

Ensemble based Multiscale Reservoir Management under Geological Uncertainty

Mircea-Alexandru Helici



Ensemble based Multiscale Reservoir Management under Geological Uncertainty

by

Mircea-Alexandru Helici

to obtain the degree of Master of Science
in Applied Mathematics
at the Delft University of Technology,
to be defended publicly on Tuesday August 29, 2017 at 15:00.

Student number: 4415183

Thesis committee:	Prof. dr. ir. Arnold W. Heemink,	TU Delft, supervisor
	Prof. dr. ir. Jan Dirk Jansen,	TU Delft, supervisor
	Ir. Rafael J. Moraes,	TU Delft, supervisor
	Dr. ir. Rahul M. Fonseca,	TNO, supervisor
	Dr. ir. Martin van Gijzen,	TU Delft

An electronic version of this thesis is available at <http://repository.tudelft.nl/>.

Cover photo by Nol Anders

Acknowledgments

First and foremost, I would like to sincerely thank my supervisors, Arnold Heemink and Jan Dirk Jansen, for their continuous support and insightful guidance throughout the course of this research.

Moreover, I am thankful for the daily inputs, active participation, and unconditional patience provided by my supervisor Rafael Moraes. His comments and expertise proved to be highly valued in steering the research effort along the proper path.

With respect to my external supervisor, Rahul Fonseca, I wish to express my appreciation for the constructive insight that he provided me with, which unequivocally helped shape this thesis into its final form.

I thank the entire Petroleum Geosciences division at TNO for offering me the opportunity of taking on this research topic. Special thanks to Alin and Joana for ride sharing.

I would like to express my deepest gratitude and thanks to my parents, grandparents, sister and brother for their unconditional love and support and for always encouraging me to pursue my Masters at TU Delft. To my girlfriend, Diana, thanks for the confidence, love and for completing me.

Thanks to all my professors at TU Delft. To all my fellow classmates from Applied Mathematics, especially to Mario, Baruch, Kenny, Zoe, Nam, Shobhit and Aseem for their positive influence on me.

Last, but not least, thanks to all my friends for the great time we have spent together. Diana and Mihai, with you the Netherlands feels like home.

*Mircea-Alexandru Helici
Delft, August 2017*

Contents

1	Introduction	1
2	Flow in porous media	5
2.1	Fundamental Reservoir Properties & Concepts	5
2.1.1	Porosity	5
2.1.2	Permeability.	5
2.1.3	Viscosity	5
2.1.4	Saturation	5
2.1.5	Relative Permeability.	6
2.2	Governing Equations.	6
2.3	Numerical simulation strategies	7
2.4	Discretization	8
3	Multiscale Finite Volume Method	11
3.1	Preprocessing	11
3.2	Simulation - Discretized MSFV	13
3.3	Multi-phase flow MSFV.	14
3.4	i-MSFV	15
4	Ensemble optimization	17
4.1	Ensemble-Based Deterministic Formulation	17
4.2	EnOpt – Robust Optimization	19
4.3	StoSAG	20
5	MS-StoSAG workflow	21
6	Numerical experiments	25
6.1	Five-Spot Model	26
6.1.1	Deterministic Life-cycle Optimization	26
6.1.2	Robust Life-cycle Optimization.	29
6.1.3	Computational Efficiency.	30
6.2	Kanaal Reservoir Model	31
6.2.1	Robust Life-cycle Optimization.	32
6.2.2	Computational Efficiency.	34
7	Conclusions	37
7.1	Recommendations	37
	Bibliography	39

Introduction

There is an on-going propensity for the world economy to expand year on year due to the increasing number of emerging markets as well as the growth in global population. To sustain this, the reliance on fossil fuels will continue, even if the costs for renewable energy plunges. In consequence, the “easy oil” supply, i.e. hydrocarbon which is to some extent easy to extract, can no longer fulfill the need (Fonseca, 2015). Thus, this is on one hand forcing the industry to develop new technologies to increase the recovery of hydrocarbon, while on the other hand making them exploit more difficult reservoirs (David, 2010).

By nature in reservoirs the hydrocarbon i.e. petroleum and natural gas, reside in the pores of the rocks. When wells are drilled to extract the hydrocarbon resources, because the pressure in the reservoir is higher than the one in the wells, the hydrocarbon is naturally pushed into the wells and up to the surface. As the underground pressure is slowly decreasing, the rocks compress pushing even more fluids into the wells. However, typically a small percentage of up to 10% of the hydrocarbons are produced this way (Schlumberger, 2016a).

For most applications a pressure-maintenance method is used from the beginning of the hydrocarbon recovery process. The most common pressure maintenance technique used is waterflooding. This is applied in order to maintain the reservoir's pressure and further displace hydrocarbon towards the production wells. Another advantage of this method is that it can limit the compression of the rocks, reducing the impact on the geometry of the reservoir and the layers above known as subsidence (surface deformation). This would also be beneficial if for example, the reservoir is below a city in which case any land alterations can cause serious social and economic problems. The water flooding process is well understood and in recent year there has been a growing number of applications which have investigated the possibility to optimize this process for a wide range of objectives. While the physical mechanism governing the water flooding process are well understood it is nearly impossible to describe with full certainty the geological and fluid properties of the reservoirs which employ water flooding or any recovery mechanism. Thus when modeling, especially when optimizing, any hydrocarbon recovery process it is essential to account for a variety of different types of uncertainties.

The uncertainties in reservoir modeling stem from many different sources. An analysis of the geological and fluid characteristics of the reservoir must take place before the recovery process is started, this process is not one without uncertainty. Further data about the reservoir is gathered by running lab experiments on the rocks acquired during the drilling of the wells and by a series of geological surveys, in addition to seismic data. To better handle uncertainty in the measurements, the geologists create a set of geological models of the reservoir starting from the collected data. To avoid any confusion around the word “model” and to follow the literature standards, the geological models that present the permeability field of the reservoir will be referred to as “realizations”.

The realizations are used as an input for a simulator in order to compute the production responses of the model or set of models i.e. realizations. The simulator is a tool, commercial

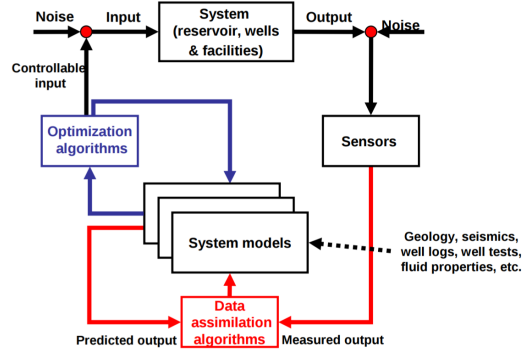


Figure 1.1: Illustration of the closed-loop reservoir management framework. Taken from (Jansen et al., 2009)

or academic, that contains a set of governing equations which describe fluid flow in the reservoir. The reservoir simulator used for running the experiments presented in this thesis is an in-house research oriented one. It was developed by R. Moraes based on the theoretical framework described in (Moraes et al., 2016), (Moraes et al., 2017).

In order to create a proactive management strategy for the whole extraction process Closed-Loop Reservoir Management (CLRM) was introduced (Jansen et al., 2005). It is inspired from control theory, meteorology and oceanography disciplines and it uses a combination of model-based optimization and data assimilation in order to maximize the performance of the reservoir with respect to hydrocarbon recovery or other economic measures (Jansen et al., 2005). There are two inner loops involved in this process as depicted in **Fig. 1.1** taken from (Jansen et al., 2009). The red loop, known as computer assisted history matching, uses the production data from the sensors to make changes to the parameters/properties in the reservoir models in order to match the production data, improving this way the representations of the reservoir thereby improving the predictability of the models. The second loop, known as the optimization loop and coloured in blue one in **Fig. 1.1**, searches for optimal control strategies using the modified reservoir realizations as initial conditions. The present thesis will focus only on the optimization loop.

For the optimization loop, CLRM can use both gradient-based methods, as well as gradient-free methods. In (Hou et al., 2015) these options were compared and it was concluded that gradient-free methods require many more simulations in comparison to the gradient-based methods. Considering that running the simulations represents the most computationally demanding step in the optimization process, it can be concluded that the gradient-based methods represent the most viable solution of the two. To compute the most accurate gradient efficiently the adjoint method would be the ideal solution (Jansen, 2011).

The adjoint method however is not available on most of the commercial simulators, due to the intricate implementation that would be required, especially when dealing with complex physics or with different control types such as bottom hole pressures, water injection rates, oil production rates, etc. The adjoint method has to be implemented in the simulator code which is not openly available for the commercial simulators. Consequently, this led to the development of the stochastic gradient-based optimization methods (Lorentzen et al., 2006) which treat the simulator as a black box and approximate the gradient by performing many forward simulations. One of these methods is Ensemble Optimization (EnOpt) introduced by (Chen, 2008b). To optimize the objective function EnOpt uses a set of perturbed control vectors and their associated objective functions and approximates the gradient using linear regression. EnOpt can be used for a single realization, i.e. deterministic optimization, or to better account for the geological uncertainty an ensemble of equiprobable geological realizations it is used, i.e. robust optimization, (Van Essen et al., 2009). Stochastic Simplex Approximate Gradient (StoSAG) method (Fonseca et al., 2014) is an improved version of EnOpt for robust optimization, which in theory produces higher-quality gradients. One of the most attractive features of StoSAG is its computational efficiency. In this thesis we will primarily focus on robust optimization.

Reservoir models are usually of large ($10^4 - 10^6$) dimension coupled with complex physical processes are usually time consuming to simulate. When accounting for uncertainty, i.e. working with an ensemble of models, each model realization has to be simulated. This translates into an increase in the total number of simulations needed and as mentioned above, the most time consuming operations during the optimization process are the reservoir simulations, so any speedup in the forward simulations is welcomed. Reduced-Order Modeling (ROM) (Kaleta et al., 2011) amongst others have shown the viability of using lower dimensional representations of the reservoir model to achieve significant speedup in simulation time. However this improved computational efficiency is achieved with the cost of oversimplifying the physics. An alternative solution is the Multiscale method (MS) (Hou and Wu, 1997), (Jenny et al., 2003) which solves the system at coarser scale while preserving the fine-scale representation, managing this way to speed up the process whilst taking into account the physics. In order to preserve the fine-scale representation on the coarser system, MS uses a set of basis functions (BF) which are solutions to local problems computed before the actual simulation starts and which rarely need to be recomputed locally. Due to the localization assumptions used when building the basis functions, the solution of the MS scheme is not as accurate as the fine-scale solution. However, these differences can be reduced if an iterative scheme like i-MSFV (iterative Multiscale Finite Volume) is employed (Hajibeygi et al., 2008), (Hajibeygi and Jenny, 2011). MS methods have been demonstrated to be an efficient and accurate technique for reservoir simulation (Tene et al., 2016).

Research Question

The main objective of the thesis is to "investigate the applicability of Multiscale Reservoir Simulation techniques in conjunction with approximate gradient techniques such as StoSAG to achieve optimized life-cycle reservoir management strategies which incorporate geological uncertainty".

Approach and Hypothesis

In order to answer the research question mentioned above, in this thesis we have proposed the MS-StoSAG workflow. The workflow uses i-MSFV as the forward simulation strategy and StoSAG to compute the gradient of the objective function. The hypothesis is that the MS-StoSAG workflow provides a computational speedup and reliable control strategies for the reservoir management, despite using approximations of the fine-scale models for the forward simulations and an approximate gradient. The main focus is on robust optimization strategies since they maximize the objective function over an ensemble of realizations.

This thesis is structured into seven chapters, including the introduction. In the second chapter, some properties of a reservoir are presented along with the equations that describe the flow in porous media. The discretization of these equations is then used to exemplify how the model is used inside a simulator. In the third chapter, the Multiscale Finite Volume method and the iterative method i-MSFV are presented. The fourth chapter gives an introduction on Ensemble Optimization by presenting the standard formulation of EnOpt and improved formulations e.g. StoSAG. In the fifth chapter we present the MS-StoSAG workflow and discuss the various components which make up the workflow as well the general settings used for the efficient use of this workflow for the numerical experiments. To test the research question, a series of numerical experiments were conducted using two synthetic reservoir models of different dimension whose results are presented in the sixth chapter. The chapter also provides results into a sensitivity analysis which investigates the impact of MS parameters such as coarsening ratio and heterogeneity contrast on the quality of the gradient provided by the MS-StoSAG workflow. In the final chapter the outcome and conclusions of the experiments are presented. This also includes some possible future directions in which the project can grow.

2

Flow in porous media

To understand the flow in porous media we first outline a few of the basic properties of the rocks and fluids within a reservoir. Secondly, starting from the law of mass conservation, we define the pressure and transport equations. Our main interest is the two-phase flow. The equations are non-linear and do not have an analytical solution, so they need to be solved numerically. Thirdly, we compare two classic numerical methods used in reservoir simulation to assess their similarities and differences. Lastly, we discretize the governing equations of flow using one of the numerical methods presented.

2.1. Fundamental Reservoir Properties & Concepts

This section highlights a few of the fundamental concepts related to reservoir rock and fluid properties. If a property is independent of the location inside the reservoir, then the rock is referred to as homogeneous with respect to that property. Otherwise, if it varies with location, it is heterogeneous. A reservoir is isotropic with respect to a property, if the property is independent of the direction within the medium. Else, if the property varies with respect to the direction, it is said to be an anisotropic medium.

2.1.1. Porosity

Fluid bearing reservoir rocks are composed of grains in between which exists spaces where the fluids reside. These spaces are known as pore spaces. For a hydrocarbon bearing reservoir the hydrocarbons reside in the pores of a rock. The fraction of the rock that consists of the pore space is called porosity. This means that it is a dimensionless unit and it is usually denoted by ϕ .

2.1.2. Permeability

Permeability of the rock, sometimes referred to as absolute permeability, represents the capacity of a rock to conduct fluids through its interconnected pores. It is commonly indicated by k and it is measured in length square (area), therefore in SI it is expressed in m^2 , but in reservoir engineering Darcy units are used: $1D = 9.869233 \times 10^{-13} \approx 10^{-12} m^2$.

2.1.3. Viscosity

Viscosity is a measure of a fluid's internal resistance to flow. In the SI units it is expressed in Pascal seconds (Pa s) which is called the Poiseuille (Pl) after Jean Léonard Marie Poiseuille. Most common, the cgs unit is used, centipoise (cP) = 10^{-3} Pl.

Generally, the water viscosity is twice the oil viscosity, but this is affected by a series of factors like the oil composition, pressure, temperature, etc.

2.1.4. Saturation

The saturation of a phase (water or oil) is the fraction of the pore space that it occupies. The common notation is S followed by subscript that corresponds to the phase. The saturations

should sum up to 1, so for two-phase flow (2.1) has to be satisfied.

$$S_w + S_o = 1 \quad (2.1)$$

The residual saturation represents the amount of the phase (fraction of pore space) that is trapped or is irreducible. For example, during water flooding, oil will be displaced, but there will still be a percentage that can not be moved and it is known as residual oil saturation S_{or} . This is an equivalent of irreducible water or connate water saturation S_{wc} .

2.1.5. Relative Permeability

The relative permeability describes the behavior of a phase in the presence of another (Jansen, 2013b). The relative permeabilities to the water and oil phases are denoted by k_{rw} , respectively k_{ro} . These values are usually determined from laboratory experiments on core samples of rocks that were extracted while drilling the wells. Water is pushed through a cylindrical piece of rock initially saturated with oil. The values obtained are typically provided in the form of tables or mathematical expressions with parameters that fit the experimental results. In literature, several such expressions are described, but the one used in this thesis is the so-called Corey model given by

$$k_{rw} = k_{rw}^0 S^{n_w} \quad (2.2)$$

$$k_{ro} = k_{ro}^0 (1 - S)^{n_o} \quad (2.3)$$

where k_{rw}^0 and k_{ro}^0 are the end-point relative permeabilities, n_w and n_o are the Corey exponents and S represents a scaled saturation defined as

$$S := \frac{S_w - S_{wc}}{1 - S_{or} - S_{wc}} \quad , \quad 0 \leq S \leq 1 \quad (2.4)$$

2.2. Governing Equations

The mathematical formulations that describe the flow and transport in a porous medium are governed by the mass conservation equation. The mass balance equation for a cell, group of cells and even for the whole reservoir follows the logic:

$$\text{mass rate in} - \text{mass rate out} + \frac{\text{mass accumulated}}{\text{unit time}} + \text{source term} = 0 \quad (2.5)$$

For two-phase immiscible flow we can write the mass conservation equation for each phase:

$$\nabla \cdot (\rho_w \mathbf{v}_w) + \frac{\partial(\rho_w \phi S_w)}{\partial t} - \rho_w q_w = 0 \quad (2.6)$$

$$\nabla \cdot (\rho_o \mathbf{v}_o) + \frac{\partial(\rho_o \phi S_o)}{\partial t} - \rho_o q_o = 0 \quad (2.7)$$

where subscripts w and o are used to identify the phases, water and oil. In (2.6) and (2.7) $\nabla \cdot$ is the divergence operator, ρ is the density, $\mathbf{v}$ is the volumetric flow rate, t is time and q is the source term. Positive values of q correspond to injection, while negative values to production.

The flow through permeable media is usually expressed through Darcy's Law which was experimentally determined by Henry Darcy and it is commonly used as a replacement to the momentum conservation equation (Aziz and Settari, 1979). In this thesis we will neglect the effects of the gravity and capillary forces, so the simplified version of Darcy's Law states that the volumetric flow rate is proportional to the gradient of the local fluid pressure (Jansen, 2013b):

$$\mathbf{v}_w = -\frac{k_{rw}}{\mu_w} \mathbf{K} \cdot \nabla p \quad (2.8)$$

$$\mathbf{v}_o = -\frac{k_{ro}}{\mu_o} \mathbf{K} \cdot \nabla p \quad (2.9)$$

where μ is the fluid viscosity, $\mathbf{K}$ is the permeability tensor and ∇p is the pressure gradient. Note that in the absence of a difference in pressure, no flow will occur. If there is a pressure gradient, the fluid will flow from areas with high pressure towards the ones with low pressure.

We define the total velocity

$$\mathbf{v}_t := \mathbf{v}_w + \mathbf{v}_o \quad (2.10)$$

and the sum of the source terms for all phases

$$q_t := q_w + q_o \quad (2.11)$$

If we divide each of the equations (2.6), (2.7) by the density of the corresponding phase, add them up and apply (2.1), we obtain the flow equation

$$\nabla \cdot \mathbf{v}_t = q_t \quad (2.12)$$

The mobility of a phase is defined as the ratio of the absolute and relative permeabilities and viscosity of that phase. This means that the mobilities of water and oil phases are defined as (Jansen, 2013b)

$$\lambda_w := \frac{k_{rw}(S_w)}{\mu_w} \quad (2.13)$$

$$\lambda_o := \frac{k_{ro}(S_w)}{\mu_o} \quad (2.14)$$

while the total mobility is

$$\lambda_t := \lambda_w + \lambda_o \quad (2.15)$$

We can now rewrite (2.12) as

$$-\nabla \cdot (\lambda_t \mathbf{K} \cdot \nabla p) = q_t \quad (2.16)$$

and by defining fractional flow (Jansen, 2013b)

$$f_w := \frac{v_w}{v_w + v_o} = \frac{\lambda_w}{\lambda_w + \lambda_o} \quad (2.17)$$

we can finally write the transport equation

$$\frac{\partial(\phi S_w)}{\partial t} + \nabla \cdot (f_w \mathbf{v}_t) = f_w q_t \quad (2.18)$$

For two-phase waterflooding problems the compressibility of the phases does not play a major role, therefore it can be neglected (Hajibeygi, 2011). The accumulation term in (2.5) is zero for incompressible flow and the densities ρ_w , ρ_o and porosity ϕ are constant.

2.3. Numerical simulation strategies

The governing equations described in the previous section do not have a general analytical solution. Therefore, they need to be solved using a numerical method. There are several numerical schemes used in reservoir simulators, but the Fully Implicit Method (FIM) is the most commonly used one. In FIM the equations for pressure and saturation are coupled and simultaneously solved at each time step. The terms are treated implicitly, so the parameters are dependent on each other at the same time step. The main advantage of this method is the unconditional stability, meaning that it allows large time-steps. However, the computational cost per time-step is considerable.

An alternative scheme is Implicit Pressure Explicit Saturation (IMPES) which first solves the pressure equation implicitly and then the saturation dependent coefficients are explicitly computed based on the old time-step values (Aziz and Settari, 1979). IMPES is computationally more attractive than FIM, but in some cases it can lead to less accurate results. This can happen in the regions with a strong coupling between pressure and saturation as it often occurs in near-well bore (Moraes et al., 2017). The transport errors are highly dependent on the mass conservation errors of the velocity field, so for the solution

of the pressure equation a discretization technique which provides mass conservative velocity fields is required (Hajibeygi, 2011). IMPES requires small saturation time-steps to maintain numerical stability, but this leads to a lower numerical diffusion which typically increases the accuracy of the results (Aziz and Settari, 1979).

Selecting one of the two numerical methods to be used in a reservoir simulator is not a trivial task, since both methods have their advantages and disadvantages. Using an iterative method like Newton-Raphson to solve the nonlinear equations can help both FIM and IMPES achieve the same solution accuracy. As a summary of this section, we compare the two numerical methods in **Table 2.1**.

Table 2.1: Comparison between FIM and IMPES methods.

	FIM	IMPES
Stability	More	Less
Memory	More	Less
CPU time/step	More	Less
Step size	Long	Short
Numerical diffusion	More	Less
Total CPU time	Case dependent	Case dependent

2.4. Discretization

As it was already mentioned, the PDEs that describe the flow in porous media do not have general analytical solutions, therefore, they need to be solved using numerical methods. In this section we use the two-point flux-approximation (TPFA) finite-volume discretization in space and IMPES discretization in time for the governing equations of 2D immiscible and incompressible flow. The advantage of using a finite-volume discretization is that it is conservative by design.

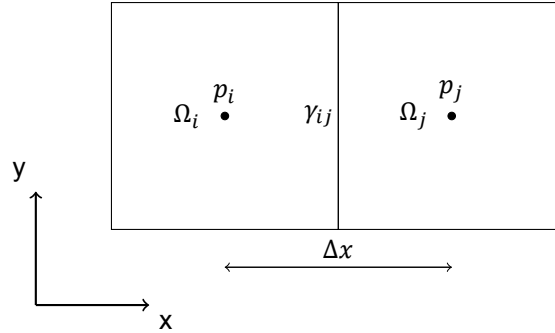


Figure 2.1: Two adjacent control volumes used to define the TPFA discretization

We will assume no-flow boundary condition for the reservoir Ω and will divide it into a set of finite control volumes Ω_k . By finite volume discretization on domain Ω_i we derive the mass-balance equation starting from (2.16):

$$\int_{\Omega_i} -\nabla \cdot (\mathbf{K} \lambda_t \nabla p) dx = \int_{\Omega_i} q_t dx \quad (2.19)$$

The interface between two neighboring control volumes Ω_i and Ω_j is denoted by γ_{ij} . It has area A_{ij} and normal vector n_{ij} . Now, we will assume that $v_t = \lambda_t \nabla p$ is smooth enough so (2.19) holds and apply the divergence theorem

$$\int_{\gamma_{ij}} -\mathbf{K} \lambda_t \nabla p \cdot n_{ij} ds = \int_{\Omega_i} q_t dx \quad (2.20)$$

Using the two-point flux approximation we write the flux across γ_{ij} as

$$\mathbf{v}_{ij} = - \int_{\gamma_{ij}} \mathbf{K} \lambda_t \nabla p \cdot \mathbf{n}_{ij} ds \quad (2.21)$$

We compute the absolute permeabilities using the harmonic mean (2.22) and we use an upwind scheme for the interfacial fluid properties, like pressure gradient, fractional flows and mobilities:

$$k_{ij} = \frac{2}{\frac{1}{k_i} + \frac{1}{k_j}} \quad (2.22)$$

If we define the transmissibilities across the interface γ_{ij} as

$$T_{ij} := \frac{2A_{ij}}{\Delta x} \cdot \left(\frac{1}{\lambda_i} + \frac{1}{\lambda_j} \right)^{-1} \quad (2.23)$$

we can now write the sum over all control volumes neighboring Ω_i

$$\sum_k T_{ik}(p_i - p_k) = \int_{\Omega_i} q_t dx \quad (2.24)$$

To summarize, we can write (2.16) for 2D incompressible flow in matrix form as

$$\mathbf{T}^{n-1} \mathbf{p}^n = \mathbf{q}^n \quad (2.25)$$

where n represents the time-step and $\mathbf{T}$ is the transmissibility matrix defined as (Lunde, 2007)

$$\mathbf{T}_{ij} = \begin{cases} \sum_k T_{ik} & \text{if } i = j \\ -T_{ij} & \text{if } \Omega_i \cap \Omega_j \neq \emptyset \\ 0 & \text{otherwise} \end{cases} \quad (2.26)$$

In conclusion, $\mathbf{T}$ is a penta-diagonal matrix for 2D problems and it is tri-diagonal for 1D. Note that the transmissibility matrix is singular, so (2.25) cannot be solved directly. To make $\mathbf{T}$ non-singular, the pressure for at least one grid block should be prescribed or well models should be used (Jansen, 2013b).

The discretization of the saturation equation (2.18) can be written as:

$$\mathbf{s}^n = \mathbf{s}^{n-1} - \mathbf{F}^{n-1}(\mathbf{v}_t^n - \mathbf{q}_t^n) \quad (2.27)$$

$\mathbf{F}$ is a diagonal matrix with elements

$$\mathbf{F}_{ii} = \frac{\Delta t}{\phi_i V_{\Omega_i}} f_{w_i} \quad (2.28)$$

where V_{Ω_i} is the volume of the domain Ω_i . Note that the fractional flow is evaluated at the previous time-step. From (2.27) we can also see that it is needed to know the initial saturation of the reservoir to be able to solve the transport problem.

To summarize the IMPES method, we assume p, v, S_w are known at time t . To simulate for the next time-step $t + \Delta t$ we first assume S_w is fixed and λ, f_w and f_o become functions of space. Pressure is updated by solving (2.25) then the velocity v is computed using Darcy's Law. The saturation is explicitly computed using (2.27) assuming that v and p are constant. These steps are repeated for every new time-step.

The size of Δt is restricted by the Courant–Friedrichs–Lewy (CFL) condition of the explicit scheme (Coats et al., 2003). The pressure equation is elliptic and it describes smooth solutions since the changes in pressure are instantly propagated through the whole reservoir (Lie, 2016). In theory, large time-steps can be used for this equation. There are two possible strategies for time-stepping in this situation: The first one is to use synchronous time-steps, the same Δt for both equations, but these time increments have to be small enough to satisfy

the CFL conditions for the saturation equation. The other strategy is using an asynchronous time-stepping strategy (Chen et al., 2004), by using large time-steps Δt for the pressure equation, and then while maintaining the pressure solution and velocity field constant, the transport problem is solved many times using smaller time-steps (substeps) until the sum of the substeps is equal to Δt :

$$\Delta t = \sum_k \Delta t_{s_k} \quad (2.29)$$

In conclusion, both FIM and IMPES are important discretization strategies that can be used for real-life applications. In this chapter we have mainly focused on IMPES since it serves as a starting point for the Multiscale method presented in the following chapter of this thesis.

Multiscale Finite Volume Method

Solving very large systems, like the ones used in reservoir simulation, in full detail is computationally demanding. This can be even more problematic when many simulations are needed, like for example during optimization. This is why there is an ongoing interest in approximate simulation techniques that can provide a good accuracy-computational efficiency trade-off. Reduced Order Models (ROM) (Kaleta et al., 2011) can speed-up the computation with the cost of oversimplifying the physics. An alternative is to use Multiscale methods (Hou and Wu, 1997), (Jenny et al., 2003) which constructs and solves a coarse-scale representation of the problem, however is able to accurately reconstruct the solution back in its original fine-scale representation. The Multiscale Finite Volume Method (MSFV) provides a mass conservative approximate solution of the original fine-scale problem. The differences between the fine-scale solution and the approximate MS one can be reduced if an iterative scheme, e.g. the Iterative Multiscale Finite Volume Method (i-MSFV) is used (Hajibeygi et al., 2008). In this chapter we present the MSFV for two-phase incompressible flow on 2D grids and the i-MSFV method.

The Multiscale method approximates the fine-scale solution by deconstructing it into a series of local problems. The numerical solutions of the local functions are called basis functions and they are used to construct a conservative (finite volume) coarse-scale system. The solution of this system is then mapped onto the original fine-scale grid. The method can be summarized as follows (Jenny et al., 2005):

1. Preprocessing:
 - 1.1. construct the primal and dual-coarse grids
 - 1.2. construct the restriction operator
 - 1.3. compute the basis functions
 - 1.4. assemble the prolongation operator based on basis functions
2. Simulation
 - 2.1. Restriction step: construct the coarse scale system
 - 2.2. Solving step: compute pressure on the coarse grid
 - 2.3. Prolongation step: reconstruct the pressure for the fine grid using the pressure computed in the previous step
 - 2.4. Construct the fine-scale conservative velocity field (for multi-phase problems only)

3.1. Preprocessing

The MSFV method relies on a primal coarse grid and on a dual coarse grid. For the former each of the cells is defined as a control volume $\check{\Omega}_k$ ($k \in [1, N_c]$) which contains a subdivision

of the fine underlying grid. One fine cell from each subdivision (typically the central one) is defined as the coarse node or primal center node or vertex. The dual coarse grid is composed of $\tilde{\Omega}_k$ cells ($k \in [1, N_d]$) defined by the cells connecting the coarse nodes **Fig. 3.1**. For 2D problems, the fine cells are divided into three categories Wang et al. (2014): interior (white), edge (blue) and vertex (red) cells.

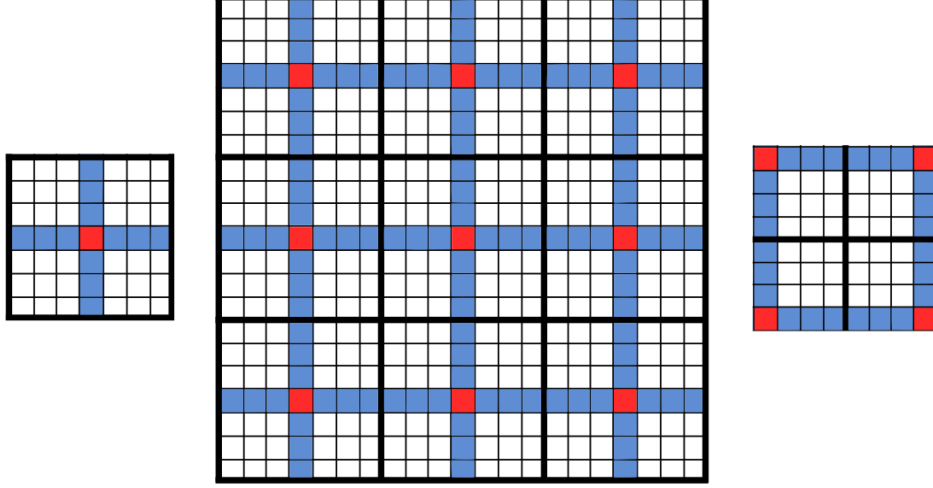


Figure 3.1: Control volume $\tilde{\Omega}_k$ (left); Primal and dual grids overlaying the fine scale domain (center); $\tilde{\Omega}_k$ cell of the dual coarse grid (right)

MSFV uses the basis functions to derive a multi-point stencil for solving the flow equation on a coarse grid. It is accurate because it incorporates the fine scale model features. The basis functions are computed locally, at the dual cell domains, as numerical solutions of equation 2.25 on $\tilde{\Omega}_k$ without the right hand side (RHS):

$$\nabla \cdot (\lambda \nabla \phi_k^i) = 0 \quad \text{on } \tilde{\Omega}_k \quad (3.1)$$

where ϕ_k^i denotes the basis function associated with the coarse node i in the dual coarse block $\tilde{\Omega}_k$. The basis functions are defined to be zero outside each dual cell (local support).

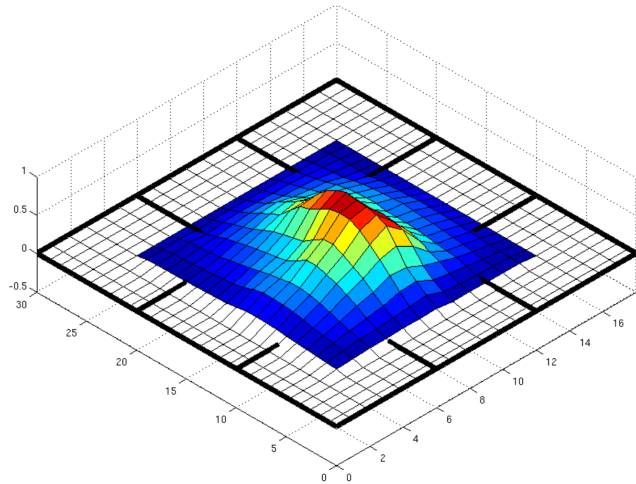


Figure 3.2: Representation of a basis function. Figure taken from (Aarnes et al., 2009)

The basis function for each point x_k is continuous **Fig. 3.2** when summing over the dual

coarse blocks that share that point

$$\phi^k = \sum_i \phi_i^k \quad (3.2)$$

In essence, for each dual coarse grid block the same system is solved using a different right hand side. For each corner point, the right hand side has the value one in a single cell corresponding to that corner and zero for the others. These solutions, see **Fig. 3.2**, are then used to assemble the basis function for each primary grid center point. The basis functions are computed only once, as a preprocessing step, and they only need to be updated if the local properties within the specific dual block are changed (Jenny et al., 2005).

The original MSFV method does not model sources at the subgrid scale, as basis functions do not account for RHS terms. To counteract this, the "well functions" were introduced (Jenny and Lunati, 2009). Well functions are computed similarly to the basis functions, the flow problem, eq. (2.25), is solved considering unity solution at the well location (i.e. $p_w = 1$). The dimensions and shape of the restriction and prolongation operators have to be changed accordingly.

These basis functions are used to construct the prolongation operator $\mathbf{P}$. Each column of $\mathbf{P}$ will hold the basis function for one of the nodes.

$$\mathbf{P} = [\phi^1 \phi^2 \dots \phi^{N_c}] \quad (3.3)$$

$\mathbf{P}$ has size $N_f \times N_c$, but each well function should be added as a separate column to the prolongation operator. To test if the prolongation operator is properly constructed, one can check if the sum of the elements on each row is 1 (partition of unity):

$$\sum_j \mathbf{P}(i, j) = 1 \quad \forall i \in [1, N_f] \quad (3.4)$$

The other operator that is built during the preprocessing step is the restriction operator $\mathbf{R}$. In the MSFV method, it represents the sum of the fine-scale conservation equations over the primal-coarse grid blocks. In this case, it does not depend on the basis functions and hence it can be built in parallel to any other preprocessing step. $\mathbf{R}$ has size $N_c \times N_f$ and it is defined as (Wang et al., 2014), (Tene et al., 2016):

$$\mathbf{R}(i, j) = \begin{cases} 1 & \text{if the fine-cell } j \text{ is contained in the primal-coarse grid block } i \\ 0 & \text{otherwise} \end{cases} \quad (3.5)$$

If well functions are used, then $\mathbf{R}$ will have one more row for each of the wells. They will contain a single value of 1 in the location of the well. This also means that in the first N_c rows the wells should not be considered as being part of the dual-coarse blocks. As a test, the sum over each column of $\mathbf{R}$ should be equal to 1. This matrix is constructed once and it does not change during the simulation.

3.2. Simulation - Discretized MSFV

We rewrite the discretized equation for incompressible flow (2.25) using the subscript f to denote the fine-scale reference problem

$$\mathbf{T} \mathbf{p}_f = \mathbf{q}_f \quad (3.6)$$

where $\mathbf{p}_f$ is the pressure field on the fine grid and $\mathbf{q}_f$ is the source term vector for the same problem.

Multiscale approximates the fine-scale pressure $\mathbf{p}_f$ by $\mathbf{p}'$ which is a mapping of the coarse scale solution $\mathbf{p}_c$ using the prolongation operator $\mathbf{P}$

$$\mathbf{p}_f \approx \mathbf{p}' = \mathbf{P} \mathbf{p}_c \quad (3.7)$$

Substituting eq. (3.7) into eq. (3.6) gives us

$$\mathbf{T} \mathbf{P} \mathbf{p}_c = \mathbf{q}' \quad (3.8)$$

It is obvious that eq. (3.8) is an overdetermined system of equations. In order to reduce it to a square system (size $N_c + \text{number of wells}$) it can be multiplied to the left by the restriction operator $\mathbf{R}$.

$$\mathbf{R} \mathbf{T} \mathbf{P} \mathbf{p}_c = \mathbf{R} \mathbf{q}' \quad (3.9)$$

In the restriction step of MSFV the coarse-scale system is built using the prolongation and restriction operators. $\mathbf{T}_c = \mathbf{R} \mathbf{T} \mathbf{P}$ and $\mathbf{q}_c = \mathbf{R} \mathbf{q}'$

$$\mathbf{p}_c = (\mathbf{R} \mathbf{T} \mathbf{P})^{-1} \mathbf{R} \mathbf{q} \quad (3.10)$$

The inverse in (3.10) is never computed. $\mathbf{p}_c$ is obtained by solving the linear system $\mathbf{T}_c \mathbf{p}_c = \mathbf{q}_c$.

$$\mathbf{p}' = \mathbf{P}(\mathbf{R} \mathbf{T} \mathbf{P})^{-1} \mathbf{R} \mathbf{q} \quad (3.11)$$

Finally, MSFV is mapping $\mathbf{p}_c$ to fine-scale using (3.7).

3.3. Multi-phase flow MSFV

In the context of this work, an Implicit Pressure Explicit Saturation (IMPES) strategy (see chapter 2.3) is employed. This means that flow and transport are solved sequentially, and that the dependency of the flow equation on saturation is lagged in time.

To honor the mass balance for multi-phase problems it is required to have a conservative fine-scale velocity field. The MSFV solution is conservative at the coarse scale, i.e. for each coarse volume $\check{\Omega}_k$, but the fine-scale solution $\mathbf{v}'$ is non-conservative due to the discontinuities at the volume interfaces of the dual grid (Jenny et al., 2005).

$$\mathbf{v}' = -\lambda \cdot \nabla \mathbf{p}' \quad (3.12)$$

To solve saturation transport in fine scale, a conservative fine-scale velocity field $\mathbf{v}''$ that is consistent with $\mathbf{v}'$ is needed. To compute it, an additional local Neumann problem needs to be solved (Jenny et al., 2005):

$$-\nabla \cdot (\lambda \cdot \nabla \mathbf{p}_k'') = \mathbf{q} \quad \text{on} \quad \check{\Omega}_k \quad (3.13)$$

$$(\lambda \cdot \nabla \mathbf{p}_k'') \cdot \bar{\mathbf{n}}_k = (\lambda \cdot \nabla \mathbf{p}') \cdot \bar{\mathbf{n}}_k \quad \text{on} \quad \partial \check{\Omega}_k \quad (3.14)$$

Using (3.12) and (3.13) the conservative fine-scale velocity field $\mathbf{v}''$ is computed (Hajibeygi et al., 2008):

$$\mathbf{v}'' = \begin{cases} -\lambda \cdot \nabla \mathbf{p}_k'' & \text{on} \quad \check{\Omega}_k \\ -\lambda \cdot \nabla \mathbf{p}' & \text{on} \quad \partial \check{\Omega}_k \end{cases} \quad (3.15)$$

Then the hyperbolic saturation equation (2.18) is integrated explicitly (Jenny et al., 2005) for the current time step using the updated flow field. Note that when computing the flow field, the total mobility λ is lagged one time-step. For two-phase flow computing the saturation of one phase is enough thanks to (2.1)

Since the mobility λ depends on saturation, the basis functions have to be recomputed for those local domains where mobility changes fall beyond a threshold value. This can be tested with the following condition (Jenny et al., 2005):

$$\frac{1}{1 + \epsilon_\lambda} < \frac{\lambda^n}{\lambda^{n-1}} < 1 + \epsilon_\lambda \quad (3.16)$$

where $\epsilon_\lambda > 0$ is a user defined threshold and the superscripts n and $n - 1$ denote the previous two time steps.

3.4. i-MSFV

The differences between the pressure computed using MSFV $\mathbf{p}'$ and the fine scale solution $\mathbf{p}_f$ come from the localization assumptions used to compute the basis functions (Hajibeygi et al., 2008). An iterative scheme can compensate for these discrepancies. Iterative Multiscale Finite Volume method (i-MSFV) (Hajibeygi and Jenny, 2011) is able to reduce the high frequency errors via some smoothing iterations on the fine scale solution, while the low frequency errors are solved by the MSFV coarse scale solution. In other words, MSFV is rewritten in residual form and iteratively solves the resulting system of equations until the convergence criterion is satisfied (Hajibeygi et al., 2008). Being an iterative method, i-MSFV is used for maintaining the desired user-defined accuracy. It is also possible to set up a maximum number of iterations since the coarse-scale solutions of i-MSFV are conservative after any iteration (Hajibeygi and Jenny, 2011). Numerical experiments show that on average only a few iterations are necessary to improve the MSFV results significantly (Hajibeygi, 2011). In (Tene et al., 2016) it is shown that convergence success rate of 100% can be obtained by iterative smoothing.

The i-MSFV algorithm can be summarized as follows:

```

while  $\|\mathbf{r}^v\| > \epsilon$ : do
    Adaptively compute the basis functions: check (3.16) and recompute basis
        functions if needed
    Multiscale stage: restriction, solve for  $\mathbf{p}_c^v$ , prolongate to fine scale  $\mathbf{p}'^v$ 
    Smoothing stage: smooth  $\mathbf{p}'^v$  until error below the threshold  $\epsilon_{sm}$ 
    Update: solution  $\mathbf{p}'^{v+1}$  and residual  $\mathbf{r}^{v+1} = \mathbf{q}^v - \mathbf{T}\mathbf{p}'^{v+1}$ ;  $\mathbf{p}'^v \leftarrow \mathbf{p}'^{v+1}$  and  $\mathbf{r}^v \leftarrow \mathbf{r}^{v+1}$ 
end

```

Algorithm 1: The i-MSFV algorithm

where the subscripts v and $v + 1$ represent the previous and current Newton–Raphson iteration, r is the residual, ϵ and ϵ_{sm} are user defined tolerances, $\mathbf{p}_c$ is the pressure of the coarse scale system and $\mathbf{p}'$ is the reconstructed fine-scale solution.

In conclusion, i-MSFV improves on the MSFV method by iteratively reducing the errors caused by the localization assumptions and by allowing the user to define a threshold for the accuracy of the solution.

Ensemble optimization

When dealing with any optimization problem, one should define the objective function that is going to be either maximized or minimized, depending on the case, as well as the controllable parameters also referred to as inputs or controls. For hydrocarbon production optimization common input parameters are the states of the wells that can be controlled, bottom-hole pressure, rates, states of the Inflow Control Valve (ICVs). The two most common objective functions are ultimate recovery (UR) and net present Value (NPV) (Fonseca, 2015).

As already mentioned, among the gradient-based methods the adjoint method (Jansen, 2011) provides the most accurate gradient and is computationally very efficient. However this method has one notable disadvantage, that it has to be implemented in the simulator code. Most of the commercial simulators do not come with this method by default. This is due to the increasing complexity as the level of details in physics grows, and code would be needed for every control type. As an alternative, stochastic methods are non-intrusive, treating the simulator as a black box. One of these methods is Ensemble-based Optimization (EnOpt), proposed by (Chen, 2008b) and (Chen et al., 2009). It provides an approximate gradient and it is computationally less attractive than the adjoint method, but it is simulator agnostic, meaning that it can work with any simulator.

Note that in order to evaluate the objective function $J(\mathbf{u}, \mathbf{m})$ for a given control vector $\mathbf{u}$ and a geological model $\mathbf{m}$, a complete simulation run is needed, see **Fig. 4.1**.

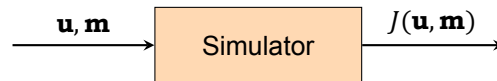


Figure 4.1: Schematic of the process

4.1. Ensemble-Based Deterministic Formulation

In this section the standard formulation of the EnOpt method for a single realization (deterministic) as proposed by (Chen et al., 2009) will be presented. This method uses an ensemble of randomly generated control vectors to compute the gradient, concept that was firstly introduced by (Lorentzen et al., 2006). For the initial step, the control vector $\mathbf{u}$ is chosen by the user. If we divide the total simulation time in control time-steps, intervals with constant settings for the wells, the control vector $\mathbf{u}$ has length N equal to the number of control variables multiplied by number of the control time-steps. A simulation has to be run in order to compute $J(\mathbf{u}, \mathbf{m})$. A multivariate Gaussian distribution ensemble with size M is generated with distribution mean $\mathbf{u}$ and a predefined covariance matrix $\mathbf{C}$. Traditionally $\mathbf{C}$ is kept constant, but it can also be updated during the optimization process as described in (Fonseca, 2015). The ensemble members are generated using

$$\mathbf{u}_i = \mathbf{u} + \mathbf{C}^{1/2} \mathbf{z}_i \quad (4.1)$$

where $\mathbf{C}^{1/2}$ is computed using Cholesky decomposition and $\mathbf{z}_i$ is drawn from a univariate Gaussian distribution. M simulations are needed in order to evaluate $J(\mathbf{u}_i, \mathbf{m})$ for $i = \{1, 2 \dots M\}$. The control anomaly matrix $\mathbf{U}$ holds on each of the columns the difference between each control realization and ensemble mean, so it is assembled as

$$\mathbf{U} = [\mathbf{u}_1 - \bar{\mathbf{u}} \quad \mathbf{u}_2 - \bar{\mathbf{u}} \quad \dots \quad \mathbf{u}_M - \bar{\mathbf{u}}] \quad (4.2)$$

where

$$\bar{\mathbf{u}} = \frac{1}{M} \sum_{i=1}^M \mathbf{u}_i \quad (4.3)$$

and the difference between each corresponding objective function and the mean forms the objective function anomaly

$$\mathbf{j} = [J(\mathbf{u}_1) - \bar{J} \quad J(\mathbf{u}_2) - \bar{J} \quad \dots \quad J(\mathbf{u}_M) - \bar{J}]^T \quad (4.4)$$

where $\bar{J}$ is the average of the objective function

$$\bar{J} = \frac{1}{M} \sum_{i=1}^M J(\mathbf{u}_i) \quad (4.5)$$

The gradient is computed using

$$\mathbf{g} = (\mathbf{U}\mathbf{U}^T)^{-1}\mathbf{U}^T\mathbf{j} = \mathbf{U}^\dagger\mathbf{j} \quad (4.6)$$

where $\dagger$ indicates the Moore-Penrose pseudo inverse which can easily be computed using singular value decomposition (SVD). The control vector for the next iteration can be computed using any gradient based technique, but in this example, the steepest descent algorithm (Nocedal and Wright, 2006) is used

$$\mathbf{u}^{l+1} = \mathbf{u}^l + \alpha \mathbf{g} \quad (4.7)$$

where α is the step size and the superscript l represents the iteration index. A simulation is run using $\mathbf{u}^{l+1}$ and based on the results $J(\mathbf{u}^{l+1}, \mathbf{m})$ is computed. Since it is a maximization problem, if $J(\mathbf{u}^{l+1}, \mathbf{m}) > J(\mathbf{u}^l, \mathbf{m})$, the algorithm continues by generating the new ensemble with distribution mean $\mathbf{u}^{l+1}$ and if not, the step size is reduced $\alpha = \alpha/2$ and $\mathbf{u}^{l+1}$ is recomputed.

According to (Fonseca, 2015), in order to prove that the singly-smoothed search direction is a reasonable approximation of $\mathbf{C}$ times the true gradient $\mathbf{g}$, (Chen, 2008b) had to make three assumptions. The first one, (4.8), is mainly related to the deterministic optimization, while the other two (4.15), (4.18) have an impact on the results of robust optimization.

The first assumption is that the mean of the ensemble corresponds with the mean of the Gaussian distribution:

$$\bar{\mathbf{u}}^l = \frac{1}{N} \sum_{i=1}^N \mathbf{u}_i^l \approx \mathbf{u}^l \quad (4.8)$$

For most of the applications, the degrees of freedom in the control vector $\mathbf{u}$ are bounded. When $\mathbf{u}$ is very close to one of the boundaries the domain of the Gaussian distribution might need to be truncated. This can lead to an average control vector $\bar{\mathbf{u}}$ different than $\mathbf{u}^l$ which would contradict the first assumption (4.8). For the deterministic case the solution proposed by (Do and Reynolds, 2013) is to use the current control vector $\mathbf{u}^l$ and its corresponding objective function value $J(\mathbf{u}^l)$ to calculate the control and objective function anomalies $\mathbf{U}$ and $\mathbf{j}$:

$$\mathbf{U} = [\mathbf{u}_1 - \mathbf{u}^l \quad \mathbf{u}_2 - \mathbf{u}^l \quad \dots \quad \mathbf{u}_M - \mathbf{u}^l] \quad (4.9)$$

$$\mathbf{j} = [J(\mathbf{u}_1) - J(\mathbf{u}^l) \quad J(\mathbf{u}_2) - J(\mathbf{u}^l) \quad \dots \quad J(\mathbf{u}_M) - J(\mathbf{u}^l)] \quad (4.10)$$

4.2. EnOpt – Robust Optimization

In order to compensate for the geological uncertainty in the reservoir model, optimization is performed over an ensemble of equiprobable geological realizations. This methodology is known as ‘robust optimization’ and it was first introduced by (Van Essen et al., 2009), while (Chen, 2008b) and (Chen et al., 2009) introduced the original En-Opt formulation for robust optimization.

We will consider an ensemble of N geological realizations. For the first step an initial guess for the control vector is needed. N simulations will be performed, every realization with the initial guess and the mean objective function is computed. In this particular version, an ensemble of N control vectors are generated similarly to the deterministic case. Then, they are coupled in a 1:1 ratio, each realization with a control vector. (Chen, 2008b) presented an argumentation in favor of this coupling. The alternative of simulating each realization with every control vector would mean a number of N^2 and would make the method computationally expensive. Note that for robust optimization the objective function is the mean of the set $\{J(\mathbf{u}_i, \mathbf{m}_i) | i = 1 \dots N\}$. The control and objective function anomalies are assembled using:

$$\mathbf{U} = [\mathbf{u}_1 - \bar{\mathbf{u}} \quad \mathbf{u}_2 - \bar{\mathbf{u}} \quad \dots \quad \mathbf{u}_N - \bar{\mathbf{u}}] \quad (4.11)$$

where

$$\bar{\mathbf{u}} = \frac{1}{N} \sum_{i=1}^N \mathbf{u}_i \quad (4.12)$$

and

$$\mathbf{j} = [J(\mathbf{u}_1, \mathbf{m}_1) - \bar{J} \quad J(\mathbf{u}_2, \mathbf{m}_2) - \bar{J} \quad \dots \quad J(\mathbf{u}_N, \mathbf{m}_N) - \bar{J}]^T \quad (4.13)$$

with $\bar{J}$ is

$$\bar{J} = \frac{1}{N} \sum_{i=1}^N J(\mathbf{u}_i, \mathbf{m}_i) \quad (4.14)$$

and

The next steps are identical to the deterministic case, the gradient is computed and the control vector is updated. In order to check if the new solution improves the objective function all realizations have to be simulated using the updated control function.

As we mentioned in the previous section, (Chen, 2008b) made two more assumptions, one of which:

$$\overline{J(\mathbf{u}^l, \mathbf{m})} = \frac{1}{N} \sum_{i=1}^N J(\mathbf{u}_i^l, \mathbf{m}_i) \approx J(\bar{\mathbf{u}}^l, \mathbf{m}) \approx J(\mathbf{u}^l, \mathbf{m}) \quad (4.15)$$

which suggests that for any realization $\mathbf{m}_i$

$$J(\mathbf{u}_i^l, \mathbf{m}_i) = \overline{J(\mathbf{u}^l, \mathbf{m})} \quad (4.16)$$

which leads to

$$J(\mathbf{u}^l, \mathbf{m}_1) \approx J(\mathbf{u}^l, \mathbf{m}_2) \approx \dots \approx J(\mathbf{u}^l, \mathbf{m}_N) \quad (4.17)$$

For a valid $\mathbf{u}^l$, (4.17) can only be true if the differences between the models are very small, which is contradicting the main reason for running robust optimization to compensate for the high geological uncertainties.

The third assumption is the first-order Taylor expansion:

$$\begin{aligned} J(\mathbf{u}_k, \bar{\mathbf{m}}_k) &= J(\mathbf{u}^l, \bar{\mathbf{m}}) + \nabla_u J(\mathbf{u}^l, \bar{\mathbf{m}})(\mathbf{u}_k^l - \mathbf{u}^l) + \nabla_m J(\mathbf{u}^l, \bar{\mathbf{m}})(\mathbf{m}_k - \bar{\mathbf{m}}) \\ &+ \frac{1}{2!} [\nabla_u (\nabla_u J(\mathbf{u}^l, \bar{\mathbf{m}}))(\mathbf{u}_k^l - \mathbf{u}^l)^2 + 2 \nabla_m (\nabla_u J(\mathbf{u}^l, \bar{\mathbf{m}}))(\mathbf{u}_k^l - \mathbf{u}^l)(\mathbf{m}_k - \bar{\mathbf{m}}) \\ &+ \nabla_m (\nabla_m J(\mathbf{u}^l, \bar{\mathbf{m}}))(\mathbf{m}_k - \bar{\mathbf{m}})^2] \end{aligned} \quad (4.18)$$

The second order term can be written:

$$\mathbf{e}_{2,k} = [\mathbf{u}_k^l - \mathbf{u}^l \quad \mathbf{m}_k - \tilde{\mathbf{m}}] \mathbf{H}_k^l [\mathbf{u}_k^l - \mathbf{u}^l \quad \mathbf{m}_k - \tilde{\mathbf{m}}]^T \quad (4.19)$$

where $\mathbf{H}_k^l$ denotes the Hessian matrix.

As already pointed out by (Fonseca et al., 2015), the error (4.19) the approximation is of order $\mathcal{O}(\max_i\{(\|\mathbf{u}_i^l - \mathbf{u}^l\|_2 + \|\mathbf{m}_i - \tilde{\mathbf{m}}\|_2)^2\})$ which if the uncertainty is large, $\|\mathbf{m}_i - \tilde{\mathbf{m}}\|_2$ is large, making the overall error large.

4.3. StoSAG

Stochastic Simplex Approximative Gradient (StoSAG) is another variation of EnOpt. It was introduced by (Fonseca et al., 2014) and it improves the standard EnOpt formulation by computing the control anomaly in a similar manner as proposed by (Do and Reynolds, 2013), not relaying on the first assumption (4.8), and by significantly modifying the objective function anomaly (4.21).

The control anomaly is assembled as following:

$$\mathbf{U} = [\mathbf{u}_1 - \mathbf{u}^l \quad \mathbf{u}_2 - \mathbf{u}^l \quad \dots \quad \mathbf{u}_N - \mathbf{u}^l] \quad (4.20)$$

And the objective function anomaly:

$$\mathbf{j} = [J(\mathbf{u}_1, m_1) - J(\mathbf{u}^l, m_1) \quad J(\mathbf{u}_2, m_2) - J(\mathbf{u}^l, m_2) \quad \dots \quad J(\mathbf{u}_N, m_N) - J(\mathbf{u}^l, m_N)]^T \quad (4.21)$$

By subtracting with respect to the individual objective functions in (4.21) the effect of the outliers on the gradient estimation is reduced and the second assumption (4.15) is avoided.

We now write the Taylor expansion of the StoSAG formulation (4.21):

$$\begin{aligned} J(\mathbf{u}_k, \mathbf{m}_k) &= J(\mathbf{u}^l, \mathbf{m}_k) + \nabla_u J(\mathbf{u}^l, \mathbf{m}_k)(\mathbf{u}_k^l - \mathbf{u}^l) + \nabla_m J(\mathbf{u}^l, \mathbf{m}_k)(\mathbf{m}_k - \tilde{\mathbf{m}}) \\ &\quad + \frac{1}{2!} [\nabla_u (\nabla_u J(\mathbf{u}^l, \mathbf{m}_k))(\mathbf{u}_k^l - \mathbf{u}^l)^2 + 2\nabla_m (\nabla_u J(\mathbf{u}^l, \mathbf{m}_k))(\mathbf{u}_k^l - \mathbf{u}^l)(\mathbf{m}_k - \tilde{\mathbf{m}}) \\ &\quad + \nabla_m (\nabla_m J(\mathbf{u}^l, \mathbf{m}_k))(\mathbf{m}_k - \tilde{\mathbf{m}})^2] \end{aligned} \quad (4.22)$$

Similar to the previous case, we can write the second order term:

$$\mathbf{e}_{2,k} = [\mathbf{u}_k^l - \mathbf{u}^l] \mathbf{H}_k^l [\mathbf{u}_k^l - \mathbf{u}^l] \quad (4.23)$$

This time we can see that the error of the approximation of the first-order Taylor expansion for StoSAG is of order $\mathcal{O}(\max_i\{(\|\mathbf{u}_i^l - \mathbf{u}^l\|_2^2)\})$ which goes to zero as $\|\mathbf{u}_i^l - \mathbf{u}^l\|_2^2 \rightarrow 0$ and this is satisfied if $\mathbf{u}_i^l \sim \mathcal{N}(\alpha_l, \mathbf{C})$ where $\alpha_l \rightarrow 0$ as $l \rightarrow 0$.

It is important to note that StoSAG has the same computational complexity as standard EnOpt for generating a search direction at each iteration of the steepest ascent algorithm and it usually performs significantly better (Fonseca et al., 2015) than standard EnOpt (Chen, 2008b) and the modified version (Do and Reynolds, 2013).

5

MS-StoSAG workflow

In this chapter we introduce the MS-StoSAG workflow. It is based on the StoSAG framework (Fonseca et al., 2014) since it was proved to be a reliable stochastic optimization method in many applications, e.g. (Fonseca et al., 2014), (Lu, 2017), (Chen, 2008a). We propose using a Multiscale (MS) simulator that uses the iterative method i-MSFV (Hajibeygi et al., 2008) for evaluating the model responses in order to improve the computational efficiency of the optimization workflow.

The schematic of the MS-StoSAG workflow is presented in **Fig. 5.1**. The workflow requires a control vector as a starting strategy that will then be stochastically perturbed. One can observe in **Fig. 5.1** that the chosen objective function is the Net Present Value (NPV) which is defined in (5.1) (Fonseca, 2015). Once the gradient is computed, it is used to update the control vector in a simple steepest-ascent algorithm (Nocedal and Wright, 2006). The responses of the realizations with the updated control values are evaluated and if the objective function is not improved, then the step-size is reduced by half. The framework is flexible and it allows different objective functions and more sophisticated optimization algorithms.

$$J = \sum_{k=1}^K \frac{[(q_{o,k}) \cdot r_o - (q_{wp,k}) \cdot r_{wp} - (q_{wi,k}) \cdot r_{wi}] \cdot \Delta t_k}{(1 + b)^{\frac{t_k}{\tau_t}}} \quad (5.1)$$

where $q_{o,k}$ represents the oil production rate in m^3/day , $q_{wp,k}$ is the water production rate in m^3/day , $q_{wi,k}$ is the water injection rate in m^3/day , r_o is the price of oil produced in $\$/\text{m}^3$, r_{wp} is the cost of produced water in $\$/\text{m}^3$, r_{wi} is the cost of injected water in $\$/\text{m}^3$, Δt_k is the difference between consecutive time steps in days, b is the discount factor expressed as a fraction per year, t_k is the cumulative time in days corresponding to time step k , and τ_t is the reference time period for discounting, typically one year.

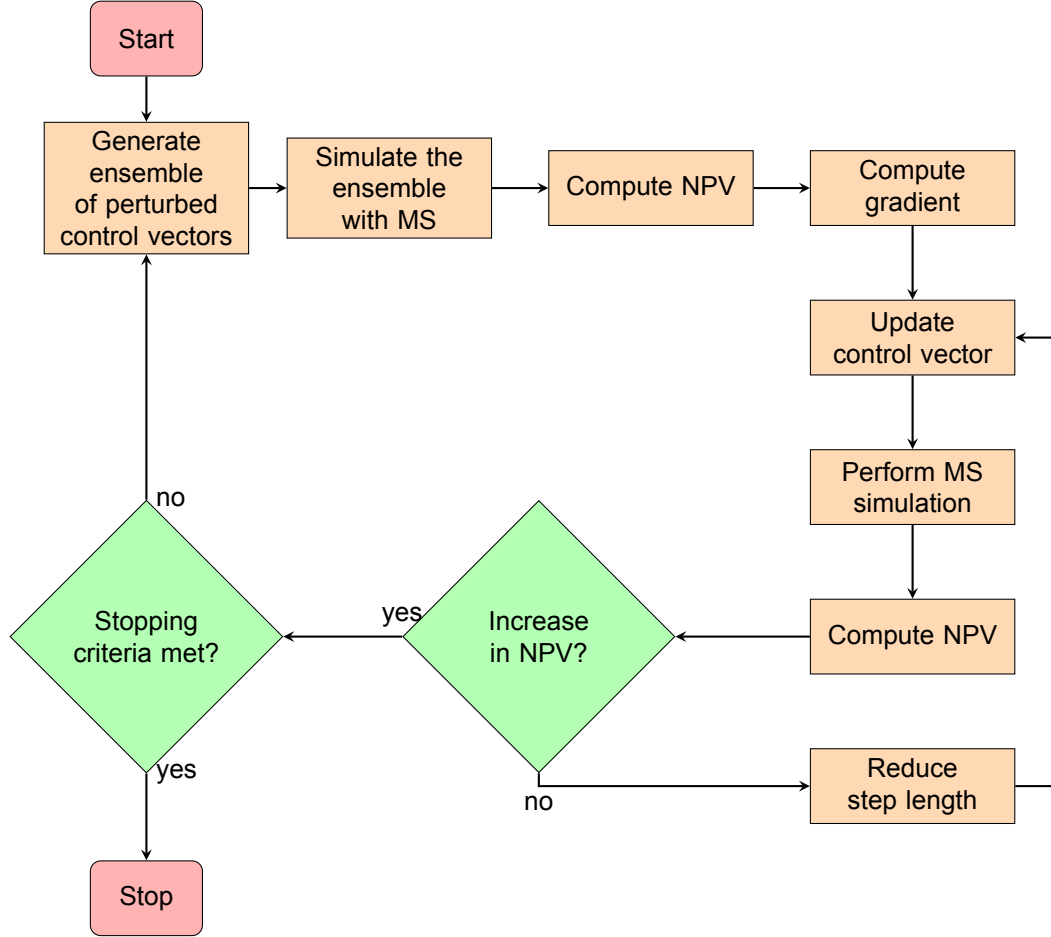


Figure 5.1: The MS-StoSAG optimization workflow.

The MS simulator used was developed by Moraes R. as described in (Moraes et al., 2016) and was benchmarked against other research simulators like MATLAB Reservoir Simulation Toolbox (MRST) (Lie et al., 2012). The simulator uses the iterative Multiscale Finite Volume method (i-MSFV) (Hajibeygi et al., 2008) and the schematic can be found in **Fig. 5.2**. In the workflow, the basis functions (BF) are reconstructed only when a given error metric e exceeds the predefined tolerance E , see (3.16).

The discrepancies that arise due to the localization assumptions are reduced by the i-MSFV smoothing step. The simulator allows for user-defined thresholds to be imposed for the fine scale residual of the multiscale solution and for the residual of the smoothing loop. The residual norm is computed using the L_2 -norm. The simulator uses the stabilized bi-conjugate gradient iterative solver (BiCGSTAB) (van der Vorst, 1992) as fine-scale smoother. Since all Krylov subspace algorithms are often useless without preconditioning (Che, 2006), BiCGSTAB is preconditioned with the dual threshold incomplete LU factorization (ILUT) algorithm (Saad, 1994).

The conservative velocity field is reconstructed directly from the smoothed pressure field. The simulator uses an asynchronous time-stepping strategy (Chen et al., 2004) because the transport solver time-step Δt_s is limited by CFL conditions, while the flow solver time-step Δt_p is not. The pressure and the velocity field are kept unchanged until the transport solver time t_s reaches Δt_p . The simulator runs until the final time t_f is reached and the output is used to compute the objective function.

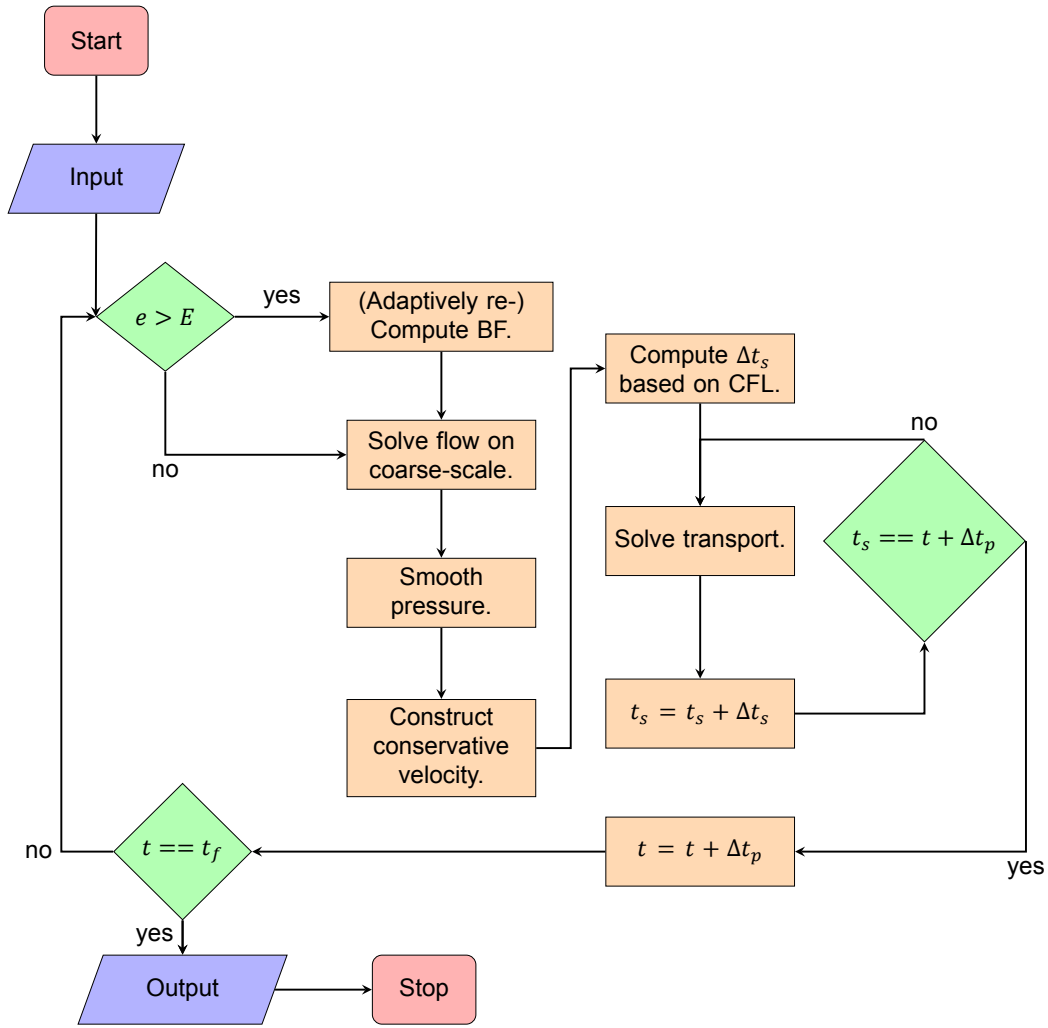


Figure 5.2: Schematic description of the MS reservoir simulation.

We are interested to evaluate if this workflow that uses an approximated forward simulation to compute an approximate gradient can not only provide computational speedup, but reliable results as well. Note that the objective is not to reduce the overall number of optimization iterations needed, but the time required to evaluate the model responses.

6

Numerical experiments

In this chapter we present the results of several optimization experiments conducted using the MS-StoSAG workflow detailed in the previous chapter. We compare these results with the optimization experiments that use a fine-scale solution for the forward simulations.

We use two models with different grid sizes for the reservoir. The first one is the Five-Spot model on which we first perform a deterministic optimization to compare the fully implicit (FIM) and the implicit pressure explicit saturation (IMPES) methods. Then, for the MS experiments, we perform both deterministic and robust optimization. The second model, the Kanaal model, is larger and for this one we run robust optimization experiments only, since the interest is to account for the geological uncertainty. For both models we analyze the computational speedup gained when running MS simulations in comparison with a similar StoSAG optimization workflow that uses fine-scale methods to evaluate the model responses.

To test the efficiency of the proposed MS-StoSAG workflow for life-cycle optimization, we ran several experiments on the two different models. For all optimization experiments, the controls used are bottom-hole pressures (BHP) for both injection and production wells. These values are bound between a minimum value of 280 bar and a maximum value of 300 bar for the production wells, while the BHP in the injection wells are bound between a minimum value of 300 bar and maximum value of 320 bar. An economic objective function is used, Net Present Value (NPV), as defined in (5.1) with the constants defined in **Table 6.1**. By using a zero discount factor we compute the undiscounted NPV.

Table 6.1: Constants used for computing the NPV.

	Value	Unit
Oil price r_o	252	\$/m ³
Cost of injected water r_{wi}	60	\$/m ³
Cost of produced water r_{wp}	30	\$/m ³
Discount factor b	0	%/year

The stopping criterion for the optimization experiments is the maximum number of optimization iterations. This number differs for each of the two models and was set by analyzing the convergence rate of the objective function.

For a better comparison, all numerical methods plotted in the same figure share the same schedule. Hence, both the time-steps for simulation and the control time-steps are imposed, reducing this way the differences that would result from allowing different Δt . We imposed the same tolerances for the residuals for all i-MSFV coarsening ratios. The threshold for the residual norm of the multiscale solution was defined as 10^{-1} , while the fine scale residual norm tolerance after smoothing was set to 10^{-6} .

6.1. Five-Spot Model

The Five-Spot Model is a simple synthetic 2D model, consisting of a 21×21 regular mesh, four production wells in the corners and one injection well placed in the central cell, as depicted in **Fig. 6.1**. The size of the grid blocks is $33.3 \times 33.3 \times 2m$ and the porosity is set constant to 0.3 and the fluid properties are defined in **Table 6.2**.

There are 1000 realizations (permeability fields) of the Five-Spot model that were generated via the decomposition of a reference permeability “image” using Principal Component Analysis parameterization as presented by (Jansen, 2013a).

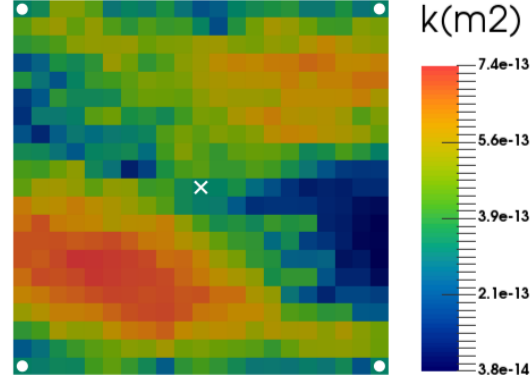


Figure 6.1: Well placement for the Five-Spot Model on top of the permeability field of one of the realizations

Table 6.2: Fluid properties for the Five-Spot Model.

Property	Value	Unit
Oil dynamic viscosity (μ_o)	0.5×10^{-3}	Pa s
Water dynamic viscosity (μ_w)	1.0×10^{-3}	Pa s
End-point relative permeability, oil (k_{row})	0.9	—
End-point relative permeability, water (k_{rw})	0.6	—
Corey exponent, oil (N_o)	2.0	—
Corey exponent, water (N_w)	2.0	—
Residual-oil saturation ($S_{or} = 0.2$)	0.2	—
Connate-water saturation (S_{wc})	0.0	—

6.1.1. Deterministic Life-cycle Optimization

For deterministic optimization a single realization is used and an ensemble of 50 perturbed control vectors with a perturbation size set to 10% of the difference between the max and min bounds of the controls. The initial control strategy consists in maintaining constant BHP of 310 bars in the injector well and constant BHP of 290 bars in the production wells. Since there are 5 wells, there are 5 control variables per control time step. We simulate 6 control time steps of 720 days each, thus we optimize 30 variables. The stopping criterion is set to 25 optimization iterations.

FIM vs IMPES

The fully implicit (FIM) and the implicit pressure explicit saturation (IMPES) schemes are presented in section 2.3. Out of the two, FIM is the most commonly used technique. In the examples presented in this thesis both FIM and IMPES achieved the same solution accuracy. **Fig. 6.2** illustrates the evolution of the objective function for fine-scale techniques FIM and IMPES. Note the difference in NPV in the first iteration, when they are both using the same control vector. The two methods provided similar NPV and we can see the difference in optimal control strategy in **Fig. 6.3**. This does not imply that IMPES always returns higher NPV. If we run again the experiment, the same realization but with slightly different settings,

e.g. different time-steps or different initial strategy, we would expect to see situations where FIM generates higher NPV than IMPES. There are several reasons for which we see different results when comparing the two numerical methods, e.g. IMPES has lower numerical diffusion (Aziz and Settari, 1979).

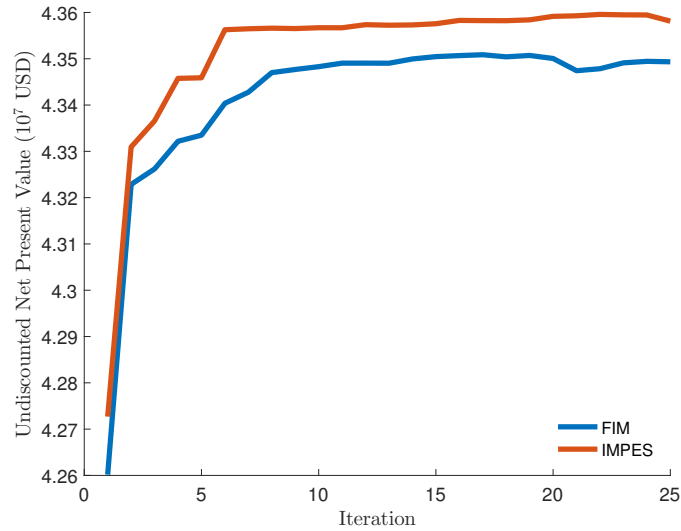


Figure 6.2: Comparison of the optimization performance between FIM and IMPES

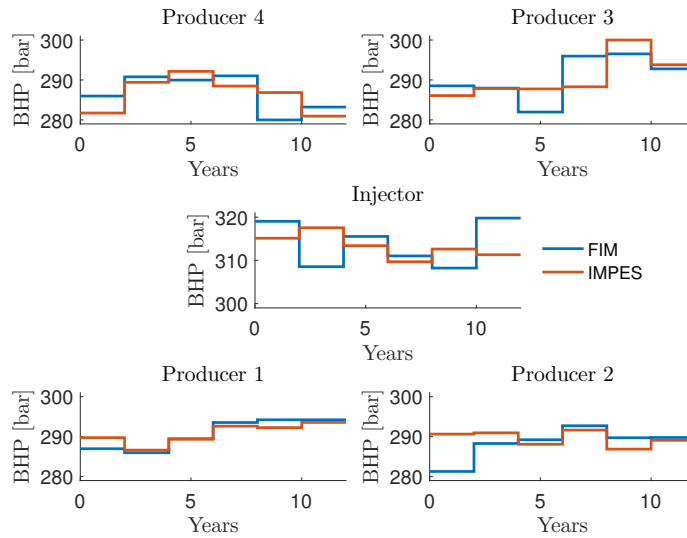


Figure 6.3: Comparison of the optimal control strategies from FIM and IMPES scheme on fine scale model

Multiscale Optimization

When comparing results from coarse-scale simulation models it is imperative to have a reference fine-scale solution. Considering that the MS reservoir simulator used is based on the IMPES coupling, for the remainder of the thesis IMPES will be the only fine-scale method used for comparison with the coarse-scale results. Hence, we compare 3 different models: fine-scale using IMPES and two coarsening ratios for i-MSFV, 3×3 and 7×7 . The objective values after each optimization iteration are illustrated in **Fig. 6.4** and we observe that the optimal strategies produce similar NPV values. The i-MSFV 3×3 model achieves the highest NPV, followed by the fine-scale IMPES model and the i-MSFV 7×7 model. It might seem counter-intuitive that the coarsest model used produced the highest NPV, but since the same tolerances for the residuals were imposed, both i-MSFV coarsening ratios provide "equally good" approximations of the fine scale IMPES. This can also be seen in **Fig. 6.6** where

the best control strategies generated by the two coarsening ratios of i-MSFV were simulated at fine scale using IMPES, for validation purposes. From **Fig. 6.6** we can also compute the relative error of the two approximations. This way we obtain 0.05% relative error for iMSFV 3×3 model and 0.04% for the 7×7 model. Note that the approximated values can be both higher or lower than the "true" fine-scale result, see **Fig. 6.6**. During our experiments we could not identify a pattern that could suggest consistent results of coarsening ratios over or under-estimating the NPV value.

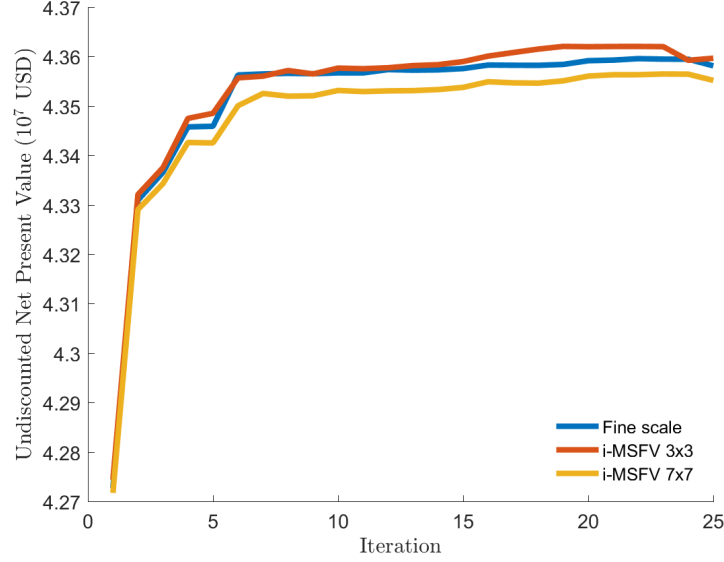


Figure 6.4: Comparison of the objective function through the iteration process for IMPES and the two i-MSFV models

Fig. 6.5 depicts the comparison of the optimal control strategies obtained from the different models. We observe that the control strategies are very similar to each other, thus the small differences in NPV.

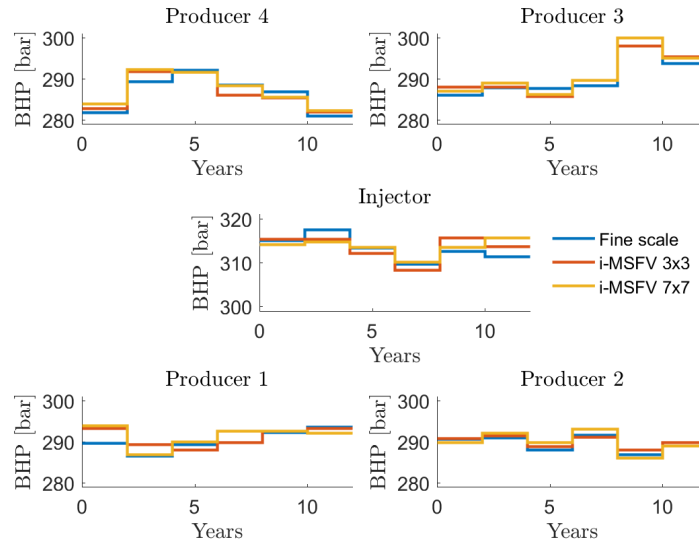


Figure 6.5: Comparison of the optimal control strategies between IMPES and two i-MSFV models

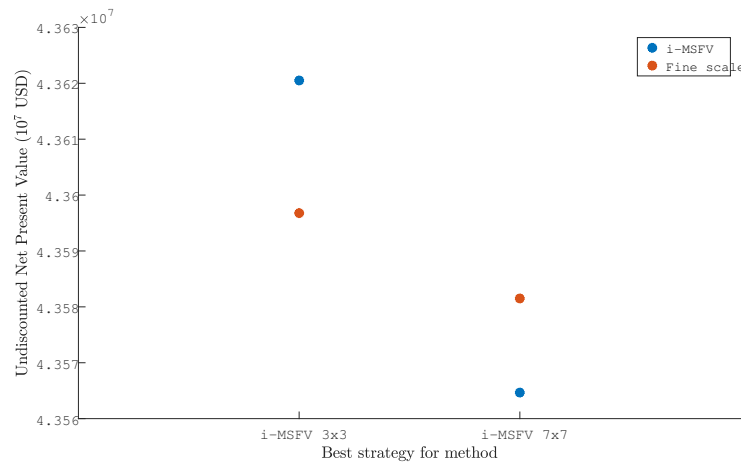


Figure 6.6: Validation for the i-MSFV results - deterministic

6.1.2. Robust Life-cycle Optimization

To account for the geological uncertainty robust optimization uses an ensemble of equiprobable geological model realizations. For this experiment 50 equiprobable geological realizations were selected, 2 of the permeability fields are illustrated in **Fig. 6.7**. The ensemble was chosen by simulating all 1000 realizations with the same control vector and uniformly sampling realizations that obtained the highest, lowest and intermediate objective function values. Note that the realization used for the deterministic optimization is part of this ensemble. The basis functions were built independently for each geological realization.

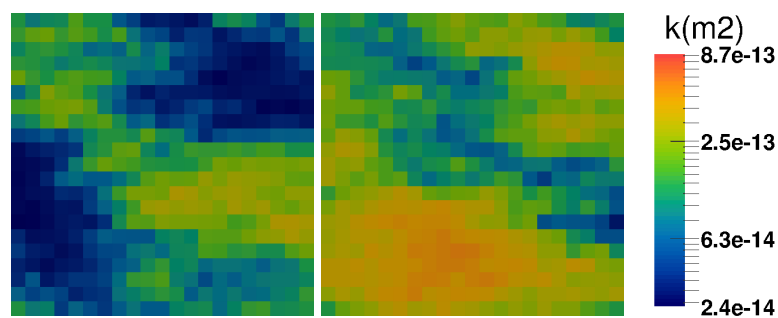


Figure 6.7: Two different permeability realizations from the ensemble of 1,000 members used in the 2D numerical experiments.

For the robust optimization experiments the objective function is mean NPV over the 50 realizations. The same coarsening ratios were used as for the deterministic optimization and the same initial control vector. **Fig. 6.8** illustrates the evolution of the mean NPV through the optimization process. We observe that the fine scale IMPES model achieves the highest NPV followed by the i-MSFV 7×7 model and the i-MSFV 3×3 model.

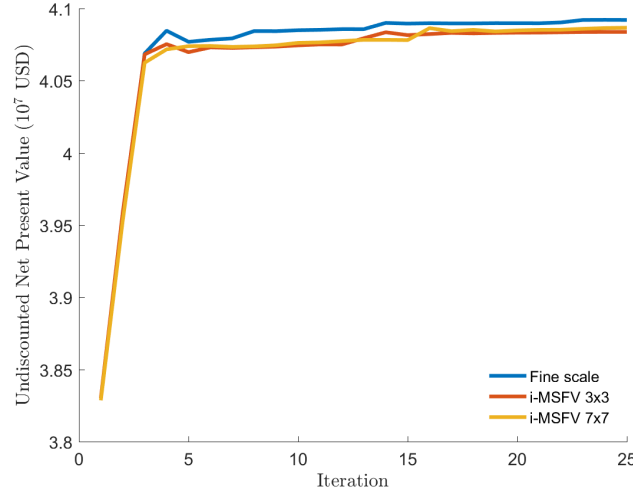


Figure 6.8: Comparison of the mean objective function value for the three different models, IMPES (blue), i-MSFV 7×7 (orange) and i-MSFV 3×3 (red)

The optimal strategies for fine scale IMPES and i-MSFV 3×3 can be visualized in **Fig. 6.9**. This time we can see that the strategies are quite different. This behavior was expected since it is known that small differences in the results of the forward simulations can lead to differences in control strategies. From **Fig. 6.10** it can be concluded that i-MSFV provided good approximations of the fine scale model. We obtained a relative error of 0.07% for iMSFV 7×7 and an order of magnitude lower for the 3×3 approximation.

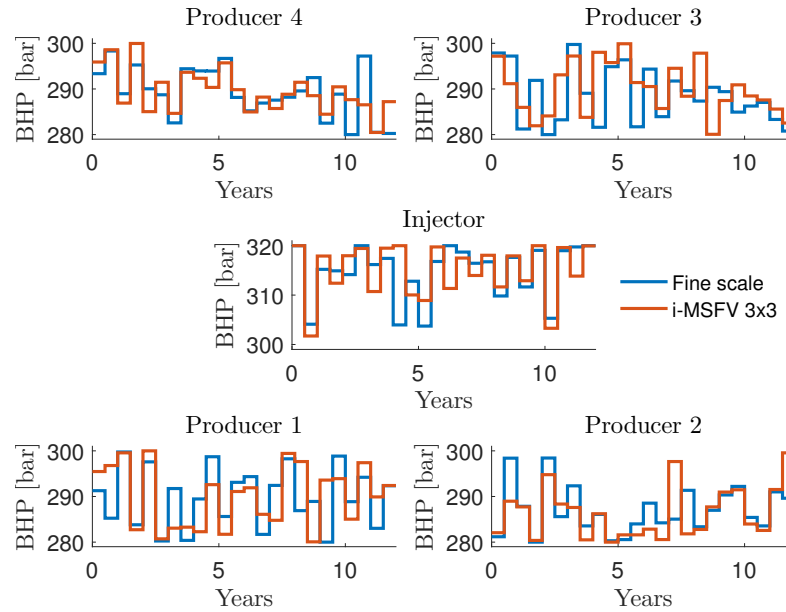


Figure 6.9: Comparison of the controls that generated the highest NPV

6.1.3. Computational Efficiency

Table 6.3 provides a comparison of the time spent for the linear solver for all four methods and the mean number of smoothing steps needed by the two coarsening ratios of i-MSFV. It is well-known that the linear system solution is the most time-consuming part in reservoir simulation (Moraes et al., 2017) and for i-MSFV it accounts for computing the coarse system and fine-scale smoothing. Although the construction of basis functions adds an overhead

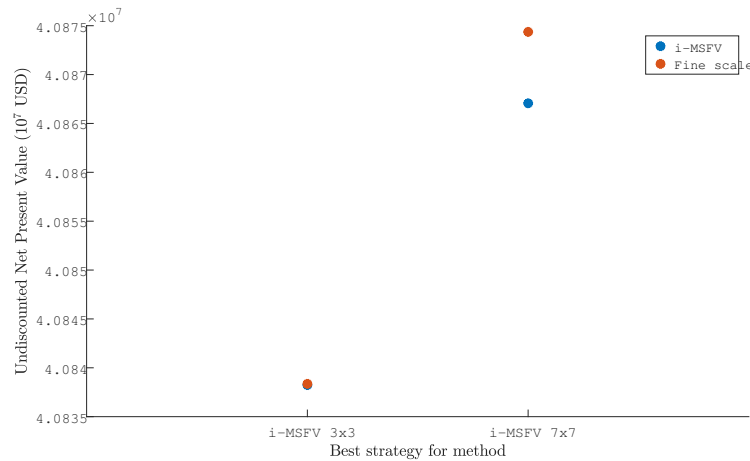


Figure 6.10: Validation for the i-MSFV results - robust

to the total simulation time (Tene et al., 2016), in these experiments the basis functions did not require to be recomputed. More than this, once computed, the basis functions can be stored in a file and read at the beginning of the next optimization iteration, since the basis functions account for the properties of the realization and not of the control.

By using linear solver time to measure the speed up, we see improvement of more than 4 times in comparison to fine-scale IMPES. The difference in the average number of smoothing steps is negligible between the two coarsening ratios, but this is model and problem dependent. In order to better analyze the efficiency of the proposed framework a larger model is proposed below.

Table 6.3: Comparison of the computational effort and average number of smoothing steps for the different i-MSFV cases

Simulation strategy	Time for linear solver (sec)	Number of smoothing steps
FIM 21×21	6	
IMPES 21x21	4	
i-MSFV 7x7	<1	1
i-MSFV 3x3	<1	2

6.2. Kanaal Reservoir Model

To illustrate the effectiveness of the proposed workflow a second larger model is considered. Comparing the results of the two fine-scale models is out of the scope of this thesis. The results of the coarse-scale models will only be compared with fine-scale IMEPS since the MS simulator used is based on this strategy.

We use a 99×99 2D model of with grid blocks of $10 \times 10 \times 5$ m. It is a larger model of about 10,000 grid blocks compared to the 441 grid blocks in the Five-spot model. The geological description is inspired from typical channelized North Sea reservoirs, hence the name Kanaal (channel in Dutch). The channels have permeabilities ranging from 250-700 mD and they are bounded by sandy shale with permeabilities ranging from 10-50 mD. The rock porosity is constant of 20% and the fluid properties are presented in **Table 6.4**. We used a line drive well configuration with 3 injection wells and 6 production wells as illustrated in **Fig. 6.11**.

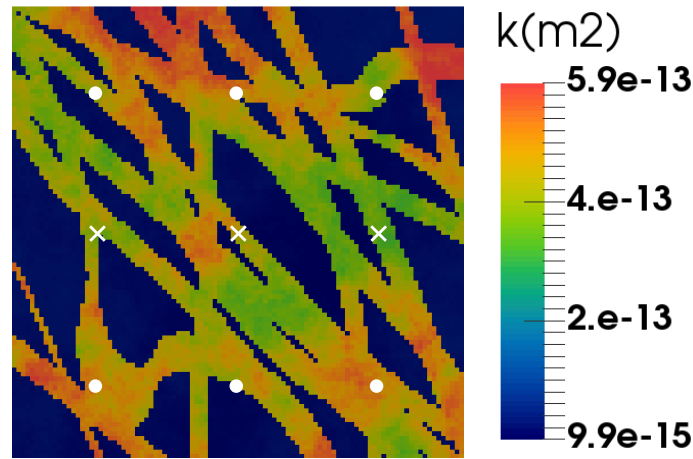


Figure 6.11: Illustration of the well configuration used for the Kanaal reservoir model with injectors in the middle denoted by X

Table 6.4: Fluid properties for the Kanaal reservoir model

Property	Value	Unit
Oil dynamic viscosity (μ_o)	2×10^{-3}	Pa s
Water dynamic viscosity (μ_w)	1.0×10^{-3}	Pa s
End-point relative permeability, oil (k_{row})	0.9	–
End-point relative permeability, water (k_{rw})	0.8	–
Corey exponent, oil (N_o)	2.0	–
Corey exponent, water (N_w)	2.0	–
Residual-oil saturation ($S_{or} = 0.2$)	0.2	–
Connate-water saturation (S_{wc})	0.0	–

An ensemble of 50 geological realizations has been created using the geological modeling software Petrel (Schlumberger, 2016b). The realizations have been constrained to well logs from 4 exploration wells which were generated from a base case model. The base case model does not form part of the ensemble of model realizations. The permeability fields of the two geological models are illustrated in **Fig. 6.12**.

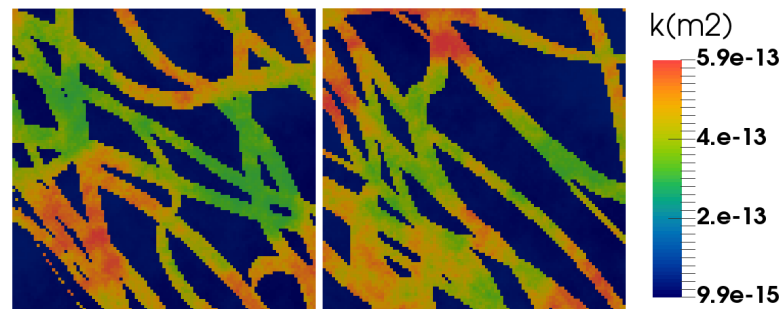


Figure 6.12: Two different permeability realizations from an ensemble of 100 members

6.2.1. Robust Life-cycle Optimization

For this larger model we focus on robust optimization experiments only. The total simulation time is 12 years, the controls are bottom hole pressure for all wells and can be changed every 6 months, resulting a total of 24 control time steps. Thus, we optimize for 9 wells $\times$ 24 control time steps, a total of 216 controls. The controls are bound in a similar manner with the previous model. For the production wells the BHP takes values in between the minimum of 280 bar and maximum 300 bar, while for the injection wells the minimum value is 300 bar and maximum 320 bar.

The initial control strategy corresponds to the minimum bound for the production wells, i.e. constant pressure of 280 bars and maximum bound for the injection wells, a constant pressure of 320 bars. The following coarsening ratios are compared: fine-scale (IMPES), 33×33 , 11×11 and 3×3 . In **Fig. 6.13** we can see the mean NPV for all these experiments and the fine scale. Note that all methods provide a better mean NPV than the reactive control strategy, which was simulated on the fine-scale and consists in maintaining the initial control vector until a production well produces more than 82% water (economic water cut), moment in which it is closed. This reactive process is repeated until all production wells are shut. From **Fig. 6.13** we also observe that different coarsening ratios achieve different optimal mean NPV values. The fine-scale IMEPS achieves the highest NPV, followed by i-MSFV 11×11 , i-MSFV 33×33 and i-MSFV 3×3 . Note that the same tolerances were imposed for all three coarsening ratios.

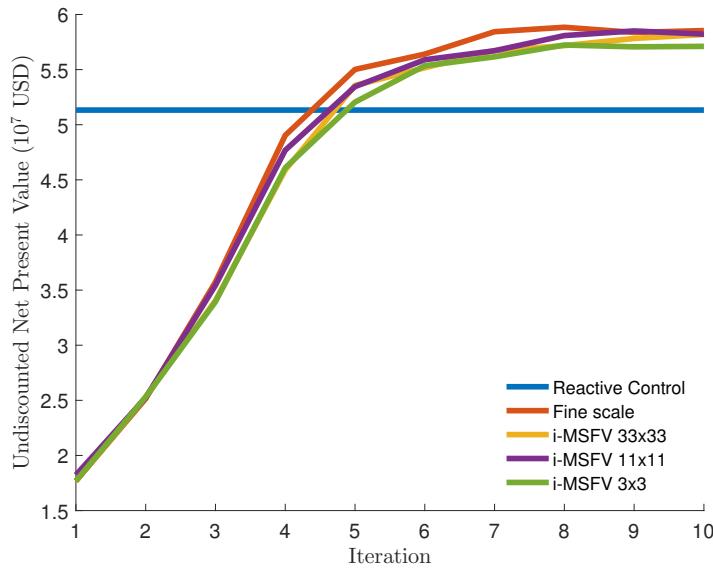


Figure 6.13: Comparison of objective function value between IMPES and the different i-MSFV models used

In **Fig. 6.14** we can see the comparison between the optimal control strategies for the fine scale IMPES model and i-MSFV 11×11 model. Note that the objective values are very close, yet the control strategies significantly differ from each other and the same trend is observed for the other seven wells. This is in accordance with the observation in earlier studies; see, e.g. Jansen et al. (2008). We can draw the same conclusion from **Fig. 6.15** which illustrates the optimal controls for two different i-MSFV coarsening ratios. The validation results presented in **Fig. 6.16** confirm that the tolerances imposed to the i-MSFV models were sufficient and all coarsening ratios provide a good approximation of the fine-scale model. For this case we obtained a relative error 0.01% for iMSFV 3×3 , 0.2% for iMSFV 11×11 and 0.01% for the 33×33 coarsening ratio. Even though in **Fig. 6.16** it might seem that the i-MSFV provided a better approximation of the fine scale than it did for the Five-Spot model, the differences have the same magnitude.

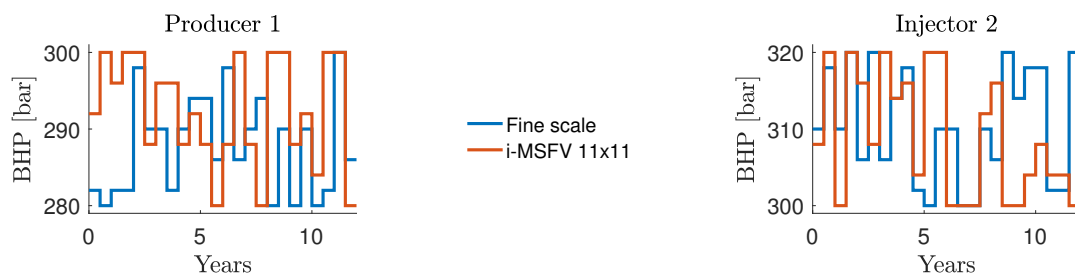


Figure 6.14: Comparison of the optimal control strategies from IMPES and i-MSFV 11×11 model

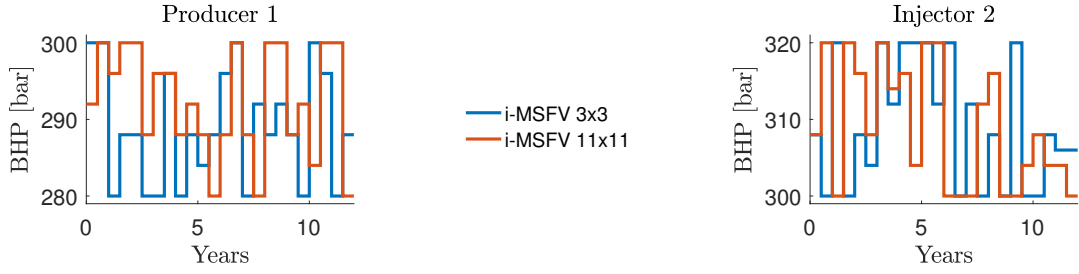


Figure 6.15: Comparison of the optimal control strategies from i-MSFV 11x11 and i-MSFV 3x3 model

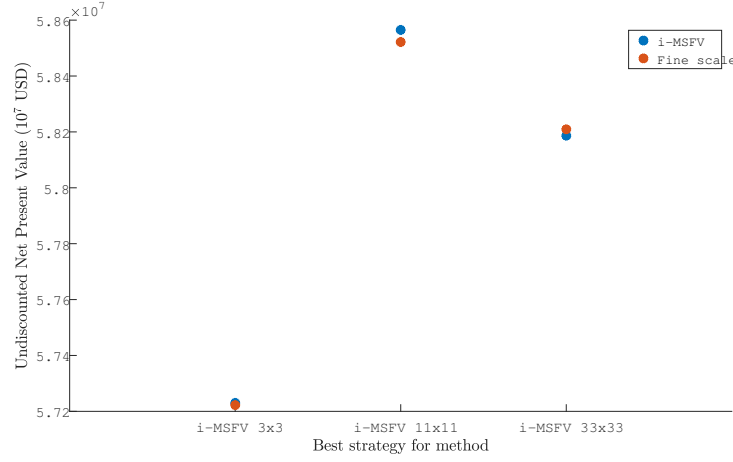


Figure 6.16: Validation for the i-MSFV results

In **Table 6.5** we provide a comparison of the production volumes of oil and water, as well as the volume of injected water obtained during the validation of the optimal control strategies for the i-MSFV models compared to the fine scale IMPES model. We observe that higher volumes of oil and water are produced in the i-MSFV models compared to the fine scale IMPES model. We can also observe that the higher the coarsening ratio, the higher are the production and injection volumes. We have not yet been able to find a conclusive explanation for these trends.

Table 6.5: Comparison of optimal produced and injected volumes for the different simulation models

Simulation Strategy	Cumulative Oil Produced	Cumulative Water Produced	Cumulative Water Injected
IMPES	3.82e5	2.88e5	6.37e5
i-MSFV 33x33	3.87e5	3.09e5	6.97e5
i-MSFV 11x11	3.86e5	3.03e5	6.90e5
i-MSFV 3x3	3.92e5	3.32e5	7.24e5

6.2.2. Computational Efficiency

In this section we analyze the computational efficiency and compare the speedup gained for the different coarsening ratios. For the Kanaal model an approximately 5 times speedup in computational efficiency is achieved when compared to the fine-scale as observed in **Table 6.6**. The tolerances of the residual for the MS solution and the smoothing steps are the same for all the cases. This time we can see a clear difference in between the mean number of smoothing steps required, coarser models needed more smoothing steps. From **Table 6.6** we can see that the best speedup was achieved by i-MSFV 11x11. It might seem counter intuitive that the coarsest model did not provide the best speed-up and it was slightly faster than 33×33 . To analyze the time spent in the linear solver we can first assume that the time needed for doing a smoothing step is almost constant since it is applied to the fine-scale

reconstructed pressure vector, so the coarsening ratio does not influence it. We can observe that even if the size of the system that has to be solved for the 33×33 ratio is larger than the one for iMSFV 3×3 , because it needed three times fewer smoothing steps it obtained similar speed-up. The problem of determining a priori which coarsening ratio to use for the best speed-up is far from being trivial especially since it is highly dependable on the model. Further experimentations on more complicated models might lead to the development of a set of "good practices" for making an "educated guess" on selecting the coarsening ratio.

Table 6.6: Comparison of the computational effort and average number of smoothing steps for the different i-MSFV cases

Simulation strategy	Speed-up for linear solver	Number of smoothing steps
IMPES	1	
i-MSFV 33×33	4.44	5
i-MSFV 11×11	5.69	12
i-MSFV 3×3	4.62	18

Conclusions

In this thesis we presented a computationally efficient workflow based on multiscale and stochastic optimization (MS-StoSAG) and tested it on two synthetic test cases. Based on the numerical experiments presented we can conclude that this workflow provided improvements in the objective function similar to the ones obtained when running optimization of the fine-scale, but in a more computationally efficient manner.

In most cases the optimal strategies were significantly different even though the NPV values were similar. In these experiments the difference in optimal control strategies is more pronounced when running robust optimization. The differences in optimal control strategies were expected, since it is known that small changes in the results of the forward simulations can lead to a different control vector. The optimal strategies of the MS experiments were validated on the fine-scale and we saw that the approximated results were very close to the exact ones, confirming the accuracy of the multiscale forward simulation models. It is not possible to know prior to the experiments which method will provide the highest improvement in NPV.

For the numerical experiments presented, the smoothing iterations of i-MSFV managed to compensate for not recomputing the basis function. It would be interesting to test if this holds when the simulation involves more complex physics like gravity effects, capillarity, higher mobility ratios, etc. However, even if the basis functions need to be recomputed for more complex scenarios, they are built adaptively and infrequently.

Reduced-order models are computationally very efficient, but because of the oversimplification in physics, the results can sometimes be significantly different compared to the fine-scale model. It is mandatory to validate the optimized strategy on the fine-scale model to test the accuracy of the results. More than this, when working with reduced order modeling methods such as POD (van Doren et al., 2006) or TPWL (Cardoso et al., 2010), it is also required to run several high-fidelity simulations a-priori to develop the reduced order model. As opposed, the workflow presented only requires the computation of the basis functions, which is a cheap process. Since the transport problem is always solved on the fine scale, there is no need to validate the optimal strategies as it was confirmed by the results in this paper.

For the larger Kanaal model an approximately five times speedup in computational efficiency is achieved and we expect the computational advantage to increase for larger-sized reservoir models. In this thesis, transport equation was solved on the fine-scale to improve the accuracy of our results.

7.1. Recommendations

The implementation of the MS reservoir simulator does not yet exploit the full potential of MS methods. These methods are known for being suited to utilize modern high performance-computing architectures. For example, the basis functions are extremely parallelizable.

Since the BF do not depend on the controls applied to a reservoir and considering that every realization is simulated at least two times for each ensemble optimization iteration, the basis functions can be reused for the same realization. In other words, once the basis functions of a realization are build, they can be saved in a file and can be read from there next time the model response needs to be evaluated.

Further improvement in computational efficiency can be achieved if the transport problem is solved at coarse-scale as suggested in (Krogstad et al., 2011).

In conclusion, the MS-StoSAG workflow achieved computationally efficient results of significant practical value, making it an attractive solutions that deserves further improvements and testing on more complex cases.

Bibliography

- (2006). *Computational Methods for Multiphase Flows in Porous Media*. SIAM. <http://resolver.tudelft.nl/uuid:6000459e-a0cb-40d1-843b-81650053e093>.
- Aarnes, J. E., Lie, K.-A., Kippe, V., and Krogstad, S. (2009). *Multiscale Methods for Subsurface Flow*, pages 3–48. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Aziz, K. and Settari, A. (1979). *Petroleum reservoir simulation*. Applied Science Publishers Ltd., London, U.K.
- Cardoso, M. A., Durlowsky, L. J., et al. (2010). Use of reduced-order modeling procedures for production optimization. *SPEJ*, 15(02):426–435. SPE-119057-PA. <http://dx.doi.org/10.2118/119057-PA>.
- Chen, B. (2008a). *A Stochastic Simplex Approximate Gradient for Production Optimization of WAG and Continuous Water Flooding*. PhD thesis, The University of Oklahoma.
- Chen, Y. (2008b). *Ensemble-Based Closed-Loop Production Optimization*. PhD thesis, The University of Tulsa.
- Chen, Y., Oliver, D., and Zhang, D. (2009). Efficient ensemble-based closed-loop production optimization. *SPE Journal*, 14(4):634–645. SPE-112873-PA. <http://dx.doi.org/10.2118/112873-PA>.
- Chen, Z., Huan, G., and Li, B. (2004). An improved IMPES method for two-phase flow in porous media. *Transport in Porous Media*, 54(3):361–376. <http://dx.doi.org/10.1023/B:TIPM.00000003667.86625.15>.
- Coats, K. et al. (2003). IMPES stability: The CFL limit. *SPEJ*, 8(03):291–297. SPE-85956-PA. <http://dx.doi.org/10.2118/85956-PA>.
- David, T. (2010). The End of Easy Oil. <https://www.technologyreview.com/s/420404/the-end-of-easy-oil/> (accessed 14 January 2016).
- Do, S. T. and Reynolds, A. C. (2013). Theoretical connections between optimization algorithms based on an approximate gradient. *Computational Geosciences*, 17(6):959–973. <http://dx.doi.org/10.1007/s10596-013-9368-9>.
- Fonseca, R., Stordal, A., Leeuwenburgh, O., Van den Hof, P., and Jansen, J. (2014). Robust ensemble-based multi-objective optimization. In *ECMOR XIV-14th European conference on the mathematics of oil recovery*. Sicily, Italy, 8-11 September 2014. <https://www.scopus.com/inward/record.uri?eid=2-s2.0-84908153652&partnerID=40&md5=b7532d2661cbd061688178ec4c05b1dd>.
- Fonseca, R.-M. (2015). *A Modified Gradient Formulation for Ensemble Optimization under Geological Uncertainty*. PhD thesis, Delft University of Technology.
- Fonseca, R. M., Chen, B., Jansen, J. D., and Reynolds, A. (2015). A Theoretical Understanding of Approximate Ensemble-Based Stochastic Gradients for Optimization Under Uncertainty. In *International Journal for numerical methods in engineering*. Wiley InterScience.
- Hajibeygi, H. (2011). *Iterative Multiscale Finite Volume Method for Multiphase Flow in Porous Media with Complex Physics*. PhD thesis, ETH Zurich.

- Hajibeygi, H., Bonfigli, G., Hesse, M. A., and Jenny, P. (2008). Iterative multiscale finite-volume method. *Journal of Computational Physics*, 227(19):8604–8621. <http://dx.doi.org/10.1016/j.jcp.2008.06.013>.
- Hajibeygi, H. and Jenny, P. (2011). Adaptive iterative multiscale finite volume method. *Journal of Computational Physics*, 230(3):628–643. <http://dx.doi.org/10.1016/j.jcp.2010.10.009>.
- Hou, J., Zhou, K., Zhang, X.-S., Kang, X.-D., and Xie, H. (2015). A review of closed-loop reservoir management. *Petroleum Science*, 12(1):114–128.
- Hou, T. Y. and Wu, X.-H. (1997). A multiscale finite element method for elliptic problems in composite materials and porous media. *Journal of computational physics*, 134(1):169–189. <http://dx.doi.org/10.1006/jcph.1997.5682>.
- Jansen, J. (2011). Adjoint-based optimization of multi-phase flow through porous media – A review. *Computers & Fluids*, 46(1):40 – 51. 10th {ICFD} Conference Series on Numerical Methods for Fluid Dynamics (ICFD 2010).
- Jansen, J. D. (2013a). *A simple algorithm to generate small geostatistical ensembles for subsurface flow simulation. Research note.* Dept. of Geoscience and Engineering, Delft University of Technology, The Netherlands. <http://resolver.tudelft.nl/uuid:6000459e-a0cb-40d1-843b-81650053e093>.
- Jansen, J. D. (2013b). *A Systems Description of Flow Through Porous Media.* Springer International Publishing. <http://dx.doi.org/10.1007/978-3-319-00260-6>.
- Jansen, J.-D., Bosgra, O. H., and Van den Hof, P. M. (2008). Model-based control of multiphase flow in subsurface oil reservoirs. *Journal of Process Control*, 18(9):846–855. <http://dx.doi.org/10.1016/j.jprocont.2008.06.011>.
- Jansen, J.-D., Brouwer, D., Naevdal, G., and Van Kruijsdijk, C. (2005). Closed-loop reservoir management. *First Break*, 23(1):43–48.
- Jansen, J.-D., Brouwer, R., and Douma, S. G. (2009). Closed Loop Reservoir Management. SPE-119098-MS. <http://dx.doi.org/10.2118/119098-MS>.
- Jenny, P., Lee, S., and Tchelepi, H. (2005). Adaptive multiscale finite-volume method for multiphase flow and transport in porous media. *Multiscale Modeling & Simulation*, 3(1):50–64. <http://dx.doi.org/10.1137/030600795>.
- Jenny, P., Lee, S. H., and Tchelepi, H. A. (2003). Multi-scale finite-volume method for elliptic problems in subsurface flow simulation. *J. Comput. Phys.*, 187(1):47–67.
- Jenny, P. and Lunati, I. (2009). Modeling complex wells with the multi-scale finite-volume method. *Journal of Computational Physics*, 228(3):687–702. <http://dx.doi.org/10.2118/99690-MS>.
- Kaleta, M. P., Hanea, R. G., Heemink, A. W., and Jansen, J.-D. (2011). Model-reduced gradient-based history matching. *Computational Geosciences*, 15(1):135–153.
- Krogstad, S., Hauge, V. L., and Gulbransen, A. (2011). Adjoint multiscale mixed finite elements. *SPEJ*, 16(01):162–171. SPE-119112-PA. <http://dx.doi.org/10.2118/119112-PA>.
- Lie, K.-A. (2016). *An Introduction to Reservoir Simulation Using MATLAB: User Guide for the Matlab Reservoir Simulation Toolbox (MRST).* SINTEF ICT. <http://www.sintef.no/contentassets/8af8db2e42614f7fb94fb0c68f5bc256/mrst-book-2016.pdf>.
- Lie, K.-A., Krogstad, S., Ligaarden, I. S., Natvig, J. R., Nilsen, H. M., and Skaflestad, B. (2012). Open-source matlab implementation of consistent discretisations on complex grids. *Computational Geosciences*, 16(2):297–322. <http://dx.doi.org/10.1007/s10596-011-9244-4>.

- Lorentzen, R. J., Berg, A., Naevdal, G., and Vefring, E. H. (2006). A New Approach For Dynamic Optimization Of Water Flooding Problems. *2006 SPE Intelligent Energy Conference and Exhibition*, 1:253–263. <http://dx.doi.org/10.1016/j.jcp.2008.09.026>.
- Lu, Ranran; Forouzanfar, F. R. A. C. (2017). Bi-Objective Optimization of Well Placement and Controls Using StoSAG. *SPE Reservoir Simulation Conference, 20-22 February*, SPE-182705-MS. <http://dx.doi.org/10.2118/182705-MS>.
- Lunde, T. (2007). *Comparison Between Mimetic and Two-Point Flux-Approximation Schemes on PEBI-grids*. PhD thesis, University of Oslo.
- Moraes, R., Fonseca, R., Helici, M., Heemink, A., and Jansen, J. D. (2017). Improving the Computational Efficiency of Approximate Gradients Using a Multiscale Reservoir Simulation Framework. SPE-182620-MS. <https://doi.org/10.2118/182620-MS>.
- Moraes, R., Rodrigues, J., Hajibeygi, H., and Jansen, J. D. (2016). Multiscale gradient computation for subsurface flow models. *Journal of Computational Physics*, (submitted 13 July 2016).
- Nocedal, J. and Wright, S. (2006). *Numerical optimization*. Springer Science & Business Media.
- Saad, Y. (1994). ILUT: A dual threshold incomplete LU factorization. *Numerical linear algebra with applications*, 1(4):387–402. <http://dx.doi.org/10.1002/nla.1680010405>.
- Schlumberger (2016a). Oilfield Glossary. http://www.glossary.oilfield.slb.com/Terms/p/primary_recovery.aspx (accessed 25 November 2016).
- Schlumberger (2016b). Petrel. <http://www.software.slb.com/products/petrel> (accessed 21 November 2016).
- Ṭene, M., Al Kobaisi, M. S., and Hajibeygi, H. (2016). Algebraic multiscale method for flow in heterogeneous porous media with embedded discrete fractures (F-AMS). *Journal of Computational Physics*. <http://www.sciencedirect.com/science/article/pii/S0021999116302376>.
- van der Vorst, H. A. (1992). Bi-CGSTAB: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems. *SIAM Journal on Scientific and Statistical Computing*, 13(2):631–644.
- van Doren, J. F., Markovinović, R., and Jansen, J.-D. (2006). Reduced-order optimal control of water flooding using proper orthogonal decomposition. *Computational Geosciences*, 10(1):137–158. <http://dx.doi.org/10.1007/s10596-005-9014-2>.
- Van Essen, G., Zandvliet, M., Van den Hof, P., Bosgra, O., Jansen, J.-D., et al. (2009). Robust waterflooding optimization of multiple geological scenarios. *SPEJ*, 14(01):202–210. SPE-102913-PA. <http://dx.doi.org/10.2118/102913-PA>.
- Wang, Y., Hajibeygi, H., and Tchelepi, H. A. (2014). Algebraic multiscale solver for flow in heterogeneous porous media. *Journal of Computational Physics*, 259:284–303. <http://dx.doi.org/10.1016/j.jcp.2013.11.024>.