Multiscale Analytical Derivative Formulations for Improved Reservoir Management

de Moraes, Rafael J.

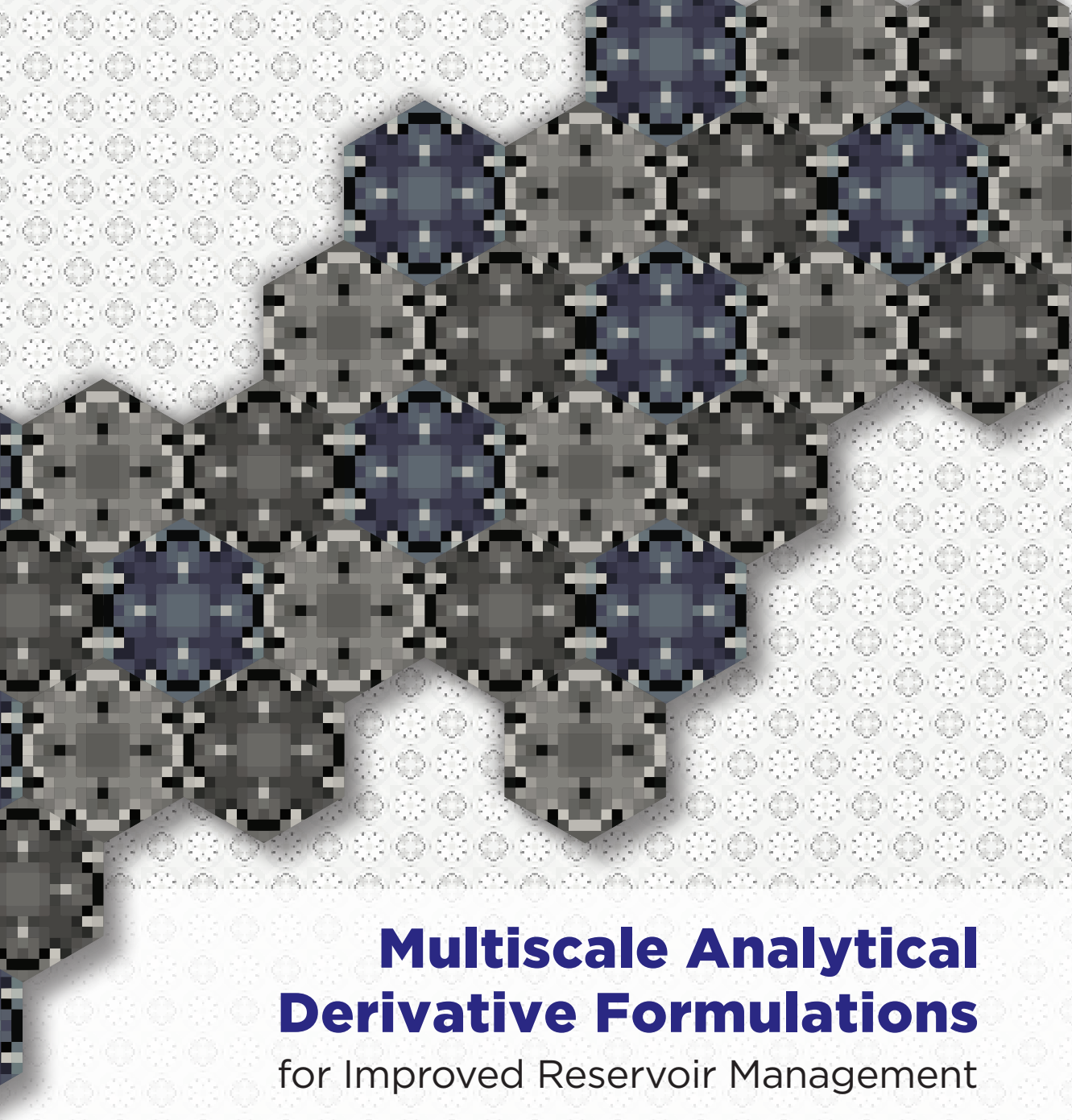
DOI
10.4233/uuid:31b1847e-e32c-482e-9e8f-286de866e751

Publication date
2018

Document Version
Final published version

Citation (APA)
de Moraes, R. J. (2018). *Multiscale Analytical Derivative Formulations for Improved Reservoir Management*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:31b1847e-e32c-482e-9e8f-286de866e751

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.



Multiscale Analytical Derivative Formulations

for Improved Reservoir Management

Rafael J. de Moraes

Propositions

accompanying the dissertation

MULTISCALE ANALYTICAL DERIVATIVE FORMULATIONS FOR IMPROVED RESERVOIR MANAGEMENT

by

Rafael MORAES

1. The multiscale adjoint gradient is accurate enough to be used by optimization algorithms (this thesis).
2. Sequentially-coupled forward simulation strategies (mainly IMPES) are easier to formulate than FIM strategies. The respective adjoint models are, however, the opposite (Chapter 3).
3. The employment of reduced order models in reservoir management studies does not make the methodology easier and should not be only faced as a means to reduce computation cost. Instead, it should be faced as a means to represent the model given the uncertainty/knowledge of the system (e.g. Chapter 7).
4. Automatic differentiation is currently not the solution for partial derivative computations for adjoint models since the current computational infrastructure does not meet the minimum development requirements.
5. Reservoir simulation is retrograde when compared to other CFD fields. I blame uncertainty for that.
6. *Shurahi*, a Japanese saying, meaning *Shu*, learning fundamentals, *ha*, breaking with traditions, *ri*, transcendence, describe the path to mastery. It is a failure if a PhD candidate ends his research without appreciating all these stages.
7. A PhD research is the process of assessing/realizing the extent of your own ignorance.
8. It's human nature, specially in academia, to recognize some one else's achievements only if comfortable with your own.
9. Personality traits should be determined based on the individuals and not on cultural background.
10. Let n be the number of children at a home. Let N be the number of caregivers / housekeepers at a home. $N \geq n+1$ is a necessary condition to keep the family's household coordinated.

These propositions are regarded as opposable and defendable, and have been approved as such by the promotor prof. dr. ir. J. D. Jansen and co-promotor dr. H. Hajibeygi.

MULTISCALE ANALYTICAL DERIVATIVE FORMULATIONS FOR IMPROVED RESERVOIR MANAGEMENT

MULTISCALE ANALYTICAL DERIVATIVE FORMULATIONS FOR IMPROVED RESERVOIR MANAGEMENT

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus, Prof. dr. ir. T.H.J.J. van der Hagen,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op
maandag 19 november 2018 om 10:00 uur

door

Rafael JESUS DE MORAES

Master of Science in Civil Engineering,
Federal University of Rio de Janeiro, Rio De Janeiro, Brazil,
geboren te Nova Friburgo, Brazil.

Dit proefschrift is goedgekeurd door de promoteoren.

Samenstelling promotiecommissie bestaat uit:

Rector Magnificus,	voorzitter
Prof. dr. ir. J. D. Jansen	Technische Universiteit Delft, promotor
Dr. H. Hajibeygi	Technische Universiteit Delft, copromotor
Onafhankelijke leden:	
Prof. dr. Y. Efendiev	Texas A&M University, United States of America
Prof. dr. ir. L. J. Sluys	Technische Universiteit Delft
Prof. dr. ir. A. W. Heemink	Technische Universiteit Delft
Dr. S. Krogstad	SINTEF, Norway
Overige leden:	
Dr. J. R. P. Rodrigues,	Petrobras Research & Development Center, Brazil

Dr. J. R. P. Rodrigues heeft in belangrijke mate aan de totstandkoming van het proefschrift bijgedragen.

DARSim
Delft Advanced Reservoir Simulation

PETROBRAS

Keywords: multiscale simulation, analytical derivative computation, adjoint method, life-cycle optimization, data assimilation

Printed by: GildePrint.

Front & Back: Art & design: Rafael J. de Moraes. Design: Filipe Daflon (dpixel.com.br). Square (back) and hexagonal (front) tessellation stimulated by the mathematically-inspired work of Dutch graphic artist M.C. Escher. Inner pattern is derived from a mathematical result of this thesis (see Figure 4.7).

ISBN 978-94-6186-990-6

An electronic version of this dissertation is available at
http://repository.tudelft.nl/.

To my guiding star, Estela, and my leading light, Helena.

R. J. de Moraes

CONTENTS

SUMMARY

The exploitation of subsurface resources is, inevitably, surrounded by uncertainty. Limited knowledge on the economical, operational, and geological setting are just a few instances of sources of uncertainty. From the geological point of view, the currently available technology is not able to provide the description of the fluids and rock properties at the necessary level of detail required by the mathematical models utilized in the exploitation decision-making process. However, even if a full, accurate description of the subsurface was available, the outcome of such hypothetical mathematical model would likely be computationally too expensive to be evaluated considering the currently available computational power, hindering the decision making process.

Under this reality, geoscientists are consistently making effort to improve the mathematical models, while being inherently constrained by uncertainty, and to find more efficient ways to computationally solve these models.

Closed-loop Reservoir Management (CLRM) is a workflow that allows the continuous update of the subsurface models based on production data from different sources. It relies on computationally demanding optimization algorithms (for the assimilation of production data and control optimization) which require multiple simulations of the subsurface model. One important aspect for the successful application of the CLRM workflow is the definition of a model that can both be run multiple times in a reasonable timespan and still reasonably represent the underlying physics.

Multiscale (MS) methods, a reservoir simulation technique that solves a coarser simulation model, thus increasing the computational speed up, while still utilizing the fine-scale representation of the reservoir, figures as an accurate and efficient simulation strategy.

This thesis focuses on the development of efficient algorithms for subsurface models optimization by taking advantage of multiscale simulation strategies. It presents (1) multiscale analytical derivative computation strategies to efficiently and accurately address the optimization algorithms employed in the CLRM workflow and (2) novel strategies to handle the mathematical modeling of subsurface management studies from a multiscale perspective. On the latter, we specifically address a more fundamental multiscale aspect of data assimilation studies: the assimilation of observations from a distinct spatial representation compared to the simulation model scale.

As a result, this thesis discusses in detail the development of mathematical models and algorithms for the derivative computation of subsurface model responses and their application into gradient-based optimization algorithms employed in the data assimilation and life-cycle optimization steps of CLRM. The advantages are improved computational efficiency with accuracy maintenance and the ability to address the subsurface management from a multiscale view point not only from the forward simulation perspective, but also from the inverse modeling side.

PREFACE

This thesis collates the developements made during the course of my PhD research project, from January 2015 to June 2018. During this period I was under the Petrobras' program for educational development of human resources, when I was released from my activities at the Research Center and could mostly focus on my PhD research.

The ultimate goal of the research was to investigate the application of multiscale simulation strategies into reservoir management studies. From this broadly stated research goal, the objective was naturally shaped along the process into the development of analytical derivative computation methods applied to gradient-based optimization techniques, a fundamental aspect of optimization algorithms.

The chapters consist of either published or submitted conference and journal papers produced during the course of the PhD research. Because the thesis is a compilation of the papers, any chapter can be read individually and no reading order is particularly required. However, it should be noted that chapter 2 provides a detailed presentation of the basic framework for all derivative calculation methods and algorithms presented in the thesis. Similarly, the developments presented in chapter 4 forms the basis for all the multicale gradient computation methods here presented.

The introduction chapter aims to provide the research background and the fundamental mathematics associated with the developments. Literature review of the state of the art of the different fundamental components are provided individually at the introduction of each chapter. Also, it is expected that the reader is familiar with some common nomenclature used in the Petroleum Engineering community.

Even though all the mathematical developments have been demonstrated in petroleum reservoir applications, the main target of this research project, all developments are firstly described in an abstraction level that allows them to be applied in other fields of science, e.g. meteorology, oceanography, system & control, among others.

Multiscale is a somewhat overloaded terminology, used to describe the conciliation of different levels of modeling representations. In the specific way the multiscale concept has been applied in this work, we consider that it is utilized between any 'arbitrary scales that are adjacent'. This is consistent with the multiscale applications found in the subsurface simulation literature when the technique is employed as a means to reduce computational cost. And among all multiscale aspects associated with multiscale reservoir simulation, this work is focused on the details of derivative computation.

Even though this thesis, in its majority, is focused on the analytical computation of derivatives, a discussion on how stochastic gradient computation can take advantage of MS simulation can be found in Appendix A.

Rafael Jesus de Moraes
Delft, May 2018

1

INTRODUCTION

1

The decision making process associated to a petroleum reservoir exploitation strategy is often based on numerical simulation studies. The reservoir simulation model is built based on a very limited knowledge of the reservoir geological description, fluid characterization and distribution, and rock properties. All this uncertainty leads to often inaccurate forecast simulations of the field production once the mathematical models rely on this uncertain data. However, during the life cycle of a petroleum field, more data may become available as a result of the reservoir exploitation itself (e.g. volume rates and pressures registered for all wells in the field) and of the strategy to acquire more data (e.g. seismic surveys, well logs). All this information can help to understand the fluids distributions and rock properties and hence help to reduce the uncertainty.

Closed-loop Reservoir Management (CLRM) is a workflow that allows the continuous update of reservoir models based on data from different sources and is a topic of active research [1–4] to make it a tool to be utilized in the decision making process. The workflow is illustrated in Fig. 1.1.

CLRM is a combination of model-based optimization and data assimilation (computer-assisted history matching) [5, 6]. The aim is to maximize reservoir performance, in terms of recovery or financial measures, over the life of the reservoir by changing reservoir management from a periodic to a near-continuous process [7]. Model based optimization workflows aim to produce reservoir management strategies of significant practical value. It relies on computationally demanding optimization algorithms (for data assimilation data and well control optimization) which require multiple reservoir simulations. One important aspect for the successful application of the CLRM workflow is the definition of a model that can both be run multiple times in a reasonable timespan and accurately enough represent the underlying physical phenomena. Techniques such as Reduced-Order Modeling (ROM) [8–11] and upscaling [12–14] are common techniques to create models which are faster to be evaluated. Gradient based optimization techniques have been reported to be the most efficient approaches when it comes to life cycle model-based optimization [15].

One key aspect of the workflow is the up/down scaling of the system models (represented by a the green ellipsis in Fig. 1.1). Low-order models are built because on both data assimilation (represented by the red ellipsis Fig. 1.1) and optimization (represented by the blue ellipsis in Fig. 1.1) loops many evaluations of the system model are performed, and hence a fast, yet accurate, low fidelity representation of the high-order model is highly desirable. Different strategies have been employed to construct low-order models. The straightest strategy is to directly use the high-order reservoir model (i.e. the geological model) as the system-model. This strategy is the most accurate one; though computationally too expensive, and often prohibitive for real-scale models. Upscaling techniques can provide a dynamic model in a coarser scale that is faster to be evaluated than the high-order models; however at the expense of accuracy and de-attachment from geological model. Another approach is to create a Reduced Order Model (ROM) which although provides a very fast low-order system models, those are only valid on a certain range of application and may not be mass conservative [10].

MS methods [16, 17], an upcoming reservoir simulation technique, however, solves a coarser simulation model, thus increasing the computational speed up, while still utilizing features of the fine scale model, hence maintaining the geological description. This

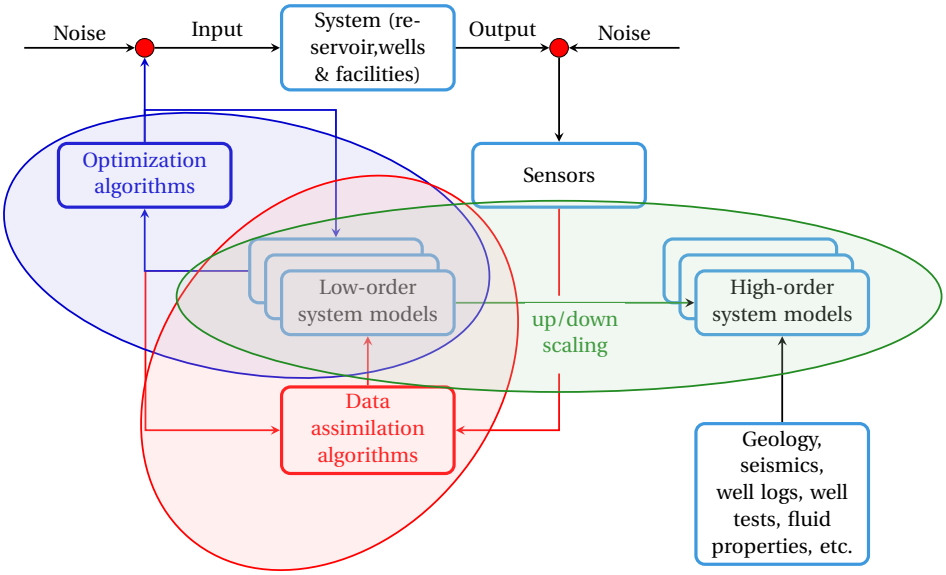


Figure 1.1: Closed-loop reservoir management. Adapted from [7].

provides an accurate and physically consistent solution of the reservoir model which is a distinct advantage over the aforementioned methods.

In addition to the performance/accuracy dilemma, one problem that arises when assimilating those various types of data into a reservoir simulation model is that those data are available in completely different representation scales (i.e. different resolutions). For instance, the production data is, of course, a result of the various microscope phenomena involved in the fluids displacement in the porous media, in contrast to the seismic resolution, that can typically have a high horizontal resolution (in the order of tens of meters), but is not unusual to observe lower vertical resolution (in the order of meters), but never on the microscope level. Hence, it is given that when assimilating data represented in different resolutions some sort of manipulation of the data is necessary, which certainly lead to loss of important information. Multiscale, with its ability to represent and solve the same reservoir discretized model in different scales, should handle the scale change issues.

In the light of those challenges, next we introduce the research objectives in section 1.1, followed by the fundamental concepts that form the basis for the developments in this thesis.

1.1. RESEARCH OBJECTIVES

The aim of this PhD research is twofold: to investigate the utilization of MS techniques as an efficient and accurate forward simulation alternative, also allowing the assimilation of production data in the scale they become available. More specifically, the main purpose of the research is to develop multiscale-based methods applied to closed-loop reservoir

management studies and shed lights on the following research questions:

1. **How can the optimization algorithms applied in the CLRM workflow benefit from MS-based forward models?**

   We aim to employ MS simulation techniques to address the performance/accuracy dilemma. In other words, we focus on taking advantage of MS methods ability to solve the flow problem in a reduced representation of the system, while providing accurate fine-scale solutions.

2. **Can we design improved data assimilation and optimization strategies based on multiscale simulation principles?**

   Specifically, we focus on the utilization of multiscale techniques not only from the forward simulation point of view. Instead, because the forward simulation is multiscale, we aim to develop multiscale analytical derivative computation methods which are efficient and accurate to be used in gradient-based optimization techniques.

3. **Can multiscale methods address the multi-level nature of inverse problems?**

   We are interested in addressing the spatial scale dissimilarity between simulation model and observed data from a multiscale simulation perspective, so that upscaling/downscaling and consequential lost of information is avoided.

1.2. FORWARD MODELING

The development of the reservoir model reservoir simulator involves the numerical solution of the governing equations describing the flow of fluids in a porous media. The governing equations are, basically, the mass balance for all the phases present in the system and the momentum balance for fluid substituted by Darcy's law [18]. The discretized equations can be generically expressed in residual form in terms of the mass balance individual operators for the mass accumulation, flux, and source terms as [19]

$$\begin{aligned}\mathbf{g}\left(\mathbf{x}^{n+1},\mathbf{x}^{n}\right) &= \frac{\mathbf{m}\left(\mathbf{x}^{n+1}\right)-\mathbf{m}\left(\mathbf{x}^{n}\right)}{\Delta t}+\mathbf{f}\left(\mathbf{x}^{n+1}\right)+\mathbf{q}\left(\mathbf{x}^{n+1}\right)\\ &= \dot{\mathbf{m}}\left(\mathbf{x}^{n+1},\mathbf{x}^{n}\right)+\mathbf{f}\left(\mathbf{x}^{n+1}\right)+\mathbf{q}\left(\mathbf{x}^{n+1}\right)=\mathbf{0}.\end{aligned} \tag{1.1}$$

Let N_x be the total number of primary variables that needs to be solved for at the time-step n. In Eq. (1.1), $\mathbf{g}\in\mathbb{R}^{N_x}$ is the model equation vector, $\mathbf{x}\in\mathbb{R}^{N_x}$ the state vector, $\dot{\mathbf{m}}\in\mathbb{R}^{N_x}$ is the mass accumulation vector, $\mathbf{f}\in\mathbb{R}^{N_x}$ in the flux vector, and $\mathbf{q}\in\mathbb{R}^{N_x}$ the source terms vector. In the scope of this thesis, as it is commonly done in the literature, pressure and phases saturations are the state variables (natural variable formulation).

We employ a backward Euler discretization in time in Eq. (1.1). Different spatial discretization schemes can be used to numerically discretize the governing equations [20], however locally mass conservative schemes are preferred and the Finite Volume method is the most practiced in reservoir simulation [21–23]. The resulting non-linear system of equations is solved via an iterative numerical scheme, usually Newton's method [24], as

$$\frac{\partial\mathbf{g}\left(\mathbf{x}^{k}\right)}{\partial\mathbf{x}}\left(\mathbf{x}^{k+1}-\mathbf{x}^{k}\right)=-\mathbf{g}\left(\mathbf{x}^{k}\right), \tag{1.2}$$

where k is the Newton iteration index. Eq. (1.2) requires the computation of derivatives of the discretized residual equations (Eq. (1.1)) with respect to the primary variables $\mathbf{x}$, i.e.,

$$\frac{\partial \mathbf{g}}{\partial \mathbf{x}} = \frac{\partial \dot{\mathbf{m}}}{\partial \mathbf{x}} + \frac{\partial \mathbf{f}}{\partial \mathbf{x}} + \frac{\partial \mathbf{q}}{\partial \mathbf{x}}, \tag{1.3}$$

where $\frac{\partial \bullet}{\partial \mathbf{x}} \in \mathbb{R}^{N_x \times N_x}$ represents the first order partial derivative matrices w.r.t. the primary variables of Eq. (1.1).

1.2.1. PARTIAL DERIVATIVE MATRICES COMPUTATION

As discussed in [25], the computation of partial derivative matrices can be performed by, basically, three different ways that, in summary, have the following pros and cons: (1) manual differentiation [26], efficient, but error prone and not flexible; (2) numerical differentiation [24], flexible, but not efficient; (3) automatic differentiation (AD) [27]: flexible and general, but efficient implementation is complex. In addition to this list, more recently, the Operator Based Linearization technique (OBL) was [28] introduced. In summary, it creates tables for the individual terms of Eq. (1.1) and performs the linearization by table interpolation.

The choice of a specific calculation method relies on the assessment of a number of factors that one wants to prioritize, like flexibility, efficiency, ease of implementation, among others. For instance, although it consists of a rather flexible framework, the reservoir simulator presented by [29] computes the Jacobian matrix either manually or numerically, avoiding eventual performance issues that one might face when performing the derivative calculation by AD. On this matter, the efficiency of the derivative computations can be preserved by tuning the automatic differentiation calculations using a diversity of computational implementation techniques [25].

1.2.2. TIME DISCRETIZATION AND COUPLING STRATEGIES

Different coupling strategies can be employed, requiring different numerical solution strategies. Fully implicit (FIM) strategies solve for all primary variables in the same (nonlinear) system of equations. The discrete-in-time nonlinear model equations represented in Eq. (1.1) can be represented in the following implicit (residual) form

$$\begin{cases} \mathbf{g}^n_{\mathbf{x}_1}\left(\mathbf{x}_1^{n-1},\ldots,\mathbf{x}_{N_c}^{n-1},\mathbf{x}_1^{n},\ldots,\mathbf{x}_{N_c}^{n},\theta\right) = \mathbf{0} \\ \qquad\qquad \vdots \\ \mathbf{g}^n_{\mathbf{x}_{N_c}}\left(\mathbf{x}_1^{n-1},\ldots,\mathbf{x}_{N_c}^{n-1},\mathbf{x}_1^{n},\ldots,\mathbf{x}_{N_c}^{n},\theta\right) = \mathbf{0}, \end{cases} \tag{1.4}$$

where $\mathbf{x}_c^n \in \mathbf{R}^{N^n_{\mathbf{x}_c}}$, $c \in \{1,\ldots,N_c\}$, is the set of primary variables associated with the model equations $\mathbf{g}^n_{\mathbf{x}_c}$, and N_c the total number of coupled equations. Such coupling has the advantage of being more stable (allow for larger time-steps) but the solution of the resulting system of equations requires handling larger linear systems.

Sequential strategies suppose a weakly-coupled system, allowing for pressure and saturation(s) to be solved by different system of equations with the dependency of pressure in saturation lagged in time. This strategy allows the employment of the best numerical

solution strategies for each of the physical phenomenon (flow and transport), however suffer from stability issues for strongly coupled systems, requiring smaller time-steps [30].

1.3. MULTISCALE RESERVOIR SIMULATION

Multiscale methods, in particular MS Finite Volume methods (MSFV) [16], are simulation techniques that aim the solution of the flow problem with a lower computational cost by incorporating the fine-scale model features into a coarse-scale operator [31–33], i.e.,

$$\mathbf{x} \approx \mathbf{x}' = \mathbf{P}\left(\mathbf{RAP}\right)^{-1}\mathbf{Rq} = \mathbf{P}\check{\mathbf{A}}^{-1}\check{\mathbf{q}}. \tag{1.5}$$

Let N_F be the total number of cells in the discretized fine-scale model and N_C be the total number of cells employed to build the coarse-scale system. So, $\mathbf{x}' \in \mathbb{R}^{N_F}$ is the approximate fine-scale solution, $\mathbf{R} \in \mathbb{R}^{N_C \times N_F}$ the restriction operator, $\mathbf{A} \in \mathbb{R}^{N_F \times N_F}$ the fine-scale system matrix, $\mathbf{P} \in \mathbb{R}^{N_F \times N_C}$ the prolongation operator, and $\check{\mathbf{A}} \in \mathbb{R}^{N_C \times N_C}$ the coarse-scale system matrix. This is achieved via the representation of the fine-scale model in a coarser-scale (restriction) and via the introduction of basis functions, which are obtained via the solution of the flow equation

$$-\nabla \cdot \left(\lambda \cdot \nabla \varphi_i\right) = 0, \tag{1.6}$$

in smaller domains and with special prescribed values

$$\delta_{ij} = \begin{cases} 1, \text{if } i = j \\ 0, \text{if } i \neq j \end{cases} \tag{1.7}$$

where $\lambda = \mathbf{K}/\mu$, $\mathbf{K}$ is the absolute permeability tensor, μ the fluid viscosity, φ_i is the i-th basis function, and δ_{ij} is the i-th basis function value at the j-th vertex. The basis functions will compose the prolongation operator

$$\mathbf{P} = \begin{bmatrix} \boldsymbol{\varphi}_1 & \boldsymbol{\varphi}_2 & \cdots & \boldsymbol{\varphi}_{N_C-1} & \boldsymbol{\varphi}_{N_C} \end{bmatrix}, \tag{1.8}$$

which is responsible for bringing the coarse-scale solution back to the fine-scale discretization (prolongation). In Eq. (1.8), $\boldsymbol{\varphi} \in \mathbb{R}^{N_F}$ is the vector with the basis function values. The construction of the prolongation operator involves the definition of a primal coarse grid (where the coarse solution will be obtained) and a dual coarse grid, on whose grid blocks the basis functions will be computed on. These grid concepts are represented in Fig. 1.2. In the MS Finite Volume setting [16], the restriction operator corresponds to the integral over the each primal-coarse block, i.e.,

$$\mathbf{R}_{i,j} = \begin{cases} 1, \text{if fine-cell } j \text{ belongs to primal-coarse block } i \\ 0, \text{otherwise.} \end{cases} \tag{1.9}$$

Because the local flow problems solved to compute the basis functions do not take into account the source terms, well functions [34] were introduced to capture the effects of the source term in the approximated MS solution. The basis function and well

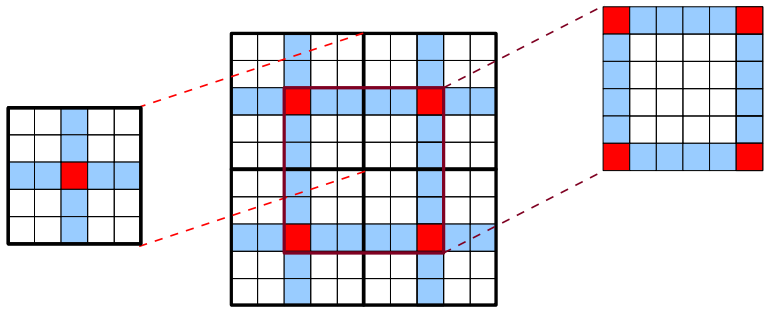


Figure 1.2: Griding elements, in 2D, for the construction of the MSFV restriction and prolongation operators. Super-imposed to the fine grid (in the center) is drawn (in bold lines) the primal coarse grid. In the left-hand side a primal coarse grid cell is illustrated. The cells in red, the centroids of the primal grid blocks, are the vertices of the dual grid blocks (represented on the right hand side). The vertices are them connected to their neighbors, defining the edges of the dual grid cells. The result is the dual coarse grid.

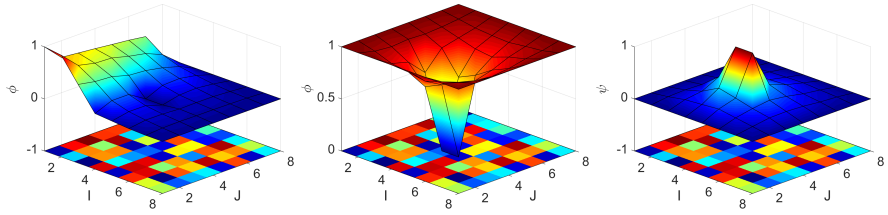


Figure 1.3: (a) Basis function, (b) sum of all basis functions and (c) well function for a given heterogeneous porous media (represented in the bottom of the plots) in a given dual-grid cell containing a well (source term). The well perforates two fine grid blocks in the center of the dual grid block.

function are illustrated in Fig. 1.3. The computation of well/basis functions are only performed once in the simulation, as a pre-processing step. Because their construction is completely independent, the computation is highly parallelizable [35].

Due to the localization assumptions utilized to compute the basis functions, the pressure solution obtained via a MSFV scheme is not as accurate as the fine scale solution. However, these discrepancies can be resolved if an iterative scheme is employed [36]. The Iterative Multiscale Finite Volume method (i-MSFV) is capable of resolving these differences by resolving the high frequency errors via some iterations in the fine scale and resolving the low frequency errors via the MSFV coarse scale solution. In brief, the method consists of re-writing the MSFV in residual form

$$\delta \mathbf{x}'^{\nu} = \mathbf{P}\left(\mathbf{RAP}\right)^{-1}\mathbf{R}\mathbf{r}^{\nu-1} \tag{1.10}$$

which is suitable to be solved by an iterative scheme as

$$\delta \mathbf{x}_{o}^{\nu} = \mathbf{M}\mathbf{r}_{o}^{\nu}, \tag{1.11}$$

where

$$\mathbf{r}^{\nu} = \mathbf{q} - \mathbf{A}\mathbf{x}'^{\nu-1}, \tag{1.12}$$

and

$$\mathbf{r}_\sigma^\nu = \mathbf{q} - \mathbf{A}\mathbf{x}'^\nu, \tag{1.13}$$

so that

$$\mathbf{x}'^\nu = \mathbf{x}'^{\nu-1} + \delta\mathbf{x}'^\nu + \delta\mathbf{x}_\sigma^\nu. \tag{1.14}$$

Here, ν is the i-MSFV iteration index, $\mathbf{r} \in \mathbb{R}^{N_F}$ and $\mathbf{r}_\sigma \in \mathbb{R}^{N_F}$ are residual vectors and $\delta\mathbf{x}' \in \mathbb{R}^{N_F}$ and $\delta\mathbf{x}_\sigma \in \mathbb{R}^{N_F}$ are, respectively, the approximate fine-scale and the smoothing solution corrections. The Eq. (1.11) is solved iteratively [37] until a convergence criterion is met, where $\mathbf{M} \in \mathbb{R}^{N_F \times N_F}$ is a factorization of $\mathbf{A}$ which construction depends on the iterative solver employed in the smoothing step.

The improved accuracy that can be reached is illustrated in Fig. 1.4. In this simulation example [15], one quarter of 5-spot pattern with injection and production pressures of, respectively, 2 Pa and 1 Pa, a fine scale (a 21x21 grid size) solution accuracy is obtained after sufficiently iterating in a i-MSVF scheme (with a 3x3 grid size). In this case, 7 iterations are necessary to reach a residual convergence tolerance of 10e-7.

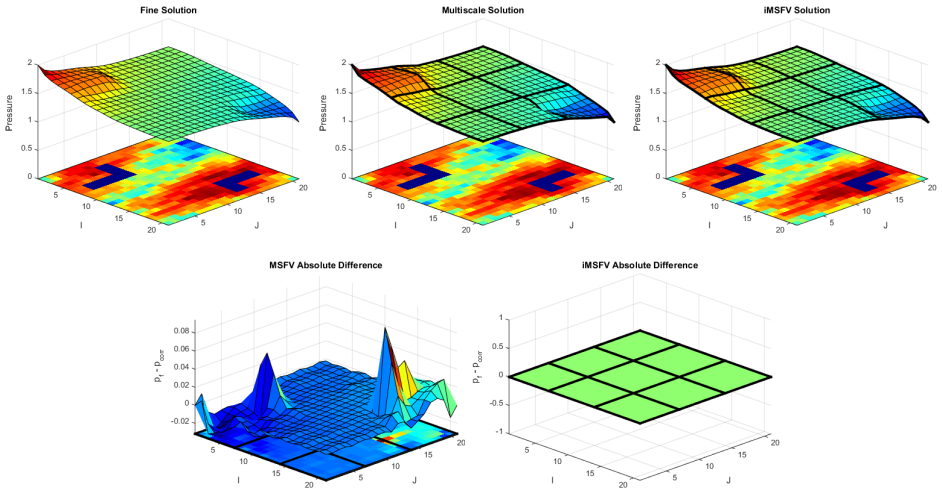


Figure 1.4: Improved solution accuracy by applying i-MSFV. (a) Fine scale pressure solution. (b) MSFV pressure solution. (c) i-MSFV pressure solution. (d) Absolute error between fine and MSFV solution. (e) Absolute error between fine and i-MSFV solution. In this case, 7 iterations are necessary to reach a residual convergence tolerance of 10e-7. In plots (a), (b) and (c) the permeability distribution is shown in the bottom and the surface represents the pressure distribution (in Pa). The fine lines represent the fine grid cells, whereas the bold lines represent the coarse grid blocks.

Although the MSFV approximated solution, by design, is conservative at the coarse scale, it is not conservative in the fine scale. And because we must use the pressure solution to compute a velocity field which will be used to solve the transport equation, it is important that a conservative velocity field is obtained before the saturation distribution is computed. For such aim, an additional local problem must be solved to obtain a conservative fine-scale velocity field [38]. The additional problem, some sort of a post-processing of the approximated MSFV pressure solution, provides a new pressure solu-

tion, which is conservative, by solving Neumann problems at the primal grid cells with the coarse velocity field prescribed as boundary condition

$$u'' = \begin{cases} -\lambda \cdot \nabla p''_k \text{ on } \check{\Omega}_k \\ -\lambda \cdot \nabla p' \text{ at } \partial\check{\Omega}_k \end{cases}, \tag{1.15}$$

where u'' is the interfacial conservative velocity, p'' the conservative pressure computed for the cell at the interior primal-grid domain $\check{\Omega}$ and p' the approximate pressure (given by Eq. (1.5)) at primal grid interfaces $\partial\check{\Omega}$. The new velocity field u'' can be then used in the transport equation solution. The idea is illustrated in Fig. 1.5.

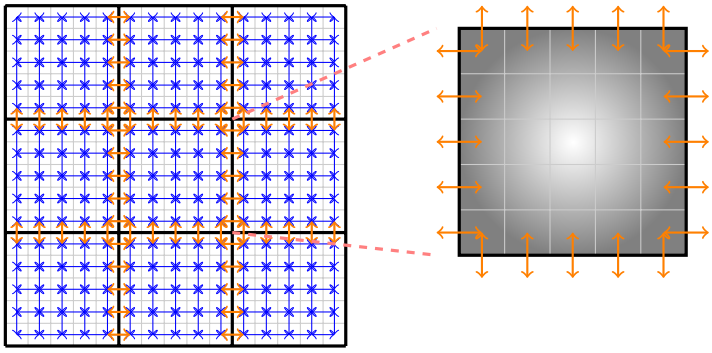


Figure 1.5: Illustration of conservative velocity field reconstruction. The orange arrows represent the interfacial velocity orthogonal components computed from the (conservative) coarse-scale solution $\mathbf{p}'$. These components are utilized as boundary conditions in the construction of $\mathbf{p}''$, represented by the gray-scale pressure field (right). The blue arrows represent the interfacial velocity orthogonal components computed from $\mathbf{p}''$.

When the MSFV method was described, it was mentioned that the basis functions can only be computed in the beginning of the simulation, as a pre-processing step to the time-stepping process. But because of total mobility changes due to saturation changes in the multiphase flow, the basis function might not be as accurate as in the beginning of the simulation given that it is numerically computed taking into account the total mobility. Re-construction of the basis function can be performed adaptively, as opposed to the computation of all basis function for all dual grid cells, a much more computationally expensive approach, by only tracking the most prominent total mobility changes in the fine scale and then re-computing the basis function of the dual grid cells that contain the fine grid cells which change in mobility in time exceed a given (user specified) tolerance [38].

1.4. DATA ASSIMILIATION

Data assimilation, parameter estimation, or as more often referred to in the Petroleum field, History Matching, can be described in the inverse problem theory presented in [5]. Given a set of observable responses

$$\mathbf{y} = \mathbf{h}(\mathbf{x}, \theta), \tag{1.16}$$

computed from the forward model equations, where $\mathbf{y} \in \mathbb{R}^{N_y}$, an inverse problem can be determined so that observations from the real system can be used to estimate the N_θ model parameters $\theta \in \mathbb{R}^{N_\theta}$ by 'solving'

$$\mathbf{d}_{obs} = \mathbf{h}(\mathbf{x}, \theta) + \epsilon, \tag{1.17}$$

where $\mathbf{d}_{obs} \in \mathbb{R}^{N_y}$ is a vector of real observations, $\epsilon \in \mathbb{R}^{N_y}$ is a vector representing the errors associated with the observed measurements (see Fig. 1.1). However, due to modelling errors of different sources, represented by $\eta \in \mathbb{R}^{N_y}$, it follows that

$$\mathbf{d}_{obs} = \mathbf{h}_F(\mathbf{x}, \theta) + \eta + \epsilon. \tag{1.18}$$

Assuming η and ϵ are independent and Gaussian [6], it follows that

$$\mathbf{e} = \eta + \epsilon \tag{1.19}$$

is Gaussian with mean given by the N_d zero vector and covariance

$$\mathbf{C}_D = \mathbf{C}_{D\eta} + \mathbf{C}_{D\epsilon}. \tag{1.20}$$

In most application of inverse theory to subsurface model parameter estimation, only $C_{D\epsilon}$ is properly represented, and, eventually, the modelling errors compensated by considering larger values of observational errors. In other words, the modeling error is neglected and its impact arbitrarily counteracted by a larger ϵ.

The problem of estimating the θ from $\mathbf{d}_{obs}$ (Eq. (1.17)) can be expressed, in a probabilistic setting, via Bayes' rule as [6]

$$f(\theta, \mathbf{d}_{obs}) = \frac{f(\mathbf{d}_{obs}|\theta)\, f(\theta)}{f(\mathbf{d}_{obs})}. \tag{1.21}$$

In the Bayesian framework considered in this thesis, we assume the *a priori* probabiblity density function $f(\mathbf{d}_{obs}|\theta)$ to be Gaussian, so that

$$f(\theta) \propto e^{-\frac{1}{2}(\theta-\theta_{prior})^T \mathbf{C}_\theta^{-1}(\theta-\theta_{prior})}, \tag{1.22}$$

where C_θ is the *a priori* model parameter covariance matrix and θ_{prior} the *a priori* model parameter distribution. Also, following the assumption of normally distributed measurement errors, and assuming the measurement errors independent of θ_{prior}, $f(\theta, \mathbf{d}_{obs})$, the conditional *a posteriori* distribution, follows [6]

$$f(\theta, \mathbf{d}_{obs}) \propto e^{-\frac{1}{2}(\theta-\theta_{prior})^T \mathbf{C}_\theta^{-1}(\theta-\theta_{prior}) + \frac{1}{2}(\mathbf{h}(\mathbf{x},\theta)-\mathbf{d}_{obs})^T \mathbf{C}_D^{-1}(\mathbf{h}(\mathbf{x},\theta)-\mathbf{d}_{obs})}. \tag{1.23}$$

The maximum *a posteriori* (MAP) model parameters are the ones that maximize $f(\theta, \mathbf{d}_{obs})$. Hence, we can define an objective function

$$\begin{aligned} O(\mathbf{h}, \theta) = &\frac{1}{2}\left(\theta - \theta_{prior}\right)^T \mathbf{C}_\theta^{-1}\left(\theta - \theta_{prior}\right) + \\ &\frac{1}{2}\left(\mathbf{h}(\mathbf{x}, \theta) - \mathbf{d}_{obs}\right)^T \mathbf{C}_D^{-1}\left(\mathbf{h}(\mathbf{x}, \theta) - \mathbf{d}_{obs}\right), \end{aligned} \tag{1.24}$$

such that the MAP estimate can be obtained as an optimization problem

$$\begin{aligned} \underset{\theta}{\text{minimize}} \quad & O(\boldsymbol{h}(\theta)) \\ \text{subject to} \quad & \boldsymbol{g}(\boldsymbol{x},\theta) = \mathbf{0}, \\ & \theta \in [\theta_{min}, \theta_{max}]. \end{aligned} \tag{1.25}$$

1.5. LIFE-CYCLE OPTIMIMIZATION

Life-cycle optimization concerns the maximization of a given economic objective (e.g. net present value or recovery factor), by manipulating the input variables, i.e. the control parameters (e.g. the well bottom hole pressures or rates) [15], often subject to operational constraints (e.g. maximum water-cut or maximum gas-oil ratio). This optimization problem can be expressed as

$$\begin{aligned} \underset{\theta}{\text{maximize}} \quad & O(\boldsymbol{h}(\theta)) \\ \text{subject to} \quad & \boldsymbol{g}(\boldsymbol{x},\theta) = \mathbf{0}, \\ & \mathbf{c}(\boldsymbol{x},\theta) = \mathbf{0}, \\ & \mathbf{d}(\boldsymbol{x},\theta) < \mathbf{0}, \\ & \theta \in [\theta_{min}, \theta_{max}], \end{aligned} \tag{1.26}$$

where $\mathbf{c}$ and $\mathbf{d}$ are, respectively, equality and inequality constraint vectors and, now, the objective function O expresses the economic objective one aims to maximize and can be generically expressed as

$$O(\boldsymbol{h}(\theta)) = \sum_{n=1}^{N} O_n(\mathbf{h}_n(\theta)), \tag{1.27}$$

where O_n represents the contribution to O in each time-step n (e.g. produced oil revenues and injection water costs) during that time interval.

1.6. FORWARD MODEL'S RESPONSES DERIVATIVE COMPUTATION

Different derivative computation methods can be employed to compute Eq. (1.29). Analytical methods (namely the Direct [6] and Adjtoint [39] methods) are described in the literature as the most efficient and accurate ones. However, due to their implementation complexity and inflexibility, stochastic derivative methods, e.g. EnOpt [3], StoSAG [40] for life-cycle optimization, and Kalman filter and its derivations, e.g. EnKF [41], ES-MDA [42], have become popular choices for the solution of the problems described by Eq. (1.26) and Eq. (1.25), respectively.

In the case that a gradient-based optimization technique is chosen as an alternative to solve Eq. (1.25) and Eq. (1.25), given any objective function

$$O = O(\mathbf{h}(\mathbf{x},\theta)), \tag{1.28}$$

the gradient w.r.t. the parameters reads

$$\nabla_{\theta} O = \left(\frac{dO}{d\theta}\right)^T = \left(\frac{dO}{d\mathbf{h}}\frac{d\mathbf{h}}{d\theta}\right)^T = \mathbf{G}^T \nabla_{\mathbf{h}} O, \tag{1.29}$$

where $\mathbf{G} \in \mathbb{R}^{N_y \times N_\theta}$ is the so-called sensitivity matrix [6].

1.7. IMPLEMENTATION NOTES AND SOFTWARE

A brief overview of the software employed in the developments of this thesis, as well as high-level consideration about the implementation is provided in this section. The ideas employed in this development are largely similar to the ideas presented in [43–45].

The aspects of the forward modeling, including the MS simulation strategies, as well as the data assimilation and life-cycle optmization algorithms presented in the previous sections were implemented in a research oriented, flexible/extensible reservoir simulator that was used in the CLRM numerical experiments presented in this thesis. The simulator was designed based on object-oriented programming and generic programming techiques and implemented using the C++ programming language [46].

With respect to extensibility, the AD method via the operator overloading technique [47] was employed not only in the partial derivative computation required by Eq. (1.3), but also to allow for easier computation of derivative information required by gradient-based optimization techniques. The choice mainly relies on the fact that, once the base computational framework is in place, the computation of the derivatives 'comes for free'. This is an important factor when developing CLRM workflows that by itself is largely based on optimization techniques that make extensive usage of derivatives. The FAD-BAD++ library [48] was employed in the operator overloading AD. It employs extensive expression templates and meta-programming techniques [49] to provide differentiability to scalar data-types. Although its usage into reservoir simulation is limited performance-wise [25], in the scope of this thesis it was not considered a severe drawback. However the usage of other libraries could be considered given the extensive utilization of generic program in the code implementation.

With respect to the spatial numerical discretization that leads to Eq. (1.1), a two-point flux approximation is used to approximate the interfacial flux. Upwind weighting is utilized to evaluate the fluid related properties at the interface, while harmonic average is used to evaluate the rock properties at the interface. With respect to time discretization and coupling between flow and transport, the fully implicit (FIM), implicit pressure explicit saturation (IMPES) and sequential implicit strategies have been implemented as requirements for the developments present in this thesis. Details on all these methods can be found in [21].

Additionally, the Eigen C++ library [50] was utilized for basic linear algebra operations and data structures, algorithms and linear solvers. The concepts presented by [51] were employed in the gridding module of the simulator, allowing for the efficient implementation of grid related algorithms (e.g. Jacobian assembly) and flexible grid representation.

1.8. THESIS OUTLINE

In the view of the research objectives defined in Section 1.1 and the mathematical modeling strategies just discussed, the remainder of this thesis is organized in chapters which aim to address the CLRM challenges as follows.

Chapter 2 This chapter presents the fundamental mathematical framework for the analytical computation of forward model resposes' derivatives, suitable to efficiently solve Eq. (1.29). This is the basis for the remaining developments of this thesis. Even though all applications of the framework in the scope of this thesis requires the computation of objective function gradients, it is shown how different derivative information can be computed via the generic framework.

Chapter 3 This chapter discusses how the derivative information required to solve the optimization problems defined in Eq. (1.25) and Eq. (1.26) is computed when a sequentially coupled system of equation as in Eq. (1.4) is employed in the forward simulation. More specifically, we address the gradient computation of sequentially coupled flow and transport equations.

Chapter 4 This chapter discusses the newly introduced MS gradient computation method employed to solve Eq. (1.29) efficiently when a MS simulation strategy (Eq. (1.5)) is employed in the forward simulation. Applications focused on the solution of Eq. (1.24) illustrates the performance of the method.

Chapter 5 This chapter discusses the extension of the MS gradient computation presented in 4 to multiphase flows. This requires the definition of the derivative model associated to Eq. (1.15). More specifically, we focus on the efficient gradient computation of sequentially coupled flow and transport equations, as presented in Chapter Eq. (3), by employing MS simulation strategies.

Chapter 6 This chapter discusses how the quality of the MS gradient computed via the strategy presented in Chapter 4 can be improved by employing a i-MSFV forward simulation. This is achieved by augmenting the the MS derivative model with Eq. (1.10). The performance of the method is shown in challenging geological settings.

Chapter 7 The framework developed in Chapter 4 is applied to the assimilation of spatially distributed data. To that end, a multiscale objective function, a variant of Eq. (1.24) is introduced, where the regularization term is kept at the original parameter description scale and the data misfit term is kept at the observation scale. It is shown how more robust the solution of Eq. (1.25) can be, specially in the context of uncertainty quantification.

Appendix A A discussion on how stochastic derivative methods can take advantage of MS methods is presented. A multiscale-Stochastic simplex approximate gradient – MS-StoSAG – workflow is employed to the solution of Eq. (1.26), where all required forward

simulation are performed using the concepts discussed in Section 1.3 and the computation of Eq. (1.29) peformed by the StoSAG algorithm.

Concluding remarks, as well as future research perspectives, are summarized at Chapter 8. Directions on how to address the research questions posed in section 1.1, in the view of the developements presented in this thesis, are presented in section 8.2. Also, a multiscale version of the CLRM as presented in Fig. 1.1 based on the results achieved in this research project.

REFERENCES

[1] J.-D. Jansen, D. Brouwer, G. Naevdal, and C. Van Kruijsdijk, *Closed-loop reservoir management,* First Break **23**, 43 (2005).

[2] P. Sarma, K. Aziz, and L. J. Durlofsky, *Implementation of adjoint solution for optimal control of smart wells,* in *SPE reservoir simulation symposium* (Society of Petroleum Engineers, 2005).

[3] Y. Chen, D. S. Oliver, and D. Zhang, *Efficient ensemble-based closed-loop production optimization,* SPE J. **14**, 634 (2009).

[4] C. Wang, G. Li, A. C. Reynolds, *et al.*, *Production optimization in closed-loop reservoir management,* SPE journal **14**, 506 (2009).

[5] A. Tarantola, *Inverse problem theory and methods for model parameter estimation*, Vol. 89 (siam, 2005).

[6] D. S. Oliver, A. C. Reynolds, and N. Liu, *Inverse theory for petroleum reservoir characterization and history matching* (Cambridge University Press, 2008).

[7] J.-D. Jansen, O. H. Bosgra, and P. M. Van den Hof, *Model-based control of multiphase flow in subsurface oil reservoirs,* Journal of Process Control **18**, 846 (2008), http://dx.doi.org/10.1016/j.jprocont.2008.06.011.

[8] J. F. van Doren, R. Markovinović, and J.-D. Jansen, *Reduced-order optimal control of water flooding using proper orthogonal decomposition,* Computational Geosciences **10**, 137 (2006), http://dx.doi.org/10.1007/s10596-005-9014-2.

[9] M. Cardoso, L. Durlofsky, and P. Sarma, *Development and application of reduced-order modeling procedures for subsurface flow simulation,* International journal for numerical methods in engineering **77**, 1322 (2009).

[10] M. Cardoso and L. Durlofsky, *Linearized reduced-order models for subsurface flow simulation,* Journal of Computational Physics **229**, 681 (2010).

[11] J. D. Jansen and L. J. Durlofsky, *Use of reduced-order models in well control optimization,* Optim. Eng. , 1 (2016).

[12] C. Farmer, *Upscaling: a review,* International journal for numerical methods in fluids **40**, 63 (2002).

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

[14] L. J. Durlofsky, *Upscaling and gridding of fine scale geological models for flow simulation,* in *8th International Forum on Reservoir Simulation Iles Borromees, Stresa, Italy*, Vol. 2024 (2005).

[15] J. D. Jansen, *Adjoint-based optimization of multi-phase flow through porous media–a review,* Computers & Fluids **46**, 40 (2011).

[16] P. Jenny, S. H. Lee, and H. A. Tchelepi, *Multi-scale finite-volume method for elliptic problems in subsurface flow simulation,* J. Comput. Phys. **187**, 47 (2003).

[17] T. Y. Hou and X.-H. Wu, *A multiscale finite element method for elliptic problems in composite materials and porous media,* J. Comput. Phys. **134**, 169 (1997).

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

[19] *ECLIPSE 100 Technical description*, Schlumberger.

[20] R. J. LeVeque, *Conservative methods for nonlinear problems,* in *Numerical Methods for Conservation Laws* (Springer, 1990) pp. 122–135.

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

[22] L.-K. Fung, A. Hiebert, L. X. Nghiem, *et al.*, *Reservoir simulation with a control-volume finite-element method,* SPE Reservoir Engineering **7**, 349 (1992).

[23] D. W. Peaceman, *Fundamentals of numerical reservoir simulation*, Vol. 6 (Elsevier, 2000).

[24] T. H. Michael, *Scientific computing: an introductory survey* (The McGraw-Hill Companies Inc.: New York, NY, USA, 2002).

[25] R. Younis and K. Aziz, *Parallel automatically differentiable data-types for next-generation simulator development,* in *SPE Reservoir Simulation Symposium* (Society of Petroleum Engineers, 2007).

[26] T. Ertekin, J. H. Abou-Kassen, and G. R. King, *Basic applied reservoir simulations* (Society of Petroleum Engineers, 2001).

[27] L. B. Rall, *Automatic differentiation: Techniques and applications,* (1981).

[28] D. V. Voskov, *Operator-based linearization approach for modeling of multiphase multi-component flow in porous media,* Journal of Computational Physics **337**, 275 (2017).

[29] A. D. Hiebert, M. Khoshkbarchi, P. H. Sammon, I. N. Alves, J. Rodrigues, A. J. Belien, B. Howell, F. E. Saaf, P. Valvatne, *et al.*, *An advanced framework for simulating connected reservoirs, wells and production facilities,* in *SPE Reservoir Simulation Symposium* (Society of Petroleum Engineers, 2011).

[30] K. Coats, *Impes stability: selection of stable timesteps,* SPE Journal **8**, 181 (2003).

[31] H. Hajibeygi, *Iterative multiscale finite volume method for multiphase flow in porous media with complex physics*, Ph.D. thesis, ETH Zurich (2011).

[32] H. Zhou and H. A. Tchelepi, *Two-stage algebraic multiscale linear solver for highly heterogeneous reservoir models,* SPE J. **17**, 523 (2012).

[33] Y. Wang, H. Hajibeygi, and H. A. Tchelepi, *Algebraic multiscale solver for flow in heterogeneous porous media,* J. Comput. Phys. **259**, 284 (2014).

[34] P. Jenny and I. Lunati, *Modeling complex wells with the multi-scale finite-volume method,* J. Comput. Phys. **228**, 687 (2009).

[35] A. M. Manea, J. Sewall, H. A. Tchelepi, *et al.*, *Parallel multiscale linear solver for highly detailed reservoir models,* SPE Journal **21**, 2 (2016).

[36] H. Hajibeygi, G. Bonfigli, M. A. Hesse, and P. Jenny, *Iterative multiscale finite-volume method,* J. Comput. Phys. **227**, 8604 (2008).

[37] Y. Saad, *Iterative methods for sparse linear systems* (SIAM, 2003).

[38] P. Jenny, S. Lee, and H. Tchelepi, *Adaptive multiscale finite-volume method for multiphase flow and transport in porous media,* Multiscale Modeling & Simulation **3**, 50 (2005).

[39] G. Chavent, M. Dupuy, and P. Lemmonier, *History matching by use of optimal theory,* Society of Petroleum Engineers Journal **15**, 74 (1975).

[40] R. M. Fonseca, B. Chen, J. D. Jansen, and A. C. Reynolds, *A stochastic simplex approximate gradient (StoSAG) for optimization under uncertainty,* Int. J. Numer. Meth. Eng. (2016), 10.1002/nme.5342.

[41] S. I. Aanonsen, G. Nævdal, D. S. Oliver, A. C. Reynolds, and B. Vallès, *The ensemble Kalman filter in reservoir engineering–a review,* SPE J. **14**, 393 (2009).

[42] A. A. Emerick and A. C. Reynolds, *Ensemble smoother with multiple data assimilation,* Computers & Geosciences **55**, 3 (2013).

[43] R. Younis, K. Aziz, *et al.*, *Parallel automatically differentiable data-types for next-generation simulator development,* in *SPE Reservoir Simulation Symposium* (Society of Petroleum Engineers, 2007).

[44] D. DeBaun, T. Byer, P. Childs, J. Chen, F. Saaf, M. Wells, J. Liu, H. Cao, L. Pianelo, V. Tilakraj, *et al.*, *An extensible architecture for next generation scalable parallel reservoir simulation,* in *SPE Reservoir Simulation Symposium* (2005).

[45] A. D. Hiebert, M. Khoshkbarchi, P. H. Sammon, I. N. Alves, J. Rodrigues, A. J. Belien, B. Howell, F. E. Saaf, P. Valvatne, *et al.*, *An advanced framework for simulating connected reservoirs, wells and production facilities,* in *SPE Reservoir Simulation Symposium* (Society of Petroleum Engineers, 2011).

[46] B. Stroustrup, *The C++ programming language* (Pearson Education India, 2000).

[47] L. B. Rall and G. F. Corliss, *An introduction to automatic differentiation*, Computational Differentiation: Techniques, Applications, and Tools **89** (1996).

[48] C. Bendtsen and O. Stauning, *FADBAD, a flexible C++ package for automatic differentiation*, Tech. Rep. (Technical Report IMM–REP–1996–17, Department of Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark, 1996).

[49] D. Vandevoorde and N. M. Josuttis, *C++ Templates* (Addison-Wesley Longman Publishing Co., Inc., 2002).

[50] G. Guennebaud, B. Jacob, *et al.*, *Eigen v3*, http://eigen.tuxfamily.org (2010).

[51] G. Berti, *Gral—the grid algorithms library*, Future Generation Computer Systems **22**, 110 (2006).

2

MATHEMATICAL FRAMEWORK FOR THE ANALYTICAL COMPUTATION OF FORWARD MODEL RESPONSES DERIVATIVES

In this work a mathematical framework that addresses the computation of derivative information arising from PDE systems is proposed. It is presented how both direct and adjoint methods are seamlessly addressed. In other words, both algorithms are described in an abstraction level such that any derivative information can be computed given that the proper Jacobians from the forward model are provided. No assumptions with respect to the forward model space or time approximations is made. The mathematical and computational abstractions, implementation details, as well as the overall design of the computational framework are discussed. Applications to different optimization studies (data assimilation and life-cycle optimization), utilizing different forward model numerical discretizations (e.g. sequential couplings and multiscale approaches) illustrate the application of the framework.

The developments presented in this chapter are based on the design document ***Initial Design for Forward and Adjoin Derivative Calculation in DRMS***, written by Dr. José R. P. Rodrigues and Dr. Hans Kraaijevanger for the Dynamic Reservoir Modelling System (DRMS) Joint Venture, composed by Computer Modelling Group (CMG), Shell, and Petrobras. The partner companies' authorization to use parts of this design document in the development of this thesis work is appreciated.

The computation of derivative information is a key factor in many numerical applications. Data assimilation, control optimization, uncertainty quantification, sensitivity analysis are just a few instances of studies that can take advantage of derivative information. In the case of multivariate functionals arising from the numerical discretization of PDEs, the computation of accurate derivative information via analytical (i.e. direct and adjoint) methods is an involved task and often rely on application-based implementations.

Although of uptmost importance, the discussion of how to compute derivative information is often mingled with the dicussion of the optmization algorithm and/or with the optmization problem.

In various areas of engineering discussions have been held about the relative merits of first discretizing the forward flow equations in time and thereafter deriving the discrete-time adjoint equations (the first-discretize-then-differentiate approach), versus first deriving the continuous-time adjoint equations and then discretizing the forward and adjoint equations in time (the first-differentiate-then-discretize approach) [1]. Most authors seem to agree that both methods can be applied as long as the forward and backward equations are truly each others adjoint, which implies discretization at identical moments in time of the forward and backward equations using identical discretization schemes. These aspects were discussed in [2] and more recently re-evaluated in relation to adjoints for multi-component (compositional) simulation [3]. Currently available large-scale reservoir simulation packages all follow the first-discretize-then-differentiate approach.

The original idea of generic Direct and Adjoint algorithms for sensitivity matrix building, sensitivity matrix and transpose sensitivity matrix vector products was originally presented in [4].

The alternative implicit differentiation derivation of the adjoint equations were originally proposed by [4, 5].

2.1. MATHEMATICAL FRAMEWORK FOR ANALYTICAL DERIVATIVE INFORMATION COMPUTATION

The set of non-linear algebraic equations for a given time-step n will be written as

$$\mathbf{g}^n\left(\mathbf{x}^{n-1},\mathbf{x}^n,\theta\right)=\mathbf{0}, \tag{2.1}$$

$\mathbf{g}^n:\mathbb{R}^{N_X^{n-1}\times N_X^n\times N_\theta}\rightarrow\mathbb{R}^{N_X^n}$, where $\mathbf{x}_c^n\in\mathbb{R}^{N_\mathbf{x}^n}$ is the state vector containing all primary variables for time-step n and $\theta\in\mathbb{R}^{N_\theta}$ is the vector of parameters with respect to which we aim to compute the derivative information. The initial conditions will be assumed to be given by

$$\mathbf{g}^0\left(\mathbf{x}^0,\theta\right). \tag{2.2}$$

The functions defining the set of observable responses $\mathbf{y}\in\mathbb{R}^{N_Y}$ for a time-step are described as

$$\mathbf{y}^n=\mathbf{h}^n\left(\mathbf{x}^{n-1},\mathbf{x}^n,\theta\right), \tag{2.3}$$

where $\mathbf{h}^n:\mathbb{R}^{N_X^{n-1}\times N_X^n\times N_\theta}\rightarrow\mathbb{R}^{N_Y^n}$ represents the output equations

All instances of $\mathbf{g}^n$ as defined in Eq. (2.1) for all time-steps can be collated in a function $\boldsymbol{g} : \mathbb{R}^{N_X \times N_\theta} \rightarrow \mathbb{R}^{N_X}$, where $N_X = \sum_{n=0}^{N} N_X^n$ is the total number of primary variables for all time-steps, such that the system of non-linear equations is represented as

$$\boldsymbol{g}(\boldsymbol{x}, \theta) = \mathbf{0}, \tag{2.4}$$

where $\mathbf{x} \in \mathbb{R}^{N_X}$ is the (column) vector of primary variables for all time-steps. Similarly, all instances of $\mathbf{y}^n$ as defined in Eq. (2.3) for all time-steps can be collated in a function $\boldsymbol{h} : \mathbf{R}^{N_X \times N_\theta} \rightarrow \mathbf{R}^{N_Y}$, where $N_Y = \sum_{n=0}^{N} N_Y^n$ represents the total number of responses for all time-steps, so that

$$\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}(\theta), \theta). \tag{2.5}$$

The collated state vector $\mathbf{x}$ is a function of θ through Eq. (2.4), so that we can write $\mathbf{x} = \mathbf{x}(\theta)$. Hence the response vector $\mathbf{y}$ can also be seen as a function of θ only.

The total derivatives of the response vector $\mathbf{y}$ with respect to the parameter vector θ, known as the sensitivity matrix, can be computed by deriving Eq. (2.5) with respect to θ, i.e.,

$$\mathbf{G} = \frac{d\boldsymbol{h}}{d\theta} = \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}} \frac{d\boldsymbol{x}}{d\theta} + \frac{\partial \boldsymbol{h}}{\partial \theta}. \tag{2.6}$$

In order to find a relationship that defines $\frac{d\boldsymbol{x}}{d\theta}$, Eq. (2.4) is differentiated with respect to θ

$$\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}} \frac{d\boldsymbol{x}}{d\theta} + \frac{\partial \boldsymbol{g}}{\partial \theta} = \mathbf{0}, \tag{2.7}$$

so that

$$\frac{d\boldsymbol{x}}{d\theta} = -\left(\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\right)^{-1} \frac{\partial \boldsymbol{g}}{\partial \theta}. \tag{2.8}$$

Substituting Eq. (2.8) in Eq. (2.6) gives

$$\mathbf{G} = -\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}} \left(\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\right)^{-1} \frac{\partial \boldsymbol{g}}{\partial \theta} + \frac{\partial \boldsymbol{h}}{\partial \theta}. \tag{2.9}$$

In many circumstances it is convenient to consider θ as a function of an underlying (smaller) parameter vector $\tilde{\theta}$

$$\theta = \phi(\tilde{\theta}) \tag{2.10}$$

where $\phi : \mathbb{R}^{N_{\tilde{\theta}}} \rightarrow \mathbb{R}^{N_\theta}$. This allows flexibility to easily deal with situations where there are dependencies among the parameters for which the derivatives are being calculated. Examples is reservoir simulation are vertical permeability as a multiple of horizontal permeability, permeability as function of porosity or porosity as function of net-to-gross ratio. Typically, the time-stepping process does not need to know these relationships and, therefore, it is desirable to separate them out of the core derivative calculation algorithm as well.

Likewise, by introducing an arbitrary non-linear function of the responses,

$$\varphi = \varphi(\mathbf{y}) \tag{2.11}$$

it is possible to consider combined responses in a way that is transparent to the forward simulation. Examples in reservoir simulation are net-present value (NPV) and a history matching misfit function.

Taking into account the observations made above, we will be interested in defining algorithms applicable to the modified response function

$$\tilde{\mathbf{y}}(\mathbf{x},\tilde{\theta}) = \varphi(\mathbf{y}(\mathbf{x},\phi(\tilde{\theta}))). \tag{2.12}$$

The sensitivity matrix for $\tilde{\mathbf{y}}$ is given by

$$\tilde{\mathbf{G}} = \frac{d\tilde{\mathbf{y}}}{d\tilde{\theta}} = \frac{\partial\phi}{\partial y}\mathbf{G}\frac{\partial\varphi}{\partial\theta} \tag{2.13}$$

This is the motivation for considering the problem for calculating **WGV** for arbitrary matrices **V** of order $N_\theta \times p$ and **W** of order $m \times N_Y$, where $m = N_\phi$ and $p = N_{\tilde{\theta}}$.

Therefore, any derivative information can be computed if the sensitivity matrix given by Eq. (2.9) is pre- and post-multiplied by arbitrary matrices **W** and **V**

$$\mathbf{WGV} = -\mathbf{W}\frac{\partial \boldsymbol{h}}{\partial \mathbf{x}}\left(\frac{\partial \boldsymbol{g}}{\partial \mathbf{x}}\right)^{-1}\frac{\partial \boldsymbol{g}}{\partial \theta}\mathbf{V} + \mathbf{W}\frac{\partial \boldsymbol{h}}{\partial \theta}\mathbf{V}. \tag{2.14}$$

The key aspect that defines the computational performance of the gradient computation is the order of the operations involving $\left(\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\right)^{-1}$. Based on that, both the direct [6] and adjoint analytical methods to compute the necessary derivative information can be defined.

Hence, the forward method first computes **GV**, whereas the backward method first computes **WG**. Once the matrix **GV** or **WG** has been obtained, the calculation of **WGV** is trivial and does hardly add to the overall computation cost. For this reason, we will consider the forward method as a method for calculating **GV**, and the backward method as a method for calculating **WG**.

Following the discussion above, two algorithms are proposed:

- A forward method to calculate the product **GV**, where **V** is an arbitrary matrix. Since p systems with matrix are solved Eq. (2.17), the cost is proportional to n simulations.
- A backward (adjoint) method to calculate the product **WG**, where **W** is an arbitrary matrix. Since m systems with matrix are solved Eq. (2.21), the cost is proportional to m simulations.

2.1.1. REMARKS ABOUT THE FRAMEWORK

The advantages of this approach are as follows:

1. By playing with **W** and **V** it is possible, with these two algorithms, to address the derivative calculation problems encountered in inverse and optimization problems. In fact, we have

(a) By applying the forward method with $\mathbf{V}$ equal to the identity matrix, we see that we can calculate the sensitivity matrix $\mathbf{G}$ with a cost proportional to the number of parameters (N_θ). Alternatively, by applying the backward method with $\mathbf{W}$ equal to the identity matrix, we see that $\mathbf{G}$ can be computed with a cost proportional to the number of responses (N_Y).

(b) By taking $n = 1$ in the forward method or $p = 1$ in the backward method, $\mathbf{W}$ or $\mathbf{V}$ reduces to a vector, showing that the product of $\mathbf{G}$ or $\mathbf{G}^T$ with an arbitrary vector can be obtained with a cost proportional to a single simulation. (Note that the matrix-vector product $\mathbf{G}^T\mathbf{W}^T$ can be obtained by transposing the product $\mathbf{WG}$.)

(c) If φ in Eq. (2.11) is a scalar ($N_\varphi = 1$), its total derivative with respect to θ, equal to $\dfrac{d\varphi}{d\theta} = \dfrac{\partial\varphi}{\partial\mathbf{y}}\mathbf{G}$, can be obtained using the backward method with a cost proportional to one simulation.

(d) Even other derivative requests could be accommodated under the two algorithms above, allowing for easy incorporation of new methodologies in optimization and history matching. For instance, the history matching misfit function can be viewed as a sum of the misfit for some data series, each one referring to a particular data (well rate, pressure, etc) for several times. By taking N_φ equal to the number of data series, with each component in φ corresponding to one of the data series, the backward method can be used to calculate the gradient for each data series part with a cost proportional to the number of data series.

2. The definition of the objective functions and simulator responses for which derivatives are available is flexible. In fact, since the particular expression for the objective function Eq. (2.11) is totally separated from the derivative calculation algorithm, new objective functions can be easily defined, as long as the gradient with respect to the simulator responses is provided.

3. Once the parameter vector θ for which derivatives are calculated inside the simulator is defined, it is possible to extend it to other parameter vectors $\tilde{\theta}$ in a completely transparent way via the transformation ϕ Eq. (2.10). One can define a new parameter vector $\tilde{\theta}$ and have their derivatives calculated efficiently, as long as they provide the derivatives of the simulator internal parameter vector θ w.r.t. their parameter vector θ (see Eq. (2.13)).

4. As will be clear when the algorithms are fully described, they share the same basic components, differing mainly on how the calculations are done , so that the implementation of both does not represent too much extra effort. As a matter of fact, developing the forward method first would allow for implementing and testing many of the components needed for the more complex backward method in a simpler context, also allowing for cross verification of the results from both implementations. The fact that products with arbitrary matrices $\mathbf{W}$ and $\mathbf{V}$ are being done only adds a very low extra complexity to the implementation, when compared to a specific algorithm, for instance to calculate the gradient through adjoint.

In short, the proposed approach of implementing one forward and one backward algorithm for multiplying **G** from left and right with arbitrary matrices is concise, adding very little complexity when compared to standard gradient calculation through adjoint, and yet is general, being able to meet all requirements from derivative use and allowing for easy extensibility in terms of objective functions.

2

2.1.2. The Forward Method

If **W** is factored out in Eq. (2.14), it can be rewritten as

$$\mathbf{GV} = \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}}\mathbf{Z} + \frac{\partial \boldsymbol{h}}{\partial \theta}\mathbf{V}, \tag{2.15}$$

where

$$\mathbf{Z} = -\left(\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\right)^{-1}\frac{\partial \boldsymbol{g}}{\partial \theta}\mathbf{V}, \tag{2.16}$$

is solved from

$$\left(\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\right)\mathbf{Z} = -\frac{\partial \boldsymbol{g}}{\partial \theta}\mathbf{V}. \tag{2.17}$$

The linear system described in Eq. (2.17) can be re-written in a block-wise form for each time-step n:

$$\left(\begin{array}{c|c|c|c} \frac{\partial \mathbf{g}^0}{\partial \mathbf{x}^0} & & & \\ \hline \frac{\partial \mathbf{g}^1}{\partial \mathbf{x}^0} & \frac{\partial \mathbf{g}^1}{\partial \mathbf{x}^1} & & \\ \hline & \ddots & \ddots & \\ \hline & & \frac{\partial \mathbf{g}^N}{\partial \mathbf{x}^{N-1}} & \frac{\partial \mathbf{g}^N}{\partial \mathbf{x}^N} \end{array}\right)\left(\begin{array}{c} \mathbf{Z}^0 \\ \hline \mathbf{Z}^1 \\ \hline \vdots \\ \hline \mathbf{Z}^N \end{array}\right) = -\left(\begin{array}{c} \frac{\partial \mathbf{g}^0}{\partial \theta}\mathbf{V} \\ \hline \frac{\partial \mathbf{g}^1}{\partial \theta}\mathbf{V} \\ \hline \vdots \\ \hline \frac{\partial \mathbf{g}^N}{\partial \theta}\mathbf{V} \end{array}\right). \tag{2.18}$$

Equation Eq. (2.18) shows that the matrix **Z** can be fully determined by recursively marching forward in time and solving a system with the matrix for each time-step. Once defined how the matrix **Z** can be computed, Eq. (2.18) can be used to compute the sensitivity matrix. The algorithm is described in Algorithm 1.

2.1.3. The Backward Method

Now, if **V** is factored out in Eq. (2.14), the equation can be rewritten as

$$\mathbf{WG} = \mathbf{Z}\frac{\partial \boldsymbol{g}}{\partial \theta} + \mathbf{W}\frac{\partial \boldsymbol{h}}{\partial \theta}, \tag{2.19}$$

where

$$\mathbf{Z} = -\mathbf{W}\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}}\left(\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\right)^{-1} \tag{2.20}$$

Algorithm 1: Right multiplying the sensitivity matrix by an arbitrary matrix via the direct method.

Input : $\frac{\partial \boldsymbol{g}}{\partial \mathbf{x}}, \frac{\partial \boldsymbol{g}}{\partial \theta}, \frac{\partial \boldsymbol{h}}{\partial \mathbf{x}}, \frac{\partial \boldsymbol{h}}{\partial \theta}, \mathbf{V}$

Output: $\mathbf{GV}$

foreach $n = 0, 1, 2, \ldots, N$ **do**

  foreach $j = 1, 2, \ldots, p$ **do**

    Solve for the $j-th$ column of $\mathbf{Z}^n$: $\mathbf{z}_j^n = \left(\frac{\partial \mathbf{g}^n}{\partial \mathbf{x}^n}\right)^{-1}\left(-\frac{\partial \mathbf{g}}{\partial \theta}\mathbf{v}_j - \frac{\partial \mathbf{g}^n}{\partial \mathbf{x}^{n-1}}\mathbf{z}_j^{n-1}\right)$

  If there are responses at n, compute $(\mathbf{GV})^n = \frac{\partial \mathbf{h}^n}{\partial \mathbf{x}^n}\mathbf{Z}^n + \frac{\partial \mathbf{h}^n}{\partial \theta}\mathbf{V}$

is solved from

$$\mathbf{Z}\left(\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\right) = -\mathbf{W}\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}}. \tag{2.21}$$

The linear system described in Eq. (2.21) can be re-written in a block-wise form for each time-step n as

$$\begin{aligned}
&\left(\mathbf{Z}^0 \;\middle|\; \mathbf{Z}^1 \;\middle|\; \ldots \;\middle|\; \mathbf{Z}^N\right) \times \begin{pmatrix} \frac{\partial \mathbf{g}^0}{\partial \mathbf{x}^0} & & & \\ \frac{\partial \mathbf{g}^1}{\partial \mathbf{x}^0} & \frac{\partial \mathbf{g}^1}{\partial \mathbf{x}^1} & & \\ & \ddots & \ddots & \\ & & \frac{\partial \mathbf{g}^N}{\partial \mathbf{x}^{N-1}} & \frac{\partial \mathbf{g}^N}{\partial \mathbf{x}^N} \end{pmatrix} = \\
&-\left(\mathbf{W}^0 \;\middle|\; \mathbf{W}^1 \;\middle|\; \ldots \;\middle|\; \mathbf{W}^N\right) \times \begin{pmatrix} \frac{\partial \mathbf{h}^0}{\partial \mathbf{x}^0} & & & \\ \frac{\partial \mathbf{h}^1}{\partial \mathbf{x}^0} & \frac{\partial \mathbf{h}^1}{\partial \mathbf{x}^1} & & \\ & \ddots & \ddots & \\ & & \frac{\partial \mathbf{h}^N}{\partial \mathbf{x}^{N-1}} & \frac{\partial \mathbf{h}^N}{\partial \mathbf{x}^N} \end{pmatrix}
\end{aligned} \tag{2.22}$$

One should note that Eq. (2.22) is solved backward in time. Now, by taking the transpose of Eq. (2.22), the linear system of equations that must be solved for each time-step

for the adjoint method reads

$$\begin{aligned}\left(\mathbf{Z}^n\right)^T = \left(\frac{\partial \mathbf{g}^n}{\partial \mathbf{x}^n}\right)^{-T} \times \\ \left(-\left(\mathbf{W}^n \frac{\partial \mathbf{h}^n}{\partial \mathbf{x}^n}\right)^T - \left(\mathbf{W}^{n+1} \frac{\partial \mathbf{h}^{n+1}}{\partial \mathbf{x}^n}\right)^T - \left(\frac{\partial \mathbf{g}^{n+1}}{\partial \mathbf{x}^n}\right)^T \left(\mathbf{Z}^{n+1}\right)^T\right).\end{aligned} \tag{2.23}$$

Equation 2.22 shows that the matrix $\mathbf{Z}$ can be fully determined by recursively marching backward in time. Once defined how the matrix $\mathbf{Z}$ can be computed, Eq. (2.22) can be employed to compute the sensitivity matrix. The algorithm is described in Algorithm 2.

Algorithm 2: Left multiplying the sensitivity matrix by an arbitrary matrix via the adjoint Method.

Input : $\frac{\partial \boldsymbol{g}}{\partial \mathbf{x}}, \frac{\partial \boldsymbol{g}}{\partial \theta}, \frac{\partial \boldsymbol{h}}{\partial \mathbf{x}}, \frac{\partial \boldsymbol{h}}{\partial \theta}, \mathbf{W}$
Output: $\mathbf{WG}$

1 **foreach** $n = N, \ldots, 2, 1, 0$ **do**
2 **foreach** $i = 1, 2, \ldots, m$ **do**
3 Solve for the $i-th$ column of $\left(\mathbf{Z}^n\right)^T$:
$$\left(\mathbf{z}_i^n\right)^T = \left(\frac{\partial \mathbf{g}^n}{\partial \mathbf{x}^n}\right)^{-T} \times \left(-\left(\mathbf{w}_i^n \frac{\partial \mathbf{h}^n}{\partial \mathbf{x}^n}\right)^T - \left(\mathbf{w}_i^{n+1} \frac{\partial \mathbf{h}^{n+1}}{\partial \mathbf{x}^n}\right)^T - \left(\frac{\partial \mathbf{g}^{n+1}}{\partial \mathbf{x}^n}\right)^T \left(\mathbf{z}_i^{n+1}\right)^T\right)$$
4 Update $(\mathbf{WG})^n = (\mathbf{WG})^n + \mathbf{Z}^n \frac{\partial \mathbf{g}^n}{\partial \theta} + \mathbf{W}^n \frac{\partial \mathbf{h}^n}{\partial \theta}$

2.1.4. A LAGRANGE MULTIPLIER DERIVATION, INTERPRETATION AND ASSOCIATION WITH THE FRAMEWORK

A widely used strategy to derive an adjoint model is the Lagrange multipliers method [7]. Given an objective function $O(\mathbf{x}, \theta) : \mathbb{R}_X^N \times \mathbb{R}_\theta^N \to \mathbb{R}$, an augmented objective function, or Lagrangian, can be defined as

$$L(\mathbf{x}, \theta, \lambda) = O(\mathbf{x}, \theta) + \lambda^T \boldsymbol{g}, \quad \mathbf{l} : \mathbb{R}_X^N \times \mathbb{R}_\theta^N \times \mathbb{R}_X^N \to \mathbb{R} \tag{2.24}$$

where $\lambda \in \mathbb{R}_X^N$. It is largely dicussed in the literature [6, 8, 9] how the adjoint model can be obtained from Eq. (2.24) for a scalar objective function. Analogously, we consider a Lagrangian for the model responses $\boldsymbol{h}$, so that

$$\mathbf{l}(\mathbf{x}, \theta, \Lambda) = \boldsymbol{h}(\mathbf{x}, \theta) + \Lambda^T \boldsymbol{g}, \quad \mathbf{l} : \mathbb{R}_X^N \times \mathbb{R}_\theta^N \times \mathbb{R}_X^N \to \mathbb{R} \tag{2.25}$$

where $\Lambda \in \mathbb{R}_X^N \times \mathbb{R}^{N_Y}$. By applying the necessary fist-order optimality conditions it follows that

$$\begin{cases} \dfrac{\partial \mathbf{l}}{\partial \Lambda} = \boldsymbol{g} = \mathbf{0} \\ \dfrac{\partial \mathbf{l}}{\partial \mathbf{x}} = \dfrac{\partial \boldsymbol{h}}{\partial \mathbf{x}} + \Lambda^T \dfrac{\partial \boldsymbol{g}}{\partial \mathbf{x}} = \mathbf{0} \\ \dfrac{\partial \mathbf{l}}{\partial \theta} = \dfrac{\partial \boldsymbol{h}}{\partial \mathbf{x}} \dfrac{\partial \mathbf{x}}{\partial \theta} + \dfrac{\partial \boldsymbol{h}}{\partial \theta} + \Lambda^T \left(\dfrac{\partial \boldsymbol{g}}{\partial \mathbf{x}} \dfrac{\partial \mathbf{x}}{\partial \theta} + \dfrac{\partial \boldsymbol{g}}{\partial \theta} \right) = \mathbf{0}. \end{cases}$$

It follows that

$$\Lambda^T = -\frac{\partial \boldsymbol{h}}{\partial \mathbf{x}} \left(\frac{\partial \boldsymbol{g}}{\partial \mathbf{x}} \right)^{-1} \tag{2.26}$$

Hence,

$$\frac{\partial \boldsymbol{h}}{\partial \theta} = -\frac{\partial \boldsymbol{h}}{\partial \mathbf{x}} \left(\frac{\partial \boldsymbol{g}}{\partial \mathbf{x}} \right)^{-1} \frac{\partial \boldsymbol{g}}{\partial \theta} + \frac{\partial \boldsymbol{h}}{\partial \theta}. \tag{2.27}$$

which, instead of simply computing the gradient of a given scalar objective function, the derivation would lead to if starting from Eq. (2.24), computes the full sensitivity matrix.

Now, following the same reasoning as exposed in 2.1, Eq. (2.14) can be reproduced if the Lagrange multiplier is used and Eq. (2.27) is pre- and post-multiplied, respectively, by $\mathbf{W}$ and $\mathbf{V}$.

Also, it can be noted Eq. (2.21) is, in fact, the Lagrange multiplier as found following the Lagrangian formulation after pre-multiplication by $\mathbf{V}$, so that Eq. (2.26) can be re-written as

$$\Lambda^T = -\mathbf{W} \frac{\partial \boldsymbol{h}}{\partial \mathbf{x}} \left(\frac{\partial \boldsymbol{g}}{\partial \mathbf{x}} \right)^{-1}. \tag{2.28}$$

The Lagrange multipliers can be interpreted as sensitivities of the objective function value with respect to deviations from the constraints. In case of adjoining the (nonlinear) porous media flow equations with Lagrange multipliers, this implies that the multipliers are the sensitivities of the objective function with respect to the residuals of the flow equations, i.e., to the residual error that remains after approximately solving the nonlinear equations with the aid of Newton-Rapson iteration.

2.2. APPLICATIONS OF THE FRAMEWORK

In this section we present how our framework can be employed to address the most common problems associated to subsurface model studies.

2.2.1. COMPUTATION OF DERIVATIVE INFORMATION FOR OPTIMIZATION ALGORITHMS

Many studies require the employment of optimization algorithms and different algorithms present different performances according to the particularities of the underlying problem being solved. In this section, we discuss how the mathematical framework here presented address the computation of different derivative information required by the different optimization algorithms. More detailed information on the different optimization algorithms can found in [10]. We do not cover optimization algorithms that requires the true Hessian computation, as for subsurface flow problems the computation of the

full Hessian is perceived to be too expensive. For the application of Newton's method to control optimization see [11].

In history matching algorithms, as well as in sensitivity analysis studies, the sensitivity matrix $\mathbf{G}$, defined as the matrix of derivatives of all responses with respect to all parameters, plays an important role.

In Gauss-Newton methods, the matrix product $\mathbf{G}^T\mathbf{G}$ is directly used. The Levenberg-Marquardt method's update equation can be written as

$$\left(\mathbf{G}^T\mathbf{G} + \lambda\mathbf{I}\right)\delta_\theta = \mathbf{G}^T\mathbf{r} \tag{2.29}$$

Conjugate gradient methods require products of $\mathbf{G}$ and $\mathbf{G}^T$ with arbitrary vectors. The CG method's update equation with line search can be written as

$$\theta_{k+1} = \theta_k + \alpha_k\left(-\nabla_\theta O_k + \beta * \mathbf{s}_{k-1}\right) \tag{2.30}$$

where α_k is computed by any line search algorithm from

$$\alpha_k = \arg\min_\alpha O\left(\theta_k + \alpha\mathbf{s}_k\right) \tag{2.31}$$

Quasi-Newton methods require the gradient of the objective function (and possibly constraints). It's been reported the limited memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) algorithm is one that provides very good permance for history mathcing problems. The BFGS algorithm update equation can be written as

$$\theta_{k+1} = \theta_k + \alpha_k\mathbf{B}_k^{-1}\left(-\nabla_\theta O_k\right) \tag{2.32}$$

where α_k is also given by Eq. (2.31).

2.2.2. COMPUTING DERIVATIVE OF DIFFERENT OBJECTIVE FUNCTIONS

In order to illustrate how the framework can be directly used to compute the derivative information of any given objective function in the form

$$O = O\left(\mathbf{h}\left(\theta\right)\right),$$

consider we wish to employ an optimization algorithm the directly used the gradient of the objective fucntion the gradient w.r.t. the parameters

$$\nabla_\theta O = \left(\frac{dO}{d\theta}\right)^T = \left(\frac{dO}{d\mathbf{h}}\frac{d\mathbf{h}}{d\theta}\right)^T = \mathbf{G}^T\nabla_\mathbf{h} O. \tag{2.33}$$

Next we consider two of the most important reservoir management studies, data assimilation and control optimization.

DATA ASSIMILATION STUDIES

In data assimilation studies (parameter calibration, history matching) one is interested to calibrate the uncertain parameters (e.g. grid-block permeabilities) by assimilating observed data from the real system. The problem can be usually seen as a least square problem minimization. If one looks at the history matching problem under a Bayesian

framework and under the usual Gaussian assumptions [12], the maximum a posteriori model can be obtained from the minimization of the following objective function

$$O(\mathbf{h}(\theta),\theta) = \frac{1}{2}\left(\theta-\theta_{prior}\right)^T \mathbf{C}_\theta^{-1}\left(\theta-\theta_{prior}\right) + \frac{1}{2}\left(\mathbf{h}(\mathbf{x},\theta)-\mathbf{d}_{obs}\right)^T \mathbf{C}_D^{-1}\left(\mathbf{h}(\mathbf{x},\theta)-\mathbf{d}_{obs}\right), \tag{2.34}$$

whose gradient is given by

$$\nabla_\theta O = \mathbf{C}_\theta^{-1}\left(\theta-\theta_{prior}\right) + \left(\frac{d\mathbf{h}}{d\theta}\right)^T \mathbf{C}_D^{-1}\left(\mathbf{h}(\mathbf{x},\theta)-\mathbf{d}_{obs}\right) = \mathbf{C}_\theta^{-1}\left(\theta-\theta_{prior}\right) + \mathbf{G}^T\mathbf{m}. \tag{2.35}$$

where

$$\mathbf{m} = \mathbf{C}_D^{-1}\left(\mathbf{h}(\mathbf{x},\theta)-\mathbf{d}_{obs}\right), \tag{2.36}$$

is an auxiliary vector, so the gradient of O can be written as $\nabla_\theta O = (\mathbf{m}^T\mathbf{G})^T$ and Algorithm 2, with $\mathbf{W} = \mathbf{m}^T$, calculates $\nabla_\theta O$ with a cost proportional to one extra forward simulation, making the Adjoint Method a much more efficient way to calculate the gradient than the Direct Method [4] [9]

CONTROL OPTIMIZATION STUDIES

Life-cycle optimization studies [3, 5, 13–15] are those where one is interested to maximize an economic indicator (e.g. Net present value) by manipulating the controllable parameters of the system (e.g. the well rates). We assume that the economic objective can be generically expressed as

$$O(\boldsymbol{h}(\theta)) = \sum_{n=1}^{N} O_n(\mathbf{h}_n(\theta)), \tag{2.37}$$

An NPV objective function can be generically expressed as

$$J = \mathbf{r}^T\boldsymbol{h}, \tag{2.38}$$

where $\mathbf{h}$ are the well rates (produced oil and water and injected water rates in a waterflooding setting [14, 16]) and $\mathbf{r}$ are the cost/revenues associated with the wells' operation.

This allows us to write the gradient of Eq. (3.64) as

$$\nabla_\theta J = \mathbf{r}^T\frac{d\boldsymbol{h}}{d\theta} = \mathbf{r}^T\mathbf{G}. \tag{2.39}$$

Eq. (3.64) allows the adjoint method to be employed in the computation of $\nabla_\theta J$ by making $\mathbf{W} = \mathbf{r}^T$. Just like in the data assimilation case, $\nabla_\theta J$ is efficiently computed with cost proportional to one backward simulation using the adjoint method.

REFERENCES

[1] J. D. Jansen, *Gradient-based optimization of flow through porous media,* .

[2] D. R. Brouwer, *Dynamic water flood optimization with smart wells using optimal control theory* (Delft University of Technology Delft, The Netherlands, 2004).

[3] D. Kourounis, L. J. Durlofsky, J. D. Jansen, and K. Aziz, *Adjoint formulation and constraint handling for gradient-based optimization of compositional reservoir flow,* Computational Geosciences **18**, 117 (2014).

[4] J. R. P. Rodrigues, *Calculating derivatives for automatic history matching,* Computational Geosciences **10**, 119 (2006).

[5] J. F. B. M. Kraaijevanger, P. J. P. Egberts, J. R. Valstar, and H. W. Buurman, *Optimal waterflood design using the adjoint method,* in *SPE Reservoir Simulation Symposium* (Society of Petroleum Engineers, 2007).

[6] F. Anterion, R. Eymard, and B. Karcher, *Use of parameter gradients for reservoir history matching,* in *SPE Symposium on Reservoir Simulation* (Society of Petroleum Engineers, 1989).

[7] J. L. Lagrange, *Mécanique analytique*, Vol. 1 (Mallet-Bachelier, 1853).

[8] G. Chavent, M. Dupuy, and P. Lemmonier, *History matching by use of optimal theory,* Society of Petroleum Engineers Journal **15**, 74 (1975).

[9] G. Gao, A. C. Reynolds, *et al.*, *An improved implementation of the lbfgs algorithm for automatic history matching,* in *SPE Annual Technical Conference and Exhibition* (Society of Petroleum Engineers, 2004).

[10] J. Nocedal and S. Wright, *Numerical optimization* (Springer Science & Business Media, 2006).

[11] E. Suwartadi, S. Krogstad, and B. Foss, *Second-order adjoint-based control for multiphase flow in subsurface oil reservoirs,* in *Decision and Control (CDC), 2010 49th IEEE Conference on* (IEEE, 2010) pp. 1866–1871.

[12] D. S. Oliver, A. C. Reynolds, and N. Liu, *Inverse theory for petroleum reservoir characterization and history matching* (Cambridge University Press, 2008).

[13] P. Sarma, K. Aziz, and L. J. Durlofsky, *Implementation of adjoint solution for optimal control of smart wells,* in *SPE reservoir simulation symposium* (Society of Petroleum Engineers, 2005).

[14] G. van Essen, M. Zandvliet, P. Van den Hof, O. Bosgra, J.-D. Jansen, *et al.*, *Robust waterflooding optimization of multiple geological scenarios,* SPE Journal **14**, 202 (2009).

[15] J. D. Jansen, *Adjoint-based optimization of multi-phase flow through porous media–a review,* Computers & Fluids **46**, 40 (2011).

[16] D. Brouwer, J. Jansen, *et al., Dynamic optimization of water flooding with smart wells using optimal control theory,* in *European Petroleum Conference* (Society of Petroleum Engineers, 2002).

3

COMPUTING DERIVATIVE INFORMATION OF SEQUENTIALLY COUPLED SUBSURFACE MODELS

A generic framework for the computation of derivative information required for gradient-based optimization using sequentially coupled subsurface simulation models is presented. The proposed approach allows for the computation of any derivative information with no modification of the mathematical framework. It only requires the forward model Jacobians and the objective function to be appropriately defined. The flexibility of the framework is demonstrated by its application in different reservoir management studies. The performance of the gradient computation strategy is demonstrated in a synthetic waterflooding model, where the forward model is constructed based on a sequentially coupled flow-transport system. The methodology is illustrated for a synthetic model, with different types of applications of data assimilation and life-cycle optimization. Results are compared with the classical fully-coupled (FIM) forward simulation. Based on the presented numerical examples, it is demonstrated how, without any modifications of the basic framework, the solution of gradient-based optimization models can be obtained for any given set of coupled equations. The sequential derivative computation methods deliver similar results compared to FIM methods, while being computationally more efficient.

The material presented in this chapter has been published in the Journal of Computational Geosciences (2018) [1] and presented at the SIAM Geosciences Conference (2018) [2].

The exploitation of subsurface resources frequently involves complex physics and geology. Thermal, geomechanical, and chemical processes are just a few phenomena that sometimes must be accounted for, while the domain is often governed by parameters that typically change several orders of magnitude over a wide range of spatial scales. Numerical simulation of such complex processes can be done with fully-implicit methods (FIMs) and sequentially coupled approaches. Even though FIMs provide the most stable simulation platform [3], many efforts have successfully lead to stable and efficient sequential simulations (e.g. [4]). Sequential simulation is often specially attractive for coupled processes of different physical natures, which often operate on different time scales or have different spatial support (e.g. local versus global effects). It is worth to be mentioned that multiscale methods [5–8] and model reduction techniques [9, 10] have been mainly developed for globally acting processes, and thus function optimally when they are used in sequential frameworks. Note that such a framework would also benefit some preconditioning methods that use sequential strategies (operator splitting) for the solution of linear system of equations. This holds for FIMs simulations using Constrained Pressure Restriction (CPR) preconditioning [11–13], where a pressure-like system is being extracted from the FIM Jacobian in order to enhance the convergence.

Ultimately, numerical simulation will support reservoir management studies which are often based on optimization techniques. It has been shown that gradient-based optimization techniques are the most efficient ones when applied, for instance, to life-cycle optimization [14, 15] and history matching [16–18] studies. Moreover, it is well known that the most efficient/accurate gradient computation technique is the adjoint method [15]. Even though a large body of the literature has been dedicated to this topic, most of it discusses the adjoint model for FIMs systems. In this case, the adjoint model is obtained by transposing the forward model system of equations [14, 17]. Also, even though the mathematical framework presented by [19] and [20] does not limit the derivation of the adjoint equations to any particular solution strategy, no explicit discussion on how it can be applied to sequentially coupled system of equations was presented. A multiscale adjoint method applied to life-cycle optimization is presented by [21], in which a sequential solution of flow and transport is employed, such that, consequentially, the adjoint model also follows a sequential solution strategy. However, in that work, the discussion is focused on the promising computational savings provided by multiscale simulation and not so much detail is given as to what extent the gradient computation itself can impose challenges.

The present work presents a general gradient computation formulation for sequentially-coupled models. An implicit differentiation strategy [19, 20] is extended to coupled systems of equations. The algorithms for the derivative computation of simulator responses neither depend on the objective function type, nature of the parameters, nor on any specific model coupling. Instead, it is shown how derivative information can be computed based on any coupling strategy. Using a chain-rule formalism, we firstly introduce a generic framework capable of computing the specific derivative information required by any given optimization algorithm. Next, it is shown how such computation is done for sequentially coupled flow and transport. Thereafter, numerical examples including both data assimilation and life-cycle optimization are presented.

3.1. MATHEMATICAL FRAMEWORK FOR THE COMPUTATION OF GRADIENT INFORMATION OF COUPLED SYSTEM OF EQUATIONS

We consider a system of discrete-in-time nonlinear model equations in implicit (residual) form:

$$\begin{cases} \mathbf{g}^n_{\mathbf{x}_1}\left(\mathbf{x}^{n-1}_1,\dots,\mathbf{x}^{n-1}_{N_c},\mathbf{x}^n_1,\dots,\mathbf{x}^n_{N_c},\theta\right)=\mathbf{0} \\ \vdots \\ \mathbf{g}^n_{\mathbf{x}_{N_c}}\left(\mathbf{x}^{n-1}_1,\dots,\mathbf{x}^{n-1}_{N_c},\mathbf{x}^n_1,\dots,\mathbf{x}^n_{N_c},\theta\right)=\mathbf{0}, \end{cases} \tag{3.1}$$

where $\mathbf{x}^n_c \in \mathbf{R}^{N^n_{\mathbf{x}_c}}$, $c \in \{1,\dots,N_c\}$, is the set of primary variables associated with the model equations $\mathbf{g}^n_{\mathbf{x}_c}$, and N_c the total number of coupled equations. The superscript n denotes the time-step index and $\theta \in \mathbf{R}^{N_\theta}$ is the vector of parameters with respect to which we aim to compute derivative information. There are $N^n_X = \sum_{c=1}^{N_c} N_{\mathbf{x}_c}{}^n$ primary variables at time-step n and N_θ parameters. Note that the initial conditions are assumed to be

$$\begin{cases} \mathbf{g}^0_{\mathbf{x}_1}\left(\mathbf{x}^0_1,\dots,\mathbf{x}^0_{N_c},\theta\right)=\mathbf{0} \\ \vdots \\ \mathbf{g}^0_{\mathbf{x}_{N_c}}\left(\mathbf{x}^0_1,\dots,\mathbf{x}^0_{N_c},\theta\right)=\mathbf{0}. \end{cases} \tag{3.2}$$

The functions defining the set of observable responses for a time-step are described as

$$\mathbf{y}^n = \mathbf{h}^n\left(\mathbf{x}^{n-1}_1,\dots,\mathbf{x}^{n-1}_{N_c},\mathbf{x}^n_1,\dots,\mathbf{x}^n_{N_c},\theta\right), \tag{3.3}$$

where $\mathbf{h}^n$ represents the output equations [22]. There are N^n_Y observations in time-step n.

Let

$$\mathbf{g}^n\left(\mathbf{x}^n,\mathbf{x}^{n-1},\theta\right) = \begin{pmatrix} \mathbf{g}^n_{\mathbf{x}_1} \\ \vdots \\ \mathbf{g}^n_{\mathbf{x}_{N_c}} \end{pmatrix}, \tag{3.4}$$

be the set of model equations, where $\mathbf{g}^n : \mathbf{R}^{N^{n-1}_X \times N^n_X \times N_\theta} \to \mathbf{R}^{N^n_X}$,

$$\mathbf{x}^n = \begin{pmatrix} \mathbf{x}^n_1 \\ \vdots \\ \mathbf{x}^n_{N_c} \end{pmatrix}, \tag{3.5}$$

be the state vector, where $\mathbf{x}^n \in \mathbf{R}^{N^n_X}$, and Eq. (3.3) be re-defined as

$$\mathbf{y}^n = \mathbf{h}^n\left(\mathbf{x}^{n-1},\mathbf{x}^n,\theta\right), \tag{3.6}$$

where $\mathbf{h}^n : \mathbf{R}^{N_X^{n-1} \times N_X^n \times N_\theta} \rightarrow \mathbf{R}^{N_Y^n}$.

A "super-vector" notation [19, 20] is used to capture the evolution in time. All instances of $\mathbf{g}^n$ as defined in Eq. (3.4) for all time-steps, can be collated in a function $\boldsymbol{g} : \mathbf{R}^{N_X \times N_\theta} \rightarrow \mathbf{R}^{N_X}$, where $N_X = \sum_{n=0}^{N} N_X^n$ is the total number of primary variables for all time-steps, such that the system of non-linear equations is represented as

$$\boldsymbol{g}(\boldsymbol{x}(\theta), \theta) = \mathbf{0}. \tag{3.7}$$

3

Note that we use bold italic font to indicate super vectors and just bold to indicate ordinary vectors.

Eq. (5.26) indicates the dependency of the forward model equations on both the primary variables and the model parameters, even though the model equations are only solved for $\boldsymbol{x}$ and the dependency on θ has to be taken into account for the implicit differentiation strategy that will be employed later on.

Similarly, all instances of $\mathbf{y}^n$ as defined in Eq. (3.6) for all time-steps can be collated in a function $\boldsymbol{h} : \mathbf{R}^{N_X \times N_\theta} \rightarrow \mathbf{R}^{N_Y}$, where $N_Y = \sum_{n=0}^{N} N_Y^n$ represents the total number of responses for all time-steps, so that

$$\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}(\theta), \theta). \tag{3.8}$$

Following the same implicit differentiation strategy as in [19] and [23], the sensitivity matrix $\mathbf{G}$ (i.e., sensitivity of the responses with respect to the parameters) can be computed by deriving Eq. (5.27) with respect to θ, i.e.,

$$\mathbf{G} = \frac{d\boldsymbol{h}}{d\theta} = \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}} \frac{d\boldsymbol{x}}{d\theta} + \frac{\partial \boldsymbol{h}}{\partial \theta}. \tag{3.9}$$

In order to find a relationship that defines $\frac{d\boldsymbol{x}}{d\theta}$, Eq. (5.26) is differentiated with respect to θ

$$\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}} \frac{d\boldsymbol{x}}{d\theta} + \frac{\partial \boldsymbol{g}}{\partial \theta} = \mathbf{0}, \tag{3.10}$$

so that

$$\frac{d\boldsymbol{x}}{d\theta} = -\left(\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\right)^{-1} \frac{\partial \boldsymbol{g}}{\partial \theta}. \tag{3.11}$$

Substituting Eq. (3.11) in Eq. (3.9) gives

$$\mathbf{G} = -\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}} \left(\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\right)^{-1} \frac{\partial \boldsymbol{g}}{\partial \theta} + \frac{\partial \boldsymbol{h}}{\partial \theta}. \tag{3.12}$$

In order to keep the framework general (in terms of which type of derivative information can be computed), the sensitivity matrix is pre- and post-multiplied by arbitrary matrices $\mathbf{V}$ (of size $N_\theta \times p$) and $\mathbf{W}$ (of size $m \times N_Y$)

$$\mathbf{WGV} = -\mathbf{W} \frac{\partial \boldsymbol{h}}{\partial \mathbf{x}} \left(\frac{\partial \boldsymbol{g}}{\partial \mathbf{x}}\right)^{-1} \frac{\partial \boldsymbol{g}}{\partial \theta} \mathbf{V} + \mathbf{W} \frac{\partial \boldsymbol{h}}{\partial \theta} \mathbf{V}. \tag{3.13}$$

The key aspect that defines the computational performance of the gradient computation is the order of the operations involving $\left(\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\right)^{-1}$. Based on that, both the direct [24] and adjoint [16] analytical methods to compute the necessary derivative information can be defined.

If $\mathbf{W}$ is factored out in Eq. (6.18), it can be rewritten as

$$\mathbf{GV} = \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}}\mathbf{Z} + \frac{\partial \boldsymbol{h}}{\partial \theta}\mathbf{V}, \tag{3.14}$$

where

$$\mathbf{Z} = -\left(\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\right)^{-1}\frac{\partial \boldsymbol{g}}{\partial \theta}\mathbf{V}, \tag{3.15}$$

is solved from

$$\left(\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\right)\mathbf{Z} = -\frac{\partial \boldsymbol{g}}{\partial \theta}\mathbf{V}. \tag{3.16}$$

The linear system described in Eq. (6.25) can be re-written in a block-wise form for each time-step n:

$$\begin{pmatrix} \frac{\partial \mathbf{g}^0}{\partial \mathbf{x}^0} & & & \\ \frac{\partial \mathbf{g}^1}{\partial \mathbf{x}^0} & \frac{\partial \mathbf{g}^1}{\partial \mathbf{x}^1} & & \\ & \ddots & \ddots & \\ & & \frac{\partial \mathbf{g}^N}{\partial \mathbf{x}^{N-1}} & \frac{\partial \mathbf{g}^N}{\partial \mathbf{x}^N} \end{pmatrix} \begin{pmatrix} \mathbf{Z}^0 \\ \mathbf{Z}^1 \\ \vdots \\ \mathbf{Z}^N \end{pmatrix} = - \begin{pmatrix} \frac{\partial \mathbf{g}^0}{\partial \theta}\mathbf{V} \\ \frac{\partial \mathbf{g}^1}{\partial \theta}\mathbf{V} \\ \vdots \\ \frac{\partial \mathbf{g}^N}{\partial \theta}\mathbf{V} \end{pmatrix}, \tag{3.17}$$

where, from Eq. (3.4) and Eq. (3.5) one can write

$$\frac{\partial \mathbf{g}^n}{\partial \mathbf{x}^n} = \begin{pmatrix} \frac{\partial \mathbf{g}^n_{\mathbf{x}_1}}{\partial \mathbf{x}^n_1} & \cdots & \frac{\partial \mathbf{g}^n_{\mathbf{x}_1}}{\partial \mathbf{x}^n_{N_c}} \\ \vdots & \ddots & \vdots \\ \frac{\partial \mathbf{g}^n_{\mathbf{x}_{N_c}}}{\partial \mathbf{x}^n_1} & \cdots & \frac{\partial \mathbf{g}^n_{\mathbf{x}_{N_c}}}{\partial \mathbf{x}^n_{N_c}} \end{pmatrix}, \tag{3.18}$$

and

$$\frac{\partial \mathbf{g}^n}{\partial \mathbf{x}^{n-1}} = \begin{pmatrix} \frac{\partial \mathbf{g}^n_{\mathbf{x}_1}}{\partial \mathbf{x}^{n-1}_1} & \cdots & \frac{\partial \mathbf{g}^n_{\mathbf{x}_1}}{\partial \mathbf{x}^{n-1}_{N_c}} \\ \vdots & \ddots & \vdots \\ \frac{\partial \mathbf{g}^n_{\mathbf{x}_{N_c}}}{\partial \mathbf{x}^{n-1}_1} & \cdots & \frac{\partial \mathbf{g}^n_{\mathbf{x}_{N_c}}}{\partial \mathbf{x}^{n-1}_{N_c}} \end{pmatrix}. \tag{3.19}$$

Here N is the total number of time-steps and the partitioning lines indicate which matrix and vector terms belong to each time-step. Also, from Eq. (3.6) and Eq. (3.5)

$$\frac{\partial \mathbf{h}^n}{\partial \mathbf{x}^n} = \begin{pmatrix} \frac{\partial \mathbf{h}^n}{\partial \mathbf{x}_1^n} & \cdots & \frac{\partial \mathbf{h}^n}{\partial \mathbf{x}_{N_c}^n} \end{pmatrix}, \tag{3.20}$$

and

$$\frac{\partial \mathbf{h}^n}{\partial \mathbf{x}^{n-1}} = \begin{pmatrix} \frac{\partial \mathbf{h}^n}{\partial \mathbf{x}_1^{n-1}} & \cdots & \frac{\partial \mathbf{h}^n}{\partial \mathbf{x}_{N_c}^{n-1}} \end{pmatrix}. \tag{3.21}$$

This solution strategy is known in the literature as the *Forward Method* [19], *Gradient Simulator* [24], or *Direct Method* [18]. Note that auxiliary matrix $\mathbf{Z}$ has dimensions of $N_X \times p$ and, therefore, it requires $N \times p$ linear systems to be solved. Hence, the cost of computing $\mathbf{GV}$ is proportional to the number of columns in $\mathbf{V}$, i.e., p.

Now, if $\mathbf{V}$ is factored out in Eq. (6.18), the equation can be rewritten as

$$\mathbf{WG} = \mathbf{Z}\frac{\partial \boldsymbol{g}}{\partial \theta} + \mathbf{W}\frac{\partial \boldsymbol{h}}{\partial \theta}, \tag{3.22}$$

where

$$\mathbf{Z} = -\mathbf{W}\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}}\left(\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\right)^{-1} \tag{3.23}$$

is solved from

$$\mathbf{Z}\left(\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\right) = -\mathbf{W}\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}}. \tag{3.24}$$

The linear system described in Eq. (6.37) can be re-written in a block-wise form for each time-step n as

$$\begin{aligned} &\left(\mathbf{Z}^0 \;\middle|\; \mathbf{Z}^1 \;\middle|\; \cdots \;\middle|\; \mathbf{Z}^N\right) \times \left(\begin{array}{c|c|c|c} \frac{\partial \mathbf{g}^0}{\partial \mathbf{x}^0} & & & \\ \hline \frac{\partial \mathbf{g}^1}{\partial \mathbf{x}^0} & \frac{\partial \mathbf{g}^1}{\partial \mathbf{x}^1} & & \\ \hline & \ddots & \ddots & \\ \hline & & \frac{\partial \mathbf{g}^N}{\partial \mathbf{x}^{N-1}} & \frac{\partial \mathbf{g}^N}{\partial \mathbf{x}^N} \end{array}\right) = \\ &-\left(\mathbf{W}^0 \;\middle|\; \mathbf{W}^1 \;\middle|\; \cdots \;\middle|\; \mathbf{W}^N\right) \times \left(\begin{array}{c|c|c|c} \frac{\partial \mathbf{h}^0}{\partial \mathbf{x}^0} & & & \\ \hline \frac{\partial \mathbf{h}^1}{\partial \mathbf{x}^0} & \frac{\partial \mathbf{h}^1}{\partial \mathbf{x}^1} & & \\ \hline & \ddots & \ddots & \\ \hline & & \frac{\partial \mathbf{h}^N}{\partial \mathbf{x}^{N-1}} & \frac{\partial \mathbf{h}^N}{\partial \mathbf{x}^N} \end{array}\right) \end{aligned} \tag{3.25}$$

One should note that Eq. (6.38) is solved backward in time. Now, by taking the transpose of Eq. (6.38), the linear system of equations that must be solved for each time-step for the adjoint method reads

$$\begin{aligned}\left(\mathbf{Z}^n\right)^T &= \left(\frac{\partial \mathbf{g}^n}{\partial \mathbf{x}^n}\right)^{-T} \times \\ \left(-\left(\mathbf{W}^n \frac{\partial \mathbf{h}^n}{\partial \mathbf{x}^n}\right)^T - \left(\mathbf{W}^{n+1} \frac{\partial \mathbf{h}^{n+1}}{\partial \mathbf{x}^n}\right)^T - \left(\frac{\partial \mathbf{g}^{n+1}}{\partial \mathbf{x}^n}\right)^T \left(\mathbf{Z}^{n+1}\right)^T\right).\end{aligned} \tag{3.26}$$

This solution strategy is known in the literature as the *Adjoint* (or *Backward*) *Method* (Chavent, 1975). Note that now $\mathbf{Z}$ has dimensions of $N_X \times m$, hence it requires $N \times m$ linear systems to be solved. As such, the cost of computing $\mathbf{WG}$ is proportional to the number of rows in $\mathbf{W}$, i.e., m.

Although the derivation as presented so far is general, in order to properly formulate the actual method to analytically compute the derivative information, the structure of the partial derivative matrices involved in the computations must be taken into account. This is only possible if the specific coupling strategy and the proper dependencies of the model equations and primary variables are taken into account. Therefore, in the rest for the paper we focus our studies on sequentially coupled multiphase flow simulations.

3.1.1. REMARKS ABOUT THE FRAMEWORK

The appropriate selection of the arbritrary matrices $\mathbf{W}$ and $\mathbf{V}$ allows one single framework to compute any derivative information and avoids the expensive computation of $\mathbf{G}$. For instace, in case of quasi-Newton methods [25], the gradient of the objective function $O = O\left(\mathbf{y}(\theta)\right)$ is directly required. Via the chain-rule, one can write

$$\nabla_\theta O = \left(\frac{dO}{d\theta}\right)^T = \left(\frac{dO}{d\mathbf{h}}\frac{d\mathbf{h}}{d\theta}\right)^T = \mathbf{G}^T \nabla_{\mathbf{h}} O. \tag{3.27}$$

The operation $(\mathbf{WG})^T = \mathbf{G}^T\mathbf{W}^T$ gives the product of $\mathbf{G}^T$ with the (column) vector $\mathbf{W}^T = \nabla_{\mathbf{h}} O$. Hence, the adjoint method can be efficiently employed to compute the objective function gradient with respect to the parameters, as described in Eq. (3.27).

Now, in case of conjugate gradient methods [25], products of $\mathbf{G}$ and $\mathbf{G}^T$ with arbitrary vectors are required. The product $\mathbf{GV}$, with $n = 1$ can be efficiently computed by the direct method while, the product $\mathbf{G}^T\mathbf{W}^T$, with $m = 1$ can be efficiently computed using the adjoint method.

Another factor that maintains the flexibility of the framework is the formal partitioning of $\boldsymbol{g}$ and $\boldsymbol{x}$ according to the coupling of the equations. The computation of the auxiliary matrix $\mathbf{Z}$ in Eq. (6.24) and Eq. (3.26) will follow the partition of the $\boldsymbol{g}$. Once $\mathbf{Z}$ is fully determined, the sensitivity matrix products Eq. (6.23) and Eq. (6.35) remain unchanged. Hence, the framework requires the Jacobians of $\boldsymbol{g}$ w.r.t. $\boldsymbol{x}$ and θ to be determined from the coupled forward model equations.

We highlight that linear system solutions involving $\dfrac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}$ are required on both direct and adjoint methods (see Eq. (6.25) and Eq. (6.37)) in order to determine the auxiliary matrix $\mathbf{Z}$. More specifically, the unique derivative information computation requires

$\frac{\partial \mathbf{g}^n}{\partial \mathbf{x}^n}$ to be full-rank. This is true in most of the cases given that this is the same partial derivative matrix required by the forward simulation. For instance, this matrix represents the Jacobian used by Newton-Raphson non-linear solvers, typically employed in the forward simulation.

The importance of the implementation separation between the forward model and the adjoint model was previously highlighted by [14], who also presented a discussion about memory requirements related to the storage of the partial derivatives (or states required to re-evaluate them during the backward runs). In that work, the computational aspects were discussed in the context of an optimal control problem using FIM. Note that, as shown in the previous section, the framework presented in our paper is readily applicable to different coupling strategies or derivative computation problems.

Also, both the direct and adjoint methods are treated in the same framework. The direct method is usually directly associated with the forward simulation. All the derivative information related computation is usually presented as part of the forward time-stepping process. Here, it is shown that it can also be achieved in complete separation from the forward simulation. The requirement is the same as for the adjoint method: the required Jacobians must be stored / re-evaluated for the derivative information required at a later stage. However, in one hand the separation from the forward simulation reduces the code intrusion, on the other hand this strategy requires the storage of the partial derivative matrices also for the direct method. Even though it has implications from a memory usage perspective, the computational efficiency of the direct method remains the same considering an efficient strategy to dump/load the partial derivative matrices from the hard-disk or their reconstruction from the primary variables states (similar concerns are associated with the adjoint method).

3.2. APPLICATIONS OF THE FRAMEWORK: LIFE-CYCLE OPTIMIZATION AND ASSISTED HISTORY MATCHING OF SEQUENTIALLY COUPLED FLOW AND TRANSPORT FORWARD MODEL

3.2.1. ALGEBRAIC DESCRIPTION OF FORWARD MODEL EQUATIONS

The computation of derivative information for sequentially coupled systems is illustrated in the context of flow and transport in heterogeneous porous media. More specifically, two-phase, immiscible, incompressible flow is considered, with no gravity and capillary effects. The total mass balance (flow) equation is given by

$$-\nabla \cdot \left(\lambda \mathbf{K} \cdot \nabla p\right) = \nabla \cdot (u) = q, \tag{3.28}$$

where u is the total velocity, $\mathbf{K}$ the absolute permeability tensor and p is the pressure [3]. The total mobility is given by $\lambda = \lambda_o + \lambda_w$, with the subscripts o and w standing for, respectively, oil and water, and the total source term is given by $q = q_o + q_w$.

The transport equation for a given phase α can be written as

$$\phi \frac{\partial S_\alpha}{\partial t} + \nabla \cdot \left(f_\alpha \mathbf{u}\right) = q_\alpha, \tag{3.29}$$

where S_α and f_α are, respectively, the saturation and fractional flow of phase α. The system is closed via the saturation constraint

$$\sum_{\alpha=o,w} S_\alpha = 1. \tag{3.30}$$

The discrete form of Eq. (3.28) reads

$$\mathbf{g}_p^n = \mathbf{A}^{n-1}\mathbf{p}^n - \mathbf{q}^{n-1} = \mathbf{0}, \tag{3.31}$$

where $\mathbf{p}^n \in \mathbf{R}^{N_b}$ and $\mathbf{q}^{n-1} \in \mathbf{R}^{N_b}$ are vectors of pressure and source terms, respectively, N_b is the number of grid blocks, and $\mathbf{A}^{n-1} \in \mathbf{R}^{N_b \times N_b}$ is the system matrix. Interfacial rock properties are computed by means of harmonic averages for the absolute permeabilities, whereas an upwind scheme is employed for interfacial fluid properties (i.e. mobilities). The dependency of the fluid mobilities on the saturation is treated lagged in time because of the sequential solution strategy.

The discrete form of Eq. (5.5) reads

$$\mathbf{g}_s^n = \mathbf{V}\left(\mathbf{s}^n - \mathbf{s}^{n-1}\right) + \mathbf{F}^t\mathbf{u}^n - \mathbf{q}_\alpha^t = \mathbf{0}, \tag{3.32}$$

where $\mathbf{s} \in \mathbf{R}^{N_b}$, $\mathbf{F}^t \in \mathbf{R}^{N_b \times N_I}$, and $\mathbf{u}^n \in \mathbf{R}^{N_I}$ are, respectively, the saturation vector, the upwind fractional flow matrix and the vector containing the normal to grid interfaces velocity components, with N_I being the number of grid interfaces,

$$\mathbf{V} = \mathbf{I}\frac{\mathbf{V}_\phi}{\Delta t}, \tag{3.33}$$

where $\mathbf{V} \in \mathbf{R}^{N_b \times N_b}$, and $\mathbf{u}^n$ is computed from

$$\mathbf{g}_u^n = \mathbf{u}^n - \mathbf{\Lambda}^{n-1}\mathbf{p}^n = \mathbf{0}, \tag{3.34}$$

where $\mathbf{\Lambda}^{n-1} \in \mathbf{R}^{N_I \times N_b}$ is the transmissibilty matrix. Futhermore, we highlight that, in our implementation, α is considered to be water, and hence water saturation is a primary variable. Therefore, all references to saturation found from here on are w.r.t. water saturation. Additionally, Δt is the time-step size, $\mathbf{V}_\phi \in \mathbf{R}^{N_b}$ is the vector containing the grid block pore-volumes, and $\mathbf{I}$ is the identity matrix.

The de-coupling of Eq. (5.8) and Eq. (5.9) allows the system to be solved sequentially, with no dependency of Eq. (5.8) on $\mathbf{s}^n$. If $t = n-1$, the fractional flow and source terms are evaluated at the previous time-step. This is the so-called implicit-pressure explicit-saturation (IMPES) discretization in time [3]. However, to avoid time-step size limitations [26], the so-called sequential implicit strategy (IMPSAT) can be defined by making $t = n$. Although Eq. (5.9) now has a non-linear dependency on $\mathbf{s}^n$, this scheme allows for larger time-steps.

3.2.2. GRADIENT COMPUTATION

From the discrete forward simulation equations Eq. (5.8), Eq. (5.9), and Eq. (3.34), Eq. (3.1) can be specialized as

$$\begin{cases} \mathbf{g}_p^n\left(\mathbf{p}^n, \mathbf{s}^{n-1}, \theta\right) = \mathbf{0} \\ \mathbf{g}_u^n\left(\mathbf{p}^n, \mathbf{u}^n, \mathbf{s}^{n-1}, \theta\right) = \mathbf{0} \\ \mathbf{g}_s^n\left(\mathbf{p}^n, \mathbf{u}^n, \mathbf{s}^{n-1}, \mathbf{s}^n, \theta\right) = \mathbf{0}, \end{cases} \tag{3.35}$$

where $\mathbf{g}_p^n$, $\mathbf{g}_u^n$ and $\mathbf{g}_s^n$ are, respectively, the vector-valued equations describing flow (pressure) and transport (saturation) at time-step n. The equations that determine the initial conditions are assumed to be

$$\begin{cases} \mathbf{g}_p^0\left(\mathbf{p}^0,\theta\right)=\mathbf{0} \\ \mathbf{g}_u^0\left(\mathbf{p}^0,\mathbf{u}^0,\theta\right)=\mathbf{0} \\ \mathbf{g}_s^0\left(\mathbf{p}^0,\mathbf{u}^0,\mathbf{s}^0,\theta\right)=\mathbf{0}. \end{cases} \tag{3.36}$$

From Eq. (3.35), let

$$\mathbf{g}^n=\begin{pmatrix}\mathbf{g}_p^n\\ \mathbf{g}_u^n\\ \mathbf{g}_s^n\end{pmatrix}. \tag{3.37}$$

Also, based on the corresponding primary variables associated to Eq. (3.35), Eq. (3.5) can be redefined as

$$\mathbf{x}^n=\begin{pmatrix}\mathbf{p}^n\\ \mathbf{u}^n\\ \mathbf{s}^n\end{pmatrix}. \tag{3.38}$$

The functions defining the set of observable outputs at time-step n will be assumed to be functions of both $\mathbf{p}^n$ and $\mathbf{s}^t$, i.e.,

$$\mathbf{y}^n=\mathbf{h}^n\left(\mathbf{p}^n,\mathbf{s}^t,\theta\right), \tag{3.39}$$

which, in the case of IMPES reads

$$\mathbf{y}^n=\mathbf{h}^n\left(\mathbf{p}^n,\mathbf{s}^{n-1},\theta\right), \tag{3.40}$$

and for IMPSAT reads

$$\mathbf{y}^n=\mathbf{h}^n\left(\mathbf{p}^n,\mathbf{s}^n,\theta\right). \tag{3.41}$$

From Eq. (3.38), Eq. (3.39) can be re-written as

$$\mathbf{y}^n=\mathbf{h}^n\left(\mathbf{x}^{n-1},\mathbf{x}^n,\theta\right). \tag{3.42}$$

Based on Eq. (3.35), Eq. (3.38), Eq. (6.27) and Eq. (6.28) can be redefined, now taking into account the appropriate dependencies of equations and variables for the flow-transport coupling, as

$$\frac{\partial\mathbf{g}^n}{\partial\mathbf{x}^n}=\begin{pmatrix}\dfrac{\partial\mathbf{g}_p^n}{\partial\mathbf{p}^n} & \mathbf{0} & \mathbf{0}\\ \dfrac{\partial\mathbf{g}_u^n}{\partial\mathbf{p}^n} & \dfrac{\partial\mathbf{g}_u^n}{\partial\mathbf{u}^n} & \mathbf{0}\\ \dfrac{\partial\mathbf{g}_s^n}{\partial\mathbf{p}^n} & \dfrac{\partial\mathbf{g}_s^n}{\partial\mathbf{u}^n} & \dfrac{\partial\mathbf{g}_s^n}{\partial\mathbf{s}^n}\end{pmatrix}=\begin{pmatrix}\mathbf{A}^{n-1} & \mathbf{0} & \mathbf{0}\\ -\boldsymbol{\Lambda}^{n-1} & \mathbf{I} & \mathbf{0}\\ -\dfrac{\partial\mathbf{q}_\alpha^n}{\partial\mathbf{p}^n} & \mathbf{F}^t & \dfrac{\partial\mathbf{g}_s^n}{\partial\mathbf{s}^n}\end{pmatrix}, \tag{3.43}$$

and

$$\frac{\partial \mathbf{g}^n}{\partial \mathbf{x}^{n-1}} = \begin{pmatrix} \mathbf{0} & \mathbf{0} & \dfrac{\partial \mathbf{g}_p^n}{\partial \mathbf{s}^{n-1}} \\ \mathbf{0} & \mathbf{0} & \dfrac{\partial \mathbf{g}_u^n}{\partial \mathbf{s}^{n-1}} \\ \mathbf{0} & \mathbf{0} & \dfrac{\partial \mathbf{g}_s^n}{\partial \mathbf{s}^{n-1}} \end{pmatrix}. \tag{3.44}$$

Furthermore, based on Eq. (3.38) and Eq. (3.42), it follows that

$$\frac{\partial \mathbf{h}^n}{\partial \mathbf{x}^n} = \begin{pmatrix} \dfrac{\partial \mathbf{h}^n}{\partial \mathbf{p}^n} & \mathbf{0} & \dfrac{\partial \mathbf{h}^n}{\partial \mathbf{s}^n} \end{pmatrix}, \tag{3.45}$$

and

$$\frac{\partial \mathbf{h}^n}{\partial \mathbf{x}^{n-1}} = \begin{pmatrix} \mathbf{0} & \mathbf{0} & \dfrac{\partial \mathbf{h}^n}{\partial \mathbf{s}^{n-1}} \end{pmatrix}. \tag{3.46}$$

Note that $\dfrac{\partial \mathbf{h}^n}{\partial \mathbf{s}^n} = \mathbf{0}$ in Eq. (3.45) if a sequential explicit method is used. On the other hand, $\dfrac{\partial \mathbf{h}^n}{\partial \mathbf{s}^{n-1}} = \mathbf{0}$ in Eq. (3.46) if a sequential implicit method is used.

Also, one should note that $\dfrac{\partial \mathbf{g}_s^n}{\partial \mathbf{s}^n}$ in Eq. (3.43) becomes diagonal if IMPES is used and non-diagonal if IMPSAT is used. On the other hand, $\dfrac{\partial \mathbf{g}_s^n}{\partial \mathbf{s}^{n-1}}$ becomes diagonal in Eq. (3.44) if IMPSAT is used and non-diagonal if IMPES is used.

THE DIRECT METHOD

If Eq. (3.43), Eq. (3.44), Eq. (3.45) and Eq. (3.46) are used in Eq. (6.26), the algorithm to compute the required gradient information using the direct method can be defined for the flow-transport coupling. The linear systems that must be solved for the flow equation in the direct method, for every time-step n, are given by

$$\mathbf{Z}_p^n = \left(\frac{\partial \mathbf{g}_p^n}{\partial \mathbf{p}^n}\right)^{-1} \left(\frac{\partial \mathbf{g}_p^n}{\partial \theta}\mathbf{V} - \frac{\partial \mathbf{g}_p^n}{\partial \mathbf{s}^{n-1}}\mathbf{Z}_s^{n-1}\right), \tag{3.47}$$

for the pressure equation,

$$\mathbf{Z}_u^n = \left(\frac{\partial \mathbf{g}_u^n}{\partial \mathbf{u}^n}\right)^{-1} \left(\frac{\partial \mathbf{g}_u^n}{\partial \theta}\mathbf{V} - \frac{\partial \mathbf{g}_u^n}{\partial \mathbf{p}^n}\mathbf{Z}_p^n - \frac{\partial \mathbf{g}_u^n}{\partial \mathbf{s}^{n-1}}\mathbf{Z}_s^{n-1}\right) \tag{3.48}$$

for the velocity equation, and

$$\mathbf{Z}_s^n = \left(\frac{\partial \mathbf{g}_s^n}{\partial \mathbf{s}^n}\right)^{-1} \left(\frac{\partial \mathbf{g}_s^n}{\partial \theta}\mathbf{V} - \frac{\partial \mathbf{g}_s^n}{\partial \mathbf{p}^n}\mathbf{Z}_p^n - \frac{\partial \mathbf{g}_s^n}{\partial \mathbf{u}^n}\mathbf{Z}_u^n - \frac{\partial \mathbf{g}_s^n}{\partial \mathbf{s}^{n-1}}\mathbf{Z}_s^{n-1}\right) \tag{3.49}$$

for the transport equation.

From Eq. (3.47) and Eq. (3.49), Eq. (6.23) can be redefined based on the partitioning

$$\mathbf{Z}^n = \begin{pmatrix} \mathbf{Z}_p^n \\ \mathbf{Z}_u^n \\ \mathbf{Z}_s^n \end{pmatrix}, \tag{3.50}$$

so that the computation of the product **GV** at time-step n is given by

$$(\mathbf{GV})^n = \mathbf{G}^n\mathbf{V} = \frac{\partial \mathbf{h}^n}{\partial \theta}\mathbf{V} - \frac{\partial \mathbf{h}^n}{\partial \mathbf{p}^n}\mathbf{Z}_p^n - \frac{\partial \mathbf{h}^n}{\partial \mathbf{s}^n}\mathbf{Z}_s^n - \frac{\partial \mathbf{h}^n}{\partial \mathbf{s}^{n-1}}\mathbf{Z}_s^{n-1}. \tag{3.51}$$

Now, the direct method algorithm can be defined and is depicted in Algorithm 3.

Algorithm 3: Right multiplying the sensitivity matrix by an arbitrary matrix via the direct method.

Input : Partial derivative matrices of $\mathbf{g}_p$, $\mathbf{g}_u$ and $\mathbf{g}_s$ w.r.t. $\mathbf{x}$ and θ, $\mathbf{V}$
Output: $\mathbf{GV}$

1 **foreach** $n = 0, 1, 2, \ldots, N$ **do**
2 **foreach** $j = 1, 2, \ldots, p$ **do**
3 Solve for the $j-th$ column of $\mathbf{Z}_p^n$ using Eq. (3.47).
4 Solve for the $j-th$ column of $\mathbf{Z}_u^n$ using Eq. (3.48).
5 Solve for the $j-th$ column of $\mathbf{Z}_s^n$ using Eq. (3.49).
6 If there are responses at n, compute $(\mathbf{GV})^n$ using Eq. (3.51)

THE ADJOINT METHOD

By transposing Eq. (3.43), Eq. (3.44), Eq. (3.45) and Eq. (3.46) and replacing them in Eq. (3.26), the linear system that must be solved for the flow equation, for every time-step n, now reads

$$\begin{aligned} \left(\mathbf{Z}_p^n\right)^T = \left(\frac{\partial \mathbf{g}_p^n}{\partial \mathbf{p}^n}\right)^{-T} \times \\ \left(-\left(\frac{\partial \mathbf{g}_u^n}{\partial \mathbf{p}^n}\right)^T \left(\mathbf{Z}_u^n\right)^T - \left(\frac{\partial \mathbf{g}_s^n}{\partial \mathbf{p}^n}\right)^T \left(\mathbf{Z}_s^n\right)^T - \left(\mathbf{W}^n \frac{\partial \mathbf{h}^n}{\partial \mathbf{p}^n}\right)^T\right) \end{aligned} \tag{3.52}$$

for the velocity equation

$$\left(\mathbf{Z}_u^n\right)^T = -\left(\frac{\partial \mathbf{g}_u^n}{\partial \mathbf{u}^n}\right)^{-T} \left(\frac{\partial \mathbf{g}_s^n}{\partial \mathbf{u}^n}\right)^T \left(\mathbf{Z}_s^n\right)^T \tag{3.53}$$

and for the transport equation

$$\begin{aligned}\left(\mathbf{Z}_s^n\right)^T = \left(\frac{\partial \mathbf{g}_s^n}{\partial \mathbf{s}^n}\right)^{-T} \Bigg(&- \left(\mathbf{W}^n \frac{\partial \mathbf{h}^n}{\partial \mathbf{s}^n}\right)^T - \left(\mathbf{W}^{n+1} \frac{\partial \mathbf{h}^{n+1}}{\partial \mathbf{s}^n}\right)^T \\ &- \left(\frac{\partial \mathbf{g}_p^{n+1}}{\partial \mathbf{s}^n}\right)^T \left(\mathbf{Z}_p^{n+1}\right)^T - \left(\frac{\partial \mathbf{g}_u^{n+1}}{\partial \mathbf{s}^n}\right)^T \left(\mathbf{Z}_u^{n+1}\right)^T \\ &- \left(\frac{\partial \mathbf{g}_s^{n+1}}{\partial \mathbf{s}^n}\right)^T \left(\mathbf{Z}_s^{n+1}\right)^T \Bigg).\end{aligned} \tag{3.54}$$

By blocking Eq. (6.35) in time we have

$$\mathbf{WG} = \sum_{n=0}^{N} \left(\mathbf{Z}_p^n \frac{\partial \mathbf{g}_p^n}{\partial \theta} + \mathbf{Z}_u^n \frac{\partial \mathbf{g}_u^n}{\partial \theta} + \mathbf{Z}_s^n \frac{\partial \mathbf{g}_s^n}{\partial \theta} + \mathbf{W}^n \frac{\partial \mathbf{h}^n}{\partial \theta} \right). \tag{3.55}$$

The adjoint algorithm for the sequential coupling is described in Algorithm 4. The gradient computation does not only involve a backward simulation, but the solution of pressure and transport related terms in the backward run is reversed when compared to the order in which the equations are solved in the forward simulation.

We highlight the importance of the backward simulation to exactly follow the forward simulation steps. In our implementation, because in the backward simulation we precisely follow the time-stepping strategy taken in the forward simulation, given that any stability issue has been addressed in the forward simulation (e.g. due to CFL conditions), the backward simulation also has no stability issues. In other words, the backward formulation, for both IMPES and IMPSAT, is as stable as the forward simulation, given that the same time-stepping strategy is employed in the backward simulation. Furthermore, because the tangent linear model as obtained during the Newton iterations is unconditionally stable, the backward model, which is, by definition, the transpose of the tangent linear model, also must be stable.

Algorithm 4: Left multiplying the sensitivity matrix by an arbitrary matrix via the adjoint Method.

Input : Partial derivative matrices of $\mathbf{g}_p$, $\mathbf{g}_u$ and $\mathbf{g}_s$ w.r.t. $\mathbf{x}$ and θ, $\mathbf{W}$
Output: $\mathbf{WG}$

1 **foreach** $n = N, \ldots, 2, 1, 0$ **do**
2 **foreach** $i = 1, 2, \ldots, m$ **do**
3 Solve for the $i-th$ column of $\left(\mathbf{Z}_s^n\right)^T$ using Eq. (3.54).
4 Solve for the $i-th$ column of $\left(\mathbf{Z}_u^n\right)^T$ using Eq. (3.53).
5 Solve for the $i-th$ column of $\left(\mathbf{Z}_p^n\right)^T$ using Eq. (3.52).
6 Update ($\mathbf{WG}$) using Eq. (3.55).

3.2.3. GRADIENT COMPUTATION AND OPTIMIZATION FOR DATA ASSIMILATION

In data assimilation studies one is interested to incorporate reponses (or observations) from the real system into the numerical model by updating the (uncertain) model parameters so that the model's response reproduces the system observations. From a mathematical point of view, this exercise can be appoached as an optimization problem

$$\begin{aligned} \underset{\theta}{\text{minimize}} \quad & O(\boldsymbol{h}(\boldsymbol{x},\theta)) \\ \text{subject to} \quad & \boldsymbol{g}(\boldsymbol{x},\theta) = \mathbf{0}, \\ & \theta \in [\theta_{min},\theta_{max}], \end{aligned} \tag{3.56}$$

where O is usually an objective function that represents the misfit between observed data and model responses. In data assimilation problems, θ represents the uncertain parameters, which are usually bounded between the upper and lower bounds θ_{min} and θ_{max}. A commonly used misfit objective function [18], with a regularization term, is given by

$$\begin{aligned} O(\boldsymbol{y},\theta) = & \frac{1}{2}(\theta-\theta_{prior})^T \mathbf{C}_\theta^{-1}(\theta-\theta_{prior}) + \\ & \frac{1}{2}(\boldsymbol{h}(\boldsymbol{x},\theta)-\mathbf{d}_{obs})^T \mathbf{C}_D^{-1}(\boldsymbol{h}(\boldsymbol{x},\theta)-\mathbf{d}_{obs}), \end{aligned} \tag{3.57}$$

where $\mathbf{C}_\theta$ is the parameter covariance matrix, θ_{prior} is the vector containing a prior estimate of the uncertain parameters, $\mathbf{d}_{obs}$ the observed data one desires to match, and $\mathbf{C}_D$ the data covariance matrix. The gradient of Eq. (3.57) is given by

$$\begin{aligned} \nabla_\theta O = & \mathbf{C}_\theta^{-1}(\theta-\theta_{prior}) + \left(\frac{d\boldsymbol{h}}{d\theta}\right)^T \mathbf{C}_D^{-1}(\boldsymbol{h}(\boldsymbol{x},\theta)-\mathbf{d}_{obs}) = \\ & \mathbf{C}_\theta^{-1}(\theta-\theta_{prior}) + \mathbf{G}^T\mathbf{m}. \end{aligned} \tag{3.58}$$

Since calculating the gradient using the adjoint method requires computational cost proportional to one extra simulation, while the direct method requires cost proportional to N_θ extra simulations, the adjoint method is computationally the most efficient one. Note that

$$\mathbf{m} = \mathbf{C}_D^{-1}(\boldsymbol{h}(\boldsymbol{x},\theta)-\mathbf{d}_{obs}), \tag{3.59}$$

where $\mathbf{m}$ is an auxiliary vector, so the gradient of O can be written as $\nabla_\theta O = (\mathbf{m}^T\mathbf{G})^T$. Moreover, Algorithm 4, with $\mathbf{W} = \mathbf{m}^T$, calculates $\nabla_\theta O$ with a cost proportional to one extra simulation, instead of proportional to the number of parameters as in the direct method. For this reason, in the data assimilation studies shown here, the adjoint method is used when evaluating Eq. (7.20).

3.2.4. GRADIENT COMPUTATION AND OPTIMIZATION FOR LIFE-CYCLE OPTIMIZATION

Life-cyle optimization aims to find the optimal set of control input parameters that maximizes an economic objective (e.g. the recovery factor or the net present value). This

problem can also be represented as an optimization problem

$$\begin{aligned} \underset{\theta}{\text{maximize}} \quad & O(\boldsymbol{h}(\boldsymbol{x},\theta)) \\ \text{subject to} \quad & \boldsymbol{g}(\boldsymbol{x},\theta) = \mathbf{0}, \\ & \mathbf{c}(\boldsymbol{x},\theta) = \mathbf{0}, \\ & \mathbf{d}(\boldsymbol{x},\theta) < \mathbf{0}, \\ & \theta \in [\theta_{min},\theta_{max}], \end{aligned} \tag{3.60}$$

where $\mathbf{c}$ and $\mathbf{d}$ represent, respectively, equality and inequality operational contraints (e.g. maximum injection pressure). Now, θ represent the control parameters (e.g. well bottom-hole pressures or rates).

Here, let us assume the economical objective function $O = J$ to be the net present value, which is given in a simplified way by [15]

$$J = \sum_{n=1}^{N} \frac{[(q_{o,n}) \cdot r_o - (q_{wp,n}) \cdot r_{wp} - (q_{wi,n}) \cdot r_{wi}] \cdot \Delta t_n}{(1+b)^{\frac{t_n}{\tau_t}}}. \tag{3.61}$$

In Eq. (3.61), $q_{o,n}$ represents the oil production rate in m^3/day, $q_{wp,n}$ is the water production rate in m^3/day, $q_{wi,n}$ is the water injection rate in m^3/day, r_o is the price of oil produced in $\$/m^3$, r_{wp} is the cost of produced water in $\$/m^3$, r_{wi} is the cost of injected water in $\$/m^3$, Δt_n is the difference between consecutive time steps in days, b is the discount factor expressed as a fraction per year, t_n is the cumulative time in days corresponding to time-step n, and τ_t is the reference time period for discounting, typically one year.

The well rates are computed via the Peaceman [27] formulation as

$$q(\mathbf{x},\theta) = T\lambda_\alpha \left(p_b - p_w\right), \tag{3.62}$$

where p_b is the grid-block pressure, p_w is the well-bore pressure, T is a connectivity index, and λ_α is the mobility of phase α.

Eq. (3.61) can be re-written in vectorial form as

$$J = \mathbf{r}_o^T \mathbf{q}_o - \mathbf{r}_{wp}^T \mathbf{q}_{wp} - \mathbf{r}_{wi}^T \mathbf{q}_{wi}, \tag{3.63}$$

where $\mathbf{q}_o \in \mathbf{R}^N$, $\mathbf{q}_{wp} \in \mathbf{R}^N$, $\mathbf{q}_{wi} \in \mathbf{R}^N$, and

$$\mathbf{r}_o = \begin{bmatrix} \frac{r_o \Delta t_1}{(1+b)^{\frac{t_1}{\tau_t}}} & \cdots & \frac{r_o \Delta t_N}{(1+b)^{\frac{t_N}{\tau_t}}} \end{bmatrix}^T,$$

$$\mathbf{r}_{wp} = \begin{bmatrix} \frac{r_{wp} \Delta t_1}{(1+b)^{\frac{t_1}{\tau_t}}} & \cdots & \frac{r_{wp} \Delta t_N}{(1+b)^{\frac{t_N}{\tau_t}}} \end{bmatrix}^T,$$

$$\mathbf{r}_{wi} = \begin{bmatrix} \frac{r_{wi} \Delta t_1}{(1+b)^{\frac{t_1}{\tau_t}}} & \cdots & \frac{r_{wi} \Delta t_N}{(1+b)^{\frac{t_N}{\tau_t}}} \end{bmatrix}^T.$$

Furthermore, Eq. (3.63) can be rewritten as

$$J = \mathbf{r}^T \boldsymbol{h}, \tag{3.64}$$

where

$$\boldsymbol{h} = \begin{bmatrix} \mathbf{q}_o^T & -\mathbf{q}_{wp}^T & -\mathbf{q}_{wi}^T \end{bmatrix}^T \quad , \quad \mathbf{r} = \begin{bmatrix} \mathbf{r}_o^T & \mathbf{r}_{wp}^T & \mathbf{r}_{wi}^T \end{bmatrix}^T .$$

This allows us to write the gradient of Eq. (3.64) as

$$\nabla_\theta J = \mathbf{r}^T \frac{d\boldsymbol{h}}{d\theta} = \mathbf{r}^T \mathbf{G}. \tag{3.65}$$

Eq. (3.64) allows the adjoint method to be employed in the computation of $\nabla_\theta J$ by making $\mathbf{W} = \mathbf{r}^T$. Just like in the data assimilation case, $\nabla_\theta J$ is efficiently computed with cost proportional to one backward simulation using the adjoint method.

3.2.5. ALGORITHM COMPLEXITY ANALYSIS

As already mentioned, sequential methods can lead to efficient simulation strategies. Because the direct and adjoint derivative computation methods are tightly related to the numerical method employed in the forward simulation, a computational efficiency gain is also observed in these derivative computation methods.

The computational efficiency of the methods is assessed via an asymptotic analysis. In the analysis, only the most computationally intensive operations involved in the algorithms are considered. Hence, because the cost of solving linear system of equations overwhelms the cost of the matrix-vector products, only the former is considered over the latter. The cost associated to the solution of a linear system is considered to be $\mathcal{O}\left(\alpha N^\beta\right)$, where α and β are constants dependent of the linear solvers employed, and N is the size of the system.

Let us consider the computational cost associated to solve the derivative information for each time-step perfomed in the forward simulation for the different methods (FIM, IMPSAT and IMPES). In the FIM case, for each column of $\mathbf{V}$ for the direct method, or each row of $\mathbf{W}$ for the adjoint method, a linear system of size $2 \times N_b$ must be solved, leading to a complexity $\mathcal{O}_{FIM}(\alpha(2 \times N_b)^\beta)$. In the IMPSAT case, a linear system must be solved for the flow and tranport equations, leading to a complexity $\mathcal{O}_{IMPSAT}(\alpha {N_b}^\beta + \alpha {N_b}^\beta)$. Now, in the IMPES case, the saturation can be obtained via an negligble matrix-vector multiplication, which requires the solution of only one linear system of size N_b, leading to a complexity of $\mathcal{O}_{IMPES}(\alpha {N_b}^\beta)$. Fig. 3.1 illustrates the cost ratios $\mathcal{O}_{IMPSAT}/\mathcal{O}_{FIM}$ and $\mathcal{O}_{IMPES}/\mathcal{O}_{FIM}$ for different values of β. It is considered that the linear solver employed in the solution of the different coupling strategies' systems are equally efficient (i.e. same β).

It is possible to see that it is always more or equally efficient to solve the resulting linear system(s) of equations in a sequential manner than using a FIM. Another aspect that is not captured in our analysis is that once we have the system de-coupling, it is possible to employ more efficient solution strategies based on the underlying physics and on the resulting system of equations' properties.

However the cost per time-step associated to the sequential gradient computation methods are smaller or equal to the FIM gradient computation method. Due to numerical instabilities, sequential methods (mainly IMPES) usually require more time-steps

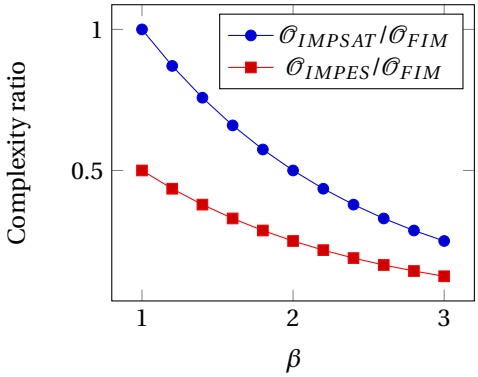


Figure 3.1: Computational complexity ratio between IMPES and FIM (red) and IMPSAT and FIM (blue) for different values of β for one time-step.

than FIM methods due to the limitations imposed by the CFL condition. Therefore, there is a tradeoff between number of time-steps and time-step cost,

$$\left(\frac{\mathcal{O}_{Seq}}{\mathcal{O}_{FIM}}\right)_{Total} = \frac{N_{Seq}^{TS}}{N_{FIM}^{TS}} \frac{\mathcal{O}_{Seq}}{\mathcal{O}_{FIM}},$$

where N_{Seq}^{TS} is the total number of time-steps taken in the sequential (either IMPES of IMPSAT) simulation, N_{FIM}^{TS} is the total number of time-steps taken in the FIM simulation, and $\mathcal{O}_{Seq}$ is the cost associated to the sequential simulation (either IMPES or IMPSAT).

Furthermore, both superior efficiency and stability could be achieved if an adaptive implicit sequential coupling [28, 29] is employed. The framework here presented could be directly employed by properly accouting for the explicit/implicit cells in the computation of the partial derivative matrices. The implementation of an AIM derivative computation method in a fully featured simulator has been used in the literature [19].

Also, we highlight that, although not captured in the above computational asymptotic analisys, it is important to note that the more time-steps are taken by the forward simulation, the more extra information (partial derivative matrices) must be computed and assembled, as well as stored/re-evaluated at each time-step to be later used in the backward simulation.

3.3. NUMERICAL EXPERIMENTS

A synthetic model is considered as proof of concept (see Fig. A.5). It is a 2D inverted five-spot model, consisting of a 21×21 equidistant Cartesian mesh with grid block dimensions of $33.3 \times 33.3 \times 2$ m. The reservoir porosity is constant and equal to 0.3. The fluid properties are described in Table A.2. The uncertainty around the absolute permeability distribution is represented by an ensemble of different permeability realizations. The ensemble is generated via the decomposition of a reference permeability "image" using Principal Component Analysis parametrization [30]. Fig. A.8 illustrates 4 different

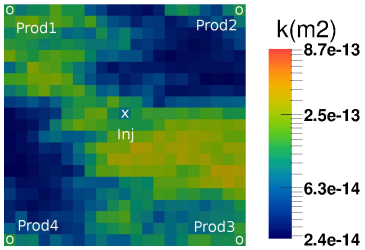


Figure 3.2: The synthetic inverted five-spot model used in the numerical experiments. One of the 1,000 permeability realizations is shown.

Table 3.1: Fluid properties for 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	–

permeability realizations from the ensemble of 1,000 members.

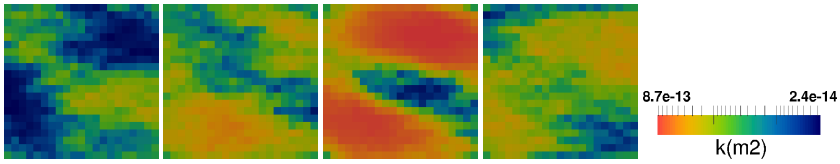


Figure 3.3: Four different permeability realizations from the ensemble of 1,000 members used in the data assimilation study.

3.3.1. GRADIENT ACCURACY

In order to quantify how much the gradients computed by the presented sequential methods deviate from those computed using a FIM method, we calculate the angle between the gradient given by the FIM method and the gradients given by the IMPES and IMPSAT sequential methods. The angle between the gradient vectors can be computed as

$$\alpha = \cos^{-1}\left(\nabla_{\theta}^{T}\hat{O}_{FIM}\,\nabla_{\theta}\hat{O}_{Seq}\right), \tag{3.66}$$

where,

$$\nabla_{\theta}\hat{O}_{FIM} = \frac{\nabla_{\theta}O_{FIM}}{\|\nabla_{\theta}O_{FIM}\|_2} \tag{3.67}$$

and

$$\nabla_{\theta}\hat{O}_{Seq} = \frac{\nabla_{\theta}O_{Seq}}{\left\|\nabla_{\theta}O_{Seq}\right\|_2}. \tag{3.68}$$

Also, $\nabla_{\theta}O_{FS}$ and $\nabla_{\theta}O_{Seq}$ denote the FIM and the sequential (IMPES and FIM) analytical gradients, respectively. As a minimum requirement, acceptable MS gradients are obtained if α is much smaller than 90^o [31].

The error metric has been computed for both the direct (Algorithm 3) and adjoint (Algorithm 4) methods. The metric is assessed for the gradient of the misfit objective function (Eq. (7.20)) and the life-cycle optimization function (Eq. (3.65)), which experiments setup are described, respectively, in Sections 3.3.2 and 3.3.3 3.2. Also, the metric is evaluated considering the gradient computed at the initial parameter values.

Table 3.2: Angle (in degrees) between gradient vectors computed via the FIM method and the IMPES and IMPSAT methods for the synthetic inverted five-spot test case.

	$\nabla_{\theta}O$		$\nabla_{\theta}J$	
	Direct	Adjoint	Direct	Adjoint
$\alpha_{FIM-IMPES}(^o)$	5.5845	5.5845	0.3427	0.3427
$\alpha_{FIM-IMPSAT}(^o)$	5.2232	5.2232	0.5508	0.5508

It can be observed that the angles for both direct and adjoint methods are equally accurate. This is an expected result giving that the difference between the two algorithms is the order in which the operations are evaluated. Also, the angles indicate that algorithms here presented provide gradients that are consistent with the FIM derivative calculation method. That is an indication that, when the gradient computed via the sequential derivative computation algorithms are utilized by a gradient-based algorithm, the optimization solution path should not be too different from an optimization performed utilizing gradients computed by a FIM derivative calculation algorithm. This will be illustrated next, when the gradients are employed in different optimization exercises.

3.3.2. WATER-FLOODING DATA ASSIMILATION

In the data assimilation studies shown here, we run the minimization problem defined by Eq. (7.7) by setting the objective function to be Eq. (3.57) and defining the vector of parameters as the natural logarithm of the permeability in each grid cell

$$\theta = \begin{bmatrix}\ln k_1 & \ln k_2 & \dots & \ln k_{N_b}\end{bmatrix}^T. \tag{3.69}$$

The covariance matrix $\mathbf{C}_{\theta}$ is computed from the ensemble of realizations as

$$\mathbf{C}_{\theta} = \frac{1}{N_e - 1}\left(\Theta - \mu\mathbf{e}^T\right)\left(\Theta - \mu\mathbf{e}^T\right)^T \tag{3.70}$$

where Θ is the $N_b \times N_e$ matrix whose j-th column is given by the member of the ensemble θ_j, $j \in \{1, ..., N_e\}$,

$$\mu = \frac{1}{N_e}\sum_{j=1}^{N_e}\theta_j \tag{3.71}$$

is the ensemble mean, and $\mathbf{e} = [\mathbf{1}, ..., \mathbf{1}]^T$ is a vector of ones of size $N_e \times 1$. In Eq. (3.57), the prior is taken to be the ensemble mean

$$\theta_{prior} = \mu. \tag{3.72}$$

Additionally, $\mathbf{C}_D$ is a diagonal matrix [18] given by

$$\mathbf{C}_D = \sigma^2 \mathbf{I}, \tag{3.73}$$

where σ^2 is the variance of the data measurement error.

The optimization utilizes a limited-memory Broyden Fletcher Goldfarb Shanno (LBFGS) implementation [25]. The LBFGS algorithm requires the objective function gradient. The misfit objective function gradient given by Eq. (7.20) can be computed via the adjoint method (Algorithm 4) with a cost proportional to one simulation backward in time. The optimization stopping criterion is determined by the minimum objective function (OF) value that is possible to satisfy a given noise level [18]. Following the definition of the model parameters in Eq. (3.69), next we show how the framework can be employed with no modifications by defining different model responses.

PERMEABILITY ESTIMATION FROM WELL PRODUCTION DATA ASSIMILATION

In this exercise, we make the responses to be the water rates at the production wells at certain observation times

$$\boldsymbol{h}(\boldsymbol{x}, \theta) = \left[\mathbf{q}_{w_{obs}}^{\text{Prod1}\,T} \quad \mathbf{q}_{w_{obs}}^{\text{Prod2}\,T} \quad \mathbf{q}_{w_{obs}}^{\text{Prod3}\,T} \quad \mathbf{q}_{w_{obs}}^{\text{Prod4}\,T} \right]^T. \tag{3.74}$$

The observed data is generated using a twin experiment. One realization of the permeability ensemble was randomly chosen to be considered the "truth". The water rates resulting from the simulation of 10 years of the model, with a 5% white noise level to represent the measurement error, were considered to be the observed data. The water rates are considered to be observed at every six months.

The water well rate matches for the FIM, IMPES and IMPSAT methods, as obtained from the optimizations, are presented, respectively, in Fig. 3.4, Fig. 3.5, and Fig. 3.6. It can be noted that the gradients computed from the three different forward simulations are successfully employed in the optimization algorithm, leading to matched responses that accurately reproduce the observed data.

The matched permeability fields are shown in Fig. 3.7. The resulting permeability fields are also in good agreement with the reference "truth" model, in particular the permeability orientation.

It can be noted from Fig. 3.8 that the optimizations for the three different forward model coupling strategies follow a similar iteration path. That was expected based on the angles shown in Table 3.2. Since the IMPES and IMPSAT gradients almost do not deviate from the FIM gradient, providing a similar search path, similar minima are found.

SEISMIC DATA ASSIMILATION

In this exercise it is aimed to demonstrated how the framework can seamlessly accommodate different types of model responses. The observed data is now considered to be a spatially distributed response. More specifically, we consider the reservoir pressure

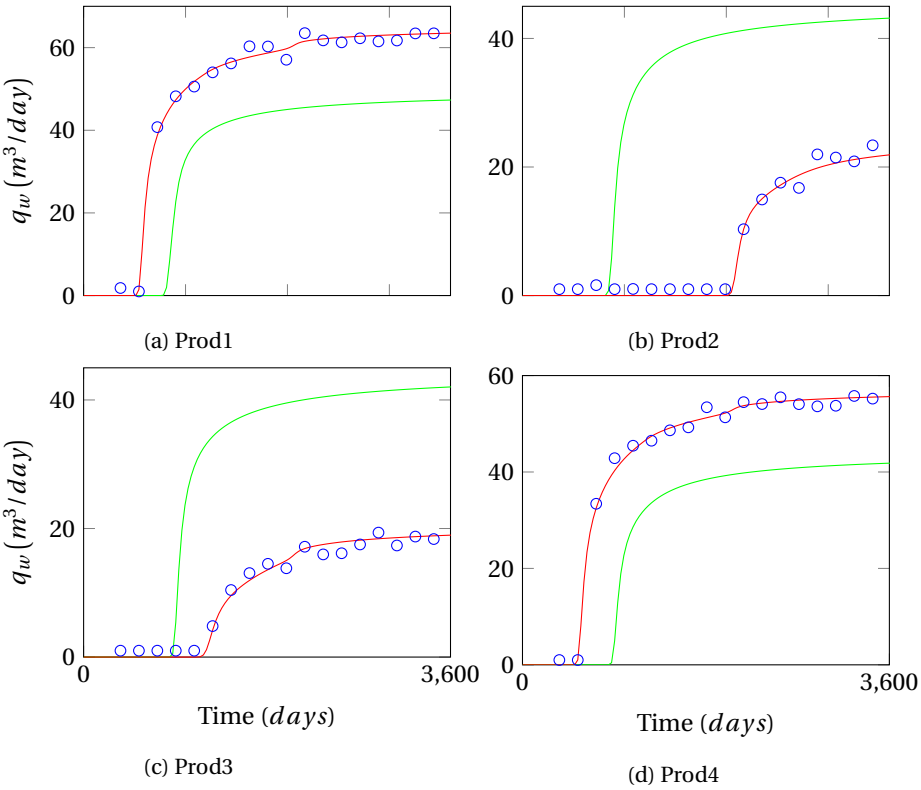


Figure 3.4: Model responses, i.e. well rates, for the well data assimilation exercise utilizing the FIM method. In the figures, the green line represents the initial well rates, the blue circles the observed rates, and the red lines the rates after matching.

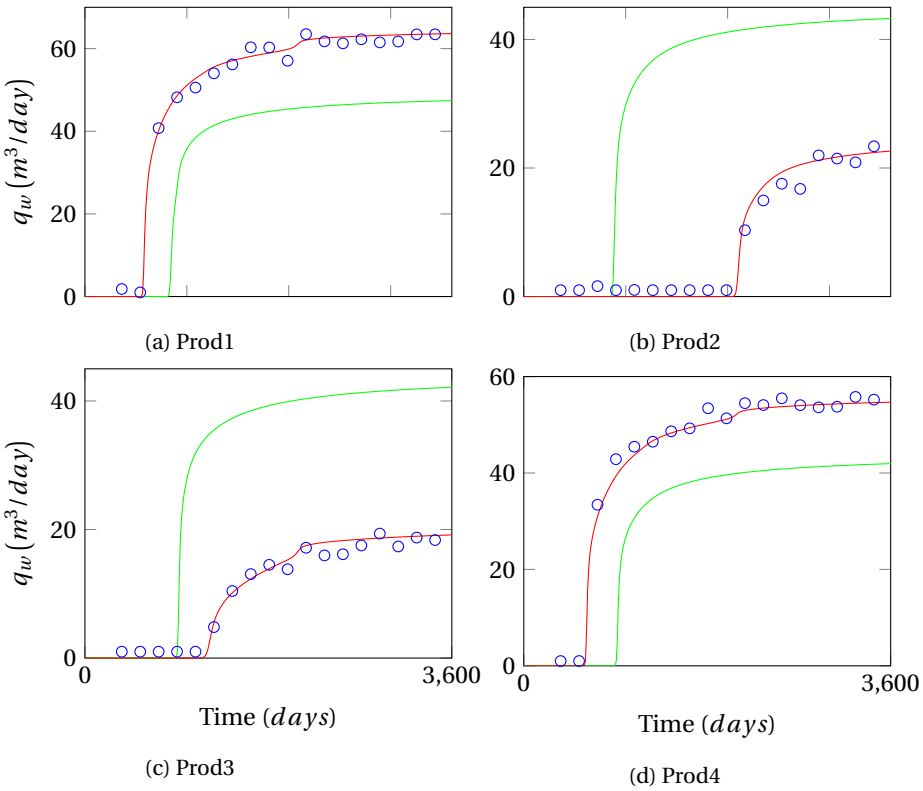
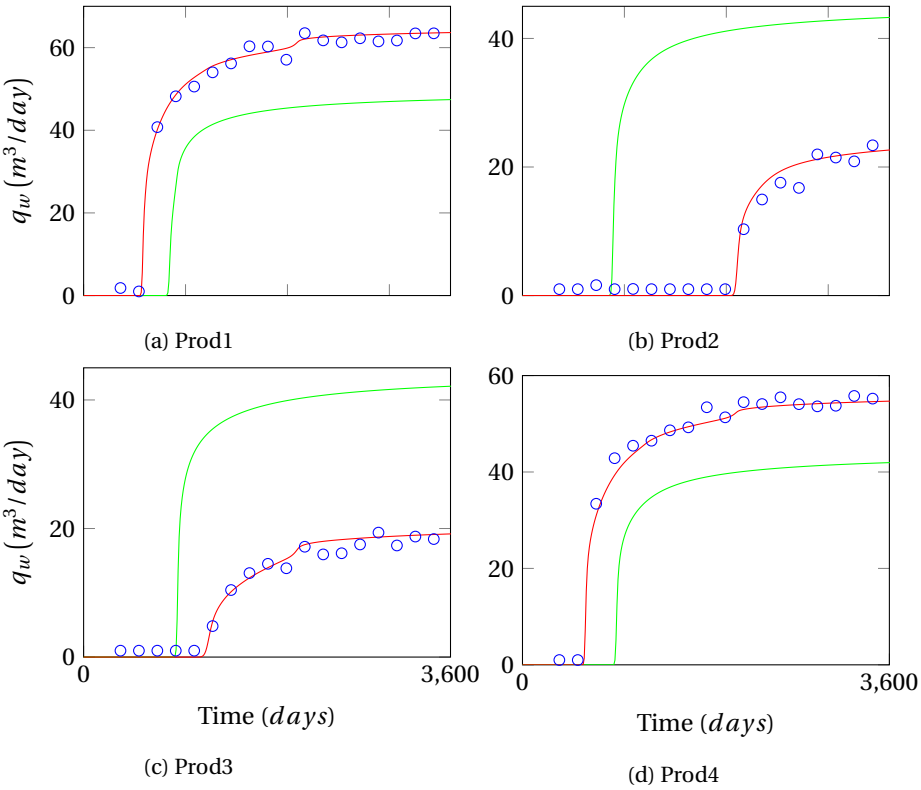


Figure 3.5: Model responses, i.e. well rates, for the well data assimilation exercise utilizing the IMPES method. In the figures, the green line represents the initial well rates, the blue circles the observed rates, and the red lines the rates after matching.



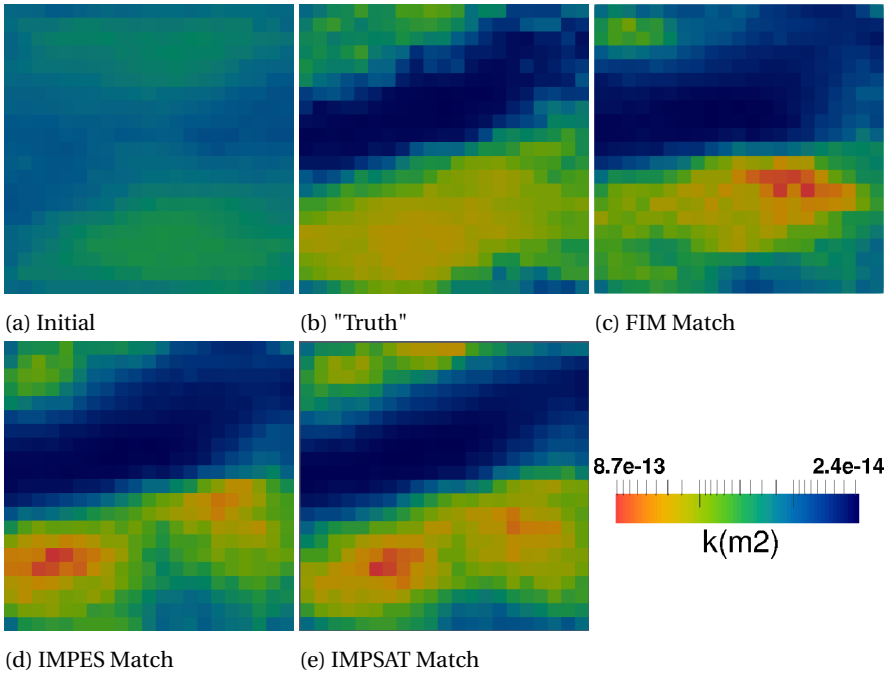
(a) Prod1

(b) Prod2

(c) Prod3

(d) Prod4

Figure 3.6: Model responses, i.e. well rates, for the well data assimilation exercise utilizing the FIM method. In the figures, the green line represents the initial well rates, the blue circles the observed rates, and the red lines the rates after matching.



(a) Initial (b) "Truth" (c) FIM Match

(d) IMPES Match (e) IMPSAT Match

Figure 3.7: Permeability field update for the well data assimilation exercise. Initial permeability field (a), permeability field from "Truth" (b) and permeability field after match utilizing the FIM (c), the IMPES (d) and the IMPSAT (e) methods.

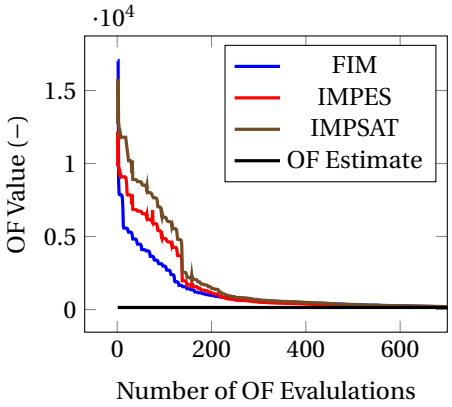


Figure 3.8: Optimization performance for the well production data assimilation exercises with gradients computed from FIM (blue), IMPES (red), and IMPSAT (brown) forward simulations. The OF Estimate line (black) indicates the minimum OF value that is possible to satisfy the observed data noise level [18].

distribution to be the observed data. Such data can be obtained e.g. from a seismic survey. The reservoir pressure can be attained from the seismic images if techniques like the ones presented in [32] and [33] are used. In this case, the derivative computation framework can be applied just like in the previous exercise by making

$$\boldsymbol{h}(\boldsymbol{x},\theta) = \begin{bmatrix} p_1 & p_2 & \dots & p_N \end{bmatrix}^T. \tag{3.75}$$

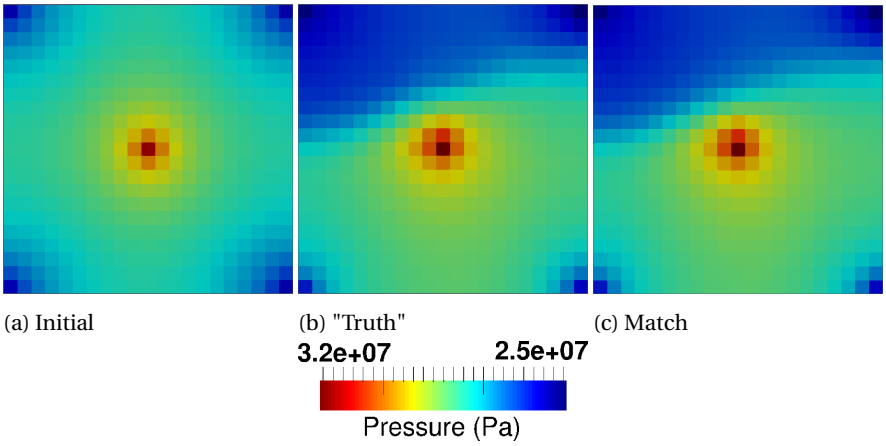
The observed pressure values are taken from the same twin experiment used in the well data history matching shown previously. Also, in the forward simulation, the flow and transport equations are sequentially coupled using the IMPES strategy.

We note that the framework is still applicable if any other spatially distributed property is considered as an observation, e.g. the more widely used impedances, provided that the necessary Jacobians of $\boldsymbol{h}$ (e.g. via the derivatives of the petro-elastic equations) are available [34, 35].

In this experiment, perfect observed data is considered, with measurement error in the range of those usually employed in synthetic studies (see e.g. [18]), is employed.

The pressure match is illustrated in Fig. 3.9, while the resulting permeability field after the data assimilation is presented in Fig. 3.10.

Here, the IMPES method is employed in the forward simulation.



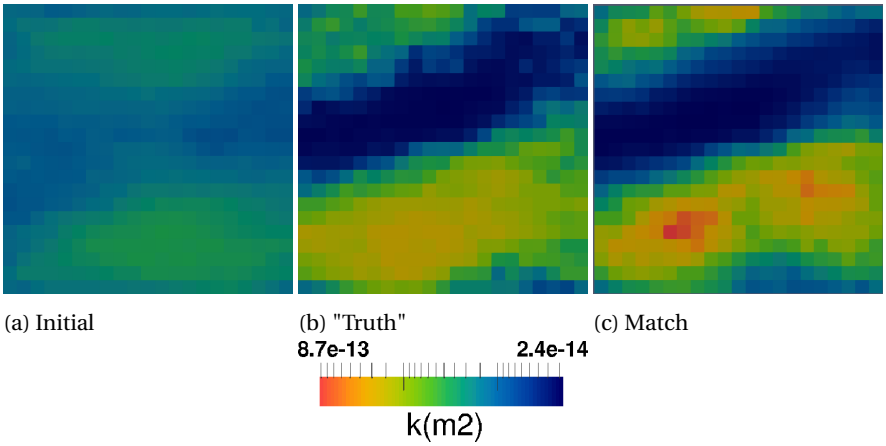
(a) Initial (b) "Truth" (c) Match

Figure 3.9: Model responses, i.e. pressure distribution, for the seismic data assimilation exercise. Initial pressure distribution (a), response from the "truth" (b) and pressure distribution after the match (c).

Once again, the data assimilation process results in a matched model that recovers the twin experiment response. Just like in the previous example, the resulting permeability field is in good agreement with the reference.

3.3.3. WATER-FLOODING LIFE-CYCLE OPTIMIZATION

In the life-cycle optimization studies shown here, we run the maximization problem defined by Eq. (3.60) by setting the objective function as in Eq. (3.64) and defining the



(a) Initial (b) "Truth" (c) Match

Figure 3.10: Permeability field update for the seismic data assimilation exercise. Initial permeability field (a), permeability field from "Truth" (b) and permeability field after match (c).

vector of parameters to be the well bottom-hole pressures at some given control times

$$\theta = \begin{bmatrix} p^1_{w_1} & \cdots & p^1_{w_{N_w}} & \cdots & p^{K_C}_{w_1} & \cdots & p^{K_C}_{w_{N_w}} \end{bmatrix}^T \tag{3.76}$$

where N_w is the total number of controlled wells and K_C is the total number of timesteps when a control change occurs. The economical parameters for oil production are defined in Table A.1.

Table 3.3: Economic parameters associated with oil production.

	Value	Unit
Oil price	252	$/m^3
Cost of injected water	60	$/m^3
Cost of produced water	30	$/m^3

By allowing all well bottom-hole pressure values (5 in total) to change every 6 control time steps of 720 days gives a total number of control parameters equal to 30. The values of the bottom-hole pressures are bounded for the production wells between a minimum value of 28 MPa and a maximum value of 30 MPa. The injection well pressures are bounded between a minimum value of 30 MPa and maximum value of 32 MPa. The initial starting strategy is one wherein the injector well operates at a constant BHP of 31 MPa and the production wells at a constant BHP of 29 MPa.

In this exercise, the optimization utilizes the steepest ascent algorithm [25]. The line-search step length is reduced by half if the newly proposed controls given by the gradient do not lead to an increase of the objective function. This backtracking is allowed to be repeated five times. The control parameters are normalized with respect to the difference between the bounds, and a normalized gradient is considered in the line search direction computation.

Here, we run three different optimizations with different forward model coupling strategies: FIM, IMPES, and IMPSAT.

The optimization performance is illustrated in Fig. 3.12 and the optimal control parameters found by the optimization of the different coupling strategies can be found in Fig. 3.11.

It can be noted that, for the three couplings considered, the framework provides consistent objective function gradients that provide similar search directions. Similar optimized NPVs are also achieved, with an NPV increase of approximately 20%. Furthermore, except for small deviations in the injection well, the optimal control strategies are nearly identical between the different optimization runs.

REFERENCES

[1] R. J. de Moraes, J. R. P. Rodrigues, H. Hajibeygi, and J. D. Jansen, *Computing derivative information of sequentially coupled subsurface models,* Computational Geosciences (2018), 10.1007/s10596-018-9772-2.

[2] R. J. de Moraes, H. Hajibeygi, and J. D. Jansen, *Multiscale gradient computation for sequentially coupled flow and transport in heterogeneous porous media,* in *Conference on Mathematical and Computational Issues in the Geosciences.*

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

[4] S. Lee, Y. Efendiev, and H. Tchelepi, *Hybrid upwind discretization of nonlinear two-phase flow with gravity,* Advances in Water Resources **82**, 27 (2015).

[5] T. Y. Hou and X.-H. Wu, *A multiscale finite element method for elliptic problems in composite materials and porous media,* J. Comput. Phys. **134**, 169 (1997).

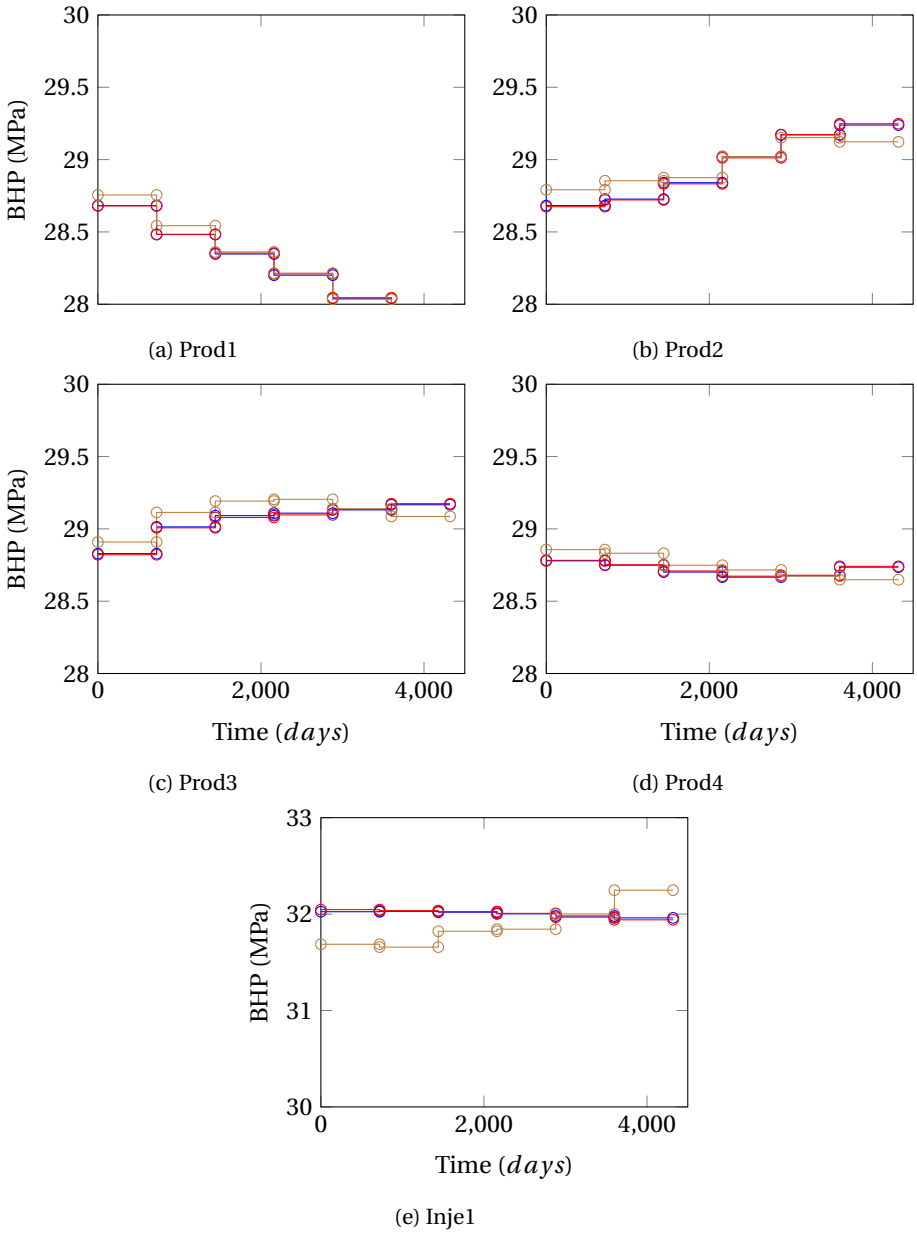
[6] P. Jenny, S. Lee, and H. Tchelepi, *Adaptive multiscale finite-volume method for multiphase flow and transport in porous media,* Multiscale Modeling & Simulation **3**, 50 (2005).

[7] H. Hajibeygi and P. Jenny, *Adaptive iterative multiscale finite volume method,* Journal of Computational Physics **230**, 628 (2011).

[8] O. Møyner and K.-A. Lie, *A multiscale restriction-smoothed basis method for high contrast porous media represented on unstructured grids,* Journal of Computational Physics **304**, 46 (2016).

[9] M. Cardoso, L. Durlofsky, and P. Sarma, *Development and application of reduced-order modeling procedures for subsurface flow simulation,* International journal for numerical methods in engineering **77**, 1322 (2009).

[10] J. F. van Doren, R. Markovinović, and J.-D. Jansen, *Reduced-order optimal control of water flooding using proper orthogonal decomposition,* Computational Geosciences **10**, 137 (2006), http://dx.doi.org/10.1007/s10596-005-9014-2.



(a) Prod1

(b) Prod2

(c) Prod3

(d) Prod4

(e) Inje1

Figure 3.11: Optimal controls (bottom-hole pressures) resulting from the optimization exercises with gradients computed from FIM (blue), IMPES (red), and IMPSAT (brown) forward simulations.

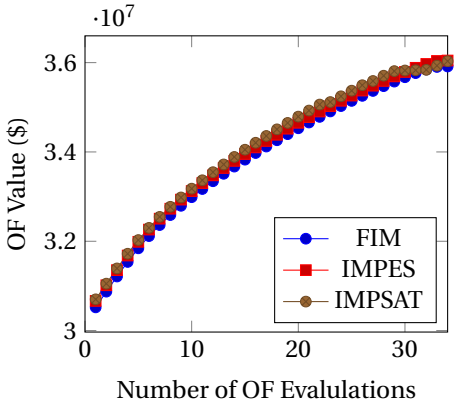


Figure 3.12: Optimization performance for the optimization exercises with gradients computed from FIM (blue), IMPES (red), and IMPES (brown) forward simulations.

[11] J. R. Wallis, R. Kendall, and T. Little, *Constrained residual acceleration of conjugate residual methods,* in *SPE Reservoir Simulation Symposium* (Society of Petroleum Engineers, 1985).

[12] C. Han, J. Wallis, P. Sarma, G. Li, M. Schrader, and W. H. Chen, *Adaptation of the cpr preconditioner for efficient solution of the adjoint equation,* SPE Journal **18**, 207 (2013).

[13] M. Cusini, C. van Kruijsdijk, and H. Hajibeygi, *Algebraic dynamic multilevel (adm) method for fully implicit simulations of multiphase flow in porous media,* Journal of Computational Physics **314**, 60 (2016).

[14] P. Sarma, K. Aziz, and L. J. Durlofsky, *Implementation of adjoint solution for optimal control of smart wells,* in *SPE reservoir simulation symposium* (Society of Petroleum Engineers, 2005).

[15] J. D. Jansen, *Adjoint-based optimization of multi-phase flow through porous media–a review,* Computers & Fluids **46**, 40 (2011).

[16] G. Chavent, M. Dupuy, and P. Lemmonier, *History matching by use of optimal theory,* Society of Petroleum Engineers Journal **15**, 74 (1975).

[17] R. Li, A. Reynolds, and D. Oliver, *History matching of three-phase flow production data,* SPEJ **8**, 328 (2003).

[18] D. S. Oliver, A. C. Reynolds, and N. Liu, *Inverse theory for petroleum reservoir characterization and history matching* (Cambridge University Press, 2008).

[19] J. R. P. Rodrigues, *Calculating derivatives for automatic history matching,* Computational Geosciences **10**, 119 (2006).

[20] J. F. B. M. Kraaijevanger, P. J. P. Egberts, J. R. Valstar, and H. W. Buurman, *Optimal waterflood design using the adjoint method,* in *SPE Reservoir Simulation Symposium* (Society of Petroleum Engineers, 2007).

[21] S. Krogstad, V. L. Hauge, and A. Gulbransen, *Adjoint multiscale mixed finite elements,* SPE Journal **16**, 162 (2011).

[22] J. D. Jansen, *Gradient-based optimization of flow through porous media, v.3,* (2016).

[23] R. J. de Moraes, J. R. Rodrigues, H. Hajibeygi, and J. D. Jansen, *Multiscale gradient computation for flow in heterogeneous porous media,* Journal of Computational Physics **336**, 644 (2017).

[24] F. Anterion, R. Eymard, and B. Karcher, *Use of parameter gradients for reservoir history matching,* in *SPE Symposium on Reservoir Simulation* (Society of Petroleum Engineers, 1989).

[25] J. Nocedal and S. Wright, *Numerical optimization* (Springer Science & Business Media, 2006).

[26] K. Coats, *Impes stability: selection of stable timesteps,* SPE Journal **8**, 181 (2003).

[27] D. W. Peaceman, *Interpretation of well-block pressures in numerical reservoir simulation with nonsquare grid blocks and anisotropic permeability,* Society of Petroleum Engineers Journal **23**, 531 (1983).

[28] G. Thomas and D. Thurnau, *Reservoir simulation using an adaptive implicit method,* Society of Petroleum Engineers Journal **23**, 759 (1983).

[29] P. Forsyth Jr and P. Sammon, *Practical considerations for adaptive implicit methods in reservoir simulation,* Journal of Computational Physics **62**, 265 (1986).

[30] J. D. Jansen, *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, 2013).

[31] R. M. Fonseca, S. S. Kahrobaei, L. J. T. Van Gastel, O. Leeuwenburgh, and J. D. Jansen, *Quantification of the impact of ensemble size on the quality of an ensemble gradient using principles of hypothesis testing,* in *SPE Reservoir Simulation Symposium* (Society of Petroleum Engineers, 2015).

[32] C. MacBeth, M. Floricich, and J. Soldo, *Going quantitative with 4d seismic analysis,* Geophysical Prospecting **54**, 303 (2006).

[33] D. Lumley, M. Meadows, S. Cole, and D. Adams, *Estimation of reservoir pressure and saturations by crossplot inversion of 4d seismic attributes,* in *SEG Technical Program Expanded Abstracts 2003* (Society of Exploration Geophysicists, 2003) pp. 1513–1516.

[34] O. Gosselin, S. Aanonsen, I. Aavatsmark, A. Cominelli, R. Gonard, M. Kolasinski, F. Ferdinandi, L. Kovacic, and K. Neylon, *History matching using time-lapse seismic (huts),* in *SPE Annual Technical Conference and Exhibition* (Society of Petroleum Engineers, 2003).

[35] A. A. Emerick, R. Moraes, and J. Rodrigues, *History matching 4d seismic data with efficient gradient based methods,* in *EUROPEC/EAGE Conference and Exhibition* (Society of Petroleum Engineers, 2007).

4

MULTISCALE GRADIENT COMPUTATION FOR FLOW IN HETEROGENEOUS POROUS MEDIA

An efficient multiscale (MS) gradient computation method for subsurface flow management and optimization is introduced. The general, algebraic framework allows for the calculation of gradients using both the Direct and Adjoint derivative methods. The framework also allows for the utilization of any MS formulation that can be algebraically expressed in terms of a restriction and a prolongation operator. This is achieved via an implicit differentiation formulation. The approach favors algorithms for multiplying the sensitivity matrix and its transpose with arbitrary vectors. This provides a flexible way of computing gradients in a form suitable for any given gradient-based optimization algorithm. No assumption w.r.t. the nature of the problem or specific optimization parameters is made. Therefore, the framework can be applied to any gradient-based study. In the implementation, extra partial derivative information required by the gradient computation is computed via automatic differentiation. A detailed utilization of the framework using the MS Finite Volume (MSFV) simulation technique is presented. Numerical experiments are performed to demonstrate the accuracy of the method compared to a fine-scale simulator. In addition, an asymptotic analysis is presented to provide an estimate of its computational complexity. The investigations show that the presented method casts an accurate and efficient MS gradient computation strategy that can be successfully utilized in next-generation reservoir management studies.

The material presented in this chapter has been published in the Journal of Computational Physics **336**, (2017) [1] and in the proceedings of the European Conference on the Mathematics of Oil Recovery (ECMOR) XV (2016) [2].

Model-based reservoir management techniques typically rely on the information provided by derivatives. For instance, in sensitivity analysis studies, derivatives can be directly used to identify the most influential parameters in the reservoir response. Also, derivative information can be utilized in history matching [3] and control optimization [4] studies, where gradient-based optimization techniques are employed in the minimization/maximization of an objective function.

These types of model-based reservoir management studies are computationally demanding. They require multiple evaluations of the reservoir model in order to compute its response under the influence of different inputs. Reduced-order modeling (ROM) techniques have been employed to reduce the computational cost of the reservoir response evaluation. In sensitivity analysis studies, response surface models and design of experiments are often used to reduce the computational costs (see, e.g., [5]). In history matching and optimization studies, techniques like upscaling [6], streamline simulation [7], and proper orthogonal decomposition [8] are employed to create reservoir models that are faster to evaluate. However, ROM and upscaling methods usually do not provide accurate enough system responses due to excessive simplifications of the fluid-rock physics and heterogeneous geological properties. To resolve this challenge, Multiscale (MS) methods have been developed [9–11].

MS methods solve a coarser simulation model, thus increasing the computational speed, while resolving the fine scale heterogeneities. Note that the specific multiscale methods addressed here, map between fine and coarse grids that are both at continuum (Darcy) scale, but with different computational grid resolutions. Moreover, the map between the nested fine and coarse grids is developed by using multiscale basis functions [9]. The basis functions are local solutions of the fine-scale equation, which are adaptively updated [12, 13] and allow the MS coarse system to account for the fine-scale heterogeneities (which typically do not have separation of scale). Note that in contrast with MultiGrid (MG) methods, MS methods are not developed as linear solvers, but are most efficient if they are used –similar as in this paper– to provide approximate conservative fine-scale solutions (crucial for multiphase systems). MS methods are found efficient and accurate for large-scale reservoir models [14, 15]. Important in this class of MS methods (compared to upscaling methods) is that the coarse-scale solutions are mapped onto the original fine scale, using the same basis functions. As such, errors can be calculated against the fine-scale reference systems (and not upscaled averaged ones). This allows for the development of convergent iterative MS procedures [16–18]. Recent developments include MS formulations for fractured media [19, 20] with compositional effects [14, 15, 21] and complex well configurations [22] and gravitational effects [23]. In addition, algebraic formulation of the method has made it convenient to be integrated within existing simulators using structured [24, 25] and unstructured [26–28] grids. The method has been also extended to fully-implicit formulations where all unknowns cross multiple dynamically-defined scales [29]. Although these developments are found efficient, they are mainly limited to forward simulation modeling. In this paper the focus is on the use of MS methods within reservoir management workflows.

Reservoir management techniques include optimization algorithms, in which calculation of derivatives plays an important role. The classical approaches for calculation of derivatives are either computationally expensive or inaccurate. For instance, numerical

differentiation (see, e.g., [30, 31]) suffers from discretization approximations and truncation errors, and is impractical when the number of parameters is large. Alternatively, analytical methods –Direct Methods (or Gradient Simulators) [32] and Adjoint Methods [33, 34]– can provide accurate and efficient derivatives under appropriate conditions (to be further discussed in the Section 4.1). However, the use of analytical methods has not been extensively adopted mainly because they are code-intrusive and require a substantial implementation effort. On this issue, automatic differentiation can partly alleviate the burden of computing derivative information [35]. Additionally, most commercial simulators do not provide analytical derivative capabilities nor do they provide access to extend their functionality in this direction. Partially due to these drawbacks, ensemble methods such as the Ensemble Kalman Filter (EnKF) have become very popular in the data assimilation community [36]. Similarly, stochastic approximate gradient techniques such as ensemble optimization (EnOpt) and the stochastic simplex approximate gradient (StoSAG) method are increasingly being used for life cycle optimization [37, 38]. These methods, however, by construction, provide an approximation of the gradient.

Multiscale gradient computation has been studied in the past. A Multiscale finite-volume Adjoint (MSADJ) method has been applied to sensitivity computation [39], where the global adjoint problem is solved via a set of coupled sub-grid problems described at a coarser scale. The coarse-scale sensitivities are then interpolated to the local fine grid by reconstructing the local variability of the model parameters with the aid of solving embedded adjoint sub-problems. In a follow up paper [40], the MSADJ method was efficiently applied to inverse problems of single-phase flow in heterogeneous porous media. Also, an efficient Multiscale Mixed Finite Element method has been developed for multiphase adjoint formulations, where both pressure and saturations are solved at the coarse scale [41]. In contrast to MSADJ, this method did not require fine-scale quantities.

The present development introduces a mathematical framework to compute sensitivities (gradients) in a multiscale strategy. The framework enables the same computational efficiency as existing multiscale methods [39–41]. However, its formulation allows for full flexibility with respect to the types of gradient information that are required by the different model-based reservoir management studies. This is achieved via an implicit differentiation strategy, as opposed to the more traditional Lagrange multiplier formulation. Also, the formulation naturally provides not only the Adjoint Method, but also the Direct Method. It is important to note that, although in this work the multiscale finite volume (MSFV) is being studied, the proposed MS-gradient method is not restricted to a specific MS method. Instead, it can be utilized in combination with any MS (and multilevel) strategy which is expressed in terms of restriction and prolongation operators.

This paper is structured as follows. First, the multiscale gradient computation method is derived based on the MS reservoir model equations and the respective model responses. The computation of the required prolongation (matrix of basis functions) operator derivatives is developed within the Multiscale Finite Volume (MSFV) formulation. Computational complexity of the method is also discussed from a theoretical point of view via asymptotic analysis of the algorithms. Thereafter, the Numerical Experiments section describes a systematic investigation of the validation, robustness, and accuracy of the MS-gradient method for test cases of increasing complexity. Because the proposed method is quite fundamental, the experiments are aimed at evaluating the gradient

computation itself, rather than any specific application.

4.1. DERIVATION OF THE MULTISCALE GRADIENT COMPUTATION MATHEMATICAL FRAMEWORK

4.1.1. FORWARD SIMULATION MODEL

The derivation of the forward simulation model's response with respect to the parameters starts by firstly describing its equations in a generic, purely algebraic form. The analysis is restricted to quasi-steady state problems. In that case, the set of equations that describes the forward simulation at the fine scale can be algebraically expressed, without any assumption regarding the underlying physical model, as

$$\mathbf{g}_F(\mathbf{x},\theta) = \mathbf{0}, \tag{4.1}$$

$$\mathbf{x} = \mathbf{x}(\theta). \tag{4.2}$$

Once the model state is determined, the observable responses of the forward model are computed. The forward model responses (typically wellbore pressures and/or rates) may not only depend on the model state, but also on the parameters themselves, and can be expressed as

$$\mathbf{y}_F = \mathbf{h}_F(\mathbf{x},\theta), \tag{4.3}$$

where $\mathbf{h}_F : \mathbb{R}^{N_F} \times \mathbb{R}^{N_\theta} \rightarrow \mathbb{R}^{N_y}$ represents the output equations [42]. It is assumed that $\mathbf{g}_F$ can be described as

$$\mathbf{g}_F(\mathbf{x},\theta) = \mathbf{A}(\theta)\mathbf{x} - \mathbf{q}(\theta), \tag{4.4}$$

where $\mathbf{A}(\theta)$ is an $N_F \times N_F$ matrix and $\mathbf{q}(\theta)$ is an N_F vector.

4.1.2. ALGEBRAIC MULTISCALE FORMULATION OF FLOW IN HETEROGENEOUS POROUS MEDIA

MS methods provide accurate and efficient solutions to the flow equations by incorporating the fine-scale heterogeneities in a coarse-scale operator [10]. This is achieved by basis functions, which are local solutions of the governing equations without right-hand-side (RHS) terms, subject to approximate local boundary conditions. These local basis functions construct the prolongation (interpolation) operator, $\mathbf{P} = \mathbf{P}(\theta)$, which is an $N_F \times N_C$ matrix that maps (interpolates) the coarse-scale solution to the fine-scale resolution. For the purpose of the development presented here, the required basic knowledge about the multiscale strategy is that a coarse-scale system can be algebraically described once the restriction operator, $\mathbf{R} = \mathbf{R}(\theta)$, is defined as an $N_C \times N_F$ matrix which maps the fine scale to the coarse scale (more information can be found in [43], [24]). Let $\check{\mathbf{x}} \in \mathbb{R}^{N_C}$ be the coarse scale solution ($N_C \ll N_F$), $\mathbf{R} = \mathbf{R}(\theta)$ be an $N_C \times N_F$ matrix mapping from the fine scale to the coarse scale and $\mathbf{P} = \mathbf{P}(\theta)$ be an $N_F \times N_C$ matrix mapping from the coarse scale to the fine scale. $\check{\mathbf{x}}$ is obtained by solving

$$\check{\mathbf{g}} = (\mathbf{RAP})\check{\mathbf{x}} - \big(\mathbf{Rq}\big) = \check{\mathbf{A}}\check{\mathbf{x}} - \check{\mathbf{q}} = \check{\mathbf{0}}. \tag{4.5}$$

Finally, the approximated fine-scale solution $\mathbf{x}'$ is obtained by interpolating the coarse scale solution $\check{\mathbf{x}}$, i.e.,

$$\mathbf{x}' = \mathbf{P}\check{\mathbf{x}}. \tag{4.6}$$

4.1.3. DERIVATIVE CALCULATION OF SIMULATOR RESPONSES

In order to derive an expression to compute the multiscale derivatives with respect to the parameters, an implicit differentiation scheme is followed [44, 45]. The implicit differentiation scheme facilitates both the Direct and the Adjoint methods, in contrast to the formulations based on the standard Lagrange multiplier [4, 30]. In addition, through its specific algebraic form, it allows for providing any gradient information required by the selected optimization algorithm.
Based on Eq. (4.6), the function $\mathbf{g}'$ is defined as

$$\mathbf{g}' = \mathbf{x}' - \mathbf{P}\check{\mathbf{x}} = \mathbf{0}, \tag{4.7}$$

which represents the multiscale procedure to find approximate primary variables at the fine scale. The state vector is now described as a combination of both sets of primary variables at the fine and coarse scales, i.e.,

$$\mathbf{x} = \begin{bmatrix} \check{\mathbf{x}} \\ \mathbf{x}' \end{bmatrix}, \tag{4.8}$$

and, similarly, the model equations are represented as a combination of the equations at both scales, i.e.,

$$\mathbf{g}(\mathbf{x}, \theta) = \begin{bmatrix} \check{\mathbf{g}} \\ \mathbf{g}' \end{bmatrix} = \mathbf{0}. \tag{4.9}$$

The definition of the state vector as in Eq. (7.32), as a key aspect of this development, allows for describing the state not only at the fine scale, but also at the coarse scale. The simulator responses $\mathbf{y}$ obtained from the multiscale method are represented as

$$\mathbf{y} = \mathbf{h}(\mathbf{x}, \theta). \tag{4.10}$$

The sensitivity matrix $\mathbf{G}$ is then computed by obtaining the derivative of Eq. (7.34) with respect to θ as

$$\mathbf{G} = \frac{d\mathbf{h}}{d\theta} = \frac{\partial \mathbf{h}}{\partial \mathbf{x}}\frac{d\mathbf{x}}{d\theta} + \frac{\partial \mathbf{h}}{\partial \theta}. \tag{4.11}$$

In order to find a relationship that defines the derivative of the state vector with respect to the parameters, Eq. (7.33) is differentiated with respect to θ

$$\frac{\partial \mathbf{g}}{\partial \mathbf{x}}\frac{d\mathbf{x}}{d\theta} + \frac{\partial \mathbf{g}}{\partial \theta} = \mathbf{0}, \tag{4.12}$$

so that

$$\frac{d\mathbf{x}}{d\theta} = -\left(\frac{\partial \mathbf{g}}{\partial \mathbf{x}}\right)^{-1}\frac{\partial \mathbf{g}}{\partial \theta}. \tag{4.13}$$

Substituting Eq. (4.13) in Eq. (4.11) gives

$$\mathbf{G} = -\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\left(\frac{\partial \mathbf{g}}{\partial \mathbf{x}}\right)^{-1}\frac{\partial \mathbf{g}}{\partial \theta} + \frac{\partial \mathbf{h}}{\partial \theta}. \tag{4.14}$$

From Eq. (4.5), Eq. (4.7), Eq. (7.32) and Eq. (7.33), it follows that

$$\frac{\partial \mathbf{g}}{\partial \mathbf{x}} = \begin{bmatrix} \mathbf{RAP} & \mathbf{0} \\ -\mathbf{P} & \mathbf{I} \end{bmatrix}. \tag{4.15}$$

Note that

$$\left(\frac{\partial \mathbf{g}}{\partial \mathbf{x}}\right)^{-1} = \begin{bmatrix} (\mathbf{RAP})^{-1} & \mathbf{0} \\ \mathbf{P}(\mathbf{RAP})^{-1} & \mathbf{I} \end{bmatrix} \tag{4.16}$$

holds. Thus, Eq. (4.14) can be restated as

$$\mathbf{G} = \left(\frac{\partial \mathbf{h}}{\partial \check{\mathbf{x}}} + \frac{\partial \mathbf{h}}{\partial \mathbf{x}'}\mathbf{P}\right)(\mathbf{RAP})^{-1}\frac{\partial \check{\mathbf{g}}}{\partial \theta} + \frac{\partial \mathbf{h}}{\partial \mathbf{x}'}\frac{\partial \mathbf{g}'}{\partial \theta} + \frac{\partial \mathbf{h}}{\partial \theta}. \tag{4.17}$$

By deriving Eq. (4.5) with respect to θ one finds

$$\frac{\partial \check{\mathbf{g}}}{\partial \theta} = \left[\frac{\partial \mathbf{R}}{\partial \theta}(\mathbf{AP}) + \mathbf{R}\frac{\partial \mathbf{A}}{\partial \theta}\mathbf{P} + (\mathbf{RA})\frac{\partial \mathbf{P}}{\partial \theta}\right]\check{\mathbf{x}} - \frac{\partial \mathbf{R}}{\partial \theta}\mathbf{q} - \mathbf{R}\frac{\partial \mathbf{q}}{\partial \theta} \tag{4.18}$$

and also, from Eq. (4.7),

$$\frac{\partial \mathbf{g}'}{\partial \theta} = -\frac{\partial \mathbf{P}}{\partial \theta}\check{\mathbf{x}}. \tag{4.19}$$

Note that the partial derivatives of the matrices **A**, **R** and **P** with respect to the vector of parameters θ result in (third order) tensors. The appropriate interpretation of tensor operations can be found in Appendix A. The substitution of Eq. (6.19) and Eq. (4.19) in Eq. (4.17) leads to an expression that will serve as the basis for the multiscale gradient computation, i.e.,

$$\begin{aligned} \mathbf{G} = &\left(\frac{\partial \mathbf{h}}{\partial \check{\mathbf{x}}} + \frac{\partial \mathbf{h}}{\partial \mathbf{x}'}\mathbf{P}\right)(\mathbf{RAP})^{-1} \\ &\left\{\left[\frac{\partial \mathbf{R}}{\partial \theta}(\mathbf{AP}) + \mathbf{R}\frac{\partial \mathbf{A}}{\partial \theta}\mathbf{P} + (\mathbf{RA})\frac{\partial \mathbf{P}}{\partial \theta}\right]\check{\mathbf{x}} - \frac{\partial \mathbf{R}}{\partial \theta}\mathbf{q} - \mathbf{R}\frac{\partial \mathbf{q}}{\partial \theta}\right\} - \\ &\frac{\partial \mathbf{h}}{\partial \mathbf{x}'}\frac{\partial \mathbf{P}}{\partial \theta}\check{\mathbf{x}} + \frac{\partial \mathbf{h}}{\partial \theta}. \end{aligned} \tag{4.20}$$

Eq. (4.20) is quite general, i.e., it can be employed to compute gradients for any MS (and multi-level) method, given that the partial derivatives of **R** and **P** are provided.

For the sake of simplicity and to be coherent with the numerical experiments presented below, from now on the dependency of the restriction operator **R** and the right-hand-side vector **q** on the parameters is neglected. This is consistent with the MSFV method (where the restriction operator is based on a finite volume integration operator at coarse scale (see e.g. [24]). After these simplifications, Eq. (4.20) can be rewritten as

$$\mathbf{G} = \left(\frac{\partial \mathbf{h}}{\partial \bar{\check{\mathbf{x}}}} + \frac{\partial \mathbf{h}}{\partial \mathbf{x}'}\mathbf{P}\right)(\mathbf{RAP})^{-1}\mathbf{R}\left(\frac{\partial \mathbf{A}}{\partial \theta}\mathbf{P} + \mathbf{A}\frac{\partial \mathbf{P}}{\partial \theta}\right)\bar{\check{\mathbf{x}}} - \frac{\partial \mathbf{h}}{\partial \mathbf{x}'}\frac{\partial \mathbf{P}}{\partial \theta}\bar{\check{\mathbf{x}}} + \frac{\partial \mathbf{h}}{\partial \theta}. \tag{4.21}$$

The order of the operations involving the $(\mathbf{RAP})^{-1}$ term is crucial to determine the computational performance of the sensitivity matrix computation. This order defines two different algorithms known in the literature as the Direct Method and the Adjoint Method. The following sections discuss how the two methods are derived for the MS scenario defined by Eq. (4.21), as well as the (dis)advantages of utilizing each of them.

4.2. COMPUTATION OF GRADIENT INFORMATION: GENERALIZATION OF THE FRAMEWORK

According to the type of study one wants to perform, different derivative information has to be provided. For instance, for optimization methods, Quasi-Newton methods require the gradient of the objective function (and possibly constraints). In history matching algorithms, as well as in sensitivity analysis studies, the sensitivity matrix **G**, defined as the matrix of derivatives of all responses with respect to all parameters, plays an important role. In Gauss-Newton methods, the matrix product $\mathbf{G}^T\mathbf{G}$ is directly used, while conjugate gradient methods require products of **G** and $\mathbf{G}^T$ with arbitrary vectors. More detailed information on the different optimization algorithms can found in [46].

To preserve the general applicability of our method, the general problem of multiplying **G** by arbitrary matrices **W** (of order $m \times N_Y$) and **V** (of order $N_\theta \times n$) from the left and the right, respectively, is considered. When $n = 1$, **GV** is simply the product of **G** with a vector, whereas, when $m = 1$, $(\mathbf{WG})^T = \mathbf{G}^T\mathbf{W}^T$ gives the product of $\mathbf{G}^T$ with the (column) vector $\mathbf{W}^T$. Those examples show how, by defining algorithms to calculate **GV** and **WG** for arbitrary **V** and **W**, different types of derivative information can be accommodated in a single framework.

4.2.1. DIRECT METHOD

The derivation starts by considering the calculation of **GV** for a given $N_\theta \times n$ matrix **V**. From Eq. (4.21), by defining

$$\mathbf{Z} = \left[(\mathbf{RAP})^{-1}\mathbf{R}\left(\frac{\partial \mathbf{A}}{\partial \theta}\mathbf{P} + \mathbf{A}\frac{\partial \mathbf{P}}{\partial \theta}\right)\breve{\mathbf{x}}\right]\mathbf{V}, \tag{4.22}$$

the product **GV** can be rewritten as

$$\mathbf{GV} = \left(\frac{\partial \mathbf{h}}{\partial \breve{\mathbf{x}}} + \frac{\partial \mathbf{h}}{\partial \mathbf{x}'}\mathbf{P}\right)\mathbf{Z} - \left(\frac{\partial \mathbf{h}}{\partial \mathbf{x}'}\frac{\partial \mathbf{P}}{\partial \theta}\breve{\mathbf{x}}\right)\mathbf{V} + \frac{\partial \mathbf{h}}{\partial \theta}\mathbf{V}. \tag{4.23}$$

Here, **Z** is obtained as the solution to the following linear system of equations

$$(\mathbf{RAP})\,\mathbf{Z} = \left[\mathbf{R}\left(\frac{\partial \mathbf{A}}{\partial \theta}\mathbf{P}\breve{\mathbf{x}} + \mathbf{A}\frac{\partial \mathbf{P}}{\partial \theta}\breve{\mathbf{x}}\right)\right]\mathbf{V}. \tag{4.24}$$

This is known as the *Forward Method* [44], *Gradient Simulator* [32], or *Direct Method* [30]. Note that **Z** has dimensions of $N_C \times n$ and, therefore, it requires n linear systems to be solved. Hence, the cost of computing **GV** is proportional to the number of columns in **V**. In particular, to obtain the full sensitivity matrix **G** with the Direct Method, in which case one has to set **V** equal to the identity matrix of order N_θ, the cost will be proportional to the number of parameters.

The algorithm to compute the product of the sensitivity matrix with a matrix via the Direct Method is depicted in Algorithm 5.

In Algorithm 5, the notation '., j'represents the j-th column of a matrix. Algorithm 5 requires the computation of the product of $\partial\mathbf{P}/\partial\theta\breve{\mathbf{x}}$ by a column vector. This is discussed in the next section when Algorithm 7 is presented.

Algorithm 5: Right multiplying the sensitivity matrix by an arbitrary matrix via the Direct Method.

Input : $\mathbf{R}$, $\mathbf{A}$, $\mathbf{P}$, $\frac{\partial \mathbf{A}}{\partial \theta}$, $\check{\mathbf{x}}$, $\frac{\partial \mathbf{h}}{\partial \check{\mathbf{x}}}$, $\frac{\partial \mathbf{h}}{\partial \mathbf{x}'}$, $\frac{\partial \mathbf{h}}{\partial \theta}$, $\mathbf{V}$
Output: $\mathbf{GV}$

Compute $\boldsymbol{\alpha} = \left(\frac{\partial \mathbf{h}}{\partial \check{\mathbf{x}}} + \frac{\partial \mathbf{h}}{\partial \mathbf{x}'}\mathbf{P}\right)$ and $\boldsymbol{\beta} = \frac{\partial \mathbf{A}}{\partial \theta}\mathbf{P}\check{\mathbf{x}}$
foreach $j = 1, 2, ..., n$ **do**
  Compute $\boldsymbol{\gamma} = \left(\frac{\partial \mathbf{P}}{\partial \theta}\check{\mathbf{x}}\right)\mathbf{V}_{.,j}$; `// Algorithm 7, where` $\mathbf{m} = \mathbf{V}_{.,j}$
  Compute $\boldsymbol{\delta} = \mathbf{R}\left(\boldsymbol{\beta}\mathbf{V}_{.,j} + \mathbf{A}\boldsymbol{\gamma}\right)$
  Solve $\mathbf{z} = (\mathbf{RAP})^{-1}\boldsymbol{\delta}$
  Compute $(\mathbf{GV})_{.,j} = \alpha\mathbf{z} - \frac{\partial \mathbf{h}}{\partial \mathbf{x}'}\gamma + \frac{\partial \mathbf{h}}{\partial \theta}\mathbf{V}_{.,j}$

4

4.2.2. ADJOINT METHOD

Next the calculation of $\mathbf{WG}$ for a given $m \times N_Y$ matrix $\mathbf{W}$ is considered. From Eq. (4.21), defining

$$\mathbf{Z}^T = \mathbf{W}\left(\frac{\partial \mathbf{h}}{\partial \check{\mathbf{x}}} + \frac{\partial \mathbf{h}}{\partial \mathbf{x}'}\mathbf{P}\right)(\mathbf{RAP})^{-1}, \tag{4.25}$$

one obtains

$$\mathbf{WG} = \mathbf{Z}^T\mathbf{R}\left(\frac{\partial \mathbf{A}}{\partial \theta}\mathbf{P} + \mathbf{A}\frac{\partial \mathbf{P}}{\partial \theta}\right)\check{\mathbf{x}} - \mathbf{W}\frac{\partial \mathbf{h}}{\partial \mathbf{x}'}\frac{\partial \mathbf{P}}{\partial \theta}\check{\mathbf{x}} + \mathbf{W}\frac{\partial \mathbf{h}}{\partial \theta}, \tag{4.26}$$

which can be rearranged as

$$\mathbf{WG} = \left(\mathbf{Z}^T\mathbf{R}\frac{\partial \mathbf{A}}{\partial \theta}\mathbf{P}\check{\mathbf{x}} + \left(\mathbf{Z}^T\mathbf{RA} - \mathbf{W}\frac{\partial \mathbf{h}}{\partial \mathbf{x}'}\right)\frac{\partial \mathbf{P}}{\partial \theta}\check{\mathbf{x}}\right) + \mathbf{W}\frac{\partial \mathbf{h}}{\partial \theta}. \tag{4.27}$$

Multiplying Eq. (4.25) by $\mathbf{RAP}$ from the right and transposing, $\mathbf{Z}$ is obtained as the solution to the following linear system of equations

$$(\mathbf{RAP})^T\mathbf{Z} = \left(\frac{\partial \mathbf{h}}{\partial \check{\mathbf{x}}} + \frac{\partial \mathbf{h}}{\partial \mathbf{x}'}\mathbf{P}\right)^T\mathbf{W}^T. \tag{4.28}$$

This is known in the literature as the *Adjoint* (or *Backward*) *Method* [34]. Note that now $\mathbf{Z}$ has dimensions of $N_C \times m$, hence it requires m linear systems to be solved. As such, the cost of computing $\mathbf{WG}$ is proportional to the number of rows in $\mathbf{W}$. In particular, to obtain the full sensitivity matrix $\mathbf{G}$ with the Adjoint Method, in which case one has to set $\mathbf{W}$ equal to the identity matrix of order N_Y, the cost will be proportional to the number of responses.

The algorithm to compute the product of the sensitivity matrix with a matrix from the left via the backward method is depicted in Algorithm 6.

Algorithm 6 requires a left multiplication of $\partial\mathbf{P}/\partial\theta\check{\mathbf{x}}$ by a row vector $\mathbf{m}^T$. This will be further discussed in the next section, where Algorithm 8 is presented.

4.2.3. REMARKS ABOUT THE FRAMEWORK

This paper presents a novel technique for computation of the gradients using multiscale methods. It entails some important features which are summarized in this section.

Algorithm 6: Left multiplying the sensitivity matrix by an arbitrary matrix via the Adjoint Method.

Input : $\mathbf{R}$, $\mathbf{A}$, $\mathbf{P}$, $\frac{\partial \mathbf{A}}{\partial \theta}$, $\check{\mathbf{x}}$, $\frac{\partial \mathbf{h}}{\partial \tilde{\mathbf{x}}}$, $\frac{\partial \mathbf{h}}{\partial \mathbf{x}'}$, $\frac{\partial \mathbf{h}}{\partial \theta}$, $\mathbf{W}$
Output: $\mathbf{WG}$

1 Compute $\boldsymbol{\alpha} = \left(\frac{\partial \mathbf{h}}{\partial \tilde{\mathbf{x}}} + \frac{\partial \mathbf{h}}{\partial \mathbf{x}'}\mathbf{P}\right)$ and $\boldsymbol{\beta} = \mathbf{R}\frac{\partial \mathbf{A}}{\partial \theta}\check{\mathbf{x}}$
2 **foreach** $i = 1, 2, ..., m$ **do**
3 Solve $\mathbf{z} = (\mathbf{RAP})^{-T}\boldsymbol{\alpha}\boldsymbol{W}_{i,\cdot}^{T}$
4 Compute $\mathbf{m}^T = \left(\mathbf{z}^T\mathbf{RA} - \mathbf{W}_{i,\cdot}\frac{\partial \mathbf{h}}{\partial \mathbf{x}'}\right)$
5 Compute $\boldsymbol{\gamma} = \mathbf{m}^T\left(\frac{\partial \mathbf{P}}{\partial \theta}\check{\mathbf{x}}\right)$; `// Algorithm 8`
6 Compute $(\mathbf{WG})_{i,\cdot} = \mathbf{z}^T\boldsymbol{\beta} + \boldsymbol{\gamma} + \mathbf{W}_{i,\cdot}\frac{\partial \mathbf{h}}{\partial \theta}$

4

First of all, the general formulation of the method accommodates both the Direct and Adjoint approaches. Note that the previously-developed multiscale gradient calculations were all applicable to Adjoint formulation only [39–41].

It is important to also note that the algebraic multiscale-gradient formulation, presented here, does not make any assumption with respect to neither the nature of the problem nor the specific optimization parameters. This flexible framework provides any gradient information which is required by the chosen optimization algorithm. Note that the previous methods have been developed for specific types of parameters (e.g., permeability in [39]). As such, they cannot be readily used to provide gradient information if the required parameters by the chosen optimization algorithm are not those the methods were developed for. For instance, the work presented in [39] and [40] addressed the computation of sensitivities where the parameters were specifically the grid-block permeability values. This specific parameter leads to the sensitivity of the coarse-scale quantities (transmissibility) being related to the fine-scale permeability via additional local adjoint problems for the basis functions. As such, the method presented in [39, 40] is not applicable if, e.g., well controls are being used as optimization parameters. On the other hand, the work presented in [41] develops an algorithm that allows the well controls being used. However, it does not account for the sensitivity calculations of the coarse-scale quantities with respect to the fine-scale parameters. Instead, only a coarse-scale adjoint problem was solved to compute the required gradient information.

The implicit differentiation strategy of the present method makes it more flexible, compared to those methods developed based on the Lagrange multiplier formulation (as in [39–41]). Note that the Lagrange multiplier formulations are developed based on an objective function, which requires to be pre-defined for each specific application [4, 30]. Instead, the presented implicit differentiation strategy [44, 45] is built on a more generic system, i.e., the sensitivity matrix. More importantly, the pre- and post-multiplication of this sensitivity matrix by arbitrary matrices provides a flexible way to conform it to the specific type of gradient information needed (without requiring the pre-computation of the full sensitivity matrix).

Finally, the multiscale-gradient formulation is developed through an algebraic for-

mulation, which is convenient to be implemented in the next-generation optimization frameworks, and also benefits all multilevel approaches that can be described through restriction and prolongation operations such as multigrid [47] and domain-decomposition [48, 49]methods.

4.3. PARTIAL DERIVATIVE OF MSFV PROLONGATION OPERATOR WITH RESPECT TO THE PARAMETERS

A particular aspect of the methodology is the computation of partial derivatives of the prolongation operator with respect to the parameters, i.e., $\partial \mathbf{P}/\partial\theta$ in Eq. (4.21), and operations with this tensor. This is particularly challenging because the computation of $\mathbf{P}$ might involve complex operations, e.g., the solution of linear systems in the case of MSFV methods.

In this work, the MSFV method is considered; extensions to other MS methodologies can be obtained along the same lines. In this case, $\mathbf{P}$ is the assembly of all basis functions obtained via the solution of local flow problems, i.e.,

$$-\nabla\cdot\left(\lambda\cdot\nabla\varphi\right)=0, \tag{4.29}$$

where λ is the mobility and φ is the basis function value. The dual-grid sub domains where the basis functions are computed are defined as follows. A primal coarse grid (on which the conservative coarse-scale system is constructed) and a dual coarse grid, which is obtained by connecting coarse nodes, are defined on a given fine-scale grid. A coarse node is a fine cell inside (typically at the centre of) each coarse cell. The basis functions are solved locally on these dual coarse cells. Such overlapping coarse and dual coarse grids are crucial for conservative solutions in MSFV. An illustration of the MSFV grids is provided in Figure 5.1. The prolongation operator can be expressed in terms of the basis functions corresponding to each coarse cell $j=1,\ldots,N_C$ as

$$\mathbf{P}=\begin{bmatrix}\boldsymbol{\varphi}_1 & \boldsymbol{\varphi}_2 & \cdots & \boldsymbol{\varphi}_{N_C-1} & \boldsymbol{\varphi}_{N_C}\end{bmatrix}, \tag{4.30}$$

where the basis function belonging to cell j, $\boldsymbol{\varphi}_j$, is at column j, being a vector of dimension N_F [24]. The construction of the basis functions $\boldsymbol{\varphi}_j$ is based on the Finite Volume (FV) discretization of Eq. (4.29) on the dual coarse grid cells. In order to compute the basis functions, reduced-dimensional problems are first solved at the edge cells to provide Dirichlet boundary conditions for internal cells. For the sake of clarity, let us assume a Cartesian two-dimensional solution domain (although extension to 3D is straightforward). More specifically, let Λ_j be the set of all edges emanating from coarse node j (in 2D, $\#\Lambda_j=4$). For each edge $e\in\Lambda_j$, let

$$\mathbf{A}^E_{e,j}\boldsymbol{\varphi}^E_{e,j}-\mathbf{b}^E_{e,j}=\mathbf{0} \tag{4.31}$$

be the reduced-dimension discrete form of Eq. (4.29) along the edge e, where $\boldsymbol{\varphi}^E_{e,j}$ is the basis function at that edge, and $\mathbf{b}^E_{e,j}$ is the RHS which results from specifying corner values of $\varphi^E_{e,j}=1$ at the vertex j and $\varphi^E_{e,j}=0$ at the opposite vertex of e. If Γ_j is the set

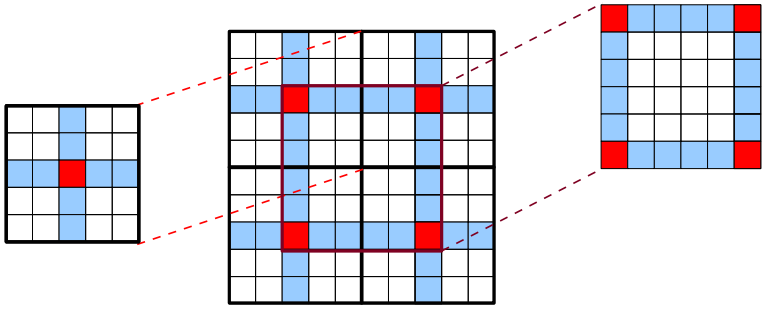


Figure 4.1: Illustration of MSFV coarse grids for a 2D domain. Given a fine-scale grid (shown in light solid black lines), the coarse grid (shown in solid bold black) is imposed as a non-overlapping partition of the computational domain. The coarse nodes (vertices) are then selected (red cells). Connecting coarse nodes constructs the dual-coarse grid (blue cells) where basis functions are solved.

of all faces which have coarse node j as a common vertex ($\#\Gamma_j = 4$ in 2D), for each face $f \in \Gamma_j$, according to Eq. (4.29), one can solve

$$\mathbf{A}^F_{f,j}\boldsymbol{\varphi}^F_{f,j} - \sum_{e\in\Lambda_j} \mathbf{E}^F_{e,f}\boldsymbol{\varphi}^E_{e,j} = \mathbf{0}, \tag{4.32}$$

for $\boldsymbol{\varphi}^F_{f,j}$, which is the basis function at the internal (face) cells. Also, $\mathbf{E}^F_{e,f}$ is a transformation matrix that appropriately assembles a vector represented in the edge topology of e into the face topology of f. Note that $\mathbf{E}^F_{e,f}$ is zero when e does not belong to the boundary of f and that Eq. (4.32) implicitly defines the boundary condition as zero for any boundary cells of f which do not belong to any of the edges in Λ_j. Finally, $\boldsymbol{\varphi}_j$ is the result of assembling the contribution of each $\boldsymbol{\varphi}^F_{f,j}$, $f \in \Gamma_j$, into the overall fine mesh topology:

$$\boldsymbol{\varphi}_j = \sum_{f\in\Gamma_j} \mathbf{E}_f\boldsymbol{\varphi}^F_{f,j}, \tag{4.33}$$

where $\mathbf{E}_f$ is a transformation matrix that assembles a vector represented in the face topology of f into a size N_F vector following the fine grid topology.

The derivative of $\boldsymbol{\varphi}_j$ w.r.t. the parameters is obtained from Eq. (4.33), i.e.,

$$\frac{\partial\boldsymbol{\varphi}_j}{\partial\theta} = \sum_{f\in\Gamma_j} \mathbf{E}_f\frac{\partial\boldsymbol{\varphi}^F_{f,j}}{\partial\theta}. \tag{4.34}$$

Differentiating Eq. (4.32), one can write

$$\frac{\partial\mathbf{A}^F_{f,j}}{\partial\theta}\boldsymbol{\varphi}^F_{f,j} + \mathbf{A}^F_{f,j}\frac{\partial\boldsymbol{\varphi}^F_{f,j}}{\partial\theta} - \sum_{e\in\Lambda_j} \mathbf{E}^F_{e,f}\frac{\partial\boldsymbol{\varphi}^E_{e,j}}{\partial\theta} = \mathbf{0}, \tag{4.35}$$

from which it follows that

$$\frac{\partial\boldsymbol{\varphi}^F_{f,j}}{\partial\theta} = \left(\mathbf{A}^F_{f,j}\right)^{-1}\left(-\frac{\partial\mathbf{A}^F_{f,j}}{\partial\theta}\boldsymbol{\varphi}^F_{f,j} + \sum_{e\in\Lambda_j} \mathbf{E}^F_{e,f}\frac{\partial\boldsymbol{\varphi}^E_{e,j}}{\partial\theta}\right). \tag{4.36}$$

Similarly, from Eq. (4.31),

$$\frac{\partial \boldsymbol{\varphi}_{e,j}^{E}}{\partial \theta} = -\left(\mathbf{A}_{e,j}^{E}\right)^{-1} \frac{\partial \mathbf{A}_{e,j}^{E}}{\partial \theta} \boldsymbol{\varphi}_{e,j}^{E} \tag{4.37}$$

holds. Combining Eq. (4.36) and Eq. (4.37) into Eq. (4.34) results in

$$\frac{\partial \boldsymbol{\varphi}_{j}}{\partial \theta} = \sum_{f \in \Gamma_j} \mathbf{E}_f \left(\mathbf{A}_{f,j}^{F}\right)^{-1} \left(-\frac{\partial \mathbf{A}_{f,j}^{F}}{\partial \theta} \boldsymbol{\varphi}_{f,j}^{F} - \sum_{e \in \Lambda_j} \mathbf{E}_{e,f}^{F} \left(\mathbf{A}_{e,j}^{E}\right)^{-1} \frac{\partial \mathbf{A}_{e,j}^{E}}{\partial \theta} \boldsymbol{\varphi}_{e,j}^{E} \right). \tag{4.38}$$

The third order tensor defining the derivative of $\mathbf{P}$ w.r.t. the parameters is obtained by grouping all $\partial \boldsymbol{\varphi}_j / \partial \theta$ at its slices,

$$\frac{\partial \mathbf{P}}{\partial \theta} = \left[\begin{array}{cccc} \frac{\partial \boldsymbol{\varphi}_1}{\partial \theta} & \frac{\partial \boldsymbol{\varphi}_2}{\partial \theta} & \cdots & \frac{\partial \boldsymbol{\varphi}_{N_C-1}}{\partial \theta} \quad \frac{\partial \boldsymbol{\varphi}_{N_C}}{\partial \theta} \end{array} \right]_{N_F \times N_C \times N_\theta}. \tag{4.39}$$

See Appendix A for a discussion on this notation. As discussed in Algorithm 5, the product $(\partial \mathbf{P}/\partial \theta \check{\mathbf{x}})\mathbf{m}$, where $\check{\mathbf{x}} \in \mathbb{R}^{N_C}$ and $\mathbf{m} \in \mathbb{R}^{N_\theta}$, is required. Since

$$\frac{\partial \mathbf{P}}{\partial \theta} \check{\mathbf{x}} = \sum_{j=1}^{N_C} \frac{\partial \boldsymbol{\varphi}_j}{\partial \theta} \check{\mathbf{x}}_j \tag{4.40}$$

where $\check{\mathbf{x}}_j$ denotes the j-th entry of vector $\check{\mathbf{x}}$, it follows, from Eq. (4.38), that

$$\begin{aligned} \left(\frac{\partial \mathbf{P}}{\partial \theta} \check{\mathbf{x}}\right)\mathbf{m} = \sum_{j=1}^{N_C} \check{\mathbf{x}}_j \sum_{f \in \Gamma_j} \mathbf{E}_f \left(\mathbf{A}_{f,j}^{F}\right)^{-1} \left(-\left(\frac{\partial \mathbf{A}_{f,j}^{F}}{\partial \theta} \boldsymbol{\varphi}_{f,j}^{F}\right)\mathbf{m} - \right. \\ \left. \sum_{e \in \Lambda_j} \mathbf{E}_{e,f}^{F} \left(\mathbf{A}_{e,j}^{E}\right)^{-1} \left(\frac{\partial \mathbf{A}_{e,j}^{E}}{\partial \theta} \boldsymbol{\varphi}_{e,j}^{E}\right)\mathbf{m} \right). \end{aligned} \tag{4.41}$$

Algorithm 7 depicts the solution procedure to calculate $(\partial \mathbf{P}/\partial \theta \check{\mathbf{x}})\mathbf{m}$ based on Eq. (4.41).

In a similar manner, Algorithm 6 requires the product $\mathbf{m}^T(\partial \mathbf{P}/\partial \theta \check{\mathbf{x}})$, where $\check{\mathbf{x}} \in \mathbb{R}^{N_C}$ and $\mathbf{m} \in \mathbb{R}^{N_F}$. From Eq. (4.38) and Eq. (4.40), one obtains

$$\begin{aligned} \mathbf{m}^T \left(\frac{\partial \mathbf{P}}{\partial \theta} \check{\mathbf{x}}\right) = \sum_{j=1}^{N_C} \check{\mathbf{x}}_j \sum_{f \in \Gamma_j} \mathbf{m}^T \mathbf{E}_f \left(\mathbf{A}_{f,j}^{F}\right)^{-1} \left(-\frac{\partial \mathbf{A}_{f,j}^{F}}{\partial \theta} \varphi_{f,j}^{F} - \right. \\ \left. \sum_{e \in \Lambda_j} \mathbf{E}_{e,f}^{F} \left(\mathbf{A}_{e,j}^{E}\right)^{-1} \left(\frac{\partial \mathbf{A}_{e,j}^{E}}{\partial \theta} \varphi_{e,j}^{E}\right) \right). \end{aligned} \tag{4.42}$$

By defining

$$\boldsymbol{\gamma}_{f,j}^{T} = \mathbf{m}^T \mathbf{E}_f \left(\mathbf{A}_{f,j}^{F}\right)^{-1} \tag{4.43}$$

and

$$\boldsymbol{\delta}_{e,f,j}^{T} = \gamma^T \mathbf{E}_{e,f}^{F} \left(\mathbf{A}_{e,j}^{E}\right)^{-1}, \tag{4.44}$$

Algorithm 7: Computation of the product $(\partial \mathbf{P}/\partial\theta\check{\mathbf{x}})\mathbf{m}$, where $\check{\mathbf{x}} \in \mathbb{R}^{N_C}$ and $\mathbf{m} \in \mathbb{R}^{N_\theta}$.

Input : $\check{\mathbf{x}}, \mathbf{m}$, $\boldsymbol{\varphi}^E_{e,j}$, $\mathbf{A}^E_{e,j}$, $\frac{\partial \mathbf{A}^E_{e,j}}{\partial\theta}$, $j = 1,...,N_C$, $e \in \Lambda_j$, $\boldsymbol{\varphi}^F_{f,j}$, $\mathbf{A}^F_{f,j}$, $\frac{\partial \mathbf{A}^F_{f,j}}{\partial\theta}$, $j = 1,...,N_C$, $f \in \Gamma_j$

Output: $\left(\frac{\partial \mathbf{P}}{\partial\theta}\check{\mathbf{x}}\right)\mathbf{m}$

Set: $\left(\frac{\partial \mathbf{P}}{\partial\theta}\check{\mathbf{x}}\right)\mathbf{m} = \mathbf{0}$

foreach $j = 1,2,...,N_C$ *(primal coarse nodes)* **do**

  foreach $f \in \Gamma_j$ **do**

    Compute $\boldsymbol{\alpha} = -\left(\frac{\partial \mathbf{A}^F_{f,j}}{\partial\theta}\boldsymbol{\varphi}^F_{f,j}\right)\mathbf{m}$

    foreach $e \in \Lambda_j$ **do**

      Compute $\boldsymbol{\beta} = \left(\frac{\partial \mathbf{A}^E_{e,j}}{\partial\theta}\boldsymbol{\varphi}^E_{e,j}\right)\mathbf{m}$

      Solve local edge system $\boldsymbol{\gamma} = \left(\mathbf{A}^E_{e,j}\right)^{-1}\boldsymbol{\beta}$

      Accumulate result in $\boldsymbol{\alpha}$: $\boldsymbol{\alpha} = \boldsymbol{\alpha} - \mathbf{E}^F_{e,f}\boldsymbol{\gamma}$

    Solve local face system $\boldsymbol{\delta} = \left(\mathbf{A}^F_{f,j}\right)^{-1}\boldsymbol{\alpha}$

    Accumulate result in $\left(\frac{\partial \mathbf{P}}{\partial\theta}\check{\mathbf{x}}\right)\mathbf{m}$: $\left(\frac{\partial \mathbf{P}}{\partial\theta}\check{\mathbf{x}}\right)\mathbf{m} = \left(\frac{\partial \mathbf{P}}{\partial\theta}\check{\mathbf{x}}\right)\mathbf{m} + \check{\mathbf{x}}_j\mathbf{E}_f\boldsymbol{\delta}$

Eq. (4.42) can be rewritten as

$$\mathbf{m}^T\left(\frac{\partial \mathbf{P}}{\partial\theta}\check{\mathbf{x}}\right) = \sum_{j=1}^{N_C}\check{\mathbf{x}}_j\sum_{f\in\Gamma_j}\left(-\boldsymbol{\gamma}^T_{f,j}\left(\frac{\partial \mathbf{A}^F_{f,j}}{\partial\theta}\boldsymbol{\varphi}^F_{f,j}\right) - \sum_{e\in\Lambda_j}\boldsymbol{\delta}^T_{e,f,j}\left(\frac{\partial \mathbf{A}^E_{e,j}}{\partial\theta}\varphi^E_{e,j}\right)\right). \tag{4.45}$$

Right multiplying Eq. (4.43) by $\mathbf{A}^F_{f,j}$ and transposing, one can see that $\boldsymbol{\gamma}_{f,j}$ is obtained as

$$\left(\mathbf{A}^F_{f,j}\right)^T\boldsymbol{\gamma}_{f,j} = \mathbf{E}^T_f\mathbf{m}. \tag{4.46}$$

Note that $\mathbf{E}^T_f$ is the transformation matrix that assembles a vector following the fine grid topology into its proper restriction to the face topology of f. Analogously, $\boldsymbol{\delta}_{e,f,j}$ is obtained by solving

$$\left(\mathbf{A}^E_{e,j}\right)^T\boldsymbol{\delta}_{e,f,j} = \left(\mathbf{E}^F_{e,f}\right)^T\boldsymbol{\gamma}. \tag{4.47}$$

Similar as for $\mathbf{E}^T_f$, $(\mathbf{E}^F_{e,f})^T$ is the transformation matrix that assembles a vector following the face topology of f into its restriction to the edge topology of e. Equations Eq. (4.45), Eq. (4.46) and Eq. (4.47) are the basis for the algorithm to calculate $\mathbf{m}^T(\partial\mathbf{P}/\partial\theta\check{\mathbf{x}})$, as depicted in Algorithm 4. Note that the face systems are solved before the edge ones, in contrast to the order of the MS calculation.

Algorithm 8: Computation of the product $\mathbf{m}^T(\partial\mathbf{P}/\partial\theta\check{\mathbf{x}})$, where $\check{\mathbf{x}} \in \mathbb{R}^{N_C}$ and $\mathbf{m} \in \mathbb{R}^{N_F}$.

Input : $\check{\mathbf{x}}$, $\mathbf{m}$, $\varphi^E_{e,j}$, $\mathbf{A}^E_{e,j}$, $\frac{\partial \mathbf{A}^E_{e,j}}{\partial\theta}$, $j = 1,...,N_C$, $e \in \Lambda_j$, $\varphi^F_{f,j}$, $\mathbf{A}^F_{f,j}$, $\frac{\partial \mathbf{A}^F_{f,j}}{\partial\theta}$, $j = 1,...,N_C$, $f \in \Gamma_j$

Output: $\mathbf{m}^T\left(\frac{\partial\mathbf{P}}{\partial\theta}\check{\mathbf{x}}\right)$

Set: $\mathbf{m}^T\left(\frac{\partial\mathbf{P}}{\partial\theta}\check{\mathbf{x}}\right) = 0$
foreach $j = 1,2,...,N_C$ **do**
  foreach $f \in \Gamma_j$ **do**
    Solve transpose local face system $\gamma = \left(\mathbf{A}^F_{f,j}\right)^{-T}\mathbf{E}^T_f\mathbf{m}$
    Compute $\alpha = -\gamma^T\left(\frac{\partial \mathbf{A}^F_{f,i}}{\partial\theta}\varphi^F_{f,i}\right)$
    foreach $e \in \Lambda_j$ **do**
      Solve transpose local edge system $\delta = \left(\mathbf{A}^E_{e,j}\right)^{-T}\left(\mathbf{E}^F_{e,f}\right)^T\gamma$
      Accumulate result in α : $\alpha = \alpha - \delta^T\left(\frac{\partial \mathbf{A}^E_{e,j}}{\partial\theta}\varphi^E_{e,j}\right)$
    Accumulate result in $\mathbf{m}^T\left(\frac{\partial\mathbf{P}}{\partial\theta}\check{\mathbf{x}}\right)$: $\mathbf{m}^T\left(\frac{\partial\mathbf{P}}{\partial\theta}\check{\mathbf{x}}\right) = \mathbf{m}^T\left(\frac{\partial\mathbf{P}}{\partial\theta}\check{\mathbf{x}}\right) + \check{\mathbf{x}}_j\alpha$

4.3.1. PROLONGATION AND ITS DERIVATIVE IN THE PRESENCE OF WELLS

Wells and other fine-scale source terms are considered in the multiscale formulation by supplementary well basis functions [19, 22]. Well functions are local solutions to Eq. (4.29) subject to Dirichlet conditions of 1 at the well cell and 0 at the coarse nodes. Hence, considering well functions, the prolongation operator (for porous rock) is enriched and reads

$$\mathbf{P} = \left[\begin{array}{ccc|ccc} \varphi_1 & \cdots & \varphi_{N_C} & \psi_1 & \cdots & \psi_{N_W} \end{array}\right], \tag{4.48}$$

where each well function ψ_w adds a column vector to the porous rock prolongation operator. Note that there are N_W wells in the domain. For rate-constrained wells, one has to add additional rows to the prolongation (and consequently, the coarse-scale system) in order to solve for the well pressures as additional unknowns [19]. The derivative of Eq. (4.48) w.r.t. the parameters is given by

$$\frac{\partial\mathbf{P}}{\partial\theta} = \left[\begin{array}{ccc|ccc} \frac{\partial\varphi_1}{\partial\theta} & \cdots & \frac{\partial\varphi_{N_C}}{\partial\theta} & \frac{\partial\psi_1}{\partial\theta} & \cdots & \frac{\partial\psi_{N_W}}{\partial\theta} \end{array}\right]. \tag{4.49}$$

Computation of the partial derivatives of the well basis functions follows similar strategy as discussed for Eq. (4.38).

4.4. COMPUTATIONAL ASPECTS OF THE MS-GRADIENT METHOD

4.4.1. PARTIAL DERIVATIVE COMPUTATION AND AUTOMATIC DIFFERENTIATION

In the computation of analytical gradients, a significant part of the overall implementation effort is associated with the computation of partial derivative matrices. This computation step often requires access to the source code because not all partial derivatives are readily available, as opposed to the state variable derivatives via the Jacobian matrix (see, e.g., inputs required by Algorithm 5 and Algorithm 6). To calculate the partial derivative matrices an Automatic Differentiation (AD) approach is used, because of its flexibility and accuracy. A review of different AD approaches is provided by [35]. In our work, an Operator Overloading AD technique (Bendtsen and Stauning 1996) based on Expression Templates [50] is applied. This facilitates the computation and assemblage of all the partial derivative matrices because they are obtained at the same time as when the system matrix and the simulator response are computed.

4.4.2. COMPUTATIONAL EFFICIENCY

An asymptotic analysis is performed to assess the efficiency of the MS-gradient information, compared with the fine-scale gradient computation. Note that the computational cost of linear system assembly and matrix-vector products are negligible when compared to the cost involved in solving the linear system of equations. The complexity of solving a linear system of size N is assumed to be $\mathcal{O}(aN^b)$, where a and b are constants associated with the specific linear solver employed.

Following Algorithm 5, n (number of columns in the $\mathbf{V}$ matrix) linear systems of size N_C must be solved to fully define the gradient information, hence its computational complexity reads $\mathcal{O}(n(a_{MS}{N_C}^{b_{MS}}))$. Additionally, Algorithm 7 consists of solving $N_L \times n \times N_C$ linear systems of size $N_R = N_F/N_C$ to provide the partial derivative of $\mathbf{P}$ w.r.t. the parameters. Here, N_R is the multiscale coarsening ratio, and N_L is the number of local problems that must be solved per coarse grid vertex (4 in 2D and 8 in 3D problems). Thus, the computational complexity of Algorithm 7 is $\mathcal{O}(N_L n N_C(a_{MS}{N_R}^{b_{MS}}))$, which results in a complexity of $\mathcal{O}^D_{MS}(n(a_{MS}{N_C}^{b_{MS}} + N_L N_C(a_{MS}{N_R}^{b_{MS}})))$ for the MS Direct Method.

A similar analysis can be performed for Algorithm 6 and Algorithm 8, resulting in an estimate of the complexity of the MS Adjoint Method, i.e., $\mathcal{O}^A_{MS}(m(a_{MS}{N_C}^{b_{MS}}+N_L N_C(a_{MS}{N_R}^{b_{MS}})))$, where m is the number of columns of $\mathbf{W}$.

The complexity of the fine-scale gradient computation for the Direct and Adjoint Methods is given by $\mathcal{O}(n(a_{FS}{N_C}^{b_{FS}}))$ and $\mathcal{O}(m(a_{FS}{N_C}^{b_{FS}}))$, respectively. Therefore, the cost ratio between the MS and fine-scale computational efficiency can be defined as

$$\frac{\mathcal{O}_{MS}}{\mathcal{O}_{FS}} = \frac{a_{MS}}{a_{FS}}\left(\frac{{N_C}^{b_{MS}}}{{N_F}^{b_{FS}}} + N_L\frac{{N_F}^{b_{MS}}}{{N_F}^{b_{FS}}}{N_C}^{(1-b_{MS})}\right), \tag{4.50}$$

which holds for both the Direct and the Adjoint Method.

For the sake of simplicity, it is assumed that the solver employed to the MS system is equally efficient to the one employed to the fine-scale system, i.e., $a_{MS} = a_{FS}$ and

$b_{MS} = b_{FS} = b$. As such, the expression defining the cost ratio simplifies to

$$\frac{\mathcal{O}_{MS}}{\mathcal{O}_{FS}} = \frac{1}{N_R^b} + N_L \frac{N_R^{b-1}}{N_F^{b-1}}. \tag{4.51}$$

For a fixed number of fine grid cells N_F, it is mainly the coarsening ratio N_R that determines the complexity of the MS-gradient algorithm. To illustrate this point, a domain with $N_F = 10^7$ fine-scale grid cells is considered. The MS-gradient speed-up, $\mathcal{O}_{MS}/\mathcal{O}_{FS}$, for different coarsening ratios N_R for both 2D and 3D cases is presented in Figure 4.2, where $b = 1.3$.

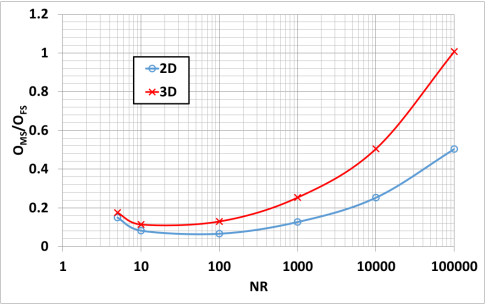


Figure 4.2: Cost ratio between MS and fine-scale gradient computation methods, as a function of the multiscale coarsening ratio, N_R, for a 2D (blue) and 3D (red) domain with $N_F = 10^7$ fine-scale cells.

4.5. NUMERICAL EXPERIMENTS

In this section, performance of the MS-gradient method is studied for single-phase incompressible flow in heterogeneous porous media. The following numerical experiments are presented to first validate and then assess the accuracy of the gradient information computed by the method. For this purpose, a misfit objective function with no regularization term

$$O(\theta) = \frac{1}{2}(\mathbf{h}(\mathbf{x},\theta) - \mathbf{d}_{obs})^T \mathbf{C}_D^{-1} (\mathbf{h}(\mathbf{x},\theta) - \mathbf{d}_{obs}), \tag{4.52}$$

with a gradient

$$\nabla_\theta O = \mathbf{G}^T \mathbf{C}_D^{-1} (\mathbf{h}(\mathbf{x},\theta) - \mathbf{d}_{obs}), \tag{4.53}$$

is considered [30]. In all experiments, the fitting parameters are cell-centered permeabilities. The observed quantity, $\mathbf{d}_{obs}$, is the fine scale pressure at the location of (non-flowing) observation wells, therefore

$$\frac{\partial \mathbf{h}}{\partial \mathbf{x}'} = \mathbf{I}, \tag{4.54}$$

and

$$\frac{\partial \mathbf{h}}{\partial \check{\mathbf{x}}} = \mathbf{0}. \tag{4.55}$$

The tensor $\partial \mathbf{A}/\partial\theta$ represents partial derivatives of the system (transmissibility) matrix with respect to permeability. Note that the results are expressed in terms of a non-dimensional pressure, i.e.,

$$p_D = \frac{p - p_{prod}}{p_{inj} - p_{prod}}, \tag{4.56}$$

where p_{inj} and p_{prod} are the injection and production pressures, respectively. In all the experiments, $p_{inj} = 1.0$ and $p_{prod} = 0.0$, the grid-block dimensions are $\Delta x = \Delta y = \Delta z = 1$ m and the fluid viscosity is 1.0×10^{-3} Pa s. In addition, in all the following test cases, well basis functions are included.

4.5.1. VALIDATION EXPERIMENTS

The MS-gradient method is validated against the numerical differentiation method with a higher-order, two-sided Taylor approximation

$$\begin{aligned}\nabla_\theta O_i = \frac{1}{2\delta\theta_i}\big(&O\left(\theta_i,\ldots,\theta_{i-1},\theta_i+\delta\theta_i,\theta_{i+i},\ldots,\theta_{N_\theta}\right) - \\ &O\left(\theta_i,\ldots,\theta_{i-1},\theta_i-\delta\theta_i,\theta_{i+i},\ldots,\theta_{N_\theta}\right)\big),\end{aligned} \tag{4.57}$$

where δ is a multiplicative parameter perturbation. The relative error can be defined as

$$\varepsilon = \frac{\|\nabla_\theta O_{FD} - \nabla_\theta O_{AN}\|_2}{\|\nabla_\theta O_{AN}\|_2}, \tag{4.58}$$

where $\nabla_\theta O_{FD}$ is obtained by performing the appropriate number of multiscale reservoir simulations required to evaluate Eq. (5.65) and $\nabla_\theta O_{AN}$ is obtained by either employing the Direct or the Adjoint Method to evaluate Eq. (6.49). Note that

$$\mathbf{m} = \mathbf{C}_D^{-1}\left(\mathbf{h}\left(\mathbf{x},\theta\right) - \mathbf{d}_{obs}\right), \tag{4.59}$$

where $\mathbf{m}$ is an auxiliary vector, so the gradient of O can be written as $\nabla_\theta O = (\mathbf{m}^T\mathbf{G})^T$ and Algorithm 5, with $\mathbf{W} = \mathbf{m}^T$, calculates $\nabla_\theta O$ with a cost proportional to one extra MS simulation, making the Adjoint Method a much more efficient way to calculate the gradient than the Direct Method. Moreover, for all cases, unless stated otherwise, for simplicity, it is assumed $\mathbf{C}_D = \mathbf{I}$.

In order to validate the proposed derivative calculation methods, as well as their implementation, the linear decrease of the error ε by decreasing the perturbation value δ (see chapter 8 of [31]) from 10^{-1} to 10^{-4} (the range within which only discretization errors are observed) is investigated.

The investigation is carried out in three examples of increasing complexity. The first case is a one-dimensional, homogeneous medium with 45 grid blocks. A primal coarse grid of just 3 grid blocks is employed (coarsening ratio of 15). Injection and production wells are located at, respectively, cells 1 and 45. One observation well is located at the centre of the domain, i.e. at grid block 23. The pressure measured in the observation well is taken from a reference case with a randomly sampled permeability field. The non-dimensional injection and production pressures are one and zero, respectively. Figure 4.3 illustrates the setup for this experiment.

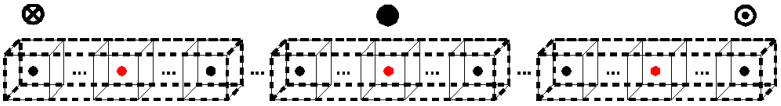
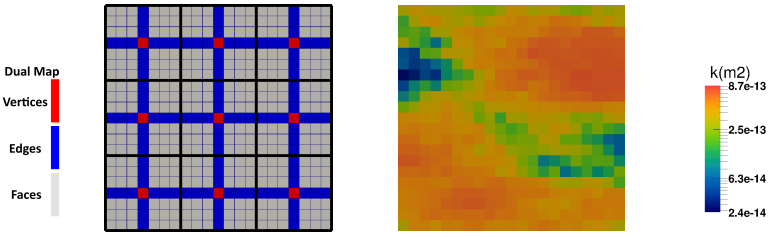


Figure 4.3: Fine and coarse grids and wells setup for 1D numerical experiments. The solid thin lines represent the fine grid-blocks. The bold dashed lines represent the primal-coarse grid blocks. Vertex cells identified with red circles. The crossed circle represents the injection well, the dotted circle the production well, and the solid circle the observation well.

In the other two experiments, the accuracy of the method is assessed for 2D test cases, one homogeneous and another one heterogeneous. In both cases the fine grid size is 21x21, while the coarse grid size is 3x3 (coarsening ratio of 7x7). The primal- and dual-coarse grids are illustrated in Figure 6.1a.



(a) MS grids (b) Reference permeability

Figure 4.4: (a) Fine, primal and dual coarse grids for 2D validation test cases and (b) reference permeability field.

The permeability field is extracted from 1,000 geological realizations, as shown in Figure 6.1b, and serves to compute the observed pressure. For the heterogeneous case, another geological realization is chosen from the ensemble. The ensemble is generated via the decomposition of a reference permeability "image" using Principal Component Analysis parameterization. Figure A.8 illustrates 4 different permeability realizations from the ensemble. See [51] for more details.

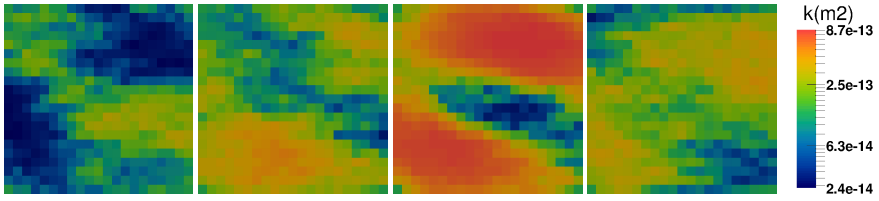


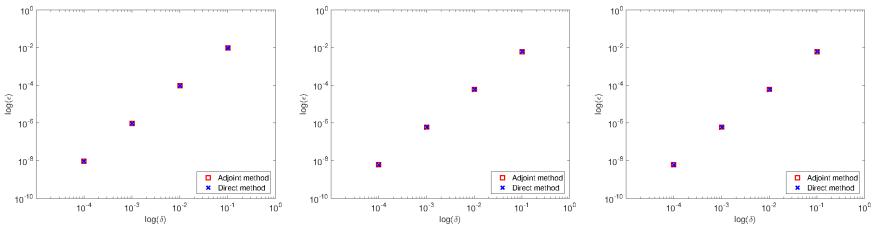
Figure 4.5: Four different permeability realizations from the ensemble of 1,000 members used in the 2D numerical experiments.

A quarter five-spot well configuration is considered, with two observation wells close to the operating wells. The well positions and operating pressures are described in Table 4.1.

Table 4.1: Well configuration for the homogeneous, two-dimensional case.

Well	Fine scale position (I, J)	Well type
INJE	(1, 1)	Injection
PROD	(21,21)	Production
OBSWELL1	(3, 3)	Observation
OBSWELL2	(19, 19)	Observation

Figure 4.6 shows that the MS-gradient and fine-scale methods have the same order of accuracy with respect to the perturbation δ, for all cases. The expected behaviour of linearly decreasing error values as the perturbation size decreases is observed in all experiments. One can also note that the Direct and Adjoint Methods are equally accurate, i.e., they both provide the analytical gradient at the same accuracy. Also, it is important to notice that the localization assumptions involved in 2D are consistently represented by the analytical methods. The result of the third experiment (Figure 4.6c) indicates the correctness of the method when applied to compute gradients for heterogeneous media.



(a) 1D, homogeneous (b) 2D, homogeneous (c) 2D, heterogeneous

Figure 4.6: Validation of MS gradient computation method via comparison with numerical differentiation. (a) One-dimensional, homogeneous, (b) two-dimensional homogenous and (c) two-dimensional heterogeneous steady-state flow test cases.

4.5.2. GRADIENT ACCURACY

In order to assess the quality of the MS gradient, the angle between fine-scale and MS normalized gradients, i.e.,

$$\alpha = \cos^{-1}\left(\nabla_\theta^T \hat{O}_{FS}\, \nabla_\theta \hat{O}_{MS}\right), \tag{4.60}$$

is measured. Here,

$$\nabla_\theta \hat{O}_{FS} = \frac{\nabla_\theta O_{FS}}{\|\nabla_\theta O_{FS}\|_2} \tag{4.61}$$

and

$$\nabla_\theta \hat{O}_{MS} = \frac{\nabla_\theta O_{MS}}{\|\nabla_\theta O_{MS}\|_2}. \tag{4.62}$$

Also, $\nabla_\theta O_{FS}$ and $\nabla_\theta O_{MS}$ denote the fine-scale and MS analytical gradients, respectively. As a minimum requirement, acceptable MS gradients are obtained if α is much smaller than 90^o [52].

Firstly, for the 1D test case, both methods result in $\alpha = 0^o$, indicating that fine scale and MS gradients are perfectly aligned in this case. This is due to the fact that, in 1D, no approximations (due to localization) are made in the MS solution, and thus, in the MS gradient computation.

For the 2D homogenous case, the fine scale and MS gradients result in $\alpha = 10.97^o$, using the same setup depicted in Figure 6.1a. Although the gradients are practically pointing in the same direction, a deviation between the two is observed. To better investigate this difference, Figure 4.7a and Figure 4.7b present the fine-scale and MS pressure solutions, respectively, and Figure 4.7c shows the difference between these two pressure solutions. Figure 4.7d and Figure 4.7e present the absolute, normalized gradient of the OF computed via fine-scale and MS gradient methods, respectively. Figure 4.7f shows the difference between the absolute, normalized gradient computed by the two methods.

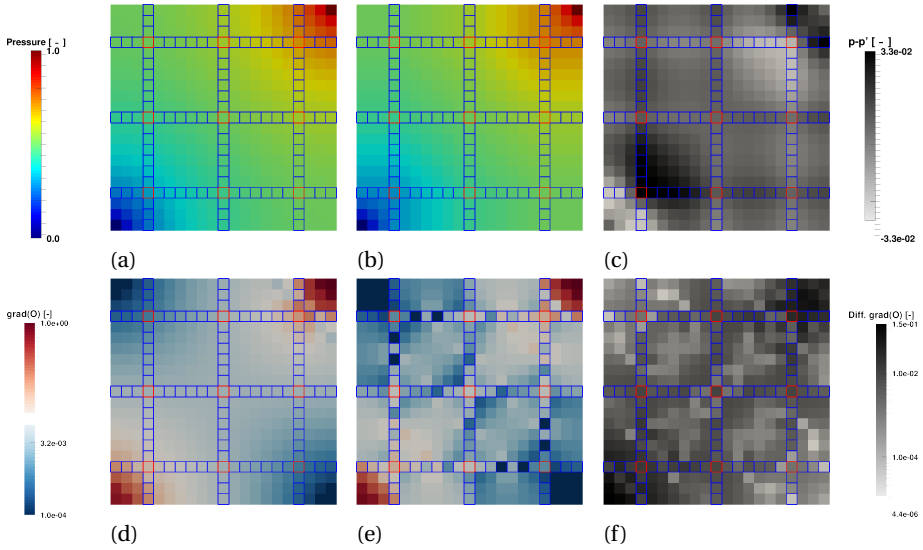


Figure 4.7: Fine scale (p) pressure solution (a) and MSFV (p') pressure solution (b). Difference between fine-scale and MSFV pressure solutions (c). Absolute, normalized fine scale gradient (d) and MSFV (e). Difference between fine-scale and MS absolute, normalized gradients (f). In all figures the dual-coarse grid is shown. Also shown, in the red boxes, are the coarse nodes.

The difference between the MS and fine-scale methods is due to the localization assumption in the calculation of the basis functions [16]. Note that, thanks to the well functions, the multiscale solutions are accurately capturing the wells.

4.5.3. EFFECT OF HETEROGENEITY DISTRIBUTION AND COARSENING RATIO

In the first case, OF gradients are computed for the whole ensemble of heterogeneous permeability fields (see Figure 4.4). Moreover, the same realization depicted in Figure 6.1b is considered as the reference from which the observed pressures are computed.

Table 4.2: Well configuration for the homogeneous, two-dimensional case.

Well	Fine scale position (I, J)	Well type
INJE	(11, 11)	Injection
PROD1	(1, 1)	Production
PROD2	(21, 1)	Production
PROD3	(1, 21)	Production
PROD4	(21, 21)	Production
OBSWELL1	(3, 3)	Observation
OBSWELL2	(19, 3)	Observation
OBSWELL3	(3, 19)	Observation
OBSWELL4	(19, 19)	Observation

An inverted five-spot well pattern is employed, whiled four observation wells are placed close to the production wells. The well configuration is depicted in Table 4.2.

Two different coarsening ratios are applied, one resulting in a coarse grid of 3x3 (as illustrated in Figure 6.1a) and another one resulting in a coarse grid of 7x7. The angles between the fine scale gradient and the MS gradient for each realization are computed, using Eq. (4.60). A histogram illustrating the angle distribution for each coarsening ratio is presented in Figure 4.8.

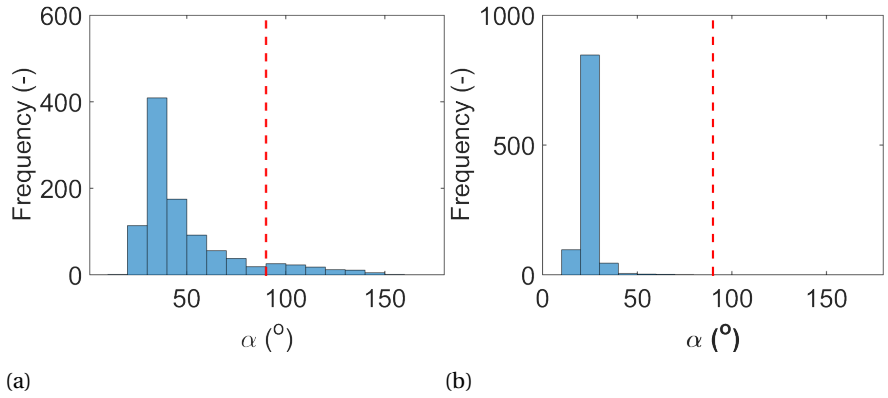
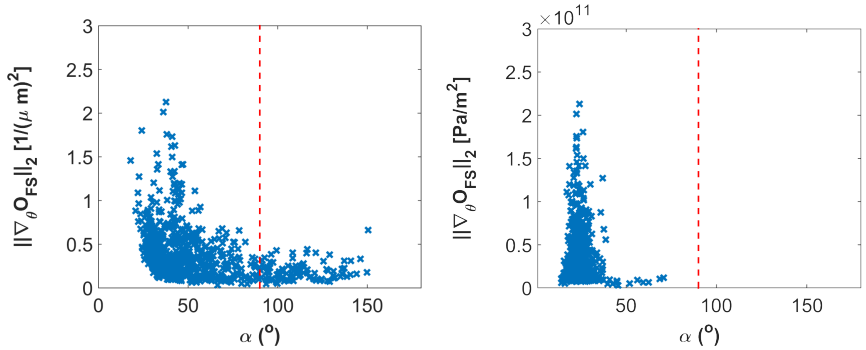


Figure 4.8: Histograms of angles between fine scale and multiscale gradients for 3x3 (a) and 7x7 (b) coarse grids. The fine-scale computational domain contains 21x21 grid cells. The vertical red dashed line indicates the 90^o limit.

Note that, for this case, the quality of the gradients is significantly improved once the coarse grid size of 3x3 is increased to 7x7. Important to note is that, as shown in Figure 4.9, the MS gradients are least accurate when the norm of the gradient vector is small. Therefore, they are not expected to have a major effect on the optimization procedure.

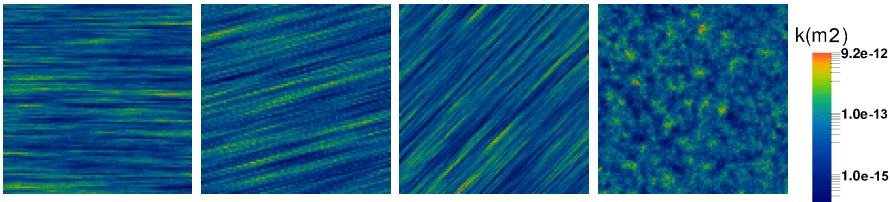
For the 3x3 coarse grid case, 9.71% of the angles are greater than 90^o, indicating that some MS gradients point in the opposite direction of the decreasing OF direction. On the other hand, if a 7x7 coarse grid is used no angle is greater than 90^o.



(a) (b)

Figure 4.9: Cross-plots of fine scale gradient norm vs. α for the 1000 realizations in the heterogeneous ensemble for (a) 3x3 and (b) 7x7 coarse grid. Note that MS gradients are accurate for the cases with large values of gradient norms. Their relatively inaccurate estimates (large α) happen mostly when the norms of gradients are small. The vertical red dashed line indicates the 90^o limit.

In order to further explore this point, four sets of 20 equiprobable realizations of log-normally distributed permeability fields with a spherical variogram and dimensionless correlation lengths of $\Psi_1 = 0.5$ and $\Psi_2 = 0.02$ are generated using sequential Gaussian simulations [53]. For each set, the variance and the mean of $\ln(k)$ are 2.0 and 3.0, respectively, where k is the grid block permeability. As depicted in Figure 6.5, for the realizations with a long correlation length, the angles between the permeability layers and the horizontal axis are 0^o, 15^o, and 45^o. A patchy (small correlation length) pattern is also considered (Figure 6.5d). Compared with the previous set, the permeability contrast is much higher in this case.



(a) 0^o (b) 15^o (c) 45^o (d) Patchy

Figure 4.10: Permeability distribution of four different realizations taken from the sets of 20 geostatistically equiprobable permeability fields with 0^o (a), 15^o (b), and 45^o (c) correlation angles. Also, a patchy field (d) with a small correlation length is considered.

The fine-scale and coarse grids contain 100 x 100 and 20 x 20 cells, respectively. The well configuration utilized in this numerical experiment is depicted in Table 6.3.

From Figure 4.11, one can observe that all cases with $\alpha > 50^o$ are associated with small gradient norms and that, for all geological sets, the MS gradient provides gradient directions that are very accurate, compared with the fine scale solution, when the gradient norms are large.

Table 4.3: Well configuration for Case 2.

Well	Fine scale position (I, J)	Well type
INJE	(1, 1)	Injection
PROD	(100, 100)	Production
OBSWELL1	(3, 3)	Observation
OBSWELL2	(98, 98)	Observation

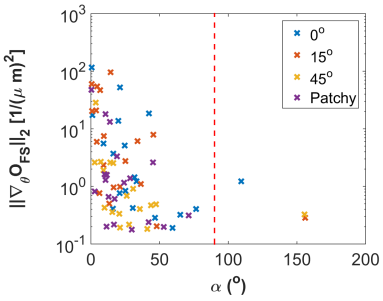


Figure 4.11: Cross-plot of fine scale gradient norm (log scale) vs. α for the four sets of 20 equiprobable permeability realizations. The vertical red dashed line indicates the 90^o limit.

Note also that, in the case of a small gradient norm, the OF has a weak dependency on the parameters in the vicinity of the point where the gradient is being calculated. As such, the overall performance of the optimization algorithm is not expected to be affected by replacing the fine-scale gradient calculation with the MS version. For practical purposes, as will be shown by the next numerical experiment, acceptable results in optimization studies can be achieved by using approximated multiscale gradients. In general, an iterative multiscale strategy [16] should be used in order to guarantee the quality of the multiscale gradient. The key component of such a development is to relate the quality of the approximate parameters (often measured via residuals) with the quality of the gradients. The employment of an iterative scheme imposes challenges associated with, for instance, the computation of extra partial derivative information that arises from the smoothing step. These challenges fall beyond the scope of this paper.

4.5.4. PARAMETER ESTIMATION STUDY

In order to investigate how the approximate MS gradient performs in an optimization algorithm, a parameter estimation study is performed. In this study, the same permeability ensemble as illustrated in Figure A.8 is used. The permeability field employed to create the synthetic data is illustrated in Figure 6.1b. The initial guess is randomly chosen from the ensemble. The MS method employs 7x7 coarse grid cells. The well configuration is presented in 4.1. Differently from the other experiments, a misfit objective

Table 4.4: Matched data for parameter estimation study utilizing gradients computed via fine scale Adjoint Method.

Well	Observed Pressure [-]	Initial Pressure [-]	Matched Pressure [-]	Percent Error [%]
OBSWELL1	0.2508	0.1693	0.2505	0.024
OBSWELL2	0.6918	1.8305	0.6920	0.012

function with a regularization term [30], i.e.

$$O(\mathbf{y},\theta) = \frac{1}{2}\left(\theta-\theta_{prior}\right)^T \mathbf{C}_\theta^{-1}\left(\theta-\theta_{prior}\right) + \frac{1}{2}\left(\mathbf{h}(\mathbf{x},\theta)-\mathbf{d}_{obs}\right)^T \mathbf{C}_D^{-1}\left(\mathbf{h}(\mathbf{x},\theta)-\mathbf{d}_{obs}\right), \tag{4.63}$$

is considered, where $\theta \in \mathbb{R}^{N_F}$ is the vector of parameters, taken to be the natural logarithm of the permeability in each grid cell. The covariance matrix $\mathbf{C}_\theta$ is computed from the ensemble of realizations as

$$\mathbf{C}_\theta = \frac{1}{N_e-1}\left(\theta-\mu\mathbf{e}^T\right)\left(\theta-\mu\mathbf{e}^T\right)^T \tag{4.64}$$

where Θ is the $N_F \times N_e$ matrix whose j-th column is given by the member of the ensemble θ_j, $j \in \{1,...,N_e\}$,

$$\mu = \frac{1}{N_e}\sum_{j=1}^{N_e}\theta_j \tag{4.65}$$

is the ensemble mean, and $\mathbf{e} = [\mathbf{1},...,\mathbf{1}]^T$ is a vector of ones of size $N_e \times 1$. In Eq. (4.63), the prior is taken to be the ensemble mean,

$$\theta_{prior} = \mu. \tag{4.66}$$

$\mathbf{C}_D$ is a diagonal matrix given by [30]

$$\mathbf{C}_D = \sigma^2\mathbf{I}, \tag{4.67}$$

where σ^2 is the variance of the data measurement error. In this experiment, the standard deviation of the pressure measurement error is $\sigma \approx 0.03$ (note that the measurement error is also non-dimensional). This represents a (very accurate) measurement error in the range of those usually employed in synthetic study cases (see e.g. [30]).

The optimization utilizes an LBGFS implementation [46], with a convergence criterion in the form of a small OF gradient norm value (10 m^{-2}). The matches obtained from the optimizations utilizing fine scale and MS Adjoint Methods are presented in Table 4.4 and Table 4.5, respectively.

It is clear that good matches are obtained via both gradient computation strategies. All matched pressure errors are in the order of 10^{-4}, while the measurement errors are

Table 4.5: Matched data for parameter estimation study utilizing gradients computed via fine scale Adjoint Method.

Well	Observed Pressure [-]	Initial Pressure [-]	Matched Pressure [-]	Percent Error [%]
OBSWELL1	0.2508	0.1704	0.2498	0.080
OBSWELL2	0.6918	0.8284	0.6929	0.065

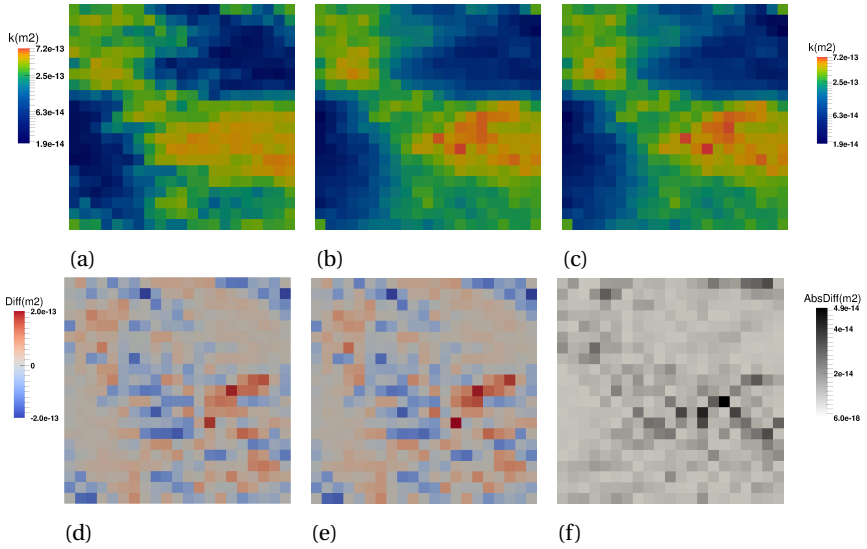


Figure 4.12: Permeability field updates. Initial permeability field (a), after model calibration with fine scale (b) and MS (c) gradient computation. Difference between initial permeability field and fine scale (d) and MS(e) model calibration. Absolute difference between fine scale and MS permeability fields after model update is also shown (f).

10^{-5}. The permeability fields estimated by both gradient methods are illustrated in Figure 4.12. The same figure also shows the differences between initial and updated permeability fields, as well as the fine scale vs. MS estimated parameters.

It is clear that the updates employed when a MS gradient is provided (Figure 4.12e) to the optimizer are close to the updates found when a fine scale gradient (Figure 4.12d) is utilized.

Finally, the performance of the optimization algorithm is assessed when both (fine scale and MS) gradients are utilized. Figure 4.13 illustrates the evolution of the normalized OF along the optimization process, where the OF values are normalized by its initial value.

Although MS-gradient approach is much more efficient (recall Figure 4.2), and both optimizations lead to good matches when converged, the MS-gradient based optimization converges slower than the one based on the fine scale gradients. It is noted that the quality of the MS solution can be further improved through an iterative procedure [16],

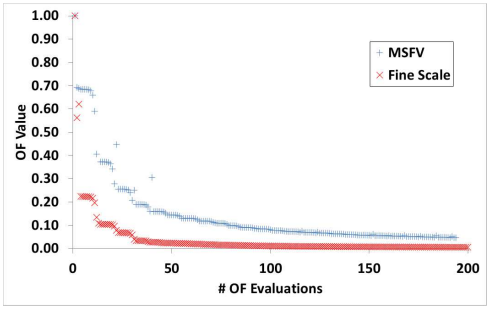


Figure 4.13: Evolution of the normalized OF value for model calibration utilizing both fine scale and MSFV gradients.

which will be subject of our future research.

REFERENCES

[1] R. J. de Moraes, J. R. Rodrigues, H. Hajibeygi, and J. D. Jansen, *Multiscale gradient computation for flow in heterogeneous porous media,* Journal of Computational Physics **336**, 644 (2017).

[2] R. Moraes, J. Rodrigues, H. Hajibeygi, and J. D. Jansen, *Multiscale gradient computation for subsurface flow models,* Journal of Computational Physics, (submitted 13 July 2016) (2016).

[3] D. S. Oliver and Y. Chen, *Recent progress on reservoir history matching: a review,* Computat. Geosci. **15**, 185 (2011).

[4] J. D. Jansen, *Adjoint-based optimization of multi-phase flow through porous media–a review,* Computers & Fluids **46**, 40 (2011).

[5] B. Yeten, A. Castellini, B. Guyaguler, and W. Chen, *A comparison study on experimental design and response surface methodologies,* in *SPE Reservoir Simulation Symposium* (Society of Petroleum Engineers, 2005).

[6] L. J. Durlofsky, *Upscaling and gridding of fine scale geological models for flow simulation,* in *8th International Forum on Reservoir Simulation Iles Borromees, Stresa, Italy*, Vol. 2024 (2005).

[7] A. Datta-Gupta, M. King, and S. of Petroleum Engineers (U.S.), *Streamline simulation: theory and practice*, SPE textbook series (Society of Petroleum Engineers, 2007).

[8] J. D. Jansen and L. J. Durlofsky, *Use of reduced-order models in well control optimization,* Optim. Eng. , 1 (2016).

[9] T. Y. Hou and X.-H. Wu, *A multiscale finite element method for elliptic problems in composite materials and porous media,* J. Comput. Phys. **134**, 169 (1997).

[10] P. Jenny, S. H. Lee, and H. A. Tchelepi, *Multi-scale finite-volume method for elliptic problems in subsurface flow simulation,* J. Comput. Phys. **187**, 47 (2003).

[11] Y. Efendiev and T. Y. Hou, *Multiscale finite element methods: theory and applications*, Vol. 4 (Springer Science & Business Media, 2009).

[12] S. H. Lee, H. Zhou, and H. Tchelepi, *Adaptive multiscale finite-volume method for nonlinear multiphase transport in heterogeneous formations,* J. Comput. Phys. **228**, 9036 (2009).

[13] H. Hajibeygi and P. Jenny, *Adaptive iterative multiscale finite volume method,* Journal of Computational Physics **230**, 628 (2011).

[14] A. Kozlova, Z. Li, J. R. Natvig, S. Watanabe, Y. Zhou, K. Bratvedt, and S. Lee, *A real-field multiscale black-oil reservoir simulator,* in *SPE Reservoir Simulation Symposium* (Society of Petroleum Engineers, 2015).

[15] M. Cusini, A. A. Lukyanov, J. Natvig, and H. Hajibeygi, *Constrained pressure residual multiscale (cpr-ms) method for fully implicit simulation of multiphase flow in porous media,* J. Comput. Phys. **299**, 472 (2015).

[16] H. Hajibeygi, G. Bonfigli, M. A. Hesse, and P. Jenny, *Iterative multiscale finite-volume method,* J. Comput. Phys. **227**, 8604 (2008).

[17] H. Zhou and H. A. Tchelepi, *Two-stage algebraic multiscale linear solver for highly heterogeneous reservoir models,* SPE J. **17**, 523 (2012).

[18] D. Cortinovis and P. Jenny, *Iterative galerkin-enriched multiscale finite-volume method,* J. Comput. Phys. **277**, 248 (2014).

[19] M. Ţene, M. S. Al Kobaisi, and H. Hajibeygi, *Algebraic multiscale method for flow in heterogeneous porous media with embedded discrete fractures (f-ams),* J. Comput. Phys. **321**, 819 (2016).

[20] Y. Efendiev, S. Lee, G. Li, J. Yao, and N. Zhang, *Hierarchical multiscale modeling for flows in fractured media using generalized multiscale finite element method,* Int. J. Geomath. **6**, 141 (2015).

[21] H. Hajibeygi and H. A. Tchelepi, *Compositional multiscale finite-volume formulation,* SPE J. **19**, 316 (2014).

[22] P. Jenny and I. Lunati, *Modeling complex wells with the multi-scale finite-volume method,* J. Comput. Phys. **228**, 687 (2009).

[23] I. Lunati and P. Jenny, *Multiscale finite-volume method for density-driven flow in porous media,* Comput. Geosci. **12**, 337 (2008).

[24] Y. Wang, H. Hajibeygi, and H. A. Tchelepi, *Algebraic multiscale solver for flow in heterogeneous porous media,* J. Comput. Phys. **259**, 284 (2014).

[25] M. Ţene, Y. Wang, and H. Hajibeygi, *Adaptive algebraic multiscale solver for compressible flow in heterogeneous porous media,* J. Comput. Phys. **300**, 679 (2015).

[26] O. Møyner and K.-A. Lie, *A multiscale restriction-smoothed basis method for high contrast porous media represented on unstructured grids,* Journal of Computational Physics **304**, 46 (2016).

[27] E. Parramore, M. G. Edwards, M. Pal, and S. Lamine, *Multiscale finite-volume cvd-mpfa formulations on structured and unstructured grids,* Multiscale Model. Simul. **14**, 559 (2016).

[28] S. Shah, O. Møyner, M. Tene, K.-A. Lie, and H. Hajibeygi, *The multiscale restriction smoothed basis method for fractured porous media (f-msrsb),* J. Comput. Phys. **318**, 36 (2016).

[29] M. Cusini, C. van Kruijsdijk, and H. Hajibeygi, *Algebraic dynamic multilevel (adm) method for fully implicit simulations of multiphase flow in porous media,* Journal of Computational Physics **314**, 60 (2016).

[30] D. S. Oliver, A. C. Reynolds, and N. Liu, *Inverse theory for petroleum reservoir characterization and history matching* (Cambridge University Press, 2008).

[31] T. H. Michael, *Scientific computing: an introductory survey* (The McGraw-Hill Companies Inc.: New York, NY, USA, 2002).

[32] F. Anterion, R. Eymard, and B. Karcher, *Use of parameter gradients for reservoir history matching,* in *SPE Symposium on Reservoir Simulation* (Society of Petroleum Engineers, 1989).

[33] W. H. Chen, G. R. Gavalas, J. H. Seinfeld, and M. L. Wasserman, *A new algorithm for automatic history matching,* SPE J. **14**, 593 (1974).

[34] G. Chavent, M. Dupuy, and P. Lemmonier, *History matching by use of optimal theory,* Society of Petroleum Engineers Journal **15**, 74 (1975).

[35] R. Younis and K. Aziz, *Parallel automatically differentiable data-types for next-generation simulator development,* in *SPE Reservoir Simulation Symposium* (Society of Petroleum Engineers, 2007).

[36] S. I. Aanonsen, G. Nævdal, D. S. Oliver, A. C. Reynolds, and B. Vallès, *The ensemble Kalman filter in reservoir engineering–a review,* SPE J. **14**, 393 (2009).

[37] Y. Chen, D. S. Oliver, and D. Zhang, *Efficient ensemble-based closed-loop production optimization,* SPE J. **14**, 634 (2009).

[38] R. M. Fonseca, B. Chen, J. D. Jansen, and A. C. Reynolds, *A stochastic simplex approximate gradient (StoSAG) for optimization under uncertainty,* Int. J. Numer. Meth. Eng. (2016), 10.1002/nme.5342.

[39] J. Fu, H. A. Tchelepi, and J. Caers, *A multiscale adjoint method to compute sensitivity coefficients for flow in heterogeneous porous media,* Advances in water resources **33**, 698 (2010).

[40] J. Fu, J. Caers, and H. A. Tchelepi, *A multiscale method for subsurface inverse modeling: Single-phase transient flow,* Adv. Water Resour. **34**, 967 (2011).

[41] S. Krogstad, V. L. Hauge, and A. Gulbransen, *Adjoint multiscale mixed finite elements,* SPE Journal **16**, 162 (2011).

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

[43] H. Zhou and H. A. Tchelepi, *Operator-based multiscale method for compressible flow,* SPE J. **13**, 523 (2008).

[44] J. R. P. Rodrigues, *Calculating derivatives for automatic history matching,* Computational Geosciences **10**, 119 (2006).

[45] J. F. B. M. Kraaijevanger, P. J. P. Egberts, J. R. Valstar, and H. W. Buurman, *Optimal waterflood design using the adjoint method,* in *SPE Reservoir Simulation Symposium* (Society of Petroleum Engineers, 2007).

[46] J. Nocedal and S. Wright, *Numerical optimization* (Springer Science & Business Media, 2006).

[47] U. Trottenberg, C. Oosterlee, and A. Schueller, *Multigrid* (Elsevier Academic Press, 2001).

[48] W. G. B. Smith, P. Bjorstad, *Domain Decomposition* (Cambridge University Press, 2004).

[49] J. E. Aarnes, *Efficient domain decomposition methods for elliptic problems arising from flows in heterogeneous porous media,* Comput. Visual Sci. **8**, 93 (2005).

[50] N. M. Josuttis, *C++ Templates: The Complete Guide* (Addison-Wesley Professional, 2003).

[51] J. D. Jansen, *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, 2013).

[52] R. M. Fonseca, S. S. Kahrobaei, L. J. T. Van Gastel, O. Leeuwenburgh, and J. D. Jansen, *Quantification of the impact of ensemble size on the quality of an ensemble gradient using principles of hypothesis testing,* in *SPE Reservoir Simulation Symposium* (Society of Petroleum Engineers, 2015).

[53] N. Remy, A. Boucher, and J. Wu, *Applied Geostatistics with SGeMS: A User's Guide* (Cambridge University Press, 2009).

5

MULTISCALE GRADIENT COMPUTATION FOR SEQUENTIALLY COUPLED FLOW AND TRANSPORT IN HETEROGENEOUS POROUS MEDIA

We present the extension of our recently developed Multiscale (MS) gradient computation method to multiphase flow heterogeneous porous media. Assuming a flow and transport sequentially coupled forward simulation, the flow equation is computed using a multiscale finite volume (MSFV) formulation and the transport equation is computed in the fine scale after reconstruction of mass conservative velocity field. The analytical gradient computation strategy follows a generic and flexible formulation that provides derivative information to any gradient-based optimization algorithm. Both the Direct and Adjoint methods are addressed. While the expensive operations involved in the computation of the gradients are performed at coarse scale, fine scale derivative information (e.g. with respect to uncertain parameters) is obtained via the partial derivatives of local MS basis functions. The gradients computed via the MS methods are validated against the one computed via numerical differentiation. The accuracy of the MS gradient are verified against fine-scale gradient computation. The investigation in the proposed synthetic models indicate the potential of the method to be applied to optimization-based reservoir management studies.

Parts of this chapter have been previously published in the proceedings of the SPE Reservoir Simulation Conference (RSC) (2017) [1].

Gradient computation is a fundamental component on optimization algorithms. Numerical differentiation [2, 3], stochastic ensemble-based [4–6], and analytical [7, 8] methods are the most commonly used gradient computation methods. Among those, analytical methods, more specifically the adjoint method [9–11], have been reported to be most accurate and efficient ones [3, 12]. Although not a limitation, adjoint methods have been traditionally employed in fully implicit simulations [13]. In this scenario, the adjoint equation is the transpose of the forward system Jacobian [13]. It is shown in [7] how the adjoint equation can generically account for different coupling strategies by properly considering the lagged-in-time primary variable dependencies.

Another important aspect associated with optimization algorithms is its computational cost. Because many forward simulation runs are required to evaluate the objective function value and its gradient during the optimization process, it is common to develop simulation models that are faster evaluated ([14–16]) to make optimization studies feasible. Multiscale (MS) reservoir simulation [17, 18], a strategy the accurately and efficiently solve flow in heterogeneous porous media, offers unique characteristics that makes it an interesting forward simulation technique for optimization studies. A class of MS methods [19–23], specifically applied to reservoir simulation, employs a mapping between the nested fine and coarse grids defined by MS basis functions [18]. This mapping allows the solution of the flow problem in the coarse grid, thus with less degrees of freedom. At the same time, the mapping allows for the representation of the primary variables in the fine grid. For a review on the main developments on MS reservoir simulation, see [24].

The employment of MS methods as forward simulation strategy allows for the computation of MS gradients. Basically, MS gradient computation provides an approximated gradient based on MS approximated solutions. In [25] it is shown how the sensitivities of a misfit objective function with respect to fine-scale grid-block permeabilities can be computed from the coarse-scale transmissibilities via the solution of local adjoint problems for the MS basis functions. In a follow up work [26], it is shown the performance for the technique in single-phase, transient flow subsurface history matching problem. In [27] it is shown how a MS adjoint strategy is employed to compute gradient information using only coarse scale information, in a sequentially coupled simulation strategy. In the context of life-cycle optimization, where fine-scale sensitivity information is needed, they also employ a coarsening strategy to solve the transport equation [28]. Also, in [29], reservoir simulator's outputs computed from approximated iterative MS Finite Volume (i-MSFV) [21] solutions are successfully used in the computation of stochastic ensemble-based gradient via the Stochastic Simplex Approximate Gradient (StoSAG) [5]. In [30], the finite difference method is combined with a Gauss-Newton approach to approximated the sensitivity matrix via linear interpolation of locally built quadratic functions. Recently, a generic mathematical framework to compute MS gradient information via the direct and adjoint method has been developed [31]. No assumption with respect to the parameters nor the optimization problem nature is made. The performance of MS gradients is demonstrated in single-phase flow optimization models.

In the present work, it is shown how the framework presented in [31] is used to address multiphase flow MS gradient computation. As fine-grid resolution is necessary to accurately capture the sharp, local saturation fronts [32]. Even though MS extensions for the solution of hyperbolic transport equations exist [28, 33], the transport equation is

considered to be solved in fine scale after appropriated conservative velocity field reconstruction [34]. And because MS simulations most commonly require a sequential coupling of flow and transport equations, the framework presented in [31] must be extended to accommodate the latter. Due to its algebraic description, conservative velocity field construction and transport equations can be appropriately accommodated. The time-stepping nature of multiphase flow is captured via a super-vector notation [7, 8, 35].

5.1. DESCRIPTION AND ALGEBRAIC REPRESENTATION OF THE FORWARD MODEL EQUATIONS FOR SEQUENTIALLY COUPLED SOLUTION OF FLOW AND TRANSPORT

In order to properly assess the dependencies of the model equations and primary variables, necessary to develop the analytical gradient methods, the governing equations and the respective solution strategies are discussed. In the scope of this work, and consistent with the main developments regarding MS methods [17, 18], the forward simulation model is built based on a sequential coupling strategy for flow and transport [36].

5.1.1. GOVERNING EQUATIONS, FINE SCALE DISCRETIZATION AND ALGEBRAIC DESCRIPTION OF THE FORWARD MODEL EQUATIONS

Two-phase, immiscible flow is described by mass conservation equation for each phase $\alpha \in \{o, w\}$

$$\frac{\partial}{\partial t}\left(\phi \rho_\alpha S_\alpha\right) + \nabla \cdot \left(\rho_\alpha \mathbf{u}_\alpha\right) = \rho_\alpha q_\alpha, \tag{5.1}$$

where ϕ is the porosity and S_α, ρ_α and q_α are, respectively, saturation, density, and source for phase α. Neglecting gravity and capillary forces, the Darcy velocity for phase α, $\mathbf{u}_\alpha$, can be expressed as

$$\mathbf{u}_\alpha = -\lambda_\alpha \mathbf{K} \cdot \nabla \mathbf{p}, \tag{5.2}$$

where λ_α is the mobility of phase α, i.e. the ratio between the phase relative permeability and viscosity, and $\mathbf{K}$ is the absolute permeability tensor. For incompressible fluid and porous-rock, each mass conservation equation can be divided by the respective (constant) density ρ_α and summed, which yields to

$$-\nabla \cdot (\mathbf{u}_t) = q_t, \tag{5.3}$$

where q_t is the sum of all phases' source terms and

$$\mathbf{u}_t = \lambda_t \mathbf{K} \cdot \nabla p, \tag{5.4}$$

is the total velocity. In Eq. (5.4), λ_t is the total mobility (the sum of all phase mobilities). Finally, one transport equation can be written as

$$\phi \frac{\partial S_\alpha}{\partial t} + \nabla \cdot \left(f_\alpha \mathbf{u}_t\right) = q_\alpha, \tag{5.5}$$

where f_α is the fractional flow of phase α, and is computed as

$$f_\alpha = \frac{\lambda_\alpha}{\lambda_t}, \tag{5.6}$$

where λ_α is the mobility of phase α. The system is closed via the saturation constraint

$$\sum_{\alpha=o,w} S_\alpha = 1. \tag{5.7}$$

Using a two-point flux finite volume discretization (TPFA) in space and sequential discretization in time [36], the discretized form of Eq. (5.3) can be written as

$$\mathbf{g}_p^n = \mathbf{A}^{n-1}\mathbf{p}^n - \mathbf{q}^{n-1} = \mathbf{0}, \tag{5.8}$$

where $\mathbf{p}^n \in \mathbb{R}^{N_F}$ and $\mathbf{q}^{n-1} \in \mathbb{R}^{N_F}$ are pressure and source terms for all phases vectors, N_F being the number of fine grid blocks, and $\mathbf{A}^{n-1} \in \mathbb{R}^{N_F \times N_F}$ is the system matrix. Interfacial rock properties are computed by means of harmonic averages for the absolute permeabilities, whereas an upwind scheme is employed for interfacial fluid properties (i.e. mobilities). The dependency of the fluid mobilities on the saturation is treated lagged in time because of sequential solution strategy.

The discretization of Eq. (5.5) can be written as

$$\mathbf{g}_s^n = \mathbf{V}\left(\mathbf{s}^n - \mathbf{s}^{n-1}\right) + \tilde{\mathbf{F}}^t\mathbf{u}^n - \mathbf{q}_\alpha^t = \mathbf{0}, \tag{5.9}$$

where $\mathbf{s} \in \mathbb{R}^{N_F}$, $\tilde{\mathbf{F}}^t \in \mathbb{R}^{N_F \times N_I}$, and $\mathbf{u}^n \in \mathbb{R}^{N_I}$ are, respectively, the saturation vector, the fractional flow matrix and the velocity vector velocity orthogonal components at the interfaces, with N_I being the number of fine grid interfaces, and

$$\mathbf{V} = \frac{\mathbf{V}_\phi}{\Delta t}\mathbf{I}, \tag{5.10}$$

where $\mathbf{V} \in \mathbf{R}^{N_F \times N_F}$. Also, Δt is the time-step size, $\mathbf{V}_\phi \in \mathbb{R}^{N_F}$ is the vector containing the grid block pore-volumes, and $\mathbf{I}$ is the identity matrix.

The de-coupling of equations Eq. (5.8) and Eq. (5.9) allows the system to be solved sequentially, with no dependency of Eq. (5.8) on $\mathbf{s}^n$. Note that the non-linear dependency of Eq. (5.18) on $\mathbf{s}^t$ defines the flow-transport coupling. If $t = n-1$, the fractional flow and source term are evaluated at the previous time-step. This allows $\mathbf{s}^n$ to be easily obtained from Eq. (5.9). However, due to instabilities, this scheme has limitations on the time-step size. This is the so-called implicit-pressure explicit-saturation (IMPES) discretization in time [36], or simply sequential explicit. On the other hand, if $t = n$, a sequential implicit strategy is difined, where Eq. (5.9) now has a non-linear dependency on $\mathbf{s}^n$. This scheme allows for larger times steps, however requires a non-linar solution of Eq. (5.9).

5.1.2. MULTISCALE DISCRETIZATION AND ALGEBRAIC DESCRIPTION OF THE FORWARD MODEL EQUATIONS

Instead of solving an expensive $N_F \times N_F$ linear system from Eq. (5.8), the flow equation is solved using a MS method. MS is an efficient strategy to solve system of equations arising

from the discretization of elliptic PDE's. More specifically, we employ a multiscale finite volume (MSFV) formulation [17, 19], which is capable of providing mass conservative solutions. However, note that the transport equation is computed at the fine-scale, after reconstruction of mass conservative velocity field [34]. The conservative velocity field is reconstructed from the approximate MS pressure field via the solution of Neumann local problems based on the mass-conservative fluxes at the coarse-grid block boundaries. A MSFV solution strategy is employed to solve Eq. (5.8). Two sets of (dual and primal) coarse grids are constructed in order to formulate a MSFV method (see Fig. 5.1). The resulting MS system can be algebraically expressed as [19]

$$(\mathbf{RAP})^{n-1}\check{\mathbf{p}}^n = \mathbf{R}^{n-1}\mathbf{q}^n, \tag{5.11}$$

where $\mathbf{R} \in \mathbb{R}^{N_C \times N_F}$ is the restriction operator, $\mathbf{P} \in \mathbb{R}^{N_F \times N_C}$ the prolongation operator and $\check{\mathbf{p}} \in \mathbb{R}^{N_C}$ the coarse pressure solution, with N_C being the number of coarse grid blocks. The interpolated fine-scale pressure is obtained by means of the prolongation operator as

$$\mathbf{p}'^n = \mathbf{P}^{n-1}\check{\mathbf{p}}^n, \tag{5.12}$$

where $\mathbf{p}' \in \mathbb{R}^{N_F}$ is the approximated fine scale pressure solution.

Figure 5.1: Illustration of MSFV coarse grids for a 2D domain. Given a fine-scale grid (shown in light solid black lines), the coarse grid (shown in solid bold black) is imposed as a non-overlapping partition of the computational domain. The coarse nodes (vertices) are then selected (red cells). Connecting coarse nodes constructs the dual-coarse grid (blue cells) where basis functions are constructed. The boundary of the dual-cell k is represented by $\partial\check{\Omega}_k$ and the $k-th$ primal-cell domain by $\check{\Omega}_k$

Although the approximated pressure can provide a conservative velocity field at the coarse scale

$$\mathbf{u}' = -\lambda \cdot \nabla p', \tag{5.13}$$

the velocity vector $\mathbf{u}' \in \mathbb{R}^{N_I}$ is not conservative at the fine-scale. Hence, an additional local problem [34, 37]

$$-\nabla \cdot \left(\lambda \cdot \nabla p''_k\right) = q'' \tag{5.14}$$

is solved on the primal coarse grid cell domain $\check{\Omega}_k$ with the Neumann boundary conditions

$$\underbrace{\left(\lambda \cdot \nabla p''_k\right) \cdot \bar{\mathbf{n}}_k}_{\mathbf{u}''} = \underbrace{\left(\lambda \cdot \nabla p'\right) \cdot \bar{\mathbf{n}}_k}_{\mathbf{u}'} \text{ at } \partial\check{\Omega}_k. \tag{5.15}$$

Here, p'' is the corrected pressure solution and q'' are the source terms. The subscript k denotes for the domain corresponding to the $k-th$ coarse grid, and $\bar{\mathbf{n}}_k$ is the unit vector normal to $\partial\check{\Omega}_k$.

Equations Eq. (5.14) can be written in discrete form as

$$\mathbf{A}''^{n-1}\mathbf{p}''^n = \mathbf{q}''^n + \mathbf{E}_I^C\left(\Lambda'^{n-1}\mathbf{p}'^n\right), \tag{5.16}$$

where $\mathbf{A}''^{n-1} \in \mathbb{R}^{N_F \times N_F}$ results from the assemblage of the local discretizations of Eq. (5.14) into one global block diagonal matrix and $\mathbf{q}''^n \in \mathbb{R}^{N_F}$ is the vector containing source

terms and $\mathbf{E}_I^C \in \mathbf{R}^{N_F \times N_I}$ is the operator that maps the Neumann boundary conditions imposed by Eq. (5.15) from the interfaces to the cells and $\Lambda'^{n-1} \in \mathbb{R}^{N_I \times N_F}$ is the transmissibilty matrix built based on $\mathbf{p}'^n$.

Using the corrected pressure $\mathbf{p}''^n \in \mathbb{R}^{N_F}$ from the solution of Eq. (5.16) to calculate

$$\mathbf{u} = \begin{cases} -\lambda \cdot \nabla p''_k \text{ on } \breve{\Omega}_k, \\ -\lambda \cdot \nabla p' \text{ at } \partial\breve{\Omega}_k, \end{cases} \tag{5.17}$$

provides a conservative velocity field inside the $\breve{\Omega}_k$ domain. The velocity field computed via Eq. (5.17) is, therefore, suited to be used in the solution of Eq. (5.9). The discrete form of Eq. (5.17) can be expressed as

$$\mathbf{u}^n = \Lambda'^{n-1}\mathbf{p}'^n + \Lambda''^{n-1}\mathbf{p}''^n, \tag{5.18}$$

where $\Lambda''^{n-1} \in \mathbb{R}^{N_I \times N_F}$ is the transmissibilty matrix built based on $\mathbf{p}''$. Note that $\Lambda' \neq \mathbf{0}$ only at the boundaries of the primal coarse cells, $\partial\breve{\Omega}_k$. Similarly, $\Lambda'' \neq \mathbf{0}$ only on the interior of the primal coarse cells and has a block diagonal structure. Fig. 5.2 illustrates the final conservative velocity field.

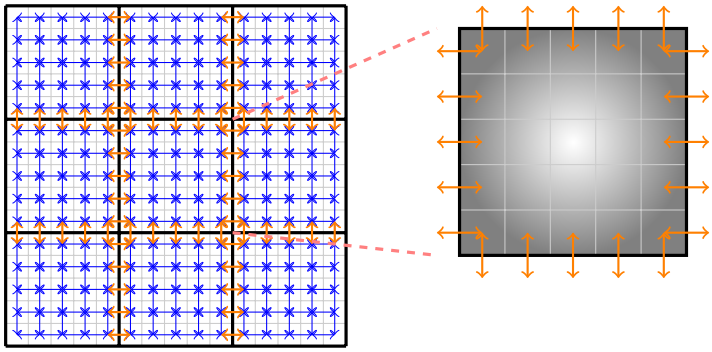


Figure 5.2: Illustration of conservative velocity field reconstruction. The orange arrows represent the interfacial velocity orthogonal components computed from the (conservative) coarse-scale solution. These components are utilized as boundary conditions in the construction of $\mathbf{p}''$, represented by the gray-scale pressure field (right). The blue arrows represent the interfacial velocity orthogonal components computed from $\mathbf{p}''$.

As discussed in [31], the MS solution for the flow equation for time-step n can be algebraically described by

$$\breve{\mathbf{g}}^n\left(\breve{\mathbf{p}}, \mathbf{s}^{n-1}, \theta\right) = (\mathbf{RAP})^{n-1}\breve{\mathbf{p}}^n - \mathbf{R}^{n-1}\mathbf{q}^n = \breve{\mathbf{A}}^{n-1}\breve{\mathbf{p}}^n - \breve{\mathbf{q}}^n = \breve{\mathbf{0}}, \tag{5.19}$$

and

$$\mathbf{g}'^n\left(\breve{\mathbf{p}}, \mathbf{p}', \mathbf{s}^{n-1}, \theta\right) = \mathbf{p}'^n - \mathbf{P}^{n-1}\breve{\mathbf{p}}^n = \mathbf{0}. \tag{5.20}$$

Note that $\mathbf{A}^{n-1} = \mathbf{A}^{n-1}\left(\mathbf{s}^{n-1}\right)$ and $\mathbf{P}^{n-1} = \mathbf{P}^{n-1}\left(\mathbf{s}^{n-1}\right)$. Furthermore, although prolongation depends on time, it is only rebuilt infrequently, depending on how much the mobilities change in the corresponding coarse grid block [34].

From Eq. (5.18) one can write

$$\mathbf{g}_u\left(\mathbf{u}^n,\mathbf{p}'^n,\mathbf{p}''^n,\mathbf{s}^{n-1},\theta\right)=\mathbf{u}^n-\Lambda'^{n-1}\mathbf{p}'^n-\Lambda''^{n-1}\mathbf{p}''^n=\mathbf{0}. \tag{5.21}$$

Note that $\Lambda'^{n-1}=\Lambda'^{n-1}\left(\mathbf{s}^{n-1}\right)$ and $\Lambda''^{n-1}=\Lambda''^{n-1}\left(\mathbf{s}^{n-1}\right)$. Also, from Eq. (5.16), one can define

$$\mathbf{g}''^n\left(\mathbf{p}^n,\mathbf{p}'^n,\mathbf{s}^{n-1},\theta\right)=\mathbf{A}''^n\mathbf{p}''^n-\mathbf{q}''^n-\mathbf{E}_I^C\left(\Lambda'^{n-1}\mathbf{p}'^n\right)=\mathbf{0}. \tag{5.22}$$

Equation Eq. (5.9) can be algebraically expressed as

$$\mathbf{g}_s^n\left(\mathbf{p}'^n,\mathbf{u}^n,\mathbf{s}^{n-1},\mathbf{s}^n,\theta\right)=\mathbf{s}^t-\mathbf{s}^{n-1}-\mathbf{F}^t\mathbf{u}^n-\mathbf{q}_\alpha^t=\mathbf{0} \tag{5.23}$$

where

$$\mathbf{F}^t\left(\mathbf{s}^t\right)=\mathbf{V}^{-1}\tilde{\mathbf{F}}^t \tag{5.24}$$

is an upwind fractional flow operator, $\mathbf{F}^t\in\mathbb{R}^{N_F\times N_I}$.

5.2. MATHEMATICAL FRAMEWORK FOR THE COMPUTATION OF FLOW-TRANSPORT SEQUENTIALLY COUPLED, MULTISCALE GRADIENT COMPUTATION

The mathematical framework for the computation of gradient information for sequatially coupled system for equations is discussed in [38]. It is presented in a very general setup. Given that the forward model equations are described in a generic, purely algebraic form, the framework can be applied to the computation of any desired derivative information. In this direction, the multiscale forward model equations will be described in such way the algorithms presented in [38] can be employed in the computation of multiscale gradients. But, firstly, we brifly describe the derivative computation framework.

5.2.1. GRADIENT COMPUTATION MATHEMATICAL FRAMEWORK

It is discussed in [7, 31, 38] how any derivative information can be efficiently computed from the sensitivity matrix as

$$\mathbf{WGV}=-\mathbf{W}\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}}\left(\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\right)^{-1}\frac{\partial \boldsymbol{g}}{\partial \theta}\mathbf{V}+\mathbf{W}\frac{\partial \boldsymbol{h}}{\partial \theta}\mathbf{V}, \tag{5.25}$$

where $\mathbf{V}$ (of order $N_\theta\times p$) and $\mathbf{W}$ (of order $m\times N_Y$) are arbitrary matrices, defined based on the derivative information one wants to obtain.

In Eq. (6.18), a super-vector notation [7, 8] is used to capture the evolution in time, hence

$$\boldsymbol{g}\left(\boldsymbol{x}\left(\theta\right),\theta\right)=\mathbf{0}, \tag{5.26}$$

where ,

$$\boldsymbol{y}=\boldsymbol{h}\left(\boldsymbol{x}\left(\theta\right),\theta\right). \tag{5.27}$$

The key aspect that defines the computational performance of the gradient computation is the order of the operations involving $\left(\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\right)^{-1}$. Based on that, both the Direct

and Adjoint analytical methods to compute the necessary derivative information can be defined.

If $\mathbf{W}$ is factored out in Eq. (6.18), it can be rewritten as

$$\mathbf{GV} = \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}}\mathbf{Z} + \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{\theta}}\mathbf{V}. \tag{5.28}$$

where

$$\mathbf{Z} = -\left(\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\right)^{-1}\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{\theta}}\mathbf{V}, \tag{5.29}$$

is solved from

$$\left(\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\right)\mathbf{Z} = -\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{\theta}}\mathbf{V}. \tag{5.30}$$

The linear system described in Eq. (6.25) can be re-written in a block-wise fashion for each time-step n:

$$\begin{pmatrix} \frac{\partial \mathbf{g}^0}{\partial \mathbf{x}^0} & & & \\ \hline \frac{\partial \mathbf{g}^1}{\partial \mathbf{x}^0} & \frac{\partial \mathbf{g}^1}{\partial \mathbf{x}^1} & & \\ \hline & \ddots & \ddots & \\ \hline & & \frac{\partial \mathbf{g}^N}{\partial \mathbf{x}^{N-1}} & \frac{\partial \mathbf{g}^N}{\partial \mathbf{x}^N} \end{pmatrix} \begin{pmatrix} \mathbf{Z}^0 \\ \hline \mathbf{Z}^1 \\ \hline \vdots \\ \hline \mathbf{Z}^N \end{pmatrix} = -\begin{pmatrix} \frac{\partial \mathbf{g}^0}{\partial \boldsymbol{\theta}}\mathbf{V} \\ \hline \frac{\partial \mathbf{g}^1}{\partial \boldsymbol{\theta}}\mathbf{V} \\ \hline \vdots \\ \hline \frac{\partial \mathbf{g}^N}{\partial \boldsymbol{\theta}}\mathbf{V} \end{pmatrix} \tag{5.31}$$

The partitioning lines indicate which matrix and vector terms belong to each time-step.

Now, if $\mathbf{V}$ is factored out in Eq. (6.18), it can be rewritten as

$$\mathbf{WG} = \mathbf{Z}\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{\theta}} + \mathbf{W}\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{\theta}}, \tag{5.32}$$

where

$$\mathbf{Z} = -\mathbf{W}\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}}\left(\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\right)^{-1} \tag{5.33}$$

is solved from

$$\mathbf{Z}\left(\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\right) = -\mathbf{W}\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}}. \tag{5.34}$$

The linear system described in Eq. (6.37) can be re-written in a block-wise fashion for each time-step n as

$$
\begin{array}{c}
\left(\begin{array}{c|c|c|c} \mathbf{Z}^0 & \mathbf{Z}^1 & \dots & \mathbf{Z}^N \end{array}\right)\times \\
\left(\begin{array}{c|c|c|c}
\dfrac{\partial \mathbf{g}^0}{\partial \mathbf{x}^0} & & & \\ \hline
\dfrac{\partial \mathbf{g}^1}{\partial \mathbf{x}^0} & \dfrac{\partial \mathbf{g}^1}{\partial \mathbf{x}^1} & & \\ \hline
 & \ddots & \ddots & \\ \hline
 & & \dfrac{\partial \mathbf{g}^N}{\partial \mathbf{x}^{N-1}} & \dfrac{\partial \mathbf{g}^N}{\partial \mathbf{x}^N}
\end{array}\right) = \\
-\left(\begin{array}{c|c|c|c} \mathbf{W}^0 & \mathbf{W}^1 & \dots & \mathbf{W}^N \end{array}\right)\times \\
\left(\begin{array}{c|c|c|c}
\dfrac{\partial \mathbf{h}^0}{\partial \mathbf{x}^0} & & & \\ \hline
\dfrac{\partial \mathbf{h}^1}{\partial \mathbf{x}^0} & \dfrac{\partial \mathbf{h}^1}{\partial \mathbf{x}^1} & & \\ \hline
 & \ddots & \ddots & \\ \hline
 & & \dfrac{\partial \mathbf{h}^N}{\partial \mathbf{x}^{N-1}} & \dfrac{\partial \mathbf{h}^N}{\partial \mathbf{x}^N}
\end{array}\right)
\end{array}
\tag{5.35}
$$

The structure of Eq. (6.38) shows that it can be solved via a back substitution, i.e. the solution of the gradient information is backward in time.

Although the derivation as presented so far is considerably general, in order to properly formulate the actual method to analytically compute the gradient information, the structure of the partial derivative matrices involved must be taken into account. And this is only possible if the specific coupling strategy and the proper dependencies of the model equations and primary variables are taken into account. In this direction, such dependencies are properly considered in the context of sequentially coupled, multiscale, multiphase flow. The framework dicussed in [38] is specialized for this specific application.

5.2.2. MULTISCALE GRADIENT COMPUTATION

Following the MS solution discussed in 5.1.2, the forward model set equations can be defined via the following partitioning

$$
\mathbf{g}^n\left(\mathbf{x}^n,\mathbf{x}^{n-1},\theta,\right) = \begin{bmatrix} \mathbf{g}_p^n \\ \mathbf{g}_u^n \\ \mathbf{g}_s^n \end{bmatrix} = \left[\begin{array}{c} \breve{\mathbf{g}}^n \\ \mathbf{g}'^n \\ \hline \mathbf{g}''^n \\ \mathbf{g}_u{}^n \\ \hline \mathbf{g}_s{}^n \end{array}\right], \tag{5.36}
$$

where

$$
\mathbf{x}^n = \mathbf{x}^n(\theta) = \left[\begin{array}{c} \breve{\mathbf{p}} \\ \mathbf{p}' \\ \hline \mathbf{p}'' \\ \mathbf{u} \\ \hline \mathbf{s} \end{array}\right]^n. \tag{5.37}
$$

The horizontal partition lines group the (sub) equations and primary variables involved in the computation of flow, velocity and transport models equations.

From Eq. (5.19), Eq. (5.20), $\mathbf{g}_p^n$ can be fully defined as

$$\mathbf{g}_p^n\left(\mathbf{x}^n,\mathbf{x}^{n-1},\theta\right) = \begin{bmatrix} \check{\mathbf{A}}^{n-1} & \\ -\mathbf{P}^{n-1} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \check{\mathbf{p}} \\ \mathbf{p}' \end{bmatrix}^n - \begin{bmatrix} \check{\mathbf{q}}^n \\ \mathbf{0} \end{bmatrix} = \mathbf{0}, \tag{5.38}$$

By differentiating Eq. (5.38) with respect to Eq. (5.37), it follows that

$$\frac{\partial \mathbf{g}_p^n}{\partial \mathbf{x}^n} = \begin{bmatrix} \check{\mathbf{A}}^{n-1} & \\ -\mathbf{P}^{n-1} & \mathbf{I} \end{bmatrix}. \tag{5.39}$$

Because the only state variable lagged in time that the model equations in the current time-step are dependent on is the saturation, the partial derivative of Eq. (5.38) with respect to $\mathbf{x}^{n-1}$

$$\frac{\partial \mathbf{g}_p^n}{\partial \mathbf{x}^{n-1}} = \begin{bmatrix} \mathbf{0} & \cdots & \frac{\partial \check{\mathbf{g}}^n}{\partial \mathbf{s}^{n-1}} \\ \mathbf{0} & \cdots & \frac{\partial \mathbf{g}'^n}{\partial \mathbf{s}^{n-1}} \end{bmatrix}. \tag{5.40}$$

From Eq. (5.22) and Eq. (5.21), $\mathbf{g}_u^n$ can be fully defined as

$$\mathbf{g}_u^n\left(\mathbf{x}^n,\mathbf{x}^{n-1},\theta\right) = \begin{bmatrix} -\mathbf{E}_I^C\Lambda'^{n-1} & \mathbf{A}''^{n-1} & \\ -\Lambda'^{n-1} & -\Lambda''^{n-1} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{p}' \\ \mathbf{p}'' \\ \mathbf{u} \end{bmatrix}^n - \begin{bmatrix} \mathbf{0} \\ \mathbf{q}'' \\ \mathbf{0} \end{bmatrix} = \mathbf{0}, \tag{5.41}$$

By differentiating Eq. (5.41) with respect to Eq. (5.37), it follows that

$$\frac{\partial \mathbf{g}_u^n}{\partial \mathbf{x}^n} = \begin{bmatrix} -\mathbf{E}_I^C\Lambda'^{n-1} & \mathbf{A}''^{n-1} & \\ -\Lambda'^{n-1} & -\Lambda''^{n-1} & \mathbf{I} \end{bmatrix}, \tag{5.42}$$

and with respect to $\mathbf{x}^{n-1}$

$$\frac{\partial \mathbf{g}_u^n}{\partial \mathbf{x}^{n-1}} = \left[\begin{array}{ccc} \mathbf{0} & \cdots & \frac{\partial \mathbf{g}''^n}{\partial \mathbf{s}^{n-1}} \\ \mathbf{0} & \cdots & \frac{\partial \mathbf{g}_u{}^n}{\partial \mathbf{s}^{n-1}} \end{array}\right]. \tag{5.43}$$

Because the transport equation is solved at fine scale, after the proper reconstruction of the conservative velocity field, the partitioning of $\mathbf{g}_s^n$ Eq. (5.23) is kept unchanged when compared to the fine-scale derivative computation strategy [38].

Next, by utilizing the partitions of the forward model equations as just presented, the MS version of the Direct and Adjoint methods algorithms for sequentially coupled forward simulation are derived as specializations of the Direct and Adjoint methods for fine-scale simulation.

5.2.3. MULTISCALE DIRECT METHOD

By utilizing the partition

$$\mathbf{Z}_p^n = \begin{bmatrix} \check{\mathbf{Z}}^n \\ \mathbf{Z}'^n \end{bmatrix}, \tag{5.44}$$

and utilizing Eq. (5.39), $\mathbf{Z}_p^n$ can be computed as

$$\begin{bmatrix} \check{\mathbf{A}}^{n-1} & \\ -\mathbf{P}^{n-1} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \check{\mathbf{Z}}^n \\ \mathbf{Z}'^n \end{bmatrix} = \begin{bmatrix} \dfrac{\partial \check{\mathbf{g}}}{\partial \theta}\mathbf{V} - \dfrac{\partial \check{\mathbf{g}}^n}{\partial \mathbf{s}^{n-1}}\mathbf{Z}_s^{n-1} \\ \dfrac{\partial \mathbf{g}'}{\partial \theta}\mathbf{V} - \dfrac{\partial \mathbf{g}'^n}{\partial \mathbf{s}^{n-1}}\mathbf{Z}_s^{n-1} \end{bmatrix}. \tag{5.45}$$

Similarly, for the velocity equation,

$$\begin{bmatrix} -\mathbf{E}_I^C \Lambda'^{n-1} & \mathbf{A}''^{n-1} & \\ -\Lambda'^{n-1} & -\Lambda''^{n-1} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{Z}'^n \\ \mathbf{Z}''^n \\ \mathbf{Z}_u^n \end{bmatrix} = \begin{bmatrix} \dfrac{\partial \mathbf{g}''}{\partial \theta}\mathbf{V} - \dfrac{\partial \mathbf{g}''^n}{\partial \mathbf{s}^{n-1}}\mathbf{Z}_s^{n-1} \\ \dfrac{\partial \mathbf{g}_u}{\partial \theta}\mathbf{V} - \dfrac{\partial \mathbf{g}_u^n}{\partial \mathbf{s}^{n-1}}\mathbf{Z}_s^{n-1} \end{bmatrix}, \tag{5.46}$$

and for the transport equation,

$$\mathbf{Z}_s^n = \frac{\partial \mathbf{g}_s^n}{\partial \theta}\mathbf{V} - \frac{\partial \mathbf{g}_s^n}{\partial \mathbf{p}'^n}\mathbf{Z}'^n - \mathbf{F}^{n-1}\mathbf{Z}_u^n - \frac{\partial \mathbf{g}_s^n}{\partial \mathbf{s}^{n-1}}\mathbf{Z}_s^{n-1}. \tag{5.47}$$

Now, the algorithm to compute the gradient information using the Direct method on a MS fashion can be fully defined, and it is described in 9.

5.2.4. MULTISCALE ADJOINT METHOD

By utilizing the partition defined in Eq. (5.44) and Eq. (5.39) and Eq. (5.40), $\mathbf{Z}_p^n$ can be computed as

$$\begin{bmatrix} \left(\check{\mathbf{A}}^{n-1}\right)^T & \left(-\mathbf{P}^{n-1}\right)^T & & & \\ & \mathbf{I} & \left(-\mathbf{E}_I^C \Lambda'^{n-1}\right)^T & \left(-\Lambda'^{n-1}\right)^T & \dfrac{\partial \mathbf{g}_s}{\partial \mathbf{p}'} \end{bmatrix} \begin{bmatrix} \left(\check{\mathbf{Z}}^n\right)^T \\ \left(\mathbf{Z}'^n\right)^T \\ \left(\mathbf{Z}''^n\right)^T \\ \left(\mathbf{Z}_u{}^n\right)^T \\ \left(\mathbf{Z}_s{}^n\right)^T \end{bmatrix} = \begin{bmatrix} -\left(\mathbf{W}^n \dfrac{\partial \mathbf{h}^n}{\partial \mathbf{p}'^n}\right)^T \end{bmatrix}. \tag{5.48}$$

From Eq. (5.42) and Eq. (5.43), $\mathbf{Z}_u^n$ can be computed as

$$\begin{bmatrix} \left(\mathbf{A}''^{n-1}\right)^T & \left(-\Lambda''^{n-1}\right)^T \\ & \mathbf{I} \end{bmatrix} \begin{bmatrix} \left(\mathbf{Z}''^n\right)^T \\ \left(\mathbf{Z}_u{}^n\right)^T \end{bmatrix} = \begin{bmatrix} -\left(\mathbf{F}^{n-1}\right)^T \left(\mathbf{Z}_s^n\right)^T \end{bmatrix}. \tag{5.49}$$

5

Algorithm 9: Multiscale Gradient Computation via the Direct Method.

Input : Partial derivative matrices of $\mathbf{g}_p$ and $\mathbf{g}_s$ w.r.t. $\mathbf{x}$ and θ, $\mathbf{V}$
Output: $\mathbf{GV}$

foreach $n = 0, 1, 2, \ldots, N$ **do**
- Compute $\alpha = \left(\frac{\partial \mathbf{A}}{\partial \theta}\mathbf{P}\check{\mathbf{p}} - \frac{\partial \mathbf{q}}{\partial \theta}\right)^n$
- Compute $\eta = \left(\frac{\partial \mathbf{A}}{\partial \mathbf{s}^{n-1}}\mathbf{P}\check{\mathbf{p}} - \frac{\partial \mathbf{q}}{\partial \mathbf{s}^{n-1}}\right)^n$
- **foreach** $j = 1, 2, \ldots, p$ **do**
  - Compute $\gamma_\theta = \frac{\partial \mathbf{P}}{\partial \theta}\check{\mathbf{p}}\mathbf{V}_{.,j}$; `// Algorithm 3 in [31], where` $\mathbf{m} = \mathbf{V}_{.,j}$
  - Compute $\beta_\theta = \mathbf{R}^{n-1}\left(\alpha \mathbf{V}_{.,j} + \mathbf{A}\gamma_\theta\right)$
  - Compute $\gamma_s = \frac{\partial \mathbf{P}}{\partial \mathbf{s}^{n-1}}\check{\mathbf{p}}\mathbf{Z}_{s}{}_{.,j}^{n-1}$; `// Algorithm 3 in [31], where` $\mathbf{m} = \mathbf{Z}_{s}{}_{.,j}^{n-1}$
  - Compute $\beta_s = \mathbf{R}^{n-1}\left(\eta \mathbf{Z}_{s}{}_{.,j}^{n-1} + \mathbf{A}\gamma_s\right)$
  - Solve $\check{\mathbf{Z}}_{.,j}^{n} = \left(\check{\mathbf{A}}^{n-1}\right)^{-1}(\beta_\theta - \beta_s)$
  - Compute $\mathbf{Z'}_{.,j}^{n} = \mathbf{P}^{n-1}\check{\mathbf{Z}}_{.,j}^{n} - \gamma_\theta - \gamma_s$
  - Solve $\mathbf{Z''}_{.,j}^{n} = \left(\mathbf{A''}^{n-1}\right)\left(\frac{\partial \mathbf{g''}}{\partial \theta}\mathbf{V}_{.,j} - \frac{\partial \mathbf{g''}^{n}}{\partial \mathbf{s}^{n-1}}\mathbf{Z}_{s}{}_{.,j}^{n-1} + \mathbf{E}_I^C \Lambda'^{n-1}\mathbf{Z'}_{.,j}^{n}\right)$
  - Compute $\mathbf{Z}_{u}{}_{.,j}^{n} = \Lambda'^{n-1}\mathbf{Z'}_{.,j}^{n} + \Lambda''^{n-1}\mathbf{Z''}_{.,j}^{n} + \frac{\partial \mathbf{g}_u}{\partial \theta}\mathbf{V}_{.,j} - \frac{\partial \mathbf{g}_u^n}{\partial \mathbf{s}^{n-1}}\mathbf{Z}_{s}{}_{.,j}^{n-1}$
  - Compute $\mathbf{Z}_{s}{}_{.,j}^{n} = \frac{\partial \mathbf{g}_s^n}{\partial \theta}\mathbf{V}_{.,j} - \frac{\partial \mathbf{g}_s^n}{\partial \mathbf{s}^{n-1}}\mathbf{Z}_{s}{}_{.,j}^{n-1} + \mathbf{F}^{n-1}\mathbf{Z}_{u}{}_{.,j}^{n}$
  - If there are responses at n, compute $(\mathbf{GV})_{.,j}^{n} = \frac{\partial \mathbf{h}^n}{\partial \mathbf{p'}^n}\mathbf{Z'}_{.,j}^{n} + \frac{\partial \mathbf{h}^n}{\partial \mathbf{s}^{n-1}}\mathbf{Z}_{s}{}_{.,j}^{n-1} + \frac{\partial \mathbf{h}^n}{\partial \theta}\mathbf{V}_{.,j}$

And $\mathbf{Z}_s^n$ can be computed as

$$\begin{aligned}\left(\mathbf{Z}_s^n\right)^T = \\ -\left(\mathbf{W}^{n+1}\frac{\partial \mathbf{h}^{n+1}}{\partial \mathbf{s}^n}\right)^T \\ -\left(\frac{\partial \check{\mathbf{g}}^{n+1}}{\partial \mathbf{s}^n}\right)^T \check{\mathbf{Z}}^{n+1} - \left(\frac{\partial \mathbf{g'}^{n+1}}{\partial \mathbf{s}^n}\right)^T \mathbf{Z'}^{n+1} - \left(\frac{\partial \mathbf{g''}^{n+1}}{\partial \mathbf{s}^n}\right)^T \mathbf{Z''}^{n+1} - \left(\frac{\partial \mathbf{g}_u^{n+1}}{\partial \mathbf{s}^n}\right)^T \mathbf{Z}_u^{n+1} \\ -\left(\frac{\partial \mathbf{g}_s{}^{n+1}}{\partial \mathbf{s}^n}\right)^T \left(\mathbf{Z}_s^{n+1}\right)^T.\end{aligned} \tag{5.50}$$

The algorithm to compute the gradient information using the Adjoint method on a MS fashion can be now fully defined, and it can be found in 10.

Algorithm 10: Multiscale Gradient Computation via the Adjoint Method.

Input : Partial derivative matrices of $\mathbf{g}_p$ and $\mathbf{g}_s$ w.r.t. $\mathbf{x}$ and θ, $\mathbf{W}$
Output: $\mathbf{WG}$

Set $\mathbf{WG} = \mathbf{0}$
foreach $n = N, \ldots, 2, 1, 0$ **do**
  Compute $\alpha = \left(\frac{\partial \mathbf{A}}{\partial \theta}\mathbf{P}\check{\mathbf{p}} - \frac{\partial \mathbf{q}}{\partial \theta}\right)^n$
  Compute $\alpha = \mathbf{R}\alpha$
  foreach $i = 1, 2, \ldots, m$ **do**
    Compute $\mathbf{m}^T = \check{\mathbf{Z}}_{i,\cdot}^{n+1}\mathbf{RA} + \mathbf{Z'}_{i,\cdot}^{n+1}$
    Compute $\xi = \mathbf{m}^T \left(\frac{\partial \mathbf{P}}{\partial \mathbf{s}^{n-1}}\check{\mathbf{p}}\right)^n$; // Algorithm 4 in [31]
    Compute $\beta_{s_{i,\cdot}} = \check{\mathbf{Z}}_{i,\cdot}^{n+1}\eta + \xi + \left(\frac{\partial \mathbf{g''}^{n+1}}{\partial \mathbf{s}^n}\right)^T \left(\mathbf{Z''}_{i,\cdot}^{n+1}\right)^T + \left(\frac{\partial \mathbf{g}_u^{n+1}}{\partial \mathbf{s}^n}\right)^T \left(\mathbf{Z}_{u_{i,\cdot}}^{n+1}\right)^T + \left(\frac{\partial \mathbf{g}_s^{n+1}}{\partial \mathbf{s}^n}\right)^T \left(\mathbf{Z}_{s_{i,\cdot}}^{n+1}\right)^T$
    Compute $\left(\mathbf{Z}_{s_{i,\cdot}}^n\right)^T = -\left(\mathbf{W}^n \frac{\partial \mathbf{h}^n}{\partial \mathbf{s}^n}\right)^T - \left(\mathbf{W}_{i,\cdot}^{n+1}\frac{\partial \mathbf{h}^{n+1}}{\partial \mathbf{s}^n}\right)^T - \beta_{s_{i,\cdot}}$
    Compute $\left(\mathbf{Z}_{u_{i,\cdot}}^n\right)^T = \left(\mathbf{F}^{n-1}\right)^T \left(\mathbf{Z}_{s_{i,\cdot}}^n\right)^T$
    Solve $\left(\mathbf{Z''}_{i,\cdot}^n\right)^T = \left(\mathbf{A''}^n\right)^{-T} \left(\Lambda''^{n-1}\right)^T \left(\mathbf{Z}_{u_{i,\cdot}}^n\right)^T$
    Compute $\left(\mathbf{Z'}_{i,\cdot}^n\right)^T = \left(\Lambda'^{n-1}\right)^T \left(\mathbf{Z''}_{i,\cdot}^n\right)^T \left(\mathbf{E}_I^C\right)^T + \left(\Lambda'^{n-1}\right)^T \left(\mathbf{Z}_{u_{i,\cdot}}^n\right)^T - \left(\mathbf{W}_{i,\cdot}^n \frac{\partial \mathbf{h}^n}{\partial \mathbf{p'}^n}\right)^T$
    Solve $\left(\check{\mathbf{Z}}_{i,\cdot}^n\right)^T = \left(\check{\mathbf{A}}^n\right)^{-T} \left(\mathbf{P}^n\right)^T \left(\mathbf{Z'}_{i,\cdot}^n\right)^T$
    Compute $\mathbf{m}^T = \check{\mathbf{Z}}_{i,\cdot}^n \mathbf{RA} - \mathbf{Z'}_{i,\cdot}^n$
    Compute $\gamma = \mathbf{m}^T \left(\frac{\partial \mathbf{P}}{\partial \theta}\check{\mathbf{p}}\right)^n$; // Algorithm 4 in [31]
    Compute $\nu = \check{\mathbf{Z}}_{i,\cdot}^n \alpha - \gamma + \mathbf{Z''}_{i,\cdot}^n \frac{\partial \mathbf{g''}^n}{\partial \theta} + \mathbf{Z}_{u_{i,\cdot}}^n \frac{\partial \mathbf{g}_u^n}{\partial \theta} + \mathbf{Z}_{s_{i,\cdot}}^n \frac{\partial \mathbf{g}_s^n}{\partial \theta} + \mathbf{W}_{i,\cdot}^n \frac{\partial \mathbf{h}^n}{\partial \theta}$
    Compute $(\mathbf{WG})_{i,\cdot} = (\mathbf{WG})_{i,\cdot} + \nu$
  Compute $\eta = \left(\frac{\partial \mathbf{A}}{\partial \mathbf{s}^{n-1}}\mathbf{P}\check{\mathbf{p}} - \frac{\partial \mathbf{q}}{\partial \mathbf{s}^{n-1}}\right)^n$

5.2.5. COMPUTATIONAL AND IMPLEMENTATION ASPECTS OF THE METHODS

COMPUTATION OF PARTIAL DERIVATIVE MATRICES AND AUTOMATIC DIFFERENTIATION

One key aspect of analytical gradient computation is the need for some additional partial derivative matrices that are not required by standard forward simulation procedures.

The partial derivatives of the model equations with respect to the primary variables lagged in time, $\frac{\partial \mathbf{g}^n}{\partial \mathbf{x}^{n-1}}$, can be fully determined from the dependencies of Eq. (5.19), Eq. (5.20), Eq. (5.22), Eq. (5.21) and Eq. (5.23) on $\mathbf{x}^{n-1}$, or more specifically on $\mathbf{s}^{n-1}$ in the scope of the present work. Also, the dependencies of the model equations on the parameters must be taken into account in order to fully determine $\frac{\partial \boldsymbol{g}}{\partial \theta}$.

Ultimately, a reservoir simulation code is an algorithm comprised of many sub-algorithms, each of them dependent on a given pre-determined set of parameters. The output of each (sub-algorithm) will be consumed by another (sub-) algorithm and, finally, by the (sub-) algorithms responsible for the computation of model equations. A convenient way to analytically compute the partial derivatives of the model equations w.r.t. their dependent parameters/variables is Automatic Differentiation (AD). In brief, AD is a computational technique that computes the partial derivatives of a given function w.r.t. its arguments at the same time the function value is evaluated. More information about the fundamentals of AD can be found in [39], and for applications of AD in reservoir simulation problems we refer to [40–42]. The same discussion holds when it comes to the computation of the partial derivatives of the model output, namely $\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}}$ and $\frac{\partial \boldsymbol{h}}{\partial \theta}$.

The partial derivatives of Eq. (5.19), Eq. (5.20), and Eq. (5.22) w.r.t. θ can be computed, respectively, as

$$\frac{\partial \check{\mathbf{g}}^n}{\partial \theta} = \left(\mathbf{R}\frac{\partial \mathbf{A}}{\partial \theta}\mathbf{P} + \mathbf{R}\mathbf{A}\frac{\partial \mathbf{P}}{\partial \theta} + \frac{\partial \mathbf{R}}{\partial \theta}\mathbf{A}\mathbf{P}\right)\check{\mathbf{p}} - \frac{\partial \mathbf{R}}{\partial \theta}\mathbf{q} - \mathbf{R}\frac{\partial \mathbf{q}}{\partial \theta}, \tag{5.51}$$

$$\frac{\partial \mathbf{g}'^n}{\partial \theta} = \frac{\partial \mathbf{P}}{\partial \theta}\check{\mathbf{p}}, \tag{5.52}$$

and

$$\frac{\partial \mathbf{g}''^n}{\partial \theta} = \frac{\partial \mathbf{A}''^{n-1}}{\partial \theta}\mathbf{p}''^n - \frac{\partial \mathbf{q}''^n}{\partial \theta} - \mathbf{E}_I^C\frac{\partial \Lambda'^{n-1}}{\partial \theta}\mathbf{p}'^n, \tag{5.53}$$

Similarly, of Eq. (5.19), Eq. (5.20), and Eq. (5.22) w.r.t. $\mathbf{x}^{n-1}$ can be expressed, respectively, as

$$\frac{\partial \check{\mathbf{g}}^n}{\partial \mathbf{s}^{n-1}} = \left(\mathbf{R}\frac{\partial \mathbf{A}}{\partial \mathbf{s}^{n-1}}\mathbf{P} + \mathbf{R}\mathbf{A}\frac{\partial \mathbf{P}}{\partial \mathbf{s}^{n-1}} + \frac{\partial \mathbf{R}}{\partial \mathbf{s}^{n-1}}\mathbf{A}\mathbf{P}\right)\check{\mathbf{p}} - \frac{\partial \mathbf{R}}{\partial \mathbf{s}^{n-1}}\mathbf{q} - \mathbf{R}\frac{\partial \mathbf{q}}{\partial \mathbf{s}^{n-1}}, \tag{5.54}$$

$$\frac{\partial \mathbf{g}'^n}{\partial \mathbf{s}^{n-1}} = \frac{\partial \mathbf{P}}{\partial \mathbf{s}^{n-1}}\check{\mathbf{p}}, \tag{5.55}$$

and

$$\frac{\partial \mathbf{g}''^n}{\partial \mathbf{s}^{n-1}} = \frac{\partial \mathbf{A}''^{n-1}}{\partial \mathbf{s}^{n-1}}\mathbf{p}''^n - \frac{\partial \mathbf{q}''^n}{\partial \mathbf{s}^{n-1}} - \mathbf{E}_I^C\frac{\partial \Lambda'^{n-1}}{\partial \mathbf{s}^{n-1}}\mathbf{p}'^n, \tag{5.56}$$

The partial derivatives of $\mathbf{A}''^{n-1}$, result in a third-order tensor. The interpretation of the operations involving such terms can be found in [31].

Not only the computation of the partial derivatives requires proper treatment, in order to efficiently evaluate the expressions determining the products, but also special attention must be paid to the order of the operations. This is not only important to determine the Adjoint and Direct methods, but it is also important in some other steps of the computation when operations involve the prolongation operator (in those cases where it depends on the parameters), namely the partial derivatives of $\mathbf{P}$ in Eq. (5.51), Eq. (5.52), and Eq. (5.54), Eq. (5.55). This is discussed next.

PROLONGATION OPERATOR PARTIAL DERIVATIVE

The construction of $\mathbf{P}$ usually requires complex operations. For instance, in the Multiscale Finite Volume (MSFV) method, the construction of the basis functions require the solution of linear systems arising from the local flow problems defined in the dual-grid cells. The efficient computation of the operations involving the partial derivative of the MSFV $\mathbf{P}$ with respect to θ on the form of $\left(\frac{\partial \mathbf{P}}{\partial \theta}\right)\mathbf{m}$ for the Direct method and in the form of $\mathbf{m}^T\left(\frac{\partial \mathbf{P}}{\partial \theta}\right)$ for the Adjoint method are described in [31]. In the case of the Adjoint method, the operations are also performed in an adjoint-like fashion, with computation cost proportional to the number of dual-coarse grid cells and independent of the number of parameters. The same strategy can be employed in the operations involving $\frac{\partial \mathbf{P}}{\partial \mathbf{s}^{n-1}}$.

Since the present work deals with time-stepping problems, this computation must be performed at all time-steps due to changes in λ. But, because basis functions are only infrequently re-constructed, an efficient implementation can take advantage of this fact. In other words, $\mathbf{P}$ is only infrequently and adaptevily updated following a certain criteria, e.g. spatial changes in the mobility ratio [34]. For instance, a direct solver can be utilized for the computation of the prolongation partial derivatives, such that the system matrix is only factorized once and solved multiple times for different right-hand sides.

In the limit case when the dependency of $\mathbf{P}$ on $\mathbf{s}^{n-1}$ is neglected, the operations involving the partial derivatives of $\mathbf{P}$ in Eq. (5.54) and Eq. (5.55) can be disregarded. In other words, if $\mathbf{P}$ is not updated due to changes in λ in forward simulation, only the operations involving $\frac{\partial \mathbf{P}}{\partial \theta}$ must be computed, and only in the beginning of the simulation. Of course, this assumption is case dependent and hence only valid in certain scenarios. This option is highlighted in 10 and 9.

THE CONSERVATIVE VELOCITY FIELD RECONSTRUCTION PARTIAL DERIVATIVE

Along with the inexpensive (as shown by [31]) solution of a linear system of equations arising from $\check{\mathbf{g}}$, the model equation $\mathbf{g}''$ also requires the solution of a linear system. More specifically, for a given time-step n and a given set of outputs i of the same time-step, one can write

$$\mathbf{Z}''^{n}_{.,j} = \left(\mathbf{A}''^{n-1}\right)\left(\frac{\partial \mathbf{g}''}{\partial \theta}\mathbf{V}_{.,j} - \frac{\partial \mathbf{g}''^{n}}{\partial \mathbf{s}^{n-1}}\mathbf{Z}_{s}{}^{n-1}_{.,j} + \mathbf{E}_I^C \Lambda'^{n-1}\mathbf{Z}'^{n}_{.,j}\right), \tag{5.57}$$

for the Direct method and

$$\left(\mathbf{Z}''_{i,.}{}^{n}\right)^T = \left(\mathbf{A}''^{n-1}\right)^{-T}\left(\left(\Lambda''^{n-1}\right)^T\left(\mathbf{Z}_{u_{i,.}}{}^{n-1}\right)^T\right). \tag{5.58}$$

for the Adjoint method. Because of the block structure of the operands of Eq. (5.57) and Eq. (5.58), $\mathbf{Z}''$ can be determined similarly to how the velocity reconstruction is computed in the forward simulation, via the solution of local systems corresponding to the primal-coarse grid cell domains. This results on a embarrassingly parallel algorithm, where all local system solution can be solved independently by different processes.

COMPUTATIONAL EFFICIENCY

The computational efficiency of the methods is assessed via an asymptotic analysis. In the analysis, only the most computationally intensive operations involved in the algorithms are considered. Hence, because the cost of solving linear system of equations overwhelms the cost of the matrix-vector products, only the former is considered over the latter.

The cost associated to the solution of a linear system will be considered to be $\mathcal{O}\left(aN^b\right)$, where a and b are constants dependent of the linear solver employed, and N is the size of the system. For the sake of simplicity, it will be considered that equally efficient solvers are used for the solution of MS and fine-scale linear systems. However, in an actual, efficient, implementation, this should not be the case.

Firstly, let's consider the computational cost associated to solve the derivative information for each time-step performed in the forward simulation. For each column of $\mathbf{V}$ for the Direct method, or each row of $\mathbf{W}$ for the Adjoint, a linear system of size N_C corresponding to the model equation $\check{\mathbf{g}}$, N_C linear systems of size $N_R = N_F/N_C$ associated with the model equation $\mathbf{g}''$, and a linear system of size N_F corresponding to $\mathbf{g}_s$ must be solved in the MS case. Here, N_R is the MS coarsening ratio. In addition, there is the cost associated to the solution of the partial derivative of $\mathbf{P}$ with respect to $\mathbf{s}^{n-1}$ and θ. The partial derivative computation with respect to each of these variables is given by $\mathcal{O}(N_L N_C(a_{MS}{N_R}^{b_{MS}}))$ (from the analysis presented in [31]), where N_L is the number of local problems that must be solved per coarse grid vertex (4 in 2D and 8 in 3D problems). That leads to an estimated computational complexity $\mathcal{O}_{MS}(a{N_F}^b + a{N_C}^b + N_C(a{N_R}^b) + N_L N_C(a{N_R}^b) + N_L N_C(a{N_R}^b))$. In the fine-scale case, a linear system must be solved for each flow and tranport equations, leading to a complexity $\mathcal{O}_{FS}(a{N_F}^b + a{N_F}^b)$.

Ultimately, one is interested in the computational gain of the MS gradient computation over a fine-scale strategy. The computational cost ratio can be estimated by

$$\frac{\mathcal{O}_{MS}}{\mathcal{O}_{FS}} = \frac{1}{2}\left(1 + \frac{1}{N_R^b} + \frac{N_R^{b-1}}{N_F^{b-1}}(1+2N_L)\right). \tag{5.59}$$

Eq. Eq. (5.59) represents the computational cost ratio upper bound, in the sense that $\mathbf{P}$ is assumed to be reconstructed every time-step. However, this is not done in practice, as discussed in Section 5.2.5. There might be cases that no basis function update is required throughout the simulation, and hence the partial derivative of $\mathbf{P}$ w.r.t. $\mathbf{s}^{n-1}$ can be neglected. This scenario gives us the computational cost ratio lower bound and is expressed by

$$\frac{\mathcal{O}_{MS}}{\mathcal{O}_{FS}} = \frac{1}{2}\left(1 + \frac{1}{N_R^b} + \frac{N_R^{b-1}}{N_F^{b-1}}(1+N_L)\right). \tag{5.60}$$

Eqs. Eq. (5.61) and Eq. (5.60) consider that a linear system must be solved for the transport equation. However, in the IMPES case, the saturation can be obtained via an

negligble matrix-vector operation. Because of that, the computational cost ratio upper and lower bounds for the IMPES coupling read, respectively,

$$\frac{\mathcal{O}_{MS}}{\mathcal{O}_{FS}} = \frac{1}{N_R^b} + \frac{N_R^{b-1}}{N_F^{b-1}}(1+2N_L), \tag{5.61}$$

and

$$\frac{\mathcal{O}_{MS}}{\mathcal{O}_{FS}} = \frac{1}{N_R^b} + \frac{N_R^{b-1}}{N_F^{b-1}}(1+N_L). \tag{5.62}$$

Note that the MS gradient computation becomes unattractive when the coarsening ratio is such that the solution of the local systems required by the construction of the basis functions and by the conservative velocity field reconstruction becomes relatively expensive. Clearly, the selection of an optimal coarsening ratio is important, however the computational efficiency is not the only factor in this task. For instance, the ability of the basis function to capture fine-scale features must also be taken into consideration. Also, the velocity field reconstruction might not be necessary if an iterative MS strategy [21] is employed. However, the gradient computation formulation should also take into account the fine-scale smoothing process. This is a topic of an ongoing research. Alternatively, a coarse scale solution could be considered for the transport as well (see e.g. [27, 43]). Although this option has not been investigated in the scope of this work, the generic framework here presented is capable of accomodating this alternative.

Another point that must be highlighted is that, however the cost per time-step associated to the IMPES gradient computation is smaller than the cost compared to the Sequential Implicit gradient computation, due to the well known instabilities imposed by the CFL condition, IMPES simulations usually require (many) more time-steps than Sequential Implicit simulations. Therefore, there is a tradeoff between number of time-steps and time-step cost,

$$\frac{\mathcal{O}_{MS}^{Seq}}{\mathcal{O}_{FS}^{IMPES}} = \frac{N_{TS}^{Seq}}{N_{TS}^{IMPES}} a N_F^b \approx \frac{CFL_{IMPES}}{CFL_{Seq}} a N_F^b,$$

which indicates that the Sequential Implicit strategy is more efficient when $CFL_{Seq} > 1$, given that the condition $CFL_{IMPES} \leq 1$ must be met if one wants to guarantee stability. On top of that, although not captured in the above asymptotic analisys, it is important to note that the more time-steps taken by the forward simulation, the more extra information (partial derivative matrices) must be computed and stored at each time-step to be later used in the backward simulation. This is discussed next.

5.3. NUMERICAL EXPERIMENTS

The following numerical experiments are presented to first validate and then assess the accuracy of the gradient information computed by the method presented in this work. For this purpose, a misfit objective function with no regularization term

$$O(\theta) = \frac{1}{2}(\mathbf{h}(\mathbf{x},\theta) - \mathbf{d}_{obs})^T \mathbf{C}_D^{-1} (\mathbf{h}(\mathbf{x},\theta) - \mathbf{d}_{obs}), \tag{5.63}$$

with a gradient

$$\nabla_\theta O = \mathbf{G}^T \mathbf{C}_D^{-1} \left(\mathbf{h}\left(\mathbf{x},\theta\right) - \mathbf{d}_{obs}\right), \tag{5.64}$$

is considered ([3]). In the experiments, the history matching parameters are cell-centered permeabilities and the observed quantity is oil rate at the production wells. The wells are controlled by minimum bottom-hole pressure (BHP).

5.3.1. VALIDATION

The MS-gradient method is validated against the numerical differentiation method for a higher-order, two-sided Taylor approximation of

$$\begin{aligned}\nabla_\theta O_i = \frac{1}{2\delta\theta_i}\big(&O\left(\theta_i,\ldots,\theta_{i-1},\theta_i+\delta\theta_i,\theta_{i+i},\ldots,\theta_{N_\theta}\right) - \\ &O\left(\theta_i,\ldots,\theta_{i-1},\theta_i-\delta\theta_i,\theta_{i+i},\ldots,\theta_{N_\theta}\right)\big),\end{aligned} \tag{5.65}$$

where δ is a multiplicative parameter perturbation. The relative error can be defined as

$$\varepsilon = \frac{\|\nabla_\theta O_{FD} - \nabla_\theta O_{AN}\|_2}{\|\nabla_\theta O_{AN}\|_2}, \tag{5.66}$$

where $\nabla_\theta O_{FD}$ is obtained by performing the appropriate number of multiscale reservoir simulations required to evaluate Eq. (5.65) and $\nabla_\theta O_{AN}$ is obtained by either employing the Adjoint MS gradient computation here depicted to evaluate Eq. (5.64). Note that the framework here presented can be used by making

$$\mathbf{W}^T = \mathbf{C}_D^{-1}\left(\mathbf{h}\left(\mathbf{x},\theta\right) - \mathbf{d}_{obs}\right), \tag{5.67}$$

so that the gradient of O can be written as $\nabla_\theta O = (\mathbf{WG})^T$. For all cases, for simplicity, it is assumed $\mathbf{C}_D = \mathbf{I}$.

In order to validate the proposed derivative calculation method, as well as the implementation, we investigate the linear decrease of the error ε by decreasing the perturbation value δ ([2]) from 10^{-1} to 10^{-4} (the range within which discretization errors dominate).

The first case is a one-dimensional, homogeneous medium with 45 grid blocks. A primal coarse grid of just 3 grid blocks is employed (coarsening ratio of 15). Injection and production wells are located at, respectively, at the first and last vertices of the primal grid blocks. Fig. 5.3 illustrates the setup for this experiment. The observed oil-rate is generated from a reference, randonly distributed pemeability field. In the second case the refernce permeability field is a homogeneous field and the gradients are computed based on the randomly distributed permeability field. This allows to assess the proper capture of fine-scale heterogeneities by the prolongation operator partial derivative w.r.t. to the grid-block permeabilities.

The expected behaviour of linearly decreasing error values as the perturbation size decreases is observed in the experiments, as shown in Fig. 5.4.

5.3.2. GRADIENT ACCURACY

The metric used to evaluate the quality of the MS gradient is the angle between fine-scale and MS normalized gradients

$$\alpha = \cos^{-1}\left(\nabla_\theta^T \hat{O}_{FS}\, \nabla_\theta \hat{O}_{MS}\right), \tag{5.68}$$

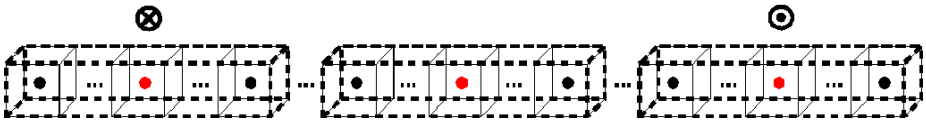


Figure 5.3: Fine and coarse grids and wells setup for 1D numerical experiments. The solid thin lines represent the fine grid-blocks. The bold dashed lines represent the primal-coarse grid blocks. Vertex cells are identified via red circles. The crossed circle represents the injection well and the dotted circle the production well.

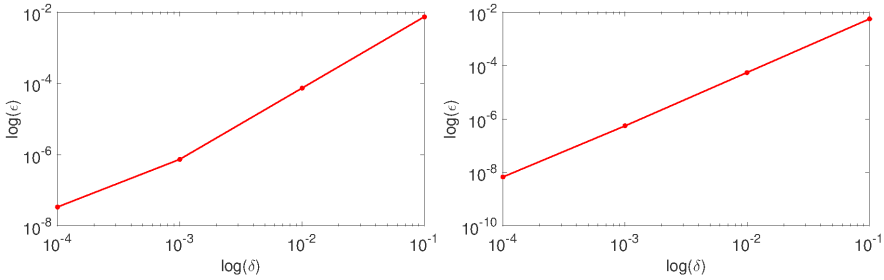


Figure 5.4: Validation of MS gradient computation method via comparison with numerical differentiation for the one-dimensional, (a) homogeneous and (b) heterogenous validation cases.

where

$$\nabla_\theta \hat{O}_{FS} = \frac{\nabla_\theta O_{FS}}{\|\nabla_\theta O_{FS}\|_2} \tag{5.69}$$

and

$$\nabla_\theta \hat{O}_{MS} = \frac{\nabla_\theta O_{MS}}{\|\nabla_\theta O_{MS}\|_2}. \tag{5.70}$$

Here, $\nabla_\theta O_{FS}$ is the gradient computed via a fine-scale IMPES adjoint and $\nabla_\theta O_{MS}$ denotes the MS adjoint gradient computed via the method developed in the present work. As a minimum requirement, acceptable MS gradients are obtained if α is much smaller than 90^o [44].

Both homogeneous and heterogeneous cases result in $\alpha = 0^o$, indicating that fine-scale and MS gradients are perfectly aligned in this case. This is due to the fact that, in 1D, no approximations (due to localization) are made in the MS solution, and thus, in the MS gradient computation. As exposed in [31], the same behaviour should not be expected in higher dimensions problem, where basis functions localization assumptions impact the quality of the gradient.

REFERENCES

[1] R. J. de Moraes, J. R. Rodrigues, H. Hajibeygi, and J. D. Jansen, *Multiscale gradient computation for flow in heterogeneous porous media,* Journal of Computational Physics **336**, 644 (2017).

[2] T. H. Michael, *Scientific computing: an introductory survey* (The McGraw-Hill Companies Inc.: New York, NY, USA, 2002).

[3] D. S. Oliver, A. C. Reynolds, and N. Liu, *Inverse theory for petroleum reservoir characterization and history matching* (Cambridge University Press, 2008).

[4] Y. Chen, D. S. Oliver, and D. Zhang, *Efficient ensemble-based closed-loop production optimization,* SPE J. **14**, 634 (2009).

[5] R. M. Fonseca, B. Chen, J. D. Jansen, and A. C. Reynolds, *A stochastic simplex approximate gradient (StoSAG) for optimization under uncertainty,* Int. J. Numer. Meth. Eng. (2016), 10.1002/nme.5342.

[6] A. S. Stordal, S. P. Szklarz, and O. Leeuwenburgh, *A theoretical look at ensemble-based optimization in reservoir management,* Mathematical Geosciences **48**, 399 (2016), http://dx.doi.org/10.1007/s11004-015-9598-6.

[7] J. R. P. Rodrigues, *Calculating derivatives for automatic history matching,* Computational Geosciences **10**, 119 (2006).

[8] J. F. B. M. Kraaijevanger, P. J. P. Egberts, J. R. Valstar, and H. W. Buurman, *Optimal waterflood design using the adjoint method,* in *SPE Reservoir Simulation Symposium* (Society of Petroleum Engineers, 2007).

[9] W. H. Chen, G. R. Gavalas, J. H. Seinfeld, and M. L. Wasserman, *A new algorithm for automatic history matching,* SPE J. **14**, 593 (1974).

[10] G. Chavent, M. Dupuy, and P. Lemmonier, *History matching by use of optimal theory,* Society of Petroleum Engineers Journal **15**, 74 (1975).

[11] W. F. Ramirez, *Application of optimal control theory to enhanced oil recovery*, Vol. 21 (Elsevier, 1987).

[12] J. D. Jansen, *Adjoint-based optimization of multi-phase flow through porous media–a review,* Computers & Fluids **46**, 40 (2011).

[13] R. Li, A. Reynolds, and D. Oliver, *History matching of three-phase flow production data,* SPEJ **8**, 328 (2003).

[14] J. F. van Doren, R. Markovinović, and J.-D. Jansen, *Reduced-order optimal control of water flooding using proper orthogonal decomposition,* Computational Geosciences **10**, 137 (2006), http://dx.doi.org/10.1007/s10596-005-9014-2.

[15] M. Cardoso and L. Durlofsky, *Linearized reduced-order models for subsurface flow simulation,* Journal of Computational Physics **229**, 681 (2010).

[16] J. D. Jansen and L. J. Durlofsky, *Use of reduced-order models in well control optimization,* Optim. Eng. , 1 (2016).

[17] P. Jenny, S. H. Lee, and H. A. Tchelepi, *Multi-scale finite-volume method for elliptic problems in subsurface flow simulation,* J. Comput. Phys. **187**, 47 (2003).

[18] T. Y. Hou and X.-H. Wu, *A multiscale finite element method for elliptic problems in composite materials and porous media,* J. Comput. Phys. **134**, 169 (1997).

[19] Y. Wang, H. Hajibeygi, and H. A. Tchelepi, *Algebraic multiscale solver for flow in heterogeneous porous media,* J. Comput. Phys. **259**, 284 (2014).

[20] H. Zhou and H. A. Tchelepi, *Operator-based multiscale method for compressible flow,* SPE J. **13**, 523 (2008).

[21] H. Hajibeygi, G. Bonfigli, M. A. Hesse, and P. Jenny, *Iterative multiscale finite-volume method,* J. Comput. Phys. **227**, 8604 (2008).

[22] Y. Efendiev and T. Y. Hou, *Multiscale finite element methods: theory and applications,* Vol. 4 (Springer Science & Business Media, 2009).

[23] O. Møyner and K.-A. Lie, *A multiscale restriction-smoothed basis method for high contrast porous media represented on unstructured grids,* Journal of Computational Physics **304**, 46 (2016).

[24] K. Lie, O. Møyner, J. Natvig, A. Kozlova, K. Bratvedt, S. Watanabe, and Z. Li, *Successful application of multiscale methods in a real reservoir simulator environment,* in *ECMOR XIV-15th European Conference on the Mathematics of Oil Recovery* (2016) amsterdam, The Netherlands, 29 August-1 September 2016. http://dx.doi.org/10.3997/2214-4609.201601893.

[25] J. Fu, H. A. Tchelepi, and J. Caers, *A multiscale adjoint method to compute sensitivity coefficients for flow in heterogeneous porous media,* Advances in water resources **33**, 698 (2010).

[26] J. Fu, J. Caers, and H. A. Tchelepi, *A multiscale method for subsurface inverse modeling: Single-phase transient flow,* Adv. Water Resour. **34**, 967 (2011).

[27] S. Krogstad, V. L. Hauge, and A. Gulbransen, *Adjoint multiscale mixed finite elements,* SPE Journal **16**, 162 (2011).

[28] J. E. Aarnes, V. L. Hauge, and Y. Efendiev, *Coarsening of three-dimensional structured and unstructured grids for subsurface flow,* Advances in Water Resources **30**, 2177 (2007).

[29] R. Moraes, R.-M. Fonseca, M. Helici, A. Heemink, and J. D. Jansen, *Improving the computational efficiency of approximate gradients using a multiscale reservoir simulation framework,* in *SPE Reservoir Simulation Symposium* (Society of Petroleum Engineers, 2017).

[30] G. Gao, J. Vink, C. Chen, Y. El Khamra, and M. Tarrahi, *Distributed gauss-newton method for history matching problems with multiple best matches,* in *ECMOR XV-15th European Conference on the Mathematics of Oil Recovery* (2016).

[31] R. Moraes, J. Rodrigues, H. Hajibeygi, and J. D. Jansen, *Multiscale gradient computation for subsurface flow models,* Journal of Computational Physics, (submitted 13 July 2016) (2016).

[32] M. Cusini, C. van Kruijsdijk, and H. Hajibeygi, *Algebraic dynamic multilevel (adm) method for fully implicit simulations of multiphase flow in porous media,* Journal of Computational Physics **314**, 60 (2016).

[33] H. Zhou and H. A. Tchelepi, *Two-stage algebraic multiscale linear solver for highly heterogeneous reservoir models,* SPE J. **17**, 523 (2012).

[34] P. Jenny, S. Lee, and H. Tchelepi, *Adaptive multiscale finite-volume method for multiphase flow and transport in porous media,* Multiscale Modeling & Simulation **3**, 50 (2005).

[35] J. D. Jansen, *Gradient-based optimization of flow through porous media, v.3,* (2016).

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

[37] H. Hajibeygi and P. Jenny, *Adaptive iterative multiscale finite volume method,* Journal of Computational Physics **230**, 628 (2011).

[38] R. J. de Moraes, J. R. Rodrigues, H. Hajibeygi, and J. D. Jansen, *Computing derivative information of sequentially coupled subsurface models,* Computational Geosciences (under review (June, 2018)).

[39] G. Corliss, *Automatic differentiation of algorithms: from simulation to optimization,* Vol. 1 (Springer Science & Business Media, 2002).

[40] R. Younis and K. Aziz, *Parallel automatically differentiable data-types for next-generation simulator development,* in *SPE Reservoir Simulation Symposium* (Society of Petroleum Engineers, 2007).

[41] Y. Zhou, H. A. Tchelepi, B. T. Mallison, *et al.*, *Automatic differentiation framework for compositional simulation on unstructured grids with multi-point discretization schemes,* in *SPE Reservoir Simulation Symposium* (Society of Petroleum Engineers, 2011).

[42] X. Li and D. Zhang, *A backward automatic differentiation framework for reservoir simulation,* Computational Geosciences **18**, 1009 (2014).

[43] S. H. Lee, H. Zhou, and H. A. Tchelepi, *Adaptive multiscale finite-volume method for nonlinear multiphase transport in heterogeneous formations,* Journal of Computational Physics **228**, 9036 (2009).

[44] R. M. Fonseca, S. S. Kahrobaei, L. J. T. Van Gastel, O. Leeuwenburgh, and J. D. Jansen, *Quantification of the impact of ensemble size on the quality of an ensemble gradient using principles of hypothesis testing,* in *SPE Reservoir Simulation Symposium* (Society of Petroleum Engineers, 2015).

6

ITERATIVE MULTISCALE GRADIENT COMPUTATION FOR HETEROGENEOUS SUBSURFACE FLOW

.

We introduce a semi-analytical iterative multiscale derivative computation methodology that allows for error control and reduction to any desired accuracy, up to fine-scale precision. The model responses are computed by the multiscale forward simulation of flow in heterogeneous porous media. The derivative computation method is based on the augmentation of the model equation and state vectors with the smoothing stage defined by the iterative multiscale method. In the formulation, we avoid additional complexity involved in computing partial derivatives associated to the smoothing step. We account for it as an approximate derivative computation stage. The numerical experiments illustrate how the newly introduced derivative method computes misfit objective function gradients that converge to fine-scale one as the iterative multiscale residual converges. The robustness of the methodology is investigated for test cases with high contrast permeability fields. The iterative multiscale gradient method casts a promissing approach, with minimal accuracy-efficiency tradeoff, for large-scale heterogeneous porous media optimization problems.

The material presented in this chapter has been submitted to Advances in Water Resources journal (*under review*)

Derivative computation is an important aspect of gradient-based optimization algorithms. When the objective (or cost) function evaluation involves the numerical simulation of discretized partial derivative equations (PDE), efficient gradient computation is of utmost importance. It is well documented in the literature that the most efficient and accurate way of computing derivatives are the analytical Direct [1–3] (if the number of parameters is greater than the number cost functionals) and Adjoint [2–7] (when the number of cost functionals is greater than the number of parameters) methods. However, even when considering efficient gradient methods, due to the necessity to evaluate the forward model and the derivative information many times, up until the optimality conditions are met, techniques to reduce the forward model simulation cost have been proposed [7–9].

Multiscale (MS) simulation methods [10, 11] have been increasingly employed for the efficient solution of elliptic [12] and parabolic [13] equations, more specifically subsurface flow problems in highly heterogeneous porous media. Also, many developments to extend its applicability to extended physics have been observed in the recent years [14].

Moreover, MS derivative computation strategies, based on MS forward simulation models, has also been subject of study. MS adjoint formulation (MS-ADJ) for single-phase subsurface flow have been presented in [15, 16]. MS adjoint computation methods for multiphase flow have also been developed [17, 18]. More recently, a mathematical framework for MS computation of derivative information has been developed [19]. It has been highlighted in [15, 19] that inaccurate MS gradient computations could lead to inaccurate gradient directions. However, as it is indicated in [19], strategies that improve the MS forward simulation solution (e.g. refinement of the MS coarse grid) result in better gradient estimates. Moreover, in [15] it is suggested that an iterative MS gradient computation strategy could resolve the multiscale gradient inaccuracies.

In this work, we develop an iterative multiscale gradient computation strategy which converges to the fine-scale gradient solution, thus allowing for error control and reduction for multiscale gradients. The derivative computation method is based on the generic mathematical framework introduced in [19]. By augmenting the MS model equation and the state vectors with the i-MSFV smoothing stage, the framework is capable of providing derivative information at any desired accuracy, up to fine-scale precision. The augmentation is addressed by the implicit differentiation strategy. In the formulation, we avoid additional complexity involved in computing partial derivatives associated to the smoothing step by also only approximately solving the derivative state equation associated to the it. Also, the strategy seamlessly accommodates both the Direct and Adjoint methods.

The remaining of this paper is organized as follows. Firstly, we present an algebraic formulation for the i-MSFV method, suitable for the derivation of the derivative computation methods. Next, we derive the Direct and Adjoint methods to compute derivative information, following the i-MSFV framework, when the algorithms are presented. Numerical validation of the method against numerical differentiation is presented. The numerical experiments conducted show that fine-scale gradient can be reproduced via the i-MSFV method if the forward simulation converges to a small enough residual tolerance. We show numerical evidences that there is a relationship between the gradient quality and the i-MSFV solution residual by comparing fine-scale gradient and i-MSFV

gradients and relating the difference between the two with the pressure error norm. Conclusion remarks are finally presented in the last section.

6.1. ALGEBRAIC AND ALGORITHMIC DESCRIPTION OF THE MULTISCALE ITERATIVE METHOD

We consider the set of equations that algebraically describes the forward simulation at the fine scale, without any assumption regarding the underlying physical model, as [19]

$$\mathbf{g}_F(\mathbf{x},\theta)=\mathbf{0}, \tag{6.1}$$

where $\mathbf{g}_F:\mathbb{R}^{N_F}\times\mathbb{R}^{N_\theta}\rightarrow\mathbb{R}^{N_F}$ represents the set of algebraic forward model equations, $\mathbf{x}\in\mathbb{R}^{N_F}$ is the state vector (which, for single-phase flow, contains the grid block pressures), $\theta\in\mathbb{R}^{N_\theta}$ is the vector of parameters, and the subscript F refers to 'fine scale'. There are N_F fine-scale cells and N_θ parameters. Eq. (7.10) implicitly assumes a dependency of the state vector $\mathbf{x}$ on the parameters θ, i.e.

$$\mathbf{x}=\mathbf{x}(\theta). \tag{6.2}$$

Once the model state is determined, the observable responses of the forward model are computed. The forward model responses may not only depend on the model state, but also on the parameters themselves, and can be expressed as

$$\mathbf{y}_F=\mathbf{h}_F(\mathbf{x},\theta), \tag{6.3}$$

where $\mathbf{h}_F:\mathbb{R}^{N_F}\times\mathbb{R}^{N_\theta}\rightarrow\mathbb{R}^{N_y}$ represents the output equations [20]. It is assumed that $\mathbf{g}_F$ can be described as

$$\mathbf{g}_F(\mathbf{x},\theta)=\mathbf{A}(\theta)\mathbf{x}-\mathbf{q}(\theta), \tag{6.4}$$

where $\mathbf{A}(\theta)\in\mathbb{R}^{N_F}\times\mathbb{R}^{N_F}$ matrix and $\mathbf{q}(\theta)\in\mathbb{R}^{N_F}$.

A two-stage multiscale (MS) solution strategy [10, 21] can be devised by firstly computing a coarse scale solution

$$\check{\mathbf{g}}(\check{\mathbf{x}},\theta)=(\mathbf{RAP})\check{\mathbf{x}}-(\mathbf{Rq})=\check{\mathbf{A}}\check{\mathbf{x}}-\check{\mathbf{q}}=\check{\mathbf{0}}, \tag{6.5}$$

$\check{\mathbf{g}}:\mathbb{R}^{N_C}\times\mathbb{R}^{N_\theta}\rightarrow\mathbb{R}^{N_C}$, where N_C is the number of coarse grid-blocks, and then an approximated fine-scale solution

$$\mathbf{g}'(\mathbf{x}',\check{\mathbf{x}},\theta)=\mathbf{x}'-\mathbf{P}\check{\mathbf{x}}=\mathbf{0}, \tag{6.6}$$

where $\mathbf{g}':\mathbb{R}^{N_F}\times\mathbb{R}^{N_C}\times\mathbb{R}^{N_\theta}\rightarrow\mathbb{R}^{N_F}$.

The so-called prolongation operator $\mathbf{P}=\mathbf{P}(\theta)$ which is an $N_F\times N_C$ matrix that maps (interpolates) the coarse-scale solution to the fine-scale resolution. The so-called restriction operator $\mathbf{R}=\mathbf{R}(\theta)$ is defined as an $N_C\times N_F$ matrix which maps the fine scale to the coarse scale. More information about how these operators are constructed for the Multiscale Finite Volume (MSFV) method can be found in [12, 21]. Let $\check{\mathbf{x}}\in\mathbb{R}^{N_C}$ be the coarse scale solution ($N_C\ll N_F$), and $\mathbf{x}'\in\mathbb{R}^{N_F}$ the approximated fine-scale solution.

The iterative multiscale strategy [22] can be devised by considering versions of Eq. (7.13) and Eq. (7.14) written in residual form. Let $\mathbf{x}^{\nu-1}$ be an approximate solution to Eq. (7.13) at iteration $\nu-1$ and

$$\mathbf{r}^{\nu-1}=\mathbf{q}-\mathbf{A}\mathbf{x}^{\nu-1} \tag{6.7}$$

be the corresponding residual. A multiscale improvement to this approximation can be devised by writing Eq. (7.14) in residual form as

$$\check{\mathbf{g}}^{\nu}\left(\check{\mathbf{x}}^{\nu},\mathbf{x}^{\nu-1},\theta\right)=\check{\mathbf{A}}\check{\mathbf{x}}^{\nu}-\check{\mathbf{r}}^{\nu-1}=\check{\mathbf{0}}, \tag{6.8}$$

where

$$\check{\mathbf{r}}^{\nu-1}=\mathbf{R}\mathbf{r}^{\nu-1}, \tag{6.9}$$

$\check{\mathbf{r}}^{\nu-1}\in\mathbb{R}^{N_C}$. Here, $\check{\mathbf{x}}^{\nu}$ is redefined as the coarse scale correction. Redefining Eq. (7.15), we have

$$\mathbf{g}'^{\nu}\left(\mathbf{x}'^{\nu},\check{\mathbf{x}}^{\nu},\theta\right)=\mathbf{x}'^{\nu}-\mathbf{P}\check{\mathbf{x}}^{\nu}=\mathbf{0}, \tag{6.10}$$

such that $\mathbf{x}'^{\nu}$ now represents the approximate fine-scale correction at iteration ν, i.e.,

$$\mathbf{x}^{\nu-1/2}=\mathbf{x}^{\nu-1}+\mathbf{x}'^{\nu} \tag{6.11}$$

is an approximate solution of Eq. (7.13) augmented with the correction from the coarse-scale calculation.

The approximate solution provided by Eq. (6.11) can be improved if successive smoothing steps are employed [22]. Let

$$\mathbf{r}_{\sigma}^{\nu-1}=\mathbf{q}-\mathbf{A}\mathbf{x}^{\nu-1/2}=\mathbf{q}-\mathbf{A}\left(\mathbf{x}^{\nu-1}+\mathbf{x}'^{\nu}\right), \tag{6.12}$$

$\mathbf{r}_{\sigma}^{\nu-1}\in\mathbb{R}^{N_F}$, be the smoothed residual obtained from the approximation given by Eq. (6.11) and

$$\mathbf{g}_{\sigma}^{\nu}\left(\mathbf{x}'^{\nu},\mathbf{x}_{\sigma}^{\nu},\mathbf{x}^{\nu-1},\theta\right)=\mathbf{A}\mathbf{x}_{\sigma}^{\nu}-\mathbf{r}_{\sigma}^{\nu-1}=\mathbf{0}, \tag{6.13}$$

a version of Eq. (7.13) written in residual form. Here $\mathbf{x}_{\sigma}^{\nu}\in\mathbb{R}^{N_F}$ is the smoothed fine-scale correction at iteration ν. The solution smoothing is obtained by solving Eq. (6.13) using any iterative solver up to a prescribed (loose) tolerance or (small) maximum number of iterations [13, 22]. The solution for a given iteration ν is, hence, obtained from

$$\mathbf{g}_{x}^{\nu}\left(\mathbf{x}^{\nu},\mathbf{x}'^{\nu},\mathbf{x}_{\sigma}^{\nu},\mathbf{x}^{\nu-1},\theta\right)=\mathbf{x}^{\nu}-\mathbf{x}^{\nu-1}-\mathbf{x}'^{\nu}-\mathbf{x}_{\sigma}^{\nu}=\mathbf{0}, \tag{6.14}$$

where $\mathbf{g}_{x}^{\nu}:\mathbb{R}^{N_F}\times\mathbb{R}^{N_F}\times\mathbb{R}^{N_F}\times\mathbb{R}^{N_F}\times\mathbb{R}^{N_\theta}\rightarrow\mathbb{R}^{N_F}$.

The MS iterative strategy is fully depicted in Algorithm 11, where $\|\,.\,\|$ represents the 2-norm.

In Algorithm 11, ϵ and ϵ_{σ} are, respectively, the user-defined tolerances for the outer-loop and smoothing step. That allows to control the smoothing step as a relative improvement starting from the MS solution approximate solution. An investigation of an optimal relationship between the number of outer loops and the number of smoothing steps is presented in [13].

6.2. ITERATIVE MULTISCALE GRADIENT COMPUTATION

For the developments that will follow in this section, it is convenient to write the set of equations that is solved in every iteration, namely Eq. (6.8), Eq. (6.10), Eq. (6.13) and Eq.

Algorithm 11: Iterative Multiscale Method [22].

Input : $\mathbf{A}, \mathbf{R}, \mathbf{P}, \mathbf{q}, \epsilon, \epsilon_\sigma$
Output: Approximate solution for the linear system $\mathbf{Ax} = \mathbf{q}$

Set $\mathbf{x}^0 = \mathbf{0}$
for $\nu = 1, 2, \ldots$ **do**
  Compute $\mathbf{r}^{\nu-1} = \mathbf{q} - \mathbf{A}\mathbf{x}^{\nu-1}$
  If $\frac{\| \mathbf{r}^{\nu-1} \|}{\| \mathbf{r}^0 \|} < \epsilon$, quit with solution given by $\mathbf{x} = \mathbf{x}^{\nu-1}$
  Solve $\check{\mathbf{x}}^\nu = \check{\mathbf{A}}^{-1} \left(\mathbf{R}\mathbf{r}^{\nu-1} \right)$
  Compute $\mathbf{x}'^\nu = \mathbf{P}\check{\mathbf{x}}^\nu$
  Compute $\mathbf{r}_\sigma^{\nu-1} = \mathbf{q} - \mathbf{A}\left(\mathbf{x}^{\nu-1} + \mathbf{x}'^\nu \right)$
  Iteratively solve $\mathbf{x}_\sigma^\nu = \mathbf{A}^{-1}\mathbf{r}_\sigma^{\nu-1}$ until $\frac{\| \mathbf{r}_\sigma^{\nu-1} - \mathbf{A}\mathbf{x}_\sigma^\nu \|}{\| \mathbf{r}_\sigma^{\nu-1} \|} < \epsilon_\sigma$
  Update $\mathbf{x}^\nu = \mathbf{x}^{\nu-1} + \mathbf{x}'^\nu + \mathbf{x}_\sigma^\nu$

(6.14), in matrix form as

$$\left(\begin{array}{cccc|cccc} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{RA} & \check{\mathbf{A}} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{P} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A} & \mathbf{0} & \mathbf{A} & \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I} & \mathbf{0} & -\mathbf{I} & -\mathbf{I} & \mathbf{I} \end{array}\right) \times \left(\begin{array}{c} \check{\mathbf{x}}^{\nu-1} \\ \mathbf{x}'^{\nu-1} \\ \mathbf{x}_\sigma^{\nu-1} \\ \mathbf{x}^{\nu-1} \\ \hline \check{\mathbf{x}}^\nu \\ \mathbf{x}'^\nu \\ \mathbf{x}_\sigma^\nu \\ \mathbf{x}^\nu \end{array}\right) - \left(\begin{array}{c} \mathbf{Rq} \\ \mathbf{0} \\ \mathbf{q} \\ \mathbf{0} \end{array}\right) = \mathbf{0} \tag{6.15}$$

One must note, however, that, in the i-MSFV procedure, Eq. (6.13) is solved only approximately and, therefore, stricly speaking the equation in the third row of Eq. (6.15) does not hold. The idea here is to describe the procedure in an algebraic manner, ignoring this approximation, in order to facilitate the presentation of the derivative calculation algorithms in the next section. Once the derivative calculation methods are obtained under the assumption that the algebraic relations in Eq. (6.15) hold, the same type of smoothing approach employed in the i-MSFV to resolve high frequency errors will be used in the solution of the derivative information. This results in a practical semi-analytical algorithm for derivative calculation in an iterative formulation. Note that a fully analytical procedure would require calculating the derivative of the smoothing ope-

rator employed in the i-MSFV (step 8 in Algorithm 11), which can be quite complex, due to its nonlinear character [23]. The proposed semi-analytical approach also becomes general and applicable to any iterative procedure used in step 8, regardless of its nature. A truly analytical derivative method would require the implementation of derivative calculation for each iterative procedure used. More details on the implication of this assumption are discussed in 12.

Considering all equations that must be solved for all iterations $\nu \in \{1,\dots,N_\nu\}$, they can be collated in a super-vector [2, 6, 20] fashion as

$$\boldsymbol{g}(\boldsymbol{x},\theta) = \begin{pmatrix} \check{\mathbf{g}}^1\left(\check{\mathbf{x}}^1,\mathbf{x}'^0,\mathbf{x}_\sigma^0,\theta\right) \\ \mathbf{g}'^1\left(\mathbf{x}'^1,\check{\mathbf{x}}^1,\theta\right) \\ \mathbf{g}_\sigma^1\left(\mathbf{x}_\sigma^1,\mathbf{x}'^1,\mathbf{x}'^0,\mathbf{x}_\sigma^0,\theta\right) \\ \mathbf{g}_x^1\left(\mathbf{x}^1,\mathbf{x}_\sigma^1,\mathbf{x}'^1,\mathbf{x}^0,\theta\right) \\ \check{\mathbf{g}}^2\left(\check{\mathbf{x}}^2,\mathbf{x}'^1,\mathbf{x}_\sigma^1,\theta\right) \\ \mathbf{g}'^2\left(\mathbf{x}'^2,\check{\mathbf{x}}^2,\theta\right) \\ \mathbf{g}_\sigma^2\left(\mathbf{x}_\sigma^2,\mathbf{x}'^2,\mathbf{x}'^1,\mathbf{x}_\sigma^1,\theta\right) \\ \mathbf{g}_x^2\left(\mathbf{x}^2,\mathbf{x}_\sigma^2,\mathbf{x}'^2,\mathbf{x}^1,\theta\right) \\ \vdots \\ \check{\mathbf{g}}^{N_\nu}\left(\check{\mathbf{x}}^{N_\nu},\mathbf{x}'^{N_\nu-1},\mathbf{x}_\sigma^{N_\nu-1},\theta\right) \\ \mathbf{g}'^{N_\nu}\left(\mathbf{x}'^{N_\nu},\check{\mathbf{x}}^{N_\nu},\theta\right) \\ \mathbf{g}_\sigma^{N_\nu}\left(\mathbf{x}_\sigma^{N_\nu},\mathbf{x}'^{N_\nu},\mathbf{x}'^{N_\nu-1},\mathbf{x}_\sigma^{N_\nu-1},\theta\right) \\ \mathbf{g}_x^{\nu}\left(\mathbf{x}^{N_\nu},\mathbf{x}_\sigma^{N_\nu},\mathbf{x}'^{N_\nu},\mathbf{x}^{N_\nu-1},\theta\right) \end{pmatrix} = \mathbf{0}, \tag{6.16}$$

and the super-state vector defined as

$$\boldsymbol{x}(\theta) = \begin{pmatrix} \begin{pmatrix} \check{\mathbf{x}}^1 \\ \mathbf{x}'^1 \\ \mathbf{x}_\sigma^1 \\ \mathbf{x}^1 \end{pmatrix}^T & \cdots & \begin{pmatrix} \check{\mathbf{x}}^{N_\nu-1} \\ \mathbf{x}'^{N_\nu-1} \\ \mathbf{x}_\sigma^{N_\nu-1} \\ \mathbf{x}^{N_\nu-1} \end{pmatrix}^T & \begin{pmatrix} \check{\mathbf{x}}^{N_\nu} \\ \mathbf{x}'^{N_\nu} \\ \mathbf{x}_\sigma^{N_\nu} \\ \mathbf{x}^{N_\nu} \end{pmatrix}^T \end{pmatrix}^T. \tag{6.17}$$

Note that we use bold-italic in the notation to represent super-vectors.

It is discussed in [2, 19, 24] how any derivative information can be efficiently computed from the pre and post multiplication of the sensitivity matrix $\mathbf{G}$, $\mathbf{G} \in \mathbb{R}^{N_y \times N_\theta}$, by arbitrary matrices as

$$\mathbf{WGV} = -\mathbf{W}\frac{\partial \mathbf{h}}{\partial \boldsymbol{x}}\left(\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\right)^{-1}\frac{\partial \boldsymbol{g}}{\partial \theta}\mathbf{V} + \mathbf{W}\frac{\partial \mathbf{h}}{\partial \theta}\mathbf{V}, \tag{6.18}$$

where $\mathbf{V}$ (of order $N_\theta \times p$) and $\mathbf{W}$ (of order $m \times N_y$) are arbitrary matrices, defined based on the derivative information one wants to obtain.

Eq. (6.18) requires the partial derivative of the model equations w.r.t. the parameters. From Eq. (7.14) and Eq. (7.15), as discussed in [19], it follows that

$$\frac{\partial \check{\mathbf{g}}^{\nu}}{\partial \theta} = \left[\frac{\partial \mathbf{R}}{\partial \theta}(\mathbf{AP}) + \mathbf{R}\frac{\partial \mathbf{A}}{\partial \theta}\mathbf{P} + (\mathbf{RA})\frac{\partial \mathbf{P}}{\partial \theta}\right]\check{\mathbf{x}}^{\nu} - \frac{\partial \mathbf{R}}{\partial \theta}\mathbf{q} + \\ -\mathbf{R}\frac{\partial \mathbf{q}}{\partial \theta} + \left(\frac{\partial \mathbf{R}}{\partial \theta}\mathbf{A} + \mathbf{R}\frac{\partial \mathbf{A}}{\partial \theta}\right)\mathbf{x}^{\nu-1}, \tag{6.19}$$

$$\frac{\partial \mathbf{g}'^{\nu}}{\partial \theta} = -\frac{\partial \mathbf{P}}{\partial \theta}\check{\mathbf{x}}^{\nu}, \tag{6.20}$$

and deriving Eq. (6.13) w.r.t. θ

$$\frac{\partial \mathbf{g}_{\sigma}^{\nu}}{\partial \theta} = \frac{\partial \mathbf{A}}{\partial \theta}\mathbf{x}_{\sigma}^{\nu} + \frac{\partial \mathbf{A}}{\partial \theta}\mathbf{x}'^{\nu} - \frac{\partial \mathbf{q}}{\partial \theta} + \frac{\partial \mathbf{A}}{\partial \theta}\mathbf{x}^{\nu-1}. \tag{6.21}$$

Also, from Eq. (6.14), it follows that

$$\frac{\partial \mathbf{g}_{x}^{\nu}}{\partial \theta} = \mathbf{0} \tag{6.22}$$

For the sake of simplicity and in coherence with the MSFV method used in the numerical experiments, the dependency of $\mathbf{R}$ on θ is neglected.

Now, the order the operations in Eq. (6.18) are evaluated define the Direct and Adjoint methods. The derivation of both methods will be discussed in the next sections.

6.2.1. THE DIRECT METHOD

If $\mathbf{W}$ is factored out in Eq. (6.18), it can be rewritten as

$$\mathbf{GV} = \frac{\partial \mathbf{h}}{\partial \boldsymbol{x}}\mathbf{Z} + \frac{\partial \mathbf{h}}{\partial \theta}\mathbf{V}, \tag{6.23}$$

where

$$\mathbf{Z} = -\left(\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\right)^{-1}\frac{\partial \boldsymbol{g}}{\partial \theta}\mathbf{V}, \tag{6.24}$$

is solved from

$$\left(\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\right)\mathbf{Z} = -\frac{\partial \boldsymbol{g}}{\partial \theta}\mathbf{V}. \tag{6.25}$$

The linear system described in Eq. (6.25) can be re-written in a block-wise fashion for each iteration ν:

$$\left(\begin{array}{c|c|c|c} \frac{\partial \mathbf{g}^0}{\partial \boldsymbol{x}^0} & & & \\ \hline \frac{\partial \mathbf{g}^1}{\partial \boldsymbol{x}^0} & \frac{\partial \mathbf{g}^1}{\partial \boldsymbol{x}^1} & & \\ \hline & \ddots & \ddots & \\ \hline & & \frac{\partial \mathbf{g}^{N_v}}{\partial \boldsymbol{x}^{N_v-1}} & \frac{\partial \mathbf{g}^{N_v}}{\partial \boldsymbol{x}^{N_v}} \end{array}\right)\left(\begin{array}{c} \mathbf{Z}^0 \\ \hline \mathbf{Z}^1 \\ \hline \vdots \\ \hline \mathbf{Z}^{N_v} \end{array}\right) = -\left(\begin{array}{c} \frac{\partial \mathbf{g}^0}{\partial \theta}\mathbf{V} \\ \hline \frac{\partial \mathbf{g}^1}{\partial \theta}\mathbf{V} \\ \hline \vdots \\ \hline \frac{\partial \mathbf{g}^{N_v}}{\partial \theta}\mathbf{V} \end{array}\right), \tag{6.26}$$

where, from Eq. (6.8), Eq. (6.10), and Eq. (6.13)

$$\frac{\partial \mathbf{g}^{\nu}}{\partial \boldsymbol{x}^{\nu}} = \begin{pmatrix} \check{\mathbf{A}} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -\mathbf{P} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A} & \mathbf{A} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I} & -\mathbf{I} & \mathbf{I} \end{pmatrix}, \tag{6.27}$$

and

$$\frac{\partial \mathbf{g}^{\nu}}{\partial \boldsymbol{x}^{\nu-1}} = \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{RA} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I} \end{pmatrix}. \tag{6.28}$$

The partitioning lines indicate which matrix and vector terms belong to each iteration.

Substituting Eq. (6.27) and Eq. (6.28) in Eq. (6.26), follows that

$$\left(\begin{array}{cccc|cccc|c|cccc} \check{\mathbf{A}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & & & & & & & & \\ -\mathbf{P} & \mathbf{I} & \mathbf{0} & \mathbf{0} & & & & & & & & \\ \mathbf{0} & \mathbf{A} & \mathbf{A} & \mathbf{0} & & & & & & & & \\ \mathbf{0} & -\mathbf{I} & -\mathbf{I} & \mathbf{I} & & & & & & & & \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{RA} & \check{\mathbf{A}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & & & & \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{P} & \mathbf{I} & \mathbf{0} & \mathbf{0} & & & & \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A} & \mathbf{0} & \mathbf{A} & \mathbf{A} & \mathbf{0} & & & & \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I} & \mathbf{0} & -\mathbf{I} & -\mathbf{I} & \mathbf{I} & & & & \\ \hline & & & & \ddots & & & & \ddots & & & \\ \hline & & & & & & & & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{RA} \\ & & & & & & & & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ & & & & & & & & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A} \\ & & & & & & & & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I} \end{array}\right) \times$$

$$\left(\check{\mathbf{Z}}^0 \quad \mathbf{Z}'^0 \quad \mathbf{Z}_{\sigma}^0 \quad \mathbf{Z}_x^0 \middle| \check{\mathbf{Z}}^1 \quad \mathbf{Z}'^1 \quad \mathbf{Z}_{\sigma}^1 \quad \mathbf{Z}_x^1 \middle| \cdots \middle| \check{\mathbf{Z}}^{N_v} \quad \mathbf{Z}'^{N_v} \quad \mathbf{Z}_{\sigma}^{N_v} \quad \mathbf{Z}_x^{N_v}\right)^T =$$

$$-\left(\frac{\partial \check{\mathbf{g}}^0}{\partial \theta}\mathbf{V} \quad \frac{\partial \mathbf{g}'^0}{\partial \theta}\mathbf{V} \quad \frac{\partial \mathbf{g}_{\sigma}^0}{\partial \theta}\mathbf{V} \quad \mathbf{0} \middle| \frac{\partial \check{\mathbf{g}}^1}{\partial \theta}\mathbf{V} \quad \frac{\partial \mathbf{g}'^1}{\partial \theta}\mathbf{V} \quad \frac{\partial \mathbf{g}_{\sigma}^1}{\partial \theta}\mathbf{V} \quad \mathbf{0} \middle| \cdots \middle| \frac{\partial \check{\mathbf{g}}^{N_v}}{\partial \theta}\mathbf{V} \quad \frac{\partial \mathbf{g}'^{N_v}}{\partial \theta}\mathbf{V} \quad \frac{\partial \mathbf{g}_{\sigma}^{N_v}}{\partial \theta}\mathbf{V} \quad \mathbf{0}\right)^T, \tag{6.29}$$

For every iteration, the linear system that must be solved for the coarse-scale equation is

$$\check{\mathbf{Z}}^{\nu} = \check{\mathbf{A}}^{-1}\left(-\frac{\partial \check{\mathbf{g}}^{\nu}}{\partial \theta}\mathbf{V} - \mathbf{R}\mathbf{A}\mathbf{Z}_x^{\nu-1}\right), \tag{6.30}$$

for the fine-scale approximate solution equation

$$\mathbf{Z}'^{\nu} = -\mathbf{P}\check{\mathbf{Z}}^{\nu} - \frac{\partial \mathbf{g}'^{\nu}}{\partial \theta}\mathbf{V}, \tag{6.31}$$

and for the smoothing equation

$$\mathbf{Z}_{\sigma}^{\nu} = \mathbf{A}^{-1}\left(-\frac{\partial \mathbf{g}_{\sigma}^{\nu}}{\partial \theta}\mathbf{V} - \mathbf{A}\mathbf{Z}'^{\nu} - \mathbf{A}\mathbf{Z}_x^{\nu-1}\right). \tag{6.32}$$

As pointed out above, it would not be feasible to fully solve Eq. (6.32). In order to obtain a practical derivative calculation method, the same smoothing procedure used in the i-MSFV is applied here, i.e., $\mathbf{Z}_\sigma^\nu$ is obtained by solving Eq. (6.32) using any iterative solver up to a prescribed (loose) tolerance or a (small) maximum number of iterations.

Finally,

$$\mathbf{Z}_x^\nu = \mathbf{Z}_x^{\nu-1} + \mathbf{Z}'^\nu + \mathbf{Z}_\sigma^\nu. \tag{6.33}$$

Observing that $\mathbf{h}$ only depends on $\mathbf{x} = \mathbf{x}^{N_\nu}$, Eq. (6.23) can be simplified to

$$\mathbf{GV} = \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \mathbf{Z}_x^{N_\nu} + \frac{\partial \mathbf{h}}{\partial \theta} \mathbf{V}. \tag{6.34}$$

Now the Direct method for the iterative multiscale gradient computation can be fully determined. It is depicted in Algorithm 12. Note that references to superscript -1 correspond to zero terms.

Algorithm 12: Post multiplication of $\mathbf{G}$ by $\mathbf{V}$ via the Direct method.

Input : $\mathbf{R}$, $\mathbf{A}$, $\mathbf{P}$, $\boldsymbol{x}$, $\frac{\partial \mathbf{A}}{\partial \theta}$, $\frac{\partial \mathbf{h}}{\partial \mathbf{x}}$, $\frac{\partial \mathbf{h}}{\partial \theta}$, $\mathbf{V}$, ϵ_σ
Output: The $\mathbf{GV}$ product

foreach $j = 1, 2, \ldots, n$ **do**
  foreach $\nu = 0, 1, \ldots, N_\nu$ **do**
    Compute $\boldsymbol{\beta} = \frac{\partial \mathbf{P}}{\partial \theta} \check{\mathbf{x}}^\nu \mathbf{V}_{.,j}$; // Algorithm 3, in [19] where $\mathbf{m} = \mathbf{V}_{.,j}$
    Compute $\boldsymbol{\alpha} = \mathbf{RA}\boldsymbol{\beta} + \mathbf{R}\left(\frac{\partial \mathbf{A}}{\partial \theta}\mathbf{x}^{\nu-1} - \frac{\partial \mathbf{q}}{\partial \theta} + \frac{\partial \mathbf{A}}{\partial \theta}\mathbf{P}\check{\mathbf{x}}^\nu\right)\mathbf{V}_{.,j}$
    Solve $\check{\mathbf{Z}}_{.,j}^\nu = \check{\mathbf{A}}^{-1}\left(-\boldsymbol{\alpha} - \mathbf{RA}\mathbf{Z}_{x}{}_{.,j}^{\nu-1}\right)$
    Compute $\mathbf{Z}'^\nu_{.,j} = -\mathbf{P}\check{\mathbf{Z}}_{.,j}^\nu + \boldsymbol{\beta}$
    Compute $\frac{\partial \mathbf{g}_\sigma^\nu}{\partial \theta} = \frac{\partial \mathbf{A}}{\partial \theta}\mathbf{x}_\sigma^\nu + \frac{\partial \mathbf{A}}{\partial \theta}\mathbf{x}'^\nu - \frac{\partial \mathbf{q}}{\partial \theta} + \frac{\partial \mathbf{A}}{\partial \theta}\mathbf{x}^{\nu-1}$
    Compute $\delta = \left(-\frac{\partial \mathbf{g}_\sigma^\nu}{\partial \theta}\mathbf{V}_{.,j} - \mathbf{A}\mathbf{Z}'^\nu_{.,j} - \mathbf{A}\mathbf{Z}_{x}{}_{.,j}^{\nu-1}\right)$
    Iteratively solve $\mathbf{Z}_\sigma{}_{.,j}^{\nu} = \mathbf{A}^{-1}\delta$ until $\frac{\| \delta - \mathbf{A}\mathbf{Z}_\sigma{}_{.,j}^{\nu} \|}{\| \delta \|} < \epsilon_\sigma$
    Compute $\mathbf{Z}_x{}_{.,j}^{\nu} = \mathbf{Z}_x{}_{.,j}^{\nu-1} + \mathbf{Z}'^\nu_{.,j} + \mathbf{Z}_\sigma{}_{.,j}^{\nu}$
  Compute $(\mathbf{GV})_{.,j} = \frac{\partial \mathbf{h}}{\partial \mathbf{x}}\mathbf{Z}_x{}_{.,j}^{N_\nu} + \frac{\partial \mathbf{h}}{\partial \theta}\mathbf{V}_{.,j}$

6.2.2. THE ADJOINT METHOD

Now, if $\mathbf{V}$ is factored out in Eq. (6.18), it can be rewritten as

$$\mathbf{WG} = \mathbf{Z}\frac{\partial \boldsymbol{g}}{\partial \theta} + \mathbf{W}\frac{\partial \mathbf{h}}{\partial \theta}, \tag{6.35}$$

where

$$\mathbf{Z} = -\mathbf{W}\frac{\partial \mathbf{h}}{\partial \boldsymbol{x}}\left(\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\right)^{-1} \tag{6.36}$$

is solved from

$$\mathbf{Z}\left(\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\right) = -\mathbf{W}\frac{\partial \mathbf{h}}{\partial \boldsymbol{x}}. \tag{6.37}$$

The linear system described in Eq. (6.37) can be re-written in a block-wise fashion for each iteration ν as

$$\begin{array}{c} \left(\mathbf{Z}^0 \;\middle|\; \mathbf{Z}^1 \;\middle|\; \dots \;\middle|\; \mathbf{Z}^{N_\nu}\right) \times \\ \left(\begin{array}{c|c|c|c} \frac{\partial \mathbf{g}^0}{\partial \boldsymbol{x}^0} & & & \\ \hline \frac{\partial \mathbf{g}^1}{\partial \boldsymbol{x}^0} & \frac{\partial \mathbf{g}^1}{\partial \boldsymbol{x}^1} & & \\ \hline & \ddots & \ddots & \\ \hline & & \frac{\partial \mathbf{g}^{N_\nu}}{\partial \boldsymbol{x}^{N_\nu-1}} & \frac{\partial \mathbf{g}^{N_\nu}}{\partial \boldsymbol{x}^{N_\nu}} \end{array}\right) \end{array} = -\mathbf{W}\frac{\partial \mathbf{h}}{\partial \boldsymbol{x}} \tag{6.38}$$

The structure of Eq. (6.38) shows that it can be solved via back substitution, i.e. the solution of the gradient information is backward in the iterations.

Substituting Eq. (6.27) and Eq. (6.28) in Eq. (6.38), follows that

$$\begin{array}{c} \left(\check{\mathbf{Z}}^0 \;\; \mathbf{Z}'^0 \;\; \mathbf{Z}_\sigma^0 \;\; \mathbf{Z}_x^0 \;\middle|\; \check{\mathbf{Z}}^1 \;\; \mathbf{Z}'^1 \;\; \mathbf{Z}_\sigma^1 \;\; \mathbf{Z}_x^1 \;\middle|\; \dots \;\middle|\; \check{\mathbf{Z}}^{N_\nu} \;\; \mathbf{Z}'^{N_\nu} \;\; \mathbf{Z}_\sigma^{N_\nu} \;\; \mathbf{Z}_x^{N_\nu}\right) \times \\ \left(\begin{array}{c|c|c} \begin{array}{cccc} \check{\mathbf{A}} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -\mathbf{P} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A} & \mathbf{A} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I} & -\mathbf{I} & \mathbf{I} \end{array} & & \\ \hline \begin{array}{cccc} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{RA} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I} \end{array} & \begin{array}{cccc} \check{\mathbf{A}} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -\mathbf{P} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A} & \mathbf{A} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I} & -\mathbf{I} & \mathbf{I} \end{array} & \\ \hline & \ddots & \ddots \\ \hline & & \begin{array}{cccc} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{RA} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I} \end{array} & \begin{array}{cccc} \check{\mathbf{A}} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -\mathbf{P} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A} & \mathbf{A} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I} & -\mathbf{I} & \mathbf{I} \end{array} \end{array}\right) \end{array} = \\ -\mathbf{W}\frac{\partial \mathbf{h}}{\partial \boldsymbol{x}} \end{array} \tag{6.39}$$

From Eq. (6.39) and recalling that $\mathbf{h}$ only depends on $\mathbf{x} = \mathbf{x}^{N_v}$, the (transposed) linear system that must be solved to compute $\mathbf{Z}^v$ is given by

$$\left(\begin{array}{cccc|cccc} \check{\mathbf{A}}^T & -\mathbf{P}^T & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{A}^T & -\mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}^T & -\mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{A}^T\mathbf{R}^T & \mathbf{0} & \mathbf{A}^T & -\mathbf{I} \end{array}\right) \times \left(\begin{array}{c} \left(\check{\mathbf{Z}}^v\right)^T \\ \left(\mathbf{Z}'^v\right)^T \\ \left(\mathbf{Z}_\sigma^v\right)^T \\ \left(\mathbf{Z}_x^v\right)^T \\ \hline \left(\check{\mathbf{Z}}^{v+1}\right)^T \\ \left(\mathbf{Z}'^{v+1}\right)^T \\ \left(\mathbf{Z}_\sigma^{v+1}\right)^T \\ \left(\mathbf{Z}_x^{v+1}\right)^T \end{array}\right) = \mathbf{0}, \tag{6.40}$$

for $v \neq N_v$, while $\mathbf{Z}^{N_v}$ is calculated from

$$\begin{pmatrix} \check{\mathbf{A}}^T & -\mathbf{P}^T & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{A}^T & -\mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}^T & -\mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{pmatrix} \times \begin{pmatrix} \left(\check{\mathbf{Z}}^{N_v}\right)^T \\ \left(\mathbf{Z}'^{N_v}\right)^T \\ \left(\mathbf{Z}_\sigma^{N_v}\right)^T \\ \left(\mathbf{Z}_x^{N_v}\right)^T \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ -\left(\dfrac{\partial \mathbf{h}}{\partial \mathbf{x}}\right)^T \mathbf{W}^T \end{pmatrix}. \tag{6.41}$$

The equation that computes the "adjoint state" associated with $\mathbf{g}_x^v$ reads

$$\left(\mathbf{Z}_x^v\right)^T = \left(\mathbf{Z}_x^{v-1}\right)^T - \mathbf{A}^T\mathbf{R}^T\left(\check{\mathbf{Z}}^{v-1}\right)^T - \mathbf{A}^T\left(\mathbf{Z}_\sigma^{v-1}\right)^T, v \neq N_v, \tag{6.42}$$

and

$$\left(\mathbf{Z}_x^{N_v}\right)^T = -\left(\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right)^T \mathbf{W}^T. \tag{6.43}$$

The "adjoint state" for $\mathbf{g}_\sigma^v$ is calculated from

$$\left(\mathbf{Z}_\sigma^v\right)^T = \mathbf{A}^{-T}\left(\mathbf{Z}_x^v\right)^T. \tag{6.44}$$

As discussed for the direct method, in the proposed derivative calculation method, Eq. (6.44) is solved as a smoothing step only, using any iterative solver up to a prescribed tolerance or number of iterations.

For $\mathbf{g}'^v$, the adjoint state is given by

$$\left(\mathbf{Z}'^v\right)^T = \left(\mathbf{Z}_x^v\right)^T - \mathbf{A}^T\left(\mathbf{Z}_\sigma^v\right)^T, \tag{6.45}$$

and finally for $\check{\mathbf{g}}^v$

$$\check{\mathbf{Z}}^v = \left(\check{\mathbf{A}}^{-T}\right)\left(\mathbf{P}^T\mathbf{Z}'^v\right). \tag{6.46}$$

Eq. (6.35) can be written as a sum where each term corresponds to the contribution of one iteration

$$\mathbf{WG} = \sum_{\nu=0}^{N^\nu} \left(\check{\mathbf{Z}}^\nu \frac{\partial \check{\mathbf{g}}^\nu}{\partial \theta} + \mathbf{Z}'^\nu \frac{\partial \mathbf{g}'^\nu}{\partial \theta} + \mathbf{Z}_\sigma^\nu \frac{\partial \mathbf{g}_\sigma^\nu}{\partial \theta} \right) + \mathbf{W} \frac{\partial \mathbf{h}}{\partial \theta}. \tag{6.47}$$

The algorithm can now be fully defined and is presented in 13. Note the use of the notation $(\mathbf{WG})^\nu$ to denote the partial sums in Eq. (6.47) and that references to superscript $\nu+1$ correspond to zero terms.

Algorithm 13: Pre multiplication of **G** by **W** via the Adjoint method.

Input : $\mathbf{R}, \mathbf{A}, \mathbf{P}, \boldsymbol{x}, \frac{\partial \mathbf{A}}{\partial \theta}, \frac{\partial \mathbf{h}}{\partial \mathbf{x}}, \frac{\partial \mathbf{h}}{\partial \theta}, \mathbf{W}, \epsilon_\sigma$

Output: The **WG** product

foreach $i = 1, 2, \ldots, m$ **do**

  foreach $\nu = N_\nu, \ldots, 1, 0$ **do**

    Set $\left(\mathbf{Z}_x^\nu\right)_{i,.}^T = \begin{cases} -\left(\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right)^T \mathbf{W}^T, \text{ if } \nu = N_\nu \\ \left(\mathbf{Z}_x^{\nu+1}\right)_{i,.}^T - \mathbf{A}^T \left(\mathbf{Z}_\sigma^{\nu+1}\right)_{i,.}^T - \mathbf{A}^T \mathbf{R}^T \left(\check{\mathbf{Z}}^{\nu+1}\right)_{i,.}^T, \text{ otherwise} \end{cases}$

    Iteratively solve $\left(\mathbf{Z}_\sigma^\nu\right)_{i,.}^T = \mathbf{A}^{-T} \left(\mathbf{Z}_x^\nu\right)_{i,.}^T$ until $\frac{\| \mathbf{Z}_x^\nu - \mathbf{A}\mathbf{Z}_{\sigma .,j}^{\ \nu} \|}{\| \mathbf{Z}_x^\nu \|} < \epsilon_\sigma$

    Compute $\left(\mathbf{Z}'^\nu\right)_{i,.}^T = \left(\mathbf{Z}_x^\nu\right)_{i,.}^T - \mathbf{A}^T \left(\mathbf{Z}_\sigma^\nu\right)_{i,.}^T$

    Solve $\check{\mathbf{Z}}_{i,.}^\nu = \left(\check{\mathbf{A}}^{-T}\right) \left(\mathbf{P}\check{\mathbf{Z}}_{i,.}^{\nu+1}\right)$

    Compute $\boldsymbol{\alpha}^T = \check{\mathbf{Z}}_{i,.}^\nu (\mathbf{RA}) - \mathbf{Z}'^\nu_{i,.}$

    Compute $\beta = \boldsymbol{\alpha} \frac{\partial \mathbf{P}}{\partial \theta} \check{\mathbf{x}}^\nu$; `// Algorithm 4 in [19] where` $\mathbf{m}^T = \boldsymbol{\alpha}^T$

    Compute $\boldsymbol{\gamma} = \mathbf{R} \frac{\partial \mathbf{A}}{\partial \theta} \mathbf{P}\check{\mathbf{x}}^\nu - \mathbf{R} \frac{\partial \mathbf{q}}{\partial \theta} + \mathbf{R} \frac{\partial \mathbf{A}}{\partial \theta} \mathbf{x}^{\nu-1}$

    Compute $\frac{\partial \mathbf{g}_\sigma^\nu}{\partial \theta} = \frac{\partial \mathbf{A}}{\partial \theta} \mathbf{x}_\sigma^\nu + \frac{\partial \mathbf{A}}{\partial \theta} \mathbf{x}'^\nu - \frac{\partial \mathbf{q}}{\partial \theta} + \frac{\partial \mathbf{A}}{\partial \theta} \mathbf{x}^{\nu-1}$

    Update $(\mathbf{WG})_{i,.}^\nu = (\mathbf{WG})_{i,.}^{\nu+1} + \beta + \check{\mathbf{Z}}_{i,.}^\nu \boldsymbol{\gamma} + \mathbf{Z}_{\sigma\, i,.}^\nu \frac{\partial \mathbf{g}_\sigma^\nu}{\partial \theta}$

  Update $\mathbf{WG} = (\mathbf{WG})^0 + \mathbf{W} \frac{\partial \mathbf{h}}{\partial \theta}$

REMARKS ABOUT THE FRAMEWORK

An alternative formulation for the i-MSFV formulation has been previously proposed and investigated in [23]. In that work, both the state and the model equation vectors explicitly account for the smoothing stage. The formulation here proposed is based on two observations. Firstly, the implementation of the aforementioned variant, although offers more control over the gradient quality, relies on the ability of computing partial derivative matrices of smoothing step w.r.t. the parameters. More specifically, it implies on the knowledge of how the precondition **M** of the system matrix **A** is built. For simpler iterative strategies, e.g. Jacobi, which construction of **M** can be simple, the computa-

tion of $\frac{\partial \mathbf{M}}{\partial \theta}$ is relatively simple. However, simpler iterative methods are usually less efficient. Also, the requirement of knowing the construction of $\mathbf{M}$ hampers the utilization of 'black-box' type of pre-conditioners. Secondly, it has been shown in [13] that only a limited number of smoothing steps are necessary to result in an efficient i-MSFV solution strategy. Hence, not much extra control would be achieved.

Note that the linear solvers employed in lines 9 and 4 of, respectively, algorithms 12 and 13, are the same solvers employed in the solution for the forward simulation, using the same prescribed (loose) tolerance. Hence, the algorithms shares the same computational advantages and solving for the approximated derivative information arising from the smoothing step.

The backward algorithm requires storing all intermediate states generated during the iterations in the forward run. It also requires solving many systems for each backward time-step. If the iteration process in the forward run goes until machine precision is reached, then essentially the fully coupled system has been solved to fine-scale precision and it might be more beneficial to neglect the iteration history and aim to solve the fine-scale system of adjoint equations in a more efficient way, given that the derivative computation problem is linear. However, we highlight that, as it has been shown in the literature [19, 25], approximate gradients computed from approximate solutions are already sufficient to efficiently/successfully lead the optimization process to the optimal solution. How accurate this gradient will need to be is dependent on different aspects, e.g. the optimization algorithm, how early/late the optimization process is, among others. The algorithm here proposed has the advantage of controlling the gradient quality, as will be shown in the numerical experiments presented in the next section.

6.3. NUMERICAL EXPERIMENTS

Given the fundamental nature of the developments here presented, we focus the numerical experiments on the validation of the computation and assessment of the gradient quality provided by the iterative multiscale method introduced.

Our experiments will be based on the evaluation of the gradient of a misfit objective function with no regularization term [3]

$$O(\theta) = \frac{1}{2}(\mathbf{h}(\mathbf{x},\theta) - \mathbf{d}_{obs})^T \mathbf{C}_D^{-1}(\mathbf{h}(\mathbf{x},\theta) - \mathbf{d}_{obs}), \tag{6.48}$$

with a gradient

$$\nabla_\theta O = \mathbf{G}^T \mathbf{C}_D^{-1}(\mathbf{h}(\mathbf{x},\theta) - \mathbf{d}_{obs}), \tag{6.49}$$

where $\mathbf{C}_D \in \mathbb{R}^{N_Y \times N_Y}$ is the so-called data covariance matrix. Here, $\mathbf{C}_D$ will be considered a diagonal matrix given by [3]

$$\mathbf{C}_D = \sigma^2 \mathbf{I}, \tag{6.50}$$

where σ^2 is the variance of the data measurement error.

In all experiments, the fitting parameters are cell-centered permeabilities. The observed quantity, $\mathbf{d}_{obs}$, is the fine scale pressure at the location of (non-flowing) observation wells, therefore

$$\frac{\partial \mathbf{h}}{\partial \mathbf{x}'} = \mathbf{I}, \tag{6.51}$$

and

$$\frac{\partial \mathbf{h}}{\partial \check{\mathbf{x}}} = \mathbf{0}. \tag{6.52}$$

In all experiments, the standard deviation of the pressure measurement error is $\sigma \approx 0.03$ (note that the measurement error is also non-dimensional). This represents a (very accurate) measurement error in the range of those usually employed in synthetic study cases (see e.g. [3]).

Note that the wells are controlled by bottom-hole pressure, expressed in terms of a non-dimensional pressure, i.e.,

$$p_D = \frac{p - p_{prod}}{p_{inj} - p_{prod}}, \tag{6.53}$$

where p_{inj} and p_{prod} are the injection and production pressures, respectively. In all the experiments, $p_{inj} = 1.0$ and $p_{prod} = 0.0$, the grid-block dimensions are $\Delta x = \Delta y = \Delta z = 1$ m and the fluid viscosity is 1.0×10^{-3} Pa s. In addition, in all the following test cases, well basis functions are included.

The metric utilized to to assess the i-MSFV gradient quality is the angle between fine-scale and i-MSFV normalized gradients, i.e.,

$$\alpha = \cos^{-1}\left(\nabla_\theta^T \hat{O}_{FS}\, \nabla_\theta \hat{O}_{MS}\right). \tag{6.54}$$

Here,

$$\nabla_\theta \hat{O}_{FS} = \frac{\nabla_\theta O_{FS}}{\|\nabla_\theta O_{FS}\|_2} \tag{6.55}$$

and

$$\nabla_\theta \hat{O}_{MS} = \frac{\nabla_\theta O_{MS}}{\|\nabla_\theta O_{MS}\|_2}. \tag{6.56}$$

Also, $\nabla_\theta O_{FS}$ and $\nabla_\theta O_{MS}$ denote the fine-scale and MS analytical gradients, respectively. As a minimum requirement, acceptable MS gradients are obtained if α is much smaller than 90^o [25]. And to prove our hypothesis, we particularly interested in observing the behaviour of the metric as more accurate i-MSFV solutions are computed.

In our i-MSFV implementation, the iterative process is controlled by the outer loop residual ϵ and the pre-conditioner smoother error tolerance ϵ_σ. The Krylov subspace biconjugate gradient stabilized method (BiCGSTAB, [26]) is employed in the smoothing stage.

6.3.1. VALIDATION EXPERIMENTS

Before focusing in the validation, in this section we will validate the iMS-gradient method against the numerical differentiation method with a higher order, two-sided Taylor approximation

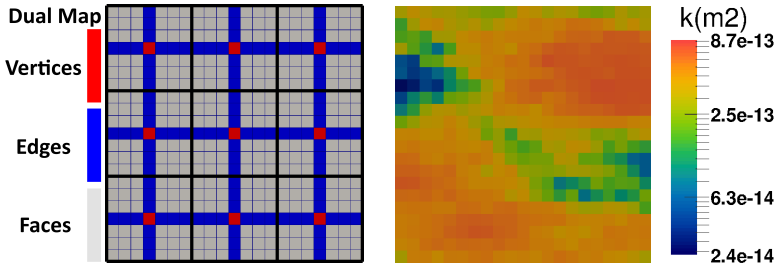
$$\begin{aligned}\nabla_\theta O_i = \frac{1}{2\delta\theta_i}\big(&O(\theta_1,\cdots,\theta_{i-1},\theta_i+\delta\theta_i,\theta_{i+1},\cdots,\theta_{N_\theta}) - \\ &O(\theta_1,\cdots,\theta_{i-1},\theta_i-\delta\theta_i,\theta_{i+1},\cdots,\theta_{N_\theta})\big)\end{aligned} \tag{6.57}$$

where we consider δ to be a multiplicative parameter perturbation. We define the relative error as

$$\varepsilon = \frac{||\nabla_\theta O_{NUM} - \nabla_\theta O_{iMS}||_2}{||\nabla_\theta O_{iMS}||_2} \tag{6.58}$$

Here $\nabla_\theta O_{NUM}$ is obtained by performing proper amount of iterative multiscale reservoir simulations required to evaluate equation [6.57]. $\nabla_\theta O_{iMS}$ is obtained by employing the iterative Direct or Adjoint gradient method.

To evaluate the correctness of the proposed iterative gradient computation methods, as well as their implementation, we investigate the linear decrease of the relative error ε by decreasing the parameter perturbation δ from 10^{-1} to 10^{-4}. This investigation is carried out in two distinct examples. Both have a fine grid of 21×21 grid blocks. We employ a 7×7 coarsening ratio, giving a 3×3 coarse grid. The reference twin-experiment is generated with permeability realization number 992. Fig. 6.1 illustrates the fine-, coarse- and dual-grid cells along with the reference permeability.



(a) MS grids (b) Reference permeability

Figure 6.1: Validation experiments setup. (a) Primal (bold black line), dual (identified by the blue cells) and fine grids (thin black lines). (b) Reference permeability field used in the twin experiment.

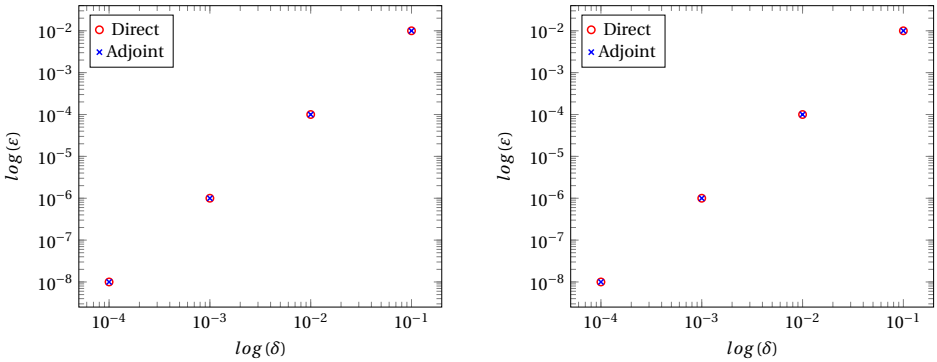
Next we determine the well positions. We use the so-called quarter well spot. Here, two observation wells are placed near operating wells. The full specifications can be found in Table [6.1].

Well	Fine scale position (I, J)	Well Type
INJE	(1, 1)	Injection
PROD	(21, 21)	Production
OBS1	(3, 3)	Observation
OBS2	(19, 19)	Observation

Table 6.1: Validation experiments well setup.

The results of this experiment are found in Fig. 6.2. Here, we use a single outer iteration. We use a very tight smoothing tolerance of $\epsilon_\sigma = 5 \times 10^{-8}$ to ensure that the numerical gradient method produces accurate enough gradients. First of all we can see that the fine-scale numerical gradient method and the iterative MS-gradient methods

are of the same order of accuracy with respect to the perturbation δ, for all different cases considered. In all experiments, we can see the linear decreasing behaviour of the relative error values ε as the perturbation δ decreases. Also we may easily see that the Adjoint and Direct method are equally accurate. They provide the analytical gradient at the same accuracy. The second experiment indicates the correctness of the method when it is applied to heterogeneous porous media problems.



(a) Homogeneous

(b) Heterogeneous

Figure 6.2: Validation of the i-MSFV gradient computation method compared with a numerical gradient. (*a*) represents the homogeneous test case, while (*b*) represents the heterogeneous test case.

6.3.2. INVESTIGATION OF I-MSFV CONVERGENCE BEHAVIOUR AND GRADIENT QUALITY

In this investigation, the error metric given by Eq. (6.54) is evaluated for a whole ensemble of heterogeneous permeability fields. The ensemble is generated via the decomposition of a reference permeability "image" using Principal Component Analysis parameterization. Figure 6.3 illustrates 4 different permeability realizations from the ensemble. See [27] for more details.

The fine-scale and coarse grids contain 21 x 21 and 7 x 7 cells, respectively. An inverted five-spot well pattern is employed, whiled four observation wells are placed close to the production wells. The well configuration is depicted in Table 6.2.

The synthetic observed pressures at the observation well locations are created via the classical twin-experiment strategy, which permeability field is extracted from the 1,000 geological realizations (the first one in Fig. 6.3).

The robustness of the method is illustrated in Fig. 6.4. It is possible to observe that the angle between fine-scale gradient and the i-MSFV gradient is smaller the tighter we make the outer-loop residual tolerance. Moreover, the variance also goes to almost zero as we set the residual tolerance to $1e^{-4}$. For this set of (relatively small contrast permeability contrast) permeability fields perfect alignment with with fine-scale gradient is reached if the tolerance is set to $1e^{-5}$.

We highlight that the permeability contrast of this ensemble is not high. Next, we assess the the robustness for the method for geological settings with higher permeability

Table 6.2: Well configuration for the homogeneous, two-dimensional case.

Well	Fine scale position (I, J)	Well type
INJE	(11, 11)	Injection
PROD1	(1, 1)	Production
PROD2	(21, 1)	Production
PROD3	(1, 21)	Production
PROD4	(21, 21)	Production
OBSWELL1	(3, 3)	Observation
OBSWELL2	(19, 3)	Observation
OBSWELL3	(3, 19)	Observation
OBSWELL4	(19, 19)	Observation

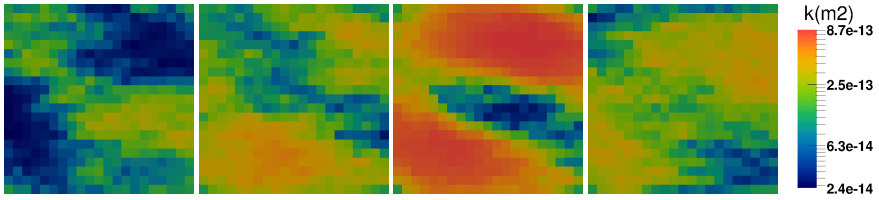


Figure 6.3: Four different permeability realizations from the ensemble of 1,000 members used in the 2D numerical experiments.

contrasts.

6.3.3. ROBUSTNESS WITH RESPECT TO HETEROGENEITY CONTRAST AND DISTRIBUTION

In order to further explore the point about the robustness of the method w.r.t. heterogeneity contrast and distribution, four sets of 20 equiprobable realizations of log-normally distributed permeability fields with a spherical variogram and dimensionless correlation lengths of $\Psi_1 = 0.5$ and $\Psi_2 = 0.02$ are generated using sequential Gaussian simulations [28]. For each set, the variance and the mean of $\ln(k)$ are 2.0 and 3.0, respectively, where k is the grid block permeability. As depicted in Fig. 6.5, for the realizations with a long correlation length, the angles between the permeability layers and the horizontal axis are 0^o, 15^o, and 45^o. A patchy (small correlation length) pattern is also considered (Figure 6.5d). Compared with the previous set, the permeability contrast is much higher in this case.

The fine-scale and coarse grids contain 100 x 100 and 20 x 20 cells, respectively. The well configuration utilized in this numerical experiment is depicted in Table 6.3.

The observed data is generated from a twin-experiment associated with (the first) permeability realization of each set.

In this experiment, $\epsilon = 1.0e^{-6}$ and $\epsilon_\sigma = 1.0e^{-1}$. The box-plot shown in Fig. 6.7 summarizes the required total number of smoothing steps, for all outer i-MSFV steps.

The grid orientation effect [29] impact on the performance of the i-MSFV method is clear in this example. The more the heterogeneity orientation is aligned with the flow

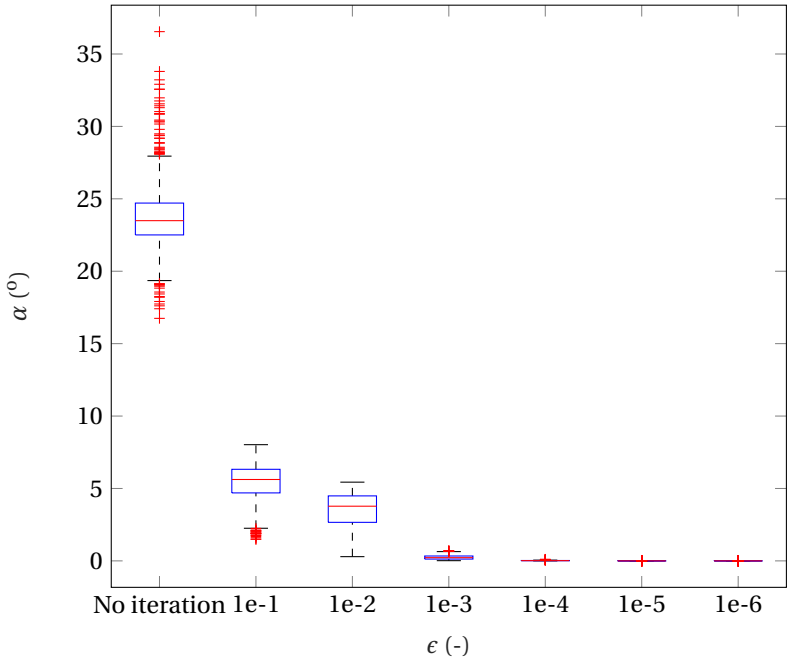
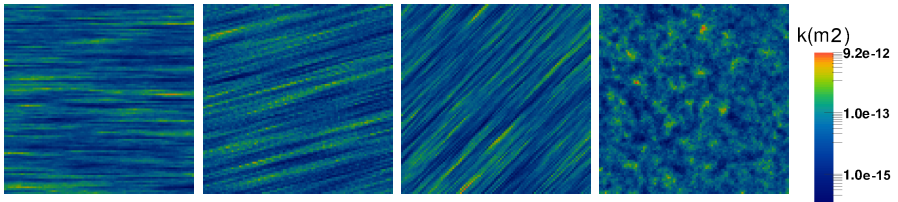


Figure 6.4: Box-plot illustrating the angle α between fine-scale gradient and i-MSFV gradient computed for the 1,000 member ensemble as a function of the outer-loop tolerance error ϵ. "No iteration" is equivalent to the MSFV gradient computation presented in [19].

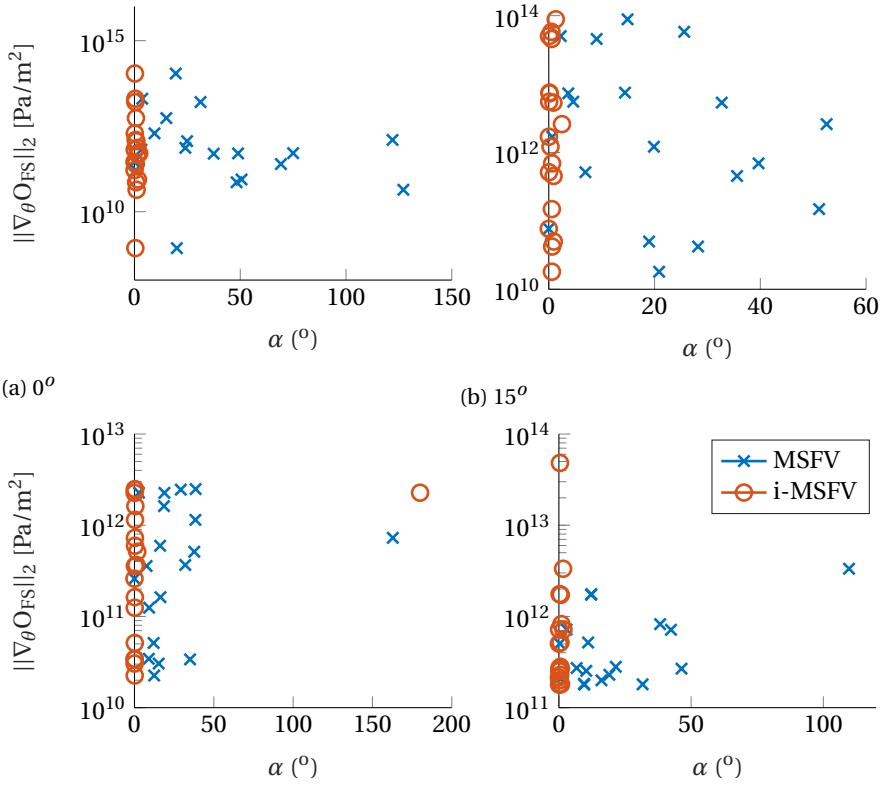
Table 6.3: Well configuration for Case 2.

Well	Fine scale position (I, J)	Well type
INJE	(1, 1)	Injection
PROD	(100, 100)	Production
OBSWELL1	(3, 3)	Observation
OBSWELL2	(98, 98)	Observation



(a) 0^o (b) 15^o (c) 45^o (d) Patchy

Figure 6.5: Permeability distribution of four different realizations taken from the sets of 20 geostatistically equiprobable permeability fields with different correlation angles (a-c). Also, a patchy field (d) with a small correlation length is considered.



(a) 0^o (b) 15^o (c) 45^o (d) Patchy

Figure 6.6: Illustration of gradient quality improvement when an i-MSFV gradient computation strategy is employed in comparison to a MSFV computation strategy. The x-axis represent the angle α between fine-scale and the MSFV gradient (illustrated by blue crosses) and the i-MSFV gradient with error tolerance $\epsilon = 10^{-6}$ (illustrated by orange circles).

orientation, the less is the number of required iterations. Also, in relation to Fig. 6.6, the more challenging the forward problem, the more challenging it is to compute i-MSFV

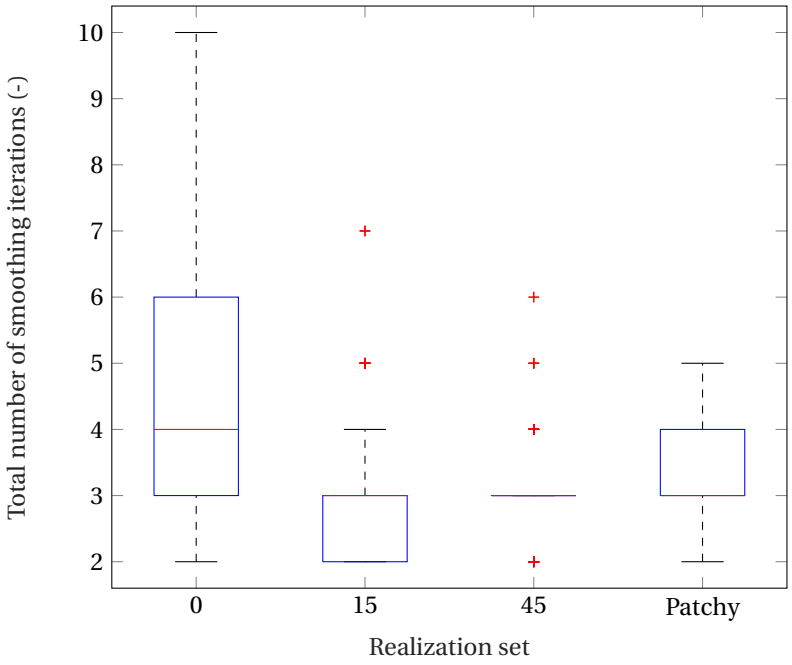
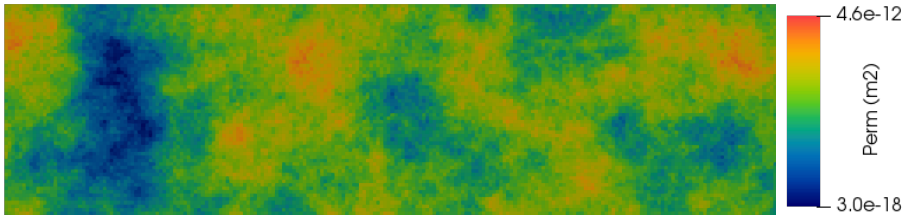


Figure 6.7: Box-plot representing the total number of smoothing iterations required to compute the misfit OF gradient for the different permeability ensembles with correlation angles 0^o, 15^o and 45^o and with a small correlation length (patchy).

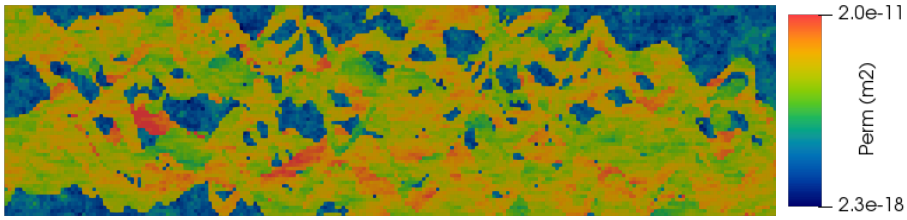
gradient in accordance to the fine-scale gradient. Nevertheless, in all cases, almost all i-MSFV realization gradients are perfectly aligned with fine-scale gradient, demonstrating the robustness of the method.

6.3.4. SPE-10 COMPARATIVE TEST CASE

Now, we investigate the performance of our method in the SPE-10 comparative case [30], regarded as challenging model for upscaling [31] and multiscale simulation [32] techniques. Here, we consider the 2-D flow simulation of both top and bottom layer of the original 3D model, which permeability fields are illustrated in 6.8.



(a) Top layer



(b) Bottom layer

Figure 6.8: SPE-10 comparative test case: top (a) and bottom (b) layer permeability fields.

The fine grid dimensions is 60 x 220, while we employ a 12 x 20 coarse grid in the MS simulation. A quarter five-spot well setting is considered. Four observation wells are deliberetely positioned in low permeability regions, surrounded by high permeability regions. The well positions are described in Table 6.4.

Table 6.4: Well configuration for the SPE-10 comparative test case.

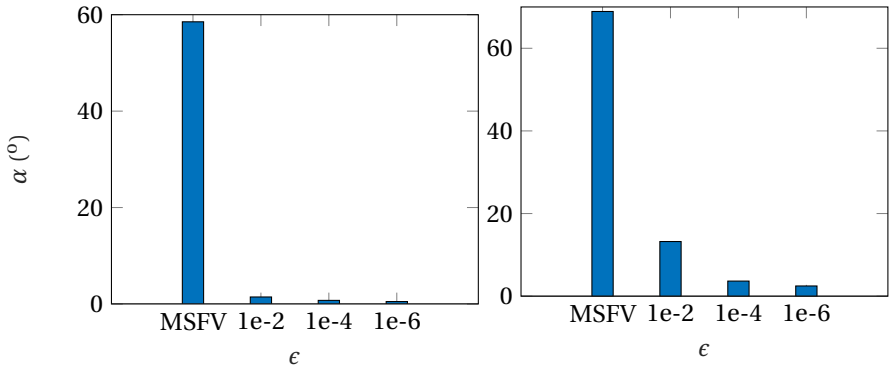
Well	Fine scale position (I, J)	Well type
INJE	(1, 220)	Injection
PROD	(60, 1)	Production
OBSWELL1	(33, 5)	Observation
OBSWELL2	(28, 50)	Observation
OBSWELL3	(28, 83)	Observation
OBSWELL4	(43, 204)	Observation

We note that it has been reported (see e.g. [32]) that the MSFV method provides non-

monotone pressure solutions when solving pressure for the bottom layer (6.8b), which strategies to improve the MS solution, among them the i-MSFV [33] here considered, are necessary to address the issue.

In this twin experiment, the reference permeability field used to compute the observations is considered homogeneous with value equals to 1e-13.

In this experiment, we fix $\epsilon_\sigma = 1.0e^{-2}$ and vary $\epsilon = 1.0e^{-2}, 1.0e^{-4}, 1.0e^{-6}$ to evaluate whether the same convergence behaviour of the i-MSFV gradient toward fine-scale gradient direction is observed.



(a) Top layer

(b) Bottom layer

Figure 6.9: i-MSFV gradient quality (angle α between fine-scale and i-MSFV gradients) as a function of residual error ϵ for the SPE-10 top layer (a) and bottom layer (b).

Once again we observe that the method provide accurate gradients, even considering this challenging geological setting, for both top and bottom layers of the model.

6.3.5. DISCUSSION

Based on the results acquired from the different numerical cases of increasing complexity we demonstrated that our newly introduced method can provide accurate gradients, up to fine-scale accuracy, if small enough residual tolerances are employed in the i-MSFV forward simulation. Also, it is shown that only a few i-MSFV/smoothing iterations are necessary to have reasonably accurate gradients. However, we highlight that, from an optimization point of view, gradients that are not fully in accordance with fine-scale gradient are necessary. As long as the gradient roughly points to the correct up/downward direction, the optimization process is able to progress towards the maximum/minimum. This is particularly true in the early iterations, the more steep region of the objective function. On the other hand, as the optimization process approach the optimum, more accurate gradients are required for precise stop criteria evaluation. The convergent behaviour of the i-MSFV gradient, which is demonstrated to be directly related to the outer-residual tolerance, an estimate defined *a-priori*, should allow for an error control of the gradient computation, which ultimately would allow control of the optimization process performance.

REFERENCES

[1] F. Anterion, R. Eymard, and B. Karcher, *Use of parameter gradients for reservoir history matching,* in *SPE Symposium on Reservoir Simulation* (Society of Petroleum Engineers, 1989).

[2] J. R. P. Rodrigues, *Calculating derivatives for automatic history matching,* Computational Geosciences **10**, 119 (2006).

[3] D. S. Oliver, A. C. Reynolds, and N. Liu, *Inverse theory for petroleum reservoir characterization and history matching* (Cambridge University Press, 2008).

[4] G. Chavent, M. Dupuy, and P. Lemmonier, *History matching by use of optimal theory,* Society of Petroleum Engineers Journal **15**, 74 (1975).

[5] R. Li, A. Reynolds, and D. Oliver, *History matching of three-phase flow production data,* SPEJ **8**, 328 (2003).

[6] J. F. B. M. Kraaijevanger, P. J. P. Egberts, J. R. Valstar, and H. W. Buurman, *Optimal waterflood design using the adjoint method,* in *SPE Reservoir Simulation Symposium* (Society of Petroleum Engineers, 2007).

[7] J. D. Jansen, *Adjoint-based optimization of multi-phase flow through porous media– a review,* Computers & Fluids **46**, 40 (2011).

[8] M. Cardoso, L. Durlofsky, and P. Sarma, *Development and application of reduced-order modeling procedures for subsurface flow simulation,* International journal for numerical methods in engineering **77**, 1322 (2009).

[9] J. F. van Doren, R. Markovinović, and J.-D. Jansen, *Reduced-order optimal control of water flooding using proper orthogonal decomposition,* Computational Geosciences **10**, 137 (2006), http://dx.doi.org/10.1007/s10596-005-9014-2.

[10] P. Jenny, S. H. Lee, and H. A. Tchelepi, *Multi-scale finite-volume method for elliptic problems in subsurface flow simulation,* J. Comput. Phys. **187**, 47 (2003).

[11] T. Y. Hou and X.-H. Wu, *A multiscale finite element method for elliptic problems in composite materials and porous media,* J. Comput. Phys. **134**, 169 (1997).

[12] H. Zhou and H. A. Tchelepi, *Operator-based multiscale method for compressible flow,* SPE J. **13**, 523 (2008).

[13] M. Ţene, Y. Wang, and H. Hajibeygi, *Adaptive algebraic multiscale solver for compressible flow in heterogeneous porous media,* J. Comput. Phys. **300**, 679 (2015).

[14] K.-A. Lie, O. Møyner, J. R. Natvig, A. Kozlova, K. Bratvedt, S. Watanabe, and Z. Li, *Successful application of multiscale methods in a real reservoir simulator environment,* Computational Geosciences **21**, 981 (2017).

[15] J. Fu, H. A. Tchelepi, and J. Caers, *A multiscale adjoint method to compute sensitivity coefficients for flow in heterogeneous porous media,* Advances in water resources **33**, 698 (2010).

[16] J. Fu, J. Caers, and H. A. Tchelepi, *A multiscale method for subsurface inverse modeling: Single-phase transient flow,* Adv. Water Resour. **34**, 967 (2011).

[17] S. Krogstad, V. L. Hauge, and A. Gulbransen, *Adjoint multiscale mixed finite elements,* SPE Journal **16**, 162 (2011).

[18] R. Moraes, J. Rodrigues, H. Hajibeygi, J. Jansen, *et al.*, *Multiscale gradient computation for multiphase flow in porous media,* in *SPE Reservoir Simulation Conference* (Society of Petroleum Engineers, 2017).

[19] R. J. de Moraes, J. R. Rodrigues, H. Hajibeygi, and J. D. Jansen, *Multiscale gradient computation for flow in heterogeneous porous media,* Journal of Computational Physics **336**, 644 (2017).

[20] J. D. Jansen, *Gradient-based optimization of flow through porous media, v.3,* (2016).

[21] Y. Wang, H. Hajibeygi, and H. A. Tchelepi, *Algebraic multiscale solver for flow in heterogeneous porous media,* J. Comput. Phys. **259**, 284 (2014).

[22] H. Hajibeygi, G. Bonfigli, M. A. Hesse, and P. Jenny, *Iterative multiscale finite-volume method,* J. Comput. Phys. **227**, 8604 (2008).

[23] P. Frank, *Iterative multiscale adjoint gradient computation for incompressible flow optimization in porous media,* (2017).

[24] R. J. de Moraes, J. R. Rodrigues, H. Hajibeygi, and J. D. Jansen, *Computing derivative information of sequentially coupled subsurface models,* Computational Geosciences (under review (June, 2018)).

[25] R. M. Fonseca, S. S. Kahrobaei, L. J. T. Van Gastel, O. Leeuwenburgh, and J. D. Jansen, *Quantification of the impact of ensemble size on the quality of an ensemble gradient using principles of hypothesis testing,* in *SPE Reservoir Simulation Symposium* (Society of Petroleum Engineers, 2015).

[26] Y. Saad, *Iterative methods for sparse linear systems* (SIAM, 2003).

[27] J. D. Jansen, *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, 2013).

[28] N. Remy, A. Boucher, and J. Wu, *Applied Geostatistics with SGeMS: A User's Guide* (Cambridge University Press, 2009).

[29] K. Aziz *et al.*, *Reservoir simulation grids: opportunities and problems,* Journal of Petroleum Technology **45**, 658 (1993).

[30] M. Christie, M. Blunt, *et al.*, *Tenth spe comparative solution project: A comparison of upscaling techniques,* in *SPE Reservoir Simulation Symposium* (Society of Petroleum Engineers, 2001).

[31] L. J. Durlofsky, *Upscaling and gridding of fine scale geological models for flow simulation,* in *8th International Forum on Reservoir Simulation Iles Borromees, Stresa, Italy*, Vol. 2024 (2005).

[32] H. Hajibeygi, *Iterative multiscale finite volume method for multiphase flow in porous media with complex physics*, Ph.D. thesis, ETH Zurich (2011).

[33] H. Hajibeygi and P. Jenny, *Adaptive iterative multiscale finite volume method,* Journal of Computational Physics **230**, 628 (2011).

7

MULTISCALE DATA ASSIMILATION OF SPATIALLY DISTRIBUTED DATA

In data assimilation problems, various types of data are naturally linked to different spatial resolutions (e.g. seismic and electromagnetic data), and these scales are usually not coincident to the subsurface simulation model scale. Alternatives like up/downscaling of the data and/or the simulation model can be used, but with potential loss of important information. To address this issue, a novel Multiscale (MS) data assimilation method is introduced. The overall idea of the method is to keep uncertain parameters and observed data at their original representation scale, avoiding up/downscaling of any quantity. The method relies on a recently developed mathematical framework to compute adjoint gradients via a MS strategy in an algebraic framework. The fine-scale uncertain parameters are directly updated and the MS grid is constructed in a resolution that meets the observed data resolution. This formulation therefore enables a consistent assimilation of data represented at a coarser scale than the simulation model. The misfit objective function is constructed to keep the MS nature of the problem. The regularization term is represented at the simulation model (fine) scale, whereas the data misfit term is represented at the observed data (coarse) scale. The computational aspects of the method are investigated in a simple synthetic model, including an elaborate uncertainty quantification step, and compared to up / downscaling strategies. The experiment shows that the MS strategy provides several potential advantages compared to more traditional scale conciliation strategies: 1) expensive operations are only performed at the coarse scale; 2) the matched uncertain parameter distribution is closer to the 'truth'; 3) faster convergence behaviour occurs due to faster gradient computation; and 4) better uncertainty quantification results are obtained. Clearly, larger-scale test are required to further quantify these potential benefits of MS data assimilation. The proof-of-concept example considered in this paper demonstrates how to consistently formulate such a gradient-based MS data assimilation strategy in an

The material presented in this chapter has been submitted to the Journal of Computational Geosciences (*submitted*) and in the proceedings of the European Conference on the Mathematics of Oil Recovery (ECMOR) XVI (2018) [1].

algebraic framework which allows for implementation in available computational platforms.

Subsurface simulation models should be conditioned to field data, whenever possible, in order to reduce uncertainty in the model parameters and hence increase forecasting reliability. Well production data (time-series of fluid rates and pressures), seismic surveys and well testing pressure data are some instances of field data that can be assimilated in order to better estimate the uncertain parameters. In addition, spatially distributed observations (e.g. seismic, electromagnetics) provide valuable information that can considerably improve the assimilation process [2]. For instance, over the past decades, an increasing number of seismic monitoring cases has been observed [3–8]. One of the main advantages of time-lapse seismic data is its ability to approximate the pressure/fluid distribution inside the reservoir. This may considerably help to gain insight about the subsurface fluid displacement process. Moreover, it may help to characterize the formation, either via improved static geological modeling, or via dynamic assimilation (inverse modeling). One hurdle in the process of assimilating spatially distributed information is the fact that, more often than not, the observed data and the forward model are described at different spatial scales. In fact, it is an open question at which scale data should best be assimilated: the simulation model scale or the observed data scale [2].

The relevancy of addressing the multiscale nature of inverse problems is observed in the recent literature on the topic. A collection of articles about multiscale forward modeling strategies and multiscale challenges associated to inverse problems can be found in [9]. In [10], the authors propose a multiscale data assimilation scheme based on the decomposition of the objective function so that the error covariance can be estimated for distinct spatial scales. A multiscale parameter field discretization designed to reduce the dimensionality of the inverse problem via an adaptive grid refinement is presented in [11]. The impact of the scale dissimilarity in terms of observation information content and the parameter space size on the ensemble collapse in ensemble based methods [12] has been addressed by [13–15] via upscaling/homogenization techniques [16, 17]. The authors also benefit from coarse scale simulations to improve the inverse problem computational efficiency. In [18] a multiscale method is proposed which accounts for microscale features by assuming that they can be represented by a low-dimensional parameterization. Nonetheless, the aforementioned works are based on the assumption that the fine-scale uncertain parameters can be homogenized and represented at a coarser scale.

On the other hand, data assimilation strategies based on multiscale (MS) simulation [19, 20] have also been developed. MS methods are efficient simulation strategies capable of solving the flow problem at a coarser grid, while being capable of accurately representing fine-scale heterogeneities. An adjoint-based multiscale finite volume method for computation of sensitivities has been presented in [21] and later extended to time-dependent [22] single-phase flow in porous media. More recently, a general framework for the computation of multiscale gradients has been introduced in [23], with an extension to multiphase flows [24]. The latter two are based on a general framework for derivative computation, whose algebraic nature does not rely on any assumption regarding the nature of the parameters, observations, or objective function type. Also, in [25] a multiscale inversion technique is presented based on the Markov-chain Monte Carlo method that also relies on the generalized multiscale finite element method [26].

Despite this body of work found in the in the inverse modeling literature, when one is interested in assimilating spatially distributed data, there is an implicit assumption that the observed data is described at the same scale of the parameters is usually made. Actually, assuming one is not interested in changing the scale of the model parameter description, some treatment must be employed in the change of the observed data or the forward model response scale. The literature indicates that up/downscaling of the observed data to the forward model scale, as a pre-processing step with respect to the data assimilation process, is the most employed strategy in practice [27–29]. In the present work, we are particularly interested in addressing the spatial scale dissimilarity between observations and the discretized forward model. In many applications, there is no observability of the spatially distributed data (due to limitations in the acquisition process; e.g. in terms of resolution) at the parameter resolution which is necessary to accurately describe important physical phenomena.

Another important aspect to be considered in data assimilation and uncertainty quantification (UQ) studies is the fact that those rely on computationally demanding algorithms. Different techniques such as Monte Carlo (MC) methods [30], Ensemble Kalman Filter (EnKF) and derivations [12, 31, 32] and randomized maximum likelihood (RML) [33] are developed to perform those studies. A comparison between the different techniques is provided by [34]. Regardless of the technique, a common feature they share is the necessity of performing many forward model runs in order to reasonably sample the posterior probability distribution of the reservoir uncertain parameters. As already mentioned, upscaling [16, 17], can build faster, reasonably accurate forward models that can speed up the sampling process. However, to accurately represent some physical phenomena, e.g. mixing, diffusion, fluid fronts, or compositional capillary effects, fine-scale resolution is of utmost importance. Hence, the ability of keeping the high fidelity description of geological parameters is fundamental for an adequate reservoir characterization. Partial-differential-equation-(PDE)-constrained optimization techniques can be employed in the solution of the inverse problem. In this case, it is well known that gradient-based algorithms are the most efficient ones, mainly if combined with efficient gradient computation. And it is also well known that gradients obtained with the adjoint method [35–38] are the most efficient and accurate ones.

The objective of this work is to develop and demonstrate an inverse modelling method that, at the same time, (1) is computationally efficient, (2) addresses the scale dissimilarity issue, with minimum loss of information, and (3) is capable of updating the highest fidelity model description. To this end, we exploit multiscale (MS) simulation strategies in order to (1) speed-up the forward simulation, while preserving fine-scale geological features, (2) efficiently compute gradient information and (3) seamlessly conciliate model and observed data scales. For a comprehensive review on the recent developments associated with MS methods applied to reservoir simulation, see [10].

The remainder of this paper is organized as follows. Firstly, a brief overview about how data assimilation is approached from a Bayesian perspective is presented. Next, we state our target forward model, consisting of incompressible single-phase flow in heterogeneous porous media. Also, we revisit the MS solution of the flow equation in a purely algebraic presentation. Thereafter, we discuss the data assimilation problem setup, focusing on the challenges of assimilating spatially distributed data. More specifically, we

discuss alternatives on how to conciliate data and model scales. Then, we introduce our multiscale data assimilation strategy, consisting of, basically, a MS objective function and a MS gradient computation strategy. The method here employed is largely based on the MS gradient computation strategy discussed by [23]. We compare the different data conciliation methods and our newly introduced method based on a synthetic 2D case. We focus our experiments on both the maximum a-posteriori (MAP) estimate and UQ via the RML method. A discussion about the results and the challenges that the method can encounter are presented next. Finally, a summary of the developments and results, as well as future research perspectives, are presented.

7.1. PRELIMINARIES

7.1.1. PROBLEM STATEMENT

Let N_d denote the number of space dimensions. Let $\Omega \subset \mathbb{R}^{N_d}$ be the problem domain with boundary $\partial\Omega$. Let $\in \mathbb{R}^{N_d}$ be an arbitrary space position. Let $\in \mathbb{R}^{N_d}$ be an unit normal vector to $\partial\Omega$. Our analysis focuses on phenomena governed by an elliptic PDE equation, denoted by g, in the form

$$(g)\begin{cases} \nabla\cdot(()\nabla()) = q(), & \in \Omega \\ \nabla() = \Gamma, & \in \partial\Omega_\Gamma \\ () = \Upsilon, & \in \partial\Omega_\Upsilon \end{cases} \tag{7.1}$$

where $\partial\Omega = \partial\Omega_\Upsilon \cup \partial\Omega_\Gamma$, $\partial\Omega_\Upsilon \cap \partial\Omega_\Gamma = \emptyset$, $= ()$ is the variable of interest, $q = q()$ is the sink/source term, and is the heterogeneous uncertain coefficient which we aim to estimate via the assimilation of real system observations. We assume has no separation of scales, hence, homogenization techniques would lead to unavoidable approximations to the effective property.

Let

$$y = h(,) \tag{7.2}$$

be the observable model responses, then the inverse problem

$$= h(,) + \epsilon \tag{7.3}$$

can provide an estimate for given the description of the real observation errors ϵ. We assume that , the real system data, can only be observed at a resolution that is coarser or equal to the resolution at which is described.

7.1.2. INVERSE PROBLEM AS A PDE CONSTRAINED OPTIMIZATION

We base our developments on a Bayesian framework. Let N_y be number of observable responses, N_θ be the number of model parameters and N_x the number of primary (state) variables. According to Bayes' theorem, the posterior probability distribution function (PDF) can be computed as

$$f(\theta|\mathbf{d}_{obs}) = \frac{f(\mathbf{d}_{obs}|\theta)\, f(\theta)}{f(\mathbf{d}_{obs})}, \tag{7.4}$$

where $\theta \in \mathbb{R}^{N_\theta}$ is the vector of model parameters and $\mathbf{d}_{obs} \in \mathbb{R}^{N_y}$ is the vector observable responses. If the *a priori* PDF of the uncertain parameters, $f(\theta)$, and the measurement errors from the observations are assumed Gaussian, it can be shown that the conditional *a posteriori* distribution is given by [39]

$$f(\theta|\mathbf{d}_{obs}) \propto \exp\left(-O\left(\mathbf{y}(\mathbf{x},\theta),\theta\right)\right), \tag{7.5}$$

where the objective function $O \in \mathbb{R}$ is given by

$$\begin{aligned} O\left(\mathbf{y}(\mathbf{x},\theta),\theta\right) = &\frac{1}{2}\left(\theta-\theta_{prior}\right)^T \mathbf{C}_\theta^{-1}\left(\theta-\theta_{prior}\right) + \\ &\frac{1}{2}\left(\mathbf{y}(\mathbf{x},\theta)-\mathbf{d}_{obs}\right)^T \mathbf{C}_D^{-1}\left(\mathbf{y}(\mathbf{x},\theta)-\mathbf{d}_{obs}\right). \end{aligned} \tag{7.6}$$

In the above equations, $\mathbf{y} \in \mathbb{R}^{N_y}$ is the vector of model responses (outputs), $\mathbf{x} \in \mathbb{R}^{N_x}$ is the state vector, $\theta_{prior} \in \mathbb{R}^{N_\theta}$ is the prior mean, $\mathbf{C}_\theta \in \mathbb{R}^{N_\theta \times N_\theta}$ is the parameter covariance matrix and $\mathbf{C}_D \in \mathbb{R}^{N_y \times N_y}$ is the covariance matrix of the measurement errors.

The solution of Eq. (7.6) can be stated as a PDE-constrained optimization problem as [40]

$$\begin{aligned} \underset{\theta}{\text{minimize}} \quad & O\left(\mathbf{y}(\mathbf{x},\theta),\theta\right) \\ \text{subject to} \quad & \mathbf{g}(\mathbf{x},\theta) = \mathbf{0}, \\ & \theta \in [\theta_{min},\theta_{max}], \end{aligned} \tag{7.7}$$

where $\mathbf{g}: \mathbb{R}^{N_x} \times \mathbb{R}^{N_\theta} \rightarrow \mathbb{R}^{N_x}$ represents the set of forward model equations and $\theta_{min} \in \mathbb{R}^{N_\theta}, \theta_{max} \in \mathbb{R}^{N_\theta}$ are, respectively, the parameter lower and upper bound vectors. The efficient solution of Eq. (7.7), resulting from the discretization of Eq. (7.1), requires gradient-based methods [41] combined with efficient gradient computation methods. For this purpose, by applying the chain rule to Eq. (7.6) it follows that

$$\begin{aligned} \nabla_\theta O\left(\theta,\mathbf{y}(\mathbf{x},\theta)\right) &= \left(\frac{dO}{d\theta}\right)^T \\ &= \left(\frac{\partial O}{\partial \theta}\right)^T + \mathbf{G}^T \nabla_{\mathbf{y}} O, \end{aligned} \tag{7.8}$$

where $\mathbf{G} \in \mathbb{R}^{N_y \times N_\theta}$ is the so-called sensitivity matrix, representing the sensitivity of the responses w.r.t. the parameters. Efficient gradient methods are analytical, and more specifically in inverse problems where the number of parameters is greater than the number of output functionals, the adjoint method is the most accurate, efficient method [40]. The efficient computation of the right-multiplication of $\mathbf{G}$ by an arbitrary vector (as in Eq. (7.8)) via the adjoint method is discussed in [37].

7.1.3. RANDOMIZED MAXIMUM LIKELIHOOD (RML)

RML [33] is an approximated sampling method for UQ, which obtains the j-th sample of the posterior PDF distribution by solving Eq. (7.7) for a given sample $\theta_{uc,j}$ from a normal distribution $\mathcal{N}(\theta_{prior}, \mathbf{C}_\theta)$ and a given sample $\mathbf{d}_{obsuc,j}$ from $\mathcal{N}(\mathbf{d}_{obs}, \mathbf{C}_D)$. Therefore,

Eq. (7.6) can be re-written for $O_j\left(\mathbf{y}(\mathbf{x},\theta),\theta\right)$ as

$$\begin{aligned} O_j\left(\mathbf{y}(\mathbf{x},\theta),\theta\right) = &\frac{1}{2}\left(\theta-\theta_{uc,j}\right)^T\mathbf{C}_\theta^{-1}\left(\theta-\theta_{uc,j}\right)+ \\ &\frac{1}{2}\left(\mathbf{y}(\mathbf{x},\theta)-\mathbf{d}_{obs\,uc,j}\right)^T\mathbf{C}_D^{-1}\left(\mathbf{y}(\mathbf{x},\theta)-\mathbf{d}_{obs\,uc,j}\right). \end{aligned} \tag{7.9}$$

Hence, a minimization problem has to be solved for every j-th posterior PDF sample one wants to estimate. This is only feasible with efficient gradient computation methods, as described in the previous section.

7.2. THE FORWARD MODEL

The set of discretized equations that describes the forward simulation at the fine scale can be algebraically expressed as [23]

$$\mathbf{g}_F(\mathbf{x},\theta)=\mathbf{0}, \tag{7.10}$$

where $\mathbf{g}_F:\mathbb{R}^{N_F}\times\mathbb{R}^{N_\theta}\rightarrow\mathbb{R}^{N_F}$ represents the set of algebraic forward model equations resulting from the numerical discretization of Eq. (7.1) over a fine grid $\in\mathbb{R}^{N_F}$, $\mathbf{x}\in\mathbb{R}^{N_F}$ is the state vector and the subscript F refers to 'fine scale'. There are N_F fine-scale cells. Eq. (7.10) implicitly assumes a dependency of the state vector $\mathbf{x}$ on the parameters θ, i.e.

$$\mathbf{x}=\mathbf{x}(\theta). \tag{7.11}$$

Once the model state is determined, the observable responses of the forward model are computed. The forward model responses may not only depend on the model state, but also on the parameters themselves, and can be expressed as

$$\mathbf{y}_F=\mathbf{h}_F(\mathbf{x},\theta), \tag{7.12}$$

where $\mathbf{h}_F:\mathbb{R}^{N_F}\times\mathbb{R}^{N_\theta}\rightarrow\mathbb{R}^{N_y}$ represents the output equations [42]. It is assumed that $\mathbf{g}_F$ can be described as

$$\mathbf{g}_F(\mathbf{x},\theta)=\mathbf{A}(\theta)\mathbf{x}-\mathbf{q}(\theta), \tag{7.13}$$

where $\mathbf{A}=\mathbf{A}(\theta)\in\mathbb{R}^{N_F}\times\mathbb{R}^{N_F}$ represents the elliptic discrete operator and $\mathbf{q}=\mathbf{q}(\theta)\in\mathbb{R}^{N_F}$ is a vector of source terms and boundary conditions.

7.2.1. MULTISCALE SIMULATION

A multiscale (MS) solution strategy can be algebraically devised [43, 44] by firstly computing a coarse scale solution

$$\check{\mathbf{g}}(\check{\mathbf{x}},\theta)=(\mathbf{RAP})\check{\mathbf{x}}-\left(\mathbf{Rq}\right)=\check{\mathbf{A}}\check{\mathbf{x}}-\check{\mathbf{q}}=\check{\mathbf{0}}, \tag{7.14}$$

where-after an approximate fine-scale solution is formed as

$$\mathbf{g}'\left(\mathbf{x}',\check{\mathbf{x}},\theta\right)=\mathbf{x}'-\mathbf{P}\check{\mathbf{x}}=\mathbf{0}. \tag{7.15}$$

Let $\check{\mathbf{x}}\in\mathbb{R}^{N_C}$ be the coarse scale solution ($N_C\ll N_F$), and $\mathbf{x}'\in\mathbb{R}^{N_F}$ the approximated fine-scale solution. The prolongation operator $\mathbf{P}=\mathbf{P}(\theta)$ is an $N_F\times N_C$ matrix that maps

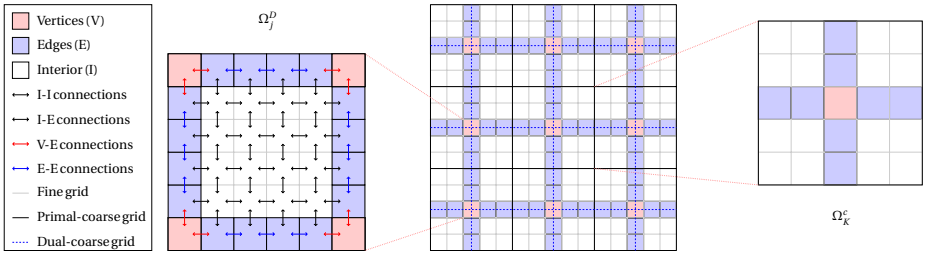


Figure 7.1: Multiscale finite volume grids and illustration of interfacial connections between cells used in the wirebasket ordering.

(interpolates) the coarse-scale solution to the fine-scale, where N_C is the number of coarse grid-blocks. The restriction operator $\mathbf{R} = \mathbf{R}(\theta)$ is defined as an $N_C \times N_F$ matrix which maps the fine scale to the coarse scale.

In multiscale methods, the scaling operators are constructed based on locally supported basis functions. Different strategies to build MS basis functions are available in the literature [19, 20, 26, 45]. In this work, we employ the multiscale finite volume (MSFV) method [20]. However, we emphasize, as will be clear from the formulation, that the framework allows the employment of different MS methods, as long as they can be expressed in terms of $\mathbf{R}$ and $\mathbf{P}$. Next we discuss the MSFV basis function construction.

CONSTRUCTION OF SCALING OPERATORS VIA THE MSFV METHOD

The MSFV discretization relies on two overlapping coarse grids, namely the primal and dual coarse grids, which are superimposed on a given fine grid. The grids are illustrated in Fig. 7.1. The primal-coarse grid contains N_C control volumes $\breve{\Omega}_i$, $i \in \{1,\ldots,N_C\}$, and the dual-coarse grid contains N_D local domains $\tilde{\Omega}_j$, $j \in \{1,\ldots,N_D\}$.

The MSFV basis functions are constructed based on local solutions of the elliptic governing equation Eq. (7.1) for every $\tilde{\Omega}_j$, with no right-hand-side and subject to special boundary conditions [20, 44]

$$\begin{cases} \nabla \cdot \left(() \nabla \varphi_j^i () \right) = 0, & \in \tilde{\Omega}_j \\ \nabla_\parallel \cdot \left(() \nabla \varphi_j^i () \right)_\parallel = 0, & \in \partial\tilde{\Omega}_j \\ \varphi_j^i (_k) = \delta_{ik}, & \forall_k \in \{1,...,N_C\}, \end{cases} \tag{7.16}$$

where φ_j^i is the basis function associated with the vertex i in $\tilde{\Omega}_j$, the subscript $\parallel$ represents the projection along $\partial\tilde{\Omega}_j$, δ_{ik} is the Kronecker delta, and $k \in \{1,\ldots,2^{N_d}\}$ denotes the vertices in $\tilde{\Omega}_j$ (N_d is the spatial dimensionality of the problem – 1, 2 or 3).

Assuming a finite volume discretization of Eq. (7.1), the basis functions, and hence the prolongation operator, can be constructed directly from the given fine-grid linear system matrix as [46]

$$\mathbf{P} = \mathscr{P} \begin{bmatrix} \mathbf{A}_{II}^{-1}\mathbf{A}_{IE}\tilde{\mathbf{A}}_{EE}^{-1}\mathbf{A}_{EV} \\ -\tilde{\mathbf{A}}_{EE}^{-1}\mathbf{A}_{EV} \\ -\mathbf{I}_{VV} \end{bmatrix}, \tag{7.17}$$

after $\mathbf{A}$ in Eq. (7.13) is re-ordered in a wirebasket ordering [47] as

$$\begin{aligned}\mathbf{g} &= \mathscr{P}\begin{bmatrix}\mathbf{g}_I\\ \mathbf{g}_E\\ \mathbf{g}_V\end{bmatrix}\\ &= \left(\mathscr{P}\begin{bmatrix}\mathbf{A}_{II} & \mathbf{A}_{IE} & \mathbf{0}\\ \mathbf{A}_{EI} & \mathbf{A}_{EE} & \mathbf{A}_{EV}\\ \mathbf{0} & \mathbf{A}_{VE} & \mathbf{A}_{VV}\end{bmatrix}\mathscr{P}^T\right)\mathscr{P}\begin{bmatrix}\mathbf{x}_I\\ \mathbf{x}_E\\ \mathbf{x}_V\end{bmatrix} - \mathscr{P}\begin{bmatrix}\mathbf{q}_I\\ \mathbf{q}_E\\ \mathbf{q}_V\end{bmatrix},\end{aligned} \tag{7.18}$$

where $\mathscr{P}\in\mathbb{R}^{N_F\times N_F}$ is a permutation matrix that reorders from wirebasket to natural ordering, $\mathbf{A}_{II}\in\mathbb{R}^{N_I\times N_I}$, $\mathbf{A}_{IE}\in\mathbb{R}^{N_I\times N_E}$, $\mathbf{A}_{EI}\in\mathbb{R}^{N_E\times N_I}$, $\mathbf{A}_{EE}\in\mathbb{R}^{N_E\times N_E}$, $\mathbf{A}_{EV}\in\mathbb{R}^{N_E\times N_V}$, $\mathbf{A}_{VE}\in\mathbb{R}^{N_V\times N_E}$, $\mathbf{A}_{VV}\in\mathbb{R}^{N_V\times N_V}$ are, respectively, the sub-matrices of $\mathbf{A}$ corresponding to the interior-interior, interior-edge, edge-interior, edge-edge, edge-vertex, vertex-edge and vertex-vertex cell connections and N_I, N_E and N_V are, respectively, the total number of interior, edge and vertex cells in the fine grid. The interfacial connections are illustrated in Fig. 7.1. Also, $\mathbf{g}_I\in\mathbb{R}^{N_I}$, $\mathbf{g}_E\in\mathbb{R}^{N_E}$, $\mathbf{g}_V\in\mathbb{R}^{N_V}$, $\mathbf{x}_I\in\mathbb{R}^{N_I}$, $\mathbf{x}_E\in\mathbb{R}^{N_E}$, $\mathbf{x}_V\in\mathbb{R}^{N_V}$, and $\mathbf{q}_I\in\mathbb{R}^{N_I}$, $\mathbf{q}_E\in\mathbb{R}^{N_E}$, $\mathbf{q}_V\in\mathbb{R}^{N_V}$ are, respectively, the model equations, and the state and source term sub-vectors corresponding to the interior, edge and vertex cells.

Note that the construction of $\mathbf{P}$ requires setting $\mathbf{A}_{VE}=\mathbf{0}$, $\mathbf{A}_{EI}=\mathbf{0}$, and $\mathbf{A}_{VV}=\mathbf{I}_V V$ and likewise the corresponding entries $\mathbf{A}_{EE}$, resulting in $\tilde{\mathbf{A}}_{EE}$, which is equivalent to the localization assumptions required to build the basis functions as stated in Eq. (7.16) [46].

If the FV method is used in the fine-scale system discretization, the restriction operator can be defined as the sum of the equations of all the fine cells contained in the coarse cell, i.e. [46]

$$\mathbf{R}_{i,j}=\begin{cases}1, \text{if} \quad \Omega_i\subset\check{\Omega}_c\\ 0, \text{otherwise.}\end{cases}\quad \left(c=1,\ldots,N_C; f=1,\ldots,N_F\right), \tag{7.19}$$

hence, in combination with Eq. (7.17), establishing the Multiscale Finite Volume (MSFV) method. Also, a Galerkin restriction operator could be used by making

$$\mathbf{R}=\mathbf{P}^T,$$

and hence, in combination with Eq. (7.17), establishing the MS Finite Element (MSFE) method. While the MSFV is conservative by construction, the MSFE provides monotone solutions.

7.3. DATA ASSIMILATION PROBLEM SETUP

7.3.1. ADJOINT GRADIENT COMPUTATION

The maximum *a posteriori* probability (MAP, [40]) of the uncertain parameters is obtained by solving the optimization problem stated in Eq. (7.7), with the objective function (OF) given by Eq. (7.6) (and using Eq. (7.12)), the gradient which is given by

$$\begin{aligned}\nabla_\theta O &= \mathbf{C}_\theta^{-1}\left(\theta-\theta_{prior}\right)+\left(\frac{d\mathbf{h}}{d\theta}\right)^T\mathbf{C}_D^{-1}\left(\mathbf{h}\left(\mathbf{x},\theta\right)-\mathbf{d}_{obs}\right)\\ &= \mathbf{p}+\mathbf{G}^T\mathbf{m},\end{aligned} \tag{7.20}$$

where

$$\mathbf{p} = \mathbf{C}_{\theta}^{-1}\left(\theta - \theta_{prior}\right), \tag{7.21}$$

and

$$\mathbf{m} = \mathbf{C}_{D}^{-1}\left(\mathbf{h}\left(\mathbf{x},\theta\right) - \mathbf{d}_{obs}\right), \tag{7.22}$$

$\mathbf{m} \in \mathbb{R}^{N_Y}$. Following an implicit differentiation strategy [37, 38], the sensitivity matrix $\mathbf{G}$ can be obtained from the total derivative of Eq. (7.12) with respect to θ as follows [23]:

$$\mathbf{G} = \frac{d\mathbf{h}}{d\theta} = \frac{\partial \mathbf{h}}{\partial \mathbf{x}}\frac{d\mathbf{x}}{d\theta} + \frac{\partial \mathbf{h}}{\partial \theta} = -\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\left(\frac{\partial \mathbf{g}}{\partial \mathbf{x}}\right)^{-1}\frac{\partial \mathbf{g}}{\partial \theta} + \frac{\partial \mathbf{h}}{\partial \theta}, \tag{7.23}$$

where $\frac{\partial \mathbf{g}}{\partial \mathbf{x}} \in \mathbb{R}^{N_F} \times \mathbb{R}^{N_F}$, $\frac{\partial \mathbf{g}}{\partial \theta} \in \mathbb{R}^{N_F} \times \mathbb{R}^{N_\theta}$, $\frac{\partial \mathbf{h}}{\partial \mathbf{x}} \in \mathbb{R}^{N_y} \times \mathbb{R}^{N_F}$, $\frac{\partial \mathbf{h}}{\partial \theta} \in \mathbb{R}^{N_y} \times \mathbb{R}^{N_\theta}$ are, respectively, partial derivative matrices obtained from the derivation of Eq. (7.10) and Eq. (7.12) with respect to $\mathbf{x}$ and θ.

The product $\mathbf{G}^T\mathbf{m} = \left(\mathbf{m}^T\mathbf{G}\right)^T$ can be solved at costs proportional to one backward simulation, regardless the number of model parameters, via the adjoint method, by pre-multiplying Eq. (7.23) by the transpose of Eq. (7.22), as discussed in [37]. By defining

$$\mathbf{z} = \left(\frac{\partial \mathbf{g}}{\partial \mathbf{x}}\right)^{-T}\left(\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right)^{T}\mathbf{m}, \tag{7.24}$$

it follows that

$$\mathbf{m}^T\mathbf{G} = -\mathbf{z}^T\frac{\partial \mathbf{g}}{\partial \theta} + \mathbf{m}^T\frac{\partial \mathbf{h}}{\partial \theta}. \tag{7.25}$$

7.3.2. CONCILIATION OF SPATIALLY DISTRIBUTED DATA AND FORWARD MODEL SCALES

In the data assimilation of spatially distributed observations, Eq. (7.6) assumes that the observations $\mathbf{d}_{obs}$ and the model responses $\mathbf{h}$ are described at the same scale. This is often not the case. Due to resolution issues and acquisition limitations, observations are often not available at the scale of the model responses. Therefore, if no MS simulation is available, either $\mathbf{d}_{obs}$ must be downscaled to the simulation scale or $\mathbf{h}$ must be upscaled to the observation scale.

The downscaling of observed data can be expressed as

$$\mathbf{d}_{obs}{}' = \mathbf{D}\breve{\mathbf{d}}_{obs}, \tag{7.26}$$

where $\mathbf{D}$ is an $N_F \times N_C$ downscaling operator, $\breve{\mathbf{d}}_{obs} \in \mathbb{R}^{N_C}$ is the coarse scale observation and $\mathbf{d}' \in \mathbb{R}^{N_F}$ is the interpolated observation at the fine scale.

Additionally, one must be able to describe the data covariance matrix $\mathbf{C}_D$, originally at (coarse) observation scale, at the fine scale. This can be achieved by setting

$$\mathbf{C}'_D = \mathbf{D}\breve{\mathbf{C}}_D\mathbf{D}^T, \tag{7.27}$$

where $\mathbf{C}'_D$ is the covariance matrix represented at the fine scale. It is simple to show that Eq. (7.27) holds because of the linearity of the expectation operator given the Gaussian assumptions. From Eq. (7.26), the expectation of $\mathbf{d}_{obs}{}'$ is given by

$$E[\mathbf{d}_{obs}{}'] = \mathbf{D}E[\breve{\mathbf{d}}_{obs}]. \tag{7.28}$$

The covariance of $\mathbf{d}_{obs}{}'$ can then be computed as (Emerick, A. A., personal communication, March 23, 2018)

$$\begin{aligned} Cov[\mathbf{d}_{obs}{}'] &= E[(\mathbf{d}_{obs}{}' - E[\mathbf{d}_{obs}{}'])(\mathbf{d}_{obs}{}' - E[\mathbf{d}_{obs}{}'])^T] \\ &= E[\mathbf{D}(\breve{\mathbf{d}}_{obs} - E[\breve{\mathbf{d}}_{obs}])(\breve{\mathbf{d}}_{obs} - E[\breve{\mathbf{d}}_{obs}])^T\mathbf{D}^T] \\ &= \mathbf{D}E[(\breve{\mathbf{d}}_{obs} - E[\breve{\mathbf{d}}_{obs}])(\breve{\mathbf{d}}_{obs} - E[\breve{\mathbf{d}}_{obs}])^T]\mathbf{D}^T \\ &= \mathbf{D}Cov[\breve{\mathbf{d}}_{obs}]\mathbf{D}^T \end{aligned} \tag{7.29}$$

Alternatively, one could upscale the model responses as

$$\breve{\mathbf{d}}_{obs} = \mathbf{U}\mathbf{h}, \tag{7.30}$$

where $\mathbf{U}$ is an $N_C \times N_F$ upscaling operator, and solve Eq. (7.6) by setting $\mathbf{d}_{obs} = \breve{\mathbf{d}}_{obs}$. One advantage over the dowscaling strategy is that $\mathbf{C}_D$ is kept at its original scale.

We highlight that we only consider strategies that change observed data / response scale and do not consider strategies that change the original uncertain parameters description scale. This is because we aim to update the most accurate description of the model parameters, so that important fine-scale features (crucial to describe the physical phenomena) are not lost.

7.4. MULTISCALE DATA ASSIMILATION

An MS solution strategy provides a coarse-scale solution that can, theoretically, be represented at any resolution coarser than the fine-scale resolution. In data assimilation studies, where the spatially distributed data resolution is known and is coarser than the model resolution, the MS grid can be chosen to be at the same resolution as the assimilation grid. This allows spatially distributed model responses to be computed at the same scale as the observed data. Next, we devise a multiscale data assimilation procedure based on this feature. This allows us, instead of manipulating the data and/or the uncertain parameters, to accurately compute responses at the observed data scale.

Therefore, a multiscale objective function is introduced by re-writting Eq. (7.6) as

$$\begin{aligned} O_{MS}(\breve{\mathbf{h}}, \theta) = &\frac{1}{2}(\theta - \theta_{prior})^T\mathbf{C}_\theta^{-1}(\theta - \theta_{prior}) + \\ &\frac{1}{2}\left(\breve{\mathbf{h}}(\breve{\mathbf{x}}, \theta) - \breve{\mathbf{d}}_{obs}\right)^T\mathbf{C}_D^{-1}\left(\breve{\mathbf{h}}(\breve{\mathbf{x}}, \theta) - \breve{\mathbf{d}}_{obs}\right), \end{aligned} \tag{7.31}$$

where $\breve{\mathbf{h}}$ is the response at the (coarse) observation scale and $\breve{\mathbf{x}}$ is the coarse state variable, computed by Eq. (7.14). Hence, the misfit term is computed at the coarse scale – the scale where data is assimilated – and the regularization term is described at the fine scale – the scale at the model parameters are described.

7.4.1. MULTISCALE GRADIENT COMPUTATION

As discussed in [23], the state vector can be described as a combination of both sets of primary variables at the fine and coarse scales, i.e.,

$$\mathbf{x} = \begin{bmatrix} \breve{\mathbf{x}} \\ \mathbf{x}' \end{bmatrix}, \tag{7.32}$$

and, similarly, the model equations can be represented as a combination of the equations at both scales, i.e.,

$$\mathbf{g}(\mathbf{x},\theta) = \begin{bmatrix} \check{\mathbf{g}} \\ \mathbf{g}' \end{bmatrix} = \mathbf{0}. \tag{7.33}$$

The definition of the state vector as in Eq. (7.32) is a key aspect of this development. It allows the description of the state not only at the fine scale, but also at the coarse scale. The simulator responses $\boldsymbol{y}$ obtained from the multiscale method are represented as

$$\check{\mathbf{y}} = \check{\mathbf{h}}(\mathbf{x},\theta), \tag{7.34}$$

the sensitivity matrix $\mathbf{G}$ can be computed in a multiscale fashion as

$$\begin{aligned}\mathbf{G}' = &\left(\frac{\partial\check{\mathbf{h}}}{\partial\check{\mathbf{x}}} + \frac{\partial\check{\mathbf{h}}}{\partial\mathbf{x}'}\mathbf{P}\right)(\mathbf{RAP})^{-1}\mathbf{R}\left(\frac{\partial\mathbf{A}}{\partial\theta}\mathbf{P} + \mathbf{A}\frac{\partial\mathbf{P}}{\partial\theta}\right)\check{\mathbf{x}} - \\ &\frac{\partial\check{\mathbf{h}}}{\partial\mathbf{x}'}\frac{\partial\mathbf{P}}{\partial\theta}\check{\mathbf{x}} + \frac{\partial\check{\mathbf{h}}}{\partial\theta}.\end{aligned} \tag{7.35}$$

Eq. (7.35) allows capturing derivative informations of the coarse response, namely $\dfrac{\partial\check{\mathbf{h}}}{\partial\check{\mathbf{x}}}$ and $\dfrac{\partial\check{\mathbf{h}}}{\partial\theta}$.

In a similar way that MS methods represent (interpolate) the coarse scale solution at the fine scale, the partial derivative of the prolongation operator w.r.t. the model parameters, $\dfrac{\partial\mathbf{P}}{\partial\theta}$ in Eq. (7.35), allows the representation of the model parameters at the fine-scale, even though the primary variables are not solved at the model scale.

The gradient of Eq. (7.31) w.r.t. the model parameters at fine scale can be computed as

$$\begin{aligned}\nabla_\theta O_{MS} = \mathbf{C}_\theta^{-1}\left(\theta - \theta_{prior}\right) + \left(\frac{d\check{\mathbf{h}}}{d\theta}\right)^T \mathbf{C}_D^{-1}\left(\check{\mathbf{h}}(\check{\mathbf{x}},\theta) - \check{\mathbf{d}}_{obs}\right) = \\ \mathbf{C}_\theta^{-1}\left(\theta - \theta_{prior}\right) + \mathbf{G}'^T\check{\mathbf{m}},\end{aligned} \tag{7.36}$$

where

$$\check{\mathbf{m}} = \mathbf{C}_D^{-1}\left(\check{\mathbf{h}}(\check{\mathbf{x}},\theta) - \check{\mathbf{d}}_{obs}\right). \tag{7.37}$$

The product $\mathbf{G}'^T\check{\mathbf{m}} = \left(\check{\mathbf{m}}^T\mathbf{G}'\right)^T$ can be solved at costs proportional to one coarse-scale backward simulation, regardless of the number of model parameters, via the MS adjoint method presented in [23], by pre-multiplying Eq. (7.35) by the transpose of Eq. (7.36), defining

$$\check{\mathbf{z}} = (\mathbf{RAP})^{-T}\left(\frac{\partial\check{\mathbf{h}}}{\partial\check{\mathbf{x}}} + \frac{\partial\check{\mathbf{h}}}{\partial\mathbf{x}'}\mathbf{P}\right)^T\check{\mathbf{m}}, \tag{7.38}$$

and rearranging the terms, it follows that

$$\check{\mathbf{m}}^T\mathbf{G}' = \check{\mathbf{z}}^T\alpha + \beta\frac{\partial\mathbf{P}}{\partial\theta}\check{\mathbf{x}} + \check{\mathbf{m}}^T\frac{\partial\check{\mathbf{h}}}{\partial\theta}, \tag{7.39}$$

where

$$\alpha = \mathbf{R}\frac{\partial \mathbf{A}}{\partial \theta}\mathbf{P}\check{\mathbf{x}}, \tag{7.40}$$

and

$$\beta = \mathbf{z}^T\mathbf{R}\mathbf{A} - \check{\mathbf{m}}^T\frac{\partial \check{\mathbf{h}}}{\partial \mathbf{x}'}. \tag{7.41}$$

SCALING OPERATORS PARTIAL DERIVATIVE COMPUTATION

The partial derivative computation of MSFV basis functions was originally discussed in [21] and recast in an algebraic, general mathematical framework expressed in terms of $\mathbf{P}$ in [23]. An efficient algorithm that computes the product $\beta\frac{\partial \mathbf{P}}{\partial \theta}\mathbf{x}$ in a backward-fashion was originally introduced in [23]. Here, we recast this computation in terms of the wirebasket [46] submatrices of $\mathbf{A}$ and $\frac{\partial \mathbf{g}}{\partial \theta}$.

[Partial derivative of MSFV prolongation operator w.r.t. θ] Let $\mathbf{A} \in \mathbb{R}^{N_F \times N_F}$ be the elliptic discrete operator (Eq. (7.13)), $\mathscr{P} \in \mathbb{R}^{N_F \times N_F}$ the permutation matrix that reorders from wirebasket to natural ordering, and $\frac{\partial \mathbf{g}^{\varphi}}{\partial \theta} \in \mathbb{R}^{N_F \times N_\theta}$ the partial derivative of Eq. (7.13) w.r.t. θ, then $\frac{\partial \mathbf{P}}{\partial \theta}$, the partial derivative of the MSFV prolongation operator w.r.t. θ can be computed as

$$\frac{\partial \mathbf{P}}{\partial \theta} = \mathscr{P}\begin{bmatrix} \mathbf{A}_{II}^{-1}\left(\mathbf{A}_{IE}\check{\mathbf{A}}_{EE}^{-1}\frac{\partial \mathbf{g}_E^{\varphi}}{\partial \theta} - \frac{\partial \mathbf{g}_I^{\varphi}}{\partial \theta}\right) \\ -\check{\mathbf{A}}_{EE}^{-1}\frac{\partial \mathbf{g}_E^{\varphi}}{\partial \theta} \\ \mathbf{0} \end{bmatrix}, \tag{7.42}$$

where $\frac{\partial \mathbf{g}_I^{\varphi}}{\partial \theta} \in \mathbb{R}^{N_I \times N_\theta}$ and $\frac{\partial \mathbf{g}_E^{\varphi}}{\partial \theta} \in \mathbb{R}^{N_E \times N_\theta}$ are the partial derivative submatrices from $\frac{\partial \mathbf{g}^{\varphi}}{\partial \theta} \in \mathbb{R}^{N_F \times N_\theta}$ after the wirebasket reordering of Eq. (7.13) and the application of the MSFV localization assumptions and boundary conditions.

Proof 1 *The basis function can be computed from Eq. (7.18), after applying the MSFV localization assumptions and applying the appropriate boundary conditions*

$$\mathscr{P}^T\mathbf{g}^{\varphi}\left(\mathbf{x} = \boldsymbol{\varphi}, \theta\right) = \begin{bmatrix} \mathbf{g}_I^{\varphi} \\ \mathbf{g}_E^{\varphi} \\ \mathbf{g}_V^{\varphi} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{II} & \mathbf{A}_{IE} & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{A}}_{EE} & \mathbf{A}_{EV} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_{VV} \end{bmatrix}\begin{bmatrix} \\ \\ \end{bmatrix} - \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \tag{7.43}$$

whose partial derivative w.r.t. θ*, reads*

$$\begin{bmatrix} \frac{\partial \mathbf{g}_I^{\varphi}}{\partial \theta} \\ \frac{\partial \mathbf{g}_E^{\varphi}}{\partial \theta} \\ \frac{\partial \mathbf{g}_V^{\varphi}}{\partial \theta} \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathbf{A}_{II}}{\partial \theta} + \frac{\partial \mathbf{A}_{IE}}{\partial \theta} \\ \frac{\partial \tilde{\mathbf{A}}_{EE}}{\partial \theta} + \frac{\partial \mathbf{A}_{EV}}{\partial \theta} \\ \mathbf{0} \end{bmatrix}. \tag{7.44}$$

The partial derivative of Eq. (7.17) *w.r.t.* θ *is*

$$\frac{\partial \mathbf{P}}{\partial \theta} = \mathscr{P}\begin{bmatrix} \mathbf{A}_{II}^{-1}\left(\mathbf{A}_{IE}\left(\check{\mathbf{A}}_{EE}^{-1}\left(-\dfrac{\partial \check{\mathbf{A}}_{EE}}{\partial \theta} - \dfrac{\partial \mathbf{A}_{EV}}{\partial \theta}\right)\right) - \dfrac{\partial \mathbf{A}_{IE}}{\partial \theta} - \dfrac{\partial \mathbf{A}_{II}}{\partial \theta}\right) \\ \tilde{\mathbf{A}}_{EE}^{-1}\left(-\dfrac{\partial \tilde{\mathbf{A}}_{EE}}{\partial \theta} - \dfrac{\partial \mathbf{A}_{EV}}{\partial \theta}\right) \\ \mathbf{0} \end{bmatrix}. \tag{7.45}$$

Substituting Eq. (7.44) *in Eq.* (7.45)*, it follows that*

$$\frac{\partial \mathbf{P}}{\partial \theta} = \mathscr{P}\begin{bmatrix} \mathbf{A}_{II}^{-1}\left(\mathbf{A}_{IE}\tilde{\mathbf{A}}_{EE}^{-1}\dfrac{\partial \mathbf{g}_E^{\varphi}}{\partial \theta} - \dfrac{\partial \mathbf{g}_I^{\varphi}}{\partial \theta}\right) \\ -\tilde{\mathbf{A}}_{EE}^{-1}\dfrac{\partial \mathbf{g}_E^{\varphi}}{\partial \theta} \\ \mathbf{0} \end{bmatrix}.$$

Hence, likewise the MSFV prolongation operator can be fully determined directly from the fine-scale system matrix, while the partial derivative of the prolongation operator w.r.t. the model parameters can fully determined from both $\mathbf{A}$ and the partial derivative matrix $\dfrac{\partial \mathbf{g}}{\partial \theta}$.

Note that, even though Proposition 7.4.1 indicates the important ability of determining $\dfrac{\partial \mathbf{P}}{\partial \theta}$ from $\dfrac{\partial \mathbf{g}}{\partial \theta}$, it does not provide enough information about how to efficiently compute this partial derivative. It is discussed in [23] how to efficiently compute the left/right multiplication of $\dfrac{\partial \mathbf{P}}{\partial \theta}$ in the context of, respectively, the direct and adjoint methods. Algorithm 4 in that paper presents and efficient way to compute the product $\beta \dfrac{\partial \mathbf{P}}{\partial \theta}\check{\mathbf{x}}$, at costs proportional to the number of coarse cells and independent of the number of parameters, suitable to be used in combination with Eq. (7.39) for its efficient solution.

7.5. NUMERICAL EXPERIMENTS

We focus our analysis on incompressible, single phase-flow. In our experiments, a simple synthetic model is considered as proof of concept (see Fig. A.5). It is a 2D inverted five-spot model, consisting of a 21×21 equidistant Cartesian mesh with grid block sizes of $33.3 \times 33.3 \times 2$ m. The reservoir porosity is constant and equal to 0.3. A fluid dynamic viscosity of 0.5×10^{-3} Pa.s is considered. The wells are controlled by bottom-hole pressure (BHP). The injection pressure is 35 MPa and the production wells' BHP is 25 MPa.

The uncertainty around the absolute permeability distribution is represented by an ensemble of different permeability realizations. The ensemble is generated via the decomposition of a reference permeability 'image'using Principal Component Analysis (PCA) parameterization [48]. Fig. A.8 illustrates four different permeability realizations from

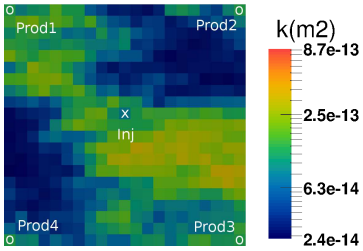


Figure 7.2: The synthetic inverted five-spot model used in the numerical experiments. One of the 1,000 permeability realizations is shown.

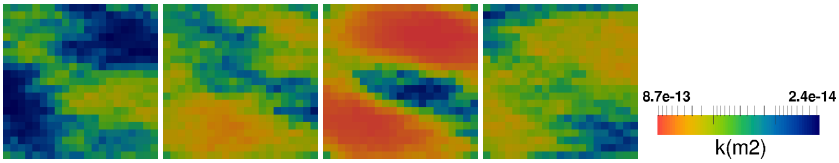


Figure 7.3: Four different permeability realizations from the ensemble of 1,000 members of the toy model numerical experiment.

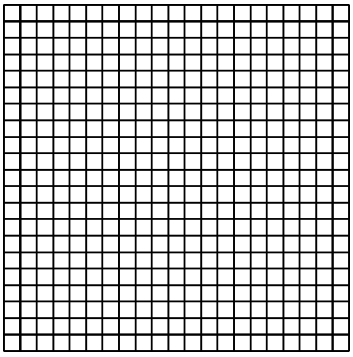
the ensemble of 1,000 equiprobable permeability realizations. In order to focus on the MS aspect of the data assimilation process, we assume that pressures can be approximately extracted from a time-lapse seismic survey [49–53]. However, it is important to note that this is not a limitation. If one is interested to perform the data assimilation in different domains, say, in the impedances domain [2], the additional complexity involved is the appropriate incorporation of seismic forward model equations in the forward model set of equations [54] and, consequentially, the computation of the appropriate partial derivative information necessary to compute Eq. (7.20).

The 'true'observed data, $\mathbf{p}_{true}$, is obtained from a twin experiment, where a MS simulation is run using a permeability field randomly chosen from an ensemble of equiprobable model realizations. The coarse observed data is the coarse scale pressure $\check{\mathbf{x}}$ computed with Eq. (7.14), while the (hypothetical, for comparison purposes) fine observed data is computed with Eq. (7.13). The pressure measurement errors are considered to follow a spherical covariance model with a 1% standard deviation in all experiments and correlation lengths equal to 1 grid-block size. Noise is added to the data by setting

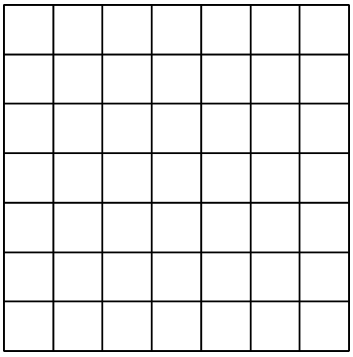
$$\mathbf{p}_{obs} = \mathbf{p}_{true} + \sqrt{\mathbf{C}_D}\mathbf{z}, \tag{7.46}$$

where $\mathbf{z}$ is sampled from $\mathcal{N}(\mu = 0, \sigma^2 = 1)$ and $\sqrt{\mathbf{C}_D}$ is computed from a Cholesky decomposition. More details on the procedure can be found in [40]. The resulting noisy observed coarse and fine pressure fields are illustrated, respectively, in Fig. 7.4d and Fig. 7.4c.

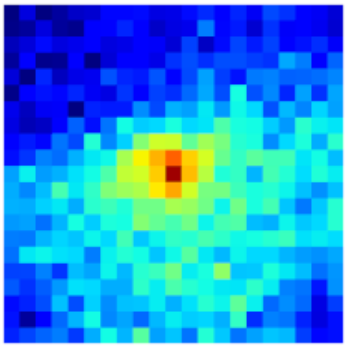
We consider the observation grid (Fig. 7.4b) where the observed data is represented to be three times coarser than the model grid (Fig. 7.4a) in the x and y directions. Hence, it has 7×7 grid-blocks with grid block size $99.9 \times 99.9 \times 2$ m.



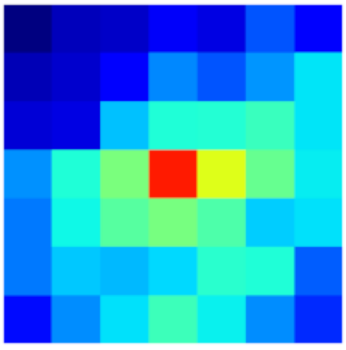
(a) Model grid



(b) Observation grid



(c) Fine-scale observation



(d) Coarse-scale observation

Figure 7.4: Schematic representation of the model (fine) grid (a) and observation (coarse) grid (b). Also, the (noisy) pressure data distribution observed at the fine-grid (hypothetical complete observations) (c) and at the actual observation grid resolution (d).

The covariance matrix $\mathbf{C}_\theta$ is computed from the ensemble of realizations as

$$\mathbf{C}_\theta = \frac{1}{N_e - 1}\left(\Theta - \boldsymbol{\mu}\mathbf{e}^T\right)\left(\Theta - \boldsymbol{\mu}\mathbf{e}^T\right)^T, \tag{7.47}$$

where Θ is the $N_F \times N_e$ matrix whose j-th column is given by the member of the ensemble θ_j, $j \in \{1, ..., N_e\}$,

$$\mu = \frac{1}{N_e}\sum_{j=1}^{N_e}\theta_j, \tag{7.48}$$

is the ensemble mean, and $\mathbf{e} = [\mathbf{1}, ..., \mathbf{1}]^T$ is a vector of ones of size $N_e \times 1$. The prior is taken to be the ensemble mean,

$$\theta_{prior} = \mu. \tag{7.49}$$

In the fine scale data assimilation strategy, an adjoint model [37] is used to calculate the OF gradient given by Eq. (7.6). In the multiscale strategy, we employ the MS adjoint gradient computation depicted in Algorithm 4 in [23]. Because the spatially distributed observed data at the coarse scale is the primary variable itself, in Eq. (7.31) we have

$$\frac{\partial \check{\mathbf{h}}}{\partial \check{\mathbf{x}}} = \check{\mathbf{I}}, \tag{7.50}$$

where $\check{\mathbf{I}}$ is the $N_C \times N_C$ identity matrix, and

$$\frac{\partial \check{\mathbf{h}}}{\partial \mathbf{x}'} = \mathbf{0}. \tag{7.51}$$

when pressure is observed at the coarse scale, and

$$\frac{\partial \mathbf{h}}{\partial \mathbf{x}} = \mathbf{I}, \tag{7.52}$$

when pressure is observed at the fine-scale.

Also, because in this case the relationship between the primary variables and the outputs is not a function of the parameter, it follows that

$$\frac{\partial \check{\mathbf{h}}}{\partial \theta} = \check{\mathbf{0}}, \tag{7.53}$$

and

$$\frac{\partial \mathbf{h}}{\partial \theta} = \mathbf{0}. \tag{7.54}$$

We utilize the limited memory Broyden-Fletcher-Goldfarb-Shanno (LBFGS) algorithm as presented in [41], as it is the most efficient algorithm to deliver optimization results for the solution of Eq. (7.7) [55].

7.5.1. CONSTRUCTION OF SCALING OPERATORS

OBSERVED DATA DOWNSCALING

Two different approaches on how to deal with the scale dissimilarity via downscaling are considered here. In the first one, we downscale the response measured at the coarse scale by setting

$$\mathbf{D} = \mathbf{R}^T. \tag{7.55}$$

where $\mathbf{R}$ is the MSFV restriction operator. This strategy can be viewed as a constant interpolation of the coarse scale observations at the fine-scale model scale.

In the second strategy, we build a multiscale prolongation operator $\mathbf{P}_{prior} = \mathbf{P}\left(\theta_{prior}\right)$, whose columns are comprised of local multiscale basis functions [20], and prolong (interpolate) the coarse scale information by setting

$$\mathbf{D} = \mathbf{P}_{prior}. \tag{7.56}$$

Note that $\mathbf{P}_{prior}$ is static and can be viewed as a MS downscaling operator.

In the aforementioned strategies, Eq. (7.6) can be used by making $\mathbf{d}_{obs} = \mathbf{d}_{obs}{}'$ and a conventional gradient-based optimization to solve Eq. (7.7) is run at the model (fine) scale.

MODEL RESPONSE UPSCALING

Two upscaling strategies are considered. In the first one, a simple arithmetic average is applied by setting

$$\mathbf{U} = \mathbf{M}, \tag{7.57}$$

where,

$$\mathbf{M}\left(c, f\right) = \begin{cases} \dfrac{1}{N_F^C}, & \text{if } f \in \Omega_c \\ 0, & \text{otherwise} \end{cases}. \tag{7.58}$$

In Eq. (7.58), N_F^C is the number of fine grid cells within a given coarse cell C, Ω_c is the c-th primal coarse grid domain and f is the fine-grid cell index.

In the second upscaling strategy, we again build a prolongation operator $\mathbf{P}_{prior}$ based on θ_{prior} and upscale the observed response by setting

$$\mathbf{U} = \mathbf{P}^{\dagger}, \tag{7.59}$$

where the $\dagger$ symbol denotes the Moore-Penrose pseud-inverse. Here, we construct $\mathbf{P}^{\dagger}$ from its truncated singular value decomposition (TSVD) [56]

$$\mathbf{P}^{\dagger} = \mathbf{\Sigma}\mathbf{\Lambda}\mathbf{\Delta}^T = \begin{bmatrix}\mathbf{\Sigma}_p & \mathbf{\Sigma}_0\end{bmatrix}\begin{bmatrix}\mathbf{\Lambda}_p & \mathbf{0} \\ \mathbf{0} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{\Delta}_p^T \\ \mathbf{\Delta}_0^T\end{bmatrix} = \mathbf{\Sigma}_p\mathbf{\Lambda}_p\mathbf{\Delta}_p^T \tag{7.60}$$

where $\mathbf{\Sigma} \in \mathbb{R}^{N_F \times N_F}$ and $\mathbf{\Delta} \in \mathbb{R}^{N_C \times N_C}$ are orthonormal matrices, $\mathbf{\Lambda} \in \mathbb{R}^{N_F \times N_C}$ is a diagonal matrix containing the singular values of $\mathbf{P}$, and the subscript p indicates the first p columns of the matrices corresponding the p non-zero singular values.

7.5.2. MAXIMUM *a posteriori* PROBABILITY (MAP) ESTIMATE

In this section we assess the performance of computing the MAP estimate via the newly introduced method in comparison the to down/upscaling strategies discussed before. Therefore, six different data assimilation strategies are considered, namely:

1. fine-scale data assimilation with constant prolongation downscaling of observed pressure (Eq. (7.55));
2. fine-scale data assimilation with prior MS prolongation downscaling of observed pressure (Eq. (7.56));
3. fine-scale data assimilation with arithmetic average to upscale the simulated pressure (Eq. (7.58));
4. fine-scale data assimilation with pseudo-inverse of MS prolongation to upscale the simulated pressure (Eq. (7.59));
5. multiscale data assimilation strategy introduced in this work;
6. fine-scale data assimilation with complete observations available at the model (fine) scale.

The latter, a hypothetical situation, is considered as the reference case, as if enough resolution was available to resolve the observed property at the (fine) model scale. Also, note that MS operators are used in strategies 1, 2 and 4. For comparison purposes, we consider the objective function normalized by the number of data points N_d. Furthermore, according to [39, 40], an acceptable data match is achieved when $\frac{O}{N_d} \approx 0.5$.

Firstly, we present a qualitative discussion based on the MAP conditioned permeability fields and final matched pressure fields in comparison to the respective 'true'permeability and pressure fields. The results for the fine-scale, complete observation data assimilation exercise is illustrated in Fig. 7.5, followed by the results from the downscaling and upscaling data conciliation strategies, represented, respectively, by Fig. 7.6 and Fig. 7.7. Lastly, the results of the data assimilation using our MS data assimilation strategy are illustrated in Fig. 7.8.

From a qualitative point of view, the matched responses from all data assimilation strategies, except the responses obtained by the constant interpolation downscaling (Fig. 7.6g) and arithmetic average upscaling (Fig. 7.7g), are fairly similar to the observed data. The pressure matches are both in accordance with the fine-scale pressure match (Fig. 7.5f) and with the 'true' pressure field. However, the simpler up/downscaling strategies result in somewhat poorer matches around the injection well.

Also, it is possible to observe that, from the point of view of the conditioned permeability fields, all assimilation strategies were capable of recovering the main features. Furthermore, not much difference is noted in the results when comparing the upscaling to the downscaling matched permeability fields. In order to better assess the quality of the parameter matches, we investigate the permeability distribution from the different matching exercises. Therefore, the density functions of the matched permeability fields are plotted in Fig. 7.9. It is possible to note that, even though the initial permeability distribution is considerably far from the true model (due to the rather homogeneous

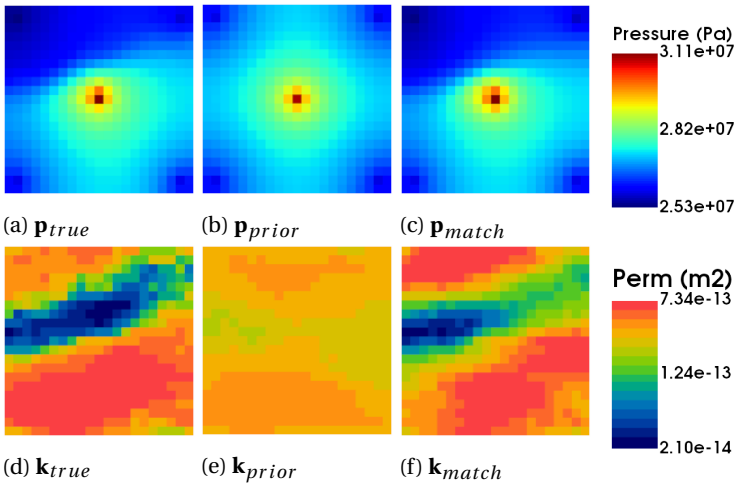


Figure 7.5: Data assimilation results, fine-scale data assimilation, complete, fine-scale observations. In the first row, the (a) true, (b) initial and (c) matched pressure fields are shown. In the second row, (d) the 'true', (e) the prior and (f) the conditioned permeability fields are shown.

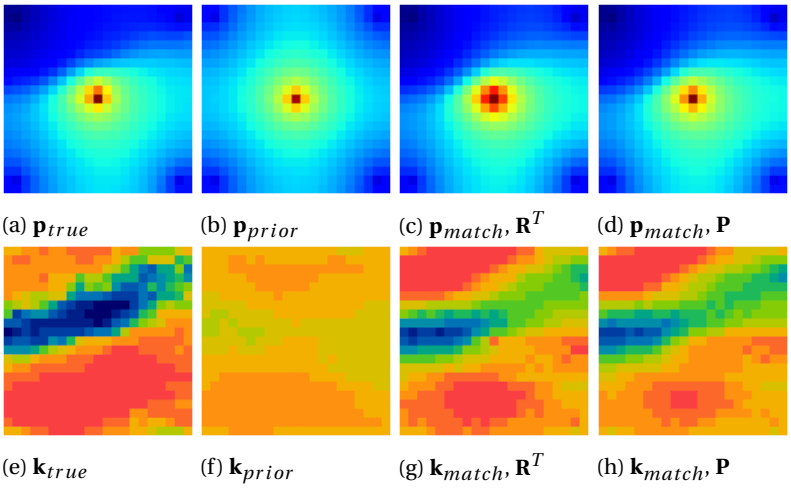


Figure 7.6: Data assimilation results, fine-scale data assimilation, downscaling of data observations. In the first row, the (a) true, (b) initial, (c) matched using the constant interpolation ($\mathbf{R}^T$) and (d) matched using the MS prolongation operator ($\mathbf{P}$) pressure fields are shown. In the second row, (d) the 'true', (e) the prior, (g) the conditioned using $\mathbf{R}^T$ and (g) the conditioned using $\mathbf{P}$ permeability fields are shown. The color maps follow the color bar found in Fig. 7.5

.

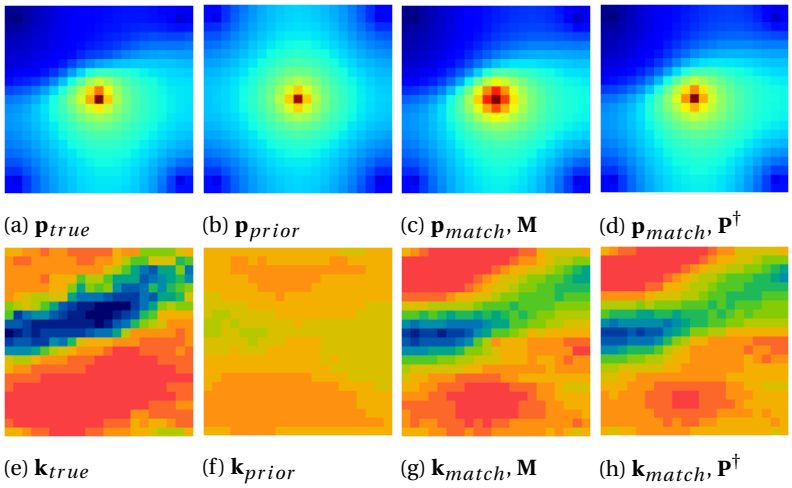


Figure 7.7: Data assimilation results, fine-scale data assimilation, upscaling of model responses. In the first row, the (a) true, (b) initial, (c) matched using arithmetic average ($\mathbf{M}$) and (d) matched using the MS prolongation operator pseudo-inverse ($\mathbf{P}^{\dagger}$) pressure fields are shown. In the second row, (d) the 'true', (e) the prior, (g) the conditioned using $\mathbf{M}$ and (g) the conditioned using $\mathbf{P}^{\dagger}$ permeability fields are shown. The color maps follow the color bars found in Fig. 7.5.

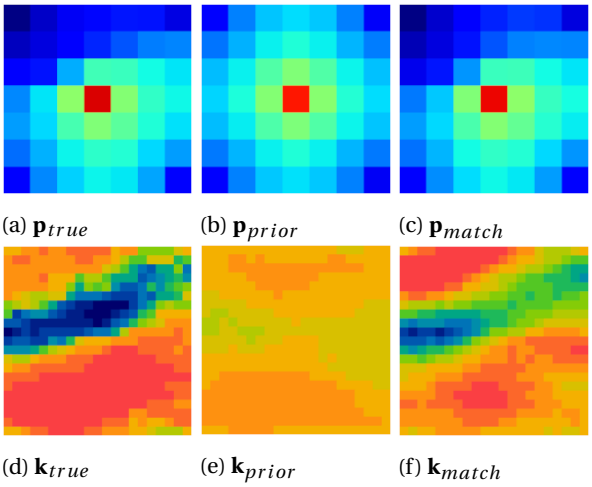


Figure 7.8: Data assimilation results, multiscale data assimilation. In the first row, the (a) true, (b) initial and (c) matched pressure fields are shown. In the second row, (d) the 'true', (e) the prior and (f) the conditioned permeability fields are shown. The color maps follow the color bars found in Fig. 7.5. Note that MS assimilation provides a fine-scale improved permeability field (f) while the observation is at coarse scale.

prior used in the MAP), the complete observation, fine-scale strategy is capable of reproduction of the reference permeability density function. But, more importantly, the MS data assimilation, with coarse scale only observations, can also provide a permeability field whose density function is consistent with the 'true'permeability density function. Also, it can be noted that the permeability fields obtained by the other strategies are also consistent with the 'true'permeability distribution.

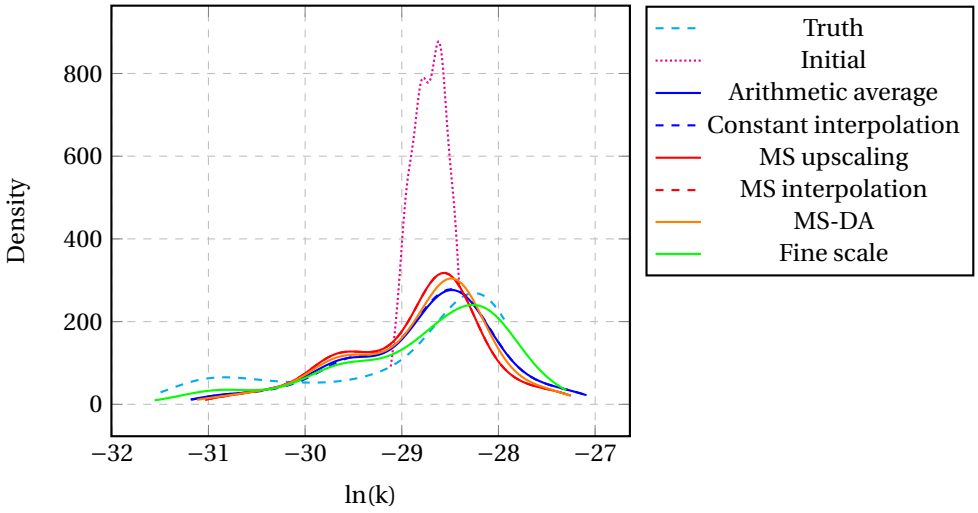


Figure 7.9: Permeability conditioned marginal PDFs for the different data assimilation strategies.

We also analyse the optimization convergence behaviour shown in Fig. 7.10 for quantitative assessment of the match. The fine-scale reference data assimilation reaches a normalized OF value very close to the ideal value of 0.5, while all other data assimilation strategies reach values relatively higher, with the simpler constant interpolation and arithmetic average scaling strategies reaching slightly higher OF values. It is important to note that the optimization behaviour is remarkably similar for all up/downscaling strategies, as well as for the MS data assimilation strategy here presented.

7.5.3. UNCERTAINTY QUANTIFICATION

A RML is run for 100 randomly chosen permeability realization from the 1,000 members ensemble (Fig. A.8), for each data conciliation strategy. In order to estimate the conditioned permeability distribution for each permeability realization, a LBFGS optimization is run for each chosen member. The results for the exercise are shown in Fig. 7.11.

It can be observed that the permeability marginal PDFs conditioned to the pressure data obtained by the MS data assimilation here introduced (Fig. 7.11e) are closer to the reference fine-scale conditioned PDFs (Fig. 7.11f). Additionally, by observing the spread of the conditioned PDFs obtained from the RML employing the up/downscaling strategies, one can note that the MS strategy is also capable of somewhat better representing the uncertainty.

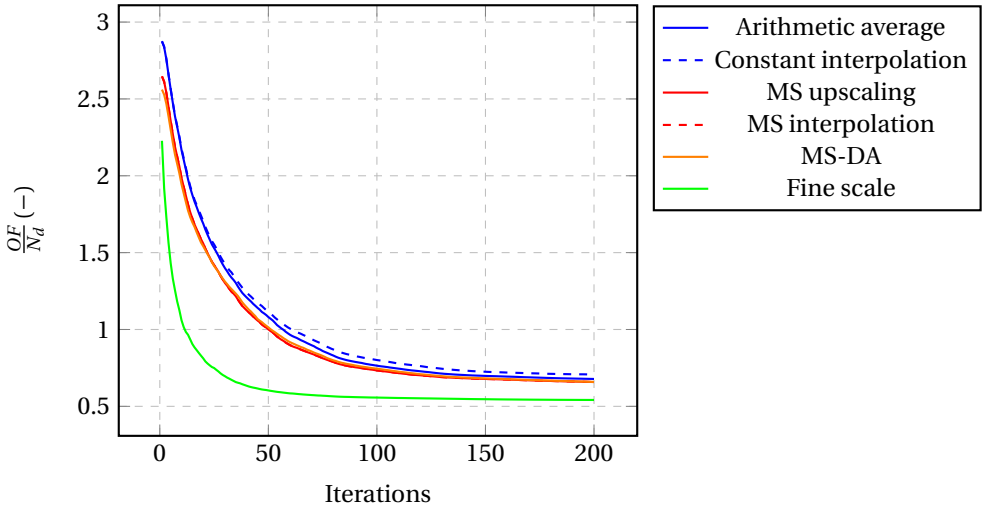


Figure 7.10: Optimization performance of the data assimilation utilizing the 6 different scale conciliation strategies as presented in this section. Note that the FS represent the hypothetical case where both observed data and model parameters are at fine scale (i.e. item 6 on our list of strategies).

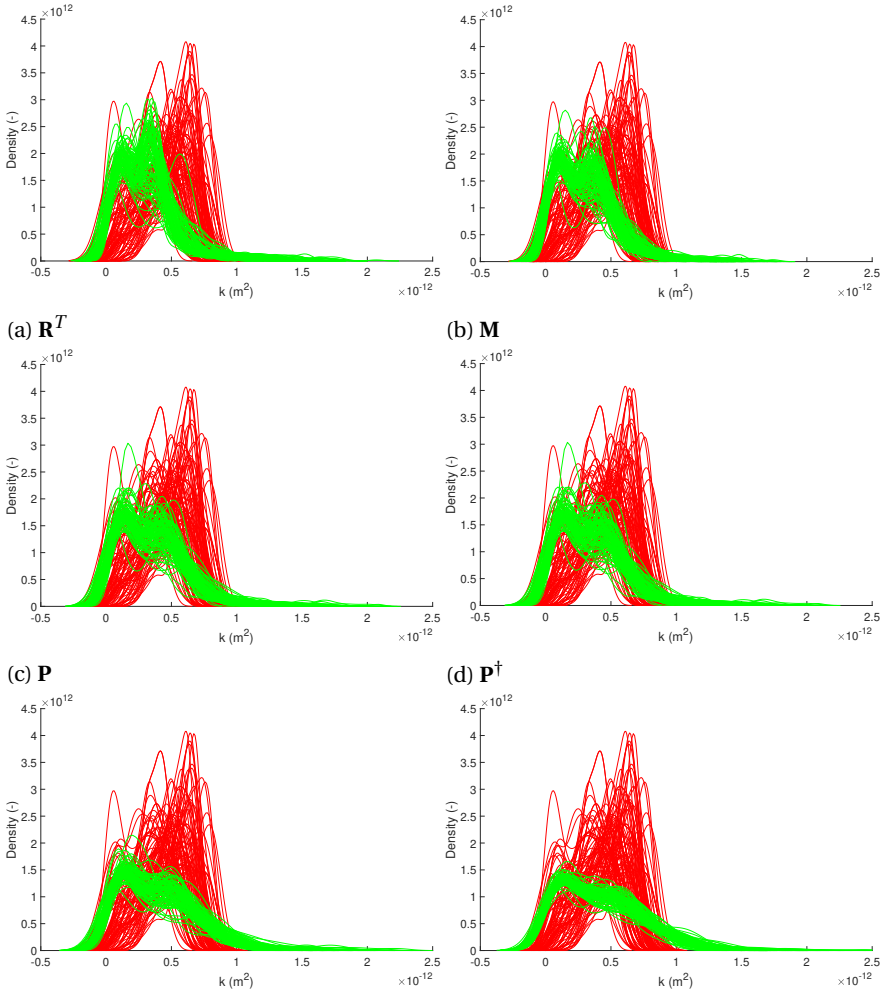
7.5.4. DISCUSSION

Firstly, even though all data assimilation strategies were capable of achieving similar MAP estimates, one should note that the synthetic case used in the experiments has low permeability contrasts. Moreover the five-spot configuration is very simple and the well spacing is relatively dense compared to the characteristic size of the heterogeneities. Nonetheless, given the good results observed in the employment of our MS data assimilation strategy, we believe that the performance of the method in more challenging scenarios is worth investigating. A systematic study of the effects of the underlying geological complexity on the MS assimilation procedure is necessary.The MS ability to preserve fine-scale features is expected to allow for more detailed description of the fine-scale uncertain parameters. Additionally, we emphasize the importance of the proper representation of the measurement errors at different scales. One must take into account the data redundancy in the case of downscaling the observed data to the model scale.

One could consider a third, fine-scale only, MS-based approach, based on the reconstruction of the prolongation operator at every optimizer iteration γ, so that changes in the permeability during the optimization process are also captured by the basis functions update. Hence, one could write

$$\mathbf{d}^{\prime\gamma}_{obs} = \mathbf{P}^{\gamma}\mathbf{d}^{\breve{}}_{obs}, \tag{7.61}$$

where $\mathbf{P}^{\gamma}$ is the reconstructed prolongation operator at every optimization iteration γ. This can only be achieved at the expense of the reconstruction of the basis function every γ . We performed studies (not reported here) where we neglect the partial derivative of $\mathbf{P}$ w.r.t θ but we did update $\mathbf{P}$. Similar results were obtained when $\mathbf{P}$ is not updated and only based on the prior (Eq. (7.56)), as reported here. Moreover, it is discussed in [23]



(a) $\mathbf{R}^T$ (b) $\mathbf{M}$

(c) $\mathbf{P}$ (d) $\mathbf{P}^\dagger$

(e) Full Multiscale (f) Fine scale

Figure 7.11: RML probability density functions for 100 permeability realizations randomly chosen from the original ensemble of 1,000 realizations (see Fig. A.8). The curves in red represent the prior permeability distributions, while the curves in green the conditioned permeability distributions. (a) Constant interpolation downscaling ($\mathbf{R}^T$), (b) arithmetic average downscaling ($\mathbf{M}$), (c) MS prolongation operator downscaling, (d) MS prolongation operator upscaling $\mathbf{P}^\dagger$, (e) the multiscale data assimilation strategy and (f) the reference fine-scale.

how to efficiently compute $\partial \mathbf{P}/\partial \theta$. This can be an alternative to further take advantage of MS principles even when a MS forward simulation is not available.

In this work we employ a two-stage MS simulation strategy, and consequently a two-stage MS gradient computation strategy. We make the primal MS coarse grid to be coincident to the observation grid resolution. However, the idea of the MS data assimilation can be extended to seamlessly address data available at multiple scales, or even consider one, or multiple MS grid resolution(s) for assimilation purposes only and different one(s) for the forward simulation. To this end, multilevel multiscale strategies [57] could be applied. Following the same multilevel multiscale strategy, data acquired at different scales (e.g. electromagnetics, high resolution close to the well, along with seismic data, low resolution in the vertical direction) could also be seamlessly and simultaneously be assimilated.

Following our studies, and also reported by [23] and [21], it can be noted that MSFV gradients can be less accurate for highly heterogeneous media. In addition, one may want to have error control on the MSFV gradient quality for practical applications. Furthermore, LBFGS proposes under/overshooting updates, mainly close to the wells [58], which also configures a challenging scenario for the MSFV gradient computation. These challenges can be addressed from the optimization point of view or from the gradient computation perspective. The former can be considered via data misfit damping or parameter constraints [58]. The latter by improved MS gradient quality, via more accurate MSFV solutions [59, 60]. An iterative MSFV gradient computation, following the solution strategy proposed by [60] could allow for additional error control over the gradient computation.

REFERENCES

[1] R. J. de Moraes, H. Hajibeygi, and J. D. Jansen, *A multiscale method for data assimilation,* in *ECMOR XVI-16th European Conference on the Mathematics of Oil Recovery, Barcelona, Spain, 3-6 September* (2018).

[2] A. A. Emerick, R. Moraes, J. Rodrigues, *et al.*, *History matching 4d seismic data with efficient gradient based methods,* in *EUROPEC/EAGE Conference and Exhibition* (Society of Petroleum Engineers, 2007).

[3] D. Ullmann De Brito, L. Caletti, R. Moraes, *et al.*, *Incorporation of 4d seismic in the re-construction and history matching of marlim sul deep water field flow simulation model,* in *SPE EUROPEC/EAGE Annual Conference and Exhibition* (Society of Petroleum Engineers, 2011).

[4] D. Ullmann De Brito, R. Moraes, A. A. Emerick, *et al.*, *The marlim field: incorporating time-lapse seismic in the assisted history matching,* in *SPE Latin American and Caribbean Petroleum Engineering Conference* (Society of Petroleum Engineers, 2010).

[5] A. A. Emerick, A. C. Reynolds, *et al.*, *History-matching production and seismic data in a real field case using the ensemble smoother with multiple data assimilation,* in *SPE Reservoir Simulation Symposium* (Society of Petroleum Engineers, 2013).

[6] J.-A. Skjervheim, G. Evensen, S. I. Aanonsen, B. O. Ruud, T.-A. Johansen, *et al., Incorporating 4d seismic data in reservoir simulation models using ensemble kalman filter,* in *SPE Annual Technical Conference and Exhibition* (Society of Petroleum Engineers, 2005).

[7] V. E. J. Haugen, L.-J. Natvik, G. Evensen, A. M. Berg, K. M. Flornes, G. Naevdal, *et al., History matching using the ensemble kalman filter on a north sea field case,* in *SPE Annual Technical Conference and Exhibition* (Society of Petroleum Engineers, 2006).

[8] Y. Zhang, D. S. Oliver, *et al., History matching using the ensemble kalman filter with multiscale parameterization: A field case study,* SPE Journal **16**, 307 (2011).

[9] I. G. Graham, T. Y. Hou, O. Lakkis, and R. Scheichl, *Numerical analysis of multiscale problems*, Vol. 83 (Springer Science & Business Media, 2012).

[10] Z. Li, J. C. McWilliams, K. Ide, and J. D. Farrara, *A multiscale variational data assimilation scheme: formulation and illustration,* Monthly Weather Review **143**, 3804 (2015).

[11] J. Wan and N. Zabaras, *A bayesian approach to multiscale inverse problems using the sequential monte carlo method,* Inverse Problems **27**, 105004 (2011).

[12] S. I. Aanonsen, G. Nævdal, D. S. Oliver, A. C. Reynolds, and B. Vallès, *The ensemble Kalman filter in reservoir engineering–a review,* SPE J. **14**, 393 (2009).

[13] K. Fossum and T. Mannseth, *A novel multilevel method for assimilating spatially dense data,* in *ECMOR XVI-16th European Conference on the Mathematics of Oil Recovery* (2018).

[14] T. Mannseth and K. Fossum, *Assimilating spatially dense data for subsurface applications—balancing information and degrees of freedom,* Computational Geosciences , 1 (2018).

[15] K. Fossum and T. Mannseth, *Coarse-scale data assimilation as a generic alternative to localization,* Computational Geosciences **21**, 167 (2017).

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

[17] C. Farmer, *Upscaling: a review,* International journal for numerical methods in fluids **40**, 63 (2002).

[18] C. Frederick and B. Engquist, *Numerical methods for multiscale inverse problems,* arXiv preprint arXiv:1401.2431 (2014).

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

[20] P. Jenny, S. H. Lee, and H. A. Tchelepi, *Multi-scale finite-volume method for elliptic problems in subsurface flow simulation,* J. Comput. Phys. **187**, 47 (2003).

[21] J. Fu, H. A. Tchelepi, and J. Caers, *A multiscale adjoint method to compute sensitivity coefficients for flow in heterogeneous porous media,* Advances in water resources **33**, 698 (2010).

[22] J. Fu, J. Caers, and H. A. Tchelepi, *A multiscale method for subsurface inverse modeling: Single-phase transient flow,* Adv. Water Resour. **34**, 967 (2011).

[23] R. J. de Moraes, J. R. Rodrigues, H. Hajibeygi, and J. D. Jansen, *Multiscale gradient computation for flow in heterogeneous porous media,* Journal of Computational Physics **336**, 644 (2017).

[24] R. Moraes, J. Rodrigues, H. Hajibeygi, J. Jansen, *et al., Multiscale gradient computation for multiphase flow in porous media,* in *SPE Reservoir Simulation Conference* (Society of Petroleum Engineers, 2017).

[25] E. T. Chung, Y. Efendiev, B. Jin, W. T. Leung, and M. Vasilyeva, *Generalized multiscale inversion for heterogeneous problems,* arXiv preprint arXiv:1707.08194 (2017).

[26] Y. Efendiev, J. Galvis, and T. Y. Hou, *Generalized multiscale finite element methods (gmsfem),* Journal of Computational Physics **251**, 116 (2013).

[27] V. Gervais-Couplet, F. Roggero, M. D. Feraille, L. Ravalec-Dupin, A. Seiler, *et al., Joint history matching of production and 4d-seismic related data for a north sea field case,* in *SPE Annual Technical Conference and Exhibition* (Society of Petroleum Engineers, 2010).

[28] O. Gosselin, S. Aanonsen, I. Aavatsmark, A. Cominelli, R. Gonard, M. Kolasinski, F. Ferdinandi, L. Kovacic, K. Neylon, *et al., History matching using time-lapse seismic (huts),* in *SPE Annual Technical Conference and Exhibition* (Society of Petroleum Engineers, 2003).

[29] M. Le Ravalec, E. Tillier, S. Da Veiga, G. Enchéry, and V. Gervais, *Advanced integrated workflows for incorporating both production and 4d seismic-related data into reservoir models,* Oil & Gas Science and Technology–Revue d'IFP Energies nouvelles **67**, 207 (2012).

[30] N. Liu, D. S. Oliver, *et al., Evaluation of monte carlo methods for assessing uncertainty,* SPE Journal **8**, 188 (2003).

[31] G. Evensen, *Data assimilation: the ensemble Kalman filter* (Springer Science & Business Media, 2009).

[32] A. A. Emerick and A. C. Reynolds, *Ensemble smoother with multiple data assimilation,* Computers & Geosciences **55**, 3 (2013).

[33] D. S. Oliver, N. He, and A. C. Reynolds, *Conditioning permeability fields to pressure data,* in *ECMOR V-5th European Conference on the Mathematics of Oil Recovery* (1996).

[34] A. A. Emerick and A. C. Reynolds, *Investigation of the sampling performance of ensemble-based methods with a simple reservoir model,* Computational Geosciences **17**, 325 (2013).

[35] G. Chavent, M. Dupuy, and P. Lemmonier, *History matching by use of optimal theory,* Society of Petroleum Engineers Journal **15**, 74 (1975).

[36] J. D. Jansen, *Gradient-based optimization of flow through porous media, v.3,* (2016).

[37] J. R. P. Rodrigues, *Calculating derivatives for automatic history matching,* Computational Geosciences **10**, 119 (2006).

[38] J. F. B. M. Kraaijevanger, P. J. P. Egberts, J. R. Valstar, and H. W. Buurman, *Optimal waterflood design using the adjoint method,* in *SPE Reservoir Simulation Symposium* (Society of Petroleum Engineers, 2007).

[39] A. Tarantola, *Inverse problem theory and methods for model parameter estimation*, Vol. 89 (siam, 2005).

[40] D. S. Oliver, A. C. Reynolds, and N. Liu, *Inverse theory for petroleum reservoir characterization and history matching* (Cambridge University Press, 2008).

[41] J. Nocedal and S. Wright, *Numerical optimization* (Springer Science & Business Media, 2006).

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

[43] H. Zhou and H. A. Tchelepi, *Operator-based multiscale method for compressible flow,* SPE J. **13**, 523 (2008).

[44] Y. Wang, H. Hajibeygi, and H. A. Tchelepi, *Algebraic multiscale solver for flow in heterogeneous porous media,* J. Comput. Phys. **259**, 284 (2014).

[45] O. Møyner and K.-A. Lie, *A multiscale restriction-smoothed basis method for high contrast porous media represented on unstructured grids,* Journal of Computational Physics **304**, 46 (2016).

[46] H. Zhou, H. A. Tchelepi, *et al.*, *Two-stage algebraic multiscale linear solver for highly heterogeneous reservoir models,* SPE Journal **17**, 523 (2012).

[47] J. Wallis and H. A. Tchelepi, *Apparatus, method and system for improved reservoir simulation using an algebraic cascading class linear solver,* (2010), uS Patent 7,684,967.

[48] J. D. Jansen, *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, 2013).

[49] A. Tura and D. E. Lumey, *Estimating pressure and saturation changes time-lapse avo data,* in *SEG Technical Program Expanded Abstracts 1999* (Society of Exploration Geophysicists, 1999) pp. 1655–1658.

[50] M. Landrø, *Discrimination between pressure and fluid saturation changes from time-lapse seismic data,* Geophysics **66**, 836 (2001).

[51] S. Cole, D. Lumley, M. Meadows, and A. Tura, *Pressure and saturation inversion of 4d seismic data by rock physics forward modeling,* in *SEG Technical Program Expanded Abstracts 2002* (Society of Exploration Geophysicists, 2002) pp. 2475–2478.

[52] D. Lumley, M. Meadows, S. Cole, and D. Adams, *Estimation of reservoir pressure and saturations by crossplot inversion of 4d seismic attributes,* in *SEG Technical Program Expanded Abstracts 2003* (Society of Exploration Geophysicists, 2003) pp. 1513–1516.

[53] C. MacBeth, M. Floricich, and J. Soldo, *Going quantitative with 4d seismic analysis,* Geophysical Prospecting **54**, 303 (2006).

[54] A. A. Emerick, R. Moraes, and J. Rodrigues, *Calculating seismic attributes within a reservoir flow simulator,* in *Latin American & Caribbean Petroleum Engineering Conference* (Society of Petroleum Engineers, 2007).

[55] G. Gao, A. C. Reynolds, *et al.*, *An improved implementation of the lbfgs algorithm for automatic history matching,* SPEJ **11**, 5 (2006).

[56] G. Strang, *Introduction to linear algebra* (Wellesley-Cambridge Press Wellesley, MA, 1993).

[57] M. Cusini, C. van Kruijsdijk, and H. Hajibeygi, *Algebraic dynamic multilevel (adm) method for fully implicit simulations of multiphase flow in porous media,* Journal of Computational Physics **314**, 60 (2016).

[58] G. Gao, A. C. Reynolds, *et al.*, *An improved implementation of the lbfgs algorithm for automatic history matching,* in *SPE Annual Technical Conference and Exhibition* (Society of Petroleum Engineers, 2004).

[59] Y. Wang, H. Hajibeygi, and H. A. Tchelepi, *Monotone multiscale finite volume method,* Computational Geosciences **20**, 509 (2016).

[60] H. Hajibeygi and P. Jenny, *Adaptive iterative multiscale finite volume method,* Journal of Computational Physics **230**, 628 (2011).

8

CONCLUSIONS & RESEARCH PERSPECTIVES

In Section 8.1 a detailed discussion about the findings obtained on each of the chapters presented in this thesis is presented. In Section 8.2, it is discussed how the developments here presented helped to address, or at least further understand, the research questions posed in 1.1. Finally, research directions that could be pursued based on our findings are discussed in 8.3

8.1. CONCLUSIONS

From the developments presented in each chapter, the following conclusion can be drawn.

Chapter 2 Two basic algorithms are necessary to address the adjoint and direct methods. The complexity relies on how the adjoint states are computed, which are delegated to functors. These functors can be specialized according to the forward model and type of responses involved in the derivative computation problem.

Chapter 3 An efficient, general framework that addresses the derivative information computation of sequentially coupled system of equations is presented. The flexibility of the framework is illustrated in small data assimilation and life-cycle optimization studies in which the forward model's flow and transport equations are sequentially coupled. In the applications, it is shown how different objective functions (i.e. NPV and least-squares misfit), parameters (i.e. BHPs and grid-block permeabilities), and responses (i.e. well rates and grid-block pressures) can be accounted for in the computation without any change in the framework. Numerical results of a simple synthetic model demonstrates that the framework can be successfully employed to optimization studies. It is shown that the sequential derivative computation methods deliver similar results compared to the classical FIM methods. Furthermore, the computational asymptotic

analysis of the presented algorithms shows that the sequential derivative computation methods are more computationally efficient when compared to FIM methods.

Chapter 4 A Multiscale gradient computation strategy was developed based on a general algebraic formulation that does not depend on a particular objective function. This was possible by recasting the calculation of derivatives as multiplying a sensitivity matrix and its transpose with arbitrary matrices. The proposed framework is capable of providing different types of gradient information. Such flexibility allows the employment of the framework to any reservoir management study that requires gradient information. Also, the formulation naturally provides both Direct and Adjoint Methods. It was shown, via an asymptotic analysis, that the MS gradient computation strategy can be considerably more efficient than the fine scale strategy. Due to the algebraic nature of the formulation, the presented strategy can be applied to any MS (and multilevel) methodology.

The accuracy of the developed MS gradient computation strategy is studied for a set of examples with increasing complexity. The investigations show that the sources of inaccuracies in the MS solution (e.g. localization assumptions) also result in inaccuracies in the gradient computation. However, fortunately, strategies to improve the MS solution also improve the MS gradient accuracy. Another important observation is that greater angles between MS and fine scale gradient vectors are associated with small values of the gradient norms. Because small gradient norms indicate a weak association between parameters and responses, such differences are typically not very relevant in optimization. Lastly, in our example, the gradient directions, and therefore the convergence behavior of the gradient-based optimization procedure, were similar when either the fine scale or the MS strategy was applied. All this indicates that the presented method allows for accurate, yet less computationally expensive, gradient computations that can be successfully utilized in reservoir management studies.

Extension of the presented algorithm to include iterative multiscale strategies [1] is important for real-field applications. The key component of this extension is to develop an a-priori estimate of the quality of the gradients. Furthermore, for multiphase simulations, the introduced augmented state vector should also include the reconstruction of conservative field [2]. These are subjects of ongoing research.

Chapter 5 We present a framework to efficiently compute gradient information required by computer-assisted history matching studies via an Adjoint, multiscale formulation. No assumption regarding the type of HM parameter or observed data is made during the derivation of the method. Following an IMPES coupling solution strategy, we show how to derive the Adjoint formulation of sequentially coupled simulation strategies. In the formulation of the forward model equations, the flow equation is solved by a MS strategy, while the transport equation is solved in the fine-scale. No assumption is made with respect to which MS method is employed other than that it has to be described in terms of restriction and prolongation operators. A conservative velocity field is reconstructed from the inexact MS pressure field via the solution of Neumann local problems based on the mass-conservative fluxes at the coarse-grid block boundaries. The equations associated with the aforementioned steps are rigorously incorporated in the adjoint formulation by algebraically keeping them in the set of forward model equa-

tions. Only relatively small linear system of equations must be solved, given that they arise from either the independent, local basis function system of equations, the (also independent) local problems solved for the conservative velocity field computation, or the MS coarse system. This fact grants the method efficiency and opportunity for taking advatage of parallel, high-performance computing environments. The partial derivative matrices required by the Adjoint formulation are computed via Automatic Differentiation. The validation numerical experiments show, in a straightforward, fit for purpose setup, that all the complexities involved in the computation of MS gradients of a multiphase flow problem are approprietely captured by the formulation. The expected accuracy of the method is demonstrated when compared to fine-scale gradient.

Chapter 6 We introduce iterative multiscale gradient computation methods based on the iterative Multiscale Finite Volume (i-MSFV) simulation method. By firstly re-casting the i-MSFV method in an algebraic framework, we derive a flexible, multi-purpose derivative computation framework which accounts for the Direct and Adjoint methods. The computational efficiency of the methods is discussed and it is shown that they share advantages equivalent to the i-MSFV method. The methods are validated via numerical experiments against numerical differentiation. It is demonstrated that the newly introduced methods address the challenges encountered by the MSFV gradient computation, specifically associated with high heterogeneity contrast. Accurate gradients, as accurate as fine-scale gradient computation, are obtained for challenging geological models, presenting high permeability contrasts (e.g. the SPE-10 comparative test case). It is shown that accurate gradients are obtained if the i-MSFV model converges to an error tolerance of $1.0e^{-6}$. However, we highlight that such accuracy is not necessary and only a few i-MSFV/smoothing iterations are necessary to acquired reasonably accurate gradients. Further control of the gradient quality via an *a-priori* error estimate along the optimization process should allow for a computationally efficient optimization method. More specifically, a less accurate gradient estimate usually suffice in the early optimization iterations, when the objective function convergence is more steep, while more accurate gradients are required in the more flat regions for a more precise stopping criterion. Hence, the determination of an *a-priori* error estimate should be a key development.

Chapter 7 Our numerical experiments indicate that the presented method has the potential to outperform strategies that rely on upscaling/downscaling of model responses / observed data. An important result is the ability of our MS data assimilation strategy to closely reproduce the reference fine-scale uncertainty quantification results. Applications in more complex cases, and for different types of assimilation problems, should give more insights about the computational and methodological advantages of MS data assimilation, as indicated by the results of the simple example addressed in our study. Our paper demonstrates how to consistently formulate such MS data assimilation strategy, in particular in combination with the use of adjoint-based techniques to efficiently obtain MS gradient information, and in an algebraic framework which allows for implementation in existing computational platforms.

Appendix A In this paper we presented a computationally efficient multiscale based stochastic optimization (MS-StoSAG) workflow, and applied it to two synthetic test cases. Deterministic and robust optimization experiments were performed to illustrate that significant improvements in objective function values (10-15% relative to the reactive control strategy) are attainable in a computationally efficient manner. The validation of our results highlights the accuracy of the multiscale forward simulation models. The results from our experiments show that an approximately five-times speedup can be achieved with scope for even higher speedups with our proposed workflow. We have illustrated that our MS-StoSAG workflow can achieve computationally efficient results, which improves the applicability of robust life-cycle optimization.

8.2. ADDRESSING THE RESEARCH OBJECTIVES

In Section 1.1, the hypotheses explored during the PhD research were posed as research questions. Now, it is discussed how the different developments in the scope of this thesis contributed to mature the answers to those questions.

1. **How can the optimization algorithms applied in the CLRM workflow benefit from MS-based forward models?**

   The forward simulation required by the data assimilation and optimization loops in CLRM management studies represented in Fig. 1.1 benefit from the efficient solution of the flow equation provided by the MSFV method. The sequential coupling strategy discussed in chapter 3 allows for the employment of the efficient solution strategies, more suited for the different underlying physics.

   The MS-StoSAG method introduced in A exploits the fact that basis function can be built only once for the entire optimization process. A offline/online type of strategy, via the i-MSFV method, is employed to compensate for the fact that basis function are not reconstructed/updated. The workflow casts an efficient strategy for robust life-cyle optimization. Remakable speed-ups (4-5 times) are achieved in such types of studies.

2. **Can we design improved data assimilation and optimization strategies based on multiscale simulation principles?**

   Novel, efficient derivative computation strategies for gradient-based optimization algorithms have been introduced. The analytical derivative methods explores a wide spectrum of MS simulation aspects, ranging from the more fundamental ones, discussed in Chapter 4, to the multiphase aspects, as discussed in Chapter 5 and how to further improve the MS gradient to any desired accuracy (Chapter 6). The careful analysis of the algorithms' complexity demostrates the computational efficiency potential of the methodologies.

3. **Can multiscale methods address the multi-level nature of inverse problems?**

   A novel multiscale data assimilation strategy was introduced in Chapter 7. Its potential to robustly address the simulation model and spatially distributed data scale conciliation is specially highlighted in an uncertainty quantification study,

when we show its superior performance when compared to the alternative down / upscaling strategies. Futhermore, based on the mathematical framework presented in that chapter, a multiscale data assimilation framework can be devised as illustrated in Fig. 8.1.

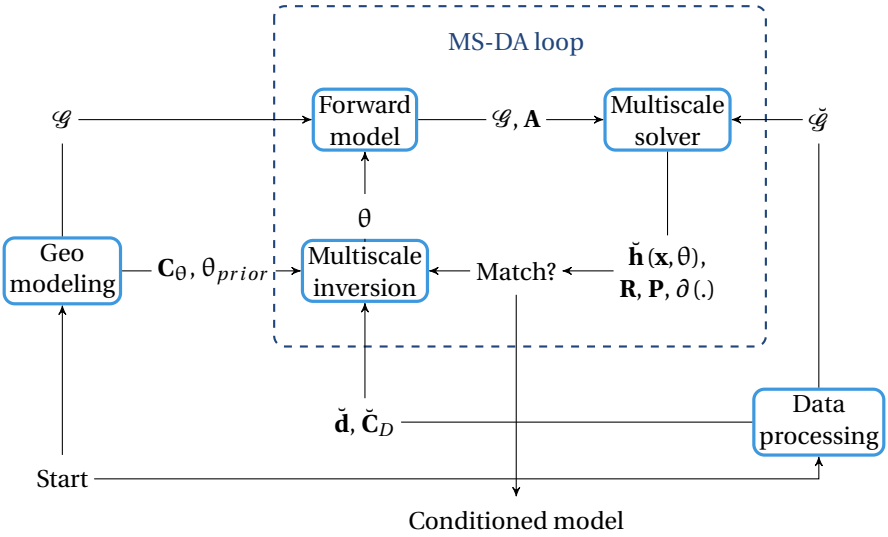


Figure 8.1: Multiscale inversion framework. In the picture $\mathscr{G}$ stands for the geo-model grid and $\check{\mathscr{G}}$ stands for the primal coarse grid.

8.3. RESEARCH PERSPECTIVES

Evaluate the Developments in More Complex Models Even though some complex geological scenarios have been considered as test cases throughout this thesis, most of the developments were tested in relatively simple models. The applicability of the developements here presented have to be tested in more realistic simulation scenarios. More complex fluid models, realistic grids (in terms of size and topology/geometry), well distribution and production planning & scheduling can be not only a challenge for multiscale simulations, but will, consequently, represent challenges for the MS-based strategies presented in this thesis. Fortunately, the developments here presented are general enough to account for such complexities. However the actual performance of them will only be known after their application to real models. This would require developments on both the mathematical and the computational modeling aspects. Some of them are considered below.

An Unified Framework for Derivative Computation of System Responses The general algebraic framework presented in Chapter 2 has been shown to be flexible to accomodate many different derivative computation requirements throughout this thesis. The mathematical framework could be independently deployed, i.e. as a library, and serve as

a computational model for the analytical derivative computation of partial differential equations in general. In other words, the developments here presented are not limited to reservoir managment applications, but could, theoretically, be readily employed is different areas of science, e.g. metheorology, oceanography, system & control, among others.

Efficient, Flexible Partial Derivative Computation The utilization of analytical derivative methods (specifically the adjoint method) have been hindered by the implementation complexity and due to its lack of flexibility. Even though the framework presented in Chapter 2 represents an important step towards an easier implementation of adjoint models, the computation of the required partial derivative matrices requires special attention. Even though AD provides improved flexibiliy, current implementations are not efficient. Modern linearization strategies, like OBL [3], should be investigated as flexible, efficient alternatives for the partial derivative matrices computations.

Efficient Multiscale Derivative Computation for Multiphase Flows & Accounting for More Physics The developments presented in Chapter 5 consider transport to be solved in the fine scale, after the necessary conservative velocity field reconstruction. Even though it is advantageous to represent transport at this scale so that important phenomena are accurately represented, efficient transport solution could be achieved if done at the coarse scale. In that direction, the Adaptive Dynamic Multiscale (ADM) [4] method could be considered as an efficient, yet accurate alternative for the derivative computation for multiphase flows, including compositional simulations [5]. Furthermore, more physics should be accounted for in the forward simulation and the MS derivative computation strategies assessed for those scenarios. For instance, because compressible flows are efficiently addressed by the i-MSFV method [6], the developments presented in Chapter 6 could be readly applied for such types of simulations.

Simultaneous Assimilation of Data Represented in Multiple Scales Even though the multiscale data assimilation methodology introduced in Chapter 7 has been applied in the data assimilation of coarse-scale observations, nothing prevents the framework to be employed in the smulteneous assimilation of data observed at different scales. For instance, eletromagnetics can provide high-resolution observations near to the wellbore, while time-lapse seismic provides global pressure / fluid distribution estimations. The two-level MS assimilation strategy here presented could readily address the challenge of joint assimilation of both types of observations. Furthermore, a multilevel MS simlulation strategy could allow not only for multiple observation scales, but also allow for the simultaneous separation of gelogical scale, dynamic simulation scale and multiple observation scales.

Accounting for Modeling Errors in Data Assimilation Studies Even though modelling errors are acknowledged to occur in the simulation of subsurface processes, those are not account in inverse modelling (i.e. parameter estimation) studies. Usually, only the errors associated with the measurements are properly accounted for. More specifically, it is discussed in Section 1.4 that modeling errors are, if not neglected, only empirically

accounted for (see Eq. (1.18) in that section). A methodology to account for modelling errors associated with upscaling of subsurface models could be devised based on multiscale simulation principles.

Multiscale as efficient simulation strategy for uncertainty quantification Randomized maximum likelihood (RML), as employed in Chapter 7, relies on gaussianity assumptions, whereas Monte Carlo techniques aim to accurately sample the *posterior* probability distribution. Even though the former is efficient when combined to efficient derivative computation streategies, the latter is computationally very expensive. Two-state Markov-chain Monte Carlo (McMC) [7, 8] and Multi-level Monte Carlo (MLMC) [9] have been proposed to reduce the computational cost of the sampling by evaluating less computationally demanding forward models when selecting/discarding models. Multilevel multiscale methods could be an alternative to efficiently and accurately decide on the trial models. Furthermore, the derivative calculation here proposed can provide an efficient way to improve trial models. With the evolution of computational power, Monte Carlo methods are becoming more feasible, while techniques to reduce the model evaluation cost is also of interest.

Adjoint-aided Improved Multiscale Forward Simulation The Lagrange multipliers, obtained directly from either the Lagrangian or indirectly as shown in 2 can be interpreted as sensitivities of the objective function value with respect to deviations from the constraints. In case of adjoining the (nonlinear) porous media flow equations with Lagrange multipliers, this implies that the multipliers are the sensitivities of the objective function with respect to the residuals of the flow equations, i.e., to the residual error that remains after approximately solving the nonlinear equations with the aid of Newton-Rapson iteration. This fact could be investigated as a mean for improved multiscale-simulation-based optimization, e.g. for adaptive convergence control of the nonlinear solver, adaptive convergence control of the iterative fine-scale smoothing step, or for adaptive coarsening criteria in the multiscale partitioning of the computational domain.

REFERENCES

[1] H. Hajibeygi, G. Bonfigli, M. A. Hesse, and P. Jenny, *Iterative multiscale finite-volume method,* J. Comput. Phys. **227**, 8604 (2008).

[2] I. Lunati and P. Jenny, *Multiscale finite-volume method for compressible multiphase flow in porous media,* J. Comput. Phys. **216**, 616 (2006).

[3] D. V. Voskov, *Operator-based linearization approach for modeling of multiphase multi-component flow in porous media,* Journal of Computational Physics **337**, 275 (2017).

[4] M. Cusini, C. van Kruijsdijk, and H. Hajibeygi, *Algebraic dynamic multilevel (adm) method for fully implicit simulations of multiphase flow in porous media,* Journal of Computational Physics **314**, 60 (2016).

[5] M. Cusini, B. Fryer, C. van Kruijsdijk, and H. Hajibeygi, *Algebraic dynamic multilevel method for compositional flow in heterogeneous porous media,* Journal of Computational Physics **354**, 593 (2018).

[6] M. Ţene, Y. Wang, and H. Hajibeygi, *Adaptive algebraic multiscale solver for compressible flow in heterogeneous porous media,* J. Comput. Phys. **300**, 679 (2015).

[7] Y. Efendiev, T. Hou, and W. Luo, *Preconditioning markov chain monte carlo simulations using coarse-scale models,* SIAM Journal on Scientific Computing **28**, 776 (2006).

[8] Y. Efendiev, A. Datta-Gupta, V. Ginting, X. Ma, and B. Mallick, *An efficient two-stage markov chain monte carlo method for dynamic data integration,* Water Resources Research **41** (2005).

[9] P. Kumar, C. Oosterlee, and R. Dwight, *A multigrid multilevel monte carlo method using high-order finite-volume scheme for lognormal diffusion problems,* International Journal for Uncertainty Quantification **7**, 57 (2017).

A

An Efficient Robust Optimization Workflow using Multiscale Simulation and Stochastic Gradient

We present an efficient workflow that combines multiscale (MS) forward simulation and stochastic gradient computation (MS-StoSAG) for the optimization of well controls applied to waterflooding under geological uncertainty. A two-stage iterative Multiscale Finite Volume (i-MSFV), a mass conservative reservoir simulation strategy, is employed as the forward simulation strategy. MS methods provide the ability to accurately capture fine scale heterogeneities, and thus the fine-scale physics of the problem, while solving for the primary variables in a more computationally efficient coarse-scale simulation grid. In the workflow, the construction of the basis fuctions is performed at an offline stage and they are not reconstructed/update throughout the optimization process. Instead, inaccuracies due to outdated basis functions are addressed by the i-MSFV smoothing stage. The Stochastic Simplex Approximate Gradient (StoSAG) method, a stochastic gradient technique, is employed to compute the gradient of the objective function using forward simulation responses. Our experiments illustrate that i-MSFV simulations provide accurate forward simulation responses for the gradient computation, with the advantage of speeding up the workflow due to faster simulations. Speed-ups up to a factor of five on the forward simulation, the most computationally expensive step of the optimization workflow, were achieved for the examples considered in the paper. Additionally, we investigate the impact of MS parameters such as coarsening ratio and heterogeneity contrast on the optimization. The combination of speed and accuracy of MS forward simulation with the flexibility of the StoSAG technique allows for a flexible and efficient optimization workflow suitable for large-scale problems.

The material presented in this appendix has been published in the Journal of Petroleum Science and Engineering **172**, (2019) [1] and in the proceedings of the SPE Reservoir Simulation Conference (RSC) 2017 [2].

A

We consider the life-cycle optimization of hydrocarbon production by manipulating well controls (pressure, rates or valve settings) for a given configuration of wells, a process also known as long-term production optimization or recovery optimization. In particular we consider robust optimization of strategies which maximize an objective function over an ensemble of reservoir model realizations to capture the effect of geologic uncertainty [3]. The most efficient optimization techniques for this purpose are gradient-based with gradients computed using the adjoint method [4]. However, the implementation of an adjoint is time-consuming and requires access to the simulator source code. This has led to a growing popularity of stochastic gradient-based optimization techniques which are easy to implement and flexible to use with different reservoir simulators as well as a different control types. One of the most used stochastic gradient-based techniques is Ensemble Optimization (EnOpt) which was first introduced by [5]. EnOpt uses a number of perturbed control vectors around a known control strategy where the number of perturbations is much lower than the total number of elements in the control vector (i.e. the number of well controls times the number of control time periods over the life time of the reservoir). The associated objective function values of these perturbed controls are used to approximate the gradient using linear regression.

For optimization of a single reservoir model, i.e. deterministic optimization, the computational effort to estimate the gradient increases with the number of perturbed controls to be evaluated. However, it is shown in [5] that for optimization over multiple reservoir models, i.e. for robust optimization, pairing one reservoir model to one perturbed control strategy leads to a similar computational efficiency as gradient-based techniques. An improved version of EnOpt for robust optimization was introduced in [6], called Stochastic Simplex Approximate Gradient (StoSAG), which is theoretically more sound and produces higher-quality gradients compared to EnOpt. Inspite of the attractive computational efficiency many high fidelity simulations need to be run within the optimization workflow. For real field cases this can be computationally demanding. The forward simulation is the most time consuming aspect of any robust optimization workflow. Thus it is essential to improve the forward simulation performance. Additionally, it was shown in [7] that, although the one-to-one pairing is computationally efficient, higher quality StoSAG gradients can be achieved by using a larger number of perturbations. Thus improving the computational efficiency of individual forward simulations could also be used to achieve higher quality StoSAG gradients by simulating a larger number of perturbed control strategies which will consequently improve the overall optimization process.

An increase in computational efficiency for robust optimization workflows is usually achieved by two general approaches. The first approach is to use a subset of model realizations for the optimization which reduces the number of simulations needed, while the second approach is by using faster simulation models. An overview of different approaches and workflows for using a subset of model realizations along with the advantages and disadvantages of these methods can be found in [8]. One of the ways to improve the computational efficiency of forward simulation models in optimization is through the use of Reduced Order Models (ROM) [9–11]. Alternatively, there, has been an increase in the applicability of different simulation strategies to speed up the computational process. One such technique is the Multiscale (MS) method [12, 13]. MS methods aim

to solve the equations at a coarser scale, yet preserving the fine scale description. MS methods have increasingly been demonstrated as an efficient and accurate technique for reservoir simulation [14–17]. For an overview of important developments in MS methods we refer to [18]. Among these developments, an important one in the scope of the present work is the development of iterative MS methods, more specifically the iterative Multiscale Finite Volume method (i-MSFV) [19, 20]. Due to the localization assumptions utilized in the construction of the MS basis functions, the solution obtained via an MS scheme is not as accurate as the fine-scale solution. However, these discrepancies can be resolved if an iterative scheme is employed. Moreover, because i-MSFV provides a fine-scale error estimate and an approximate, however fully conservative velocity field, it allows for both accurate and efficient simulation with control of the error estimate [21].

Upscaling/homogenization techniques [22], aim to compute effective model parameters to represent fine-scale properties at a coarser, computationally affordable, scale. Even though upscaling methods can provide accurate production/injection rates [23], which is essential for stochastic gradient-based techniques, it is often important to enable the accurate modeling of physical phenomena that requires a fine-scale heterogeneity representation, e.g. front displacements, capillarity effects, and mixing. On the other hand, MS simulation strategies aim to compute coarse scale primary variables, but are still able to represent an approximate solution at the fine scale, which is an advantage over upscaling techniques. However, this leads to, consequently, approximate reservoir responses, which will be utilized by stochastic gradient computation methods. On that note, it is shown in [24, 25] that analytical gradients (e.g. adjoint-based ones) computed via MS strategies provide an accurate enough approximation of the true gradient, capable of providing optimization results comparable to fine-scale optimizations. Also, the quality of stochastic gradients compared to analytically computed ones is discussed in [7].

The remainder of the paper is organized as follows. First, we discuss the StoSAG optimization methodology followed by the multiscale reservoir simulation framework which has been implemented in-house and used in this study. We then provide the workflow used for the optimization experiments performed in this paper. The computational gain of the workflow is illustrated via an analysis of the computation complexity of the algorithms involved. Following the theoretical descriptions of these two building blocks we illustrate the advantages and computational gains achieved on two different reservoir models for optimization cases with and without geological uncertainties.

A.1. THEORETICAL FRAMEWORK

A.1.1. STOCHASTIC GRADIENT COMPUTATION

In model-based optimization problems, traditionally, controllable variables, in individual wells or at field scale, such as injection or production rates, bottom-hole pressures (BHP) or inflow control valve (ICV) settings etc., are manipulated to maximize the value of an objective function such as Net Present Value (NPV) or Ultimate Recovery (UR). The controllable variables commonly referred to as controls are denoted by the vector $\mathbf{u}$ which consists of all the variables to be optimized at the different control time steps. In this section we briefly outline the StoSAG method first introduced in [26]. A detailed

A

description of the method can be found in [6].

For a control vector $\mathbf{u}^\ell \in \mathbb{R}^{N_u}$, where N_u is the total number of controls and ℓ is the optimization iteration index, we generate an ensemble of multi-variate Gaussian distributed perturbed control vectors $\{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_M\}$, M being total number of ensemble members, around the control vector $\mathbf{u}^\ell$ with a user defined covariance matrix $\mathbf{C} \in \mathbb{R}^{N_u \times N_u}$. At the first iteration the choice of the control vector is user-defined and at subsequent iterations determined by the optimization algorithm. The covariance matrix $\mathbf{C}$ is usually kept constant throughout the optimization although methods exists to adaptively update the covariance matrix see e.g. [27, 28]. The objective function J chosen to be optimized is then evaluated for each of the perturbed control vectors $\{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_M\}$ which leads to a corresponding set of objective function values $\{J_1, J_2, \ldots, J_M\}$.

To estimate the stochastic gradient we assemble a mean-shifted matrix of the control vectors $\mathbf{U}$ defined as

$$\mathbf{U} = [\mathbf{u}_1 - \mathbf{u}^\ell \quad \mathbf{u}_2 - \mathbf{u}^\ell \cdots \mathbf{u}_M - \mathbf{u}^\ell], \tag{A.1}$$

and a mean-shifted vector of the corresponding objective function values $\mathbf{j}$ defined as

$$\mathbf{j} = \left[J(\mathbf{u}_1) - J(\mathbf{u}^\ell) \quad J(\mathbf{u}_2) - J(\mathbf{u}^\ell) \cdots J(\mathbf{u}_M) - J(\mathbf{u}^\ell)\right]^T. \tag{A.2}$$

The equations described above can be used for deterministic (single geological model) optimization. Recently many papers have investigated the theoretical and practical applications of stochastic gradients for robust optimization; see e.g. [26, 29] and references therein. In this paper we use a 1:1 ratio, i.e. one reservoir model to one perturbed control strategy, to robustly estimate the gradient. This leads to a slightly different notation for the vector of objective function anomalies given in Eq. (A.2) which for robust optimization is defined as

$$\mathbf{j} = \left[J(\mathbf{u}_1, \mathbf{m}_1) - J(\mathbf{u}^\ell, \mathbf{m}_1) \quad J(\mathbf{u}_2, \mathbf{m}_2) - J(\mathbf{u}^\ell, \mathbf{m}_2) \cdots J(\mathbf{u}_M, \mathbf{m}_M) - J(\mathbf{u}^\ell, \mathbf{m}_M)\right]^T. \tag{A.3}$$

where m are the geological model realizations which are equal in number to the number of perturbed control vectors. The StoSAG gradient is then calculated via linear regression and is given by [6]

$$\mathbf{g} = (\mathbf{U}(\mathbf{U}^T))^\dagger \mathbf{U}\mathbf{j}, \tag{A.4}$$

where the superscript † indicates the Moore-Penrose pseudo inverse and $\mathbf{g} \in \mathbb{R}^{N_u}$. The gradient calculated above is then utilized in a simple steepest ascent algorithm [30] to calculate an updated control vector until convergence is achieved. [31] have shown that the formulation provided above has many commonalities with other stochastic gradient methods such as Simultaneous Perturbation Stochastic Approximation (SPSA) [32].

A.1.2. MULTISCALE RESERVOIR SIMULATION

Multiscale (MS) methods [12, 13], in summary, use locally constructed MS basis functions to enable the solution of the original fine-scale problem at a coarse scale and allow the solution to be accurately represented back to the fine-scale representation of the reservoir model.

In the context of this work we consider an immiscible, incompressible two-phase flow regime. Because MS methods are well-suited to solve elliptic problems, a sequential strategy to solve the governing equations is traditionally considered. Here, an Implicit in Pressure, Explicit in Saturation (IMPES) coupling is employed. For further explanations about the governing equations and solution strategies see, e.g., [33].

The two-level [34] MS system can be algebraically expressed as [35, 36]

$$(\mathbf{RAP})\,\check{\mathbf{p}} = (\mathbf{Rq}) \tag{A.5}$$

where $\mathbf{R}$ is an $N_C \times N_F$ restriction operator, $\mathbf{P}$ is an $N_F \times N_C$ prolongation operator, $\check{\mathbf{p}} \in \mathbb{R}^{N_C}$ is the coarse-scale pressure solution, $\mathbf{q} \in \mathbb{R}^{N_F}$ the source terms vector, and $\mathbf{A}$ is the $N_F \times N_F$ system matrix resulting from the flow equation discretization [33]. Also, N_F and N_C are, respectively, the fine grid and coarse grid number of gridblocks. The interpolated fine-scale pressure is obtained by

$$\mathbf{p}' = \mathbf{P}\check{\mathbf{p}}, \tag{A.6}$$

where $\mathbf{p}' \in \mathbf{R}^{N_F}$ is the approximate fine scale solution. The prolongation operator $\mathbf{P}$ is constructed based on local basis functions. Different strategies to build the basis functions are available in the literature [12, 13, 37, 38]. The restriction operator $\mathbf{R}$ can be either constructed as the transpose of the prolongation operator in finite-element-based multiscale methods, or simply be based on the grid geometry in finite-volume-based ones [13].

Following the MS pressure equation solution, we solve for the saturations at the fine scale. Because of the hyperbolic nature of the transport equations, a fine grid is necessary in order to capture the local, sharp saturation fronts. Hyperbolic conservative laws require locally conservative velocity fields. However, MS approximate solutions are not conservative at the fine scale. There are different alternatives to build a fine-scale conservative velocity field to be used in the solution of the transport equation at the fine scale [20, 21, 39].

Finally, because the Courant–Friedrichs–Lewy (CFL) condition for numerical stability [40] can be very restrictive in terms of the time-step size selection, we employ an asynchronous time-stepping strategy [41]. One should note that sequential implicit [42] and adaptive implicit [33] simulation strategies could be considered to address the CFL time-step restrictions. However, we emphasize that IMPES strategies can deliver efficient solution strategies due to the straight forward solution of saturation – it can be seen as as simple matrix-vector multiplication. Some simulators rely on IMPES and deliver very good performances [43]. In the context of optimization, which is the focus of this work, any efficient simulation coupling strategy could be considered. Also, MS strategies introduced by [44, 45], which solve the transport equation at the coarse scale, could be used in our proposed MS-StoSAG workflow.

A.2. MS-StoSAG workflow

The main idea of the workflow we present next is to speed up the forward simulations required for the evaluation of the objective function and for the StoSAG gradient computation used at every iteration during optimization. In this direction, regarding MS

A

simulations, we want to avoid the overhead to the total simulation time due to the construction of basis function [14]. This can be achieved by employing the offline/online basis function construction strategy introduced by [46]. The construction of the basis function, for all ensemble members, is performed at an offline stage, outside the main optimization loop. Online updates to the MS system are then performed in an online stage so that any required improvement due to MS solution approximations are compensated. The ensemble optimization workflow combined with the online/offline MS simulation strategy is shown in Fig. A.1.

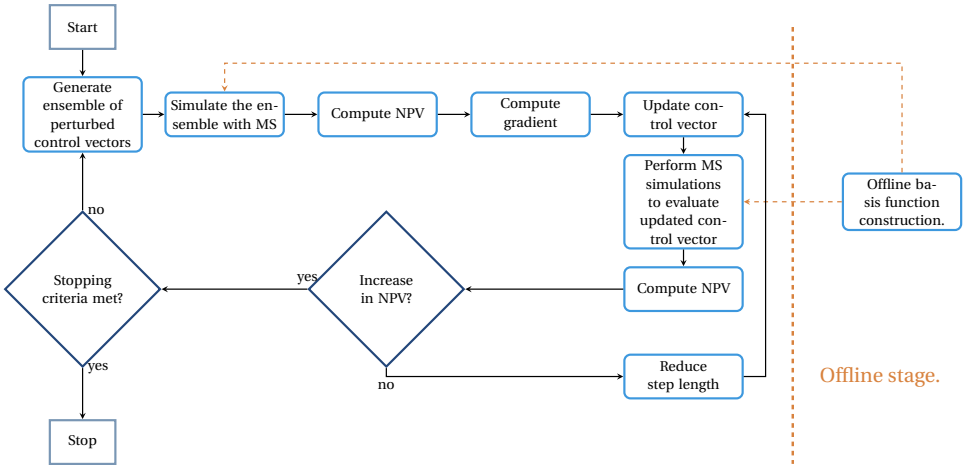


Figure A.1: The MS-StoSAG optimization workflow. MS reservoir simulation is used in the evaluation of the model responses.

Note that both the simulation of the ensemble members for NPV calculation and for the StoSAG gradient computation benefit from faster MS simulations. Basis functions are only built once, at the beginning of the optimization process, at the offline stage.

In our implementation, although a simple steepest-ascent with a variable step size optimization algorithm has been used, the framework is flexible to allow for the use of more sophisticated algorithms. Furthermore, for the robust optimization considered in this work, the uncertainty associated with the fine-scale parameters is accounted for through an ensemble of equiprobable geological realizations. [47, 48] have addressed the application of MS methods for stochastic subsurface fluid flow modeling. Even though these MS methods have not been considered in this work, it can be observed that the framework here presented is independent of a particular MS method, and hence any formulation, including the stochastic MS version, could be considered in our proposed MS-StoSAG workflow.

A key aspect of our MS-StoSAG workflow is the online stage. This online stage happens during each forward MS simulation. In our simulator the flow (pressure) equation, is solved via the Multiscale Finite Volume (MSFV) method [13]. The MSFV method, by construction, is mass conservative [13]. The restriction operator in Eq. (A.5) is based on a finite volume integration operator at coarse scale resulting on a matrix of 0's and 1's (see e.g. [36]) , whilst the the prolongation operator is based on basis functions con-

structed via the local solutions of the governing flow equation, subject to assumed local boundary conditions, without right-hand-side (RHS) terms

$$-\nabla\cdot\left(\lambda\cdot\nabla\varphi\right)=0, \tag{A.7}$$

where λ is the mobility and φ is the basis function value. This involves the definition of a primal coarse grid (on which the conservative coarse-scale system is constructed) and a dual coarse grid, which is obtained by connecting coarse nodes. A coarse node is a fine cell inside (typically at the center of) each coarse cell. Basis functions are solved locally on these dual coarse cells. Such overlapping coarse and dual coarse grids are crucial for conservative solutions in MSFV. An illustration of the MSFV grids is provided in Fig. A.2.

As basis functions do not account for RHS terms, well terms are considered as supplementary functions, called "well functions" [49]. Well functions are local solutions of the flow problem considering unity solutions at the well (i.e., $p_w = 1$). An illustration of basis and well functions is provided in Fig. A.3. The prolongation operator can be expressed in terms of the basis functions corresponding to each coarse cell $j = 1,\ldots,N_C$, and each well function ψ_w corresponding to all $w = 1,\ldots,N_w$ wells adds a column vector to the porous rock prolongation operator

$$\mathbf{P}=\left[\begin{array}{cccc|ccc}\varphi_1 & \cdots & \varphi_{N_C} & | & \psi_1 & \cdots & \psi_{N_W}\end{array}\right]. \tag{A.8}$$

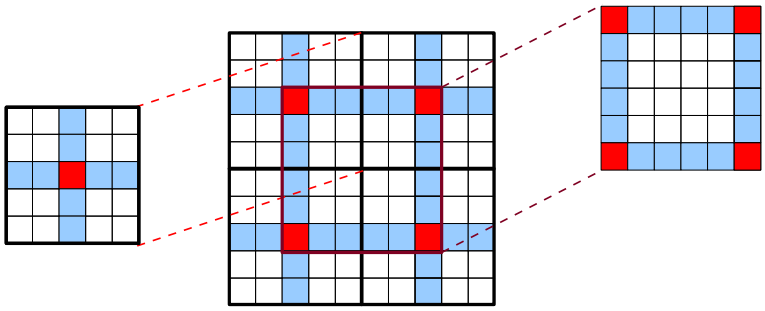


Figure A.2: Illustration of MSFV coarse grids for a 2D domain. Given a fine-scale grid (shown in light solid black lines), the coarse grid (shown in solid bold black) is imposed as a non-overlapping partition of the computational domain. The coarse nodes (vertices) are then selected (filled in red cells). Connecting coarse nodes constructs the dual-coarse grid (highlighted in blue) where basis functions are solved.

We highlight that, in our implementation, the computation of well/basis functions is only performed at the offline stage, as a pre-processing step. The basis functions are not updated because, firstly, the smoothing step of the iterative Multiscale Finite Volume (i-MSFV) method (discussed below) addresses the discrepancies left over from the MS solution stage and, secondly, it is more computationally efficient in the optimization context since basis functions can be computed a priori, i.e. at an offline stage. Building basis functions is a key element in the multiscale simulation framework. An optimum balance between efficiency and accuracy is directly associated with the coarsening ratio employed in the construction of the coarse grid. The ability of capturing the fine-scale heterogeneities and the position of the coarse nodes [50] are examples of factors that

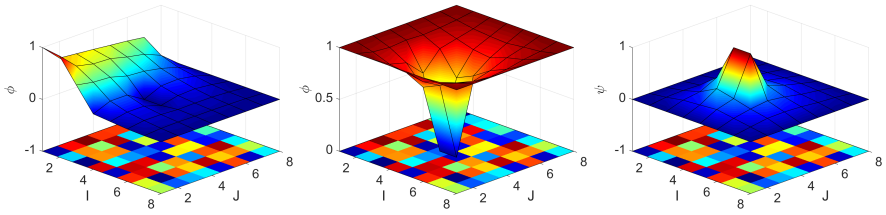


Figure A.3: (a) Basis function, (b) sum of all basis functions and (c) well function for a given heterogeneous porous media (represented in the bottom of the plots) in a given dual-grid cell containing a well (source term). The well perforates two fine grid blocks in the center of the dual grid block.

influence the choice of the coarsening ratio. In [51] it is indicated that, generally, a coarsening factor approximately equal to the square root of the number of grid-blocks in each direction leads to a good performance/efficiency trade-off. For the robust optimization experiments considered in this paper the basis functions for every single geological model realization have been built. However, because we do so at the offline stage, the computational cost of building all the basis function is relatively small when compared to the overall optimization cost.

The approximate multiscale pressure solution provided by Eq. (A.6) is inaccurate when compared to the fine-scale solution of the pressure equation. Firstly, by design, the MSFV solution reflects the localization assumptions utilized to compute the basis functions in Eq. (A.8). Secondly, non-monotone solutions may occur for geological settings with high permeability contrasts, specially in low permeability regions [52] (as demonstrated, e.g., for the SPE10 benchmark case [53]). Again, the non-monotone solutions are associated with the construction of the MSFV basis functions. Different strategies have been proposed in the literature to address this issue, e.g. the monotone MSFV (m-MSFV) [50] and the Multiscale Restriction Smoothed Basis (MsRSB) [38] methods. However, in this work, all these discrepancies can be resolved if an iterative scheme is employed [20].

The iterative Multiscale Finite Volume method (i-MSFV) is capable of resolving these differences by resolving the high frequency errors via some iterative (smoothing) solutions at the fine scale and resolving the low frequency errors via the MSFV coarse-scale solution. In brief, the method consists of re-writing Eqs. (A.5) and Eq. (A.6) in residual form and iteratively solving the resulting system of equations until a convergence criterion is met. Note that i-MSFV delivers conservative coarse-scale solutions after any iteration stage. As such, it is not used as a linear solver, but to maintain the desired user-defined accuracy. This is, in our implementation, the online stage that addresses the MS solution inaccuracies. This is illustrated in Fig. A.4.

Regarding the conservative velocity field reconstruction, we also rely on the i-MSFV method, by smoothing the fine-scale i-MSFV solution until a sufficiently small fine-scale residual is achieved.

We address the CFL time-stepping restrictions by solving the transport equation using small time-steps, hence honoring the CFL condition, but keeping pressure and velocity fields unchanged. The transport solver time-step Δt_s is limited by CFL conditions, but

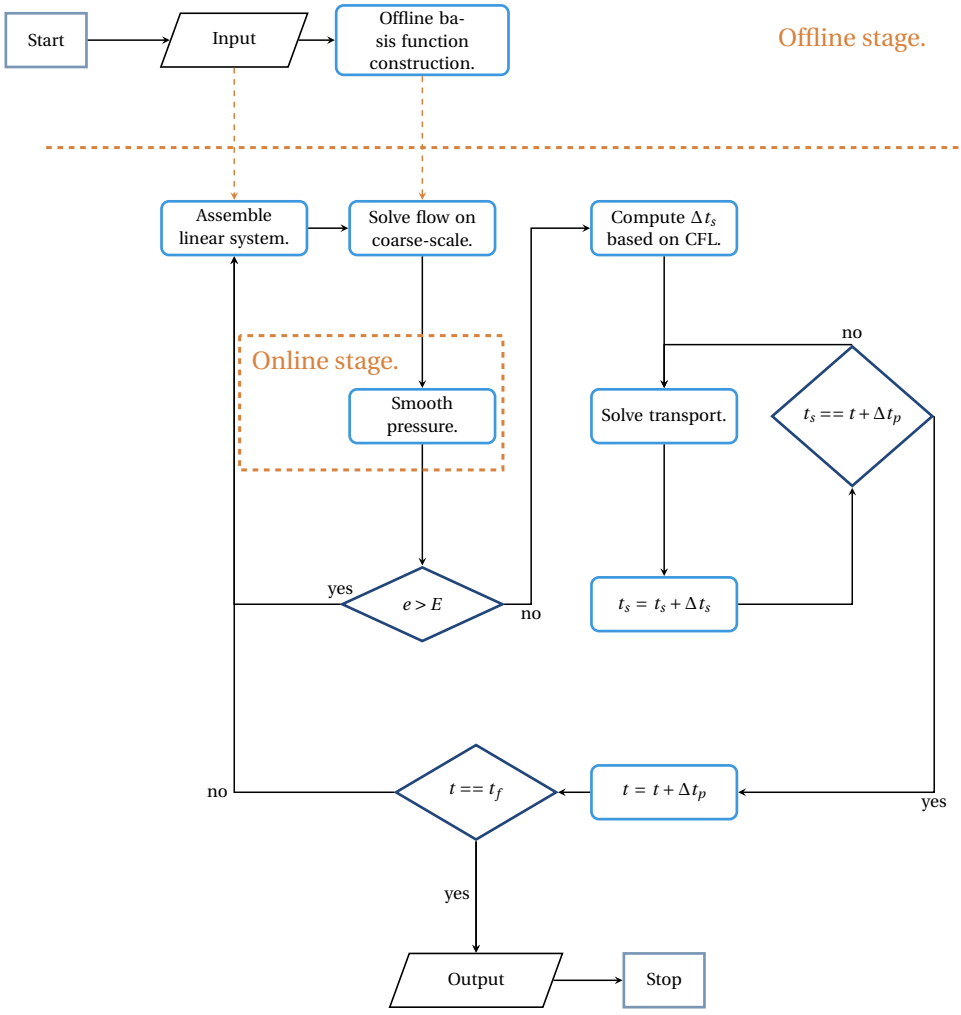


Figure A.4: Schematic description of the MS reservoir simulation.

the flow solver time-step Δt_p is not. The velocity field is kept unchanged until the transport solver time t_s reaches Δt_p. The simulator outputs the information required to compute the gradient and objective function when the final time t_f is reached.

A.2.1. A NOTE ABOUT COMPUTATIONAL COMPLEXITY

We measure the computational efficiency of our workflow by comparing the relative computational complexity of a steepest ascent iteration using our MS-StoSAG strategy to the alternative of employing a fine-scale simulation in the forward-model evaluations.

In the scope of our applications, because of the underlying physics and solution strategies employed, the computational cost of the forward simulation can be assumed to be the cost of the pressure equation linear system solution. This is because (1) for incompressible flow there is no partial derivative computation involved, (2) in the IMPES strategy the solution of the transport equation is proportional to a matrix-vector multiplication. Furthermore, the computational complexity of all other steps of the workflow can also be considered negligible when compared to the cost of solving a linear system.

The complexity of solving a linear system of size N is assumed to be $\mathcal{O}(aN^b)$, where a and b are constants associated with the specific linear solver employed. One steepest ascent iteration, disregarding any backtracking for the sake of simplicity, requires the evaluation of M forward simulations to evaluate all realizations' objective function values. Additionally, in order to estimate the StoSAG gradient in a 1:1 approach, we need to evaluate another M forward simulations. Therefore, $\mathcal{O}_{FS}(2 \times M \times aN_F^b)$ is the cost of one steepest ascent iteration if fine-scale simulation is employed and $\mathcal{O}_{MS}(2 \times M \times N_{iMS} \times (aN_C^b + N_S \times aN_F^b))$ is the cost for the MS simulation when empoying a two-stage i-MSFV strategy, where N_{iMS} is the total number of i-MSFV iterations and N_S is the total number of smoothing steps per i-MSFV iteration.

Given that we are interested in evaluating the relative gain of the workflow compared to ensemble optimization when fine-scale forward simulations are employed, we can say that, given our setting, this can be assessed by simply computing the ratio

$$\begin{aligned}\frac{\mathcal{O}_{FS}}{\mathcal{O}_{MS}} &= \frac{\mathcal{O}_{FS}(2 \times M \times aN_F^b)}{\mathcal{O}_{MS}(2 \times M \times N_{iMS} \times (aN_C^b + N_S \times aN_F^b))} \\ &= \frac{\mathcal{O}_{FS}(aN_F^b)}{\mathcal{O}_{MS}(N_{iMS} \times (aN_C^b + N_S \times aN_F^b))}.\end{aligned}$$

Thus, because of the discussion above, and because our steepest ascent stopping criteria is the maximum number of iterations, the computational gain of the MS-StoSAG workflow for the MS setting we use can be directly estimated from the computational cost ratio between fine-scale and MS cost to solve the flow equation. This will be the metric used when evaluating the computational cost in the numerical experiments.

A.3. NUMERICAL EXPERIMENTS

We illustrate the application of our proposed framework for life-cycle waterflooding optimization of two different models. For both models, the controls are bottom-hole pres-

sures in the injection and production wells. The objective function used for optimization is a standard economic objective, Net Present Value (NPV), as defined in Eq. (A.9), for which we use the prices provided in **Table A.1**.

$$J = \sum_{k=1}^{K} \frac{[(q_{o,k}) \cdot r_o - (q_{wp,k}) \cdot r_{wp} - (q_{wi,k}) \cdot r_{wi}] \cdot \Delta t_k}{(1+b)^{\frac{t_k}{\tau_t}}} \tag{A.9}$$

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 $\$/m^3$, r_{wp} is the cost of produced water in $\$/m^3$, r_{wi} is the cost of injected water in $\$/m^3$, 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.

Table A.1: Costs associated with oil production.

	Value	Unit
Oil price	252	$\$/m^3$
Cost of injected water	60	$\$/m^3$
Cost of produced water	30	$\$/m^3$

In the MS simulation of all numerical experiments there is no basis function reconstruction. Instead, we delegate the resolution of the discrepancies left by the localization assumptions and inaccuracies due to outdated basis functions to the i-MSFV smoothing step. Also, the conservative velocity field is reconstructed directly from the smoothed pressure field. Therefore, we require the i-MSFV loop to reduce the fine scale residual with one order of magnitude, while the fine scale residual tolerance after smoothing is set to 10^{-6}. A l^2-norm is employed to compute the residual norm. The stabilized bi-conjugate gradient iterative solver with ILUT preconditioner [54] is used as the fine-scale smoother.

A.3.1. TOY MODEL - FIVE-SPOT MODEL

In the first numerical experiment, in order to evaluate the proposed workflow in a relatively controlled environment, a toy-model, consisting of a simple synthetic 2D inverted five-spot model is considered. It consists of a 21×21 regular mesh with grid block size dimensions of $33.3 \times 33.3 \times 2$ m. The reservoir porosity is constant and equal to 0.3. The permeability distribution is depicted in Fig. A.5 and the fluid properties are described in **Table A.2**. For the optimization, we have 5 control variables per control time step. In this exercise we use 6 control time steps of 720 days each, thus we optimize 30 variables. The values of the bottom-hole pressures are bounded for the production wells between a minimum value of 28 MPa and a maximum value of 30 MPa. The injection well pressures are bounded between a minimum value of 30 MPa and maximum value of 32 MPa.

DETERMINISTIC LIFE-CYCLE OPTIMIZATION

We first consider a single model realization. For this purpose we use an ensemble of 50 perturbed control vectors with a perturbation size defined as 10% of the difference bet-

A

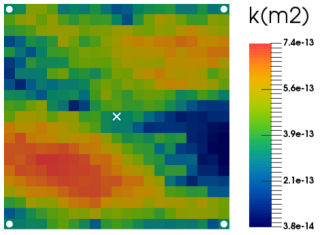


Figure A.5: The permeability realization used for deterministic optimization and well placement

Table A.2: Fluid properties for 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 ($S_w c$)	0.0	–

ween the min and max bounds of the controls. The stopping criteria used is a maximum number of 25 optimization iterations. The initial starting strategy is one wherein the injector well operates at a constant BHP of 31 MPa and the production wells at a constant BHP of 29 MPa. When working with reduced order or upscaled models it is imperative to compare and validate the results with respect to the fine scale model realization.

Multiscale Optimization We consider two different coarsening ratios to test the efficiency of our proposed MS-StoSAG workflow. The two coarsening ratios are 7x7 and 3x3. Thus a deterministic optimization with the optimization parameters discussed above is performed for 3 different models, fine scale, i-MSFV 7x7 and i-MSFV 3x3. The optimization process is illustrated in Fig. A.6. We observe that all three models find optimal strategies which produce approximately similar NPV values. We also observe that while the i-MSFV 7x7 model achieves a slightly lower NPV compared to the fine scale model, the i-MSFV 3x3 model achieves a higher NPV than the other two models. Thus the coarsest model used produced the highest NPV which may be counter-intuitive. Cross validation of the i-MSFV strategies on the 21x21 fine scale model produces NPV values which are indistinguishable from each other which could be the result of a relatively simple small model. Fig. A.7 is a comparison of the optimal control strategies obtained from the different models. We observe that all the strategies are very similar to each other thus the objective function values are also similar.

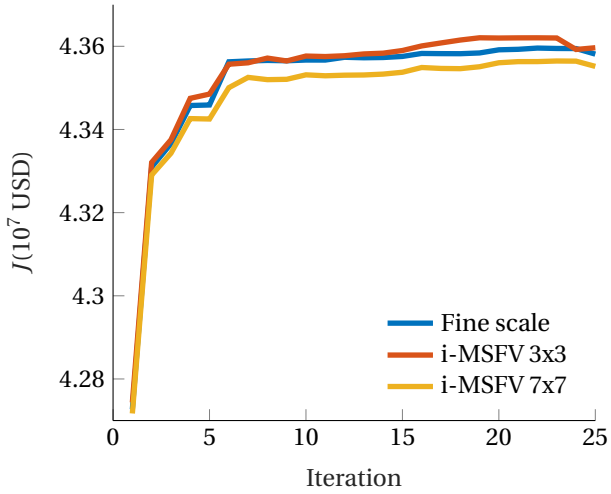


Figure A.6: Comparison of the objective function through the iteration process for fine scale and the two i-MSFV strategies, toy model.

Sensitivity to Optimization Parameters The parameters which influence the computation of a stochastic gradient such as the ensemble size of the perturbed controls, perturbation sizes to generate the ensemble of controls, random seeds used to generate the perturbed controls were varied and tested with the different i-MSFV models. The results from this exercise followed the exact same trends as reported in earlier publications see e.g. [7] who investigated the impact of gradient quality based on the various parameter choices.

Robust Life-cycle Optimization Inclusion of uncertainty within the optimization framework is usually accounted for by utilizing an ensemble of equiprobable geological model realizations. In this paper a large ensemble of 1,000 realizations of the five-spot model was generated via the decomposition of a reference permeability "image" using Principal Component Analysis parameterization. See [55] for more details. Fig. A.8 illustrates 4 different permeability realizations from a reduced ensemble of 50 members used in the optimization experiments. The reduced ensemble was selected by simulating all 1000 realizations with the same control vector and uniformly sampling realizations based on the highest, lowest and intermediate objective function values. We acknowledge that there exists a range of formal clustering techniques to carefully select a representative ensemble. For the proof-of-concept optimization experiments in this paper the representativeness of the selected ensemble to a larger ensemble is deemed to be outside the scope of this paper.

The objective function used for the robust optimization experiments is the expected NPV calculated over the 50 realizations. Fig. A.9 illustrates the evolution of the mean NPV through the optimization process for the different simulation models. The same coarsening ratios and initial starting strategy has been used as the deterministic case. The fine scale strategy achieves the highest objective function values followed by the

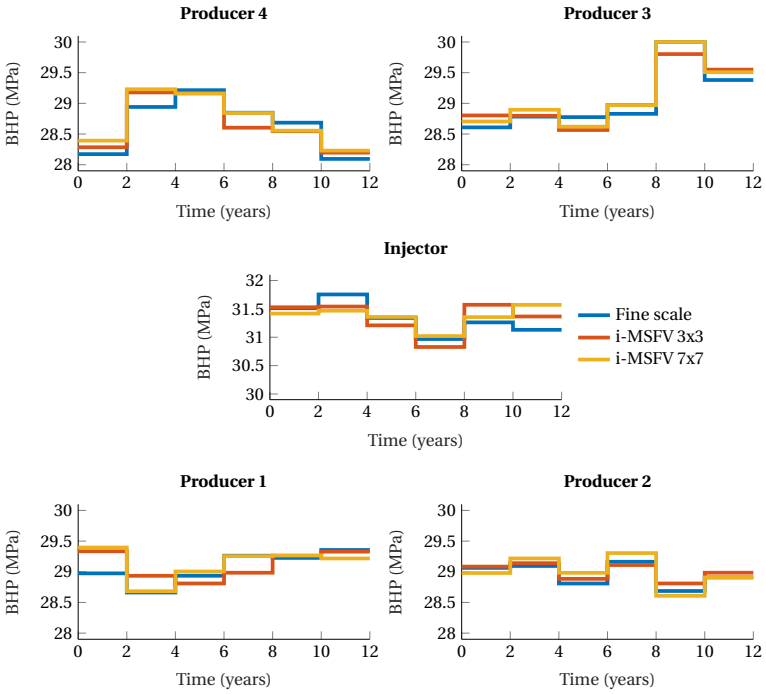


Figure A.7: Comparison of the optimal control strategies between fine scale and two i-MSFV strategies, toy model.

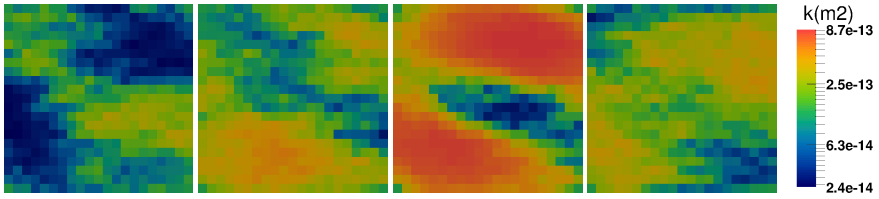


Figure A.8: Four different permeability realizations from the ensemble of 1,000 members used in the toy model numerical experiments.

i-MSFV 7x7 model and the i-MSFV 3x3 model. Though the five-spot model is relatively small it is reassuring that, irrespective of the different models (fine scale or i-MSFV) used, very similar objective function values have been achieved when geological uncertainties are accounted for.

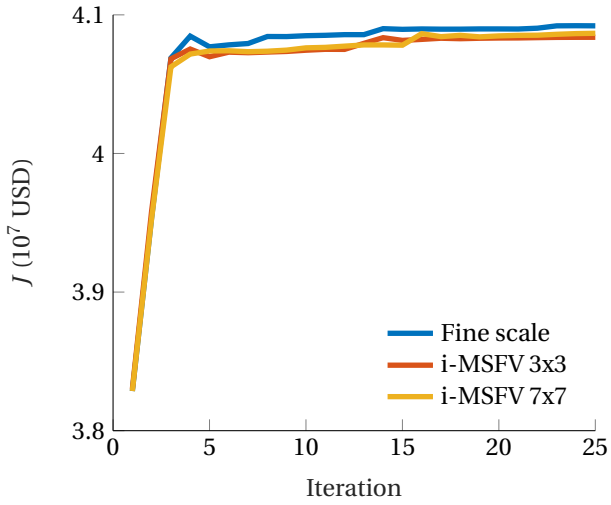


Figure A.9: Comparison of the mean objective function value for the three different simulation strategies, fine scale (blue), i-MSFV 7x7 (orange) and i-MSFV 3x3 (red), toy model.

Fig. A.10 is a comparison of the optimal control strategies for two different models, fine scale and i-MSFV 3x3. The optimal control strategies achieved after robust optimization are quite different from the strategies achieved for deterministic optimization which could be the impact of geological uncertainties, the correspondingly different basis functions and impact of coarsening ratio on the different realizations. A-priori determination of the impact of these factors on the results is non-trivial and is probably model dependent.

COMPUTATIONAL EFFICIENCY

Table A.3 provides a comparison of the computational efficiency and speedup gained for the different models used for the experiments. We use the linear solver time to measure the speed up. This accounts for all computations performed by the i-MSFV solution strategy, i.e solution of the coarse system plus fine-scale smoothing. Also, it is well-known that the linear system solution is the most time-consuming part in reservoir simulation framework [33].

An approximately factor 4 speedup can be achieved for this relatively small five-spot model. The average number of smoothing steps throughout the optimization is approximately similar, irrespective of the coarsening ratio used which is model- and problem-dependent. The tolerances of the residual for the MS solution and the smoothing steps are the same for all cases. In order to understand and validate the efficiency of the proposed framework a larger model is used below.

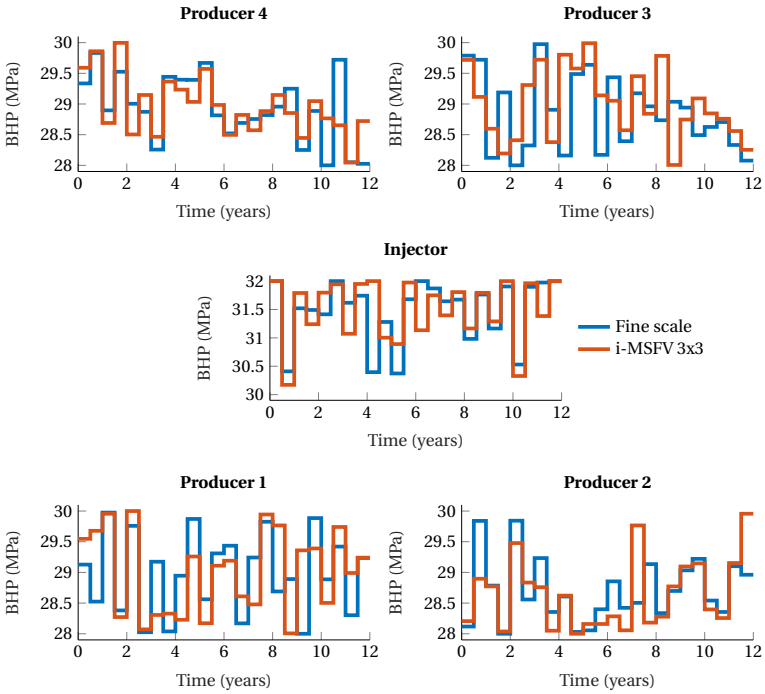


Figure A.10: Comparison of the controls that generated the highest NPV, toy model.

Table A.3: Comparison of the computational effort and average number of smoothing steps ($\bar{N}_S$) for the different i-MSFV cases, toy model.

Simulation strategy	$\frac{\mathcal{O}_{FS}}{\mathcal{O}_{MS}}$	$\bar{N}_S$
i-MSFV (7x7)	4.3	1
i-MSFV (3x3)	4.7	2

A.3.2. KANAAL RESERVOIR MODEL

A second example is considered to illustrate the effectiveness of our proposed workflow. We use a 99 x 99 2D model with grid blocks of 10 × 10 × 5 m. Thus we have approx. 10,000 grid blocks in this model compared to the 441 grid blocks in the five-spot model. The geological description used in this model has been inspired from a typical channelized North Sea reservoir. The channels (Kanaal in Dutch) are set in a sandy shale background with permeabilities ranging from 10-50 mD. The channels have permeabilities ranging from 250-700mD with constant porosities of 20%. This reservoir is developed using a line drive well configuration with 3 injection wells and 6 production wells as illustrated in Fig. A.11. The fluid properties of this reservoir are given in **Table A.4**.

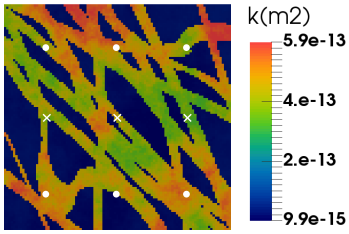


Figure A.11: Illustration of the well configuration used for the Kanaal reservoir model with injectors in the middle row denoted by X.

An ensemble of 50 geological realizations has been created using geological modeling software [56] to be used for the robust optimization experiments. Each of the realizations has been constrained to well logs from 4 exploration wells which were generated from a base-case model. This base-case model does not form part of the ensemble of model realizations. An illustration of a subset of models is illustrated in Fig. A.12. The life cycle period for this reservoir is 12 years. We allow the controls, i.e. the bottomhole pressures in all wells, to be manipulated once every 6 months, i.e. 24 we have control time steps. Thus for the 9 wells we optimize a total of 9x24 = 216 controls. The values of the bottomhole pressures are bounded for the production wells between a minimum value of 28 MPa and a maximum value of 30 MPa. The injection well pressures are bounded between a minimum value of 30 MPa and maximum value of 32 MPa.

ROBUST LIFE-CYCLE OPTIMIZATION

For this example we focus on robust optimization experiments. The initial control strategy corresponds to the bottomhole pressures on their maximum bound for the injectors, i.e. a constant pressure of 32 MPa, and on their minimum bound for the producers,

A

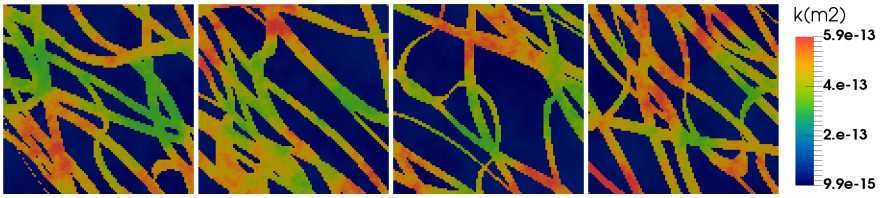


Figure A.12: Four different permeabilty realizations from an ensemble of 100 members, Kanaal model.

Table A.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 ($S_w c$)	0.0	–

i.e. 28 MPa. Different coarsening ratios and their impact on the optimization are investigated. The coarsening ratios for this example are 3x3, 11x11 and 33x33. The fine scale model is 99x99. We observe in Fig. A.13 that all the optimization experiments find solutions that achieve an objective function value higher than a reactive control strategy, where the production wells are shut-in once the water-oil ratio exceeds a preset maximum. The economic water cut for the reactive control strategy for the prices considered in this paper is 82%. The reactive control strategy has been used as a reference solution when evaluating the performance of life-cycle optimization techniques in line with the approach first introduced in [3]. We also observe from Fig. A.13 that different coarsening ratios achieve different optimal mean NPV values with similar trends as observed for the five-spot model. It is also important to note that, in this case, the optimization runs performed with the MS-StoSAG workflow provide mean NPV values higher than the one obtained by the fine-scale optimization.

COMPARISON OF OPTIMAL STRATEGIES

Fig. A.14 is an illustration of the optimal controls for 2 wells for the fine scale model and the coarser scale i-MSFV 11x11 model. The optimal controls obtained by the other models are significantly different from each other with the same trend observed for the other seven wells. This is a fact that has been consistently observed in the literature, first noted in [57]. Considerably different control settings can lead to relatively similar NPV values due to the over-parameterization of the optimization problem. This is in accordance with numerous other studies; see, e.g., [58–60]. A comparison of the optimal controls for two different i-MSFV models is illustrated in Fig. A.15. Once again, we observe that different optimal control strategies achieve very similar objective function

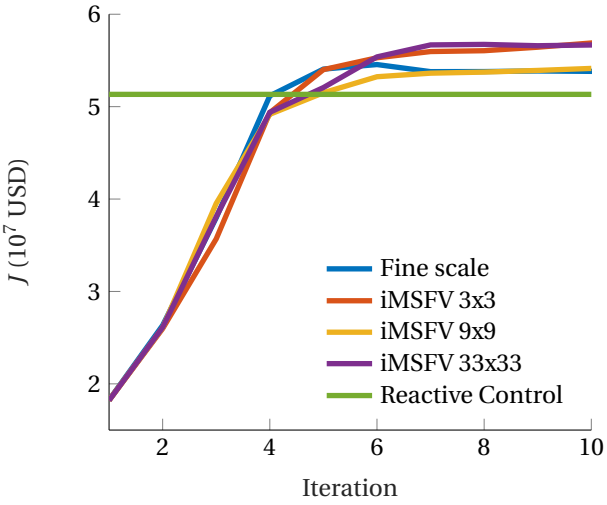


Figure A.13: Comparison of objective function value between fine scale and the different i-MSFV strategies used, Kanaal model.

values.

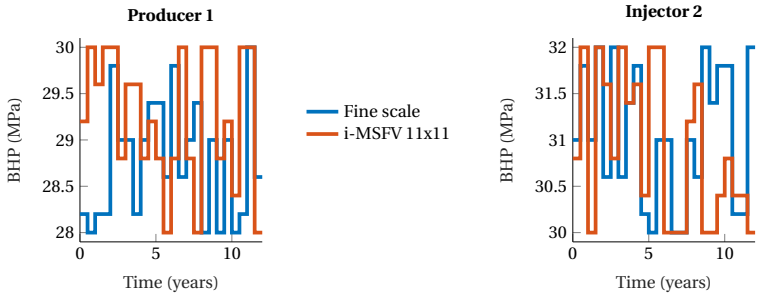


Figure A.14: Comparison of the optimal control strategies from fine scale and i-MSFV 11x11 strategy, Kanaal model.

In addition to visually recognising differences in the optimal control strategies we provide in **Table A.5** a comparison of the optimal volumes of oil and water produced and injected for the different simulation models. We observe an interesting trend in the results. Higher volumes of oil and water are produced in the i-MSFV models compared to the fine scale model. Additionally we observe that the higher the coarsening ratio, the higher are the production and injection volumes. An important aspect to be considered here is the ability of the local basis functions to represent the fine-scale heterogeneities. The coarsening ratio and the positioning of the coarse node vertices are instrumental in achieving higher quality basis functions. Different strategies exist to achieve higher quality basis functions (e.g. [50]), however, in our experiments, we rely on the ability of the i-MSFV to correct the MSFV solution approximations/errors.

A

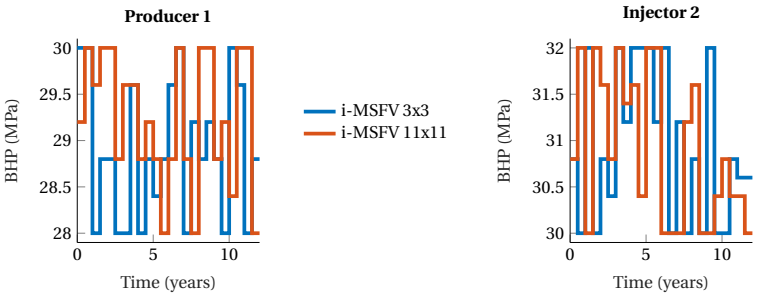


Figure A.15: Comparison of the optimal control strategies from i-MSFV 11x11 and i-MSFV 3x3 strategies, Kanaal model.

Table A.5: Comparison of optimal total produced oil (Q_o) and water ($Q_{w_{prod}}$) and total injected volume ($Q_{w_{inj}}$) for the different simulation strategies, Kanaal model.

Simulation Strategy	$Q_o(m^3)$	$Q_{w_{prod}}(m^3)$	$Q_{w_{inj}}(m^3)$
Fine scale	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

COMPUTATIONAL EFFICIENCY

In this section we compare the computational efficiency and speedup gained for the different coarsening ratios. For the larger Kanaal model an approximately 5 times speedup in computational efficiency is achieved as observed in Table A.6. The tolerances of the residual for the MS solution and the smoothing steps are the same for all the cases. We also observe a clear trend in the average number of smoothing steps required for the different coarsening ratios with a higher number of smoothing steps being required for a coarser model.

Table A.6: Comparison of the computational effort and average number of smoothing steps ($\bar{N}_S$) for the different i-MSFV cases, Kanaal model.

Simulation strategy	$\frac{\mathcal{O}_{FS}}{\mathcal{O}_{MS}}$	$\bar{N}_S$
i-MSFV 33x33	4.44	5
i-MSFV 11x11	5.69	12
i-MSFV 3x3	4.62	18

A.4. DISCUSSION

The multiscale implementation used in this work can be further improved upon to obtain an even higher computational efficiency similar to the results reported in [25]. The speedup in the computational efficiency reported in [25] was obtained with coarse-scale

representations for the both the flow and transport equations. In this paper the flow problem is solved at the coarse scale while the transport equation has been solved on the fine scale which improves the accuracy of our results.

Even though no basis function reconstructions were performed in our experiments, and the i-MSFV fine-scale smoothing performed remarkably well, that might not be the case when more complex physics are involved in the simulation model (e.g. gravity effects, capillarity, higher mobility ratios). However, even under more complex scenarios, MS strategies should still deliver considerable speedups given that basis functions need only be built adaptively and infrequently. Also, the employment of different velocity field reconstruction techniques could be necessary in the case that i-MSFV becomes expensive. But again, techniques like those presented by [20, 39] benefit from adaptivity. Furthermore, the implementation of our MS reservoir simulator does not yet capitalize on all potential advantages of MS methods. MS methods are well suited to take full advantage of modern high performance-computing architectures. For instance, the solution of basis functions are embarrassingly parallelizable [17, 61], an advantage with respect to global-based ROM approaches. Moreover, we expect that the computational advantage will increase for larger-sized reservoir models, relying on the model-size-dependent computational performance of MS methods, as discussed, e.g., in [13].

When working with coarse scale models for optimization it is imperative to validate the optimized strategy using the fine scale model to understand and quantify the impact of optimization. Reduced-order or upscaled models, though computationally very efficient, do not always produce similar production responses compared to the fine scale model. In our approach, because the transport problem is always solved on the fine scale, the optimal strategies need not be validated as is confirmed by the results in this paper. This is an additional attractive feature of the workflow proposed in this paper. When working with ROM methods such as POD [11] or TPWL [10], multiple high-fidelity simulations need to be performed a-priori (akin to an offline stage) to develop the ROM and afterwards to verify the accuracy of the results. In our workflow we alleviate the need for these expensive pre- and re-computations. In our MS-StoSAG workflow the offline stage i.e., computation of the basis functions in the MS method is extremely cheap compared to the overall cost of the optimization process.

REFERENCES

[1] R. Moraes, R.-M. Fonseca, M. Helici, A. Heemink, and J. D. Jansen, *An efficient robust optimization workflow using multiscale simulation and stochastic gradients,* (under review, (June 2018)).

[2] R. Moraes, R.-M. Fonseca, M. Helici, A. Heemink, and J. D. Jansen, *Improving the computational efficiency of approximate gradients using a multiscale reservoir simulation framework,* in *SPE Reservoir Simulation Symposium* (Society of Petroleum Engineers, 2017).

[3] G. van Essen, M. Zandvliet, P. Van den Hof, O. Bosgra, J.-D. Jansen, *et al., Robust waterflooding optimization of multiple geological scenarios,* SPE Journal **14**, 202 (2009).

A

[4] J. D. Jansen, *Adjoint-based optimization of multi-phase flow through porous media–a review,* Computers & Fluids **46**, 40 (2011).

[5] Y. Chen, D. S. Oliver, and D. Zhang, *Efficient ensemble-based closed-loop production optimization,* SPE J. **14**, 634 (2009).

[6] R. M. Fonseca, B. Chen, J. D. Jansen, and A. C. Reynolds, *A stochastic simplex approximate gradient (StoSAG) for optimization under uncertainty,* Int. J. Numer. Meth. Eng. (2016), 10.1002/nme.5342.

[7] R. M. Fonseca, S. S. Kahrobaei, L. J. T. Van Gastel, O. Leeuwenburgh, and J. D. Jansen, *Quantification of the impact of ensemble size on the quality of an ensemble gradient using principles of hypothesis testing,* in *SPE Reservoir Simulation Symposium* (Society of Petroleum Engineers, 2015).

[8] E. G. D. Barros, S. Maciel, R. J. Moraes, and R. M. Fonseca, *Automated clustering based scenario reduction to accelerate robust life-cycle optimization,* in *ECMOR XVI-16th European conference on the mathematics of oil recovery* (EAGE, 2018 - to appear).

[9] J. D. Jansen and L. J. Durlofsky, *Use of reduced-order models in well control optimization,* Optim. Eng. , 1 (2016).

[10] M. Cardoso and L. Durlofsky, *Linearized reduced-order models for subsurface flow simulation,* Journal of Computational Physics **229**, 681 (2010).

[11] J. F. van Doren, R. Markovinović, and J.-D. Jansen, *Reduced-order optimal control of water flooding using proper orthogonal decomposition,* Computational Geosciences **10**, 137 (2006), http://dx.doi.org/10.1007/s10596-005-9014-2.

[12] T. Y. Hou and X.-H. Wu, *A multiscale finite element method for elliptic problems in composite materials and porous media,* J. Comput. Phys. **134**, 169 (1997).

[13] P. Jenny, S. H. Lee, and H. A. Tchelepi, *Multi-scale finite-volume method for elliptic problems in subsurface flow simulation,* J. Comput. Phys. **187**, 47 (2003).

[14] M. Ţene, Y. Wang, and H. Hajibeygi, *Adaptive algebraic multiscale solver for compressible flow in heterogeneous porous media,* J. Comput. Phys. **300**, 679 (2015).

[15] M. Cusini, A. A. Lukyanov, J. Natvig, and H. Hajibeygi, *Constrained pressure residual multiscale (cpr-ms) method for fully implicit simulation of multiphase flow in porous media,* J. Comput. Phys. **299**, 472 (2015).

[16] A. Kozlova, Z. Li, J. R. Natvig, S. Watanabe, Y. Zhou, K. Bratvedt, and S. Lee, *A real-field multiscale black-oil reservoir simulator,* in *SPE Reservoir Simulation Symposium* (Society of Petroleum Engineers, 2015).

[17] A. Kozlova, D. Walsh, S. Chittireddy, Z. Li, J. Natvig, S. Watanabe, and K. Bratvedt, *A hybrid approach to parallel multiscale reservoir simulator,* in *ECMOR XIV-15th European Conference on the Mathematics of Oil Recovery* (2016) amsterdam,

The Netherlands, 29 August-1 September 2016. http://dx.doi.org/10.3997/2214-4609.201601889.

[18] K. Lie, O. Møyner, J. Natvig, A. Kozlova, K. Bratvedt, S. Watanabe, and Z. Li, *Successful application of multiscale methods in a real reservoir simulator environment,* in *ECMOR XIV-15th European Conference on the Mathematics of Oil Recovery* (2016) amsterdam, The Netherlands, 29 August-1 September 2016. http://dx.doi.org/10.3997/2214-4609.201601893.

[19] H. Hajibeygi, G. Bonfigli, M. A. Hesse, and P. Jenny, *Iterative multiscale finite-volume method,* J. Comput. Phys. **227**, 8604 (2008).

[20] H. Hajibeygi and P. Jenny, *Adaptive iterative multiscale finite volume method,* Journal of Computational Physics **230**, 628 (2011).

[21] H. Hajibeygi, S. H. Lee, I. Lunati, *et al.*, *Accurate and efficient simulation of multiphase flow in a heterogeneous reservoir with error estimate and control in the multiscale finite-volume framework,* SPEJ **17**, 1 (2012), sPE-141954-PA. http://dx.doi.org/10.2118/141954-PA.

[22] L. J. Durlofsky, *Upscaling and gridding of fine scale geological models for flow simulation,* in *8th International Forum on Reservoir Simulation Iles Borromees, Stresa, Italy*, Vol. 2024 (2005).

[23] H. Li and L. J. Durlofsky, *Local–global upscaling for compositional subsurface flow simulation,* Transport in Porous Media **111**, 701 (2016).

[24] R. J. de Moraes, J. R. Rodrigues, H. Hajibeygi, and J. D. Jansen, *Multiscale gradient computation for flow in heterogeneous porous media,* Journal of Computational Physics **336**, 644 (2017).

[25] S. Krogstad, V. L. Hauge, and A. Gulbransen, *Adjoint multiscale mixed finite elements,* SPE Journal **16**, 162 (2011).

[26] R. Fonseca, A. Stordal, O. Leeuwenburgh, P. Van den Hof, and J. Jansen, *Robust ensemble-based multi-objective optimization,* in *ECMOR XIV-14th European conference on the mathematics of oil recovery* (2014) sicily, Italy, 8-11 September 2014. https://www.scopus.com/inward/record.uri?eid=2-s2.0-84908153652&partnerID=40&md5=b7532d2661cbd061688178ec4c05b1dd.

[27] R. Fonseca, O. Leeuwenburgh, P. Van den Hof, J.-D. Jansen, *et al.*, *Improving the ensemble-optimization method through covariance-matrix adaptation,* SPEJ **20**, 155 (2014), sPE-163657-PA. http://dx.doi.org/10.2118/163657-PA.

[28] A. S. Stordal, S. P. Szklarz, and O. Leeuwenburgh, *A theoretical look at ensemble-based optimization in reservoir management,* Mathematical Geosciences **48**, 399 (2016), http://dx.doi.org/10.1007/s11004-015-9598-6.

[29] Y. Chen, *Ensemble-Based Closed-Loop Production Optimization*, Ph.D. thesis, The University of Oklahoma (2008).

A

[30] J. Nocedal and S. Wright, *Numerical optimization* (Springer Science & Business Media, 2006).

[31] S. T. Do and A. C. Reynolds, *Theoretical connections between optimization algorithms based on an approximate gradient,* Computational Geosciences **17**, 959 (2013), http://dx.doi.org/10.1007/s10596-013-9368-9.

[32] J. C. Spall, *Multivariate stochastic approximation using a simultaneous perturbation gradient approximation,* IEEE transactions on automatic control **37**, 332 (1992), http://dx.doi.org/10.1109/9.119632.

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

[34] H. Zhou and H. A. Tchelepi, *Two-stage algebraic multiscale linear solver for highly heterogeneous reservoir models,* SPE J. **17**, 523 (2012).

[35] H. Zhou and H. A. Tchelepi, *Operator-based multiscale method for compressible flow,* SPE J. **13**, 523 (2008).

[36] Y. Wang, H. Hajibeygi, and H. A. Tchelepi, *Algebraic multiscale solver for flow in heterogeneous porous media,* J. Comput. Phys. **259**, 284 (2014).

[37] Y. Efendiev and T. Y. Hou, *Multiscale finite element methods: theory and applications*, Vol. 4 (Springer Science & Business Media, 2009).

[38] O. Møyner and K.-A. Lie, *A multiscale restriction-smoothed basis method for high contrast porous media represented on unstructured grids,* Journal of Computational Physics **304**, 46 (2016).

[39] P. Jenny, S. Lee, and H. Tchelepi, *Adaptive multiscale finite-volume method for multiphase flow and transport in porous media,* Multiscale Modeling & Simulation **3**, 50 (2005).

[40] K. Coats *et al.*, *IMPES stability: The CFL limit, SPE Reservoir Simulation Symposium,*, SPEJ **8**, 291 (2003), sPE-85956-PA. http://dx.doi.org/10.2118/85956-PA.

[41] Z. Chen, G. Huan, and B. Li, *An improved IMPES method for two-phase flow in porous media,* Transport in Porous Media **54**, 361 (2004), http://dx.doi.org/10.1023/B:TIPM.0000003667.86625.15.

[42] P. Jenny, S. H. Lee, and H. A. Tchelepi, *Adaptive fully implicit multi-scale finite-volume method for multi-phase flow and transport in heterogeneous porous media,* Journal of Computational Physics **217**, 627 (2006).

[43] D. Tolstolytkin, E. Borovkov, I. Rzaev, K. Y. Bogachev, *et al.*, *Dynamic modeling of samotlor field using high resolution model grids,* in *SPE Russian Oil and Gas Exploration & Production Technical Conference and Exhibition* (Society of Petroleum Engineers, 2014).

[44] Y. Efendiev, T. Hou, and T. Strinopoulos, *Multiscale simulations of porous media flows in flow-based coordinate system,* Computational Geosciences **12**, 257 (2008).

[45] S. H. Lee, H. Zhou, and H. A. Tchelepi, *Adaptive multiscale finite-volume method for nonlinear multiphase transport in heterogeneous formations,* Journal of Computational Physics **228**, 9036 (2009).

[46] Y. Efendiev, S. Lee, G. Li, J. Yao, and N. Zhang, *Hierarchical multiscale modeling for flows in fractured media using generalized multiscale finite element method,* Int. J. Geomath. **6**, 141 (2015).

[47] J. E. Aarnes and Y. Efendiev, *Mixed multiscale finite element methods for stochastic porous media flows,* SIAM Journal on Scientific Computing **30**, 2319 (2008).

[48] P. Dostert, Y. Efendiev, and T. Y. Hou, *Multiscale finite element methods for stochastic porous media flow equations and application to uncertainty quantification,* Computer Methods in Applied Mechanics and Engineering **197**, 3445 (2008).

[49] P. Jenny and I. Lunati, *Modeling complex wells with the multi-scale finite-volume method,* J. Comput. Phys. **228**, 687 (2009).

[50] Y. Wang, H. Hajibeygi, and H. A. Tchelepi, *Monotone multiscale finite volume method,* Computational Geosciences **20**, 509 (2016).

[51] Y. Wang, H. Hajibeygi, and H. A. Tchelepi, *Algebraic multiscale solver for flow in heterogeneous porous media,* Journal of Computational Physics **259**, 284 (2014).

[52] H. Hajibeygi, *Iterative multiscale finite volume method for multiphase flow in porous media with complex physics*, Ph.D. thesis, ETH Zurich (2011).

[53] M. Christie, M. Blunt, *et al.*, *Tenth spe comparative solution project: A comparison of upscaling techniques,* in *SPE Reservoir Simulation Symposium* (Society of Petroleum Engineers, 2001).

[54] Y. Saad, *ILUT: A dual threshold incomplete LU factorization,* Numerical linear algebra with applications **1**, 387 (1994), http://dx.doi.org/10.1002/nla.1680010405.

[55] J. D. Jansen, *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, 2013).

[56] Schlumberger, *Petrel,* (2016), http://www.software.slb.com/products/petrel (accessed 21 November 2016).

[57] J.-D. Jansen, R. Brouwer, S. G. Douma, *et al.*, *Closed loop reservoir management,* in *SPE reservoir simulation symposium* (Society of Petroleum Engineers, 2009).

[58] J.-D. Jansen, O. H. Bosgra, and P. M. Van den Hof, *Model-based control of multiphase flow in subsurface oil reservoirs,* Journal of Process Control **18**, 846 (2008), http://dx.doi.org/10.1016/j.jprocont.2008.06.011.

[59] G. Van Essen, P. Van den Hof, J.-D. Jansen, *et al.*, *Hierarchical long-term and short-term production optimization,* SPE Journal **16**, 191 (2011).

[60] S. T. Do and A. C. Reynolds, *Theoretical connections between optimization algorithms based on an approximate gradient,* Computational Geosciences **17**, 959 (2013), http://dx.doi.org/10.1007/s10596-013-9368-9.

[61] A. M. Manea, J. Sewall, H. A. Tchelepi, *et al.*, *Parallel multiscale linear solver for highly detailed reservoir models,* SPE Journal **21**, 2 (2016).

B

Tensors and Tensor Operations Interpretation

B

The partial derivatives of **R**, **P** and **A** matrices with respect to the vector of parameters θ result in third-order tensors. These tensors can be interpreted as a stack of matrices, with each matrix representing the partial derivatives of each of its columns with respect to the parameters. For example, a third order tensor defined by Eq. (4.39) reads

$$\frac{\partial \mathbf{P}}{\partial \theta} = \left[\begin{array}{cccc} \frac{\partial \varphi_1}{\partial \theta} & \frac{\partial \varphi_2}{\partial \theta} & \cdots & \frac{\partial \varphi_{N_C-1}}{\partial \theta} \quad \frac{\partial \varphi_{N_C}}{\partial \theta} \end{array} \right]_{N_F \times N_C \times N_\theta}, \tag{B.1}$$

where $\partial \mathbf{P}/\partial\theta$ is an $N_F \times N_C \times N_\theta$ third-order tensor that can be interpreted as a stack of N_θ matrices of partial derivatives $\partial \varphi_j/\partial\theta$, $j = 1, ..., N_C$, each of dimension $N_F \times N_C$.

We illustrate the operations with this third-order tensor by explaining the operations involved in the partial derivative Eq. (4.5) w.r.t. θ, disregarding the dependency of **R** on θ, i.e.,

$$\begin{aligned} \frac{\partial \check{\mathbf{g}}}{\partial \theta}_{N_C \times N_\theta} = & \mathbf{R}_{N_C \times N_F} \left(\overbrace{\frac{\partial \mathbf{A}}{\partial \theta}_{N_F \times \boxed{N_F} \times N_\theta} \mathbf{P}_{\boxed{N_F} \times N_C}}^{1} + \right. \\ & \left. \underbrace{\mathbf{A}_{N_F \times \boxed{N_F}} \frac{\partial \mathbf{P}}{\partial \theta}_{\boxed{N_F} \times N_C \times N_\theta}}_{2} \right) \check{\mathbf{x}}_{N_C}. \end{aligned} \tag{B.2}$$

The subscripts represent the dimensions of the tensors. The product 1 in Eq. (B.2) can be interpreted as a stack of matrices resulting from N_θ products between matrices $\partial \mathbf{A}/\partial\theta\big|_{:,:,k}$ of dimension $N_F \times N_F$ and the matrix **P** of dimension $N_F \times N_C$, $k = 1, ..., N_\theta$, and, therefore is a third-order tensor $N_F \times N_C \times N_\theta$. Analogously, product 2 also results in a third-order tensor of size $N_F \times N_C \times N_\theta$ as a stack of the products **A** times $\partial \mathbf{P}/\partial\theta\big|_{:,:,k}$, $k = 1, ..., N_\theta$. In the equations, the "squared" dimensions highlight the dimensions which are operated on in each of the matrix products. Now consider the product of the term under parenthesis in Eq. (B.2) with the vector $\check{\mathbf{x}}$,

$$\frac{\partial \check{\mathbf{g}}}{\partial \theta}_{N_C \times N_\theta} = \mathbf{R}_{N_C \times N_F} \underbrace{\left(\frac{\partial \mathbf{A}}{\partial \theta} \mathbf{P} + \mathbf{A} \frac{\partial \mathbf{P}}{\partial \theta} \right)_{N_F \times \boxed{N_C} \times N_\theta} \check{\mathbf{x}}_{\boxed{N_C}}}_{1}. \tag{B.3}$$

The product 1 in Eq. (B.3) can be interpreted as N_θ products between matrices $(\partial \mathbf{A}/\partial\theta \mathbf{P} + \mathbf{A}\partial \mathbf{P}/\partial\theta)\big|_{:,:,k}$ of dimensions $N_F \times N_C$ and the vector $\check{\mathbf{x}}$ of dimension N_C , $k = 1, ..., N_\theta$, hence leading to a matrix of dimensions $N_F \times N_\theta$, as follows

$$\frac{\partial \check{\mathbf{g}}}{\partial \theta}_{N_C \times N_\theta} = \mathbf{R}_{N_C \times \boxed{N_F}} \left(\left(\frac{\partial \mathbf{A}}{\partial \theta} \mathbf{P} + \mathbf{A} \frac{\partial \mathbf{P}}{\partial \theta} \right) \check{\mathbf{x}} \right)_{\boxed{N_F} \times N_\theta}. \tag{B.4}$$

Finally, the last operation is a simple product between two matrices with the dimensions indicated in Eq. (B.4), leading to

$$\frac{\partial \check{\mathbf{g}}}{\partial \theta}_{N_C \times N_\theta} = \left(\left(\mathbf{R} \left(\frac{\partial \mathbf{A}}{\partial \theta} \mathbf{P} + \mathbf{A} \frac{\partial \mathbf{P}}{\partial \theta} \right) \right) \check{\mathbf{x}} \right)_{N_C \times N_\theta}, \tag{B.5}$$

from where one can notice that the operations at the right-hand side lead to a matrix with the dimensions equal to those of the left-hand side matrix.

About the Author

Rafael was born and raised in the picturesque city of Nova Friburgo located on the outskirts of Rio de Janeiro, Brazil. During his formative years he was always intrigued by the dynamics of processes & machines. These interests were pursued and further strengthened during his graduation studies in Mechanical Engineering at Universidade do Estado do Rio de Janeiro. After graduating *Cum Laude*, he joined Engineering Software & Scientific Software (ESSS), through whom he was seconded as a scientific developer in the development of advanced reservoir simulation technologies and state-of-the-art algorithms for history matching, and life cycle optimization tools to the Petrobras Research and Development Center (CENPES). During this time, he developed a passion for scientific research which led him to pursue an M.Sc. degree in Civil & Petroleum Engineering from Universidade do Federal do Rio de Janeiro, with a specialization in numerical methods for reservoir simulation. Before the completion of his M.Sc. degree program, Rafael was hired by Petrobras as a Reservoir Simulation Engineer at CENPES. During the next few years, Rafael, coordinated and contributed to several national and international R&D projects. In 2010, he was assigned as the Senior Petrobras Representative at the Dynamic Reservoir Modelling System (DRMS) Joint Venture between Petrobras, Shell and CMG which was focused on the development of a highly specialized next generation reservoir simulator. This assignment took him and his beautiful wife Thays to Calgary, Alberta, Canada, where they lived for more than 4 years during which their angelic twin daughters, Estella and Helena, were born. In addition to being the Senior Petrobras representative in Calgary, Rafael also served as the Uncertainty team lead. He was awarded the prestigious CMG *Momento* in recognition of his immense contribution to success achieved by the DRMS JV. After enjoying the snowy slopes of Calgary, Rafael was sent in 2015 by Petrobras to pursue his Doctoral Studies (PhD), at Delft University of Technology, the Netherlands, under the supervision of Prof. dr.ir. Jan-Dirk Jansen and Dr. Hadi Hajibeygi. During his doctoral studies he was awarded a Fellowship for exceptional new researchers by UCLA at the Institute of Applied and Pure Mathematics, Los Angeles, California, USA. After completion of his fellowship he returned to the Netherlands to successfully defend his doctoral thesis in November 2018 all the while enjoying, in addition to his research, the sights, sounds and culture of Europe. He particularly cherished the joy of biking his daughters on his Bakfiets (Special Dutch Family Bicycle) against the strong winds in the flat plains of the Netherlands.

Rahul-Mark Fonseca
Utrecht, October 2018

LIST OF PUBLICATIONS

The following is a listing of the journal and conference publications that resulted from the PhD research, in reverse chronological order.

JOURNAL PAPERS

6. **R.J. de Moraes**, J.R.P. Rodrigues, H. Hajibeygi, J.D. Jansen, *Multiscale gradient computation for sequentially coupled flow and transport in heterogeneous porous media*, (**to be submitted**).

5. **R.J. de Moraes**, H. Hajibeygi, J.D. Jansen, *A multiscale method for data assimilation of spatially distributed data*, Computational Geosciences, (**Submitted**).

4. **R.J. de Moraes**, W. de Zeeuw, J.R.P. Rodrigues, H. Hajibeygi, J.D. Jansen, *Iterative multiscale gradient computation for heterogeneous subsurface flow*, Advances in Water Resources, (**under review**).

3. **R.J. de Moraes**, R.M. Fonseca, M. A. Helici, A. W. Heemink, J.D. Jansen, *An efficient robust optimization workflow using multiscale simulation and stochastic gradients*, Journal of Petroleum Sciences and Engineering **172**, (2019).

2. **R.J. de Moraes**, J.R.P. Rodrigues, H. Hajibeygi, J.D. Jansen, *Computing derivative information of sequentially coupled subsurface models*, Computational Geosciences (**online**) (2018).

1. **R.J. de Moraes**, J.R.P. Rodrigues, H. Hajibeygi, J.D. Jansen, *Multiscale gradient computation for flow in heterogeneous porous media*, Journal of Computational Physics **336**, 644 (2017).

CONFERENCE PROCEEDINGS

7. W. de Zeeuw, **R.J. de Moraes**, A. W. Heemink, J.D. Jansen, *Adjoint-based Adaptive Convergence Control of the Iterative Finite Volume Multiscale Method*, in SPE Reservoir Simulation Conference, Society of Petroleum Engineers, (2019).

6. **R.J. de Moraes**, H. Hajibeygi, J.D. Jansen, *A Multiscale Method For Data Assimilation*, in ECMOR XVI - 16th European Conference on the Mathematics of Oil Recovery, EAGE Publications BV, (2018).

5. **R.J. de Moraes**, H. Hajibeygi, J.D. Jansen, *Multiscale Data Assimilation of Spatially Distributed Information*, in 10th International Conference on Porous Media and Annual Meeting, Interpore, (2018).

4. **R.J. de Moraes**, H. Hajibeygi, J.D. Jansen, *Multiscale Gradient Computation for Sequentially Coupled Flow and Transport in Heterogeneous Porous Media*, in SIAM Conference on Mathematical and Computational Issues in the Geosciences, SIAM, (2017).

3. **R.J. de Moraes**, R. M. Fonseca, M. Helici, A. W. Heemink, J. D. Jansen, *Improving the Computational Efficiency of Approximate Gradients Using a Multiscale Reservoir Simulation Framework*, in SPE Reservoir Simulation Conference, Society of Petroleum Engineers, (2017).

2. **R.J. de Moraes**, J.R.P. Rodrigues, H. Hajibeygi, J.D. Jansen, *Multiscale Gradient Computation for Multiphase Flow in Porous Media*, in SPE Reservoir Simulation Conference, Society of Petroleum Engineers, (2017).

1. **R.J. de Moraes**, J.R.P. Rodrigues, H. Hajibeygi, J.D. Jansen, *Multiscale Gradient Computation for Subsurface Flow Models*, in ECMOR XV - 15th European Conference on the Mathematics of Oil Recovery, EAGE Publications BV, (2016).

ACKNOWLEDGEMENTS

Once a friend told me that the PhD success is owned by the PhD candidate solely. I cannot claim that. That is very fortunate. And I say fortunate because I was very lucky to have the opportunity to interact with many knowledgeable people who helped with the entire process, in different dimensions. I could not get to the end of it without interacting with them. Not only in the technical aspect. But also in all other aspects involved in this complex route that must be pursued to get to the end of a PhD. I am not naming every one individually here, but if you have actually been part of this process in any shape or form, I am sure you know how thankful I am.

My promotor prof. dr. ir. Jan Dirk Jansen, who always had the right words to push me further. For the trust he deposited on me, for instance by saying 'to maximize the impact of my talents'. For the opportunities he promoted, like my participation in the Data Assimilation Summer School in Romenia, the fellowship at UCLA, and so many others. Thank you for your efforts to make me a better researcher.

My copromotor, dr. Hadi Hajibeygi, for introducing me the multiscale world, for always being challenging and honest with my research. For letting me lead research activities in DARSim. The interactions with the group has provided me a insightful, broadening experience.

My Petrobras supervisor (and more importantly, friend), dr. José R. P. Rodrigues, for his endless patience while discussing every single detail of the adjoint formulation and implementation, continued support, availability, and, last but not least, friendship.

I am thankful to Petrobras for giving me the opportunity to undergo its H&R development program in order to pursue this PhD and for sponsoring all my PhD research.

My paranymphs, Flávia Pacheco, for not only being so encouraging in my professional career, but also for always being the most supportive friend when I most needed, and Matteo Cusini, the friend I found in Delft who walked along with me every corner of my PhD path, inside and outside the university. My friend Rahul Fonseca, who I also only found in Delft, but understands me and advises me as if we know each other for a life time. Thank you all for helping me to make it through.

I am especially grateful to the opportunity to have MSc students working on topics related to my PhD research project: Mircea Helici, who worked in the MS-StoSAG workflow implementation, Frank Pennekamp, who worked on the i-MSFV derivative computation formulation, Wessel de Zeeuw, who worked on both the i-MSFV derivative computation and in the multiscale-based goal-oriented method developments. Their dedication to their MSc projects and high-quality work had an impactful result on my research.

My wife Thays Moraes, who not only walked along with me the ups and downs of this PhD, but has also been my main supporter in life. I would never be able to thrive at any aspect of life without you!

Rafael Jesus de Moraes
Delft, October 2018

The improved understanding of subsurface resources and, consequently, improved decisions on how to efficiently exploit them rely on mathematical and computational models. The improvements are usually guided by optimization algorithms that require derivative information. This thesis discusses the development of efficient and accurate derivative computation techniques based on multiscale simulation strategies. The multiscale aspect of this research is twofold. On one hand, it allows for more efficient computational algorithms. On the other, the inherently multiscale nature of the physical process can be more appropriately accommodated by the mathematical models.

ISBN 978-1-2345678-9-7

TU Delft