Andreas Rath, Bsc.

Master Thesis 2018 supervised by:
Holger Ott, Univ.-Prof. Dipl.-Phys. Dr.rer.nat.
Siroos Azizmohammadi, PhD

Mapping Structured Grid Properties on Triangulated Meshes

dpe DEPARTMENT PETROLEUM ENGINEERING

MONTAN UNIVERSITÄT

I dedicate this thesis to my parents for always
supporting me in the last years

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been published elsewhere. This dissertation is the outcome of my own work using only cited literature.

Erklärung

Hiermit erkläre ich, dass der Inhalt dieser Dissertation, sofern nicht ausdrücklich auf die Arbeit Dritter Bezug genommen wird, ursprünglich ist und nicht an anderer Stelle veröffentlicht wurde. Diese Dissertation ist das Ergebnis meiner eigenen Arbeit mit nur zitierter Literatur.

Andreas Rath, 19 September 2018

Acknowledgements

I want to acknowledge the following people for supporting me in the past years and making this work possible.

First, I want to thank my family for supporting me in every possible way. Their advice was not only appreciated regarding my education in the recent years, also their counsel in private concerns was important and enlightening.

Another person I want to thank is Siroos Azizmohammadi who guided me throughout my studies and supervised me in both, bachelor and master thesis.

In the end, I also want to thank all my friends who were there for me and were the necessary distraction of a stressful university day.

Abstract

The purpose of this work is to identify a reliable and effective method to map porous media properties like e.g. porosity, from a structured grid to an unstructured triangulated mesh. The generation of the unstructured triangulated grid regarding this thesis strongly relies upon a wavelet-based coarsening algorithm, which leads to a non-uniform structured grid. Depending on the applied coarsening-level one can detect differences in the results of the triangulated grid, which ultimately affects the simulated flow behavior. The computation of properties on the generated triangulated grid is of importance as it provides more flexibility regarding flow simulations.

The final objective of this thesis is to identify the most promising mapping algorithm to assign properties from structured on unstructured grids. In general, two major categories of mapping techniques exist, which will be discussed in more detail. Furthermore, two-phase flow simulations, using MRST library, are conducted on the considered grid structures to investigate the simulated flow behavior on different grids. The most important parameters in this study are the coarsening- level, produced volumes and the well placement.

In order to determine the most reliable mapping technique, the algorithms of interest are applied on two datasets. For evaluation of the methods, three different aspects are considered that are complexity of implementation, accuracy and runtime. Produced oil, breakthrough time and water cut are selected as output parameters for the flow simulations. In addition, a set of other parameters was tested to get additional insight in the mapping algorithm, which will be discussed in the thesis.

As a conclusion, the simplest algorithm, the inverse distance method, shows the best performance for this specific mapping purpose. Furthermore, the simulation results show some interesting effects; by using a triangulated grid, the oil production decreases the higher the coarsening level is picked. This can be traced back to the calculation of pressure that is biased by the mapped properties. Moreover, the simulation results indicate a faster breakthrough if using a structured uniform instead of a coarsened grid.

Zusammenfassung

Ziel dieser Arbeit ist es, eine zuverlässige und effektive Methode für das Mapping von Lagerstättenparametern, wie etwa Porosität, auf unstrukturierte, triangulierte Gitter zu finden. Die Erstellung des unstrukturierten, triangulierten Gitters beruht auf der Geometrie eines nicht gleichförmigen, strukturierten Gitters, welches durch einen wellenbasierenden Vergröberungsalgorithmus erzeugt wurde. Dieser Algorithmus erlaubt es, verschiedene Vergröberungen der Gitterstruktur zu erzeugen, womit schlussendlich die Geometrie des triangulierten Gitters und damit auch Simulationsergebnisse beeinflusst werden können. Die Berechnung der Gitterparameter, die den triangulierten Gitter zugewiesen werden sollen, sind von großer Bedeutung, da sie ebenfalls die Simulationseigenschaften beeinflussen.

Aus diesem Grund soll für diesen speziellen Anwendungszweck der vielversprechenste Algorithmus identifiziert und durch einen Programmcode bereitgestellt werden. Grundsätzlich kann zwischen zwei Kategorien von Mappingalgorithmen unterschieden werden, welche im Literaturteil der Arbeit genauer beschrieben werden. Anschließend sollen mehrphasige Flüssigkeitssimulationen mithilfe von MRST Unterschiede zwischen den verschiedenen Gitterstrukturen aufzeigen, wobei mehrere Faktoren berücksichtigt werden. Die wichtigsten Faktoren sind dabei der verwendete Vergröberungsfaktor des wellenbasierenden Algorithmus, das produzierte Ölvolumen und die Platzierung der Bohrungen.

Um festzustellen welcher Mappingalgorithmus die vielversprechensten Resultate liefert, wurden die vorgestellten Algorithmen an zwei Datensätzen getestet. Grundlage der Bewertung waren Anwendbarkeit, Genauigkeit und Laufzeit, welche für jede Simulation einzeln bewertet wurden. Anchließend wurden Simulationen basierend auf den Resultaten des optimalen Algorithmus durchgeführt und dabei Faktoren wie produziertes Ölvolumen, Durchbruchzeit und Wasseranteil beobachtet.

Letztendlich stellte sich heraus, dass der einfachste Algorithmus, die Inverse Distanzmethode, die besten Ergebnisse generiert. Weiters kann festgestellt werden, dass das produzierte Ölvolumen unter Verwendung von triangulierten Gittern sinkt, je größer die Vergröberung festgelegt wurde. Dieses Phänomen kann darauf zurückgeführt werden, dass der berechnete Druck durch die geringfügige Veränderung der Gitterparameter während des Mappings, vom initial berechneten Druck abweicht. Des weiteren kann durch mehrere Simulationen gezeigt

werden, dass es bei strukturierten gleichförmigen Gittern zu einem erschnellten Durchbruch von Injektionswasser kommt.

Table of Contents

List of Figures

List of Tables

Abbreviations

MRST	Matlab Reservoir Simulation Toolbox
PEBI	Perpendicular Bisectional
SGS	Sequential Gaussian Simulation

Chapter 1

Introduction

A vital part of modern reservoir engineering is to conduct flow simulations on a broad variety of problems. Implementing the different problems with all their complexity to establish a perfect copy of the reality can sometimes be problematic. The purpose of this thesis will be to describe difficulties coming along with mapping grid properties from structured grids to triangulated meshes. Furthermore, different approaches regarding grid transformation to unstructured meshes and mapping techniques with all their corresponding effects on simulation are investigated.

1.1 Background and Context

Today, finite -difference, -volume and element methods are the most common numerical discretization schemes used for the realization of reservoir simulation. All of them strongly rely on the corresponding grid structure. Generally, grids are categorized into structured and unstructured grids, whereas both can further be classified into uniform or non- uniform structures.

To be more specific, the grid structure and size changes the accuracy of the simulation regarding different aspects like numerical dispersion, simulation time or topological correctness of the simulation environment. Figure 1 shows the latter mentioned, sketching a 2D example of a reservoir separated into two compartments by a discontinuity.

A problem created by the discontinuity and boundaries of the model is that a structured uniform grid is not able to capture the geological detail if a coarse grid is used. The structured grid cells will not fit in the discontinuous model, making it on the one hand necessary to neglect some cells and simplify the model or on the other hand to use special gridding techniques.

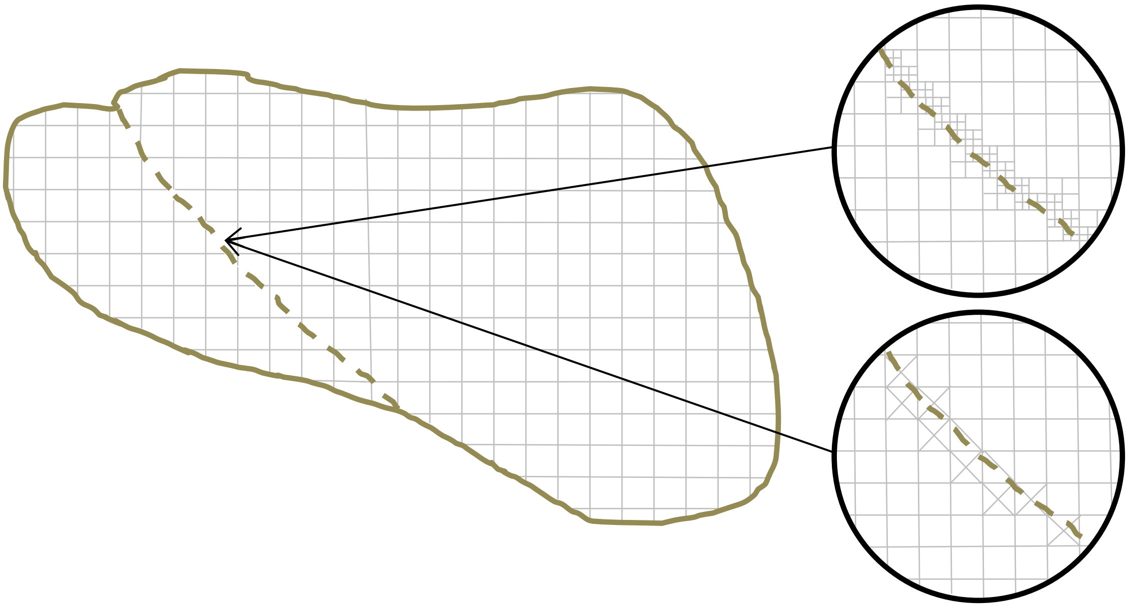


Figure 1: Topology displayed by a non- uniform structured grid and a hybrid grid

One technique would be to refine the grid close to the discontinuous structures to capture more detail. Another possibility would be to use a hybrid grid, utilizing the advantages of both, the structured and unstructured grids. As shown by the figure above, a simple square mesh represents the structured grid. The advantage of it is the low computational expense to calculate flow between the blocks because of the almost perfect center-to-center orthogonality. Unfortunately this fact only holds if a gradually refinement is performed and the aspect ratio is approximately one.

Instead of executing an excessive grid refinement, as done by the first technique, an unstructured grid is used. This allows generating structures that are more complicated, without having a severe grid refinement.

The selection of the unstructured grid depends further on the simulation dimension and on the available computational power. For a two dimensional environment there would be for example the possibility for using triangulated or PEBI grids, which have both advantages and disadvantages regarding computational power or geometrical properties. Particularly the first one mentioned will be very important for this thesis since it will be the main method to convert the structured grid into an unstructured one.

A major problem coming along while using a grid refinement or triangulation is that the initial cell splits into two or more parts. At first sight, this fact might not bother one, but it is essential for further simulation applications. Assuming to use a finite volume discretization scheme implies that all the physical properties, describing the specific simulation region, are in the barycenter of the cell. Furthermore, if considering a two-dimensional domain, a cell would be divided into two parts, each with a new center. Accordingly, the assignment of properties for the generated unstructured grid, based on the initial structured properties, is necessary. Considering that every reservoir has a certain degree of heterogeneity and anisotropy, the new assigned properties should be slightly different. To capture this trend, caused by heterogeneity and anisotropy, various aspects are important for consideration. A few of them would be:

- Point set density available for mapping
- Neighborhood size of point set used for mapping
- Type of mapping algorithm used
    - Deterministic techniques
    - Stochastic techniques

1.2 Scope and Objectives

The objective of this work is to investigate the impact of using triangulated grids instead of uniform and non- uniform structured grids on flow simulation. For this purpose, triangulation of a non- uniform structured grid to an unstructured grid is carried out. Based on the geometric properties of the unstructured grid and the uniform structured grid, the mapping of physical properties, as porosity or permeability, is done. In the end, flow simulations show how the grid structure impacts parameters, like breakthrough time or produced phase volumes.

1.3 Outline

The literature review in chapter 2 gives a brief state of the art overview of the used techniques to achieve the described objectives of chapter 1.2. It is divided into two subchapters describing first, the methods of triangulations and secondly the different possibilities for property mapping.

Chapter 3 discusses the code implementation of the triangulation and mapping algorithm. Associated with that, it provides information about made assumptions during the code implementation and limitations that go with it.

Chapter 4 investigates the setup of the simulation in more detail. Important parts covered by these investigations are for example the placement of wells and the used initial simulation environment to assure comparability for all simulations. Moreover, to get a better understanding of the flow simulations, the used porosity and permeability fields are described closer.

In the end, one can find the results in chapter 5. First, it will point out major findings regarding the different mapping algorithms. Subsequently, it gives a detailed description of all the simulation results regarding the impact of grid structure, the coarsening factor and the placement of wells.

Chapter 2

State of the Art

2.1 Triangulation

The use of triangulations is a vital part in simulation studies. Without them, approaches like finite element discretization would not be imaginable and many engineering problems would be problematic to solve. Although, finite elements discretization is a hardly used technique in reservoir simulation, triangulations have found their way in this field. To be more specific it is necessary to describe complicated structures like faults, irregular features and boundaries and also property anisotropies. Instead of using corner point grids, which have some downsides as explained in chapter 1.1, it is easier to use unstructured grids as triangulated ones. Using a finite volume discretization technique, these grids are already implemented in commercial software. Even though, unstructured grids are already available in commercial simulation software their use is just for small approaches, but not for complete reservoir simulations. The explanation for the limited application of unstructured grids is the higher computational effort and longer simulation runtimes coming along with it. Because of economic reasons, these long runtimes are not accepted by the industry. Therefore, in order to adjust simulation times to a reasonable scale, simplifications on the simulation domain are accepted, which is on the expense of the simulations accuracy.

The following chapters present several possibilities to triangulate structured grids and give an insight on the runtime performance of different methods. Important geometric properties of the generated triangles will be shown. In addition, more advanced approaches of triangulations will be discussed and the corresponding advantages and disadvantages will be depicted.

Generally, if one wants to triangulate a point set, two options exist. First, there is the possibility to execute a triangulation of a finite point set S, where T is the triangulation of S that contains all necessary vertices. Then the sum of all T is the convex hull of the point set S (Cheng, et al., 2013). In the case mentioned above the triangulation would be a point set triangulation. Another possible case that can occur is a polygon triangulation. This is the case if a point set is not a convex polygon point set, but it has a defined convex hull. Figure 2 shows the difference between both cases (Gärtner, et al., 2015).

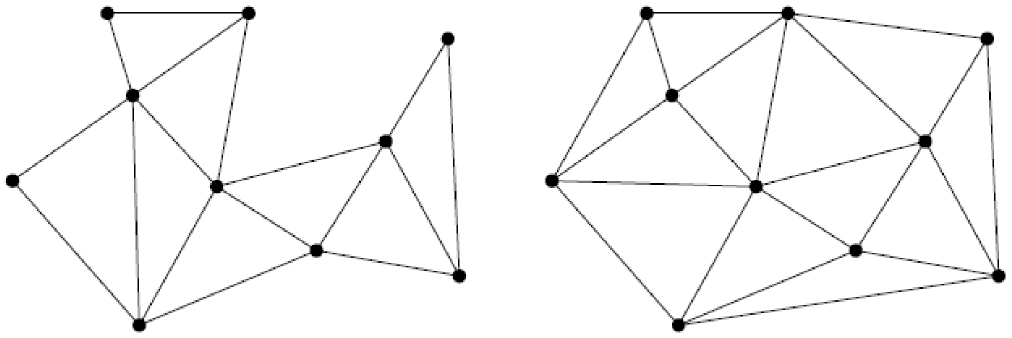


Figure 2: Polygon triangulation (left) and point set triangulation (right) (Gärtner, et al., 2015)

The difference between the two types of triangulations is defined by the difference in convexity. The point set on the left is a so- called non- convex polygon and the point set on the right is a convex polygon. One possibility to describe the expression of convexity is to draw a straight line between two points of the point set. If the straight line is leaving the convex hull to reach the other point of the same polygon, it is a non-convex polygon. Figure 3 shows this procedure.

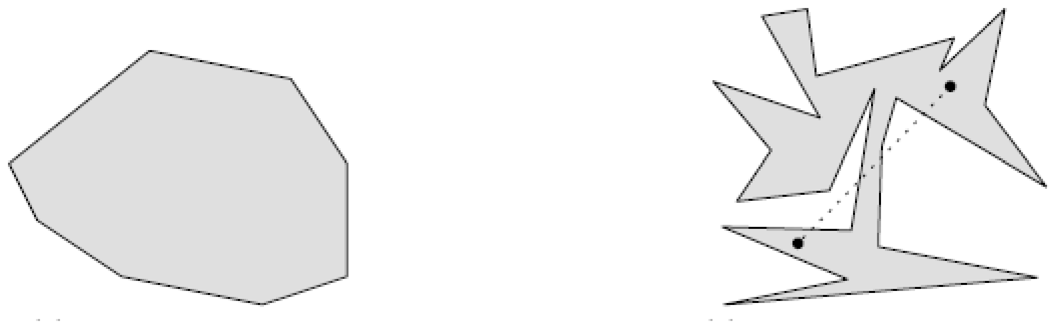


Figure 3: Convex polygon (left) and non-convex polygon (right) (Gärtner, et al., 2015)

In particular, the point set triangulation will be of importance in the course of this thesis. This is the case because the given simulation domain is a convex point set, having no local triangulation planned for the beginning.

2.1.1 Lexicographic Triangulation

The simplest way to generate a triangulation is to use a scan or lexicographic triangulation in which the vertices of the point set S are sorted lexicographically. This can be understood as sorting the vertices bottom- up in x- direction first, and if the x coordinates of two vertices are the same, the y- coordinate is taken into consideration. Subsequent to that, the triangulation is performed, adding point after point of S to the triangulation.

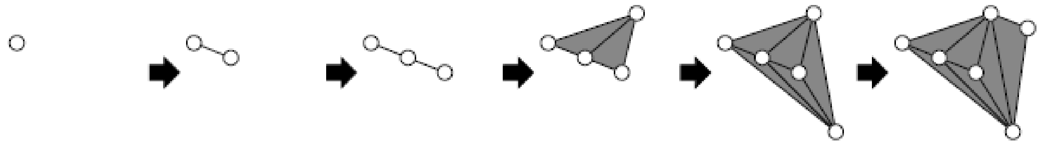


Figure 4: Steps of a lexicographical triangulation (Cheng, et al., 2013)

Another condition to set edges between the vertices is that every newly added point can only be connected with the other vertices if it has a direct connection to it. Simply said, the connection of the new added point must not cross an existing edge of the former triangulated vertices. Figure 4 shows this principle on the basis of a simple example.

Although the algorithm is easily applicable, it is not used for meshing purposes. This is because of its geometric properties as can be seen in the figure below.

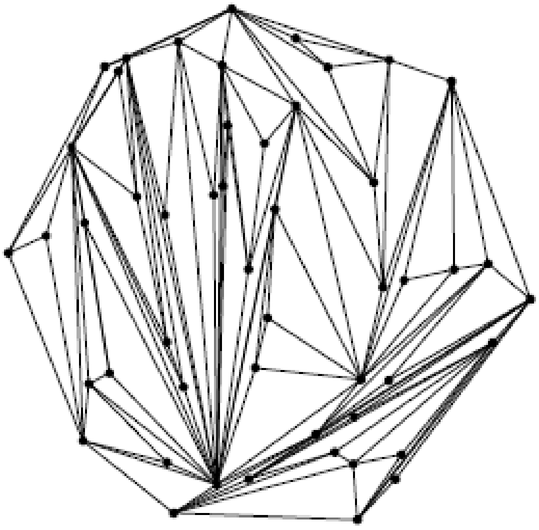


Figure 5: Lexicographic triangulation of a large point set (Gärtner, et al., 2015)

The main disadvantage of this lexicographic triangulation gets clear after having a look at the point set triangulation of the example above. Most of the triangles are not equal- sided triangles, but acute-angled ones. These acute-angles result in long drawn- out triangles, having bad simulation properties. As mentioned before the center of gravity contains the property of a cell. Bearing this in mind, one can imagine what kind of problems are caused for simulation purposes if using a mesh generated by a lexicographic triangulation. The triangle covers a vast spatial domain into one direction but perpendicular to that only a small domain. From this, it follows that for the first case one automatically assumes a constant property and for the latter mentioned properties can change more easily. This might not be the choice for real field scale simulations, in particular for heterogeneous and anisotropic reservoirs. Therefore, equal-sided triangles are favorably used.

For this reason, there is no further consideration of the lexicographic triangulation in the course of this thesis.

2.1.2 Delaunay Triangulation

The most prominent method to triangulate a point set is represented by the Delaunay triangulation. A noteworthy fact to mention is that a Delaunay triangulation is not a special kind of triangulation method. It is rather a collection of different algorithms, yielding triangles, which fulfill certain properties. In general, two basic categories of Delaunay triangulation methods can be distinguished:

- Sequential Delaunay algorithms
- Parallel Delaunay algorithms

Both have their advantages and disadvantages. Principally, a sequential algorithm is much easier to implement compared to a parallel one. From the perspective of the runtime on the other hand, the parallel algorithms are at an advantage compared to the sequential ones. Su, et al. (1994) pointed out that most of the parallel algorithm implementations failed and less of them delivered reasonable performances (Su, 1994).

Since the focus of this thesis does not rely on the runtime of the triangulation, it is not constructive to consider parallel algorithms further. As a result, the focus will be on sequential algorithms for the upcoming chapters.

2.1.2.1 Advantages and Disadvantages

For now, the described triangulations were not satisfying for simulation purposes because of one major disadvantage, the acute angled triangles. This major disadvantage does not concern point set triangulations, generated by Delaunay triangulations. One major advantage of a Delaunay triangulation is that it maximizes the minimum angle of a triangle. This special characteristic delivers triangles, which are almost equally-sided. By considering this characteristic, the generated triangles are as similar as possible, yielding favorable simulation models. (De Loera, et al., 2010) Figure 6 shows the result of a Delaunay triangulation of the same point set as in Figure 5.

Comparing the results delivered by the lexicographic and the Delaunay triangulation, one can perfectly see why the triangles generated by the Delaunay triangulation are more useful for simulation purposes.

Another major advantage of a Delaunay triangulation is it geometrical duality to the Voronoi diagram, which is also a very important and useful grid structure for simulation purposes. (De Loera, et al., 2010)

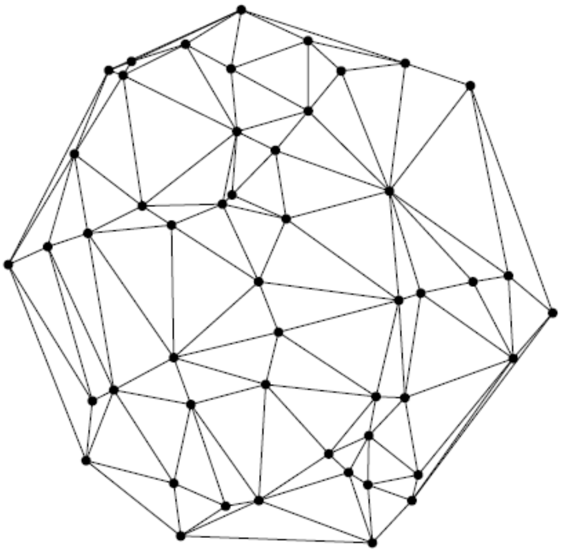


Figure 6: Delaunay triangulation of a large point set (Gärtner, et al., 2015)

A disadvantage of the Delaunay triangulation techniques is that compared to lexicographic triangulation it is harder to implement and requires higher computational power. However, several algorithms are proposed to work properly and a vast literature is available, describing their advantages and disadvantages. The next chapters will give a brief overview on some of these algorithms and highlight their advantages and disadvantages.

2.1.2.2 Peculiarity of Delaunay Triangulations

A remarkable peculiarity in relation with Delaunay triangles is that every circumcircle of a triangle τ, of a triangulation T, in a point set S, contains only the vertices of τ and no other vertices of S. The eponym of the Delaunay triangulation himself, Boris Nikolavevich Delaunay, showed this behavior. To support this hypothesis two major definitions exist.

Definition: 1: In the context of a finite point set S, a triangle τ is strongly Delaunay if its vertices are in S and its closed circumdisk contains no point in S except the vertices of τ. An edge e is strongly Delaunay if its vertices are in S and it has at least one closed circumdisk that contains no point in S except the vertices of e. Every point in S is a strongly Delaunay vertex. (Cheng, et al., 2013)

The second major definition regarding Delaunay triangulations is given by the Delaunay lemma.

Definition: 2: Let T be a triangulation of a point set S. The following three statements are equivalent. (Cheng, et al., 2013)

- *Every triangle in T is Delaunay (i.e. T is Delaunay)*
- *Every edge in T is Delaunay*

- *Every edge in T is locally Delaunay*

Summarizing the Delaunay lemma one can say that one requirement requires the other and if one is not given, the triangulation cannot be Delaunay.

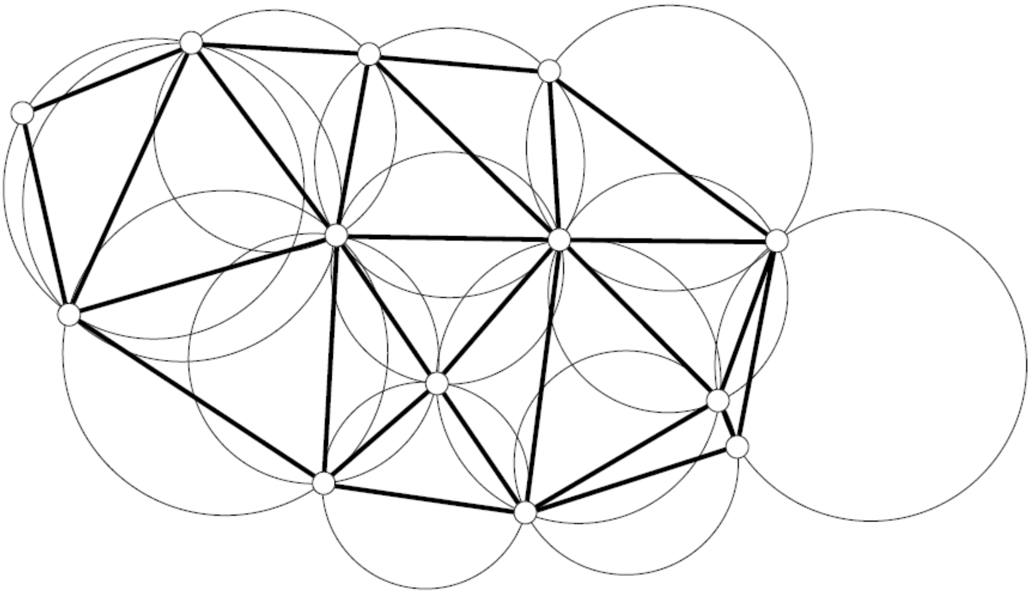


Figure 7: Delaunay triangulation showing the empty circumcircle property (Cheng, et al., 2013)

2.1.2.3 Delaunay Triangulation Algorithms

The previous chapters mentioned the prerequisites that a triangulation is Delaunay and why it is favorable compared to other triangulations like the lexicographical one. Generally, it is possible to generate Delaunay triangles with several different techniques, but some of them have advantages or also disadvantages. Su and Drysdale (1997) give a compact summary of various algorithms, where they divide the main techniques of Delaunay triangulations into five categories: (Su & Drysdale, 1997)

- Divide- and Conquer algorithms
- Sweepline algorithms
- Incremental algorithms
- Gift wrapping algorithms
- Convex hull based algorithms

In general, the runtime of these methods is described by the O notation, which gives a “time” approximation using the total amount of edges in a triangulation. This kind of approximation is necessary because running a triangulation on different computers yields different runtimes because of difference in computational power. As a result, it is reasonable to express a worst-case runtime scenario in dependence of the maximal number of edges in a triangulation. Although this worst-case scenario approximates the runtime, there is no guarantee that the algorithms converge towards it. Depending on the ordering of the point set, literature indicates that most of the time, they are faster than expected (Su, 1994).

In regard of the Divide- and Conquer algorithm the worst- case runtime scenario is $O(n\log(n))$. To achieve this runtime, two different steps are vital. First, a division of the point set into smaller subsets, followed by a triangulation solves the problem locally. Merging the local solutions are representing the second vital step. A problem that remains is that during the merging process some triangles might not be Delaunay. Therefore it necessary to check the empty circumcircle property again and applying a flip algorithm, if it is not fulfilled.

Compared to the Divide- and Conquer algorithms, the Incremental algorithms are representing a worst- case convergence behavior of $O(n^2)$. One can say that in general these algorithms are slower than the ones converging with $O(n\log(n))$, but as literature is pointing out only in particular cases. Furthermore, Su and Drysdale (1997) propose an improved incremental algorithm that has depending on the point distribution a $O(n)$ converging behavior. (Su & Drysdale, 1997)

To conclude the chapter, one can say that most of the state of the art triangulation methods converge against $O(n\log(n))$, leaving behind a broad variety of algorithms that can be implemented for the upcoming challenges in the thesis.

2.1.3 Constrained Delaunay Triangulation

The constrained Delaunay triangulation represents a more advanced possibility to triangulate a point set. In some applications, it is necessary to pre-define a few edges in order to better approximate specific details, as for example a sharp change of reservoir properties. Applying these constraints is the major difference compared to the conventional Delaunay triangulations and brings along a large difficulty. Triangulating a conventional point set without any constraints always has at least one solution that is Delaunay. Using these constraints opens the possibility that some triangles might not fulfill the empty circumcircle property anymore resulting in non-Delaunay triangles. Although, not fulfilling all the Delaunay criteria, still the minimum angles of the triangles are maximized, which results in favorable triangles for simulation purposes (Gärtner, et al., 2015).

Literature gives insight into different kinds of available algorithms, dividing it first into optimal and asymptotically optimal algorithms. Kallmann, et al. (2003) further mentioned that in regard to asymptotically optimal algorithms the Divide- and Conquer and the Sweepline algorithms are available. In the case of optimal algorithms, only the divide and Conquer algorithms are known. (Kallmann, et al., 2003)

The following figure shows a simple example of a constrained Delaunay triangulation.

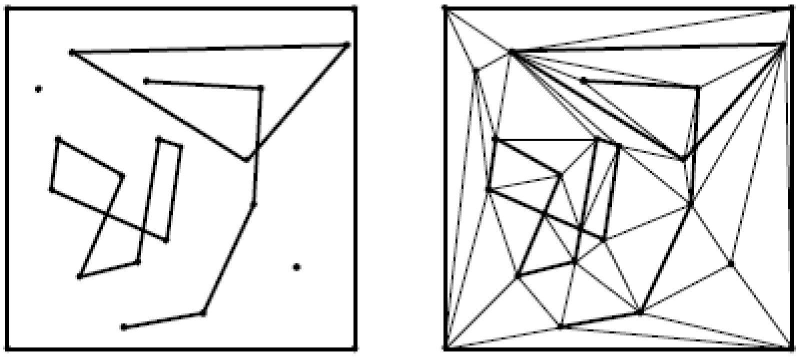


Figure 8: Constrained Delaunay triangulation (Kallmann, et al., 2003)

2.2 Mapping Techniques

Mapping techniques represent another essential part in reservoir simulation. After defining the grid structure, the corresponding parameters for each grid cell are assigned accordingly. Since reservoirs have a large extend and only a small part is known by measurement data, it is necessary to populate the grid cell properties using statistical techniques, known as geo-statistics. The field of geo-statistics provides many different algorithms to map properties from a few measurement points to a complete reservoir model. The two main categories of algorithms are:

- Deterministic algorithms
- Stochastic algorithms

2.2.1 Deterministic Algorithms

A major advantage if using deterministic methods is that the value of an undefined cell is calculated according to a specific equation. This does not rise the necessity of implementing data distributions and random sampling of values during the calculation of the properties. In particular, the result generated via a deterministic algorithm is always the same and does not change if the same data is run with the same boundary conditions. Two of the most prominent algorithms regarding deterministic methods are:

- Kriging
- Inverse Distance Weighting

2.2.1.1 Kriging

Kriging is one of the most applied and well-known mapping algorithms in geo modelling, supported by a vast literature. To cover a broad spectrum of engineering problems, several derivations of the Kriging algorithm were developed including:

- Simple Kriging
- Ordinary Kriging
- External Drift Kriging
- Co-Kriging

For the sake of compactness, not all of the Kriging algorithms are explained in the following, but the most important one regarding this thesis. The chosen algorithm is the ordinary Kriging, which uses a mean value as a basis for modification by the neighboring grid block values as shown below.

$$x_0 = m_{PS} + \sum_{i=1}^{n} w_i(x_i - m_{PS}) \tag{2.1}$$

x_0 .... *Calculated property*

m_{PS} .... *Mean value of nearest neighborhood*

w_i .... *Weightings of proximate cell values*

x_i .... *Proximate cell value*

As shown by the equation above, a mean value that is re-estimated for each cell property calculation is modified by the weighted sum of the proximity cells. Juxtaposed to the simple Kriging, one calculates the mean value differently. More precisely, it is calculated by the properties in the nearest neighborhood that is prior defined by variograms. Figure 9 shows the search radius for the nearest neighbor properties.

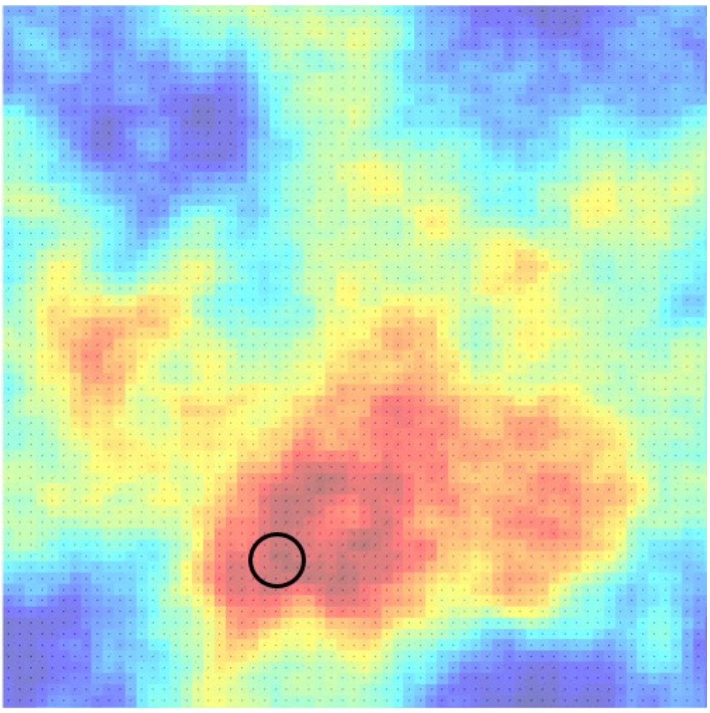


Figure 9: Property field with corresponding cell centroids

The next vital factor in order to calculate the property of a new cell is the weighting vector. It contains the weightings for the proximate cell values and describes their influence on the newly calculated cell. To calculate these weightings one can define variograms that describe the spatial correlation of the data. After deriving the spatial correlation and its corresponding correlation values, a system of linear equations of the form Ax = b can be derived.

$$\boldsymbol{Cov}\,\vec{w} = \overrightarrow{Cov_0} \tag{2.2}$$

$\boldsymbol{Cov}$ .... *Covariance matrix*

$\vec{w}$ .... *Weighting vector*

$\overrightarrow{Cov_0}$ .... *Covariance vector in relation to the new cell*

Equation (2.2) provides a possibility to determine the weightings for equation (2.1). By the pre-definition of variograms, one can calculate the covariance by the following equations (Doyen, 2007):

- Spherical

$$C\left(\frac{h}{a}\right) = \begin{cases} 1 - \dfrac{3h}{2a} + \dfrac{1}{2}\left(\dfrac{h}{a}\right)^3 & if\ h\ \leq a \\ 0 & if\ h > a \end{cases} \tag{2.3}$$

- Exponential

$$C\left(\frac{h}{a}\right) = exp\left(-\frac{3h}{a}\right) \tag{2.4}$$

- Gaussian

$$C\left(\frac{h}{a}\right) = \exp\left(-\frac{3h^2}{a^2}\right) \tag{2.5}$$

C .... *Calculated covariance*

h .... *Center to center distance of cells*

a .... *Range of variogram*

Equations (2.3) to (2.5) represent normalized parametric equations that create covariances in a range between 0 and 1. In order to scale them to the correct size the sill of the derived variograms and the nugget effect can be used.

So far, variograms were just mentioned as tools to define the range of spatial correlation and to determine the similarity of the different points to each other. In the following, a brief description of the different aspects of variograms shows their construction and their main parameters.

In general, one can decide between experimental and parametric variograms, which are both important in the upcoming description. First, an experimental variogram is constructed using points that are sampled from different lag distances. Next, by plotting the average square difference of the sampled points against the distance h, one yields the variogram (Doyen, 2007).

$$\gamma^*_{(h)} = \frac{1}{2N_{(h)}} \sum_{i=1}^{N_{(h)}} (x_{i+h} - x_i)^2 \tag{2.6}$$

$\gamma^*_{(h)}$ .... *Dissimilarity of point set to origin*

$N_{(h)}$ .... *Number of points in sampled point set in distance h*

x_{i+h} .... *Property of point at distance i+h*

x_i .... *Property of point at origin*

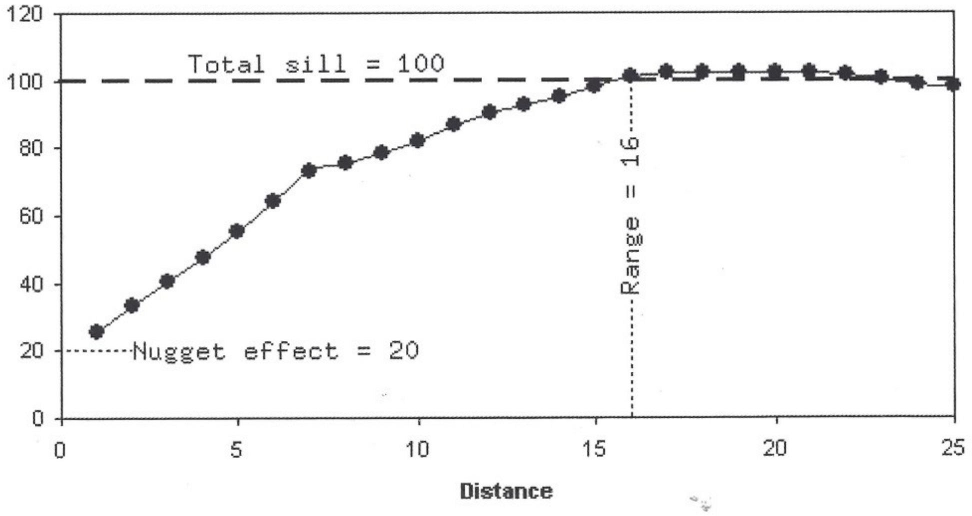


Figure 10: Experimental variogram with various points of dissimilarity at distance h (Doyen, 2007)

The figure above shows an experimental variogram and its corresponding basic parameters. The first parameter is the sill value that can be found at the plateau of the experimental curve. It indicates the dissimilarity of the origin and the different point sets in distance h and is an important factor for calculating the covariances afterwards. The next important factor is the range of the variogram. It is defined by the start of the plateau and gives an indication for the maximal correlation rage of the data. The last important parameter is the nugget effect. Its basic statement is that the data at the origin also has a certain degree of dissimilarity, due to measurement errors and averaging.

After defining the parameters, a parametric model is used to approximate the experimental data. The following three equations are an example for parametric models in regard of variogram fitting.

- Spherical

$$\gamma\left(\frac{h}{a}\right) = \begin{cases} \left[1 - \left(1 - \frac{3h}{2a} + \frac{1}{2}\left(\frac{h}{a}\right)^3\right)\right] \mathrm{S} + \mathrm{N} & if\ h\ \leq a \\ \mathrm{S} + \mathrm{N} & if\ h > a \end{cases} \tag{2.7}$$

- Exponential

$$\gamma\left(\frac{h}{a}\right) = \left[1 - exp\left(-\frac{3h}{a}\right)\right] S + N \tag{2.8}$$

- Gaussian

$$\gamma\left(\frac{h}{a}\right) = \left[1 - \exp\left(-\frac{3h^2}{a^2}\right)\right] S + N \tag{2.9}$$

S .... *Sill*

N .... *Nugget*

A more precise look at equations (2.7) to (2.9) shows a similarity between equations (2.3) to (2.5). This similarity is because variograms are a measurement of dissimilarity and covariances measure the similarity of data. Both, similarity and dissimilarity are connected to each other. The following equation provides the connection between covariance and variogram:

$$\gamma_{(h)} = C_{(0)} - C_{(h)} \tag{2.10}$$

$\gamma_{(h)}$ .... *Value of dissimilarity at distance h*

$C_{(0)}$ .... *Value of similarity at origin (1 if using normalized models)*

$C_{(h)}$ .... *Value of similarity at distance h*

The final answer why one should use a parametric model is manifold. First, if using an experimental model for defining unknown regions that are far away from the origin, data might not be defined anymore and no value can be determined. On the other hand if using a parametric model, one is more flexible because the certain value is calculated via the parametric equation. A second reason for using parametric models is the construction of the covariance matrix. For the necessary matrix inversion to calculate the weightings, all values need to be positive. Using an experimental model might not produce just positive values, which is problematic for the algorithm implementation (Doyen, 2007).

The following figure shows a fitted parametric model to experimental data.

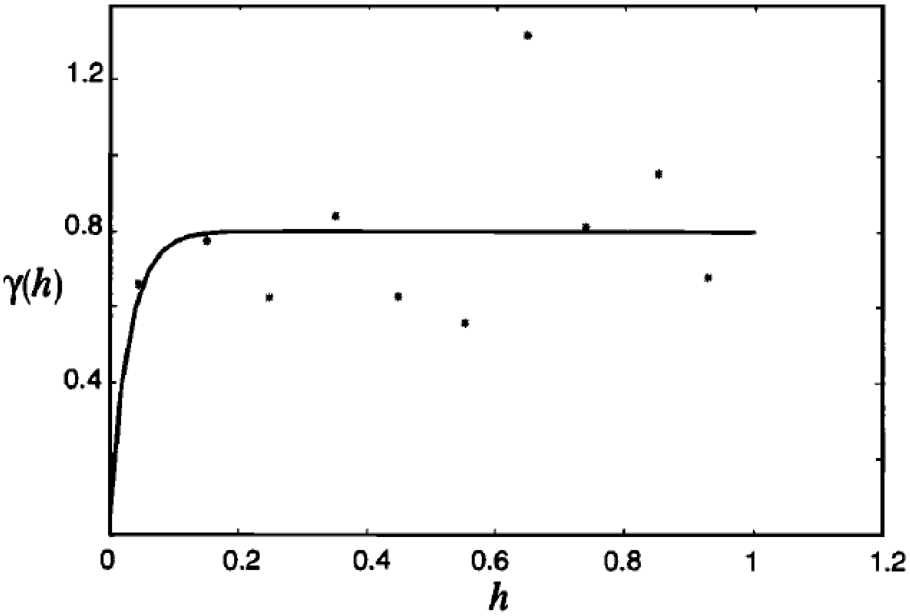


Figure 11: Parametric variogram fitted to experimental data (Kitandis, 1997)

2.2.1.2 Inverse Distance Weighting

A simpler deterministic mapping algorithm is known as the inverse distance weighting. This algorithm does not take any spatial correlation of data into account. The advantage of this algorithm is that no variograms or other correlation techniques are necessary, but the downside of this is that one cannot account for any directional behavior, like anisotropies.

To determine the value of a new cell, simply the neighboring cells are used. One can do this by applying weights calculated due to their distance, putting the largest weights on the closest data points. This simplicity might also be one of the greatest disadvantages. As mentioned, the algorithm does not consider any directional behavior. For example, having four measurement points in every direction and at the same distance from the origin, the same weighting for each point will be used, even if there is no correlation in this direction (Webster & Oliver, 2007).

$$x_0 = \sum_{i=1}^{n} x_i w_i \tag{2.11}$$

With:

$$w_i = \frac{1}{|c_i - c_0|^{\beta}} \qquad with\, \beta > 0 \tag{2.12}$$

c_i .... *Coordinate of neighboring point*

c_0................ *Coordinate of point of interest*

β .... *Inversely weighting factor (Common value is 2)*

2.2.2 Stochastic Algorithms

Stochastic algorithms represent the second major type of mapping techniques. When juxtaposing deterministic and stochastic algorithms, one can say that stochastic methods never produce the same result if using the same data. Nevertheless, each of the results are from a mathematical point of view valid. Although these algorithms provide advantages regarding heterogeneity, not all reservoir properties should be mapped using an algorithm of this origin. One of these properties is for example the water saturation. Common reservoir engineering standard is that this property is initialized using capillary pressure measurements or by utilizing Leverett J-functions. Both methods assure an equilibrated system, which would not be the case if using a stochastic approach.

One of the simplest and widespread methods of stochastic approaches is the Sequential Gaussian Simulation (SGS). It utilizes the result of the deterministic Kriging approach and further uses the property distribution to randomly assign a value within the variance range. To be more specific the following steps have to be fulfilled:

- Randomly pick a non-simulated cell
- Calculation of the Kriging result x_i^{SK}
- Calculation of variance of the Kriging result within the used point set $\sigma_{i,SK}$
- Put Kriging result and variance borders in distribution curve
- Randomly select one value within the borders in the distribution curve
- Assign the picked value from the distribution curve to grid
- Use newly calculated value as additional control point for further calculations
- Start from first point until all cells have an assigned value

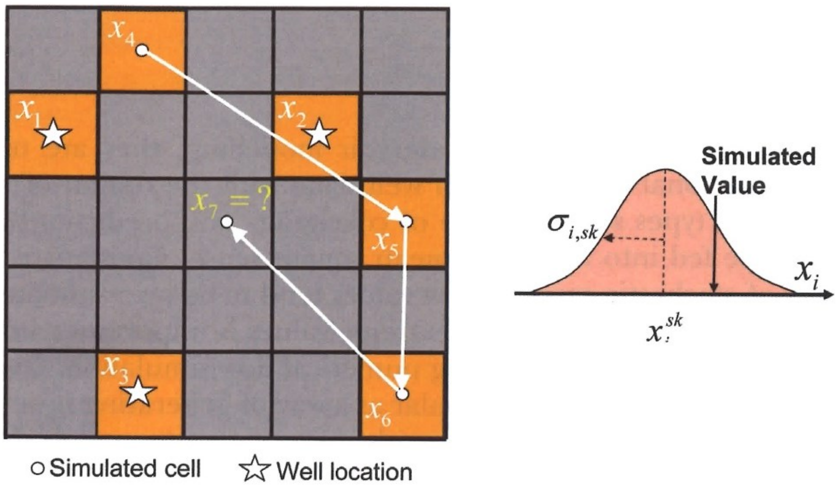


Figure 12: Sequential Gaussian Simulation (Doyen, 2007)

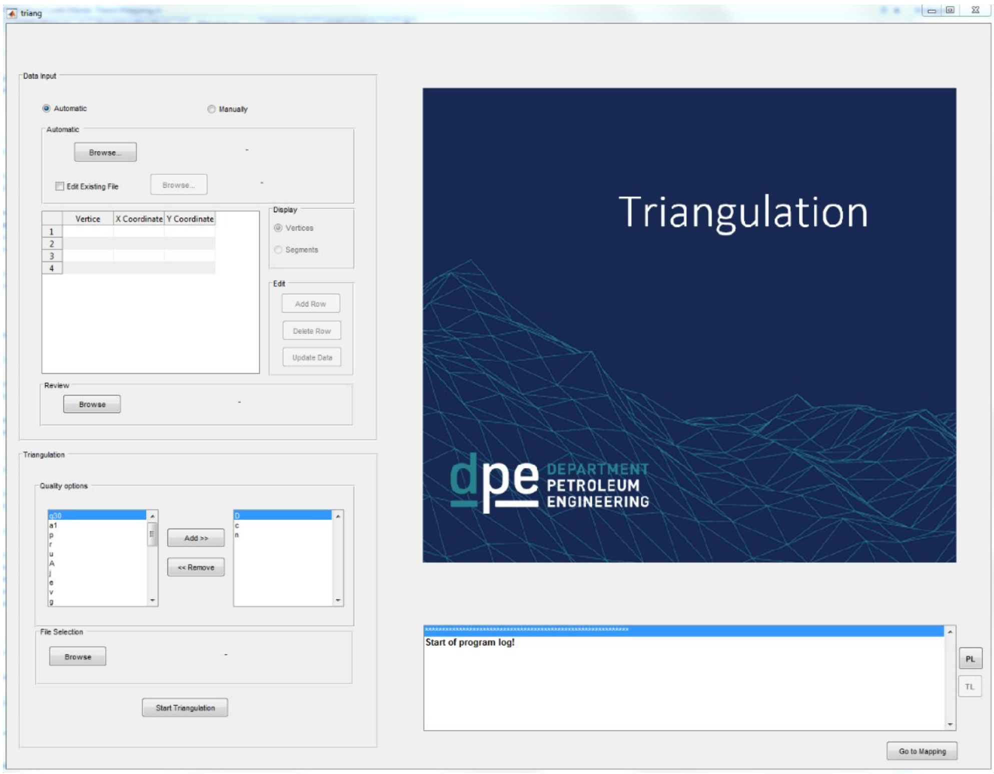
Another remarkable point to mention in regard to SGS is its dependence on the Gaussian distribution of data. If the properties for mapping are differently distributed, as for example bimodal, one has to transform this first to a Gaussian distribution. This can be achieved by a normal score transform. Consequently, the transformation has to be inserted in the algorithm above. Furthermore, a back transformation of the data has to be computed after a value is assigned to the cell, resulting in a tremendous increase in computational cost.

Chapter 3

Code Implementation

The previous chapters gave an outlook on the content of the thesis and an overview of the state of the art algorithms. The following chapter will give more detail regarding the implementation of the theory. Then, the capabilities of the code and all its corresponding limitations are discussed in more detail.

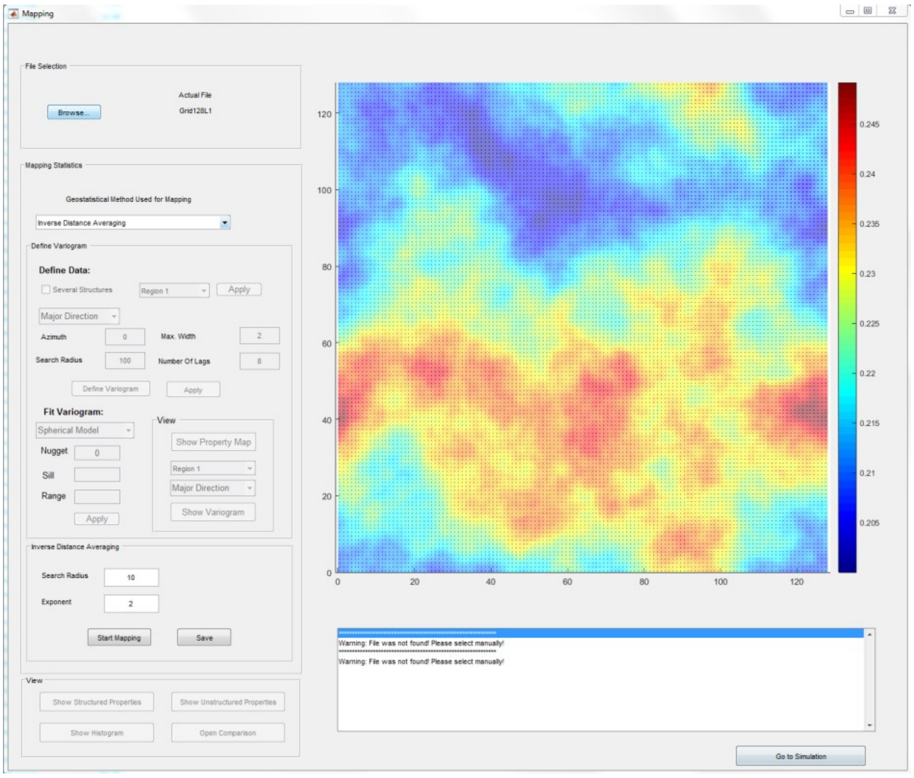
The code developed in the course of this thesis consists of three main windows. When running the startup file of the thesis, the first window opens, providing different options for triangulation of point sets (Figure 13).



```
************************************************************
Start of program log!
```

Figure 13: Initial triangulation window

After triangulating the selected point set, with its corresponding quality parameters one can proceed with the mapping window. This is done by clicking on the “Go to Mapping” button.



Mapping

File Selection

Actual File: Grid128L1

Mapping Statistics

Geostatistical Method Used for Mapping: Inverse Distance Averaging

Define Variogram

Define Data:

□ Several Structures — Region 1

Major Direction

Azimuth: 0

Max. Width: 2

Search Radius: 100

Number Of Lags: 8

Fit Variogram:

Spherical Model

Nugget: 0

Sill:

Range:

View: Region 1 — Major Direction

Inverse Distance Averaging

Search Radius: 10

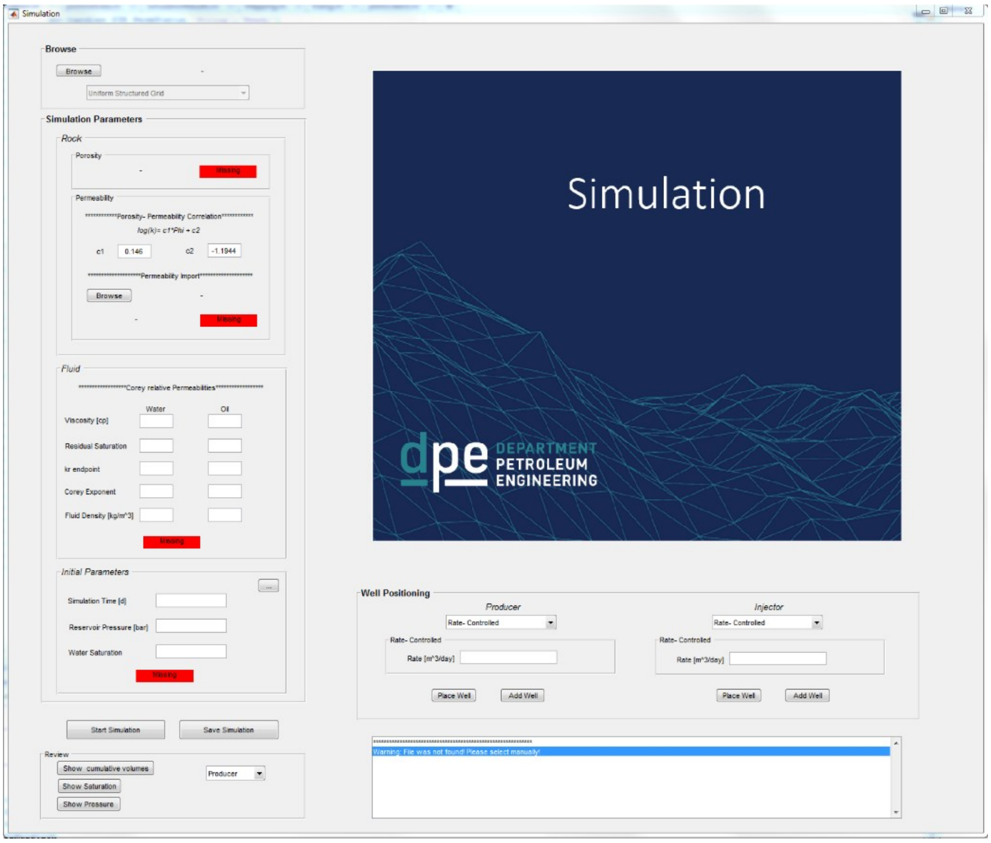
Exponent: 2

```
*************************************************
Warning: File was not found! Please select manually!
*************************************************
Warning: File was not found! Please select manually!
```

Figure 14: Initial mapping window

While executing the “Go to Mapping” button, the code automatically loads all information and displays the corresponding property field (Figure 14). Using a drop down menu one can use the desired mapping algorithm and fine-tune a variety of parameters. Another possibility is to save the final solution of the mapped file and reload it by using the browse function. The simulation window can then be opened by clicking on the “Go to Simulation” button.

When opening the simulation window (Figure 15), an algorithm performs an automatic setup of the reservoir rock according to the MRST standards. One can change the generated rock afterwards by browsing for a new file or uploading new permeability fields manually. Depending on the available and already implemented information, the window returns a status of the four vital parameters in order to start a simulation. Unless every parameter section changed from red to green light, the simulation cannot start. Results can afterwards be evaluated by an additional results window.



Uniform Structured Grid

************Porosity- Permeability Correlation************

*log(k)= c1*Phi + c2*

c1 0.146 c2 -1.1944

********************Permeability Import********************

******************Corey relative Permeabilities******************

Producer

Rate- Controlled

Rate- Controlled

```
Warning: File was not found! Please select manually!
```

Figure 15: Initial Simulation window

3.1 Triangulation

The basis of all triangulations regarding this thesis is a structured non- uniform grid (Figure 16) generated using a wavelet-based transformation algorithm on a fine structured grid. Depending on the applied coarsening of the algorithm, the cells of the initial fine structured grid are combined to larger cells.

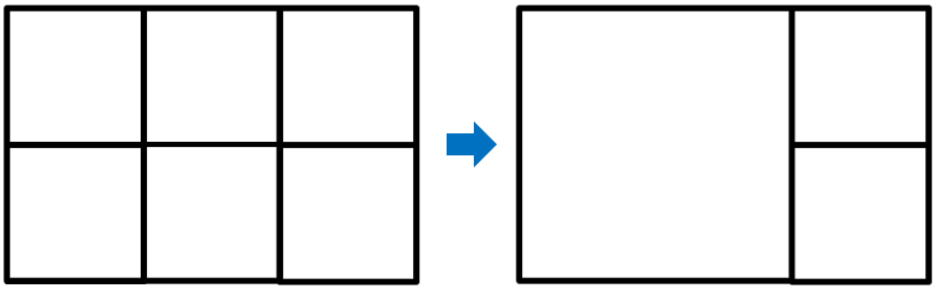


Figure 16: Transformation of structured uniform grid to structured non- uniform grid

Table 1 shows an explanation of the different coarsening factors for each data file used.

Table 1: Coarsening factors of each data file with the corresponding number of grid blocks

	Number of grid blocks		
Dataset	**Structured uniform grid**	**Structured non-uniform grid**	**Triangulated grid**
Grid64L1	4096	1543	6490
Grid64L2	4096	1051	4850
Grid128L1	16384	4972	25160
Grid128L2	16384	2383	16530
GridSPE10L1	13200	3561	19974
GridSPE10L2	13200	1440	12904

Although, upscaling is a necessary process in the simulation workflow, it underlies certain limitations. One of them is for example the aspect ratio between neighboring grid blocks. Having a severe difference in size, might be problematic for the simulation solver, resulting in convergence problems. Therefore, if a change of grid size is applied via the coarsening algorithm, a gradual change of the grid blocks is favorable and a prerequisite for further modifications. Figure 17 shows both, an acceptable coarsening and a non-acceptable one.

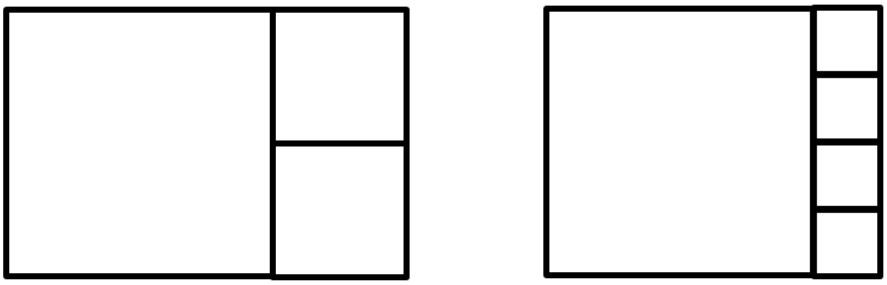


Figure 17: Acceptable aspect ratio (left) and not acceptable aspect ratio (right)

Considering all prerequisites for simulation purposes, triangulation can be done on structured non- uniform grids. In the course of this thesis, it was decided to go for an open source software, instead of using the implemented MATLAB function or a self- coded algorithm. The used software is called “Triangle” and was developed by Jonathan Richard Shewchuk, who did an extensive research on 2D mesh generation. During its development, he introduced several algorithms of different nature and compared them in regard to robustness and runtime performance. In the end he came to the same conclusion as Su and Drysdale that the divide and conquer algorithm with alternating cuts outperformed the other algorithms and implemented it in Triangle. Furthermore, Triangle also provides Rupert’s mesh refinement algorithm that allows to insert further vertices into a point set automatically in order to guarantee for certain quality characteristics, as for example the smallest angle in a triangulation (Shewchuk, 1996).

For all these reasons, the decision fell on the use of Triangle instead of other software. To simplify its usage, its code was compiled and converted into an executable file that can be called by MATLAB. For further simplification a user interface was developed that allows processing the raw data by Triangle and lists all available options. Figure 13 presents the initial appearance of the triangulation window. It is divided into two sections, the data section and the triangulation section, respectively.

The data input section provides the possibility to browse for pre-defined data files or an option to enter raw data manually. Moreover, the imported data can be further modified by inserting vertices or segments as constraints. After applying all modifications, the data is saved in a ".poly" file that is needed for the following triangulation.

The second section provides the triangulation options. Before triangulation starts, one has to define quality parameters. Depending on the used parameters, the result can vary in regard to minimum angles, areas or number of total vertices. The default selection of quality parameters is as follows:

- D: Construct a Delaunay triangulation out of the entered ".poly" file
- c: Compute the convex hull of the entered point set
- n: Include a ".neigh" file in the program output

One should not remove a parameter of the provided default selection without further knowledge of the provided code. Although removing parameters can be problematic, it can be of advantage to add further quality parameters. One of them might be the minimum angle criteria q_. It defines the minimum allowed angle of all triangles in the triangulation and assures that all angles are greater than the specified one. Another parameter of interest is a_. It defines the maximal area for each triangle and inserts additional vertices in order to fulfill the limitation. The user interface lists all quality parameters and gives detailed information by right- clicking the certain property.

The disadvantage coming along if using Triangle is that it only allows triangulation of planar point sets. Triangle does not provide the option to triangulate 3D problems, which makes it necessary to use different software. For the further progress of this thesis, this disadvantage was not problematic since it is also limited to 2D problems.

As a result, future work regarding this thesis is to modify the code further that also 3D point sets can be processed. One way to achieve this goal might be to use MATLAB's built in triangulation function that does not provide as much quality criteria as Triangle does, but provides the option to process 3D point sets.

Another interesting development for future work might be an automatic constraint function. To be more specific the triangulation of square cells, as it is the case for the used data sets, always yields two possible solutions for a Delaunay triangulation. Figure 18 shows a sketch for one triangulated cell and its solutions.

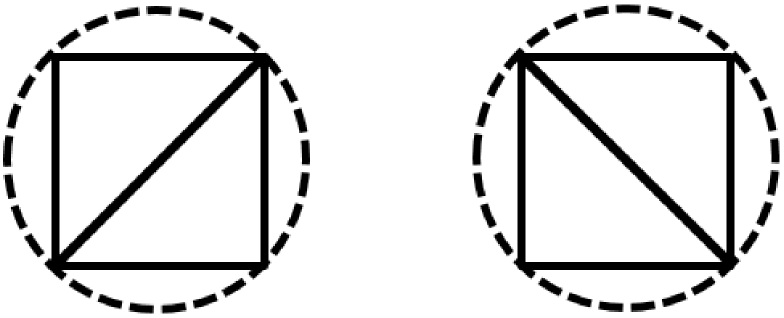


Figure 18: Delaunay triangulations of equal quality

From a geometrical point of view, both triangles are equivalent, but from different aspects, one triangulation might be better than the other one. One way to determine which triangulation is better is to utilize the fine property field of the initial structured grid. Using derivatives one might be able to detect sharp property changes. These property changes should be accounted for by pre-defining edges, perpendicular to the change.

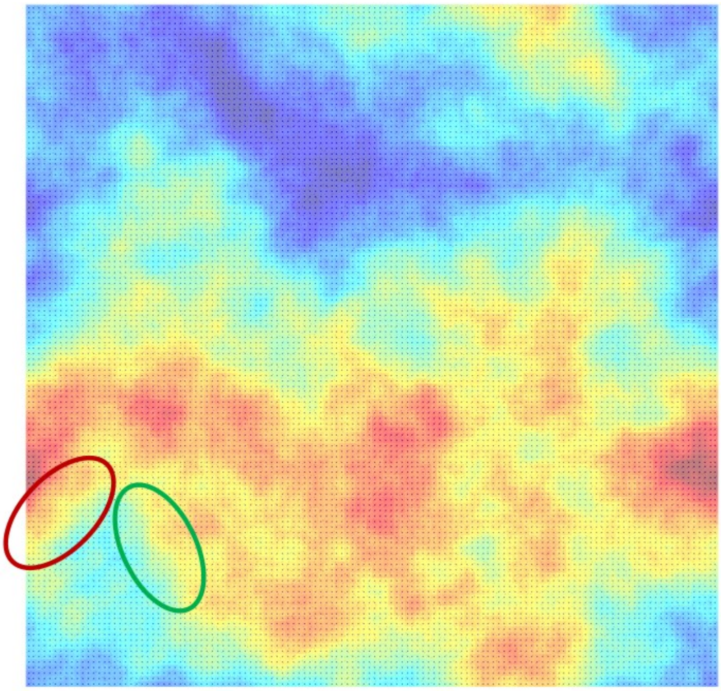


Figure 19: Automatic pre-definition of edges

As shown by the figure above, the left triangulation of Figure 18 might be preferable for the red area, whereas the right triangulation of Figure 18 might be better for the green area.

3.2 Mapping

The second main window offered by the developed code opens the possibility to map the initial grid properties to the previously generated triangulated mesh. One can imagine a scenario as shown below by Figure 20.

Combining both, the geometric data of the triangulated grid and the properties of the initial fine structured grid, and using them as input data, yields the triangulated and mapped grid. Important factors for property mapping are the centroids of triangles and of the property grid blocks. Depending on their spatial distribution, mapping algorithms will assign new properties to the triangulated grid. Putting both, the triangulated grid and the fine map on top of each other, one can see the importance of the centroids.

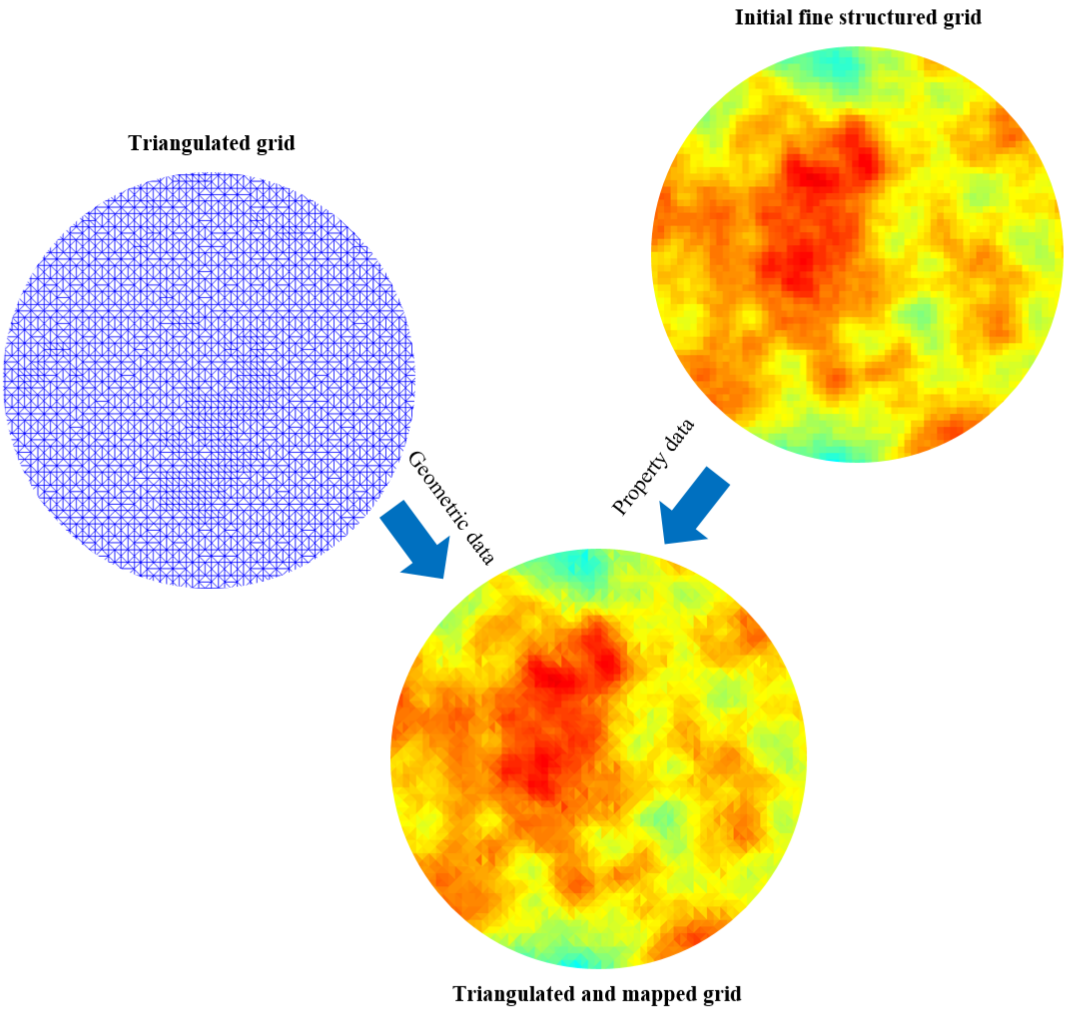


Figure 20: Mapping workflow

As Figure 21 shows, the properties of the triangles should be slightly different from the initial properties of the structured fine grid because of their different location. The used mapping algorithm decides how this difference is weighted. A Kriging algorithm utilizes for example the center-to-center distance to calculate weightings based on spatial correlation ranges. Compared to that an inverse distance weighting is easier to apply since it is based only on the center-to-center distances. Nevertheless, regardless which algorithm is used, the geometric connection of the fine structured grid and the triangulated one is important because it is one of the key factors for assigning the new properties.

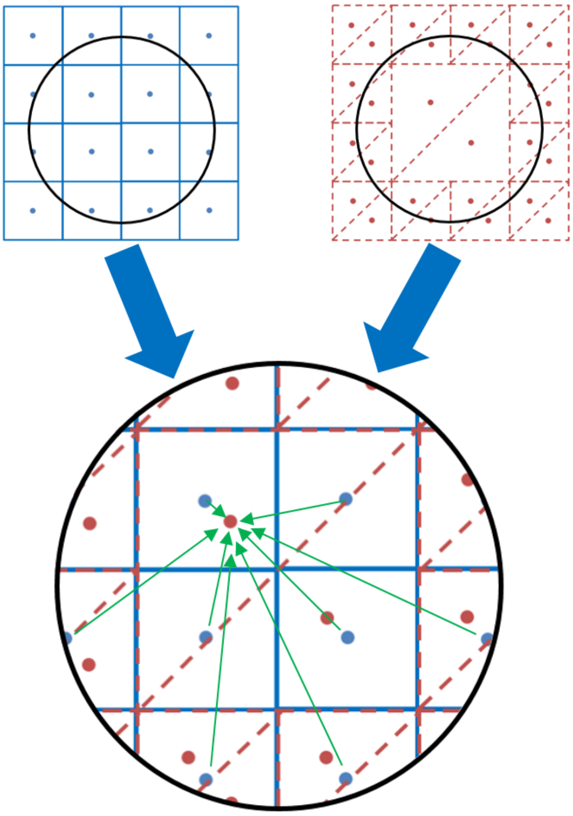


Figure 21: Mapping scheme

In general, the code offers the possibility to map properties using the Inverse Distance Weighting and the Kriging algorithm. Both algorithms are deterministic methods and calculate a cell property explicitly, not randomly. The reason for choosing only a deterministic approach is that the fine grid already determines the property structure. Since it is assumed that the initially defined structure represents reality, it should not be biased by a random distribution applied through stochastic approaches. In the following subchapters, all required parameters and assumptions made to implement the algorithms are a matter of discussion.

3.2.1 Inverse Distance Weighting

To initialize the inverse distance weighting algorithm, one has to import a triangulated file, using the provided browse button in the file section panel. After successfully loading the file, two basic parameters for executing the mapping have to be defined. The search radius for defining proximate points of the initial point set is the first parameter of importance. Depending on the point set density, one should choose a radius that sufficient points are available for the mapping process. Nevertheless, if the radius is picked too large and the point set around the point of interest is sparse, the approximation of the new points might not be correct. Therefore, large search radii combined with sparse data sets, are not a good initial condition if using an inverse distance weighting method. The second parameter of importance is the weighting exponent. A standard is to pick a value of 2, which inversely weights the points as the square of distance.

The code provides a default input for both parameters, using a radius of 10 meters for the definition of the closest points and a value of 2 for the inverse weighting.

3.2.2 Kriging

The initialization of the Kriging algorithm starts by selecting a triangulated file, using the browse button. After importing the necessary data, one has to use the drop down menu to select the Kriging option. Selecting the Kriging option enables the “Define Variogram” panel that provides a variety of options.

The first option provided, is the choice to select between a single structure or a combination of several structures. Thinking of a model in reservoir scale, there will always be several structures involved. One can detect these features by a significant change of the properties magnitude or a change of the directional behavior. To account for these sharp changes, a state of the art method is to use rock typing and introduce different regions.

To account for several regions with the provided code, the user interface displays the initial property distribution that allows seeing these changes. By enabling the several regions option, one activates a brush tool that provides the ability to define the region type by colorizing the displayed property map. Figure 22 shows two examples of regional anisotropy.

The figure on the left shows a single anisotropy with a directional behavior from NW to SE. Compared to that the figure on the right shows two anisotropies having a directional behavior from NW to SE and from NE to SW. In addition, also a significant change in the magnitude of the property’s value can be observed. To account for this significant change, one should apply the previously described option.

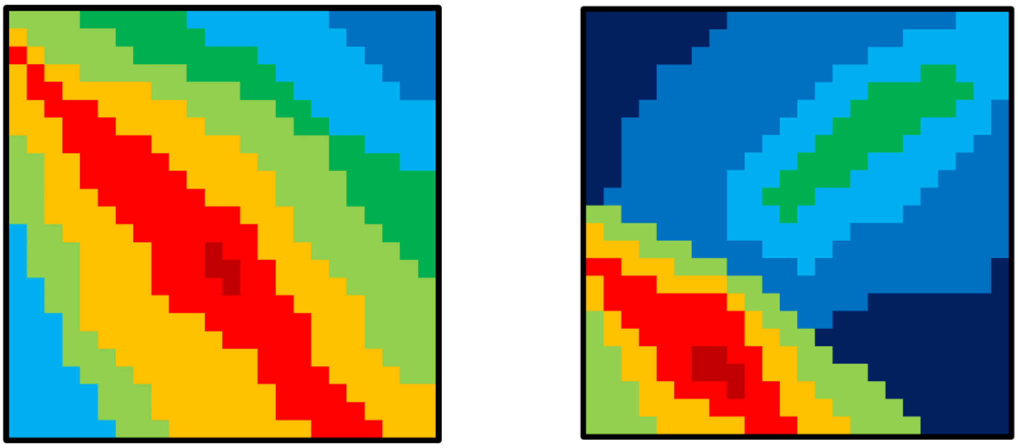


Figure 22: Regional anisotropy

The next important functionality is the distinction between the principle directions of the anisotropy. Considering a three-dimensional case one can distinguish between three different principal directions, which are major, minor and vertical direction. Accordingly, a two-dimensional problem has only two principal directions to fit, which are major and minor direction for a horizontal plane. To fit a direction one has to look at the initial property map and place a search area with the corresponding lag distances. This can be achieved by providing the four basic parameters that define the search area. Simply clicking on the property map after defining the basic parameters defines the search area as it is shown by the following figure.

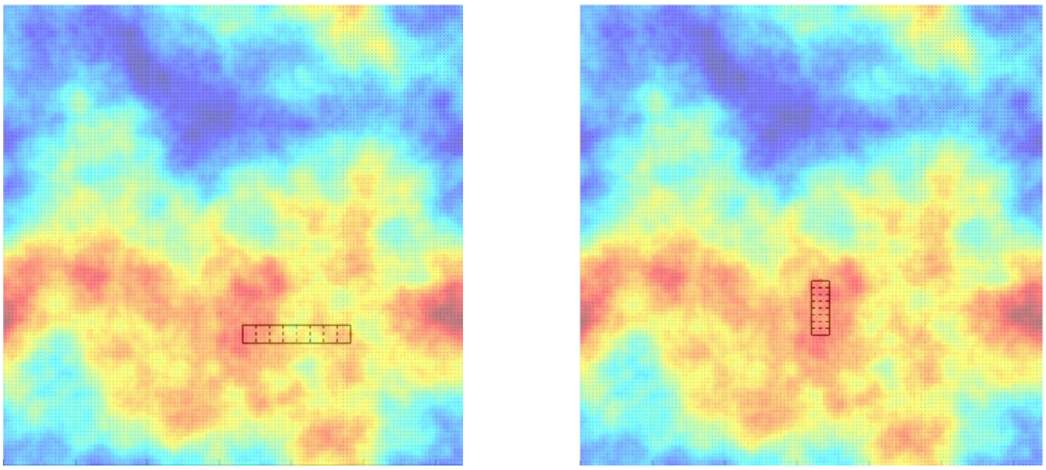


Figure 23: Variogram search area in major direction (left) and minor direction (right)

The correct parameters that define the search area and its placement are fully dependent on the judgement of the operator. In the figure above one can see a clear directional behavior from west to east making it to the major direction. Perpendicular to it one places the minor direction with slightly different parameters for the search area.

After placing the search area, the apply button calculates the experimental variogram using the point sets within the different lag distances. Based on these experimental points of the variogram a parametric model is fitted for the Kriging algorithm afterwards (Figure 24). Depending on the experimental variogram, one can decide between a Spherical, Gaussian or Exponential parametric model as shown by equations (2.7) to (2.9). The sill and range values allow scaling the variogram to the correct size. The only factor that must not be changed is the nugget effect. This can be explained by the quality of the point set that is used for this thesis. Normally, Kriging and the corresponding variograms are applied on data sets that are sparse and have several measurements at one grid cell. Compared to that the data sets for this thesis are dense and have only one value per cell. As a result if having only one value per cell, no inaccuracy of the data can be assumed, resulting in a nugget effect of 0.

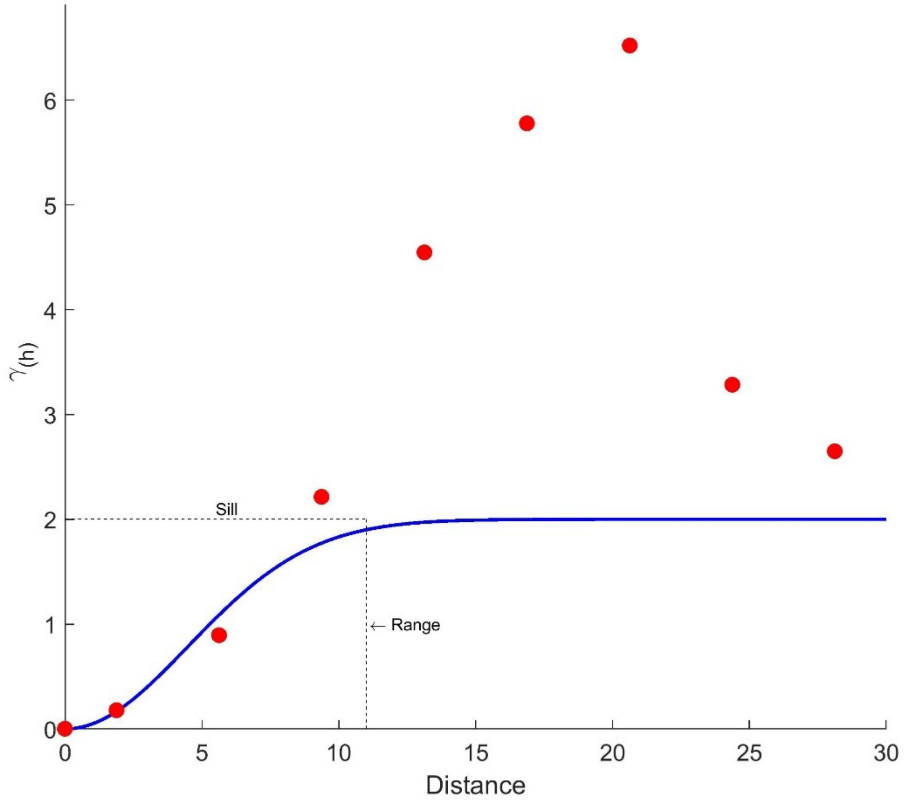


Figure 24: Fitted parametric model to experimental variogram

The next step is to calculate the cell values by utilizing the defined variograms for each region. First, the implemented algorithm looks up all proximate cell values depending on the correlation range of the major and minor direction variogram. In this case, it becomes obvious that the correlation length of both might not be the same. Therefore, the search area is not represented by a circle as shown by Figure 9, but by an ellipse with a major axis having the correlation length of the major variogram and a minor axis having the correlation length of the minor variogram. Another important fact is the pointing direction of the ellipse. Depending on the fixed azimuth of the major and minor axis, the ellipse will also rotate as shown by Figure 25.

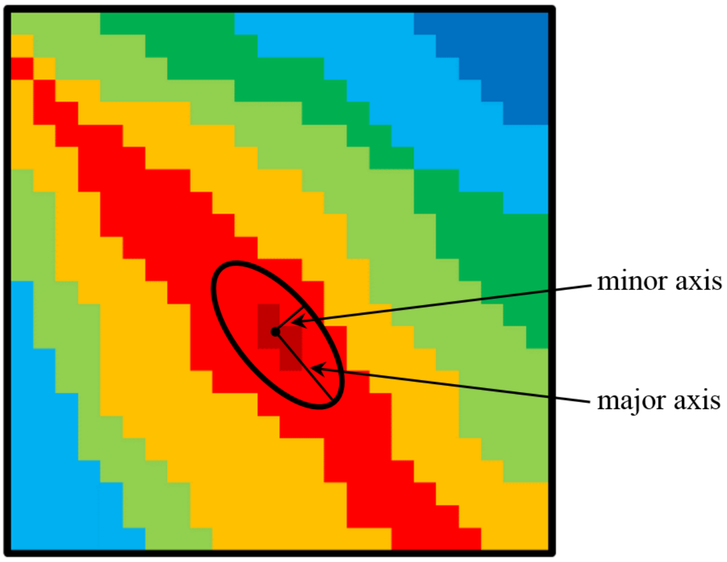


Figure 25: Rotation of search ellipse

The equation used to define the point set within the elliptic search area is derived by the following relationship:

$$\frac{x'^2}{a^2} + \frac{y'^2}{b^2} = 1 \tag{3.1}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix} \begin{pmatrix} x - x_m \\ y - y_m \end{pmatrix} \tag{3.2}$$

$$\frac{[\cos(\alpha)\,(x - x_m) - \sin(\alpha)(y - y_m)]^2}{a^2} + \frac{[\sin(\alpha)\,(x - x_m) + \cos(\alpha)(y - y_m)]^2}{b^2} - 1 = 0 \tag{3.3}$$

x' .... *Rotated x- coordinate*

y' .... *Rotated y- coordinate*

x.................. *x- coordinate of research point*

y.................. *y- coordinate of research point*

x_m .... *x- coordinate of ellipse center (point of interest)*

y_m............... *y- coordinate of ellipse center (point of interest)*

a .... *Major axis*

b .... *Minor axis*

α................. *Azimuth*

Inserting equation (3.2) in equation (3.1) yields (3.3). This equation describes if a point (x,y) is within an ellipse rotated by azimuth α. If the result of equation (3.3) is smaller than 0, point (x,y) lies within the ellipse. All points that yield results greater that 0 are outside of the ellipse and can get excluded of the point set.

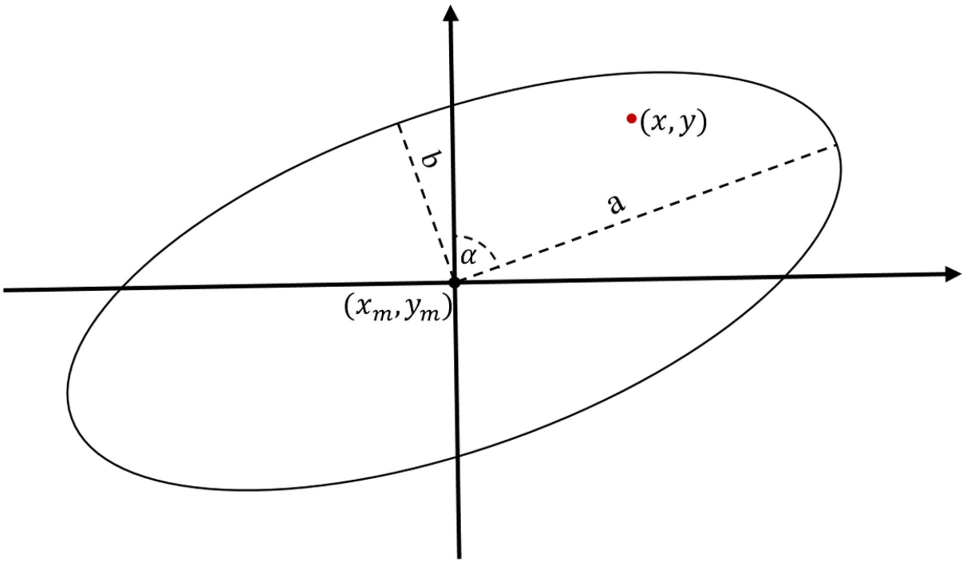


Figure 26: Rotated ellipse for point set determination

After defining the proximate points to the cell of interest, the implemented algorithm applies a filter depending on the center-center distances and the azimuth. The reason for implementing an additional filter after defining the proximate point set are runtime and memory issues. As described the point sets used for this thesis are dense, which implies also a large number of proximate points. While calculating the Kriging value one has to invert matrices that is computational expensive and memory intensive if using large point sets. For example, the computational time increases eightfold if the number of points is twice the size. Therefore, it is necessary to limit the number of points in order to achieve a reasonable runtime and memory performance. As a rule of thumb, one can take a point set of 12 as a minimum, which generates acceptable results. By increasing the number of points the result gets more reliable, however a limit of 32 points is fixed in most algorithms. This is because results generated by a larger number of points are not improved significantly. (Kelkar & Perez, 2002)

To achieve the filtering of the proximate point set, the implemented algorithm first divides the search ellipse in three different sectors. After that, each sector is further divided into two preferred regions and two secondary regions. Using these subdivisions, the algorithm first adds points of the midmost sector and the preferred regions (S_{1p}) to the point set and checks if more than 12 data points are available. If the size of the point set is greater than 12 the filtering is completed otherwise the algorithm proceeds and is adding the remaining points of the first sector (S_{1s}) to the point set and checks again for the number of data points. Depending if the condition is fulfilled the algorithm proceeds until every point is added, which should only be the case if one uses a sparse data set.

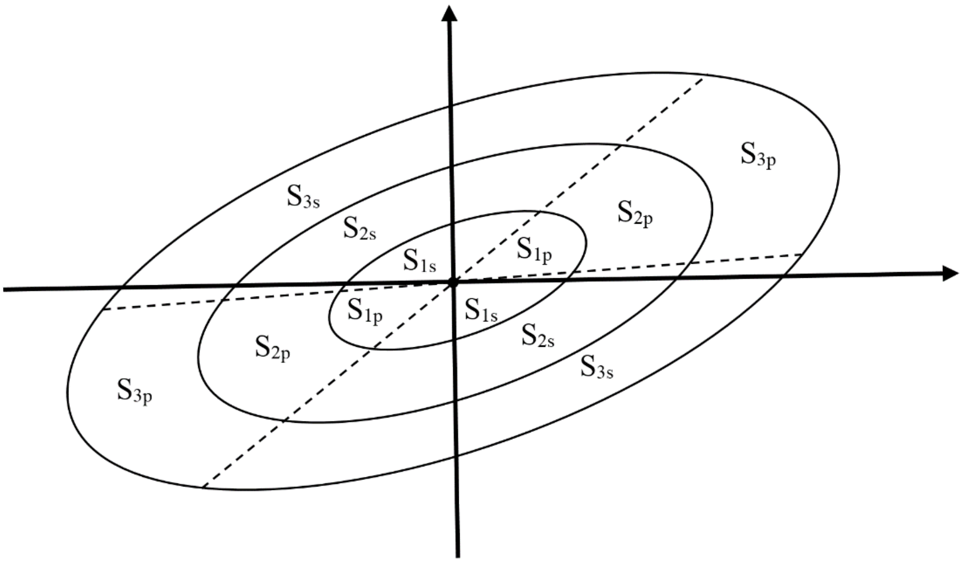


Figure 27: Sectorial division for further filtering

Subsequent to the definition of the filtered point set one can start calculating the covariance matrix and vector to solve for the Kriging weights as shown by equation (2.2). A major issue if one wants to solve for the covariances is that the equations presented so far are only parametric models for a symmetrically radial correlation length. As already emphasized, in real problems properties will mostly have a directional behavior, hence an elliptical behavior if describing the correlation length. Therefore, equations (2.7) to (2.9) need a slight modification in order to account for an elliptical behavior:

- Spherical

$$C\left(\frac{h}{r_e}\right) = \begin{cases} \left(1 - \frac{3h}{2r_e} + \frac{1}{2}\left(\frac{h}{r_e}\right)^3\right)\mathrm{S} + \mathrm{N} & if\ h \le r_e \\ \mathrm{S} + \mathrm{N} & if\ h > r_e \end{cases} \tag{3.4}$$

- Exponential

$$C\left(\frac{h}{r_e}\right) = \left[exp\left(-\frac{3h}{r_e}\right)\right]S + N \tag{3.5}$$

- Gaussian

$$C\left(\frac{h}{r_e}\right) = \left[\exp\left(-\frac{3h^2}{{r_e}^2}\right)\right]S + N \tag{3.6}$$

r_e .... *Correlation length in any direction*

To modify equation (2.7) to (2.9) one must substitute the circular correlation range a by the elliptical correlation length r_e that is influenced by the azimuth. Using the modification as shown by equations (3.4) to (3.6) one is able to calculate the covariances of the system. In principal, most of the necessary data is available. Sill and nugget effect are defined by the fitted variograms, whereas h needs to be calculated using the two cell center coordinates. The only component, which is more complicated to calculate, is the elliptical correlation length r_e. For its calculation, first a linear function, based on the two center coordinates is calculated. Next one has to find the root of the combination of the linear function and the function of the rotated ellipse as shown in (3.3). Rearranging the polynomial and expressing x explicitly yields the following quadratic equation.

$$x_{e1,2} = hx^2 + ix + j \tag{3.7}$$

with:

$$h = \frac{(\cos(\alpha) - ksin(\alpha))^2}{a^2} + \frac{(\sin(\alpha) + kcos(\alpha))^2}{b^2} \tag{3.8}$$

$$\begin{aligned} i &= \frac{2\big(ke\; sin^2(\alpha) - x_m\, cos^2(\alpha) + \cos(\alpha)\sin(\alpha)(kx_m - e)\big)}{a^2} \\ &+ \frac{2\big(ke\; cos^2(\alpha) - x_m sin^2(\alpha) + \cos(\alpha)\sin(\alpha)(e - kx_m)\big)}{b^2} \end{aligned} \tag{3.9}$$

$$j = \frac{(x_m \cos(\alpha) + e\; sin(\alpha))^2}{a^2} + \frac{(x_m \sin(\alpha) - e\cos(\alpha))^2}{b^2} - 1 \tag{3.10}$$

$$e = d - y_m \tag{3.11}$$

α................ *Azimuth*

k................ *Slope of linear function*

a................ *Major axis*

b................ *Minor axis*

d................ *y- intercept of linear function*

x_m .... *x- coordinate of ellipse center*

y_m.............. *y- coordinate of ellipse*

$x_{e1,2}$............ *x- coordinates of root*

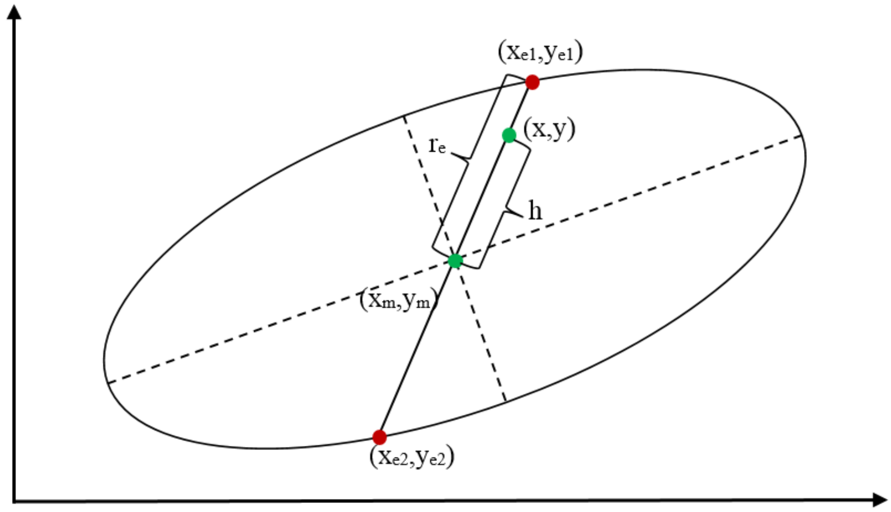


Figure 28: Calculation parameters of covariance

Figure 28 indicates two solutions, which is also verified by the derived equation (3.7). A major step for calculating the correct value of r_e is to determine the reasonable solution of $x_{e1,2}$. In the end, one can calculate the corresponding y-value y_e, using x_e and determine the distance r_e by the Pythagorean theorem as shown.

$$r_e = \sqrt{(x_m - x_e)^2 + (y_m - y_e)^2} \tag{3.12}$$

Finally, the covariances can be calculated by equations (3.4) to (3.6). A major downside of this algorithm as explained is that for the proper scaling of the covariances, the sill value is necessary. This is problematic because of two reasons. First, sill values are not necessarily the same for the major and minor direction (Figure 29) and secondly no azimuth dependent conversion between major and minor direction sills is implemented in the described algorithm.

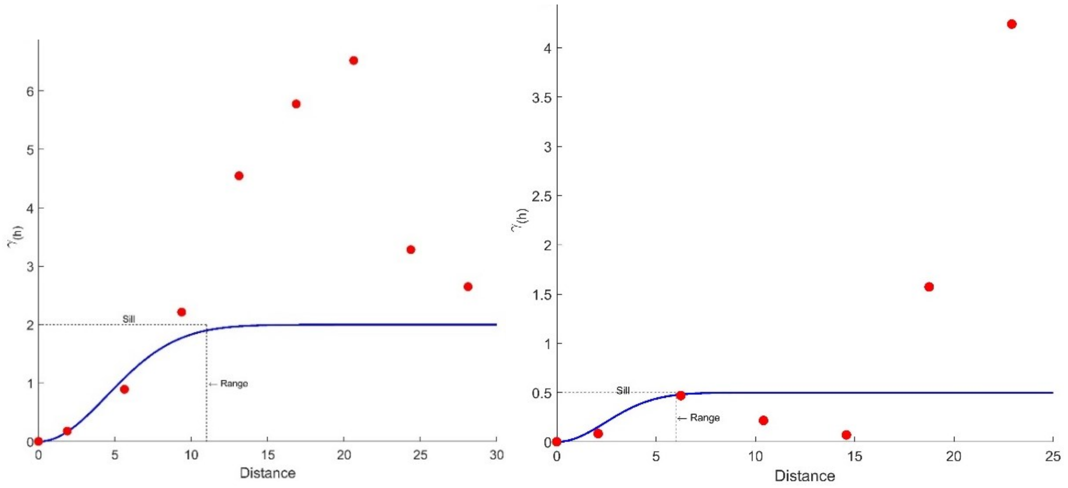


Figure 29: Major (left) and minor (right) direction variograms with different sill

The current implementation of this algorithm neglects this problem and uses the major direction sill for all scaling purposes. This creates, depending on the difference of major and minor sill, calculation errors that create outliners while calculating the cell values. Therefore, it is advisable to use this algorithm only if a small difference in sill occurs.

Although the explained algorithm works best if having same sill values, a small modification can improve its functionality to detect outliners. As mentioned, the algorithm generates outliners due to calculation errors caused by different sill values. In order to detect them one can utilize the dense point set surrounding the calculated property of the triangulated grid. To be more specific a mean value and the corresponding standard deviation can be computed using the surrounding data points. Furthermore, one can introduce a sensitivity factor to obtain more control for fine-tuning if still the algorithm produces some outliners. In this context, the sensitivity factor controls the allowed standard deviation around the computed mean value. As a default value the program uses the closest four points next to the computed triangulated cell and twice the standard deviation as an allowed interval. In the end, the algorithm as implemented should reproduce an acceptable property field.

As a final tool, the code provides a review window that displays the property maps, histograms and if existing, variograms. This review option allows a perfect direct comparison between Kriging and Inverse Distance Weighting results, while it is also providing a deeper insight in the different grid structures. Therefore, it is a vital tool for evaluating the generated results.

Chapter 4

Simulation Setup and Property Validation

In the end, not only direct results, like property magnitude and distribution after mapping, were of interest, but more interestingly the difference in simulation behavior. Therefore, three different sets of data were mapped first and subsequently simulated using a pre-defined simulation setup. The first two data sets were synthetic maps generated by a fBm (fractional Brownian motion) algorithm, with the first one having a single modal distribution and an extent of 64x64 meters. Contrary to that, the second synthetic data set shows a multimodal distribution and an extension of 128x128m. The third data set used, was taken from the SPE 10 comparative paper. It provides two different model setups, of different nature to test upscaling techniques. The first provided model is a vertical cut of a 2D- gas injection problem consisting out of 2000 cells initially. The second model is a 3D- water flooding setup consisting out of 1.1 million cells. The reasoning behind the low number of cells in the geological model was to ensure a reasonable simulation runtime. As a result, both models allow simulations on fine and upscaled grid basis, which ensures a perfect comparability (Christie & Blunt, 2001).

The software of choice for all simulation purposes was decided to be MRST that provides a vast selection of fluid flow solvers and a detailed literature of the concept behind them. To simplify the simulation setup using MRST, an additional user interface shows all the necessary parameters and suggests some default values for simulation. In the end, the user must define parameters of four different categories in order to setup a complete reservoir model. First, the rock has to be defined by its minimum parameters, porosity and permeability. Secondly, the definition of the reservoir fluid is of importance followed by the third category of parameters that give information about time and boundary conditions. In the end, also the placement of producing and injecting wells is important to disturb the initial reservoir condition.

4.1 Reservoir Rock Setup

In order to define the basic parameters of the rock, one has to browse for a valid file type using the provided user interface. The valid file types are represented by a “.out” extension that contains the data of the structured grids and a “.dat” extension that contains the data of the triangulated grid. A vital parameter that is contained by these files is porosity, given in fraction of the bulk volume. While loading these file types, an automatic assignment of porosity to the corresponding grid is done. By default, the code also calculates a value for permeability using the Timur relationship. (Timur, 1968)

$$\log k = c_1\phi + c_2 \tag{4.1}$$

k................. *Permeability [mD]*

c_1, c_2 .... *Correlation parameters*

ϕ................ *Porosity [%]*

After loading and calculating the necessary data, the implemented algorithm defines the rock automatically and checks its consistency with the corresponding grid. Another possibility one can utilize is to upload a permeability file manually. Using the permeabilities browse button and selecting the desired file can easily do this. After successfully uploading the file, the rock-struct is re-defined and checked for its consistency.

A simplification, one might already have noticed is the negligence of compressibility. The current code setup assumes an incompressible rock and all its corresponding assumptions.

4.2 Reservoir Fluid Setup

The next important parameters to determine are the reservoir fluid properties. In order to initialize a proper fluid description, the code uses a Corey model, provided by MRST that describes the behavior of relative permeability. A simplification that comes along by using the mentioned fluid description is that one assumes incompressible- and immiscible flow. A major motivation for introducing this assumption was because of simplicity reasons. A compressible and miscible fluid flow approach would require more input parameters that are hard to determine, especially for the synthetic datasets. Using an incompressible flow approach instead, might not produce reliable production results, but allows an indication of the major behavior of a simulation regarding grid structure.

For the initialization of the Corey model, five different parameters for oil and water are required, respectively. To achieve a comparable result the fluids of all models were initialized using the same parameters, which were taken from the data set of the SPE 10 paper and are listed in the following table.

Table 2: Reservoir Fluid Parameters

Fluid Parameters		
	Water	**Oil**
Viscosity μ [Pas] 10^{-3}	0.3	3
S_{res}	0.2	0.2
$k_{(Send)}$	1	1
n	2	2
Density ρ [kg/m^3]	1025	849

Another limitation in the current code is the possibility of defining only one PVT region. Although real reservoirs with a large extent always have several PVT regions, this simplification does not bother the upcoming simulation runs. The reasoning behind this is that the used data sets only present a very homogeneous and limited part of a reservoir, making the possibility to account for several region obsolete.

4.3 Initial Parameters

The third section of parameters describes the initial setting of the reservoir and boundary conditions applied on the model edges. A basic description one has to provide is the initial reservoir pressure, saturation and the desired simulation time. Defining these parameters already allows running simulations using a default setting regarding simulation time intervals and boundary conditions. The table below shows the default parameters when using a setup as mentioned.

Table 3: Default setting of initial simulation parameter

Time	
dt	5 days
Boundary Condition	
Type:	Constant Pressure Boundary
Value:	p_{init}
Saturation at boundary:	[S_{winit} 1-S_{winit}]

To improve the models flexibility, one can define user specific settings like changing the constant pressure and saturation at the boundaries. Furthermore, the code allows changing between a constant pressure and constant influx constraint at the boundary. The only additional parameter that is necessary for enabling a constant influx option is a rate definition. At this time, it is important to point out that a rate should not be picked arbitrarily. The influx rate at the boundary faces is strongly dependent on the cell size. In the case of both synthetic data sets, a cell of the structured grid has a size of 1 m^3. If one applies, for example a constant influx at each boundary face of 5 m^3/day as a boundary condition the pressure response will be abnormally high. Therefore, it is wise if one thinks about a realistic boundary influx first if enabling this option. Using the current version of the code, one can only assign the same pressure or rate towards all principal directions. Simpler said, one is not able to apply a different rate or pressure for a northern boundary face than for a southern one.

As a conclusion, Table 4 shows the basic initial parameters used for the upcoming simulation runs. All results shown in the results and discussion sections are based upon these values.

Table 4: Initial simulation parameters

Initial Parameters	
p_{init} [bar]	200
Initial water saturation [-]	0.2
Simulation time [days]	120

4.4 Well Definition

The last important section to complete the simulation setup is to define sources and sinks in the system. In this matter, MRST provides several options, whereas the used one was to apply the Peaceman well model. Using a well model allowed a monitoring for the produced rates and bottomhole pressures, respectively. This was one major criteria for the selection of a source/ sink term. The reason for this is that if one knows for example the rates, the breakthrough time from injector to producer can be determined easily. Furthermore, a difference in produced volume or an abnormal pressure response can be detected. Since, the initial idea of running simulations was for exactly these reasons, the definition of producers and injectors was of major importance.

Therefore, the implemented algorithm provides an interactive option to assign wells to the grid by clicking on the desired cell. Moreover, it provides the possibility to select between a pressure- or rate-controlled well system, which allows more flexibility in regard to the simulation setup.

The placement of the wells was done for each data set individually, but followed a certain rule. In order to monitor grid orientation effects, each data set was simulated twice, using two different well setups. The first setup of wells uses a producer that is oriented to the injector by an angle of approximately 45 degrees. Compared to that the second well setup was picked to have no directional angle involved.

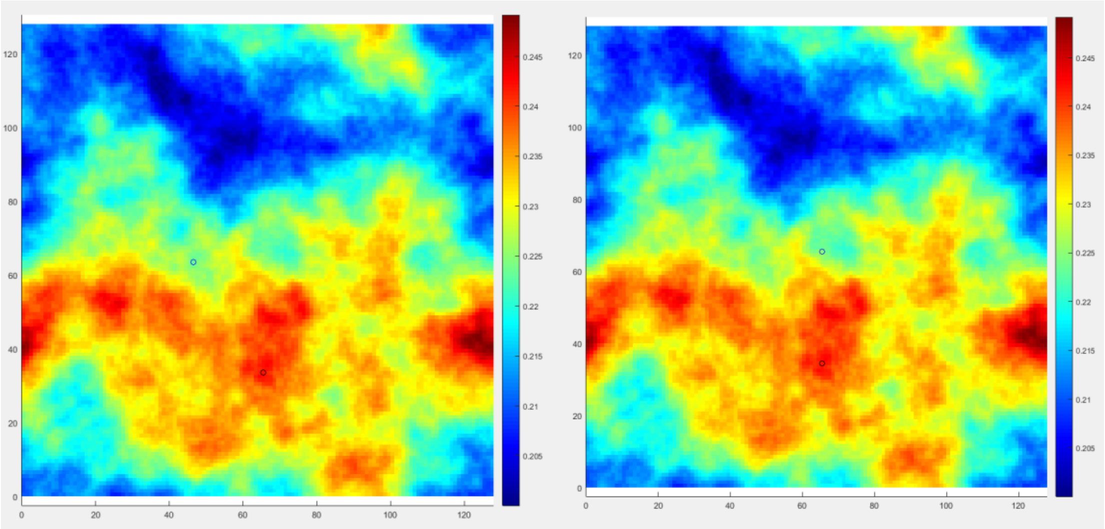


Figure 30: Well setup for simulation: Setup W1 (left) and Setup W2 (right)

Table 5 shows a brief summary of the control type and values used fort the well setup.

Table 5: Well setup

Producer	
Type	Pressure- controlled
pwf [bar]	190
Injector	
Type	Pressure- controlled
p_{wf} [bar]	210

Chapter 5

Results and Discussion

The following chapter gives a detailed overview regarding all generated results in the course of this thesis. This includes the outcome of both, the effectiveness and accuracy of the mapping algorithm and the difference in simulation behavior. First, a comparison and analysis of the implemented mapping algorithms gives a recommendation for a preferred method and indicates advantages of its usage. After that, simulation runs of the mapped field, generated by the preferred algorithm highlight the impact of the new grid structure with its corresponding property field.

5.1 Mapping

In order to determine the most efficient and accurate mapping algorithm, the use of two synthetic datasets provide an insight on the time effort for applying an algorithm. Furthermore, the computational cost and accuracy were of importance for rating, which algorithm performs better to map structured grid properties on unstructured triangulated meshes.

5.1.1 64x64m Synthetic Data

The application of both mapping techniques was done first on the 64x64m synthetic dataset (Figure 31). The data showed an anisotropy around the center of the data set with a porosity range between 20 to 25%. At a closer look the anisotropy shows a slight N-E trend, which indicates that data for mapping purposes should preferably picked from this direction.

As the data shows a slightly directional trend, the first implemented technique is the Kriging algorithm since it is not possible to account for trends using the inverse distance weighting method.

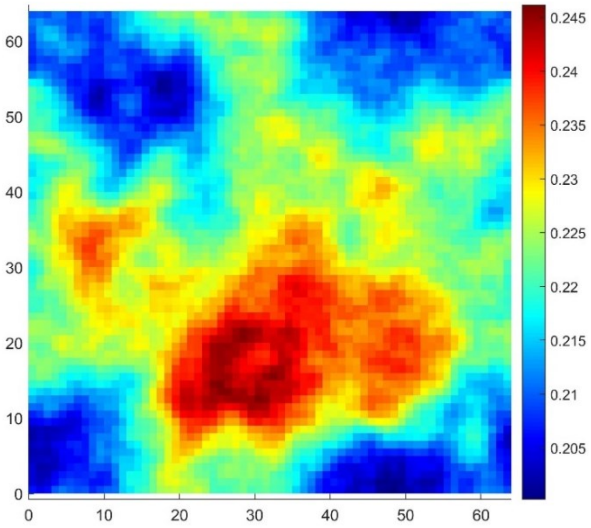


Figure 31: 64x64m synthetic dataset

First, the variograms in both principal directions were fitted, using an azimuth of 70 for the major direction and an azimuth of 340 for the minor direction. Furthermore, the search radius of the major direction was determined to be 20m using 8 lags whereas the search radius of the minor direction was 15m using 6 lags. Figure 32 show both search areas using the mentioned data.

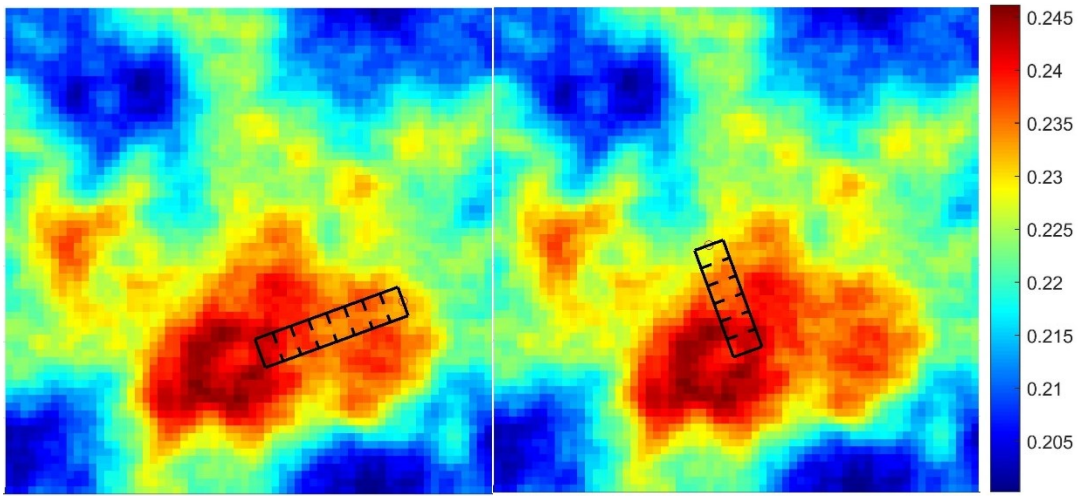


Figure 32: Variogram search area in major (left) and minor (right) direction

Applying both search areas as shown by the figure above yields the experimental variograms in Figure 33.

At a glance, one is able to see that the sill values of major and minor direction are of completely different magnitude. Therefore, using the implemented algorithm will create outliners that the code has to detect as already explained in chapter 3.2.2. To accomplish a satisfying fit for the parametric variogram in the major direction, one can see that a Gaussian type model using a sill of 3.5 and a range of 9m shows the best result. After this, the best fit for the minor direction experimental variogram is found using a Gaussian type model with a sill of 0.05 and a range of 4m.

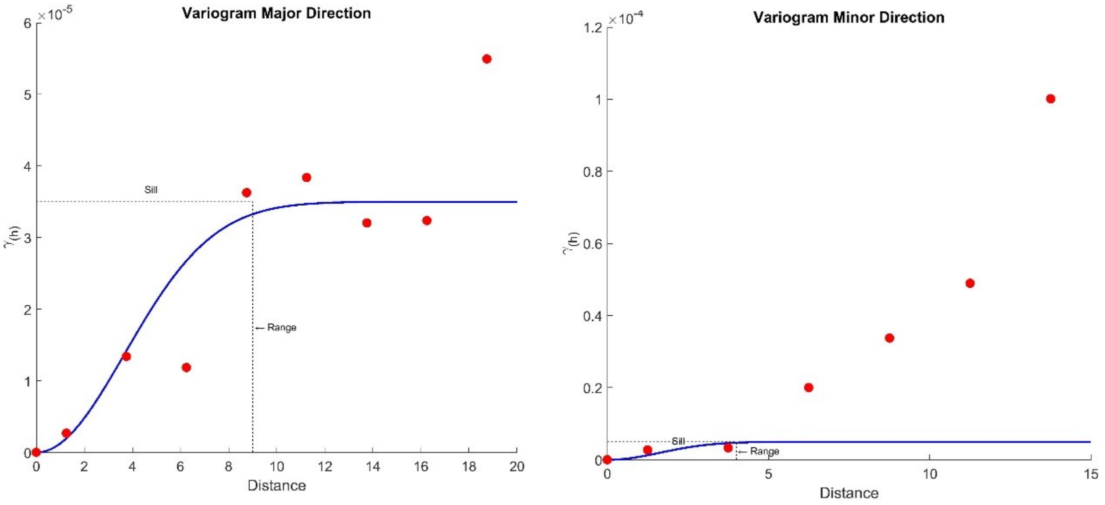


Figure 33: Variogram in major (left) and minor (right) direction

Applying the Kriging algorithm, using the determined parametric variograms yields the following figure.

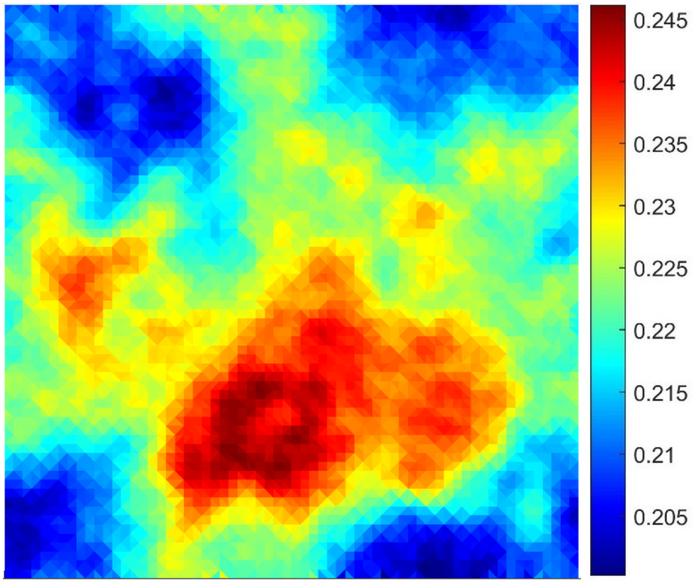


Figure 34: Mapping result of synthetic 64x64m grid using Kriging

To reach the result from the figure above, a total runtime of 25.8 seconds using an Intel(R) Xenon(R) X5570 3.06 GHz processor was necessary. The total number of detected outliner was 28 by using the default-detecting algorithm on a point set with a total cell number of 6490.

One can get additional insight in the result using some statistical display as for example a histogram that shows the change in the properties occurrence. As one can see in Figure 35, there is not a significant change between the property distribution of the triangulated and the uniform structured grid. Only the non- uniform structured grid shows a significant deviation from the initial property distribution. Another value of interest is the change in pore volume if comparing the triangulated grid and the structured uniform one. Calculating the cell volume by assuming a cell height of 1m yields a pore volume of 909.8915m^3 for the triangulated grid and a pore volume of 909.8937m^3 for the structured uniform grid. This slight deviation also shows an excellent fit between both maps. To see the total change in property distribution one can have a look at the following figure. The maximal change of one property interval is around 1 percent, which is again a sign for an excellent fit.

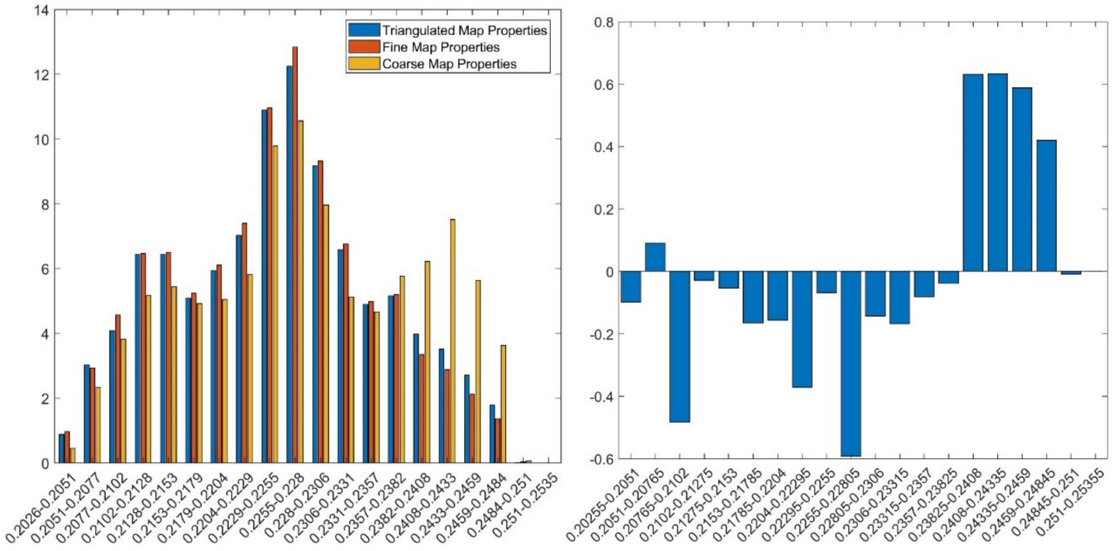


Figure 35: ***Left:*** *Property distribution in percent using Kriging for triangulated grid (blue), uniform structured grid (red) and non- uniform structured grid (yellow);* ***right:*** *Total change of property distribution in percent using Kriging: Triangulated to structured uniform grid*

After analyzing the result of the Kriging algorithm, the application of the inverse distance technique was done. The following results based upon the default setting of the code, whereas the search radius for determining the point set is 10m and the weighting exponent is 2. Using these parameters one can yield a result in a runtime of 0.55 seconds.

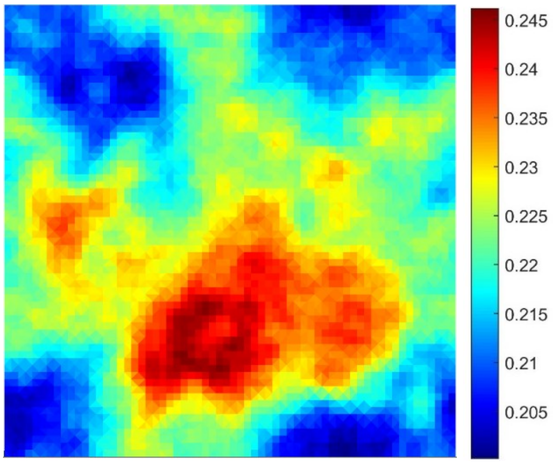


Figure 36: Mapping result of synthetic 64x64m grid using inverse distance weighting

At a glance, the mapped result looks again to be an excellent fit. Figure 37 supports this as one can see that there is again no significant change in the property distribution if comparing the triangulated and structured uniform grid. Furthermore, the pore volume of 909.9005m^3 shows that only a slight change of compared to the initially 909.8937m^3 occurs. Again, the total change of property intervals is of interest and shows that the maximal change of a category is below 1 percent as well.

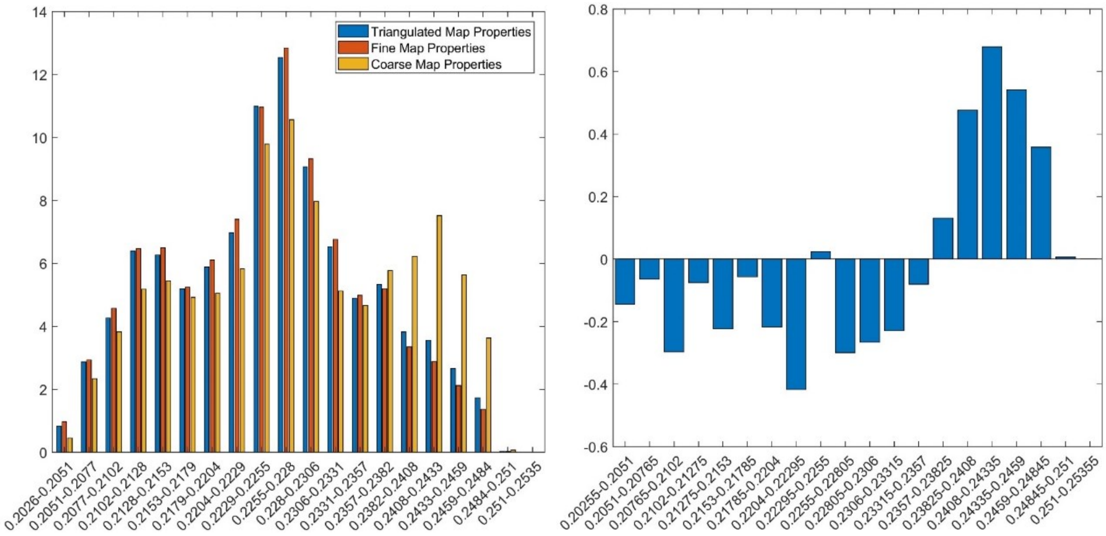


*Figure 37: **Left:** Property distribution in percent using inverse distance weighting on triangulated grid (blue), uniform structured grid (red) and non- uniform structured grid (yellow); **right:** Total change of property distribution in percent using inverse distance: Triangulated to structured uniform grid*

5.1.2 128x128m Synthetic Data

In order to verify the mapping results of the 64x64m grid one can use the second synthetic dataset as an object of comparison. The 128x128m dataset (Figure 38) contains like the previous example an anisotropy that indicates a directional trend towards east of the property to fit. Furthermore, the porosity values have a range between 20 and 25%.

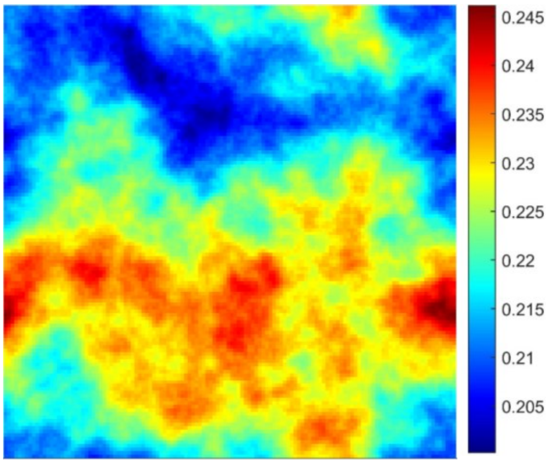


Figure 38: 128x128m synthetic dataset

To apply the Kriging algorithm on the current data set, the major and minor direction of the variograms were defined first. As mentioned before the data shows a directional trend towards east, which indicates an azimuth of 90 degrees for the major direction and corresponding to that an azimuth of 0 degrees for the minor direction. The search radius for the major direction was picked to be 30m using 8 lags whereas the search radius of the minor direction was picked to be 25m using 6 lags.

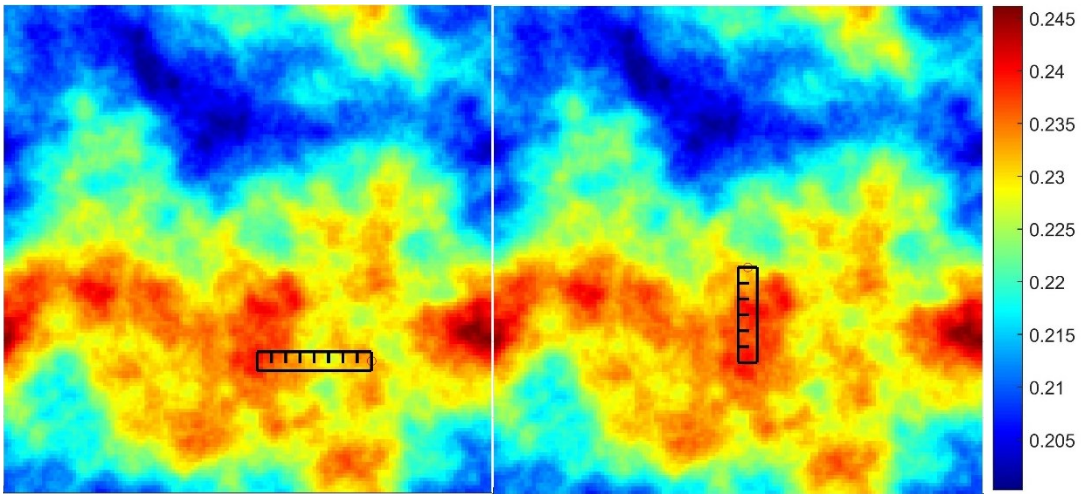


Figure 39: Variogram search area in major (left) and minor (right) direction

These settings, yield experimental variograms as shown by Figure 40. Again, a Gaussian type was found to be the best fit for the used dataset. To fit the major direction parametric model, a sill of 2 and a range of 11m was used, whereas a sill of 0.5 and a range of 6m was used for the minor direction parametric model.

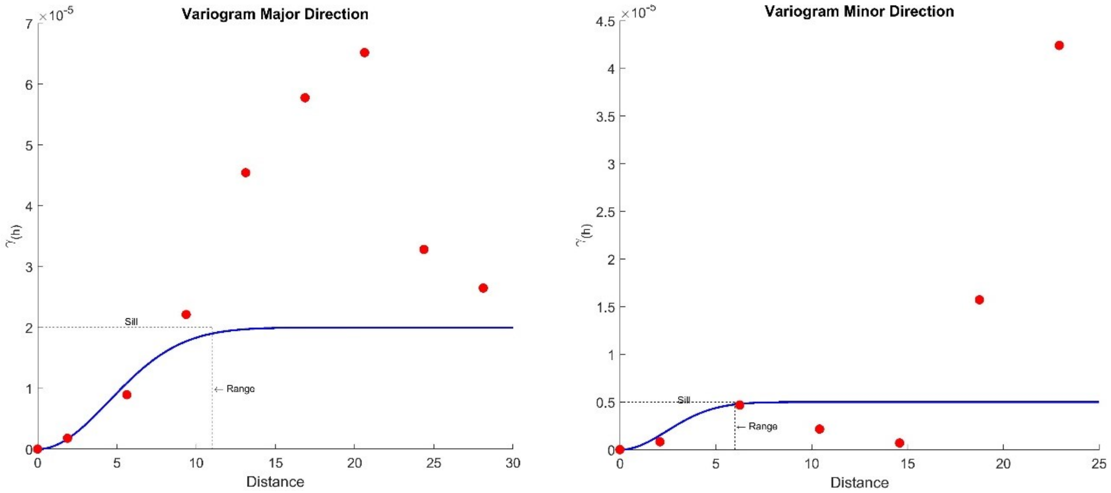


Figure 40: Variogram in major (left) and minor (right) direction

Applying these settings yields the Kriging result as shown by Figure 41. The default outliner detection found 454 abnormal values and corrected them as explained in section 3.2.2. The total runtime of the algorithm was 147 seconds using the Intel(R) Xenon(R) X5570 3.06 GHz processor.

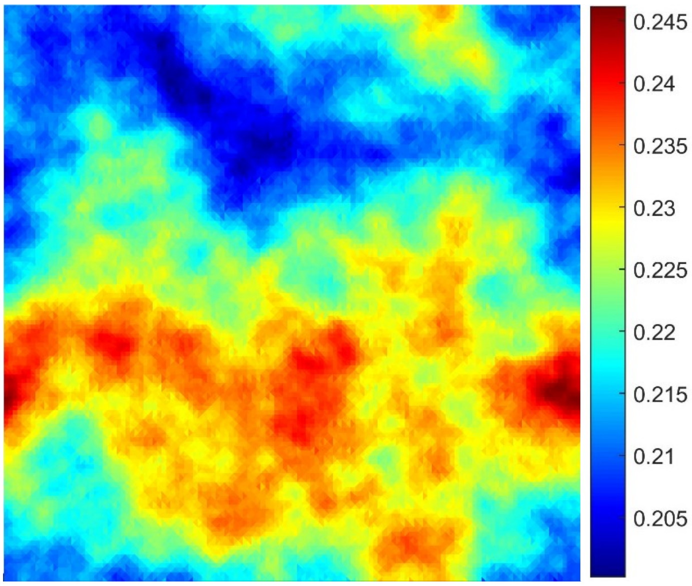


Figure 41: Mapping result of synthetic 64x64m grid using Kriging

Next, the histogram (Figure 42) shows that the mapped properties have an excellent fit if comparing the triangulated with the structured uniform grid properties. Moreover, the pore volume of both grids, the triangulated and the structured uniform one, was in accordance. With an assumed height of 1m, the triangulated grid has a pore volume of 3661.2243m^3 compared to 3661.2175m^3 of volume for the structured uniform grid. Similar to the result of the 64x64m dataset the Kriging algorithm produced an output with property category deviations of lower than 1 percent.

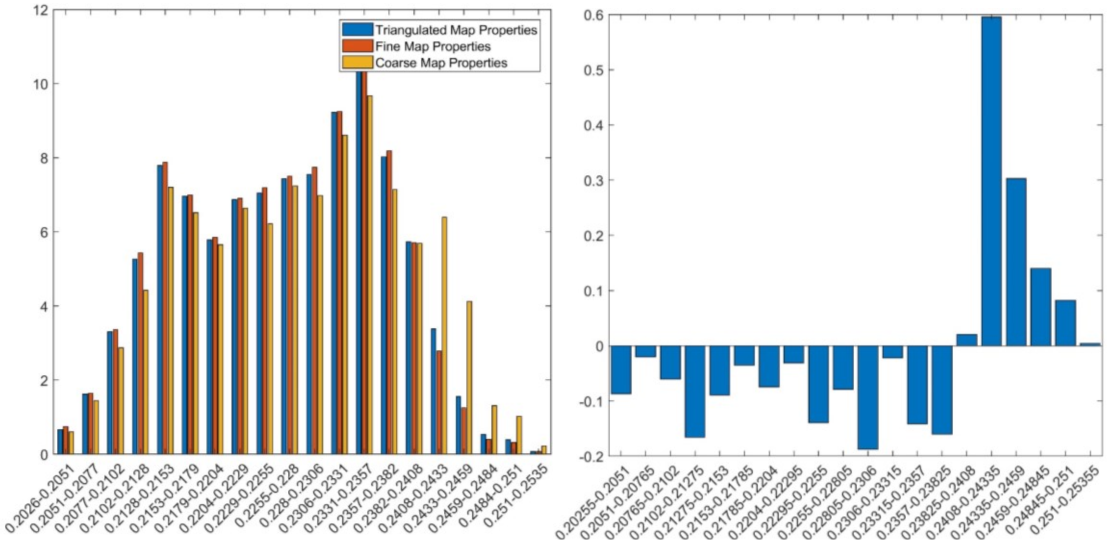


*Figure 42: **Left:** Property distribution in percent using Kriging on triangulated grid (blue), uniform structured grid (red) and non- uniform structured grid (yellow); **right:** Total change of property distribution in percent using Kriging: Triangulated to structured uniform grid*

After the successful implementation of the Kriging algorithm, also the inverse distance technique was tested. To compare the result with the previous dataset, the default settings of the inverse distance was applied again using a 10m search radius and a weighting exponent of 2. Afresh, the inverse distance technique generated excellent results as shown in the following figure.

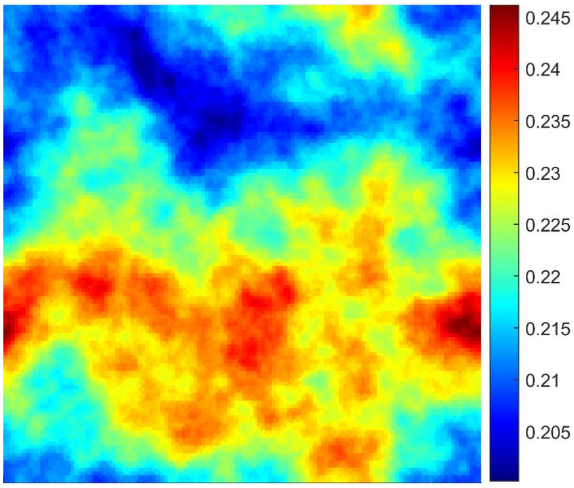


Figure 43: Mapping result of synthetic 128x128m grid using inverse distance

Like the 64x64m dataset, this is also supported by a closer look at the property histogram. Each category shows an excellent match if comparing the triangulated grid with the structured uniform grid. In the end, the triangulated and the structured uniform property distributions showed a difference lower than 1 percent.

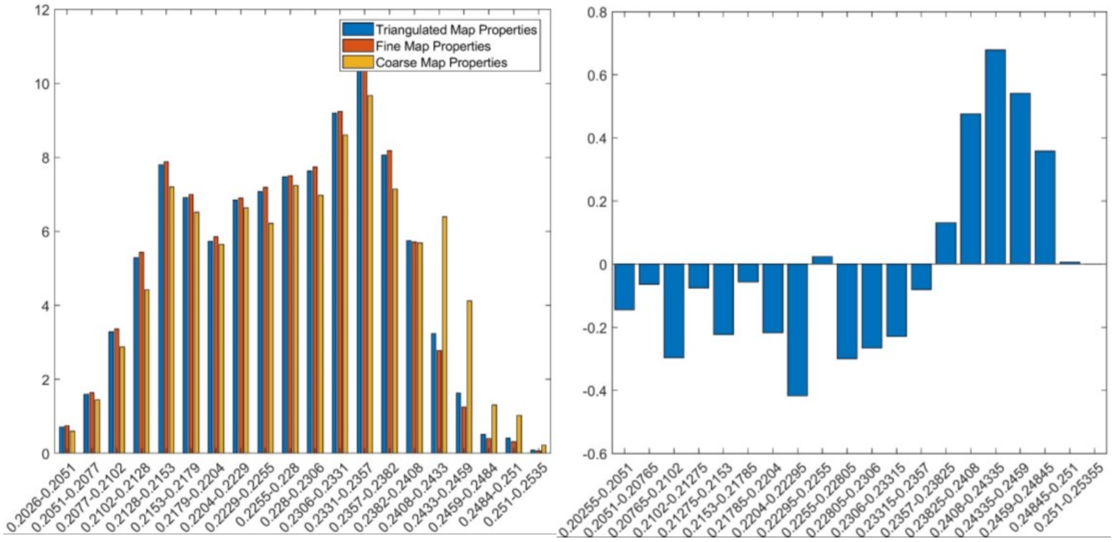


*Figure 44: **Left:** Property distribution in percent using inverse weighting distance on triangulated grid (blue), uniform structured grid (red) and non- uniform structured grid (yellow); **right:** Total change of property distribution in percent using inverse distance weighting: Triangulated to structured uniform grid*

5.1.3 Mapping Discussion

In order to determine which algorithm performs better for this specific mapping purpose, the results of chapter 5.1.1 and 5.1.2 are compared in the following. The first question that arises is how one defines performance. In this context, the determination of performance relies on three major categories:

- Complexity of implementation
- Accuracy
- Runtime

In regard of the first category, one can clearly tell that the inverse distance-weighting algorithm is easier to implement than the Kriging algorithm. The reason for this is the difference in calculating the weightings. Both methods assign a greater weighting to closer data points, but additionally to that, the Kriging technique also applies a directional preference. In order to estimate this directional preference, the determination of variograms is necessary resulting in additional expenses regarding parameter definition. Although the Kriging technique requires more input than the inverse distance algorithm, it is still an easy method to use. Several derivations of this technique are available, requiring more parameters, hence are harder to implement.

The next category of interest is the accuracy of the implemented algorithms. As Figure 35 and Figure 42 show, both utilizations of the Kriging algorithm delivered excellent results having deviations less than 1% from the initial property distribution. Nonetheless, the accuracy of the implemented version of the algorithm also relies on the effectiveness of the outliner detection. Allowing an error bar too large, might affect the accuracy of the result. Therefore, one should carefully revise each result regarding abnormal errors generated by an insensitive setting of the outline detection. Compared to that, Figure 37 and Figure 44 show that also the inverse distance technique generates excellent result with errors less than 1%. At a glance, the result was surprising, but was clear after further consideration of the specific problem. Since the dataset of the structured uniform grid is used as a source for mapping, the point set density is high. As a result, there is always a known data point close to the point of interest, which is assigned a high weighting factor. This also explains why the directional behavior of both datasets is mapped properly. The single value close to point of interest receives a weighting high enough that all the other values of the point set have only a minor influence on the result. The only limitation of this algorithm is at the interface of a sharp property change. Depending on the geometry setup, a slight change of these interfaces can occur, which is not too problematic.

The last category of interest is the total runtime of the implemented algorithms. Considering the results of both datasets clearly shows a superiority of the inverse distance technique compared to the Kriging algorithm. This superiority is caused by two different reasons. First, the not optimized data structure for the Kriging algorithm in the current codes version and secondly the number of processes that have to be calculated additionally to run the Kriging algorithm. Therefore, one can say that the inverse distance technique has a clear advantage compared to Kriging.

To conclude the comparison of all three categories, it can be said that for this special case of mapping the inverse distance technique is the better choice. First, it is easier to implement and applicable to all kinds of datasets. Next, its accuracy is not dependent on the detection of outliners and finally its runtime will always be faster than the Kriging algorithm. The following table gives a short rating of all evaluated categories for both methods.

Table 6: Rating of implemented algorithms

	Kriging	**Inverse Distance**
Complexity of	3	1
Accuracy	1	1
Runtime	2	1

5.1.4 SPE 10 dataset

Ultimately, the mapping of the SPE 10 dataset (Christie & Blunt, 2001) was conducted by using the preferred algorithm, the inverse distance-weighting method. Since the triangulation of the datasets is limited to the 2D domain, only the first layer of the SPE 10 paper is picked. Both the initial data map and the triangulated data map can be seen in the figure below.

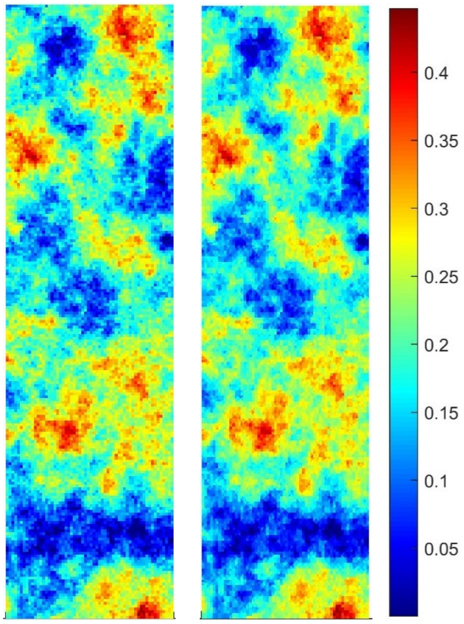


Figure 45: Initial property map (left) and triangulated one (right) of SPE10 layer 1 using inverse distance weighting

The fit of the inverse distance method is again excellent concerning the property map and regarding the statistics. The maximum difference of the property distribution is lower than 0.2 percent in total, which can be seen by Figure 46. Furthermore, also the pore volume indicates a satisfying result with 2567.4194 m^3 for the structured uniform grid and 2567.5417m^3 for the triangulated map.

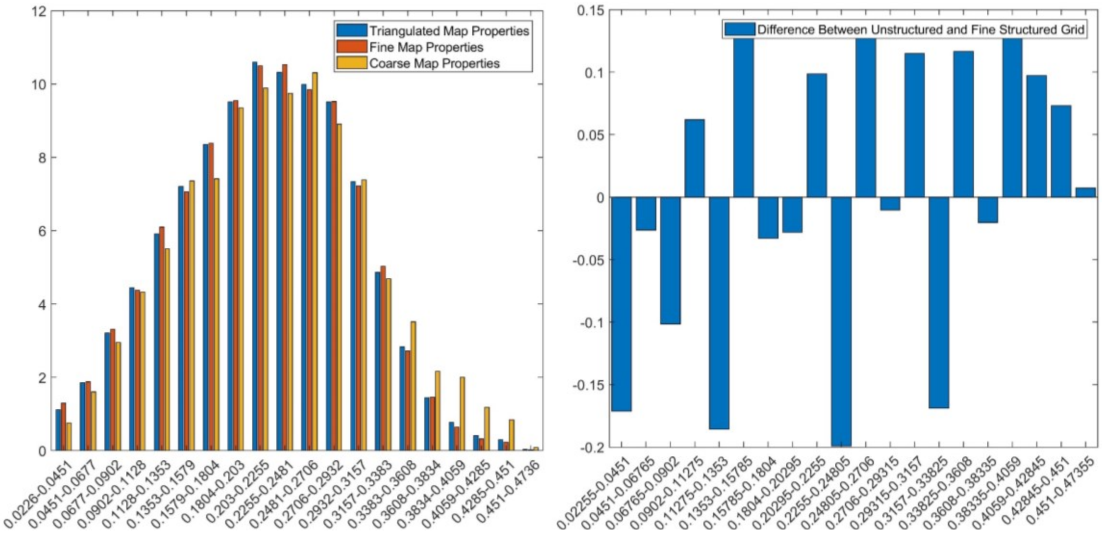


*Figure 46: **Left:** Property distribution in percent using inverse weighting distance on triangulated grid (blue), uniform structured grid (red) and non- uniform structured grid (yellow); **right:** Total change of property distribution in percent using inverse distance weighting: Triangulated to structured uniform grid*

5.2 Simulation

The following chapter gives a detailed overview of all simulation results generated in the course of this thesis. Chapters 4.1 to 4.4 show a summary of all underlying base case conditions in order to reach the upcoming results. In total, a number of 30 simulation runs indicate the differences caused by coarsening factor, grid type and well placement.

To keep track of the large number of simulations, it was necessary to introduce a specific nomenclature to identify the different simulation runs. In general, this nomenclature consists of five different specifiers:

Grid128_IDBCT1W1

- Name of field
- Type of grid
- Case classification
- Time step size
- Well setup

As mentioned above, the first part in the used nomenclature is the name of the oilfield or reservoir. A vital part is to separate the name from the other specifiers by an underscore sign as shown by the example. Then the specifier of the used grid followed by the case is followed by the specifier time step size given in days. In the end, the well setup is of importance.

Table 7: Grid type specifiers

Specifier	Grid type
ID	Unstructured triangulated grid using inverse distance as mapping technique
U	Structured uniform grid
NU	Structured non-uniform grid

5.2.1 The impact of grid structure

A factor of major investigation is the impact of the grid structure on the simulation result. In order to visualize possible effects, a monitoring of oil production rate, water cut and pressure/ saturation field was conducted. Having a closer look at the daily oil production and the corresponding water cut first, reveals an interesting result. The results of both synthetic data sets show a similar production rate for the structured uniform and non-uniform grids, whereas the triangulated grids seem to deliver the lowest production rate (Figure 47, Figure 63, Figure 64).

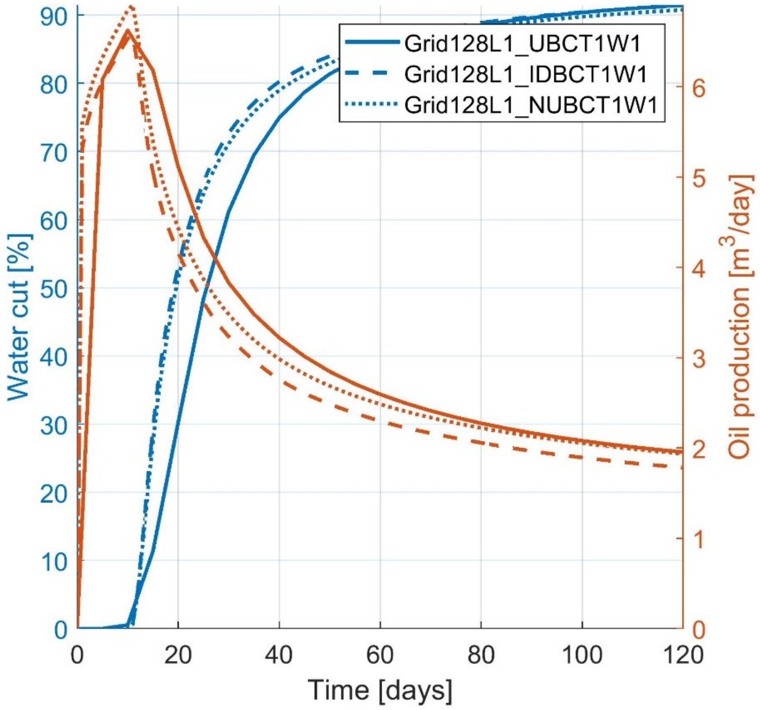


Figure 47: Comparison of oil production rate and water cut from different grids: 128x128m Grid

The used data set of the SPE10 paper shows a slightly different result. The production rate of the structured uniform grid is clearly higher as with the triangulated and structured non-uniform grid. In case of the water cut of all grids, no exact behavior crystallizes from the simulation results.

The question that remains is why the production rates vary from grid-structure to grid-structure. This is of interest, because looking at the cumulative production of both, water and oil, one can already see a clear difference between the produced volumes. In regard to the cumulative produced oil, Figure 48 shows that after only 120 days of simulation the result deviates for approximately 30m^3 from structured uniform to triangulated grid. All other simulation results follow this trend as well (Figure 65, Figure 66).

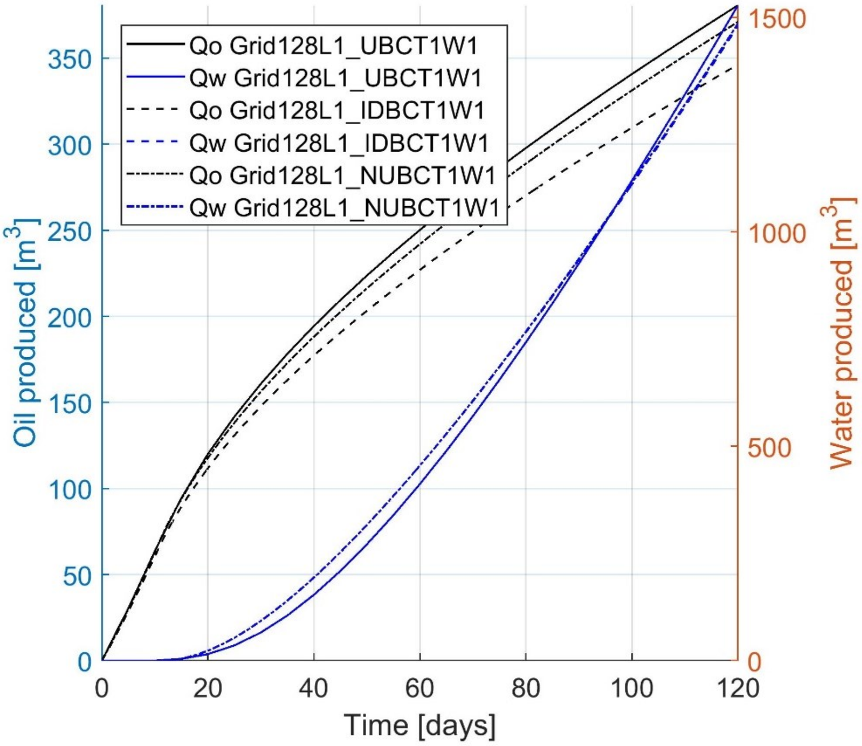


Figure 48: Cumulative produced volumes comparing different grid- structures: 128x128m Grid

To explain this phenomenon of different production behavior, the final saturation map of both, structured uniform and triangulated grid is plotted in Figure 49.

A part of the behavior of fluid production as shown by Figure 48 can be explained by the difference of the saturation fields. Both fields show a different extent, but a similarity of the different saturation trends. Although, the approximation of both fields seem similar, one can detect that the saturation transition of saturation changes looks smoother for the structured fine grid. Using this phenomenon, one can explain a relative change from oil to water production hence a change in water cut. A problem that still cannot be explained is the change of total produced volumes. As shown by Figure 48, the cumulative production is not only a relative change of water and oil production, furthermore a significant volume of oil and a minor volume of water is produced less.

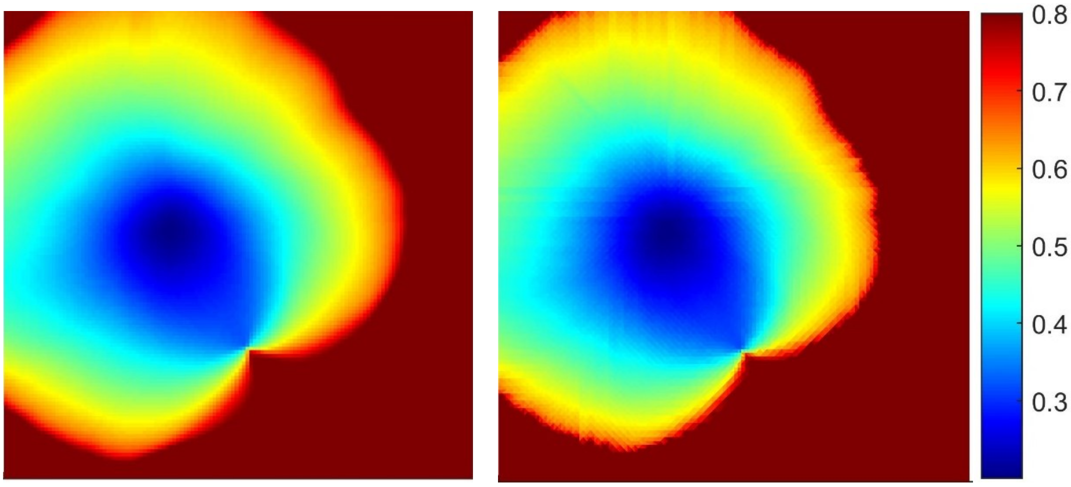


Figure 49: Saturation map of Grid128L1_UBCT1W1 (left) and Grid128L1_IDBCT1W1 (right)

To find an explanation for the change of total produced volume, a close look at the pressure distribution of both grids is helpful. Since both simulations are based on the same initial and boundary conditions, the only factor that influences the production of fluid volumes, for an incompressible example, is pressure.

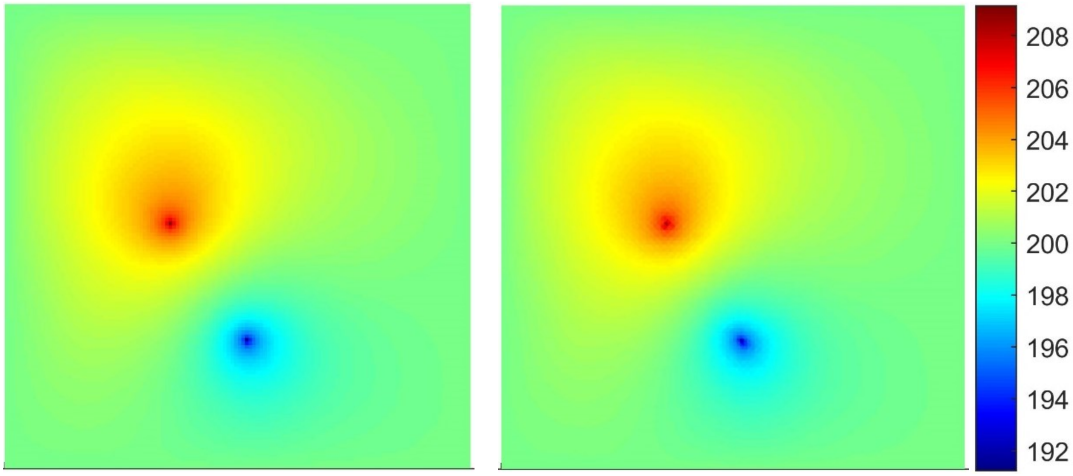


Figure 50: Pressure map of Grid128L1_UBCT1W1 (left) and Grid128L1_IDBCT1W1 (right)

At first sight, both pressure fields look the same, not showing a reason why a change in production rates should occur. To improve the knowledge of both pressure fields one can compare the calculated value for each cell. An issue that arises if one wants to compare the calculated values is that MRST uses a finite volume based approach to calculate the certain cell pressures, hence it calculates values at the cells barycenter. Since the grid structure of both model is different, also the number and location of centers is different resulting in a different matrix size of the pressure solution. Therefore, the result of both pressure fields are not directly comparable. In order to compare both fields, a conversion of them into images followed by an image analysis was conducted. Figure 51 shows the total difference of the greyscale image comparison and indicates a difference in the pressure field. Some parts of this difference in pressure can be explained by the different structure of grid blocks. The triangulated grid consist of a higher number of grid blocks hence has a higher resolution. This fact is the major cause of the phenomenon looking alike a pressure front that propagates into the field. Another part of pressure change that is caused in the near wellbore regions might be caused by the different grid structure that ultimately affects the production rate.

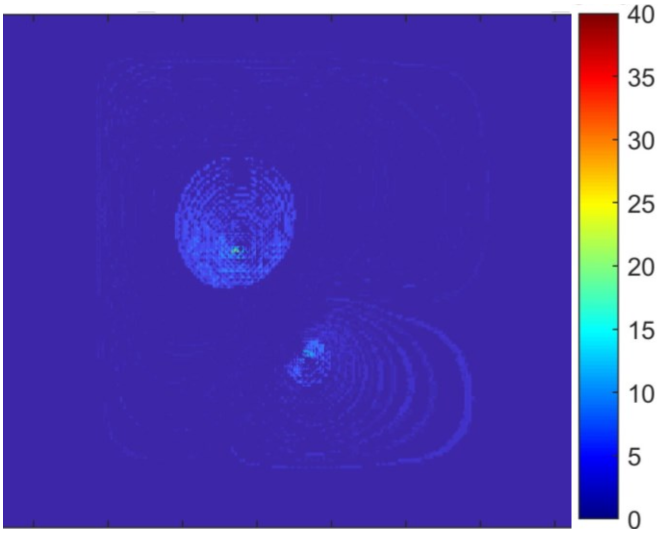


Figure 51: Image comparison of pressure field showing total difference in greyscale: Grid128L1_UBCT1W1 and Grid128L1_IDBCT1W1

One might argue that the already applied coarsening on the triangulated grid possibly creates the mentioned effects. Therefore, the structured uniform grid was triangulated and mapped directly, without applying any coarsening algorithm, yielding the following results.

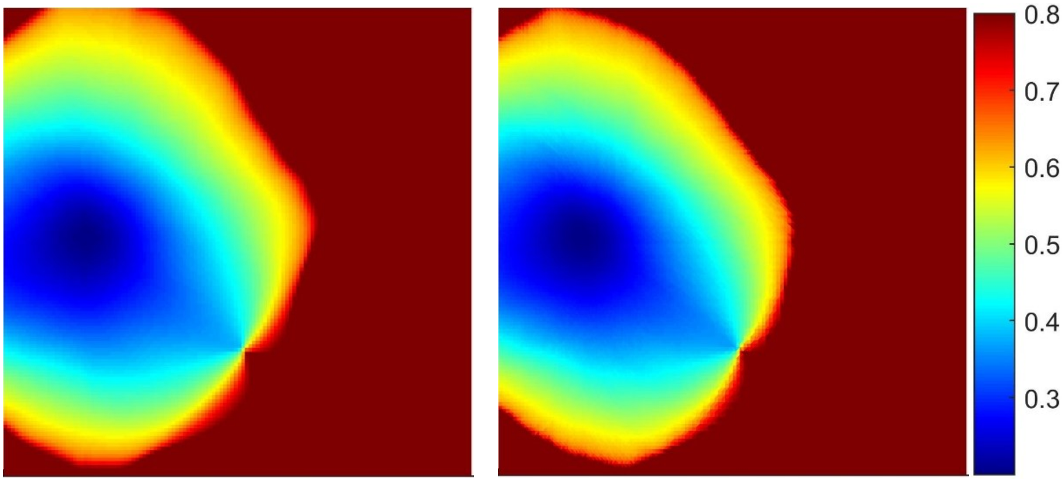


Figure 52: Saturation map of structured uniform grid (left) and triangulated grid (right) without coarsening

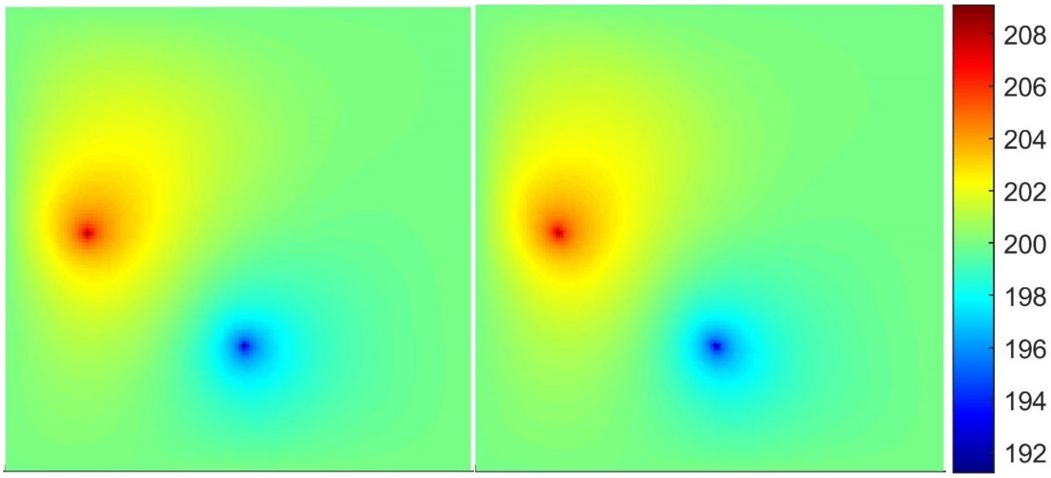


Figure 53: Saturation map of structured uniform grid and triangulated grid without coarsening

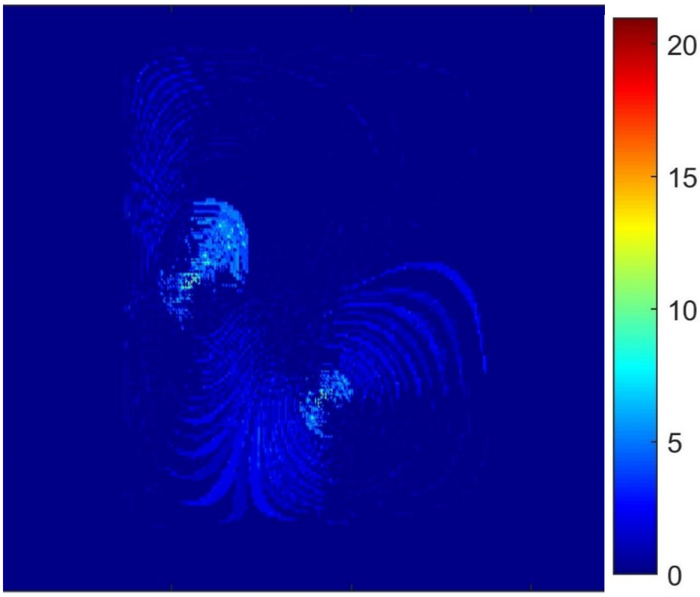


Figure 54: Image comparison of pressure field showing total difference in greyscale: structured uniform grid to triangulated grid without coarsening

Again, the propagation of the saturation field is slightly different, showing a slower change in saturation for the triangulated grid. This supports the observation seen by the difference in pressure as shown by Figure 54. Another remarkable fact is the absolute difference of the respective pressure fields decreases with a decrease in the coarsening factor. Looking at Figure 51 and Figure 54 and comparing the magnitude of maximum difference, one can see that the values of the coarsened case are larger.

To summarize the observed results, one can say that the production rate of the system changes due to a difference in the pressure calculation. Another phenomenon caused by this trend is a slower propagation of the saturation front, hence a relative change of the produced volume phases.

5.2.2 Impact of coarsening factor

The next factor of interest is the impact of the used coarsening factor on the simulation result. To have a clear understanding of it, simulations of all three datasets were performed and the produced cumulative oil and water volume were monitored. The results of all data sets show a similar behavior in regard to the produced oil volume. More precisely, the produced oil of the structured uniform grid is the highest, whereas the oil production of the triangulated grids is smaller in every other case. This phenomenon seems to become more severe for higher coarsening factors but more drastically for the produced water volumes than the produced oil. (Figure 55, Figure 67, Figure 68)

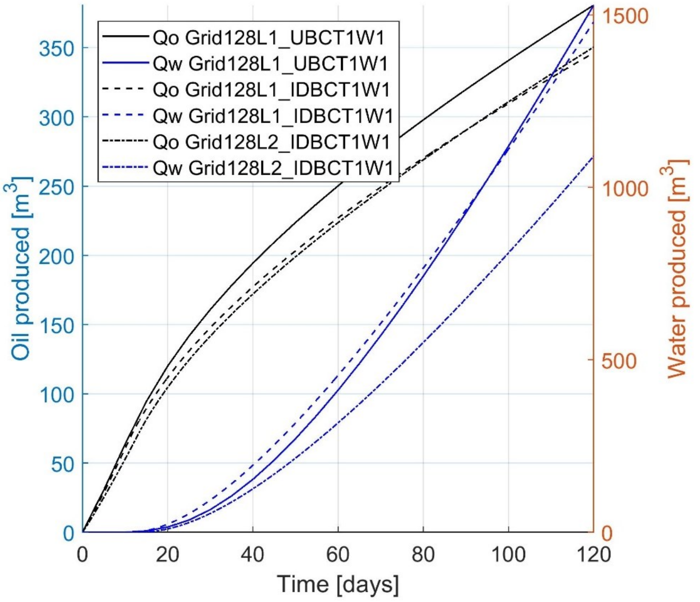


Figure 55: Cumulative volumes of different triangulated grids: 128x128m Grid

The vital question is what is causing this remarkable change? As already explained in the previous chapter, the difference of the grid structure causes a change of the calculated pressure field and therefore a difference in the produced fluid volumes. This has also an impact on the propagation of the saturation change. In order to explain the observed effect of different coarsening levels on the produced volumes a detailed look in the final saturation field of all three cases is important (Figure 56, Figure 69, Figure 70).

In accordance to the previous chapter, the propagation of the triangulated saturation front is slower than the structured uniform saturation front. This effect seems to increase as the coarsening level increases. As explained before, the difference of the pressure solution might cause this change in saturation propagation, but additional parameters also have an impact. To be more specific the calculation of the pressure is based on several parameters like porosity, permeability, viscosity and compressibility. Using this knowledge one can have a look at the porosity histogram that was generated during the mapping process. A remarkable feature is a shift of the property distribution to the higher porosity ranges, the larger the coarsening factor gets (Figure 57, Figure 71, Figure 72).

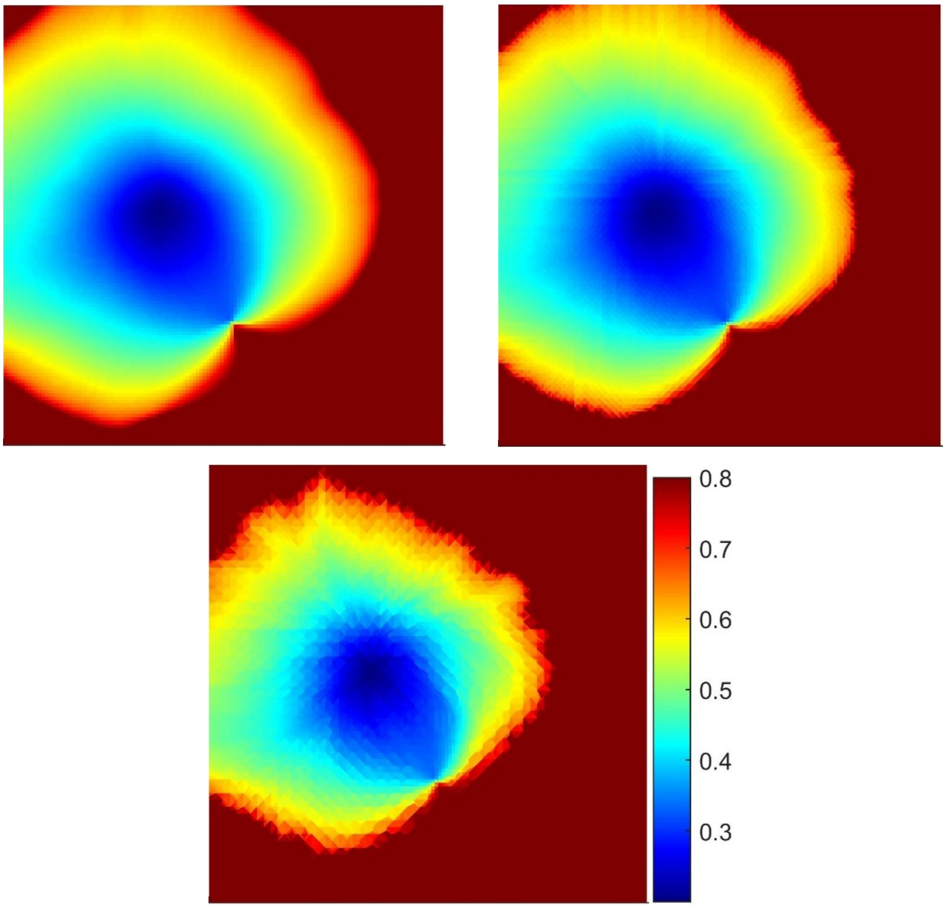


Figure 56: Saturation fields of Grid128L1_UBCT1W1 (top left), Grid128L1_IDBCT1W1 (top right) and Grid128L2_IDBCT1W1 (bottom)

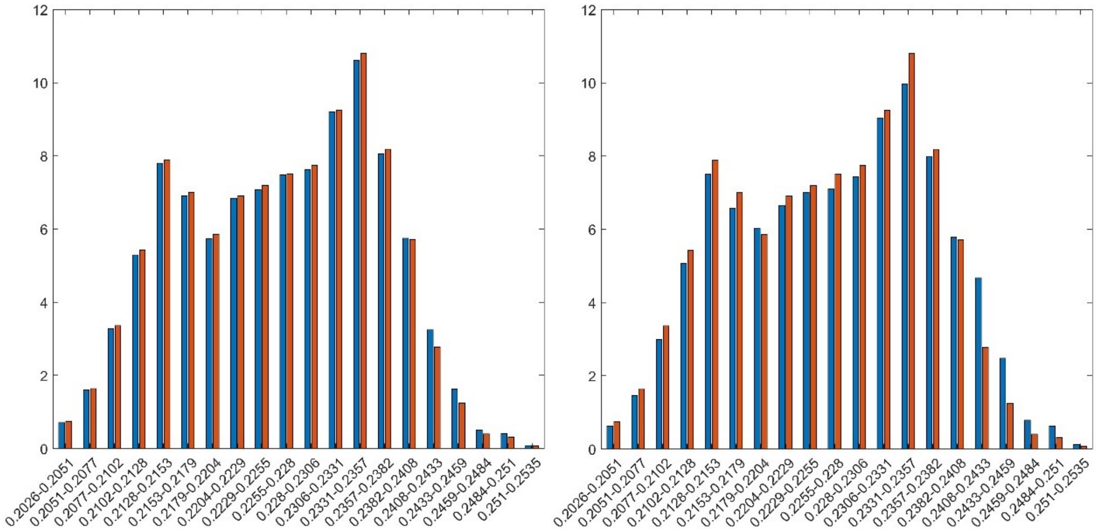


Figure 57: Shift of porosity distribution due to coarsening level on 128x128m Grid: first level (left), second level (right)

This feature is a vital fact if one thinks of the parameters that affect the calculation of pressure. For the current simulation- setup, this means that not only this change of porosity influences the pressure calculation, but also the permeability as it is calculated by a porosity-permeability correlation. Furthermore, by an increase of the porosity within a cell, also the volume of fluid in the cell increases, which ultimately affects the change of saturation. The reason for this is that if the wells produce a constant volume, more volume must be produced until the injected fluid reaches the production well. Ultimately, this means that the higher the coarsening factor, the later the breakthrough one can expect. This explanation perfectly fits to the observed simulation results as Figure 58, Figure 73and Figure 74 show.

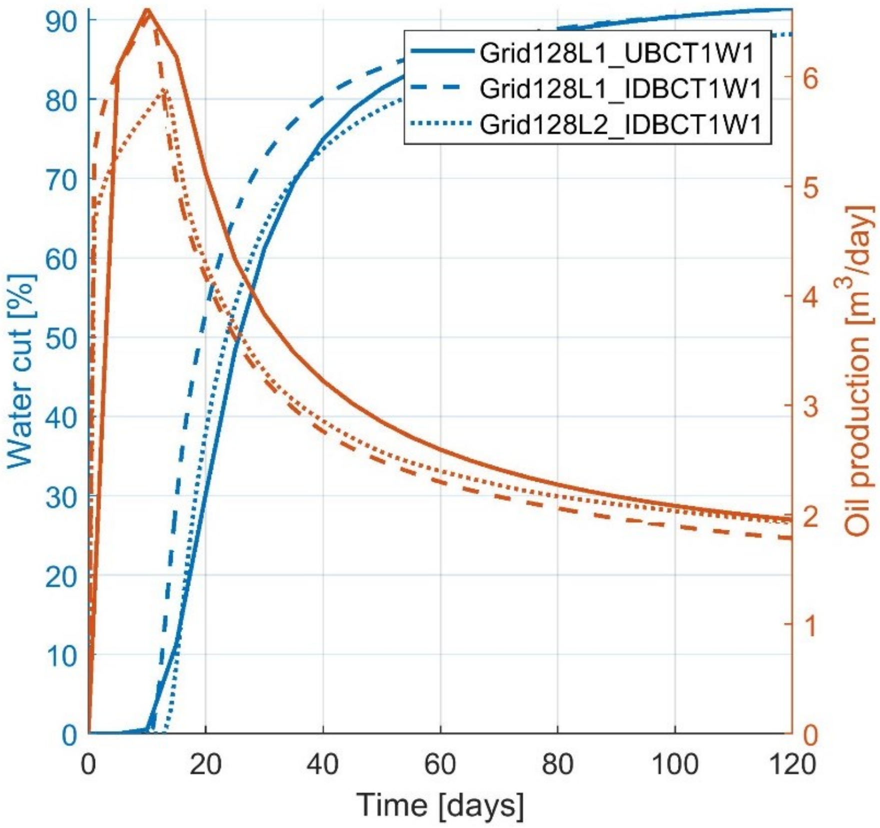


Figure 58: Comparison of oil production rate and water cut for different coarsening levels: 128x128m Grid

5.2.3 Well Placement

The last factor of interest investigated by this thesis is the effect of the well placement while comparing a triangulated and structured uniform grid. For this reason, the breakthrough times of both well-setups, explained in chapter 4.4, are compared. Having a close look at well setup W1, one can detect a slight difference in the breakthrough time if comparing the triangulated with the structured uniform grid.

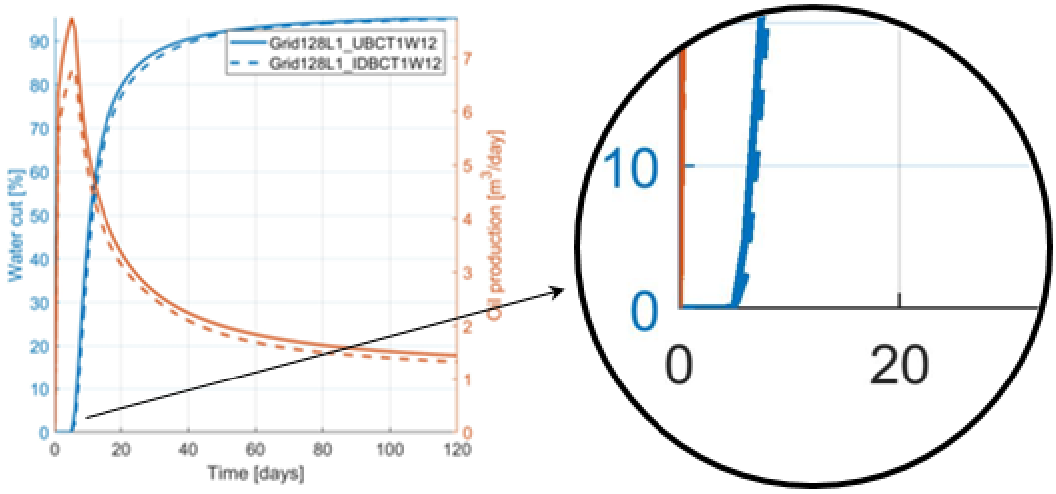


Figure 59: Well setup W1 breakthrough time: Grid128

Similar to the response of well setup W1, one can identify a difference in breakthrough time in setupW2 as well.

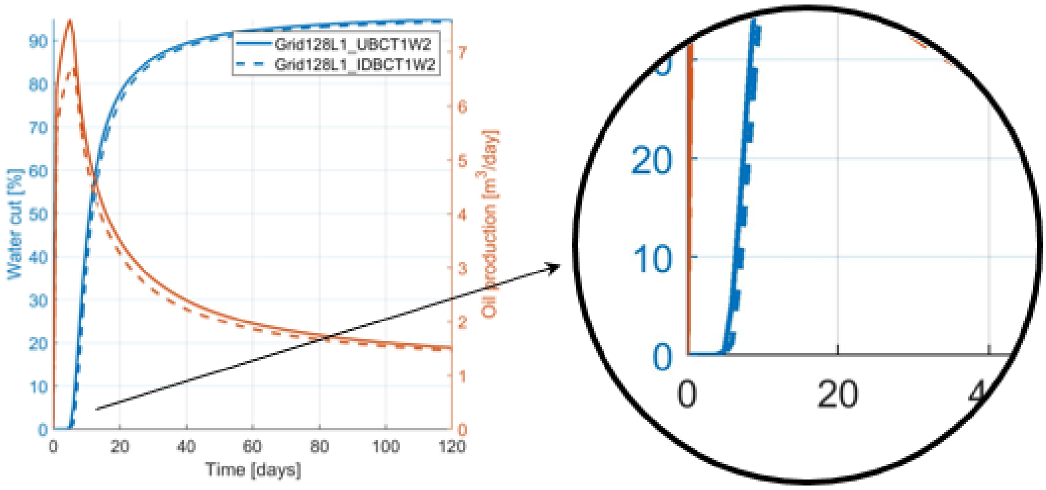


Figure 60: Well setup W2 breakthrough time: Grid128

As already identified before, both results indicate a later breakthrough if using a triangulated grid. Furthermore, no significant change in breakthrough time can be detected while comparing well setup W1 or W2. Figure 60 only shows a slightly increased time for breakthrough if using well setup W2. This difference can be caused on the one hand by the different grid structure or on the other hand by the difference in properties. In order to determine which of these effects cause the slight difference, a simulation of a constant porosity case with 20% is performed. As a result, the latter mentioned effect is eliminated; hence, if still a change is visible one can assume that the well placement and grid structure causes this difference in breakthrough.

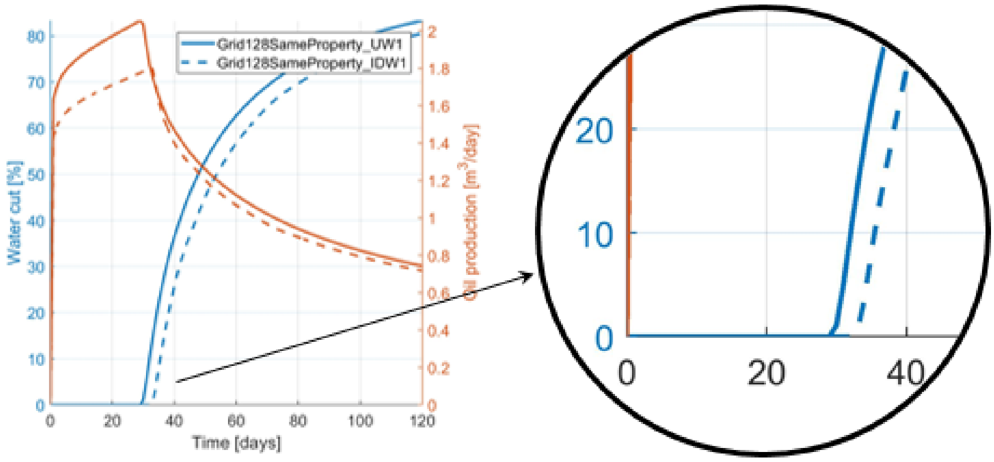


Figure 61: Well setup W1 breakthrough time: Grid128 with constant porosity of 20%

Figure 61 and Figure 62 show well setup W1 and W2, respectively. Both results do not indicate a recognizable change in breakthrough time, which means that the well placement does not seem to be an influencing factor.

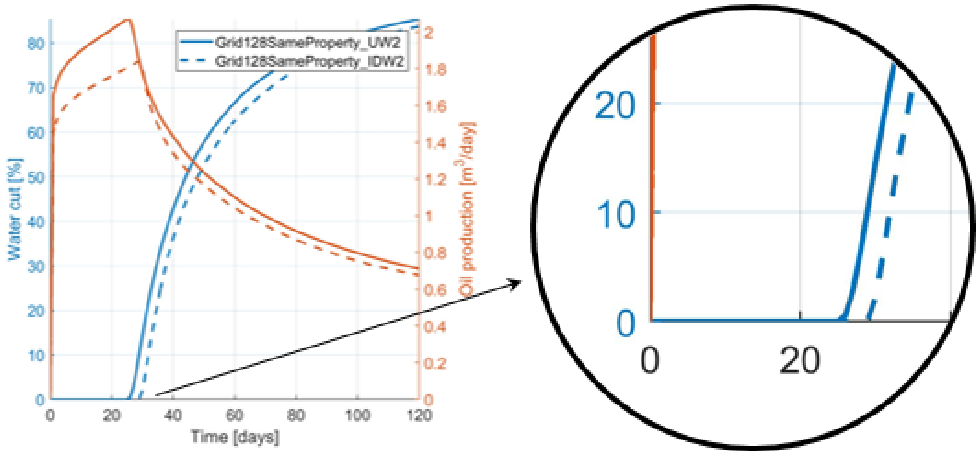


Figure 62: Well setup W2 breakthrough time: Grid128 with constant porosity of 20%

Chapter 6

Conclusion

6.1 Summary

The previous chapters gave a detailed overview on the assumptions made in the course of this thesis and show interesting results regarding both, the mapping algorithm and simulations. Concerning the mapping algorithm, one can identify a clear advantage of one of the simplest deterministic algorithms, the inverse distance weighting method, by the applied rating scheme. Factors of major investigation during the rating process were complexity of implementation, accuracy and runtime.

Based on the results of the inverse distance method, several simulation setups show that the utilization of a triangulated grid causes a reduction in the liquid production rate if comparing with a structured uniform grid. Furthermore, this effect seems to increase the larger the applied coarsening factor gets, having more impact on the produced water than the produced oil volumes. A comparison of the triangulated grid and structured uniform grid pressure fields, using image analysis, unsheathed a difference of both field. Ultimately, one can show that a possible driving force of this difference in pressure is a slight shift in the property histogram, applied during the mapping process.

6.2 Evaluation and Future Work

Concluding, all planned objectives are met in the course of this thesis. First, the most appropriate mapping algorithm for this special kind of grid conversion is identified. Secondly, several simulation runs show all effects on the screening parameters, whereas each observed phenomenon could be explained using the generated results.

Although, all objectives were met, one can still improve and further verify the obtained results. A major point of interest is the difference of the calculated pressure fields when using a triangulated or structured uniform grid. One can further investigate in the causes of this difference and try to show if only the difference in property distribution is the origin of this difference or if additional factors are of interest. A point to start would be for example the discretization scheme used for calculating pressure. The simulator, MRST, uses a finite volume approach to solve the pressure distribution at the barycenter's of the grid blocks. Compared to that, CSMP++ uses a finite element approach to solve for the pressure distribution at the grid nodes and interpolates the pressure at the barycenter. Comparing on the one hand the solution of the finite volume approach and on the other hand, the finite element approach might be an interesting additional value and give insight in changes if different discretization schemes are used.

Additionally to that, also an improvement of the triangulation and the mapping can be done by implementing an automatic constraint algorithm as explained in chapter 3.1. Applying this algorithm might have an effect on the simulation result and will allow representing a better picture of reality.

Furthermore, an extension of the code including the possibility to process three-dimensional data and simulate it afterwards would be interesting.

Chapter 7

References

Cheng, S.-. W., Dey, T. K. & Shewchuk, J. R., 2013. *Delaunay Mesh Generation.* Boca Raton: Taylor & Francis Group, LLC.

Christie, M. A. & Blunt, M. J., 2001. *Tenth SPE Comparative Solution Project: A Comparision of Upscaling Techniques.* Houston: SPE.

De Loera, J., Rambau, J. & Santos, F., 2010. *Triangulations: Structures for Algorithms and Applications.* Berlin Heidelberg: Springer.

Doyen, P., 2007. *Seismic Reservoir Characterization: An Earth Modelling Perspective.* Houten: EAGE.

Gärtner, B., Hoffmann, M. & Welzl, E., 2015. *Geometry: Combinatorics & Algorithms.* Zurich: ETH Zurich.

Kallmann, M., Bieri, H. & Thalmann, D., 2003. *Fully Dynamic Constrained Delaunay Triangulations,* s.l.: USC Robotics Research Lab.

Kelkar, M. & Perez, G., 2002. *Applied Geostatistics for Reservoir Characterization.* s.l.:Society of Petroleum Engineers.

Kitandis, P., 1997. *Introduction to Geostatistics: Applications to Hydrogeology.* Stanford: Cambridge University Press.

Shewchuk, J. R., 1996. Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator. *Applied Computational Geometry: Towards Geometric Engineering,* Volume 1148, pp. 203-222.

Su, P., 1994. *Efficient Parallel Algorithms for Closest Point Problems.* Hanover, New Hampshire: Dartmouth College.

Su, P. & Drysdale, S., 1997. *A comparison of sequential Delaunay triangulation algorithms,* Pittsburgh, Hanover: Elsevier.

Timur, A., 1968. *An Investigation of Permeability, Porosity & Residual Water Saturation Relationships for Sandstone Reservoirs.* Louisiana: SPWLA Logging Symposium.

Webster, R. & Oliver, M. A., 2007. *Geostatistics for Envoronmental Scientists.* 2nd ed. Chichester: John Wiley & Sons, Ltd.

Appendix A

Appendix

A.1 Simulation Results

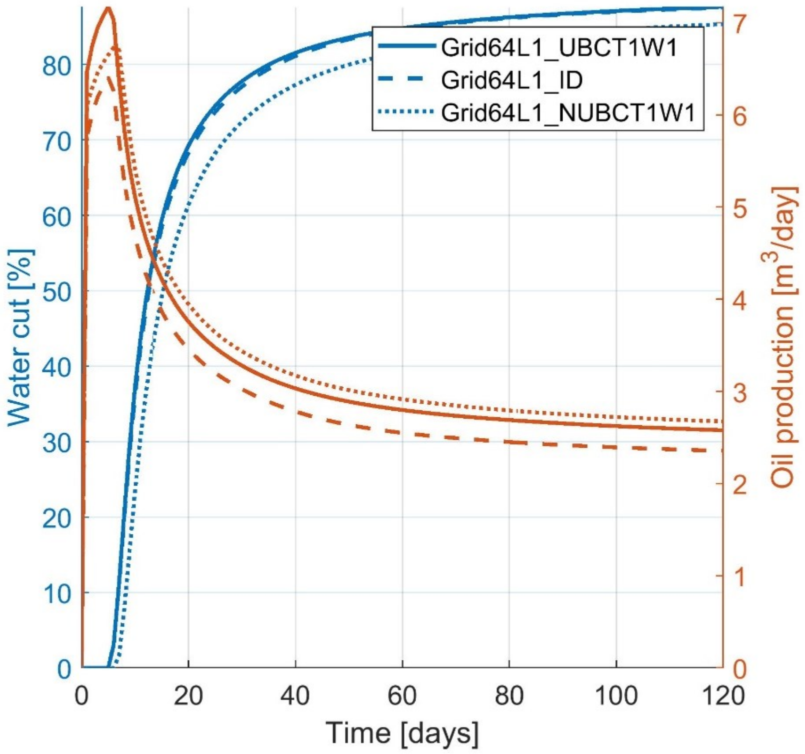


Figure 63: Comparison of oil production rate and water cut from different grids: 64x64m Grid

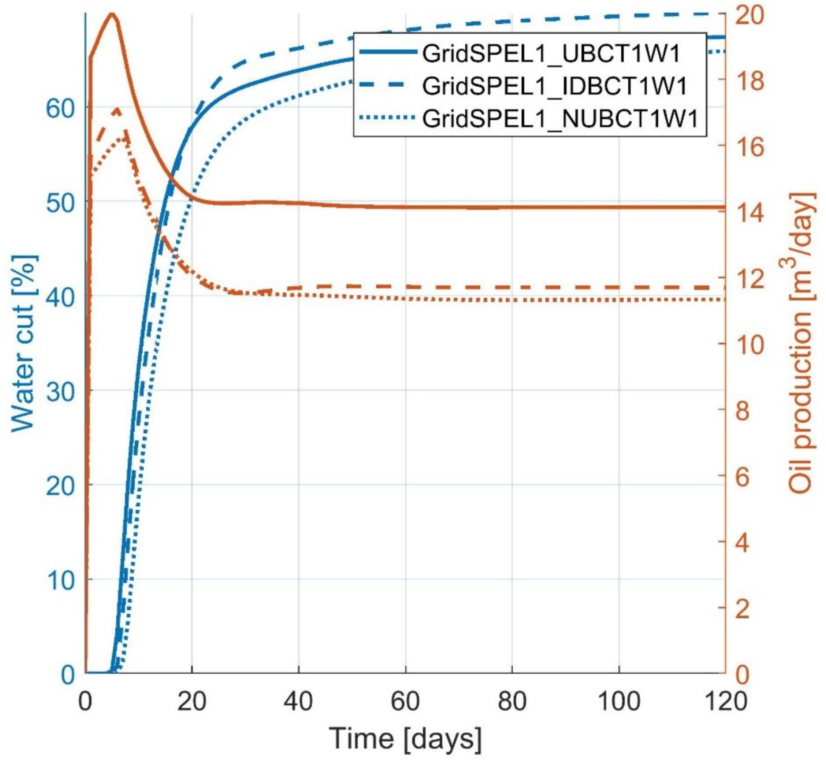


Figure 64: Comparison of oil production rate and water cut from different grids: SPE10 Grid

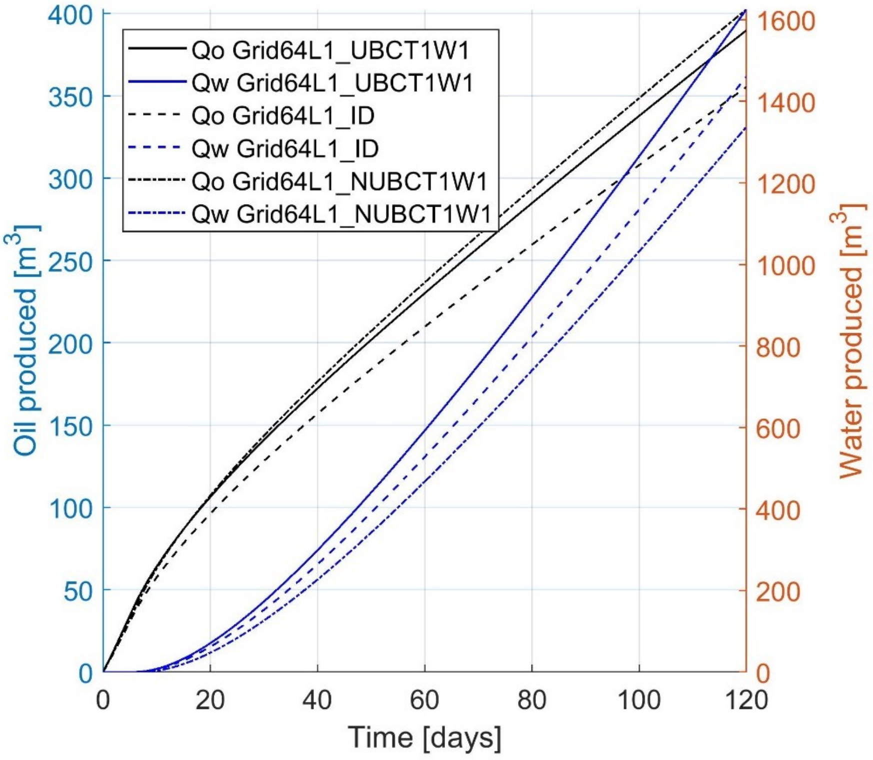


Figure 65: Cumulative produced volumes comparing different grid- structures: 64x64m Grid

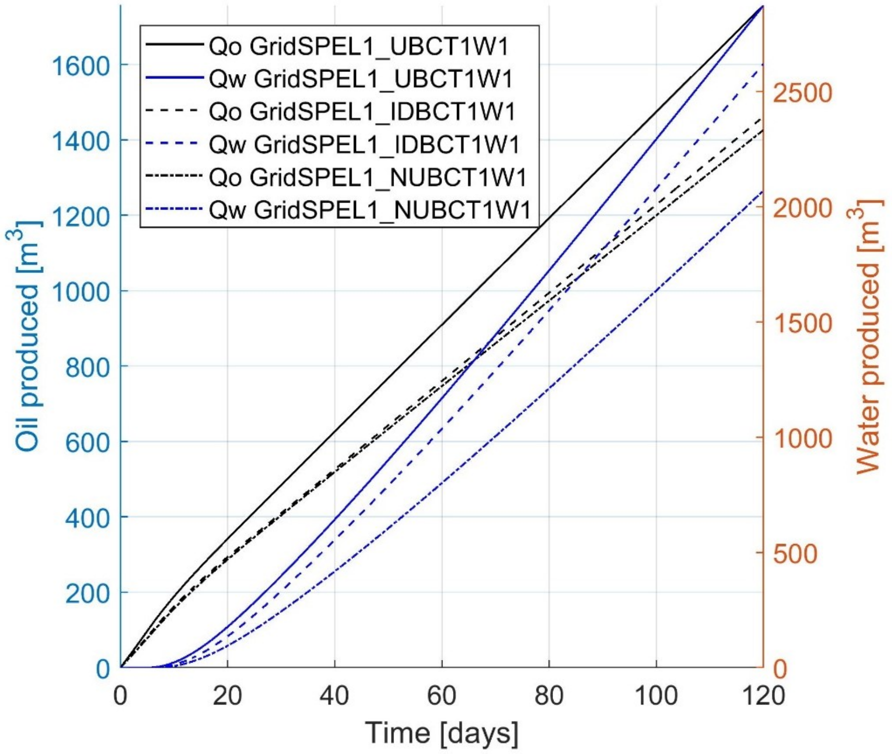


Figure 66: Cumulative produced volumes comparing different grid- structures: SPE10 Grid

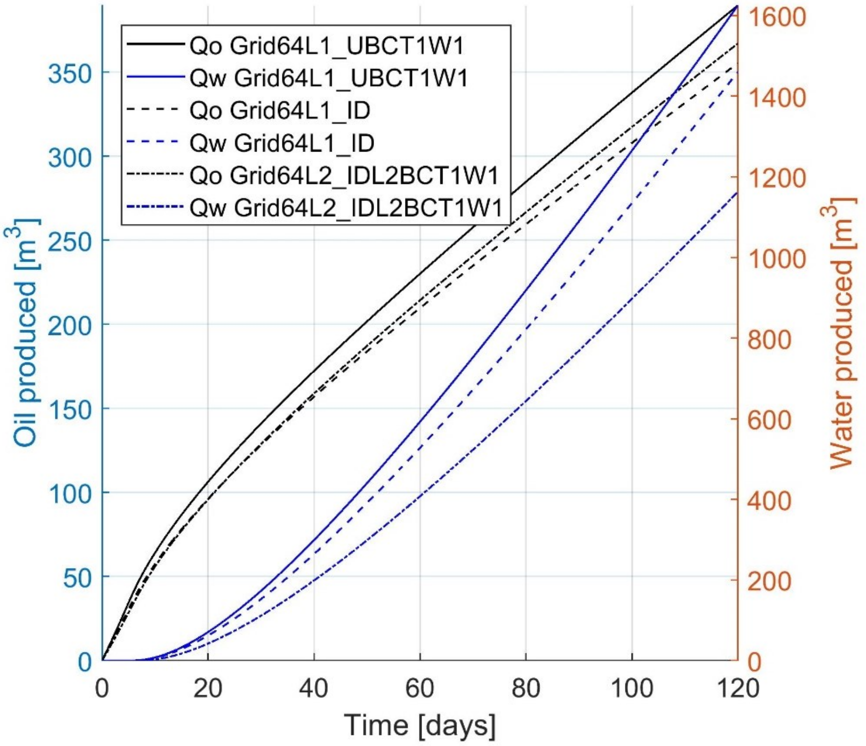


Figure 67: Cumulative volumes of different triangulated grids: 64x64m Grid

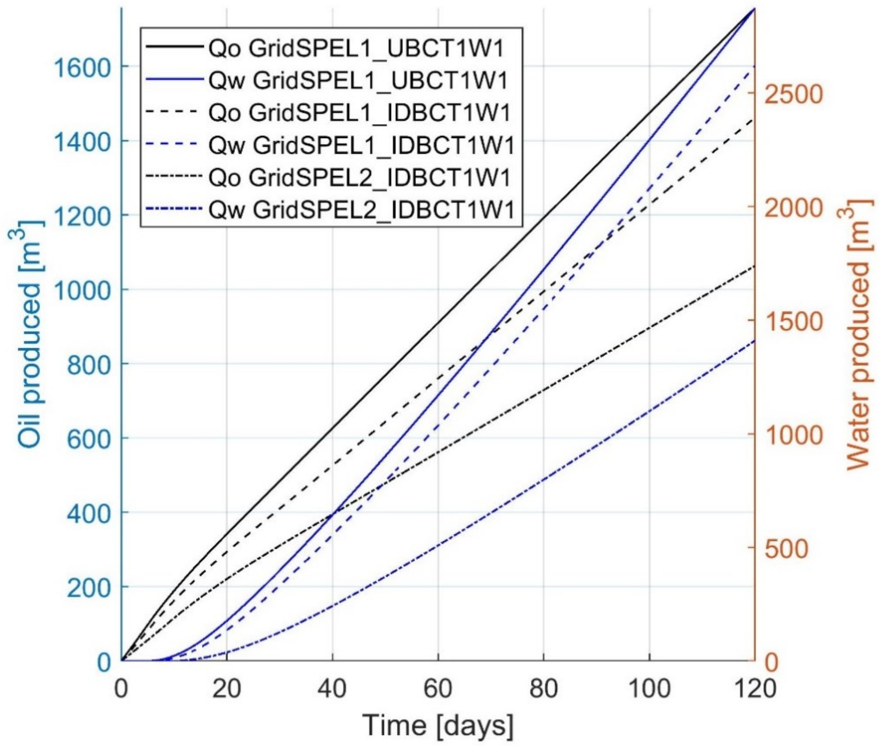


Figure 68: Cumulative volumes of different triangulated grids: SPE10 Grid

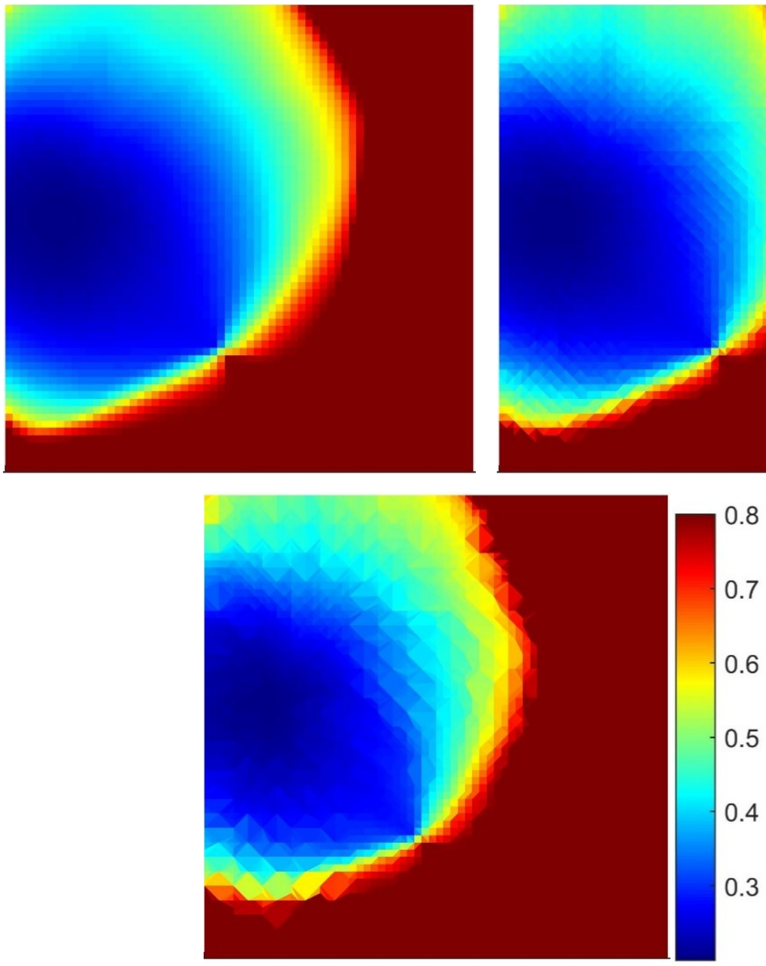


Figure 69: Saturation fields of Grid64L1_UBCT1W1 (top left), Grid64L1_IDBCT1W1 (top right) and Grid64L2_IDBCT1W1 (bottom)

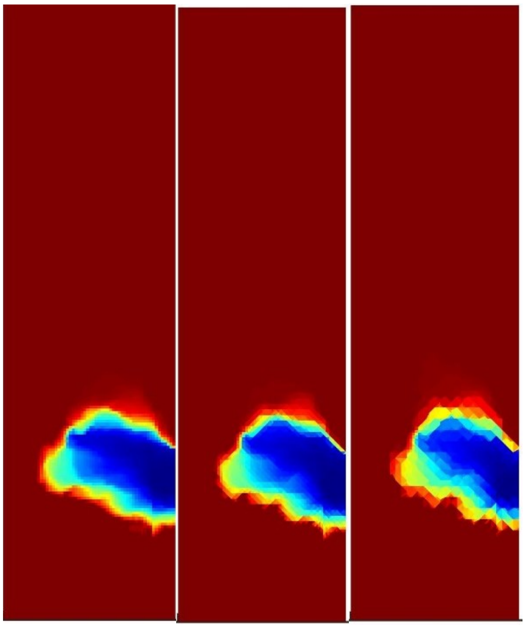


Figure 70: Saturation fields of GridSPE10L1_UBCT1W1 (top left), GridSPE10L1_IDBCT1W1 (top right) and GridSPE10L2_IDBCT1W1 (bottom)

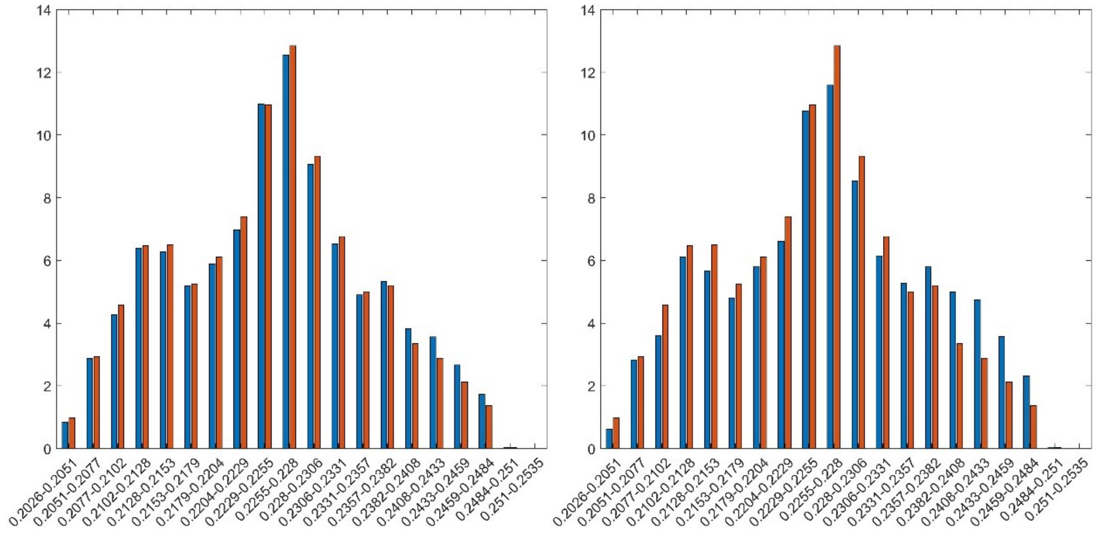


Figure 71: Shift of porosity distribution due to coarsening level: 64x64m Grid

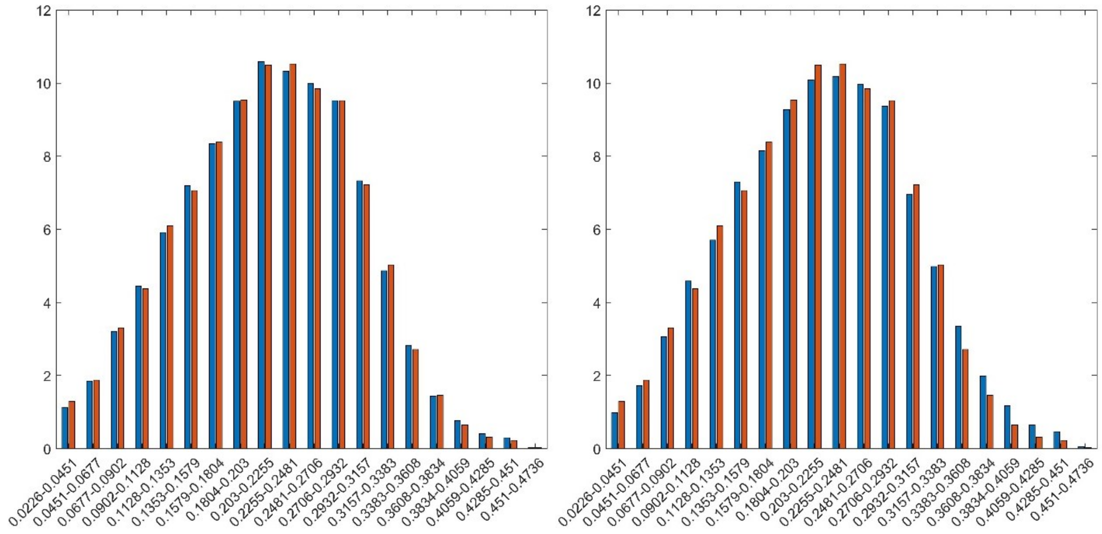


Figure 72: Shift of porosity distribution due to coarsening level: SPE10 Grid

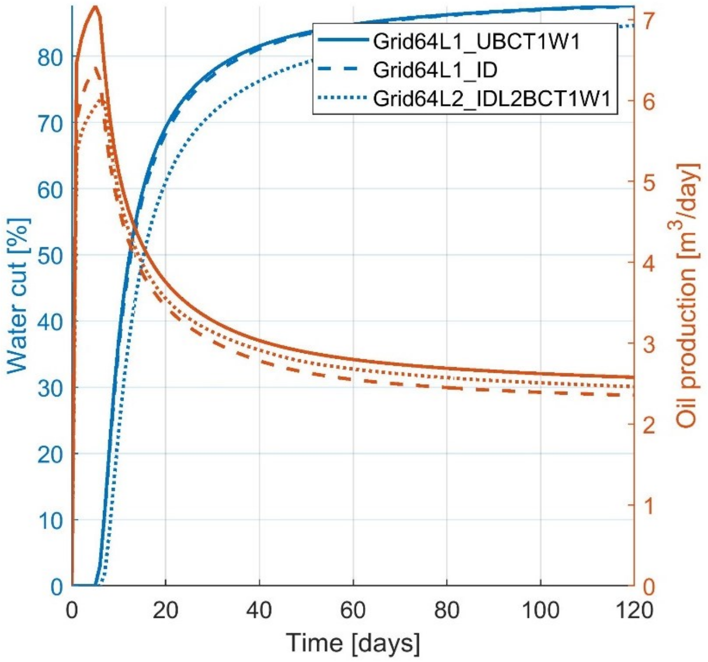


Figure 73: Comparison of oil production rate and water cut for different coarsening levels: 64x64m Grid

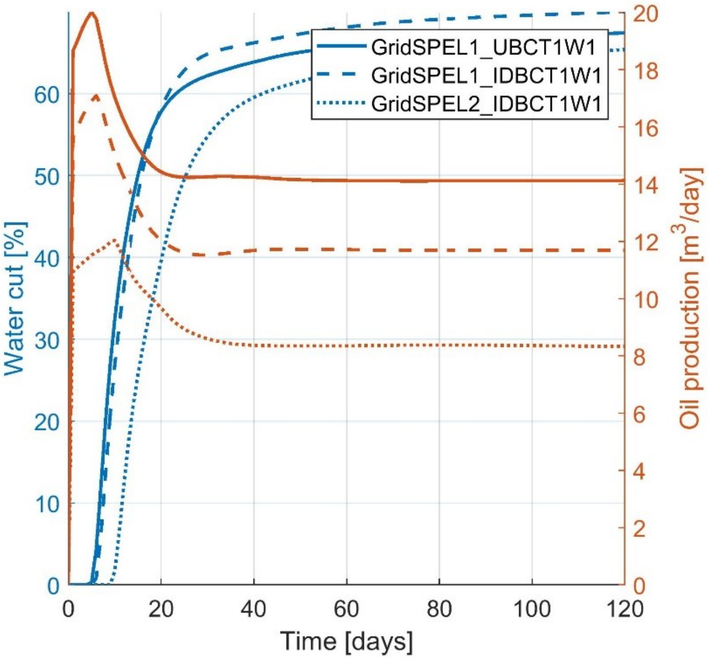


Figure 74: Comparison of oil production rate and water cut for different coarsening levels: SPE10 Grid