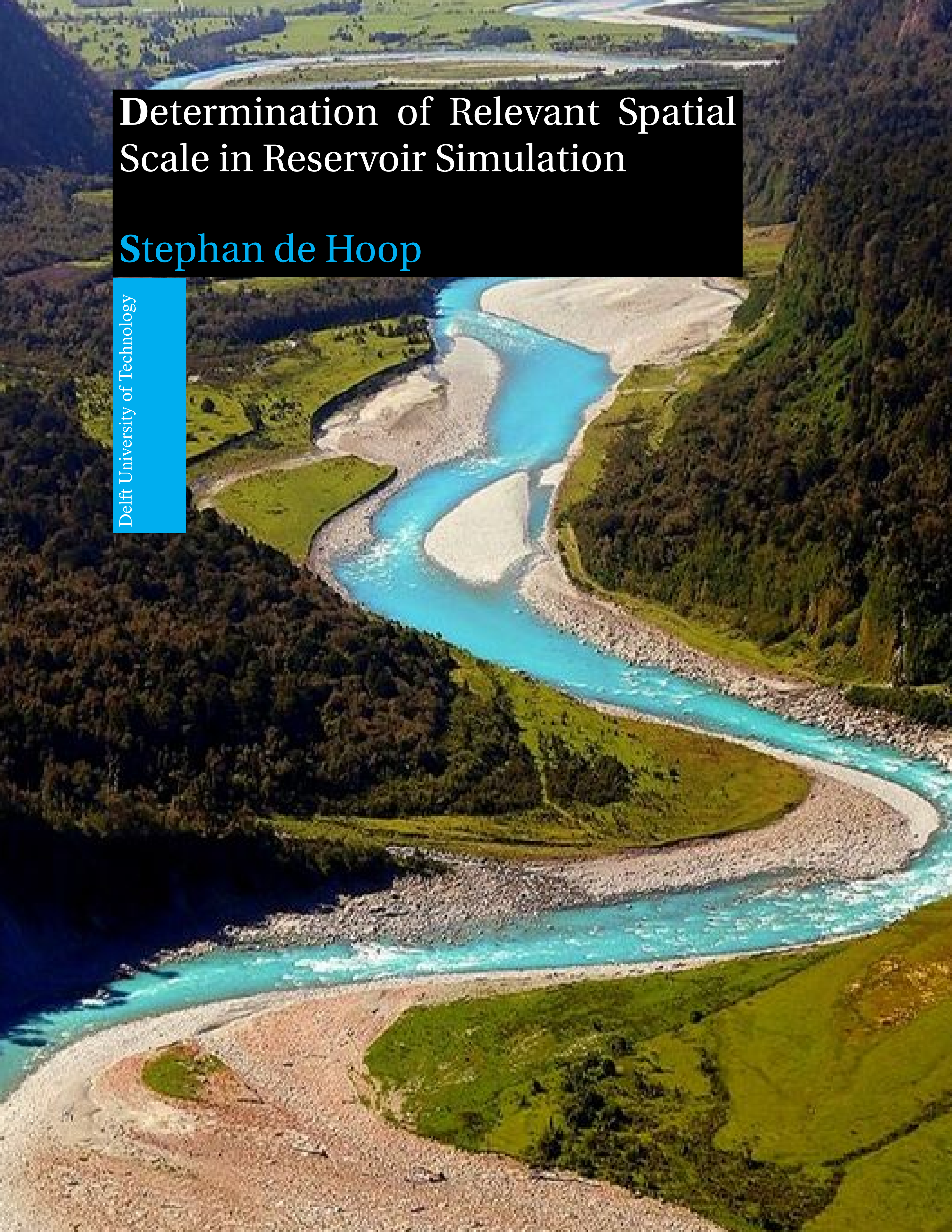


Determination of Relevant Spatial Scale in Reservoir Simulation

Stephan de Hoop

Delft University of Technology

Determination of Relevant Spatial Scale in Reservoir Simulation

by

Stephan de Hoop

in partial fulfillment of the requirements for the degree

Master of Science
in Petroleum Engineering and Geosciences

at the Delft University of Technology,
to be defended publicly on Thursday November 16, 2017 at 10:00.

Supervisors:	Dr. Denis Voskov	Associate Prof. TU Delft
	Dr. ir. Femke Vossepoel	Associate Prof. TU Delft
Committee:	Dr. Denis Voskov	Associate Prof. TU Delft
	Dr. ir. Femke Vossepoel	Associate Prof. TU Delft
	Dr. Joep Storms	Associate Prof. TU Delft
	Prof. Dr. Giovanni Bertotti	Professor TU Delft
	Dr. Andre Jung	Shell

Abstract

Under-sampling of the subsurface combined with scale differences in observations causes the estimation of geological parameters to be an ill-posed problem. As a result, only a subset of theoretically possible models can truly depict the reality. In reservoir modeling, we capture complexity and heterogeneity by representing the solution space with a large number of high-resolution models, whose spread represents uncertainties in permeability. It is not immediately evident at which scale the static and dynamic model should be formulated. Therefore, this thesis work attempts to determine a relevant spatial scale in reservoir simulation. The relevant spatial scale is subdivided into a static and dynamic spatial scale. The static one is analyzed using the Discrete Cosine Transform (DCT). A dominant basis-vector is determined which explains the predominant pattern in the reservoir model. The associated dimensions to accurately represent this basis-vector is chosen as relevant static spatial scale. A hierarchical ensemble is established using a flow-based upscaling approach, in an attempt to quantify the coarsening effect.

The hierarchical ensemble of models is simulated forward in time to represent fluid flow and associated uncertainties in its response. Dynamic analysis is done on a reduced representation of the response uncertainty, obtained via Multidimensional Scaling (MDS). An Uncertainty Trajectory is built in order to analyze the effect of time on the response uncertainty. The distance from the finest uncertainty trajectory is used to quantify the coarsening effect on a dynamic level. It is shown that the characteristics of the coarser ensemble scale behave similarly to the finest ensemble scale. This observation has led to the use of coarse information in the prediction of representative fine-scale models. Where representatives refer to using them as a subset to approximate the full fine-scale ensemble statistics.

keywords: simulation, discrete cosine transform, uncertainty quantification, clustering.

Acknowledgments

First of all, I would like to thank my supervisor Dr. Denis Voskov who has inspired me throughout my whole Master program and without whom this project would not have been possible. His excitement and ambitious mindset have always been encouraging. I am very grateful for all the help he has given me the past two years. I am very much looking forward to our future research endeavors.

Secondly, I would like to express my sincere gratitude to my supervisor Dr. Femke Vossepoel who has always provided me with valuable insights as well as keeping me on a schedule when I needed it most. Her positive attitude towards doing research are invaluable as well as always finding time for a proper discussion.

Both supervisors motivated me and gave me the opportunity to present at the SIAM Conference, an experience so valuable to me. Their contribution and support are highly appreciated.

Many thanks to Dr. Andre Jung for his immediate interest in my thesis project and willingness to assist at all times. He has been almost an additional supervisor and always showed me a warm welcome at Baker Hughes. Our discussions have greatly helped me grow as a scientist.

Hereby I would also like to thank Prof. Dr. Giovanni Bertotti and Dr. Joep Storms for being part of my committee. Besides this, both of them have been a big inspiration for me to even pursue a Master at the Delft University of Technology and for this, I am very grateful.

At this point, I would like to thank my parents for their unconditional love and support, both emotionally as well as financially. I would also like to thank my brother Richard, who has shaped me into the person I am today. Furthermore, I am grateful for having Robin and Bart as my closest friends who stood up with my constant nagging about being busy. I also want to thank Jasper for all the moments that we shared in our educational journey.

Finally, I want to thank Natalia Papatrecha, who took care of me when I needed it most. She has stood next to me, motivating me whenever I felt I couldn't do it. I will never forget as well the long nights of hard work which have led to both our success. We made it!

Contents

List of Figures

List of Tables

List of Abbreviations

2D Two-dimensional.

AD-GPRS Automatic Differentiation - General Purpose Reservoir Simulator.

AIM Adaptive Implicit Method.

BHP Bottom hole pressure.

CFL Courant–Friedrichs–Lewy.

CTU Corner-Transport Upwind.

DCT Discrete Cosine Transform.

DCU Donor-Cell Upwind.

FCT Fourier Cosine Transform.

FFT Fast Fourier Transform.

FIM Fully Implicit Method.

HF High Fidelity.

IMPALA Improved Parallel Multiple-point Algorithm Using a List Approach.

IMPES Implicit Pressure Explicit Saturation.

IMSAT Implicit Pressure and Implicit Saturation.

KLT Karhunen-Loeve Transform.

MDS Multidimensional Scaling.

MPS Multiple Point Statistics.

NG Net to Gross.

OP Orthogonal Procrustes.

PDE Partial Differential Equation.

RBF Gaussian Radial Basis Function.

RHS Right-hand side.

SIS Sequential Indicator Simulation.

SVD Singular Value Decomposition.

TI Training Image.

TOF Time of Flight.

TPFA Two-Point Flux Approximation.

1

Introduction

1.1 Problem Statement

Reservoir modeling consists of integrating several data sources in order to construct a geological model. Subsequently, fluid flow is simulated in aid of understanding the dynamic behavior of the particular (petroleum/geothermal) reservoir. Reservoir performance is greatly affected by heterogeneity, resulting from the sedimentary architecture (Haldorsen, 1986) and (Nordahl & Ringrose, 2008). In order to assess reservoir performance and predict future production behavior, the spatial distribution of lithological and petrophysical properties is required in a discretized subsurface model. Due to under-sampling of the subsurface and scale differences across measurements, the spatial distribution of these parameters is typically unknown. This uncertainty is represented through a set of equally probable reservoir models, also referred to as an ensemble of models.

Figure 1.1 shows measurements of permeability generally available in a Field Development Plan, ranging from micro- to macro-scale observations. Sub-meter scale fluctuations are classically included through upscaling of core-scale measurements to a representative value on a larger scale, referred to as the high fidelity scale at which the geological model is constructed. The spatial discretization scale at which the static (geological) and dynamic (reservoir simulation) model should be formulated is not immediately evident from the obtained measurements nor from the assumed geological conditions under which the sediments were deposited.

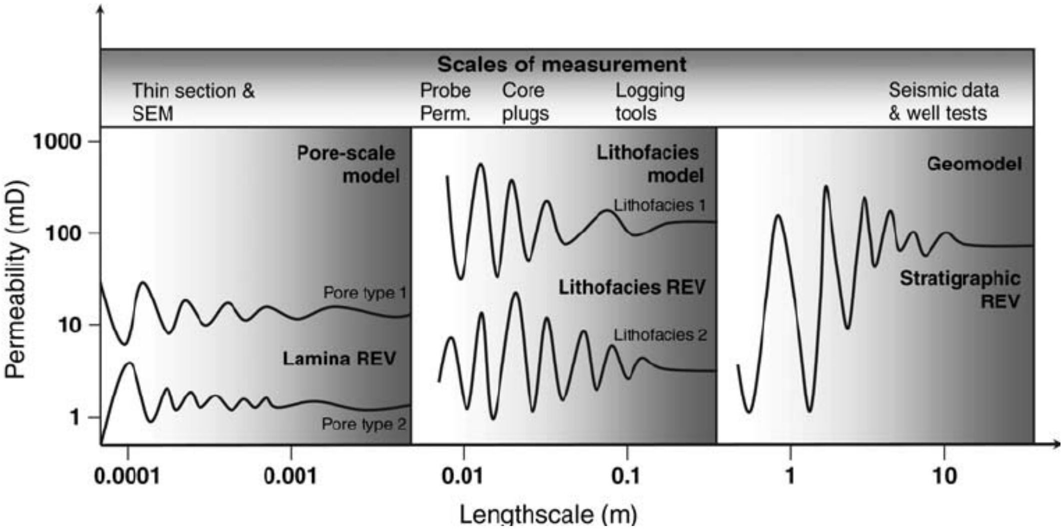


Figure 1.1: The permeability measurements across a variety of length scales, based on (Nordahl & Ringrose, 2008).

Ultimately, the objective of any reservoir characterization and simulation procedure is to assess the economic risk associated with the particular investment. To minimize this risk, quantification of the spatial uncertainty is inevitable. Please note that there is no such thing as “right” or “correct” uncertainty. This is explained in great detail by Caers (2011), who uses the analog of weather predictions: if the weather forecast predicts a 60% chance of rain, and it doesn’t rain, there is no way in which the quality or “correctness” of this uncertainty can be verified. In reservoir modeling, the same principle applies. Knowing the “right” uncertainty requires knowing the true state of the Earth (or subsurface) which subsequently nullifies the need of uncertainty quantification in the first place (it’s a paradox). Uncertainty assessment is therefore subjective, but if done in a systematic way, can lead to improved decision making (Caers, 2011).

Large ensemble sizes are typically required to represent the aforementioned spatial uncertainty (depending on the expected complexity of the reservoir) and the resulting quantification of dynamic uncertainty (also referred to as response uncertainty) is done through forward simulation of the static models in time. Depending on the size of the discretized problem, this can become a computationally expensive task, and, when explicitly including micro-scale fluctuations, it becomes virtually impossible[1] considering the current computational resources (Nordahl & Ringrose, 2008). Characterization of micro-scale pore-structures using tomography and micro-imaging is an existing and ongoing research, see e.g. (Jungreuthmayer et al., 2015), (Sok et al., 2010) or (C. Chen, Hu, Westacott, & Loveless, 2013), however inverse estimation of these micro-scale fluctuations by macroscopic well measurements (such as pressure and flow rate) is currently an unexplored area.

Ignoring over-fitting a model[2], the finest scale possible to perform reservoir simulation is generally preferred due to the numerical accuracy of the simulation. Given a certain modeling purpose, knowledge of the existence of a relevant (macroscopic) spatial scale at which the reservoir model should be formulated, can greatly assist in the uncertainty quantification process. If a relevant spatial scale exists, the response uncertainty at this particular scale should then converge to the finest scale response uncertainty. Knowing this relevant spatial scale, if coarser than the finest scale, it is possible to decrease the computational costs while still making (financial) decisions on the same response uncertainty.

The non-linear mapping (also referred to as transfer function) from the parameter space to the solution space (production data, etc.) requires investigating the existence of a static and dynamic relevant spatial scale. Even a simple dead oil model may exhibit non-linear behavior in its solution, mainly due to the dependency of relative permeability on saturation. The importance of this concept is illustrated in figure 1.2 where two seemingly different realizations (regarding the Euclidean distance between their model parameters) exhibit almost equivalent behavior in their solution/response.

---

[1] Considering the vast amount of realizations required to represent the spatial uncertainty associated with the characterization of micro-scale fluctuations.

[2] Over-fitting is the problem of learning/explaining the training data rather than recognizing patterns, which has negative implications on the predictive value. A prime example is given by (Bishop, 2006) Chapter 1 p.7, in the context of fitting a polynomial curve to some measurements. Increasing the degrees of freedom (model parameters) in this example improves the fit with the observations, however at the costs of spurious oscillations decreasing the predictive value of the model away from the data points.

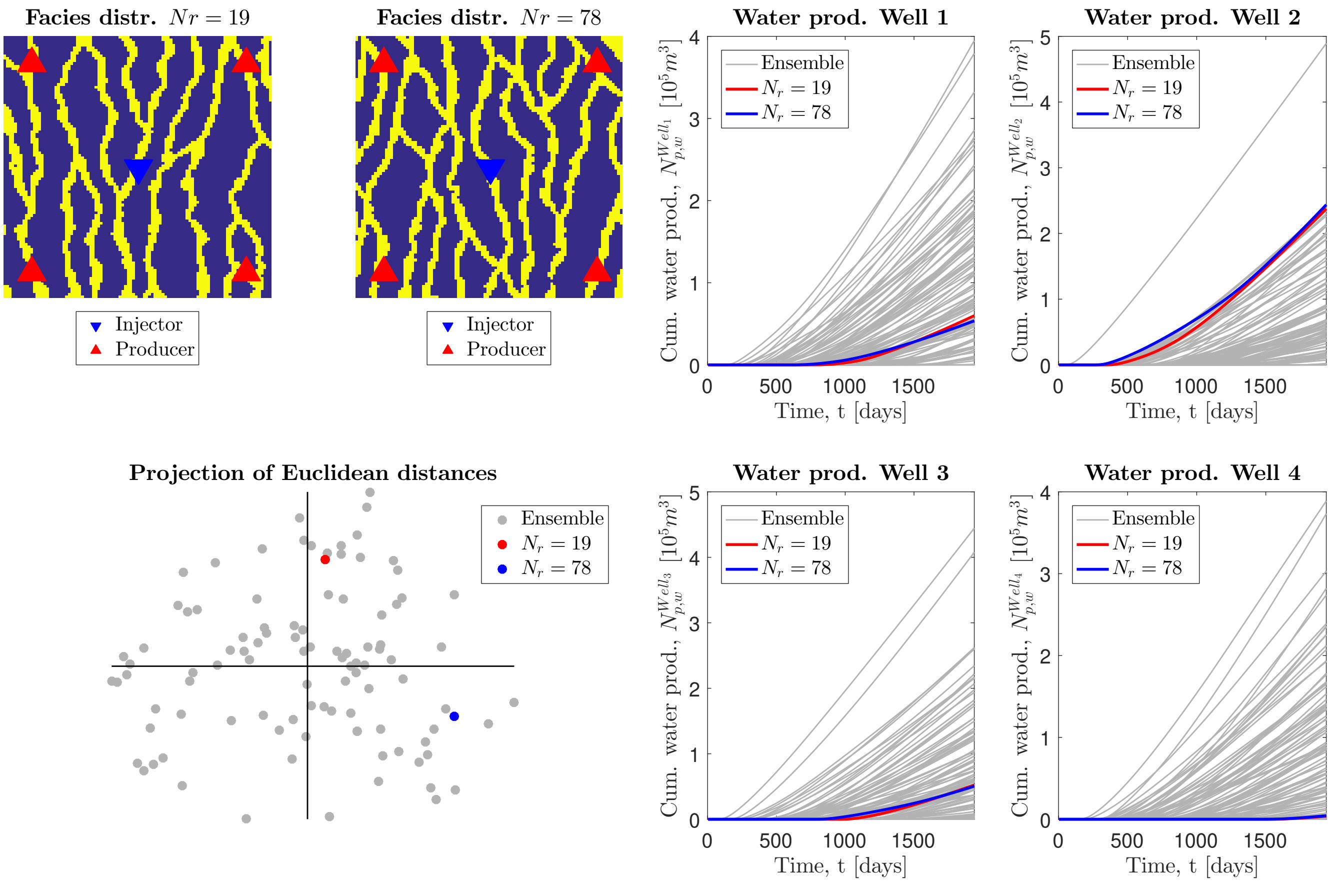


Figure 1.2: Depicting statistically equivalent flow behavior in terms of water production in each well, for two realizations which are seemingly different w.r.t. the Euclidean distance of the model parameters.

1.2 Objectives and method

The main research question is, therefore: Does there exists a relevant spatial scale in reservoir simulation? The resulting thesis objectives are:

- Establish hierarchical ensemble of reservoir simulation models.
- Understand and quantify the coarsening effect on the response uncertainty.
- Understand the extent to which coarse information can be effectively used to simulate flow behaviour.

In order to achieve the thesis objectives, the following workflow is proposed in figure 1.3.

The first step of the workflow is the generation of M “geologically realistic” reservoir models, denoted the High fidelity (Hf) M-dimensional ensemble of models. The hierarchical ensembles are created by coarsening (and refining) the Hf ensemble. Analyzing the coarsening effect on the prior (static) information is done using the Discrete Cosine Transform (DCT) on the transmissibility fields across all ensemble scales. Dynamic data is obtained by forward simulating all ensemble scales in time using AD-GPRS (Stanford, 2012) and (D. Voskov, Zhou, & Volkov, 2012), representing the fluid flow of a two-phase dead oil model (slightly compressible) in a 2D reservoir. Subsequently performing distance-based modeling of the response uncertainty in Metric Space results in quantifying the coarsening effect and possible understanding of a relevant dynamic spatial scale.

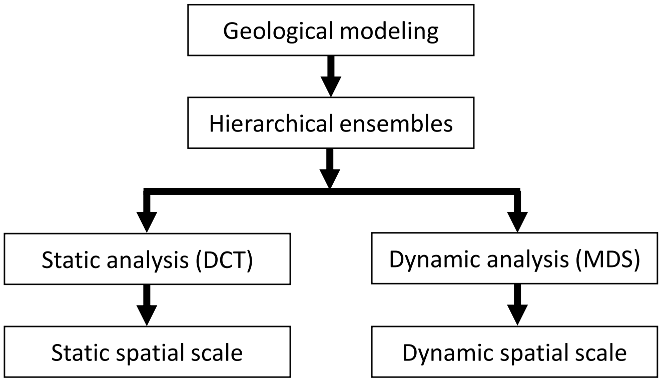


Figure 1.3: Simple schematic depicting the workflow adopted in this work.

1.3 Document Structure

The thesis report is structured in the following way. Acquiring the ensemble of models is briefly explained in Chapter 2. The necessary theoretical background for the Uncertainty Quantification and determination of relevant spatial scale is thoroughly explained in Chapter 3. The actual analysis and results are presented in Chapter 4 and 5, while the discussion and conclusion can be found in Chapter 6 and 7.

2

Geological Modeling

Generation of geological models constitutes an important key element in the quantification of uncertainty for subsurface flow problems (Sun & Durlofsky, 2017). The uncertainty in the parameters, due to undersampling of the subsurface and different scales across the measurements, are generally represented by a set of equally probable geological realizations, referred to as an ensemble of models. The ensemble of models should be complex enough in order to capture the reservoir heterogeneities and reflect a more representative image of the subsurface, compared to an oversimplified model. This thesis work, therefore, tried to avoid simplifications in the reservoir modeling process, since the objective of this work is to determine a relevant spatial scale, and oversimplified models might show a bias towards a coarser spatial scale. A fluvial depositional setting is chosen to be modeled due to the increased interest regarding flow prediction in fluvial reservoirs (Ortiz & Deutsch, 2002) and (Sun & Durlofsky, 2017). In detail, a channelized reservoir of two main facies (reservoir and non-reservoir) constitutes the simulated system since the distribution of the channelized sedimentary bodies affects the reservoir connectivity and therefore the flow (Rongier, Collon, & Renard, 2017). Also note that for the reservoir models used in this work, it is assumed that the reservoir is geologically young and therefore the main characteristics of the reservoir are explained by sedimentary processes only (diagenesis and compaction play a minor role) (Galloway & Hobday, 2012).

The generated models correspond to a fluvial depositional environment comprised of clay-rich floodplain and sand dominated channels. Fluvial reservoirs are typically complex, displaying large permeability contrasts between channel sands and the shale matrix (Henriquez, Tyler, Hurst, et al., 1990). For that reason, this thesis work uses two main facies (sand and clay, i.e. reservoir and non-reservoir) since the fluvial environment, due to its highly heterogeneous nature, constitutes a big challenge up to date in the petroleum research.

A subset of the fluvial depositional environment is the meandering environment (highly sinuous), where the main reservoir sands are deposited through migration of the channels in the form of point bars (Galloway, 1981), (Schumm, 1981) and (Ethridge & Schumm, 1977). For generating the aforementioned environments, we used the FLUMY software and adopted the sedimentological model proposed by Donselaar et al.,(2008). In this model, it is assumed that point bars are not isolated entities but connected through (thin) cross-bedded channel floor deposits caused by aggradational conditions due to upward movement of the graded profile (Miall, 2013) and (Donselaar & Overeem, 2008). This reservoir architecture is assumed to apply to mixed-load low-gradient meandering rivers under conditions of increasing accommodation space (Donselaar & Overeem, 2008).

2.1 Modeling approaches

Primary control on the flow performance of a reservoir is the spatial distribution of depositional facies (Strebelle & Levy, 2008). The typical order of constructing a reservoir model is therefore to first model the depositional facies followed by populating the facies with petrophysical properties accordingly. Geologically realistic facies distribution has shown to positively influence the representation of the uncertainty and variability of key reservoir parameters, as well as provide more accurate history matching results (Keogh, Martinius, & Osland, 2007) and (Demyanov, Rojas, Arnold, & Christie, 2013). There are several methods for establishing a geologically realistic reservoir model, e.g. a process based or stochastic approach. Besides geological realism, the ability of data conditioning is another key element in the attempt to understand or predict subsurface flow processes and its associated uncertainty.

Process-based modeling offers highly detailed geologically realistic reservoir models, however, due to the forward simulation in time, the ability to be conditioned to hard data is limited (Michael et al., 2010) and (Hoffimann, Scheidt, Barfod, & Caers, 2017). Another drawback of the process based modeling manifests itself in the computational effort of acquiring a single realization/geological model. Creation of a large ensemble of models, by variation of the input and boundary conditions (e.g. sea level rise, sediment supply), can become a tedious task. There are cases where it is shown that data condition for process-based modeling is possible in principle (Karssenberg, Tornqvist, & Bridge, 2001). However, in this particular case, it is merely achieved by a large amount of Monte Carlo simulations and it is mentioned by Karssenberg et al., (2001) that this is infeasible for real-world applications without severely increasing the computing power.

Sequential Indicator Simulation (SIS) and other variogram-based simulation techniques allow for appropriate data conditioning but lack of geological realism. Accurate simulation of curvilinear geological features such as extensively continuous channel sands is impossible using merely two-point statistical correlation functions (variogram) (Strebelle & Levy, 2008), (Remy, Boucher, & Wu, 2009), (Kim, Lee, Lee, Rhee, & Shin, 2017) and (Lee, Lim, Choe, & Lee, 2017). Object-based modeling techniques are more appropriate for simulating continuous channels, but lack flexible conditioning capabilities (Hoffimann et al., 2017), especially when “dense” data is available (Strebelle & Levy, 2008). Object-based methods have also shown to cope with difficulties of accurately representing complex channel interactions (Seifert & Jensen, 2000).

Multiple Point Statistics (MPS) is recently accepted as an appropriate alternative to the aforementioned modeling approaches. The main reasons for this constitutes the realistic depositional facies distribution, ease at which MPS honors both hard and soft data as well as well as the low computational costs (Hashemi, Javaherian, Ataee-pour, Tahmasebi, & Khoshdel, 2014), (Kim et al., 2017), (Strebelle, 2002), (Pyrcz, Boisvert, & Deutsch, 2008), (Caers & Zhang, 2004) and (Mariethoz & Caers, 2014). MPS relies on a Training Image (TI) which conceptually represents the geological patterns and spatial variability. The variogram, as a measure of geological heterogeneity, is replaced in MPS by the TI (Caers & Zhang, 2004). One of the current challenges is the use of process-based models as TI’s. This is challenging because of the complexity, non-stationarity and non-repetitiveness of these TI’s, see e.g. (Michael et al., 2010) and (Hoffimann et al., 2017).

This work utilizes the MPS implementation in the software JewelSuite$^{\text{TM}}$, which is based on the IMPALA$^{\text{TM}}$ (Improved Parallel Multiple-point Algorithm Using a List Approach), see (Straubhaar, Renard, Mariethoz, Froidevaux, & Besson, 2011) and (Straubhaar & Malinverni, 2014). For a more general overview of MPS, see (Mariethoz & Caers, 2014).

2.2 Geological models

2.2.1 Training Images

The training images used in this work are displayed in figure 2.1. The left image is the original training image used in Strebelle, (2002) representing a highly channelized fluvial depositional environment.The right image represents a part of the Ganges Delta.

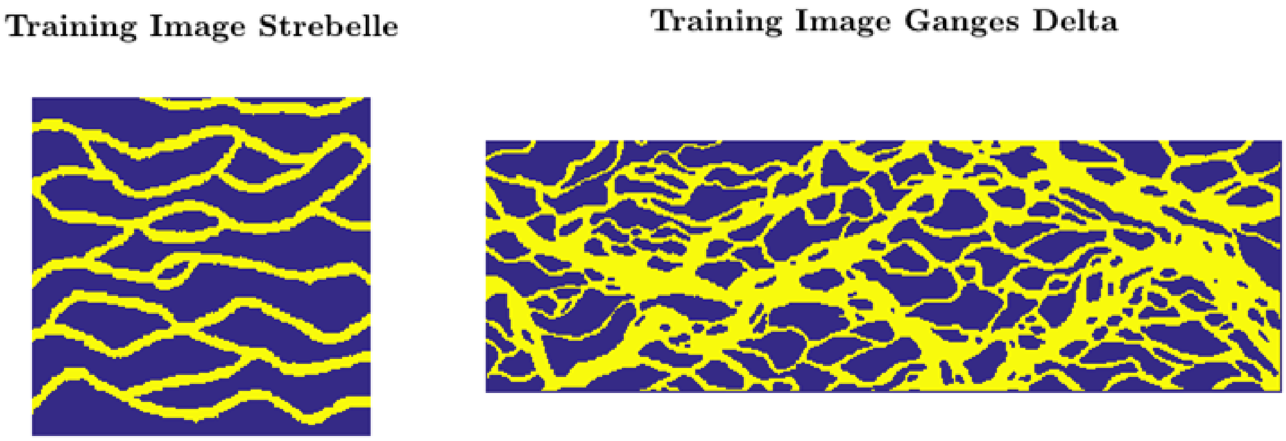


Figure 2.1: **Left** image depicts the training image used in (Strebelle, 2002) where yellow represents the high permeable reservoir facies and purple depicts the low permeable non-reservoir facies. **Right** figure displays the training image taken from (Mariethoz & Caers, 2014) depicting depositional characteristics of a part of the Ganges Delta.

2.2.2 Ensembles generated using MPS

An ensemble of model realizations reflects the uncertainty in flow response. Multiple ensembles are generated using the same training image since any affinity transformation can be used to generate a wide variety of channel-type reservoirs (Caers & Zhang, 2004).

The smallest and simplest[1] ensembles of three models generated from the left TI in figure 2.1 are shown in figure 2.2, where the difference in ensembles is the main paleoflow orientation, ranging from SW-NE to W-E (i.e. from a 45 to 0 degree angle w.r.t. the principal grid orientation).

Certain parameters, such as simulation path and search window, are required for the MPS simulation in JewelSuite$^{\mathrm{TM}}$. In order to achieve proper channel continuity, instead of choosing a random path, a stratified or unilateral simulation path is advised (Mariethoz & Caers, 2014). The size of the search template, as well as the template nesting levels, greatly affect the quality of the simulation as well as the computational time. A balance should be achieved where realizations exhibit geological realism in the sense of continuous channels but still contain a large degree of spatial variability amongst ensemble members (Arpat, 2005), instead of being a clone stamp of the training image. In this thesis work, a visual inspection similar to the one in Arpat (2005) is done on the average sand distribution (see figure 2.3). Note that at the location of the five wells, the average sand distribution is equal to one, since these locations are considered to be constrained by hypothetical “hard well data” of having found sand. A more quantitative way of analyzing the MPS output is explained in (Tan, Tahmasebi, & Caers, 2014).

Another set of ensemble members generated from the left TI in figure 2.1 are shown in figure 2.4. These models are considered to be part of another hypothetical reservoir, containing a line-drive production strategy (of four injectors and four producers), as well as being larger in width and length as the ensemble members shown in figure 2.2.

Lastly, in an attempt to create an even more heterogeneous “difficult” ensemble, the right TI in figure 2.1 (resembling characteristics from a part of the Ganges delta) was used, constrained by a repeated five-spot pattern (total of nine wells, four injector and five producers), the result shown in figure 2.5.

---

[1] Simplest in terms or production strategy, namely a five-spot with one injector in the center, surrounded by four producers at the edges of the reservoir.

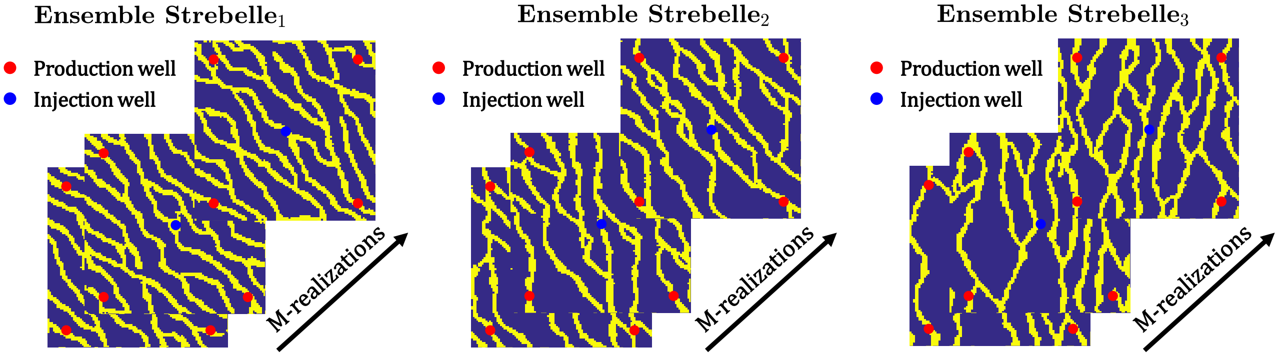


Figure 2.2: Three different ensemble generated using the Strebelle training image. M is the dimension of the ensemble and throughout this work the number of ensemble member is kept at 100 realizations. The five-spot pattern with one injector and four producers is used for constraining the stochastic simulation (sand at every well location). The size of the reservoir is rather small, namely $1000[m] \times 1000[m]$, with a $\Delta x = \Delta y = 10[m]$ leading to $N_x \times N_y = 100 \times 100$ on the Hf-scale.

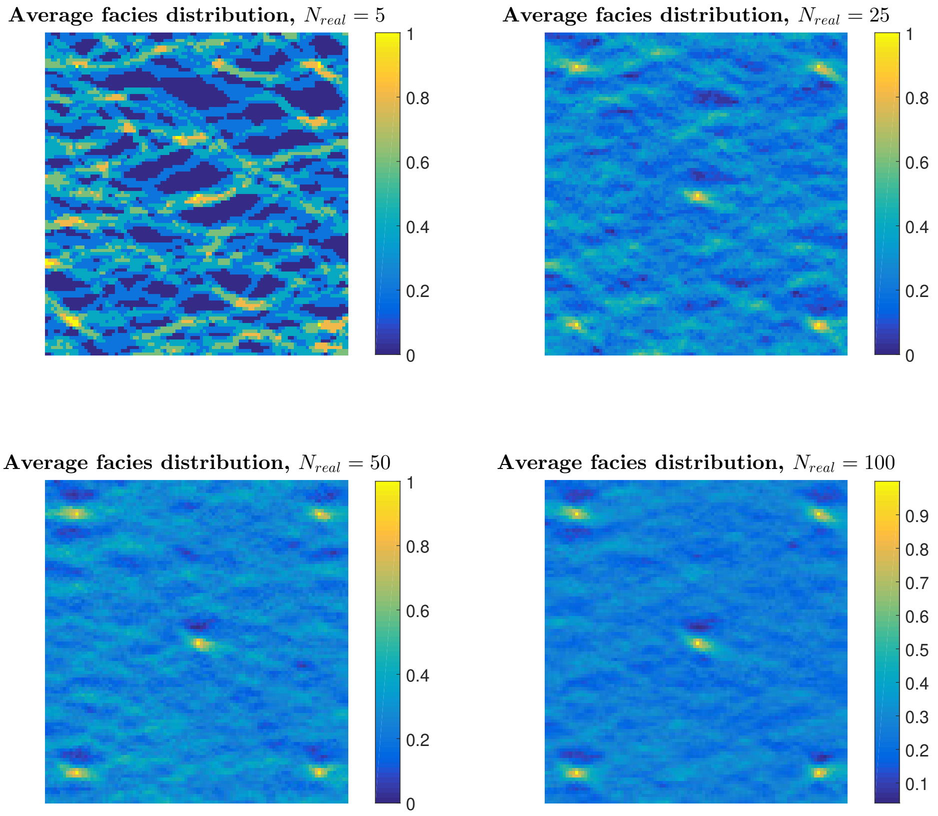


Figure 2.3: Depicting the average sand quantity of the Strebelle$_3$ ensemble, where N_{real} stands for the number of realizations included in the averaging. One means sand occurring at that particular cell for each realization, while zero means no sand at the particular location occurs in any realization included. It serves as a quick visual check for the validity (in terms of spatial uncertainty) of the stochastic simulations. If strong patterns are recognized, the stochastic simulation might be clone-stamping (copying) the training image.

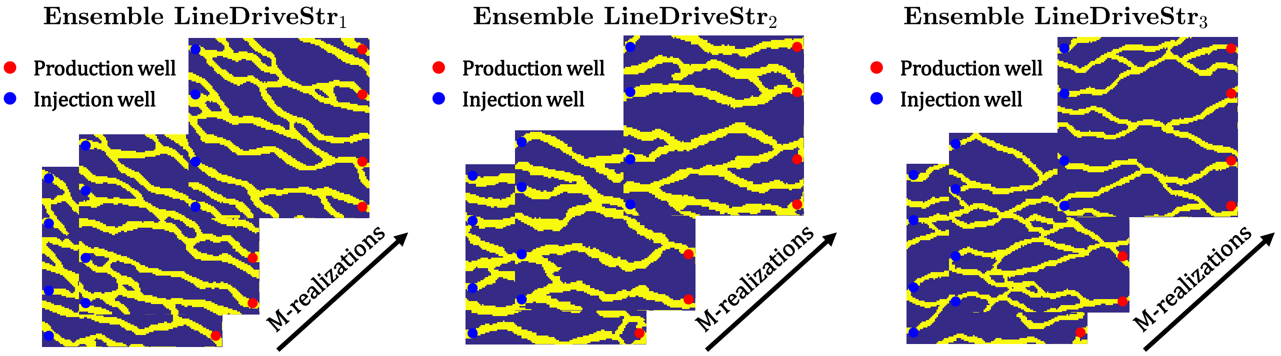

Ensemble LineDriveStr$_1$ **Ensemble LineDriveStr$_2$** **Ensemble LineDriveStr$_3$**

Figure 2.4: Three different ensemble generated using the Strebelle training image, where $M = 100$. The line-drive pattern with four injectors and four producers is used for constraining the stochastic simulation (sand at every well location). The size of the reservoir is rather small, namely $2400[m] \times 2400[m]$, with a $\Delta x = \Delta y = 20[m]$ leading to $N_x \times N_y = 120 \times 120$ on the Hf-scale.

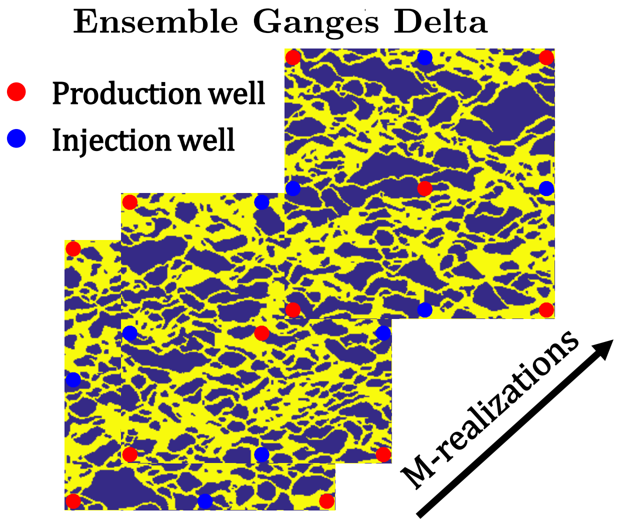

Ensemble Ganges Delta

Figure 2.5: One ensemble generated using the Ganges Delta training image, where $M = 100$. The repeated five-spot pattern with four injectors and five producers is used for constraining the stochastic simulation (sand at every well location). The size of the reservoir is rather small, namely $4800[m] \times 4800[m]$, with a $\Delta x = \Delta y = 20[m]$ leading to $N_x \times N_y = 240 \times 240$ on the Hf-scale.

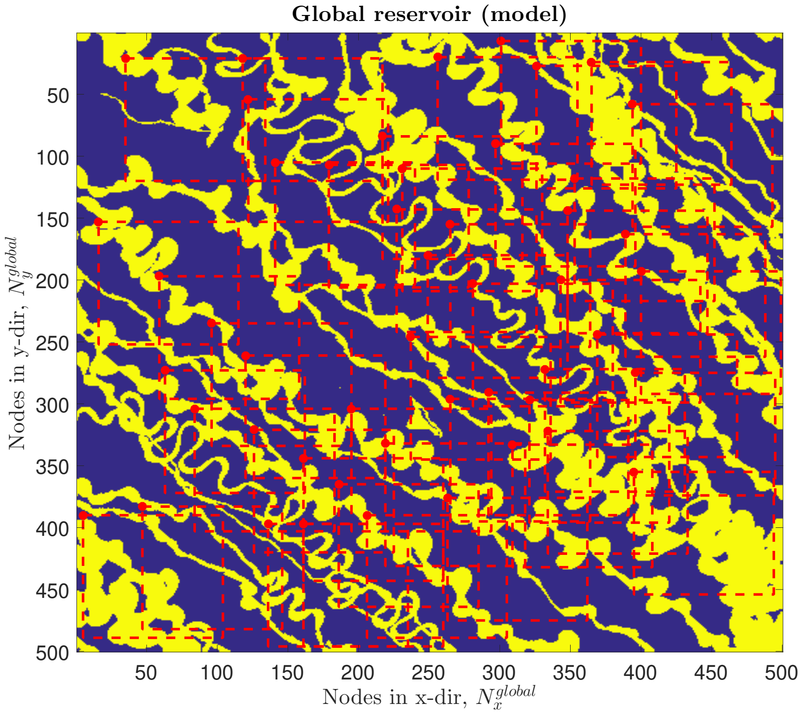


Figure 2.6: Depicting the sampling of smaller reservoir models from the larger FLUMY simulation (for ensemble Flumy$_3$). Note that realizations are at least sampled a fixed amount of distance apart, since spatial uncertainty should be reasonable. The large red dots indicates the origin of the local reservoir models. Note that the local reservoir models only include the constrained models, since much more local reservoir models are sampled, yet rejected due to no sand occurring at all five locations. Also note that the statics, shown in figure 2.7, are slightly biased to larger Net to Gross (NG).

2.2.3 Ensembles generated using FLUMY

Besides stochastic modeling, this work adopts a geometric (semi-process based) approach, using the software FLUMY, see (Grappe, Cojan, Ors, & Rivoirard, 2016) and (Lopez, Cojan, Rivoirard, & Galli, 2009). Currently, there is no actual modeling of sediment transportation in FLUMY, however, time is discretized in iterations (or time-steps) where migration is performed at each step (based on erodibility and other geometric parameters, which also affect the sinuosity of developing meanders). Data condition, by seismic or well data, is partially possible in FLUMY but not always honored in the end member of the simulation. Therefore, another way of extracting realization from FLUMY is chosen in this work. A domain, much larger than the actual reservoir domain, is simulated in FLUMY. From the resulting 3D reservoir model, 2D slices are extracted as shown in figure 2.6. On these resulting 2D models, a template with the same size as the reservoir searches for possible location at which sandy (reservoir) facies occur at the well locations. These locations are then extracted as representative 2D reservoir models. In order to keep the spatial uncertainty and variability between reservoir models, only a limited amount of “smaller” reservoir models can be extracted from the larger 2D model. The statistics of the chosen “smaller” reservoir models show a small bias towards a larger Net to Gross (NG) than the original large 2D model. The resulting ensemble members are depicted in figure 2.8, where again the main variation between the ensemble members is the orientation of the main paleoflow. Further research is necessary to evaluate the validity of the proposed sampling technique. A preliminary thought is that the clear structure exhibited by the ensemble, compared to the averaging of the Strebelle$_3$ ensemble, results in a much smaller spatial uncertainty (shown in figure 2.9).

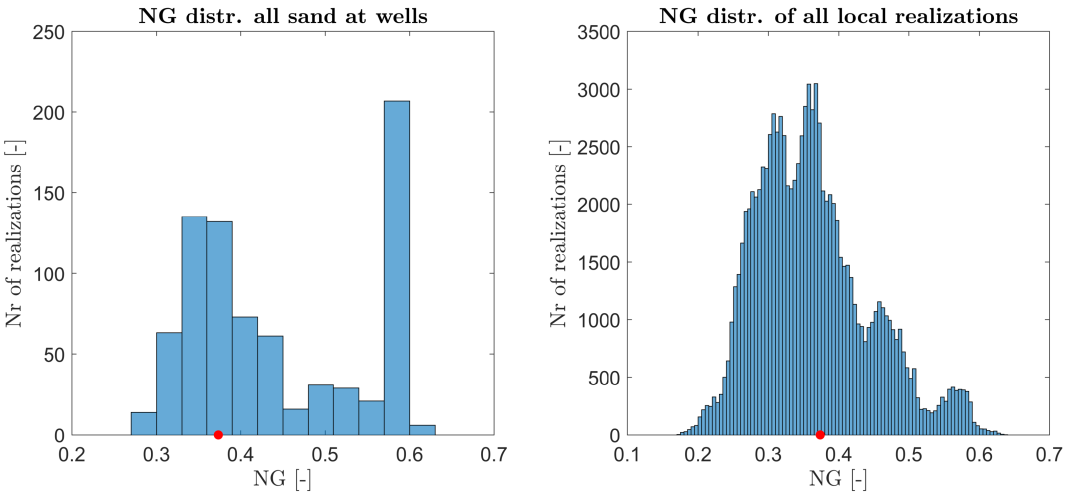


Figure 2.7: Comparison of the statistics obtained from the constrained local realizations, w.r.t all the sampled location realizations. Realizations that pass the posed constraints of having sand at all well, show increase in NG.

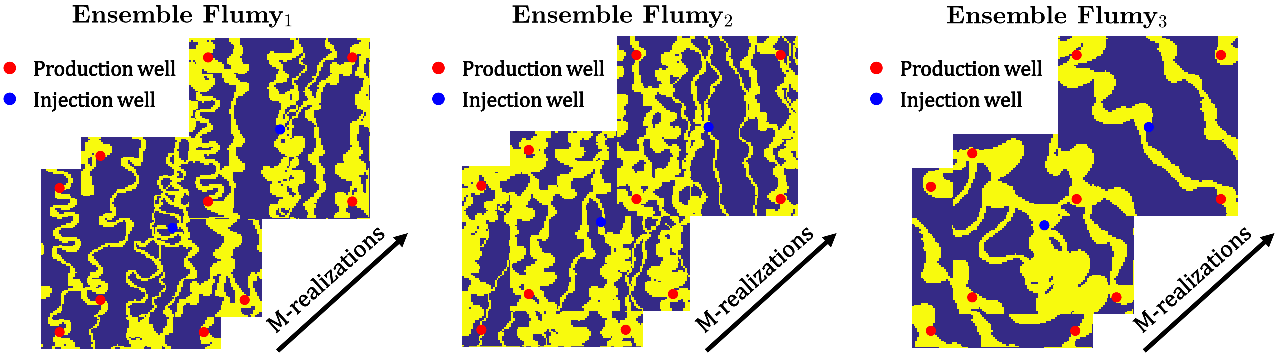


Figure 2.8: Three different ensemble generated using the software FLUMY, where $M = 100$. The five-spot pattern with one injector and four producers is in the search template for possible reservoir models in the large simulation domain, see figure 2.6. The size of the reservoir is rather small, namely $1000[m] \times 1000[m]$, with a $\Delta x = \Delta y = 10[m]$ leading to $N_x \times N_y = 100 \times 100$ on the Hf-scale.

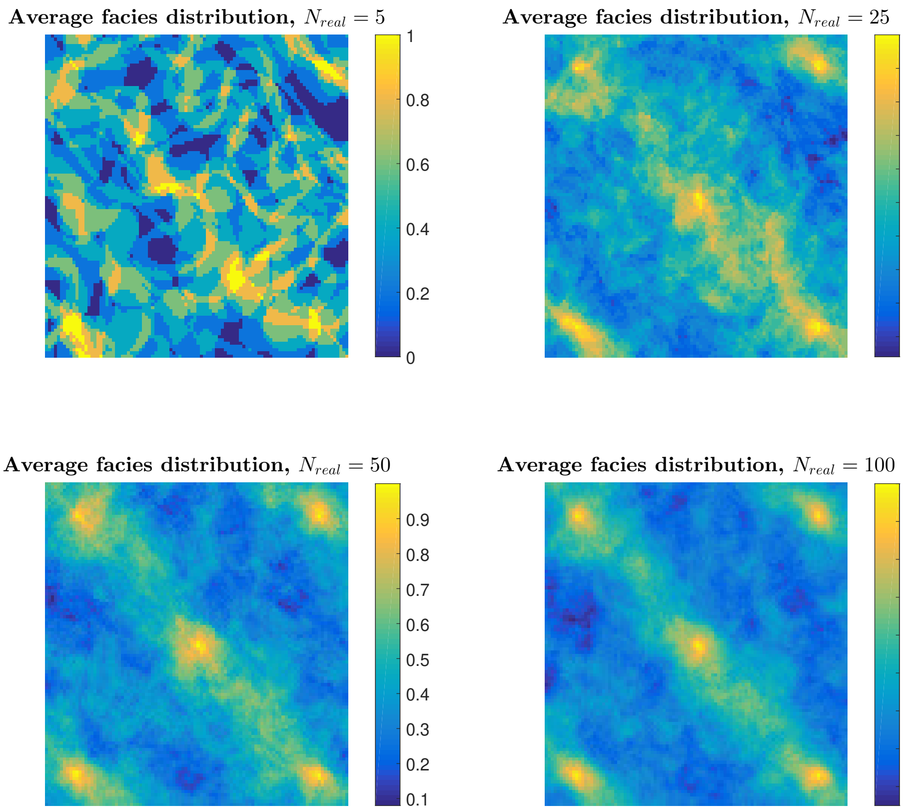


Figure 2.9: Depicting the average sand quantity of the FLUMY3 ensemble, where N_{real} stands for the number of realizations included in the averaging. The predominant paleoflow orientation of the Flumy3 ensemble is visible due to the large averaged sand facies in the NW-SE direction.

2.2.4 Petrophysical properties

For simplicity as well as due to the fact that the primary control on the flow performance of a reservoir is the spatial distribution of depositional facies (Strebelle & Levy, 2008), all ensemble members (containing binary facies distribution) are populated with homogeneous petrophysical properties within each facies. The sandy reservoir facies are assumed to have a permeability of $1000[mD]$ and a porosity of 30% while the non-reservoir facies are considered to have a permeability of $5[mD]$ and a porosity of 10%. This is assumed to create a substantially large contrast in permeability, as often seen in a fluvial depositional setting. Chapter 4.2 mimics the effect of sub-scale heterogeneity by populating the permeability and porosity using a bimodal distribution sampled from a smoothly varying Gaussian Random Field.

3

Theoretical background

This chapter describes all the necessary theory in order to understand the analysis done in chapter 4 and 5. The equations which govern fluid flow in porous media are briefly mentioned, followed by the upscaling algorithm used to obtain the coarser representation of the reservoir parameters. The Discrete Cosine Transform is used in this thesis work to analyze the static spatial scale and thoroughly explained in section 3.3. Finally, the distance-based modeling, used in the analysis of the response uncertainty is explained in the last part of this chapter.

3.1 Governing equations for Flow and Transport in Porous Media

In this work, we are using Automatic Differentiation General Purpose Research Simulator (ADGPRS) developed at Stanford University. In this chapter we only highlight the important equations and formulations used in ADGPRS (D. Voskov et al., 2012) and (Stanford, 2012). The following list of references should cover the fundamentals of numerical reservoir simulation, from basic formulations up to complex physics such as thermal and compositional simulation: (Aziz & Settari, 1979), (Bear, 2013), (Peaceman, 1977), (Helmig et al., 1997), (Mattax, Dalton, et al., 1990), (Z. Chen, Huan, & Ma, 2006) and (Jansen, 2013).

Conservation laws typically expressed through Partial Differential Equations (PDE), form the basis of any model that simulates flow through porous media. The conservation of mass for two-phase flow in porous media, with water and oil as the two immiscible phases, coupled with Darcy's law for the conservation of momentum (Aziz & Settari, 1979), is typically given in the continuous form as

$$\frac{\partial(\rho_i \phi S_i)}{\partial t} + \nabla \cdot (\rho_i \boldsymbol{v}_i) - \rho_i q_i = 0, \qquad i \in \{o, w\} \tag{3.1}$$

where ρ_i is the density, S_i is the saturation and q_i is the source term of the i-th phase respectively, ϕ is the porosity of the porous media, ∇ is the Nabla operator and $\boldsymbol{v}_i$ is the Darcy velocity of the i-th phase, given as

$$\boldsymbol{v}_i = -\frac{k_{r,i}}{\mu_i} \boldsymbol{K} \nabla (P_i - \rho_i \boldsymbol{g}), \qquad i \in \{o, w\} \tag{3.2}$$

where $k_{r,i}$ is the relative permeability, μ_i is the viscosity and P_i is the pressure of the i-th phase respectively, $\boldsymbol{K}$ is the permeability tensor and $\boldsymbol{g}$ is the directional gravitational acceleration defined as $g\nabla z$. The constraint equation typically used to close the above system of governing equations is

$$S_w + S_o = 1 \tag{3.3}$$

Equation (3.1) does accurately describe the process of water injection into a dead oil reservoir. However, for the general purpose, it is easier to write the conservation equation for each component separately and then reduce it for particular physics. This approach was adopted in the design of ADGPRS, see

(D. V. Voskov & Tchelepi, 2012; D. Voskov, 2012) for details. The conservation of mass in general form is written as:

$$F_c = \frac{\partial}{\partial t}\left(\phi \sum_p x_{cp}\rho_p S_p\right) - \nabla \cdot \sum_p x_{cp}\rho_p \boldsymbol{v}_p - \sum_p x_{cp}\rho_p q_p = 0, \quad c = 1, \ldots, n_c \tag{3.4}$$

where the subscript x_{cp} is the mass fraction of component c in phase p.

Since an analytical solution to the above systems of equations is not always available, the systems are discretized in space using a standard Finite Volume Method (FVM), e.g. Two-Point Flux Approximation where the flux depends on only two neighbors. Discretization in time can be done using a Fully Implicit Method (FIM), using a sequential Implicit Pressure Explicit Saturation (IMPES) scheme, Implicit Pressure and Saturation (IMSAT) or Adaptive Implicit Method (AIM). AIM is considered the most general formulation, i.e. other schemes are special cases of AIM (Cao, 2002). Switching between the formulations in AIM scheme is dependent on the CFL condition, which in return depends on the phases present in the system, see (D. Voskov et al., 2012) for more detail.

3.2 Flow based upscaling

This section is dedicated to the technique used in order to generate the coarser ensemble members. First, different available upscaling techniques are highlighted and the technique, chosen in this thesis, is further explained.

Generally, there is a differentiation between flow based upscaling algorithms and traditional averaging technique (also called analytical averaging methods). Traditional averaging techniques comprise arithmetic, geometric, harmonic and power averaging. These types of averaging are computationally efficient and straightforward for implementation. However, they usually don't incorporate any information of the flow itself and are based on idealized assumptions. From the other side, flow-based averaging techniques incorporate the information of the flow, either in a local, extended local, quasi-global (local-global) or global manner. The aim of these techniques is to obtain a coarser representation of the parameters which should approximate the fine-scale behavior of the system while reducing greatly the computational time.

Local

The main idea of local upscaling algorithms is that the properties of the coarse cell are solely determined by solving a local flow problem, where the domain of the local flow problem exactly comprises the target coarse cell (Y. Chen, Durlofsky, Gerritsen, & Wen, 2003). Since the steady-state single-phase pressure equation, subject to particular boundary conditions, is solved to obtain the coarse properties, the solution is heavily dependent on boundary conditions. Besides that, the global pressure field is strongly influenced by the global permeability field (He & Durlofsky, 2006). This means that coarse-scale properties obtained from local flow problems don't always accurately capture the global flow patterns from the underlying fine-scale properties. A positive aspect of the local upscaling though is that the local flow problems can be solved independently from each other, leading to easy parallelization (Y. Chen et al., 2003) and (He & Durlofsky, 2006). For more extensive overview of local upscaling techniques see (Kitanidis, 1990), (Durlofsky, 1991) and (Durlofsky, 2005).

Extended local

Extended local is a natural expansion of the local formulation, i.e. it extends the domain used for the local flow problem to include surrounding fine-scale information. The size of the surrounding region, typically denoted by r due to its similarity with a radius[1]. The extended local formulation is therefore different in

---

[1] Namely, a radius of fine-scale information surrounding the coarse cell of interest.

the boundary conditions used for the local flow problem and therefore different in its approximation of the coarse-scale property, generally an improvement over the purely local upscaling methods. For more extensive overview of the extended local formulation see e.g. (Holden & Lia, 1992), (Gomez-Hernandez, Journel, et al., 1994), (Wu, Efendiev, & Hou, 2002), (Wen, Durlofsky, & Edwards, 2003a) or (Wen, Durlofsky, & Edwards, 2003b).

Global

Global upscaling is the algorithm used in this work to obtain the coarse-scale ensemble members. Global upscaling involves solving the fine-scale steady state single-phase pressure equation on the global domain defined as

$$\nabla \cdot \boldsymbol{v} = \nabla \cdot \left(\frac{\boldsymbol{K}}{\mu} \nabla (P - \rho \boldsymbol{g}) \right) = q_{\text{well}}, \tag{3.5}$$

or sometimes also expressed in dimensionless form as

$$\nabla \cdot \boldsymbol{v} = \nabla \cdot (\boldsymbol{K} \nabla P) = q_{\text{well}}. \tag{3.6}$$

The solution to equation (3.5) for pressure, P, is then used to obtain the coarse property (permeability or transmissibility). As shown by Durlofsky et al., (2006) and Chen et al., (2003), directly upscaled transmissibility will result in a better representation of the fine-scale pressure field. This is accentuated for highly discontinuous fine-scale permeability fields, which can be found in channelized reservoirs. The reason for this is that in the step of computing the coarse transmissibility by harmonic averaging of the upscaled permeability, more weight is put on lower permeability values, therefore, underestimating the total flow and increasing the approximation error (Y. Chen et al., 2003). Avoiding the harmonic averaging step by directly upscaling the transmissibility is therefore advised in order to better capture strongly discontinuous permeability fields.

As mentioned before, the pressure solution to equation (3.5) is used to obtain the interface transmissibilities of the coarse cells. This is done by considering the following principle: for a Two-Point Flux Approximation, interface transmissibility multiplied with a pressure difference across the interface separating two coarse cells should give the flux across the particular interface. This can be written for the interface $(i+1/2, j)$ separating the two coarse cells (i, j) and $(i+1, j)$, as

$$(q_x^c)_{i+1/2,j} = (T_x^c)_{i+1/2,j} \left(P_{i,j}^c - P_{i+1,j}^c \right) \tag{3.7}$$

where $(q_x^c)_{i+1/2,j}$ is the coarse flux across the interface $(i+1/2, j)$ simply defined as the integrated fine-scale fluxes across the coarse interface, $(T_x^c)_{i+1/2,j}$ is the coarse transmissibility and ($P_{i,j}^c$ and $P_{i+1,j}^c$) are the coarse pressures obtained by arithmetically averaging the fine-scale pressures contained in each block respectively. Rewriting this equation for coarse transmissibility results in

$$(T_x^c)_{i+1/2,j} = \frac{(q_x^c)_{i+1/2,j}}{P_{i,j}^c - P_{i+1,j}^c} \tag{3.8}$$

Naturally this can be extended to transmissibility in the y-direction as

$$(T_y^c)_{i,j+1/2} = \frac{(q_y^c)_{i,j+1/2}}{P_{i,j}^c - P_{i,j+1}^c} \tag{3.9}$$

This way of obtaining coarse properties from global flow was first shown by Horne (1987) followed by Nielsen et al.,(2000) who formulated it in terms of an optimization problem (White & Horne, 1987). Global upscaling in highly heterogeneous reservoirs has one downfall, however, namely that the resulting transmissibility values might be very large (or even negative!), see (Holden & Nielsen, 2000) and (Y. Chen et al., 2003) for more analysis on the matter. An iterative procedure is generally used to obtain a positive-definite transmissibility matrix, very much desired from a numerical analysis point of view. The iterative

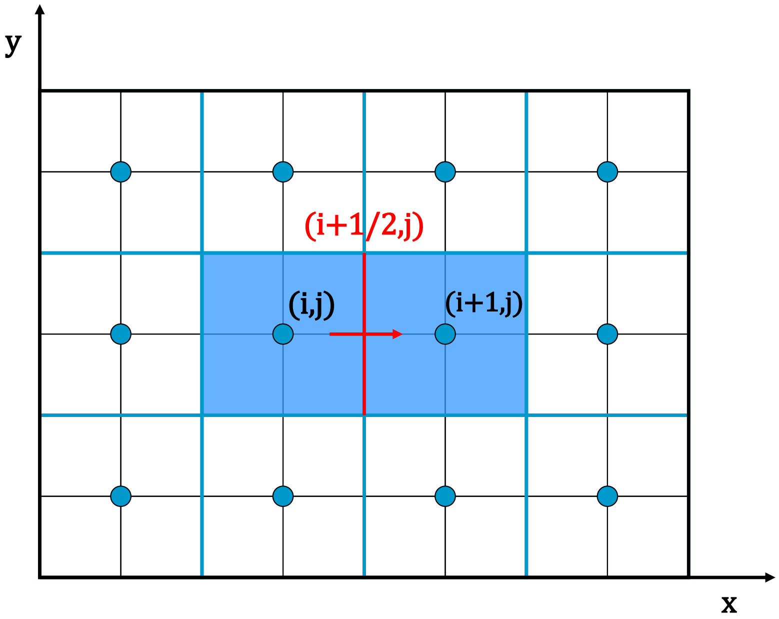


Figure 3.1: Schematic image showing the fine-grid of $N_x^f \times N_y^f = 8 \times 6$, using an upscaling factor of two in both the x- and y-direction results in the coarse grid having size $N_x^c \times N_y^c = 4 \times 3$, where the coarse pressure is indicated with the blue circle obtained by arithmetically averaging the fine scale pressure contained in the particular coarse cell. Cells in the x-direction are counted using the i-th index, while cells in the y-direction are counted using the j-th index. The formula for estimating the coarse interface transmissibility, separating the coarse blocks (i, j) and $(i+1, j)$, denoted $(i+1/2, j)$, is by equation (3.8), where the red arrow is used to indicate the integrated coarse flux.

procedure replaces negative values with geometrically averaged transmissibility values, after which the coarse pressure is recomputed using the new coarse transmissibilities. Next it followed by estimating the new coarse transmissibility from equation (3.8) where $(q_x^c)_{i+1/2,j}$ is unchanged. Generally, the convergence of zero non-negative transmissibilities values is reached within five iterations. When convergence is not reached within ten iterations, the remaining negative transmissibility values are finally set to a geometrically averaged transmissibility. The rate of convergence is heavily dependent on the degree of heterogeneity, in particular, the contrast in permeability. When the convergence is not reached, horrendously large transmissibility values might occur after several iterations (5-10). This doesn't negatively affect the pressure solution in the upscaling procedure, however, might cause numerical complications when these transmissibility values are used to perform a simulation with different conditions.

Besides flow-based upscaled transmissibility, a similar formula for a flow-based upscaled well-index for well α can be derived, given by the following equation

$$\mathrm{WI}^c_{\text{well }\alpha} = \frac{q_{\text{well }\alpha}}{P^c_{i,j} - P_{\text{well }\alpha}}, \tag{3.10}$$

where the coarse superscript is left out for the coarse-scale well flux and pressure since for strictly vertical wells in 2D model, they are generally equal to the fine-scale well flux and pressure[2]. Note that in three-dimensions, there exists a coarse-scale well flux and pressure when upscaling in the z-direction causing the fine- and coarse-scale to deviate, even for strictly vertical wells. The resulting coarse-scale well flux is just the summation of the fine-scale well fluxes (for each fine-scale grid block containing a well inside the new coarse block) and the coarse-scale pressure is the average well pressure (averaged over each fine-grid

---

[2] Unless the well is horizontal or the upscaling ratio is quite extreme, up to the case where multiple wells end up in the same coarse grid block.

containing a well with prescribed target BHP inside the new coarse grid-block). For upscaling, we used the modified MATLAB©script developed by Dr. Brad Mallison at Stanford University.

3.3 Discrete Cosine Transform

One of the most common applications of the Discrete Cosine Transform (DCT) is found in pattern recognition and lossy (image and audio) compression. This popularity is due to its strong energy compaction property and it is closely related to the optimal[3] Karhunen-Loeve Transform (KLT) (Ahmed, Natarajan, & Rao, 1974). DCT used to be popular as an approximation to the KLT[4] due to its simplicity in computation (using Fast-Fourier Transform). Recently, there is increased interest in the use of DCT in areas like Compressed History Matching, where sparsity in the transform domain is exploited by using a truncated transform to approximate spatial parameters in an inverse problem (Jafarpour, Goyal, McLaughlin, & Freeman, 2010). Another useful property of the DCT is that no prior information is required in the construction of the DCT basis (Jafarpour et al., 2010), whereas the aforementioned KLT is considered to be in the class of data-dependent bases. More information on the mathematical properties and theory of the DCT are given in (Rao & Yip, 2014) and (Jain, 1989). Additionally, Strang, (1999) provides a very insightful argument why the DCT is not merely a discrete approximation of the Fourier Cosine Transform. Specifically, it is shown that the basis functions for the DCT arise very naturally as eigenvectors of a second difference matrix, corresponding to the discretization of a homogeneous undamped harmonics problem on the closed interval $[0,\pi]$ (Strang, 1999). Depending on the choice of discretized Neumann boundary conditions, the eigenvectors of the discretization matrix represent the basis-vectors for a type of DCT. In this work, DCT-2 is used for which the formal mathematical definition is given below.

DCT-2 is a linear invertible function, that maps from $\mathbb{R}^N$ to $\mathbb{R}^N$, or $g:\mathbb{R}^N\to\mathbb{R}^N$. Note that the -2 refers to the type of DCT and not the dimension of the input signal. The one-dimensional DCT-2 on a discrete signal, $\boldsymbol{y}=[y_0 \quad y_1 \quad \dots \quad y_{N-1}]^T$, where N is the number of elements in the signal, can be expressed as:

$$g_k=\alpha_k\sum_{n=0}^{N-1}y_n\cos\left[\frac{\pi}{N}\left(n+\frac{1}{2}\right)k\right] \tag{3.11}$$

where g_k is the k-th frequency component of the transformation and α_k is a scaling factor, defined as

$$\alpha_k=\begin{cases}\sqrt{\frac{1}{N}}, & :k=0\\ \sqrt{\frac{2}{N}}, & :k=1,\dots,N-1\end{cases} \tag{3.12}$$

This scaling factor ensures that the DCT-2 matrix, associated with this linear transformation, is orthogonal (orthonormal column vectors) meaning

$$\boldsymbol{\Phi}^T\boldsymbol{\Phi}=\boldsymbol{\Phi}\boldsymbol{\Phi}^T=\boldsymbol{I} \tag{3.13}$$

where $\boldsymbol{\Phi}$ is the $N\times N$ square DCT-2 matrix, implicitly defined by equation (3.11), $\boldsymbol{I}$ is the identity matrix and the superscript T is used to indicate the matrix transpose. Note that it's not necessary to write the Hermitian transpose (conjugate matrix transpose) since only real basis-vectors and input signals are concerned in this work. Key observation from equation (3.13) is: $\boldsymbol{\Phi}^T=\boldsymbol{\Phi}^{-1}$. The above mentioned linear transformation can be defined in matrix-vector form as

$$\boldsymbol{g}=\boldsymbol{\Phi}\boldsymbol{y} \tag{3.14}$$

which can be interpreted as a simple change of basis, namely changing the original basis of the vector (standard basis on $\mathbb{R}^N$) to a basis of mutually orthogonal cosine function oscillating at different frequencies.

---

[3] Optimal refers here to its variance distribution property and rate-distortion function.

[4] For wide sense stationary processes.

Note that each row of $\boldsymbol{\Phi}$ corresponds to a basis element of the DCT basis since left multiplying equation (3.14) with $\boldsymbol{\Phi}^T$ gives

$$\boldsymbol{\Phi}^T \boldsymbol{g} = \boldsymbol{I}\boldsymbol{y} \tag{3.15}$$

where it is evident that the columns of $\boldsymbol{\Phi}^T$ are the basis-vectors of the DCT domain, such that $\boldsymbol{\Phi} = [\boldsymbol{\varphi}_0 \quad \boldsymbol{\varphi}_1 \quad \dots \quad \boldsymbol{\varphi}_{N-1}]^T$ where $\boldsymbol{\varphi}_i$ is the i-th DCT basis-vector, and the columns of $\boldsymbol{I}$ are the standard basis-vectors of $\mathbb{R}^N$ respectively. Extension of the DCT basis for signals in multiple dimensions is simply by tensor product of the one-dimensional basis vectors. For a two-dimensional data matrix $\boldsymbol{Y}$, this results in performing a one-dimensional DCT on the rows of $\boldsymbol{Y}$ followed by another one-dimensional DCT on the columns, or vice versa. Explicitly, this two dimensional DCT can be expressed as

$$g_{k_1,k_2} = \alpha_{k_1,k_2} \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} y_{n_1,n_2} \cos\left[\frac{\pi}{N_1}\left(n_1 + \frac{1}{2}\right)k_1\right] \cos\left[\frac{\pi}{N_2}\left(n_2 + \frac{1}{2}\right)k_2\right] \tag{3.16}$$

where the subscripts n_i and k_i indicate the i-th spatial and DCT component respectively, and α_{k_1,k_2} the scaling factor which ensures orthogonality. The two-dimensional linear transformation can be written in matrix-vector form as

$$\boldsymbol{G} = \left(\boldsymbol{\Phi}_2(\boldsymbol{\Phi}_1\boldsymbol{Y})^T\right)^T = \boldsymbol{\Phi}_1\boldsymbol{Y}\boldsymbol{\Phi}_2^T \tag{3.17}$$

where $\boldsymbol{\Phi}_i$ is the i-th one-dimensional square DCT-2 matrix of size $N_i \times N_i$, $i \in \{1,2\}$ and $\boldsymbol{Y}$ is the $N_1 \times N_2$ data matrix.

The strong energy compaction property[5] of the DCT is utilized in this work to identify dominant basis-vectors. The dominant basis-vector is referred to as characteristic scale/frequency of the input signal, since a DCT-2 basic-vector is fully described by its amplitude and frequency, therefore representing a characteristic scale of the input signal.

This is illustrated in 3.2 with a trivial one-dimensional example, constituting a cosine wave, $A\cos(2\pi fx + \phi) = 2\cos(4\pi x)$ being mapped to the DCT domain. It is evident that the transformed signal can be approximated by, or in this case entirely expressed as one coefficient multiplying its associated basis-vector. Note that the dimensionless frequency number k is not equal to the spatial frequency f. The reason for this is that the transformation is scale independent in its dimensionless form, $n,k \in \{0,1,\dots,N-1\}$, assuming a uniformly spaced signal with increment 1. Consequently this means that the frequency number k itself doesn't reveal information on actual spatial frequency f in [cycles/m]. Since in this work, the DCT is computed for an ensemble of models across various model scales (different number of nodes), it is important to understand the difference. For example, two similar signals with different number of points/observations (coarse and fine-scale representation) shouldn't have different dominant/characteristic $k_{\max}$, something which is not immediately evident from equation (3.11). Considering the following equation

$$y(x) = A\cos(2\pi fx + \phi) \tag{3.18}$$

which is the most general form of a cosine wave, where A is the amplitude of the wave, f the spatial frequency in [cycles/m], x the spatial position in [m] and ϕ is the phase in [radians]. Now, expressing the k-th basis-vector in the following way

$$\varphi_k(n) = \cos\left[\frac{\pi}{N\Delta x}\left(n\Delta x + \frac{\Delta x}{2}\right)k\right] \tag{3.19}$$

where Δx is the constant increment between two points/observations of the discrete signal $\mathbf{y}$. Note that this is the exact same expression for the k-th basis-vector as in equation (3.11). Substituting $L = N\Delta x$ and $x = n\Delta x$, and solving the following equation for f:

$$2\pi fx = \frac{\pi kx}{L}$$

---

[5] Magnitude of the DCT coefficients is strongly correlated with their resembling of the original signal. It is assumed that largest (absolute value) DCT coefficient describes the dominant behavior of the process/signal.

obviously yields $f = \frac{k}{2L}$. This means that two cosine waves, similarly defined as in Figure 3.2, with e.g. $N_{\text{coarse}} = 100$ and $N_{\text{fine}} = 1000$, would yield the same dominant frequency number, namely $k = 8$[-] and yield the same spatial frequency $f = 2$[cycles/m], as one would expect[6].

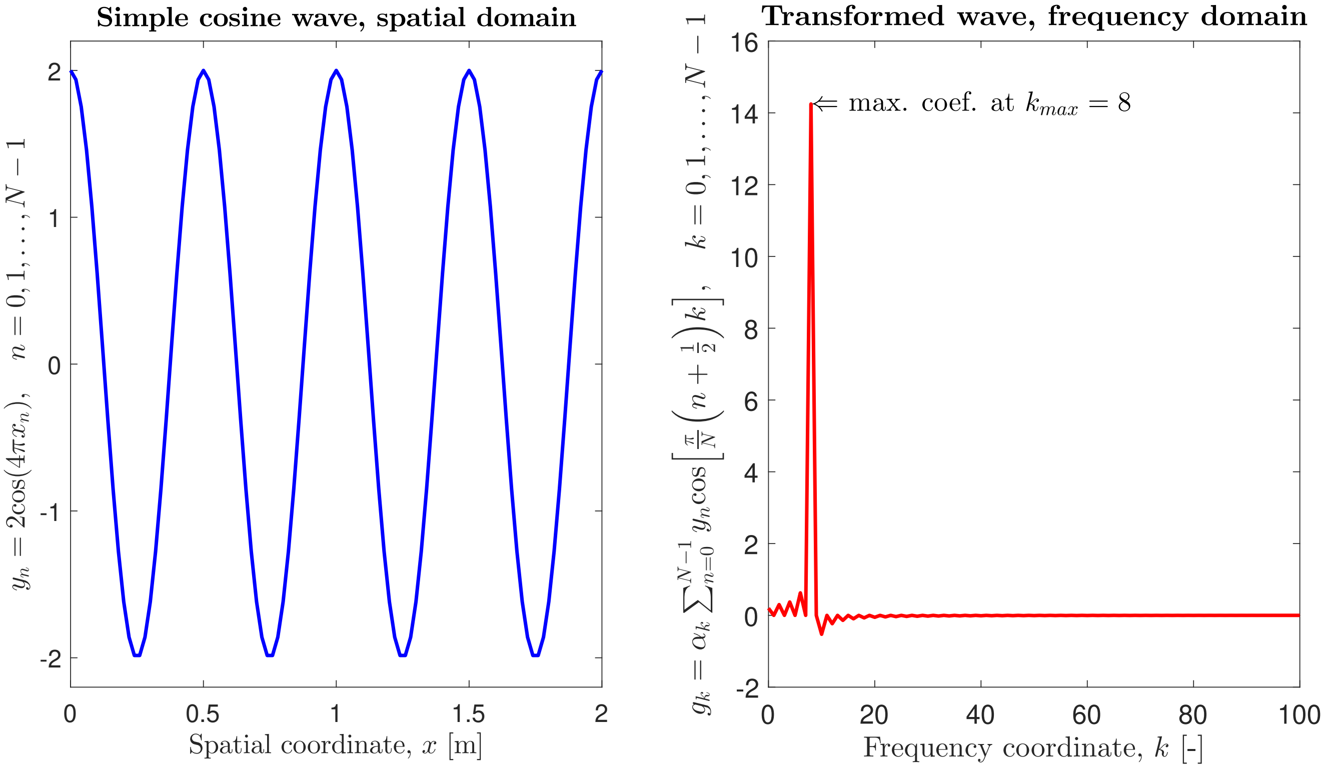


Figure 3.2: Visualization of the one-dimensional Discrete Cosine Transform (DCT) on a simple cosine wave. **The left** figure shows the cosine wave in the spatial domain whereas **the right** figure displays the transformed wave in the DCT (frequency) domain. This trivial example illustrates the strong energy compaction property of the DCT, particularly that the signal is almost entirely described by $k = 8$ while the remaining coefficients are almost zero. Note that frequency coordinate k is not equal to f, the spatial frequency in [cycles/m], see explanation above.

Important to note is that the DCT is a unitary transformation, see equations (3.11) and (3.13), such that the energy[7] of the input signal is equal to the that of the output signal (Jain, 1989), since

$$||\boldsymbol{g}||_2^2 = \sum_{k=0}^{N} |g_k|^2 = \boldsymbol{g}^T \boldsymbol{g} = (\boldsymbol{\Phi y})^T \boldsymbol{\Phi y} = \boldsymbol{y}^T \boldsymbol{\Phi}^T \boldsymbol{\Phi y} = \boldsymbol{y}^T \boldsymbol{y} = ||\boldsymbol{y}||_2^2 \tag{3.20}$$

where $||\cdot||_2^2$ is the squared l2-norm. This property is important considering a truncated DCT, where truncated means setting the lower DCT coefficients to zero. Namely that

$$||\boldsymbol{g} - \tilde{\boldsymbol{g}}||_2^2 = ||\boldsymbol{y} - \tilde{\boldsymbol{y}}||_2^2 \tag{3.21}$$

where $\tilde{\boldsymbol{g}}$ and $\tilde{\boldsymbol{y}}$ are the truncated DCT and input signals, respectively. This can be proven similarly by

$$\begin{aligned} ||\boldsymbol{g} - \tilde{\boldsymbol{g}}||_2^2 = (\boldsymbol{g} - \tilde{\boldsymbol{g}})^T (\boldsymbol{g} - \tilde{\boldsymbol{g}}) = (\boldsymbol{\Phi y} - \boldsymbol{\Phi}\tilde{\boldsymbol{y}})^T (\boldsymbol{\Phi y} - \boldsymbol{\Phi}\tilde{\boldsymbol{y}}) &= \Big(\boldsymbol{\Phi}(\boldsymbol{y} - \tilde{\boldsymbol{y}})\Big)^T \Big(\boldsymbol{\Phi}(\boldsymbol{y} - \tilde{\boldsymbol{y}})\Big) \\ &= (\boldsymbol{y} - \tilde{\boldsymbol{y}})^T \boldsymbol{\Phi}^T \boldsymbol{\Phi} (\boldsymbol{y} - \tilde{\boldsymbol{y}}) \\ &= (\boldsymbol{y} - \tilde{\boldsymbol{y}})^T (\boldsymbol{y} - \tilde{\boldsymbol{y}}) = ||\boldsymbol{y} - \tilde{\boldsymbol{y}}||_2^2 \end{aligned} \tag{3.22}$$

---

[6] The finer signal in the DCT domain simply contains much more higher-frequency basis-vectors. However, the spatial frequency of the first N_{coarse} basis-vectors are identical since the two signals are defined over the same distance.

[7] Here defined as the square of the l2-norm, inner product or the squared length of a vector.

Since generally the energy in the DCT domain is highly compacted (few large coefficients and a lot of very small ones), a truncated DCT approximation with only a few non-zero coefficients may result in a small error (in the squared l2-norm) in the spatial domain as well. An example of this is shown in Figure 3.3.

Besides showing the operations required for obtaining a truncated signal $\tilde{\boldsymbol{y}}$, it is also an appropriate way to mention another important fact about the linear transformation and the DCT in particular. If the signal $\hat{\boldsymbol{y}}$ is random, then the DCT-coefficients, $\hat{\boldsymbol{g}}$, are also random[8] and therefore spatial characteristics of a random signal (or ensemble of random signals, as are generated in this work) should be measured with respect to its expected value. Formally, the truncated input signal by inversion of truncated DCT-coefficients is obtained by introducing the diagonal restriction matrix $\boldsymbol{R}$, where $r_{i,j}$ is the (i,j)-th entry of the restriction matrix $\boldsymbol{R}$ defined as

$$r_{i,j} = \begin{cases} 1, & : i = j \wedge i \in \boldsymbol{S} \\ 0, & : \text{otherwise} \end{cases} \tag{3.23}$$

where $\boldsymbol{S}$ is a set of integers containing the non-zero entries after truncation. Usually $\boldsymbol{S}$ is defined by a threshold value for the required energy of the signal after truncation. For example, considering the one-dimensional example above: if the truncation energy is 100% then all basis-vectors should be included in the "truncation", which means that $\boldsymbol{S} = \{0, 1, \ldots, N-1\}$ and therefore $\boldsymbol{R} = \boldsymbol{I}$. This means that the truncated DCT, $\tilde{\boldsymbol{g}}$, can be expressed as

$$\tilde{\boldsymbol{g}} = \boldsymbol{R}_M \boldsymbol{g} = \boldsymbol{R}_M \boldsymbol{\Phi} \boldsymbol{y} \tag{3.24}$$

where the subscript M indicates the number of non-zero diagonal entries, which consequently means that the truncated signal, $\tilde{\boldsymbol{y}}$, can be expressed as

$$\tilde{\boldsymbol{y}} = \boldsymbol{\Phi}^T \tilde{\boldsymbol{g}} = \boldsymbol{\Phi}^T \boldsymbol{R}_M \boldsymbol{\Phi} \boldsymbol{y} = \boldsymbol{U}_M \boldsymbol{y} \tag{3.25}$$

where $\boldsymbol{U}_M = \boldsymbol{\Phi}^T \boldsymbol{R}_M \boldsymbol{\Phi}$ is the truncation matrix of rank M[9].

Figure 3.4 shows the truncation matrix $\boldsymbol{U}_M$ for the example in Figure 3.3, which clearly shows the rank 1 approximation (only using one basis-vector) since $\boldsymbol{U}_M$ is simply constructed by the tensor product of the dominant basis-vector. Alternative formulation of the truncation matrix $\boldsymbol{U}_M$ is simply as the sum of rank 1 matrices, defined by the tensor products of the basis-vectors which are included in the truncation

$$\boldsymbol{U}_M = \sum_{m=0}^{N-1} \boldsymbol{\varphi}_m \otimes \boldsymbol{\varphi}_m \delta_m = \sum_{m=0}^{N-1} \boldsymbol{\varphi}_m (\boldsymbol{\varphi}_m)^T \delta_m \tag{3.26}$$

where the subscript M stands for the rank of the truncation matrix $\boldsymbol{U}_M$ and δ_m is the Dirac-delta function with a similar definition as equation (3.23), namely

$$\delta_m = \begin{cases} 1, & : m \in \boldsymbol{S} \\ 0, & : \text{otherwise} \end{cases} \tag{3.27}$$

Note that throughout the chapter all subscripts range from 0 to $N-1$, this is the usual convention when using the DCT since that way the subscripts of the basis-vectors refer to its frequency number.

Another interesting thing to realize is that the Gaussian noise added to the cosine wave in Figure 3.3 doesn't "disappear" but since the DCT is a linear transformation, equation (3.25) for this cosine wave with added noise can be expressed as

$$\tilde{\boldsymbol{y}} = \boldsymbol{U}_1 \hat{\boldsymbol{y}} = \boldsymbol{U}_1 (\boldsymbol{y} + \boldsymbol{\varepsilon}) = \boldsymbol{U}_1 \boldsymbol{y} + \boldsymbol{U}_1 \boldsymbol{\varepsilon} \tag{3.28}$$

---

[8] Since a linear transformation of a Gaussian random vector is also a Gaussian random vector.

[9] M is the number of basic-vectors included in the truncation, generally $M << N$

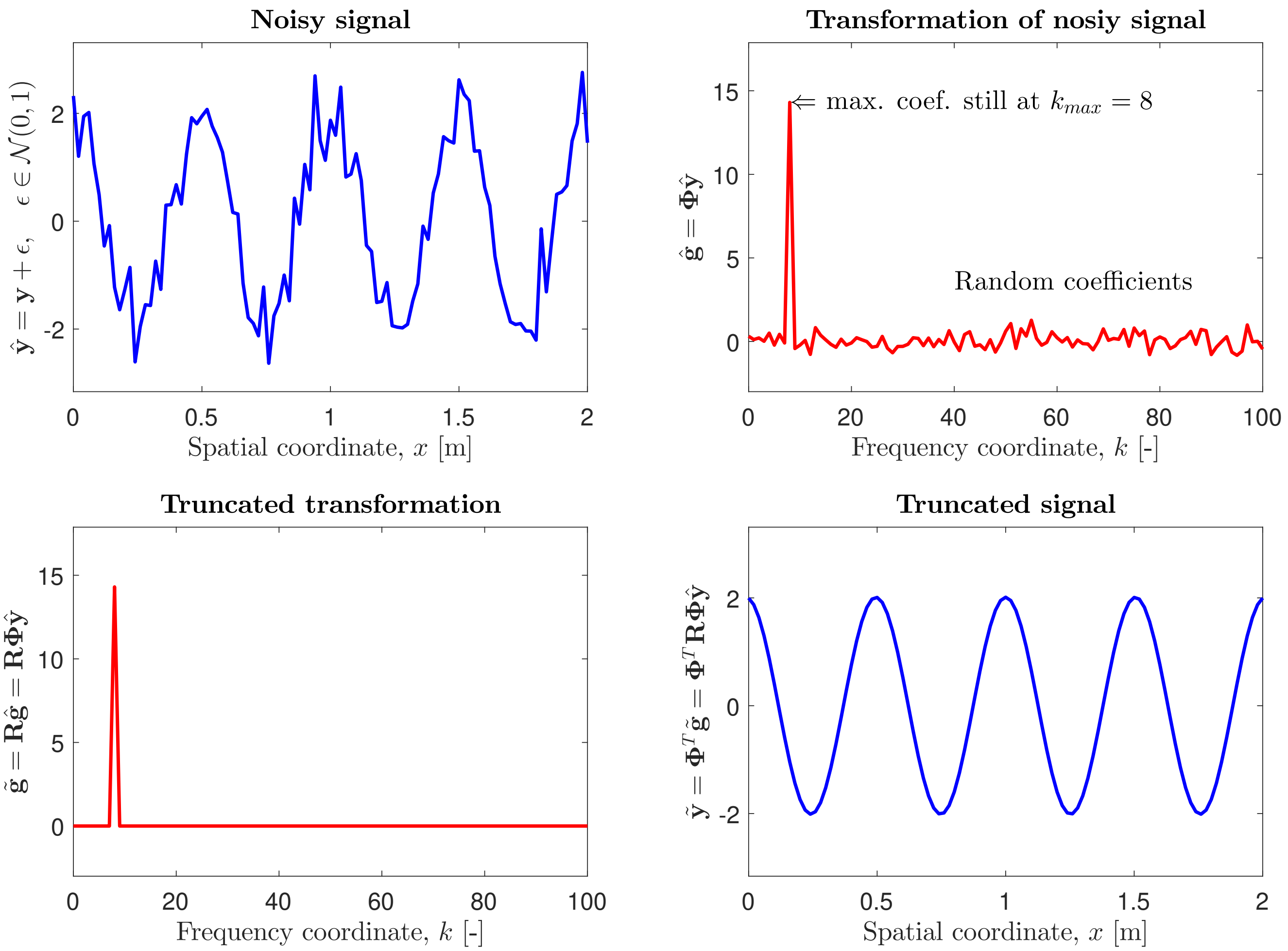


Figure 3.3: Illustration of truncated DCT on the same signal shown in Figure 3.2 but with added Gaussian noise $\varepsilon \in \mathcal{N}(0,1)$. Random vectors are indicated with the $\hat{\cdot}$. **Top left** displays the signal with added Gaussian noise. **Top right** displays the DCT of the noisy signal, where the random coefficients are clearly visible. This highlights an important observation, if the input signal is random, the DCT coefficients will be random as well and the characteristic of an ensemble of random signals is the expected value. **Bottom left** depicts the truncated DCT coefficients, simply obtained by restricting the full DCT coefficients to one non-zero coefficient, namely the largest DCT coefficient. The restriction matrix $\boldsymbol{R}$, is a diagonal matrix containing zeros everywhere, except the $k_{\max}$-th diagonal entry. See equation (3.23) for formal definition. **Bottom right** shows the truncated signal, obtained by mapping the truncated DCT back to the spatial domain.

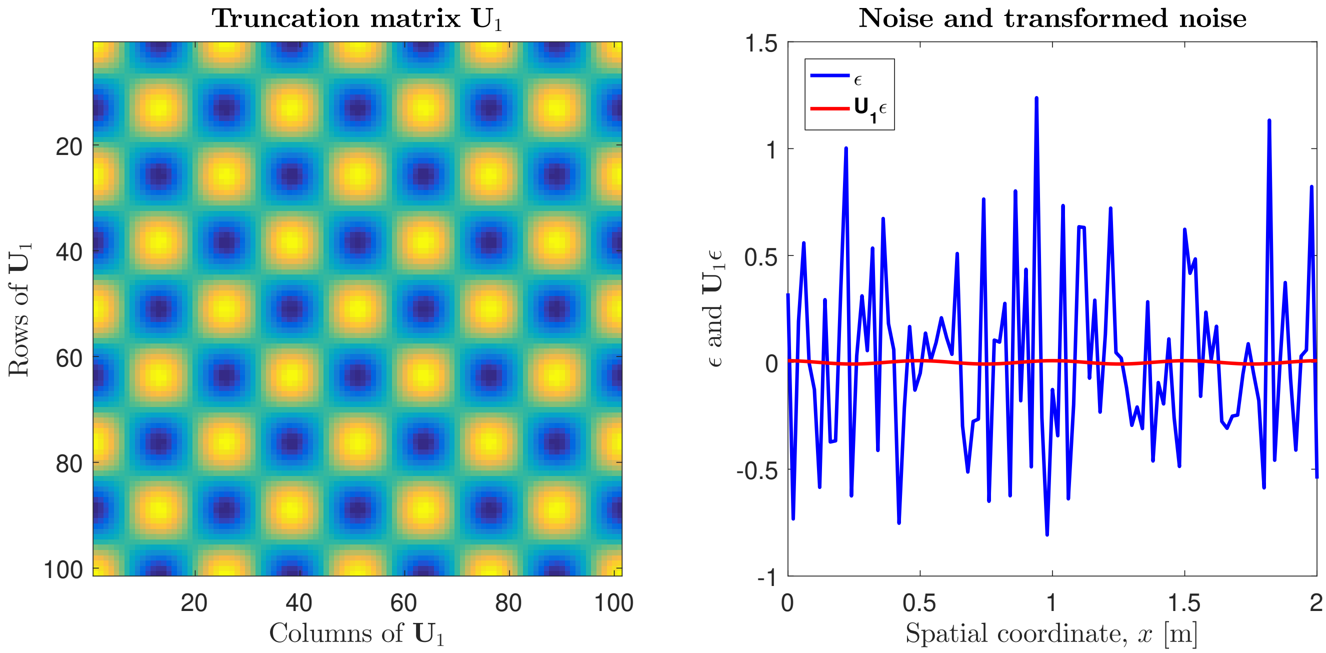


Figure 3.4: **Left** displays the truncation matrix $\boldsymbol{U}_1$ associated with the example in Figure 3.3. Clearly visible is the linear dependence of the columns resulting from the rank 1 truncation matrix, entirely defined by the dominant basis function. **Right** shows the noise which was added to the simple cosine wave and its transformed signal. The transformed noise is clearly in the column space of $\boldsymbol{U}_1$ but also heavily damped. The reason for this is the rank 1 DCT approximation using only one basis-vector (which has unit length) to construct $\boldsymbol{U}_1$, meaning that the l2-norm of the columns of $\boldsymbol{U}_1$ are much smaller than 1.

where $\boldsymbol{y} = [2\cos(4\pi x_0) \quad 2\cos(4\pi x_1) \quad \dots \quad 2\cos(4\pi x_{N-1})]^T$ and $\boldsymbol{\varepsilon} \in \mathscr{N}(0,1)$ as defined as figure 3.2 and 3.3. Since for this simple example, the reconstruction of the original signal containing merely one basis-vector, the column space of $\boldsymbol{U}_1$, denoted by $C(\boldsymbol{U}_1)$, is spanned entirely by this one (dominant) basis-vector. Because the result of a matrix-vector multiplication will always “fall” in the column space of the matrix which multiplies the vector, meaning that the random noise is simply transformed to “fall” in $C(\boldsymbol{U}_1)$. Adding two vectors that are both in this one dimensional[10] column space, $C(\boldsymbol{U}_1)$, will result in a vector that is still in $C(\boldsymbol{U}_1)$[11], providing this smooth signal displayed in the bottom right of figure 3.3.

3.4 Distance-based modeling

Distance-based modeling has recently gained interest in areas like Spatial Uncertainty Quantification and model[12] selection, see e.g. (Caers, Park, & Scheidt, 2010), (Scheidt & Caers, 2009), (Fenwick & Batycky, 2011) or (Scheidt, Caers, et al., 2009). Distance-based modeling formulates the model uncertainty/variability in terms of a distance (Caers et al., 2010), which measures the dissimilarity between any two models $X, Y \in \mathbb{R}^N$, as a scalar value namely $d(X,Y) \in \mathbb{R}$. When this distance is also a metric (see definition below) then one can speak about modeling in metric space, which has certain advantages. The reason for this is that the complexity or dimensionality can be drastically reduced in Metric Space by choosing a distance which has a large correlation (positive or negative) with the purpose of the modeling

---

[10] Note that the dimensionality of a vector space is determined by cardinality of the basis of that vector space, i.e. the number of basis-vectors in the basis.

[11] Note that this is the very definition of a subspace, other than the multiplicative property as well as containing the zero-vector.

[12] Model refers here to a single realization from an ensemble of models, i.e. a sample from a set of realizations. Where realization is defined as the outcome of a stochastic or process based simulation, which in this work serves as the input to a reservoir simulator.

endeavor (Caers et al., 2010). Due to the large intrinsic variability between all the models, a simple Euclidean distance between all of them will not result in a drastic reduction of the dimensionality, however choosing the distance between all models defined along a certain trajectory[13] might greatly reduce the dimension (Caers et al., 2010) and (Fenwick & Batycky, 2011). Note that the purpose of the modeling, governing this choice of distance, could cover the prediction of water breakthrough in a secondary oil recovery process or the concentration of a contaminant at a given point in space-time, advecting with the fluid through the reservoir.

3.4.1 Formal definitions

Definition of Metric Space

A metric space $(\boldsymbol{W}, d)$, where $\boldsymbol{W}$ is a set and d is a metric on the set $\boldsymbol{W}$ defined as the function $d : \boldsymbol{W} \times \boldsymbol{W} \to \mathbb{R}$ satisfying the following four axioms (for $x, y, z \in \boldsymbol{W}$):

1. $d(x,y) \geq 0$
2. $d(x,y) = 0 \Leftrightarrow x = y$
3. $d(x,y) = d(y,x)$
4. $d(x,z) \leq d(x,y) + d(z,y)$

In this work, the matrix $\boldsymbol{D}$ denotes the square and symmetric distance matrix, usually of size $M \times M$ where M denotes the size of the ensemble $\boldsymbol{W}$, where the (i,j)-th entry of $\boldsymbol{D}$ is defined as

$$\delta_{i,j} = d(\boldsymbol{w}_i, \boldsymbol{w}_j), \quad i,j = 1,2,\ldots,M \quad \forall \boldsymbol{w}_i, \boldsymbol{w}_j \in \boldsymbol{W} \tag{3.29}$$

where $d(\cdot,\cdot)$ satisfies the above given definition of a metric on a set, i.e. any allowable dissimilarity function that is a metric, e.g Euclidean or (Modified) Hausdorff distance (Dubuisson & Jain, 1994).

Definition of Eigenvalue Decomposition

For a more fundamental overview of the theory, the following two references should suffice (Lay, 2003) and (Strang, 1993). However, the definition of the eigenvalue decomposition is given below, since it is vital for solving the Multi-Dimensional Scaling (MDS) algebraically.

The eigenvalue decomposition is defined as a matrix factorization which diagonalizes the square $M \times M$ matrix $\boldsymbol{A}$ into

$$\boldsymbol{A} = \boldsymbol{R}\boldsymbol{\Lambda}\boldsymbol{R}^{-1} \tag{3.30}$$

where $\boldsymbol{R}$ is the $M \times M$ invertible matrix of eigenvectors (as its columns) and $\boldsymbol{\Lambda}$ is a $M \times M$ diagonal matrix with the corresponding eigenvalues (on its main diagonal). Note that in order to diagonalize $\boldsymbol{A}$ in this way, $\boldsymbol{R}$ needs to be nonsingular, i.e. there must exist M-linearly independent eigenvectors such that the rank of $\boldsymbol{R}$, $\operatorname{rank}(\boldsymbol{R}) = M$. Sometimes this decomposition is written in the following form

$$\boldsymbol{A}\boldsymbol{r}_i = \lambda_i \boldsymbol{r}_i, \quad i = 1,2,\ldots,M \quad \forall \boldsymbol{r}_i \neq 0 \tag{3.31}$$

where $\boldsymbol{r}_i$ denotes the i-th eigenvector with its corresponding eigenvalue, denoted by λ_i, such that $\boldsymbol{R} = [\boldsymbol{r}_1 \quad \boldsymbol{r}_2 \quad \ldots \quad \boldsymbol{r}_M]$ and $\boldsymbol{\Lambda} = \operatorname{diag}(\lambda_1, \lambda_2, \ldots, \lambda_M)$.

One important property of the eigenvalue decomposition is that if the matrix $\boldsymbol{A}$ is square symmetric, which is true for many applications as you can see below, the eigenvectors and eigenvalues are always real.

---

[13] Trajectory could be streamlined from well to well, where the distance could be based on the Time of Flight (TOF) along this streamline.

Moreover, the eigenvectors are orthogonal (if $\lambda_i \neq \lambda_j \quad \forall i,j \in \{1,2,\dots,M\}$) or can be chosen orthogonal (if $\lambda_i = \lambda_j$) (Searle & Khuri, 2017). This will reduce equation (3.30) to

$$\boldsymbol{D} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^T \tag{3.32}$$

where $\boldsymbol{D}$ is a square symmetric matrix and $\boldsymbol{Q}$ is an orthogonal matrix with orthonormal eigenvectors as its columns such that $\boldsymbol{Q}\boldsymbol{Q}^T = \boldsymbol{Q}^T\boldsymbol{Q} = \boldsymbol{I}$.

Definition of the centering matrix

Another special matrix, used in the classical MDS algorithm, is called the centering matrix (Marden, 1996), defined as

$$\boldsymbol{C}_M = \boldsymbol{I}_M - \frac{1}{M}\mathbf{1}\mathbf{1}^T \tag{3.33}$$

where $\boldsymbol{I}_M$ is the $M \times M$ identity matrix and $\mathbf{1}$ is a column vector of $M \times 1$ ones, such that $\mathbf{1} = [1 \quad 1 \quad \dots \quad 1]^T$. The centering matrix $\boldsymbol{C}_M$ has some interesting properties, namely the following

1. It has $M-1$ eigenvalues of 1 and one eigenvalue of 0, i.e. $\text{rank}(\boldsymbol{C}_M) = M-1$.
2. The nullspace of $\boldsymbol{C}_M$, denoted $N(\boldsymbol{C}_M)$, is one-dimensional and is clearly along the vector $\mathbf{1}$, since $\boldsymbol{C}_M\mathbf{1} = \mathbf{0}$.
3. It is a projection matrix, namely a projection onto the $M-1$ dimensional subspace which is, as defined by the four fundamental subspaces of Linear Algebra (Strang, 1993), orthogonal to $N(\boldsymbol{C}_M^T) = N(\boldsymbol{C}_M)$, since $\boldsymbol{C}_M^T = \boldsymbol{C}_M$.

These properties are useful when trying to understand the MDS algorithm and result of the MDS projection, defined below. Also note that due to $\text{rank}(\boldsymbol{C}_M) = M-1$, the linear mapping $\boldsymbol{C}_M$ is not invertible.

3.4.2 Multi-Dimensional Scaling

The above definitions are enough to introduce the algebraic formulation of the Multidimensional scaling (MDS) as defined in (Borg & Groenen, 2005)[14], (Caers et al., 2010) or (Caers, 2011). The algorithm for classical MDS, taken from (Borg & Groenen, 2005) chapter 12, is defined as follows

1. Compute the square symmetric matrix of squared dissimilarities[15] $\boldsymbol{D}.^2$ or $\boldsymbol{D}^{(2)}$
2. Double center the matrix $\boldsymbol{D}^{(2)}$, using equation (3.33) for the centering matrix $\boldsymbol{C}_M$, such that

$$\boldsymbol{B} = -\frac{1}{2}\boldsymbol{C}_M\boldsymbol{D}^{(2)}\boldsymbol{C}_M \tag{3.34}$$

3. Compute the eigenvalue decomposition of $\boldsymbol{B}$ using equation (3.30), such that

$$\boldsymbol{B} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^T \tag{3.35}$$

4. Ordering the eigenvalues as $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_m \geq 0 \geq \lambda_{m+1} \geq \dots \geq \lambda_M$ and the eigenvectors accordingly, such that $\boldsymbol{\Lambda}_m$ and $\boldsymbol{Q}_m$ contain the first m non-zero eigenvalues and associated eigenvectors respectively, the new coordinates can be expressed as

$$\boldsymbol{X}_m = \boldsymbol{Q}_m\boldsymbol{\Lambda}_m^{1/2} \tag{3.36}$$

   where the square root of a diagonal matrix is simple the square root of the diagonal entries.

---

[14] For a more detailed introduction, a geometric interpretation of the MDS or a more in-depth review of the MDS see this reference.

[15] Here $.^2$ or $^{(2)}$ denotes the square of the elements of the matrix $\boldsymbol{D}$.

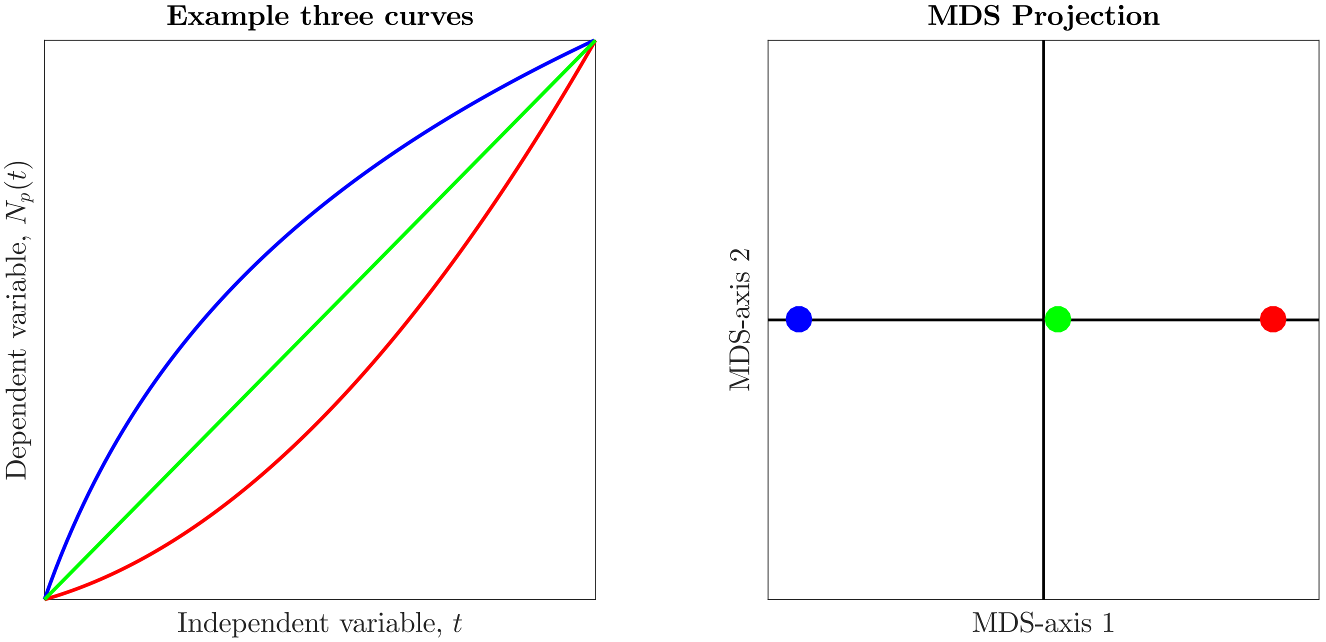


Figure 3.5: **Left** displays three simple curves, in this case generated using a logarithmic function, linear function and translated quadratic function. The Euclidean distance is computed between all the curves using equation (3.37), resulting in the distance matrix $\boldsymbol{D}$ which is then used in the classical MDS algorithm to obtain the figure on the **right**. The **right** part of this figure clearly shows that the first axis of the MDS projection explains "all" of the projection, since from this projection the original distance matrix can be reconstructed in it's entirety, while having a residual error of $||\boldsymbol{D}_{\text{recon}} - \boldsymbol{D}||_2 \approx 0$.

A nice property of the classical scaling algorithm is, as mentioned by (Borg & Groenen, 2005), that the dimensions are nested. This means that a truncated MDS, e.g. a projection onto a two-dimensional Cartesian plane, can be achieved by taking the first two dimensions of the coordinate matrix $\boldsymbol{X}_m$. Finally, if the distance function used to construct $\boldsymbol{D}$ happens to be a Euclidean distance, the coordinates of the MDS projection $\boldsymbol{X}_m$ using the classical scaling algorithm, are found up to a rotation (Borg & Groenen, 2005).

To illustrate MDS in practice, a simple example is shown below where the Euclidean distance is chosen for $d(\cdot,\cdot)$, which is computed for three artificial curves, e.g. time as independent variable and oil production as the dependent variable. The resulting distance matrix $\boldsymbol{D}$ is then used according to the algorithm defined above, resulting in the MDS coordinate projection onto a two-dimensional Cartesian plane. For completeness, the formal definition of the Euclidean distance between two vectors, $\boldsymbol{w}_i$ and $\boldsymbol{w}_j$ in $\mathbb{R}^N$ is

$$d_{i,j}^{\text{Eucl}} = d(\boldsymbol{w}_i, \boldsymbol{w}_j) = \sqrt{\sum_{n=1}^{N} (\boldsymbol{w}_{i,n} - \boldsymbol{w}_{j,n})^2} = \sqrt{(\boldsymbol{w}_i - \boldsymbol{w}_j)^T (\boldsymbol{w}_i - \boldsymbol{w}_j)} \tag{3.37}$$

Note that in this simple example with only three curves, a one-dimensional projection is sufficient. The quality of the projection[16] is heavily dependent on the size of the eigenvalues of $\boldsymbol{B}$. However, generally in Earth Science applications, no more than six dimensions are required for a sufficiently good projection (Caers et al., 2010).

---

[16] Where quality refers to the magnitude of $||\boldsymbol{D}_{\text{recon}} - \boldsymbol{D}||_2$, where $\boldsymbol{D}_{\text{recon}}$ is the reconstructed distance matrix based on the MDS coordinates (projection).

3.4.3 Clustering and model selection

The effort of dividing a set of objects into groups, such that objects in any one group are more "similar" to each member of the group than to objects outside of the group, is called clustering and the group is generally referred to as a cluster. Clustering is useful in many applications and often plays an important role in areas like data mining. Clustering in Earth Sciences is often used to derive a reduced representation of the full data set, where each cluster is said to represent a certain behavior of the full data system or set.

The main class of clustering used in this work is "Centroid-based clustering". The centroid-based clustering comprises the most well-known clustering technique, k-means clustering, as well as other methods such as k-medoids which differs from k-means in that it restricts the centroid to be a member of the data set.

The macros used to execute the clustering is done via the Statistics and Machine Learning Toolbox in MATLAB©(MathWorks, 2017), which contains the k-means and k-medoids algorithms. The first step in the algorithm of k-means is to decide on the number of clusters, denoted n_k. These clusters, or "means", can be initialized in various ways from randomly selecting the initial means (cluster centers) to various other methods see (Steinley & Brusco, 2007). The k-means algorithm is then followed by a two-step procedure which iterates between 1 and 2 until convergence is reached.

1. Assigning every data point to the closest mean (cluster center), where closest refers to the least squared Euclidean distance.
2. Updating the n_k clusters by calculating new means (centers), based on the mean of the cluster members.

See the example of a simple clustering in figure 3.6.

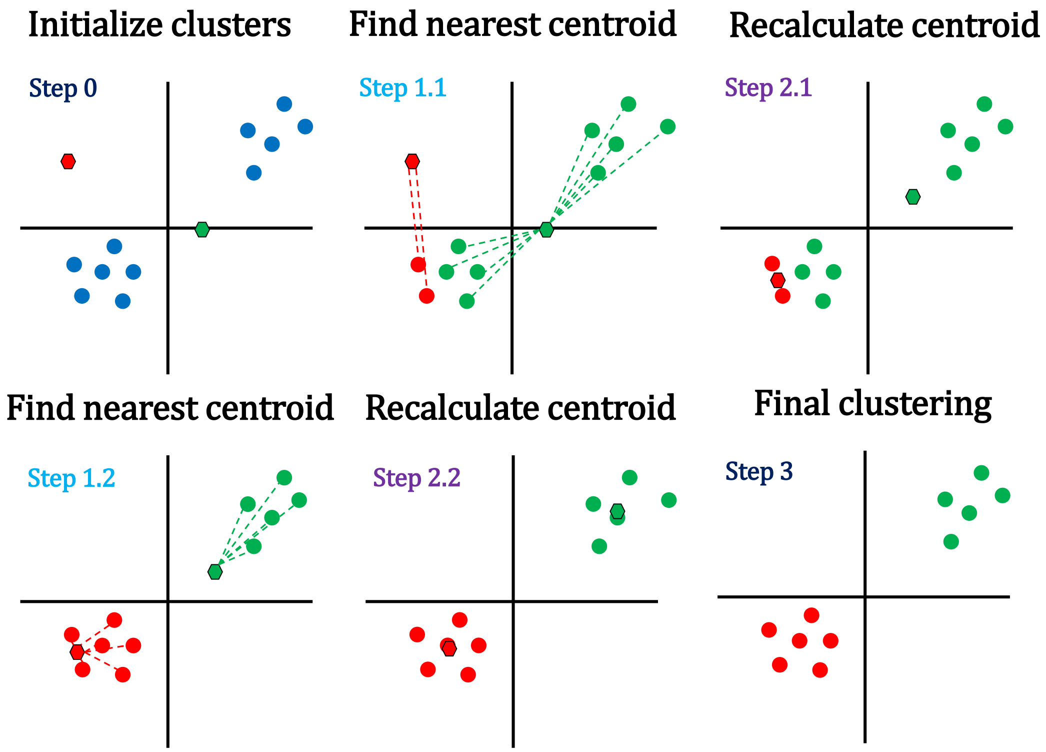


Figure 3.6: Depicting the "four" steps in k-means clustering.

The main disadvantage of k-means clustering is that it's sensitive to outliers and doesn't necessarily converge when non-Euclidean distances are used (Jin & Han, 2010). Also, the final k-means clustering depends on the initial clustering. This is typically solved by running the k-means algorithm multiple times and choosing the result with the least within-cluster variation defined as

$$W_{\text{var}} = \sum_{k=1}^{n_k} \Big(\sum_{C(i)=k} ||X_i - \bar{X}_k||_2^2 \Big) \tag{3.38}$$

where X_i is the i-th data point, $C(i) = k$ is a function that assigns data point X_i to cluster k and $\bar{X}_k$ is the mean or center of the k-th cluster. The above mentioned convergence criteria is typically set to the point where the within-cluster variation doesn't change over the next iteration.

K-medoids, on the other hand, allows for non-Euclidean distances to be used and is less sensitive to outliers. The difference in implementation, as also mentioned above, is that the center of the cluster is also a data point. Initialization and the first step of the algorithm don't differ much from that of k-means however the second step does differ. One algorithm that finds the (local) optimum configuration of medoids is Partitioning Around Medoids (PAM). By iteratively swapping the current medoid with an associated non-medoid and computing the quality of the clustering, the algorithm attempts to improve the quality of the clustering. "All possible combinations of representative and non-representative points are analyzed, and the quality of the resulting clustering is calculated for each pair. An original representative point is replaced with the new point which causes the greatest reduction in distortion function. At each iteration, the set of best points for each cluster form the new respective medoids." - (Jin & Han, 2010).

K-medoids has the advantage that the medoids can be chosen as representatives of the cluster immediately since they are part of the data set. This is a slight advantage when considering model selection process, where model selection refers to selecting representative models which approximate the behavior of the exhaustive[17] set. For a more detailed review of the clustering methods mentioned here, the reader is referred to the following references (Kaufman & Rousseeuw, 1987), (Hartigan & Wong, 1979), (Kaufman & Rousseeuw, 2009) and (Jin & Han, 2010).

3.4.4 Kernel trick

The shape and linearity of the MDS projection have implications on the resulting clustering. The non-linear behavior of clusters will not be accurately described by clustering analysis since the distances that underly the clustering are linear. This leads to the concept of a high-dimensional "Feature Space" where this linear separation or structure of the clusters does exist in the form of a hyperplane. For defining a map from any metric space, only the dot-product is necessary. Therefore the high-dimensional function that maps one metric space to another doesn't need to be known explicitly, only its dot product. The function that represents this dot product in the feature space is known as a kernel function (Caers et al., 2010). Typical kernel function used are the Gaussian Radial Basis function, defined in (Caers et al., 2010) as

$$K_{i,j} = k(\boldsymbol{x}_i, \boldsymbol{x}_j) = exp\Big(- \frac{(\boldsymbol{x}_{m,i} - \boldsymbol{x}_{m,i})^T (\boldsymbol{x}_{m,i} - \boldsymbol{x}_{m,i})}{2\sigma^2} \Big) \tag{3.39}$$

where $\boldsymbol{x}_{m,i}$ are the m-dimensional MDS projection coordinates of $\boldsymbol{x}_i$.

This procedure is known as the "Kernel Trick". The value for $\boldsymbol{\sigma}$ is typically chosen at around $0.2\big(max(\boldsymbol{X_m}) - min(\boldsymbol{X_m})\big)$ (Scheidt & Caers, 2009).

---

[17] Exhaustive meaning total set.

4

Determination spatial scale

A relevant[1] spatial scale in reservoir simulation may be subdivided in a static and dynamic spatial scale. The main reason for this distinction is that relative permeability is a non-linear function of saturation, such that a dead oil model can show highly non-linear behavior in its solution. Due to this non-linear relation, there may be a difference between the dynamic and the static spatial scales, particularly in the case that the static spatial scale is based solely on the input parameters and the dynamic spatial scale based on the actual flow response obtained from the forward simulation in ADGPRS.

4.1 Determination static spatial scale

The main tool to determine if a dominant static spatial scale exists and how it evolves when coarsening is the Discrete Cosine Transform (DCT). The DCT is explained in great detail in the previous chapter. The main property of the DCT is its efficient energy compaction which in this work is utilized to determine the dominant basis-vector of a transformation.

4.1.1 Zero-centered mean logarithmic signal

Before actually transforming a signal, the linear property of the DCT is used to illustrate that subtracting the mean from the signal $\boldsymbol{y}$, doesn't change its characteristics[2]. Consider the following discrete signal, $\boldsymbol{y} = [(100+cos(2\pi x_0)) \quad (100+cos(2\pi x_1)) \quad \dots \quad (100+cos(2\pi x_{N-1}))]^T$, where $\boldsymbol{x} = [x_0 \quad x_1 \quad \dots \quad x_{N-1}]^T$ is the position vector. The constant 100 is not actually considered as a characteristic of the signal. Using equation (3.14) and (3.28), the linear transformation can be written as

$$\boldsymbol{g}^\star = \boldsymbol{\Phi}\boldsymbol{y}^\star = \boldsymbol{\Phi}(\boldsymbol{y} - \bar{y}\mathbf{1}) = \boldsymbol{\Phi}\boldsymbol{y} - \boldsymbol{\Phi}\bar{y}\mathbf{1} \tag{4.1}$$

where the superscript $\star$ is used to indicate that the mean of the signal, denoted $\bar{y}$, is subtracted. Note that every row of $\boldsymbol{\Phi}$ contains a basis-vector, and all the basis-vectors, except the first constant zero-frequency basis-vector, have a zero mean. This last part is important since taking the inner product of a zero-mean signal with a scaled vector of ones, will result in zero, similar to the fact that the sum of a zero mean signal is also zero. Therefore, the right-hand-side (RHS) of equation (4.1) can be written as

$$\boldsymbol{\Phi}\boldsymbol{y} - \boldsymbol{\Phi}\bar{y}\mathbf{1} = \boldsymbol{g} - \boldsymbol{\Phi}\bar{y}\mathbf{1} = [g_0 \quad g_1 \quad \dots \quad g_{N-1}]^T - [g_0 \quad 0 \quad \dots \quad 0]^T = [0 \quad g_1 \quad \dots \quad g_{N-1}]^T = \boldsymbol{g}^\star \tag{4.2}$$

which means that the full transformed signal, $\boldsymbol{g} = \boldsymbol{\Phi}\boldsymbol{y}$, is equal to $\boldsymbol{g}^\star$ except for the first zero-frequency coefficient, also referred to as DCT_0 coefficient (which handles the constant translation/offset of the original signal). The first coefficient of the two resulting vectors of the RHS of equation (4.1) is actually

---

[1] Note that relevant and dominant are used interchangeably throughout this work.

[2] Except for removing the zero-coefficient arising from the constant basis-vector for zero-frequency.

g_0 which can be shown by the definition of the mean and inner product of two vectors. The first DCT basis-vector can be written as

$$\boldsymbol{\varphi}_0 = \sqrt{\frac{1}{N}}\mathbf{1} \tag{4.3}$$

such that the first DCT coefficient, g_0, is given by the inner product (since $\boldsymbol{\Phi} = [\boldsymbol{\varphi}_0 \quad \boldsymbol{\varphi}_1 \quad \dots \quad \boldsymbol{\varphi}_{N-1}]^T$)

$$< \boldsymbol{\varphi}_0, \boldsymbol{y} > = \sqrt{\frac{1}{N}}y_0 + \sqrt{\frac{1}{N}}y_1 + \ldots + \sqrt{\frac{1}{N}}y_{N-1} = \sqrt{\frac{1}{N}}\sum_{n=0}^{N-1} y_n \tag{4.4}$$

The first resulting coefficient of the matrix-vector multiplication $\boldsymbol{\Phi}\bar{y}\mathbf{1}$ can be written similarly as

$$< \boldsymbol{\varphi}_0, \bar{y}\mathbf{1} > = \sqrt{\frac{1}{N}}\bar{y} + \sqrt{\frac{1}{N}}\bar{y} + \ldots + \sqrt{\frac{1}{N}}\bar{y} = N\sqrt{\frac{1}{N}}\bar{y} \tag{4.5}$$

Defining the mean function $\mu : \mathbb{R}^N \to \mathbb{R}$, explicitly as

$$\mu(\boldsymbol{y}) = \frac{1}{N}\sum_{n=0}^{N-1} y_n = \bar{y} \tag{4.6}$$

which, after substitution in equation (4.5), results in

$$< \boldsymbol{\varphi}_0, \bar{y}\mathbf{1} > = \sqrt{\frac{1}{N}}\sum_{n=0}^{N-1} y_n \tag{4.7}$$

such that

$$g_0 = < \boldsymbol{\varphi}_0, \boldsymbol{y} > = < \boldsymbol{\varphi}_0, \bar{y}\mathbf{1} > = g_0 \tag{4.8}$$

In the present study, the static property used for the DCT constitutes the permeability or transmissibility, as these affect the flow response more than porosity. Since permeability is logarithmically related to porosity, it is chosen to calculate the characteristic of the signal $\check{\boldsymbol{y}}$, defined as

$$\check{\boldsymbol{y}} = \ln(\boldsymbol{y}) - \mu\big(\ln(\boldsymbol{y})\big)\mathbf{1} = \ln(\boldsymbol{y}) - \Big(\frac{1}{N}\sum_{n=0}^{N-1}\ln(y_n)\Big)\mathbf{1} \tag{4.9}$$

such that $\check{\boldsymbol{g}}$ is equal to $\boldsymbol{g}$, where $\boldsymbol{g} = \boldsymbol{\Phi}\ln(\boldsymbol{y})$. Using the distributive property of the linear DCT, except for the first zero-frequency coefficient. Note that $\boldsymbol{\Phi}\ln(\boldsymbol{y}) \neq \boldsymbol{\Phi}\boldsymbol{y}$. Generally, for signals with a high signal-to-noise factor they are equal up to some scaling factor $\boldsymbol{\beta}$. Clear signals do not occur often in real world applications and on top of that, the coarser representations of the transmissibility obtained with the global upscaling technique is prone to generate a few outlying (relatively large) values (Holden & Nielsen, 2000) and (Y. Chen et al., 2003). The effect on a simple one-dimensional example is shown below, where it is clearly favored to estimate the dominant characteristic of the original signal through $\check{\boldsymbol{y}}$ rather than using $\boldsymbol{y}^{\star}$, see definitions above.

Similar behavior is observed in two-dimensions, however, a larger outlying value is required to create similar erratic behavior in the transformed signal (such as seen in Figure 4.2). These larger values are however not uncommon, after the iterative procedure of the global upscaling required to converge to a positive-definite transmissibility matrix. See the example in Figure 4.1. In this example, an actual upscaled transmissibility field is used for the signal $\boldsymbol{Y}$, namely the transmissibility in the y-direction, such that $\boldsymbol{Y} = \boldsymbol{T}_y$ in this case. Clearly, the logarithmic transformation preserves the characteristics of the signal while DCT directly on the centered[3] signal results in the horrendous pattern in DCT domain indicating a sharp, irregular, discontinuity.

---

[3] Centered around the mean.

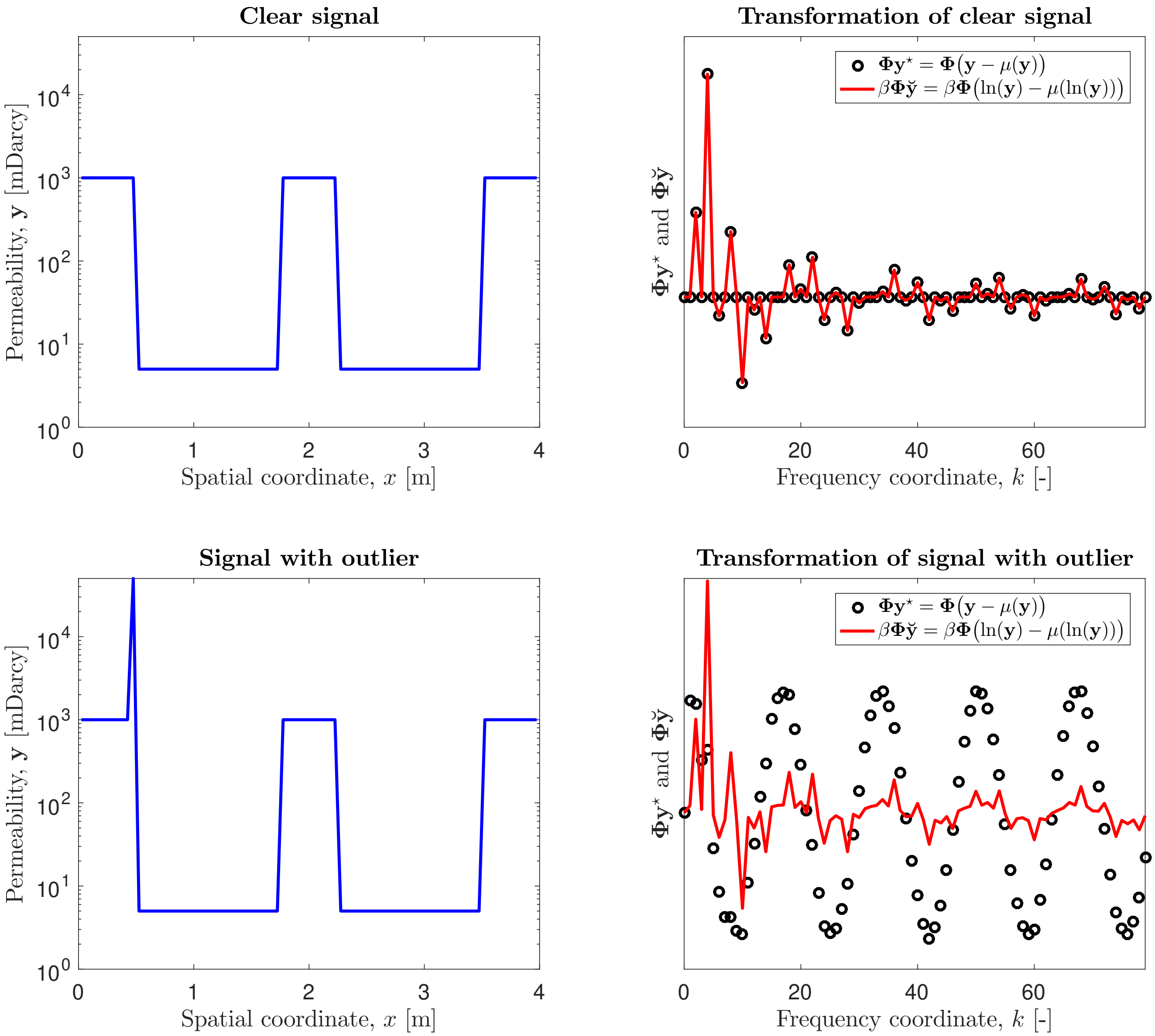


Figure 4.1: **Top left** displays the clear signal, in this case representing some sort of permeability, where the large values could indicate channel sands and the low values floodplain shales or clays. **Top right** shows the DCT of the clear signal for both $\boldsymbol{y}^\star$ and $\check{\mathbf{y}}$, which are defined in equations (4.1) and (4.9) respectively. For the clear signal, the DCT of both signals is equal up to a scaling factor β. **Bottom left** illustrates a hypothetical outlier, as could be obtained in the coarser representation of transmissibility from the global upscaling algorithm. **Bottom right** shows the DCT of the signal with outlier for both $\boldsymbol{y}^\star$ and $\check{\mathbf{y}}$. Clearly they are very different after introducing only a single outlying value. $\boldsymbol{y}^\star$ doesn't compact the energy of the signal at all while $\check{\mathbf{y}}$ still has the same structure as in the case of a clear signal, i.e. it still finds the same characteristic or dominant basis-function as the original signal. This robustness is appreciated and therefore determining the characteristics of $\check{\mathbf{y}}$ is favored in this work. Note that a sine-like shape of the coefficients in the DCT domain usually indicates a very sharp, irregular, discontinuity, such as the outlying value.

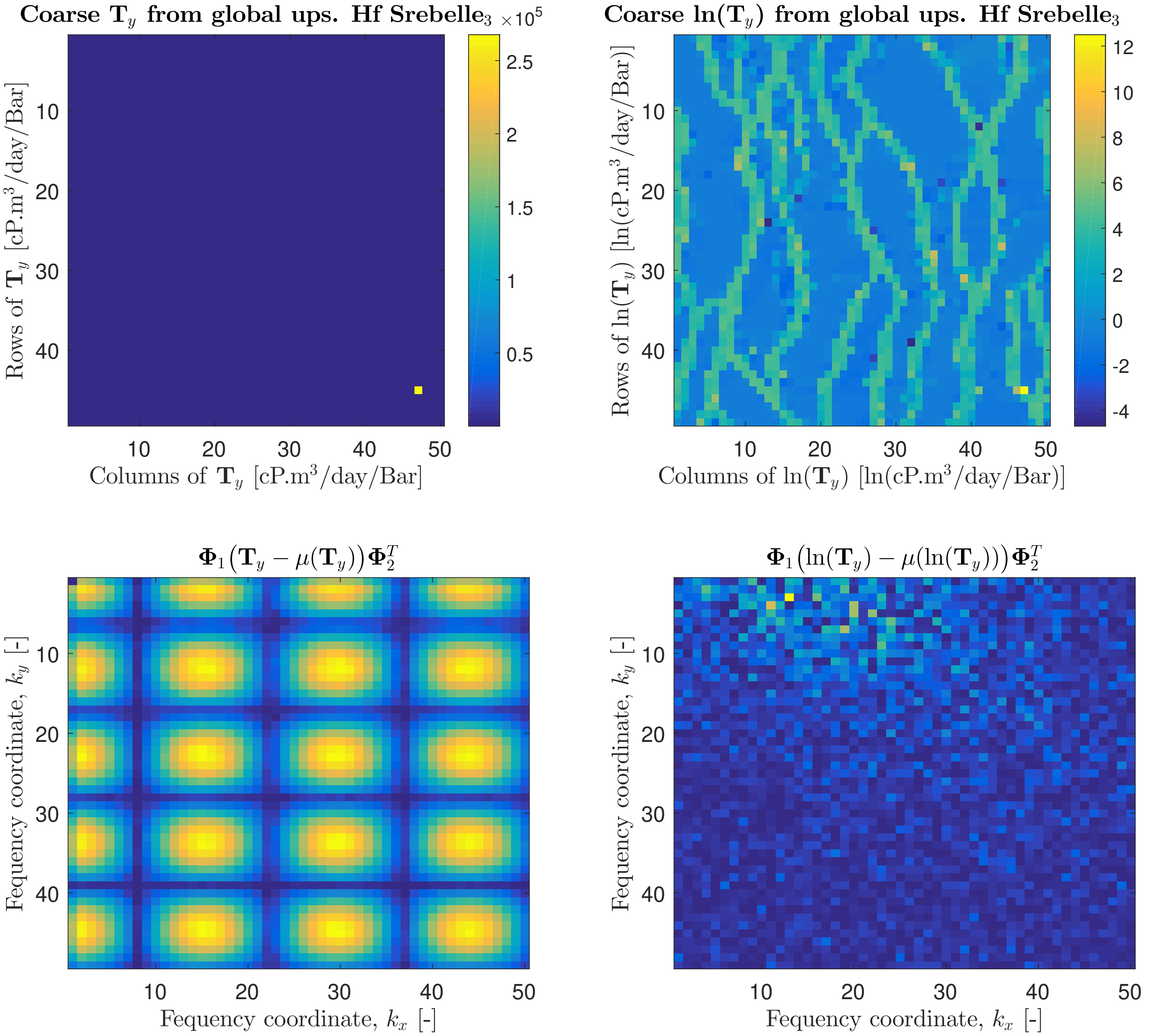


Figure 4.2: **Top left** displays the globally upscaled transmissibility in y-direction of one ensemble member of the Strebelle ensemble, clearly displaying a horrendous transmissibility value as described in previous chapter and in detail by Nielsen et al.,(2000) and Chen et al.,(2003). Note that the solution for $\boldsymbol{T}_y$ still minimizes the misfit between the fine-scale flux and coarse pressures while having zero non-zero transmissibility values and therefore is a valid solution. **Top right** shows the natural logarithm of the globally upscaled $\boldsymbol{T}_y$, still displaying the outlying value however much smaller in contrast. **Bottom left** shows similar erratic behavior of the signal in DCT domain as in the one-dimensional case (namely indicating a sharp, irregular, discontinuity). **Bottom right** illustrates a much more compact and realistic transformation of the signal.

4.1.2 DCT on ensemble of models across various coarsening levels.

Having justified the choice for transforming the characteristic of permeability $\check{\boldsymbol{y}}$ or $\check{\boldsymbol{Y}}$ rather than directly transforming the permeability $\boldsymbol{y}$ or $\boldsymbol{Y}$ respectively, the dominant basis-vectors and their evolution when coarsening is investigated. In this work, several sets of ensemble simulations with different characteristics are generated. The main ensemble which is used in this analysis is the High Fidelity (Hf) Strebelle$_3$, where the name arises from the Training Image (TI) used to generate the ensemble and the subscript 3 is used to indicate the main orientation of the of resulting channels[4]. The following image shows three particular ensemble members and their respective DCT.

As can be seen in 4.3, most of the energy of the transformation is contained in the low-frequency basis-vectors and the dominant (largest DCT coefficient) basis-vector is the same for these three particular realizations. The frequency of the dominant basis-vector is larger in the x-direction as opposed to the y-direction, this can be understood since alternations of channel and floodplain facies occur at much smaller distances compared to the wavelength (or sinuosity) of the channels. If the dominant basis-vector for each ensemble member is determined, a frequency map can be reconstructed which indicates the occurrence of a basis-vector being dominant in the ensemble members. See figure 4.4 for a graphical representation of this.

A geometric interpretation of this dominant basis-vector for the Strebelle$_3$ ensemble is shown in figure 4.5. The dominant basis-vector accurately predicts the geological features present in the Strebelle$_3$ ensemble. The average distance between the channels as well as the average wavelength of the channels is coinciding with the peaks and troughs of the 2-dimensional cosine wave.

---

[4] See chapter on Geological Modeling for more detail on the actual ensemble.

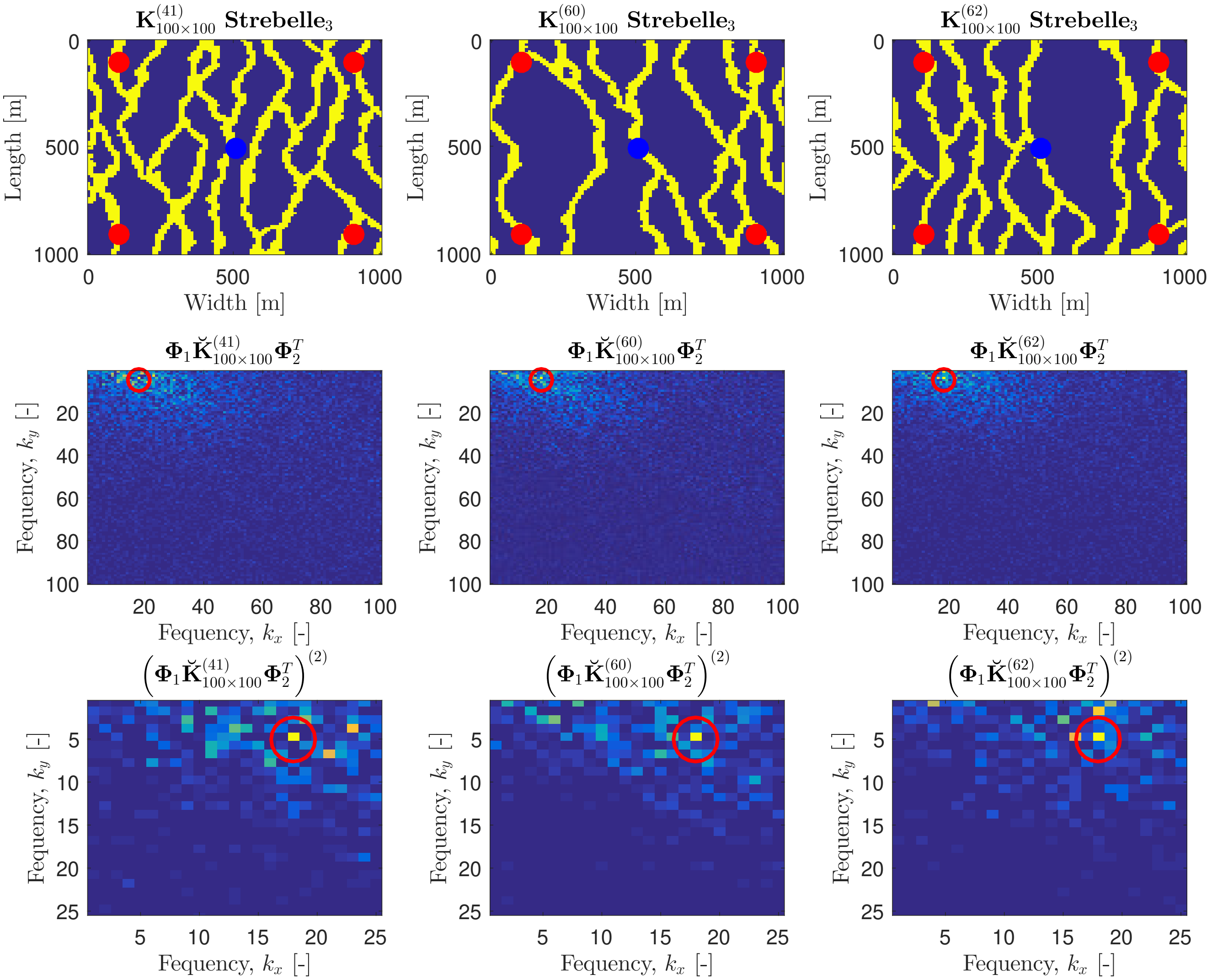


Figure 4.3: **Top row** displays three ensemble members of the Strebelle$_3$ ensemble, where the red dots indicate production wells and the blue dot represents the injector well. The yellow color indicates reservoir facies with a homogeneous permeability of 1000 [mD] whereas the purple color indicates non-reservoir facies of 5 [mD], see also chapter on Geological Modeling for more information. **Middle row** depicts the permeability in DCT domain, where the dominant basis-vector is indicated with the red circle. **Bottom row** shows a “zoomed in” version of the DCT coefficients squared, indicated with the superscript (2), in particular is shows the coefficients corresponding to the first $k_x \times k_y = 25 \times 25 = 625$ basis-vectors. Note that the colors of the **middle** and **bottom** rows are indicative of the magnitude of the DCT coefficients, namely purple equals the smallest coefficient and yellow equals the largest coefficient.

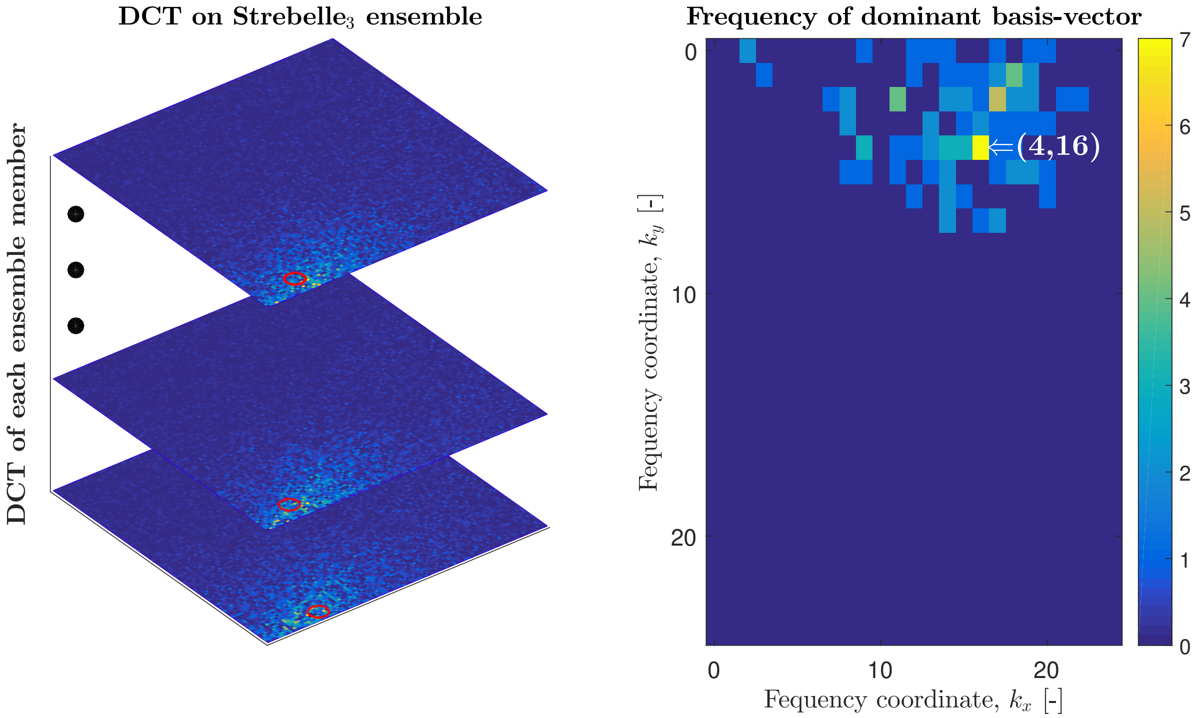


Figure 4.4: **Left** schematic shows the DCT of each ensemble member with the dominant basis-vector of each respective ensemble member indicated in red. Storing this dominant basis-vector for each ensemble member allows for the construction of a frequency map, displayed in the **right** image, where frequency means the amount of times a basis-vector was marked as dominant basis-vector of the transformation. E.g. the dominant basis-vector of the ensemble Strebelle$_3$ is considered to have a dimensionless frequency of $k_x = 16$ and $k_y = 4$ since it occurs a maximum number of times, in particular it is the dominant basis-vector for seven ensemble members.

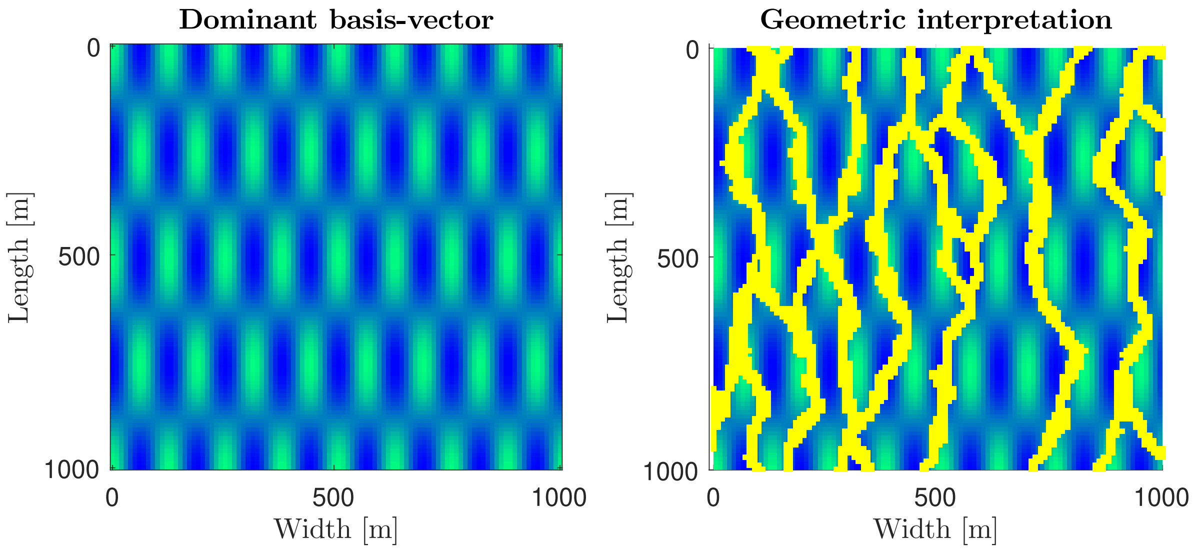


Figure 4.5: **Left** image depicts the dominant basis-vector of the Strebelle$_3$ ensemble. **Right** image displays the same basis-vector overlain with an ensemble member, showing a geometric interpretation of the dominant basis-vector. Clearly the dominant basis-vector describes the sinuosity (average wave-length of the channels) and average distance between the channels. These length measures are important parameters since they govern a minimum number of static grid-cells allowed to represent this signal and preserve the continuity and characteristics of the channels.

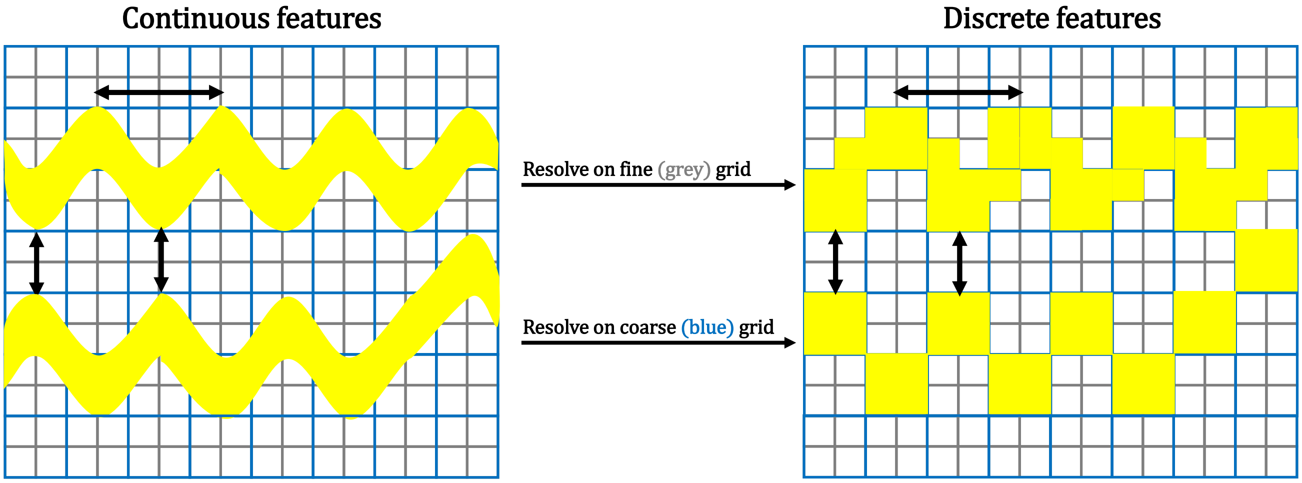


Figure 4.6: **Left** schematic shows continuous geological features (channels) projection on two Cartesian grids, blue and gray respectively. The blue grid dimensions are exactly half of the wavelength of the channels, while the gray grid is one fourth of the wavelength. **Right** schematic displays the mapping of these continuous features to the Cartesian grids. The blue grid is incapable of capturing face-to-face connectivity, an essential requirement for accurate flow and transport simulation if a Two-Point Flux Approximation is used for the spatial discretization. Appropriate connectivity, for this simple schematic, is achieved by the mapping to the gray (finer) grid.

Besides the geometric interpretation highlighted above, the frequency of the dominant basis-vector has other implications. Accurately representing a discrete cosine wave, such as the basis-vectors of the DCT transformation, requires certain grid dimensions. The minimal grid-size (Δx and Δy) required to represent the peaks and troughs accurately is half of the wavelength of the cosine wave. The resulting discrete signal will approximate a Haar-wavelet, which however doesn't capture face-to-face connectivity between grid-cells. An attempt to display this is shown in figure 4.6, where the continuous features on the left are resolved on two grid-sizes, blue and gray respectively. Δx and Δy of the blue grid corresponds to half of the wavelength of the channels, whereas the dimensions of the underlying gray grid are exactly half of the blue grid. Even though the wavelength of the channel, as well as the average distance between the channels, are preserved using half of the wavelength as grid dimension, the face-to-face connectivity isn't maintained which might cause numerical issues when a Two-Point Flux Approximation (TPFA) is used in the spatial discretization[5]. The face-to-face connectivity is nicely represented using the dimensions of the gray grid, leading to an estimation of the relevant static spatial scale, namely one-fourth of the dominant wavelength of the ensemble. Note that this is a clear disadvantage of a fixed Cartesian grid and can possibly be solved by using an unstructured grid or adaptive refinement.

The coarsening effect on the static spatial scale is examined in a similar way as the dominant basis-vector analysis on the fine-scale, however, now the parameter transmissibility is used since this is directly obtained from the global upscaling algorithm. In particular, the transmissibility in the y-direction is examined since this is parallel to the orientation of the paleoflow, causing a more continuous upscaled result w.r.t the transmissibility in the x-direction. Figure 4.8 displays the frequency maps of dominant basis-vectors across various ensemble scales.The dominant frequencies are preserved throughout the upscaling, however towards the $N_x \times N_y = 20 \times 20$ ensemble scale the dominant basis-vector shifts towards the lower frequency side of the spectrum.

---

[5] Two-Point Flux Approximation (TPFA) in a Finite Volume Method (FVM) means that the cell average of the Finite Volume is updated based on two neighboring cell for each dimension. E.g. a two-dimensional FVM-scheme requires information in four neighboring cells when using TPFA.

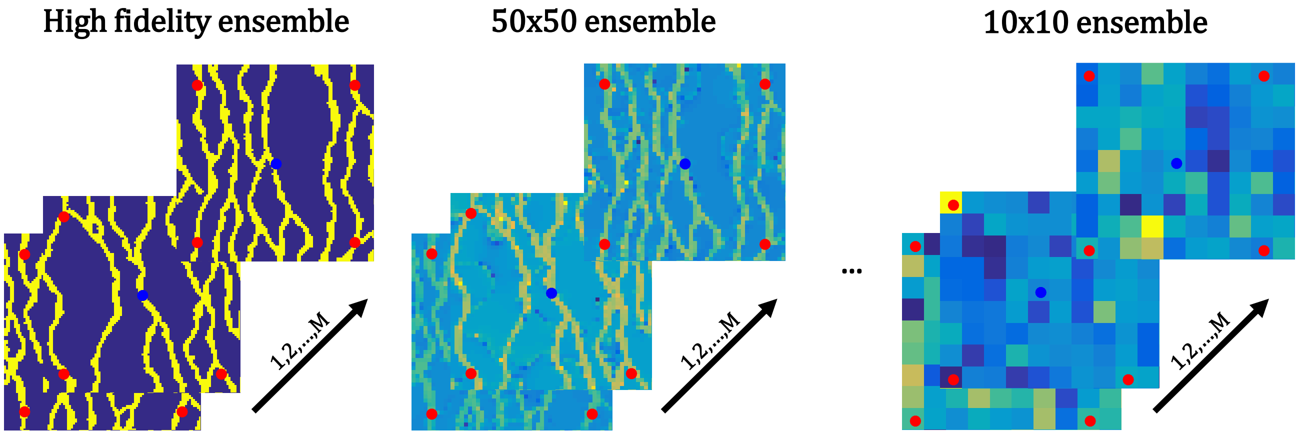


Figure 4.7: Depicting the hierarchical Strebelle$_3$ ensemble, with several levels of coarsening.

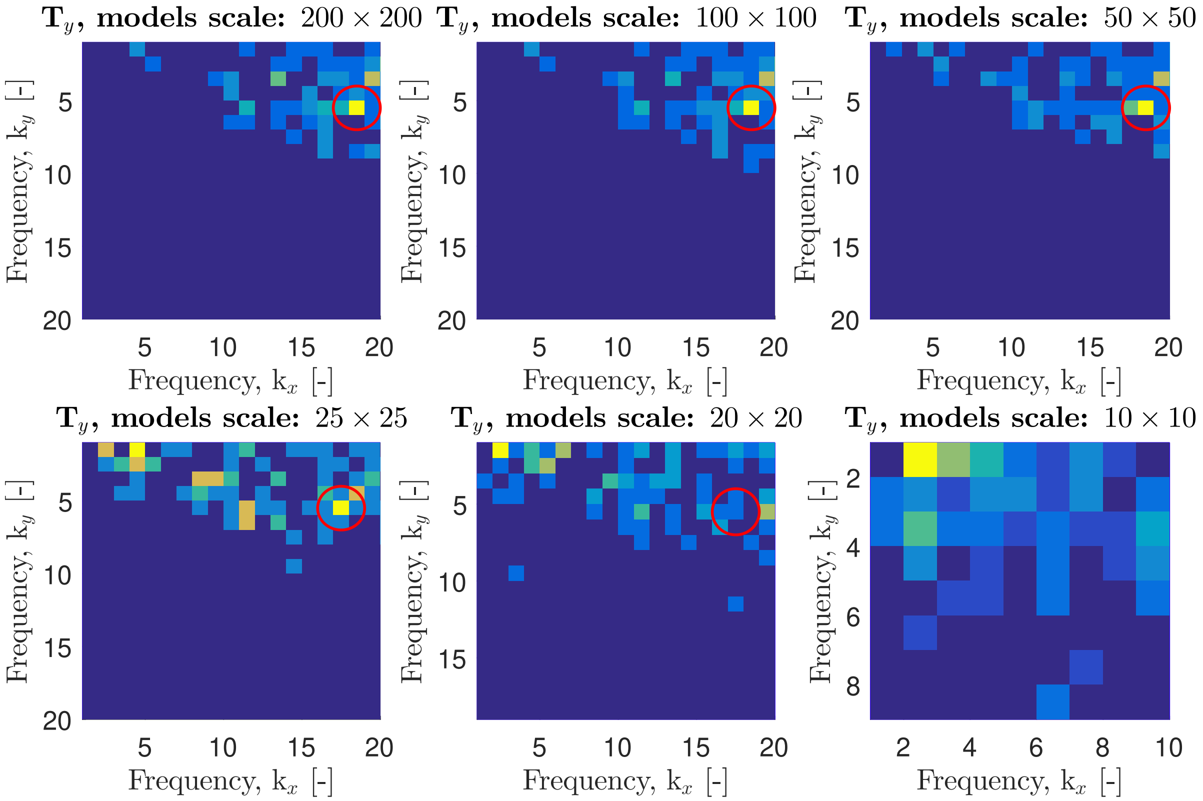


Figure 4.8: Depicting the evolution of the dominant basis-vectors across various ensemble scales, indicated in the title of each subplot by $N_x \times N_y$. The dominant characteristics are, on a static level, are clearly preserved during the upscaling algorithm, and only start to "fade away" around 20×20 model scale, while completely disappearing in the final upscaling step.

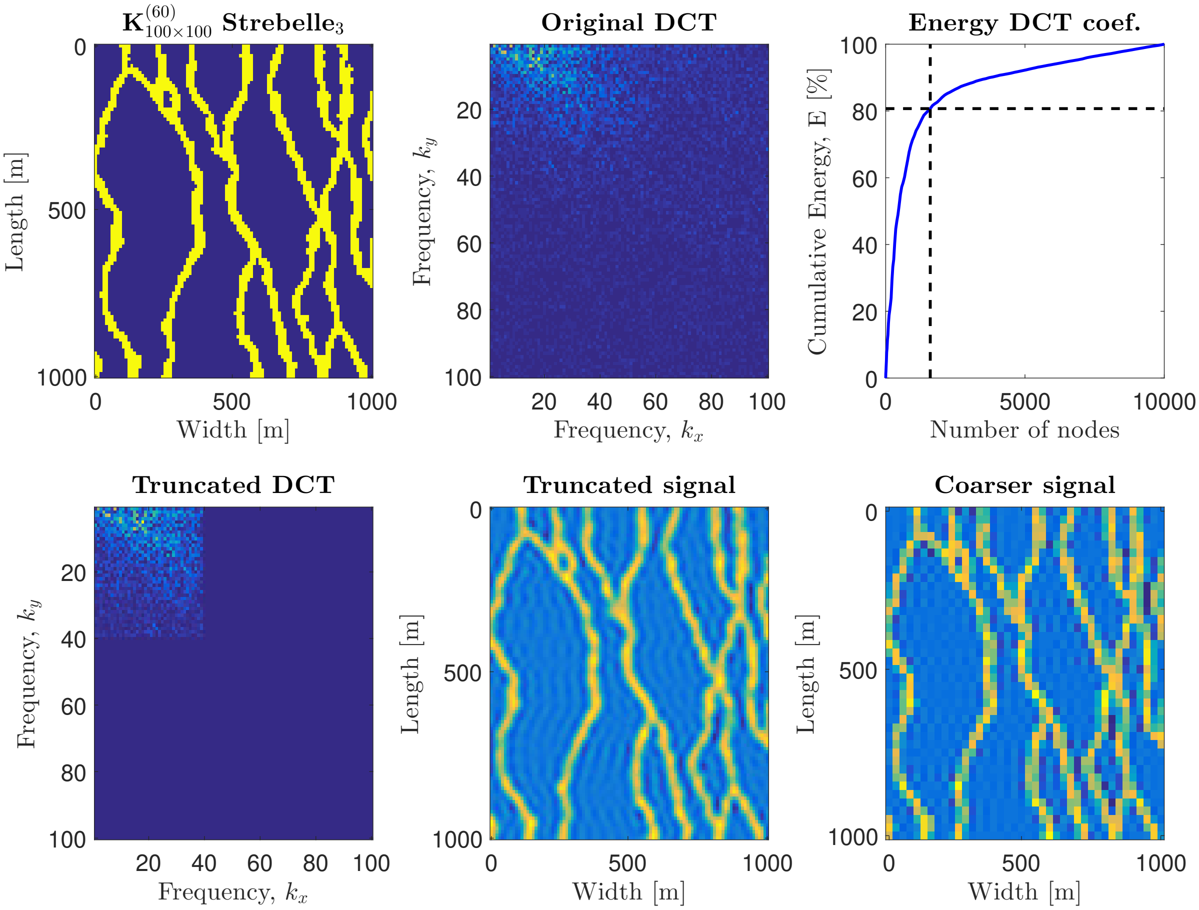


Figure 4.9: **Top left** depicts the permeability map of an ensemble member of Strebelle$_3$. **Top middle** displays the absolute values of the original DCT of the signal. **Top right** shows the cumulative energy contained in the DCT-coefficients. The arbitrary threshold, chosen at two times the largest dominant frequency of the DCT, is able to capture over 80% of the total energy of the original signal. **Bottom left** shows the non-zero elements of the truncated DCT. **Bottom middle** shows the truncated signal, using only the 40×40 non-zero DCT coefficients. **Bottom right** shows, instead of mapping the truncated DCT back to a 100×100 grid, a coarser mapping where the remaining non-zero coefficients are mapped to a grid of equal dimensions $N_x \times N_y = 40 \times 40$.

4.1.3 Total energy as a measure

Another way of viewing a relevant static spatial scale is through representing the cumulative energy of the DCT-coefficients as a function of increasing number of nodes. This is attempted in figure 4.9 for a particular realization from the Strebelle$_3$ ensemble. In figure 4.9, besides the cumulative energy, also truncated mappings using the arbitrary threshold are shown (both for limiting the amount of non-zero coefficients as well as restricting the number of nodes available for reconstructing the original signal).

Important to note is that the bottom right truncated reconstruction is similar to an upscaling algorithm. However, since it is not directly including any flow characteristics, the result is inferior to any flow based upscaling method. The main reason is that in flow-based upscaling a misfit between the coarse-scale pressure and fine-scale flux is minimized by fixing the coarse-scale transmissibilities. Coarsening using DCT will most likely provide a much poorer approximation of the fine-scale fluxes and pressure.

4.2 Determination dynamic spatial scale

There are multiple ways to examine a dynamic spatial scale. One way is to simply do a sensitivity study on a single deterministic realization[6] and examine when the deviation, for the particular modeling purpose, is unacceptable. The issue with this is illustrated in figure 4.10. The upscaling algorithm doesn't perform equally good (or bad) for each realization individually. The Cartesian grid restricts the flexibility of the upscaling with respect to features present in the geological model such as meandering (sinuous) channels, which are regionally and locally oblique to the principal grid orientation[7]. Depending on these features, the upscaling algorithm will perform differently. An accurate coarse representation will be more difficult to obtain for these oblique features compared to straight channels aligned with the principal orientation of the grid. Flow-based upscaling is much less sensitive to this than traditional averaging methods, but it is not immune. Numerically, with the TPFA upscaling and simulation scheme used in this work, it can be even better understood that upscaling of oblique features might lead to discontinuities in the coarse representation (corner point flux is not accounted for and therefore only face to face transmissibility affects the continuity of reservoir bodies), e.g. see chapter 20 of (LeVeque, 2002).

Besides the aforementioned issue, connectivity between injector(s) and producers differs a lot from one realization to another, due to the highly channelized reservoir models. Deviation from the fine-scale to coarse-scale behavior increases in reservoir models where the injector and producer are in direct communication. This acts as another restriction to simple grid-convergence studies performed on a single, or a small subset of "deterministic" models, which might not capture this, see figure 4.10.

Grid resolution could also be viewed of as an explanatory (independent) variable in determining the effect on the response (dependent) variable via Response Surface Methodology (RSM)[8]. However, in this work, the choice is made to study the dynamic spatial scale in the context of ensemble response uncertainty, i.e. how does the response uncertainty of the ensemble behave when coarsening and, therefore, would the decision making based on this uncertainty change in a real-life case. The advantage of this compared to simple grid-convergence study is that is considered the behavior of the uncertainty of the whole ensemble when coarsening, rather than determining for a single realization an allowable grid-resolution (given some set of constraints). One criteria in assessing the behavior of the response uncertainty across various scales is that the response uncertainty[9] shouldn't decrease solely by reducing the number of nodes (grid-cells). Coarsening without reduction of variance or variability in the solution is obviously impossible for any complex model. However allowable reduction in variability enables decision making on a similar uncertainty range rather than on an inherently smaller and bias uncertainty range due to averaging of parameters and solution (Omre, Lødøen, et al., 2004; O'Sullivan, Christie, et al., 2005).

Table 4.1 describes the parameters used in the reservoir simulation performed in ADGPRS. Other simulation parameters are the same across all ensembles, in particular: fluid model consists of a dead oil model, standard Brooks-Corey parameters, as well as standard tuning parameters for time-stepping and (non)-linear solver.

Since BHP is fixed and will be constant in the production wells throughout the simulation, the following one-dimensional[10] properties can be used to compare the uncertainty in flow response across various model scales: oil- and water production rate, (on a field and individual well scale) and water-cut (defined as the volumetric flow rate of water divided by the total volumetric flow rate, in this case, oil and water).

Oil- and water rate, as well as water-cut, might be favorable properties over cumulative production

---

[6] Typically a deterministic PDE is solved for M-"deterministic" reservoir models, where deterministic refers to them being equally probable, not to the actual method of acquiring the reservoir model.

[7] Regionally oblique refers to obliqueness arising from the main orientation of the channel belt/paleoflow. Locally oblique refers to obliqueness due to sinuosity of the individual channels inside the channel belt.

[8] Use a set of designed experiments to obtain the optimal response (surface), given a set of explanatory variables.

[9] Here referred to as the spread or variability in solution space, i.e. set of possible one-dimensional responses to the forward simulation of the ensemble in time (such as oil and water production).

[10] Referred to here as one-dimensional since the data comprises points in a two-dimensional space. The data is collected in time, however they describe very little on the reservoir behavior away from the well and therefore are referred to as one-dimensional data.

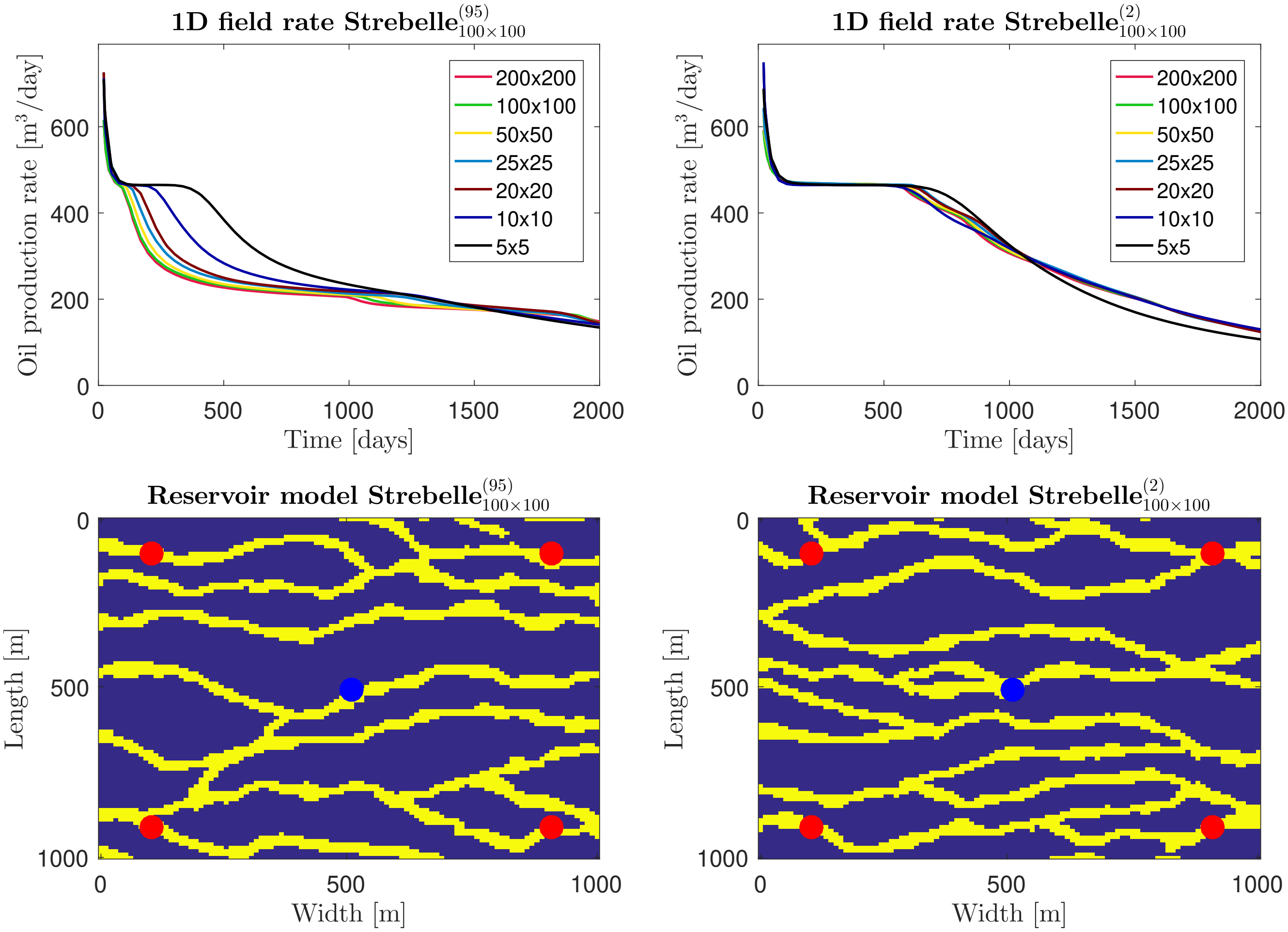


Figure 4.10: **Top row** displays one-dimensional field responses for two realizations from the Strebelle$_3$ ensemble. The difference in rate of deviation from the fine-scale is clearly visible. **Bottom row** shows the two reservoir models corresponding to the one-dimensional flow response. The left reservoir model shows slightly more features (channel branches and general sinuosity) oblique to the principal orientation of the grid as well as clear short-circuit between production (red) and injection (well), namely the bottom well. This is less pronounced in the right model, where the flow seems “easier” represented on a coarser scale (almost direct fit with only 10×10 nodes.

Table 4.1: Table of input parameters for reservoir simulation.

Ensemble	**Number of wells**	**Control**	**Value**	**Dimension Hf**	**Hf Grid size**
(name)	(inj, prod)	(inj, prod)	(inj, prod)	$(N_x \times N_y)$	$(\Delta x, \Delta y)$
Strebelle$_{1,2,3}$	(1, 4)	(rate, pressure)	(500[m^3/day], 100[bar])	(100×100)	(10, 10)
Flumy$_{1,2,3}$	(1, 4)	(rate, pressure)	(500[m^3/day], 100[bar])	(100×100)	(10, 10)
LineDrive$_{1,2,3}$	(4, 4)	(pressure, pressure)	(220[bar], 130[bar])	(120×120)	(20, 20)
Ganges	(4, 5)	(pressure, pressure)	(220[bar], 130[bar])	(240×240)	(20, 20)

for comparing flow responses. The reason for this is the smooth behavior of cumulative production to events such as water breakthrough (small change in its derivative) compared to the effect of this event on changes in rate. Also, it is expected that individual wells contain larger variability in their response than field data, which is merely a summation of all well production data. Therefore both data "scales" are considered in the analysis.

Further illustrating the problem of Uncertainty Quantification and estimation of a relevant spatial scale in reservoir simulation is the example in figure 4.11. The figure displays one-dimensional flow response obtained from a member of a different ensemble, namely Strebelle$_1$ ensemble (see chapter 2.2.2). In this example, as displayed in chapter 2.2.2, the paleoflow orientation is oblique to principal grid orientation. With the five-spot setup used in simulating the aforementioned ensemble, two production wells (well 1 and 4) are parallel to the main paleoflow orientation, i.e. generally in direct communication with the injector, while the other two wells (well 2 and 3) are perpendicular to the main paleoflow orientation. The key observation from this figure is the fact that the one-dimensional response of wells perpendicular to the paleoflow orientation, in this five-spot setup, show relative homogeneous production patterns while the reservoir clearly exhibits a highly heterogeneous facies distribution (and two-dimensional oil displacement). The result is that the misfit between the fine- and coarse scale one-dimensional responses for these wells are nearly negligible. The opposite is true for wells parallel to the paleoflow since they are in direct communication with the injector well and show a highly heterogeneous production pattern (e.g. "early" water breakthrough).

Multidimensional scaling, or MDS, is used in this work to simplify the representation of the production data in time without any considerable loss of information[11]. The spatial uncertainty in model parameters and resulting response uncertainty is represented by the spread in the points after 2D MDS projection. As shown in chapter 3.4.2, three curves in time can be easily characterized by three points on a two-dimensional plane (actually even one-dimensional plane would suffice, see figure 3.5). The spread in the projection of the points tells us much about the time-behavior and similarity of the curves. If their distance after 2D-projection using MDS is small, the curves are said to be similar in the original domain. This holds when almost all of the energy of the eigenvalues of $\boldsymbol{B}$ is in the first two eigenvalues since a two-dimensional projection is concerned. Figure 4.12 shows how the dynamic response (oil production rate, field scale) of the Strebelle$_3$ ensemble is represented as a 2D projection using classical MDS.

The orientation and position of ensemble members, represented by points after the MDS projection on a lower dimensional space, is determined by the dissimilarity matrix $\boldsymbol{D}$ up to a rotation (Borg & Groenen, 2005). This observation has led to an experiment where consecutively more elements[12] were included in the curves used in the computation of the dissimilarity matrix $\boldsymbol{D}$ such that the dissimilarity and resulting MDS is taken over growing intervals of time. This is illustrated in figure 4.13. Stacking of these projections, representing one-dimensional responses of the forward simulation up until that particular time, leads to M-"continuous" curves describing the behavior of the uncertainty in time, referred to as the "Uncertainty Trajectory". This is illustrated in figure 4.14.

Since this work attempts to clarify the evolution of the uncertainty when coarsening, two options of comparing the Uncertainty Trajectory between ensemble scales are considered. Figure 4.15 depicts one of the two possibilities of comparing multiple ensemble scales and their associated uncertainty, while figure 4.16 depicts the second option. The first option is two compute the dissimilarities solely between ensemble members of the same scale, resulting in the uncertainty of the ensemble at that particular scale. This can then be done with all ensemble scales and the resulting N_c[13] transformations can then be analyzed in a single Uncertainty Space. The other approach is to compute the dissimilarity between each ensemble member at every ensemble scale, resulting in one large dissimilarity matrix (for each time interval) as

---

[11] Energy of the first two eigenvalues is typically above 97% of the total energy of all eigenvalues when one-dimensional production data curves are used for dissimilarity matrix $\boldsymbol{D}$.

[12] Where elements refer a vector element or index, such that more elements means a larger vector.

[13] Where N_c is the number of ensemble scales in the hierarchy, from finest ensemble scale denoted $N_c = 1$ to the coarsest ensemble scale denoted $N_c = C$

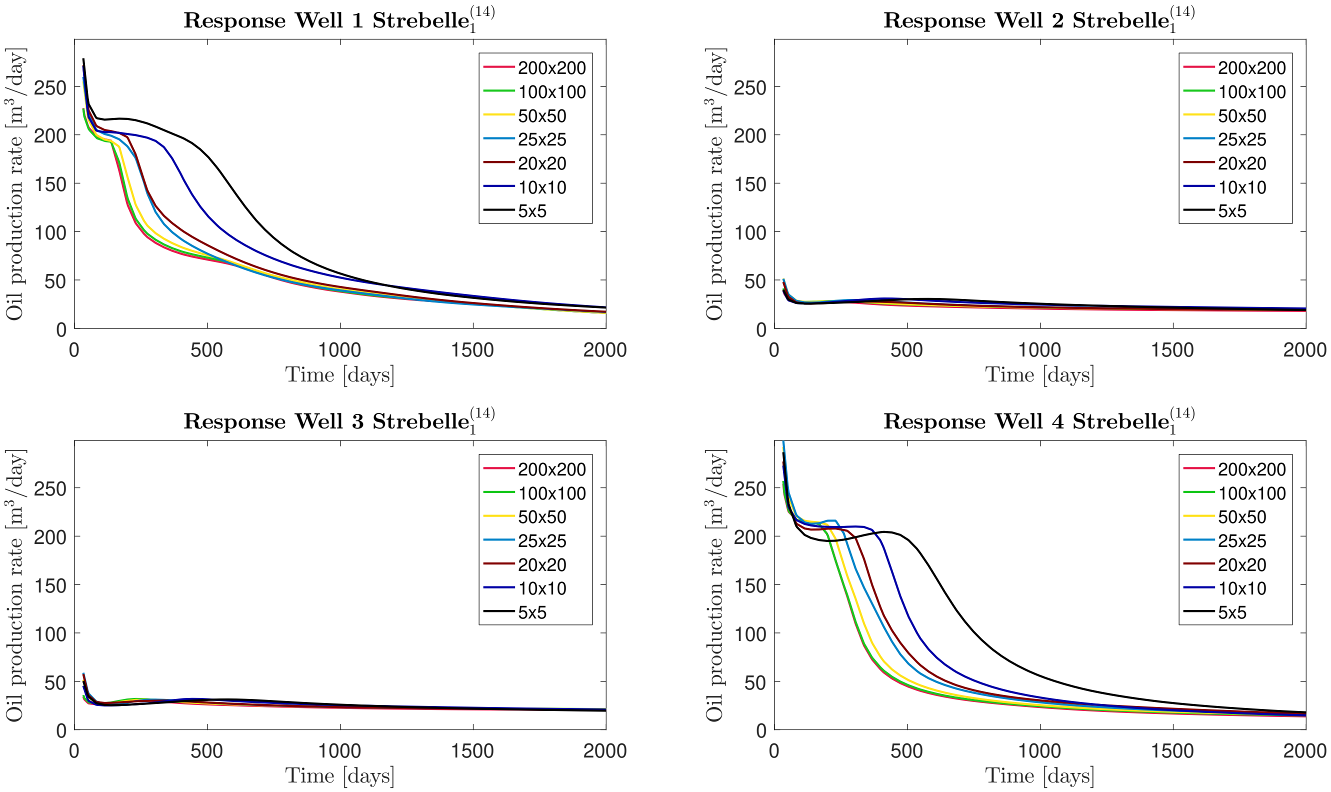


Figure 4.11: **Top left** and **bottom right**, i.e. well 1 and 4, image displays the oil production rate of wells that are parallel to the paleoflow orientation and therefore in direct communication with the injector well. Large initial production which decreases rapidly after breakthrough. Also clearly visible is the rapidly increasing misfit between the fine- and coarser scales. **Top right** and **bottom left** show two wells that are orthogonal to the main paleoflow orientation, display a homogeneous (continuous) oil production rate. The well behavior is fully explained by the most coarse representation due to the homogeneous nature of the production. Note that the reservoir is clearly heterogeneous, however the one-dimensional response of well 2 and 3 are not able to capture this behavior away from the well location.

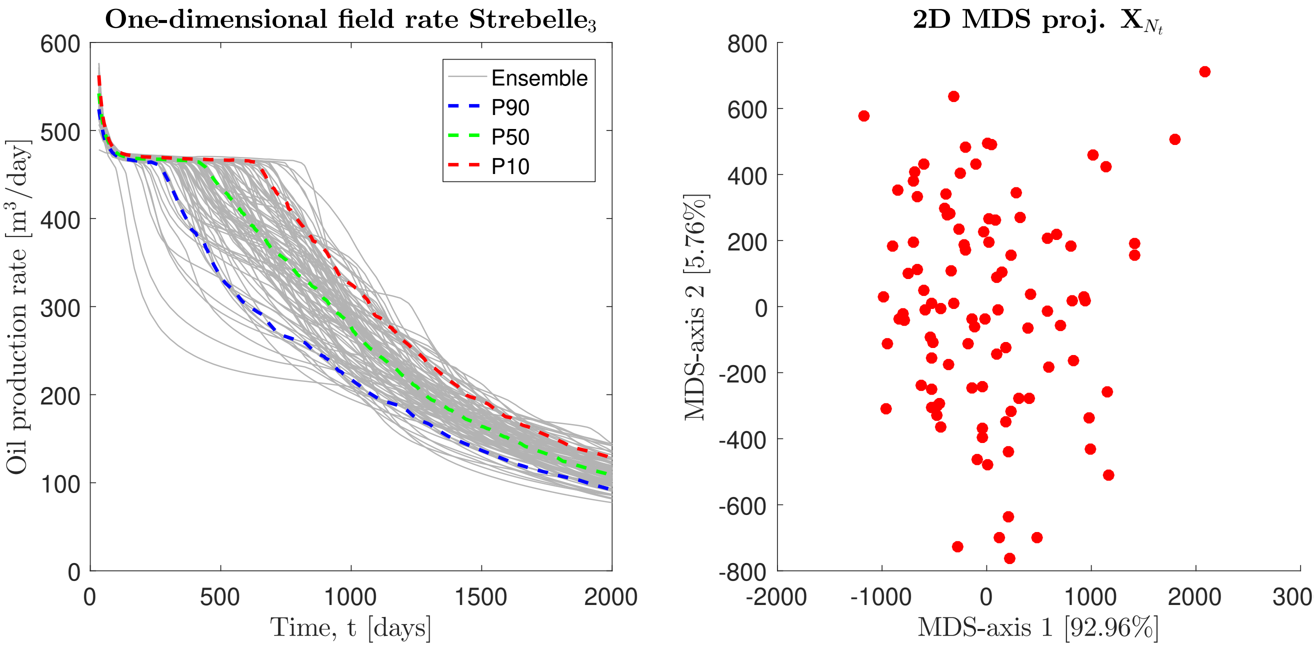


Figure 4.12: **Left** image shows the one-dimensional oil field rate for the Strebelle$_3$ ensemble. **Right** figure depicts the two-dimensional MDS projection of this ensemble response, computed over $t \in [0, t_{N_t}] = [0, t_{end}]$.

opposed to N_c dissimilarity matrices (for each time interval) in the first case.

Both methods of uncertainty quantification offer certain advantages. The first method can serve as a way to estimate how similar the behavior of the uncertainty is across several ensemble scales. The magnitude of the separation or distance between finest ensemble Uncertainty Trajectory and coarser ensemble scales is a measure of how similar realizations are behaving (in time and when coarsening). Note that this doesn't mean that there exists no error between the fine- and coarse scale solutions to the flow problem, it merely means that the coarse ensemble uncertainty has similar characteristics as the respective fine-scale ensemble uncertainty. These characteristics can then be exploited in the form of finding similar flow responses and appointing representative realizations for the said groups of flow responses, using a coarser scale flow simulations as a measure of distances between ensemble members. On top of that, clustering algorithms such as K-medoids are invariant to translations and orthogonal transformations of the data points (Kaufman & Rousseeuw, 1987). Even under the aforementioned linear transformations, the model selection algorithm based on coarse flow response distances can be quite effective in selecting representative realizations if the Uncertainty Trajectory of the particular coarse scale ensemble follows the Uncertainty Trajectory of the fine-scale ensemble closely.

The second method allows for a more direct measure of how similar the uncertainty is across ensemble scales while considering the intrinsic uncertainty within each ensemble scale simultaneously. Note that uncertainty refers here to the spread or variability in flow responses of each ensemble. If after MDS projection of the joint distance matrix, containing the distance between each ensemble member of every ensemble scale, ensemble members of the same scale are projected closer to each other than to members of different ensemble scales, there most likely exists a large upscaling bias. This can be understood considering a water injection situation as is modeled in this work, where (hypothetically) the coarse ensemble consistently predicts much later water breakthrough. This creates a bias in the coarse solutions and causes coarse solutions to be more similar to themselves as opposed to the fine-scale solutions. An attempt to depict this can be found in figure 4.21.

Using the first approach, shown in figure 4.14 and 4.15, the Uncertainty Trajectory is computed for each ensemble scale using the several properties available (as mentioned above). The deviation in Uncertainty Trajectory becomes evident when coarsening ratio increases. This is observed for each property, at the

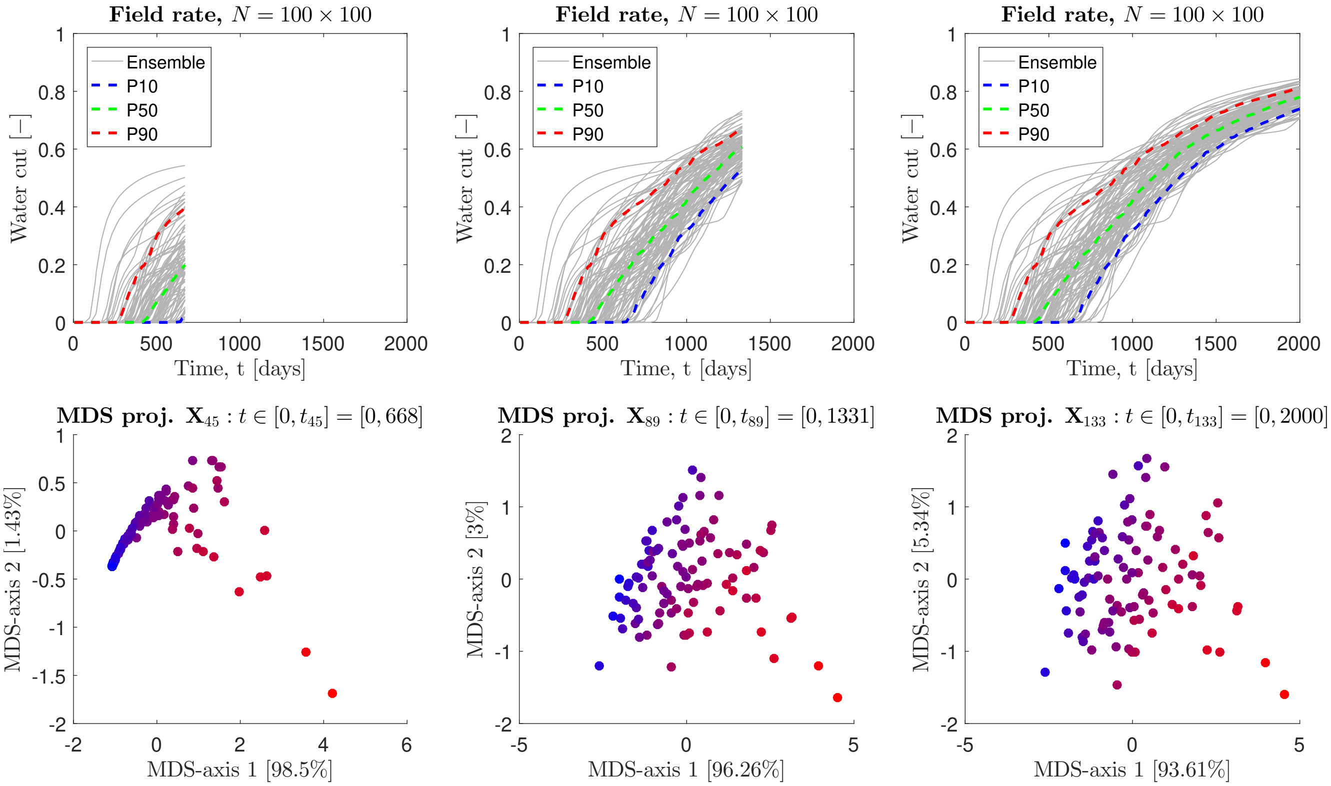


Figure 4.13: **Top row** displays the response of the Strebelle$_3$ ensemble in terms of field water cut after different times of production. **Bottom row** depicts the two-dimensional projection of the MDS, included different times of production. The projection clearly depends on the interval chosen for calculating the distance matrix $\boldsymbol{D}$. Since one point in the 2D-projection represents one realization, they can be easily color coded relative to the property used for computing the initial distance matrix $\boldsymbol{D}$. In this case the color code represents breakthrough time of water, with red being fast and blue being slow. Note that with increasing variation between the curves, due to increasing the time interval over which the distance is computed, the energy of the second eigenvalue in the projection becomes larger, denoted in % on the x- and y-axis of the plot.

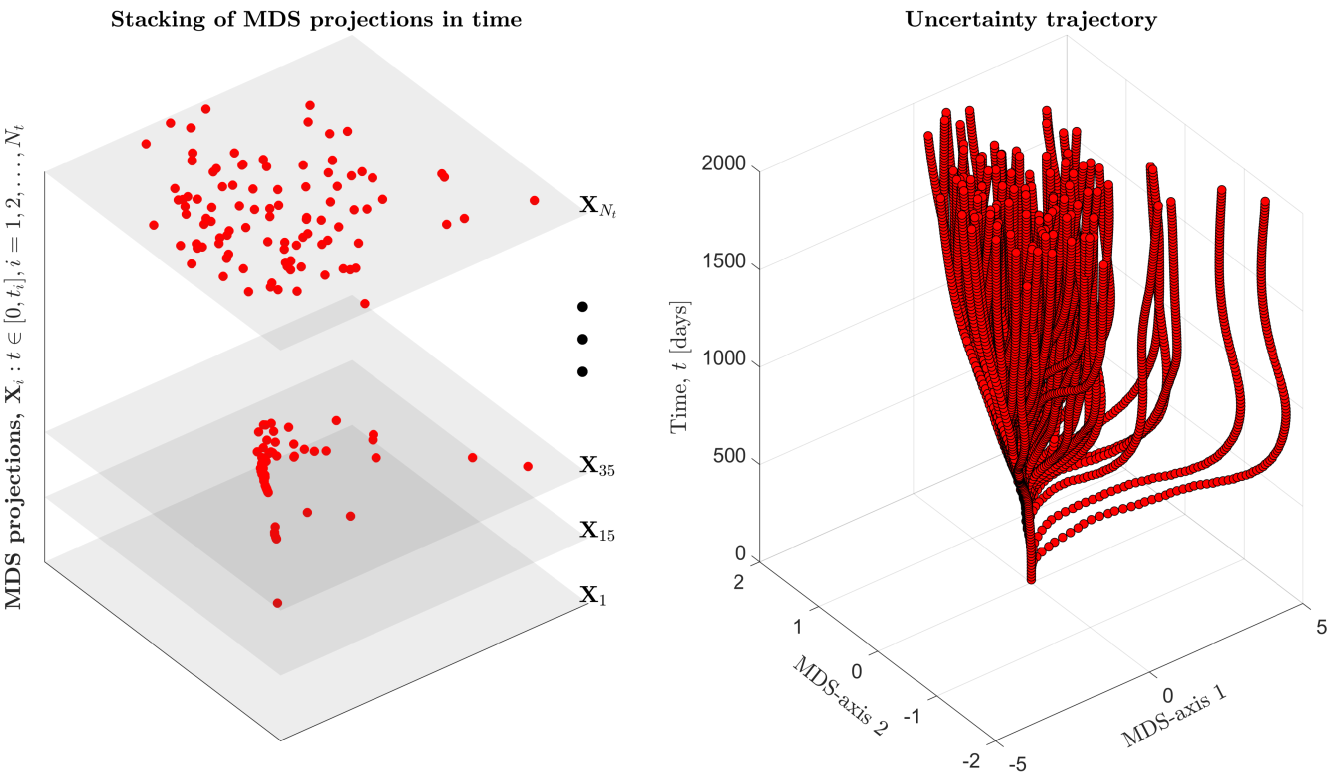


Figure 4.14: **Left** schematic depicts the MDS projections, $\boldsymbol{X}_i$ such that $t \in [0,t_i]$ and $i = 1,2,\ldots,N_t$. **Right** image displays the full spectrum of projections, referred to as the Uncertainty Trajectory. The procedure to obtain the continuous Uncertainty Trajectory is not trivial, but a representative display can be obtained following the workflow proposed in Appendix A.

well and field scale. Figure 4.17 illustrates the deviation from the fine-scale Uncertainty Trajectory for two realizations. Note that two realizations are chosen since visually comparing the 100-trajectories for seven ensemble scales isn't very informative. A more quantitative comparison, however, is shown in figure 4.18 where the integrated distance for each trajectory with respect to the finest scale is displayed.

Another way of examining the coarsening effect in terms of Uncertainty Trajectory is considering the first approach, but computing an Orthogonal Procrustes (OP) problem at each time-slice w.r.t. the finest Uncertainty Trajectory. An OP can be defined as an orthogonal matrix $\boldsymbol{R}$ which is the closest map between two matrices (Schönemann, 1966) (Zhang, 2000). The problem of finding the orthogonal matrix $\boldsymbol{R}$ that minimizes the difference between two matrices say $\boldsymbol{A}$ and $\boldsymbol{B}$, can be expressed through the Singular Value Decomposition (SVD), a generalization of the Eigenvalue Decomposition explained in chapter 3.4.2.

$$\boldsymbol{Y} = \boldsymbol{B}\boldsymbol{A}^T = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T \tag{4.10}$$

where $\boldsymbol{U}$ is the orthogonal matrix of left-singular vectors (as its columns), $\boldsymbol{\Sigma}$ is diagonal matrix containing the singular values and $\boldsymbol{V}^T$ is the orthogonal matrix of right-singular vectors (as its rows). Then, $\boldsymbol{R}$ can be found by computing

$$\boldsymbol{R} = \boldsymbol{U}\boldsymbol{V}^T \tag{4.11}$$

Matrix $\boldsymbol{B}$ in our case is the coordinate matrix obtained from the classical MDS using the fine-scale ensemble, while matrix $\boldsymbol{A}$ is the coordinate matrix obtained from the classical MDS using the coarse-scale ensembles. Since an orthogonal transformation doesn't affect the clustering (Kaufman & Rousseeuw, 1987), aligning the "data clouds" at each time-slice with the fine-scale projection provides insight in the accuracy of model selection on coarse distances. If the Uncertainty Trajectory deviates a lot, which happens especially at early distances, the coarse scale distance cannot be used for model selection. Figure 4.19 depicts the outcome of this procedure for the cumulative oil production (as in the top right of figure 4.17). The deviation is heavily dependent on the time interval used in computing the MDS projection. Near the end of the simulation time, all model scales tend to converge to the fine-scale orientation (except the 5×5 ensemble scale. Note that no scaling nor translation was required to obtain this fit, merely an MDS projection for each interval which afterward is transformed (rotated) using an orthogonal transformation (i.e. linear transformation that preserves the dot product).

The integrated distance for this modified Uncertainty Trajectory, plotted on a semi-log scale is shown in figure 4.20. This plot suggests a dynamic relevant spatial scale similar to the scale predicted on the prior using DCT. This doesn't seem to happen consistently throughout the other ensembles, which makes it not possible to confirm a clear correlation, via DCT, between the static and dynamic characteristic scale.

The mean integrated distance to the fine-scale Uncertainty Trajectory displays very similar behavior using the second approach. The main difference between the two approaches can be visualized in the following way. Consider figure 4.21 where the left plot depicts the last time-slice (namely $t \in [0, t_{end}]$) of the separate MDS projections of each ensemble scale (method 1) and the right plot shows the last time-slice of the second method where the MDS projection depends on the response of all ensemble scales and members simultaneously. The color in figure 4.21 indicates a particular model scale whereas the symbol indicates a particular realization. The most important difference between the two methods is the absence of the coarsest ensemble members on the left side of the plot associated with method 2. This can be explained by considering the fact that the MDS projection of method 2 (see figure 4.16) contains the flow response of all ensemble members of the same scale as well as across other scales in the distance computation. This means that consequently, the coarsest scale ensemble response is more similar to itself than to the finest scale response, due to an upscaling bias (consistent overestimation of the cumulative oil produced). More ensembles and their Uncertainty Trajectories are shown in the Appendix, namely Appendix B.

Sensitivity to sub-heterogeneity[14] was also investigated. The Uncertainty Trajectory is relatively insensitive to these high-frequency fluctuations of the parameters since flow in these models is mainly governed by the facies distribution, especially on a statistical (uncertainty) level. The reason for this is most likely due to the large contrast in permeability between the reservoir and non-reservoir facies, rendering the high-frequency fluctuations less impact-full. This doesn't necessarily imply flow with and without these sub-heterogeneities being similar at each model scale. Figure 4.22 displays the flow response for the same facies distribution with- and without sub-heterogeneity. The sub-heterogeneity was generated using a smoothly varying Gaussian random field, resulting in a bi-modal distribution of the permeability with the same mean as the homogeneous (in terms of permeability distribution inside facies) model, as also shown in figure 4.22. The absolute difference (denoted "misfit") between the two saturation fields displays that the difference is rather small. This is most likely caused by the fact that most of the flow (and transport) occurs through the highly permeable reservoir facies (due to the large contrast in permeability between the reservoir and non-reservoir facies).

---

[14] Where sub-heterogeneity refers to a heterogeneous representation of the properties inside a particular facies, as discussed in chapter 2.2.4.

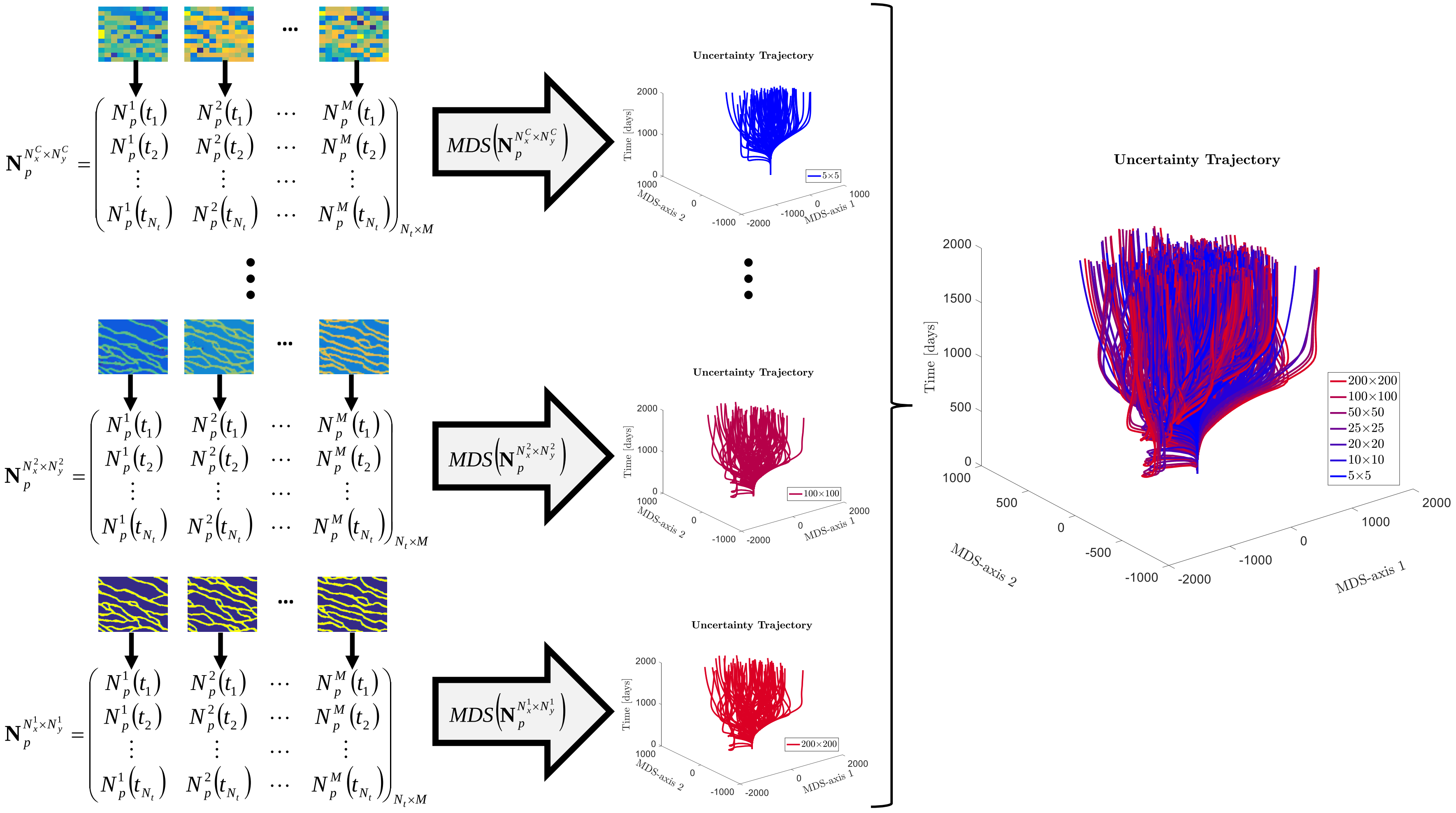


Figure 4.15: Schematic indicating one way of comparing the Uncertainty Trajectory for several ensemble scales, namely by computing the Uncertainty Trajectory separately for each ensemble scale.

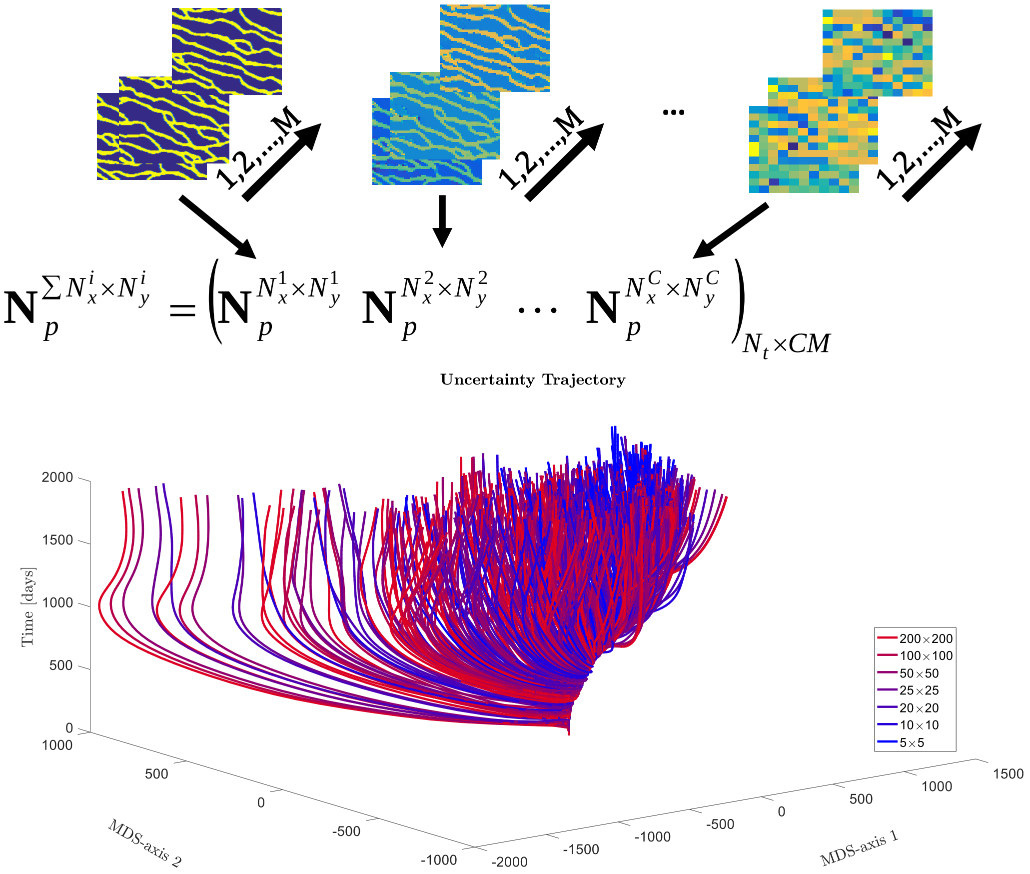
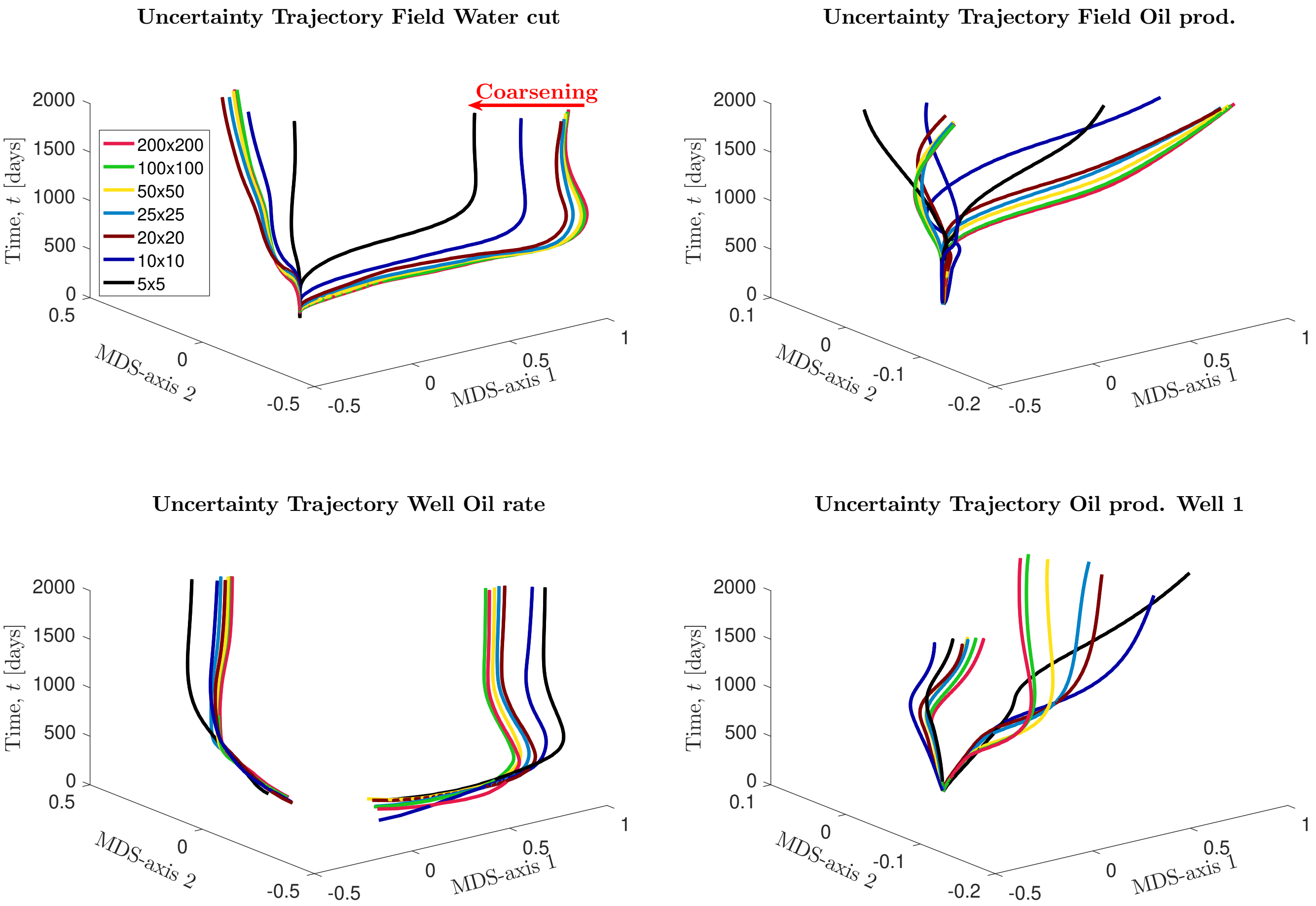


Figure 4.16: Schematic indicating the second way of comparing the Uncertainty Trajectory for several ensemble scales, namely by computing the Uncertainty Trajectory separately for each ensemble scale.


Uncertainty Trajectory Field Water cut

Uncertainty Trajectory Field Oil prod.

Uncertainty Trajectory Well Oil rate

Uncertainty Trajectory Oil prod. Well 1

Figure 4.17: **Top left** image depicts the Uncertainty Trajectory for two realizations using the Field water cut as a distance for the MDS projections. Clearly the deviation from the finest trajectory (200 × 200 model scale) increases when coarsening. The same behavior can be seen, in a different extend for the other properties.

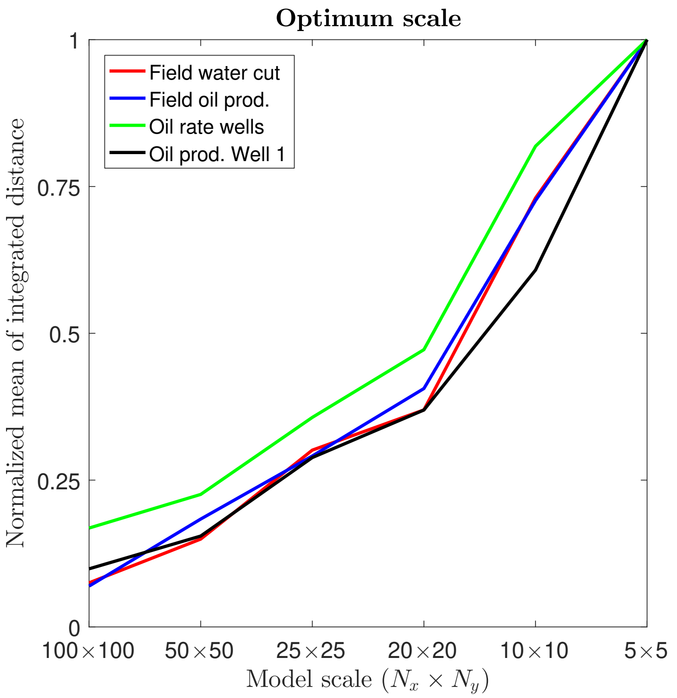


Figure 4.18: Depicting the mean of the integrated distance between the finest ensemble scale and coarser scales respectively. The distance in Uncertainty Trajectory increases when coarsening, and seems to show consistent behavior across several properties.

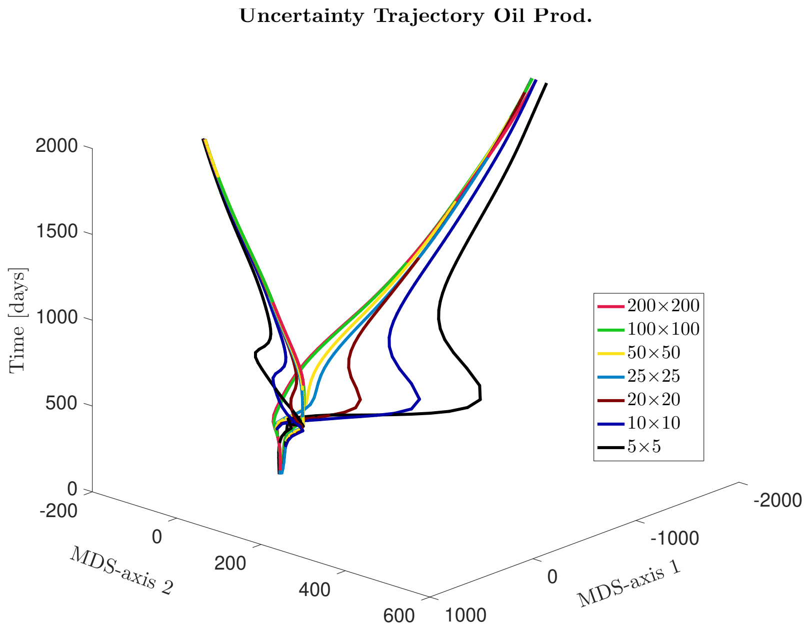


Figure 4.19: Displaying the Uncertainty Trajectory of two realizations from the Strebelle$_3$ ensemble for Cumulative Oil Production after performing an Orthogonal Procrustes problem w.r.t. to the finest scale Uncertainty Trajectory at each time-slice.

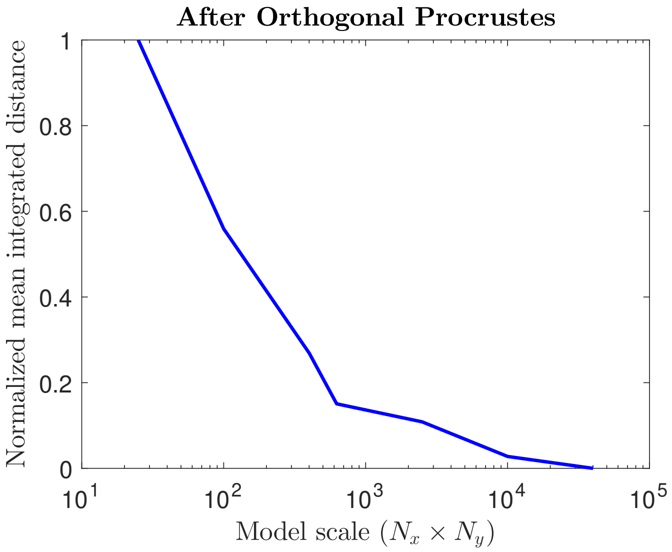


Figure 4.20: Depicting the normalized mean of the integrated distance w.r.t. the finest scale Uncertainty Trajectory (as shown in figure 4.19). The integrated distance is similar to the ones displayed in the top right of figure 4.17, however now an Orthogonal Procrustes problem is solved w.r.t. the fine-scale ensemble at each time-slice.

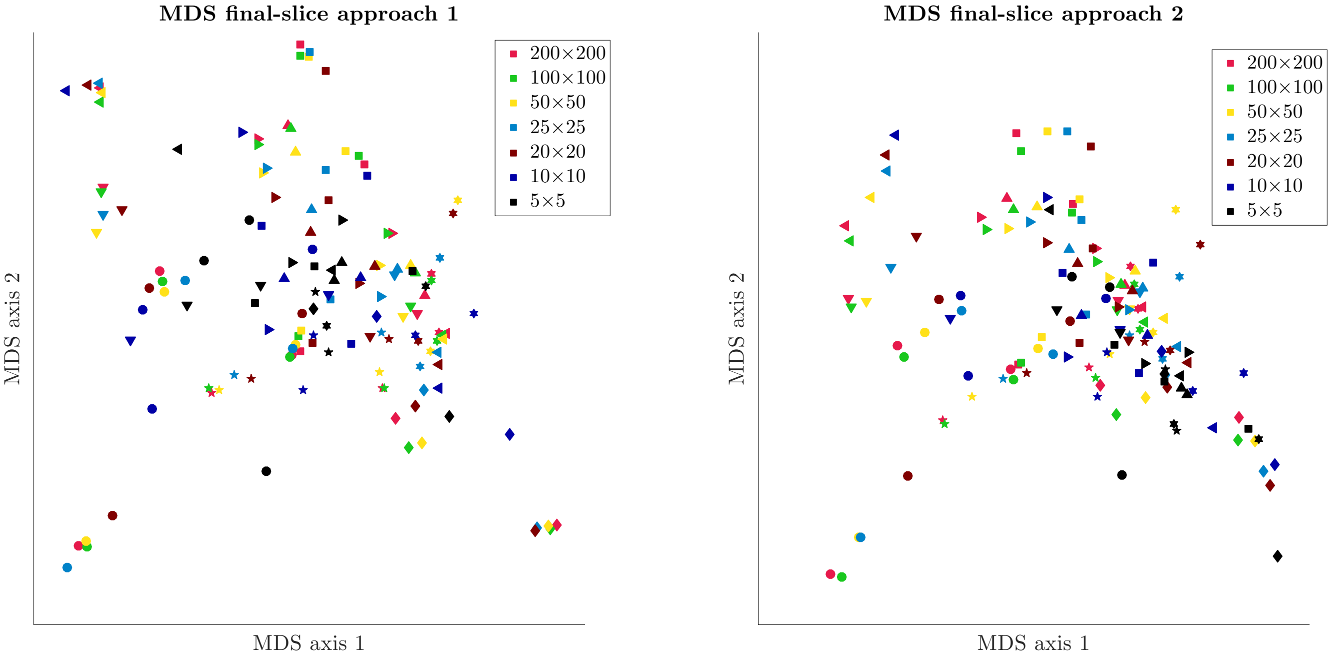


Figure 4.21: **Left** image depicts the final-slice of the MDS projection, namely such that $t \in [0, t_{end}]$, using method 1, i.e. independently projecting the ensemble scale distances. Different realizations are depicted with different symbols, while the color indicates the ensemble scale to which that particular realization (symbol) belongs. Clearly grouping or realizations can be observed with ensemble scale 20×20 typically being the last scale which can clearly be grouped together. **Right** image shows the final slice of the MDS projection using method 2, i.e. considering all the distances across all scales in one single projection (per time-slice). The key observation in this plot is the absence of the coarsest ensemble on the left side of the plot, indicating that the flow response of the coarsest ensemble is more similar to itself than to the same realization at different scales.

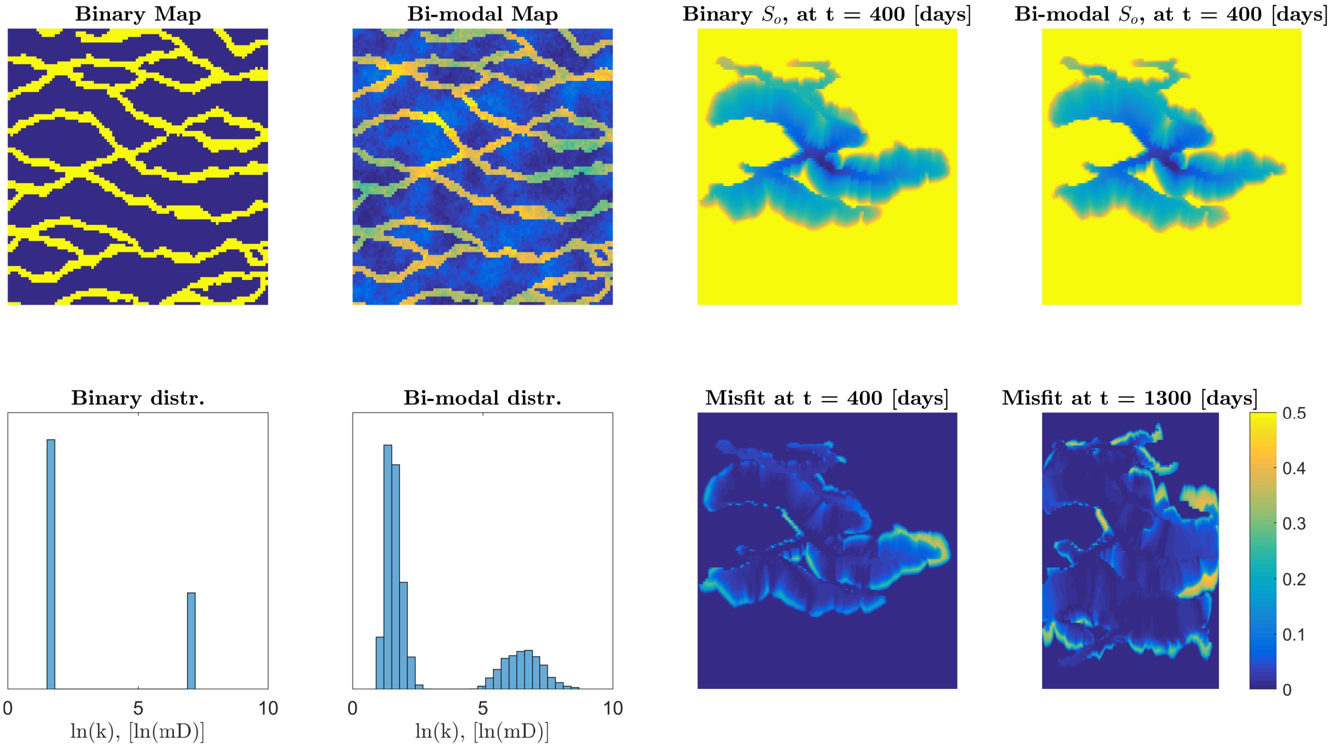


Figure 4.22: **Top left** depicts the two permeability fields of a binary (referring two homogeneous distribution in the facies) and bimodal distribution, see chapter 2.2.4. **Bottom left** shows the actual distribution and the fact that they share the same mean (so that they are able to be compared in a general sense). **Top right** displays the associated saturation fields of the two models, where blue depicts water and yellow the oil phase. **Bottom right** depicts the absolute difference between the two saturation fields at two particular times.

5

Predicting fine-scale response using coarse scale distances

This chapter is devoted to the prediction or selection of a subset of representative models which converge to the full statistics of the ensemble. This subset contains a much lower amount of simulations on the fine-scale which reduces the computational time while still offering the same quantiles used in the decision making models[1]. The main driving force behind using coarse-scale distances for computing the MDS projection was the behavior of the coarser uncertainty trajectories. If the coarser trajectory only slightly deviates from the fine-scale trajectory, the resulting "coarse"[2] MDS projection might hold a large predictive value on the response uncertainty behavior of the fine-scale. This hypothesis led to the idea of using coarse-scale ensemble responses in the computation of the distance matrix of squared distances $\boldsymbol{D}^{(2)}$, consecutively used in the MDS projection on which clustering is performed. This clustering, based on coarse distances, ultimately leads to a model selection algorithm where the selected models were simulated on the fine-scale and showed to statistically represent the total variation of the full fine-scale ensemble response. The developed methodology is quite similar to the previously presented methods by (Scheidt, Caers, Chen, & Durlofsky, 2011). This can be understood since the main methodology of the previous chapter was heavily influenced by the work of Caers et al., (2010) and Caers et al., (2009) where a proxy response like Time of Flight (TOF) or results from a streamline simulation were used to compute $\boldsymbol{D}^{(2)}$.

5.1 Quality of the model selection

The reduced representation of the ensemble response, via MDS projection on a lower two-dimensional space (plane), displays the coarse ensemble response in time as a set of points. The distance between each point in this new space is solely determined by the MDS projection[3], where each point represents the dynamic response of one member of the coarse ensemble. A clustering on this set of points is done using a k-medoids method[4] and the resulting medoid of each cluster is thus assumed to represent a particular type/class of behavior from the full ensemble. Simulating the resulting subset of the full fine-scale ensemble, namely the ensemble members which coincide with the medoids[5], it is believed that statistics, derived from this subset, can converge to the statistics of the full ensemble. This convergence depends on many factors. First of all the amount of clusters. Since for each cluster a representative is chosen, an increasing number of clusters will most likely result in a decrease in the misfit between the statistics of the subset and the statistics of the full fine-scale ensemble. However, an increase in clusters also means an increase in the

---

[1] Assuming that financial decision associated with a Field Development Plan is mainly based on the quantile response of the ensemble, i.e. the P10, P50 and P90.

[2] Coarse refers to the distance used to compute the MDS projection, namely the full response of a coarse ensemble.

[3] Up to a rotation, since Euclidean distances and a classical MDS are used (Borg & Groenen, 2005), chapter 12 on "Classical Scaling", p.262.

[4] Using the "Statistics and Machine Learning Toolbox in MATLAB©" (MathWorks, 2017). See also chapter 3.4.

[5] In clustering with k-medoids, the center (referred to as a medoid) of each cluster is part of the data, therefore a data point itself.

computational effort since every medoid has to be simulated on the fine-scale. In the limit, where a number of clusters are equal to a number of ensemble members, the subset becomes equal to the ensemble and so do the statistics. An optimum has to be found where enough clusters are chosen to ensure convergence to the full ensemble statistics but constrained by computational costs of adding too many clusters. The convergence is not only a function of the number of clusters but also a function of the "quality" of the model selection which in turn is heavily influenced by the following three things:

1. Amount of correlation (positive or negative), contained in the property used for computing the distances $\boldsymbol{D}^{(2)}$, between the fine- and respective coarse-scale.
2. Amount of energy explained by the first two eigenvalues of the MDS projection.
3. Time interval used in the computation of the distances $\boldsymbol{D}^{(2)}$.

The first point can be understood intuitively through the definition of clustering. Clustering is based on grouping objects (or data) such that objects in the resulting group are more "similar" to other objects in that group than to objects outside of the particular group. Since a representative of a cluster is chosen for fine-scale simulation, the resulting response of that simulation must have a large correlation with the property used for computing the dissimilarities in the MDS projection. For example, if cumulative oil production is used as a property to compute the matrix of squared distances $\boldsymbol{D}^{(2)}$ which consequently used in computing the MDS projection, the resulting configuration of points can be effectively clustered such that members of the same cluster describe similar behavior in the cumulative oil production. Combining all the representatives (medoids) will result in a good approximation of the overall behavior in cumulative oil production of the full ensemble. It doesn't mean, however, that the combination of these medoids will approximate other properties equally well. It will most likely not work if there is no (strong) correlation between the property used for dissimilarity and the property it tries to approximate. This is shown in figure 5.1 where the property used for $\boldsymbol{D}^{(2)}$ is field cumulative oil production, which clearly separates this property in an ordered manner, but doesn't manage to separate the water rate of well 1. The medoids for the three clusters do show an approximation of the averaged behavior, however, the method will become less robust in this prediction when clustering of similar realizations is less effective. This problem can be solved easily by computing another MDS projection using the coarse ensemble water rate of well 1 for $\boldsymbol{D}^{(2)}$ on which another clustering is done and finally simulate the intersection of the two subsets (set 1 containing representatives for field oil production, set 2 containing representatives for water rate of well 1) on the fine-scale as approximation to the full fine-scale ensemble response. Another option is to combine several properties for the computation of $\boldsymbol{D}^{(2)}$, such as oil and water production and perform a joint-estimation, similar to the joint-estimation using a proxy response in (Scheidt & Caers, 2009).

The second point is also necessary since performing a clustering on the 2D-projection of the ensemble response will only result in clusters where members of the clusters are similar to each other when the 2D-projection accurately describes the dissimilarities between the responses. This is generally achieved when at least a certain percentage of the total energy is contained in the first two eigenvalues of the projection, which is very likely to happen considering these one-dimensional responses. When dissimilarities between two-dimensional responses are concerned, for example when performing clustering on saturation maps or velocity fields, extra dimensions might be required to accurately represent the dissimilarities of the full ensemble in the lower dimensional space. Note that for clustering purposes, adding another dimension to better approximate the original matrix $\boldsymbol{D}^{(2)}$ is not a problem, only visually it cannot be represented when more than three dimensions are required.

The third point is motivated by the erratic behavior of some of the coarse trajectories in the first half of the simulation time, e.g. seen in figure 4.19.

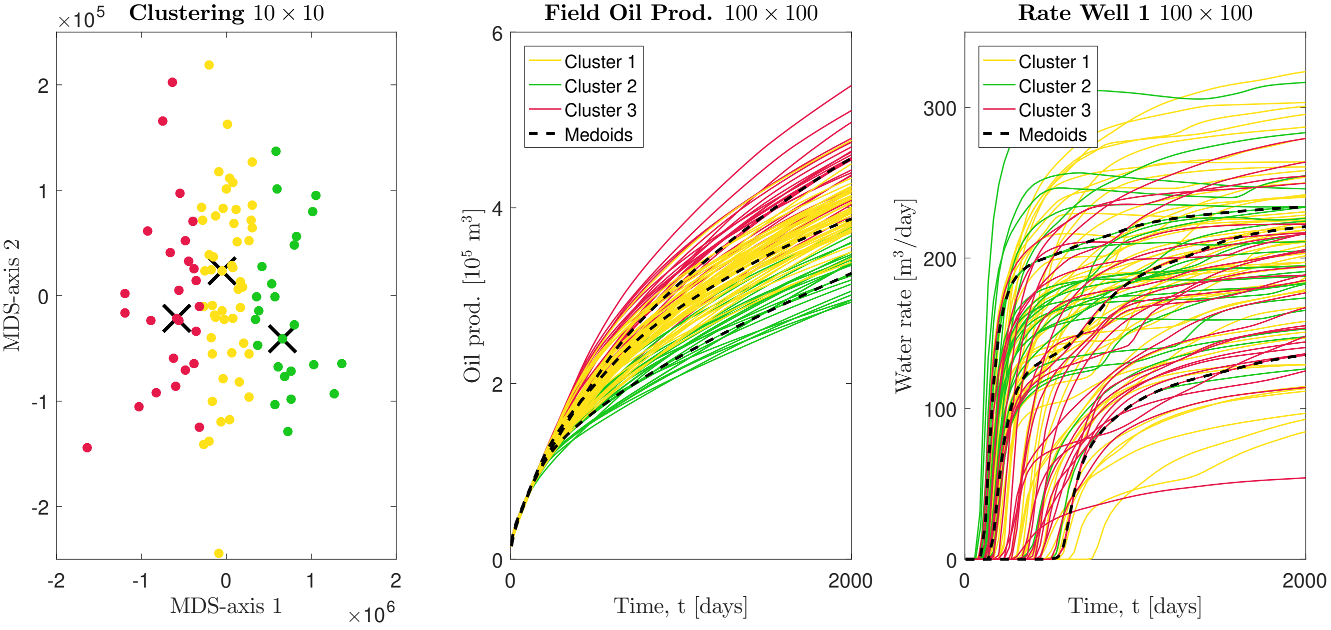


Figure 5.1: **Left** figure depicts clustering of the MDS projection, where $\boldsymbol{D}^{(2)}$ is computed using the field oil production of the coarse ensemble (number of nodes is 10×10). Note that the MDS-axis 1 is roughly one order of magnitude larger than MDS-axis 2, causing the nearly vertical separation in the clustering. The number of clusters is chosen to be three for this example. The **middle** graph shows that the clustering is very effective for identifying production classes for the field oil production of the fine-scale ensemble. The **right** figure displays a less effective prediction of similar behaving realizations when the water rate of well 1 is concerned. Note that even though the clustering for this property using field oil production for $\boldsymbol{D}^{(2)}$ is less effective in predicting similar responses for the water rate of well 1, the medoids still manage to approximate the average behavior of the water rate of well 1. However, it can be understood that if the clustering is less effective in grouping similar realizations, the method becomes less robust in predicting the average behavior. Also note that if coarse ensemble water rate of well 1 was used in computing $\boldsymbol{D}^{(2)}$, the clustering on that MDS projection would be again very effective in predicting similar responses for the fine-scale ensemble behavior of the water rate of well 1.

5.2 Use of Feature Space (Kernel Trick)

This subsection mentions briefly the possibility of using the Kernel Trick to improve clustering. Figure 5.2 shows the same clustering as the clustering in figure 5.1, but now plotted with equally scaled axes. The "vertical" separation in figure 5.1 is merely an artifact of the axes not being equally scaled. The other two images in figure 5.2 are MDS projections after using the kernel trick (see chapter 3.4.4), based on (Caers et al., 2010) chapter 3.3. The color code corresponds to the same clustering as the original clusters. The problem with the MDS projection after mapping to the feature space, where "new" dissimilarities are computed resulting in the "new" MDS projection, is that it is not always clear where to locate the boundaries of each cluster. In the original MDS space, the red and green represent very different production behavior and are clearly separated, whereas the middle image in figure 5.2 using a Gaussian Radial Basis Function (RBF) with $\sigma = 0.1$, shows green and red points very close together. This adds a difficulty of selecting the proper kernel function as well as having to define the parameter of the particular kernel function such that realizations with dissimilar behavior are actually at a great distance after MDS projection with kernel such that they are not "grouped' in the same cluster.

Usually the kernel trick is used to improve linear separation of the data by a hyperplane in a much higher dimensional space (called the feature space), which in this case might not be necessary since linear separation of different responses (see gradual change in color in top left image in figure 5.4) already exists. The right image in figure 5.2 will probably result in the same clustering as the original clustering without a kernel. Note that when there is no such linear separation as seen in the example above, where the property used in computing the distance is clearly linearly separable, using a kernel might be effective or even necessary to obtain an effective clustering. This could happen when joint estimations are concerned where the distance for the MDS projection is based on multiple properties, oil and water production simultaneously, see (Scheidt & Caers, 2009) for example. Also, in this paper, they do not cluster on the MDS projection of the dissimilarities in the feature space, but use the properties of the kernel function and the information of inner-products to expand the Euclidean distance between each element in Feature space and therefore are able to cluster directly in the feature space. This seems to be the best way to proceed, but the objective of this thesis is not to evaluate the most optimal clustering technique. Here we want to illustrate that information obtained from coarse scale simulations can greatly reduce the computational time while converging to the same statistics.

5.3 Correlation properties

As illustrated above, it is important to understand the correlation between the property used for computing $\boldsymbol{D}^{(2)}$ and the property tried to approximate by simulation of the medoids. Therefore, for several properties, the correlation is calculated. For properties on the same scale (e.g. water rate and oil production for the same ensemble scale) as well as cross-scale correlation (e.g. water rate on the 200×200 ensemble and water rate on the 10×10 ensemble scale), the figure 5.3 illustrates that there tends to exist a large correlation between each property when coarsening is applied. This suggests that the MDS projection, using a property with a large correlation between scales, can be effectively employed for model selection by k-medoids clustering.

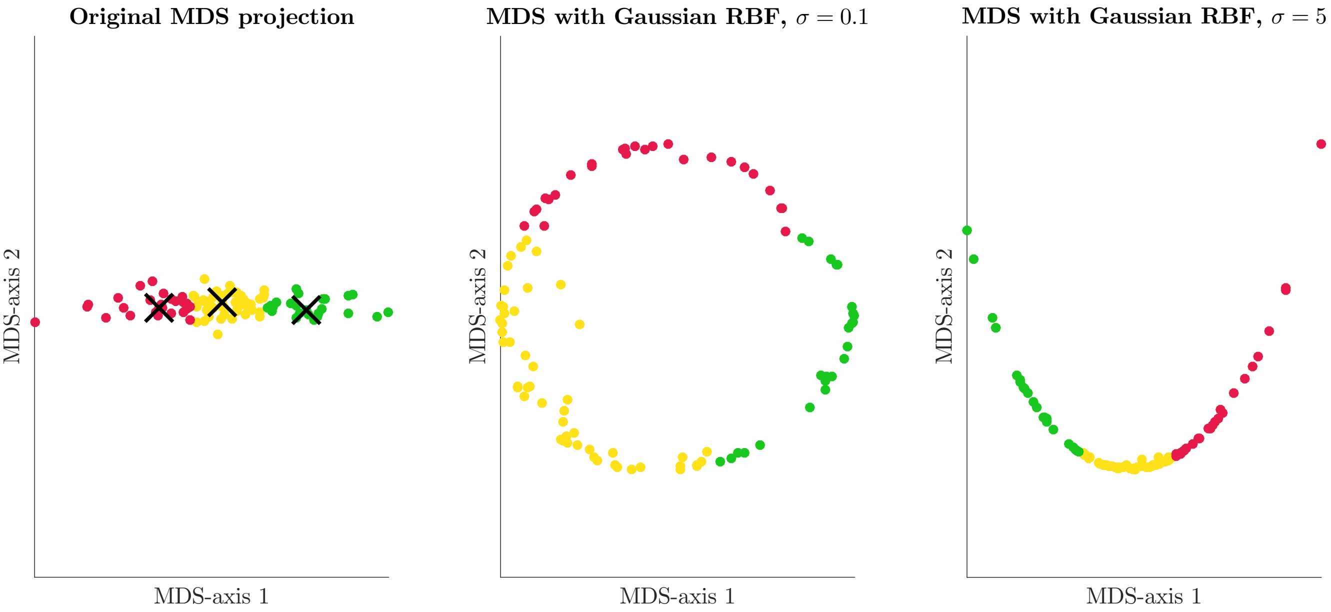


Figure 5.2: **Left** image depicts the same clustering as in figure 5.1 but now plotted with equally scaled axis. This illustrates the vertical clustering in figure 5.1, which is merely an artifact of the axes not being equally scaled. **Middle** figure illustrates a MDS projection using the kernel trick (see chapter 3.4.4) with a Gaussian Radial Basis Function (RBF) as kernel function, for $\sigma = 0.1$. The color code corresponds to the same clustering on the original MDS projection. **Right** image shows another MDS projection with kernel trick using the same Gaussian RBF but with $\sigma = 5$. The color code represents the same clustering as the original clustering. Several authors, such as (Scheidt & Caers, 2009), advise to use a bandwidth of 20% for the σ parameter in the kernel function (which is close to the $\sigma = 5$ used in the **right** image).

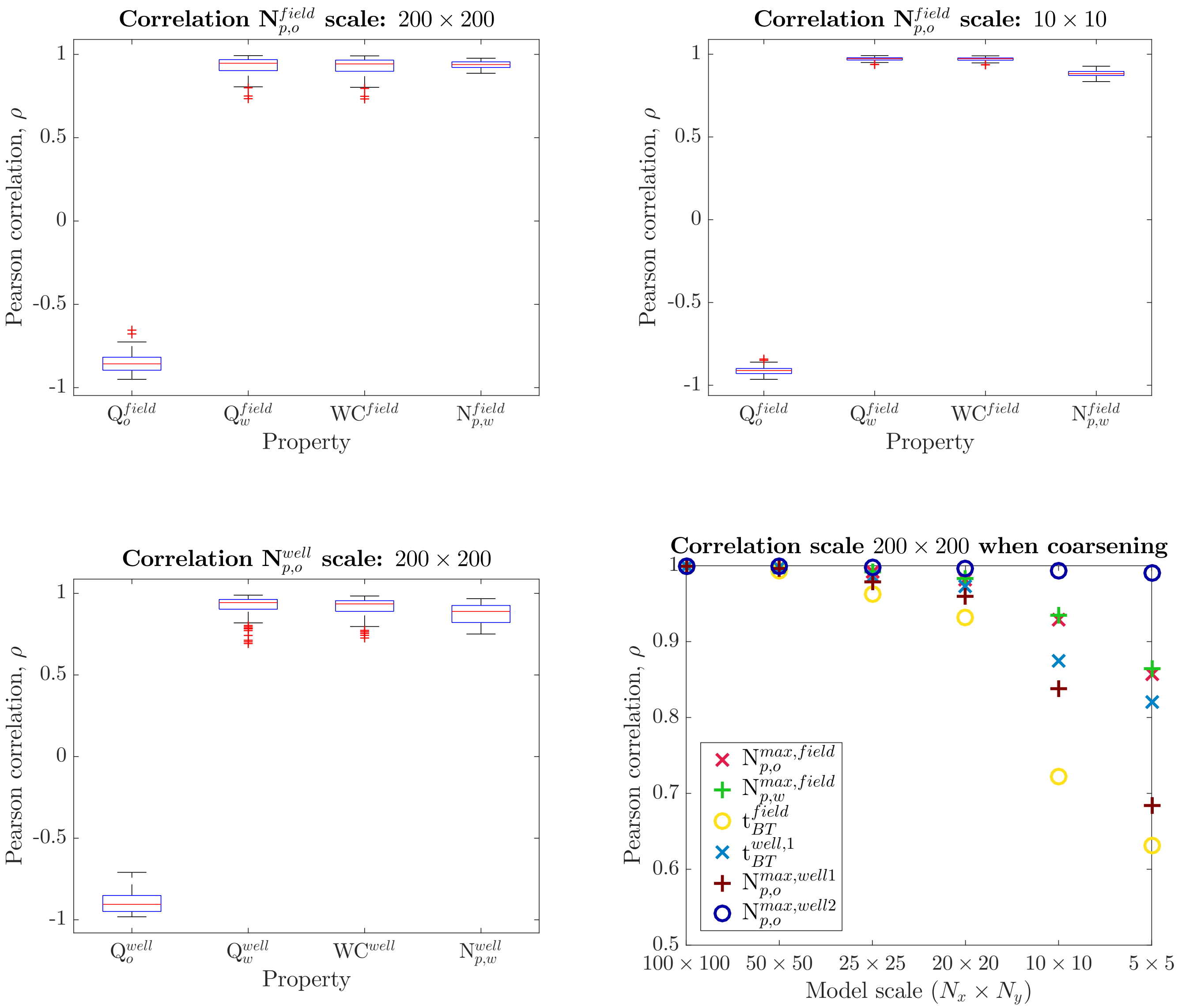


Figure 5.3: **Top left** depicts the correlation between the field oil production (denoted $N_{p,o}^{field}$) and several other properties for each ensemble member using a box-plot method, for the 200 × 200 ensemble scale. **Top right** shows the correlation between the field oil production and several other properties for each ensemble member using a box-plot method, for the 10 × 10 ensemble scale. **Bottom left** illustrates the correlation between the well oil production and several other properties for each ensemble member using a box-plot method, for the 200 × 200 ensemble scale. The three aforementioned graphs show similar behavior for well and field scale as well as fine- and coarse ensemble scales. The negative correlation for the oil rate (denoted $Q_o^{\cdot}$) is because a large cumulative oil production occurs at time of lower oil rate since oil rate tends to decline due to production of water and depletion of the field over time. **Bottom right** depicts the correlation for several properties between ensemble scale 200 × 200 and coarser ensemble scales. $N_{p,\cdot}^{max,\cdot}$ is the maximum production of the specific phase (oil or water) and particular scale (field or well) while $t_{BT}^{\cdot}$ corresponds to the time of breakthrough for a particular scale (field or well). Properties used for computation of the distance matrix of squared dissimilarities $\boldsymbol{D}^{(2)}$ which show a large correlation across several ensemble scales will result in an effective clustering.

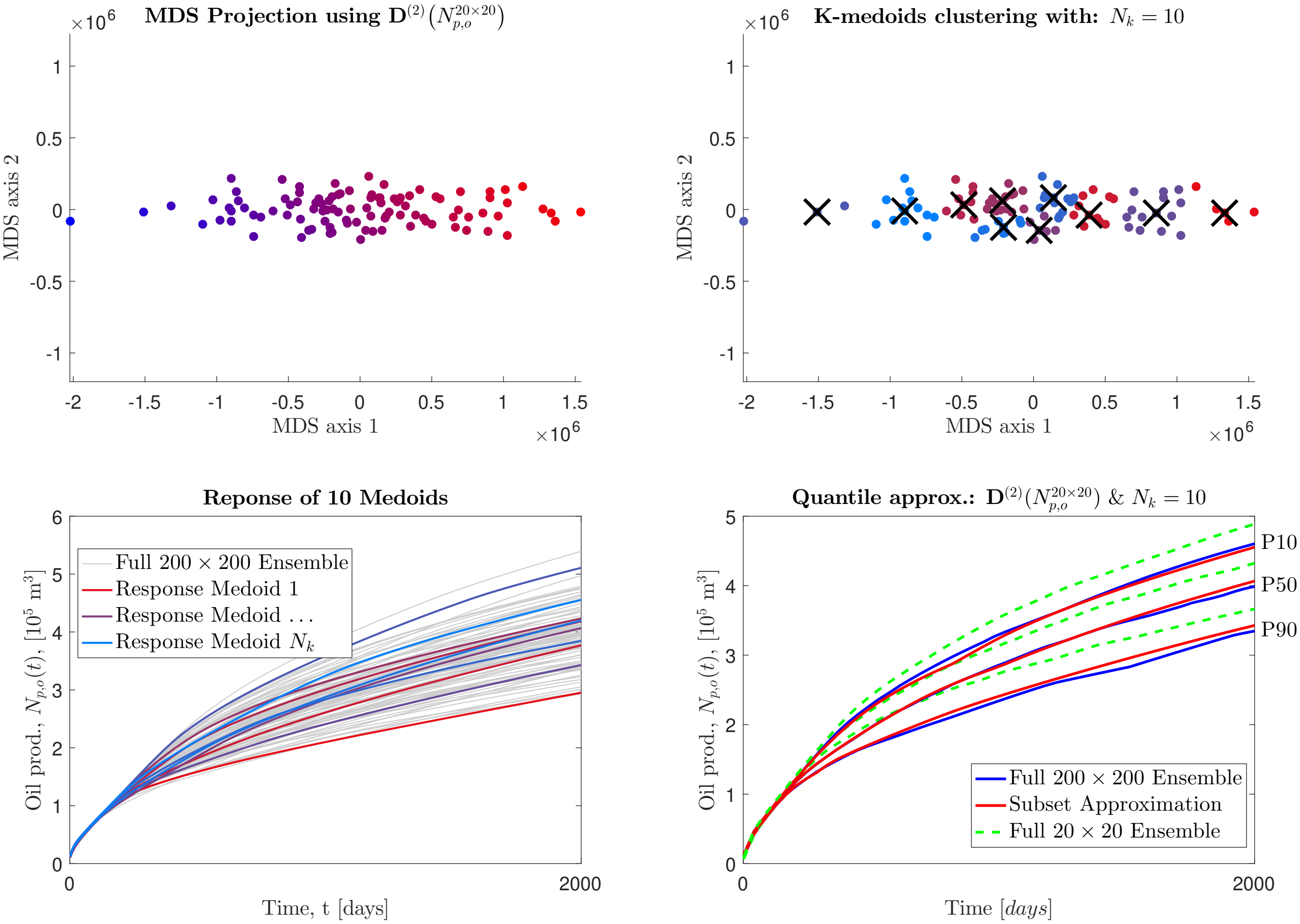


Figure 5.4: **Top left** image shows a MDS projection using cumulative oil production of the coarse (20×20) ensemble as a property for computation of the dissimilarities, denoted by $\boldsymbol{D}^{(2)}\left(N_p^{20\times20}\right)$. **Top right** image shows a K-medoids clustering, using 10 clusters denoted $N_k = 10$. **Bottom left** shows the response of these 10 medoids after forward simulation on the finest scale. Underlying these 10 responses is the "unknown" full fine-scale ensemble response. **Bottom right** image depicts the approximation of the quantiles using the subset of 10 medoids.

5.4 Approximation of ensemble statistics

The main goal is to converge to the full fine-scale ensemble statistics using a subset of fine-scale realizations, guided by coarse-scale distances computed for a particular property. An example of the general workflow is seen in figure 5.4. Two main choices are made in this process, namely:

1. Choice of particular scale and response-metric[6] for the coarse ensemble, denoted $\boldsymbol{D}^{(2)}(\cdot)$ (or for the distance in figure 5.4 by $\boldsymbol{D}^{(2)}(N_{p,o}^{N_x\times N_y})$).
2. Number of clusters, i.e. number of fine-scale simulations.

5.4.1 Determine the number of clusters

Typically, the number of clusters is determined with the Silhouette index. This index measures the separation and tightness (cohesion) of each cluster, i.e. similarity between an object and its own cluster

---

[6] Response-metric refers to the type of flow response used for computing distances between ensemble members: Oil production, water rate, etc..

relative to other clusters, see (Rousseeuw, 1987). This is formally defined as

$$s(i) = \frac{b(i) - a(i)}{max a(i), b(i)} \tag{5.1}$$

where $a(i)$ is the average dissimilarity of i to all other objects in cluster A, and $b(i)$ is the minimum averaged dissimilarity between i and other clusters (excluding A). Therefore $s(i)$ lies on the interval $[-1, 1]$ where close to -1 means a poor clustering since the within cluster dissimilarity $a(i)$ is larger than $b(i)$, while $s(i)$ close to 1 indicates a good clustering since $a(i)$ is much smaller than $b(i)$.

Only considering the actual clustering, the average Silhouette index (width) is an objective (appropriate) measure for determining the number of optimal clusters (Rousseeuw, 1987). However, is it not immediately evident if the number of clusters, or classes of flow responses, is sufficient to derive subset statistics that converge to the full ensemble statistics. The number of clusters determines the number of realizations included in the subset statistics. If the Silhouette index determines that the optimal number of clusters is two, only two realizations are chosen as representatives of the full ensemble. This means that the subset of the quantiles is only based on two realizations. Even for a perfect distance, this will result in a very poor estimation of the three quantiles. Therefore, another way of determining the number of clusters (size of the subset) should be investigated. The best choice is to use one which accounts for the actual objective of converging to full-ensemble statistics with merely a subset of realizations.

In real-world applications, the full fine-scale ensemble response is unknown, which makes determining the appropriate number of clusters required to converge to the (unknown) full fine-scale ensemble statistics even harder. However, the full coarse ensemble response is known since it is used in the MDS projection (by computation of $\boldsymbol{D}^{(2)}(\cdot)$). This means that the convergence rate between the coarse-scale subset and full ensemble can be computed as a function of a number of clusters. This is illustrated in figure 5.5. If the coarse distance is of high quality, as previously defined, the convergence between the fine-scale subset and ensemble can be similar to the convergence between the coarse-scale subset and ensemble.

The coarse scale subset converges to the full ensemble. In the interpretation of this result, we should consider the reduction in variability for coarser grids. This causes the subset for coarser simulations to converge faster to the coarse full ensemble statistics than for finer-scale subsets and ensembles which contain a larger variability in response, see figure 5.6.

5.4.2 Effect of simulation time on dissimilarities between ensemble members and resulting clustering

The clustering seems to be the most effective considering either the whole or the last 1/3 of the simulation time, in the computation of the coarse distances subsequently used in the model selection (through clustering). This is expected for the property water cut since several realizations haven't shown breakthrough in the first interval. The clustering, therefore, has difficulty predicting representative realizations which accurately describe the full simulation time. This is best observed in figure 5.7. The clustering based on coarse distances, taken over the first 1/3 of the simulation, is not able to find representative realizations that also accurately describe the later simulation times. Similar behavior is observed in figure 5.8. For both figure 5.7 and 5.8 the coarsest ensemble scale is used in the computation (most degenerate distance) and 15 representatives are selected for computation of the subset quantiles.

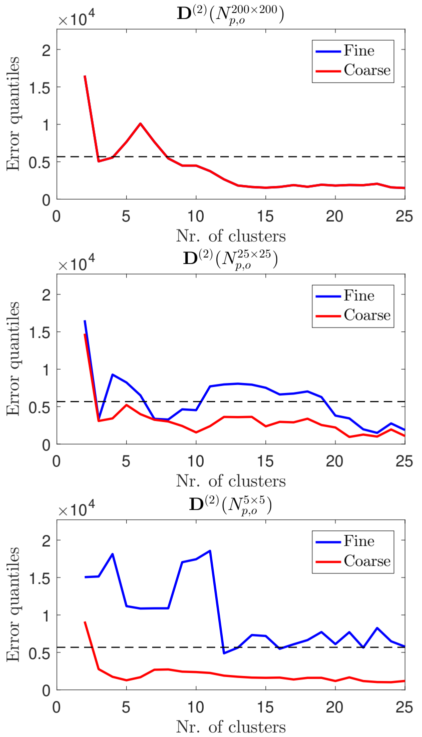


Figure 5.5: Convergence between the subset statistics and full ensemble for both the scale at which the distance is computed (denoted in red with “coarse”) as well as the scale which it is compared with (denoted in blue with “fine”), as a function of clusters. This is similar to the robustness-analysis shown in (Scheidt et al., 2009), however here it is not compared for added noise which reduces the correlation with the finest-scale but compared for the scale at which the coarse distance is computed. Similar convergence rate can be observed for the smaller upscaling ratios. This indicates that when the coarse distance is of larger quality (see above for definition), the convergence rate can predict the required number of clusters. However, caution is advised since a too coarse distance will predict a low amount of required clusters due to the reduced variance in the coarse ensemble response, see 5.6.

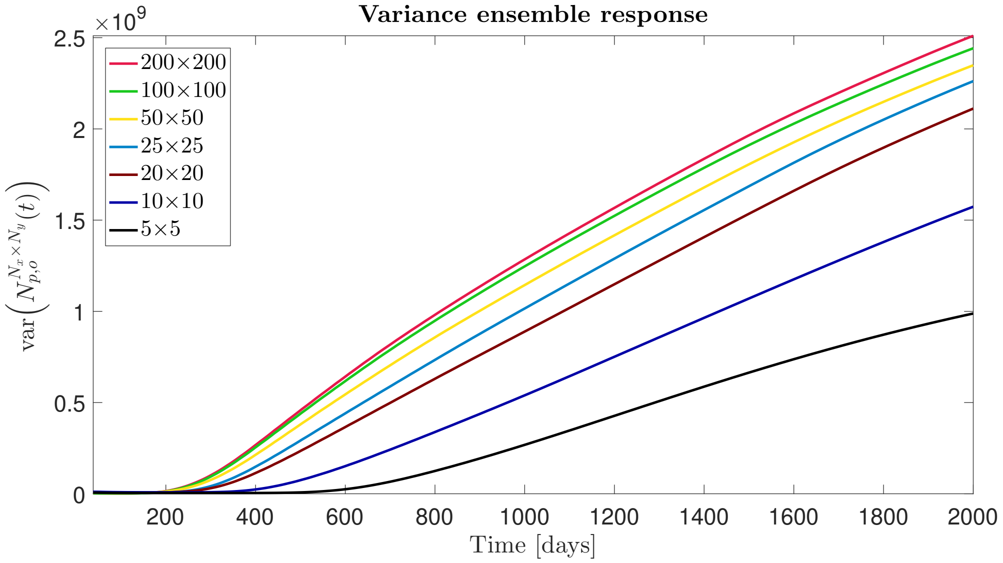


Figure 5.6: Depicts the reduction in variance when coarsening. This poses a risk to using the convergence rate between the subset of models (selected using the coarse distances) and the full coarse-scale ensemble as a measure of required number of clusters/realizations. Generally, due to lower variance, a smaller subset (less realizations) is required to converge/explain the full ensemble statistics on the coarser scale.

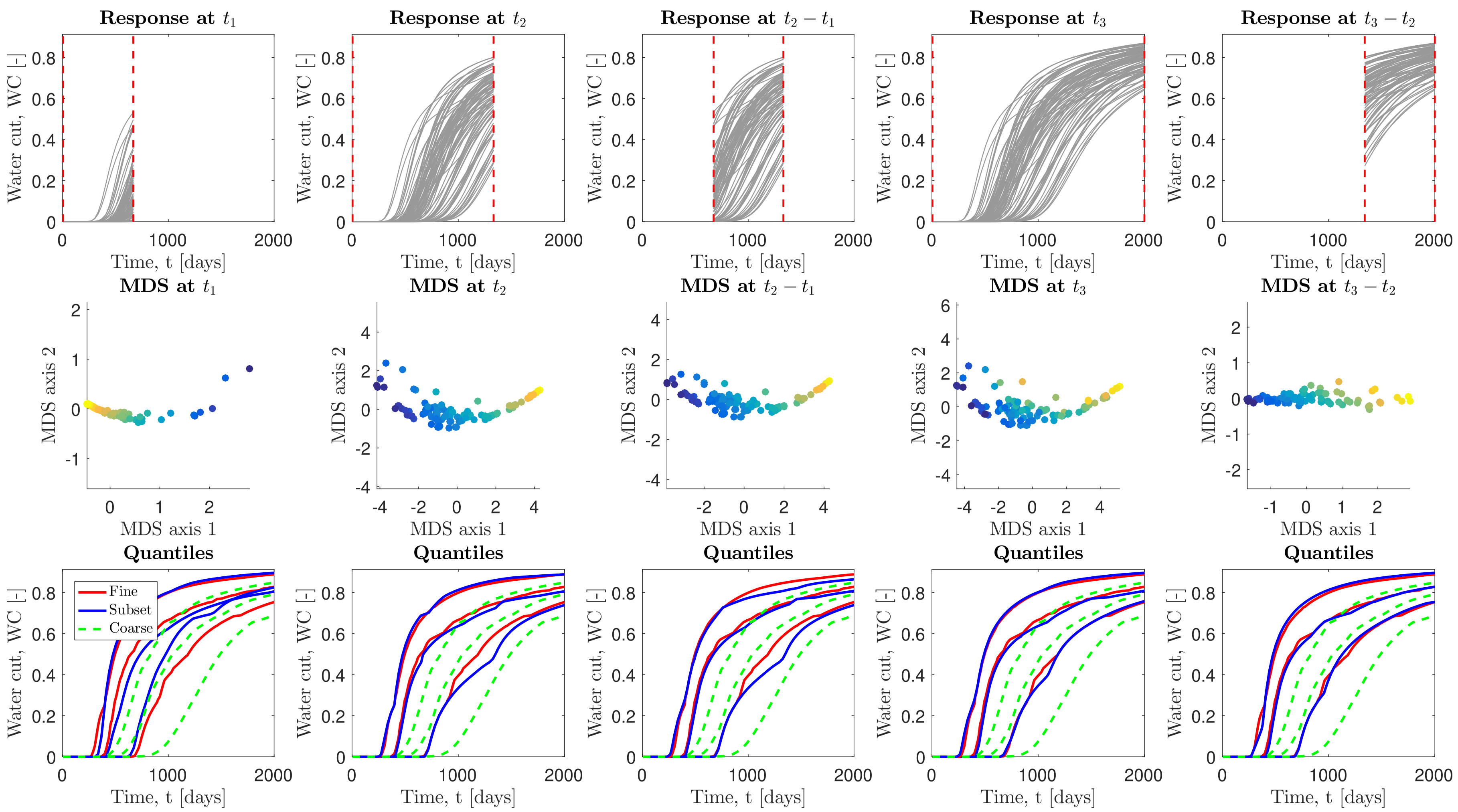


Figure 5.7: **Top row** depicts the coarse response used for computation of the distance, at several time-intervals. **Middle row** displays the effect of the time-intervals on the computed MDS projection. Color of the points corresponds to the final value of the response for that interval. **Bottom row** shows the associated quantiles for the full-ensembles (fine- and coarse-scale) as well as the computed quantiles from the K-Medoids clustering on the MDS projection in the **middle row**. The earliest time interval clearly shows the most deviation from the full fine-scale ensemble quantiles. Note that the coarsest distance is used for the distance computation, and the subset constitutes 15 realizations.

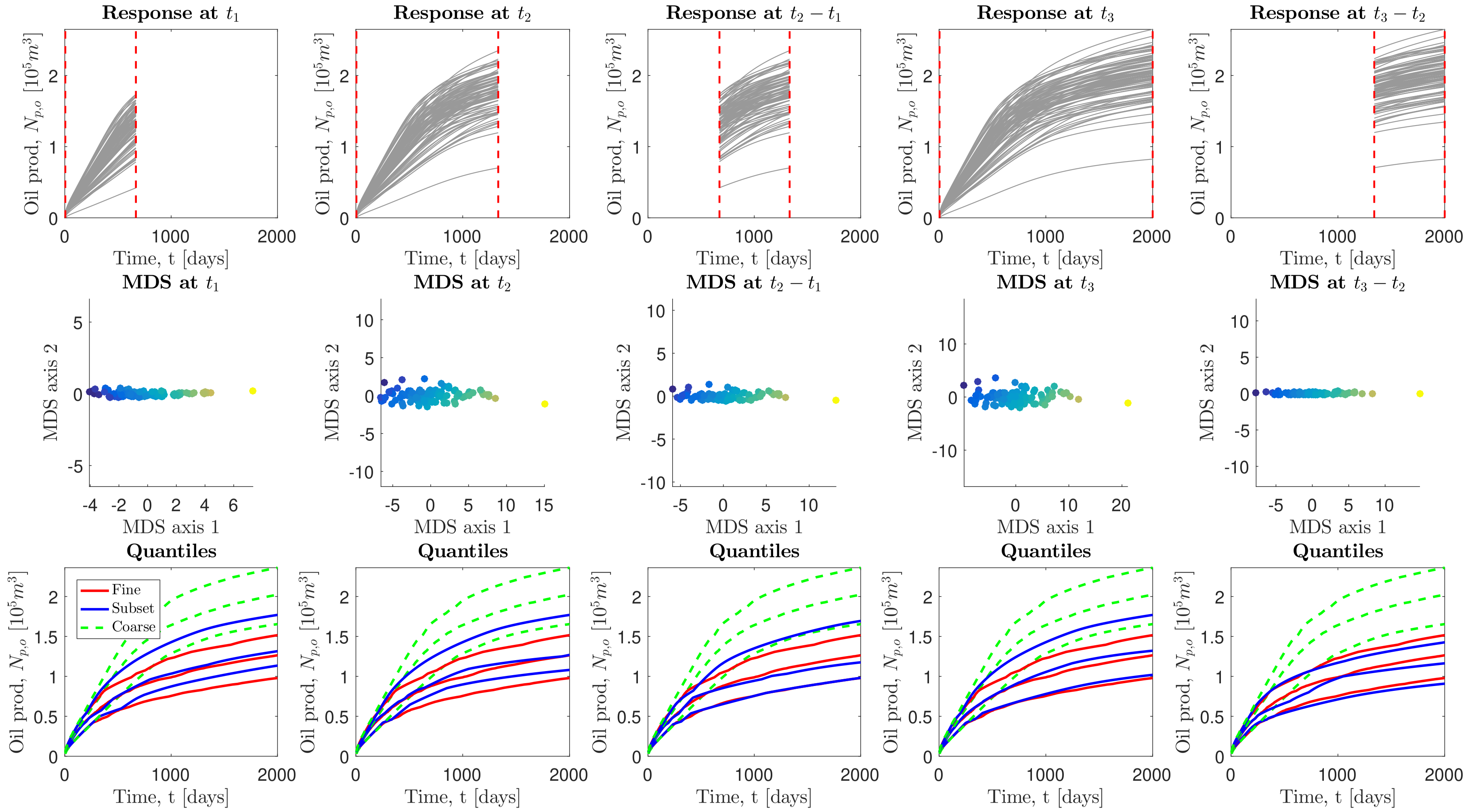


Figure 5.8: **Top row** depicts the coarse response used for computation of the distance, at several time-intervals. **Middle row** displays the effect of the time-intervals on the computed MDS projection. Color of the points corresponds to the final value of the response for that interval. **Bottom row** shows the associated quantiles for the full-ensembles (fine- and coarse-scale) as well as the computed quantiles from the K-Medoids clustering on the MDS projection in the **middle row**. Note that all MDS projections have a similar linear separation in the MDS axis 1, while the resulting subset statistics display very different behavior. This might indicate that the clustering isn't very effect since there are no actual clusters to be recognized. This could cause certain medoids to be appointed as representative which are actually not that representative. Note that the coarsest distance is used for the distance computation, and the subset constitutes 15 realizations.

6

Discussion and Conclusion

The main goal of the thesis project is to determine if there exists a relevant spatial scale in reservoir simulation. The problem of finding this relevant spatial scale is subdivided into two parts, namely finding a relevant static and dynamic spatial scale. Geological modeling resulted in several ensembles of models constituting the test cases for the presented research question. The Discrete Cosine Transform (DCT) is used to determine a relevant static spatial scale, based on the characteristics of the input signal (transmissibility fields). DCT is able to identify the dominant basis-vector, in particular, the basis-vector which explains the most significant pattern contained in the original signal. Note that a basis-vector for the two-dimensional DCT is nothing more than a cosine wave oscillating at two distinct frequencies, namely a frequency in the $x-$ and $y-$direction. Due to stochasticity in the generation of the static reservoir models, this dominant basis-vector might vary from one model to the other. Therefore, the relevant static spatial scale of the ensemble is estimated as the most frequent dominant basis-vector, across every ensemble member.

A coarser representation of the original input signal is obtained through flow-based upscaling. Outlying transmissibility values in the coarser representation are found, similar to the ones mentioned in Nielsen et al.,(2000) and Chen et al.,(2003), caused by the global upscaling. It is shown that after transforming the original signal by taking the natural logarithm and subtracting the mean, DCT is insensitive to the aforementioned outlying values. This means that the DCT is also useful in analyzing the evolution of the static characteristics of the ensemble when coarsening.

It is observed that the relevant static spatial scale remains constant across several ensemble scales, however, degenerates when the ensemble scale approaches the determined dominant scale. This is expected since the dominant characteristics of the signal cannot be accurately represented at this particular scale. A geometric interpretation of the dominant basis-vector is given. This interpretation leads to soft constraints on the allowable (Cartesian) grid-dimensions, required to accurately represent the original signal on the coarser domain. Note that a 2D cosine wave which constitutes the dominant basis-vector, oscillate in the z-direction. The channels in the geological model however oscillate in the xy-plane. This is illustrated in figure 4.6, where the face-to-face connectivity is not preserved. When using a Two-Point Flux Approximation (TPFA) in the upscaling and consecutive flow simulations, this requires additional constraints on the coarse grid-dimensions. Namely, such that fine-grid connectivity of the channels is preserved on the coarse grid. The dynamic behavior of the reservoir determines most (if not all) of the decision making. On top of that, the transfer function from the static (parameter space) to the dynamic (solution space) is highly nonlinear. Therefore knowledge on a relevant static spatial scale is not enough.

Response uncertainty is obtained through forward simulating the hierarchical ensembles in time using AD-GPRS. Uncertainty Quantification is done using a reduced representation of the ensemble response, obtained with Multidimensional Scaling (MDS). The significance of simulation time on the response uncertainty is identified via the construction of an Uncertainty Trajectory. The trajectories of all ensemble scales are compared, and deviation from the finest uncertainty trajectory is used to analyze the effect of coarsening on the response uncertainty. The magnitude of the deviation is dependent on the type of response used in the computation of the MDS projections. When the water cut is chosen as a response, the time component becomes quite evident. This is most likely caused by the bias contained in the coarser ensemble response. Bias refers to consistent later water breakthrough in the coarser responses. Near the end of production time the characteristics of the uncertainty seem similar across all ensemble scales.

This constitutes the main reason for the attempt to exploit the coarser information. Clustering algorithms such as k-medoids, are invariant to translations and orthogonal transformations (Kaufman & Rousseeuw, 2009). This means that if the coarse- and fine-scale Uncertainty Trajectories are similar at a particular time, clustering done on this time interval will be similar for both the coarse- and fine-scale. Resulting representatives from the coarse clustering should therefore in theory approximate representative fine-scale responses. If an appropriate number of clusters is found, the subset statistics can approximate the full ensemble statistics. Typically this appropriate number of clusters is determined through the Silhouette index (Rousseeuw, 1987). For our purpose, this doesn't always work. E.g., for some responses, there might only exist two real clusters (based on the structure of the projection). This means that there will only be two representative fine-scale realizations in the subset. The statistics of the subset will never approximate all three quantiles of the full fine-scale ensemble accurately. A more robust method would be to compute the convergence of the subset statistics and the full coarse-scale ensemble. The number of clusters required to obtain the desired convergence can then be used for the representatives and subsequently simulated on the fine-scale, constituting the subset statistics. Reduction of variability in the coarser responses might pose a potential threat to this methodology and should, therefore, be investigated further.

On top of the reduction in variability of the coarser response, the fine-scale ensemble response in real-world applications is unknown. A possible way to handle these problems is using the following general work-flow:

1. Construct Hf (geological) model.
2. Perform DCT to extract dominant scale.
3. Construct coarser representation using flow-based upscaling, where the grid-dimensions are governed by the dominant scale estimated by the DCT.
4. Simulate fluid flow on the full coarse-scale ensemble.
5. Determine number of clusters via convergence test.
6. Simulate fluid flow on the subset of fine-scale representatives.
7. Compute the Uncertainty Trajectory for the subset of representatives, on the fine- and coarse scale.
8. Examine distance between the fine- and coarse-scale trajectories. If the behavior of the coarse-scale trajectory is unsatisfactory, re-sample additional representatives and simulate these on the fine-scale.

This work-flow is still in an experimental phase and requires verification of robustness. Besides that, it also requires a better definition of certain terms, such as “unsatisfactory deviation in uncertainty trajectory”. The limitations of the model selection should be thoroughly understood.

In case future work is considered, an ever larger contrast in permeability between the reservoir and non-reservoir facies can be considered as well as a larger number of depositional facies. Also, note that the reservoirs in this work are considered geologically young, i.e. reservoir performance is governed by sedimentology (which governs the petrophysics) only (Galloway & Hobday, 2012). Fractures occur naturally in reservoirs (Berkowitz, 2002), and are currently heavily investigated (both from a geological and reservoir simulation point of view). The relevant spatial scale of a reservoir model could be heavily influenced by the presence of fractures and therefore this could be an interesting future research endeavor.

Note that flow-based gridding techniques in conjunction with flow-based upscaling methods, such as summarized in Durlofsky (2005), provide a better coarse representation of the geological features of the fine grid. This highlights one limitation of the performed study and could also be included in future research.

Appendices

A

DCT on LineDrive and GangesDelta

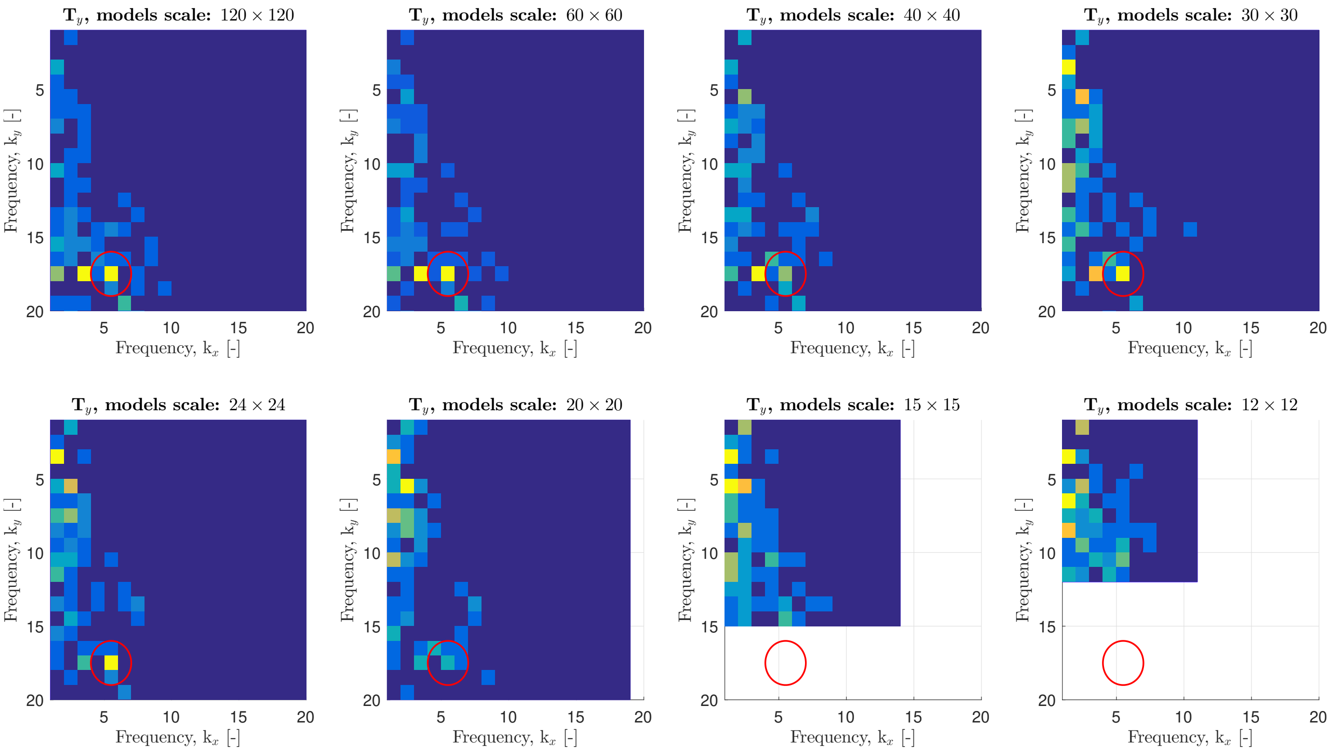


Figure A.1: Depicting the degeneration of the characteristics of the LineDrive$_3$ ensemble at the model scale 20×20.

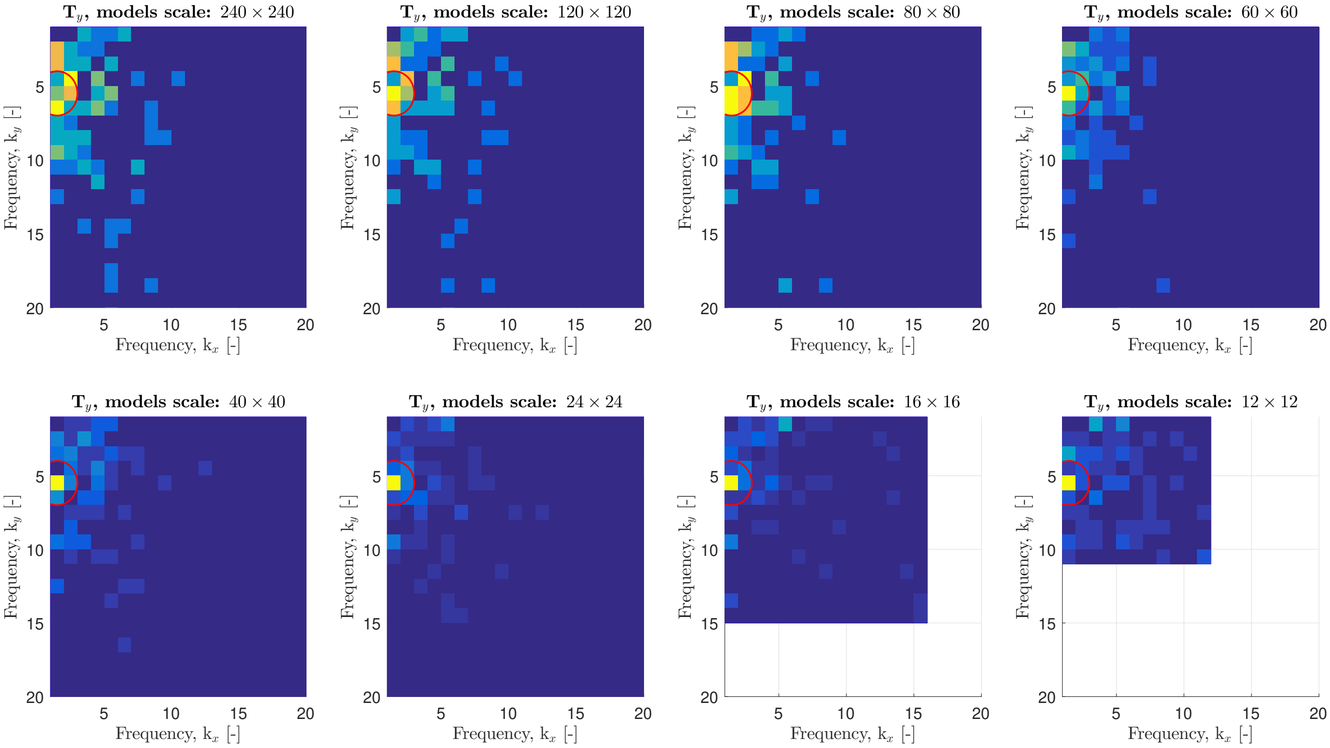


Figure A.2: Depicting the preservation of the characteristics of the GangesDelta ensemble all model scales while coarsening. Note that even at a coarse scale of 12×12, which is a factor of 20×20 upscaled, the dominant basis-vector according to the DCT is still similar to the finest scale. This is likely indicates the thick channel features, which can easily be described by a low frequency cosine wave, while the high-frequency (smaller channels) are not easily characterized by the DCT.

B

Subset statistics for full fine-scale ensembles for various properties and ensembles.

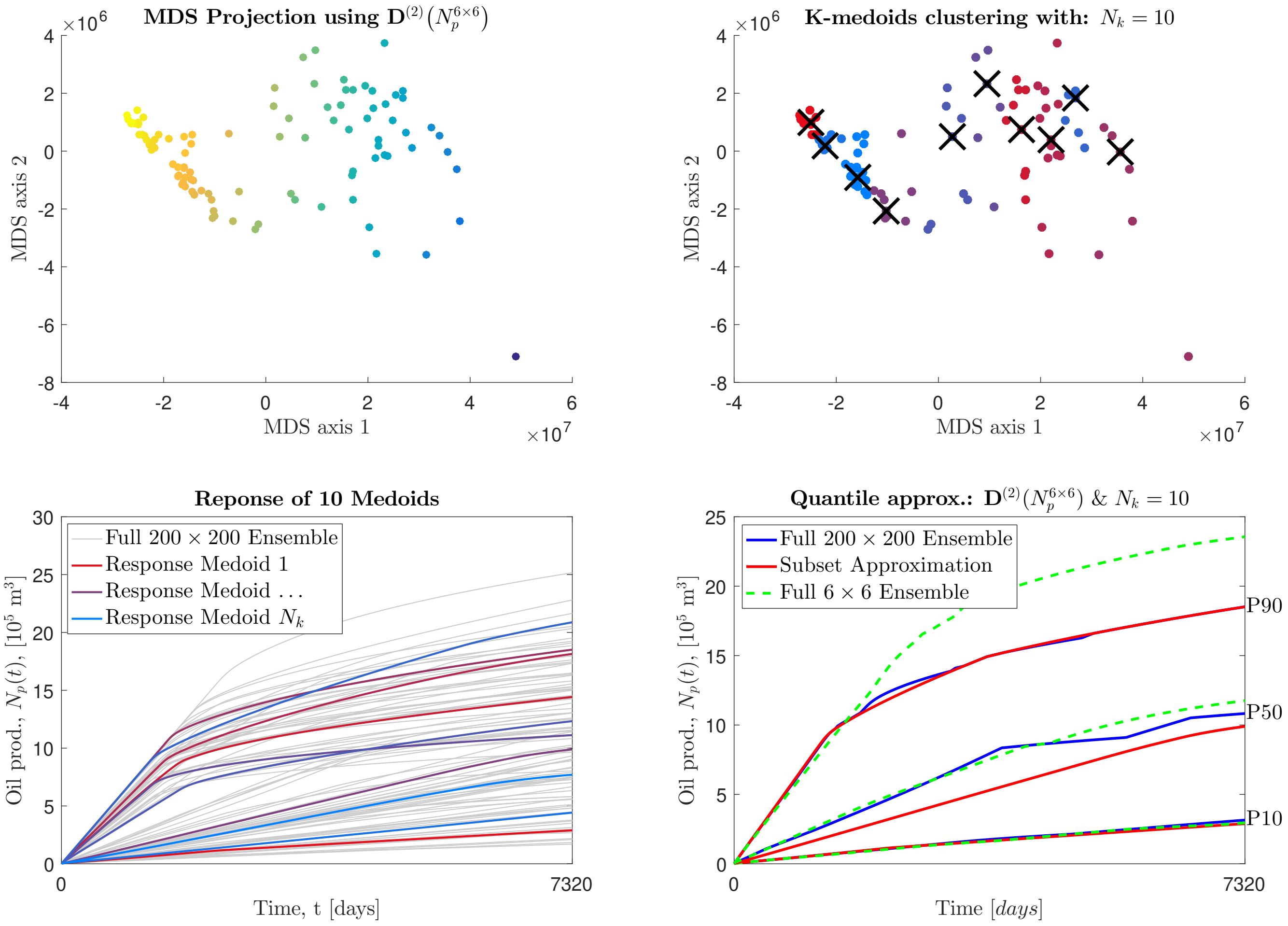


Figure B.1: Depicting coarse distance model selection, for LineDrive$_2$ ensemble well 1.

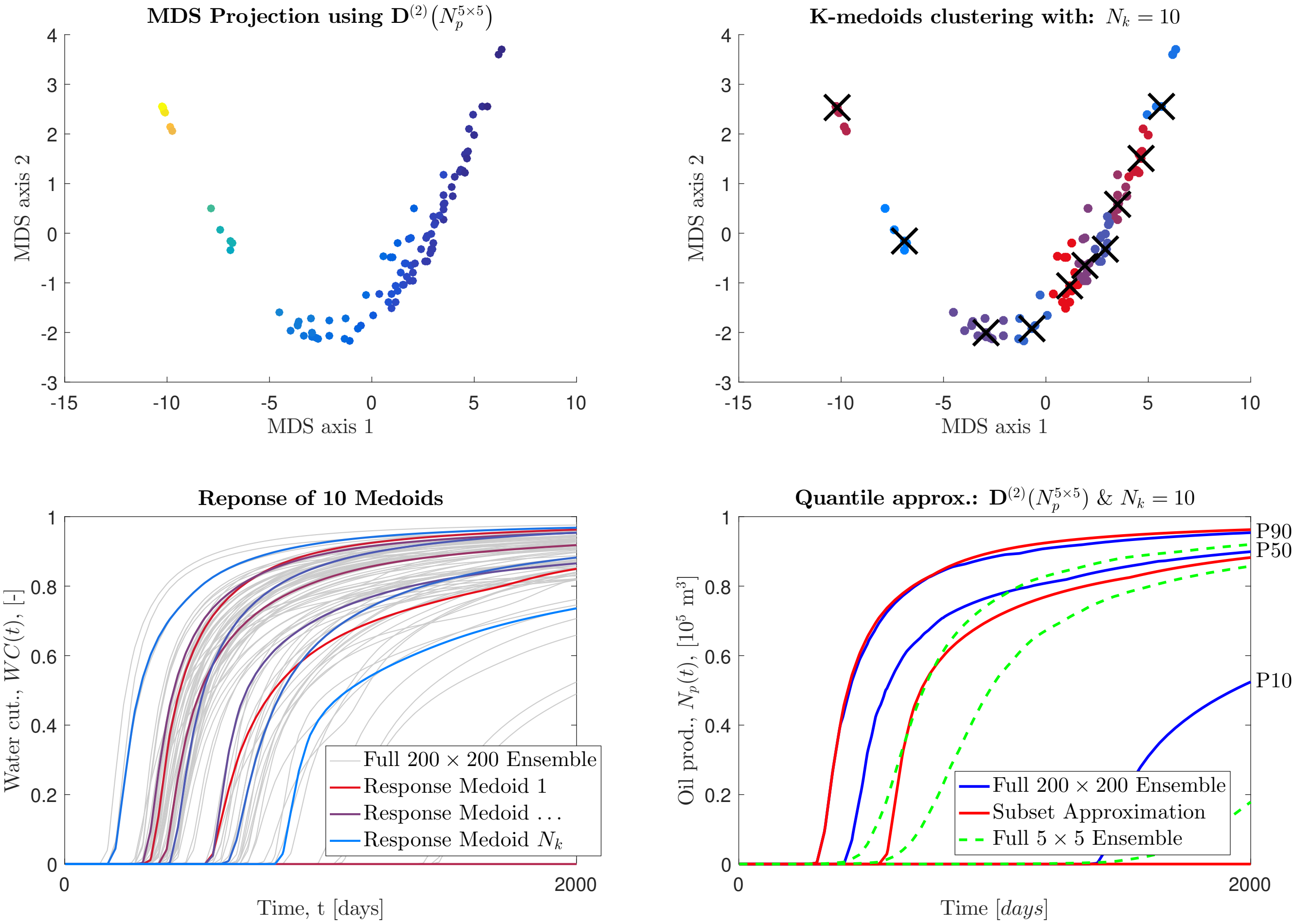


Figure B.2: Depicting coarse distance approximating water cut for Flumy$_3$ ensemble, well 1. The limitations of the clustering used in this thesis work is visible.

References

Ahmed, N., Natarajan, T., & Rao, K. R. (1974). Discrete cosine transform. *IEEE transactions on Computers*, *100*(1), 90–93.

Arpat, G. B. (2005). *Sequential simulation with patterns.* Stanford University.

Aziz, K., & Settari, A. (1979). *Petroleum reservoir simulation.* Chapman & Hall.

Bear, J. (2013). *Dynamics of fluids in porous media.* Courier Corporation.

Berkowitz, B. (2002). Characterizing flow and transport in fractured geological media: A review. *Advances in water resources*, *25*(8), 861–884.

Bishop, C. M. (2006). *Pattern recognition and machine learning.* springer.

Borg, I., & Groenen, P. J. (2005). *Modern multidimensional scaling: Theory and applications.* Springer Science & Business Media.

Caers, J. (2011). *Modeling uncertainty in the earth sciences.* John Wiley & Sons.

Caers, J., Park, K., & Scheidt, C. (2010). Modeling uncertainty of complex earth systems in metric space. In *Handbook of geomathematics* (pp. 865–889). Springer.

Caers, J., & Zhang, T. (2004). Multiple-point geostatistics: a quantitative vehicle for integrating geologic analogs into multiple reservoir models.

Cao, H. (2002). *Development of techniques for general purpose simulators* (Unpublished doctoral dissertation). Stanford University.

Chen, C., Hu, D., Westacott, D., & Loveless, D. (2013). Nanometer-scale characterization of microscopic pores in shale kerogen by image analysis and pore-scale modeling. *Geochemistry, Geophysics, Geosystems*, *14*(10), 4066–4075.

Chen, Y., Durlofsky, L., Gerritsen, M., & Wen, X. (2003). A coupled local-global upscaling approach for simulating flow in highly heterogeneous formations. *Advances in Water Resources*, *26*(10), 1041-1060.

Chen, Z., Huan, G., & Ma, Y. (2006). *Computational methods for multiphase flows in porous media.* SIAM.

Demyanov, V., Rojas, T., Arnold, D., & Christie, M. (2013). Uncertainty quantification in history matching of fluvial reservoirs with connectivity analysis and realistic geology. In *75th eage conference & exhibition incorporating spe europec 2013.*

Donselaar, M. E., & Overeem, I. (2008). Connectivity of fluvial point-bar deposits: An example from the miocene huesca fluvial fan, ebro basin, spain. *AAPG bulletin*, *92*(9), 1109–1129.

Dubuisson, M.-P., & Jain, A. K. (1994). A modified hausdorff distance for object matching. In *Pattern recognition, 1994. vol. 1-conference a: Computer vision & image processing., proceedings of the 12th iapr international conference on* (Vol. 1, pp. 566–568).

Durlofsky, L. J. (1991). Numerical calculation of equivalent grid block permeability tensors for heterogeneous porous media. *Water resources research*, *27*(5), 699–708.

Durlofsky, L. J. (2005). Upscaling and gridding of fine scale geological models for flow simulation. In *8th international forum on reservoir simulation iles borromees, stresa, italy* (Vol. 2024).

Ethridge, F. G., & Schumm, S. A. (1977). Reconstructing paleochannel morphologic and flow characteristics: methodology, limitations, and assessment.

Fenwick, D., & Batycky, R. (2011). Using metric space methods to analyse reservoir uncertainty. In *Proceedings of the 2011 gussow conference.*

Galloway, W. E. (1981). Depositional architecture of cenozoic gulf coastal plain fluvial systems.

Galloway, W. E., & Hobday, D. K. (2012). *Terrigenous clastic depositional systems: Applications to fossil fuel and groundwater resources.* Springer Science & Business Media.

Gomez-Hernandez, J. J., Journel, A. G., et al. (1994). Stochastic characterization of gridblock permeabilities. *SPE Formation Evaluation*, *9*(02), 93–99.

Grappe, B., Cojan, I., Ors, F., & Rivoirard, J. (2016). Dynamic modelling of meandering fluvial systems at the reservoir scale, flumy software. In *Second conference on forward modelling of sedimentary systems.*

Haldorsen, H. H. (1986). Simulator parameter assignment and the problem of scale in reservoir engineering. *Reservoir characterization*, *6*.

Hartigan, J. A., & Wong, M. A. (1979). Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, *28*(1), 100–108.

Hashemi, S., Javaherian, A., Ataee-pour, M., Tahmasebi, P., & Khoshdel, H. (2014). Channel characterization using multiple-point geostatistics, neural network, and modern analogy: A case study from a carbonate reservoir, southwest iran. *Journal of Applied Geophysics*, *111*, 47–58.

He, C., & Durlofsky, L. (2006). Structured flow-based gridding and upscaling for modeling subsurface flow. *Advances in Water Resources*, *29*(12), 1876-1892.

Helmig, R., et al. (1997). *Multiphase flow and transport processes in the subsurface: a contribution to the modeling of hydrosystems.* Springer-Verlag.

Henriquez, A., Tyler, K. J., Hurst, A., et al. (1990). Characterization of fluvial sedimentology for reservoir simulation modeling. *SPE Formation Evaluation*, *5*(03), 211–216.

Hoffimann, J., Scheidt, C., Barfod, A., & Caers, J. (2017). Stochastic simulation by image quilting of process-based geological models. *Computers & Geosciences*.

Holden, L., & Lia, O. (1992). A tensor estimator for the homogenization of absolute permeability. *Transport in Porous Media*, *8*(1), 37–46.

Holden, L., & Nielsen, B. (2000). Global upscaling of permeability in heterogeneous reservoirs; the output least squares (ols) method. *Transport in Porous Media*, *40*(2), 115-143.

Jafarpour, B., Goyal, V. K., McLaughlin, D. B., & Freeman, W. T. (2010). Compressed history matching: exploiting transform-domain sparsity for regularization of nonlinear dynamic data integration problems. *Mathematical Geosciences*, *42*(1), 1–27.

Jain, A. K. (1989). *Fundamentals of digital image processing.* Prentice-Hall, Inc.

Jansen, J. D. (2013). *A systems description of flow through porous media.* Springer.

Jin, X., & Han, J. (2010). K-medoids clustering. In C. Sammut & G. I. Webb (Eds.), *Encyclopedia of machine learning* (pp. 564–565). Boston, MA: Springer US.

Jungreuthmayer, C., Steppert, P., Sekot, G., Zankel, A., Reingruber, H., Zanghellini, J., & Jungbauer, A. (2015). The 3d pore structure and fluid dynamics simulation of macroporous monoliths: High permeability due to alternating channel width. *Journal of Chromatography A*, *1425*, 141–149.

Karssenberg, D., Tornqvist, T. E., & Bridge, J. S. (2001). Conditioning a process-based model of sedimentary architecture to well data. *Journal of Sedimentary Research*, *71*(6), 868–879.

Kaufman, L., & Rousseeuw, P. (1987). *Clustering by means of medoids.* North-Holland.

Kaufman, L., & Rousseeuw, P. J. (2009). *Finding groups in data: an introduction to cluster analysis* (Vol. 344). John Wiley & Sons.

Keogh, K. J., Martinius, A. W., & Osland, R. (2007). The development of fluvial stochastic modelling in the norwegian oil industry: A historical review, subsurface implementation and future directions. *Sedimentary Geology*, *202*(1), 249–268.

Kim, K. H., Lee, K., Lee, H. S., Rhee, C. W., & Shin, H. D. (2017). Lithofacies modeling by multipoint statistics and economic evaluation by npv volume for the early cretaceous wabiskaw member in athabasca oilsands area, canada. *Geoscience Frontiers*.

Kitanidis, P. K. (1990). Effective hydraulic conductivity for gradually varying flow. *Water Resources Research*, *26*(6), 1197–1208.

Lay, D. C. (2003). *Linear algebra and its applications.* Addison Wesley, Boston.

Lee, K., Lim, J., Choe, J., & Lee, H. S. (2017). Regeneration of channelized reservoirs using history-matched facies-probability map without inverse scheme. *Journal of Petroleum Science and Engineering*, *149*, 340–350.

LeVeque, R. J. (2002). *Finite volume methods for hyperbolic problems* (Vol. 31). Cambridge university press.

Lopez, S., Cojan, I., Rivoirard, J., & Galli, A. (2009). Process-based stochastic modelling: meandering channelized reservoirs. *Analogue Numer Model Sediment Syst: From Understand Predict (Special Publ. 40 of the IAS)*, *40*.

Marden, J. I. (1996). *Analyzing and modeling rank data.* CRC Press.

Mariethoz, G., & Caers, J. (2014). *Multiple-point geostatistics: stochastic modeling with training images.* John Wiley & Sons.

MathWorks, I. (2017). *Statistics and machine learning toolbox.* The MathWorks, Inc., Natick, Massachusetts, United States.

Mattax, C. C., Dalton, R. L., et al. (1990). Reservoir simulation (includes associated papers 21606 and 21620). *Journal of Petroleum Technology*, *42*(06), 692–695.

Miall, A. D. (2013). *The geology of fluvial deposits: sedimentary facies, basin analysis, and petroleum geology.* Springer.

Michael, H., Li, H., Boucher, A., Sun, T., Caers, J., & Gorelick, S. (2010). Combining geologic-process models and geostatistics for conditional simulation of 3-d subsurface heterogeneity. *Water Resources Research*, *46*(5).

Nordahl, K., & Ringrose, P. S. (2008). Identifying the representative elementary volume for permeability in heterolithic deposits using numerical rock models. *Mathematical geosciences*, *40*(7), 753–771.

Omre, H., Lødøen, O. P., et al. (2004). Improved production forecasts and history matching using approximate fluid-flow simulators. *SPE Journal*, *9*(03), 339–351.

Ortiz, J., & Deutsch, C. V. (2002). Calculation of uncertainty in the variogram. *Mathematical Geology*, *34*(2), 169–183.

O'Sullivan, A., Christie, M., et al. (2005). Solution error models: a new approach for coarse grid history matching. In *Spe reservoir simulation symposium.*

Peaceman, D. W. (1977). *Fundamentals of numerical reservoir simulation.* Elsevier Scientific Publishing Company.

Pyrcz, M. J., Boisvert, J. B., & Deutsch, C. V. (2008). A library of training images for fluvial and deepwater reservoirs and associated code. *Computers & Geosciences*, *34*(5), 542–560.

Rao, K. R., & Yip, P. (2014). *Discrete cosine transform: algorithms, advantages, applications.* Academic press.

Remy, N., Boucher, A., & Wu, J. (2009). *Applied geostatistics with sgems: a user's guide.* Cambridge University Press.

Rongier, G., Collon, P., & Renard, P. (2017). Stochastic simulation of channelized sedimentary bodies using a constrained l-system. *Computers & Geosciences*, *105*, 158–168.

Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, *20*, 53–65.

Scheidt, C., & Caers, J. (2009). Representing spatial uncertainty using distances and kernels. *Mathematical Geosciences*, *41*(4), 397–419.

Scheidt, C., Caers, J., Chen, Y., & Durlofsky, L. J. (2011). A multi-resolution workflow to generate high-resolution models constrained to dynamic data. *Computational Geosciences*, *15*(3), 545–563.

Scheidt, C., Caers, J., et al. (2009). Uncertainty quantification in reservoir performance using distances and kernel methods–application to a west africa deepwater turbidite reservoir. *SPE Journal*, *14*(04), 680–692.

Schönemann, P. H. (1966). A generalized solution of the orthogonal procrustes problem. *Psychometrika*, *31*(1), 1–10.

Schumm, S. A. (1981). Evolution and response of the fluvial system, sedimentologic implications.

Searle, S. R., & Khuri, A. I. (2017). *Matrix algebra useful for statistics.* John Wiley & Sons.

Seifert, D., & Jensen, J. (2000). Object and pixel-based reservoir modeling of a braided fluvial reservoir. *Mathematical Geology*, *32*(5), 581–603.

Sok, R. M., Knackstedt, M. A., Varslot, T., Ghous, A., Latham, S., Sheppard, A. P., et al. (2010). Pore scale characterization of carbonates at multiple scales: Integration of micro-ct, bsem, and fibsem. *Petrophysics*, *51*(06).

Stanford. (2012). *AD-GPRS, autmatic differentiation general purpose reservoir simulator.* Retrieved from `https://supri-b.stanford.edu/research-areas/ad-gprs`

Steinley, D., & Brusco, M. J. (2007). Initializing k-means batch clustering: A critical evaluation of several techniques. *Journal of Classification*, *24*(1), 99–121.

Strang, G. (1993). *Introduction to linear algebra* (Vol. 3). Wellesley-Cambridge Press Wellesley, MA.

Strang, G. (1999). The discrete cosine transform. *SIAM review*, *41*(1), 135–147.

Straubhaar, J., & Malinverni, D. (2014). Addressing conditioning data in multiple-point statistics simulation algorithms based on a multiple grid approach. *Mathematical Geosciences*, *46*(2), 187–204.

Straubhaar, J., Renard, P., Mariethoz, G., Froidevaux, R., & Besson, O. (2011). An improved parallel multiple-point algorithm using a list approach. *Mathematical Geosciences*, *43*(3), 305–328.

Strebelle, S. (2002). Conditional simulation of complex geological structures using multiple-point statistics. *Mathematical Geology*, *34*(1), 1–21.

Strebelle, S., & Levy, M. (2008). Using multiple-point statistics to build geologically realistic reservoir models: the mps/fdm workflow. *Geological Society, London, Special Publications*, *309*(1), 67–74.

Sun, W., & Durlofsky, L. J. (2017). A new data-space inversion procedure for efficient uncertainty quantification in subsurface flow problems. *Mathematical Geosciences*, 1–37.

Tan, X., Tahmasebi, P., & Caers, J. (2014). Comparing training-image based algorithms using an analysis of distance. *Mathematical Geosciences*, *46*(2), 149–169.

Voskov, D. (2012). An extended natural variable formulation for compositional simulation based on tie-line parameterization. *Transport in porous media*, *92*(3), 541–557.

Voskov, D., Zhou, Y., & Volkov, O. (2012). Technical description of ad-gprs. *Energy Resources Engineering, Stanford University.*

Voskov, D. V., & Tchelepi, H. A. (2012). Comparison of nonlinear formulations for two-phase multi-component eos based simulation. *Journal of Petroleum Science and Engineering*, *82*, 101–111.

Wen, X., Durlofsky, L., & Edwards, M. (2003a). Upscaling of channel systems in two dimensions using flow-based grids. *Transport in Porous Media*, *51*(3), 343–366.

Wen, X., Durlofsky, L., & Edwards, M. (2003b). Use of border regions for improved permeability upscaling. *Mathematical Geology*, *35*(5), 521–547.

White, C., & Horne, R. (1987). Computing absolute transmissibility in the presence of fine-scale heterogeneity. In (p. 209-220).

Wu, X.-H., Efendiev, Y., & Hou, T. Y. (2002). Analysis of upscaling absolute permeability. *Discrete and Continuous Dynamical Systems Series B*, *2*(2), 185–204.

Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, *22*(11), 1330–1334.