determine if a point is included in a line or plane) can fail due to precision problems. To mitigate those issues, a small tolerance was adopted on those checks.

3.6.3. Incremental method

An incremental method for generating a Voronoi Diagram was implemented for this research, based on Wolfman's notes (2000) for his course on the University of Washington. It is relatively simple, and can be summarized as follows:

```
Begin with four initial nodes surrounding the actual node set
Create cells for those nodes so that they form a rectangular frame to
the data points, and add them to cell_set
FOR each node n in your dataset:
       Create new cell c with node n
       FOR each cell ci in cell_set:
              Determine the perpendicular bisector (line equidistant
              from two points) between the nodes of c and ci
              FOR each edge e of cell ci:
                     Delete e if nearer to the c's node than ci's node
                     IF e intersects the bisector, cut the segment closer
                     to c's node
              IF two edges were cut, connect them with a new edge (i.e.,
              a segment of the bisector)
       Add cell c to cell_set
Clip edges outside your domain, adding new edges when necessary
```

Although the incremental method is slow for a large set of node points, it is still useful in some situations – for instance, if we want to insert a new node to a preexisting Voronoi diagram.

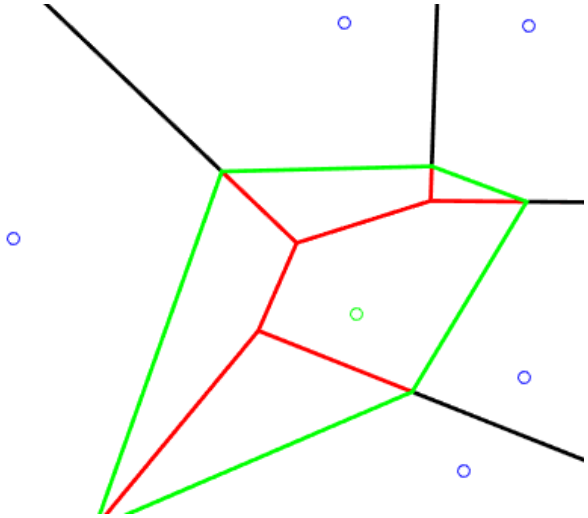

Figure 16 – Voronoi node insertion with the incremental method. Erased edges are in red, and inserted edges in green.

3.6.4. Fortune Algorithm

The Fortune algorithm is much more complex than the incremental method, and sensible to degenerate cases. The implementation for this work was based on the description given by De Berg et al. (1997).

The Fortune algorithm sweeps the domain with a horizontal line that activates events as it moves. For any given position of the sweep line, the Voronoi diagram is known at the subdomain above the so called beach line. This line is composed by parabolic arcs, each equidistant from the sweep line and an already swept by node (Figure 17).

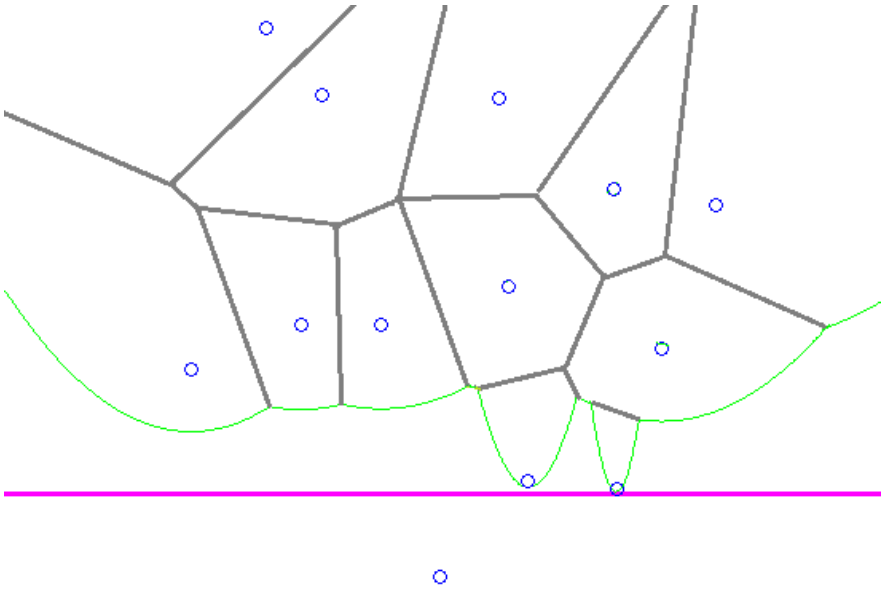

Figure 17 – Fortune algorithm elements: Beach line (green), sweep line (pink) edges and half-edges (gray) and nodes (blue circles)

Two sorts of events are possible:

I. A point event happens when the sweep line passes over a node. This event adds a new arc to the beach line, associated with the new node and adding a new cell to the diagram (Figure 18).

II. A circular event happens when the circle determined by three nodes related to consecutive arcs from the beach line is tangent to the sweep line. The central arc ends at this point, corresponding to a vertex from the Voronoi diagram, and closing a cell from the diagram (Figure 19).

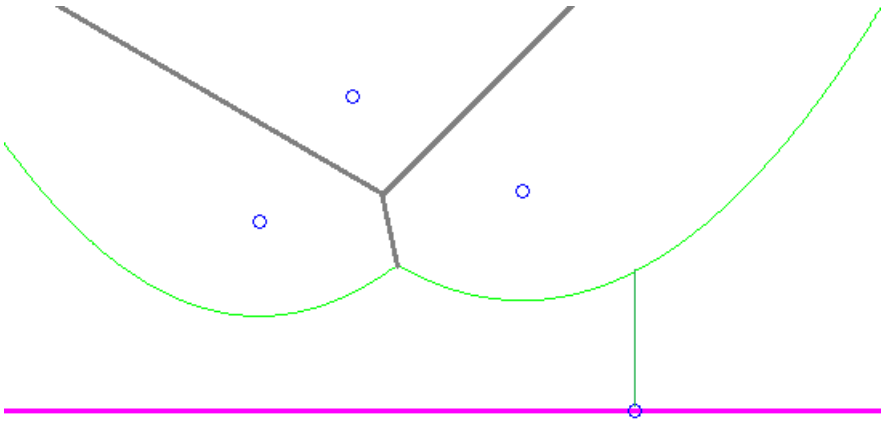

Figure 18 – Example of point event for Fortune algorithm

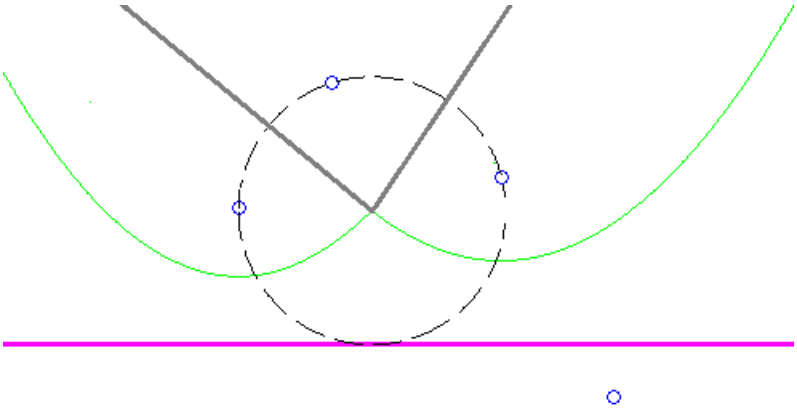

Figure 19 – Example of circular event for Fortune algorithm

The sweep line does not have to check every possible position, only activate at the following event from on a sorted queue. All point events are added at the start of the algorithm, while circular events can be added or removed from the queue whenever changes happen to the beach line. The beach line can be stored simply as the sequence of nodes that define the currently active parabolic arcs.

After a point event happens, a new edge does not grow at a single direction from its vertex, but rather grows in two directions from an intermediate point of the edge. The sweep algorithm ends when the event queue is emptied. Several edges will be connected at infinity and must be clipped according to the domain limits.

3.6.5. Related issues

For the Fortune algorithm to perform in $O(n * \log n)$ time, the event queue must be implemented as an efficient structure for insertion. This is usually achieved with a binary tree. Even though this structure was not implemented, the Fortune algorithm implementation used for this work is still much faster than the incremental algorithm. Table 2 illustrates this behavior with the time used to generate the Voronoi diagram with both algorithms in MATLAB. The range of models on this work have less than 1000 cells, and thus the Voronoi diagram can be evaluated in under a minute. In order to work efficiently with much bigger models, a binary tree implementation becomes mandatory.

Table 2 – Voronoi diagram generation time

Number of cells	Running time (s)	
	Incremental	Fortune
38	1,1	0,14
123	6,7	0,42
387	51	2,0
546	65	3,0
1180	386	10
3402	3256	74

Figure 20 presents some examples of the Voronoi grids generated throughout this work (all considering a single horizontal layer). All grids have in average less than six neighbors per cell (this is reasonable since we start with a hexagonal template). Models with higher cell density variability have a few cells with more connections – up to nine or ten in extreme cases. A higher number of average connections could have a negative impact on the numeric solver.

Voronoi cells aspect ratios are low – no cases of 'slivers', cells with long edges and small volumes that can appear with triangulations, were identified. However, a few connections can happen along very small edges. These connections increase the computational cost of the numerical flow problem, but do not contribute much to the result.

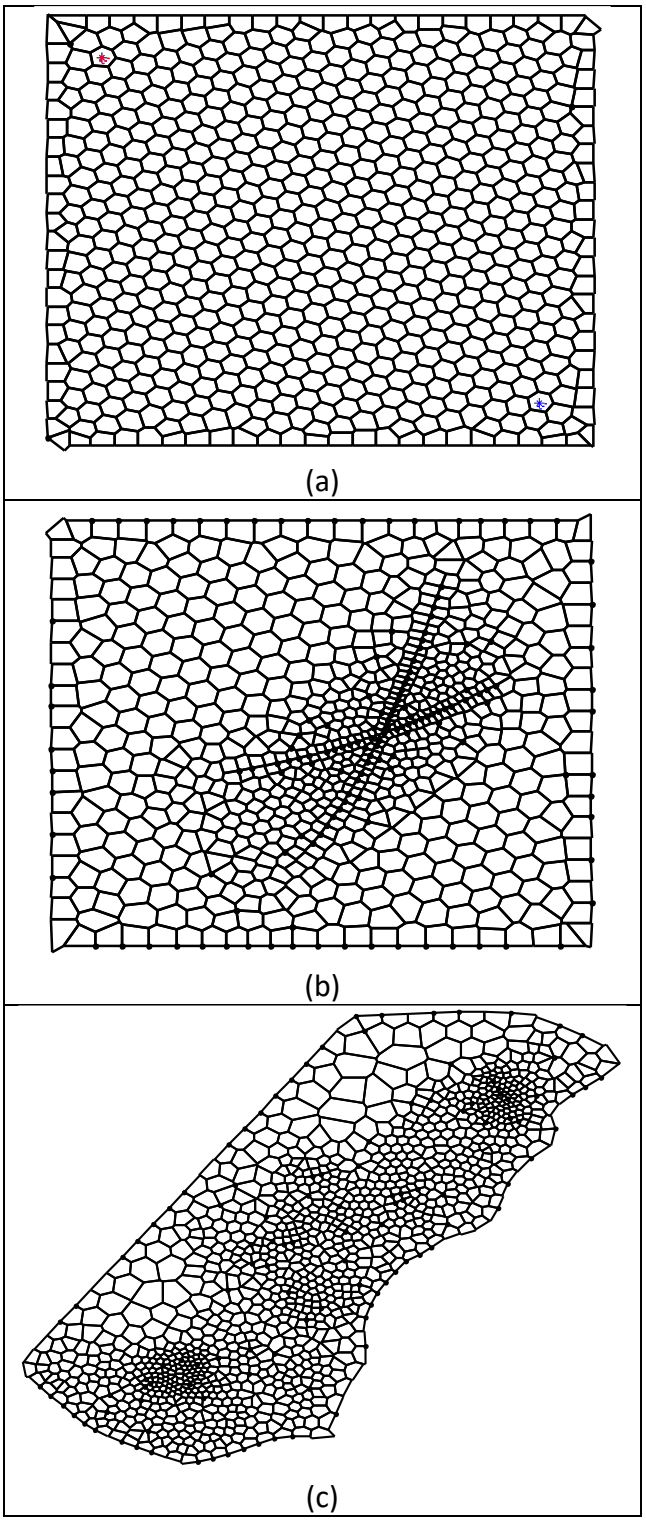

Figure 20 – Examples of Voronoi grids: (a) nearly hexagonal; (b) refinement near two faults; (c) refinement near wells and high porosity.

4
Permeability and transmissibility upscaling

This chapter begins with a review of upscaling techniques for the structured (section 4.1) and unstructured (section 4.2) grid cases. New alternatives for unstructured grid upscaling, without resorting to flow simulation, are proposed in sections 4.3 and 4.4. Some comparisons and tests with those methods are presented after that. The chapter is concluded with a short discussion on equivalent radius calculations for unstructured grids, a problem related to grid scale.

Transmissibility is frequently mentioned on those discussions. In most cases, it refers to the geometric component of transmissibility, which can be determined before flow simulation starts. The mobility component of transmissibility depends on fluid saturations and must be reevaluated at each time step. Therefore, it cannot be evaluated during upscaling.

4.1.
Permeability upscaling on structured grids

Even with the advances on computational power in the last decades, flow simulation is still a demanding task, and in many cases cannot be performed on the same refined grid as geological modelling. Thus, property upscaling is almost always a required step for flow simulation.

The general principle of upscaling is to determine equivalent properties on a coarse grid so that simulation behavior stays as close as possible as if performed on the refined grid (Figure 21). For most properties, this is an almost trivial operation – the main concern is weighing by porous volume. However, permeability upscaling is more nuanced due to directional and multiphase effects. A thorough survey of permeability upscaling methods for structured grids can be found on Renard & De Marsily (1997). Upscaling methods can be classified as heuristic, analytical or flow based.

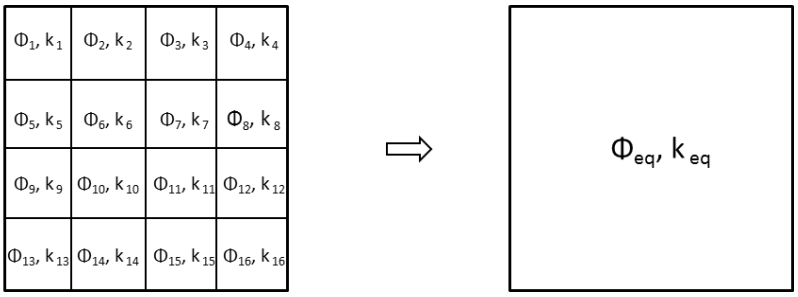

Figure 21 – General upscaling principle

4.1.1. Upscaling bounds and heuristic methods

Several methods have been proposed for establishing bounds for the permeability of an upscaled cell. The most fundamental bounds are those proposed by Wiener, that remain valid for any set of conditions. They can easily be applied, even for more complex mesh geometries. However, variations with flow direction are ignored, and the limits are too wide to be of use in many conditions. The lower Wiener bound is given by the harmonic mean of the refined permeabilities, while the upper bound is given by the arithmetic mean (eq. (21)).

$$\frac{N_i N_j N_k}{\sum_{i,j,k}^{N_i N_j N_k} \left(k_x^{i,j,k}\right)^{-1}} \ll K_{x,block}^{Wiener} \ll \frac{1}{N_i N_j N_k} \sum_{i,j,k}^{N_i N_j N_k} k_x^{i,j,k} \tag{21}$$

Cardwell & Parsons attribute different bounds for each flow direction, based on an electrical equivalence analogy. The lower bound is given by the arithmetic mean of the harmonic mean calculated on the flow direction, while the upper bound takes the harmonic mean of the arithmetic mean on the planes orthogonal to flow direction (eq. (22) and Figure 22).

$$\begin{aligned} \frac{N_i}{N_j N_k} \sum_{j,k}^{N_j N_k} \left(\sum_{i}^{N_i} \left(k_x^{i,j,k}\right)^{-1} \right)^{-1} &\ll K_{x,block}^{CardwellParsons} \\ &\ll \frac{N_i}{N_j N_k} \left(\sum_{i}^{N_i} \left(\sum_{j,k}^{N_j N_k} k_x^{i,j,k} \right)^{-1} \right)^{-1} \end{aligned} \tag{22}$$

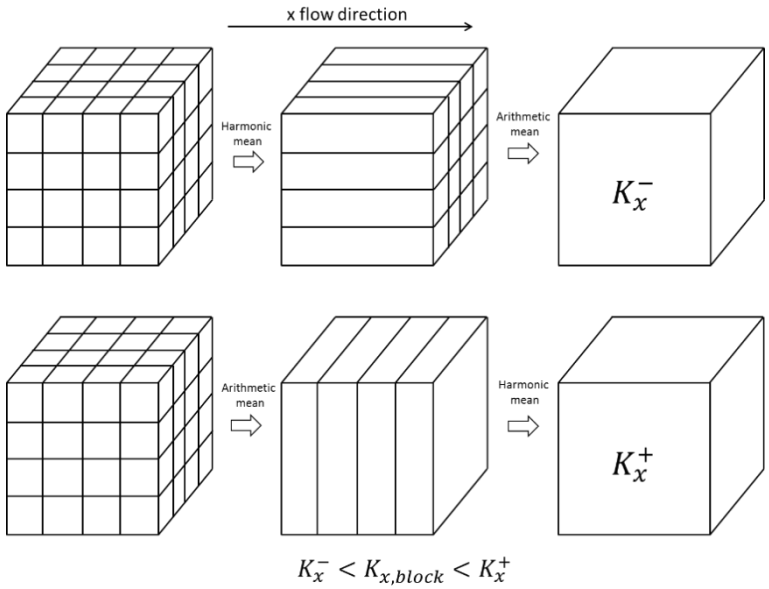

Figure 22 – Cardwell & Parsons algorithm principle (adapted from Renard & De Marsily, 1997)

Other bounds, such as Hashin and Shtrikman and Matheron, are based on assumptions that restrict their application for real reservoir modelling (binary media, for instance).

Heuristic formulas can be used to estimate upscaled permeabilities from a pair of bounds. It is possible to use a weighted average or a power average, but the definition of an exponent creates an additional parameter that needs to be fit. A simpler alternative is to take the geometric mean of the bounds (eq. (23)).

$$k_i = \sqrt{k_i^- k_i^+} \tag{23}$$

Several analytical upscaling methods have been proposed – percolation, effective media theory, streamlines. However, many of those are based on restrictive assumptions, such as a binary media.

4.1.2. Renormalization

Renormalization, and its many variations, is an analytical method with wider application. It is analogous to determining an equivalent resistance in an electrical network, by successive transformations (Figure 23). The transformations can be of the star-triangle type (King, 1989), or done one direction at a time (Le Loc'h, according to Renard & De Marsily, 1997).

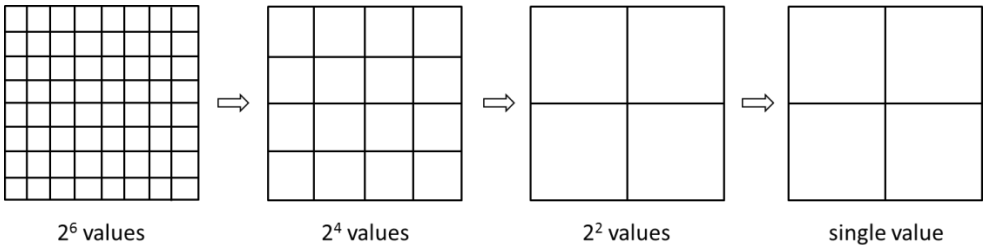

Figure 23 – Renormalization principle (adapted from Renard & De Marsily, 1997)

4.1.3. Flow-based methods

The last kind of permeability upscaling methods are those based on some sort of numerical solution of the flow problem on the refined grid, often considering a reduced domain and assuming a single phase. Local methods consider the flow problem on a single coarse cell at a time. This is actually more time efficient than solving the refined flow problem on the entire domain a single time.

A directional permeability on each main direction can be determined by fixing the pressure or head on two opposite sides of the block and measuring the total flow rate crossing the medium (eq. (24)). This is known as the permeameter condition (Figure 24).

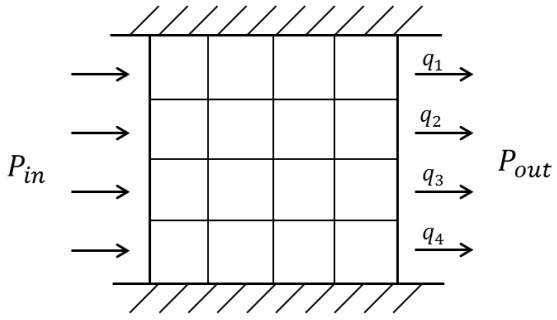

Figure 24 – Flow-based upscaling with permeameter conditions

$$k_{x,block} = \frac{\mu L}{(P_{in} - P_{out})} \sum_{jk} \frac{q_{jk}}{A_{jk}} \tag{24}$$

By using periodic or linear boundary conditions instead, and measuring the flow in all directions, it is possible to determine a full tensor permeability for the block. However, only periodic conditions can guarantee a symmetric tensor.

Using refined scale simulations on the whole domain, or considering the neighboring cells, allows non-local information to be taken into account when

estimating the permeability tensor. More than one set of boundary conditions still have to be considered. The problem usually becomes over-determined and is solved by least squares minimization.

4.1.4. Permeability versus transmissibility upscaling

Independently of the adopted upscaling method, an important concern is that they may become biased, especially with heterogeneous reservoirs. Upscaling from the most refined scale available directly to transmissibilities between coarse cells (or inter-cell permeabilities) seems to result in a smaller bias than determining coarse cell permeabilities for simulation, according to experiments by Romeu & Noetinger (1995). The transmissibility between cells is related to the equivalent inter-cell permeability by geometric aspects and phase mobility (eq. (25)).

$$T_{i,j,p} = \frac{A_{i,j}\lambda_p k_{eq,i,j}}{l_{i,j}} \tag{25}$$

This is reasonable if we consider that the flow simulator must transform the cell permeabilities into connection transmissibilities before solving the flow problem. Most commercial flow simulators adopt a harmonic average, as exemplified by eq. (26) for one-dimensional flow. This is equivalent to applying a moving average filter to the upscaled permeability distribution, smoothing heterogeneities and possibly causing information loss. Toronyi et al. (1974) show that this two-point average can deviate significantly from the refined grid permeability distribution mean, especially when there is high permeability contrast.

$$T_{i+1/2} = \frac{A_{i+1/2}}{\mu}\left(\frac{L_i/2}{k_i} + \frac{L_{i+1}/2}{k_{i+1}}\right)^{-1} \tag{26}$$

4.2. Permeability and transmissibility upscaling on unstructured grids

Permeability upscaling is significantly more complex for unstructured coarse grids. This happens because the connections between cells can occur in any arbitrary direction, and are not aligned to the main directions of the refined model. Most approaches discussed in literature work with inter-cell transmissibilities instead of cell permeabilities. As discussed in the previous section, this approach

leads to more accurate results. For unstructured grids, it is also more practical, since, if permeability was informed (either directional or as a full tensor), the simulator would have to calculate its projection in each flow direction determined by the cell connections.

Another difficulty arises from the fact that, in general, the coarse unstructured cell borders do not align with refined grid cell borders. Thus, several refined cells will only partially overlap with a specific coarse cell and particular rules must be devised to treat those cases.

Palagi et al. (1994) presents one of the few analytical methods proposed for unstructured grids, and also one of the few not to determine transmissibility. They determines permeability for each side of the connection between cells, so that in the end each cell has one permeability value for each neighbor. A user defined pattern determines the refined grid permeability sampling from the triangle formed by the cell node and the connecting edge. Then, he suggests two alternatives for estimating the permeability for each side. A simple power law averaging is proposed (eq. (27)), but will not take into account how the permeability samples are arranged. The other alternative is a parallel/series equivalence. However, since the line of points inside a triangle cannot be parallel, this must also result in some inaccuracies.

$$k_{ij,i} = \left(\sum_{n}^{n_p} k_n^w / n_p \right)^{1/w} \tag{27}$$

Empower, the ExxonMobil flow simulator capable of ustructured grid simulation, adopts a global flow based method for upscaling (Khan & Dawson, 2004; Usadi et al., 2007). Single phase simulations are performed on the so called computational grid – an unstructured grid with scale close to the refined structured grid – for several boundary conditions. The upscaled permeability for each connection is selected from calculations on the simulation that gives the highest local average pressure gradient. An alternative is to combine more than one simulation, so that the resulting pressure gradient is aligned to the connection.

Prévost et al. (2005) also uses a flow simulation to determine transmissibilities, but with a local refined grid. For each pair of neighboring cells, a rectangular structured local grid containing both cells and a small margin is extracted. Since this grid isn't aligned to the original refined grid, a rotation, with some loss of information, must be performed. Permeameter boundary conditions

are then applied in this volume to determine transmissibility between cells (Figure 25). He also discusses another method that calculates a full permeability tensor and uses a multipoint flow approximation, but that proves to be less robust.

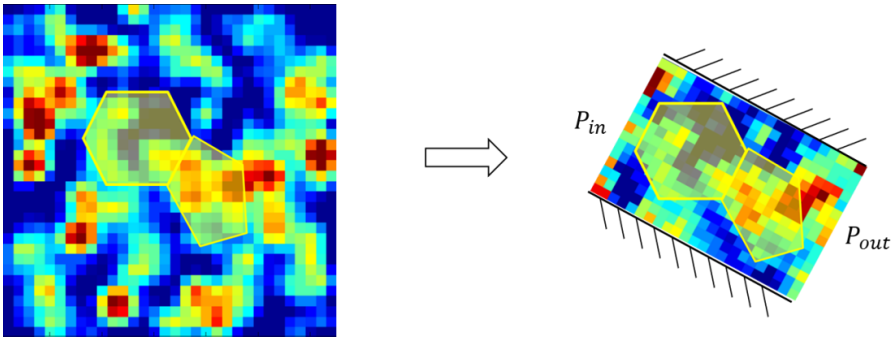

Figure 25 – Prévost unstructured upscaling strategy (adapted from Prévost, 2005)

Mlacnik's (2006) unstructured grid is determined by a single streamline simulation (see previous chapter) of the fine grid, so that the cells are aligned to streamlines and equipotential lines. For neighboring cells that are aligned along the same streamline, he can use the results from this simulation to determine transmissibility. Since there is no or very little flow (on this simulation) between neighboring cells whose connections are parallel to the streamlines, he adopts Prevost's upscaling method in those cases.

Sahimi et al. (2010) use wavelet transformations to determine the upscaled grid points and associated permeability values. This process seeks to minimize the permeability variance within each coarsened cell, so that heterogeneities are better preserved. Those coarse cells are then triangulated with a Delaunay criterion.

4.3. Proposed Cardwell & Parsons algorithm for unstructured grids

For the reasons discussed in the previous sections, it is recommended to determine transmissibilities, or equivalently, inter-cell permeabilities instead of block permeabilities for unstructured grids. This means a higher number of values (around 3 times the number of cells) have to be determined during upscaling and informed with the model. However, the transmissibilities would have to be calculated internally by the simulator anyway, with less accuracy.

As exemplified in last section, using flow simulations is a common solution for upscaling with unstructured grids. However, this implies an additional

computational cost, even if the flow simulations are all single phase. Also, some flow based methods, such as using streamlines, are dependent on the well arrangement, and may become inaccurate if well locations or rates change. There are not many published solutions that avoid simulations, and those available have some limitations.

Therefore, new methods for determining transmissibilties directly from the refined scale permeability field, without resorting to any sort of flow simulation, are discussed. The discussion is limited to upscaling in the horizontal direction, since unstructured grids in 2.5 dimensions are considered. If necessary, the vertical upscaling can be performed beforehand by a method associated with structured grids.

The first proposal is to use a heuristic calculation based on Wiener bounds, since they are unaffected by grid orientation issues. The Wiener bounds can be calculated considering all refined grid points contained in both cells, or only those inside a quadrilateral defined by the cell nodes and connection corners, as illustrated by Figure 26. The blue and green dots represent the refined cell centers contained in either unstructured cell, while the circles represent cell nodes. This domain is similar to that adopted by Palagi; however, a single permeability for both sides of the connection in this work.

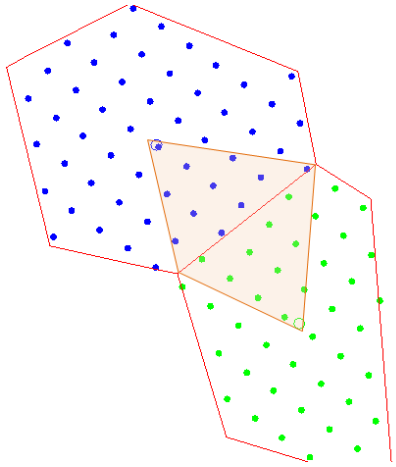

Figure 26 – Cells limits and domain for Wiener upscaling

$$k_{i,j\ wiener} = \sqrt{k^{+}_{i,j\ wiener} k^{-}_{i,j\ wiener}} \tag{28}$$

The flexibility of the Wiener bounds is also their greatest limitation. Since the spacial arrangement of the refined cells is not taken into account, directional effects cannot be reproduced. Any refined cell with null permeability is enough to

eliminate the whole connection while using a geometric mean of the bounds. Giving the same weight to all refined cells also seems to be inaccurate – clearly, the permeability of cells nearest to the coarse grid nodes should have a higher impact on the flow inside the quadrilateral than those near the cell interface, where there is a wide cross-section.

Applying Cardwell & Parsons instead of Wiener bounds addresses those problems, but also results in two main difficulties with unstructured grids. A rectangular domain is necessary to calculate the appropriate series/parallel equivalences, so that the quadrilateral used with Wiener bounds cannot be used in this case. In addition, the resulting permeabilities are components aligned with the refined grid, not with the interface between coarse cells.

The first problem can be solved by defining the domain as the smallest rectangle aligned with the refined grid that contains both coarse cell nodes, and the whole interface between them. An example is presented in Figure 27. The Cardwell & Parsons bounds can then be calculated in the x and y directions of the refined grid. A geometric mean is again applied between upper and lower bounds to obtain k_{xx} and k_{yy}.

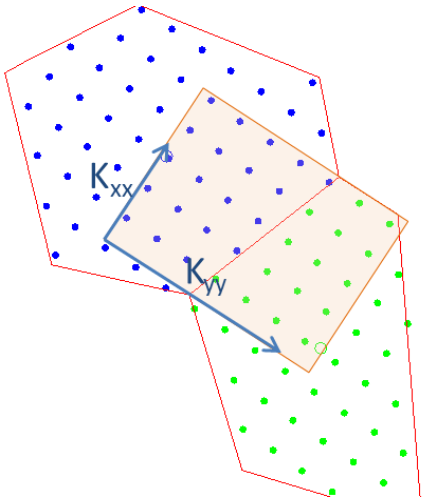

Figure 27 – Cells limits and domain for Cardwell & Parsons upscaling

The volumetric flow equation for the interface between the cells can be developed as shown by eq. (29), by assuming the principal permeability directions are aligned to the refined grid (the permeability tensor is diagonal) and considering a linear pressure gradient $\Delta\Phi_{i,j}$ between cell centers (therefore perpendicular to the cell interface). θ_n is the angle between the refined grid x direction and the interface direction.

$$
\begin{aligned}
q_{i,j} = A_{i,j}\lambda_p\left(\bar{\bar{k}}\nabla\Phi\right).\vec{n} &= \frac{A_{i,j}}{l_{i,j}}\lambda_p\left(\bar{\bar{k}}\Delta\Phi_{i,j}\vec{n}\,\right).\vec{n} \\
&= \frac{A_{i,j}}{l_{i,j}}\lambda_p\left(k_{xx}n_x\hat{\imath} + k_{yy}n_y\hat{\imath}\,\right)\Delta\Phi_{i,j}.\vec{n} \\
&= \frac{A_{i,j}}{l_{i,j}}\lambda_p\left(k_{xx}n_xn_x + k_{yy}n_yn_y\,\right)\Delta\Phi_{i,j} \\
&= \frac{A_{i,j}}{l_{i,j}}\lambda_p\left(k_{xx}\cos^2\theta_n + k_{yy}\sin^2\theta_n\,\right)\Delta\Phi_{i,j}
\end{aligned}
\tag{29}
$$

The inter-cell permeability and transmissibility are therefore given by eq. (30) and (31).

$$k_{i,j} = \left(k_{xx}\cos^2\theta_n + k_{yy}\sin^2\theta_n\,\right) \tag{30}$$

$$T_{i,j} = \frac{A_{i,j}}{l_{i,j}}\lambda_p k_{i,j} = \frac{A_{i,j}}{l_{i,j}}\lambda_p\left(k_{xx}\cos^2\theta_n + k_{yy}\sin^2\theta_n\,\right) \tag{31}$$

The greatest frailty of this method is the definition of the domain for permeability calculation, especially for highly deformed cells. As with other methods implemented for this dissertation, it works better if the original model is much more refined than the coarse model. The expected ratio of cells between models should be at least 10:1.

4.4. Proposed Renormalization algorithm for unstructured grids

The proposed modification on the Cardwell & Parsons model can lead to imprecise results on some situations, particularly for a limited refinement or highly deformed cells. An alternative method based on renormalization was proposed to overcome those limitations. The classic versions of this upscaling method assume a rectangular grid. One way to interpret renormalization is determining the equivalent transmissibility for a network of connections as series/parallel combinations, analogous to an electric network.

This leads to a more flexible application of renormalization, suitable for unstructured grids, since an equivalent electric resistance or conductance can be determined for arbitrary network arrangements, not only for rectangular networks. For instance, the resistance between two nodes can be determined by successive star-triangle transformations (Figure 28).

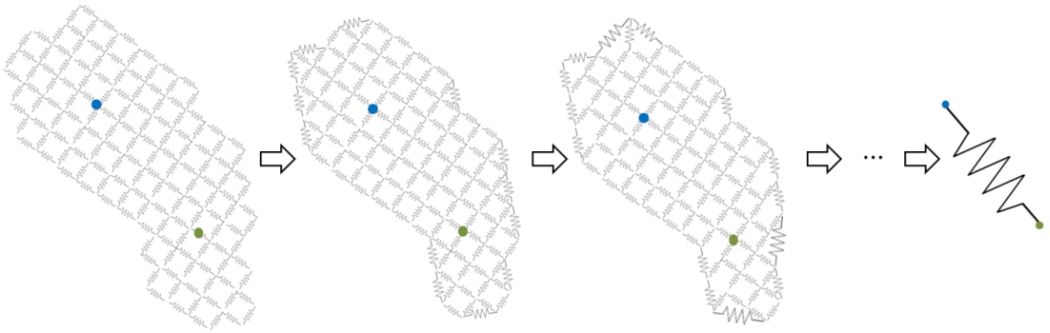

Figure 28 – Iterative renormalization for a pair of unstructured cells

The process of applying successive transformations can become time consuming and hard to determine with large, complex networks. The equivalent resistance or conductance can be determined more efficiently in those cases by applying graph theory. This solution traces back to Kirchhof (1847), but the development presented below is adapted from Wo (1982).

Considering a network of connected resistors, an external current leaving the circuit at a node i equals the sum of the currents entering the node from connected nodes, as given by eq. (32), where x_{ij} is the conductance of a resistor connecting two nodes (or zero otherwise) and V_i the potential at a node.

$$I_i = \sum_{j \neq i} x_{ij}(V_i - Vj) \tag{32}$$

This system of equations can also be represented as:

$$I_i = \sum_{j}^{N} A_{ij} V_j$$

$$A_{ij} = \begin{cases} -x_{ij}, & i \neq j \\ \sum_{k \neq i} x_{ik}\,, & i = j \end{cases} \tag{33}$$

This same system can be represented in matrix form as

$$\bar{I} = AV$$

The equivalent conductance between nodes can be determined by eq. (34), when the current I is input at node k and output at node l.

$$C_{kl} = \frac{I}{V_k - V_l} \tag{34}$$

However, since the determinant of A is null, this system cannot be solved directly. This is solved by fixing the tension at node l as zero, and removing the

associated lines and columns from the equation system (indicated by a superscripted (l) on the matrix).

$$\bar{I}^{(l)} = A^{(l)} V^{(l)}$$

The tension at node k can then be solved by applying Cramer's rule, assuming an input current equal to I at node k, -I at node l and 0 at all other nodes.

$$V_k = \frac{\left|A_k^{(l)}\right|}{|A^{(l)}|} = \frac{I\left|A^{(kl)}\right|}{|A^{(l)}|} \tag{35}$$

Therefore, by substituting the values of V_k and V_l on eq. (34), the equivalent conductance between two nodes can be determined by simply calculating the ratio between two determinants (eq. (36)). This process can be applied to reservoir permeability upscaling by replacing conductivities by transmissibility.

$$C_{kl} = \frac{\left|A^{(l)}\right|}{|A^{(kl)}|} \tag{36}$$

This is faster than solving even a single-phase, steady state flow problem, since the variables do not have to be solved at all nodes and only the transmissibility between two nodes must be determined. Another advantage of this method is that no rotation of the refined model, mandatory for some methods and a potential source of imprecisions, is necessary with this modified renormalization.

The transmissibility between neighboring nodes on the refined grid (equivalent to each resistor on the model) is determined by calculating the transmissibility from the center of the cell to the contact edge for both cells, and then taking the equivalent transmissibility (Figure 29) between nodes. Using a rectangular refined grid greatly simplifies the calculations.

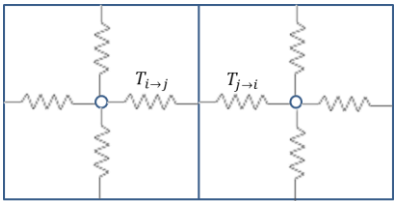

Figure 29 – Equivalent transmissibility between a pair of cells

Different boundary and flow conditions can be adopted with this method, by changing the input and output nodes, and the connected domain considered.

Two sets of conditions were tested for the upscaling problem. Both consider all refined grid cells whose center is inside either coarse cell limits, so that the no-flow boundary is defined by those limits. The first hypothesis takes the

transmissibility between the refined cells closest to the node of the coarse cells (Figure 30). This allows even the permeability of refined cells that are not located between the coarse cell nodes to be taken into account, although with a smaller weight. The second hypothesis is to consider linear flow, by using conditions similar to a permeameter (Figure 31). Instead of a single input or output node, all refined cells lying across a strip perpendicular to flow are considered to be at the same potential (connected with an infinite transmissibility).

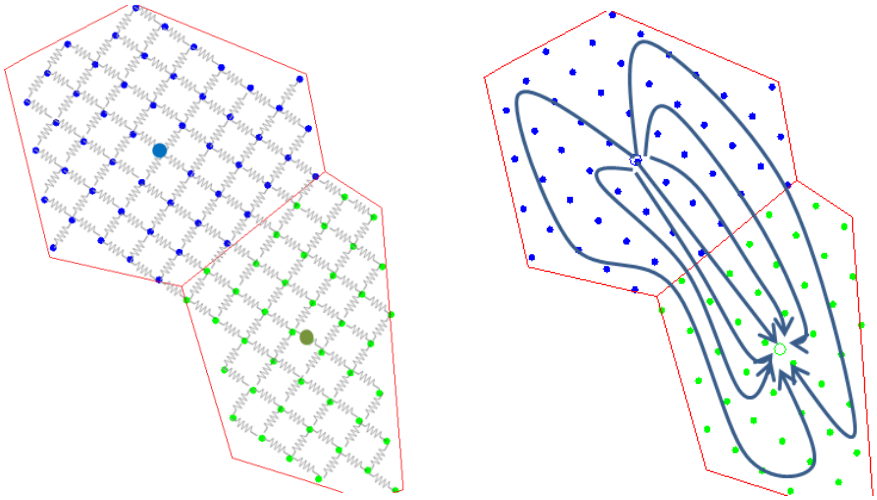

Figure 30 – Equivalent network and schematic flow for fixed pressure on the unstructured cell nodes

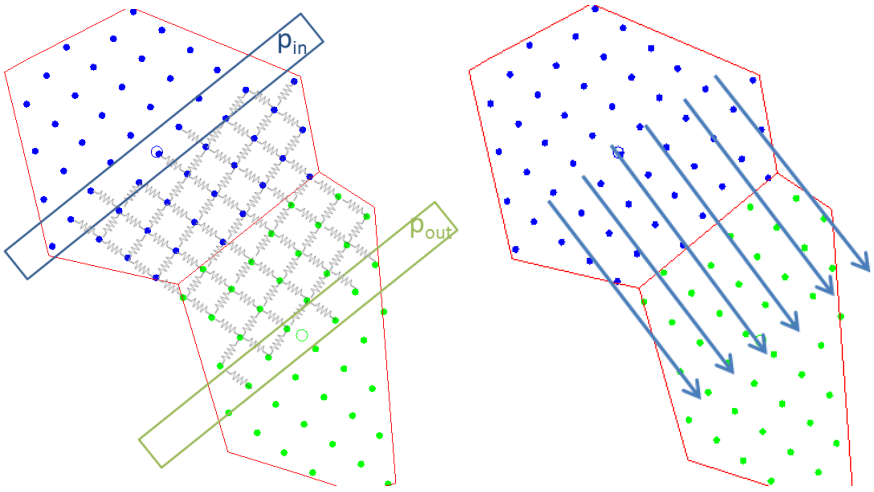

Figure 31 – Equivalent network and schematic flow for permeameter condition

It is not recommended to include a margin around the cells, as done by Prevost, because with this method this might unrealistically increase the transmissibility.

Even when the transmissibilities of the refined model are normalized before calculating the determinants, numerical errors may appear for large networks. In order to minimize those issues, when either determinant is a few orders below the largest floating point number representable in MATLAB, an alternative formula was employed. This requires obtaining the eigenvalues λ_i for both matrices, but since it only applied for small fraction of all cell pairs, the impact in calculation times is small:

$$C_{kl} = \frac{\left|A^{(l)}\right|}{\left|A^{(kl)}\right|} = \frac{\prod_i \lambda_i^{A^{(l)}}}{\prod_j \lambda_j^{A^{(kl)}}} = \frac{e^{\ln \prod_i \lambda_i^{A^{(l)}}}}{e^{\ln \prod_j \lambda_j^{A^{(l)}}}} = e^{\left(\sum_i \ln \lambda_i^{A^{(l)}} - \sum_j \ln \lambda_j^{A^{(kl)}}\right)} \qquad (37)$$

4.5. Initial tests with the proposed methods

Before running flow simulations with unstructured models to evaluate the proposed techniques, some initial tests were done to calibrate the modified Cardwell & Parsons and modified renormalization methods.

Initially, a homogeneous permeability field was considered, in order to be able to compare the proposed upscaling methods with analytical transmissibility values. Rectangular and hexagonal coarse grids with different scales were generated for testing, while the refined grid had 125x125 rectangular cells. The mean deviation from analytical transmissibility values was determined for each upscaling method and cell ratio between refined and coarse grids (Table 3 and Table 4).

Table 3 – Upscaled transmissibility ratios for rectangular coarse grids

Coarse grid cells	Cell Ratio	Normalized transmissibility values		
		Modified Cardwell & Parsons	Modified Renormalization (between nodes)	Modified Renormalization (permeameter)
25	625	100,00%	51,17%	98,08%
144	109	100,00%	59,76%	91,11%
400	39	100,00%	65,92%	86,35%
625	25	100,00%	69,17%	100,00%
900	17	100,00%	71,22%	80,94%
1600	10	100,00%	76,61%	76,52%
3600	4	100,00%	77,62%	67,97%
15625	1	100,00%	100,00%	100,00%

Table 4 – Upscaled transmissibility ratios for hexagonal coarse grids

Coarse grid cells	Cell Ratio	Normalized transmissibility values		
		Modified Cardwell & Parsons	Modified Renormalization (between nodes)	Modified Renormalization (permeameter)
123	127	100,00%	84,00%	121,50%
387	40	100,00%	91,30%	113,60%
624	25	100,00%	93,70%	108,90%
921	17	100,00%	95,15%	104,05%
1517	10	100,00%	97,65%	96,87%
2912	5	100,00%	95,15%	84,49%

The modified Cardwell & Parsons method returns the same transmissibility as an analytical calculation for all scales when permeability is homogeneous. This happens because in this case $k_{xx} = k_{yy} = k$, and the other variables involved (distance between nodes and interface area) are the same as for analytical calculation. This does not mean that this method also yields better results than others for heterogeneous models.

Renormalization considering the flow between coarse cell nodes and considering permeameter conditions (linear flow) show opposite tendencies with cell ratio variation (Figure 32). For permeameter conditions, results are closer to the expected values when the cell ratio between grids is high, and transmissibility

decreases significantly when this ratio becomes small. However, this method does determine the exact analytical permeability when there is a perfect overlap of cell boundaries with the refined grid (only possible with a rectangular coarse grid), as illustrated on Figure 33.

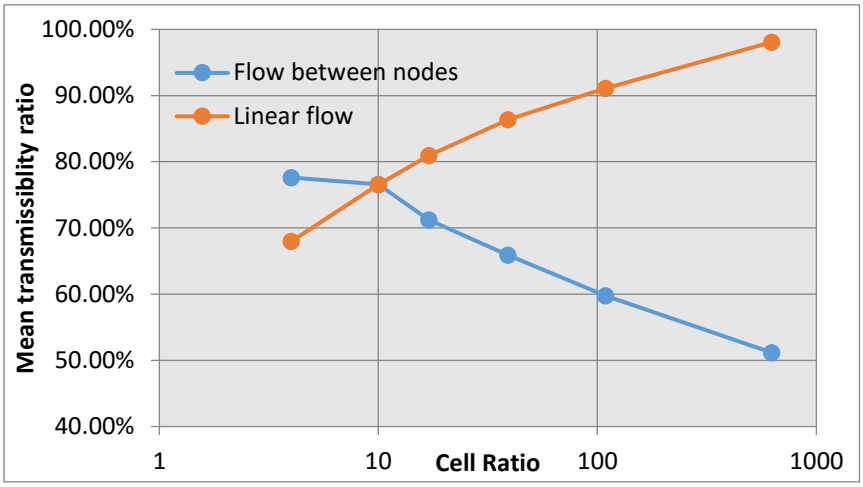

Figure 32 – Upscaled renormalization transmissibility ratios for rectangular coarse grids

The general tendency for this method can be explained by imperfect overlap (Figure 34) between grids. The algorithm takes into account all refined cells whose center are inside the coarse cell limits, even though several cells are split along the border because of imperfect overlap. Thus, a refined cell whose center is inside the coarse cell boundary will be taken into account on transmissibility calculations, even if most of its volume is not, while another may be disregarded even if nearly half its volume is inside. This effect becomes more significant when the cell ratio between models is small (high granularity).

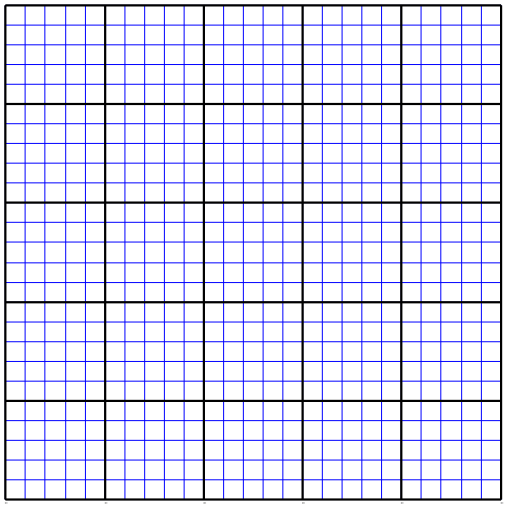

Figure 33 – Coarse grid perfectly overlapping the refined grid

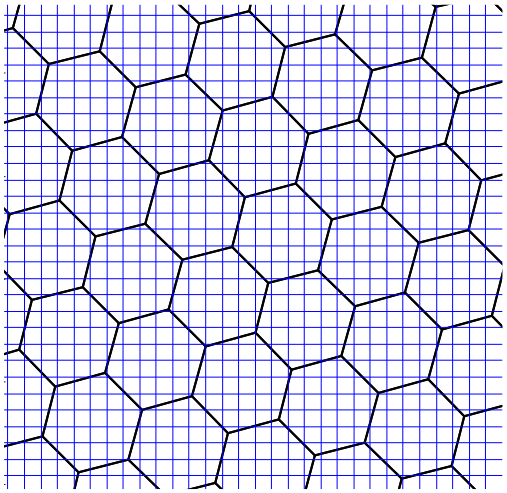

Figure 34 – Coarse grid that does not overlap the refined grid

This effect is unnoticeable when flow between cell nodes is considered for renormalization. This probably happens because for this condition the flow along the borders has little impact on the total transmissibility, since the fluid must take a longer path than the straight line connecting nodes. In this case, a small cell ratio actually improves the results, possibly because less flow paths are possible, and thus the shortest path (closest to the correct value) has a higher weight on the calculations.

With a hexagonal grid, calculated transmissibilities become higher in all cases. This effect can probably be explained by an increase in the considered interface area between cells because of the angle between coarse cell interfaces and the refined grid (Figure 35). For a variable angle θ, this expected increase in the interface area can be estimated by eq. (38).

$$E[A_\theta] = E[A_0(|\cos\theta| + |\sin\theta|)]$$
$$= {}^{A_0}/_{2\pi} \int_{-\pi}^{\pi} (|\cos\theta| + |\sin\theta|)d\theta = {}^{4A_0}/_{\pi} \cong 1.273L_0 \quad (38)$$

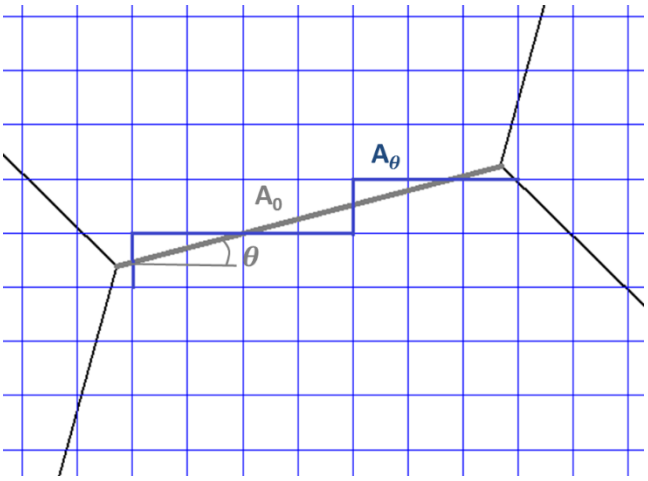

Figure 35 – Coarse cell interface and approximation from refined cells

Dividing the transmissibilities obtained for renormalization with a linear flow condition by the factor above resulted in a behavior close to that observed for the rectangular grid (Figure 36).

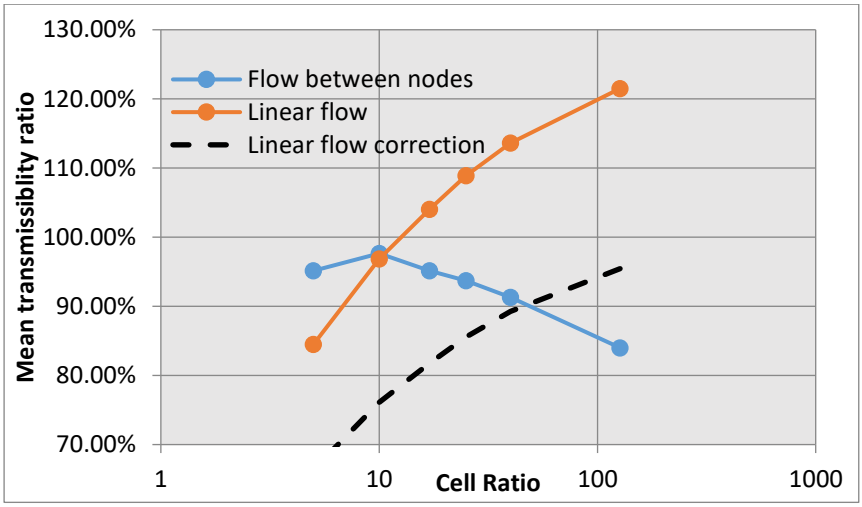

Figure 36 – Upscaled renormalization transmissibility ratios for hexagonal coarse grids

Comparisons of upscaling results for heterogeneous models can provide further insight into the proposed methods. However, since analytical calculations are not possible, there is no reference value for comparison in this case. In chapter 5, refined grid flow simulations will be used as reference. For now, we will only compare the methods between each other.

A more complex reservoir model with heterogeneous porosity and permeability was used for those comparisons (Figure 37). A mostly hexagonal unstructured grid with around 2000 cells was generated (Figure 38) using the techniques discussed on last chapter. Table 5 shows statistics for the upscaled transmissibilities with different methods.

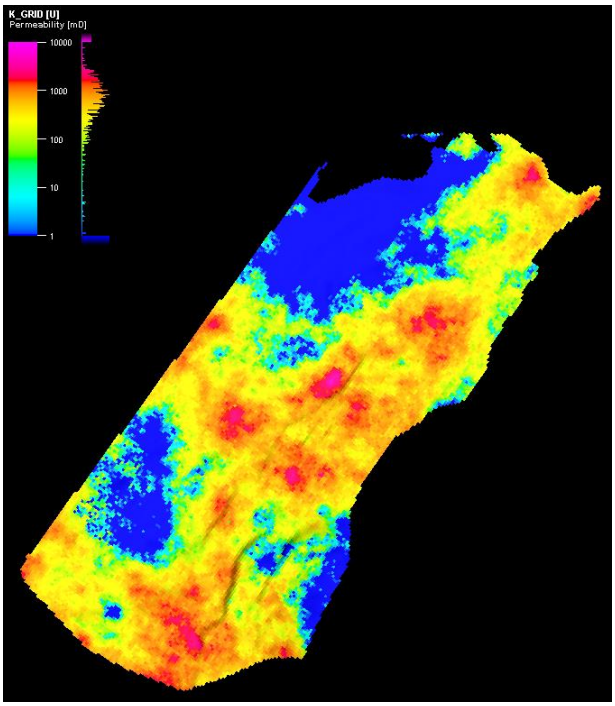

Figure 37 – Heterogeneous model permeability field

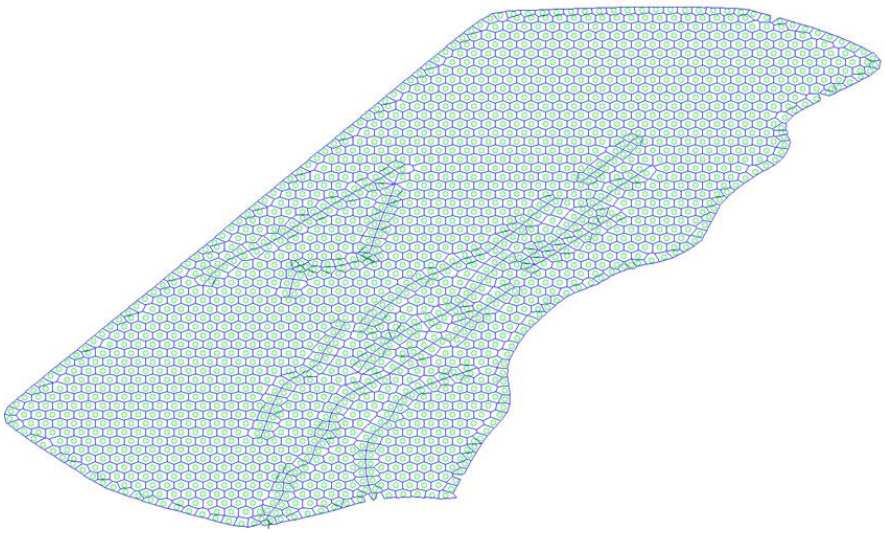

Figure 38 – Heterogeneous model unstructured grid

For the considered permeability distribution, without anisotropy or structured barriers, the results for Wiener and for Cardwell and Parsons are very similar.

With renormalization, less non-null transmissibilities are calculated, because when the edge between neighboring cells is small (near the refined cell dimensions), there may be no path to make the connection. This can be advantageous from a computational point of view, since the transmissibility matrix is sparser. However, this could also be achieved for other methods by post processing the results.

Considering flow from a central node results in smaller transmissibility than a permeameter condition, because the mean path that the fluid must pass is longer and flow barriers near the nodes have higher impact. Comparisons with other methods suggest that linear flow condition is nearer the expected results.

Table 5 – Upscaled transmissibilities for a complex reservoir model

		Transmissibility (mD.m)		
Upscaling Method	**Active connections**	**Minimum (non-null)**	**Mean**	**Maximum**
Modified Wiener	83.0%	4.76E-10	3740	71240
Modified Cardwell & Parsons	90.7%	5.24E-10	3820	70840
Renormalization (flow between nodes)	77.9%	1.79	2810	33200
Renormalization (linear flow)	79.1%	1.87	3990	66100

4.6. Equivalent radius Calculations

Well bottom-hole pressures must be correlated to the pressures of the blocks where they are located. This is usually achieved by defining an equivalent radius r_o, given by the distance from the well at which the pressure is equivalent to the simulation well block pressure (Peaceman, 1978). For an anisotropic permeability field and rectangular cell, the equivalent radius is given by (Peaceman, 1982):

$$r_o = 0.28 \frac{\sqrt{\sqrt{k_y/k_x}\Delta x^2 + \sqrt{k_x/k_y}\Delta y^2}}{\left(k_y/k_x\right)^{1/4} + \left(k_x/k_y\right)^{1/4}} \tag{39}$$

Since this formulation is based on cell dimensions and directional permeabilities, it cannot be applied for an unstructured grid model. Also, it is only valid for a well at the center of a cell and distant from the reservoir boundaries. Abou-Kassem et al. (1980) propose a generalization for irregular cells that applies for unstructured grids, based on the assumption of radial flow. It is presented here in the simpler form described by Palagi et al. (1994), without the inclusion of image wells that become necessary near boundaries:

$$r_o = \left[\exp\left(-f\overline{kh}\right)\prod_{j=1}^{n_i} {d_{ij}}^{T_{ij}}\right]^{1/\sum_{j=1}^{n_i} T_{ij}} \quad (40)$$

The values obtained by eq. (39) were compared with eq. (40) for a homogeneous model with a Cartesian grid. The results were effectively identical, with the exception of wells located on boundary cells. It is not possible to directly compare the values for real unstructured grids, since eq. (39) cannot be computed for those cases.

With heterogeneous models, the $\overline{kh}$ product from the formula above should be representative for the whole cell. Among the upscaling methods proposed in this chapter, a Wiener method is the only appropriate alternative, since $\overline{kh}$ should not assume a specific flow direction.

5
Flow Simulation and test results

In this chapter, flow simulations with simple models generated using the techniques presented in chapters 3 and 4 are performed. In section 5.1, the GSim flow simulator and its adaption for unstructured grids are discussed. The reservoir model used for the experiments is described in section 5.2, and after that the structured grid used as reference. Sections 5.4 to 5.7 present comparative results for scenarios with different reservoir properties.

Many figures in this chapter adopt a common display convention. In order to avoid repetitions with every new results figure, some initial clarifications should be made. On production time series graphs, green is adopted to indicate structured grid results, with continuous lines for the refined grid and dashed lines for the medium grid. Red (or orange) is used for unstructured grid results with renormalization, while blue indicates Cardwell & Parsons upscaling. Darker tones of those colors represent oil production curves, while lighter tones stand for water production.

Pressure and saturation maps are colored according to property values from unstructured grid models. The contour lines represent pressure and saturation from the refined structured grid for the same scenario, and adopt the same color scale. The maps represent fluid distribution after one year of production. Exceptions to those conventions will be noted along the relevant figures.

5.1.
Unstructured grid flow simulation with GSim

GSim is a reservoir multiphase simulator being developed at Pontifical Catholic University of Rio de Janeiro (PUC-Rio) in a joint project with Petrobras. The simulator was originally conceived for simulation with structured grids. However, since it adopts a plug-in approach, the modifications for unstructured grid simulation can be inserted by replacing just the relevant modules with a setup script. Duarte et al. (2015) discussed the advantages of a simulator based on such a configuration.

For a Voronoi / PEBI unstructured grid model (flow between cells is orthogonal to cell limits), the two point CVFD formulation used by GSim simulator is still valid. Thus, only minor modifications to the code are necessary to run the unstructured model. The main modifications are related to data input and output, and a few others to the cell or element model.

Geometric transmissibilities and well index are calculated directly from the refined grid for the unstructured models as part of the upscaling process, and therefore must be informed to GSim as part of the input model. Only fluid mobility factors related to transmissibilities and well productivity are calculated internally by GSim for the unstructured model.

The cell / element model had to altered, since the structured version of GSim assumes hexahedral cells, determined by the coordinates of their corners (known as corner point in reservoir simulation). Fortunately, GSim inherits support for other sorts of elements from the generic simulator it is built upon (Celes et al., 2005), so that only some functions specific for reservoir simulation, such as cell volume calculations, had to be adapted for the unstructured element. For the proposed 2.5D unstructured topology, each element is determined by an arbitrary number of corners and a thickness value.

The latest versions of GSim are capable of reading an input file similar to other reservoir simulators, which assumes corner point cells. For unstructured simulation, a generic file format supported by GSim called Neutral File (Menezes et al., 2002), which supports several element types, had to be adopted. The disadvantage was that several reservoir properties reading functions (already available on the other reader) had to be reimplemented.

5.2. Reservoir Model

The chosen reservoir model has a simple hexahedral shape, with 250m in both areal directions and 10m thickness, located at a 3000m depth. The water-oil contact was considered to be below the reservoir, so that, initially, the only water in the reservoir is connate water. A producer and a water injecting vertical wells were placed at opposite corners of the reservoir, at (25,25) and (225,225) coordinates. The wells were set to operate for four years with fixed bottom-hole pressures at,

respectively, 150 kgf/cm^2 and 250 kgf/cm^2. The maximum simulation time step was set to one day.

Different reservoir properties were considered for each test scenario, as listed in Table 6 and illustrated by Figure 39. Those scenarios will be described in more detail on the appropriate sections.

Table 6 – Reservoir scenarios summary

Scenario	Description	Minimum Φ	Maximum Φ	Minimum k	Maximum k
A	Homogeneous	25%	25%	100 mD	100 mD
B	Thick channels	25%	25%	50 mD	500 mD
C	Amalgamated channels	20%	31%	50 mD	500 mD
D	Sealing faults	25%	25%	100 mD	100 mD
E	Aligned channels	25%	25%	10 mD	500 mD

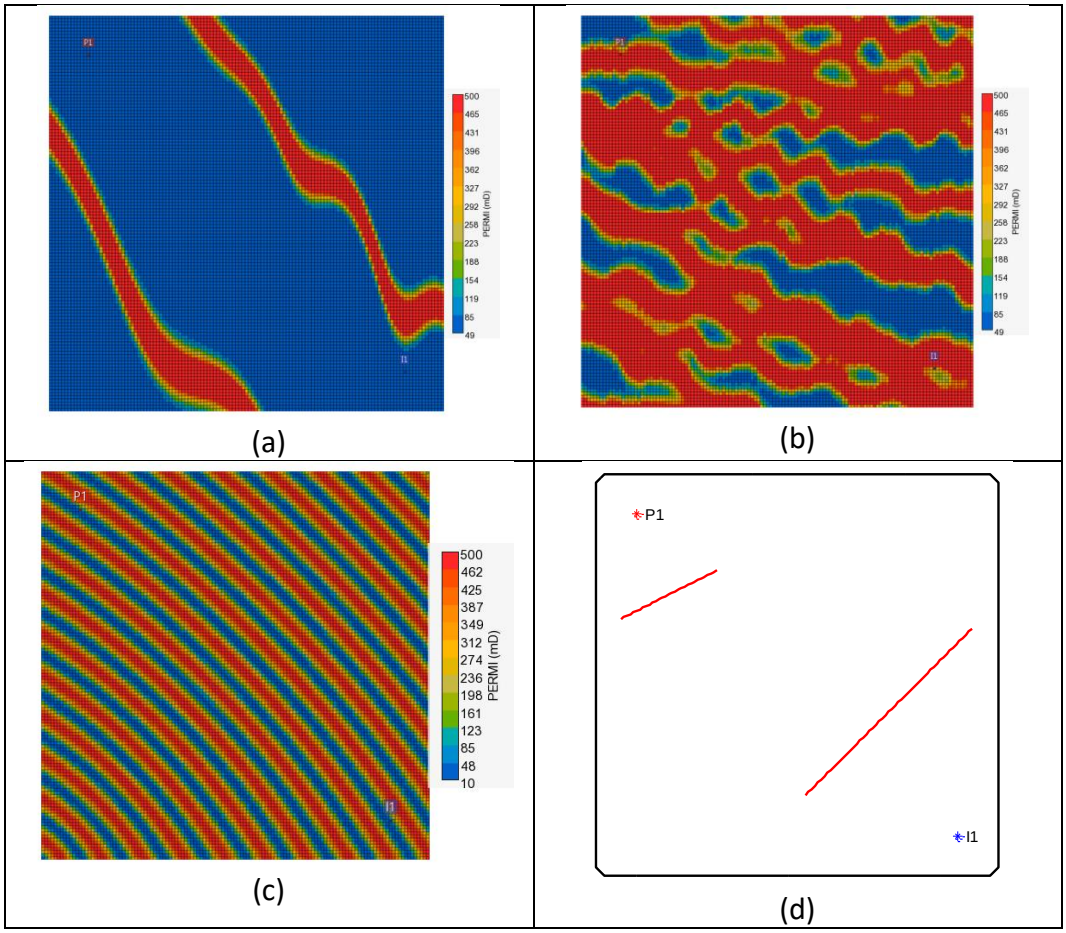

Figure 39 – Heterogeneous models: (a) permeability for thick channels; (b) permeability for amalgamated channels; (c) permeability for thin aligned channels; and (d) sealing faults positions (in red).

A single relative permeability set and a single black-oil fluid model are used for all models. Since the relative permeability is higher for oil than water and oil viscosity at reservoir conditions is slightly over 1 cp, the mobility ratio is favorable to oil displacement.

5.3. Structured Models

Structured grid models were generated for all scenarios to serve as comparison for the unstructured models. These models were simulated with the commercial reservoir flow simulator IMEX. Three levels of horizontal refinement were considered (Table 7), always with a single 10m vertical layer.

Table 7 – Structured grids summary

Grid	Areal granularity	Cell dimensions	Total number of cells
Coarse structured	5 x 5	50 m x 50 m	25
Medium structured	25 x 25	10 m x 10 m	625
Refined structured	125 x 125	2 m x 2 m	15625

The refined structured grid is considered as equivalent to a geological model without upscaling, and used as input to build all unstructured models. The results obtained with this grid for each scenario are considered as the "correct" values, and used as a reference to evaluate the results from all other grids.

The comparison of simulation results for Scenario A with structured grids (Figure 40) show that there is significant degradation of the production curves with the coarse grid, even for homogeneous permeability. For this scenario, the medium resolution grid has very good results, close to those of the refined resolution. As will be shown further on, those results degrade somewhat with heterogeneous models. The proposed objective is to achieve results as near as possible to those of the refined grid with unstructured grids, while having a number of cells similar or smaller than the medium structured grid. Therefore, the coarse structured model will not be even taken into account on the following comparisons.

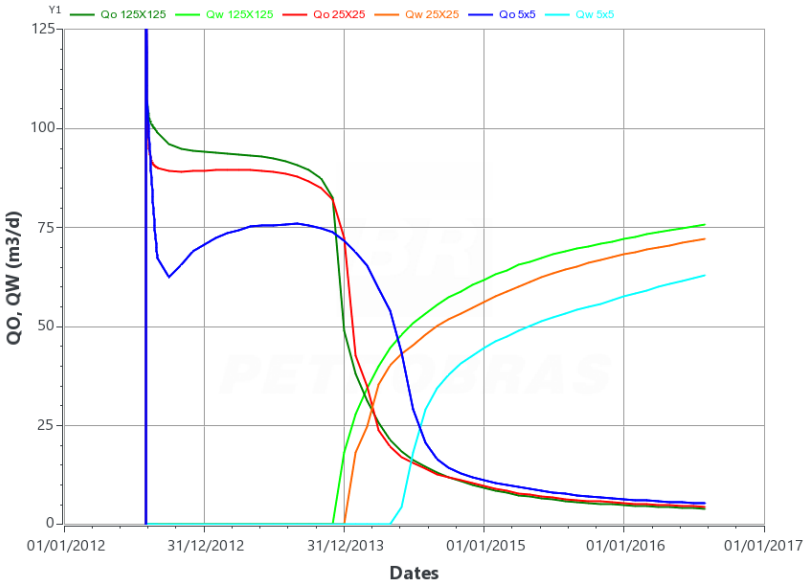

Figure 40 – Production curves for scenario A with structured grids: coarse (blue), medium (red) and refined (green).

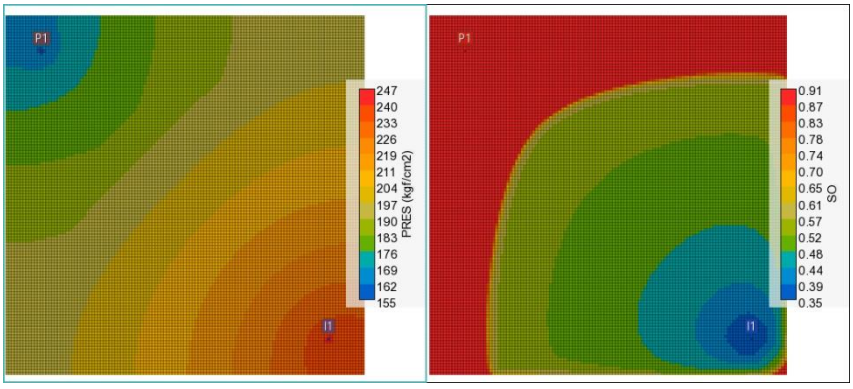

Figure 41 – Pressure and saturation maps for the refined structured grid, scenario A.

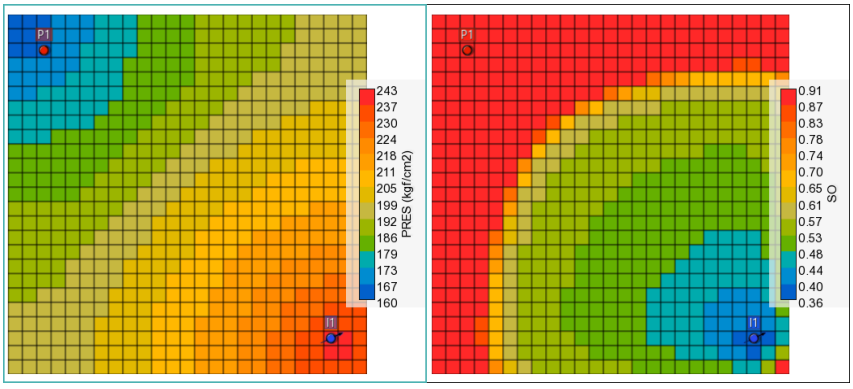

Figure 42 – Pressure and saturation maps for the medium refined structured grid, scenario A.

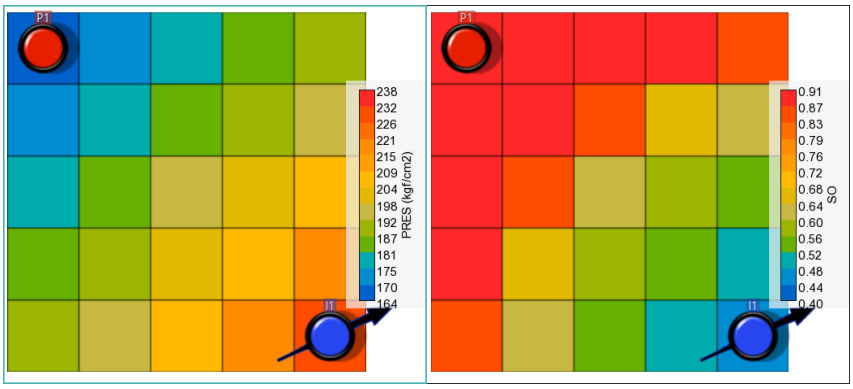

Figure 43 – Pressure and saturation maps for the coarse structured grid, scenario A.

5.4. Scenario A

5.4.1. Rectangular unstructured grid

An unstructured model with a grid geometrically identical to the medium structured grid was generated. Thus, the modifications made to GSim simulation for unstructured grids can be validated while isolated from the effects of grid geometry and upscaling algorithms.

Transmissibilities between cells for this case can be calculated analytically by a simple geometrical formula (eq. (41)), since the cells have regular shape and permeability is homogenous. The well indexes are calculated by using the equivalent radius for unstructured cells presented on the previous chapter (eq. (40)). The resulting values are practically identical to those obtained by using Peaceman's equivalent radius, except for cells on the reservoir boundary. This difference happens because image wells are disregarded on the adopted formulation.

$$T_{i,j} = A_{i,j}\lambda_p k_{i,j} = \frac{A_{i,j}\lambda_p k}{l_{i,j}} \tag{41}$$

The simulation results show an almost perfect reproduction of structured simulation results, as seen on Figure 44. The visual differences seen near breakthrough can be attributed to recorded time steps. While GSim records all simulated time steps, IMEX was set to record a single sample per month after the first, resulting in an interpolation on the graph. This result shows that GSim

simulation with unstructured grids is performing as expected, and the differences on the following comparisons can be attributed to grid geometry and upscaling.

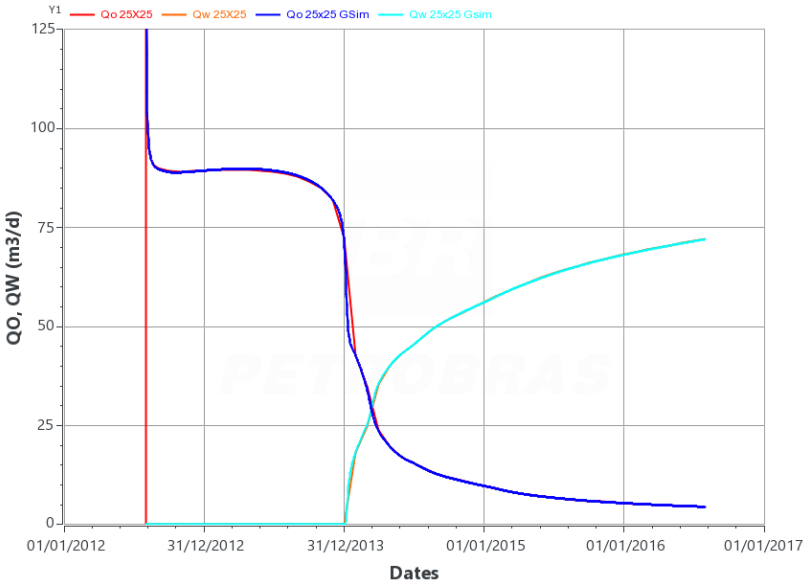

Figure 44 – Production curves for scenario A: medium structured grid with commercial simulator (red), unstructured rectangular grid with GSim (blue).

5.4.2. Hexagonal grid

A nearly hexagonal grid (Figure 45) was built to run further simulations and test the upscaling methods and GSim's performance on a model with more connections between cells. This grid deviates from the hexagonal structure near the borders and where the wells are located. The grid spacing was set to have a number of cells as close as possible to the medium structured grid (624 against 625). This same grid will be used as initial unstructured grid for testing other scenarios.

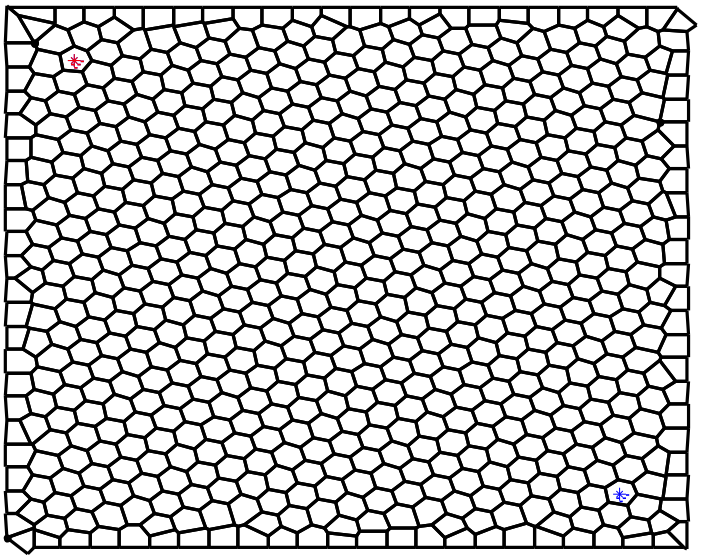

Figure 45 – Nearly hexagonal unstructured grid.

Since this scenario has homogeneous permeability, the modified Cardwell and Parsons method yields the same results as the analytical calculation used before (eq. (41), and thus can be considered the “reference” distribution value. The modified renormalization method, however, is affected by the grid mismatch effects discussed last chapter.

The simulation results show that the modified Cardwell and Parsons method nearly matches the medium structured model (Figure 46). The renormalization method is actually closer to the refined structured model. However, this is probably due to a slight overestimation of transmissibilities rather than better representing the refined model. By observing the pressure and saturation maps, we can see a good match for both cases, with the water front advancing slightly slower for Cardwell & Parsons and slightly faster for renormalization, when compared to the refined model (Figure 47 and Figure 48).

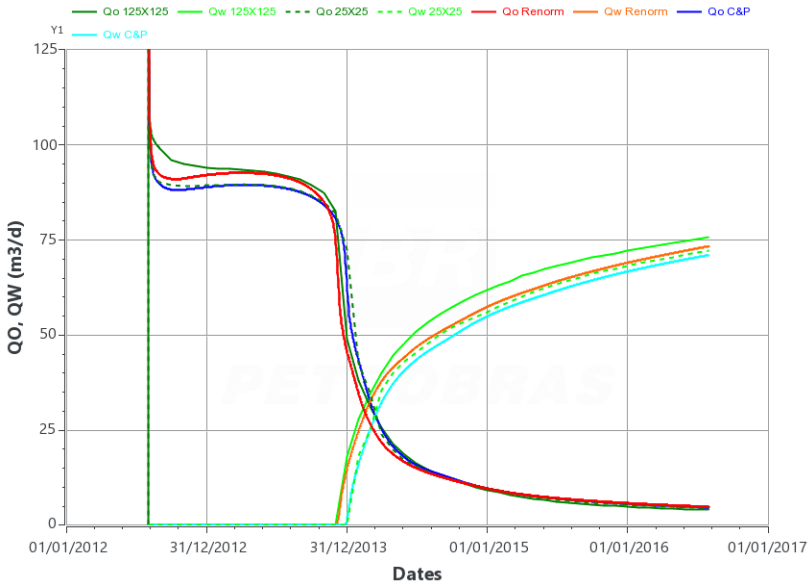

Figure 46 – Production curves for scenario A with nearly hexagonal grid.

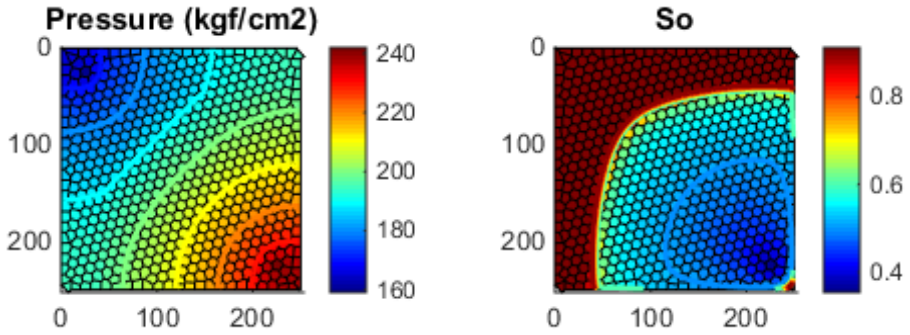

Figure 47 – Pressure and saturation maps for Cardwell & Parsons upscaling, nearly hexagonal grid, scenario A.

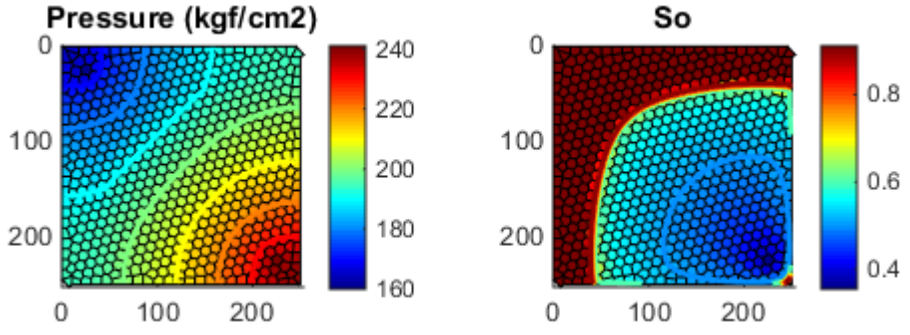

Figure 48 – Pressure and saturation maps for renormalization upscaling, nearly hexagonal grid, scenario A.

5.4.3.
Grid refinement near the wells

Additional grids were built to represent the same problem accurately with fewer cells, by having a variable cell dimension. This has the advantage of decreasing computational cost. Since this scenario is homogeneous, the only feature available to determine the spacing distribution is the location of the wells. Three grids were generated this way, by using spacing maps based on eq. (15) from section 3.4.2. In all cases, the influence of the wells was set to 200m, so that almost the entire grid was affected and the hexagonal pattern lost. Electrostatic repulsion iterations combined with space filling were used to populate the grid.

Some boundary cells have a deformed shape, due to node redistribution and the sparser density in some areas. The gridding algorithms could be improved to prevent this issue. However, the impact in the simulation results are neglectable.

The electrostatic repulsion algorithm also shows some issues. Although the regions near wells are clearly more refined, cell dimensions do not correspond to the specified spacing map. This effect is evident near the external boundaries of the model, where the spacing is enforced, resulting in cells with high aspect ratio. This happens because the spacing map is used only as a weight to repulsion with this algorithm, so that the initial density of cells and equilibrium of forces is what actually determines spacing.

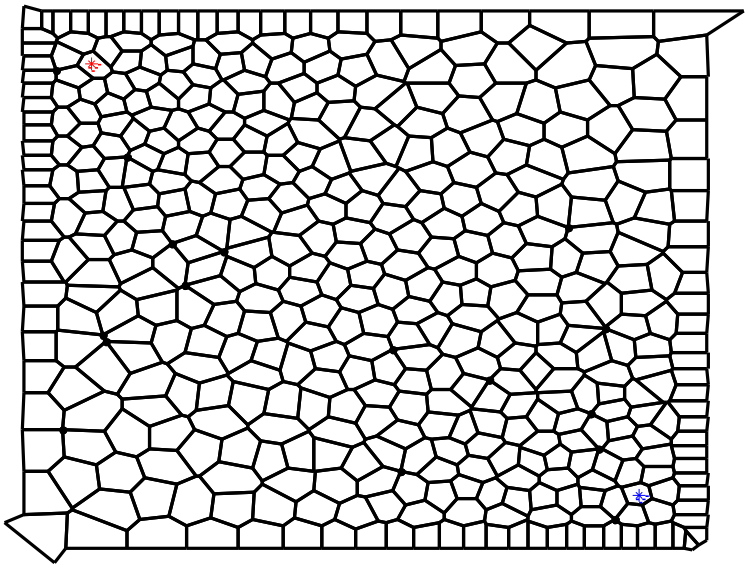

Figure 49 – Unstructured grid with well refinement (441 cells) for scenario A.

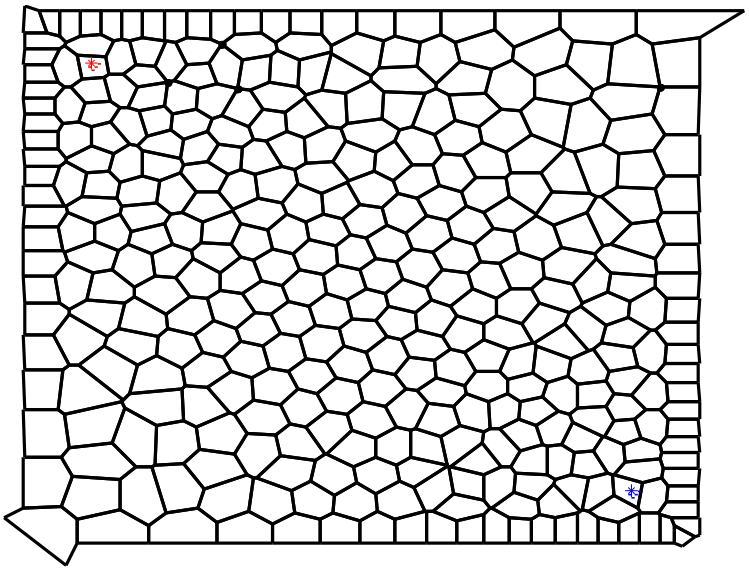

Figure 50 – Unstructured grid with well refinement (337 cells) for scenario A.

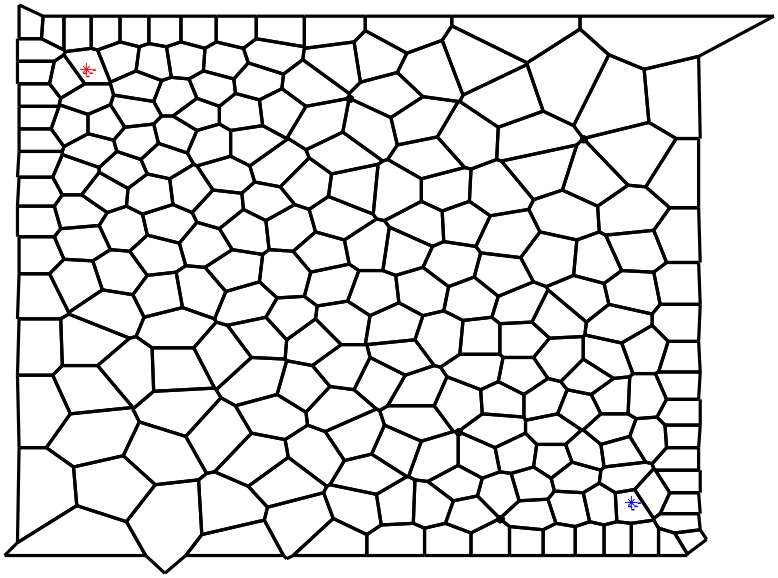

Figure 51 – Unstructured grid with well refinement (221 cells) for scenario A.

The simulation results with those grids (Figure 52, Figure 53 and Figure 54) do not deviate significantly from the medium structured grid, even though the coarsest of the three has only 35% of the cells.

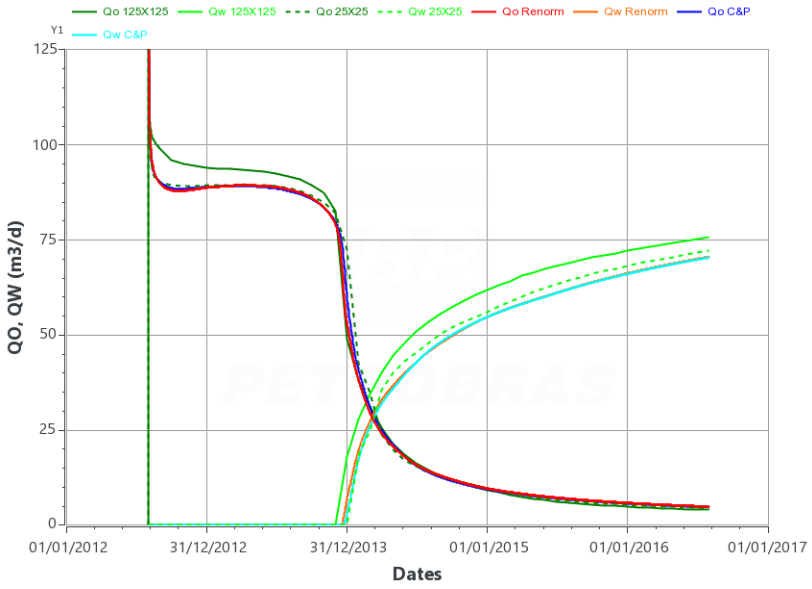

Figure 52 – Production curves for scenario A with unstructured grid with well refinement (441 cells).

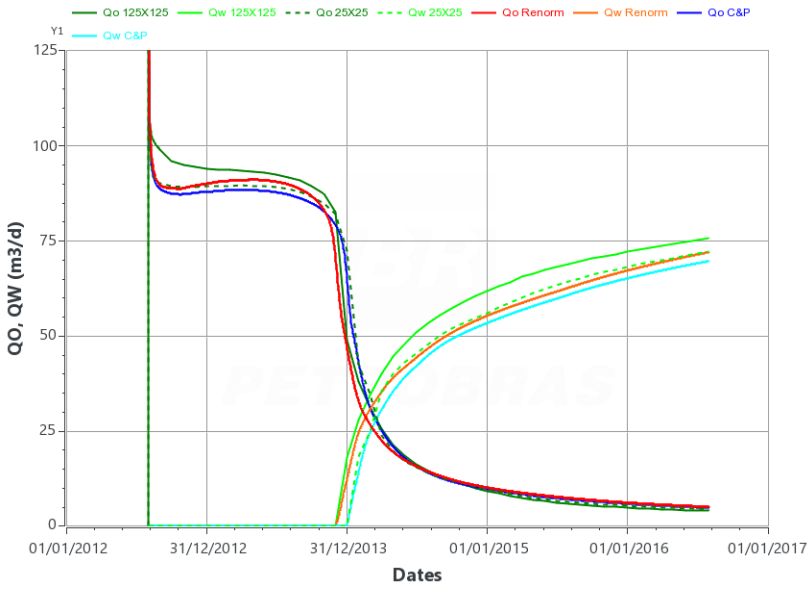

Figure 53 – Production curves for scenario A with unstructured grid with well refinement (337 cells).

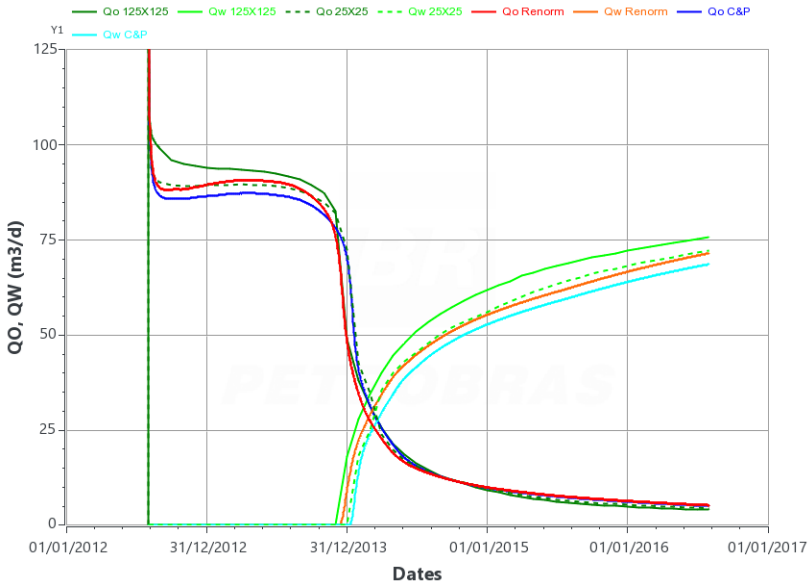

Figure 54 – Production curves for scenario A with unstructured grid with well refinement (221 cells).

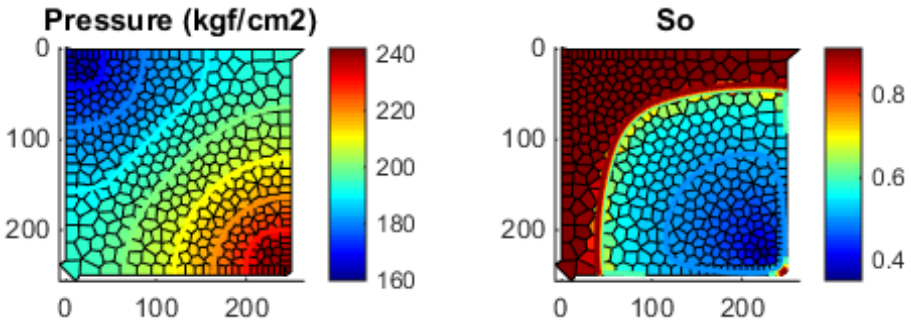

Figure 55 – Pressure and saturation maps for renormalization upscaling, grid refined near wells (441 cells), scenario A.

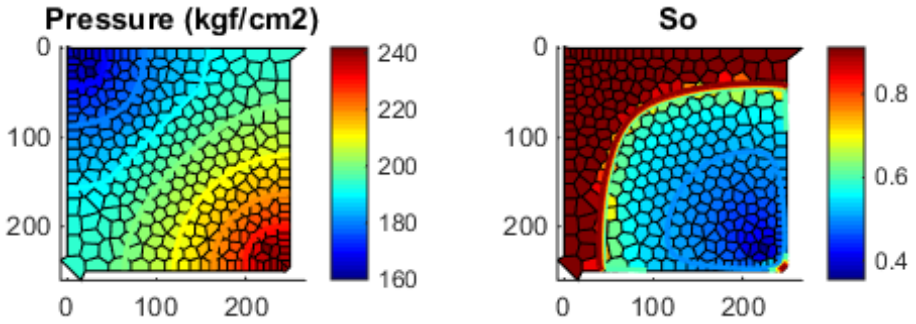

Figure 56 – Pressure and saturation maps for renormalization upscaling, grid refined near wells (337 cells), scenario A.

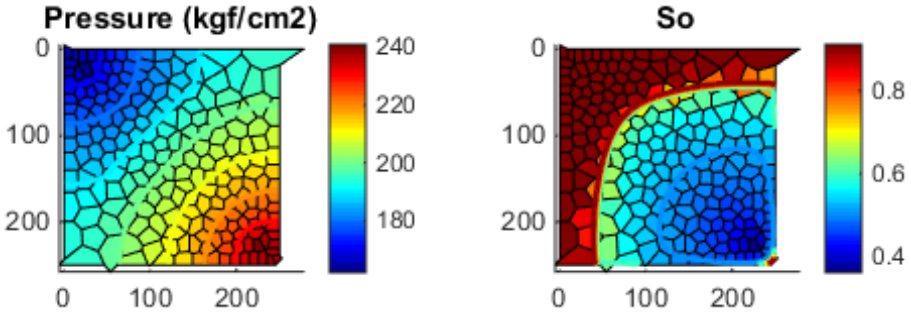

Figure 57 – Pressure and saturation maps for renormalization upscaling, grid refined near wells (221 cells), scenario A.

5.4.4. Computational cost comparisons

The simulation time needed to run several models with unstructured grids for scenario A in GSim is presented in Table 8, normalized by the time needed to run the rectangular grid. Some variation due to random effects is expected, since all values correspond to a single simulation on the same computer. All simulations were run with a fixed time step of one day throughout the simulation. The only exception is the refined rectangular model, whose time steps had to be set to one tenth of a day to guarantee solver convergence with such a small cell size. Because of this, the refined model was excluded from the computational cost comparisons.

The models with a rectangular grid run faster than other models with a similar number of cells. Those models were specified in an unstructured format, but are equivalent to structured grids. Each (interior) cell has only four connected neighbors, while the other models have cells with 6 neighbors in average. Thus, their jacobian matrix have less non-null values, and the numerical solver runs faster for the same grid size.

The advantage of the unstructured grids is to be able to represent the same problem with fewer cells. For the evaluated models, a reduction to less than half the total number of cells of the structured models in necessary to have a reduction in running time. This shows the importance of having an effective grid generation for unstructured grids.

Roughly 84% of the total time necessary to run the simulations is used to build the Jacobian matrix. This is justified by the relatively small size of the tested grids.

Table 8 – Total simulation time for scenario A

Grid type	# of cells	Total time (normalized)	Jacobian matrix setup (% of total)
Rectangular	25	1,9%	56,9%
Rectangular	625	100%	81,9%
Hexagonal	624	138,8%	84,0%
Refined near wells	441	113,7%	85,8%
Refined near wells	337	97,4%	85,7%
Refined near wells	270	47,6%	84,2%
Refined near wells	221	37,9%	83,9%
Refined near wells	196	26,5%	82,3%

5.5. Scenario B

Most reservoir models are not homogeneous. Thus, it is essential to evaluate the impact of permeability heterogeneity when using unstructured grids. Proper upscaling algorithms become important to ensure consistent results, since transmissibility can no longer be determined analytically.

Two permeability channels were placed along the reservoir in Scenario B, increasing permeability up to 500 mD in the center of the channel (Figure 58). In order to avoid increasing production rates too much, background permeability was decreased to 50 mD. For the medium structured grid, permeability was determined by a structured version of the Cardwell & Parsons algorithm, so that it yields slightly different results in the x and y directions (Figure 59). The same strategy will be adopted for the other structured heterogeneous models on the following sections.

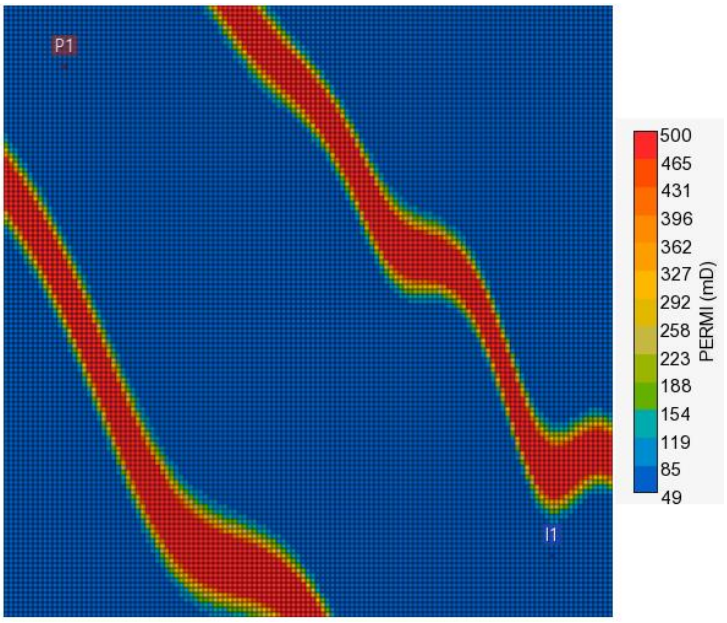

Figure 58 – Permeability map for scenario B with refined structured grid.

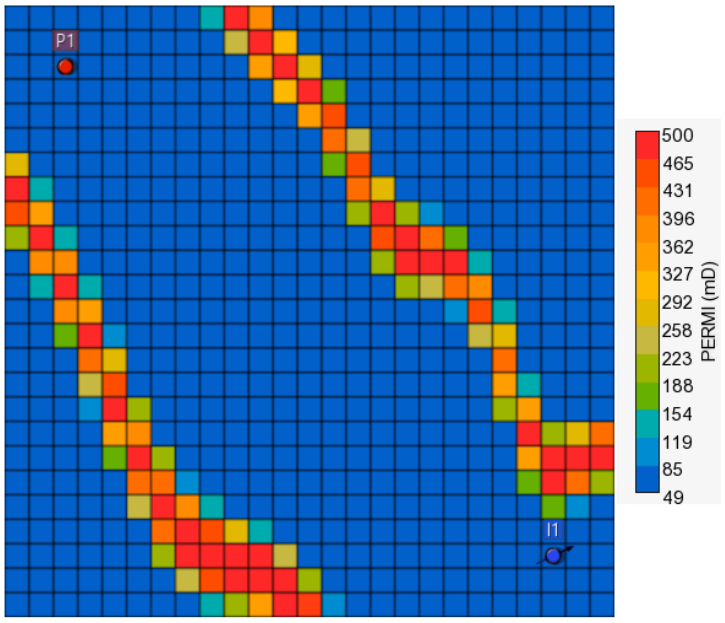

Figure 59 – Upscaled permeability map in the x direction for scenario B with medium structured grid.

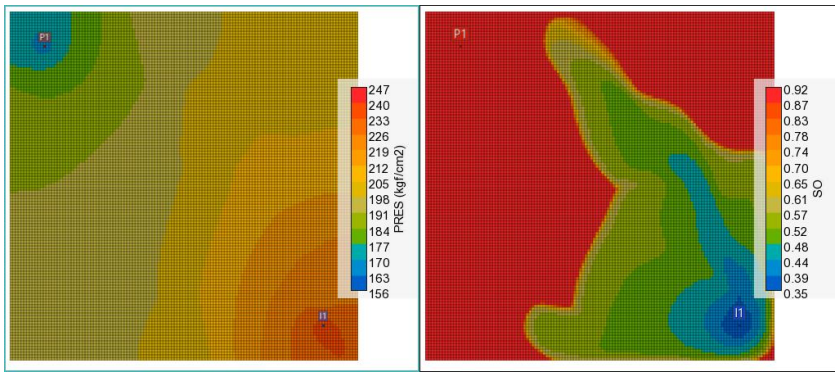

Figure 60 - Pressure and saturation maps for the refined structured grid, scenario B.

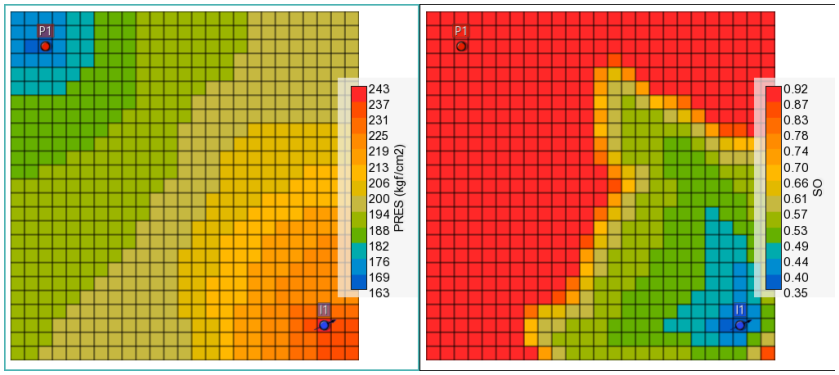

Figure 61 - Pressure and saturation maps for the medium refined structured grid, scenario B.

5.5.1. Hexagonal grid

The same hexagonal grid from section 5.4.2 was used for the initial tests. The simulation results for this scenario (Figure 62) show a higher deviation between production on the refined structured grid and the medium structured grid, with a delay of about six month on water breakthrough. The production curves for the models with the hexagonal grid (using both upscaling methods) stays between those two limits for the whole simulation, and therefore can be considered better than those of the medium structured model. Although this might be related to the upscaling algorithms employed, it probably happens because the hexagonal topology offers more directions for flow to happen between cells, thus emulating the flow in the direction of the channels better than a Cartesian model with the same number of cells can.

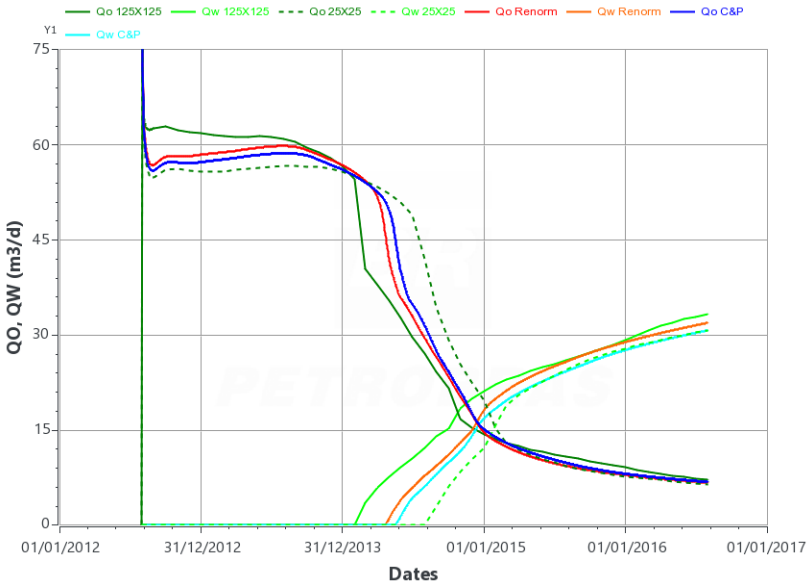

Figure 62 – Production curves for scenario B with nearly hexagonal grid.

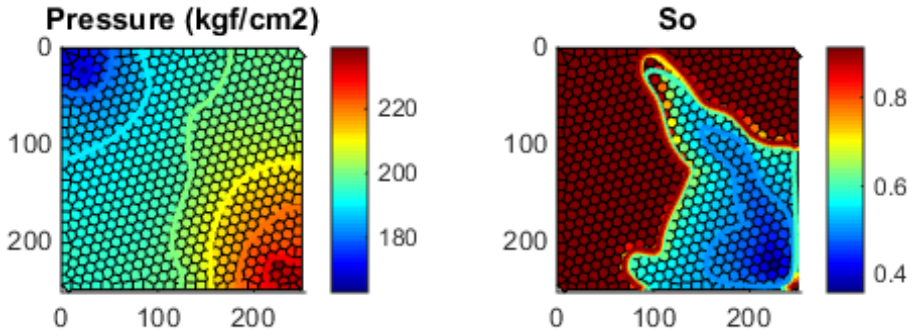

Figure 63 – Pressure and saturation maps for Cardwell & Parsons upscaling, nearly hexagonal grid, scenario B

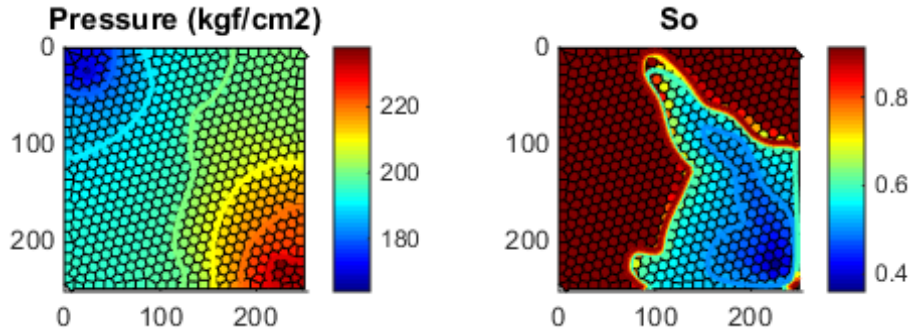

Figure 64 - Pressure and saturation maps for renormalization upscaling, nearly hexagonal grid, scenario B.

5.5.2.
Permeability based grid refinement

As was done for scenario A, unstructured grids with fewer cells can be generated to accelerate simulations. The chosen criterion was to have smaller spacing where permeability is high and higher flow is expected (on the channels). Although this could potentially be combined with refinement near the wells, no grid was generated with this strategy.

An atomic repulsion algorithm was used to distribute the cells this time. The resulting cells have dimensions closer to those determined by the spacing map than with electrostatic repulsion. The hexagonal lattice is also better preserved on the background, low density regions. However, due to the limited thickness of the channels, the total number of cells could not be reduced as much as with the homogeneous model without disfiguring the channels.

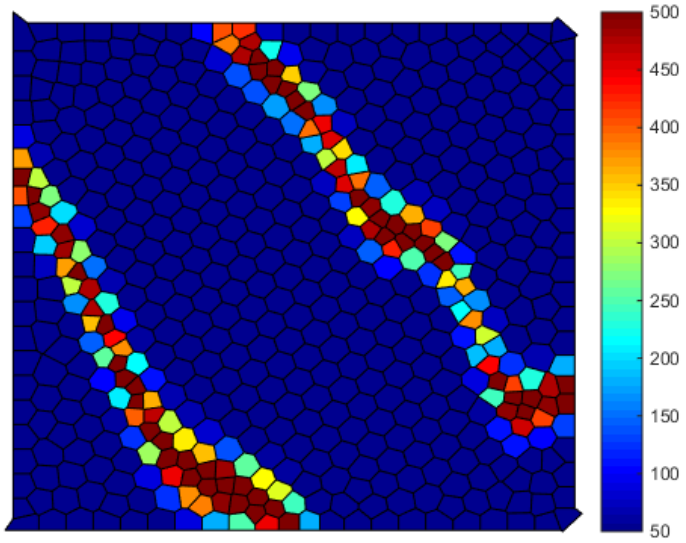

Figure 65 – Unstructured grid for scenario B with permeability based refinement (569 cells), colored by Wiener upscaled permeability.

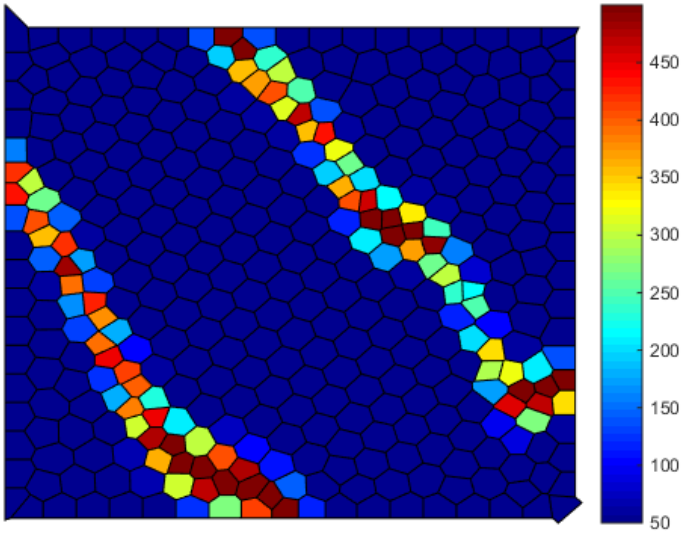

Figure 66 – Unstructured grid for scenario B with permeability based refinement (386 cells), colored by Wiener upscaled permeability.

The production results with those smaller models remain between the limits set before by the two structured models. The results with both upscaling methods are very close to each other, although renormalization has a water breakthrough a few days closer to the refined model.

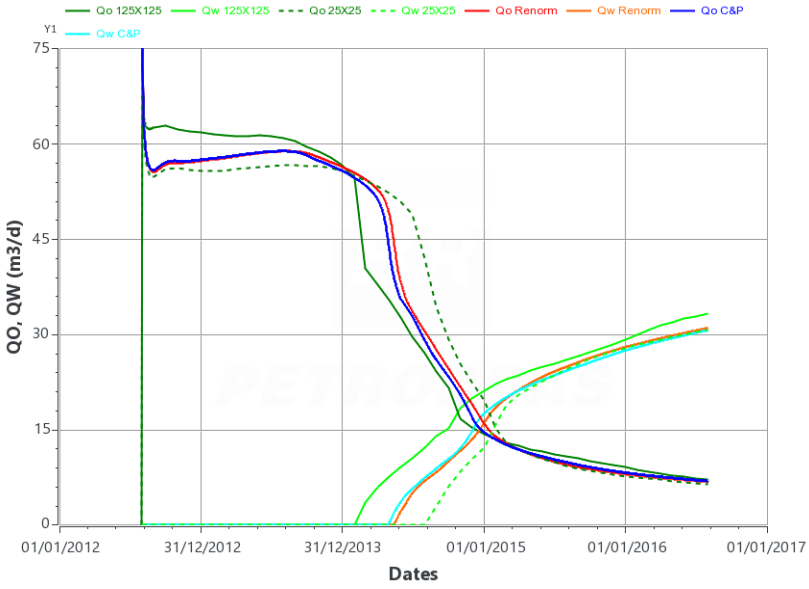

Figure 67 – Production curves for scenario B with permeability based refinement (569 cells).

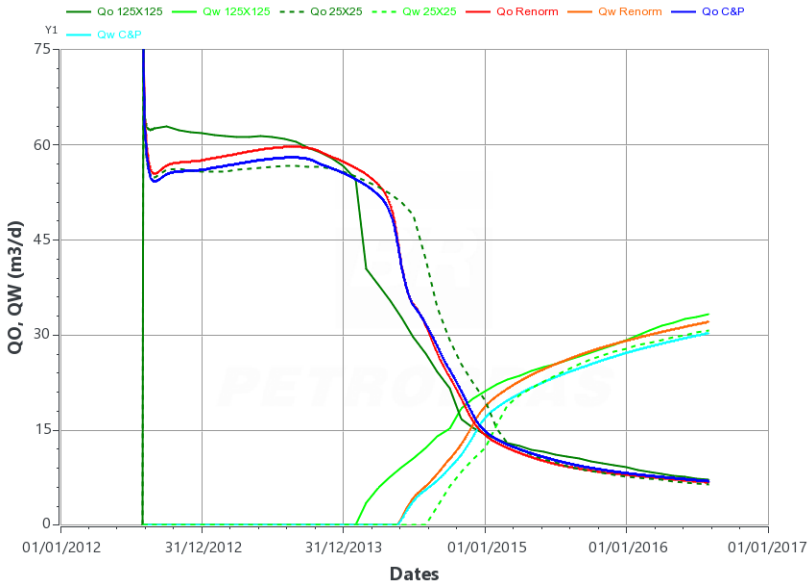

Figure 68 – Production curves for scenario B with permeability based refinement (386 cells).

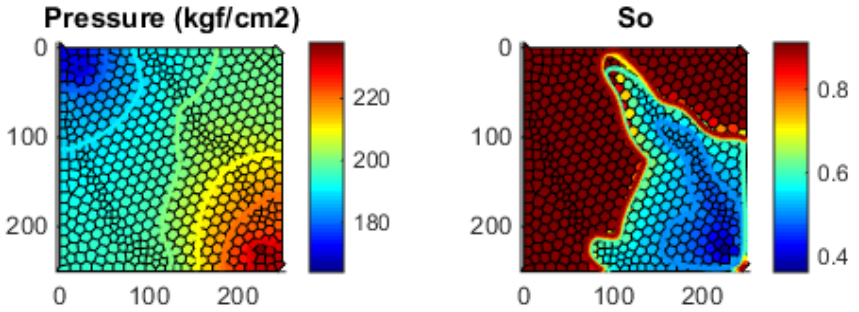

Figure 69 – Pressure and saturation maps for renormalization upscaling, permeability based refinement (569 cells), scenario B

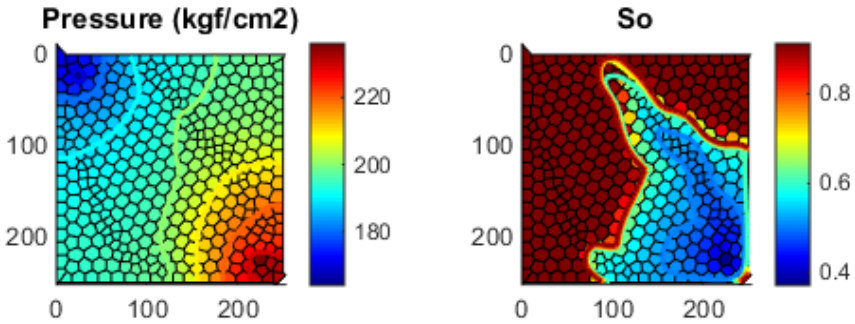

Figure 70 – Pressure and saturation maps for renormalization upscaling, permeability based refinement (386 cells), scenario B

5.6.
Scenario C

An even more heterogeneous scenario that can be considered is the amalgamation of several channels with variable thickness and direction (Figure 71 and Figure 72). Porosity was also considered heterogeneous and correlated to permeability for this scenario.

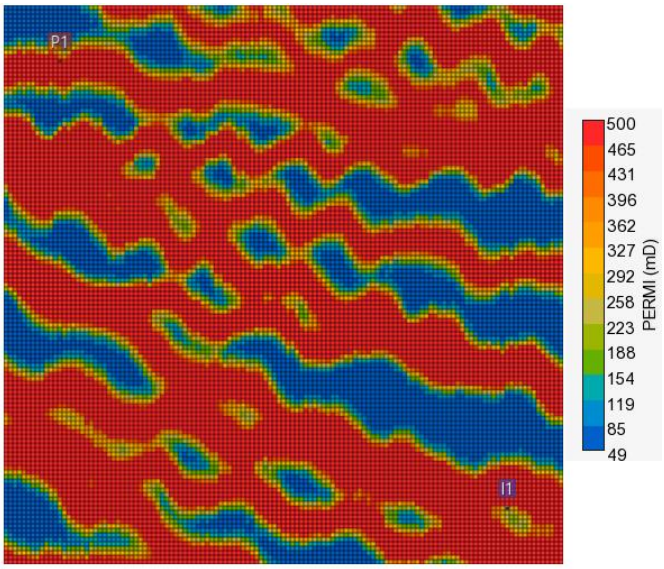

Figure 71 – Permeability map for scenario C with refined structured grid.

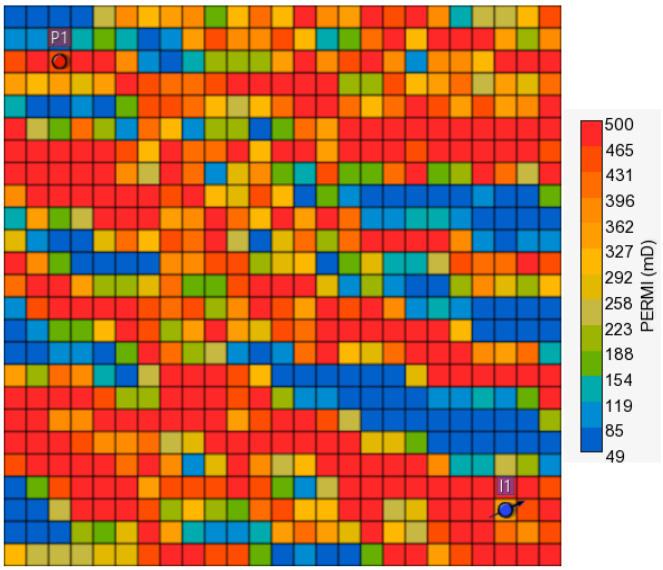

Figure 72 – Upscaled permeability map in the x direction for scenario C with medium structured grid.

5.6.1.
Hexagonal grid

The deviation between the production curves of the medium and the refined structured models was small for this scenario, against the initial expectations. Even so, results from the unstructured hexagonal grid are closer to the refined model (Figure 73).

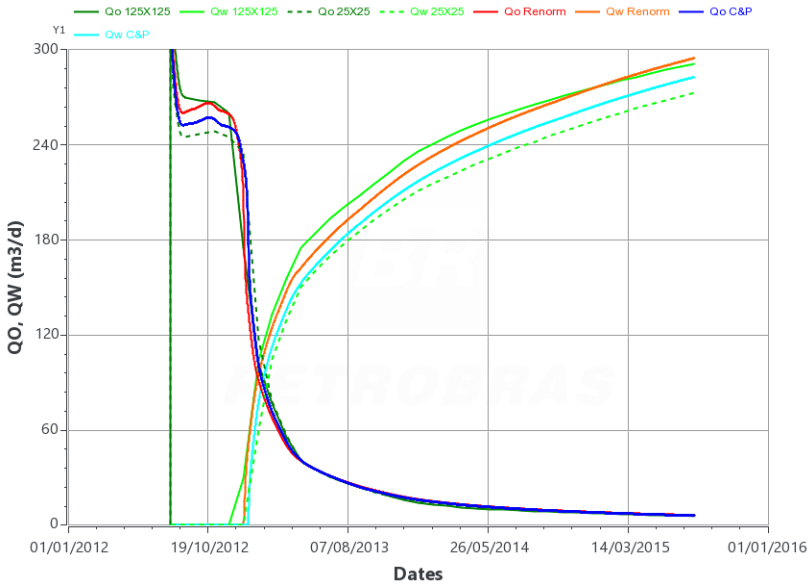

Figure 73 – Production curves for scenario C with nearly hexagonal grid.

5.6.2.
Permeability and permeability gradient grid refinement

Two criteria for the spacing map were compared for this scenario. One was based on permeability values, as was done with scenario B (Figure 74). The other criterion was based on the permeability gradient, in an attempt to better represent regions where permeability changes rapidly (Figure 75). A Gaussian smoothing filter must be applied to the gradient absolute value before it can be used as a spacing map, otherwise the spacing variations are too abrupt.

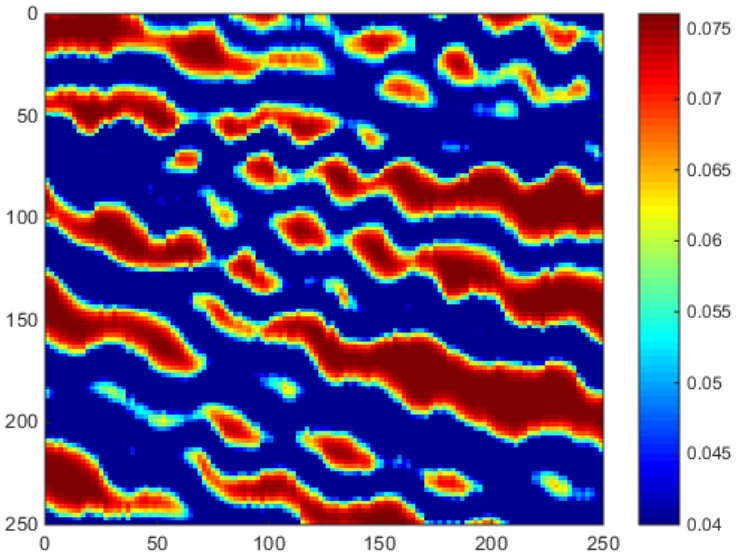

Figure 74 – Permeability based spacing map for scenario C.

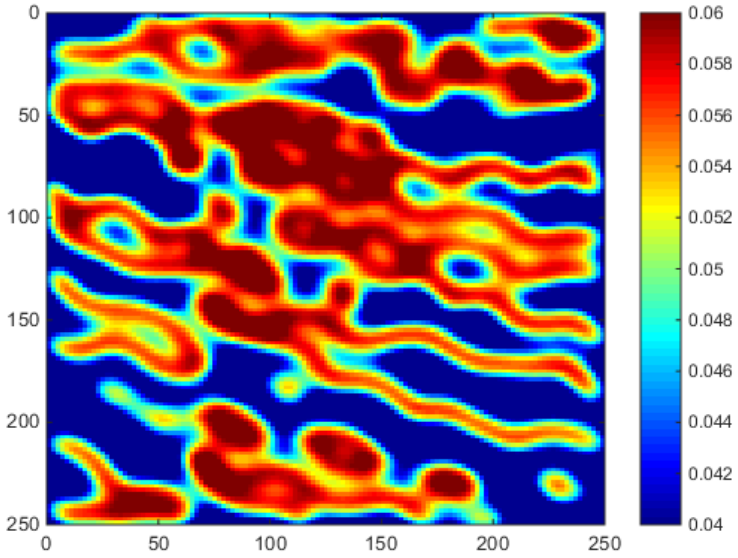

Figure 75 – Permeability gradient based spacing map for scenario C (smoothed).

The grid with permeability based spacing has 482 cells, while the one based on the gradient has 491 cells. The standard deviation from the mean cell porosity was 24% higher for the second grid, which suggests it was able to distinguish better the contrasts on the properties distribution from the refined grid (Figure 76 and Figure 77). Permeability fields for those two models were determined only with the modified renormalization method.

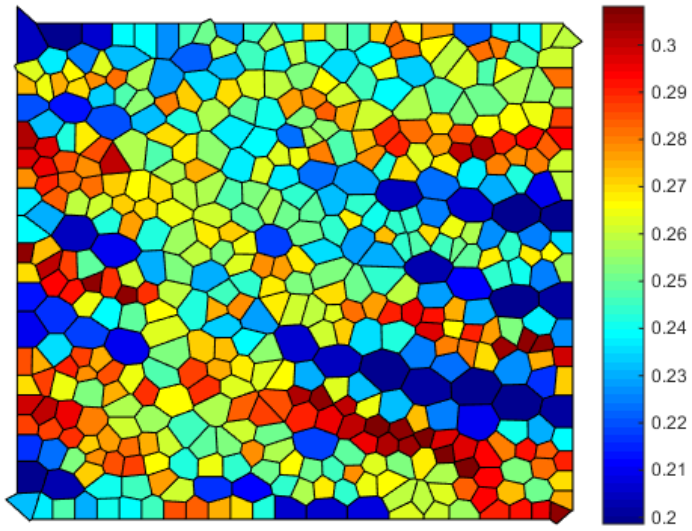

Figure 76 – Unstructured grid with permeability based refinement (482 cells) for scenario C, colored by porosity value.

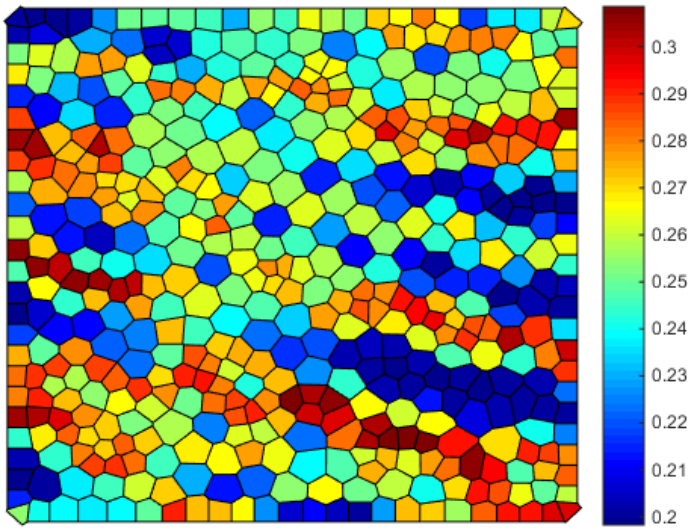

Figure 77 – Unstructured grid with permeability gradient based refinement (491 cells) for scenario C, colored by porosity value.

Production results for both grids are satisfactory, staying between the refined and medium structured grids results. However, against the original expectations, the grid with permeability based spacing had production results closer to the refined grid (Figure 78) than the other one. The results may have been affected by small variations on well index values, caused by different cell boundaries. Another hypothesis is that permeability gradient may not be an adequate criterion for this scenario, since the permeability distribution is practically bimodal and high gradients correspond simply to the border of the channels.

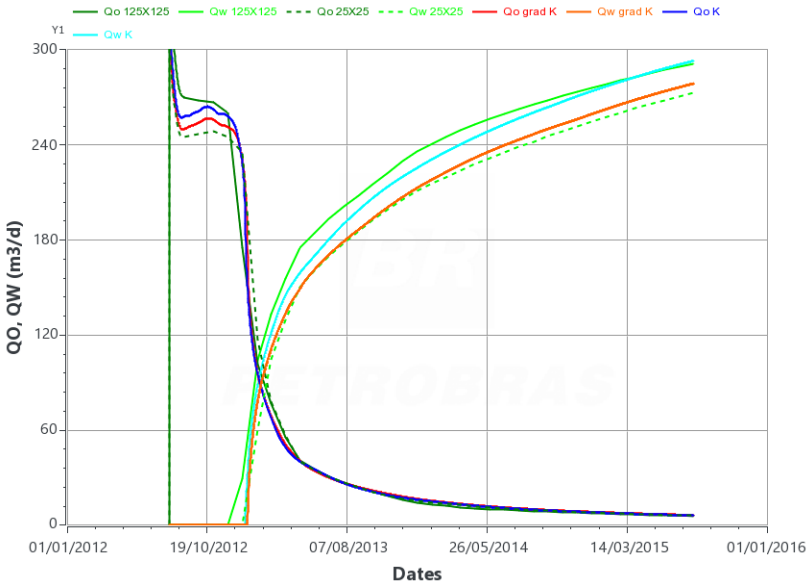

Figure 78 – Production curves for scenario C (only renormalization): permeability (blue) and permeability gradient (red) based grid refinement.

The grid generation algorithms proposed in this work were unable to determine an appropriate coarser grid for this scenario. Both the electrostatic repulsion and the atomic forces algorithms had convergence issues when used with a spacing map with larger values, such as noisy point distributions or large regions without any points. This stability problem may be a limitation of the iterative algorithms, or may be an issue of the spacing map itself.

5.7. Scenario D

Scenario D considers two straight sealing faults crossing a homogeneous reservoir between the producer and injector wells. With a structured model, unless it is aligned to one of the grid directions, the fault must be represented by alternating segments in the x and y directions (Figure 79 and Figure 80). Small portions of the porous volume may consequentially be attributed to the wrong side of the fault. Even a well too close to a fault may be placed incorrectly. Those effects become worse when coarser grids are used.

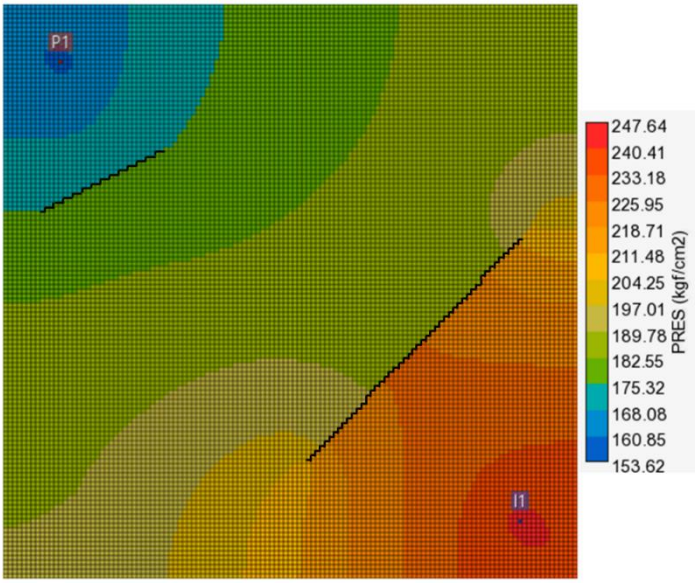

Figure 79 – Faults representation on the refined structured grid, with pressure map.

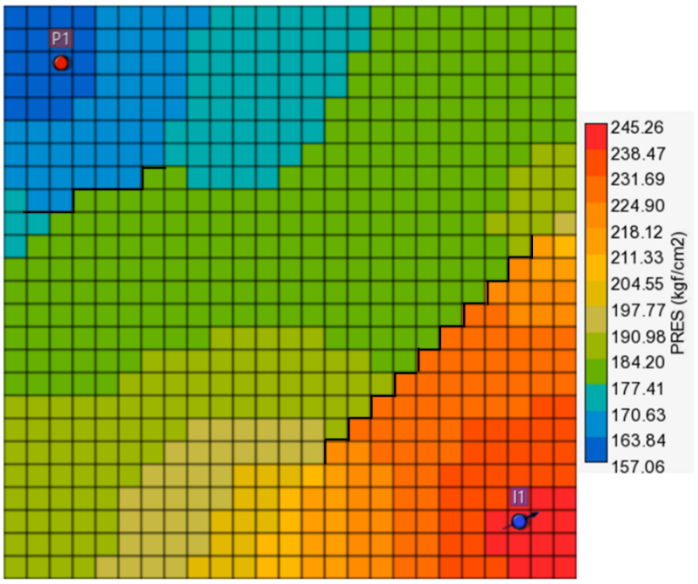

Figure 80 – Faults representation on the medium structured grid, with pressure map.

5.7.1. Hexagonal grid with fault representation

An unstructured grid with almost the same number of cells as the medium structured grid (626 versus 625) can represent the faults as straight lines and with more accurate limits (Figure 81). Even if the faults had a curvature, they could still

be represented by the current algorithm. However, intersections between faults, or between a fault and the border cause distortions in the grid, since each feature is treated separately.

As with previous scenarios, production results with the unstructured grid remain between the results of the refined and the medium structured grid (Figure 82). In some aspects, such as the definition of the pressure gradient and water advancing front, the results from the unstructured grid may even be superior to refined structured grid (Figure 83 and Figure 84), since they preserve the shape of the fault. Unfortunately, there is no way to evaluate this, except for a scenario that has analytical solution.

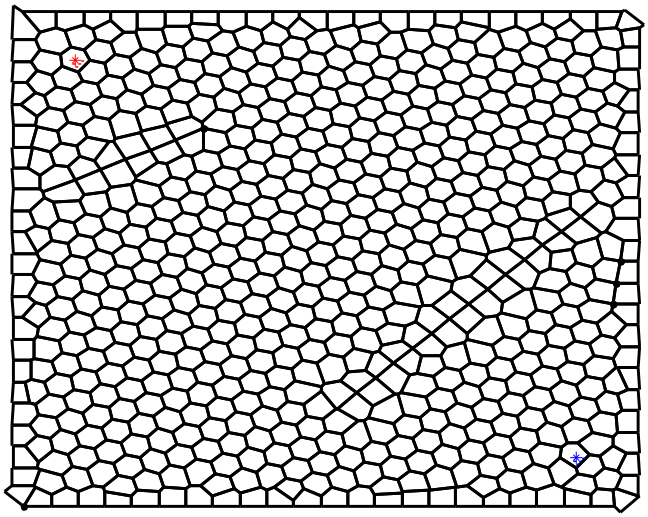

Figure 81 – Unstructured grid with faults representation for scenario D.

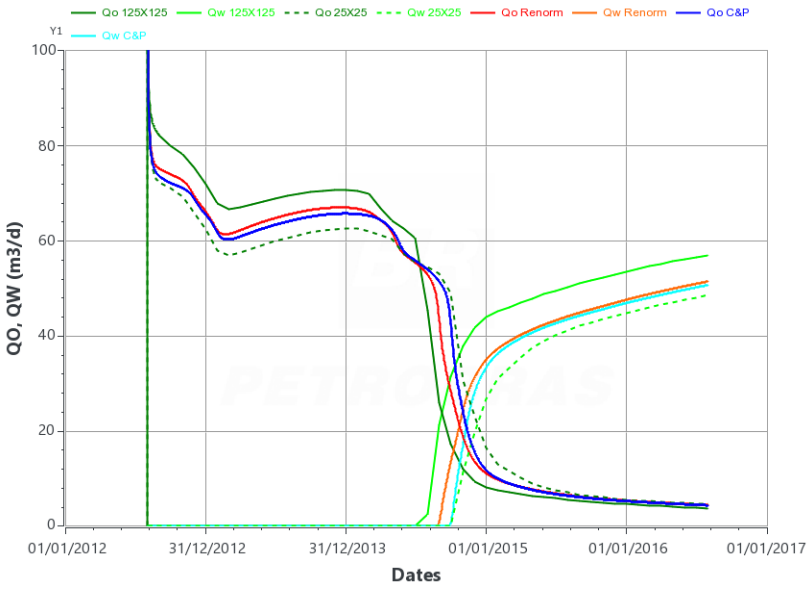

Figure 82 – Production curves for scenario D with unstructured grid.

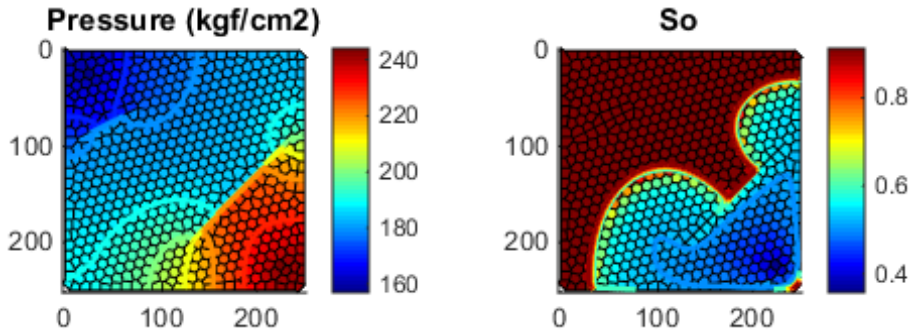

Figure 83 – Pressure and saturation maps for Cardwell & Parsons upscaling, unstructured grid with faults, scenario D.

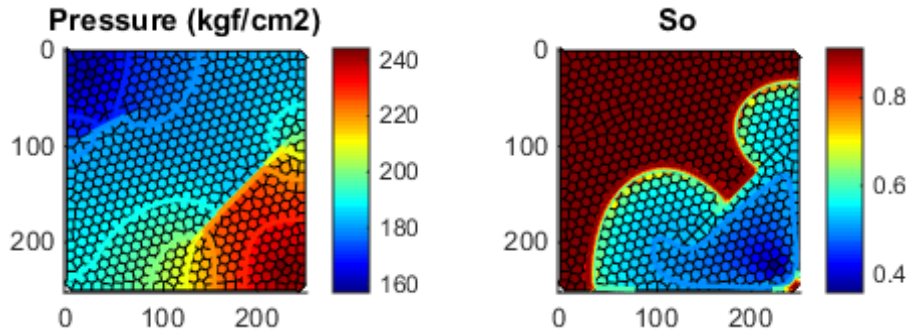

Figure 84 – Pressure and saturation maps for renormalization upscaling, unstructured grid with faults, scenario D.

Coarser unstructured grids were not generated for this specific scenario due to time constraints. Since permeability is homogeneous, two possibilities are to refine the grid near the wells or near the faults. An alternative is to refine the grid where the faults restrict the flow to a smaller area. However, this is harder to implement without estimating the flow beforehand.

5.8. Scenario E

Scenario E consists of parallel, slightly curved permeability channels (Figure 85). Although features so well structured are not usually found in real reservoirs, they serve as a test to the limits of the proposed algorithms. The distance between channel centers is around 20 meters.

The medium structured model is incapable of adequately representing the permeability field. The high frequency of the original distribution leaves the

upscaled permeability with a chequered aspect (Figure 86). The water advancing front pattern from the refined model is entirely lost (Figure 88), with production behaving closer to the expected for a homogenous permeability.

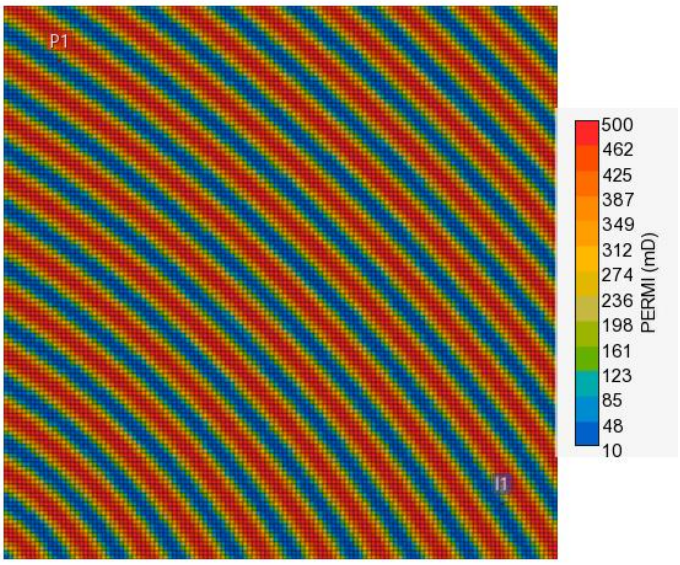

Figure 85 – Permeability map for scenario E with refined structured grid.

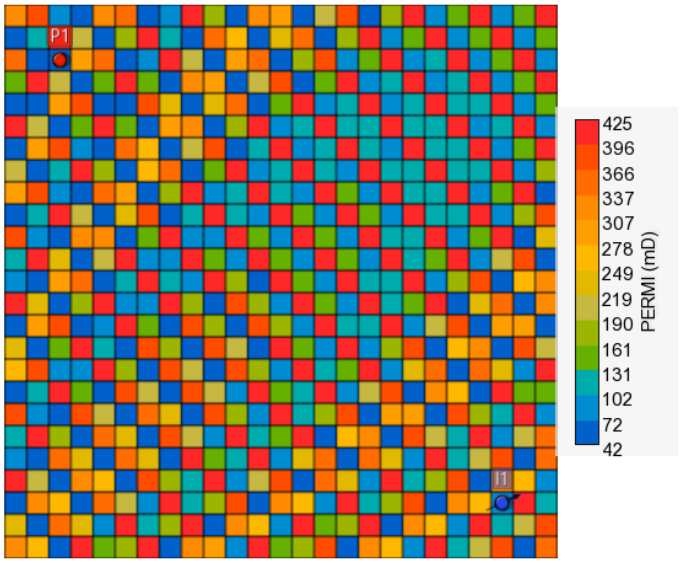

Figure 86 – Upscaled permeability map in the x direction for scenario E with medium structured grid.

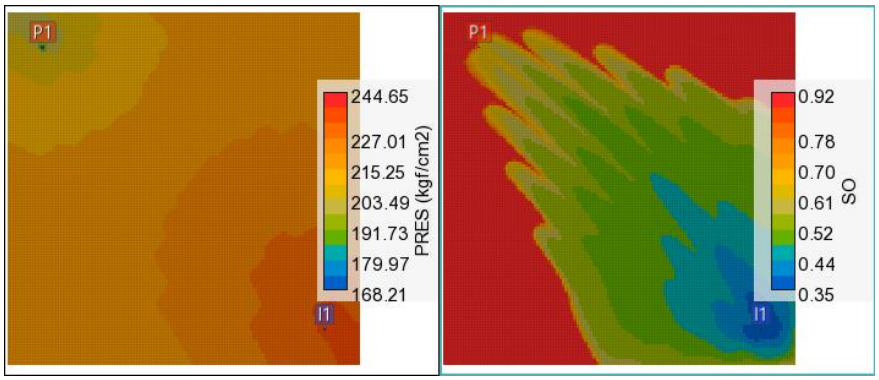

Figure 87 – Pressure and saturation maps for the refined structured grid, scenario E.

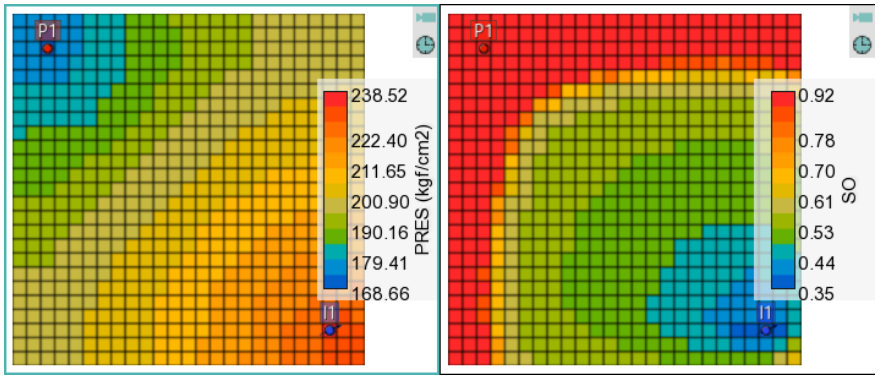

Figure 88 – Pressure and saturation maps for the medium structured grid, scenario E.

5.8.1. Hexagonal grid

Although the hexagonal grid is also incapable of reproducing the production curves from the refined model, the results are much closer than those of the medium structured model, for a similar number of cells (Figure 89). The expected water advancing front is reproduced with small deviations on this grid (Figure 90). This significant improvement can be attributed to two aspects: the hexagonal grid has connections in more directions than the rectangular one, so that the changing direction of the channels is better represented, and, as discussed in chapter 4, direct transmissibility calculations result in a higher resolution, so that the high frequency variations in permeability are better preserved.

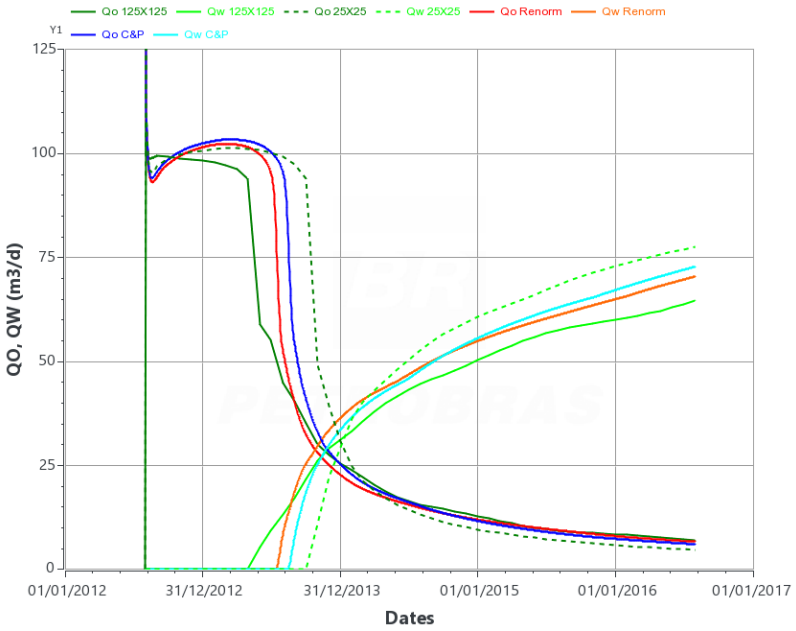

Figure 89 – Production curves for scenario E with nearly hexagonal grid.

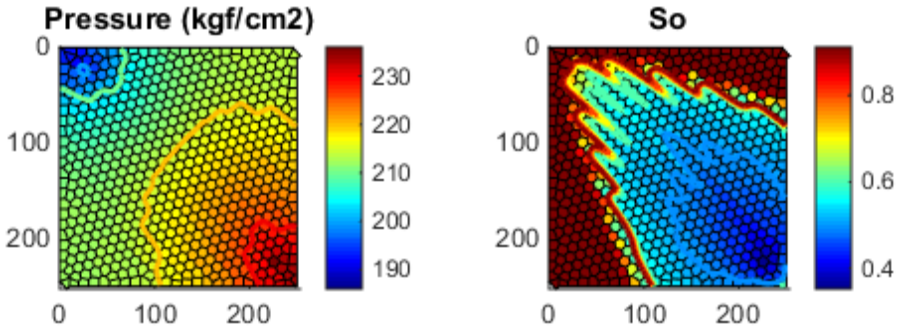

Figure 90 – Pressure and saturation maps for renormalization upscaling, hexagonal grid, scenario E.

5.8.2. Permeability pattern aligned grid

The grid generation algorithms proposed in chapter 3 were not able to represent the structure of this permeability distribution with a number of cells smaller than the medium structured grid. Over 1000 grid points were necessary to converge to a minimally representative model.

The adopted solution was to generate equally spaced grid points along the channel centers (where permeability is highest), without using a spacing map or node redistribution algorithms. This approach is not reproducible for general heterogeneous permeability fields, since it relies on the specific structure of this

scenario. The obtained grid, with 254 cells (Figure 91), had production results superior to the hexagonal model, which has over double the number of cells, when upscaling was performed by renormalization (Figure 92 and Figure 93). The model with Cardwell & Parsons upscaling had an overestimated productivity that can be attributed to a mismatch on the calculated well index level.

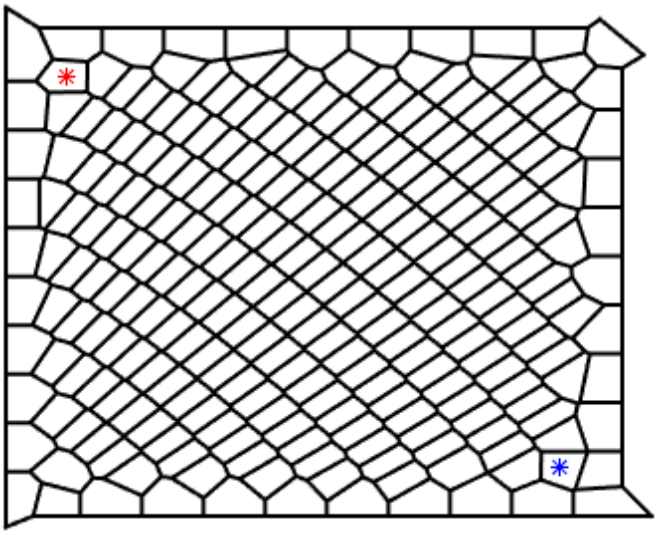

Figure 91 – Unstructured grid aligned to channels for scenario E

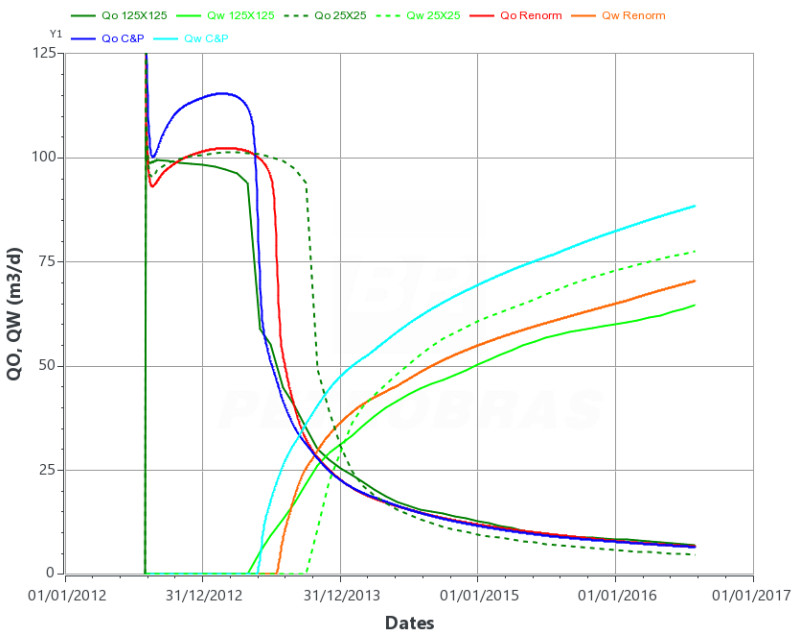

Figure 92 – Production curves for scenario E with grid aligned to channels.

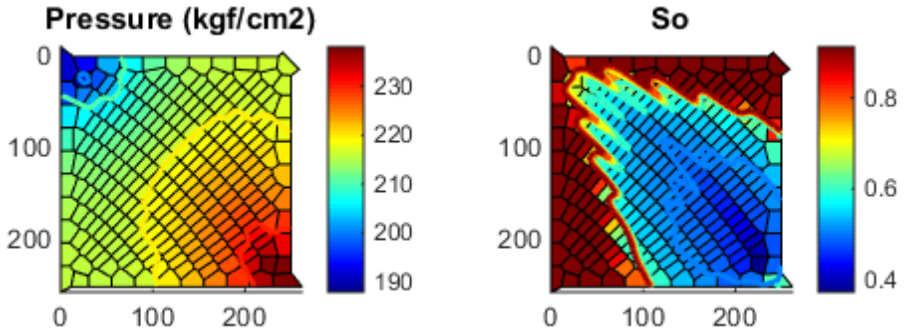

Figure 93 – Pressure and saturation maps for renormalization upscaling, cells aligned with channels (254 cells), scenario E.

6
Final Considerations

Reservoir simulation is an essential tool for reservoir engineering. However, computational cost can still be an issue for huge or complex models. Unstructured grids are a powerful resource to generate simulation models with fewer cells without sacrificing the quality of the results. Complex geological features may also be better represented with unstructured grids.

This dissertation investigated alternatives for unstructured grid generation and upscaling methods that do not require fine scale (structured) simulations. Most methods from literature rely on single-phase steady state flow simulations of the fine grid. Even with those simplifications, this can be a costly activity for huge models. Another issue is that if the well arrangement or production conditions are changed, the generated model may no longer be adequate.

The proposed grid generation strategy was to start from a regular template and then redistribute the point set according to a previously defined spacing map. Two iterative redistribution algorithms based on physical models were proposed. Several criteria for spacing maps were also investigated. From the final point distribution, the grid cells were determined by a Voronoi algorithm.

Two upscaling algorithms for unstructured grids were proposed. Those methods are modified versions of the Cardwell & Parsons and renormalization techniques for structured grids.

6.1.
Conclusions

Unstructured grids were used to represent several reservoir models. The unstructured model results for all scenarios were at least equivalent to those from structured models with a similar number of cells, and in most cases were closer to the results of more refined structured models. Even a regular hexagonal grid was shown to give superior results than a structured model with a similar number of cells for scenarios with heterogeneous reservoir properties.

Unstructured model simulation has an increase in computational cost when compared to structured models with the same number of cells. This is related to the higher number of active connections between cells that must be handled by the numerical solver with most unstructured models. This effect is compensated by the need of less cells to adequately represent the reservoir behavior, resulting is a smaller Jacobian and lower overall computational cost if the grid size is small enough.

A limiting factor for cell number reduction was the high overall drainage efficiency of the simple tested models. Full field models closer to real reservoirs often have significant areas where drainage is very low and therefore can benefit more from unstructured grid flexibility.

6.1.1. Grid generation

The insertion of grid points to represent discrete features such as well or faults causes deformations on the initial regularly spaced distribution. The grid point redistribution algorithms were effective for smoothing those deformations while preserving the discrete features on grids with constant spacing. They were also capable of refining some grids when used with variable spacing maps. However, some limitations of those algorithms were evidenced on the generation of grids with high spacing contrasts or complex features, such as intersecting faults.

The proposed redistribution algorithms, based on the equilibrium of physical forces, require several iterations and adequate parameter calibration to converge. This can become very costly for a high number of grid points (over 2000 or more) or a highly variable spacing map. The algorithms are more effective when working with a spacing map that has small variations.

The atomic forces algorithm has the advantage of enforcing the desired spacing value for the point distribution. The electrostatic forces algorithm considers the spacing map only as a weight to the forces, so that regions with a small spacing value will have a higher point density but not respect the specified spacing value. However, point distributions generated by electrostatic equilibrium are smoother and more regularly shaped than those generated by atomic forces, which can end with empty regions due to the attraction from other regions.

Simple spacing criteria, such as those based on permeability or distance from wells were shown to be effective to reduce the total number of cells without degrading the simulation results for some scenarios. However, an optimal grid point distribution is problem dependent, especially for complex heterogeneous models. Running fine scale simulations remains the most effective alternative to determine this distribution in a generic way.

Voronoi grid was shown to be a good unstructured grid alternative for reservoir simulation. The resulting grids were generally well-behaved, without deformed cells that can affect simulation convergence. Minimal modifications to the simulator were necessary to adapt it for Voronoi grid, making the implementation of the code as a plug-in rather practical. This way, future developments of the simulator can be inherited by both the structured and unstructured versions with little additional effort.

6.1.2. Upscaling

Both proposed upscaling algorithms were capable of representing the tested scenarios permeability adequately. In most cases, the results with both methods were similar, although renormalization has a tendency to estimate slightly higher transmissibility values.

The Cardwell & Parsons modified algorithm is simpler and has the smallest computational cost of the two. Originally, it was expected that this algorithm would not be able to estimate appropriate transmissibilities for reservoirs with complex permeability arrangements. However, this could not be verified on any of the tested scenarios.

The modified renormalization algorithm is very flexible. Since all calculations are made directly with transmissibilities, it should work even if the fine scale model is also unstructured. Some care must be taken with numerical issues arising when the cell ratio between models is high.

6.2. Future work possibilities

This dissertation was proposed as an initial investigation into unstructured reservoir models, associated with the development of the GSim simulator. As such, it was not expected to solve most implementation issues, but to expose alternatives for future development.

The current version of GSim adopts a fully implicit solution of the flow equations, with a constant time step throughout simulation. All computational cost comparisons used the same time step length, defined to avoid convergence issues throughout simulation. Adopting an adaptive time step in GSim may reduce simulation time significantly for some of the models, leading to a comparison closer to what is expected in a scenario with a commercial simulator. Changing the simulation solver from fully implicit to IMPES (implicit pressure and explicit saturation) or AIM (adaptive implicit method) can also have interesting effects on computational time and simulation stability.

The experimental models on this dissertation were limited to 2.5D models with a single layer. In order to better represent the sort of reservoir models in use by the petroleum industry, the experiments should be extended to models with multiple layers, and to the representation of inclined layers and fault throw. Those features lead to new questions regarding how to evaluate vertical permeability and transmissibility for unstructured models. Bigger models may also have a higher potential for reducing the total cell number by using unstructured grids.

The proposed grid generation strategy, based on the iterative redistribution of grid points from an initial regular point distribution, was not robust enough to represent complex reservoirs with a small number of cells. It also had a high computational cost for models with more than 10000 cells. A method based on an advancing front with a spacing map (Evazi, 2009) or on optimizing the global point distribution (Hale, 2002) may perform better on those challenging cases. This could be implemented as future research, or a third party grid generator could be used.

An even more ambitious step would be to propose an adaptive unstructured grid. This solution would allow to focus cell refinement, for instance, on the water advancing front of a water injection model, where the higher saturation changes are expected to happen. This approach has the potential to reduce further the total

number of cells. However, the grid would have to be reevaluated several times during simulation since the front is moving. This can have significant impact on simulation time if the gridding algorithm is not efficient. Grid generation strategies that permit the insertion and removal of cells without having to redetermine the whole grid are ideal for this application.

The current Fortune algorithm implementation is more efficient than an incremental method and can determine Voronoi grids with less than 1000 cells in under a minute. However, for bigger grids, implementing a binary tree for the Fortune event queue becomes mandatory to keep the grid generation time reasonable.

Both upscaling strategies had satisfactory results on all tested scenarios. Even so, they should be reevaluated for bigger models, intricate permeability distributions and more elaborate development strategies.

One promising line of use for unstructured grids is the representation of fractures, faults and associated damage zones. The grid can be aligned to those features and cells can even have a different aspect ratio to better represent the flow on those regions. The fact that the unstructured model informs directly the transmissibility between cells instead of cell permeability is also an advantage, since those can be directly altered to represent sealing faults, increased flow on damage zones and the direct connection between non-neighboring cells.

Another possibility of unstructured grids is to represent the flow along wells, valves and lines as cells of the same model as the reservoir. This requires further changes both to the modeling workflow and to the simulator, to allow cells with different geometry and different governing flow laws.

7
Bibliographical References

ABOU-KASSEM, Jamal H.; Aziz, Khalid. Analytical well models for reservoir simulation. **Society of Petroleum Engineers Journal**, v. 25, n. 04, p. 573-579, 1985.

AZIZ, Khalid. Reservoir simulation grids: opportunities and problems. **Journal of Petroleum Technology**, v. 45, n. 07, p. 658-663, 1993.

BAHRAINIAN, Seyed Saied; DEZFULI, Alireza Daneh. A geometry-based adaptive unstructured grid generation algorithm for complex geological media. **Computers & geosciences**, v. 68, p. 31-37, 2014.

BLAZEK, Jiri. **Computational fluid dynamics: principles and applications**. Butterworth-Heinemann, 2015.

CELES, W.; PAULINO, G.H.; ESPINHA, R., A compact adjacency-based topological data structure for finite element representation. IJNME, 64(11) 1529-1556 (2005)

CHEN, Qian-Yong; WAN, Jing; MIFFLIN, Rick T. A new multipoint flux approximation for reservoir simulation. In: **SPE Reservoir Simulation Symposium**. Society of Petroleum Engineers, 2007.

DE BERG, Mark et al. Computational geometry. In: **Computational geometry**. Springer, Berlin, Heidelberg, 1997. p. 1-17.

DUARTE, Leonardo S. et al. PolyTop++: an efficient alternative for serial and parallel topology optimization on CPUs & GPUs. **Structural and Multidisciplinary Optimization**, v. 52, n. 5, p. 845-859, 2015.

EVAZI, M.; MAHANI,H. Unstructured Coarse Grid Generation for Reservoir Flow Simulation Using Background Grid Approach. In: **SPE Middle East Oil and Gas Show and Conference**. Society of Petroleum Engineers, 2009.

GUNASEKERA, D.; COX, J.; LINDSEY, P. The generation and application of K-orthogonal grid systems. In: **SPE Reservoir Simulation Symposium**. Society of Petroleum Engineers, 1997.

HALE, Dave. Atomic meshes: from seismic imaging to reservoir simulation. In: **Proceedings of the 8th European Conference on the Mathematics of Oil Recovery**. 2002.

HEINEMANN, Z. E. et al. Modeling reservoir geometry with irregular grids. In: **SPE Symposium on Reservoir Simulation**. Society of Petroleum Engineers, 1989.

HEINRICH, Bernd. **Finite difference methods on irregular networks: a generalized approach to second order elliptic problems**. Birkhäuser, 1987.

Khan, S. A; Dawson, A. G. Method of upscaling permeability for unstructured grids. Patent US6826520B1, 2004.

KING, P. R. The use of renormalization for calculating effective permeability. **Transport in porous media**, v. 4, n. 1, p. 37-58, 1989

LEDOUX, Hugo. Computing the 3d Voronoi diagram robustly: An easy explanation. In: **4th International Symposium on Voronoi Diagrams in Science and Engineering (ISVD 2007)**. IEEE, 2007. p. 117-129.

MANCHUK, J G.; DEUTSCH, C. V. Unstructured Grid Generation Considering Reservoir Properties. **CCG Annual Report 11,** 2009.

MENEZES, Ivan F.; TILIO, Marcelo; MIRANDA, Antônio. **Neutral File.** PUC-Rio, Tecgraf. Available on: <https://web.tecgraf.puc-rio.br/neutralfile>. Accessed on: 10 nov. 2018.

MLACNIK, M.; DURLOFSKY, L. J.; Heinemann, Z. E. Sequentially adapted flow-based PEBI grids for reservoir simulation. **SPE Journal**, v. 11, n. 03, p. 317-327, 2006.

OKABE, Atsuyuki et al. **Spatial tessellations: concepts and applications of Voronoi diagrams**. John Wiley & Sons, 2009.

PALAGI, Cesar Luiz; AZIZ, Khalid. Use of Voronoi grid in reservoir simulation. **SPE Advanced Technology Series**, v. 2, n. 02, p. 69-77, 1994.

PEACEMAN, D. W. Interpretation of Well-Block Pressures in Numerical Reservoir Simulation with Nonsquare Grid Blocks and Anisotropic Permeability, paper SPE-10528. In: **Sixth SPE Symposium on Reservoir Simulation of the Society of Petroleum Engineers, New Orleans, LA**. 1982.

PEACEMAN, D. W. Interpretation of well-block pressures in numerical reservoir simulation (includes associated paper 6988). **Society of Petroleum Engineers Journal**, v. 18, n. 03, p. 183-194, 1978.

PIAZZA, R. E. **Performance Assessment of Fully Implicit Reservoir Simulator Solvers.** Rio de Janeiro, 2019. 162p. Masters dissertation – Departamento de Engenharia Mecânica, PUC-Rio.

PRÉVOST, Mathieu et al. Unstructured 3D gridding and upscaling for coarse modelling of geometrically complex reservoirs. **Petroleum Geoscience**, v. 11, n. 4, p. 339-345, 2005.

REBAY, Stefano. Efficient unstructured mesh generation by means of Delaunay triangulation and Bowyer-Watson algorithm. **Journal of computational physics**, v. 106, n. 1, p. 125-138, 1993.

RENARD, Ph; DE MARSILY, G. Calculating equivalent permeability: a review. **Advances in water resources**, v. 20, n. 5-6, p. 253-278, 1997.

ROMEU, R. K.; NOETINGER, B. Calculation of internodal transmissivities in finite difference models of flow in heterogeneous porous media. **Water Resources Research**, v. 31, n. 4, p. 943-959, 1995.

SAHIMI, Muhammad et al. Upscaled unstructured computational grids for efficient simulation of flow in fractured porous media. **Transport in porous media**, v. 83, n. 1, p. 195-218, 2010.

SETTARI, Antonin; AZIZ, Khalid. Use of irregular grid in reservoir simulation. **Society of Petroleum Engineers Journal**, v. 12, n. 02, p. 103-114, 1972.

USADI, Adam K. et al. Parallelization on Unstructured Grids. In: **SPE Reservoir Simulation Symposium**. Society of Petroleum Engineers, 2007.

WOLFMAN, S. Algorithm for generation of Voronoi Diagrams. Notes for CSE 326: Data Structures (featuring algorithms) course on University of Washington Computer Science & Engineering. Available on: <https://courses.cs.washington.edu/courses/cse326/00wi/projects/voronoi.html. 2000>. Acessed on: 10 jan. 2018.

WU, Fa-Yueh. The potts model. **Reviews of modern physics**, v. 54, n. 1, p. 235, 1982.

A
Reservoir test model description

This appendix describes in further detail the models used during chapter 5. All of them are variations of the same reservoir, with different porosity and permeability values, as specified on Table 6. The reservoir is located at a 3000m depth and has box geometry, with 250m lateral width and 10m thickness.

Rock compressibility was considered to be 50E-6 $(kgf/cm^2)^{-1}$. The adopted relative permeability curves are presented on Table 9 and Table 10.

Table 9 – Oil and water relative permeability tables

Water saturation	Kwr	Kor
0.088	0	0.8
0.1	0	0.75
0.15	0	0.563
0.2	0.002	0.411
0.25	0.005	0.288
0.3	0.011	0.193
0.35	0.02	0.122
0.4	0.034	0.07
0.45	0.053	0.036
0.5	0.079	0.015
0.55	0.111	0.005
0.6	0.151	0.001
0.65	0.2	0
1	1	0

Table 10 – Liquid and gas relative permeability tables

Liquid saturation	Krg	Krl
0.088	0.9	0
0.1	0.865	0
0.15	0.729	0
0.2	0.607	0.001
0.25	0.501	0.004
0.3	0.407	0.01
0.35	0.326	0.019
0.4	0.256	0.032

0.45	0.197	0.05
0.5	0.148	0.074
0.55	0.108	0.104
0.6	0.076	0.142
0.65	0.051	0.187
0.7	0.032	0.242
0.75	0.019	0.306
0.8	0.009	0.381
0.85	0.004	0.467
0.9	0.001	0.565
0.95	0	0.676
1	0	0.8

The reservoir is filled primarily with oil, without any water oil or gas oil contact. The only other fluid present at simulation start is connate water. The initial reservoir pressure is 200 kgf/cm^2, while the fluid bubble pressure is 100 kgf/cm^2. The (extrapolated) PVT properties used to characterize the fluid in the simulator are given by Table 11.

Table 11 – PVT properties table

Pressure (kg/cm2)	Gas Oil Ratio	Oil FVF	Gas FVF	Oil viscosity (cp)	Gas viscosity (cp)
1.03	0	1	0.8239	2.7695	0.01092
26.03	20	1	21.7978	1.7277	0.0133
31.03	22	1	26.1419	1.6713	0.01346
41.03	27	1	34.978	1.5707	0.01373
51.03	32	1	43.9965	1.4838	0.014
61.03	37	1	53.1816	1.4074	0.01428
67.02	40	1	58.7501	1.3658	0.01445
80	46	1	71.2296	1.3351	0.01484
100	56	1	90.8276	1.2611	0.01549
120	66	1	110.596	1.2006	0.01619
150	81	1	139.8711	1.1266	0.01735
180	96	1	167.8947	1.0661	0.01863
200	106	1	185.579	1.0311	0.01955
250	131	1.3689	225.7614	0.9571	0.02209
300	156	1.4287	260.2866	0.8966	0.02497
350	180	1.4884	289.8959	0.8454	0.02819

One vertical producer well is placed at (25m,25m) coordinates and set to operate with fixed 150 kgf/cm2 bottom hole pressure. A water injector well is placed at the opposite corner, operating with a 250 kgf/cm2 bottom hole pressure. The reservoir simulation runs for 4 years, with a one day maximum time step.