Surrogate-assisted reservoir history matching

Xiao, C.

DOI
10.4233/uuid:0efec7ac-3198-4934-8d15-e94986ae104a

Publication date
2021

Document Version
Final published version

Citation (APA)
Xiao, C. (2021). *Surrogate-assisted reservoir history matching*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:0efec7ac-3198-4934-8d15-e94986ae104a

SURROGATE-ASSISTED RESERVOIR HISTORY MATCHING

SURROGATE-ASSISTED RESERVOIR HISTORY MATCHING

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 Woensdag 27 januari 2021, 12.30 uur

door

Cong XIAO

Master of Petroleum Engineering,
China University of Petroleum, Beijing, China,
geboren te AnLu, China.

Dit proefschrift is goedgekeurd door de
promotor: Prof. dr. ir. A. W. Heemink
promotor: Prof. dr. ir. H. X. Lin

Samenstelling promotiecommissie:

Rector Magnificus	voorzitter
Prof. dr. ir. A. W. Heemink	Technische Universiteit Delft
Prof. dr. ir. H. X. Lin	Technische Universiteit Delft

Onafhankelijke leden:

Prof. dr. ir. J. D. Jansen	Technische Universiteit Delft
Prof. dr. ir. M. Verlaan	Technische Universiteit Delft
Prof. dr. ir. J. L. Kloosterman	Technische Universiteit Delft
Prof. dr. ir. R. G. Hanea	University of Stavanger, Norway / Equinor

Overige leden:
Dr. ir. O. Leeuwenburgh — TNO, the Netherlands

TU Delft Delft University of Technology

Keywords: reservoir simulation reduced-order modeling deep neural network data assimilation model-reduced adjoint smooth local parameterization
Printed by: Ridderprint | www.ridderprint.nl

This project was funded by China Scholarship Council (CSC).

Author email: xclmjtud@yahoo.com
ISBN 978-94-6366-365-6

An electronic version of this dissertation is available at

http://repository.tudelft.nl/.

To my family

CONTENTS

SUMMARY

Reservoir history matching is a process of iteratively adjusting reservoir model parameters using measurements. This process can be computationally demanding, particularly when numerous simulations with a high-dimensional reservoir model have to be performed. In the community of petroleum engineering, the use of surrogate modelling techniques have recently gained more and more popularity to improve the efficiency of history matching. However, it is still not possible to fully utilize their potential in realistic applications. One of the challenges is to retain high accuracy while increasing the computational efficiency using a surrogate model. In order to address this problem, two approaches for surrogate modeling are proposed and analyzed in this dissertation: A projection-based reduced-order model and a data-driven deep convolutional neural network.

In the first part of the thesis, a non-intrusive subdomain POD-TPWL method for solving gradient-based reservoir history matching problems is presented. It is a projection-based reduced-order modelling approach wherein the adjoint model of the original high-dimensional non-linear model is approximated by a subdomain reduced-order linear model. Domain decomposition and radial basis function interpolation are integrated with trajectory piece wise linearization to form the subdomain POD-TPWL algorithm. This subdomain POD-TPWL has the advantage of avoiding simulator-code intrusion. This is very attractive because the source code of the model is often not available for users of commercial code. Furthermore, by introducing domain decomposition for the reduced-order model and by restricting the number of uncertain parameter patterns to the subdomains, the number of full order simulations required for the derivation of this surrogate model is reduced drastically. The proposed local parameterization enables the application of subdomain POD-TPWL to large-scale parameter estimation problems since the number of full-order model simulations depends primarily on the number of the local parameters patterns in each subdomain. By projecting the reconstructed local parameter patterns onto numerous predefined global parameter patterns, smooth global parameter patterns are finally obtained.

The advancement of deep neural network (DNN) architectures and the availability of popular deep learning packages have particularly stimulated research on data-driven surrogate modelling for high-dimensional nonlinear systems. In the second part of the thesis, we propose two kinds of deep-learning inversion frameworks for efficiently solving large-scale history matching problems. The first deep-learning deterministic inversion framework primarily explores the possibility of applying a DNN surrogate to approximate the gradient of the objective function by making use of auto-differentiation (AD). In combination with the DNN surrogate, the AD enables us to evaluate the gradients efficiently in a parallel manner and without the need of explicitly coding of the adjoint model. Furthermore, we investigate the benefits of using a stochastic gradient optimizer instead of the full gradient optimizer, as the latter is the most common option used in the community of data assimilation. The second framework is the deep-learning stochastic inversion which constructs a deep-learning surrogate based on an image-oriented distance parameterization for ensemble-based seismic history matching. Instead of directly assimilat-

ing spatially dense seismic data, image-oriented distance parameterization is employed to extract valuable information from the water fronts. Inspired by the methodologies developed for image segmentation in the field of computer vision and image processing, we propose an advanced image-segmentation network for accurately predicting water fronts with highly-complex spatial discontinuities. In comparison with the conventional workflows entirely based on high-fidelity simulation models, experimental results show that the proposed surrogate-supported workflow achieves an accuracy equal to or better than the conventional workflow at significantly lower cost.

A comparative study of these two surrogate approaches reveals that DNN is efficient in assessing model uncertainty by generating an ensemble of realizations. However, DNN requires high-performance computing resources, e.g., GPUs, while the physical interpretation of the results is often lacking. The implementation of subdomain POD-TPWL is relatively flexible and easier to diagnose. However, the sensitivity of subdomain POD-TPWL with respect to the domain decomposition scheme used cannot be ignored, while the choice of the decomposition is still rather subjective. For small sample sizes the physics based subdomain POD-TPWL is more efficient while for larger sample sizes the improvement of the accuracy of subdomain POD-TPWL tends to be limited for a given domain decomposition scheme. On the other hand, the DNN surrogate approach does not yield satisfactory results for small sample sizes, but gets better and better with larger samples sizes.

1

INTRODUCTION

Nowadays it becomes increasingly important to produce fossil fuels, e.g., oil and gas resources, in an economical and environmental friendly manner. The concept of close-loop reservoir management has proven to be a mature and practically feasible methodology for resources recovery in an optimal control strategy. This chapter describes a generic concept of simulation-based reservoir management as an effective and automatic means for the recovery of oil gas resources and presents the motivation on quantification and reduction of oil reservoir model-form uncertainties with a focus on data-driven surrogate-assisted approaches. The research questions are also explicitly introduced in this chapter.

1

1.1. SIMULATION-BASED CLOSED-LOOP RESERVOIR MANAGEMENT

Traditional fossil fuels, e.g., oil and gas resources, are the key supply to the energy consumption around the world nowadays, even though a transition to more clean and environmental friendly natural resources production is imperative [1]. On the one hand, the rapid progress of our society exacerbates the consumption of oil and gas resources [2]. On the other hand, it is now becoming more and more tough to extract new oil and natural gas fuels. For example, our attention has been shifted from onshore to offshore, e.g., deep sea [3], from conventional fossil resource to unconventional combustible ice in polar region [4]. Therefore, the need to efficiently and economically produce limited oil reserves has increasingly gain popularity [5]. An important technique, named simulation-based closed-loop reservoir management (CLRM), has been proposed to increase the oil recovery in an economical and effective manner [6]. It is an emerging research topic based on smart oilfield technology. Taking the oilfield production system as the object, the numerical simulation technology based on the mathematical model is used to organically combine two processes of data assimilation and production optimization. Briefly speaking, the process of CLRM updates the reservoir geological model by continuously fusing the observed data and determining the distribution of reservoir fluids, and then uses optimal control algorithms to optimize oilfield development [6, 7].

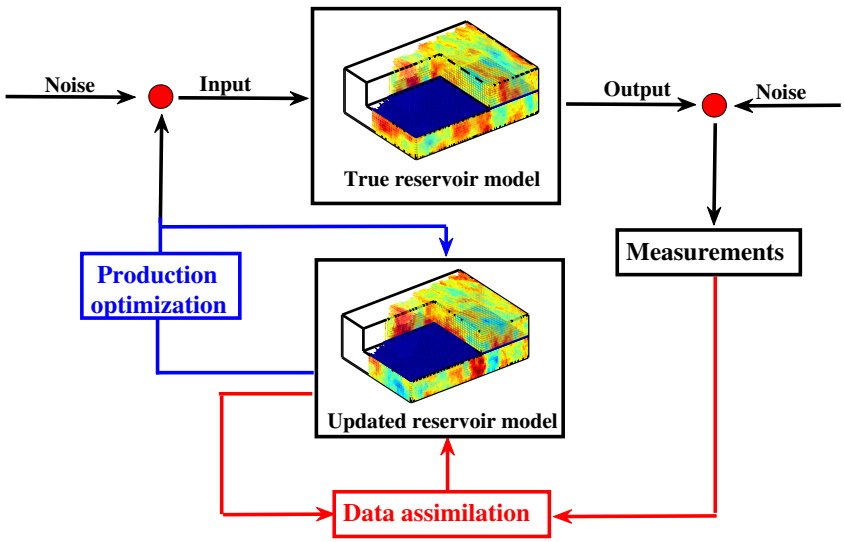


Figure 1.1: The diagram of simulation-based closed-loop reservoir management [6]. In this diagram, the red and blue parts represent the data assimilation and production optimization, respectively.

The inspiration for this CLRM concept mainly stems from data assimilation techniques commonly used when dealing with large-scale flow models in meteorology and oceanography [8], as well as model-based control theories [9]. CLRM considers reservoir development as a simulation-based closed-loop control process, and its core mainly includes two aspects: model update based on data assimilation [10, 11] and well-control optimization

based on reservoir model [12]. Fig.1.1 displays reservoir management as a simulation-based control process. As shown in the red part of this figure, reservoir engineers usually employ assisted history matching methods to characterize reservoir geological models and fluid parameters by fitting reservoir production data. The conventional logging, well testing and other methods have long data collection periods and small amounts of data, which easily lead to very large model errors and limited prediction accuracy. Smart well permanent down-hole sensor technology makes direct acquisition of more and more reservoir parameters. The accumulation of a large amount of detection data has created conditions for the application of history matching algorithm in the field of reservoir development. The blue part of the figure shows how to apply the most effective control theory to reservoir production and development based on the model update, and determine the optimal production strategy to maximize net present value via producing the oil/gas resources as much as possible.

Mathematical model constitutes an indispensable part of production optimization and data assimilation in closed-loop reservoir management framework [13]. Therefore, it must be closely connected with the key elements of CLRM process. Currently, a broad feature of the reservoir flow system is a mathematical model that is represented by a set of equations coupled with boundary and/or initial conditions [14]. The form of the model equation mainly depends on the nature of the reservoir, that is, rocks, fluids and their dynamic interactions in porous media. The knowledge of rock and fluid properties is usually very limited, although it is essential for creating accurate and representative reservoir models. The fact that rock properties usually vary in space (i.e., heterogeneity) causes other difficulties as well [15].

The concept of closed-loop reservoir management framework originally stems from the control theory, but it is often impractical to directly convert to reservoir management due to the high complexity of the realistic reservoir models [16–18]. The two processes of reservoir closed-loop management are both large-scale optimization problems, which require highly computational cost. Although computing resources increase rapidly, model complexity tends to increase at the same rate, especially for the field-scale applications. Therefore, to make the optimization workflow truly effective, it is necessary to significantly shorten the simulation time. This makes model simplification technology or alternative fast emulation method an important research area. In this dissertation, we mainly concentrate on developing efficient solutions to resolve the data assimilation process.

1.2. SURROGATE-ASSISTED DATA ASSIMILATION METHODS

1.2.1. DATA ASSIMILATION

We address the problem of computationally efficient estimation of spatially varying parameters in large-scale simulation models. Simulation models typically contain 10^5 – 10^6 spatially related but largely unknown model parameters, data are sparse and frequently indirect, and relationships between parameters and data tend to be highly nonlinear. The unknown parameters include, amongst others, the properties of discrete volumes of porous rock at depths up to a few kilometers, and functional models of interaction between fluids that are present in the pores of that rock. Measured data are typically obtained at wells that are drilled from the surface and are used to either produce fluids from the reservoir or inject fluids into it in order to displace the fluids present in the reservoir. These wells are normally sparsely distributed over large areas, leaving the generally heterogeneous

reservoir rock in-between the wells unobserved. Alternative ways of gathering information are based on geophysical techniques, such as reflection seismic, which register the arrival time and amplitude of acoustic waves that are reflected at contrasts in acoustic impedance, which in turn is affected by the density and mechanical properties of the rock and fluids. Such geophysical information is often of low spatial resolution (especially in the vertical, or depth, direction). All these aspects make the reservoir history matching a challenging computational and mathematical problem.

In the field of ocean modelling and numerical weather prediction, data assimilation has been extensively studied for several decades. Generally, the sequential and variational data assimilation algorithms are the two classical categories in the data assimilation community. Naevdal et al for the first time applied the ensemble kalman filter (EnKF) to calibrate the reservoir geological parameters [19] around the well position, and a literature review about the applications of the EnKF in petroleum engineering is referred to [20]. Chen et al was the first to use the adjoint method [21] to effectively resolve the predefined objective function in variational algorithms. Subsequent works have applied this method to the reservoir history matching and production optimization [22–24]. A review on adjoint-based reservoir optimization is given in [25]. The computational burden has been always one of the most severe challenges and hence strongly restricts the applications of data assimilation algorithms to more practical reservoir history matching problems [26–30].

1.2.2. SURROGATE MODELLING

Surrogate modelling aims at providing a faster emulation with a simplified relation between the inputs and outputs of a complex model. Surrogate models with a reduced complexity have been extensively applied in various fields with the aim of reducing computational cost. Reviews of surrogate modelling can be found in [31, 32]. To the best knowledge, the surrogate models can be roughly classified into three categories: hierarchical-based, projection-based reduced-order modelling (ROM) and data-driven surrogate models. In hierarchical-based methods the surrogate is created by simplifying the representation of the physical system, such as by ignoring certain processes, or reducing the numerical resolution or the complexity, e.g., upscaling and grid coarsening [33, 34]. Reduced-order modelling approaches have been always increasing popularity as one of the most effective means to reduce the computational effort of model-based workflows through reducing the model dimensionality. The main idea behind projection-based ROM is to construct a (linear) low-order surrogate model by projecting the dynamics of the system onto the subspace of dominant variability of the model dynamics. Most ROM strategies employ proper orthogonal decomposition (POD) [35, 36] of time series of model state 'snapshots' to identify an orthogonal set of basis functions of the subspace [37, 38]. Such ROM strategy has been applied with success in speeding up model simulations in computational fluid mechanics [39–41], subsurface flow simulations [42–44], and air pollution [45, 46].

Data-driven approaches purely depend only on data (simulated or real data) to accurately build the relations between input and quantities of interest as much as possible, such as polynomial chaos expansion [47] and fully-connected artificial neural networks [48]. Recently, the availability of high-performance units, e.g., GPU, and more and more available data greatly prompt a major advancement of machine learning methods, especially those based on deep learning techniques. This advancement subsequently stimulates some successful applications of deep-learning methods in various research areas, especially for the image recognition [49, 50] and natural language processing [51, 52]. The

advancement of deep neural networks has strongly stimulated their successful applications with promising performance [53–57]. The popularity of these methods is further enhanced by the availability of open access machine learning frameworks, such as TensorFlow [58] and PyTorch [59].

1.2.3. RESEARCH OBJECTIVE

This dissertation aims at developing surrogate-assisted methods to efficiently address large-scale reservoir history matching problem in the community of petroleum engineering. Although several optimization algorithms are capable of providing a robust way to infer unknown parameters for strongly nonlinear models, in some cases it may not be feasible to perform the history matching within an acceptable short time frame. This limitation has stimulated the development of efficient optimization strategies in oil reservoir engineering. The surrogate modelling is currently identified as one of the most promising means to improve the efficiency of reservoir simulations. In this dissertation, incorporating surrogate modelling techniques into data assimilation algorithms will be systematically investigated.

1.3. RESEARCH QUESTIONS

The contribution of this dissertation to the reservoir data assimilation community are two types of data-driven surrogate models and illustrations of their applicability to speedup reservoir history matching problems. Overall, we will address below five research questions:

Research question 1: The commonly used reduced-order modeling aims primarily at approximating the dynamic response of the full model as accurately as possible. Can approximate gradients estimated by non-intrusive Reduced Order Modelling (ROM) be used to efficiently estimate geological parameters of reservoir models with sufficient accuracy?

Research question 2: From a computational point of view, the global parameters are decomposed in each subdomain separately should be very attractive. Can local parameterization by subdomain-based decomposition help to improve the computational efficiency of ROM-based parameter estimation for large-scale problems?

Research question 3: The reduced-order modeling approach is shown to be applicable for certain problems. How can alternative surrogate modelling approaches inspired by recent advances in deep learning technology be used to speed up the history matching process?

Research question 4: Since both deep-learning surrogate model and reduced-order model are intentionally used for the gradient-based history matching, what are the relative benefits and disadvantages of deep-learning and reduced order surrogate modelling approaches for gradient-based history matching?

Research question 5: As an alternative to gradient-based seismic history matching, how can ensemble-based history matching approaches benefit from the use of deep learning surrogate models, especially for the challenging application of large-scale seismic his-

tory matching?

1.4. THESIS LAYOUT

The main content of this dissertation is arranged in the "manuscript" format ("multi-paper" format), in which the main chapters , e.g., from **Chapter 3** to **Chapter 7**, are the full or part of journal/conference papers (published or submitted). The chapters that address similar topics may share similar motivations, arguments and materials.

Chapter 1 describes the background of simulation-based reservoir management workflow and presents the motivation on surrogate-assisted approaches for speeding up reservoir history matching. The research questions are also explicitly described in this chapter.

In **Chapter 2**, we first introduce the research application of this dissertation, e.g., the reservoir history matching problem, through a mathematical definition of objective functions. A generic literature review about the data assimilation algorithms in the context of petroleum engineering is also given. Finally, the adjoint-based deterministic optimization method and ensemble-based stochastic optimization method, are derived briefly.

To answer **Research question 1**, **Chapter 3** presents a projection-based subdomain POD-TPWL for approximating the gradient of the original high-dimensional non-linear reservoir model. In **Chapter 4**, we explore the potential of applying the above reduced-order model to large-scale reservoir parameter estimation problem by representing the spatial parameters in each subdomain individually. The influences of decompositions on scalability of the methodology are investigated as well. The results of this chapter are used to answer **Research question 2**. Since our proposed projection-based reduced-order modelling is very sensitive to the domain decomposition schemes, **Chapter 5** presents an adaptive decomposition strategy for mitigating the negative effects of inappropriate domain decomposition schemes. In order to resolve **Research question 3** and **Research question 4**, **Chapter 6** mainly focuses on developing an efficient deep-learning deterministic inversion framework to efficiently address the gradient-based history matching. This chapter primarily explores the possibility of applying deep-learning surrogate to approximate the gradient of seismic history matching by using the auto-differentiation (AD). We also conduct a comparative study between subdomain POD-TPWL and deep-learning surrogate to investigate their scalable efficiency and accuracy in large-scale applications. Corresponding to the last **Research question 5**, **Chapter 7** proposes a hybrid workflow through combining deep-learning model and image-oriented distance parameterization for ensemble-based seismic history matching. Specifically, a deep neural network for image segmentation is constructed to predict the water fronts, from which valuable information can be extracted along with a significant reduction of data volume.

Finally, the last **Chapter 8** concludes the dissertation by highlighting the key contributions and discussing possible research directions for the future work.

REFERENCES

[1] M. Roser, H. Ritchie, and E. Ortiz-Ospina, *World population growth,* Our World in Data (2013).

[2] B. Dudley *et al.*, *Bp statistical review of world energy,* BP Statistical Review, London, UK, accessed Aug **6**, 2018 (2018).

[3] X. Li, C. Wang, W. Jin, and B. Ran, *A review on deep-sea sedimentation theory: Significances to oil-gas exploration,* Acta Sedimentologica Sinica **27**, 77 (2009).

[4] G. Datong, *"combustible ice", when will it can be large-scale mined?* Environmental Research and Monitoring , 19 (2017).

[5] T. Ahmad and D. Zhang, *A critical review of comparative global historical energy consumption and future demand: The story told so far,* Energy Reports **6**, 1973 (2020).

[6] 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).

[7] J. Jansen, S. Douma, D. Brouwer, *et al.*, *Closed-loop reservoir management. paper spe 119098 presented at the spe reservoir simulation symposium, the woodlands, texas, 2–4 february,* (2009).

[8] B. A. Foss, J. P. Jensen, *et al.*, *Performance analysis for closed-loop reservoir management,* SPE Journal **16**, 183 (2011).

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

[10] G. Evensen, *Data assimilation: the ensemble Kalman filter* (Springer Science & Business Media, 2009).

[11] D. S. Oliver and Y. Chen, *Recent progress on reservoir history matching: a review,* Computational Geosciences **15**, 185 (2011).

[12] P. Sarma, L. J. Durlofsky, K. Aziz, and W. H. Chen, *Efficient real-time reservoir management using adjoint-based optimal control and model updating,* Computational Geosciences **10**, 3 (2006).

[13] J. Hou, K. Zhou, X.-S. Zhang, X.-D. Kang, and H. Xie, *A review of closed-loop reservoir management,* Petroleum Science **12**, 114 (2015).

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

[15] R. E. Ewing, *The mathematics of reservoir simulation* (SIAM, 1983).

[16] A. Capolei, E. Suwartadi, B. Foss, and J. B. Jørgensen, *A mean–variance objective for robust production optimization in uncertain geological scenarios,* Journal of Petroleum Science and Engineering **125**, 23 (2015).

[17] 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).

[18] M. M. Siraj, P. M. Van den Hof, and J. D. Jansen, *Handling risk of uncertainty in model-based production optimization: a robust hierarchical approach,* IFAC-PapersOnLine **48**, 248 (2015).

[19] Æ. Geir, T. Mannseth, E. H. Vefring, *et al.*, *Near-well reservoir monitoring through ensemble kalman filter,* in *SPE/DOE improved oil recovery symposium* (Society of Petroleum Engineers, 2002).

[20] S. I. Aanonsen, G. Nævdal, D. S. Oliver, A. C. Reynolds, B. Vallès, *et al.*, *The ensemble kalman filter in reservoir engineering–a review,* Spe Journal **14**, 393 (2009).

[21] R. Li, A. C. Reynolds, D. S. Oliver, *et al.*, *History matching of three-phase flow production data,* in *SPE reservoir simulation symposium* (Society of Petroleum Engineers, 2001).

[22] S. Vlemmix, G. J. Joosten, R. Brouwer, J.-D. Jansen, *et al.*, *Adjoint-based well trajectory optimization,* in *EUROPEC/EAGE conference and exhibition* (Society of Petroleum Engineers, 2009).

[23] G. Van Essen, E. Jimenez, J. K. Przybysz-jarnut, L. Horesh, S. G. Douma, P. van den Hoek, A. Conn, *et al.*, *Adjoint-based history-matching of production and time-lapse seismic data,* in *SPE Europec/EAGE Annual Conference* (Society of Petroleum Engineers, 2012).

[24] E. Suwartadi, S. Krogstad, and B. Foss, *Adjoint-based surrogate optimization of oil reservoir water flooding,* Optimization and Engineering **16**, 441 (2015).

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

[26] M. Verlaan and A. Heemink, *Nonlinearity in data assimilation applications: A practical method for analysis,* Monthly Weather Review **129**, 1578 (2001).

[27] Y. Cao, J. Zhu, I. M. Navon, and Z. Luo, *A reduced-order approach to four-dimensional variational data assimilation using proper orthogonal decomposition,* International Journal for Numerical Methods in Fluids **53**, 1571 (2007).

[28] Y. Chen and D. Zhang, *Data assimilation for transient flow in geologic formations via ensemble kalman filter,* Advances in Water Resources **29**, 1107 (2006).

[29] A. Wilson *et al.*, *Field-scale assisted history matching using a systematic ensemble kalman smoother,* Journal of Petroleum Technology **69**, 93 (2017).

[30] B. Lin, P. Crumpton, A. Dogru, *et al.*, *Parallel implementation of ensemble kalman smoother for field-scale assisted history matching,* in *SPE Middle East Oil & Gas Show and Conference* (Society of Petroleum Engineers, 2017).

[31] S. Razavi and B. Tolson, *Review of surrogate modeling in water resources,* Water Resources Research **48** (2012).

[32] R. Alizadeh, J. K. Allen, and F. Mistree, *Managing computational complexity using surrogate models: a critical review,* Research in Engineering Design **31**, 275 (2020).

[33] H. Ates, A. Bahar, S. E.-A. Salem, M. Charfeddine, *et al.*, *Ranking and upscaling of geostatistical reservoir models using streamline simulation: A field case study,* SPE Reservoir Evaluation & Engineering **8**, 22 (2005).

[34] M. O. Salazar, J. R. Villa Piamo, *et al.*, *Permeability upscaling techniques for reservoir simulation,* in *Latin American & Caribbean Petroleum Engineering Conference* (Society of Petroleum Engineers, 2007).

[35] P. Vermeulen and A. Heemink, *Model-reduced variational data assimilation,* Monthly weather review **134**, 2888 (2006).

[36] M. U. Altaf, A. W. Heemink, and M. Verlaan, *Inverse shallow-water flow modeling using model reduction,* International journal for multiscale computational engineering **7** (2009).

[37] M. P. Kaleta, R. G. Hanea, A. W. Heemink, and J.-D. Jansen, *Model-reduced gradient-based history matching,* Computational Geosciences **15**, 135 (2011).

[38] C. Xiao, O. Leeuwenburgh, H. X. Lin, and A. Heemink, *Non-intrusive subdomain pod-tpwl for reservoir history matching,* Computational Geosciences **23**, 537 (2019).

[39] D. Xiao, F. Fang, A. G. Buchan, C. C. Pain, I. M. Navon, and J. Du, *Non-linear model reduction for the navier–stokes equations using residual deim method,* Journal of Computational Physics **263**, 1 (2014).

[40] D. Xiao, P. Yang, F. Fang, J. Xiang, *et al.*, *Non-intrusive reduced order modelling of fluid–structure interactions,* Computer Methods in Applied Mechanics and Engineering **303**, 35 (2016).

[41] S. Hijazi, G. Stabile, A. Mola, and G. Rozza, *Data-driven pod-galerkin reduced order model for turbulent flows,* Journal of Computational Physics , 109513 (2020).

[42] E. Gildin, H. Klie, A. A. Rodriguez, M. F. Wheeler, *et al.*, *Development of low-order controllers for high-order reservoir models and smart wells, SPE Annual Technical Conference and Exhibition*, (2006).

[43] R. Djeddi, A. Kaminsky, and K. Ekici, *Convergence acceleration of fluid dynamics solvers using a reduced-order model,* AIAA Journal **55**, 3059 (2017).

[44] M. A. Cardoso, *Development and application of reduced-order modeling procedures for reservoir simulation* (Stanford University, 2009).

[45] R. Djeddi, A. Kaminsky, and K. Ekici, *Convergence acceleration of fluid dynamics solvers using a reduced-order model,* AIAA Journal **55**, 3059 (2017).

[46] A. Manzoni, F. Salmoiraghi, and L. Heltai, *Reduced basis isogeometric methods (rb-iga) for the real-time simulation of potential flows about parametrized naca airfoils,* Computer Methods in Applied Mechanics and Engineering **284**, 1147 (2015).

[47] C. Dai, L. Xue, D. Zhang, and A. Guadagnini, *Data-worth analysis through probabilistic collocation-based ensemble kalman filter,* Journal of Hydrology **540**, 488 (2016).

[48] M. A. Ahmadi, *Developing a robust surrogate model of chemical flooding based on the artificial neural network for enhanced oil recovery implications,* Mathematical Problems in Engineering **2015** (2015).

[49] A. Savchenko, *Probabilistic neural network with complex exponential activation functions in image recognition using deep learning framework,* arXiv preprint arXiv:1708.02733 (2017).

[50] Y. J. Heo, S. J. Kim, D. Kim, K. Lee, and W. K. Chung, *Super-high-purity seed sorter using low-latency image-recognition based on deep learning,* IEEE Robotics and Automation Letters **3**, 3035 (2018).

[51] T. Young, D. Hazarika, S. Poria, and E. Cambria, *Recent trends in deep learning based natural language processing,* ieee Computational intelligenCe magazine **13**, 55 (2018).

[52] X. He, J. Gao, and L. Deng, *Deep learning for natural language processing: Theory and practice,* Deep Learning Technology Center Microsoft Research, Redmond, WA (2014).

[53] S. W. Cheung, E. T. Chung, Y. Efendiev, E. Gildin, Y. Wang, and J. Zhang, *Deep global model reduction learning in porous media flow simulation,* Computational Geosciences **24**, 261 (2020).

[54] S. Mo, N. Zabaras, X. Shi, and J. Wu, *Deep autoregressive neural networks for high-dimensional inverse problems in groundwater contaminant source identification,* Water Resources Research **55**, 3856 (2019).

[55] S. Mo, N. Zabaras, X. Shi, and J. Wu, *Deep autoregressive neural networks for high-dimensional inverse problems in groundwater contaminant source identification,* Water Resources Research **55**, 3856 (2019).

[56] Y. Zhu, N. Zabaras, P.-S. Koutsourelakis, and P. Perdikaris, *Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data,* Journal of Computational Physics **394**, 56 (2019).

[57] Y. Zhu and N. Zabaras, *Bayesian deep convolutional encoder–decoder networks for surrogate modeling and uncertainty quantification,* Journal of Computational Physics **366**, 415 (2018).

[58] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, *et al.*, *Tensorflow: Large-scale machine learning on heterogeneous distributed systems,* arXiv preprint arXiv:1603.04467 (2016).

[59] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, *Pytorch: An imperative style, high-performance deep learning library,* in *Advances in neural information processing systems* (2019) pp. 8026–8037.

2

PROBLEM DESCRIPTION AND FORMULATIONS

In this chapter, the system of equations governing the flow of oil and water through a reservoir is briefly described. Both the gradient-based and ensemble-based data assimilation algorithms are introduced. Furthermore, the mathematical definition of reservoir history matching is described, on the basis of which numerical experiments throughout this thesis will be conducted.

2.1. DEFINITION OF RESERVOIR HISTORY MATCHING

Water flooding is the most widely used developing strategy to increase the oil recovery after the primary oil recovery stage. Our research mainly focuses on history matching geological parameters of a reservoir with the oil/water two phase flowing system. The reservoir model is used to simulate a water-flooding system for producing oil resources. The effectiveness of the sweep of the oil by the injected water depends on, amongst other factors, the spatial heterogeneity of reservoir geological characteristics, most importantly the porosity and permeability. Without loss of generality, we simplify an explicit formula for a single simulation step of a discretized two-phase oil-water reservoir system as follows,

$$\mathbf{x}^n = \mathbf{f}^n(\mathbf{x}^{n-1}, \boldsymbol{\beta}), \quad n = 1, \cdots, N_t \tag{2.1}$$

where, $\mathbf{f}^{n+1}$ represents a nonlinear model operator. $\mathbf{x}^{n+1}$ represents the state vector (pressure and saturation in all gridblocks). n and $n+1$ represents two consecutive timesteps. N_t denotes the total number of simulation steps. $\boldsymbol{\beta}$ denotes a vector of reservoir geological parameters, which are the spatial permeability fields in our study. More details about the discretization of the governing equations can be referred to [1].

The relationship between the measured data $\mathbf{y}^m$ and the model state and reservoir parameters can be described by a measurement operator $\mathbf{h}^m$

$$\mathbf{y}^m = \mathbf{h}^m(\mathbf{x}^m, \boldsymbol{\beta}) + \mathbf{r}^m, \quad m = 1, \cdots, N_{obs} \tag{2.2}$$

where,N_{obs} is the number of time steps at which the measurements are taken (a subset of the N_t simulation time steps), and $\mathbf{r}^m$ is a vector of measurement errors for the data gathered at time step m. These errors are assumed to satisfy a Gaussian distribution $\mathbf{G}(\mathbf{0}, \mathbf{R}^m)$ where $\mathbf{R}^m$ is the measurement error covariance matrix. The specific formula of this nonlinear operator depends on the type of measured data. In this study we consider two types of data, namely well data (fluid rate and bottom-hole pressure measured at the well locations only), and grid-based saturations (assumed to be observed in all gridblocks). Grid-based saturations are used to mimic time-lapse seismic data from which water saturation can be interpreted from a seismic inversion [2].

The history matching process calibrates the uncertain parameters by minimizing an objective function defined as a sum of weighted squared differences between observed and simulated data. Incorporating prior information into the objective function can further regularize the history matching problem [3]. In general, the objective function can be described by a sum of two terms

$$\begin{aligned} J(\mathbf{x}^1, \cdots, \mathbf{x}^n, \cdots, \mathbf{x}^{N_t}, \boldsymbol{\beta}) &= \frac{1}{2}(\boldsymbol{\beta} - \boldsymbol{\beta}_b)^T \mathbf{R}_b{}^{-1}(\boldsymbol{\beta} - \boldsymbol{\beta}_b) \\ &+ \frac{1}{2}\sum_{m=1}^{N_{obs}} [\mathbf{d}_{obs}^m - \mathbf{h}^m(\mathbf{x}^m, \boldsymbol{\beta})]^T (\mathbf{R}^m)^{-1} [\mathbf{d}_{obs}^m - \mathbf{h}^m(\mathbf{x}^m, \boldsymbol{\beta})] \end{aligned} \tag{2.3}$$

where, $\mathbf{d}_{obs}^m$ represents a vector of observed data at timestep m. The reservoir parameters $\boldsymbol{\beta}$ is assumed to follow a Gaussian distribution $\mathbf{G}(\boldsymbol{\beta}_b, \mathbf{C}_b)$. In validation experiments, $\mathbf{d}_{obs}^m$ is generated by adding some noise, e.g., $\mathbf{r}^m$, to the simulated data from the reference model.

The randomized maximum likelihood (RML) procedure [4, 5] enables the assessment of the uncertainty by generating multiple 'samples' from the posterior distribution. Each of these samples is a history matched realization, which also honors any conditioning

data, e.g., well data and seismic data in our study. We also should note that RML is not an exact but rather an approximate method to generated a sample (set) of the posterior distribution. Proper correction steps are required to yield an unbiased quantification of uncertainty [6, 7]. This is not the main issue here and therefore a standard RML procedure will be implemented in this thesis.

In order to generate multiple posterior realizations using RML procedure, we need to minimize the objective functions corresponding to a set of perturbed measurements,

$$J^i_{RML}(\mathbf{x}^1,\cdots,\mathbf{x}^n,\cdots,\mathbf{x}^{N_t},\boldsymbol{\beta}) = \frac{1}{2}(\boldsymbol{\beta}-\boldsymbol{\beta}^i_b)^T\mathbf{R}_b{}^{-1}(\boldsymbol{\beta}-\boldsymbol{\beta}^i_b) + \frac{1}{2}\sum_{m=1}^{N_{obs}}[\mathbf{d}^{m,i}_{obs}-\mathbf{h}^m(\mathbf{x}^m,\boldsymbol{\beta})]^T(\mathbf{R}^m)^{-1}[\mathbf{d}^{m,i}_{obs}-\mathbf{h}^m(\mathbf{x}^m,\boldsymbol{\beta})] \quad (2.4)$$

In Eq.2.4, $\boldsymbol{\beta}^i_b \sim \mathrm{G}(\boldsymbol{\beta}_b,\mathbf{R}_b)$ and $\mathbf{d}^{m,i}_{obs} \sim \mathrm{G}(\mathbf{d}^m_{obs},\mathbf{R}_m)$ for $i = 1,2,\cdots,N_e$, where N_e is the number of perturbations to be generated. History matching is to minimize the objective functions using data assimilation algorithms which can be used to determine a parameter set that is not too far away from the prior information while minimizing the misfit between the observed and simulated data. This dissertation mainly focuses on addressing the above reservoir history matching problems in an efficient manner.

2.2. DATA ASSIMILATION IN PETROLEUM ENGINEERING

The aim of data assimilation is to infer model parameters or states through combining the dynamic model and available measurements. Data assimilation (DA) has been investigated for several decades in various fields, such as ocean modeling and numerical weather prediction. The sequential and variational algorithms are the two classical categories in the DA community. One branch of the sequential data assimilation technique is the ensemble-based Kalman filter (EnKF) proposed in [8] and its variants, including ensemble kalman smoother (EnKS), ensemble smoother (ES), ensemble square root filter (EnSQR), etc, which have been successfully applied to various fields [9–11]. Typically a suit of dynamic models runs independently forward in time (forecast step), and is continuously updated as new measurements become available (analysis step). To tackle unphysical updating problems of the sequential data assimilation methods [12], iterative schemes with an outer-loop are applied in the presence of non-linearity, for example the water-oil two phase reservoir model studied in this thesis. The fluid properties, e.g. oil/water density and relative permeability curves, depend on the primary state variables, i.e., pressure and phase saturation. This dependence introduces non-linearity in the history matching. The most thorough method among them is to rerun the forward model from the initial time using the updated parameters, and can be referred to restarted EnKF (REnKF) [13, 14]. Similarly, as variants of EnKF, iterative ES also subsequently have been developed [15–17]. In the community of reservoir engineering, Naevdal et al. first proposed the application of the ensemble kalman filter (EnKF) to calibrate the reservoir geological parameters [18], and a literature review about the applications of the EnKF in petroleum engineering is referred to [19].

Variational data assimilation, e.g., 4D-Var, is an another option to address reservoir history matching problem. Comparing to the aforementioned sequential data assimilation, 4D-Var is not limited to search the optimal solution in a prescribed finite ensemble subspace as done in the EnKF scheme. A particular efficient approach to solve 4D-Var

inversion problems is the gradient-based minimization algorithm, where the gradients of objective function with respect to the uncertain variables are efficiently obtained by incorporating an adjoint model [20]. In reservoir engineering, Chen et al was the first to use the adjoint method [21], subsequent works have applied this method into the reservoir history matching and production optimization problems [22–24]. A review on adjoint-based reservoir optimization can be found in [25]. The adjoint approach has high computational efficiency because just a single forward simulation and a single backward simulation of the adjoint model are required to compute one gradient, independent on the number of uncertain parameters. However, implementing the adjoint model generally requires access to the source code and generally requires an overwhelming programming effort, especially for high dimensional complex system. Moreover quite often the code of the forward model is not always available, especially for commercial simulators.

2.3. ADJOINT-BASED OPTIMIZATION METHOD

The key step of an adjoint gradient estimation is to determine the gradient of the objective function with respect to the parameters using the adjoint model [25]. One option to formulate the adjoint model is to employ Lagrange multipliers to replace the PDE-constrained optimization problem by an unconstrained optimization. We define an new objective function $\hat{J}$ through adjoining the system equation (Eq.2.1) to the original objective function $J(\mathbf{x}_1,\cdots,\mathbf{x}_{N_t},\boldsymbol{\beta})$ as follows

$$\hat{J}(\mathbf{x}^1,\cdots,\mathbf{x}^n,\cdots,\mathbf{x}^{N_t},\boldsymbol{\beta}) = J(\mathbf{x}^1,\cdots,\mathbf{x}^n,\cdots,\mathbf{x}^{N_t},\boldsymbol{\beta}) + \sum_{n=1}^{N_t}[\boldsymbol{\lambda}^n]^T[\mathbf{x}^n - \mathbf{f}^n(\mathbf{x}^{n-1},\boldsymbol{\beta})] \tag{2.5}$$

where $\boldsymbol{\lambda}_n^T$ represents the adjoint variables at time instance n. And then we can further obtain the total variation of $\hat{J}$ with respect to $d\mathbf{x}_n$, for n=1,$\cdots$,N_t, and $d\boldsymbol{\beta}$ is

$$d\hat{J} = dJ + \sum_{n=1}^{N_t}\boldsymbol{\lambda}_n^T d\mathbf{x}_n - \sum_{n=1}^{N_t}\boldsymbol{\lambda}_n^T\frac{\partial \mathbf{f}_n}{\partial \mathbf{x}_{n-1}}d\mathbf{x}_{n-1} - \sum_{n=1}^{N_t}\boldsymbol{\lambda}_n^T\frac{\partial \mathbf{f}_n}{\partial \boldsymbol{\beta}} \tag{2.6}$$

We can rearrange the above equation by changing the terms $\mathbf{x}_{n-1}$

$$d\hat{J} = dJ + \sum_{n=1}^{N_t}\boldsymbol{\lambda}_n^T d\mathbf{x}_n - \sum_{n=0}^{N_t-1}\boldsymbol{\lambda}_{n+1}^T\frac{\partial \mathbf{f}_{n+1}}{\partial \mathbf{x}_n}d\mathbf{x}_n - \sum_{n=1}^{N_t}\boldsymbol{\lambda}_n^T\frac{\partial \mathbf{f}_n}{\partial \boldsymbol{\beta}}d\boldsymbol{\beta} \tag{2.7}$$

and further reformulating the limits of the sums involving $\mathbf{x}_n$ produces

$$d\hat{J} = dJ + \sum_{n=1}^{N_t}(\boldsymbol{\lambda}_n^T - \boldsymbol{\lambda}_{n+1}^T\frac{\partial \mathbf{f}_{n+1}}{\partial \mathbf{x}_n})d\mathbf{x}_n - \boldsymbol{\lambda}_1^T\frac{\partial \mathbf{f}_1}{\partial \mathbf{x}_0}d\mathbf{x}_0 + \boldsymbol{\lambda}_{N_t+1}^T\frac{\partial \mathbf{f}_{N_t+1}}{\partial \mathbf{x}_{N_t}}d\mathbf{x}_{N_t} - \sum_{n=1}^{N_t}\boldsymbol{\lambda}_n^T\frac{\partial \mathbf{f}_n}{\partial \boldsymbol{\beta}}d\boldsymbol{\beta} \tag{2.8}$$

Since we focus on the uncertainty of geological parameters **m**and other conditions, e.g., boundary condition and initial condition are not time-varying, the term $\boldsymbol{\lambda}_1^T\frac{\partial \mathbf{f}_1}{\partial \mathbf{x}_0}d\mathbf{x}_0$ involving the initial condition should be $\mathbf{0}$. The term $\boldsymbol{\lambda}_{N_t+1}^T\frac{\partial \mathbf{f}_{N_t+1}}{\partial \mathbf{x}_{N_t}}d\mathbf{x}_{N_t}$ which relates to the last time index also should be zero for any differential $\mathbf{x}_{N_t}$, which implies that $\boldsymbol{\lambda}_{N_t+1} = \mathbf{0}$. In addition, the sum of the variations of the original objective function J, which is shown to

be the first term of Eq. 2.8, represents the total variations with respect to all state variables $\mathbf{x}_n$ and parameters $\boldsymbol{\beta}$

$$dJ = \sum_{n=1}^{N_t} \frac{\partial J}{\partial \mathbf{x}_n} d\mathbf{x}_n + \frac{\partial J}{\partial \boldsymbol{\beta}} d\boldsymbol{\beta} \tag{2.9}$$

Substituting the above equation into Eq.2.8 yields

$$d\hat{J} = \sum_{n=1}^{N_t} \{\frac{\partial J}{\partial \mathbf{x}_n} + \boldsymbol{\lambda}_n^T - \boldsymbol{\lambda}_{n+1}^T \frac{\partial \mathbf{f}_{n+1}}{\partial \mathbf{x}_n}\} d\mathbf{x}_n - \sum_{n=1}^{N_t} \boldsymbol{\lambda}_n^T \frac{\partial \mathbf{f}_n}{\partial \boldsymbol{\beta}} d\boldsymbol{\beta} + \frac{\partial J}{\partial \boldsymbol{\beta}} d\boldsymbol{\beta} \tag{2.10}$$

We set all coefficients of $d\mathbf{x}_n$ to zero, and then take the transpose of the above equations. Finally, the system of the adjoint equations is given as follows

$$(\frac{\partial J}{\partial \mathbf{x}_n})^T + \boldsymbol{\lambda}_n - (\frac{\partial \mathbf{f}_{n+1}}{\partial \mathbf{x}_n})^T \boldsymbol{\lambda}_{n+1} = \mathbf{0} \tag{2.11}$$

for $n = 1, \cdots, N_t$. Given the aforementioned ending condition $\boldsymbol{\lambda}_{N_d+1} = \mathbf{0}$ which can be regarded as the initial conditional of the adjoint system equation that has to be solved backwards in time. The total variation of $\hat{J}$ can then easily be obtained and the total derivatives of J with respect to the model parameters $\boldsymbol{\beta}$ is given by

$$\frac{dJ}{d\boldsymbol{\beta}} = \frac{\partial J}{\partial \boldsymbol{\beta}} - \sum_{n=1}^{N_t} \boldsymbol{\lambda}_n^T \frac{\partial \mathbf{f}_n}{\partial \boldsymbol{\beta}} \tag{2.12}$$

We note that the dynamic system equation is nonlinear, whereas the adjoint equation Eq.2.5 is linear. The Newton-Raphson method which are generally used to iteratively solve the dynamic equation requires several iterations to converge at each time instance, whereas the adjoint model uses the sensitivity of the system, e.g., $\frac{\partial \mathbf{f}_{n+1}}{\partial \mathbf{x}_n}$ and $\frac{\partial \mathbf{f}_{n+1}}{\partial \boldsymbol{\beta}}$, at the converged solution of the Newton-Raphson method. Therefore it requires only one system equation at each time instance to be solved. If the derivatives of system equation $\mathbf{f}_n$ with respect to $\mathbf{x}_n$ and $\boldsymbol{\beta}$ are available for $n = 1, \cdots, N_t$, the adjoint method can easily be used to obtain the gradient of the objective function J, which can subsequently be used for the procedure of gradient-based history matching.

2.4. ENSEMBLE SMOOTHER WITH MULTIPLE DATA ASSIMILATION

In this subsection, a commonly used ensemble-based data assimilation method will be briefly described. The ensemble smoother with multiple data assimilation (ES-MDA) was introduced in [15] originally for the purpose of calibrating the reservoir model by conditioning to spatially dense seismic data. The main idea of ES-MDA is to divide the analysis step over a predefined number of partial updates, effectively creating an iterative scheme. At each analysis step, the measurement error covariance $\mathbf{R}$, e.g., a concatenation of $\mathbf{R}^m$ at all timesteps, is inflated to $\epsilon_k \mathbf{R}$, where ϵ_k is an inflation coefficient. The choice of inflation coefficients should satisfy the condition that their harmonic sum over N_a repeated implementations of the updating step is equal to one. i.e., $\sum_{k=1}^{N_a} \frac{1}{\epsilon_k}$=1. In this study, we choose a special inflation coefficient ϵ_k such that $\epsilon_k = N_a$ for k=1,$\cdots$,N_a [26].

The ES-MDA analysis equation can be presented as follows:

$$\boldsymbol{\beta}_i^k = \boldsymbol{\beta}_i^{k-1} + \mathbf{K}^k[\mathbf{d}_{obs} + \mathbf{e}_i^k - \mathbf{g}(\boldsymbol{\beta}_i^{k-1})], \quad i = 1, \cdots, N_e \tag{2.13}$$

where, we need randomly re-sample the observation errors $\mathbf{e}_i^k$ at each iteration step with the inflated covariance, i.e., $\mathbf{e}_i^k \sim G(\mathbf{0}, \epsilon_k \mathbf{R})$. Using a compact notation, the measurements operator $\mathbf{g}(\boldsymbol{\beta})$ is a concatenation of $\mathbf{h}^m(\mathbf{x}^m, \boldsymbol{\beta})$ for the each time step, e.g., $[\mathbf{h}^1(\mathbf{x}^1, \boldsymbol{\beta}), \mathbf{h}^2(\mathbf{x}^2, \boldsymbol{\beta}), \cdots, \mathbf{h}^{N_{obs}}(\mathbf{x}^{N_{obs}}, \boldsymbol{\beta})]$. The Kalman gain for ES-MDA is formulated as

$$\mathbf{K}^k = \mathbf{C}_{\boldsymbol{\beta}\mathbf{d}}^{k-1}(\mathbf{C}_{\mathbf{dd}}^{k-1} + \epsilon_k \mathbf{R})^{-1} \tag{2.14}$$

Thus, the ES-MDA algorithm is as follows:

1. Initialize the iteration number N_a and the inflation coefficients ϵ_k for $k = 1, \cdots, N_a$.
2. For $k = 1, \cdots, N_a$:
   - Run the ensemble forward models $\mathbf{g}(\boldsymbol{\beta}_0^k)$ from initial time, i.e., i=0.
   - Sample the observation errors using $\mathbf{e}_i^k \sim (\mathbf{0}, \epsilon_k \mathbf{R})$ for each ensemble member
   - Update each ensemble member using Eq. 2.13

As indicated in Eq.2.14, the implementation of ES-MDA typically requires the inversion of a matrix $\mathbf{C} \in R^{N_{obs} \times N_{obs}}$ (where $N_{obs} = N_d \times N_t$ represents the total number of measurements) given by

$$\mathbf{C} = \mathbf{C}_{\mathbf{dd}}^{k-1} + \epsilon_k \mathbf{R} \tag{2.15}$$

where, $\mathbf{C}_{\mathbf{dd}}^{k-1}$ is a symmetric positive semi-definite matrix. Eq. 2.15 indicates that matrix $\mathbf{C}$ will be real-symmetric positive-definite as well if we choose $\mathbf{R}$ positive-definite. However, the matrix $\mathbf{C}$ may be poorly conditioned [27] in the situations where a small ensemble size is used. To address this issue when implementing the ES-MDA algorithm, we generally need to compute a pseudo-inverse of matrix $\mathbf{C}$ by use of a truncated singular value decomposition (TSVD) [28]. We also should note that other schemes that address the case of large numbers of data do not explicitly construct $\mathbf{C}$ are provided by Evensen (2004) [29].

REFERENCES

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

[2] F. Gassmann, *Elastic waves through a packing of spheres,* Geophysics **16**, 673 (1951).

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

[4] P. K. Kitanidis, *Quasi-linear geostatistical theory for inversing,* Water resources research **31**, 2411 (1995).

[5] D. S. Oliver *et al.*, *Multiple realizations of the permeability field from well test data,* SPE Journal **1**, 145 (1996).

[6] C. Chen, G. Gao, R. Li, R. Cao, T. Chen, J. C. Vink, P. Gelderblom, *et al.*, *Global-search distributed-gauss-newton optimization method and its integration with the randomized-maximum-likelihood method for uncertainty quantification of reservoir performance,* SPE Journal **23**, 1 (2018).

[7] D. S. Oliver, *Metropolized randomized maximum likelihood for improved sampling from multimodal distributions,* SIAM/ASA Journal on Uncertainty Quantification **5**, 259 (2017).

[8] G. Evensen, *Sequential data assimilation with a nonlinear quasi-geostrophic model using monte carlo methods to forecast error statistics,* Journal of Geophysical Research: Oceans **99**, 10143 (1994).

[9] P. Sakov, D. S. Oliver, and L. Bertino, *An iterative enkf for strongly nonlinear systems,* Monthly Weather Review **140**, 1988 (2012).

[10] Y. Chen and D. S. Oliver, *Ensemble randomized maximum likelihood method as an iterative ensemble smoother,* Mathematical Geosciences **44**, 1 (2012).

[11] Y. Chen and D. S. Oliver, *Levenberg–marquardt forms of the iterative ensemble smoother for efficient history matching and uncertainty quantification,* Computational Geosciences **17**, 689 (2013).

[12] K. Thulin, G. Li, S. Aanonsen, A. C. Reynolds, *et al.*, *Estimation of initial fluid contacts by assimilation of production data with enkf,* in *SPE annual technical conference and exhibition* (Society of Petroleum Engineers, 2007).

[13] Y. Gu, D. S. Oliver, *et al.*, *An iterative ensemble kalman filter for multiphase fluid flow data assimilation,* Spe Journal **12**, 438 (2007).

[14] Y. Wang, G. Li, A. C. Reynolds, *et al.*, *Estimation of depths of fluid contacts by history matching using iterative ensemble-kalman smoothers,* SPE Journal **15**, 509 (2010).

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

[16] Y. Chen, D. S. Oliver, *et al.*, *History matching of the norne full-field model with an iterative ensemble smoother,* SPE Reservoir Evaluation & Engineering **17**, 244 (2014).

[17] X. Luo, A. S. Stordal, R. J. Lorentzen, and G. Naevdal, *Iterative ensemble smoother as an approximate solution to a regularized minimum-average-cost problem: theory and applications,* arXiv preprint arXiv:1505.01135 (2015).

[18] Æ. Geir, T. Mannseth, E. H. Vefring, *et al., Near-well reservoir monitoring through ensemble kalman filter,* in *SPE/DOE improved oil recovery symposium* (Society of Petroleum Engineers, 2002).

[19] S. I. Aanonsen, G. Nævdal, D. S. Oliver, A. C. Reynolds, B. Vallès, *et al., The ensemble kalman filter in reservoir engineering–a review,* Spe Journal **14**, 393 (2009).

[20] R. Courant and D. Hilbert, *Methods of Mathematical Physics: Partial Differential Equations* (John Wiley & Sons, 2008).

[21] R. Li, A. C. Reynolds, D. S. Oliver, *et al., History matching of three-phase flow production data,* in *SPE reservoir simulation symposium* (Society of Petroleum Engineers, 2001).

[22] S. Vlemmix, G. J. Joosten, R. Brouwer, J.-D. Jansen, *et al., Adjoint-based well trajectory optimization,* in *EUROPEC/EAGE conference and exhibition* (Society of Petroleum Engineers, 2009).

[23] G. Van Essen, E. Jimenez, J. K. Przybysz-jarnut, L. Horesh, S. G. Douma, P. van den Hoek, A. Conn, *et al., Adjoint-based history-matching of production and time-lapse seismic data,* in *SPE Europec/EAGE Annual Conference* (Society of Petroleum Engineers, 2012).

[24] E. Suwartadi, S. Krogstad, and B. Foss, *Adjoint-based surrogate optimization of oil reservoir water flooding,* Optimization and Engineering **16**, 441 (2015).

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

[26] A. A. Emerick, *Analysis of the performance of ensemble-based assimilation of production and seismic data,* Journal of Petroleum Science and Engineering **139**, 219 (2016).

[27] G. Evensen, *Data assimilation: the ensemble Kalman filter* (Springer Science & Business Media, 2009).

[28] A. A. Emerick and A. C. Reynolds, *History matching time-lapse seismic data using the ensemble kalman filter with multiple data assimilations,* Computational Geosciences **16**, 639 (2012).

[29] G. Evensen, *Sampling strategies and square root analysis schemes for the enkf,* Ocean dynamics **54**, 539 (2004).

3

NON-INTRUSIVE PROJECTION-BASED SUBDOMAIN POD-TPWL

This chapter presents a non-intrusive subdomain POD-TPWL algorithm for reservoir history matching through integrating domain decomposition (DD), proper orthogonal decomposition (POD), radial basis function (RBF) interpolation and the trajectory piecewise linearization (TPWL). It is an efficient approach for model reduction and linearization of general non-linear time-dependent dynamical systems without the need of access to the legacy source code. In the subdomain POD-TPWL algorithm, firstly, a sequence of snapshots over the entire computational domain are saved and then partitioned into subdomains. From the local sequence of snapshots over each subdomain, a number of local basis vectors is formed using POD, and then the RBF interpolation is used to estimate the derivative matrices for each subdomain. Finally, those derivative matrices are substituted into a POD-TPWL algorithm to form a reduced-order linear model in each subdomain. This reduced-order linear model allows an easy implementation of the adjoint and results in an efficient adjoint-based parameter estimation procedure. Comparisons with the classic finite-difference based history matching show that our proposed subdomain POD-TPWL approach obtains a comparable accuracy at a much lower computational cost.

Parts of this chapter have been published in Computational Geosciences. **Xiao, C.**, Leeuwenburgh, O., Lin, H.X. and Heemink, A., 2019. *Non-intrusive subdomain POD-TPWL for reservoir history matching.* Computational Geosciences, 23(3), pp.537-565.

3.1. INTRODUCTION

History matching problems can be efficiently solved using a gradient-based minimization algorithm [1]. In general, significant effort is required to obtain and maintain a correct implementation of the adjoint model for complex nonlinear simulation models. Such implementations are generally intrusive, that is, they require access to the model code, which may not always be possible. Many efforts have been taken to make the implementation of the adjoint model more feasible. One way is to replace the original complex model with a surrogate that the construction of the adjoint model becomes easier. Courtier et al. [2] proposed an incremental approach by replacing a high resolution nonlinear model with an approximated linear model. Liu et al. [3, 4], developed an ensemble-based four-dimensional variational (4DEnVar) data assimilation scheme where the approximated linear model is constructed using an ensemble of model forecasts. Recently, to extend the ensemble-based tangent linear model (TLM) to more realistic applications, Frolov and Bishop et al. [5, 6] incorporated a local ensemble tangent linear model (LETLM) into 4D-Var scheme. The LETLM has the ability to capture localized physical features of dynamic models with relatively small ensemble size. However, it will become intractable for high-dimensional systems. Proper Orthogonal Decomposition (POD), a model order reduction method, is a possible approach to decrease the dimensionality of the original model. The POD approach has been applied to various disciplines, including reservoir model simulations [7, 8], and has in some cases shown significant speed up [9] .

The combination of model linearization and model reduction techniques has the potential to further ease the implementation of adjoint models for high-dimensional complex dynamic systems. Vermeulen et al [10] combined a non-intrusive perturbation-based linearization method and POD to build a reduced-order linear approximation of the original high-dimensional non-linear model. The adjoint of this reduced-order linear model can be easily constructed and therefore the minimization of the objective function can be handled efficiently. Altaf et al. [11] and Kaleta et al. [12] applied this method to a coastal engineering and reservoir history matching problem, respectively.

Alternatively, the Trajectory Piecewise Linearization (TPWL) can be classified as a model-intrusive linearization method. In TPWL, a number of full-order 'training' runs is first simulated, and then a linear model is generated through first-order expansion around the 'closest' training trajectories. In reservoir engineering, Cardoso et al. [13] were the first to integrate POD and TPWL methods and applied this strategy to oil production optimization. He et al applied the POD-TPWL method to both reservoir history matching and production optimization [14, 15]. These studies suggested that POD-TPWL has the potential to significantly reduce the runtime for subsurface flow problems [16]. A drawback, however, is that the POD-TPWL method requires access to derivative matrices used internally by the numerical solver, and therefore cannot be used with most commercial simulators [14, 17]. And while non-intrusive reduced-order linear model construction is possible [10–12], the required derivative information is estimated using a global perturbation-based finite difference method, which needs a large number of full-order simulations and is therefore computationally demanding. Furthermore, the global perturbation also hinders the extension of this method to large-scale reservoir history matching which requires retaining many POD patterns [12]. In order to avoid model intrusion and numerous full-order simulations, we propose to incorporate domain decomposition (DD) and radial basis function (RBF) interpolation into POD-TPWL to develop a new non-intrusive subdomain POD-TPWL algorithm.

RBF interpolation is mainly used to construct surrogate models [18], and has been applied e.g. to reservoir engineering and fluid dynamics [19–21]. Recently, Bruyelle et al. [22] applied the neural network-based RBF to obtain the first-order and second-order derivative information of a reservoir model and estimate the gradients and Hessian matrix for reservoir production optimization. The accuracy of RBF-based gradient approximation is determined by the sampling strategies of the interpolation data [22]. For high dimensional problems, the classical global RBF interpolation algorithm requires a large number of interpolation data to capture the flow dynamic as much as possible [23]. Moreover, the global RBF algorithm can cause some spurious long-distance correlations, which implies the possibilities to avoid some redundant interpolation data. This motivates us to develop a subdomain RBF interpolation technique for reservoir models where the domain decomposition (DD) technique potentially allows us to apply the methodology to large-scale problems. Different local RBF interpolation schemes are considered based on the details of local flow dynamics in each subdomain. The domain decomposition technique first introduced in the work of Przemieniecki [24] has been applied to various fields [25]. Lucia et al. [26] first introduced the DD method to model-order reduction for accurately tracking a moving strong shock wave. Subsequently, the DD method has also been applied to non-linear model reduction problems [27–29].

We presents a new non-intrusive subdomain POD-TPWL algorithm for subsurface flow. The key idea behind this subdomain POD-TPWL is to integrate the DD method and RBF interpolation into a model linearization procedure based on POD-TPWL. In this study, we construct the reduced-order linear model (TLM) piecewise for each subdomain instead of each state variable. After constructing the reduced-order linear model using the subdomain POD-TPWL algorithm, because of the linearity in the reduced-order subspace, the implementation of adjoint model is easy and, thus, it is convenient to incorporate this reduced-order linear model into a gradient-based reservoir history matching procedure. The runtime speedup and the accuracy of the new history matching algorithm have been assessed.

This chapter is organized as follows. The classical adjoint-based solution approach are described in Section 2. Section 3 contains the mathematical background of the conventional POD-TPWL. Section 4 gives the mathematical descriptions of domain decomposition (DD) and radial basis function (RBF) interpolation, which are used to develop the non-intrusive subdomain POD-TPWL. In addition, a workflow for combining subdomain POD-TPWL with an adjoint-based history matching algorithm is described. Section 5 discusses and evaluates an application of the new history matching workflow to two numerical 'twin' experiments involving synthetic reservoir models.

3.2. PROBLEM DESCRIPTION

In this chapter, we mainly focus on an efficient gradient-based optimization method for iteratively minimizing an objective function that quantifies the misfit between simulated and observed data, which has been defined in Chapter 2. The key step of a gradient-based minimization algorithm is to determine the gradient of the cost function with respect to the parameters by using an adjoint model. In the adjoint approach, a modified function $\hat{J}$ is obtained by adjoining the model equation to the objective function **J** which

has been defined in Chapter 2.

$$\hat{J}(\mathbf{x}^1,\cdots,\mathbf{x}^n,\cdots,\mathbf{x}^N,\boldsymbol{\beta}) = J(\mathbf{x}^1,\cdots,\mathbf{x}^n,\cdots,\mathbf{x}^N,\boldsymbol{\beta}) + \sum_{n=1}^{N}[\boldsymbol{\lambda}^n]^T[\mathbf{x}^n - \mathbf{f}^n(\mathbf{x}^{n-1},\boldsymbol{\beta})] \tag{3.1}$$

And then, the gradient of the cost function is formulated after introducing the adjoint model as follows (more details about the mathematical derivation can be found in Chapter.2,

$$[\frac{dJ}{d\boldsymbol{\beta}}]^T = \mathbf{R}_p{}^{-1}(\boldsymbol{\beta}-\boldsymbol{\beta}_p) - \sum_{n=1}^{N}[\boldsymbol{\lambda}^n]^T\frac{\partial \mathbf{f}^n}{\partial \boldsymbol{\beta}} - \sum_{m=1}^{N_0}[\frac{\partial \mathbf{h}^m(\mathbf{x}^m,\boldsymbol{\beta})}{\partial \boldsymbol{\beta}}]^T\mathbf{R}_m{}^{-1}(\mathbf{d}_o^m - \mathbf{h}^m(\mathbf{x}^m,\boldsymbol{\beta})) \tag{3.2}$$

where the adjoint model in terms of the Lagrange multipliers $\boldsymbol{\lambda}^n$ is given by

$$\boldsymbol{\lambda}^n = [\frac{\partial \mathbf{f}^{n+1}}{\partial \mathbf{x}^n}]\boldsymbol{\lambda}^{n+1} + [\frac{\partial \mathbf{h}^n(\mathbf{x}^n,\boldsymbol{\beta})}{\partial \mathbf{x}^n}]^T\mathbf{R}_n{}^{-1}(\mathbf{d}_o^n - \mathbf{h}^n(\mathbf{x}^n,\boldsymbol{\beta})) \tag{3.3}$$

for $n = N,\cdots 1$ with an ending condition $\boldsymbol{\lambda}^{N+1} = 0$. This adjoint approach has a high computational efficiency because just one forward simulation and one backward simulation are required to compute the gradient, independent on the size of the variable vector. It should be pointed out that four derivative terms, e.g, $\frac{\partial \mathbf{h}^m(\mathbf{x}^m,\boldsymbol{\beta})}{\partial \boldsymbol{\beta}}$, $\frac{\partial \mathbf{h}^n(\mathbf{x}^n,\boldsymbol{\beta})}{\partial \mathbf{x}^n}$, $\frac{\partial \mathbf{f}^n}{\partial \boldsymbol{\beta}}$ and $\frac{\partial \mathbf{f}^{n+1}}{\partial \mathbf{x}^n}$, are required in the adjoint method. We will give detailed descriptions of how to efficiently obtain these four terms using our proposed subdomain POD-TPWL algorithm in the following sections.

3.3. POD-TPWL ALGORITHM

In the TPWL scheme, one or more full order "training" runs using a set of perturbed parameters are simulated first. The states and the derivative information at each time step from these runs are used to construct the TPWL surrogate. Given the state $\mathbf{x}^n$ and parameters $\boldsymbol{\beta}$, the state $\mathbf{x}^{n+1}$ is approximated as a first-order expansion around the training solution $(\mathbf{x}_{tr}^{n+1},\mathbf{x}_{tr}^n,\boldsymbol{\beta}_{tr})$ as follows,

$$\mathbf{x}^{n+1} \approx \mathbf{x}_{tr}^{n+1} + \mathbf{E}^{n+1}(\mathbf{x}^n - \mathbf{x}_{tr}^n) + \mathbf{G}^{n+1}(\boldsymbol{\beta}-\boldsymbol{\beta}_{tr}) \tag{3.4}$$

where

$$\mathbf{E}^{n+1} = \frac{\partial \mathbf{f}^{n+1}}{\partial \mathbf{x}_{tr}^n}, \quad \mathbf{G}^{n+1} = \frac{\partial \mathbf{f}^{n+1}}{\partial \boldsymbol{\beta}_{tr}} \tag{3.5}$$

The training solution $(\mathbf{x}_{tr}^{n+1},\mathbf{x}_{tr}^n,\boldsymbol{\beta}_{tr})$ is chosen to be as 'close' as possible to the state $\mathbf{x}^n$. A detailed description of a criterion for closeness can be found in [30]. The matrices $\mathbf{E}^{n+1} \in R^{2N_g\times 2N_g}$ and $\mathbf{G}^{n+1} \in R^{2N_g\times N_g}$ represent the derivative of the dynamic model as Eq.3.1 at timestep n+1 with respect to states $\mathbf{x}_{tr}^n$ and parameters $\boldsymbol{\beta}_{tr}$ respectively. Eq.3.5 is, however, still in a high-dimensional space, e.g, $\mathbf{x}^{n+1} \in R^{2N_g}$, and $\boldsymbol{\beta} \in R^{N_g}$, which motivates the development of the POD-TPWL algorithm [30].

POD provides as means to project the high-dimensional states into an optimal lower-dimensional subspace. The basis of this subspace is obtained by performing a Singular Value Decomposition (SVD) of a snapshot matrix containing the solution states at selected time steps (snapshots) computed from training simulations. The state vector $\mathbf{x}$ can then be

represented in terms of the product of a coefficient vector $\boldsymbol{\psi}$ and a matrix of basis vectors $\boldsymbol{\Phi}$

$$\mathbf{x} = \boldsymbol{\Phi}\boldsymbol{\psi} \tag{3.6}$$

Let $\boldsymbol{\Phi}_p$ and $\boldsymbol{\Phi}_s$ represent separate matrices of basis vectors for pressure and saturation respectively [31]. In general there is no need to contain all columns of the left singular matrix in $\boldsymbol{\Phi}_p$ and $\boldsymbol{\Phi}_s$ and a reduced state vector representations are obtained by selecting only the first columns according to an energy criterion; see e.g.,[30]. To normalize the reduced state vector, the columns of $\boldsymbol{\Phi}_p$ are determined by multiplying the left singular matrix $\mathbf{U}_p$ with the singular value matrix $\boldsymbol{\Sigma}_p$ (and similarly for saturation), i.e.

$$\boldsymbol{\Phi}_p = \mathbf{U}_p\boldsymbol{\Sigma}_p, \quad \boldsymbol{\Phi}_s = \mathbf{U}_s\boldsymbol{\Sigma}_s \tag{3.7}$$

These two matrix are then assembled into a single basis matrix $\boldsymbol{\Phi}$ as follows:

$$\mathbf{x} = \boldsymbol{\Phi}\boldsymbol{\psi} = \begin{bmatrix} \boldsymbol{\Phi}_p & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Phi}_s \end{bmatrix} \boldsymbol{\psi} = \begin{bmatrix} \boldsymbol{\Phi}_p & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Phi}_s \end{bmatrix} \begin{bmatrix} \boldsymbol{\psi}_p \\ \boldsymbol{\psi}_s \end{bmatrix} \tag{3.8}$$

We also use Karhunen-Loeve expansion (KLE) or principal component analysis (PCA) to parameterize the parameter space. KLE reduces the dimension of the parameter vector by projecting the high-dimensional parameter into an optimal lower-dimensional subspace [32]. The basis of this subspace is obtained by performing an eigenvalue decomposition of the background parameter covariance matrix $\mathbf{R}_b$. If this covariance matrix is not accessible the basis can alternatively be obtained from an SVD of a matrix holding an ensemble of prior parameter realizations with ensemble mean $\boldsymbol{\beta}_b$. Including normalization of the reduced parameter vector, a random parameter vector sample $\boldsymbol{\beta}$ is generated as follows,

$$\boldsymbol{\beta} = \boldsymbol{\beta}_b + \boldsymbol{\Phi}_{\boldsymbol{\beta}}\boldsymbol{\xi}, \quad \text{with} \quad \boldsymbol{\Phi}_{\boldsymbol{\beta}} = \mathbf{U}_{\boldsymbol{\beta}}\boldsymbol{\Sigma}_{\beta} \tag{3.9}$$

where $\boldsymbol{\Phi}_{\boldsymbol{\beta}}$ denotes the matrix of parameter basis vectors, $\mathbf{U}_{\boldsymbol{\beta}}$ and $\boldsymbol{\Sigma}_{\boldsymbol{\beta}}$ are the matrix of left singular vectors and singular value matrix respectively, and $\boldsymbol{\xi}$ denotes a vector with independent Gaussian random variables with zeros mean and unit variance. A reduced parameter space representation is obtained by selecting only the first several columns of $\boldsymbol{\Phi}_{\boldsymbol{\beta}}$. The number of retained columns for basis matrix (denoted as l_p and l_s for pressure and saturation, $l_{\boldsymbol{\beta}}$ for parameter, respectively) is determined through an energy criterion [30]. We take $\boldsymbol{\Phi}_p$ as an example. We first compute the total energy E_t, which is defined as $E_t = \sum_{i=1}^{L} v_i^2$, where v_i denotes the i-th diagonal element of singular value matrix $\boldsymbol{\Sigma}_p$. The energy associated with the first l_p singular vectors is given by $E_{l_p} = \sum_{i=1}^{l_p} {v_i}^2$. Then the smallest l_p is determined such that E_{l_p}exceeds a specific fraction of E_t, e.g., 95%. The same procedure is applied to determine l_s and l_β.

Substituting Eq.3.8 and Eq.3.9 into Eq.3.4, we obtain the following POD-TPWL formula,

$$\boldsymbol{\psi}^{n+1} \approx \boldsymbol{\psi}_{tr}^{n+1} + \mathbf{E}_{\boldsymbol{\psi}}^{n+1}(\boldsymbol{\psi}^n - \boldsymbol{\psi}_{tr}^n) + \mathbf{G}_{\boldsymbol{\xi}}^{n+1}(\boldsymbol{\xi} - \boldsymbol{\xi}_{tr}) \tag{3.10}$$

$$\mathbf{E}_{\boldsymbol{\psi}}^{n+1} = \boldsymbol{\Phi}^T \frac{\partial \mathbf{f}^{n+1}}{\partial \mathbf{x}_{tr}^n}\boldsymbol{\Phi}, \quad \mathbf{G}_{\boldsymbol{\xi}}^{n+1} = \boldsymbol{\Phi}^T \frac{\partial \mathbf{f}^{n+1}}{\partial \boldsymbol{\beta}_{tr}}\boldsymbol{\Phi}_{\beta} \tag{3.11}$$

Similarly, the well model is also linearized around a close training solution $(\boldsymbol{\psi}_{tr}^{n+1}, \boldsymbol{\xi}_{tr})$ in the reduced space as follows,

$$\mathbf{y}^{m+1} \approx \mathbf{y}_{tr}^{m+1} + \mathbf{A}_{\boldsymbol{\psi}}^{m+1}(\boldsymbol{\psi}^{m+1} - \boldsymbol{\psi}_{tr}^{m+1}) + \mathbf{B}_{\boldsymbol{\xi}}^{m+1}(\boldsymbol{\xi} - \boldsymbol{\xi}_{tr}) \tag{3.12}$$

$$\mathbf{A}_{\boldsymbol{\psi}}^{m+1} = \frac{\partial \mathbf{h}^{m+1}}{\partial \mathbf{x}_{tr}^{m+1}} \boldsymbol{\Phi}, \quad \mathbf{B}_{\boldsymbol{\xi}}^{m+1} = \frac{\partial \mathbf{h}^{m+1}}{\partial \boldsymbol{\beta}_{tr}} \boldsymbol{\Phi}_{\boldsymbol{\beta}} \tag{3.13}$$

Eq.3.10 and Eq.3.12 represent the POD-TPWL system for reservoir model and well model in the reduced-order space, respectively. In general the traditional POD-TPWL method modifies the source code to output all derivative matrices [30]. In this paper, radial basis function interpolation is used to approximately estimate these derivative matrices. These derivative matrices then are substituted into POD-TPWL algorithm to form a subdomain reduced-order linear model.

3

3.4. ADJOINT-BASED OPTIMIZATION USING REDUCED-ORDER MODELLING

This section describes the mathematical background of domain decomposition (DD), and radial basis function (RBF) interpolation, which are used to construct a non-intrusive subdomain reduced-order linear model. And then a procedure to incorporate this reduced-order linear model into an adjoint-based history matching is given in the last subsection.

3.4.1. DOMAIN DECOMPOSITION METHOD

We denote a 2D or 3D computational domain as Ω. The entire domain Ω is assumed to be decomposed into S non-overlapping subdomains Ω^d , $d \in \{1, 2, \cdots, S\}$ (such that $\Omega = \bigcup_{d=1}^{S} \Omega^d$ and $\Omega^i \cap \Omega^j = 0$ for $i \neq j$). Each subdomain has local unknowns, e.g., local pressure and saturation variables. In each subdomain Ω^d, the generated snapshots within that subdomain are used to construct a set of local POD basis functions $\boldsymbol{\phi}^d$ and the corresponding POD coefficients $\boldsymbol{\psi}^{d,n+1}$ at the timestep n+1 as described in the previous section. The dynamic model is replaced by an interpolation model relating neighboring subdomain POD coefficients at the current and previous time step.

$$\boldsymbol{\psi}^{d,n+1} = \pounds^{d,n+1}(\boldsymbol{\psi}^{d,n}, \boldsymbol{\psi}^{sd,n+1}, \boldsymbol{\xi}) \tag{3.14}$$

Similarly, the dynamic well model is replaced by a second interpolation model expressed in terms of the local subdomain POD coefficient $\boldsymbol{\psi}^{d,m+1}$ and $\boldsymbol{\xi}$

$$\mathbf{y}^{d,m+1} = \hbar^{d,n+1}(\boldsymbol{\psi}^{d,m+1}, \boldsymbol{\xi}) \tag{3.15}$$

where, vector $\boldsymbol{\psi}^{d,n}$ denotes the set of POD coefficients at the time level n for the subdomain Ω^d, $\boldsymbol{\psi}^{sd,n+1}$ denotes the set of POD coefficients at time level n+1 for the surrounding subdomains Ω^{sd}. In a 2-D case, the number of surrounding subdomains associated to subdomain Ω^d is between 2 and 4. Fig.3.1 shows a maximum of four surrounding subdomains connected with the subdomain Ω^5, three surrounding subdomains connected with the subdomain $\Omega^2, \Omega^4, \Omega^6, \Omega^8$, and two surrounding subdomains connected with the subdomain $\Omega^1, \Omega^3, \Omega^7, \Omega^9$.

We propose to use RBF interpolation to obtain the derivative matrices that are required by the POD-TPWL. In addition, domain decomposition has the abilities to efficiently capture localized physical features [23], and therefore has the potential to improve the derivative estimation by local low-dimensional RBF interpolation which will be described in the next subsections.

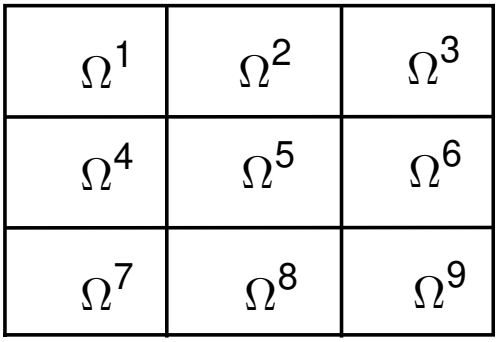


Figure 3.1: Illustration of domain decomposition in a 2-D case

3.4.2. RADIAL BASIS FUNCTION INTERPOLATION

RBF interpolation can be classified as a data-driven interpolation method [18–20]. High-dimensional interpolation needs a large number of data to obtain a satisfactory accuracy, a phenomenon often referred to as the "curse of dimensionality". To remedy this difficulty, DD approximates the global domain by the sum of the local subdomains, and therefore can be applied to form a locally low-dimensional RBF interpolation.

For subdomain Ω^d, let $\pounds^{d,n+1}(\boldsymbol{\psi}^{d,n}, \boldsymbol{\psi}^{sd,n+1}, \boldsymbol{\xi})$ denote a RBF interpolation function for the POD coefficient $\boldsymbol{\psi}^{d,n+1}$ at the time level n+1. The RBF interpolation function is a linear combination of M radial basis functions in the form of,

$$\pounds^{d,n+1}(\boldsymbol{\psi}^{d,n}, \boldsymbol{\psi}^{sd,n+1}, \boldsymbol{\xi}) = \sum_{j=1}^{M} \boldsymbol{\omega}_j^{d,n+1} \times \theta(||(\boldsymbol{\psi}^{d,n}, \boldsymbol{\psi}^{sd,n+1}, \boldsymbol{\xi}) - (\boldsymbol{\psi}_j^{d,n}, \boldsymbol{\psi}_j^{sd,n+1}, \boldsymbol{\xi}_j)||) \tag{3.16}$$

where, $||(\boldsymbol{\psi}^{d,n}, \boldsymbol{\psi}^{sd,n+1}, \boldsymbol{\xi}) - (\boldsymbol{\psi}_j^{d,n}, \boldsymbol{\psi}_j^{sd,n+1}, \boldsymbol{\xi}_j)||$ is a scalar distance using L^2 norm. $\boldsymbol{\omega}^{d,n+1}$ is a weighting coefficient vector of size M. θ is a set of specific radial basis functions.

The specific coefficient $\boldsymbol{\omega}_j^{d,n+1}$ is determined so as to ensure that the interpolation function values $\pounds^{d,n+1}$ at the training data points $(\boldsymbol{\psi}_j^{d,n}, \boldsymbol{\psi}_j^{sd,n+1}, \boldsymbol{\xi}_j)$, match the given data $\boldsymbol{\psi}_j^{d,n+1}$ exactly. This can be expressed by,

$$\mathbf{D}^{d,n+1}\boldsymbol{\omega}^{d,n+1} = \mathbf{Z}^{d,n+1} \tag{3.17}$$

where

$$\mathbf{D}^{d,n+1} = \begin{bmatrix} \theta(l^{n+1}(1,1)) & \dots & \theta(l^{n+1}(1,M)) \\ . & \theta(l^{n+1}(i,j) & . \\ \theta(l^{n+1}(M,1)) & \dots & \theta(l^{n+1}(M,M)) \end{bmatrix}$$

$$l^{n+1}(i,j) = ||(\boldsymbol{\psi}_i^{d,n}, \boldsymbol{\psi}_i^{sd,n+1}, \boldsymbol{\xi}_i) - (\boldsymbol{\psi}_j^{d,n}, \boldsymbol{\psi}_j^{sd,n+1}, \boldsymbol{\xi}_j)||,$$

$$i = 1, \cdots, M; j = 1, \cdots, M \tag{3.18}$$

$$\boldsymbol{\omega}^{d,n+1} = [\boldsymbol{\omega}_1^{d,n+1}, \boldsymbol{\omega}_2^{d,n+1}, \cdots, \boldsymbol{\omega}_M^{d,n+1}]^T \tag{3.19}$$

$$\mathbf{Z}^{d,n+1} = [\boldsymbol{\psi}_1^{d,n+1}, \boldsymbol{\psi}_2^{d,n+1}, \cdots, \boldsymbol{\psi}_M^{d,n+1}]^T \tag{3.20}$$

The weighting coefficients are determined by solving the linear system Eq.3.17. We chose Multi-Quadratic RBF in our case. l represents a Euclidean distance. ϵ denotes the shape parameters, which can be optimized using greedy algorithm [21]. Clearly, a Multi-Quadratic RBF monotonically decreases with Euclidean distance. Multi-Quadratic RBF is local and is the msot commonly used across several applications. Other type of RBF can be chosen with more specific purpose [33]. A list of well-known RBF are provided in Table 3.1.

After the construction of RBF interpolation, we can analytically estimate the derivative at the 'closest' training data point, e.g., $(\boldsymbol{\psi}_i^{d,n}, \boldsymbol{\psi}_i^{sd,n+1}, \boldsymbol{\xi}_i)$, by differentiating the RBF as follows,

$$\frac{\partial \pounds^{d,n+1}}{\partial \boldsymbol{\xi}}|_{\boldsymbol{\xi}=\boldsymbol{\xi}_i} = \sum_{j=1}^{M} \boldsymbol{\omega}_j^{d,n+1} \times \frac{\partial \theta(||(\boldsymbol{\psi}^{d,n}, \boldsymbol{\psi}^{sd,n+1}, \boldsymbol{\xi}) - (\boldsymbol{\psi}_j^{d,n}, \boldsymbol{\psi}_j^{sd,n+1}, \boldsymbol{\xi}_j)||)}{\partial \boldsymbol{\xi}}|_{\boldsymbol{\xi}=\boldsymbol{\xi}_i} \tag{3.21}$$

$$\frac{\partial \pounds^{d,n+1}}{\partial \boldsymbol{\psi}^{sd,n+1}}|_{\boldsymbol{\psi}^{sd,n+1}=\boldsymbol{\psi}_i^{sd,n+1}} = \sum_{j=1}^{M} \boldsymbol{\omega}_j^{d,n+1} \times \frac{\partial \theta(||(\boldsymbol{\psi}^{d,n}, \boldsymbol{\psi}^{sd,n+1}, \boldsymbol{\xi}) - (\boldsymbol{\psi}_j^{d,n}, \boldsymbol{\psi}_j^{sd,n+1}, \boldsymbol{\xi}_j)||)}{\partial \boldsymbol{\psi}^{sd,n+1}}|_{\boldsymbol{\psi}^{sd,n+1}=\boldsymbol{\psi}_i^{sd,n+1}} \tag{3.22}$$

$$\frac{\partial \pounds^{d,n+1}}{\partial \boldsymbol{\psi}^{d,n}}|_{\boldsymbol{\psi}^{d,n}=\boldsymbol{\psi}_i^{d,n}} = \sum_{j=1}^{M} \boldsymbol{\omega}_j^{d,n+1} \times \frac{\partial \theta(||(\boldsymbol{\psi}^{d,n}, \boldsymbol{\psi}^{sd,n+1}, \boldsymbol{\xi}) - (\boldsymbol{\psi}_j^{d,n}, \boldsymbol{\psi}_j^{sd,n+1}, \boldsymbol{\xi}_j)||)}{\partial \boldsymbol{\psi}^{d,n}}|_{\boldsymbol{\psi}^{d,n}=\boldsymbol{\psi}_i^{d,n}} \tag{3.23}$$

Table 3.1: Some well-known radial basis functions

Functions	**Definition**
Gaussian	$\theta(l) = e^{-(\frac{l}{\epsilon})^2}$
Linear Spline	$\theta(l) = l$
Multi-Quadratic	$\theta(l) = \sqrt{l^2+\epsilon^2}$
Inverse Caddric	$\theta(l) = \frac{1}{l^2+\epsilon^2}$
Cubic Spline	$\theta(l) = l^3$
Thin Plate Spline	$\theta(l) = l^2 log l$
Inverse Multistory	$\theta(l) = \frac{1}{\sqrt{l^2+\epsilon^2}}$

Similarly, the approximation Eq.3.15 also can be constructed using RBF interpolation method as follows,

$$\mathbf{y}^{d,m+1} \approx \hbar^{d,n+1}(\boldsymbol{\psi}^{d,m+1}, \boldsymbol{\xi}) = \sum_{j=1}^{M} \boldsymbol{\varepsilon}_j^{d,m+1} \times \theta(\|(\boldsymbol{\psi}^{d,m+1}, \boldsymbol{\xi}) - (\boldsymbol{\psi}_j^{d,m+1}, \boldsymbol{\xi}_j)\|) \tag{3.24}$$

The derivative at the training data by differentiating the RBF function Eq.3.14 with respect to $(\boldsymbol{\psi}_i^{d,m+1}, \boldsymbol{\xi}_i)$ can be given by

$$\frac{\partial \hbar^{d,n+1}}{\partial \boldsymbol{\xi}}|_{\boldsymbol{\xi}=\boldsymbol{\xi}_i} = \sum_{j=1}^{M} \boldsymbol{\varepsilon}_j^{d,m+1} \times \frac{\partial \theta(\|(\boldsymbol{\psi}^{d,m+1}, \boldsymbol{\xi}) - (\boldsymbol{\psi}_j^{d,m+1}, \boldsymbol{\xi}_j)\|)}{\partial \boldsymbol{\xi}}|_{\boldsymbol{\xi}=\boldsymbol{\xi}_i} \tag{3.25}$$

$$\frac{\partial \hbar^{d,n+1}}{\partial \boldsymbol{\psi}^{d,m+1}}|_{\boldsymbol{\psi}^{d,m+1}=\boldsymbol{\psi}_i^{d,m+1}} = \sum_{j=1}^{M} \boldsymbol{\varepsilon}_j^{d,m+1} \times \frac{\partial \theta(\|(\boldsymbol{\psi}^{d,m+1},\boldsymbol{\xi}) - (\boldsymbol{\psi}_j^{d,m+1},\boldsymbol{\xi}_j)\|)}{\partial \boldsymbol{\xi}}|_{\boldsymbol{\psi}^{d,m+1}=\boldsymbol{\psi}_i^{d,m+1}} \tag{3.26}$$

where, $\boldsymbol{\varepsilon}^{d,m+1}$ is a weighting coefficient vector of size M (number of training data sets), and these matrices are reconstructed for each time step correspondingly.

3.4.3. SUBDOMAIN POD-TPWL ALGORITHM

By considering the dynamic interaction between neighboring subdomains as in Eq.3.14, the coefficients $\boldsymbol{\psi}^{d,n+1}$ for the subdomain Ω^d can be obtained as follows,

$$\begin{aligned}\boldsymbol{\psi}^{d,n+1} \approx\ & \boldsymbol{\psi}_{tr}^{d,n+1} + \mathbf{E}_{\boldsymbol{\psi}_{tr}}^{d,n+1}(\boldsymbol{\psi}^{d,n} - \boldsymbol{\psi}_{tr}^{d,n}) \\ & + \mathbf{E}_{\boldsymbol{\psi}_{tr}}^{sd,n+1}(\boldsymbol{\psi}^{sd,n+1} - \boldsymbol{\psi}_{tr}^{sd,n+1}) + \mathbf{G}_{\boldsymbol{\xi}_{tr}}^{n+1}(\boldsymbol{\xi} - \boldsymbol{\xi}_{tr})\end{aligned} \tag{3.27}$$

Coupling domain decomposition and radial basis function interpolation, the derivative matrices required by POD-TPWL for the subdomain Ω^d are estimated as follows

$$\begin{aligned}\mathbf{E}_{\boldsymbol{\psi}_{tr}}^{d,n+1} &\approx \frac{\partial \pounds^{d,n+1}}{\partial \boldsymbol{\psi}^{d,n}}|_{\boldsymbol{\psi}^{d,n}=\boldsymbol{\psi}_{tr}^{d,n}} \\ \mathbf{E}_{\boldsymbol{\psi}_{tr}}^{sd,n+1} &\approx \frac{\partial \pounds^{d,n+1}}{\partial \boldsymbol{\psi}^{sd,n+1}}|_{\boldsymbol{\psi}^{sd,n+1}=\boldsymbol{\psi}_{tr}^{sd,n+1}} \\ \mathbf{G}_{\boldsymbol{\xi}}^{n+1} &\approx \frac{\partial \pounds^{d,n+1}}{\partial \boldsymbol{\xi}}|_{\boldsymbol{\xi}=\boldsymbol{\xi}_{tr}}\end{aligned} \tag{3.28}$$

where subindex *tr* refers to the nearest training point determined in terms of of vector $(\boldsymbol{\psi}^{d,n}, \boldsymbol{\psi}^{sd,n+1}, \boldsymbol{\xi})$.

Similarly, substituting Eq.3.24-Eq.3.25 into Eq.3.12, the simulated measurements $\mathbf{y}^{d,m+1}$ of the subdomain Ω^d are reformulated as

$$\mathbf{y}^{d,m+1} \approx \mathbf{y}_{tr}^{d,m+1} + \mathbf{A}_{\boldsymbol{\psi}_{tr}}^{d,m+1}(\boldsymbol{\psi}^{d,m+1} - \boldsymbol{\psi}_{tr}^{d,m+1}) + \mathbf{B}_{\boldsymbol{\xi}_{tr}}^{m+1}(\boldsymbol{\xi} - \boldsymbol{\xi}_{tr}) \tag{3.29}$$

$$\mathbf{A}_{\boldsymbol{\psi}}^{d,m+1} \approx \frac{\partial \hbar^{d,m+1}}{\partial \boldsymbol{\psi}^{d,m+1}}|_{\boldsymbol{\psi}^{d,m+1}=\boldsymbol{\psi}_{tr}^{d,m+1}}, \mathbf{B}_{\boldsymbol{\xi}_{tr}}^{m+1} \approx \frac{\partial \hbar^{d,m+1}}{\partial \boldsymbol{\xi}}|_{\boldsymbol{\xi}=\boldsymbol{\xi}_{tr}} \tag{3.30}$$

The implementation of POD-TPWL is local in each subdomain, which has the potential to capture features dominated by local dynamics better than global approximations. Therefore, we could refer to the subdomain POD-TPWL algorithm. The subdomain POD-TPWL consists of an offline stage and an online stage. (1) During the offline stage, we construct a set of local RBF and estimate the derivative information for each subdomain. Firstly, the solutions of the full-order model are saved as a sequence of snapshots over the whole computational domain and then partitioned into subdomains. From the local sequence of snapshots over each subdomain, a number of local basis vectors is formed using POD. Unlike the traditional practices in which RBF is used to construct a set of surrogates for each subdomain, we use RBF to estimate the derivative matrices for each subdomain. Finally, those estimated derivative matrices are substituted into POD-TPWL algorithm to form a reduced-order linear model in each subdomain. (2) The online stage consists of the time evolution of the dynamic state of the reduced model by iterative implementation

and solution of the subdomain POD-TPWL equations. Referring to Eq.3.27, we represent the dynamic interactions among neighboring subdomains using an implicit formula. the variables of one subdomain at current time level can be linearized around the variables of this subdomain at previous timestep and variables of neighboring subdomains at current timestep, which have not been determined. Thus, additional iterations are required. The non-adjacent subdomains almost have no direct dynamic interactions, this kind of subdomain POD-TPWL algorithm can be easily parallelized. Referring to Fig.3.1, subdomains $\Omega^1,\Omega^3,\Omega^5,\Omega^7,\Omega^9$ have no direct interactions, and therefore the subdomain POD-TPWL algorithm can be simultaneously implemented in these five subdomains. This is similar for the subdomains $\Omega^2,\Omega^4,\Omega^6,\Omega^8$. This parallelization for subdomain POD-TPWL is known as block red-black ordering [34]. The k-th iterative description of Eq.3.26 is as follows.

$$\begin{aligned}\boldsymbol{\psi}_k^{d,n+1} \approx \boldsymbol{\psi}_{tr}^{d,n+1} + \mathbf{E}_{\boldsymbol{\psi}_{tr}}^{d,n+1}(\boldsymbol{\psi}^{d,n} - \boldsymbol{\psi}_{tr}^{d,n}) \\ + \mathbf{E}_{\boldsymbol{\psi}_{tr}}^{sd,n+1}(\boldsymbol{\psi}_{k-1}^{sd,n+1} - \boldsymbol{\psi}_{tr}^{sd,n+1}) + \mathbf{G}_{\boldsymbol{\xi}_{tr}}^{n+1}(\boldsymbol{\xi} - \boldsymbol{\xi}_{tr})\end{aligned} \tag{3.31}$$

The iteration for subdomain POD-TPWL is very cheap, thus, we do not limit the maximum number of iterations. The iteration will be stopped when no further changes in the estimate of state, .e.g., pressure and saturation occur,

$$\frac{|\boldsymbol{\psi}^{k+1} - \boldsymbol{\psi}^{k}|}{max\{|\boldsymbol{\psi}^{k+1}|, 1\}} < \eta_{\boldsymbol{\psi}} \tag{3.32}$$

Nonetheless, the parallelisation is not explored in this thesis and is left as a future area of research.

3.4.4. SAMPLING STRATEGY

In our proposed subdomain POD-TPWL algorithm, training points are required for both RBF interpolation and to construct the POD basis. For POD, the snapshot matrix generated from the training simulations should accurately characterize the dynamic behavior of the system. The training simulations used to construct the RBF interpolation model should allow for accurate computation of derivative matrices. The procedure for choosing these training points will be described here.

Sampling strategy for POD. A small initial set of model parameter vectors is sampled and used as input for full-order model (FOM) simulations from which a snapshot matrix is constructed. The singular value spectrum is computed for this initial set of samples. The number of samples is then increased one at a time, i.e. adding one FOM simulation, and the SVD is recomputed, until no significant changes are observed in the singular value spectrum.

Sampling strategy for RBF. The accuracy of the RBF interpolation will be reduced if too few data points are chosen, while the computational cost increases with the number of data points, which will be prohibitive if too many points are chosen. To limit the number of FOM simulations used to construct the interpolation model for the POD coefficients we use 2-sided perturbation of each coefficient ξ_j resulting in $2 \times l_\beta + 1$ points. In some experiments we add additional points by simultaneous random sampling of perturbations $\Delta\boldsymbol{\xi}$. An alternative could be use to use Smolyak sparse grid sampling [35].

3.4.5. ADJOINT-BASED HISTORY MATCHING ALGORITHM

After constructing the linear reduced-order model using the proposed subdomain POD-TPWL methodology, it can be used within an adjoint-based history matching workflow.

The cost function evaluated using the reduced-order linear model is given as follows,

$$\begin{aligned} J(\boldsymbol{\xi}) =& \frac{1}{2}(\boldsymbol{\beta}_b + \boldsymbol{\Phi}_{\boldsymbol{\beta}}\boldsymbol{\xi} - \boldsymbol{\beta}_p)^T \mathbf{R}_p{}^{-1}(\boldsymbol{\beta}_b + \boldsymbol{\Phi}_{\boldsymbol{\beta}}\boldsymbol{\xi} - \boldsymbol{\beta}_p) \\ &+ \frac{1}{2}\sum_{d=1}^{S}\sum_{m=1}^{N_0}[\mathbf{d}_o^{d,m} - \mathbf{y}_{tr}^{d,m} - \mathbf{A}_{\boldsymbol{\psi}_{tr}}^{d,m}(\boldsymbol{\psi}^{d,m} - \boldsymbol{\psi}_{tr}^{d,m}) - \mathbf{B}_{\boldsymbol{\xi}_{tr}}^{m}(\boldsymbol{\xi} - \boldsymbol{\xi}_{tr})]^T \\ &(\mathbf{R}^m)^{-1}[\mathbf{d}_o^{d,m} - \mathbf{y}_{tr}^{d,m} - \mathbf{A}_{\boldsymbol{\psi}_{tr}}^{d,m}(\boldsymbol{\psi}^{d,m} - \boldsymbol{\psi}_{tr}^{d,m}) - \mathbf{B}_{\boldsymbol{\xi}_{tr}}^{m}(\boldsymbol{\xi} - \boldsymbol{\xi}_{tr})] \end{aligned} \tag{3.33}$$

After augmenting this cost function with the model equation Eq.3.27, the gradient with respect to the model parameters is obtained as

$$\begin{aligned} [\frac{dJ}{d\boldsymbol{\xi}}]^T =& (\boldsymbol{\phi}_{\boldsymbol{\beta}})^T \mathbf{R}_p{}^{-1}(\boldsymbol{\beta}_b + \boldsymbol{\phi}_{\boldsymbol{\beta}}\boldsymbol{\xi} - \boldsymbol{\beta}_p) \\ &- \sum_{d=1}^{S}\sum_{m=1}^{N_0}[\mathbf{B}_{\boldsymbol{\xi}_{tr}}^{m}]^T(\mathbf{R}^m)^{-1}[\mathbf{d}_o^{d,m} - \mathbf{y}_{tr}^{d,m} - \mathbf{A}_{\boldsymbol{\psi}_{tr}}^{d,m}(\boldsymbol{\psi}^{d,m} - \boldsymbol{\psi}_{tr}^{d,m}) \\ &- \mathbf{B}_{\boldsymbol{\xi}_{tr}}^{m}(\boldsymbol{\xi} - \boldsymbol{\xi}_{tr})] - \sum_{d=1}^{S}\sum_{n=1}^{N}[\mathbf{G}_{\boldsymbol{\xi}_{tr}}^{n}]^T\boldsymbol{\lambda}^{d,n} \end{aligned} \tag{3.34}$$

where $\boldsymbol{\lambda}^{d,n}$ is obtained as the solution of the adjoint model for the subdomain Ω^d as follows

$$\begin{aligned} [\mathbf{I} - (\mathbf{E}_{\boldsymbol{\psi}_{tr}}^{d,n})^T]\boldsymbol{\lambda}^{d,n} =& \sum_{d=1}^{S}[\mathbf{A}_{\boldsymbol{\psi}_{tr}}^{d,n}]^T\mathbf{R}_n{}^{-1}[\mathbf{d}_o^{d,n} - \mathbf{y}_{tr}^{d,n} \\ &- \mathbf{A}_{\boldsymbol{\psi}_{tr}}^{d,n}(\boldsymbol{\psi}^{d,n} - \boldsymbol{\psi}_{tr}^{d,n}) - \mathbf{B}_{\boldsymbol{\xi}_{tr}}^{n}(\boldsymbol{\xi} - \boldsymbol{\xi}_{tr})] + [\mathbf{E}_{\boldsymbol{\psi}_{tr}}^{sd,n}]^T\boldsymbol{\lambda}^{d,n+1} \end{aligned} \tag{3.35}$$

The minimization of the cost function Eq.3.33 is performed using a steepest descent algorithm [36] and is stopped when either one of the following stopping criteria is satisfied

- No further changes in the cost function

$$\frac{|J(\boldsymbol{\xi}^{k+1}) - J(\boldsymbol{\xi}^k)|}{max\{|J(\boldsymbol{\xi}^{k+1})|, 1\}} < \eta_J \tag{3.36}$$

- No further changes in the estimate of parameters,

$$\frac{|\boldsymbol{\xi}^{k+1} - \boldsymbol{\xi}^k|}{max\{|\boldsymbol{\xi}^{k+1}|, 1\}} < \eta_{\boldsymbol{\xi}} \tag{3.37}$$

- The maximum number of iterations has been reached. i.e

$$k <= N_{max} \tag{3.38}$$

where η_J and $\eta_{\boldsymbol{\xi}}$ are predefined error constraints and N_{max} is the maximum number of iterations.

As mentioned in [12], the solution of the reduced and linearized minimization problem based on Eq.3.33 is not necessarily the solution of the original problem. Therefore an additional stopping criterion should be introduced for the original model as follows [37],

$$N_d N_0 - 2\sqrt{2N_d N_0} \leqslant 2J(\boldsymbol{\beta}^k) \leqslant N_d N_0 + 2\sqrt{2N_d N_0} \tag{3.39}$$

where, N_0 is the number of timesteps where the measurements are taken, N_d is the number of measurements at each timestep, $\boldsymbol{\beta}^k$ represents the updated parameters vector at the k-th outer-loop. J is the cost function computed using the original reservoir model simulations.

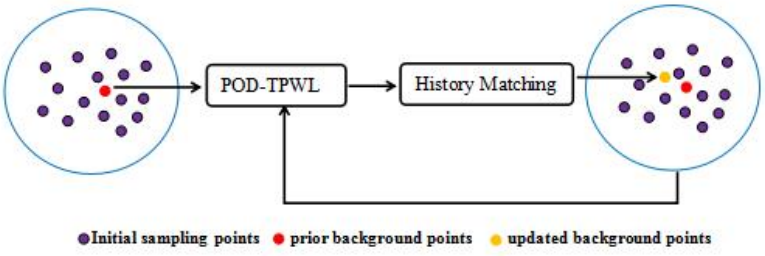


Figure 3.2: The illustration of the reconstruction method of subdomain POD-TPWL algorithm

If the objective function does not meet the stopping criterion as Eq.3.39, then additional outer-loops are required to reconstruct new reduced-order linear models using the updated parameters, and the aforementioned iterative inner-loop is performed again.

Our proposed non-intrusive subdomain POD-TPWL has computational advantages over the traditional construction of reduced-order linear models using perturbation-based finite-difference method proposed in [12], especially when the reduced-order linear model is required to be reconstructed for each outer-loop. Instead of re-perturbing the parameter and state variables one by one to approximate the derivative matrices as proposed in [12], which would require (l_p+l_s+l_β+1) FOM simulations, our algorithm runs only one additional FOM simulation using updated parameters. The updated parameters and simulated snapshots are added into the previous group of sampling interpolation points and corresponding snapshots. The derivative matrices for the updated parameters are approximated based on the updated group of interpolation points and snapshots.

3.5. NUMERICAL EXPERIMENTS AND DISCUSSION

In this section, two numerical experiments are presented that aim to demonstrate and evaluate our proposed adjoint-based history matching algorithm. The first experiment is based on a small 2D synthetic model containing 9 wells. The second experiment uses a reservoir model with 13 wells based on the SAIGUP benchmark case [38]. In our numerical experiments, MRST, a free open-source software for reservoir modeling and simulation[39], is used to run the FOM simulations.

3.5.1. CASE 1 - 2D RESERVOIR WITH 9 WELLS

DESCRIPTION OF MODEL SETTINGS

A 2D heterogeneous oil-water reservoir is considered with two-phase imcompressible flow dynamics. The reservoir contains 8 producers and 1 injector, which are labeled as P_1 to P_8, and I_1 respectively, see Fig.3.3. Detailed information about the reservoir geometry, rock properties, fluid properties, and well controls is summarized in Table 3.2.

REDUCED MODEL CONSTRUCTION

We generate an ensemble of 1000 Gaussian-distributed realizations of log-permeability.

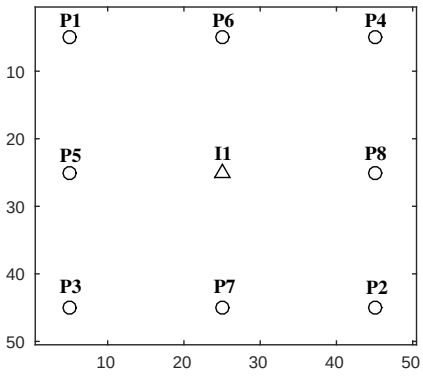


Figure 3.3: The well placement in the 2-D reservoir model for case 1

We also assume that the generated log-permeability fields are not conditioned to the permeability values at the well locations. The log-permeability fields and the corresponding porosity fields are described by the following statistics:

$$\sigma_{\boldsymbol{\beta}} = 5 \tag{3.40}$$

$$\mathbf{C}_{\boldsymbol{\beta}}(x_{i1,j1}; y_{i2,j2}] = \sigma^2_{\boldsymbol{\beta}} e^{-[(\frac{|x_{i1}-x_{i2}|}{\chi x})^2+(\frac{|y_{i1}-y_{i2}|}{\chi y})^2]} \tag{3.41}$$

$$\frac{\chi_x}{L_x} = 0.2, \frac{\chi_y}{L_y} = 0.2 \tag{3.42}$$

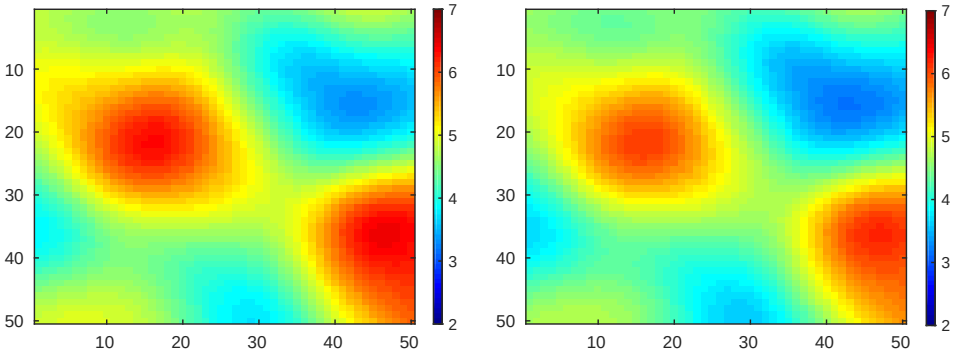
$$\boldsymbol{\phi} = 0.25(\frac{e^{\boldsymbol{\beta}}}{200})^{0.1} \tag{3.43}$$

Here, $\sigma_{\boldsymbol{\beta}}$ is the standard deviation of log-permeability $\boldsymbol{\beta}$; $\mathbf{C}_{\boldsymbol{\beta}}$ is the covariance of $\boldsymbol{\beta}$; $x_{i1,j1}$=(x_{i1},y_{j1}) denotes the coordinates of a grid block; χ_x (or χ_y) is the correlation length in x (or y) direction; and L_x (or L_y) is the domain length in x (or y) direction. The background log-permeability $\boldsymbol{\beta}_b$ is taken as the average of the 1000 realizations. One of the realizations was considered to be the truth, and is illustrated in Fig.3.4(a). The permeability field was parameterized using KL-expansion and about 95% energy is maintained, resulting in 18 permeability patterns with $l_{\boldsymbol{\beta}} = 18$ corresponding independent PCA coefficients, which are used in the workflow as a low-dimensional representation of the 2500 grid block permeability values. Fig.3.4(b) shows the projection of the 'true' permeability field in this low-dimensional subspace which shows that the truth can be almost perfectly reconstructed in this subspace. Four realizations for log-permeability field generated are additionally shown in Fig.3.5.

After having reduced the parameter space, the next step is to reduce the reservoir model. The first step is to generate a set of training runs from which snapshots will be taken. Since the required number of training runs is not known a priori we follow the following procedure: (1) generate a sample PCA coefficient vector by sampling from the set $\{-1,1\}$, (2) run a full-order model simulation with these parameters, (3) extract snapshots and form the snapshot matrix, (4) compute the singular value decomposition of the snapshot matrix (5) repeat steps (1) to 4) until changes in the singular values are insignificant.

Table 3.2: Experiment settings using MRST for case 1

Description	Value
Dimensions	50 × 50 ×1
Grid cell size	20 × 20 × 10
Number of wells	8 producers, 1 injector
Fluid density	1014 kg/m^3, 859 kg/m^3
Fluid viscosity	0.4 mP·s, 2 mP·s
Initial pressure	30 MPa
Initial saturation	S_o=0.80, S_w=0.20
Connate water saturation	S_{wc}=0.20
Residual oil saturation	S_{or}=0.20
Corey exponent, oil	4.0
Corey exponent, water	4.0
Injection rate	200 m^3/d
BHP	25 MPa
History production time	5 year
Prediction time	10 year
Timestep	0.1 year
Measurement timestep	0.2 year



(a) 'True' model in original full-order space (b) Projected 'true' model in reduced-order subspace

Figure 3.4: Comparison of the 'true' reservoir model in full-order space and in reduced-order space for Case 1

For Case 1 this produced a set of 15 training runs and 240 snapshots for pressure and saturation each.

ERROR QUANTIFICATIONS

The performance of the subdomain POD-TPWL model can be investigated by comparing the errors relative to FOM simulation for quantities of interest. Here, errors are quantified in terms of the mismatch of the fluid rate, water-cut and primal variables, i.e., pressure and saturation between the FOM solution d_{FOM} and subdomain POD-TPWL simulations d_{ROM}. For example, the average fluid rate error E_{fr} or the average water-cut error E_{wct} is

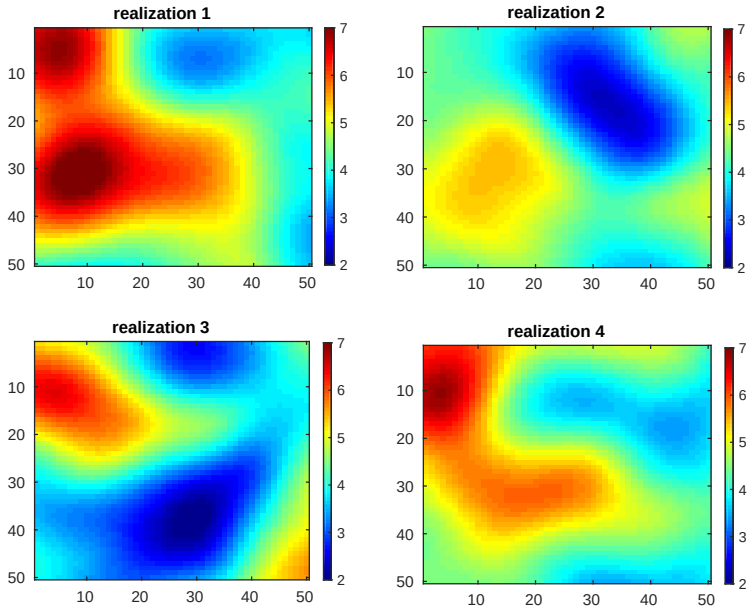


Figure 3.5: Examples of realizations of the log-permeability field generated from PCA coefficients sampled randomly from the set {-1, 1} for Case 1

calculated as

$$E_{fr} \quad \text{or} \quad E_{wct} = \frac{1}{NN_d} \sum_{i=1}^{N} \sum_{j=1}^{N_d} \frac{|(\mathbf{d}_{FOM}^{i,j} - \mathbf{d}_{ROM}^{i,j})|}{\mathbf{d}_{FOM}^{i,j}} \tag{3.44}$$

Where, $\mathbf{d}$ represents the fluid rate or water-cut. Similarly, the pressure error E_p and saturation error E_s are formulated as

$$E_p \quad \text{or} \quad E_s = \frac{1}{NN_d} \sum_{i=1}^{N} \sum_{j=1}^{N_d} \frac{|(\mathbf{x}_{FOM}^{i,j} - \mathbf{x}_{ROM}^{i,j})|}{\mathbf{x}_{FOM}^{i,j}} \tag{3.45}$$

Where, $\mathbf{x}$ represents the saturation or pressure in each gridblock at each timestep.

In our case, the initial 37 sampling points are selected as described in subsection of sampling strategy, where the j-th element $\boldsymbol{\xi}_i^j$ of the i-th PCA coefficient vector $\boldsymbol{\xi}^j$ is perturbed sequentially in 2 opposite directions (positive and negative) by a specific amplitude perturbation $\Delta\boldsymbol{\xi}_i^j$. These 37 sampling points are used to build a subdomain reduced-order linear model. Four factors, e.g, domain decomposition, projection energy, testing interval and training interval, are considered to analyze the sensitivity of model errors. The experiment settings are shown in Table 3.3. We specify the stopping criterion $\eta_{\boldsymbol{\psi}} = 10^{-3}$.

The detailed information about the error quantification can be found in our Supplementary file, the main results are summarized here. Fig.3.6 shows the error in fluid rate, water-cut, pressure and saturation as a function of the four factors. A relatively small subdomain size of 3 x 3 cells produced the most accurate results for this case. Accuracy is also improved by increasing the energy threshold and retaining more POD patterns, albeit at

an increased computational cost. Retaining 95% of the total energy during projection produces an acceptable accuracy in this case. Increasing the testing interval, which represents the maximum discrepancy between test model and linearized training model, severely deteriorates the reduced model accuracy, with the best results obtained with the [-0.1, 0.1] interval. an appropriate iteration step-length for the history matching process should be set as 0.1 based on our numerical experiments.

In terms of computational effort, the runtime for a single FOM simulation for this case was about 2 seconds on a machine with i5-4690 Intel CPUs (4 cores, 3.5GHz) and 24 GB memory using Matlab-R2015a. The subdomain POD-TPWL models, in contrast, required less than 0.2 seconds. However, the construction of subdomain POD-TPWL requires simulating 52 training models, POD, derivative estimation using RBF, plus additional overhead, which severely increases the cost. Therefore, it would not make sense to construct the subdomain POD-TPWL model unless it is to be used for a large number of simulations. Because many simulations are required in history matching applications, the subdomain POD-TPWL model should be applicable in this context. The use of subdomain POD-TPWL in conjunction with an adjoint-based data assimilation procedure is presented in the following parts.

Table 3.3: Experiment design of error quantification for case 1

Domain decomposition	Projection energy	Testing interval	training interval
3×3	90%	[-0.1,0.1]	[-0.1,0.1]
4×4	95%	[-0.2,0.2]	[-0.5,0.5]
5×5	99%	[-0.3,0.3]	[-1,1]

HISTORY MATCHING PROCEDURE

Based on the error sensitivity analysis presented above, we divide the entire domain into 9 (3 × 3) rectangle subdomains as illustrated in Fig.3.7. The choice of subdomains is fairly arbitrary at this point since we have no formal algorithm to determine the best number and design of the subdomains. The previously collected global snapshots for pressures and saturations are divided into local snapshots. For each subdomain, two separate eigenvalue problems for pressure and saturation are solved using POD. The number of reduced parameters and state patterns for each subdomain and for the global domain are listed in Table 3.4 where specific projection energy, e.g, 95% and 95%, are preserved for the pressure and saturation respectively in each subdomain.

The history period is 5 years during which observations are taken from 8 producers and 1 injector every second simulation time step (nearly 73 days) resulting in 25 time instances. Noisy observations are generated from the model with the "true" permeability field and include bottom-hole pressures (BHP) in the injector and fluid rates and water-cut (WCT) in the producers. As a result we have 200 fluid rates and 200 WCT values measured in the producers and 25 bottom-hole pressures measured in the injector, which gives in total 425 measurement data. Normal distributed independent measurement noise with a standard deviation equal to 5% of the 'true' data value, was added to all observations. The generated measurements are shown in Fig.3.8.

To analyze the results, we define two error measures based on data misfits e_{obs} and

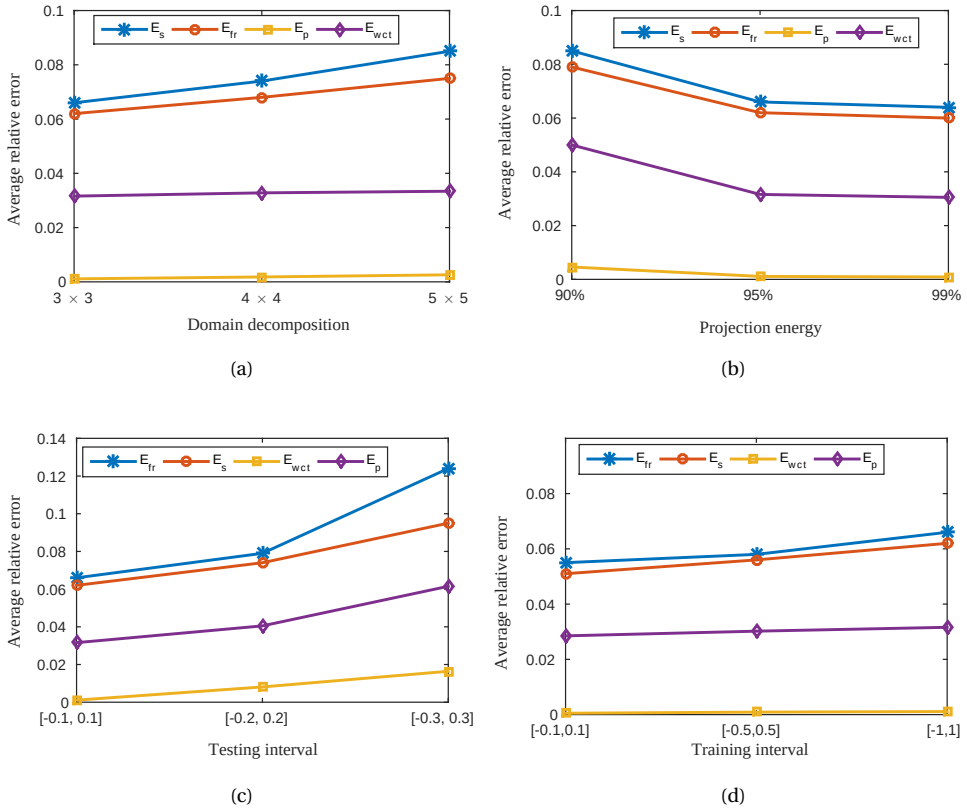


Figure 3.6: Average subdomain POD-TPWL errors for test case as a function of domain decomposition, projection energy, testing interval and training interval for case 1. Results are for TPWL models constructed using 37 training simulations

Table 3.4: Summary of the number of reduced variables for the global domain and after domain decomposition for case 1 (Note:**s** refers to saturation, **p** refers to pressure)

Domain Decomposition				Global Domain		
Index of subdomain	$\boldsymbol{\beta}$	**s**	**p**	$\boldsymbol{\beta}$	**s**	**p**
1		14	7			
2		13	6			
3		12	5			
4		13	4			
5	18	16	7	18	72	36
6		14	6			
7		13	5			
8		15	6			
9		12	5			
Total number	18	122	51	18	72	36

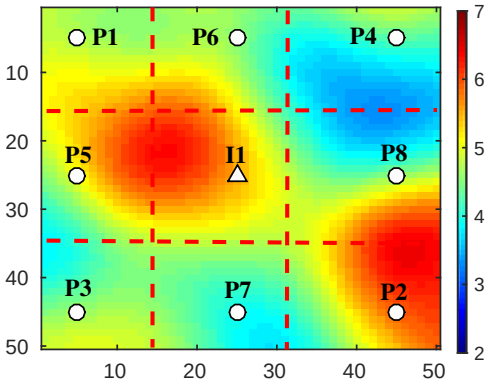


Figure 3.7: Illustration of the applied domain decomposition for Case 1.

parameter misfits $e_{\boldsymbol{\beta}}$ as follows,

$$e_{obs} = \sqrt{\frac{\sum_{i=1}^{N_o}\sum_{j=1}^{N_d}(\mathbf{d}_{obs}^{i,j} - \mathbf{d}_{upt}^{i,j})^2}{N_o N_d}} \tag{3.46}$$

$$e_{\boldsymbol{\beta}} = \sqrt{\frac{\sum_{i=1}^{N_g}(\boldsymbol{\beta}_{true}^{i} - \boldsymbol{\beta}_{upt}^{i})^2}{N_g}} \tag{3.47}$$

where, $\mathbf{d}_{obs}^{i,j}$ and $\mathbf{d}_{upt}^{i,j}$ represent the measurements and simulated data using the updated model respectively; $\boldsymbol{\beta}_{true}^{i}$ and $\boldsymbol{\beta}_{upt}^{i}$ denote the grid block log-permeability from the 'true' model and updated model respectively.

Figures 3.9, 3.10 and 3.11 and Table 3.5 show the results of the first numerical experiments, including the updated log-permeability field, the value of cost function at each iteration and the mismatch between observed data and predictions. To demonstrate the performance of our proposed methodology, we compared the results with those of finite-difference (FD) based history matching algorithm without domain decomposition and model order reduction. The total computational cost of any minimization problem strongly depends on the number of parameters. In our work, for a fair comparison, we use the same parameterization to reduce the number of parameters and implement FD based history matching in this reduced-order parameter subspace. The cost function for FD based history matching can be defined as follows,

$$\begin{aligned} J(\boldsymbol{\xi}) =& \frac{1}{2}(\boldsymbol{\beta}_b + \boldsymbol{\phi}_{\boldsymbol{\beta}}\boldsymbol{\xi} - \boldsymbol{\beta}_p)^T \mathbf{R}_p{}^{-1}(\boldsymbol{\beta}_b + \boldsymbol{\phi}_{\boldsymbol{\beta}}\boldsymbol{\xi} - \boldsymbol{\beta}_p) \\ &+ \frac{1}{2}\sum_{m=1}^{N_0}(\mathbf{d}_o^m - \mathbf{h}^m(\mathbf{x}^m, \boldsymbol{\xi}))^T(\mathbf{R}^m)^{-1}(\mathbf{d}_o^m - \mathbf{h}^m(\mathbf{x}^m, \boldsymbol{\xi})) \end{aligned} \tag{3.48}$$

The FD method is used to compute the numerical gradient of the cost function as Eq.3.48 with respect to 18 PCA coefficients. A FD gradient is determined by one-sided perturbation of each of the 18 PCA coefficients. Thus, 19 full-order model (FOM) simulations are required for each iteration step. The stopping criteria are set $\eta_J = 10^{-4}$, $\eta_{\boldsymbol{\xi}} = 10^{-3}$,

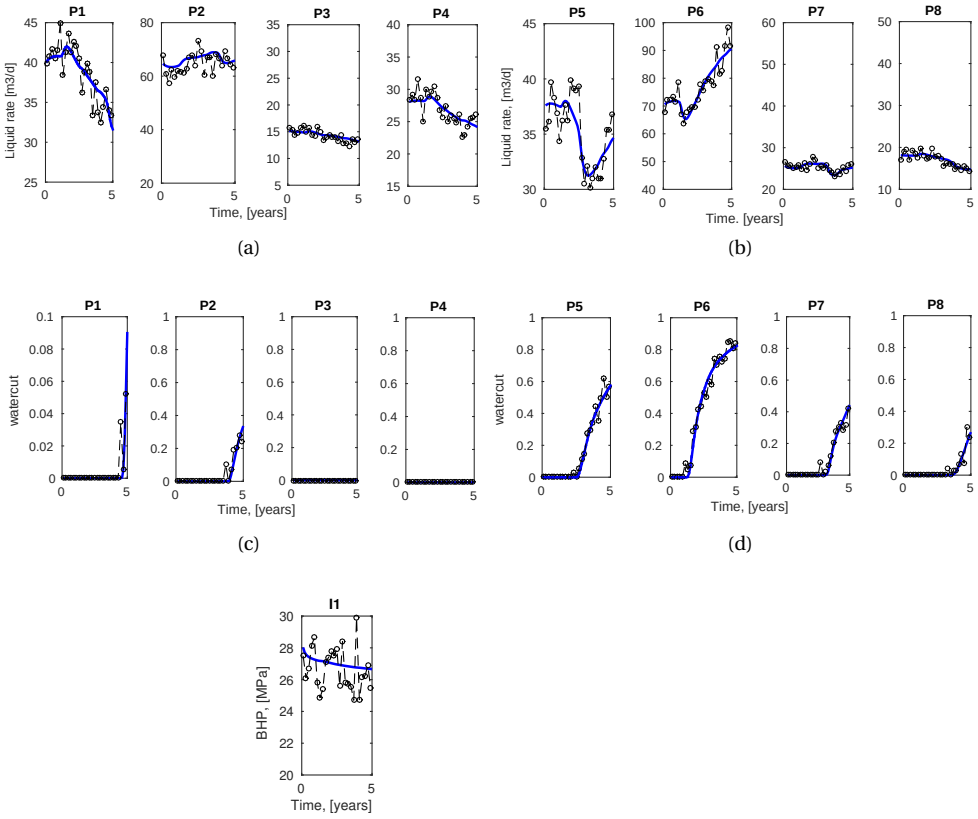


Figure 3.8: Measured quantities for case 1. Blue solid line: reference model (truth), black dashed line: noisy data.

and N_{max}=30. As can be seen from Fig.3.10 and Table 3.5, the model-reduced approach needs 55 FOM simulations, among them, 15 FOM simulations are used to collect the snapshots and 37 FOM simulations are used to construct the initial reduced-order linear model in the first outer-loop. The remaining 3 FOM simulations are used to reconstruct three new reduced-order linear models in the next three outer-loops and to calculate the cost function in the original space. Fig.3.9 shows the true, initial and final estimates of log-permeability field. In this case, the main geological structures of the the 'true' model can be reconstructed with both methods. However, the parameter estimates obtained with proposed methodology more accurately reproduce the true amplitudes than those obtained with FOM based history matching. From Fig.3.11 and Table 3.5, we can both qualitatively and quantitatively observe that the history matching process results in an improved prediction in all of the eight production wells. Fig.3.11 illustrates the data match of fluid rate and water-cut up to year 5 and an additional 5-year prediction until year 10 for all 8 producers. The prediction based on the initial model is far from that of the true model. After history matching, the predictions from the updated model match the observations very well. Also the prediction of water breakthrough time is improved for all of the

production wells, also for wells that show water breakthrough only after the history period.

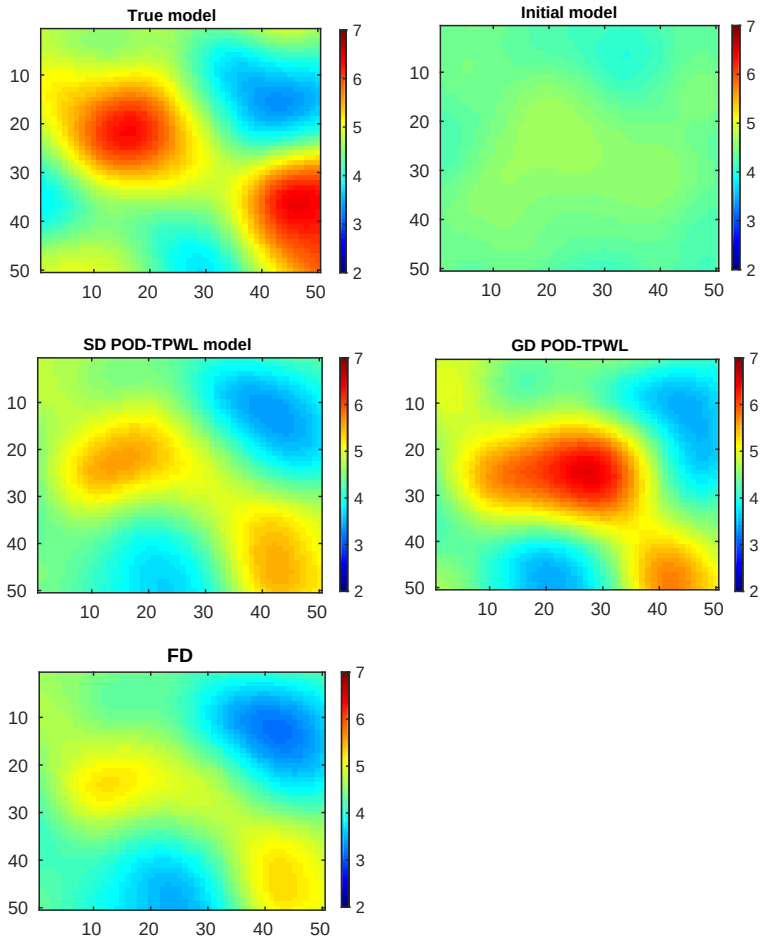
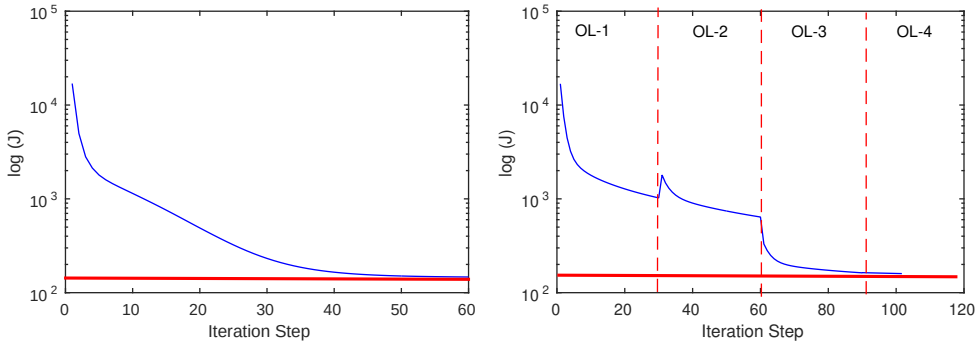


Figure 3.9: True, initial and updated log-permeability fields using SD POD-TPWL, GD POD-TPWL, and the FD method for case 1

Table 3.5: comparison between SD POD-TPWL and FD method for case 1

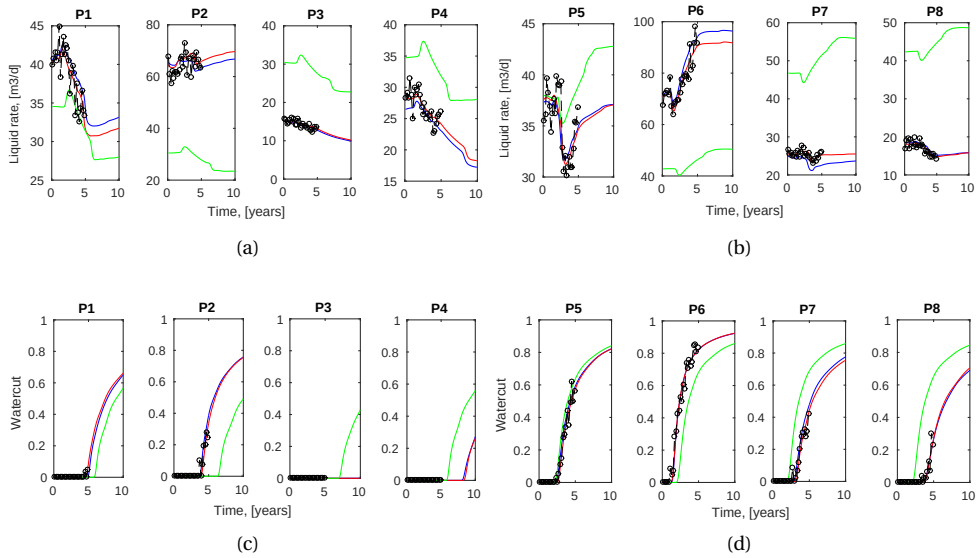
-	Iterations	FOM	$J(\boldsymbol{\xi}) \times 10^4$	e_{obs}	$e_{\boldsymbol{\beta}}$
Initial model	-	-	1.69	28.38	2.28
SD POD-TPWL	103	55	0.0160	3.35	0.68
FD	52	988	0.0153	3.28	0.72
'True' model	-	-	0.0068	2.12	0

One of the key issues for the subdomain POD-TPWL is the implementation of the domain decomposition technique. Our proposed subdomain POD-TPWL (SD POD-TPWL) can be easily generalized to the global domain POD-TPWL (GD POD-TPWL). The dif-



(a) FD method (b) subdomain POD-TPWL

Figure 3.10: Cost function value decrease using (a) finite-difference method, and (b) subdomain POD-TPWL for case 1. OL-i means the i-th outer-loop



(a) (b)

(c) (d)

Figure 3.11: Forecasts of the liquid rate and water-cut for case 1 from the initial model (green line), the reference model (blue line), and the model updated using the SD POD-TPWL (red line). Measured data are indicate by open circles.

ferences between SD POD-TPWL and GD POD-TPWL are: a) model order reduction in global domain versus reduction in each subdomain separately; b) derivative estimation using RBF interpolation in the global domain versus interpolation for each subdomain. As shown in Table 3.4, the total dimension of the reduced-order linear model is 18+122+51=191 for domain decomposition and 18+72+36 =126 for the global domain. Table 3.6 shows the total number of the reduced variables in each subdomain and in the global domain. While the total sum of the reduced variables in each subdomain is larger than that of the

global domain, the number of reduced variables in each individual subdomain is relatively small. Furthermore, these local reduced variables have the surprisingly abilities to accurately capture the flow dynamics, as suggested by Fig.3.12. Fig.3.12 shows the distribution of pressure and saturation at the final time. In this case, the reconstructions of the saturation and pressure field using a small number of patterns in each subdomain are comparable with those of the global domain. In addition, as shown in Table.3.7, both GD POD-TPWL and SD POD-TPWL can converge to satisfactory results. The SD POD-TPWL needs 55 FOM simulations, while the GD POD-TPWL algorithm requires 73 FOM simulations (15 FOM simulations are run to collect the snapshots, 55 FOM simulations are used to construct the initial reduced-order linear model in the first outer-loop, and the remaining 3 FOM simulations are used to reconstruct the reduced-order linear models in the following three outer-loops). Therefore, compared to the global RBF interpolation, the proposed local RBF interpolation technique requires only a small number of reduced variables per subdomain and is much more computationally efficient. If the dimension of the underlying model would be much larger, the GD POD-TPWL would result in a reduced-order linear model with a higher dimension and therefore more interpolation points would be required in the RBF scheme. In the SD POD-TPWL algorithm this problem is avoided since for large-scale problems the dimension of the reduced-order linear model for the subdomain does not increase significantly, we only need to activate more subdomains.

Table 3.6: The number of interpolation variables in each subdomain and global domain for case 1. Ω^d is the d-th subdomain

	Domain Decomposition	Global Domain
Ω^1	75=21(Ω^1)+19(Ω^2)+17(Ω^4)+18	
Ω^2	98=21(Ω^1)+19(Ω^2)+17(Ω^3)+23(Ω^5)+18	
Ω^3	74=19(Ω^2)+17(Ω^3)+20(Ω^6)+18	
Ω^4	97=21(Ω^1)+17(Ω^4)+23(Ω^5)+18(Ω^7)+18	
Ω^5	118=19(Ω^2)+17(Ω^4)+23(Ω^5)+20(Ω^6)+21(Ω^8)+18	126 =72 +36+18
Ω^6	95=17(Ω^3)+23(Ω^5)+20(Ω^6)+17(Ω^9)+18	
Ω^7	74=17(Ω^4)+18(Ω^7)+21(Ω^8)+18	
Ω^8	97=23(Ω^5)+18(Ω^7)+21(Ω^8)+17(Ω^9)+18	
Ω^9	76=20(Ω^6)+21(Ω^8)+17(Ω^9)+18	

For Case 1, history matching results using GD POD-TPWL are slightly better than those from the subdomain POD-TPWL, especially for the high-permeable zone, e.g, the red area in Fig.3.9. The water-front of the waterflooding process propagates forward quickly (as the blue area in Fig.3.12) and therefore there are strong dynamic interactions within this area. Our chosen domain decomposition may artificially cut off this inherent dynamic interaction between the east-south corner and the west-north corner. A flow-informed domain decomposition technique may therefore be required to identify the relevant dynamic interactions, especially for strongly heterogeneous reservoir models such as those based on strongly contrasting facies distributions or channels.

3.5.2. CASE 2 - 2D BENCHMARK MODEL WITH 13 WELLS

DESCRIPTION OF MODEL SETTINGS

In the second numerical experiment, the SAIGUP model [38] is used to test our proposed adjoint-based history matching approach. The first layer containing a total of 3895

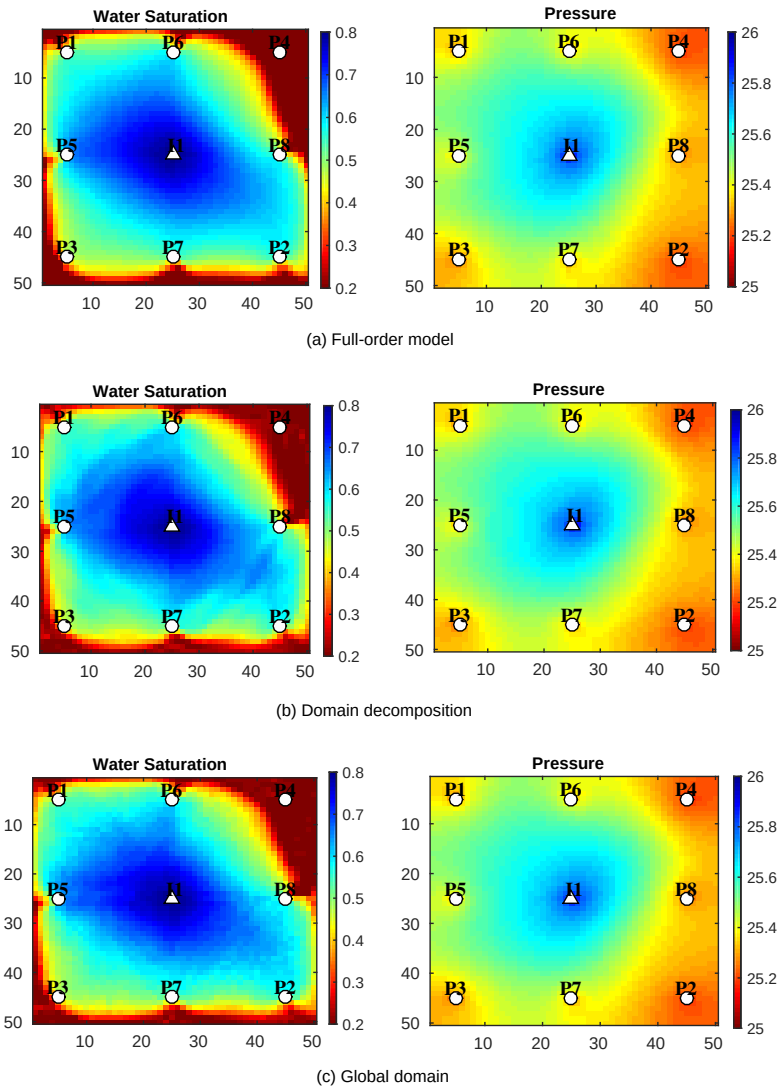


Figure 3.12: Water saturation and pressure fields from the full-order model and from SD POD-TPWL and GD POD-TPWL based models for case 1

active and 905 inactive grid cells is chosen for our test case. The realistic geological properties, e.g, faults, are preserved. The reservoir model describes an oil–water two-phase flow system with six producers and seven injectors, which are labeled from P_1 to P_6, and I_1 to I_7, see Fig.3.13. Some detailed information about reservoir geometry, rock properties, fluid properties, and well controls are shown in Table 3.8.

DESCRIPTION OF REDUCED MODEL PROCEDURE

Similarly, as Eq.3.40-Eq.3.41, we generate an ensemble of 1000 Gaussian-distributed

Table 3.7: comparison between SD POD-TPWL and GD POD-TPWL for case 1

-	FOM	$J(\boldsymbol{\xi}) \times 10^4$	e_{obs}	$e_{\boldsymbol{\beta}}$
Initial model	-	1.69	28.38	2.28
SD POD-TPWL	55	0.0160	3.35	0.68
GD POD-TPWL	73	0.0140	3.21	0.61
FD	988	0.0153	3.28	0.72
'True' model	-	0.0068	2.12	0

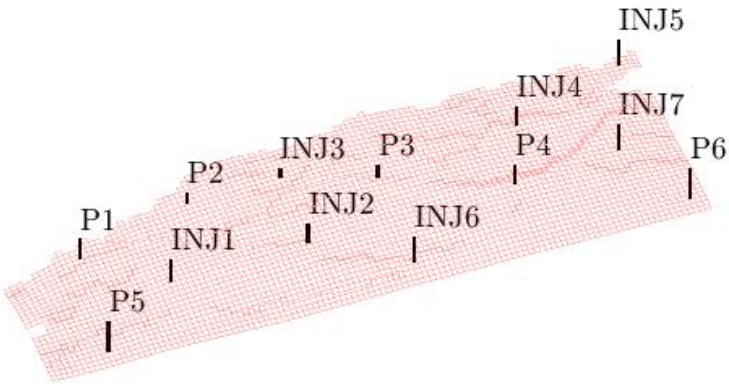
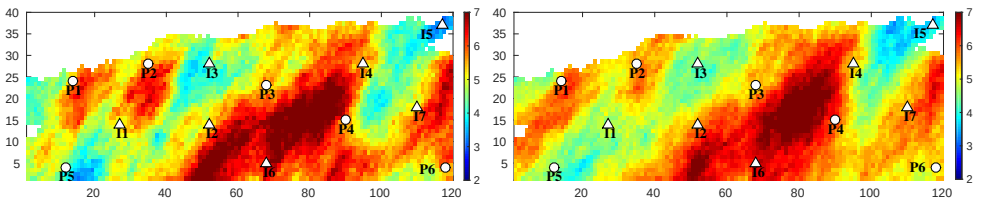


Figure 3.13: The 3-D view of the well placement for case 2

realizations of log-permeability. One of these realizations was considered to be the truth, as Fig.3.14(a). To efficiently implement our methodology in this much more realistic case, the log-permeability field was parameterized using a KL-expansion described above and about 90% energy is maintained, resulting in 44 permeability patterns to represent the uncertainty in all 3895 active grid cells. Fig.3.14(b) represents the 'true' permeability field projected onto the subspace spanned by these $l_{\boldsymbol{\beta}}$=44 PCA coefficients. Most parts of the properties of the original 'true' model is reconstructed in the reduced-order parameter subspace, while parts of properties are lost, e.g, the high-permeable area around the producer P_2. Four realizations for log-permeability field generated from random coefficients sampled from the set {-1, 1} are illustrated in Fig.3.15.



(a) 'True' model in original full-order space (b) Projected 'true' model in reduced-order subspace

Figure 3.14: The comparison of the 'true' reservoir model in full-order space and reduced-order space for case 2

Table 3.8: Experiment settings using MRST for case 2

Description	Value
Dimension	40×120×1
Number of wells	6 producers, 7 injectors
Constant porosity	0.2
Fluid density	1014 kg/m^3, 859 kg/m^3
Fluid viscosity	0.4 mP·s, 2 mP·s
Initial pressure	30 MPa
Initial saturation	S_o=0.80, S_w=0.20
Connate water saturation	S_{wc}=0.20
Residual oil saturation	S_{or}=0.20
Corey exponent, oil	4.0
Corey exponent, water	4.0
Injection rate	250 m^3/d
BHP	25 MPa
History production time	10 year
Prediction time	15 year
Timestep	0.1 year
Measurement timestep	0.2 year

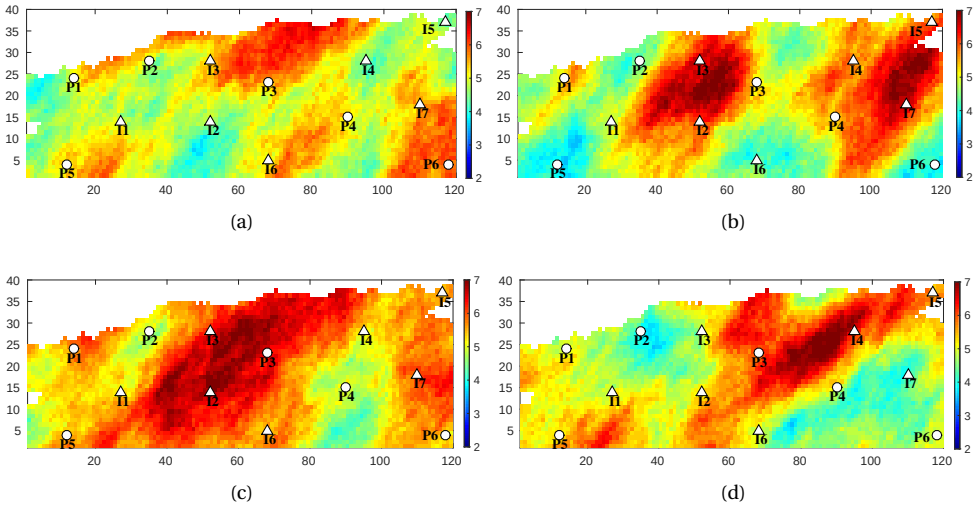


Figure 3.15: The comparison of the 'true' reservoir model in full-order space and reduced-order space for case 2

Experiments showed that the changes in singular value spectrum of the snapshot matrix are insignificant when the snapshots at every timestep are selected from 20 or more FOM simulations. These 20 FOM are also sampled from the interval {-1, 1} to effectively preserve the dynamic behavior. Finally, we collect 2000 snapshots for pressures and saturations separately. Comparing to the previous synthetic model, the existence of the faults and strong heterogeneity makes the flow dynamic o f SAIGUP more complicated. The er-

ror quantification for subdomain POD-TPWL is also implemented and we obtain some consistent results with the Case 1, thus, the detailed implementation of error quantification is not described here. To effectively capture the local physical features, we divide the whole model domain into 40 subdomains (4 subdomains in x direction times 10 subdomains in y direction) as in Fig.3.16. The inactive grids are intentionally considered to convert the original irregular model into a rectangle reservoir model, and subsequently this regular model can be conveniently decomposed into subdomains. In addition, to increase the efficiency, we do not construct the reduced-order linear model in the subdomain if the number of active grids is less than 5. For each subdomain, two eigenvalue problems separately for pressure and saturation are solved using POD. Both decompositions preserve 90% of the energy, and the number of reduced variables for each subdomain is shown in Fig.3.17.

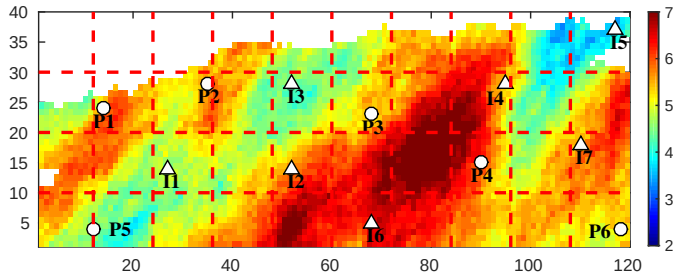


Figure 3.16: The 2-D view of domain decomposition in the 'true' model for case 2

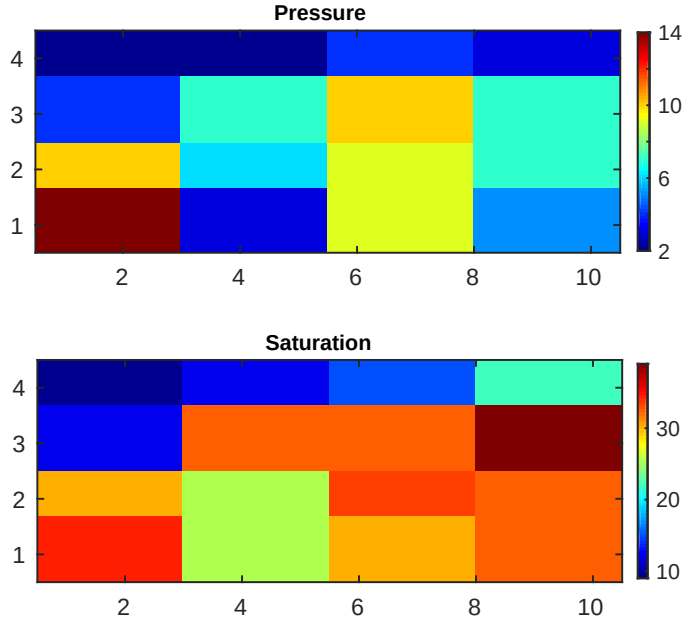


Figure 3.17: The number of reduced pressure and saturation variables in each subdomain for case 2

The history period is 10 years during which observations are taken from these 13 wells every 0.2 years, resulting in 300 fluid rates and 300 WCT values measured in the producers and 350 bottom-hole pressures measured in the injectors, which give in total 950 data points. The generated measurements for producing wells are shown in Fig.3.18.

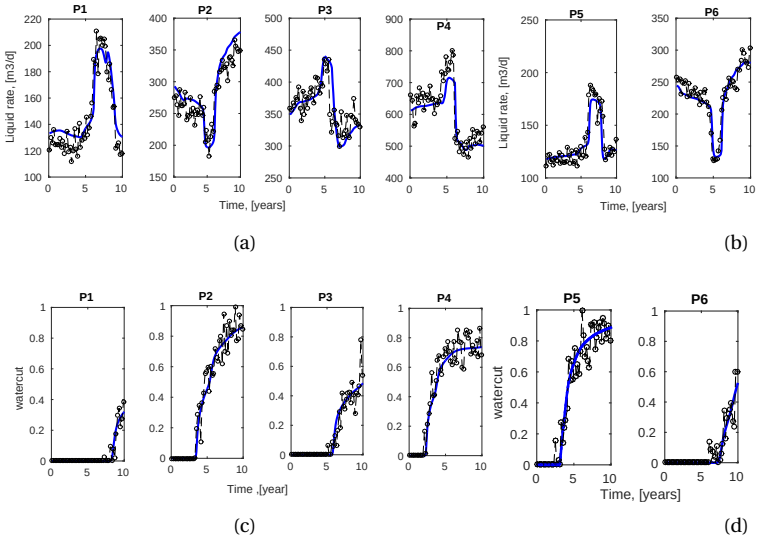


Figure 3.18: Example of synthetic measurements for case 2: Blue solid line: reference model (truth), black dashed line: noisy data.

HISTORY MATCHING RESULTS

The accuracy of RBF interpolation depends on the interpolation points, and the number of interpolation points depends on the number of interpolation variables. In the previous synthetic model (retaining 18 PCA coefficients), the number of sampling points using the simple two-sides perturbation strategy is sufficient to represent the interval {-1, 1}. However, in this case, retaining 44 PCA coefficients forces us to sample much more interpolation points. We also compare the results with those of the FD based history matching algorithm, which is implemented in the reduced-order parameter subspace spanned by the 44 log-permeability patterns. The one-side FD method is used to compute the numerical gradient of cost function as Eq.3.48 with respect to 44 PCA coefficients. Thus, 45 FOM simulations are required for each iteration step. The stopping criteria are set to $\eta_J = 10^{-4}$, $\eta_{\boldsymbol{\xi}} = 10^{-3}$, N_{max}=50 in this case.

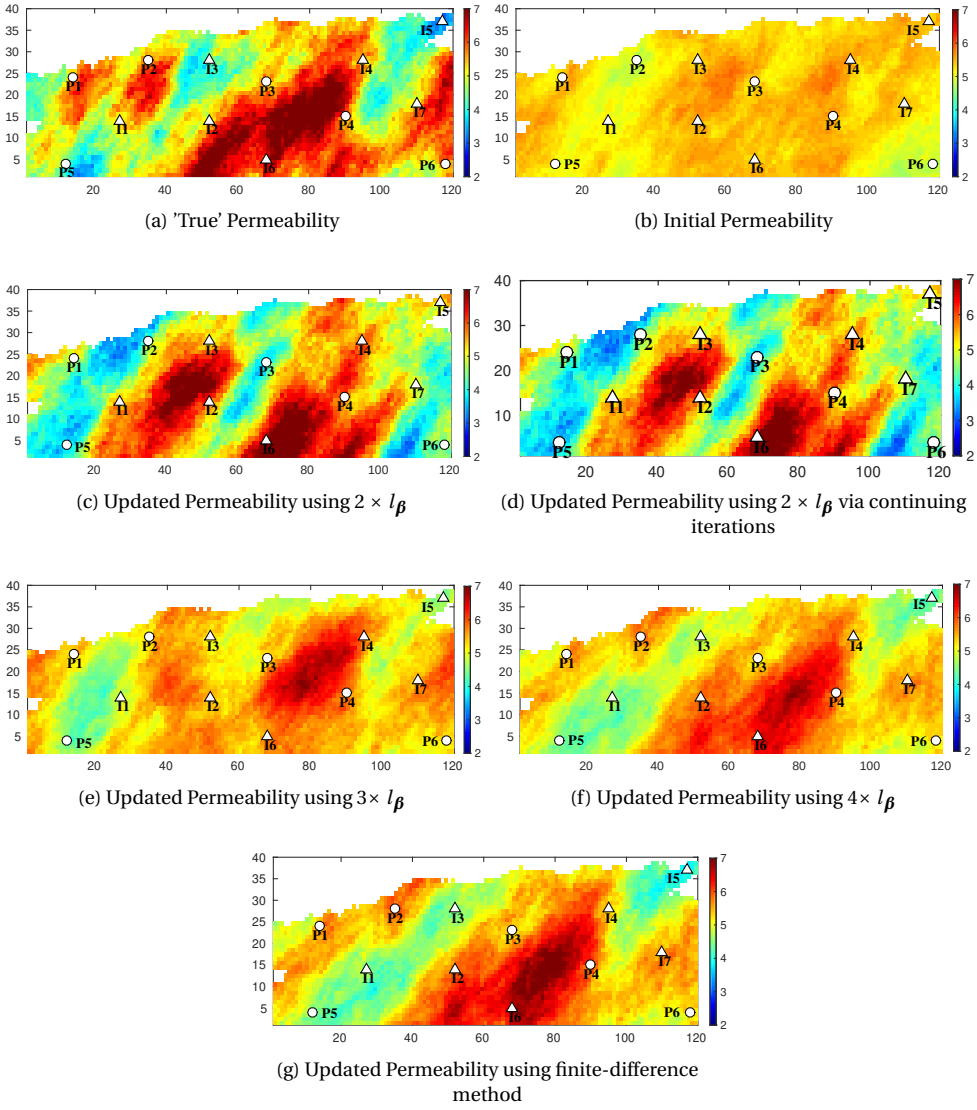
Fig.3.19, Fig.3.20 and Table 3.9 show the updated log-permeability field using different numbers of sampling points. In our case, the sampling strategy is that the first $2\times l_{\boldsymbol{\beta}}$ sampling points are selected from the set {-1,1} through perturbing each PCA coefficient sequentially in 2 opposite directions (positive and negative), and then the remaining sampling points are chosen randomly within the interval [-1, 1]. During the minimization procedure, a small iteration step is used to ensure a decreasing cost function when we use a small number of sampling points, e.g, $2\times l_{\boldsymbol{\beta}}$, which leads to a less accurate approximate gradient.

Fig.3.20 shows that using $2 \times l_{\boldsymbol{\beta}}$ sampling points leads to relatively slow convergence. In our predefined stopping criteria, totally 221 iteration steps with five outer-loops are required. Fig.3.19 illustrates that the 'true' log-permeability field is approximately calibrated when increasing the number of sampling points, e.g, $3 \times l_{\boldsymbol{\beta}}$ and even $4 \times l_{\boldsymbol{\beta}}$ in our case. It is easily recognized that using small number of sampling points leads to bad quality of approximate gradients derived from the subdomain reduced adjoint model, which will deteriorate the minimization procedure. We continue the iteration process when using $2 \times l_{\boldsymbol{\beta}}$ sampling points, to investigate whether there are potentials to obtain comparable results with that of FD method. As Fig.3.20, the vertical blue dash line represents the starting point using new stopping criterion, and the dash line represents the corresponding cost function of the new iteration process. Another 122 iteration steps and 2 outer-loops are required to reach convergence. In contrast to the old stopping criterion, the final cost function is decreased by 4.9% in this case. We should note that these additional 122 iteration steps just require 2 new FOM simulations for the two outer-loops. This property makes our methodology significantly attractive because much more iteration steps do not explosively increase the FOM simulations. Comparison between Fig.3.19(c) and Fig.3.19(d) also shows that the log-permeability filed has slight improvement when continuing the iteration process. Comparison between Fig.3.19(f) and Fig.3.19(g) demonstrates that our model-reduced approach obtains comparable updated log-permeability field with the FD method using $4 \times l_{\boldsymbol{\beta}}$ sampling points, e.g, 199 FOM simulations, and among them, 20 FOM simulations are run to collect the snapshots, 176 FOM simulations are used to construct the initial reduced-order linear model in the first outer-loop, the remaining 3 FOM simulations are used to reconstruct the reduced-order linear models in the next three outer-loops and calculate the cost function in original space, while the FD method requires 3510 FOM simulations.

Table 3.9: Comparison between SD POD-TPWL and FD method using different number of sampling points for case 2. SD POD-TPWL^1 $\times 2$ means the stopping criteria $\eta_J = 10^{-4}$, $\eta_{\boldsymbol{\xi}} = 10^{-3}$, N_{max}=50, while SD POD-TPWL^2 $\times 2$ means the stopping criteria $\eta_J = 10^{-5}$, $\eta_{\boldsymbol{\xi}} = 10^{-4}$, N_{max}=50

-	Iterations	FOM	$J(\boldsymbol{\xi}) \times 10^5$	e_{obs}	$e_{\boldsymbol{\beta}}$
Initial model	-	-	1.1417	73.5892	0.7078
SD POD-TPWL^1 $\times 2$	221	113	0.2117	18.1208	0.7837
SD POD-TPWL^2 $\times 2$	343	115	0.1998	16.3524	0.7797
SD POD-TPWL$\times 3$	194	155	0.1931	14.8523	0.5017
SD POD-TPWL$\times 4$	189	199	0.1892	11.0852	0.4182
FD	78	3510	0.1935	11.2532	0.3925
'True' model	-	-	0.1777	9.2531	0

Alternatively, we limit the sampling points within a relative small interval {-0.1, 0.1} so that only a small number of sampling points, e.g, two-side perturbations ($2 \times 44+1=89$), is required to efficiently construct a subdomain reduced-order linear model. Four realizations for log-permeability field generated from this small interval {-0.1, 0.1} are illustrated in Fig.3.21. Fig.3.22, Fig.3.23 and Fig.3.24 and Table 3.10 show the results of these numerical experiments, including the updated log-permeability field, iterative value of cost function and the mismatch between observed data and predictions. As Fig.3.24 and Table 3.10, both SD POD-TPWL and FD method are able to converge to a satisfactory minimum cost function value. FD method has relatively high convergence ratio over our SD POD-TPWL,



(a) 'True' Permeability

(b) Initial Permeability

(c) Updated Permeability using $2 \times l_{\boldsymbol{\beta}}$

(d) Updated Permeability using $2 \times l_{\boldsymbol{\beta}}$ via continuing iterations

(e) Updated Permeability using $3\times l_{\boldsymbol{\beta}}$

(f) Updated Permeability using $4\times l_{\boldsymbol{\beta}}$

(g) Updated Permeability using finite-difference method

Figure 3.19: The updated permeability under different number of sampling points for case 2. The stopping criteria for (c),(e), (f) and (g) is $\eta_J = 10^{-4}$, $\eta_{\boldsymbol{\xi}} = 10^{-3}$, N_{max}=50, while The stopping criteria for (d) is $\eta_J = 10^{-5}$, $\eta_{\boldsymbol{\xi}} = 10^{-4}$, N_{max}=50

e.g., 78 and 195 iteration steps with 4 outer-loops separately, and also yields slightly more accurate updated log-permeability. However, our model-reduced approach only needs 112 FOM simulations, while the FD method requires 3510 FOM simulations. Both PCA-based parameterization and POD-based model reduction introduce some inherent approximation errors into this minimization procedure, therefore, the SD POD-TPWL and the FD method in this case generally do not decrease the cost function to the reference

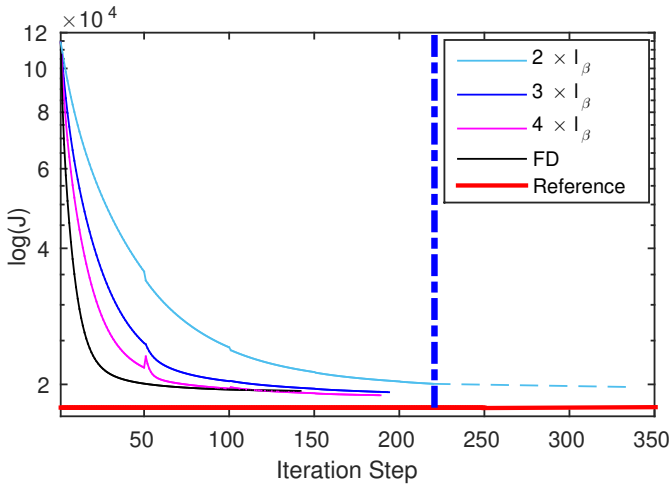


Figure 3.20: Comparison of the cost function value using different number of sampling points for case 2. The extended dash line represents the cost function using a new stopping criterion $\eta_J = 10^{-5}$, $\eta_{\boldsymbol{\xi}} = 10^{-4}$, N_{max}=50

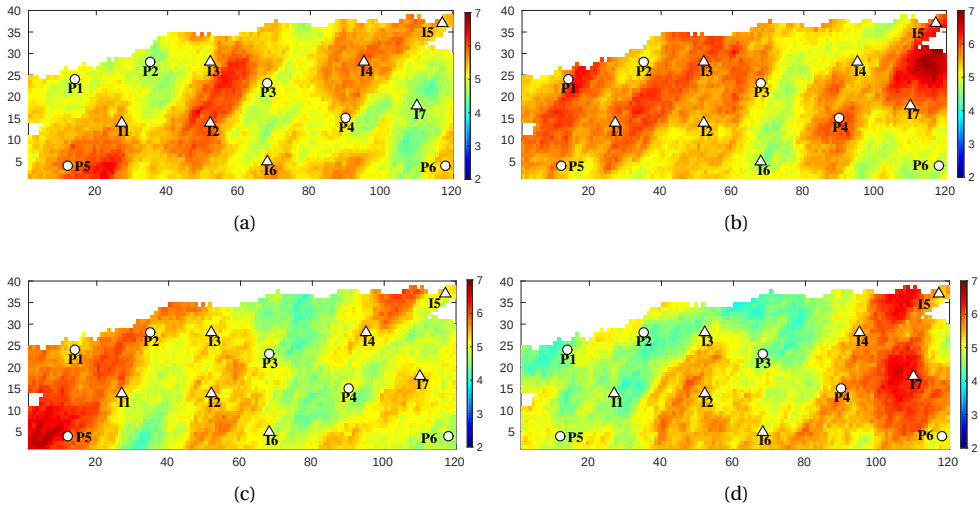
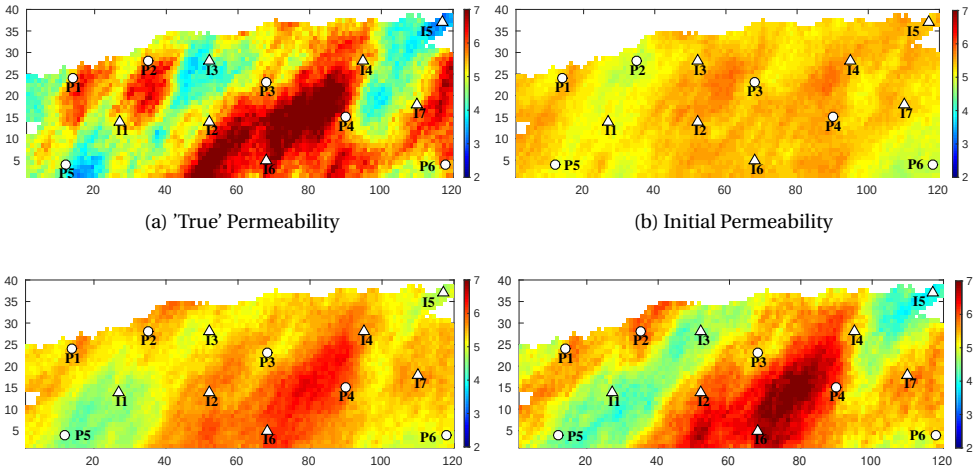


Figure 3.21: Four random realizations of the log-permeability field generated in the interval {-0.1, 0.1} for case 2

value illustrated as the bold red line. In addition, POD-based model reduction also makes our method theoretically less accurate than that of the FD method, which is demonstrated as the black line and blue line in the Fig.3.22.

There is a trade-off between the number of sampling points and the parameter interval. On the one hand, a large parameter interval generally has high possibilities to include the parameter space containing the 'true' model, thus the 'true' solution is likely to be found while needs to use a large number of sampling points. On the other hand, we also can choose a small number of sampling points within a small interval. Although there is

a larger possibility that the 'true' solution is not contained in this interval, a valid solution (based on an acceptable data mismatch) can most likely be found using a relatively small number of sampling points. Therefore, it is highly significant to choose reasonable parameter interval in practice. If we have poor prior information for the 'true' model, a relatively large parameter interval is preferable to provide relatively more accurate results, e.g., finding the 'true' solution, of history matching process, as a compensation, a large number of sampling points are required to implement our SD POD-TPWL. On the contrary, if we have good prior information for the 'true' model, a small parameter interval can be perturbed around the prior parameter field. Using a small number of sampling points enables us to obtain satisfactory history matching results.



(a) 'True' Permeability

(b) Initial Permeability

(c) Updated Permeability using new method

(d) Updated Permeability using finite-difference

Figure 3.22: Comparison of the initial, 'true' and updated log-permeability using interval {-0.1, 0.1} for case 2

Table 3.10: Comparison of subdomain POD-TPWL and FD method using interval {-0.1, 0.1} for case 2

-	Iterations	FOM	$J(\boldsymbol{\xi}) \times 10^5$	e_{obs}	$e_{\boldsymbol{\beta}}$
Initial model	-	-	1.1417	73.5892	0.7078
SD POD-TPWL	195	112	0.1928	11.1802	0.4685
FD	78	3510	0.1935	11.2532	0.3925
'True' model	-	-	0.1777	9.2531	0

3.5.3. COMPUTATIONAL ASPECTS

This section discusses the computational aspects of our proposed adjoint-based reservoir history matching algorithm. The offline computational costs for subdomain POD-TPWL algorithm comprise (1) executing parameterization using eigenvalue decomposition of the covariance matrix, (2) implementing model order reduction using POD in each subdomian, (3) conducting RBF interpolation and computing the derivative matrices. The cost of eigenvalue decomposition and POD is negligible for small models, while it will be-

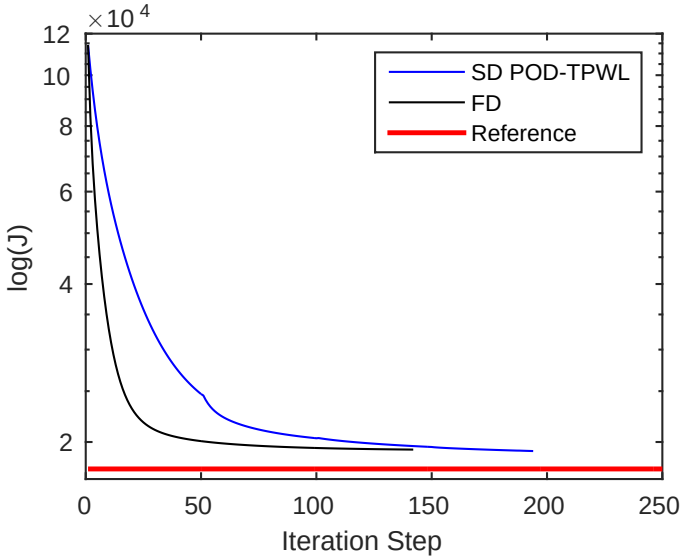


Figure 3.23: Cost function value of SD POD-TPWL and FD method using interval {-0.1, 0.1} for case 2

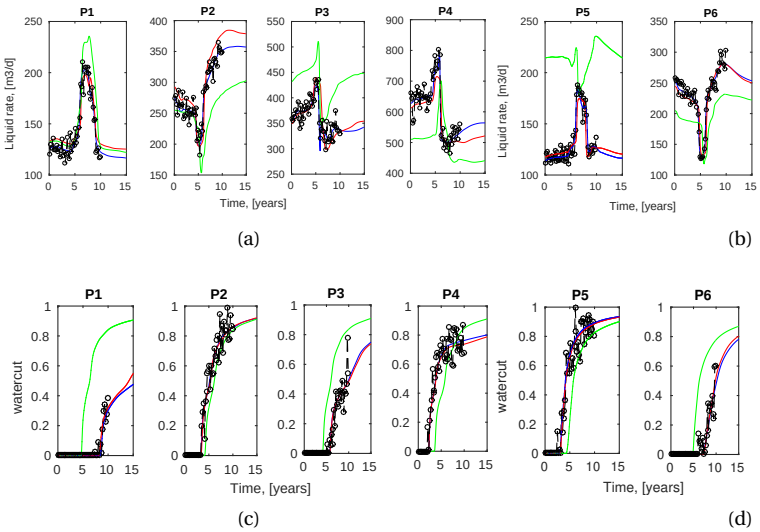


Figure 3.24: Forecast of the liquid rate and water-cut using interval {-0.1, 0.1} for case 2: green line-initial model, blue line-reference model, red line-updated model by subdomain POD-TPWL

come significant for large-scale models. In our cases, the required number of FOM simulations is roughly 2-3 times the number of PCA coefficients, e.g, 54 simulations for the synthetic model, 113 (sampling within a small interval [-0.1, 0.1]) and 199 (sampling within a relative large interval [-1,1]) FOM simulations for the modified version of 2D SAIGUP model, respectively. This process is code non-intrusive without the need of large program-

ming effort. Besides, this process is also easily parallelized. Once available, the costs of running the reduced model are negligible. We should note that the gradient-based reservoir history matching generally requires $O(10^2 - 10^4)$ FOM simulations, thus, an offline cost of $O(10-10^2)$ FOM simulations in these settings is attractive. For large-scale reservoir history matching, the main computational cost is dominated by the required number of FOM simulations. In our proposed method, most part of the FOM simulations is mainly in offline stage, which means that our method is easily implemented.

REFERENCES

[1] H. Jeffreys, B. Jeffreys, and B. Swirles, *Methods of mathematical physics* (Cambridge university press, 1999).

[2] P. Courtier, J.-N. Thépaut, and A. Hollingsworth, *A strategy for operational implementation of 4d-var, using an incremental approach,* Quarterly Journal of the Royal Meteorological Society **120**, 1367 (1994).

[3] C. Liu, Q. Xiao, and B. Wang, *An ensemble-based four-dimensional variational data assimilation scheme. part i: Technical formulation and preliminary test,* Monthly Weather Review **136**, 3363 (2008).

[4] C. Liu, Q. Xiao, and B. Wang, *An ensemble-based four-dimensional variational data assimilation scheme. part ii: Observing system simulation experiments with advanced research wrf (arw),* Monthly Weather Review **137**, 1687 (2009).

[5] S. Frolov and C. H. Bishop, *Localized ensemble-based tangent linear models and their use in propagating hybrid error covariance models,* Monthly Weather Review **144**, 1383 (2016).

[6] C. H. Bishop, S. Frolov, D. R. Allen, D. D. Kuhl, and K. Hoppel, *The local ensemble tangent linear model: an enabler for coupled model 4d-var,* Quarterly Journal of the Royal Meteorological Society **143**, 1009 (2017).

[7] E. Gildin, H. Klie, A. A. Rodriguez, M. F. Wheeler, *et al.*, *Development of low-order controllers for high-order reservoir models and smart wells, SPE Annual Technical Conference and Exhibition*, (2006).

[8] R. Djeddi, A. Kaminsky, and K. Ekici, *Convergence acceleration of fluid dynamics solvers using a reduced-order model,* AIAA Journal **55**, 3059 (2017).

[9] M. A. Cardoso, *Development and application of reduced-order modeling procedures for reservoir simulation* (Stanford University, 2009).

[10] P. Vermeulen and A. Heemink, *Model-reduced variational data assimilation,* Monthly weather review **134**, 2888 (2006).

[11] M. U. Altaf, A. W. Heemink, and M. Verlaan, *Inverse shallow-water flow modeling using model reduction,* International journal for multiscale computational engineering **7** (2009).

[12] M. P. Kaleta, R. G. Hanea, A. W. Heemink, and J.-D. Jansen, *Model-reduced gradient-based history matching,* Computational Geosciences **15**, 135 (2011).

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

[14] J. He, P. Sarma, and L. J. Durlofsky, *Reduced-order flow modeling and geological parameterization for ensemble-based data assimilation,* Computers & geosciences **55**, 54 (2013).

[15] J. He, L. J. Durlofsky, *et al.*, *Reduced-order modeling for compositional simulation by use of trajectory piecewise linearization,* SPE Journal **19**, 858 (2014).

[16] J. He, J. Sætrom, and L. J. Durlofsky, *Enhanced linearized reduced-order models for subsurface flow simulation,* Journal of Computational Physics **230**, 8313 (2011).

[17] S. Trehan and L. J. Durlofsky, *Trajectory piecewise quadratic reduced-order model for subsurface flow, with application to pde-constrained optimization,* Journal of Computational Physics **326**, 446 (2016).

[18] D. R. Jones, *A taxonomy of global optimization methods based on response surfaces,* Journal of global optimization **21**, 345 (2001).

[19] H. Klie *et al.*, *Unlocking fast reservoir predictions via nonintrusive reduced-order models,* in *SPE Reservoir Simulation Symposium* (Society of Petroleum Engineers, 2013).

[20] D. Xiao, F. Fang, C. Pain, I. Navon, and A. Muggeridge, *Non-intrusive reduced order modelling of waterflooding in geologically heterogeneous reservoirs,* in *ECMOR XV-15th European conference on the mathematics of oil recovery* (European Association of Geoscientists & Engineers, 2016) pp. cp–494.

[21] D. Xiao, F. Fang, C. Pain, I. Navon, P. Salinas, and Z. Wang, *Non-intrusive model reduction for a 3d unstructured mesh control volume finite element reservoir model and its application to fluvial channels,* International Journal of Oil, Gas and Coal Technology **19**, 316 (2018).

[22] J. Bruyelle and D. Guérillot, *Neural networks and their derivatives for history matching and reservoir optimization problems,* Computational Geosciences **18**, 549 (2014).

[23] P. P. Chinchapatnam, K. Djidjeli, and P. B. Nair, *Domain decomposition for time-dependent problems using radial based meshless methods,* Numerical Methods for Partial Differential Equations: An International Journal **23**, 38 (2007).

[24] J. S. Przemieniecki, *Matrix structural analysis of substructures,* AIAA Journal **1**, 138 (1963).

[25] X. Bian, Z. Li, and G. E. Karniadakis, *Multi-resolution flow simulations by smoothed particle hydrodynamics via domain decomposition,* Journal of Computational Physics **297**, 132 (2015).

[26] D. J. Lucia, P. I. King, and P. S. Beran, *Reduced order modeling of a two-dimensional flow with moving shocks,* Computers & Fluids **32**, 917 (2003).

[27] J. Baiges, R. Codina, and S. Idelsohn, *A domain decomposition strategy for reduced order models. application to the incompressible navier–stokes equations,* Computer Methods in Applied Mechanics and Engineering **267**, 23 (2013).

[28] D. Amsallem, M. J. Zahr, and C. Farhat, *Nonlinear model order reduction based on local reduced-order bases,* International Journal for Numerical Methods in Engineering **92**, 891 (2012).

[29] S. Chaturantabut, *Temporal localized nonlinear model reduction with a priori error estimate,* Applied Numerical Mathematics **119**, 225 (2017).

[30] J. He and L. J. Durlofsky, *Constraint reduction procedures for reduced-order subsurface flow models based on pod–tpwl,* International Journal for Numerical Methods in Engineering **103**, 1 (2015).

[31] 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).

[32] K. Fukunaga and W. L. Koontz, *Application of the karhunen-loeve expansion to feature selection and ordering,* IEEE Transactions on computers **100**, 311 (1970).

[33] Y. Wu, H. Wang, B. Zhang, and K.-L. Du, *Using radial basis function networks for function approximation and classification,* ISRN Applied Mathematics **2012** (2012).

[34] D. J. Evans, *Parallel sor iterative methods,* Parallel computing **1**, 3 (1984).

[35] S. A. Smolyak, *Quadrature and interpolation formulas for tensor products of certain classes of functions,* in *Doklady Akademii Nauk*, Vol. 148 (Russian Academy of Sciences, 1963) pp. 1042–1045.

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

[37] A. Tarantola, *Inverse problem theory and methods for model parameter estimation* (SIAM, 2005).

[38] J. D. Matthews, J. N. Carter, K. D. Stephen, R. W. Zimmerman, A. Skorstad, T. Manzocchi, and J. A. Howell, *Assessing the effect of geological uncertainty on recovery estimates in shallow-marine reservoirs: the application of reservoir engineering to the saigup project,* Petroleum Geoscience **14**, 35 (2008).

[39] K.-A. Lie, S. Krogstad, I. S. Ligaarden, J. R. Natvig, H. M. Nilsen, and B. Skaflestad, *Open-source matlab implementation of consistent discretisations on complex grids,* Computational Geosciences **16**, 297 (2012).

4

SUBDOMAIN POD-TPWL WITH SMOOTH LOCAL PARAMETERIZATION

This chapter presents a new local parameterization that decouples the computational cost of the algorithm from the number of global principle components and therefore provides attractive scaling for models with a very large number of uncertain parameter patterns. For each subdomain, we restrict to a small number of uncertain local parameter patterns, and consequently the number of full order simulation required for the derivation of the reduced order models can be reduced drastically. The computational effort of the new methodology hardly increases when the number of parameter patterns increases. We apply the new algorithm to a large-scale reservoir model parameter estimation test problem, where only 90 full order model runs are needed to estimate 282 parameter patterns while keeping satisfactory accuracy.

Parts of this chapter have been published in Journal of Computational physics. **Xiao, C.**, Leeuwenburgh, O., Lin, H.X. and Heemink, A., 2021. *Efficient estimation of space varying parameters in numerical models using non-intrusive subdomain reduced order modeling.* Journal of Computational Physics, 424, p.109867.

4.1. INTRODUCTION

Projection-based reduced order model (ROM) approaches have received attention as a way of reducing the computational effort of model-based workflows through dimensionality reduction. These reduced-order model approaches aim primarily at approximating the dynamic response of the full model as accurately as possible. Our objective here, on the other hand, is to effectively and efficiently estimate uncertain full order model parameters through approximation of the adjoint of the original high-dimensional non-linear model.

A non-intrusive POD-based method to build reduced-order linear approximations of high-dimensional non-linear subsurface flow models was proposed by Vermeulen and Heemink [1]. The adjoint of this reduced-order linear model can be easily constructed and the minimization of the model-data mismatch can therefore be handled efficiently. Altaf et al. [2] and Kaleta et al. [3] applied this method to a coastal engineering and reservoir history matching problem, respectively. This algorithm considers any simulator as black box and is completely non-intrusive with respect to the simulation code. However, the model linearization is realized through a perturbation-based finite difference method, which requires a significant number of full-order model simulations and is therefore computationally less attractive for large scale problems with many uncertain parameters. The projection-based ROM methods have restricted their applications on complex dynamic problems mainly due to the intrinsic stability and robustness problems [4]. Besides, projection-based ROMs are highly code-intrusive and their speedup potential is restricted by the strong nonlinearity [5].

In Chapter 3, we have proposed a non-intrusive approach, e.g., subdomain POD-TPWL, that combines dimensionality reduction, piece-wise linearization and domain decomposition, and demonstrated this approach on an example from the field of subsurface reservoir engineering. Compared to the POD-TPWL approach proposed in [6], this subdomain POD-TPWL has two advantages: (1) Instead of taking global basis functions to define the low-order subspace, the snapshots of dynamic states are first partitioned according to the domain decomposition strategy and then local basis functions are obtained from these partitioned snapshots. (2) The derivative matrices required in the process of model linearization are separately approximated in each subdomain using a radial basis function (RBF) interpolation method [7], such that access to the underlying model code is not required. A drawback of that approach was that only the dimension of the model state space was reduced in 'local' subdomains, whereas the number of parameters was still determined by a 'global' Principle Component Analysis (PCA) decomposition of the parameter space over the entire spatial domain. Based on numerical experiments, it was found that the total number of full-order model simulations required for subdomain POD-TPWL is roughly 2-4 times the number of global parameters. This computational cost is still too high for large-scale parameter estimation problems with a large number of uncertain parameters.

In this chapter we therefore propose a new method that incorporates a local parameter decomposition into the subdomain POD-TPWL approach. We will show that the resulting reduction in the dimensionality of the local PCA parameterizations is much larger than that obtained through global PCA parameterization. As a result a much smaller number of full-order model simulations will need to be run to solve the parameter estimation problem. A consequence of the decomposition of the spatial parameter field into spatial non-overlapping subdomains is the potential loss of smoothness. When implementing param-

eterization techniques such as PCA [8], KPCA [9], DCT [10] and DWT [11] in the global domain, it is guaranteed that the reconstructed spatial parameter field is smooth. However, applying these techniques in each subdomain separately to obtain local parametrizations will introduce discontinuities at the boundaries between neighboring subdomains. We therefore introduce an approach that resolves the non-smoothness at the boundaries of neighboring subdomains by projection of the estimated local PCA patterns onto global PCA patterns. This smooth local parameterization simultaneously inherits the advantages of global PCA (e.g., smoothness and differentiability) and of local PCA (computational efficiency). We combine this parameterization method with subdomain POD-TPWL to drastically speed up the solution of parameter estimation problems with many uncertain parameters. The methodology is assessed in detail by analyzing its performance in example applications to petroleum reservoir history matching problems.

The remainder of the chapter is organized as follows. A brief description of subdomain POD-TPWL with global parameterization and its limitation are introduced in Section 2. Section 3 describes the smooth local parameterization method to represent parameter patterns in each subdomain individually while preserving smoothness. The procedure of subdomain POD-TPWL combined with the smooth local parameterization is presented in Section 4. Section 5 describes the basic settings of two case-studies using a 2D reservoir simulation model. Section 6 analyses in detail the results of numerical experiments.

4.2. LIMITATIONS OF SUBDOMAIN POD-TPWL

It can be seen in Chapter 3 that the main idea behind the subdomain POD-TPWL is to approximate the dynamical evolution of reduced states within subdomains by a reduced-order piece-wise linear model analogous. Introducing an extra term to allow for dynamical interaction between each subdomain Ω^d and its neighboring subdomains Ω^{sd}, we postulate that the local POD coefficients $\boldsymbol{\psi}^{d,n}$ in subdomain Ω^d can be approximated by $\hat{\boldsymbol{\psi}}^{d,n}$ as follows

$$\boldsymbol{\psi}^{d,n} \approx \hat{\boldsymbol{\psi}}^{d,n} = \boldsymbol{\psi}^{d,n}_{tr} + \mathbf{E}^{d,n}_{\boldsymbol{\psi}_{tr}}(\boldsymbol{\psi}^{d,n-1} - \boldsymbol{\psi}^{d,n-1}_{tr}) + \mathbf{E}^{sd,n}_{\boldsymbol{\psi}_{tr}}(\boldsymbol{\psi}^{sd,n} - \boldsymbol{\psi}^{sd,n}_{tr}) + \mathbf{G}^{d,n}_{\boldsymbol{\xi}_{tr}}(\boldsymbol{\xi} - \boldsymbol{\xi}_{tr}) \quad (4.1)$$

where

$$\mathbf{E}^{d,n}_{\boldsymbol{\psi}_{tr}} = \frac{\partial \pounds^{d,n}}{\partial \boldsymbol{\psi}^{d,n-1}_{tr}}, \quad \mathbf{E}^{sd,n}_{\boldsymbol{\psi}_{tr}} = \frac{\partial \pounds^{d,n}}{\partial \boldsymbol{\psi}^{sd,n}_{tr}}, \quad \mathbf{G}^{d,n}_{\boldsymbol{\xi}_{tr}} = \frac{\partial \pounds^{d,n}}{\partial \boldsymbol{\xi}_{tr}} \quad (4.2)$$

Analogously, the measurements $\mathbf{y}^{d,m}$ taken in subdomain Ω^d are approximated by $\hat{\mathbf{y}}^{d,m}$ as follows

$$\mathbf{y}^{d,m} \approx \hat{\mathbf{y}}^{d,m} = \mathbf{y}^{d,m}_{tr} + \mathbf{A}^{d,m}_{\boldsymbol{\psi}_{tr}}(\boldsymbol{\psi}^{d,m} - \boldsymbol{\psi}^{d,m}_{tr}) + \mathbf{B}^{d,m}_{\boldsymbol{\xi}_{tr}}(\boldsymbol{\xi} - \boldsymbol{\xi}_{tr}) \quad (4.3)$$

with

$$\mathbf{A}^{d,m}_{\boldsymbol{\psi}_{tr}} = \frac{\partial \hbar^{d,m}}{\partial \boldsymbol{\psi}^{d,m}_{tr}}, \quad \mathbf{B}^{d,m}_{\boldsymbol{\xi}_{tr}} = \frac{\partial \hbar^{d,m}}{\partial \boldsymbol{\xi}_{tr}} \quad (4.4)$$

Above we have introduced new, so far undefined, functions $\pounds^{d,n}(\boldsymbol{\psi}^{d,n-1}, \boldsymbol{\psi}^{sd,n}, \boldsymbol{\xi})$ and $\hbar^{d,m}(\boldsymbol{\psi}^{d,m}, \boldsymbol{\xi})$ of the local reduced states $\boldsymbol{\psi}^{d,n}$ and $\boldsymbol{\psi}^{sd,n}$ and of the global reduced parameters $\boldsymbol{\xi}$. Note that only the derivatives of these models appear in our proposed subdomain formulation and that we are in principle free to choose the form of these functions. We apply the reasoning that we would like to be able to evaluate their derivatives (Eq.4.2) efficiently and that, when used in Eq.4.1, they deliver the best possible approximation $\hat{\mathbf{y}}^{d,m}$ of the full states $\boldsymbol{\psi}^{d,n}$.

We proposed to define $£^{d,n}$ and $\hbar^{d,m}$ in terms of a radial basis function (RBF) interpolation model. The RBF interpolation model can be represented as a linear combination of M radial basis functions

$$£^{d,n}(\boldsymbol{\psi}^{d,n-1}, \boldsymbol{\psi}^{sd,n}, \boldsymbol{\xi}) = \sum_{j=1}^{M} \boldsymbol{\omega}_j^{d,n} \times \theta(||(\boldsymbol{\psi}^{d,n-1}, \boldsymbol{\psi}^{sd,n}, \boldsymbol{\xi}) - (\boldsymbol{\psi}_j^{d,n-1}, \boldsymbol{\psi}_j^{sd,n}, \boldsymbol{\xi}_j)||) \tag{4.5}$$

and similarly,

$$\hbar^{d,m}(\boldsymbol{\psi}^{d,m}, \boldsymbol{\xi}) = \sum_{j=1}^{M} \boldsymbol{\varepsilon}_j^{d,m} \times \theta(\|(\boldsymbol{\psi}^{d,m}, \boldsymbol{\xi}) - (\boldsymbol{\psi}_j^{d,m}, \boldsymbol{\xi}_j)\|) \tag{4.6}$$

4

respectively, where the subscripted index j identifies one of M training runs, which significantly depends on the number of inputs, e.g., $\boldsymbol{\psi}_j^d$ and $\boldsymbol{\xi}_j$. A discussion of how to choose M was provided in Chapter 3. Subdomain POD-TPWL uses analytical RBFs $£^{d,n}$ and $\hbar^{d,m}$ to approximate the time series of full model state snapshots, which also allows us to derive the derivative matrices $\mathbf{E}_{\boldsymbol{\psi}_{tr}}^{d,n}$, $\mathbf{E}_{\boldsymbol{\psi}_{tr}}^{sd,n}$, $\mathbf{G}_{\boldsymbol{\xi}_{tr}}^{d,n}$, $\mathbf{A}_{\boldsymbol{\psi}_{tr}}^{d,m}$, and $\mathbf{B}_{\boldsymbol{\xi}_{tr}}^{d,m}$ analytically. More information on the specific formula of the radial basis function θ, and the determination of the weighting coefficients $\boldsymbol{\omega}$ can be found in Chapter 3.

It can be seen from Eq.4.1 and Eq.4.3 that the subdomain POD-TPWL system is defined in terms of local POD patterns of the model dynamics, and global patterns representing the uncertain parameter space. Our previous numerical experiments have shown that the total number of full-order model simulations required to construct the subdomain POD-TPWL model is approximately 2-4 times the number of global parameter patterns. Decreasing the number of preserved global parameter patterns will reduce the computational cost, but will result in a loss of accuracy. The dependence of the computational cost on the dimension of the global reduced parameter space will inevitably limit the application of subdomain POD-TPWL to inverse problems with a limited number of uncertain parameters or with parameters fields containing smooth and large-scale spatial correlations.

From a computational point of view, implementing subdomain POD-TPWL in terms of local parameter patterns in each subdomain, similar to the treatment of the dynamic states, therefore seems attractive. Firstly, the number of local patterns in each individual subdomain will be smaller than the number of global patterns, this description might be almost valid in some practical problems, secondly, the computational cost of the parameter estimation procedure is proportional to the typical number of local patterns in any one subdomain, this statement has been partially verified in Chapter 3. where the global parameter patterns are optimized. Such an approach we aim at proposing in this work would rely on the validity of assumption that it is possible to reconstruct global parameter fields from the local parameter solutions that satisfy criteria for acceptable solutions in the global domain. In the following section we will introduce a smooth local parameter reduction method that addresses the computational limitations associated with the use of global parameter reduction and produces smooth global space parameter patterns.

In the following section we will introduce a smooth local parameter reduction method that addresses the computational limitations associated with the use of global parameter reduction but satisfies the three assumptions above.

4.3. SMOOTH LOCAL PARAMETERIZATION

In this section, we first discuss global PCA and local PCA based representations of spatial parameter patterns in the global domain and in the subdomains, respectively. The disadvantages of each are investigated to motivate the development of the smooth local parameterization (SLP).

4.3.1. GLOBAL AND LOCAL REPRESENTATIONS OF SPATIAL PARAMETERS

In analogy to the global PCA-based parameter reduction described above, the local parameter vector $\boldsymbol{\beta}_d$ in subdomain Ω^d can be expressed as

$$\boldsymbol{\beta}^d = \boldsymbol{\beta}_m^d + \boldsymbol{\Phi}_{\boldsymbol{\beta}}^d \boldsymbol{\xi}^d, \quad d = 1, \cdots, S \tag{4.7}$$

where $\boldsymbol{\Phi}_{\boldsymbol{\beta}}^d$ is the basis matrix that projects the high-order parameter patterns onto the low-order subspace of dimension N_l^d for subdomain d. Again, an energy criterion or a basis optimization procedure can be employed to reduce the number of basis vectors [12].

In order to illustrate the global and local PCA procedures described above to reconstruct parameter fields we generate $N_r = 1000$ random Gaussian realizations of the model parameter vector, representing the log-permeability for all gridblocks of the 2D reservoir model that will be used in the numerical experiments presented later. The model is decomposed into 20 non-overlapping subdomains (4 subdomains in one direction and 5 subdomains in the second direction) as illustrated in Fig.4.2(a). $N_G = 48$ global PCA patterns are required to construct a reduced order representation that retains 95% of the energy (variability) in the original 1000 realizations. The number of local PCA patterns N_l^d for each subdomain obtained by the same procedure and the same energy criterion (but now applied in each subdomain separately) is summarized in Fig.4.1. Results for other decompositions are summarized in Table 4.1. Decomposition with a larger number of subdomains results in fewer local PCA coefficients in each subdomain. The maximum number of local patterns in any one subdomain is substantially smaller than N_G.

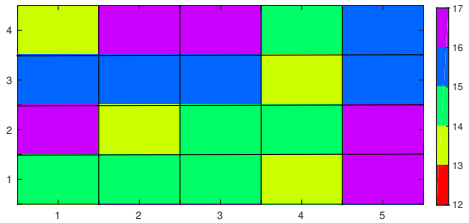


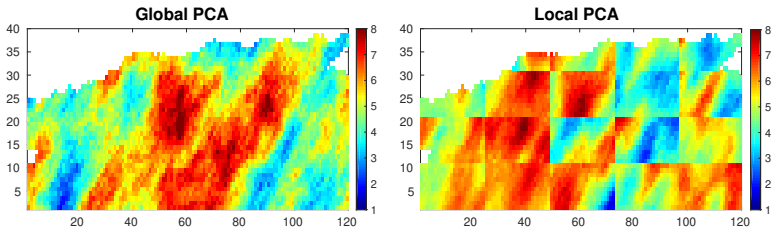
Figure 4.1: The number of local PCA coefficients in each subdomain for a domain decomposition consisting of 4 × 5 subdomains

The first two panels of Fig.4.2 show two examples of parameter patterns for random realizations of global ($\boldsymbol{\xi}$) and local ($\boldsymbol{\xi}^d$) PCA coefficients. It can be seen that sampling arbitrary local PCA coefficients will result in a non-smooth reconstructed global parameter field, while global PCA produces a smooth parameter field in a relatively high-order subspace. The maximum number of local patterns required in any subdomain is much smaller, however, than the number of global patterns that is needed. This motivates us to develop a smooth local parameterization that exploits the advantages of global PCA

Table 4.1: Total number of local PCA and global PCA patterns, and the maximum number of local PCA patterns among all subdomains corresponding to three different domain decomposition strategies, e.g., 3×4, 4×5 and 5×6, respectively.

Domain decomposition	SLP		Global PCA
	$N_L = \sum_{d=1}^{S} N_l^d$	$\max\{N_l^1,...,N_l^d,...,N_l^S\}$	N_G
3 × 4	236	18	
4 × 5	275	15	48
5 × 6	312	12	

(smooth representation) and local PCA (computationally efficiency). We will demonstrate the gain in computational efficiency in the following sections.



(a) (b)

Figure 4.2: Example parameter reconstructions using (a) global PCA, (b) local PCA, respectively.

4.3.2. SMOOTH LOCAL REPRESENTATION OF SPATIAL PARAMETERS

The results shown in Fig. 4.2(b) suggest that direct reconstruction of the global parameter field from local subdomain parameter solutions, and then stitching all subdomains together, tends to produce non-smooth results. We therefore add a post-processing step that finds the best matching global PCA reconstruction. This solution is found by minimizing an objective function consisting of the sum of squared differences between the local solution based reconstruction in terms of $\boldsymbol{\xi}^d$ and a global reconstruction in terms of global PCA coefficients $\boldsymbol{\xi}$,

$$J(\boldsymbol{\xi}) = \frac{1}{2}[\sum_{d=1}^{S} \mathbf{T}^d(\boldsymbol{\beta}_m^d + \boldsymbol{\Phi}_{\boldsymbol{\beta}}^d \boldsymbol{\xi}^d) - \boldsymbol{\beta}_m - \boldsymbol{\Phi}_{\boldsymbol{\beta}}\boldsymbol{\xi}]^T \mathbf{C}^{-1} [\sum_{d=1}^{S} \mathbf{T}^d(\boldsymbol{\beta}_m^d + \boldsymbol{\Phi}_{\boldsymbol{\beta}}^d \boldsymbol{\xi}^d) - \boldsymbol{\beta}_m - \boldsymbol{\Phi}_{\boldsymbol{\beta}}\boldsymbol{\xi}] + \frac{1}{2}\boldsymbol{\xi}^T\boldsymbol{\xi} \quad (4.8)$$

where, $\mathbf{T}^d \in R^{N_g \times N_l^d}$ is a transformation matrix with elements equal to 0 or 1 that maps a grid position in subdomain Ω^d to the corresponding grid position in the global domain, and N_l^d denotes the total number of local parameter patterns in subdomain Ω^d. The squared differences between global and local reconstructions are weighted by the matrix $\mathbf{C}$, which is the covariance matrix quantifying the prior variability of the Gaussian-distributed random parameters $\boldsymbol{\beta}$. A regularization term is included to constrain the magnitude of solutions for $\boldsymbol{\xi}$. The minimizing solution is obtained by setting the derivative of

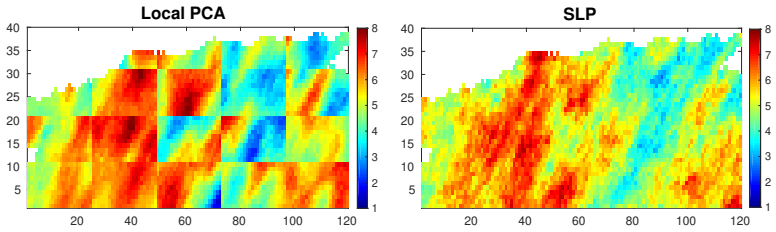
$J(\boldsymbol{\xi})$ with respect to $\boldsymbol{\xi}$ equal to zero, resulting in

$$\boldsymbol{\xi} = \frac{1}{2}\boldsymbol{\Phi}_{\boldsymbol{\beta}}^{T}\mathbf{C}^{-1}\sum_{d=1}^{S}\mathbf{T}^{d}\boldsymbol{\Phi}_{\boldsymbol{\beta}}^{d}\boldsymbol{\xi}^{d} \tag{4.9}$$

The parameter field can then be reconstructed by insertion of the solution in the formula of PCA solution, resulting in

$$\boldsymbol{\beta} = \boldsymbol{\beta}_{m} + \frac{1}{2}\boldsymbol{\Phi}_{\boldsymbol{\beta}}\boldsymbol{\Phi}_{\boldsymbol{\beta}}^{T}\mathbf{C}^{-1}\sum_{d=1}^{S}\mathbf{T}^{d}\boldsymbol{\Phi}_{\boldsymbol{\beta}}^{d}\boldsymbol{\xi}^{d} \tag{4.10}$$

It can be seen that smooth solutions for the parameter field can be expressed directly in terms of the local coefficients $\boldsymbol{\xi}^{d}$. Furthermore, this smooth local parameterization (SLP) is a differentiable and linear transformation, which makes the parameter estimation problem suitable for use of gradient-based minimization. The last panel of Fig.4.3 shows reconstructed parameter fields using proposed SLP approach. Table 4.1 summarizes the maximum number of local PCA coefficients, e.g., max $\{N_l^d\}$ and total number of local PCA patterns, e.g., N_L=$\sum_{d=1}^{S} N_l^d$, d= 1,2,...,S, among all subdomains for different domain decomposition strategies. Decomposition with a larger number of subdomains results in fewer local PCA coefficients in each subdomain.



(a) (b)

Figure 4.3: Illustration of parameter reconstruction, left: local PCA; right: SLP

A final question to be addressed is the number of local PCA patterns that need to be retained. In order to guarantee smoothness and full reconstruction of the global solution space, many local PCA patterns may be required. However, previous experiments [13] have indicated that the overall computational cost is proportional to the number of PCA patterns. We will discuss here a procedure to estimate the minimum number of required patterns that delivers efficiency and guarantees a desired accuracy.

Eq.4.9 can be written in a compact form as follows

$$\boldsymbol{\xi} = \mathbf{T}_{GL}\,\boldsymbol{\xi}_{L} \tag{4.11}$$

where $\boldsymbol{\xi}_{L} = [\boldsymbol{\xi}^{1\,T},...,\boldsymbol{\xi}^{S\,T}]^{T}$. $\mathbf{T}_{GL} \in R^{N_G \times N_L}$ is a transformation matrix for converting local PCA coefficients into global PCA coefficients, which, after unfolding, can be expressed as

$$\mathbf{T}_{GL} = \frac{1}{2}\boldsymbol{\Phi}_{\boldsymbol{\beta}}^{T}\mathbf{C}^{-1}[\mathbf{T}^{1}\boldsymbol{\Phi}_{\boldsymbol{\beta}}^{1}, \mathbf{T}^{2}\boldsymbol{\Phi}_{\boldsymbol{\beta}}^{2}, \cdots, \mathbf{T}^{d}\boldsymbol{\Phi}_{\boldsymbol{\beta}}^{d}, \cdots, \mathbf{T}^{S}\boldsymbol{\Phi}_{\boldsymbol{\beta}}^{S}] \tag{4.12}$$

The global projection basis matrix $\mathbf{\Phi}_{\boldsymbol{\beta}}$ can be written in unfolded form as follows

$$\mathbf{\Phi}_{\boldsymbol{\beta}} = [\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \cdots, \boldsymbol{\phi}_i, \cdots, \boldsymbol{\phi}_{N_G}] \tag{4.13}$$

where $\boldsymbol{\phi}_i \in R^{N_g \times 1}$, $i \in \{1, 2, \cdots, N_G\}$ is one global basis vector (we have dropped the subscript β on the individual basis vectors for notational clarity). Defining $\mathbf{e}_i$ as the unit column vector with a 1 at position i and zeros elsewhere, each individual global basis vector can be extracted as follows

$$\boldsymbol{\phi}_i = \mathbf{\Phi}_{\boldsymbol{\beta}}\, \mathbf{e}_i, \quad i = 1, 2, ..., N_G \tag{4.14}$$

The solution space for the full parameter field is spanned by the global PCA patterns. SLP must therefore be able to produce any solution for the global PCA coefficients $\boldsymbol{\xi}$. In another words, the following set of equations should have solutions $\boldsymbol{\xi}_L^i$,

$$\mathbf{T}_{GL} \boldsymbol{\xi}_L^i = \mathbf{e}_i, \quad i = 1, 2, \cdots, N_G \tag{4.15}$$

The transformation matrix $\mathbf{T}_{GL}$ should be full-row rank, which implies the total number of local PCA patterns N_L must be equal to or larger than that of global PCA patterns N_G. The rank of transformation matrix $\mathbf{T}_{GL}$ cannot be explicitly determined from Eq.4.12. Therefore, we propose a numerical procedure to determine a minimum for the total number of local patterns N_L. We assume that $\boldsymbol{\xi}_L^i(N_L)$ and $\boldsymbol{\phi}_i^*(N_L)$ as functions of preserved number of local PCA coefficients N_L are the solution of the ith equation and the reconstructed i the global PCA basis vector, respectively. We define a root mean square error $RMSE_i(N_L)$ to quantitatively characterize the accuracy of the reconstructed global PCA basis vector $\boldsymbol{\phi}_i^*$. The total $RMSE(N_L)$ is the average root mean square error of all global PCA basis vector reconstructions. The minimum N_L is determined by minimizing the total $RMSE(N_L)$ as follows

$$N_L = \arg\min_{N_L} RMSE(N_L) = \arg\min_{N_L} \sum_{i=1}^{N_G} RMSE_i(N_L) \tag{4.16}$$

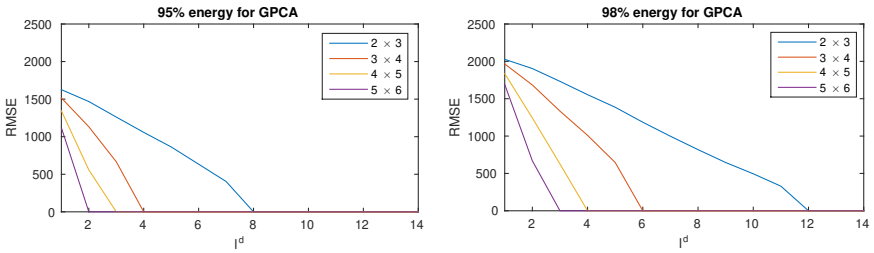
where

$$RMSE_i(N_L) = \|\boldsymbol{\phi}_i - \boldsymbol{\phi}_i^*(N_L)\|_2 = \|\boldsymbol{\phi}_i - \mathbf{\Phi}_{\boldsymbol{\beta}} \times \mathbf{T}_{GL} \boldsymbol{\xi}_L^i(N_L)\|_2 \tag{4.17}$$

We solve this problem by simply evaluating the cost function (Eq.4.16) for increasing values of N_L until a minimum is found. Fig.4.4 shows the evolution of the $RMSE(N_L)$ as a function of the number of local PCA patterns N_L for 2 different global energy criteria. It can be seen that a minimum N_L can been obtained that makes $RMSE(N_L)$ equal to zero. Table 4.2 summarizes the minimum number of local PCA patterns corresponding to different domain decomposition strategies. We can confirm that the minimum total number of local PCA patterns N_L is equal to or slightly larger than the number of global PCA patterns N_G. Fig.4.5 and Fig.4.6 show the 1st global basis vector $\boldsymbol{\phi}_1$, 72nd global basis vector $\boldsymbol{\phi}_{72}$ and their corresponding reconstructed $\boldsymbol{\phi}_1^*$ and $\boldsymbol{\phi}_{72}^*$ when different number of local PCA patterns are retained. The global basis vectors are perfectly reconstructed as long as we retain at least the minimum number of local PCA patterns. The accuracy of the parameter estimation will be investigated in the next section. This analysis only show that in case we have a sufficient number of local PCA patterns, the reconstructed local PCAs contain information about all global PCA patterns. It does not guarantee that the reconstructed global PCAs are always accurate.

Table 4.2: The minimum number of local PCA patterns corresponding to different domain decomposition strategies when $RMSE = 0$.

Energy for global PCA	95%		98%	
N_G	48		72	
Domain Decomposition	$\min\{N_l^1,...,N_l^d,...,N_l^S\}$	N_L	$\min\{N_l^1,...,N_l^d,...,N_l^S\}$	N_L
2×3	8	48	12	72
3×4	4	48	6	72
4×5	3	60	4	80
5×6	2	60	3	90



(a) (b)

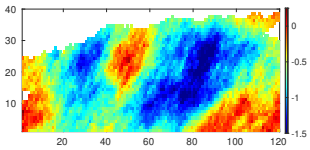
Figure 4.4: The $RMSE$ for different number of local PCA patterns in each subdomain. Four domain decomposition strategies are considered, consisting of 2×3, 3×4, 4×5 and 5×6 subdomains respectively.

4.4. ADJOINT-BASED HISTORY MATCHING WITH SMOOTH LOCAL PARAMETERIZATION

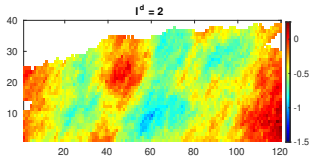
We can generate reduced-order linear models for each large subdomain individually. On the contrary, each small subdomain has to generate reduced-order linear models by considering the effects of the neighboring subdomains. This will increase the number of interpolation variables used in the RBF and hence results in additional computational cost. To speed up model linearization, we assume that each subdomain is sufficient "large "so that it can be handled individually. We should note that it will never be possible to easily clarify these local dependencies in realistic applications, and the definition of "large "or "small "is subjective. An inappropriate domain decomposition strategy, e.g., too "large "or too "small "subdomains, adversely causes spurious long-distance dependency or cuts off real dependencies among neighboring subdomains and hence deteriorates the quality of the reduced-order models. We will provide a comprehensive analysis of sensitivity with respect to the domain decomposition strategy in the section with numerical experiments.

After integrating subdomain POD-TPWL with smooth local parameterization, which is referred to as LSPT, the local POD coefficients $\hat{\boldsymbol{\psi}}^{d,n}$ of subdomain Ω^d are reformulated by directly replacing the variables $\boldsymbol{\xi}$ and $\boldsymbol{\xi}_{tr}$ defined in Eq.4.1 - Eq.4.6 with the new variables $\boldsymbol{\xi}^d$ and $\boldsymbol{\xi}_{tr}^d$. LSPT consists of an off-line stage and an on-line stage. (1) The off-line stage constructs SLP and reduced-order linear models for each subdomain. (2) The on-line stage implements LSPT given a set of new parameters.

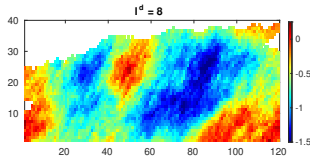
The proposed LSPT approach can be incorporated into a parameter estimation algorithm based on the adjoint method. As an approximation to the original objective function



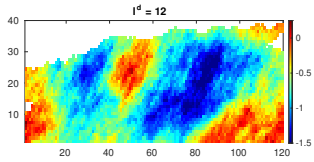
(a) 1st Basis vector $\boldsymbol{\nu}_1$



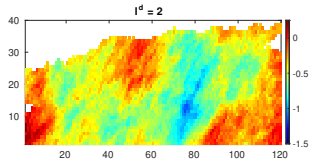
(b) 2 × 3



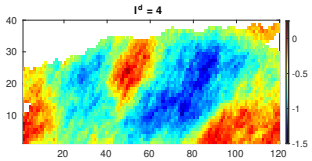
(c) 2 × 3



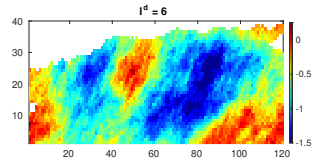
(d) 2 × 3



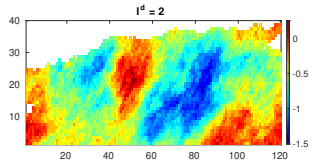
(e) 3 × 4



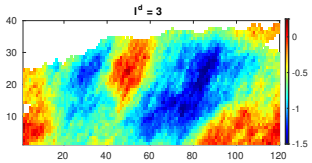
(f) 3 × 4



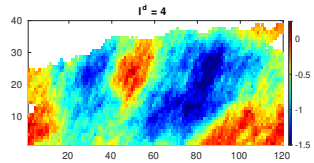
(g) 3 × 4



(h) 4 × 5



(i) 4 × 5



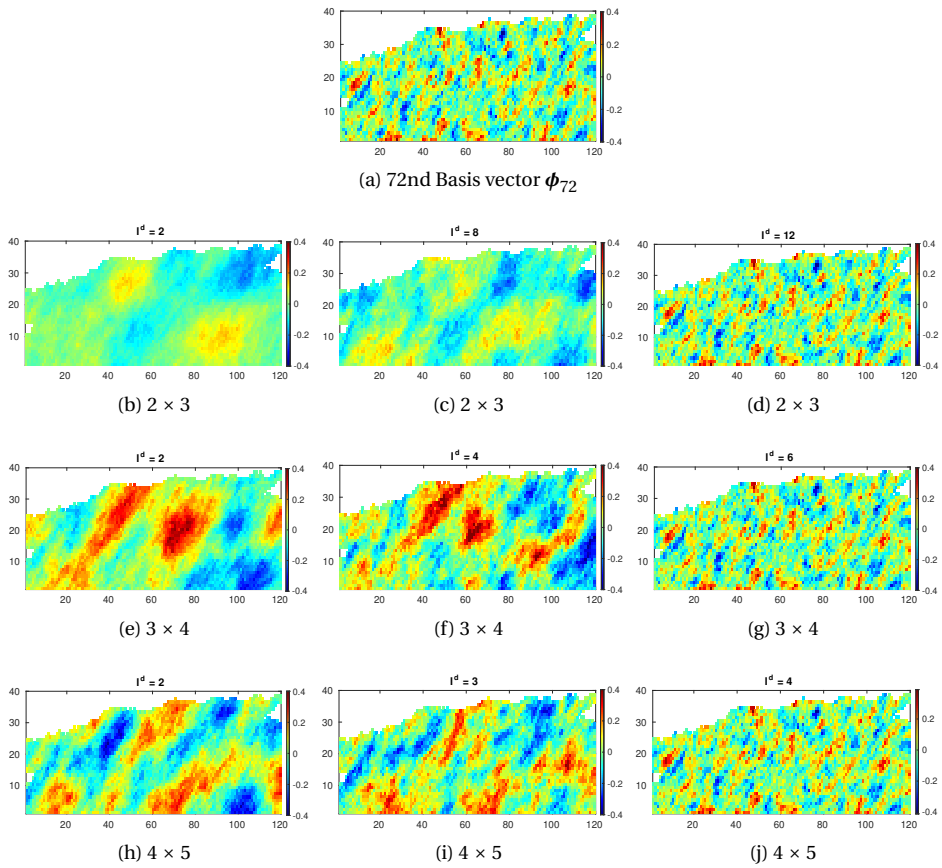
(j) 4 × 5

Figure 4.5: The reconstructed 1st global basis vector $\boldsymbol{\phi}_1$ using different domain decomposition strategies and different number of local PCA patterns in each subdomain. Four domain decomposition strategies are considered, consisting of 2×3, 3×4, 4×5 and 5×6 subdomains respectively.

J, a new objective function J_{ROM} computed using reduced-order models is defined as

$$J_{ROM}(\boldsymbol{\xi}_L) = \frac{1}{2}[\sum_{d=1}^{S} \mathbf{T}^d(\boldsymbol{\beta}_m^d + \boldsymbol{\Phi}_{\boldsymbol{\beta}}^d \boldsymbol{\xi}^d) - \boldsymbol{\beta}_m]^T \mathbf{C}^{-1} [\sum_{d=1}^{S} \mathbf{T}^d(\boldsymbol{\beta}_m^d + \boldsymbol{\Phi}_{\boldsymbol{\beta}}^d \boldsymbol{\xi}^d) - \boldsymbol{\beta}_m] + \frac{1}{2} \sum_{d=1}^{S} \sum_{m=1}^{N_{obs}} [\mathbf{d}_{obs}^{d,m} - \hat{\mathbf{y}}^{d,m}]^T (\mathbf{R}^m)^{-1} [\mathbf{d}_{obs}^{d,m} - \hat{\mathbf{y}}^{d,m}] \tag{4.18}$$

We introduce an adjoint model to compute the gradient of the objective function J_{ROM} with respect to local PCA coefficients $\boldsymbol{\xi}_L$ where we follow the mathematical derivation provided in Chapter 3. A modified objective function $\hat{J_{ROM}}(\boldsymbol{\xi}_L)$ is obtained by adjoining the



(a) 72nd Basis vector $\boldsymbol{\phi}_{72}$

(b) 2 × 3 (c) 2 × 3 (d) 2 × 3

(e) 3 × 4 (f) 3 × 4 (g) 3 × 4

(h) 4 × 5 (i) 4 × 5 (j) 4 × 5

Figure 4.6: The reconstructed 72nd global basis vector $\boldsymbol{\phi}_{72}$ using different domain decomposition strategies and different number of local PCA patterns in each subdomain. Four domain decomposition strategies are considered, consisting of 2×3, 3×4, 4×5 and 5×6 subdomains respectively.

reduced-order linear model, resulting in

$$\begin{aligned}\hat{J}_{ROM}(\boldsymbol{\xi}_L) = J_{ROM}(\boldsymbol{\xi}_L) \\ + \sum_{d=1}^{S}\sum_{n=1}^{N}[\hat{\boldsymbol{\psi}}^{d,n} - \boldsymbol{\psi}_{tr}^{d,n} - \mathbf{E}_{\boldsymbol{\psi}_{tr}}^{d,n}(\boldsymbol{\psi}^{d,n-1} - \boldsymbol{\psi}_{tr}^{d,n-1}) - \mathbf{E}_{\boldsymbol{\psi}_{tr}}^{sd,n}(\boldsymbol{\psi}^{sd,n} - \boldsymbol{\psi}_{tr}^{sd,n}) \\ - \mathbf{G}_{\boldsymbol{\xi}_{tr}^d}^{n}(\boldsymbol{\xi}^d - \boldsymbol{\xi}_{tr}^d)]^T \boldsymbol{\lambda}^{d,n}\end{aligned} \tag{4.19}$$

The gradient of J_{ROM} with respect to $\boldsymbol{\xi}^d$ for each subdomain Ω^d is derived as

$$\frac{dJ_{ROM}(\boldsymbol{\xi}_L)}{d\boldsymbol{\xi}^d} = \frac{d\hat{J}_{ROM}(\boldsymbol{\xi}_L)}{d\boldsymbol{\xi}^d} - \sum_{n=1}^{N}[\mathbf{G}_{\boldsymbol{\xi}_{tr}^d}^{n}]^T \boldsymbol{\lambda}^{d,n} \tag{4.20}$$

where

$$\frac{dJ_{ROM}(\boldsymbol{\xi}_L)}{d\boldsymbol{\xi}^d} = (\boldsymbol{\Phi}_{\boldsymbol{\beta}}^d \mathbf{T}^d)^T \mathbf{C}^{-1}[\mathbf{T}^d(\boldsymbol{\beta}_m^d + \boldsymbol{\Phi}_{\boldsymbol{\beta}}^d \boldsymbol{\xi}^d) - \boldsymbol{\beta}_m] - \sum_{m=1}^{N_0} [\mathbf{B}_{\boldsymbol{\xi}_{tr}}^{d,m}]^T \mathbf{R}_m{}^{-1}[\mathbf{d}_{obs}^{d,m} - \mathbf{y}_{tr}^{d,m} - \mathbf{A}_{\boldsymbol{\psi}_{tr}}^{d,m}(\boldsymbol{\psi}^{d,m} - \boldsymbol{\psi}_{tr}^{d,m}) - \mathbf{B}_{\boldsymbol{\xi}_{tr}^d}^m(\boldsymbol{\xi}^d - \boldsymbol{\xi}_{tr}^d)] \quad (4.21)$$

The adjoint model is expressed in terms of the Lagrange multipliers $\boldsymbol{\lambda}^{d,n}$ for subdomain Ω^d and is given by

$$[\mathbf{I} - (\mathbf{E}_{\boldsymbol{\psi}_{tr}}^{d,n})^T]\boldsymbol{\lambda}^{d,n} = [\mathbf{A}_{\boldsymbol{\psi}_{tr}}^{d,n}]^T \mathbf{R}_n{}^{-1}[\mathbf{d}_{obs}^{d,n} - \mathbf{y}_{tr}^{d,n} - \mathbf{A}_{\boldsymbol{\psi}_{tr}}^{d,n}(\boldsymbol{\psi}^{d,n} - \boldsymbol{\psi}_{tr}^{d,n}) - \mathbf{B}_{\boldsymbol{\xi}_{tr}^d}^n(\boldsymbol{\xi}^d - \boldsymbol{\xi}_{tr}^d)] + [\mathbf{E}_{\boldsymbol{\psi}_{tr}}^{sd,n}]^T \boldsymbol{\lambda}^{d,n+1} \quad (4.22)$$

for $n = N, \cdots, 1$ with an ending condition $\boldsymbol{\lambda}^{d,N+1} = 0$. The solution of the adjoint model Eq.4.22, together with the solution of Eq.4.21, can be used in Eq.4.20 to obtain the desired total derivative with respect to the local PCA coefficients in subdomain Ω^d. The full gradient with respective to all local PCA coefficients can be obtained as

$$\nabla J = [\frac{dJ_{ROM}(\boldsymbol{\xi}_L)}{d\boldsymbol{\xi}^1}, ..., \frac{dJ_{ROM}(\boldsymbol{\xi}_L)}{d\boldsymbol{\xi}^d}, ..., \frac{dJ_{ROM}(\boldsymbol{\xi}_L)}{d\boldsymbol{\xi}^S}]. \quad (4.23)$$

Once the gradient ∇J_k at the k^{th} iteration step is available, a steepest descent update of coefficients $\boldsymbol{\xi}_{k+1}^L$ is obtained by

$$\boldsymbol{\xi}_L^{k+1} = \boldsymbol{\xi}_L^k - \alpha_k \frac{\nabla J_k}{\|\nabla J_k\|_\infty} \quad (4.24)$$

Where α_k is the step length at the k^{th} iteration step [14]. The minimization process terminates when either one of the following three stopping criteria is satisfied: There is no significant change in the objective function,

$$\frac{|J_{ROM}(\boldsymbol{\xi}_L^{k+1}) - J_{ROM}(\boldsymbol{\xi}_L^k)|}{max\{|J_{ROM}(\boldsymbol{\xi}_L^{k+1})|, 1\}} < \eta_{J_{ROM}} \quad , \quad (4.25)$$

there is no significant change in the parameter estimate,

$$\frac{|\boldsymbol{\xi}_L^{k+1} - \boldsymbol{\xi}_L^k|}{max\{|\boldsymbol{\xi}_L^{k+1}|, 1\}} < \eta_{\boldsymbol{\xi}^L} \quad , \quad (4.26)$$

or the maximum number of iterations has been reached,

$$k = N_{max} \quad , \quad (4.27)$$

where $\eta_{J_{ROM}}$, $\eta_{\boldsymbol{\xi}^L}$ and N_{max} denote predefined constants and maximum iterative steps respectively. In our experiments $\eta_{J_{ROM}} = 10^{-4}$, $\eta_{\boldsymbol{\xi}^L} = 10^{-3}$. More strict criteria could possibly lead to more accurate results but we did not investigate this. Since the simulation of the reduced-order model is very cheap, we do not limit the maximum number of iterations N_{max} for the inner-loop. The initial step size α_0=0.1. Once the objective function increases, the step size is divided by 2 to improve convergence.

Oliver et al. [15] discussed the expected range of the optimal objective function. If the relationship between the simulated data and the parameters is linear, and assuming a tolerance of five standard deviations from the mean, the optimal objective function value $J(\boldsymbol{\xi}_L)$ should satisfy the inequality

$$N_{obs} - 5\sqrt{N_{obs}} \leqslant 2\,J(\boldsymbol{\xi}_L) \leqslant N_{obs} + 5\sqrt{N_{obs}} \tag{4.28}$$

where J is the objective function for the full order model and N_{obs} is the total number of measurements. Since we employ an approximate reduced order model in the minimization we apply a less strict criterion. Throughout this study, we use the following criterion

$$J(\boldsymbol{\xi}_L) \leqslant 5\,N_{obs} \tag{4.29}$$

$J(\boldsymbol{\xi}_L)$ represents the objective function computed using the full-order model. This criterion or tolerance can be considered to be an upper-bound to judge whether a set of acceptable optimized parameters has been obtained. Since the reduced-order model in Eq.4.18 is not an exact representation of the full-order model, outer-loops are typically required to update the reduced-order linear models. After convergence of an inner loop, the updated parameters are used as input for a full model simulation, which is added to the training set. An updated reduced order model is constructed as described in Sections 3 and 4, after which a new inner loop (minimization of Eq.4.18 using the reduced order model) is started. In the numerical experiments we use a fixed number of 10 outer-loop iterations, which appear to be sufficient for near-convergence in all cases, after which we evaluate the criterion Eq.4.29 for $J(\boldsymbol{\xi}_L^k)$. An overview of the full workflow is provided in Algorithm 1 for the offline stage and Algorithm 2 for the online stage. We will refer to this workflow as Local Subdomain POD-TPWL (LSPT).

Algorithm 1: Parameter and state reduction

1 Create a large set of model realizations $\boldsymbol{\beta}_i$ for $i = 1, ..., N_\beta$;
2 Form global basis matrix $\boldsymbol{\Phi}_\beta$ and coefficients $\boldsymbol{\xi}$;
3 Partition the global domain into subdomains Ω^d;
4 For local basis matrices $\boldsymbol{\Phi}_\beta^d$ and coefficients $\boldsymbol{\xi}^d$;
5 Simulate a training set of model realizations $\boldsymbol{\xi}_{tr}^d$ and collect snapshots $\mathbf{x}_{tr}^n$;
6 Compute local basis coefficients $\boldsymbol{\psi}_{tr}^{d,n}$;

4.5. RESERVOIR HISTORY MATCHING EXPERIMENTS

In this section, the LSPT method is applied to a history matching example problem based on a 2D reservoir model describing a two phase water-oil reservoir system containing 6 producers and 7 injectors, labeled P_1 to P_6, and I_1 to I_7 respectively (see Fig.4.7). The triangles and circles in the figures denote the injectors and producers, respectively. Some relevant properties of the reservoir geometry, rock properties, fluid properties, and well controls are shown in Table 4.3. An open-source reservoir simulator [16] is used to simulate the reservoir model.

We consider numerical experiments with 2 variations of LSPT. In the first one (LSPT1), we will minimize the number of local PCA patterns using (Eq. 4.16), while in the second

Algorithm 2: Adjoint-based parameter estimation with LSPT

Choose an initial $\boldsymbol{\xi}_L^0$, set $k = 0$;
while $J(\boldsymbol{\xi}_L) > 5\,N_{obs}$ **do**
  Construct the RBF model in all subdomains (Eqs. 4.5-4.6);
  Compute the derivative matrices in all subdomains (Eqs. 4.2-4.4);
  while *not converged* **do**
    Simulate the RBF model (Eqs. 4.5, 4.6;
    Solve the adjoint model (Eq. 4.22;
    Calculate the gradient (Eqs. 4.20-4.21);
    Update the parameter solution $\boldsymbol{\xi}_L^k \rightarrow \boldsymbol{\xi}_L^{k+1}$ (Eq. 4.24);
    Evaluate the reduced model objective function $J_{ROM}(\boldsymbol{\xi}_L^{k+1})$ (Eq. 4.18);
    Check convergence (Eqs. 4.24-4.27);
  end
  Simulate the full model with solution $\boldsymbol{\xi}_L$;
  Update the training set with $(\boldsymbol{\xi}^d, \boldsymbol{\psi}^{d,n})$;
  Evaluate the full-order model objective function $J(\boldsymbol{\xi})$;
  Check convergence (Eq. 4.29)
end

experiment (LSPT2) we use the local PCA patterns obtained by use of an energy cutoff criterion. For comparison purposes, the LSPT results will be compared against subdomain POD-TPWL with global parameterization (GSPT) and against results obtained with finite-difference (FD) gradients with respect to both global PCA and local PCA parameter representations, referred to as GP-FD and LP-FD respectively.

Table 4.3: Reservoir, fluid and well properties of the 2D reservoir model

Description	**Value**
Dimension	40×120×1
Number of wells	6 producers, 7 injectors
Fluid density	1014 kg/m^3, 859 kg/m^3
Fluid viscosity	0.4 mPa·s, 2 mPa·s
Initial pressure	25 MPa
Initial saturation	S_o=0.80, S_w=0.20
Connate water saturation	S_{wc}=0.20
Residual oil saturation	S_{or}=0.20
Corey exponent, oil	4.0
Corey exponent, water	4.0
Injection rate	200 m^3/d
Producer BHP	20 MPa
History production time	10 year
Prediction time	15 year
Model timestep	0.1 year
Measurement timestep	0.2 year

4.5.1. DESCRIPTION OF HISTORY MATCHING SCENARIOS

Experiments are presented for 2 history matching scenarios. Scenario S1 is characterized by the combination of a parameter field dominated by large spatial scales and a relatively small number of well data (fluid rate and watercut (WCT) at the producing wells and bottom-hole pressure (BHP) at the injector wells). In Scenario S2 the spatial scales of the parameter field are substantially smaller. At the same time, a much larger data set is available in this scenario, saturation values in all grid cells, such that it is possible to investigate the scaling of computational efficiency of our proposed methodology with the size of the problem. In both scenarios we will investigate the impact of different decompositions of the model domain into rectangular subdomains. Fig.4.7(a) and Fig.4.9(a) show a base case decomposition based on a 4×5 grid. Note that for this decompositon some subdomains contain no wells, while one of the subdomains contains 2 wells.

SCENARIO S1

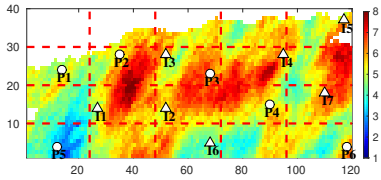
We select 1 out of 1000 generated model realizations as the truth for scenario S1 (see Fig.4.7(a)). A global reduction of the parameter space using a 95% energy cutoff criterion results in $N_G = 48$ preserved global PCA patterns. The number of local PCA coefficients retained in each subdomain is shown in Fig.4.1. The total number of local PCA coefficients $N_L = \sum_{d=1}^{S} l_d = 275$. Fig.4.7 (a) and Fig.4.7 (b) separately represents the projected 'true' permeability field using SLP and local PCA. We follow the procedure described in [13] to determine the number of full order model (FOM) simulations for the selection of snapshots. 22 simulations were run using random global PCA coefficients sampled from the set $\{-1, 1\}$, from which a total of 2200 snapshots of both pressure and saturation were extracted. For each subdomain, two separate eigenvalue problems for pressure and saturation are solved using POD on the two sets of 2200 snapshots, where 95% energy is preserved, and the number of POD patterns for each subdomain is shown in Fig.4.8.

SCENARIO S2

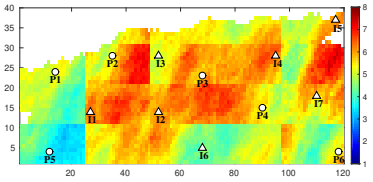
Scenario S2 aims to investigate the possibility of estimating a much larger number of parameters than in S1, given the availability of a much larger number of measured data. As in S1, a 4×5 subdomain decomposition is chosen as a base-case (see Fig.4.9(a)). A second set of 1000 Gaussian-distributed log-permeability fields is generated resulting in a set of $N_G = 282$ preserved global PCA patterns. Fig.4.9 shows the "true", and projected "true" permeability fields using local PCA and smooth local PCA (SLP) respectively. For this scenario 32 FOM simulations were run to select snapshots. The resulting number of local POD patterns in each subdomain is shown in Fig.4.10.

4.5.2. CONSTRUCTION OF THE REDUCED ORDER MODEL

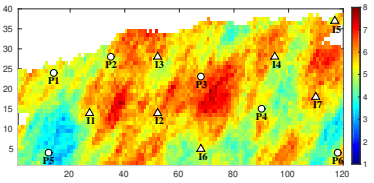
In scenario S1, a total of $53 = 22 + 2 \times 15 + 1$ full order model (FOM) simulations are run to construct the subdomain reduced-order model. 22 FOM simulations are used to collect the snapshots to construct the bases for the states, 1 FOM simulation is the specific training trajectory that is used in the linearization, and an additional $M = 30 = (2 \times 15)$ FOM simulations are run with perturbed parameter inputs to construct the subdomain reduced-order linear model (Eq. 17-18). The vectors of PCA coefficients which correspond to these 53 training models are sampled from a training interval $\boldsymbol{\xi} \in [-1, 1]$ by use of a two-sided perturbation method centered on $\boldsymbol{\beta}_{tr}$. The detailed strategy for sampling training models has been described in Chapter.3.



(a) 'True' model



(b) Projected "true" model using LP



(c) Projected "true" model where the local PCA's results are projected onto the global PCA's

Figure 4.7: Comparison of the "true" reservoir model in full-order space and reduced-order space for Scenario 1. The triangles and circles denote the injectors and producers, respectively. The global saturation and pressure snapshots are decomposed into 20 rectangle subdomains. The red dash lines represent the boundary of subdomains.

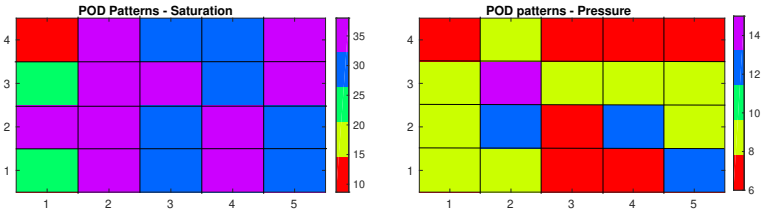
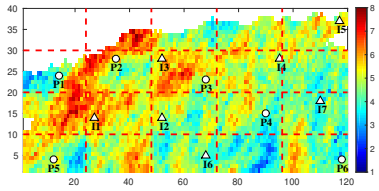


Figure 4.8: The number of reduced pressure and saturation POD patterns in each subdomain for Scenario 1. The global saturation and pressure snapshots are decomposed into 20 rectangle subdomains.

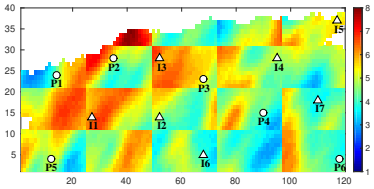
In terms of computational effort, the runtime for a single FOM simulation for this case was about 9.8 s on a machine with i5-4690 Intel CPUs (4 cores, 3.5GHz) and 24 GB memory using Matlab-R2015a. The LSPT base case models for both scenarios, by contrast, required less than 0.3 s. However, the LSPT models for two scenarios separately require simulating 53 and 72 training models plus additional overhead, which significantly increases the cost. Therefore, it would not make sense to construct the LSPT model unless it is to be used for a large number of simulations. Because many simulations are required in history matching applications, the use of LSPT models should be attractive. The results of applying of LSPT in conjunction with an adjoint-based data assimilation procedure is presented in the following parts.

4.5.3. ERROR QUANTIFICATION

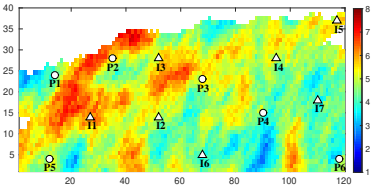
To assess the performance of LSPT, we define the following relative errors with respect



(a) 'True' model



(b) Projected 'true' model using local PCA



(c) Projected 'true' model where the local PCA's results are projected onto the global PCA's

Figure 4.9: Comparison of the "true" reservoir model in full-order space and reduced-order space for Scenario 2. The triangles and circles denote the injectors and producers, respectively. The global saturation and pressure snapshots are decomposed into 20 rectangle subdomains. The red dash lines represent the boundary of subdomains.

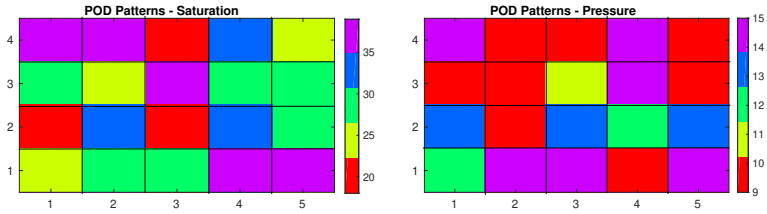


Figure 4.10: The number of reduced pressure and saturation POD patterns in each subdomain for Scenario 2. The global saturation and pressure snapshots are decomposed into 20 rectangle subdomains.

to the full model simulation reference,

$$E_d = \frac{1}{NN_d} \sum_{i=1}^{N} \sum_{j=1}^{N_d} \frac{|(\mathbf{d}_{FOM}^{i,j} - \mathbf{d}_{ROM}^{i,j})|}{\mathbf{d}_{FOM}^{i,j}} \tag{4.30}$$

where $\mathbf{d}$ represents the data vector (fluid rates, WCT , BHP and/or water saturation), and

$$E_x = \frac{1}{NN_d} \sum_{i=1}^{N} \sum_{j=1}^{N_d} \frac{|(\mathbf{x}_{FOM}^{i,j} - \mathbf{x}_{ROM}^{i,j})|}{\mathbf{x}_{FOM}^{i,j}} \tag{4.31}$$

where, $\mathbf{x}$ represents the state vector (saturation and/or pressure in each gridblock).

We analyze the dependence of the model errors with respect to domain decomposition, energy cutoff criterion, testing interval and training interval. The testing interval or training interval represent the predefined perturbation intervals from which the testing and training samples are selected. Fig.4.11 shows the RMSE error in fluid rate, water-cut,

pressure and saturation as a function of these four factors for scenario S1. For scenario S2, Fig.4.12 shows predictions of gridblock saturation and the corresponding relative errors at day 1825 and day 3650 using FOM and LSPT simulations for a relatively small subdomain size of 3×4 cells, which produced the most accurate results for this case. Accuracy is also improved by increasing the energy threshold and thus retaining more POD patterns, albeit at an increased computational cost. Retaining 95% of the total energy during projection produces an acceptable accuracy in this case. The testing intervals and training intervals represent the magnitude of two-side perturbations around the linearized trajectory corresponding to $\boldsymbol{\xi}_{tr}$. The testing case and training models will be randomly sampled from these two intervals, respectively. Increasing the testing interval, which represents the maximum discrepancy between test model and linearized training model, deteriorates the accuracy of the reduced model, with the best results obtained here with a [-0.1, 0.1] interval. For this choice of training interval an appropriate iteration step size for the history matching process should therefore be set as 0.1.

In order to evaluate the quality of parameter estimation results, we will compare the value of the final objective function value against the tolerance (Eq.4.29) and against reference objective function values for the true model (reflecting the impact of the data noise) and the projected true model (that is the best possible reconstruction of the truth given the selected PCA patterns) will be provided as well. Reconstructed parameter maps will provide a visual indication of smoothness and uniqueness of the solution. For all approaches we will list the computational cost expressed in terms of the number of full order model simulations.

4.5.4. GENERATION OF NOISE MEASUREMENTS

The historic production period is 10 years, during which well measurements are taken at 0.2 year intervals, resulting in 50 time instances in total. Normal distributed independent measurement noise with a standard deviation equal to 5% of the 'true' data value was added to all measurements. The complete well data set consist of 300 fluid rates and 300 WCT values measured in the producers and 350 bottom-hole pressures measured in the injectors (950 measured data points in total).

The seismic data used in scenario S2 correspond to the saturation values from the 'true' model simulation after 1825 days (1st monitor) and 3650 days (2nd monitor) of production, mimicking the collection of data from 2 seismic monitor surveys. For this scenario there are in total 8920 measurements. The noisy measurements for the two monitor surveys are shown in Fig.4.13.

4.6. RESULTS

4.6.1. SCENARIO S1

BASE-CASE

Fig.4.14 and Table 4.4 summarize the decrease in the objective function over the outer and inner iteration loops, and the initial and final values respectively. (Note that the jumps of the cost function values in the inner-loop iterations as Fig.4.14(a) are the starts of new outer-loop iterations.) Fig.4.15 and Fig.**??** show the true, initial (prior) and estimated (posterior) log-permeability fields, and Fig.4.16 shows the prior and poster data mismatches.

Fig.4.14 shows that GP-FD, LP-FD and GSPT obtain similar cost function values after the minimization, while our proposed LSPT obtains a slightly less accurate result. As can

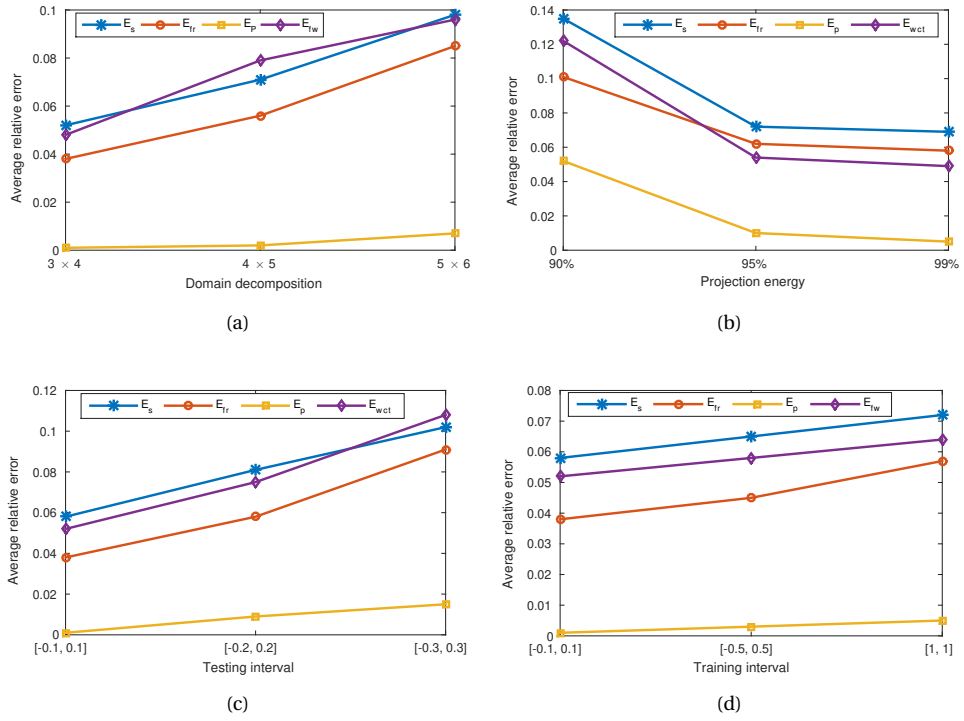


Figure 4.11: Average LSPT errors as a function of domain decomposition, projection energy, testing interval and training interval for scenario 1. Results are for LSPT models constructed using 53 training simulations for the test case.

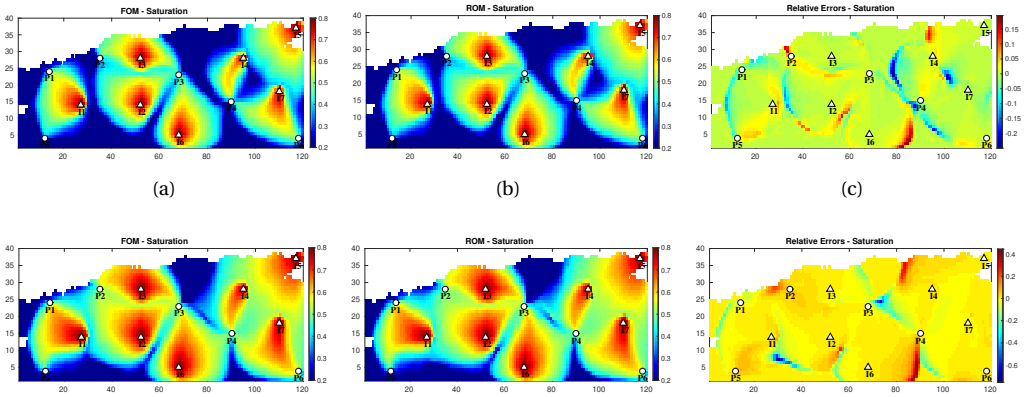
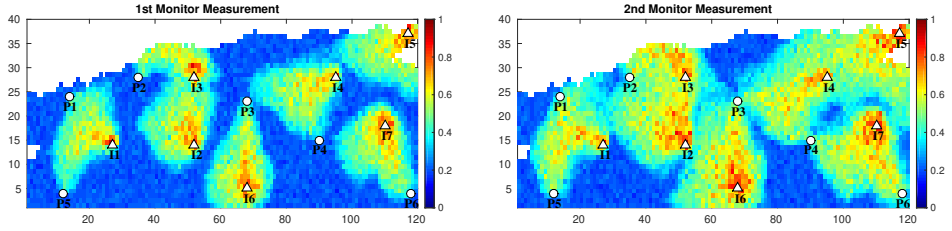


Figure 4.12: Predictions of saturation distribution and its corresponding relative errors at day 1825 and day 3650 using FOM and LSPT for scenario 2. The first row is at day 1825, while the second row is at day 3650.



(a) (b)

Figure 4.13: Noise distribution of water saturation for scenario S2. Normal distributed independent measurement noise with a standard deviation equal to 5% of the 'true' data value, was added to all observations

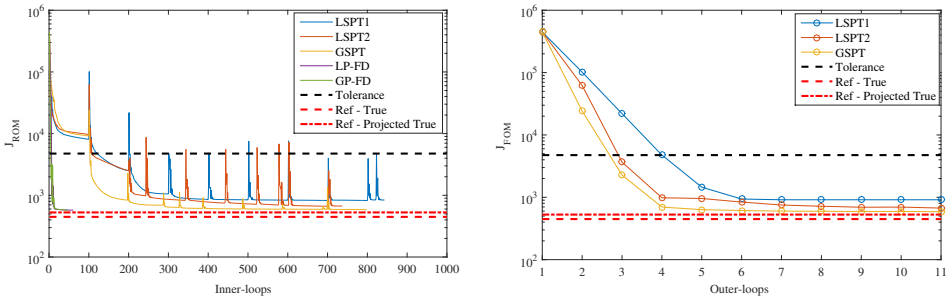
be seen from Table 4.4, both LSPT1 and LSPT2 need 62 FOM simulations (9 FOM simulation are used to update the reduced-order linear model in 9 outer-loops. GSPT needs $22+(4\times48+1)+9=224$ FOM simulations.

Fig.4.15 shows the true, initial and final updated log-permeability fields. Although LSPT2 obtains relatively low cost function values, the updated log-permeability field is non-smooth, which severely violates the (geological) assumptions underlying the model, and would therefore have to be rejected. LSPT1 produces an acceptable solution in terms of final objective function value and in terms of the spatial properties of the reconstructed parameter field. In Fig.4.15 (c) one example area (red dashed rectangle) is highlighted in which the log-permeability field is not correctly reconstructed due to lack of observations (no wells are present in the corresponding subdomain). The choice of domain decomposition may therefore have a significant influence on the performance of LSPT. We will further investigate this issue in the following sections.

Fig.4.16 illustrates the match for fluid rate and water-cut data up to year 10 and an additional 15-year prediction for all six producers. The prediction based on the initial model is far from that of the true model. After the history matching, the predictions of the updated model match the observations very well. Also the prediction of the water breakthrough time is improved for all production wells, including the wells that show water breakthrough only after the history matching period.

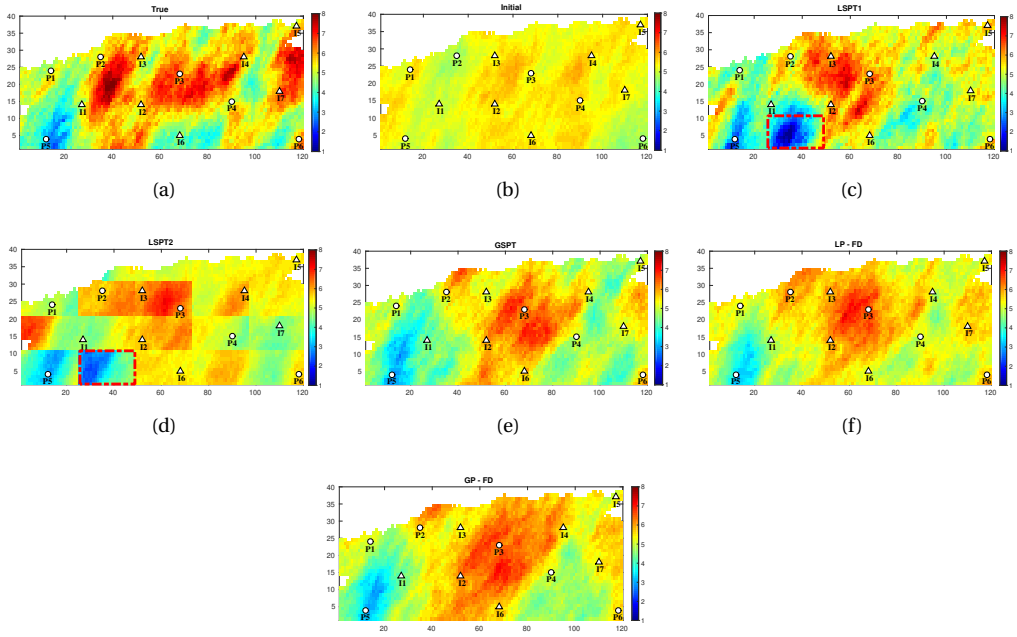
Table 4.4: The number of required FOM simulations and final objective function values for LSPT1, LSPT2, GSPT and FD method for Scenario 1. The reference values of true model and projected true model are also shown here. The domain decomposition for this base-case is 4×5.

-	Iterations	FOM	$J_{FOM}(\boldsymbol{\xi})$
Initial model	-	-	4.49×10^5
LSPT1	10	62 = 22+(2×15+1)+9	912.93
LSPT2	10	62 = 22+(2×15+1)+9	697.32
GSPT	10	224 = 22+(4×48+1)+9	587.83
LP-FD	61	4421	573.94
GP-FD	47	2773	571.73
Tolerance	-	-	4750
Ref - Projected True	-	-	528.1
Ref - True	-	-	447.4



(a) (b)

Figure 4.14: Evolution of the objective function values using LSPT, GSPT, LP-FD and GP-FD method for Scenario 1 as a function of outer-loops. The computation of the objective function for inner-loops and outer-loops uses reduced-order linear model and full-order model, respectively. The two red dash-lines separately represent reference objective function values for the true model (reflecting the impact of the data noise) and the projected true model (that is the best possible reconstruction of the truth given the selected PCA patterns).



(a) (b) (c)

(d) (e) (f)

(g)

Figure 4.15: Comparison of the updated logarithmic permeability fields from the LSPT1, LSPT2, GSPT, LP-FD and GP-FD method for Scenario 1. The true model and initial model are displayed here for a comparison.

IMPACT OF DOMAIN DECOMPOSITION STRATEGY

Fig.4.17, Fig.4.18 and Table 4.5 show the effects of domain decomposition on the minimization process and the final estimate of the log-permeability field. Four designs are

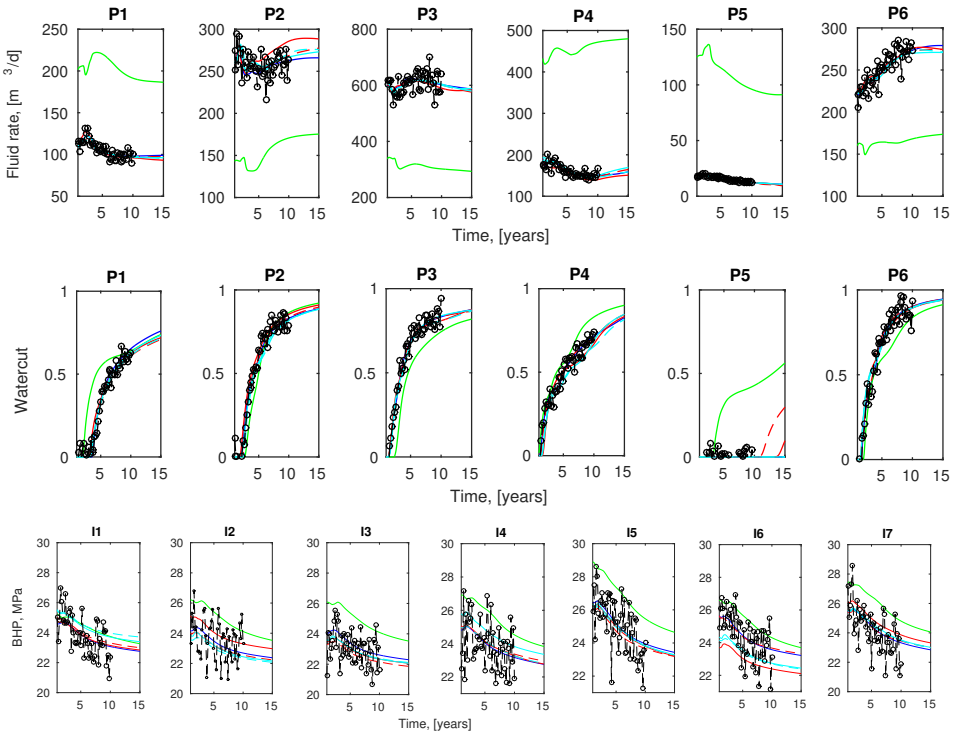


Figure 4.16: Forecast of the liquid rate, WCT and BHP for scenario S1: green line-initial model, blue line-'true' model, solid red line - LSPT, dash red line - GSPT, solid cyan line - LP-FD, dash cyan line - GP-FD

considered, consisting of 2×3, 3×4, 4×5 and 5×6 subdomains respectively. To ensure that all these four schemes converge to a final solution, we specified the maximum number of outer-loops as 15 in this experiment. The total number of local PCA coefficients and the maximum local PCA coefficients among all subdomains are summarized in Table 4.5. It can be seen that a higher number of subdomains will result in a lower number of local PCA patterns per subdomain. As a result, fewer FOM simulations are required. These numerical results demonstrate our aforementioned motivation that The number of training models depends primarily on the maximum number of local parameters in a subdomain, not on the underlying full-order model, which can be decreased by refining the domain decomposition. Fig.4.17 and Fig.4.18 demonstrate that the domain decomposition strategy has significant influence on the performance of LSPT. All four domain decomposition strategies obtain an acceptable cost function value after minimization. However, the updated log-permeability fields differ significantly, which implies that different local minimas are generated using different domain decomposition strategies. The design of decomposition should be implemented with much cares.

QUANTIFICATION OF DIFFERENT SOURCES OF ERRORS

Three main sources of errors (SOE) contribute to the over-all quality of the history matching result: (1) approximation errors of the subdomain POD-TPWL (SOE1), e.g., POD, RBF, and domain decomposition; (2) the loss of global PCA patterns due to an insufficient

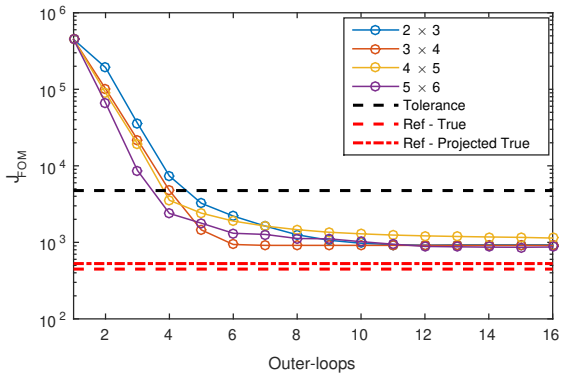


Figure 4.17: Evolution of the objective function values of LSPT using different domain decomposition strategies, 2×3, 3×4, 4×5 and 5×6, for Scenario 1.

Table 4.5: The number of required FOM simulations and cost function values of LSPT using different domain decomposition strategies, e.g., 2×3, 3×4, 4×5 and 5×6, for Scenario 1.

Domain decomposition	N_L	$\max\{l_d\}, d = 1,2,..,S$	N_G	Iterations	FOM	$J_{FOM}(\boldsymbol{\xi})$
Initial model	-	-	-	-	-	4.49×10^5
2×3	112	20	48	15	77 = 22+(2×20+1)+14	901.69
3×4	205	18		15	73 = 22+(2×18+1)+9	878.21
4×5	275	15		15	67 = 22+(2×15+1)+9	912.93
5×6	322	12		15	61 = 22+(2×12+1)+9	869.01
Tolerance	-	-	-	-	-	4750
Ref - Projected True	-	-	-	-	-	528.1
Ref - True	-	-	-	-	-	447.4

number of local PCA patterns (SOE2); and (3) only a fraction, e.g. 95%, of the full energy is preserved by the gloabl PCA (SOE3). To distinguish and quantify these three error contributions, LSPT and FD are consecutively implemented. After minimizing the cost function using LSPT, continuing minimization using FD-LP can quantify the SOE 1, while further minimization using FD-GP can quantify the sum of SOE1 and SOE2. To quantify SOE3, the cost function is minimized by successively preserving an increasing fraction of the global PCA energy, e.g., 95%, 98%, 99% and 99.5%.

We estimate the different error contributions in Table 4.6, Table.4.7 and Fig.4.19 for a decomposition strategy with 2×3 subdomains and a fixed number of outer loop iteratons (15). The impact of the number of retained local PCA patterns is tested using values of 2, 8 and 20 in all subdomains. Table 4.6 summarizes the initial, final and reference cost function values, the total sum of local PCA patterns, and the required FOM simulations. The numerical minimum of local PCA patterns required to fully cover the 48 global PCA patterns is 8. Table 4.6 (last column) and the yellow curve in Fig. Fig.4.19(b), whosing the sum of SOE1 and SOE2, suggest that not much further improvement can be obtained if a GP-FD minimization is performed, suggesting that 8 local patterns are indeed nearly sufficient for obtaining a almost identical solution. Note, however, that a much smaller number of FOM simulations is required. We will further investigate to what extent this finding remains valid when assimilating a large number of measurements in the second case-study. Fig.4.19(a) shows that the continued objective function minimization using LP-FD does not significantly decrease the cost function except in the case that only 2 local

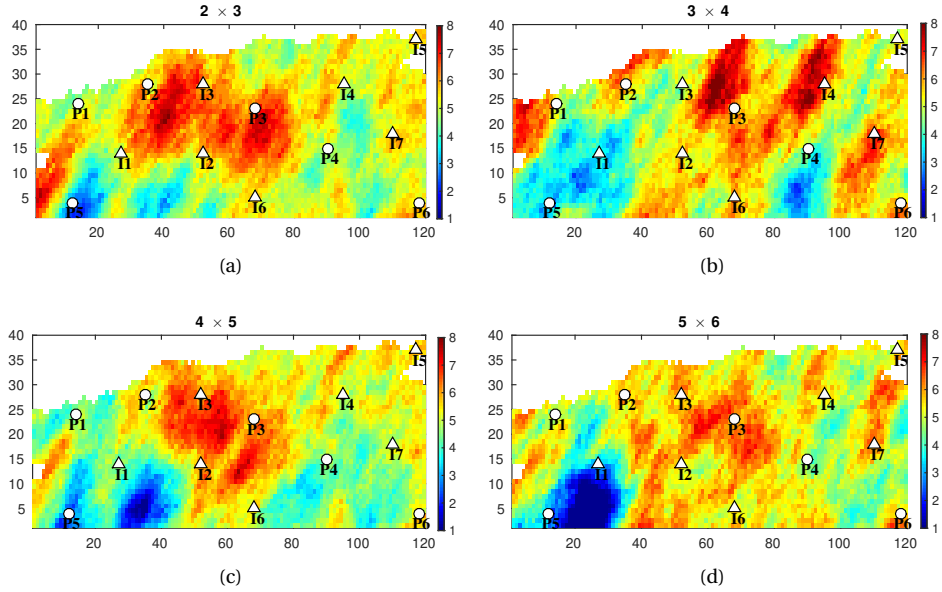


Figure 4.18: Updated logarithmic permeability fields using different domain decomposition strategies, e.g., 2×3, 3×4, 4×5 and 5×6, for Scenario 1.

patterns are retained. This implies that the SOE1 contribution is very small and almost can be ignored, as long as a minimum number of local patterns are retained. SOE2 can be decreased by increasing the number of local PCA patterns, but at the cost of additional FOM simulations.

Fig.4.19(c) and Table.4.7 indicate that SOE3 will gradually decrease with an increasing fraction of retaining (global) energy. Retaining 98% energy is sufficient to accurately represent the original parameter field in this case. An additional increase of 60 global PCA patterns from 48 to 108 requires an additional 10 local PCA patterns in each subdomain. This only requires 20 new FOM simulations.

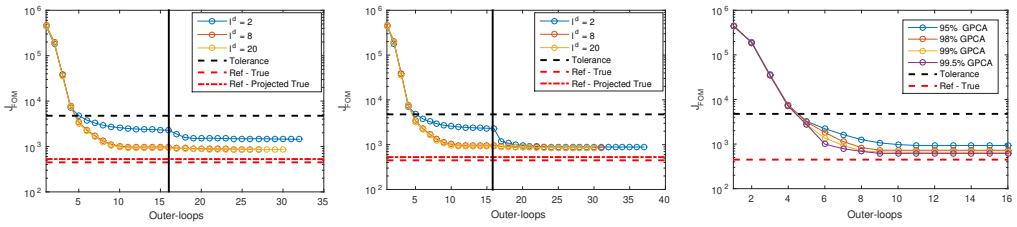


Figure 4.19: Evolution of the cost function values when the LSPT and FD are sequentially implemented. (a) continuing minimization using FD-LP; (b) continuing minimization using FD-GP; (c) the cost function is minimized through preserving more and more global PCA energy, e.g., 95%, 98%, 99% and 99.5%. The vertical black line represents the starting point of minimization using the FD method.

Table 4.6: The number of FOM simulations and the cost function values when quantifying SOE1 and SOE2 for scenario S1. A same number of local PCA patterns l_d is retained among all subdomains.

objective function				$J_{FOM}(\boldsymbol{\xi})$		
Initial model				1.01×10^5		
Tolerance				4750		
Ref - Projected True				528.1		
Ref - True				447.4		
l_d	N_L	N_G	FOM	LSPT	LP-FD	GP-FD
2	12	48	53 = 22+(8×2+1)+14	2276.15	1441	886.3
8	48		53 = 22+(2×8+1)+14	902.48	869.24	860.24
20	120		77 = 22+(2×20+1)+14	892.21	854.69	860.24

Table 4.7: The number of FOM simulations and the final objective function values when quantifying SOE3 for Scenario S1. A same number of local PCA patterns l_d is retained among all subdomains. The first column represents the preserved energy of global PCA. A decomposition strategy with 2 × 3 subdomains is used.

-	N_L	l_d	N_G	Iterations	FOM	$J_FOM(\boldsymbol{\xi})$
Initial model	-	-	-	-	-	1.01×10^5
95%	48	8	72	15	53 = 22+(2×8+1)+14	902.48
98%	72	12	72	15	61 = 22+(2×12+1)+14	738.25
99%	96	16	92	15	69 = 22+(2×16+1)+14	694.18
99.5%	108	18	104	15	73 = 22+(2×18+1)+14	621.52
Tolerance	-	-	-	-	-	4750
Ref - Projected True	-	-	-	-	-	528.1
Ref - True	-	-	-	-	-	447.4

4.6.2. SCENARIO S2

For the S2 scenario with a large number of measurements, three different domain decomposition strategies, i.e., 3×4, 4×5 and 5×6, are formed. The required minimum number of local PCA patterns corresponding to the different domain decomposition strategies is summarized in Table 4.8 while the objective function evolution over the history matching process is shown in Fig.4.20. The 4×5 domain decomposition strategy is seen to lead to the smallest objective function value in this case. Figure 4.21 shows that the true parameter field can be reconstructed very accurately when a large number of measurements is available. Only the 5 × 6 decomposition leads to a poor spatial reconstruction, consistent with a relatively high objective function value. Fig.4.22 shows the predicted saturation before and after the history matching. Compared to the initial model, the model predictions have been significantly improved.

Figure 4.23 shows the simulated and measured well data up for the 10-year history period, and simulated data for an additional 15-year prediction period. Results are shown for the initial model and for the estimated models for the three different domain decompositions, as well as for the GP-FD solution. The prediction of especially the fluid rate and bottom-hole pressure based on the initial model is quite poor. After the history matching, the predictions of all the updated model are consistent with the measurements (and associated errors).

Compared to S1, the number of global PCA patterns has been increased from 48 to 282, however, taking the 4 × 5 decomposition as an example, the required number of FOM simulations has only increased from 53 to 72. The degree of freedom for the history matching problem depends on the number of global PCA patterns, while the required FOM simulation depends on the number of local PCA patterns. It is therefore very attractive to increase the degree of the freedom by adding local PCA patterns in all subdomains. Taking the 5 × 6

domain decomposition scheme as an example, adding one local PCA pattern in each subdomain allows us to retain another 30 global PCA patterns, while only 2 more FOM simulations are added to the entire history matching procedure. These numerical results further demonstrate that introducing local parameterization makes subdomain POD-TPWL highly scalable. In other words, the required number of FOM simulations does not grow rapidly with an increasing number of uncertain parameters.

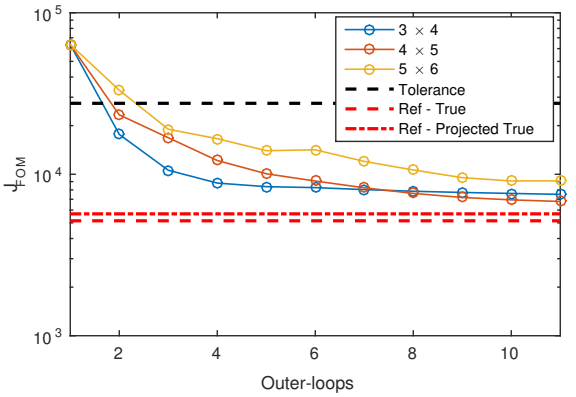


Figure 4.20: Evolution of the objective function values as a function of outer-loops using LSPT for Scenario 2. The computation of the objective function for the outer-loops uses full-order model simulations. The two red dash-lines separately represent reference objective function values for the true model (reflecting the impact of the data noise) and the projected true model (that is the best possible reconstruction of the truth given the selected PCA patterns).

Table 4.8: The number of FOM simulations and the final objective function values of LSPT using different domain decomposition strategies, e.g., 3×4, 4×5 and 5×6, for Scenario 2. A same number of local PCA patterns l_d is retained among all subdomains.

-	l_d	N_L	N_G	Iterations	FOM	$J_{FOM}(\boldsymbol{\xi})$
Initial model	-	-	-	-	-	6.39×10^4
3×4	24	288		10	90 = 32+(2×24+1)+9	7508
4×5	15	300	282	10	72 = 32+(2×15+1)+9	6783
5×6	10	300		10	62 = 32+(2×10+1)+9	9601
Tolerance	-	-	-	-	-	2.75×10^4
GP-FD	-	-	-	-	-	6416
Projected 'True' model	-	-	-	-	-	5685
'True' model	-	-	-	-	-	5149

4.6.3. COMPUTATIONAL COMPLEXITY

The computational cost of the proposed parameter estimation approach can be split into two main parts. The cost of the offline stage consists of constructing the subdomain reduced-order linear model. The cost of the online stage consists of the cost of solving the reduced system and the parameter estimation problem. We will now discuss these two stages in more detail below.

OFFLINE STAGE

The cost of executing parameterization using eigenvalue decomposition of the global

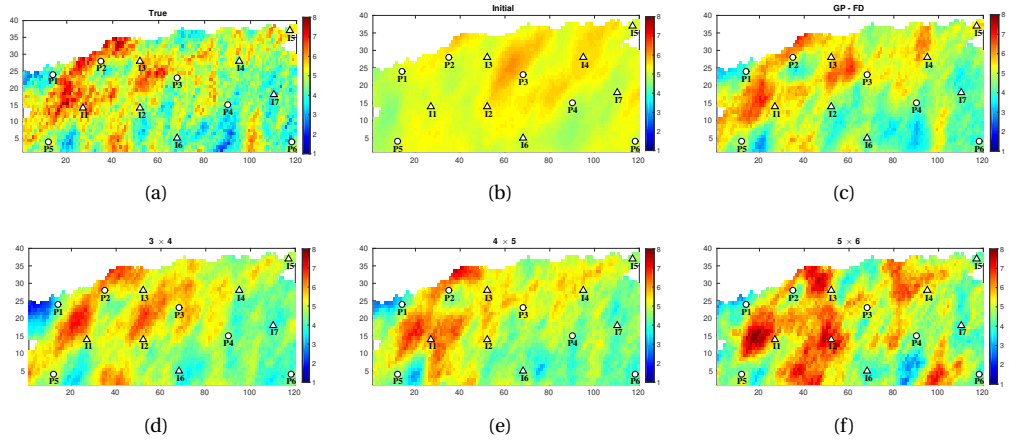


Figure 4.21: Comparison of updated permeability fields using LSPT and GP-FD for scenario S2

covariance matrix and local covariance matrix in each subdomain is negligible for small models, while it will become more significant for large-scale models. The approximate computational complexity is of order $O(N_\beta^3)$. An equivalent formulation can be derived in which the eigenvalue problem is formulated in the snapshots coordinate $\hat{\mathbf{C}}$ by $\frac{\mathbf{X}_c^T \mathbf{X}_c}{N_r - 1}$. It is a so-called method of snapshots [17]. An alternative way to calculate PCA patterns is to perform a singular value decomposition (SVD) on the matrix $\mathbf{X}_c$. It can avoid the calculation of the covariance matrix $\mathbf{C}$ or $\hat{\mathbf{C}}$. The costs of these two efficient alternatives are proportional to $O(N_r^3)$ and $O(N_\beta N_r^2)$, respectively. Both of them have a numerical advantage, because typically $N_r \ll N_\beta$.

Generating snapshots is an essential part of the POD method. The actual generation of the snapshots is done by sampling an ensemble of parameter realizations around an initial mean field. For each member in the newly generated ensemble the FOM is simulated, and the values for the state variables at each time step are saved. The computational cost of performing this part of the process, expressed in number of FOM runs is equal to the number of members in the generated ensemble, namely $O(FOM)$. Here $O(FOM)$ denotes the computational complexity for one full-order model simulation, which is associated with the model dimension N_β, the number of simulation time steps N, and the efficiency of the numerical solver of the forward modelling code. Taking Newton-Raphson iteration as an example, the corresponding computational complexity is $O(\log_2(N_\beta)N)$ [18].

The cost of solving the reduced eigenvalue problem to construct the POD is equivalent to the cost of a Singular Value Decomposition of the snapshot matrix. Since the dimension of the snapshot matrix in each subdomain is relatively low, this cost is also low.

The cost of approximating derivatives using the RBF interpolation represents the most computationally expensive part of constructing the subdomain reduced-order linear model. The computational time expressed in number of FOM runs is several times the number of local PCA patterns generated, or $O(\log_2(N_\beta)N)$.

ONLINE STAGE

The cost of solving a system of model-reduced linear equations can be neglected in

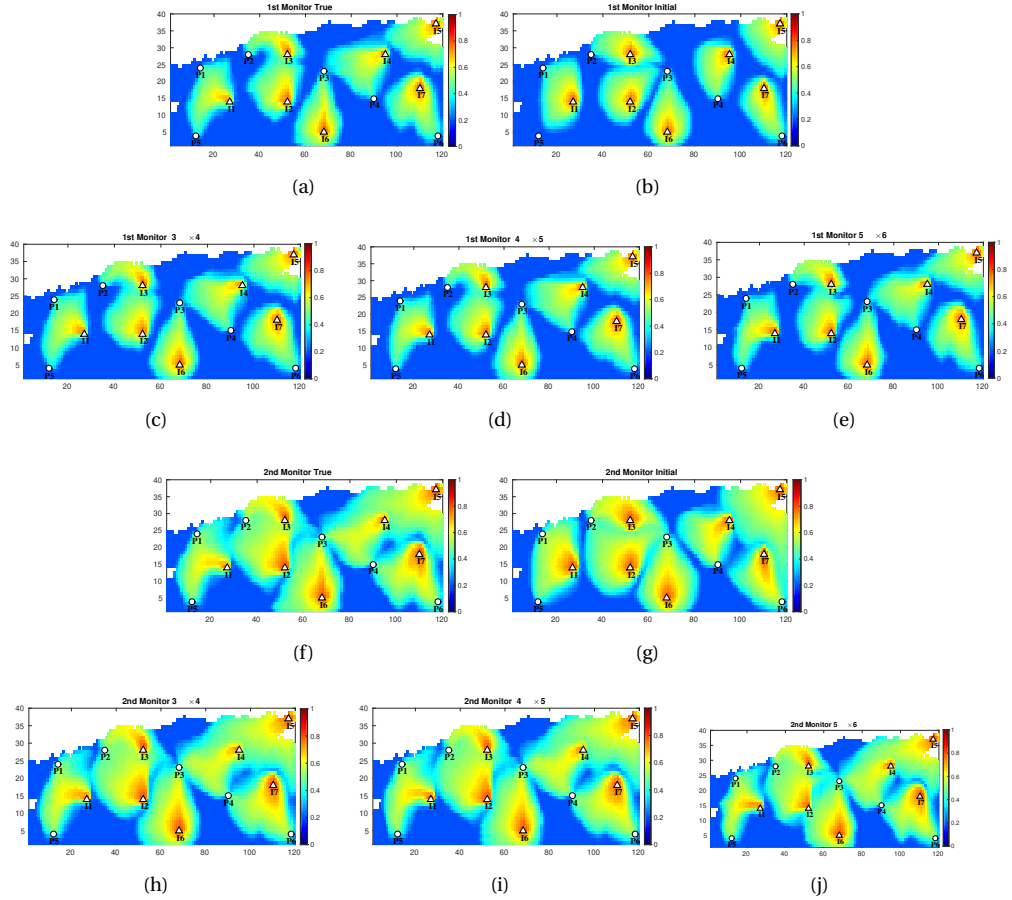


Figure 4.22: Predictions of water saturation using LSPT for Scenario 2. The subfigures (a) - (d) present the results at the 1st monitor, subfigures (e) - (h) present the results at the 2nd monitor. Large errors occur at the water fronts. Three domain decomposition strategies, e.g., 3×4, 4×5 and 5×6, are conducted.

comparison with all other contributions.

The cost of the model-reduced optimization procedure is proportional to the number of times that a new subdomain reduced-order linear models is constructed which requires one FOM run with a cost of $O(\log_2(N_\beta)N)$.

In short, the total computational cost in terms of order analysis is $(O(N_\beta^3)+O(\log_2(N_\beta)N))$. The process is code non-intrusive and does not involve overwhelming programming efforts. As the adjoint model is not always available, especially for commercial simulators, the finite-difference method can be used to approximate the gradient for use in an objectve function minimization procedure. In that case, $O(10^3-10^4)$ FOM simulations will typically be required for large-scale parameter estimation problems. An offline cost for our proposed approach of $O(10-10^2)$ FOM simulations is therefore a significant improvement. For large-scale parameter estimation problems, the computational cost is dominated by the required FOM simulations. In our proposed method, most of the FOM simulations are

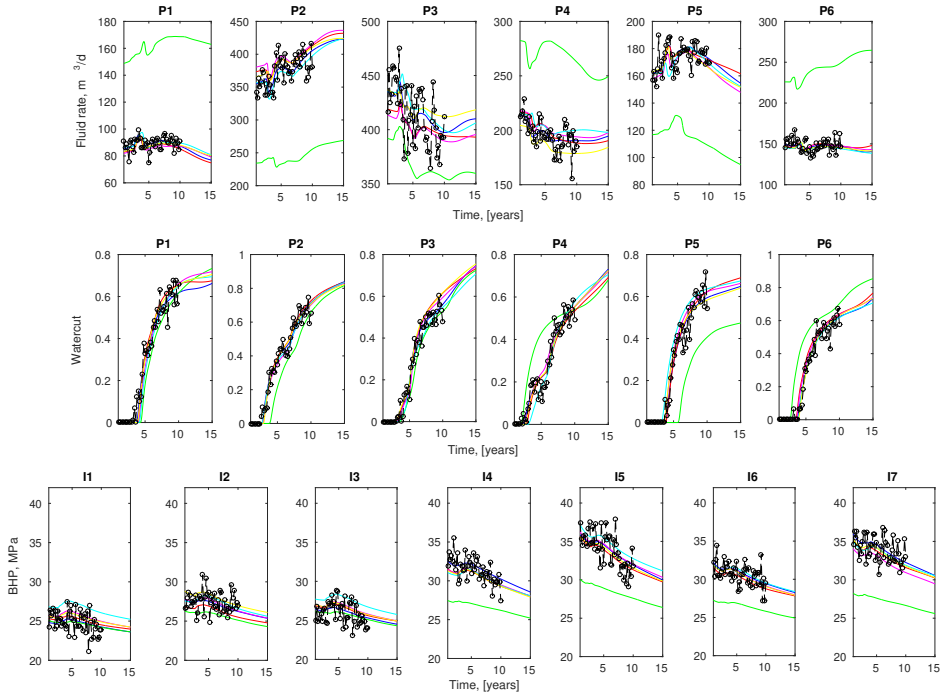


Figure 4.23: Forecast of the liquid rate, WCT and BHP for scenario S2: green line - initial model, blue line - 'true' model, red line - LSPT with 3×4 domain decomposition, magenta line-LSPT with 4×5 domain decomposition, cyan line - LSPT with 5×6 domain decomposition, yellow line - GP-FD

performed in the offline stage.

Figure 4.24 summarizes the required FOM simulations as a function of number of subdomains in this study. These results indicate computational efficiency can benefit from increasing the number of subdomains. On the other hand, the numerical results presented in Fig.4.17 indicate that the quality of the parameter field estimate may deteriorate if too small subdomains are formed. It is therefore important to find an appropriate trade-off in efficiency and accuracy by optimizing the domain decomposition strategy.

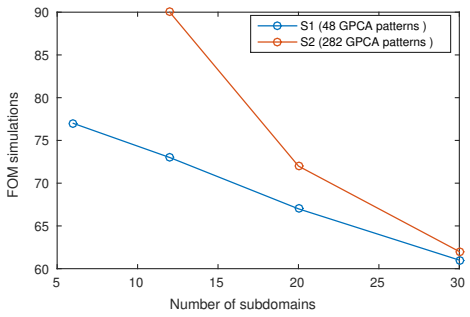


Figure 4.24: Summary of the required FOM simulations for scenario S1 and S2 in the numerical experiments.

REFERENCES

[1] P. Vermeulen and A. Heemink, *Model-reduced variational data assimilation,* Monthly weather review **134**, 2888 (2006).

[2] M. U. Altaf, A. W. Heemink, and M. Verlaan, *Inverse shallow-water flow modeling using model reduction,* International journal for multiscale computational engineering **7** (2009).

[3] M. P. Kaleta, R. G. Hanea, A. W. Heemink, and J.-D. Jansen, *Model-reduced gradient-based history matching,* Computational Geosciences **15**, 135 (2011).

[4] U. Baur, P. Benner, and L. Feng, *Model order reduction for linear and nonlinear systems: a system-theoretic perspective,* Archives of Computational Methods in Engineering **21**, 331 (2014).

[5] D. Amsallem and C. Farhat, *Interpolation method for adapting reduced-order models and application to aeroelasticity,* AIAA journal **46**, 1803 (2008).

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

[7] G. B. Wright, *Radial basis function interpolation: numerical and analytical developments,* (2003).

[8] C. Chen, G. Gao, J. Honorio, P. Gelderblom, E. Jimenez, T. Jaakkola, *et al.*, *Integration of principal-component-analysis and streamline information for the history matching of channelized reservoirs,* in *SPE Annual Technical Conference and Exhibition* (Society of Petroleum Engineers, 2014).

[9] P. Sarma, L. J. Durlofsky, and K. Aziz, *Kernel principal component analysis for efficient, differentiable parameterization of multipoint geostatistics,* Mathematical Geosciences **40**, 3 (2008).

[10] B. Jafarpour and D. B. McLaughlin, *History matching with an ensemble kalman filter and discrete cosine parameterization,* Computational Geosciences **12**, 227 (2008).

[11] Y. Chen and D. S. Oliver, *Multiscale parameterization with adaptive regularization for improved assimilation of nonlocal observation,* Water resources research **48** (2012).

[12] J. He and L. J. Durlofsky, *Constraint reduction procedures for reduced-order subsurface flow models based on pod–tpwl,* International Journal for Numerical Methods in Engineering **103**, 1 (2015).

[13] C. Xiao, O. Leeuwenburgh, H. X. Lin, and A. Heemink, *Non-intrusive subdomain pod-tpwl for reservoir history matching,* Computational Geosciences **23**, 537 (2019).

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

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

[16] K.-A. Lie, S. Krogstad, I. S. Ligaarden, J. R. Natvig, H. M. Nilsen, and B. Skaflestad, *Open-source matlab implementation of consistent discretisations on complex grids,* Computational Geosciences **16**, 297 (2012).

[17] L. Sirovich, *Turbulence and the dynamics of coherent structures. i. coherent structures,* Quarterly of applied mathematics **45**, 561 (1987).

[18] M. A. H. Veron and C. U. d. L. Rjoja, *Newton-raphson's method and convexity,* Univ. u Novom Sadu Zb. Rad. Period.-Mat. Fak. Ser. Mat. **22**, 159 (1992).

5

SUBDOMAIN POD-TPWL WITH ADAPTIVE DOMAIN DECOMPOSITION STRATEGY

Since the performance of our proposed projection-based subdomain POD-TPWL is very sensitive to how the domain is decomposed, inappropriate domain decomposition schemes negatively influence the accuracy of subdomain POD-TPWL. This chapter presents an adaptive decomposition strategy, named smooth weighted local parameterization (SWLP). The SWLP strategy progressively enlarges the weighting coefficient corresponding to the appropriate domain decomposition scheme in the process of objective function minimization. Experiment results with a 3D benchmark reservoir model have demonstrated an effective means to improve the history matching results tested on a 3D benchmark reservoir model. The solution, however, is completely restricted within a predefined group of domain decomposition schemes, and the results of this proposed adaptive strategy can always converge to the best domain decomposition scheme from this group.

Parts of this chapter have been submitted to Journal of Petroleum Science and Engineering. **Xiao, C.**, Leeuwenburgh, O., Lin, H.X. and Heemink, A., 2020. *Conditioning of Deep-Learning Surrogate Models to Image Data with Application to Reservoir Characterization.* Journal of Petroleum Science and Engineering.

5.1. ADAPTIVE DOMAIN DECOMPOSITION STRATEGY

The results of numerical experiments in Chapter 4 show that the performance of subdomain POD-TPWL is very sensitive to the domain decomposition scheme. Inappropriate decompositions lead to poor accuracy of subdomain POD-TPWL. This has motivated us to develop an adaptive decomposition strategy, namely smooth weighted local parameterization (SWLP).

This adaptive decomposition strategy defines a new objective function by weighing the results of a number of domain decomposition schemes simultaneously. The weighting coefficients $\boldsymbol{\omega} = [\boldsymbol{\omega}_1 \quad ... \quad \boldsymbol{\omega}_{L_{lp}}]^T$ (L_{lp} represents the number of domain decomposition schemes) are assigned to each decomposition schemes correspondingly. The process of minimizing this new objective function adaptively adjusts the weighting coefficients and hence selects the optimal domain decomposition correspondingly. Hereinafter we define a group of various domain decomposition schemes as a dictionary.

For the l-th domain decomposition scheme, the entire domain is decomposed into S_l subdomains and the corresponding vector of local PCA coefficients is $\boldsymbol{\xi}_L^l = [\boldsymbol{\xi}^1,..,\boldsymbol{\xi}^d,...,\boldsymbol{\xi}^{S_l}]$. The objective function J_{ROM} (which is equivalent to Eq.4.18 in Chapter 4) is reformulated as a weighted sum of objective-function $\hat{J}_{ROM}$ corresponding to different domain decomposition schemes

$$\hat{J}_{ROM} = \sum_{l=1}^{L_{lp}} \boldsymbol{\omega}_l J_{ROM}(\boldsymbol{\xi}_L^l), \quad \sum_{l=1}^{L_{lp}} \boldsymbol{\omega}_l = 1, \quad 0 \leqslant \boldsymbol{\omega}_l \leqslant 1 \tag{5.1}$$

Instead of minimizing this new objective function with respect to the local PCA coefficients $\boldsymbol{\xi}_L^l$ for all domain decomposition schemes, we can minimize global PCA coefficients $\boldsymbol{\xi}$ alternatively based on a linear relationship between $\boldsymbol{\xi}_L^l$ and $\boldsymbol{\xi}$, e.g., defined as Eq.4.12 in Chapter 4. $\mathbf{T}_{GL}^l$ represents a transformation matrix for mapping the local PCA coefficients $\boldsymbol{\xi}_L^l$ to the global PCA coefficients $\boldsymbol{\xi}$ for the l-th domain decomposition scheme, that is, $\boldsymbol{\xi} = \mathbf{T}_{GL}^l \boldsymbol{\xi}_L^l$.

It is noticeable that Eq.5.1 is a constrained optimization problem, which is generally not easy to deal with. In this study, we represent an easy-to-use approach by reformulating the weighting coefficients as functions of a new variable α and therefore make the constraint, i.e., $\sum_{l=1}^{L_{lp}} \boldsymbol{\omega}_l = 1$ and $0 \leqslant \boldsymbol{\omega}_l \leqslant 1$, automatically satisfied. We also should note that our proposed approach cannot completely represent the original constrained optimization problem, and hence might lead to negative effects. Fortunately, the negative effects almost can be negligible based on the following numerical results. Some available formulas of weighting coefficients are listed in Table 5.1

Table 5.1: Summary of the used formulas of weighting coefficients as functions of variable α.

$\boldsymbol{\omega} = [\boldsymbol{\omega}_1 \quad ... \quad \boldsymbol{\omega}_{L_{lp}}]^T$
Type 1: $\boldsymbol{\omega}_1 = \frac{1}{\sum_{l=1}^{L_{lp}} \alpha^{2(l-1)}}$, ..., $\boldsymbol{\omega}_{L_{lp}} = \frac{\alpha^{2(L_{lp}-1)}}{\sum_{l=1}^{L_{lp}} \alpha^{2(L_{lp}-1)}}$
or
Type 2: $\boldsymbol{\omega}_1 = \frac{1}{\sum_{l=1}^{L_{lp}} e^{(l-1)\alpha}}$, ..., $\boldsymbol{\omega}_{L_{lp}} = \frac{e^{(L_{lp}-1)\alpha}}{\sum_{l=1}^{L_{lp}} e^{(L_{lp}-1)\alpha}}$

Both the global PCA coefficients $\boldsymbol{\xi}$ and the new variable α need to be optimized simultaneously, the gradients can be given by

$$\hat{\mathbf{g}}_{ROM} = [\frac{\partial \hat{J}_{ROM}}{\partial \boldsymbol{\xi}}, \frac{\partial \hat{J}_{ROM}}{\partial \alpha}] \tag{5.2}$$

where

$$\begin{aligned}
\frac{\partial \hat{J}_{ROM}}{\partial \boldsymbol{\xi}} &= \sum_{l=1}^{L_{lp}} \boldsymbol{\omega}_l \frac{\partial J_{ROM}(\boldsymbol{\xi}_L^l)}{\partial \boldsymbol{\xi}} = \sum_{l=1}^{L_{lp}} \boldsymbol{\omega}_l \frac{\partial J_{ROM}(\boldsymbol{\xi}_L^l)}{\partial \boldsymbol{\xi}_L^l} \frac{d\boldsymbol{\xi}_L^l}{d\boldsymbol{\xi}} \\
\frac{\partial \hat{J}_{ROM}}{\partial \alpha} &= \sum_{l=1}^{L_{lp}} J_{ROM}(\boldsymbol{\xi}_L^l) \frac{d\boldsymbol{\omega}_l}{d\alpha}
\end{aligned} \tag{5.3}$$

In order to compute the gradient $\hat{\mathbf{g}}_{ROM}$, three terms, e.g., $\frac{\partial J_{ROM}(\boldsymbol{\xi}_L^l)}{\partial \boldsymbol{\xi}_L^l}$, $\frac{d\boldsymbol{\xi}_L^l}{d\boldsymbol{\xi}}$ and $\frac{d\boldsymbol{\omega}_l}{d\alpha}$ required in Eq.5.3 can be computed as follows:

- The term $\frac{\partial J_{ROM}(\boldsymbol{\xi}_L^l)}{\partial \boldsymbol{\xi}_L^l}$.

  The original objective function can be reformulated as a quadratic function by replacing the non-linear operator with a set of subdomain reduced-order linear models as defined in Eq.4.5 of Chapter 4. More details about the construction of reduced-order linear models using subdomain POD-TPWL algorithm can be found in Chapter 4. After formulating this new objective function, for example $J_{ROM}(\boldsymbol{\xi}_L^l)$ for the l-th domain decomposition scheme, its gradient with respect to $\boldsymbol{\xi}^{d_l}$ can be derived by introducing a model-reduced adjoint model.

- The term $\frac{d\boldsymbol{\xi}_L^l}{d\boldsymbol{\xi}}$.

  The formulation of derivative term $\frac{d\boldsymbol{\xi}_L^l}{d\boldsymbol{\xi}}$ represents the transformation between the global PCA coefficients $\boldsymbol{\xi}$ and the local PCA coefficients $\boldsymbol{\xi}_L^l$ for the l-th domain decomposition scheme. As has been mentioned in Chapter 4, transformation matrix $\mathbf{T}_{GL}^l$ is fully row-rank and therefore a pseudo-inversion can be used to approximate this derivative term.

  $$\frac{d\boldsymbol{\xi}_L^l}{d\boldsymbol{\xi}} = [\mathbf{T}_{GL}^l]^+ \tag{5.4}$$

  where, superscript + represents a pseudo-inversion operation for a matrix.

- The term $\frac{d\boldsymbol{\omega}_l}{d\alpha}$. The formulation of derivative term $\frac{d\boldsymbol{\omega}_l}{d\alpha}$ depends on the selected weighting function defined in Table 5.1. An analytical formulation of this term can be easily obtained in Table 5.2.

Once the gradient $\hat{\mathbf{g}}_{ROM}^k$ at the k-th iteration step is available, the optimal parameters for the next iteration that minimize the objective function is updated by gradient descent iterations,

$$[\boldsymbol{\xi}, \alpha]^{k+1} = [\boldsymbol{\xi}, \alpha]^k - \varepsilon^k \hat{\mathbf{g}}_{ROM}^k \tag{5.5}$$

where ε^k is a step-length at the k-th iteration step. The gradient descent algorithm is considered to have sufficiently converged when either one of the following two stopping criteria is satisfied.

- The objective function defined in Eq.5.1 hardly changes, i.e.,

$$\frac{|\hat{J}_{ROM}(\boldsymbol{\xi},\alpha)^{k+1} - \hat{J}_{ROM}(\boldsymbol{\xi},\alpha)^k|}{max\{|\hat{J}_{ROM}(\boldsymbol{\xi},\alpha)^{k+1})|,1\}} < \eta_{\hat{J}_{ROM}} \tag{5.6}$$

- The estimate of parameters almost does not change, i.e.,

$$\frac{|(\boldsymbol{\xi},\alpha)^{k+1} - (\boldsymbol{\xi},\alpha)^k|}{max\{|(\boldsymbol{\xi},\alpha)^{k+1}|,1\}} < \eta_{\boldsymbol{\xi},\alpha} \tag{5.7}$$

where, $\eta_{\hat{J}_{ROM}}$ and $\eta_{\boldsymbol{\xi},\alpha}$ denote the predefined error constraints, respectively. In our experiments, we set $\eta_{\hat{J}_{ROM}} = 10^{-4}$ and $\eta_{\boldsymbol{\xi},\alpha} = 10^{-3}$. Since simulation runs of the reduced-order model are very cheap, we do not limit the maximum number of iterations.

Table 5.2: Summary of the formulation of derivative term $\frac{d\boldsymbol{\omega}_l}{d\alpha}$.

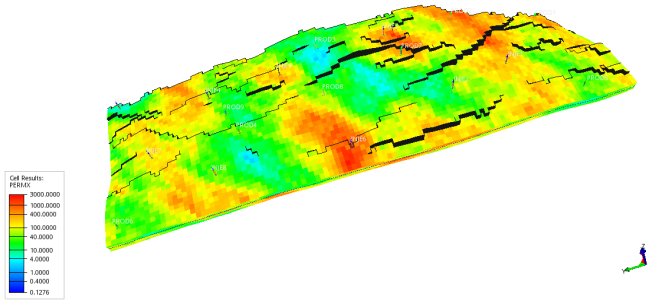
$\boldsymbol{\omega} = [\boldsymbol{\omega}_1 \quad ... \quad \boldsymbol{\omega}_{L_{lp}}]^T$
Type 1: $\boldsymbol{\omega}_1 = \frac{1}{\sum_{l=1}^{L_{lp}} \alpha^{2(l-1)}}$, ..., $\boldsymbol{\omega}_{L_{lp}} = \frac{\alpha^{2(L_{lp}-1)}}{\sum_{l=1}^{L_{lp}} \alpha^{2(L_{lp}-1)}}$
or
Type 2: $\boldsymbol{\omega}_1 = \frac{1}{\sum_{l=1}^{L_{lp}} e^{(l-1)\alpha}}$, ..., $\boldsymbol{\omega}_{L_{lp}} = \frac{e^{(L_{lp}-1)\alpha}}{\sum_{l=1}^{L_{lp}} e^{(L_{lp}-1)\alpha}}$

Once the gradient is available, various optimization methods, such as the steepest descent, can be used to minimize the objective function $\hat{J}_{ROM}$ defined as Eq.5.1. Since the projection-based subdomain POD-TPWL cannot exactly represent the original FOM, after obtaining the optimized parameters corresponding to the specific subdomain POD-TPWL surrogate model, the objective function must be evaluated using the full-order model in order to check whether a satisfactory accuracy has been obtained. If it is necessary, some additional outer-loop iterations can be performed by reconstructing new subdomain reduced-order linear models around the currently updated parameters and then performing the iterative inner-loop again until convergence.

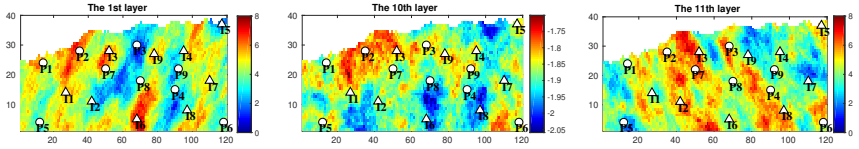
5.2. NUMERICAL EXPERIMENT SETTINGS

The numerical experiments are tested on a 3D benchmark model used in the SAIGUP project [1]. The model consists of 20 layers containing a total of 78720 active grid cells. The reservoir model describes a water-flooding system with nine producers and nine injectors, which are labeled from P_1 to P_9, and I_1 to I_9, see Fig.5.1. Details about reservoir geometry, rock properties, fluid properties, and well controls are shown in Table 5.3. We also show the relative permeability curve of this water-oil two-phase flooding system in Fig.5.2. In our numerical experiments, the open-source simulator *Flow* from the Open Porous Media (OPM) project for reservoir modeling and simulation [2], is used to run simulations.

In this case-study, the logarithmic permeability field is heterogeneous and assumed to be log-Gaussian random fields. We generate Gaussian-distributed realizations of logarithmic permeability using the Stanford Geostatistical Modeling Software (SGeMS) [3]. One of the realizations is chosen to be the reference model as illustrated in Fig.5.1. Fig.5.1(b) also shows the logarithmic permeability fields of the 1st, 10th and 11th vertical layer correspondingly. We should note that the permeability realizations in this example are not conditioned to the values in the wells. Separate random Gaussian distributions are used to model the top nine layers (zone 1) and the bottom 10 layers (zone 2), which are isolated by a vertically impermeable layer (layer 10, corresponding to zone 3). A global reduction of the parameter space using a 95% energy cutoff criterion results in a total of 303 (i.e., 155 and 148 patterns for zone 1 and zone 2, respectively.) global PCA patterns, which are used to approximately represent the original logarithmic permeability fields.



(a) The 3D view of geological realization



(b) The 2D view of the 1st, 10th and 11th horizontal layer

Figure 5.1: The illustrations of the spatial logarithmic permeability for the 1st, 10th and 11th horizontal layer. The triangles and circles denote the injectors and producers, respectively.

Table 5.3: Reservoir settings using OPM for 3D benchmark SAIGUP model.

Reservoir model settings	
Dimension	40×120×20
Number of wells	9 producers, 9 injectors
Constant porosity	0.2
Fluid density	1014 kg/m^3, 859 kg/m^3
Fluid viscosity	0.4 mP·s, 2 mP·s
Bottom-hole pressure for producers	15 MPa
Bottom-hole pressure for injectors	30 MPa
Historical production time	5400 days
Simulation timestep	30 days

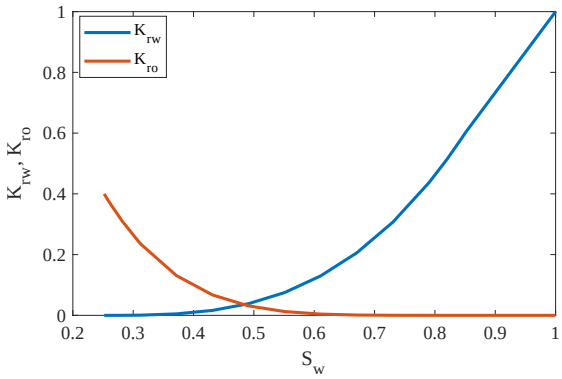
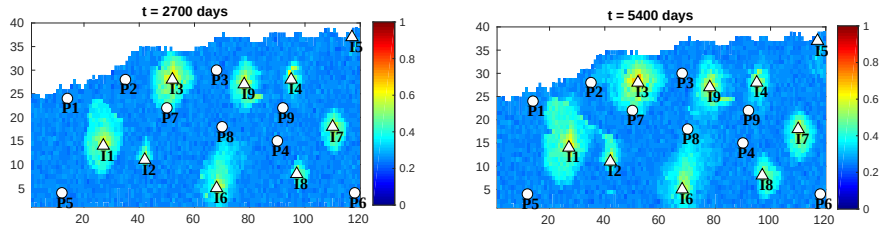
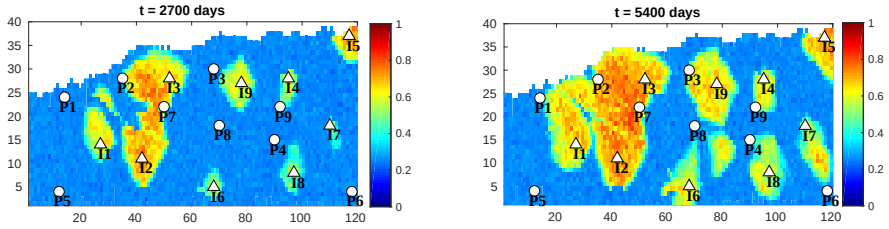


Figure 5.2: The relative permeability curves of this water-oil two-phase flooding system.

The seismic data used in this study correspond to the saturation values from the 'true' model simulation after 2700 days and 5400 days of production, which results in total 157440 measurements. The noisy measurements for the two monitor surveys are shown in Fig.5.3.



(a) The 1st layer



(b) The 11th layer

Figure 5.3: Distribution of noisy saturation at the days 2700 and days 5400 used in this comparative study. Normal distributed independent measurement noise with a standard deviation equal to 5% of the 'true' data value, was added to all observations

To assess the history matching results, we define an error measure based on parameter misfits $e_{\mathbf{m}}$ as follows,

$$e_{\mathbf{m}} = \sqrt{\frac{\sum_{i=1}^{N_m}(\mathbf{m}_{true}^i - \mathbf{m}_{upt}^i)^2}{N_m}} \tag{5.8}$$

where, $\mathbf{m}_{true}^i$ and $\mathbf{m}_{upt}^i$ denote the logarithmic permeability value of the gridblock i from

the reference model and updated model, respectively.

5.3. CONSTRUCTION OF SUBDOMAIN POD-TPWL SURROGATE MODEL

In Chapter 4, our proposed subdomain POD-TPWL is mainly tested on the synthetic or simplified version of 2D models, in which we only need to consider the dynamic associations between one subdomain and its horizontal neighboring four subdomains. In order to adapt subdomain POD-TPWL to 3D models, two domain decomposition strategies are investigated as follows:

- Strategy 1: The entire vertical layers are coupled into one layer, and only the horizontal domain decomposition is implemented. This strategy is very easy to be implemented in practical applications, however, it shows a poor vertical scalability, especially for the multi-layer reservoir models.

- Strategy 2: There are dynamic interactions among adjacent subdomains, e.g., one specific subdomain and its neighboring six subdomains, namely the subdomains at the left, right, forth, back, upward and downward direction, are considered. This strategy has high scalability both in vertical and horizontal direction. In addition, we are able to form different number of subdomains over each layer. This strategy is particularly suitable to assimilate spatial seismic data.

It can be observed from Fig.5.3 that the water-flooded area in the upper layers is smaller than that of lower layers, that is, the saturation and geological parameters have shorter-distance correlations in the upper layers than that of the lower layers. We intentionally design a 3D domain decomposition strategy that the zone 1 is decomposed into small subdomains, while the zone 3 is decomposed into relatively large subdomains. The SLP enables us to represent global PCA patterns using a relatively small number of local PCA patterns in each subdomain. Table 5.4 shows the number of preserved local PCA patterns in each subdomains for these two domain decomposition strategies. In strategy 1 we use a 2D decomposition, while in strategy 2 we use a quasi-3D decomposition consisting of two independent 2D decompositions for zone 1 and zone 2, respectively.

Table 5.4: Summary of the number of local PCA patterns. l^d is the number of local PCA patterns for the subdomain Ω^d. $(DD_x^1 \times DD_y^1, DD_x^2 \times DD_y^2)$ denote the number of subdomains at the two horizontal directions. The superscript 1 or 2 denote the zone 1 and zone 2, respectively.

2D domain decomposition			3D domain decomposition		
$(DD_x \times DD_y)$	l^d	N_L	$(DD_x^1 \times DD_y^1, DD_x^2 \times DD_y^2)$	l^d	N_L
2×3	51	306	(3×4, 2×3)	26	312
3×4	26	312	(4×5, 3×4)	13	312
4×5	16	320	(5×6, 4×5)	8	320
5×6	11	330	(6×8, 5×6)	6	360

We follow the procedure described in Chapter 3 to determine the number of FOM simulations for the collection of pressure and saturation snapshots. 42 FOM simulations are run for collecting a total number of 7560 (42 simulation models by 180 time steps) snapshots of both pressure and saturation. Instead of taking global basis functions to define

the subspace, the snapshots are first partitioned into each subdomain and then local basis functions are obtained from these partitioned snapshots. That is, these 7560 global saturation and pressure snapshots are decomposed into each subdomains. As a result, the implementation of POD for a large number of snapshots will not pose severe computational problems in this 3D model application. The number of POD patterns for each subdomain is shown in Fig.5.4 - 5.5 for these two domain decomposition strategies. In comparison to the 2D domain decomposition, a small number of POD patterns could be preserved for the 3D domain decomposition in each subdomain.

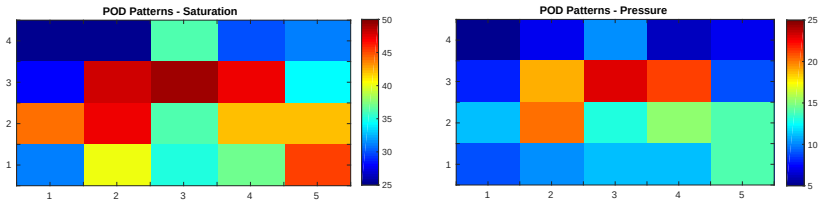
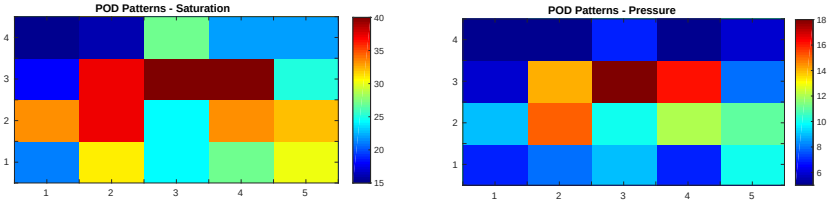
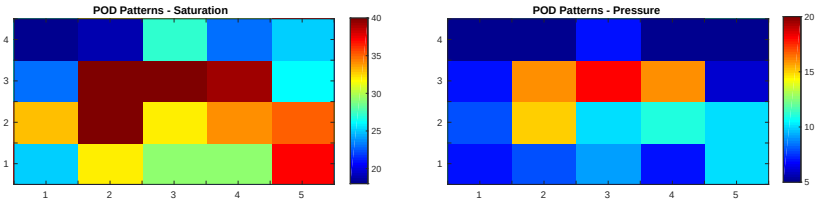


Figure 5.4: The number of local POD patterns in each subdomain for the 2D domain decomposition.



(a) Zone 1



(b) Zone 2

Figure 5.5: The number of local POD patterns in each subdomain for the 3D domain decomposition.

5.4. ASSESSMENT OF SURROGATE MODEL QUALITY

It is useful to analyze the relative error of the predicted saturation between the full-order model simulation and the subdomain POD-TPWL surrogate model. The field-average relative error in saturation over all N_t time intervals, denoted as γ_s^n, for all N_{test} samples,

is given by

$$\gamma_s = \frac{1}{N_{test} N_m N_t} \sum_{i=1}^{N_{test}} \sum_{n=1}^{N_t} \sum_{j=1}^{N_m} \frac{\|\hat{\mathbf{x}}^{i,n,j} - \mathbf{x}^{i,n,j}\|}{\mathbf{x}^{i,n,j}} \tag{5.9}$$

where, $\hat{\mathbf{x}}^{i,n,j}$ and $\mathbf{x}^{i,n,j}$ separately denote the saturation predicted from HFM and surrogate models for the testing sample i, in gridblock j at the timestep n.

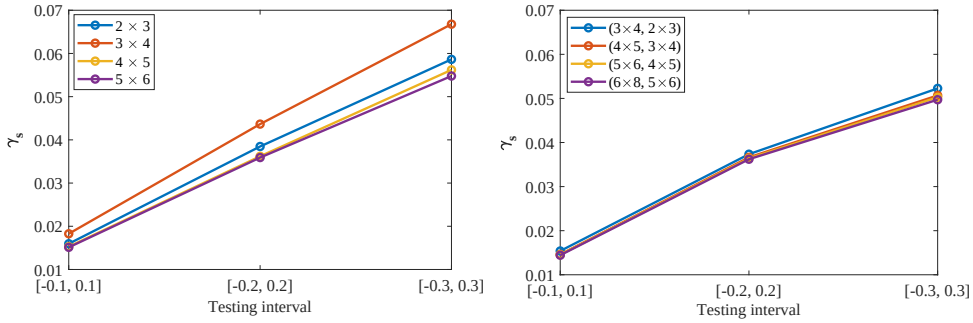
Fig.5.6 shows the dependence of the field-average relative error γ_s with respect to domain decomposition strategy and testing interval. The testing interval represents the maximum discrepancy between test model and the linearized trajectory. It can be seen in fig.5.6(a) that increasing the testing interval deteriorates the accuracy of the reduced model, e.g., from 1.82% to 6.85% for the 3 × 4 2D domain decomposition. Similar results also can be observed for the 3D domain decomposition strategy, as illustrated in Fig.5.6(b). The small γ_s values clearly indicate a high degree of accuracy in the saturation maps predicted from the subdomain POD-TPWL surrogate models. In this case-study, the field-average relative error γ_s is relatively more sensitive to the 2D domain decomposition than that of 3D domain decomposition strategy. In addition, the comparison of the time-varying phase saturation between the subdomain POD-TPWL and HFM is used to visually assess the accuracy. Fig.5.8 shows the predicted saturation distribution and the corresponding absolute errors at day 2700 and day 5400 for the 3×4 2D domain decomposition and (5×6, 4×5) 3D domain decomposition strategy, respectively. The testing model sampled from a small interval [-0.1, 0.1] produces much better predictions than that of a large interval [-0.3, 0.3].

In terms of computational effort, a single FOM simulation requires about 250 seconds on a machine with i5-4690 Intel CPUs (4 cores, 3.5GHz) and 24 GB memory using OPM-*Flow* simulator. The subdomain POD-TPWL models, by contrast, require less than 1.5 seconds. However, the construction of subdomain POD-TPWL needs run many FOM simulations. Taking the 4×5 2D domain decomposition as an example, a total number of 123 HFM simulations are performed to construct the subdomain reduced-order model. Among them, 42 FOM simulations are used to construct the basis for model reduction and remaining 81 FOM simulations are used to construct the subdomain reduced-order linear model.

5.5. HISTORY MATCHING RESULTS USING SUBDOMAIN POD-TPWL

5.5.1. STUDY OF BASE-CASE

Fig.5.9 shows the evolution of the objective function values as a function of out-loop iterations for these two domain decomposition strategies. In this case, 3×4 2D and (3×4, 2×3) 3D domain decomposition schemes obtain the best results for these two strategies, respectively. Taking strategy 1 as an example, although other 2D domain decomposition schemes almost have been convergent indicated by no further reductions of the objective functions, the final objective functions are larger than that of 3×4 2D domain decomposition scheme. This result can be partially explained by some local minimas and even failed convergences. Inappropriate domain decomposition schemes inevitably deteriorate the performance of subdomain POD-TPWL surrogate model. Too small subdomains, for example the (6×8, 5×6) 3D domain decomposition, might harmfully ignore intrinsic



(a) 2D domain decomposition

(b) 3D domain decomposition

Figure 5.6: Field-average relative error γ_s as a function of domain decomposition, testing interval and number of training samples. (a) - (b) are for the subdomain POD-TPWL with 2D and 3D domain decomposition, respectively.

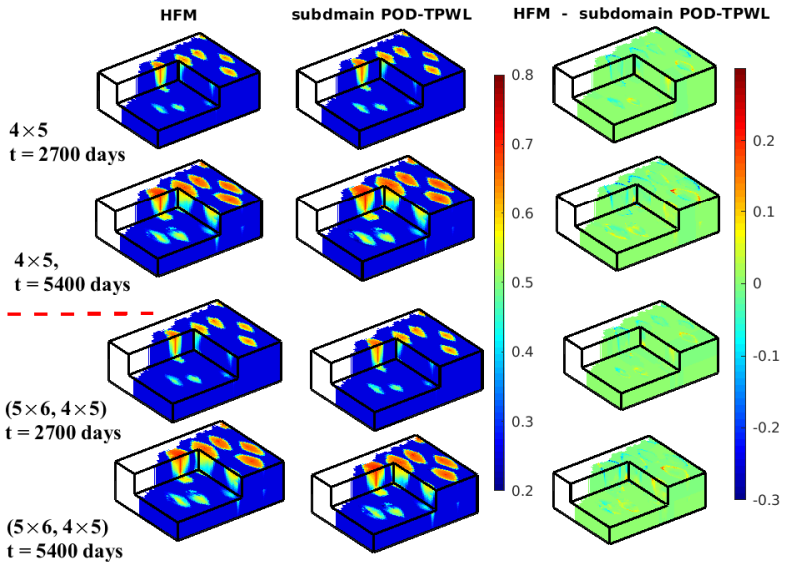
correlations among sudomains and hence cause unsatisfactory history matching results.

The required number of model simulations is shown in Table 5.5. For example, 2×3 2D domain decomposition requires 311 FOM simulations, among them, 42 FOM simulations are used to construct the basis for model reduction, 256 FOM simulations are run to construct the initial subdomain reduced-order linear model at the 1st out-loop, and additional 13 FOM simulations are required in additional 13 outer-loops. Both (3×4, 2×3) 3D and 3×4 2D domain decomposition schemes achieve comparable results with 2×3 2D domain decomposition, however, they require 158 and 311 FOM simulations, respectively. In this case-study, although 3D domain decomposition enables us to run fewer FOM simulations, 2D domain decomposition obtains slightly better results indicated by smaller final objective functions and parameter misfits $e_{\mathbf{m}}$.

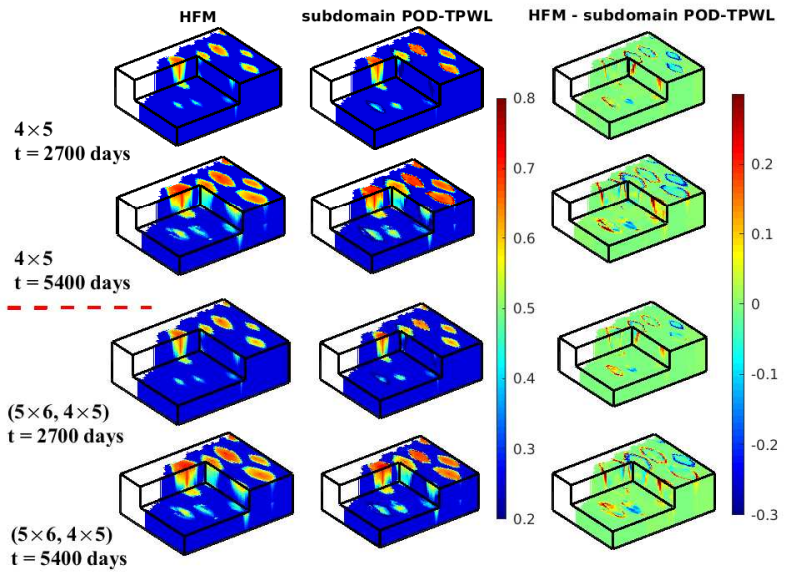
Fig.5.10 displays 2D horizontal cross sections of the initial and updated logarithmic permeability fields. In this case study, the true logarithmic permeability fields almost can be reconstructed, especially for the 3×4 2D domain decomposition scheme. The parameter misfits $e_{\mathbf{m}}$ have been drastically reduced, e.g., from 1.1142 to 0.6641. Some domain decomposition schemes, e.g., 4×5 and 5×6 2D decompositions, yield unreasonable calibrations in some subdomains, e.g., the logarithmic permeabilities are undershooting or overshooting. Overall, these results indicate that the performance of the subdomain POD-TPWL is very sensitive to the domain decomposition schemes. In order to find satisfactory history matching results, we should carefully design reasonable domain decomposition schemes, such as the 3×4 2D domain decomposition or the (3×4, 2×3) 3D domain decomposition in this case-study.

5.5.2. STUDY OF ADAPTIVE DOMAIN DECOMPOSITION STRATEGY

In order to determine the optimal solution from our predefined group of domain decomposition schemes, we have to repeatedly implement the optimization process, which generally requires intensive computations. To efficiently and automatically mitigate the negative effects of inappropriate domain decomposition schemes, e.g., (6×8, 5×6) 3D decomposition, we introduce an adaptive strategy, named smooth weighted local parameterization (SWLP).



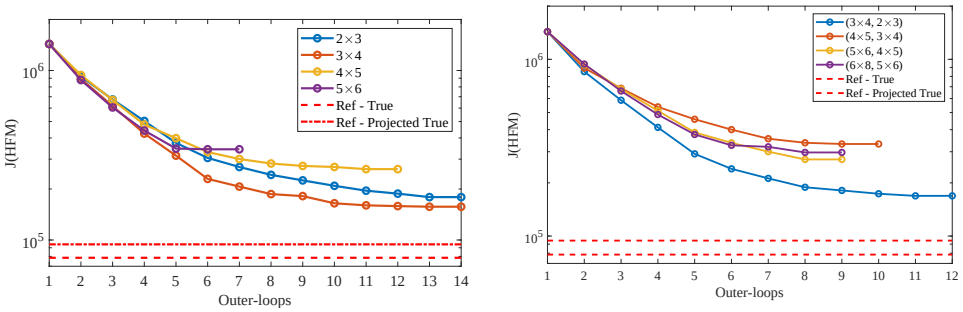
(a)



(b)

Figure 5.7: Illustration of time-varying phase saturation for a random parameter field predicted from subdomain POD-TPWL, FOM and their residual errors at 2700 days and 5400 days of production. (a) Testing interval [-0.1, 0.1]; (b) Testing interval [-0.3, 0.3].

Type 1 and **Type 2** separately denote two formulas of the weighting coefficients given in Table 5.6. As shown in Fig.5.10, **Type 1** generates comparable objective function value with the 3×4 2D domain decomposition scheme, while both **Type 1** and **Type 2** yield com-



(a) 2D domain decomposition

(b) 3D domain decomposition

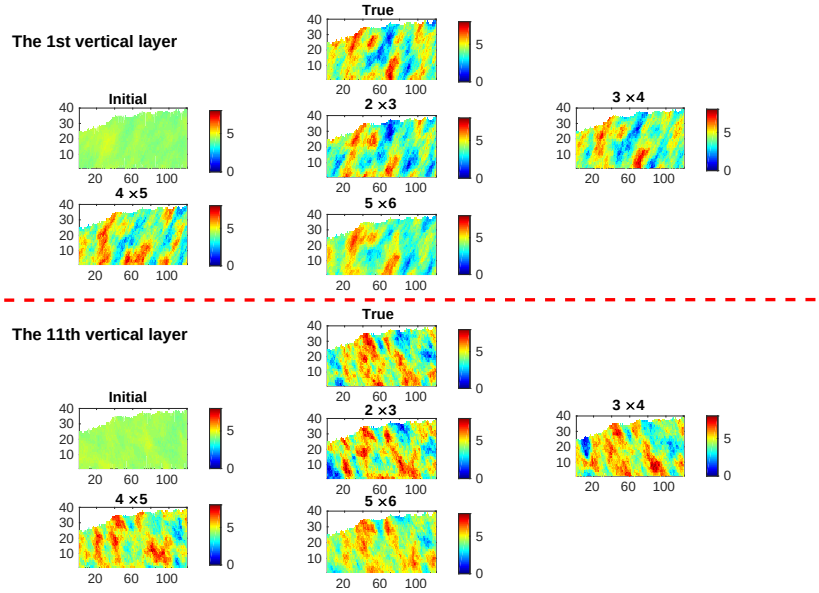
Figure 5.8: Evolution of the objective function values as a function of outer-loop iterations using subdomain POD-TPWL. The calculation of the objective functions uses the high-fidelity models.

Table 5.5: The required number of HFM simulations and final objective function values for these two domain decomposition strategies.

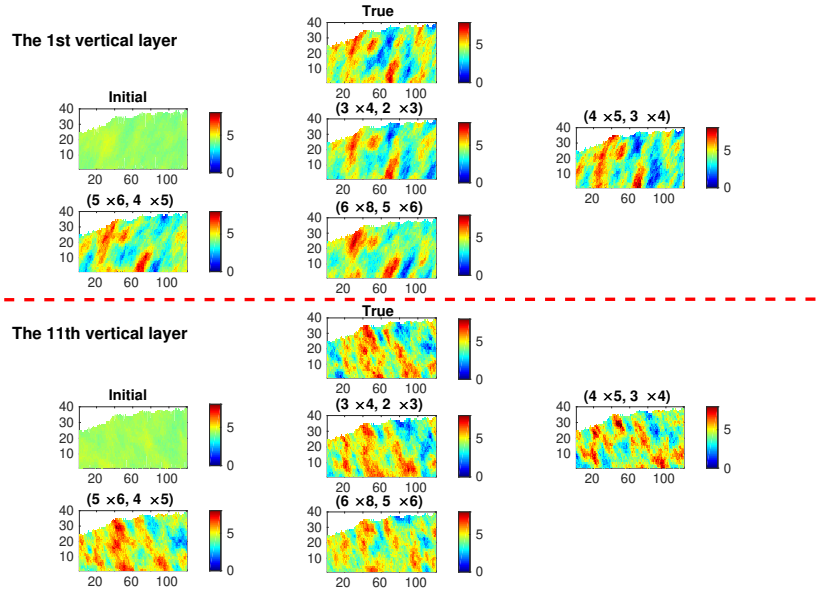
DD	Iterations	HFM	$J(HFM) \times 10^6$	$e_{\mathbf{m}}$
2D domain decomposition				
2×3	14	311	0.1791	0.8115
3×4	14	160	0.1572	0.6641
4×5	12	134	0.2618	0.8376
5×6	7	104	0.3429	0.7430
3D domain decomposition				
(3×4, 2×3)	12	158	0.1689	0.6534
(4×5, 3×4)	10	117	0.3319	0.9254
(5×6, 4×5)	9	107	0.2715	0.8300
(6×8, 5×6)	9	99	0.2973	0.9022
Ref - Projected True	0.0912			
Ref - True	0.0785			

parable objective function values with the (3×4, 2×3) 3D domain decomposition scheme. Fig.5.10 also displays the evolution of the weighting coefficients as a function of outer-loop iterations. We initialize an equal weight for each domain decomposition scheme, and then SWLP could automatically identify the specific domain decomposition whose objective function is easy to be minimized and then gradually increase its weighting coefficient correspondingly or vice versa. For example, both **Type 1** and **Type 2** have identified that (3×4, 2×3) 3D domain decomposition scheme is better than other three domain decomposition schemes in this case. This result is exactly consistent with the previous results. It also can be seen that the performance of this adaptive strategy is slightly sensitive to the formulas of weighting coefficients. **Type 1** can produce a little better result than that of **Type 2** for the problem defined in this study. This adaptive strategy can help us optimize the domain decomposition scheme effectively, particularly when a good prior knowledge of the studied problem is not available. We could consider various domain decomposition schemes as much as possible, which increases the possibility to search the optimal solution.

The required number of HFM simulations for this adaptive strategy is approximately



(a) 2D domain decomposition



(b) 3D domain decomposition

Figure 5.9: 2D horizontal cross sections of the updated logarithmic permeability fields from the subdomain POD-TPWL surrogate models corresponding to different 2D and 3D domain decomposition schemes.

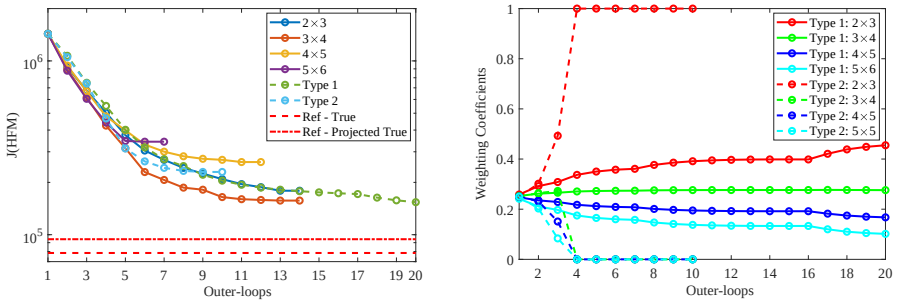
equivalent to the most expensive one among all considered domain decomposition schemes. As shown in Table 5.6, **Type 1** and **Type 2** require 158 and 156 HFM simulations for the 3D domain decomposition strategy, respectively. The results of this proposed adaptive strat-

egy can approximately converge to the best domain decomposition scheme, e.g., (3×4, 2×3). A consistent result is also obtained for the 2D domain decomposition scheme. Taking the 3D decomposition scheme as an example, we generate an ensemble of posterior solutions through starting from different initial models. Fig.5.11 depicts a boxplot of the weighting coefficients and the objective functions with and without the proposed adaptive decomposition strategy. It is noticeable that the (3×4, 2×3) decomposition obtains larger weighting coefficients for almost all ensemble members. The final objective functions also can accurately converge to the (3×4, 2×3) 3D decomposition without the proposed adaptive decomposition strategy.

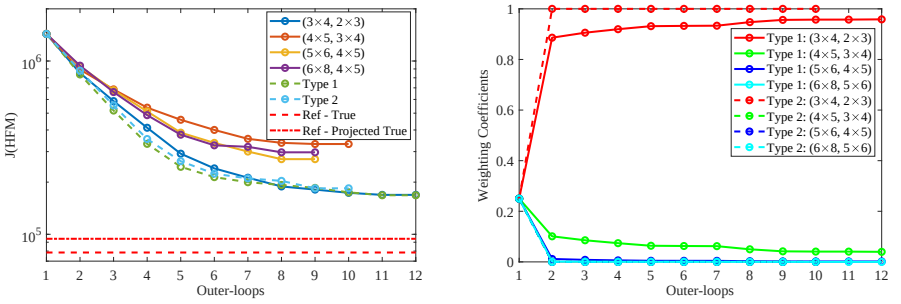
In this case-study, the proposed adaptive strategy always performs as well as the best decomposition in our predefined dictionary of domain decompositions. Nevertheless, limitations of the proposed method do exist. First, since some underlying mathematical structures are enforced by the weighting coefficients, such as exponential function (**Type 1**) and power function (**Type 2**), the SWLP strategy possibly leads to sub-optimal solutions, which are a little worse than that of the best domain decomposition scheme, as shown in Fig.5.11. Second, we should not ignore the deficiency of the proposed adaptive strategy that the final solutions are always limited to our used domain decomposition schemes, and therefore the performance of this approach strongly depends on the design of these decompositions. To cope with this issue, we should consider various domain decomposition schemes as much as possible. However, we have to construct reduced-order models for all domain decomposition schemes simultaneously, which will increase additional computational overhead and memory requirements particularly for high-dimensional models. Fortunately, the computational overhead is generally negligible when compared to the full-order model simulations. In addition, on the basis of the first several iteration steps, this adaptive strategy has correctly indicated the optimal decomposition scheme based on the largest weighting coefficient, and then we can directly minimize the objective function correspondingly.

Table 5.6: The number of required HFM simulations and final objective function values using the proposed adaptive strategy for 2D and 3D domain decomposition schemes.

-	2D domain decomposition				3D domain decomposition			
-	Iterations	HFM	$J(HFM)\times 10^6$	$e_{\mathbf{m}}$	Iterations	HFM	$J(HFM)\times 10^6$	$e_{\mathbf{m}}$
Type 1	20	317	0.1539	0.6013	12	158	0.1691	0.6645
Type 2	10	307	0.2297	0.7399	10	156	0.1842	0.6875

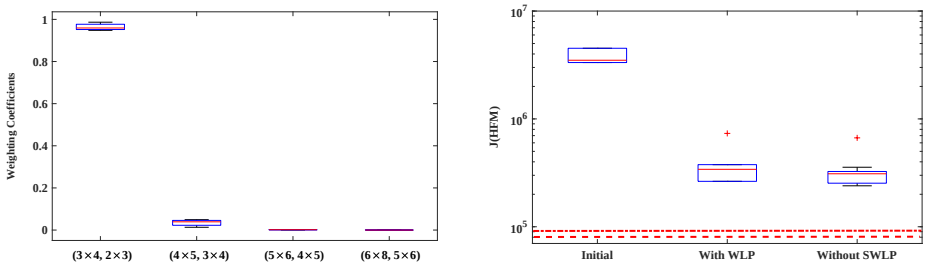


(a) 2D domain decomposition



(b) 3D domain decomposition

Figure 5.10: Evolution of the objective function values and their corresponding weights using subdomain POD-TPWL as a function of outer-loops for the proposed adaptive domain decomposition strategy.



(a) Weighting coefficients

(b) Objective functions

Figure 5.11: Boxplot of the weighting coefficients and the final objective functions using the subdomain POD-TPWL surrogate models. The 3D domain decomposition with adaptive strategy is implemented.

REFERENCES

[1] J. D. Matthews, J. N. Carter, K. D. Stephen, R. W. Zimmerman, A. Skorstad, T. Manzocchi, and J. A. Howell, *Assessing the effect of geological uncertainty on recovery estimates in shallow-marine reservoirs: the application of reservoir engineering to the saigup project,* Petroleum Geoscience **14**, 35 (2008).

[2] A. F. Rasmussen, T. H. Sandve, K. Bao, A. Lauser, J. Hove, B. Skaflestad, R. Klöfkorn, M. Blatt, A. B. Rustad, O. Sævareid, *et al., The open porous media flow reservoir simulator,* Computers & Mathematics with Applications (2020).

[3] C. V. Deutsch, A. G. Journel, *et al., Geostatistical software library and user's guide,* New York **119** (1992).

6

HISTORY MATCHING USING DEEP-LEARNING DETERMINISTIC INVERSION

This chapter presents an efficient deterministic inversion to primarily explore the possibility of applying deep neural network surrogate to approximate the gradient of seismic history matching by using auto-differentiation (AD). The AD enables us to evaluate the gradients efficiently in a parallel manner. Furthermore, the benefits of using stochastic gradient optimizers in the deep learning practice, instead of full gradient optimizers in conventional deterministic inversions, is investigated. The proposed surrogate-assisted inversion with stochastic gradient optimizer obtains a very fast convergence rate against the model noise for the test case with a high-dimensional seismic history matching problem with a large number of data and parameters. In addition, we also conduct several comparisons and evaluations with the projection-based subdomain POD-TPWL approach in terms of the computational efficiency and accuracy. Overall, the accuracy and computational efficiency provided by these two kinds of surrogate models suggest that both of them can be used to efficiently address large-scale seismic history matching problem.

This chapter has been submitted to Knowledge-Based Systems. **Xiao, C.**, Lin, H.X., Leeuwenburgh, O. and Heemink, A., 2020. *Surrogate-Assisted Gradient-Based Seismic History Matching: Comparative Study between Projection-Based Reduced-Order Modeling and Deep Neural Network.* Journal of Petroleum Science and Engineering.

6.1. INTRODUCTION

Reservoir history matching can play a key role in improving geological characterization and reducing the uncertainty of reservoir model predictions [1, 2]. Measured data are typically obtained at wells, which are normally sparsely distributed over large areas, leaving the generally heterogeneous reservoir rock in-between the wells unsampled. Alternative ways of gathering information are based on geophysical techniques, such as time-lapse (4D) seismic data, which is affected by the density and mechanical properties of the rock and fluids. The availability of such geophysical information is often less frequent in time and of low spatial resolution, but much denser compared to the well measure points. In contrast to sparse well data, seismic data generally provides sufficient spatial-information about the dynamic changes in the oil reservoirs [3].

Incorporating 4D seismic data into the history matching workflow, hereinafter referred to as seismic history matching (SHM), has been investigated for several decades. The optimization algorithms for addressing SHM problem in the field of petroleum engineering have been investigated for several decades, among them, the gradient-based deterministic inversion is one of commonly used approaches nowadays. We generally define a nonlinear objective function as the sum of squares of the difference between the recorded data and simulated predictions through running a reservoir model. Gradient-based optimization, using the adjoint model, is used to minimize the objective function through iteratively adjusting the model parameters [4]. The high dimensionality of grid-based seismic measurements and strong non-linearity, however, pose strong challenges, e.g., expensive model simulations and algorithm convergence. Although many efforts have been taken to make the implementation of the adjoint model feasible [5], a fast evaluation of the objective function and its gradient information still remains a big computational challenge. In some cases it may not be feasible to perform the history matching rapidly, because it requires numerous simulations, e.g, simulations of multiple models for the gradient approximation. The surrogate modeling is currently identified as one of the most promising means to improve the efficiency of reservoir simulations. The main focus of this chapter is on investigating surrogate-assisted optimization strategy for seismic history matching.

Surrogate modeling aims at providing a faster emulation with a simplified relation between the inputs and outputs of a complex model. It has the potential of accelerating the gradient-based optimization problems. Existing surrogate modeling approaches can be roughly classified into three categories: hierarchical-based, projection-based reduced-order model (ROM) and data-driven surrogate models. Reviews of surrogate modeling can be found in [6]. In hierarchical-based methods the surrogate is created by simplifying the representation of the physical system, such as by ignoring certain processes, or reducing the numerical resolution or the complexity, e.g., upscaling and grid coarsening [7, 8], or the complexity, e.g., nonlinearity. Projection-based ROM approaches have received attention as a way of reducing the computational effort of model-based workflows through dimensionality reduction [9, 10]. The details about extensive applications of ROMs to various research area can been found in Chapter 3.

The aforementioned ROM accurately represents the high-fidelity forward model in the reduced subspaces spanned by the projection basis which contains the hidden physics. By contrast, data-driven approaches solely depend on data (simulated or real data) to approximate the relation between input and output as accurately as possible, such as polynomial chaos expansion [11] and fully-connected artificial neural networks [12]. These data-driven surrogate models have demonstrated their feasible applicability to some de-

gree, however their use is restricted only to relatively small-dimensional problems and therefore suffer from the *curse of dimensionality* and fail to work for large-scale problems. The deep neural network (DNN) has recently attracted attention because of successful applications in several fields, including computer vision [13] and image processing [14]. Recent advances in DNN and their promising application results have prompted research on deep-learning-based surrogate modeling for high-dimensional subsurface flow systems [15]. The popularity of these methods is further enhanced by the availability of open access machine learning frameworks, such as TensorFlow [16] and PyTorch [17].

Construction of surrogate models for high-dimensional and nonlinear subsurface flow models becomes increasingly popular in various engineering application. For example, Tripathy et al (2018) proposed a fully-connected deep neural network to approximate the flow dynamic of a single phase subsurface water flow in porous media [18]. Zhong et al (2019) proposed a generative surrogate models for the dynamic plume prediction of CO_2 capture and storage problem. This type of surrogate model was built on a generative adversarial network (GAN) [19]. Subsequently, motivated by the developments in the community of image super-resolution and image segmentation tasks, the advanced DNN structures dense block [20] and residual block [21] are introduced to further improve the performance of the DNN architecture. The dense block and/or residual block enable us to design much deeper neural networks while avoiding the vanishing/exploding gradient problem. They have demonstrated impressive performance in effectively obtaining accurate surrogate models for various applications.

The DNN models are generally trained by iteratively adjusting the trainable variables that parameterize the neural network model, during which the auto-differentiation (AD) is used to calculate the gradient of the loss function with respect to neural network parameters, e.g., weights and biases. There is a clear similarity in the way how DNN and SHM iteratively update parameters, e.g., geological parameters in SHM. In this sense, the training of DNN is not that far from the gradient-based SHM. Due to the availability of AD and high-performance computing units, the evaluation of gradients for DNN is much more efficient than that of conventional SHM procedure. An interesting question remains how to take full use of the computational advantage of DNN to address the SHM problem in a short time frame.

The contribution of this chapter to the reservoir data assimilation community is investigating two kinds of surrogate models and verifying their applicability to speedup gradient-based SHM problems. Specifically, the first objective of this work is to explore the possibility of using the DNN surrogate to approximate the gradient for seismic history matching, which has not been fully investigated yet. Addressing gradient-based SHM with the use of a DNN surrogate has the advantage of facilitating the use of deep learning packages. It might benefit from powerful concepts such as AD in the deep learning packages for obtaining gradients in a very efficient manner. Since both DNN surrogate and subdomain POD-TPWL are intentionally used for the gradient-based seismic history matching, a comparative study between them is identified as the second research objective.

The remainder of this chapter is as follows: The gradient-based seismic history matching framework is defined in Section 2. Section 3 describes the deep-learning surrogate model, in which a residual U-Net and an explicit concatenation of time feature are integrated to capture both spatial and temporal features. The gradient approximation for SGD is described as well. Section 4 assesses the surrogate-assisted gradient-based inversion framework on a 3D benchmark reservoir model. A comparative study between the DNN

surrogate model and our previously proposed projection-based subdomain POD-TPWL is also presented. Finally, Section 5 summarizes our contribution and discusses future work.

6.2. GRADIENT-BASED SEISMIC HISTORY MATCHING

History matching time-lapse seismic data requires the capability to compute seismic data from a given reservoir model. The SHM workflow generally needs us to run two consecutive forward models: the fluid flow simulation model and the rock–physics model. The former predicts the reservoir state variables (pressure and saturation), and then the latter simulate the seismic response from the reservoir state variables. The Gassmann model is one of the most commonly used rock–physics model in practice [22]. The relationship between simulated saturation and spatial parameter fields can be described by a nonlinear operator. To simplify the notation without loss of generality, one such generic operator $\mathbf{h}^n$ can be simply described as follows,

$$\mathbf{x}^n = \mathbf{h}^n(\mathbf{m}), \quad n = 1, \cdots, N_t \tag{6.1}$$

where, $\mathbf{m} \in R^{N_\mathbf{m}}$ denotes the vector of spatial parameters for reservoir model. N_m is the total number of gridblocks. n denotes the simulation step. $\mathbf{x}^n \in R^{N_m}$ represents the saturation in all gridblocks, which are constrained to lie within a bound [0,1]. The seismic data is directly related to fluid saturation and pressure changes and provides information on the dynamic behavior of the reservoirs. For simplification, we directly measure fluid saturation in each grid-block without running the rock–physics model for generating the real elastic properties. That is to say, $\mathbf{x}^n$ directly represents the simulated seismic data.

Gradient descent algorithms, or more specifically a full-gradient descent (FGD), are generally used to minimize the objective function in an iterative manner. The general gradient-based seismic history matching can be defined as a PDE-constrained least squares optimization formula. The objective function, denoted as J here, is defined using full dataset with high-fidelity model (HFM) as follows.

$$J(\mathbf{m}) = \frac{1}{2}(\mathbf{m}-\mathbf{m}_b)^T \mathbf{R}_m^{-1}(\mathbf{m}-\mathbf{m}_b) + \frac{1}{2}\sum_{n=1}^{N_t}(\mathbf{d}_{obs}^n - \mathbf{x}^n)^T [\mathbf{R}_{obs}^n]^{-1}(\mathbf{d}_{obs}^n - \mathbf{x}^n) \tag{6.2}$$

where, the measurement errors for the data gathered at the timestep n are generally assumed to satisfy a Gaussian distribution $N(\mathbf{0}, \mathbf{R}_{obs}^n)$, where $\mathbf{R}_{obs}^n$ represents the measurement error covariance matrix. Here we want note that the terminology HFM is equivalent to the FOM (full-order model) used in previous two chapters.

After defining the objective functions, the key step of a gradient-based minimization algorithm is to determine the gradient of the objective function with respect to the parameters. In this paper, both projection-based ROM and DNN surrogate are used to approximate the gradients in an efficient manner.

6.3. SURROGATE MODELING USING DEEP NEURAL NETWORK

This section introduces the procedures of using a data-driven DNN to approximate the relation between geological parameters and saturation. Specifically, the neural network architecture, model training procedure and gradient approximation for the stochastic gradient descent are described.

6.3.1. NEURAL NETWORK ARCHITECTURE

A deep-learning based surrogate model representing the time-dependent process can be presented as follows:

$$\hat{\mathbf{x}}^n = \hat{\mathbf{h}}^n(\mathbf{m}, t^n, \boldsymbol{\theta}), \quad n = 1, \cdots, N_t \tag{6.3}$$

where, $\hat{\mathbf{x}}_n \in R^{N_x \times N_y \times N_z}$ is the neural network prediction (a 3D image) for the input $\mathbf{m} \in R^{N_x \times N_y \times N_z}$ at the time $\mathbf{t}^n$. N_x, N_y and N_z represent the image size. $\boldsymbol{\theta}$ denotes the trainable parameters for the neural network model. In this model, the current output $\hat{\mathbf{x}}_n$ depends only on the time-independent input permeability field $\mathbf{m}$ and time $\mathbf{t}^n$.

Recent applications of deep neural network to subsurface flow simulations have been extensively reported. For example, Jin et al proposed a data-driven DNN surrogate model with autoregressive structure for approximating the time-varying process in reservoir simulation problem [23]. Tang et al [24] developed a deep convolutional recurrent neural network architectures, specifically a combination of auto-encoder and a convolutional long short term memory recurrent network (convLSTM) [25]. Although autoregressive structure excels at temporal regression tasks, they will definitely encounter a time-dependent error accumulation problem. The recurrent neural network might become computationally demanding for the long time-serial models [26], although the time-dependent error accumulation problem hardly occurs.

To address the aforementioned drawbacks, we use an alternative deep convolutional neural network, namely time-conditioning residual U-Net (cR-U-Net), based on an integration of auto-encoder structure and the well-known Residual U-Net architecture, which was originally proposed for bio-medical image segmentation [21]. The residual U-Net architecture has demonstrated to be very effective for the cross-domain regression problem [19], such as the mapping from the spatial parameter fields to simulated saturation field studied in the paper. To learn the temporal features of dynamic models, a time-conditioning feature, as an additional channel, is concatenated to the low-dimensional representation features after the encoder part [15]. The time feature is represented as a map of the same dimension as the low-dimensional representation features. Each element of the time feature map is equal to a specific time value. This is different from the above two approaches that use either autoregressive or recurrent structures to capture the time-serial dynamics and hence is capable of effectively mitigating the error accumulation and computational issue for the long time-serial models.

The configuration of cR-U-Net is guided by the literature [24] where a similar R-U-Net architecture is designed. This cR-U-Net architecture contains two symmetric units, e.g., an encoding path and a decoding path, to capture the spatial features of the input and the output images. In addition, we also design a transition unit, including a stack of residual blocks, for connecting the encoding and the decoding unit. Fig.6.1 illustrates a schematic diagram of the proposed cR-U-Net which synoptically displays the arrangement of encoding, transition and decoding unit.

Details of the architecture of the encoding, transition and decoding unit are depicted in Figs.6.2-6.4. The encoding unit shown in Fig.6.2(a) then takes the reshaped geological permeability maps as the inputs. A set of feature maps $\mathbf{F}_k(\mathbf{m})$ (k = 1, 2, 3, 4) are extracted from four consecutive convolutional blocks and will then be fed to the symmetric decoding unit. From $\mathbf{F}_1(\mathbf{m})$ to $\mathbf{F}_4(\mathbf{m})$, the maps gradually contains much more compressed and global features. Each convolutional block consists of three continuous operations ($Conv3D$-$BatchNorm$-$ReLU$), including a 3D convolutional layer ($Conv3D$), a

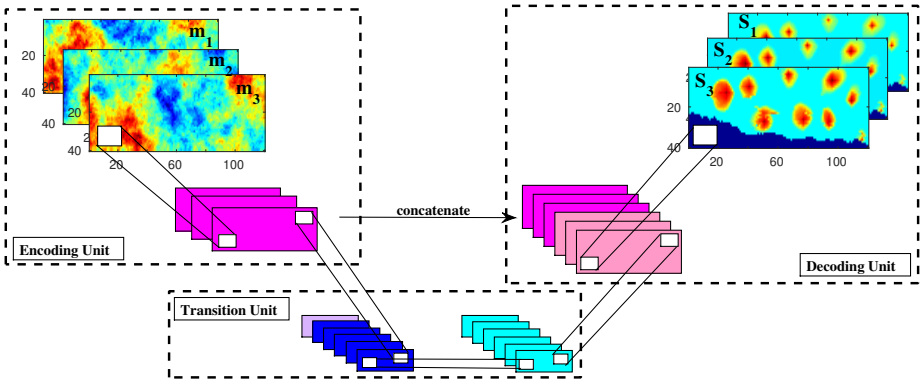


Figure 6.1: Schematic illustration of the cR-U-Net architecture for a 2D input image following the literature [24]. The cR-U-Net is composed of encoding unit, transition unit and decoding unit. The multi-scale features extracted in the encoding unit are concatenated with the upsampling features in the decoding unit to produce the final output. The transition unit concatenates and forwards the extracted multi-scale features and time feature, and then feeds them to the decoding unit. The gray shape in the transition unit represents the time feature map.

batch normalization layer ($BatchNorm$) and a rectified linear activation unit ($ReLU$), as displayed in Fig.6.2(b).

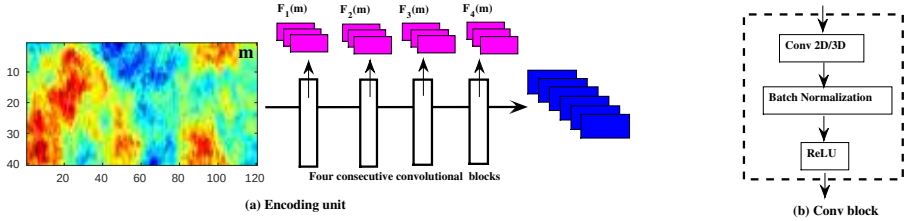


Figure 6.2: Schematic illustration of encoding unit. This unit consists of four convolutional blocks. The encoding unit accepts the logarithmic permeability image as an input. The generated multi-scale features $\mathbf{F}_k(\mathbf{m}) \in R^{N_{w,k} \times N_{h,k} \times N_{d,k}}$ ($k = 1, 2, 3, 4$), are sequentially fed to the decoding unit. Also the blue maps on the right side are the same as $\mathbf{F}_4(\mathbf{m})$.

The feature maps $\mathbf{F}_4(\mathbf{m})$ produced from the encoding unit are fed to a transition unit, see Fig.6.3(a). To learn the temporal features of the dynamic model, the time, as a conditional feature channel, is concatenated to the low-dimensional representation features, i.e., $\mathbf{F}_4(\mathbf{m})$, after the encoding part. This transition net is composed of five residual-blocks. The structure of the residual block, specifically residual convolutional ($resConv$) block, bypass the nonlinear layers through introducing an identity mapping. The nonlinear layer follows the standard structure of $Conv3D$-$BatchNorm$-$ReLU$ as illustrated in Fig.6.3(b). This special architecture of $resConv$ block can help cope with the gradient vanishing/explosion problem especially for the highly deeper network [27]. A residual block intentionally constructs connections between non-adjacent layers, designed to take full advantage of the multi-scale features in the output of the previous layers [20]. For example, a residual block with two convolutional layers is shown in Fig.6.3(b). The aim of transition unit is to pro-

duce feature maps $\mathbf{F}_5(\mathbf{m})$, which are the most complicated and informative feature maps containing both the spatial and temporal information. These maps will be then provided to the decoding unit.

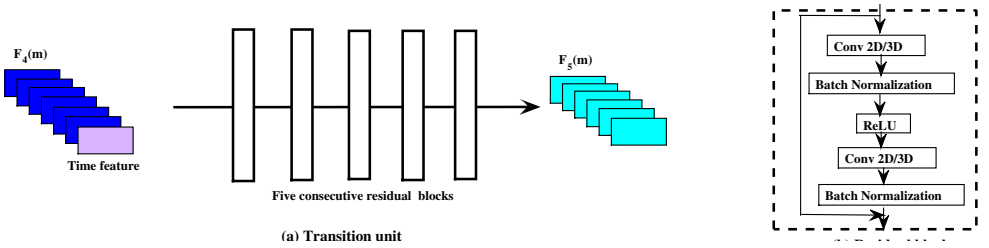


Figure 6.3: Schematic illustration of transition unit. A stack of five residual blocks is used to propagate the output of the encoding unit. In addition, the time feature as an additional channel is fed to this unit as well to capture the flow dynamic.

Fig.6.4 illustrates the structure of the decoding unit. In this unit, the produced feature maps $\mathbf{F}_5(\mathbf{m})$ from transition unit are gradually unsampled using four transposed 3D convolutional blocks, and then the output feature maps of them are concatenated with the previously extracted feature maps $\mathbf{F}_k(\mathbf{m})$ (k = 1, 2, 3, 4) in the corresponding encoding unit, respectively. Finally, the decoding unit generates the target quantities of interest, e.g., saturation maps, in the final output layer. The transposed 3D convolutional ($TConv3D$) block or upsampling block is applied here for the purpose of upsampling operations. In order to guarantee that the output $\hat{\mathbf{x}}$ is constrained to lie within the bound [0,1], the $Sigmod$ activation function is designed at the final output layer [28].

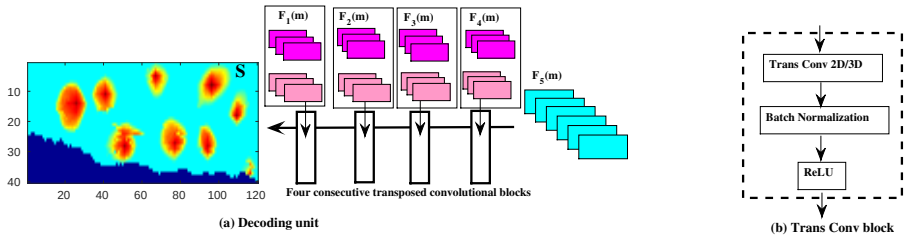


Figure 6.4: Schematic illustration of decoding unit. This unit consists of four transposed convolutional blocks corresponding to the encoding unit. Decoding unit accepts the extracted multi-scale features $\mathbf{F}_k(\mathbf{m})$ (k = 1, 2, 3, 4) to produce the target maps.

6.3.2. DATASET PREPARATION

In order to train the cR-U-Net architecture, the dataset has to contain mappings from the various permeability fields to seismic maps, i.e., spatially grid-wise saturation in this paper. In our proposed cR-U-Net surrogate model, the current output $\mathbf{x}^{i,n}$ depends only on the input permeability image $\mathbf{m}^i$ and time index t^n. We re-arrange the saturation snapshots obtained from one high-fidelity model simulation, i.e., saturation map ($\mathbf{m}^i$; $\mathbf{x}^{i,1}$,$\cdots$,$\mathbf{x}^{i,N_t}$), as N_t consecutive training samples

$$\{(\mathbf{m}^i;\mathbf{x}^{i,1},\cdots,\mathbf{x}^{i,N_t})\}_{i=1}^{N_s} \Rightarrow \{(\mathbf{m}^i,t^n;\mathbf{x}^{i,n})\}_{i=1,n=1}^{N_s,N_t} \tag{6.4}$$

and we expect that the time-dependent relationship between the geological permeability inputs and time-varying saturation outputs can be clearly captured in this time-conditional network structure. Finally, the total number of training samples fed to this cR-U-Net becomes $N_s \times N_t$.

6.3.3. TRAINING PROCEDURES

The choice of loss function is generally case-dependent. It is important to choose an appropriate loss function with the aim of improving the performance of designed network as much as possible. For this study, a loss function defined by the mean square error (MSE) is used for this cross-domain image regression problem.

$$L_{MSE}(\boldsymbol{\theta}) = \frac{1}{N_s N_t} \sum_{i=1}^{N_s} \sum_{n=1}^{N_t} \|\hat{\mathbf{x}}^{i,n} - \mathbf{x}^{i,n}\|_2^2 \tag{6.5}$$

After configuring the neural network architecture, we use the popular *Adam* optimizer to optimize the neural network parameters. Through using approximations of the first and second order moments of the gradients, the *Adam* computes learning rates for different parameters in an adaptive manner. The learning rate which controls the updates of model parameters at each iteration is 5×10^{-3}. A learning rate scheduler which drops ten times on plateau training is also adopted to stabilize the training process. Details about the iterative scheme of the *Adam* optimizer will be given in the next section. An overview of the full workflow is provided in Algorithm 3 for the offline training stage. This network was built and trained using the open-source deep learning package PyTorch [17].

Once the neural network is trained, the predictions corresponding to new inputs are straightforward. Given an arbitrary input, repeated implementations of Eq.6.3 are used to evolve this time-varying states for all N_t timesteps. Specifically, the saturation $\hat{\mathbf{x}}^n$ at the n-th timestep is sequentially predicted by providing the permeability input $\mathbf{m}$ and the time t^n. The computational cost for the new predictions almost can be neglected as compared to additional high-fidelity model simulations.

Algorithm 3: The cR-U-Net architecture: Offline training stage

```
Set an initial network trainable parameters θ^0 ;
while epoch < n_epoch do
    while minibatch < N_s × N_t do
        Calculate the gradient ▽ L_MSE (θ) using automatic-difference tool;
        Update the parameter solution using Adam(θ) → θ ;
        Evaluate the loss function function L_MSE (Eq.6.5);
        Check convergence;
    end
end
Return the optimal parameters θ
```

6.3.4. GRADIENT APPROXIMATION USING AUTO-DIFFERENTIATION

The use of DNN surrogate model in SHM enables the entire SHM procedure completely implemented using the deep learning packages. It can benefit from the deep learn-

ing practice of using minibatches of data with the stochastic-gradient descent (SGD) optimizers instead of the more conventional approach of applying an optimizer such as the FGD, e.g., as defined in Eq.6.2. The robustness and faster convergence of SGD optimizers have been extensively demonstrated in the deep-learning community [29]. In this work, we explore the potential of applying SGD to improve seismic history matching performance. The attractive benefits of using SGD stem from its efficiency for problems with a large number of data and parameters and its effectiveness for problems with noisy gradients.

Compared to FGD during which we have to run through all data points at N_t measured time steps for a single update of the parameters in a particular iteration, SGD enables us to use the data points at only one measured time step (e.g., $n \in [1, N_t]$) or a subset **bz** of $[1, N_t]$ (e.g., a collection of several random selections of indices from $[1, N_t]$) to update the parameters. If the size of dataset is huge, e.g., the spatially dense seismic data frequently collected at some time steps, it may take too long for FGD to minimize the objective function because we need use the full dataset for updating the parameters in each iteration. In contrast, SGD uses only one or a subset of data and it starts update the parameters rapidly.

We introduce the stochastic gradient as an approximation of the full gradient that can be used to speed up calculations and algorithm convergence. After training the cR-U-Net surrogate model, the approximated stochastic objective function, denoted as J_N, can be defined using a subset **bz** with trained neural network model correspondingly

$$\begin{aligned} J_N = &\frac{1}{2}(\mathbf{m}-\mathbf{m}_b)^T {\mathbf{R}_m}^{-1}(\mathbf{m}-\mathbf{m}_b) \\ &+ \frac{1}{2}\sum_{n\in\mathbf{bz}} [\mathbf{d}^n_{obs} - \hat{\mathbf{h}}^n(\mathbf{m}, t^n, \boldsymbol{\theta})]^T [\mathbf{R}^n_{obs}]^{-1} [\mathbf{d}^n_{obs} - \hat{\mathbf{h}}^n(\mathbf{m}, t^n, \boldsymbol{\theta})] \end{aligned} \tag{6.6}$$

and its corresponding stochastic gradient with respect to the parameters **m** using the sensitivity or Jacobian matrix

$$\mathbf{g}_{N,1} = {\mathbf{R}_m}^{-1}(\mathbf{m}-\mathbf{m}_b) - \sum_{n\in\mathbf{bz}} [\frac{\partial \hat{\mathbf{h}}^n}{\partial \mathbf{m}}]^T [\mathbf{R}^n_{obs}]^{-1} [\mathbf{d}^n_{obs} - \hat{\mathbf{h}}^n(\mathbf{m}, t^n, \boldsymbol{\theta})] \tag{6.7}$$

The computation of objective function gradient can be realized based on two methods. (1) The first method uses the model sensitivities $\frac{\partial \hat{\mathbf{h}}^n}{\partial \mathbf{m}}$, to compute the gradient $\mathbf{g}_{N,1}$, e.g., Eq.6.7. Each element of this matrix is the derivative of the surrogate model outputs, $\hat{\mathbf{h}}^n(\mathbf{m})$, with respect to the parameters **m**. e.g., grid-based geological permeability. (2) The second method directly calculates the gradient of the objective function J_N with respect to **m**, denoted to be $\mathbf{g}_N$. Deep-learning packages provide sufficient built-in modules to efficiently compute the function with a least-square formula, e.g., Eq.6.6, and its gradient using AD. That is, the second method does not explicitly involve the computation of sensitivities. In general, $\mathbf{g}_N$ is often cheaper to calculate than that of model sensitivities $\frac{\partial \hat{\mathbf{h}}^n}{\partial \mathbf{m}}$ in terms of the computational complexity.

Once the SHM algorithm is implemented in the deep learning packages, some sophisticated evaluations, e.g., objective function evaluation (Eq.6.6) and gradient approximation (Eq.6.7) involving high-dimensional matrix-matrix/vector multiplications, can be quickly computed. For example, they can be executed on multi-CPUs and GPUs, and run in parallel across a distributed memory computer cluster. All these features form the main motivation for investigating the use of DNN surrogate model for the SHM problem.

6.3.5. STOCHASTIC GRADIENT DESCENT OPTIMIZATION

To minimize surrogate objective function J_N, a standard stochastic gradient descent (SGD) algorithm updates the parameters at k^{th} optimization step

$$\mathbf{m}^{k+1} \leftarrow \mathbf{m}^k - \varepsilon^k \mathbf{g}_N^k \tag{6.8}$$

The aforementioned *Adam* optimization algorithm has recently seen broader applications in the community of deep learning. *Adam* leverages past gradient information to retard the descent along large gradients. This information is stored in the momentum vector $\mathbf{u}$ and squared element-wise gradient vector $\mathbf{v}$ as

$$\begin{aligned} \mathbf{u}^k &= b_u \mathbf{u}^{k-1} + (1-b_u)\mathbf{g}_N^k; \quad \hat{\mathbf{u}^k} = \frac{\mathbf{u}^k}{1-\mathbf{u}^k} \\ \mathbf{v}^k &= b_v \mathbf{v}^{k-1} + (1-b_v)[\mathbf{g}_N^k]^2; \quad \hat{\mathbf{v}^k} = \frac{\mathbf{v}^k}{1-\mathbf{v}^k} \end{aligned} \tag{6.9}$$

where, $\hat{\mathbf{u}^k}$ and $\hat{\mathbf{v}^k}$ are the unbiased momentum and squared gradient vectors, respectively. The gradient descent step proceeds as follows

$$\mathbf{m}^{k+1} = \mathbf{m}^k - \zeta^k \frac{\hat{\mathbf{u}^k}}{\sqrt{\hat{\mathbf{v}^k}} + \epsilon} \tag{6.10}$$

The above update is performed element-wise and ϵ is a small number (e.g. 10^{-8}) to avoid any division by zero in the implementation. Here we use commonly recommended default parameters values of ε^k=0.001, b_u=0.9, b_v=0.999 and ϵ=10^{-8}.

For the implementation of the SHM procedure with the deep neural network surrogate, we have used the open access deep learning package PyTorch to build a hybrid CPU/GPU computing framework, where the training data are generated using the open-source simulator OPM-$Flow$ [30] running on CPUs while both surrogate model training and seismic history matching are implemented on GPUs. All these features prompt the application of our proposed methodology to practical cases.

6.4. EXPERIMENTS AND DISCUSSION

6.4.1. DESCRIPTION OF MODEL SETTINGS

In the numerical experiment, we test our proposed surrogate-assisted seismic history matching approach on a 3D benchmark model used in Chapter 5. One of the realizations is chosen to be the reference model as illustrated in Fig.6.5. Details about reservoir geometry, rock properties, fluid properties, and well controls also can be found in Chapter 5.

We show four Gaussian-distributed realizations of logarithmic permeability and their corresponding predictions of water saturation in Fig.6.6. The geological model realizations and the saturation corresponding to day 2700 and day 5400 are consecutively displayed in this figure. It can be seen that the heterogeneity of the permeability fields results in fairly large differences between the areas that are flooded in different model realizations.

In order to quantify the history matching results, we define an error measure based on

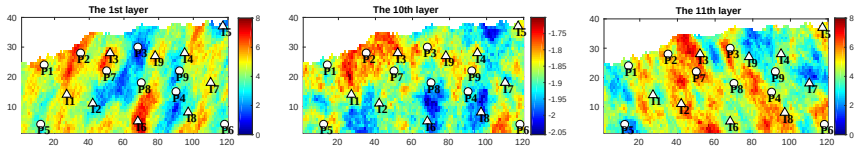


Figure 6.5: The illustrations of the spatial logarithmic permeability for the 1st, 10th and 11th vertical layer. The triangles and circles denote the injectors and producers, respectively.

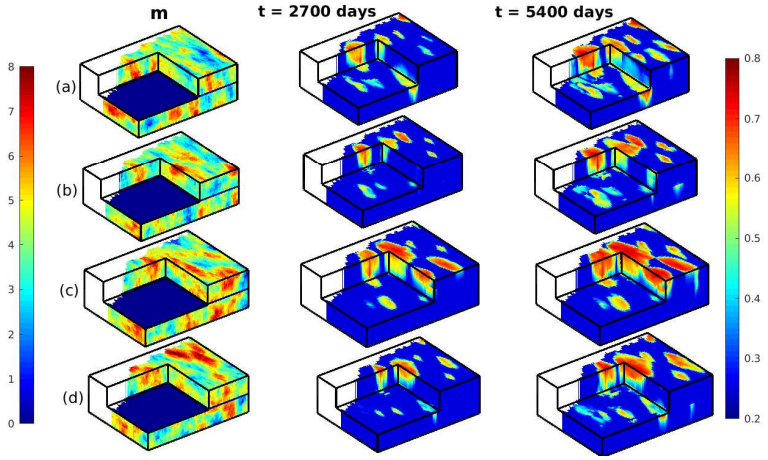


Figure 6.6: Predictions of the time-varying saturation maps corresponding to four random realizations of logarithmic permeability fields.

parameter misfits $e_{\mathbf{m}}$ as follows,

$$e_{\mathbf{m}} = \sqrt{\frac{\sum_{i=1}^{N_m}(\mathbf{m}_{true}^i - \mathbf{m}_{upt}^i)^2}{N_m}} \tag{6.11}$$

where, $\mathbf{m}_{true}^i$ and $\mathbf{m}_{upt}^i$ denote the logarithmic permeability value of the gridblock i from the reference model and updated model, respectively.

In addition, the optimal objective function value J also should satisfy the following inequality

$$J \leqslant 3\,N_{obs} \tag{6.12}$$

where, N_{obs} is the total number of measurements. This criterion can be used to quantify the accuracy of history matching results.

To evaluate the quality of the parameter estimation results, we will compare the value of the final objective function against both the criterion (Eq.6.12) and the reference objective function values for the true model. Reconstructed parameter maps will provide a visual indication of quality of the solution. For all approaches we will list the computational cost expressed in terms of the number of HFM simulations.

6.4.2. CONFIGURATION OF CR-U-NET ARCHITECTURE

The details about cR-U-Ne architecture are described in Table 6.2. The input is the log-

Table 6.1: he hyper parameter settings for training cR-U-Net model.

Training size (N_s)	100,300,500,800,1000
Re-organized training size ($N_s \times N_t$)	1000,3000,5000,8000,10000
Testing size	200
Learning rate	0.001
Optimizer	*Adam*
Batch size	10
Number of epochs	100

arithmic permeability **m**. For the first step, we reshape the geological permeability fields into three-dimensional images with the size of N_x=40, N_y=120 and N_z=20. In the decoding part, a set of feature maps $\mathbf{F}_k(\mathbf{m})$ (k = 1, 2, 3, 4) are extracted from four consecutive convolutional blocks. From $\mathbf{F}_1(\mathbf{m})$ to $\mathbf{F}_4(\mathbf{m})$, the maps gradually contains more compressed features, as illustrated in Fig.6.7 After the decoding part, the input is compressed in 128 feature maps with the size 2 × 7 × 1 and then concatenated with an additional time feature map with the same size 2 × 7 × 1. These 129 feature maps are fed to the transition unit for producing 128 feature maps with the constant size 2 × 7 × 1. Finally, these 128 feature maps are provided to the decoding unit for the output, e.g., saturation, with the size of 40 × 120 × 20.

Five different number of training samples, i.e., N_s = 100, 300, 500, 800, and 1000 samples, are used to assess the performance of cR-U-Net with respect to the training sample size. The model is run for 5400 days, and training data are collected at N_t=10 intervals of 540 days each. After reorganizing the dataset, there are in total 1000, 3000, 5000, 8000 and 10000 training samples correspondingly. The hyper parameter settings for training the cR-U-Net model are listed in Table 6.1.

Table 6.2: Illustration of cR-U-Net architecture. The size of a 3D image is N_x=40, N_y=120 and N_z=20 in this case-study.

Unit	Layer	Output Size
Encoder unit	Input (geological permeability images)	(40, 120, 20, 1)
	Conv3D-BatchNorm-ReLU, 16 kernels of size (3,3,3,2)	(20, 60, 10, 16)
	Conv3D-BatchNorm-ReLU, 32 kernels of size (3,3,3,16)	(20, 60, 10, 32)
	Conv3D-BatchNorm-ReLU, 64 kernels of size (3,3,3,32)	(10, 30, 5, 64)
	Conv3D-BatchNorm-ReLU, 128 kernels of size (3,3,3,64)	(10, 30, 5, 128)
Transition unit	Input (outputs of encoder unit + an additional time feature)	(10, 30, 5, 129)
	resConv, 128 kernels	(10, 30, 5, 128)
	resConv, 128 kernels	(10, 30, 5, 128)
	resConv, 128 kernels	(10, 30, 5, 128)
	resConv, 128 kernels	(10, 30, 5, 128)
	resConv, 128 kernels	(10, 30, 5, 128)
Decoder unit	Input (outputs of transition unit)	(10, 30, 5, 128)
	TConv3D-BatchNorm-ReLU, 128 kernels of size (3,3,3,128),	(10, 30, 5, 128)
	TConv3D-BatchNorm-ReLU, 64 kernels of size (3,3,3,128),	(20, 60, 10, 64)
	TConv3D-BatchNorm-ReLU, 32 kernels of size (3,3,3,64),	(20, 60, 10, 32)
	TConv3D-BatchNorm-ReLU, 16 kernels of size (3,3,3,32),	(40, 120, 20, 16)
	Conv3D-Sigmoid, 1 kernels of size (3,3,3,16), stride 1	(40, 120, 20, 1)

6.4.3. ASSESSMENT OF SURROGATE MODEL QUALITY

The performance of the cR-U-Net model can be investigated by comparing the predefined metrics, e.g., the field-average relative error γ_s, relative to the number of training

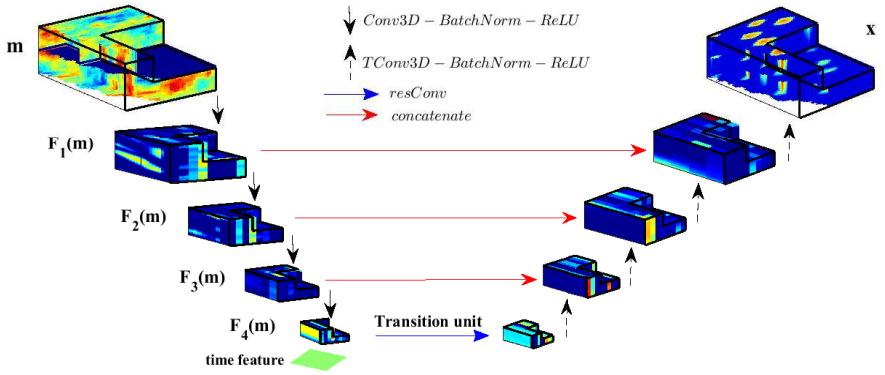


Figure 6.7: Schematic illustration of the cR-U-Net architecture using the 3D SAIGUP geological models as inputs.

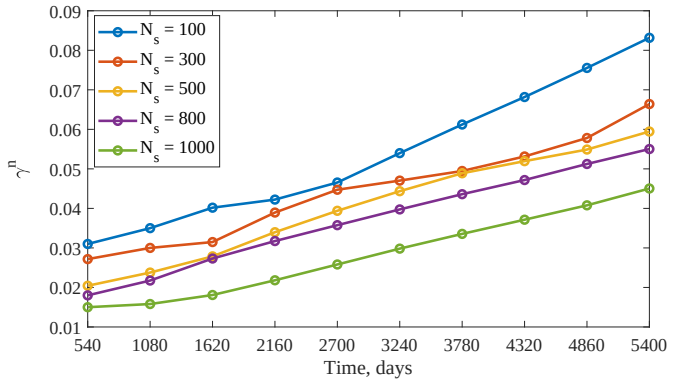


Figure 6.8: Comparison of the field-average relative error in saturation at all N_s =10 time intervals over the full N_{test} = 200 test samples. The cR-U-Net surrogate models are trained using N_s = 100, 300, 500, 800, and 1000 samples, respectively.

samples N_s for the quantities of interest, i.e., saturation values. Fig.6.8 shows the γ_s value obtained on the training ensemble of N_{test} = 200 realizations for the predicted saturation. The γ_s metrics will gradually increase as the time propagation, which reflects the degraded quality of network models. It has been demonstrated that training cR-U-Net model using N_s = 1000 samples significantly improve the network quality through achieving relatively low γ_s values. Through observing the evaluated relative errors in saturation maps over the 200 random test samples, we find the overall field-average relative errors γ_s are 2.83% and 5.37% for 1000 and 100 training samples, respectively. The small γ_s values significantly indicate a high degree of accuracy in the saturation maps predicted from the cR-U-Net model.

In addition, the plots of the time-varying saturation predictions from the two approaches are used to visually assess the accuracy of the trained network. Fig.6.9 shows the predicted saturation distribution at two time-instances, i.e., day 2700 and day 5400, corresponding to the cR-U-Net models trained using N_s = 100 and 1000 samples, respectively. It clearly can be seen that the cR-U-Net surrogate model is capable of predicting the water satura-

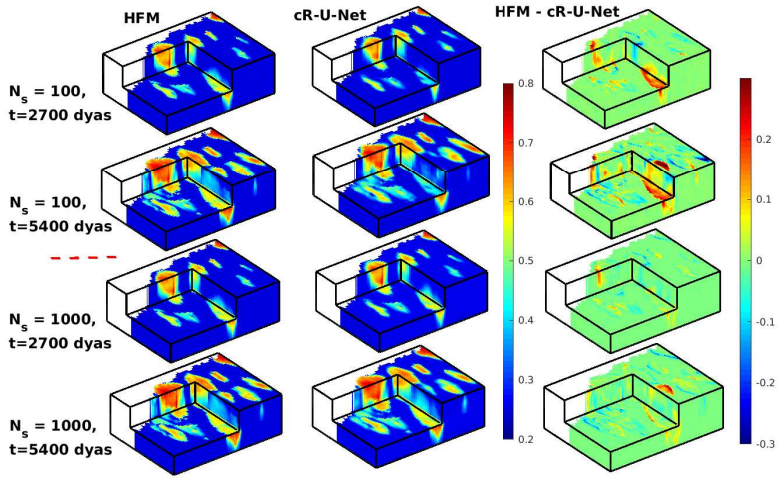


Figure 6.9: Illustration of time-varying phase saturation predicted from cR-U-Net, HFM and their absolute errors at day 2700 and day 5400. The cR-U-Net models are trained using N_s = 1000 and 100 training samples, respectively.

tion profile with a high accuracy. In addition, the corresponding relative saturation error γ_s at these two time instances are 3.92% and 2.87%, respectively. Thus, the results in Fig.6.9 can be considered to be representative in terms of surrogate model accuracy.

In terms of computational effort, the run time for a single HFM simulation for this case is about 250 seconds on a machine with i5-4690 Intel CPUs (4 cores, 3.5GHz) and 24 GB memory using OPM-*Flow* simulator. The cR-U-Net is trained using a NVIDIA Tesla P100 GPU card. The simulation of trained neural network requires about 0.1 seconds. However, the training stage of this network is computationally intensive, which includes the generation of training samples and additional overhead. It would be not useful to construct the cR-U-Net surrogate model unless it is to be used in the situation where numerous simulations are required. Because many simulations are required in seismic history matching applications, the cR-U-Net surrogate models could be applicable in this context. Although the training time can vary by case, it is just a small fraction of the time required in the conventional seismic history matching where numerous HFM simulations must be performed. The use of surrogate model in conjunction with the gradient-based seismic history matching is presented in the following section.

6.4.4. HISTORY MATCHING RESULTS USING THE CR-U-NET SURROGATE

The measurements used in this case correspond to the saturation values from the 'true' model simulation for every 540 days, mimicking the collection of a large amount of data, with a total of 787200 measurements, from ten time instances. Normal distributed independent measurement noise with a standard deviation equal to 5% of the 'true' data value was added to all measurements. We will demonstrate the feasible applicability of our proposed DNN-assisted history matching approach to assimilate a large number of measurements.

STUDY OF THE BASE-CASE

After training the cR-U-Net surrogate model, the gradients of the objective function with respect to the logarithmic permeability are computed analytically using AD, and then the $Adam$ optimizer is implemented to update the geological parameters efficiently. Batch size of one (i.e. we randomly select data from one measured timestep) is used in this base-case study. In addition to the default parameters for the $Adam$ optimizer, the learning rate ε is set to be 0.05. We do not fix the maximum number of iterations since the cost of running the cR-U-Net models is almost negligible. The minimization algorithm is considered to have sufficiently converged when either the criterion for minimum change of objective function or the logarithmic permeability at two consecutive iterations is satisfied. These two stopping criteria are set to be 0.0001 and 0.001, respectively.

Table 6.3 and Fig.6.10 display the history matching results, including the final objective functions, parameter misfits $e_{\mathbf{m}}$ and the required number of HFM simulations. The cR-U-Net models are trained using N_s = 100,300,500,800 and 1000 samples, respectively. The surrogate objective functions J_N are iteratively minimized as shown in Fig.6.10. Since the cR-U-Net cannot exactly represent the original HFM using a finite training set, we also show the high-fidelity objective functions J using original HFM in Table 6.3. It can be seen that the values of J are slightly larger than that of J_N due to the fact that the cR-U-Net surrogate model introduces additional approximation errors. In order to verify the history matching results, we also display the metric (Eq.6.12) and the reference objective function value. The final objective functions are very close to the reference value. After assimilating a large amount of measurements, the parameter misfits $e_{\mathbf{m}}$ have been significantly decreased, e.g., from 1.1141 to 0.4123 corresponding to N_s = 1000 training samples. The accuracy of history matching results gradually improve as the number o training samples as indicated by the reductions of $e_{\mathbf{m}}$ values.

Fig.6.11 shows the 2D horizontal cross sections of the updated logarithmic permeability fields for the 1st and 11th vertical layer, which can be used to assess the accuracy of the history matching results. The reference model is almost reconstructed by assimilating a large number of measurements. Surprisingly, the cR-U-Net model trained with only N_s = 100 samples is already capable of achieving a posterior realization which is visually close to the reference model. These results reveal the effectiveness of the proposed cR-U-Net surrogate model for addressing seismic history matching problems, since it can provide a useful direction to update the uncertain parameters correctly in this case-study.

To further illustrate the accuracy of the updated reservoir models, the predictions of saturation at both two specific time instances (e.g., day 2700 and day 5400) before and after history matching are depicted in Fig.6.12. Compared to the initial models, the saturation predictions of the updated models are visually close to the predictions of the true model. The correlation coefficients have been increased from the initial 70% to the final 98%.

EFFECT OF THE INITIAL MODELS

The gradient-based optimization methods inevitably get stuck in local minima for any non-convex optimization problems, such as seismic history matching in this study. Fortunately, without running additional HFM simulations at the history matching stage, the proposed cR-U-Net surrogate model enables us to efficiently generate multiple posterior solutions through starting from different initial models. Fig.6.13 depicts the parameter misfits $e_{\mathbf{m}}$ after the history matching corresponding to three different initial models. The differences among these three parameter misfits $e_{\mathbf{m}}$ indicate that several local minima

Table 6.3: Summary of history matching results using cR-U-Net surrogate model trained with different sample size. These are the results for SGD with batch size 1. We also show the tolerance (Eq.6.12) and the reference objective function values for the true model (reflecting the impact of the data noise).

N_s	$e_{\mathbf{m}}$	$J_N \times 10^6$	$J \times 10^6$	Number of HFM simulations
100	0.7130	0.9262	1.0224	100
300	0.6075	0.8425	0.8754	300
500	0.5087	0.8376	0.8577	500
800	0.4368	0.7083	0.7235	800
1000	0.4123	0.6720	0.7011	1000
Reference	0.4027			
Tolerance	2.362			

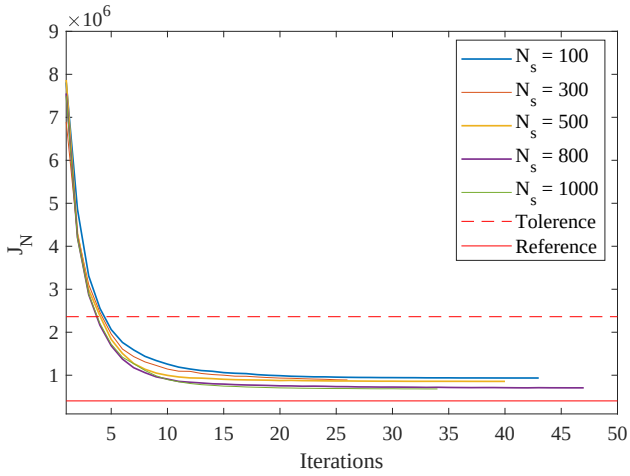


Figure 6.10: The surrogate objective function J_N as a function of iterations. The cR-U-Net surrogates are trained using N_s = 100,300,500,800 and 1000 training samples, respectively. The tolerance and the reference objective function values for the true model (reflecting the impact of the data noise) are shown in red bash and bold lines.

have been found for different initial models. It can be further visually revealed in Fig.6.14 that the updated logarithmic permeability fields corresponding to these three initial models are significantly different. We also compare in Fig.6.15 the predictions of the well water injection rate (WWIR) and well water-cut (WWCT) at all nine injectors and nine producers before and after history matching. Clearly, it can be observed that the spread of both WWIR and WWCT is significantly reduced towards the predictions of the reference model.

In this study, the total number of HFM simulations is taken as an indicator of the computational cost, since the GPU time for running cR-U-Net is negligible compared to that for running HFM model. In order to assess the model uncertainty, we can generate multiple posterior models through repeatedly implementing the gradient-based optimization. We only need to run HFM simulations in the training stage and SHM process does not involve additional HFM simulations. Our proposed surrogate-assisted history matching method is highly efficient since the training of the cR-U-Net requires only a small number of HFM simulations. For example, the cR-U-Net model trained with N_s = 100 samples is already capable of achieving good history matching results in this case-study.

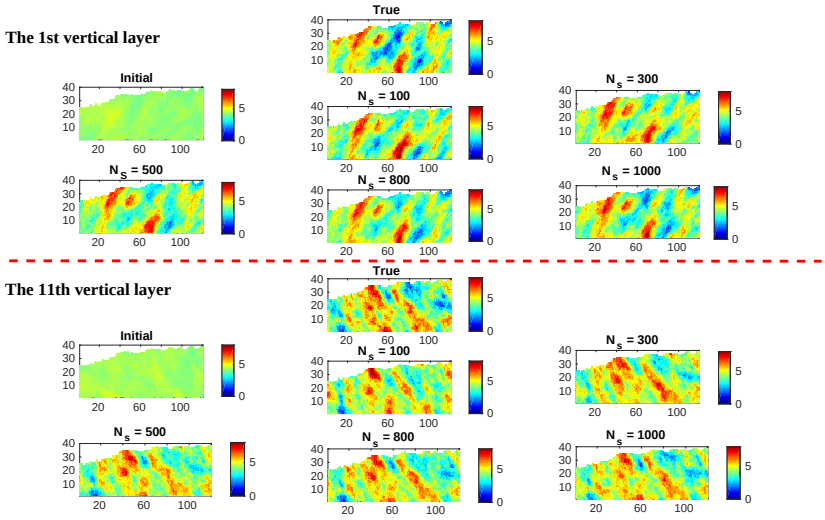


Figure 6.11: 2D horizontal cross sections of the updated logarithmic permeability under different number of sample size N_s = 100,300,500,800 and 1000 from the cR-U-Net. The 1st and 11th vertical layer are shown here.

FGD VERSUS SGD OPTIMIZERS

Since our proposed cR-U-Net model is not an exact representation of the high-fidelity model, incorporating the cR-U-Net into the SHM procedure inevitably introduces an additional source of uncertainty due to the approximation errors. In this section, the *Adam* optimizer is used to demonstrate the effectiveness of SGD optimization in reducing the approximation errors. The entire seismic data measured at 10 time instances is divided using five different batch size, e.g., 1,2,3,4 and 5, respectively. Taking the batch size 2 as an example, two random time index are selected from the 10 time steps, and then the corresponding seismic data at these two time instances are used to update the parameters in one iteration. That is to say, the entire seismic data will be fully used to update the parameters in 5 iterations.

Fig.6.16 shows the evolution of the surrogate objective function J_N with respect to the iterations and batch size for the $Adam$ optimizer. It clearly can be seen that using a smaller batch size can achieve a faster convergence rate. For example, about 35 and 118 iterations are required for a batch size 1 and 5, respectively. By contrast, the FGD optimizer requires much more iterations, e.g., after 220 iterations it still has a larger objective function value.

Fig.6.17 displays an ensemble of parameter misfits $e_{\mathbf{m}}$ corresponding to different batch size and the FGD optimization. It is noticeable that the SGD optimization achieves larger reductions of the parameter misfits than that of the FGD optimization. SGD is more effective to address surrogate-assisted seismic history matching with approximation errors. As a hyper parameter for the $Adam$ optimizer, we should chose the batch size with much care. The evolution of the parameter misfits $e_{\mathbf{m}}$ corresponding to three initial models are highlighted as well in Fig.6.17. It can be concluded that the optimal batch size depends on the initial model and the quality of the cR-U-Net model. For example, for the cR-U-Net trained with N_s = 1000 samples, batch size 3, 2 and 5 generate the best history matching results for these three initial models, respectively.

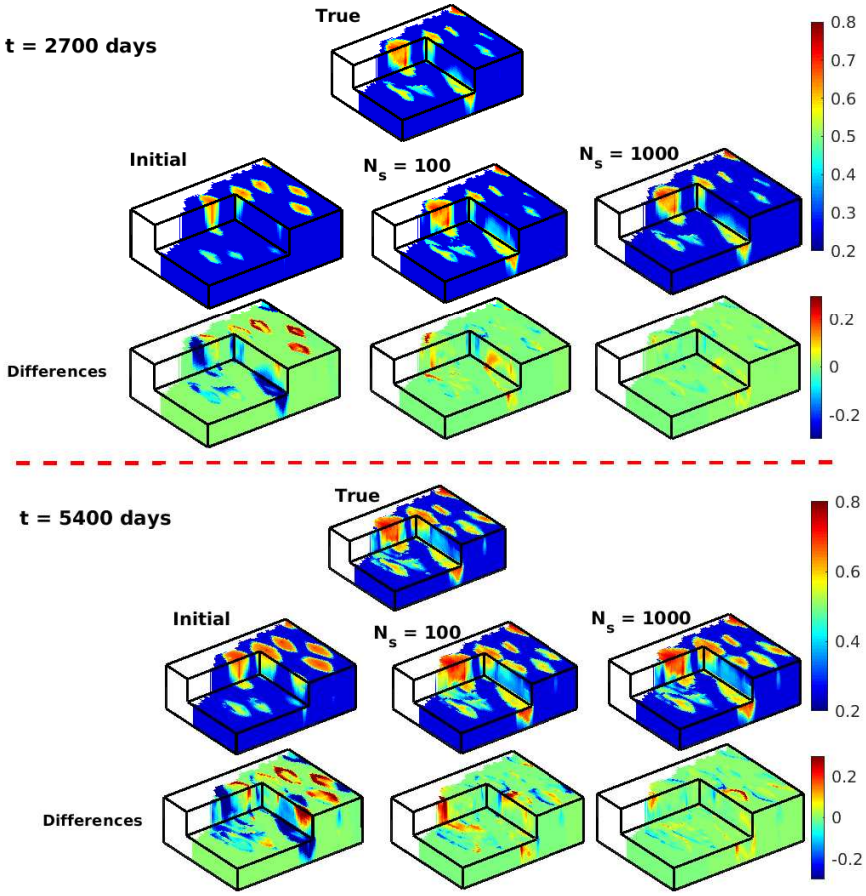


Figure 6.12: Predictions of water saturation before and after history matching and their absolute errors at 2700 days and 5400 days of production. The cR-U-Net models are trained using N_s = 100 and 1000 training samples, respectively.

Fig.6.18 depicts the updated logarithmic permeability fields corresponding to one random initial model using SGD and FGD optimization. The SGD optimization obtains similar parameter fields corresponding to different batch size, which are very consistent with the reference model, while the FGD optimization visually generates a different parameter field. SGD has proven highly successful in the process of training DNN. There is a clear similarity in the way how DNN and SHM iteratively update parameters, e.g., geological parameters in the seismic history matching. The stochastic optimization methods might easily move away from saddle points, while the FGD methods tend to get stuck in them. That is to say, our adopted stochastic optimization method might be more possible to escape from local minima. By contrast, the approximated gradients by FGD are inevitably fraught with approximation errors of cR-U-Net surrogate model.

6.4.5. cR-U-NET VERSUS SUBDOMAIN POD-TPWL SURROGATE

Since both the DNN surrogate and our previously proposed projection-based reduced-

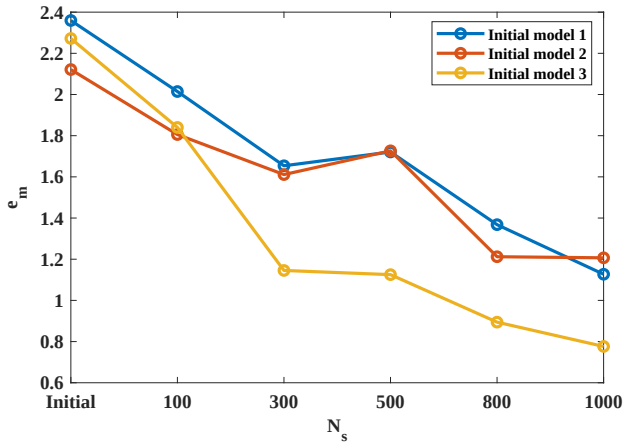


Figure 6.13: Illustration of the parameter misfits $e_{\mathbf{m}}$ with respective to the number of training samples N_s corresponding to three random initial models.

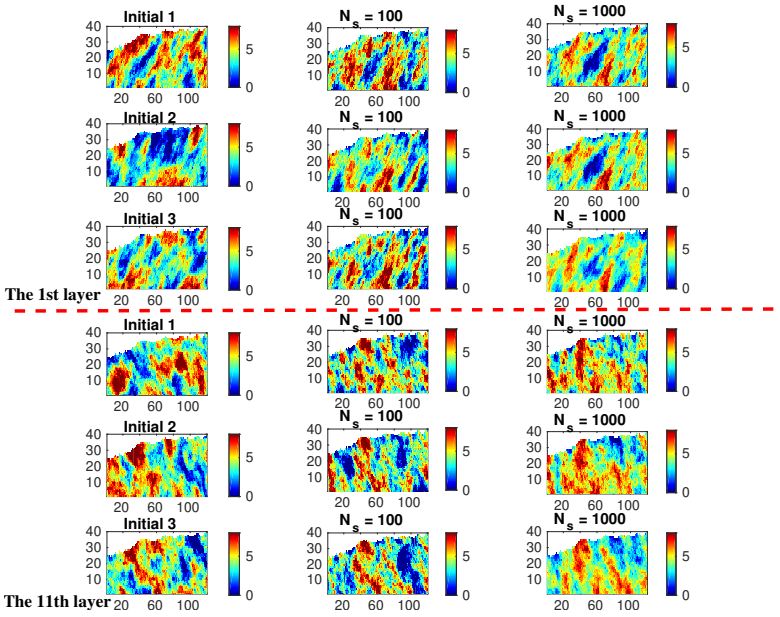
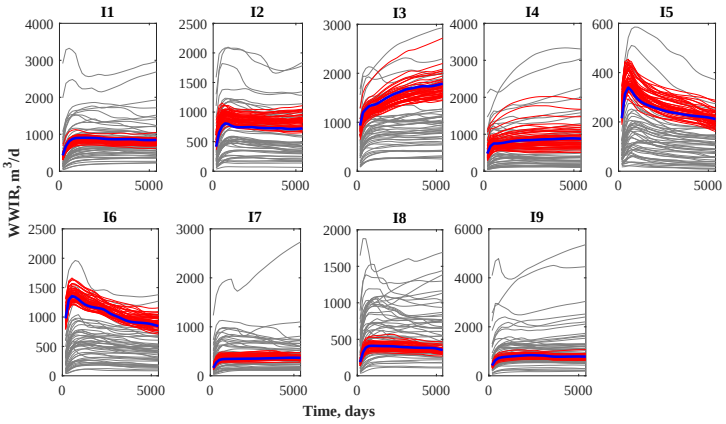
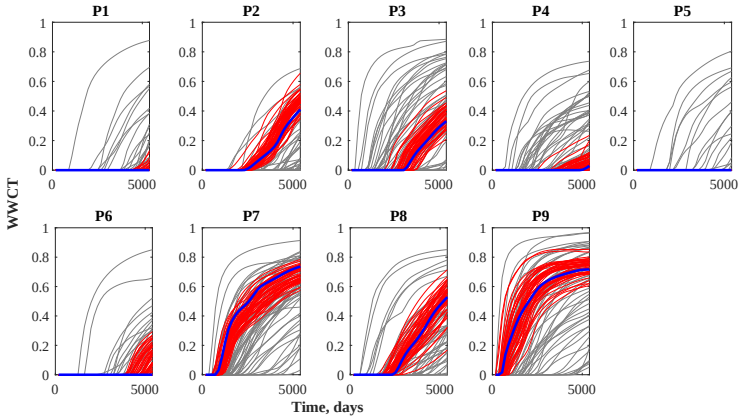


Figure 6.14: 2D horizontal cross sections of the updated logarithmic permeability corresponding to three random initial models. The cR-U-Net models are trained using N_s = 100 and 1000 samples, respectively.

order modeling, e.g., subdomain POD-TPWL, are intentionally used for the gradient-based history matching, a comparative study between them is conducted in this section. More details about the derivation of subdomain POD-TPWL and model-reduced adjoint approach for gradient approximation can be found in Chapter 5. In this case-study, we choose a fixed domain decomposition scheme, e.g., (3×4, 2×3) decomposition. The seismic data are measured corresponding to the saturation values from the 'true' model simu-



(a) Well water injection rate



(b) Well watercut

Figure 6.15: Forecast of the well water injection rate rate and well water-cut of all nine injectors and nine producers: dark-gray lines-initial models, blue lines-reference model, red lines-updated model using the cR-U-Net trained with N_s = 1000 samples.

lation after 2700 days and 5400 days of production, which results in total 157440 measurements. The noisy measurements for theses two time instances are shown in Fig.6.19.

COMPARISON OF SURROGATE MODEL QUALITY

Fig.6.20 displays the dependence of the field-average relative error γ_s with respect to domain decomposition strategy, testing interval and number of training samples. The testing interval represents the maximum discrepancy between the testing model and the reduced-order linear model. It can be seen from Fig.6.20(a) that increasing the testing interval deteriorates the accuracy of the subdomain POD-TPWL surrogate model, e.g., from 1.52% to 5.04%, for the (3×4, 2×3) domain decomposition scheme. The cR-U-Net is almost not sensitive to the testing interval, which however has a significant influence on the subdomain POD-TPWL surrogate model. For the testing model sampled from a small

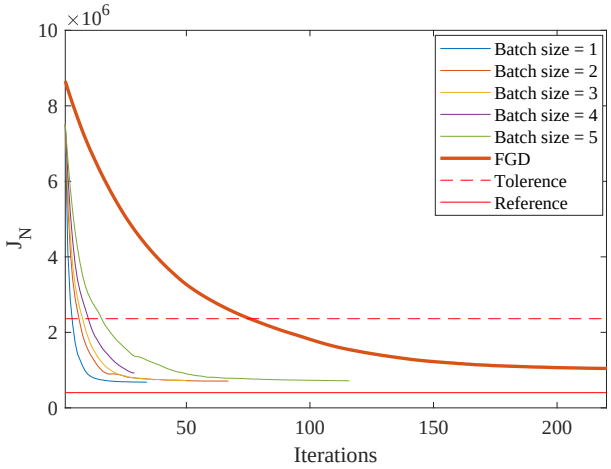
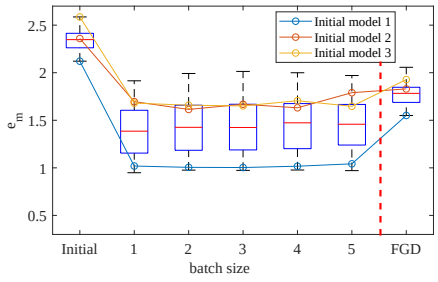


Figure 6.16: Evolution of surrogate objective function J_N as a function of iterations and batch size for the $Adam$ optimizer. The cR-U-Net surrogate models are trained with 1000 samples.

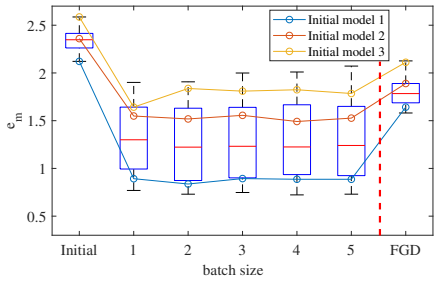
testing interval, e.g., [-0.1, 0.1], these two surrogate models almost obtain comparable accuracy, however, the accuracy of the subdomain POD-TPWL decreases gradually as the testing interval increases. More deep insights about these two types of surrogate models are worth explaining. Subdomain POD-TPWL constructs reduced-order linear model around a specific trajectory using the first-order Taylor expansion. If the testing model is far away from this trajectory, the accuracy inevitably will deteriorate. This leads to the necessity in frequently reconstructing subdomain POD-TPWL once the new testing models have large discrepancies from the current reduced-order linear model. By contrast, the cR-U-Net actually constructs a global surrogate model based on the entire training data and hence retraining cR-U-Net surrogate model is not strictly required. Definitely, filling the entire parameter space as much as possible using a large number of training samples, e.g., 1000 in this study, substantially improves the accuracy.

COMPARISON OF HISTORY MATCHING RESULTS

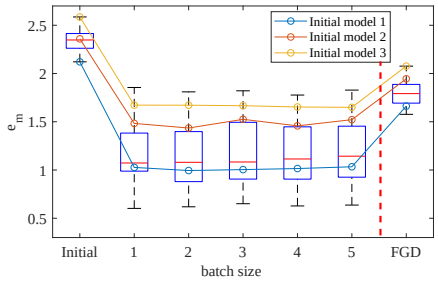
Fig.6.21(a) shows the evolution of objective function values as a function of out-loop iterations for the subdomain POD-TPWL approach. In this base-case study, (3×4, 2×3) 3D domain decomposition has been almost convergence after 12 out-loop iterations. Fig.6.21(b) and Table 6.4 display the final objective functions, the number of HFM simulations and the parameter misfits $e_{\mathbf{m}}$ for the cR-U-Net and subdomain POD-TPWL, respectively. Although the subdomain POD-TPWL obtains comparable and even smaller final objective functions than that of cR-U-Net surrogate models, the parameter misfits $e_{\mathbf{m}}$ are relatively larger. For example, the final objective functions are 0.1689×10^5 and 0.3890×10^5 for the subdomain POD-TPWL and cR-U-Net trained with 300 samples, respectively, the parameter misfits $e_{\mathbf{m}}$, however, are 0.6534 and 0.6370 correspondingly. This result indicates that subdomain POD-TPWL is noticeably susceptible to the data noise and hence produces a result, which is overfitting the noisy measurements. By contrast, the proposed cR-U-Net with SGD optimizer yields small parameter misfits $e_{\mathbf{m}}$, which are also gradually decreased as the number of training samples.



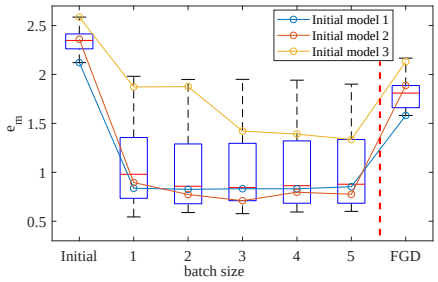
(a) N_s=100



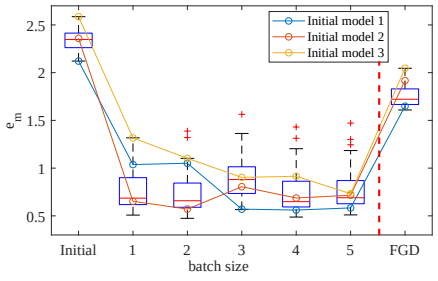
(b) N_s=300



(c) N_s=500



(d) N_s=800



(e) N_s=1000

Figure 6.17: Boxplot of the parameter misfits $e_{\mathbf{m}}$ of logarithmic permeability before and after history matching corresponding to 100 different initial models. The batch size of the $Adam$ optimizer is set to 1, 2, 3, 4 and 5, respectively.

In terms of the computational cost, the (3×4, 2×3) domain decomposition requires 158 HFM simulations, among them, 42 HFM simulations are used to collect the snapshots to construct the bases for implementing POD, 105 HFM simulations are run to construct the initial subdomain reduced-order linear model at the 1st out-loop, and additional 11 HFM simulations are required in the following 11 outer-loops. In order to achieve comparable parameter misfits $e_{\mathbf{m}}$, the cR-U-Net surrogate model should be trained with at least N_s = 300 samples. We also present an alternative to compare these two surrogate models for

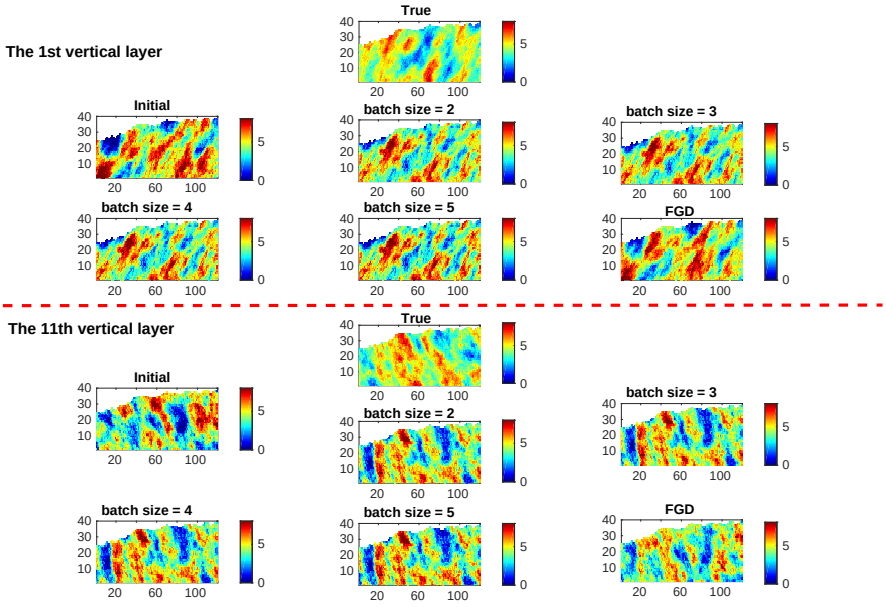
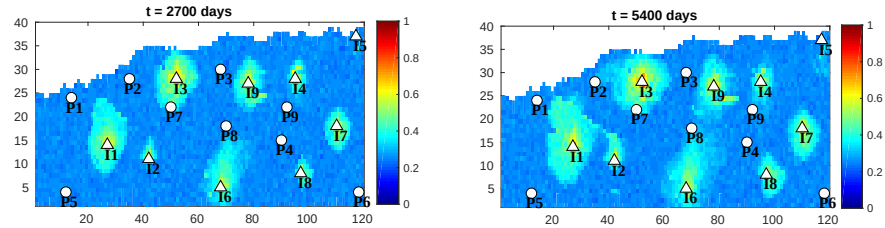
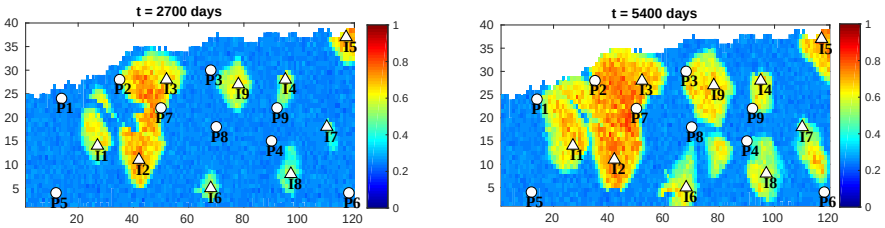


Figure 6.18: 2D horizontal cross sections of the updated logarithmic permeability for the 1st and 11th vertical layer under different batch size for SGD and FGD optimization methods. The cR-U-Net surrogate models are trained using N_s 1000 training samples.



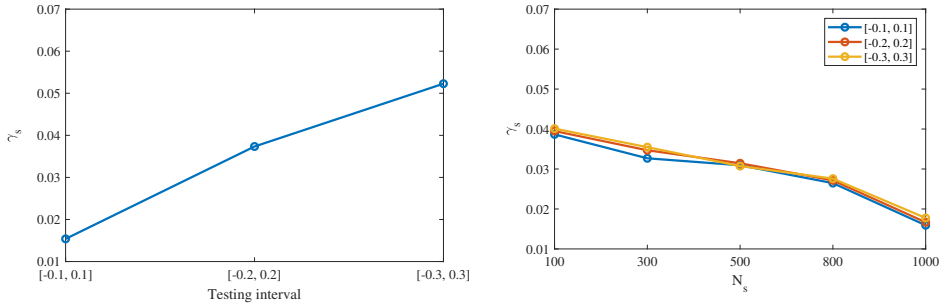
(a) The 1st layer



(b) The 20th layer

Figure 6.19: Noise distribution of water saturation at the days 2700 and days 5400 used in this comparative study. Normal distributed independent measurement noise with a standard deviation equal to 5% of the 'true' data value, was added to all observations

almost the same computational cost. We increase the sampling points for constructing the subdomain POD-TPWL surrogate models, which generally improves the accuracy of



(a) Subdomain POD-TPWL (b) cR-U-Net

Figure 6.20: Field-average relative error γ_s as a function of domain decomposition, testing interval and number of training samples. (a) Subdomain POD-TPWL; (b) cR-U-Net surrogate model.

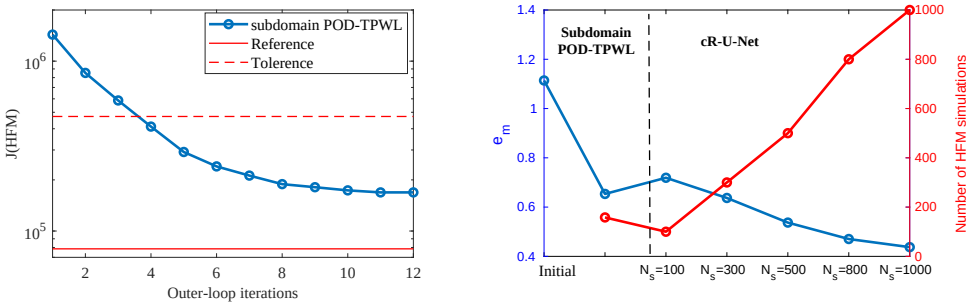
the approximated gradients. As illustrated in fig.6.22, the parameter misfits $e_{\mathbf{m}}$ gradually decreases but much slower than those for the cR-U-Net. Corresponding to the specific domain decomposition scheme, such as (3×4, 2×3) in this case-study, the accuracy does not infinitely improve as the number of training samples. These results further indicate the importance of designing appropriate domain decompositions.

Fig.6.23 shows the 2D horizontal cross sections of updated logarithmic permeability fields for a comparison of the history matching results using the cR-U-Net and subdomain POD-TPWL surrogate models. It clearly can be seen that the true logarithmic permeability fields almost can be reconstructed. In summary, subdomain POD-TPWL is slightly more efficient than that of cR-U-Net surrogate model for generating one posterior realization for this example application.. In addition, the implementation of subdomain POD-TPWL is also relatively flexible without a strict requirement for the hardware, e.g., GPU cards.

Table 6.4: The final objective function values $J(\times 10^6)$ for the subdomain POD-TPWL and cR-U-Net.

Subdomain POD-TPWL	cR-U-Net				
	N_s=100	N_s=300	N_s=500	N_s=800	N_s=1000
0.1689	0.4366	0.3890	0.2699	0.2083	0.2021
Initial	1.433				
Ref - Projected True	0.0912				
Ref - True	0.0785				

In order to generate N_e posterior realizations for quantifying the model uncertainty, the subdomain POD-TPWL has to be independently implemented for each ensemble member, which requires us to repeatedly update the reduced-order models with several additional HFM simulations at the outer-iterations. 147 HFM simulations are required to construct the initial reduced-order model at the 1st out-loop. If the number of outer-loops for the i-th ensemble member is assumed to be $N^i_{outer-loop}$, the required total number of HFM simulations for obtaining N_e posterior realizations should be $(147 + \sum_{i=1}^{N_e} \times N^i_{outer-loop})$. By contrast, the cR-U-Net surrogate model obtains N_e posterior realizations through start-



(a) (b)

Figure 6.21: Final objective functions J and parameter misfits $e_{\mathbf{m}}$ with respective to the training size and domain decomposition for the cR-U-Net and subdomain POD-TPWL, respectively.(a) The evolution of objective function with respect to the iterations for the subdomain POD-TPWL. (b) The number of HFM simulations and the parameter misfits $e_{\mathbf{m}}$ for the cR-U-Net and subdomain POD-TPWL.

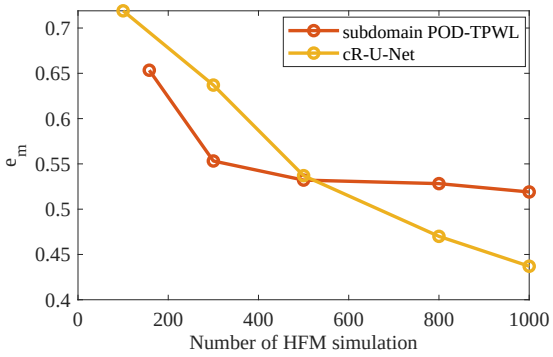


Figure 6.22: Parameter misfits $e_{\mathbf{m}}$ with respective to the number of HFM simulations for the cR-U-Net and subdomain POD-TPWL, respectively. We gradually increase the HFM simulations for constructing the subdomain POD-TPWL surrogate models.

ing from different initial models. As we have mentioned before, this process does not involves additional HFM simulations. Fig.6.24 depicts the parameter misfits $e_{\mathbf{m}}$ of $N_e = 100$ posterior models for the cR-U-Net and subdomain POD-TPWL. It is noticeable that the cR-U-Net generates smaller parameter misfits $e_{\mathbf{m}}$ than that of the subdomain POD-TPWL. In terms of computational cost, the subdomain POD-TPWL requires about, for example an average $N_{outer-loop}$=15 outer-loops, 1647 = 147+100×15 HFM simulations for generating these 100 posterior models, while the cR-U-Net only requires 100 HFM simulations for achieving even better results. The computational saving will increase linearly with the ensemble size. Overall, the cR-U-Net will be much more efficient than that of the subdomain POD-TPWL for generating an ensemble of posterior realizations.

Both methods have their own advantages and disadvantages. On the one hand, our previous results have indicated that the performance of subdomain POD-TPWL is very sensitive to the domain decomposition schemes. How to choose the optimal domain de-

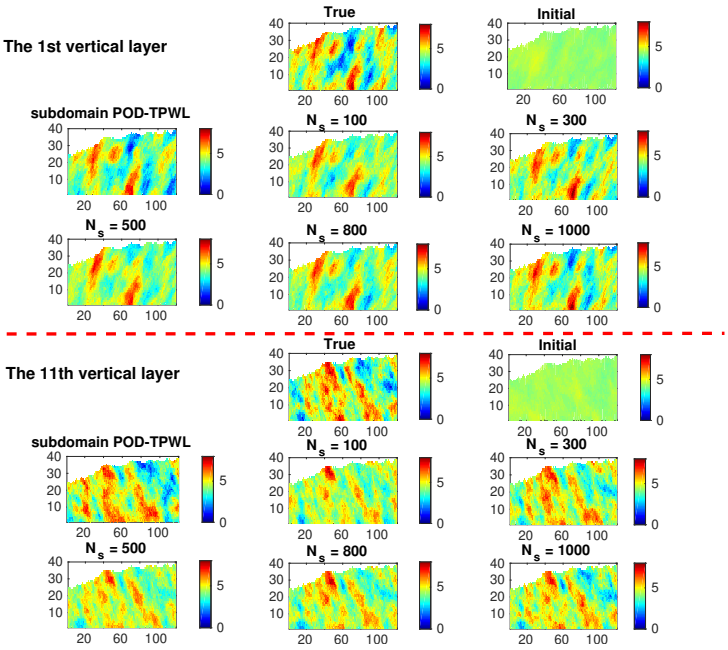


Figure 6.23: 2D horizontal cross sections of the updated logarithmic permeability fields for assessing the accuracy of the history matching results using the cR-U-Net and subdomain POD-TPWL surrogate models. The 1st and 11th vertical layer are shown here.

composition scheme is one of the most key steps for a successful implementation of the subdomain POD-TPWL. On the other hand, although the construction of the cR-U-Net is realized without being explicitly programmed with the hidden and highly complex governing physics, the choice of a DNN architecture is relatively subjective (although flexible), and its outputs lack a clear understanding. The cR-U-Net takes advantage of the high-performance computing units, such as the GPU cards, since the exiting deep learning packages can efficiently use them. The ability of a software program to fully utilize GPUs is a big advantage.

6.4.6. ADDITIONAL REMARKS

One of the most prominent merit of the proposed surrogate-assisted seismic history matching method with deep neural network is that it can perform massively parallel computations on GPUs. Nevertheless, limitations of the proposed method do exist. First, the choice of a neural network architecture is flexible, but it is still relatively subjective. Second, our proposed surrogate-assisted seismic history matching does not involve additional HFM simulations. Although the availability of GPU's should be a precondition for efficiently training the cR-U-Net surrogate model, this is not a big problem as GPUs are relatively cheap and easy to install on existing computer systems. Third, the intrinsic disadvantage of the DNN method appears to be that it may not be clear beforehand how many training samples are needed to obtain the desired accuracy. In order to obtain highly accurate DNN model, we have to blindly increase the number of training samples as much as possible in most cases. This inevitably leads to redundant model simulations

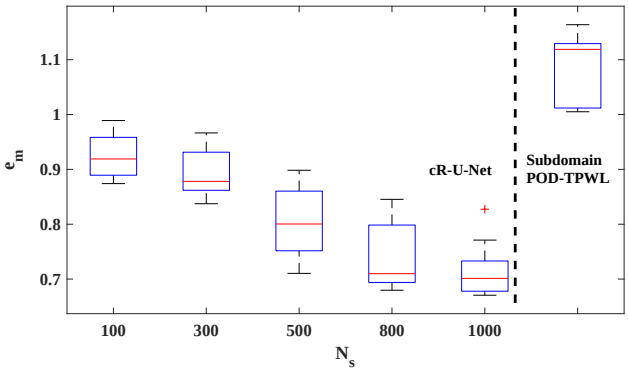


Figure 6.24: Boxplot of the parameter misfits $e_{\mathbf{m}}$ of logarithmic permeability after the history matching using cR-U-Net and subdomain POD-TPWL surrogate models. The 3D domain decomposition with adaptive strategy is implemented.

sometimes when satisfactory results can be obtained using a small dataset. Some promising results have been obtained in this paper, however, further research is needed to gain insights about how the DNN surrogate works for the gradient-based inversion, when it will work (weak- or strong non-linear model) and the trade-offs between accuracy and computational complexity. The degree of model non-linearity dominates the convergence performance of the gradient-based optimizations. Nevertheless, whether the non-linearity of the high-fidelity model is correctly represented by the DNN surrogate model has not been fully understood yet.

Our proposed DNN surrogate method has demonstrated to be very efficient in generating multiple posterior models. In the community of data assimilation, the ensemble-based assimilation methods are particularly useful to quantify the models uncertainty for the realistic cases and many successful applications have been reported in the literature [31]. A direct application of the proposed DNN surrogates to field-scale models with a large degree of freedom definitely poses huge computational challenges and memory requirements for training them. This drawback seemingly makes the DNN approach less attractive than that of the ensemble-based assimilation methods. However, directly assimilating a huge amount of data has the high possibility to cause unreliable rank deficiency and eventually ensemble collapse problem. In general, increasing ensemble size or covariance localization have always been a standard approach to resolve this issue [32, 33]. However, it will definitely complicate the implementation of ensemble-based assimilation methods in realistic applications. By contrast, the filter divergence issue hardly occurs in our proposed DNN-assisted seismic history matching methodology.

REFERENCES

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

[2] D. S. Oliver and Y. Chen, *Recent progress on reservoir history matching: a review,* Computational Geosciences **15**, 185 (2011).

[3] T. Mannseth and K. Fossum, *Assimilating spatially dense data for subsurface applications—balancing information and degrees of freedom,* Computational Geosciences **22**, 1323 (2018).

[4] H. Jeffreys, B. Jeffreys, and B. Swirles, *Methods of mathematical physics* (Cambridge university press, 1999).

[5] P. Courtier, J.-N. Thépaut, and A. Hollingsworth, *A strategy for operational implementation of 4d-var, using an incremental approach,* Quarterly Journal of the Royal Meteorological Society **120**, 1367 (1994).

[6] R. Alizadeh, J. K. Allen, and F. Mistree, *Managing computational complexity using surrogate models: a critical review,* Research in Engineering Design **31**, 275 (2020).

[7] H. Ates, A. Bahar, S. E.-A. Salem, M. Charfeddine, *et al.*, *Ranking and upscaling of geostatistical reservoir models using streamline simulation: A field case study,* SPE Reservoir Evaluation & Engineering **8**, 22 (2005).

[8] M. O. Salazar, J. R. Villa Piamo, *et al.*, *Permeability upscaling techniques for reservoir simulation,* in *Latin American & Caribbean Petroleum Engineering Conference* (Society of Petroleum Engineers, 2007).

[9] P. Vermeulen and A. Heemink, *Model-reduced variational data assimilation,* Monthly weather review **134**, 2888 (2006).

[10] M. U. Altaf, A. W. Heemink, and M. Verlaan, *Inverse shallow-water flow modeling using model reduction,* International journal for multiscale computational engineering **7** (2009).

[11] C. Dai, L. Xue, D. Zhang, and A. Guadagnini, *Data-worth analysis through probabilistic collocation-based ensemble kalman filter,* Journal of Hydrology **540**, 488 (2016).

[12] M. A. Ahmadi, *Developing a robust surrogate model of chemical flooding based on the artificial neural network for enhanced oil recovery implications,* Mathematical Problems in Engineering **2015** (2015).

[13] Y. J. Heo, S. J. Kim, D. Kim, K. Lee, and W. K. Chung, *Super-high-purity seed sorter using low-latency image-recognition based on deep learning,* IEEE Robotics and Automation Letters **3**, 3035 (2018).

[14] T. Young, D. Hazarika, S. Poria, and E. Cambria, *Recent trends in deep learning based natural language processing,* ieee Computational intelligenCe magazine **13**, 55 (2018).

[15] S. Mo, N. Zabaras, X. Shi, and J. Wu, *Deep autoregressive neural networks for high-dimensional inverse problems in groundwater contaminant source identification,* Water Resources Research **55**, 3856 (2019).

[16] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, *et al., Tensorflow: Large-scale machine learning on heterogeneous distributed systems,* arXiv preprint arXiv:1603.04467 (2016).

[17] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al., Pytorch: An imperative style, high-performance deep learning library,* in *Advances in neural information processing systems* (2019) pp. 8026–8037.

[18] R. K. Tripathy and I. Bilionis, *Deep uq: Learning deep neural network surrogate models for high dimensional uncertainty quantification,* Journal of computational physics **375**, 565 (2018).

[19] Z. Zhong, A. Y. Sun, and H. Jeong, *Predicting co2 plume migration in heterogeneous formations using conditional deep convolutional generative adversarial network,* Water Resources Research **55**, 5830 (2019).

[20] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, *Densely connected convolutional networks,* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017) pp. 4700–4708.

[21] O. Ronneberger, P. Fischer, and T. Brox, *U-net: Convolutional networks for biomedical image segmentation,* in *International Conference on Medical image computing and computer-assisted intervention* (Springer, 2015) pp. 234–241.

[22] F. Gassmann, *Elastic waves through a packing of spheres,* Geophysics **16**, 673 (1951).

[23] Z. L. Jin, Y. Liu, and L. J. Durlofsky, *Deep-learning-based reduced-order modeling for subsurface flow simulation,* arXiv preprint arXiv:1906.03729 (2019).

[24] M. Tang, Y. Liu, and L. J. Durlofsky, *A deep-learning-based surrogate model for data assimilation in dynamic subsurface flow problems,* Journal of Computational Physics , 109456 (2020).

[25] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, *Convolutional lstm network: A machine learning approach for precipitation nowcasting,* Advances in neural information processing systems **28**, 802 (2015).

[26] Z. Zhou and D. M. Tartakovsky, *Markov chain monte carlo with neural network surrogates: Application to contaminant source identification,* arXiv preprint arXiv:2003.02322 (2020).

[27] K. He, X. Zhang, S. Ren, and J. Sun, *Identity mappings in deep residual networks,* in *European conference on computer vision* (Springer, 2016) pp. 630–645.

[28] P. Sibi, S. A. Jones, and P. Siddarth, *Analysis of different activation functions using back propagation neural networks,* Journal of theoretical and applied information technology **47**, 1264 (2013).

[29] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*, Vol. 1 (MIT press Cambridge, 2016).

[30] A. F. Rasmussen, T. H. Sandve, K. Bao, A. Lauser, J. Hove, B. Skaflestad, R. Klöfkorn, M. Blatt, A. B. Rustad, O. Sævareid, *et al.*, *The open porous media flow reservoir simulator,* Computers & Mathematics with Applications (2020).

[31] A. A. Emerick and A. C. Reynolds, *History matching time-lapse seismic data using the ensemble kalman filter with multiple data assimilations,* Computational Geosciences **16**, 639 (2012).

[32] A. Emerick and A. Reynolds, *Combining sensitivities and prior information for covariance localization in the ensemble kalman filter for petroleum reservoir applications,* Computational Geosciences **15**, 251 (2011).

[33] M.-J. Yeo, S.-P. Jung, and J. Choe, *Covariance matrix localization using drainage area in an ensemble kalman filter,* Energy Sources, Part A: Recovery, Utilization, and Environmental Effects **36**, 2154 (2014).

7

HISTORY MATCHING USING DEEP-LEARNING STOCHASTIC INVERSION

Imaging-type monitoring techniques are used in monitoring dynamic processes in many domains, including medicine, engineering, and geophysics. This chapter presents an efficient workflow for application of such data for the conditioning of simulation models. In order to reduce the high computational cost and complexity of data assimilation workflows for high-dimensional parameter estimation, a residual-in-residual dense block extension of the U-Net convolutional network architecture is proposed, to predict time-evolving features in high-dimensional grids. The network is trained using high-fidelity model simulations. We present two examples of application of the trained network as a surrogate within an iterative ensemble-based workflow to estimate the static parameters of geological reservoirs based on binary-type image data, which represent fluid facies as obtained from time-lapse seismic surveys. The differences between binary images are parameterized in terms of distances between the fluid-facies boundaries, or fronts. We discuss the impact of the choice of network architecture, loss function, and number of training samples on the accuracy of results and on overall computational cost.

Parts of this chapter have been included in a manuscript submitted to Journal of Petroleum Science and Engineering. **Xiao, C.**, Leeuwenburgh, O., Lin, H.X. and Heemink, A., 2020. *Conditioning of Deep-Learning Surrogate Models to Image Data with Application to Reservoir Characterization.* Knowledge-Based Systems.

7.1. INTRODUCTION

Imaging-type monitoring techniques are relevant for monitoring of dynamic processes in many application domains, and include for example X-Ray, computed tomography (CT) and magnetic resonance imaging (MRI) techniques for medical imaging [1], satellite remote sensing for earth observation [2, 3], and seismic and electromagnetic imaging of the subsurface [4]. Applications in the earth observation domain include the prediction of spreading of air pollution [5, 6] and e.g. typhoon tracks [7]. Geophysical applications include the monitoring of CO_2 storage in aquifers [8] and the displacement of fluids in hydrocarbon reservoirs [9]. Imaging techniques deliver pixel-wise information in 2D or 3D and may be used to identify static features or anomalies or changes over time. We are especially interested in the application of such data for the conditioning of simulation models. Such applications are very common in the geosciences, where large-scale simulation models and measured data are used to monitor the state of e.g. energy and water systems, predict their future behaviour and optimize actions to achieve desired behaviour of the system.

A challenge with the use of imaging techniques for these purposes is that they tend to deliver very large number of data points that may have complex relationships to underlying (poorly-known) model parameters. Therefore compromises are often needed towards the data assimilation methods that are used to integrate the data into models, or towards the description of the data and the associated measurement noise. We intend to use state-of-the-art data assimilation (in fact, parameter estimation) methods that are able to deliver a full uncertainty characterization, especially Iterative Ensemble Smoothers (IES) [10]. Such methods characterize uncertainty in the model by a large ensemble of model realizations, where each model realization is defined by a different set of randomly sampled values for uncertain model parameters.

Two challenges are commonly recognized in the application of such data assimilation methods to imaging data: (1) the use of large data sets can lead to artificial collapse of the ensemble; (2) given the computational expense of simulating each model, the large number of uncertain grid-based parameters, and the complexity in data-parameter relationships, many iterations of the IES with a large ensemble may be required, resulting in huge over-all computational costs.

Several approaches have been attempted to deliver reduced representations of large data sets, including coarse representations [11], wavelet decomposition [12], and nonlinear reduction methods based on machine learning techniques [13]. In many cases, including the earth observation and geophysical examples mentioned earlier, the purpose of the monitoring is the identification of changes over time, which can often be characterized by the displacement of a contour value in the image. Leeuwenburgh and Arts and Zhang and Leeuwenburgh [14, 15] proposed a parameterization of monitoring data for such situations in terms of distances between corresponding contours (or iso-surfaces in 3D) in the simulated and measured images respectively, and showed that the resulting reduction in the number of data can help avoid ensemble collapse (e.g., see Trani et al [16], [17] and [18] for related approaches). In its most basic form, the contours separate the domain into regions belonging to one of two possible classes, effectively resulting in a binary image.

Here we address the second challenge, namely the high computational demand imposed by iterative parameter estimation workflows involving imaging data. In introducing the methodology we will focus on an application of time-lapse seismic monitoring for reservoir model parameter estimation, also referred to as seismic history matching in the

field of reservoir engineering [19]. Seismic data are obtained as waves that are registered in grid-based distributions of sensor locations, after being first emitted into the ground at source locations on the surface and subsequently reflected at so-called impedance contrasts in the subsurface, typically reflecting spatial changes in rock properties or fluid content. When a seismic survey is repeated at a later time, the differences between the imaging data sets can often be interpreted as changes in the distribution of different types of fluids. Examples include the displacement of water by CO_2 [8] and the displacement of oil by water or gas [9]. Given that direct access to the reservoirs, which are often found at depths of a few kilometers, is possible only at locations where wells have been drilled into the reservoir, this time-lapse seismic data can be the main source of information about changes in the system.

In order to reduce the computation cost of seismic history matching reservoir models, surrogate (proxy) model methods such as upscaled models [20] and reduced-order models[21] have been pursued. Disadvantages of these approaches are the loss of information at high spatial resolutions and non-linearity respectively. Another approach is the use of machine learning surrogates, where especially Artificial Neural Networks (ANN) have recently started receiving renewed interest. This growing interest is related to the appearance of modern architectures that support deep networks with enhanced capability of relating large numbers of inputs and outputs. Several recent studies have explored the use of Deep Neural Network (DNN) surrogates for prediction of single-phase [22, 23], two-phase [24, 25], and multi-phase [26] subsurface flow dynamics. A detailed literature review about the application of deep neural network can be found in Chapter 6. As far as we are aware, however, such approaches have not been successful yet in the context of grid-based parameter estimation based on large-volume imaging data. We will therefore propose a new surrogate modelling methodology based on machine learning or more specifically deep learning approaches that aims to deliver high-quality parameter characterization at a significantly decreased computational cost. This is motivated by rapid recent advancements in the application of deep neural networks to simulation of dynamic systems, and image processing [27], and wide availability of high-performance processing units (GPU's) and deep-learning frameworks (e.g., Tensorflow [28] and PyTorch [29]).

Motivated by the successful applications of hybridization workflow using neural network methods in a variety of research domains, the aim of this work is to propose a hybrid deep learning-based workflow to characterize reservoir heterogeneities. Specifically, a cheap-to-run surrogate model is constructed to efficiently predict imaging-type seismic data. Representation of fluid-fronts position using binarized images is simply equivalent to an image segmentation problem in the field of computer vision [30–32]. Deep learning as a promising approach has shown potentials to effectively address this kind of problem. This paper explores the potential of deep neural network to predict the spatially discontinuous fluid fronts based on the concept of image segmentation. It has been extensively acknowledged that a deeper network may approximate the predictions with higher complexity, but at the cost of difficulty in training [33, 34]. Inspired by the successful applications of image super-resolution problems [35, 36], the state-of-the-art residual-in-residual dense block (RRDB) acts as an effective means to train a deeper neural network. On the basis of standard U-Net [37] which has shown prominent advantage and applicability for object segmentation task, we adopt an improved U-Net architecture via multiple RRDB structures.

The remainder of this chapter is structured as follows: Section 2 describes a hybrid

deep-learning workflow for reservoir heterogeneity characterization. In Section 4, a systematic accuracy assessment of the proposed deep-learning segmentation model for predicting binarized fluid-fronts is presented. This section also discusses and evaluates an application of proposed hybrid workflow to characterize a synthetic 2D non-Gaussian facies model and a benchmark 3D reservoir model [38]. Finally, Section 5 highlights our contribution and points out some potential works.

7.2. IMAGE-ORIENTED DISTANCE PARAMETERIZATION

We will consider the situation in which the relevant information contained by the data can be captured by contour (or iso-surfaces), which segment the image into binary categories (for similar applications in e.g. computer vision, see e.g. [30–32]). The underlying assumption is that the information contained in the shape or outline of a feature (if it can be extracted coherently) is generally more reliable and informative than the information contained in the amplitude of individual grid cell values. We adopt a distance-based parameterization of the contour information that was designed for application to seismic history matching of subsurface reservoirs [14, 15] where the contours represent the position of the separation front between regions with high saturation of displacing and displaced fluids respectively. The dimension of the resulting data space (e.g. the number of points on fronts) is typically much lower than the original data space which is equal (or proportional) to the number of grid cells. In the following the essential elements of the parameterization are described in more detail.

In Fig.7.1, scalar values 0 and 1 are used to define binarized images $\mathbf{I}_A$ and $\mathbf{I}_B$ obtained by pre-processing underlying images A and B (not shown). Contours defining the boundaries between the two categories contained in the images are shown as well. The similarity between the two images is characterized by a map computing from the local Hausdorff distance (LHD) [39]:

$$LHD(A,B) = \mathbf{I}_A \circ \mathbf{D}_B + \mathbf{I}_B \circ \mathbf{D}_A. \tag{7.1}$$

where $\mathbf{D}_A$ and $\mathbf{D}_B$ represent distance maps for shapes A and B respectively, which are computed here using a fast marching method [15]. In essence, LHD quantifies the similarity of two images with two directed distance maps in complementary directions, i.e. $\mathbf{I}_A \circ \mathbf{D}_B$ (distance from B to A) and $\mathbf{I}_B \circ \mathbf{D}_A$ (distance from A to B). Based on Eq.7.1, three parameterization approaches developed in [15] are as follows

$$\begin{aligned} PLHDC(A_c,B_c) &= \mathbf{I}_{A_c} \circ \mathbf{D}_{B_c}, \\ LHDC(A_c,B_c) &= \mathbf{I}_{A_c} \circ \mathbf{D}_{B_c} + \mathbf{I}_{B_c} \circ \mathbf{D}_{A_c}, \\ LHDA(A_a,B_a) &= \mathbf{I}_{A_a} \circ \mathbf{D}_{B_a} + \mathbf{I}_{B_a} \circ \mathbf{D}_{A_a}. \end{aligned} \tag{7.2}$$

where the subscripts c and a denote that the shapes in the images are represented by contours (boundaries or fronts) or areas (for example, a flooded area) respectively. $PLHDC$ is the partial $LHDC$ measuring the distance only from the simulated fronts to the "observed" fronts. Because nonzero distance values exist only on the "observed" fronts, the number of data is reduced from the number of grid points on the whole image to the number of grid points on the "observed" fronts. Furthermore, the binary character of the image representation is transformed into continuous data (distances) that can be handled by the data assimilation methodology. These three distance parameterization approaches each have their own advantages and disadvantages. $PLHDC$ leads to the strongest reductions in the

number of data, but not does capture the information in both images as well as LHDC [15]. *LHDA* provides the most complete description of similarity and differences but does not reduce the number of data. Therefore, in the remainder of this paper *LHDC* will be used. In the ES-MDA procedure, the measurement innovation can now be replaced by the image dissimilarity, e.g. *LHDC*.

In many cases it may not be feasible to perform reservoir history matching exactly because it requires numerous computationally costly simulations of the numerical model. This limitation has stimulated research into efficient approximate methods, for example for parameter estimation (also referred to as history matching) in subsurface reservoir engineering applications. Surrogate modeling is currently identified as one of the most promising means to improve the efficiency of parameter estimation procedures. In the following We will consider the possibility of reducing the simulation cost by replacing simulations of the high-fidelity model by simulation of a surrogate. Specifically we will describe an efficient deep-learning segmentation model for predicting binarized images as accurately as possible.

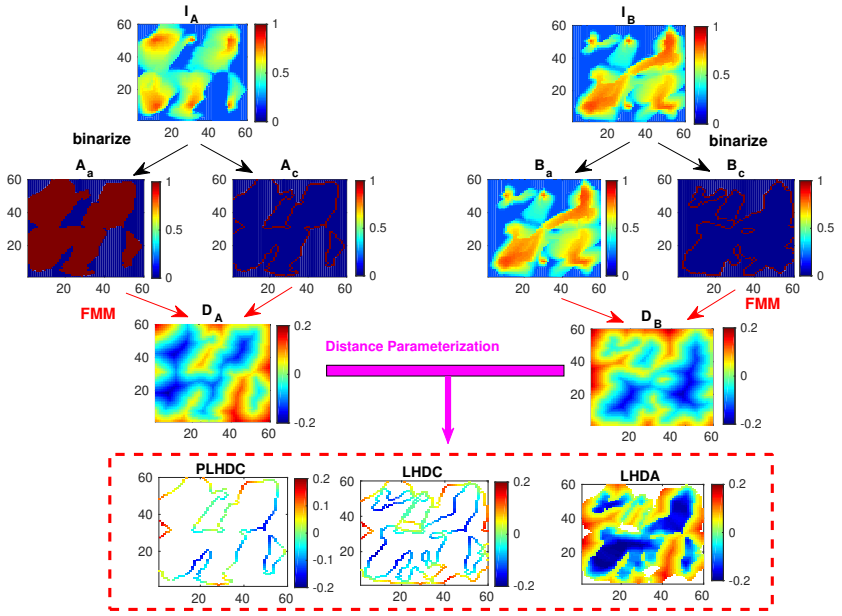


Figure 7.1: Illustrative diagram of *PLHDC*, *LHDC* and *LHDA* schemes for the image $\mathbf{I}_A$ and $\mathbf{I}_B$, respectively. The first and second row shows the maps of contour, area and distances corresponding these two images. The last row shows the dissimilarity maps of these two images using *PLHDC*, *LHDC* and *LHDA* scheme.

7.3. CONDITIONAL RESIDUAL-IN-RESIDUAL DENSE BLOCK U-NET

This section introduces the procedures of using a deep neural network to perform predictions of spatially discontinuous shape features. The section discussing the preparation of the training data-set introduces a pre-processing step to convert continuous maps into binarized images. In the network training section, we define the training losses for both

image regression and image segmentation. A two-stage training strategy consisting of alternating minimization of the regression loss and the segmentation loss suggested in the literature is applied to better approximate the discontinuous shapes in the images.

7.3.1. NEURAL NETWORK ARCHITECTURE

In this work we consider the cross-domain image segmentation problem of predicting spatially discontinuous binarized images representing changes in dynamic systems. Some recent studies has investigated the potential of using DNN surrogates to replace high-fidelity model simulations. For example, Jin et al proposed a DNN surrogate model with autoregressive structure for approximating time-varying reservoir dynamics [40]. Tang et al [41] developed a deep convolutional recurrent neural network architecture, specifically a combination of auto-encoder and a convolutional long-short-term memory recurrent network (convLSTM) [42].

Our proposed hybrid workflow shares some similarities with the one proposed by Tang et al. (2019) where also a deep-learning based history matching problem was pursued. Tang et al [41] developed a surrogate for temporal prediction of spatially continuous pressure and saturation snapshots for channelized oil reservoir models. The spatial pressure and saturation predictions were the basis for predictions of well data such as fluid rate and bottom-hole pressure, which were used in a history matching workflow aimed at characterizing the channelized reservoir system. In this paper we will demonstrate how a similar workflow could be used for parameter estimation based on (binary) imaging-type data.

The main task of the neural network is to perform an cross-domain image segmentation from the Gaussian/non-Gaussian geological parameter field to the output binary contour maps. Motivated by the changes in the data parameterization in terms of distances to the front instead of grid block saturations we develop an advanced deep convolutional neural network architecture based on the integration of a stack of residual-in-residual dense blocks (RRDB) and U-Net architecture by adapting the previous cR-U-Net proposed in Chapter 6. The overall neural network architecture is illustrated in Fig.7.2, which is almost consistent with our previously proposed cR-U-Net in Chapter 6. The detailed descriptions of each units, including encoding unit, transition unit and decoding unit, can also be found in Chapter 6.

In the cR-U-Net architecture, the extracted features in the encoding path are copied and concatenated onto the upsampled features in the decoding path, this enables the multi-scale features extracted in the encoding path to be propagated to the corresponding decoding path. The latent feature maps produced from encoding unit are propagated using a stack of residual-blocks. Our numerical results have shown that this residual blocks is capable of capture the flow dynamic of the Gaussian model. In this chapter, we proposed a time-conditioning Residual-in-Residual Dense U-Net (cRRDB-U-Net). Specifically, a stack of residual-in-residual dense block embedded into the transition unit is used to connect the encoding unit and the decoding unit. We should emphasize that this advanced RRDB proposed in [36] originally used for image super-resolution recovery is adopted to further enhance the performance of standard U-Net for the context that the oil/water transport model with non-Gaussian parameter inputs and spatially discontinuous contour outputs needs to be accurately approximated.

RESIDUAL-IN-RESIDUAL DENSE BLOCK

The latent feature maps $\mathbf{F}_4(\mathbf{m})$ produced from the last block of encoding unit are fed

to a transition unit, see Fig.7.3(c). To learn the temporal features of dynamic model, the time, as a conditional feature channel, is concatenated to the low-dimensional representation features, i.e., $\mathbf{F}_4(\mathbf{m})$, after the encoding part. This transition unit is composed of two adjacent RRDB and conventional convolutional blocks. The aim of transition unit is to produce feature maps $\mathbf{F}_5(\mathbf{m})$, which are the most complicated and informative feature maps containing both the spatial and temporal information. These maps will be then provided to the decoding unit. Specifically, the RRDB structure contains a well-designed combination of dense blocks and residual blocks. A dense block intentionally constructs connections between non-adjacent layers, designed to take full advantage of the multi-scale features in the output of the previous layers [43]. For example, a dense block with 5 layers is shown in Fig.7.3(a). The structure of the residual block, specifically residual convolutional ($resConv$) block, bypass the nonlinear layers through introducing an identity mapping. This special architecture of $resConv$ block can help cope with the gradient vanishing/explosion problem especially for the highly deeper network [44].

As we all know, a deeper network may approximate the map with higher complexity, but at the cost of difficulty in training [33, 34]. Inspired by the successful applications of image super-resolution problems [35, 36], the state of the art residual-in-residual dense block (RRDB) is used to ease the training process of a deeper neural network. We display the RRDB architecture in Fig.7.3(b). It contains a stack of special structures where the dense blocks are embedded between two adjacent residual blocks. More details about RRDB can be referred to [36]. The number of input feature maps of a dense block is N_f (Fig.7.3(a)) and is set to $N_f = 24$.

TRANSITION UNIT BASED ON RESIDUAL-IN-RESIDUAL DENSE BLOCK

The network structures of the encoding and decoding units have been described in previously proposed cR-U-Net. We employ the residual-in-residual dense block structure in the transition unit of our cRRDB-U-Net network for surrogate modeling of oil/water transport in media with non-Gaussian parameters. The transition unit is shown in Fig.7.3(c). For 2D or 3D images (fields), the 2D and 3D operations (i.e., BN and $Conv$) are used in the network without requiring any additional modifications to the network architecture. The network contains four residual-in-residual dense blocks and the feature maps are to go through a coarsen-to-refine process. A convolutional layer is first employed to extract feature maps from the raw input image. The obtained features are then passed through the residual-in-residual dense blocks and the transition convolutional layers for downsampling/upsampling of the feature maps. We arrange an adjacent position of two residual-in-residual dense blocks in the network (Fig.7.3(c)) to encourage the information flow through the coarse feature maps. That is, two adjacent blocks are placed in the most central part. An additional level of residual learning is implemented on the stacked residual-in-residual dense blocks, resulting in a three-level residual learning structure in the network.

To verify the effectiveness of RRDB structure, a comparative study of the standard U-Net with and without this block is conducted. Specifically, we integrate a stack of residual convolutional $resConv$ blocks [40] with the standard U-Net architecture, which can be referred to as cR-U-Net and also has been used in Chapter 6. In addition to the transition unit, the overall neural network architecture of the cR-U-Net is very similar to the cRRDB-U-Net. In the cR-U-Net architecture, the feature maps produced from the encoding unit are concatenated with the time value and then are fed to five residual-blocks. The archi-

tectures of cRRDB-U-Net are summarized in Table 7.1.

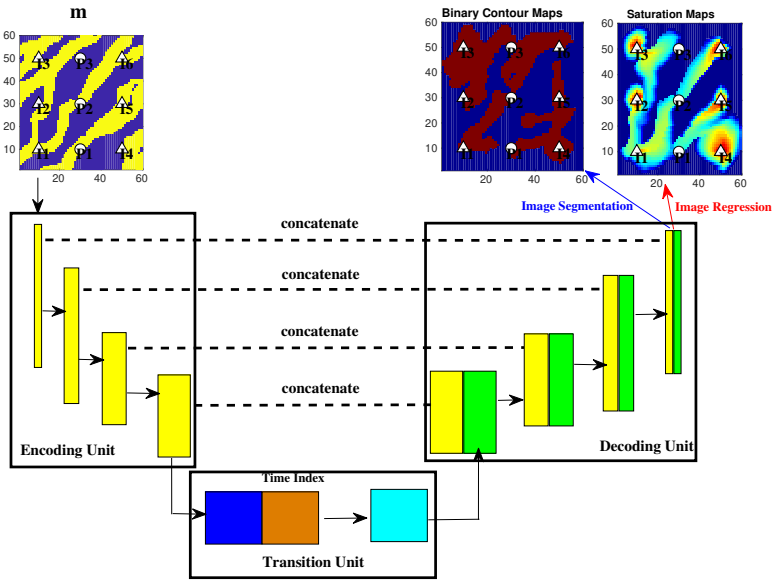


Figure 7.2: Schematic illustration of cRRDB-U-Net architecture. cRRDB-U-Net is composed of encoding unit, transition unit and decoding unit, where the local features extracted in the encoding unit are concatenated with the upsampling features in the decoding unit to produce the target images. The transition unit concatenates and forwards the extracted local features and time feature, and finally feeds them to the decoding unit.

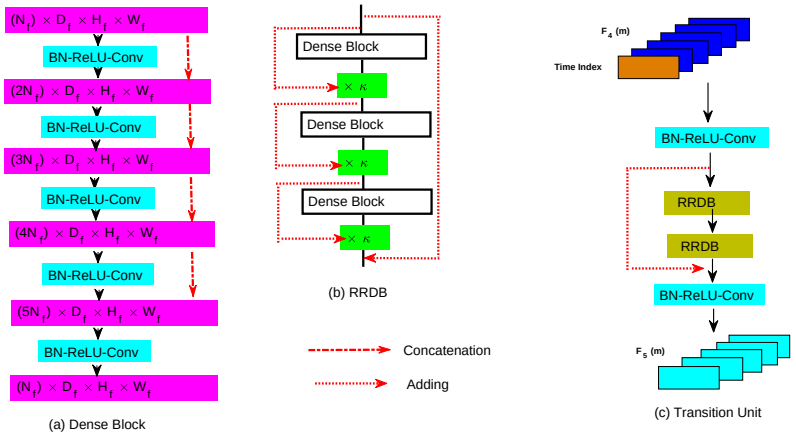


Figure 7.3: Schematic illustration of transition unit. We arrange an adjacent position of two residual-in-residual dense blocks in the network to encourage the information flow through the coarse feature maps. In addition, the time feature as an additional channel is feed to this unit as well to mimic the flow dynamic.

7.3.2. DATASET PREPROCESSING AND PREPARATION

In order to train the cRRDB-U-Net surrogate model, we generate a set of training sam-

Table 7.1: The network design of cRRDB-U-Net Architecture. This table shows the network structure for the 2D model, which can be easily extended to 3D model by using 3D convolutional operations. Here N_x and N_y denote the width and height of original images.

Unit	filter size and stride	Output Size
Encoding unit	Input	(N_x, N_y, 1)
	$BN-ReLU-Conv2D$, 16 filters of size (3,3), stride 2	($N_x/2$, $N_y/2$, 16)
	$BN-ReLU-Conv2D$, 32 filters of size (3,3), stride 1	($N_x/2$, $N_y/2$, 32)
	$BN-ReLU-Conv2D$, 64 filters of size (3,3), stride 2	($N_x/4$, $N_y/4$, 64)
	$BN-ReLU-Conv2D$, 128 filters of size (3,3), stride 1	($N_x/4$, $N_y/4$, 128)
Transition unit	Input	($N_x/4$, $N_y/4$, 129)
	$BN-ReLU-Conv2D$, 129 filters	($N_x/4$, $N_y/4$, 128)
	$RRDB$, 128 filters	($N_x/4$, $N_y/4$, 128)
	$RRDB$, 128 filters	($N_x/4$, $N_y/4$, 128)
	$BN-ReLU-Conv2D$, 128 filters	($N_x/4$, $N_y/4$, 128)
Decoding unit	Input	($N_x/4$, $N_y/4$, 128)
	$BN-ReLU-TConv2D$, 128 filters of of size (3,3), stride 1	($N_x/4$, $N_y/4$, 128)
	$BN-ReLU-TConv2D$, 64 filters of of size (3,3), stride 2	($N_x/2$, $N_y/2$, 64)
	$BN-ReLU-TConv2D$, 32 filters of size (3,3), stride 1	($N_x/2$, $N_y/2$, 32)
	$BN-ReLU-TConv2D$, 16 filters of size (3,3), stride 2	(N_x, N_y, 16)
	$Conv2D$, 1 filter of size (3,3), stride 1	(N_x, N_y, 1)

ples consisting of parameter grid as inputs and time series of binary fluid facies grid as outputs. The deep-learning based surrogate model represents the time-varying process

$$\hat{\mathbf{y}}^{i,n} = \hat{\mathbf{h}}^n(\mathbf{m}^i, t^n, \boldsymbol{\theta}), \quad n = 1,\ldots,N_t; \quad i = 1,\ldots,N_s. \tag{7.3}$$

where $\hat{\mathbf{y}}^{i,n} \in R^{N_x \times N_y}$ is the network prediction (image) at time $\mathbf{t}^n$ given the input $\mathbf{m}^i \in R^{N_x \times N_y}$, $\boldsymbol{\theta}$ denotes a vector containing all trainable parameters of the cRRDB-U-Net surrogate model, and i denotes the index of the training sample. N_s represents the total number of training samples.

Training data $\mathbf{y}^{i,n}$ is generated by simulating a high-fidelity forward simulation model (HFM) and selecting snaptshots of its output at times $\mathbf{t}^n$. We will assume that the simulations produce maps of continuous state variables, which can be used for image regression tasks. We employ a post-processing step to convert the continuous maps $\mathbf{y}^{i,n}$ to binary maps $\mathbf{y}^{i,n}$ to address the image segmentation problem addressed in the paper. The binary output is obtained by applying a pre-defined threshold value S_{con} and the grid blocks (pixels) are assigned a value 0 or 1. We define a pixel-wise indicators $\mathbf{F}$ as follows.

$$\begin{aligned} \mathbf{F}^{i,n} = 0, \mathbf{y}^{i,n} >= S_{con} \quad or \quad \mathbf{F}^{i,n} = 1, \mathbf{y}^{i,n} < S_{con} \\ \hat{\mathbf{F}}^{i,n} = 0, \hat{\mathbf{y}}^{i,n} >= S_{con} \quad or \quad \hat{\mathbf{F}}^{i,n} = 1, \hat{\mathbf{y}}^{i,n} < S_{con}. \end{aligned} \tag{7.4}$$

Fig.7.1 shows an example of continuous simulated HFM output (saturation maps) and the corresponding binarized images (binary contour maps) by applying Eq.7.4.

The output $\mathbf{y}^{i,n}$ depends solely on the input image $\mathbf{m}^i$ and time index t^n. We rearrange the data structure predicted from one high-fidelity model simulation, i.e., saturation map ($\mathbf{m}^i$; $\mathbf{y}^{i,1},\cdots,\mathbf{y}^{i,N_t}$), as N_t consecutive training samples

$$\{(\mathbf{m}^i; \mathbf{y}^{i,1},\cdots,\mathbf{y}^{i,N_t})\}_{i=1}^{N_s} \Rightarrow \{(\mathbf{m}^i, t^n; \mathbf{y}^{i,n})\}_{i=1,n=1}^{N_s,N_t}. \tag{7.5}$$

and the corresponding binarized maps

$$\{(\mathbf{m}^i; \mathbf{F}^{i,1},\cdots,\mathbf{F}^{i,N_t})\}_{i=1}^{N_s} \Rightarrow \{(\mathbf{m}^i, t^n; \mathbf{F}^{i,n})\}_{i=1,n=1}^{N_s,N_t}. \tag{7.6}$$

in this way, the temporal relationship between the inputs and time-varying outputs is clearly captured in the time-conditional network structure. The total number of training samples fed to this cRRDB-U-Net becomes $N_s \times N_t$.

7.3.3. TRAINING PROCEDURES

The choice of loss function for training neural networks is generally case-dependent. The choice of an appropriate loss function for the task at hand may strongly improve the performance of the network. The binary cross entropy (BCE) loss function is generally used for image segmentation tasks, while the mean square error (MSE) loss function is more commonly used for image regression tasks. These two loss functions can be defined as follows

$$L_{MSE}(\boldsymbol{\theta}) = \frac{1}{N_s N_t} \sum_{i=1}^{N_s} \sum_{n=1}^{N_t} \|\hat{\mathbf{y}}^{i,n} - \mathbf{y}^{i,n}\|_2^2. \tag{7.7}$$

and

$$L_{BCE}(\boldsymbol{\theta}) = \frac{1}{N_s N_t N_{\mathbf{m}}} \sum_{i=1}^{N_s} \sum_{n=1}^{N_t} \sum_{j=1}^{N_{\mathbf{m}}} \mathbf{F}^{i,n,j} \log \hat{\mathbf{F}}^{i,n,j} + (1 - \mathbf{F}^{i,n,j}) \log(1 - \hat{\mathbf{F}}^{i,n,j}). \tag{7.8}$$

Algorithm 4 summarizes a conventional one-stage (OS) strategy for training the proposed cRRDB-U-Net model for image segmentation based on the BCE lose function. A variant of stochastic gradient descent optimizers, e.g., *Adam*, is used to train cRRDB-U-Net surrogate model. *Adam* computes adaptive learning rates for different parameters using estimates of the first and second order moments of the gradients. The learning rate controlling the magnitude of updates of model parameters at each iteration is 5×10^{-3}. In addition, a learning rate scheduler which drops ten times on plateau training is applied to guarantee a good convergence performance. This network is built and trained using the deep learning package PyTorch [29].

Algorithm 4: Procedure of optimizing neural network parameters $\boldsymbol{\theta}$ of cRRDB-U-Net using conventional one-stage (OS) training strategy.

```
Set an initial network trainable parameters θ^0 ;
while epoch < n_epoch do
    while minibatch < N_s × N_t do
        Calculate the gradient ▽ L_BCE (θ) using auto-differentiation (AD) tool;
        Update the parameters θ using Adam(θ) → θ ;
        Evaluate the loss function function L_BCE (i.e., Eq.7.8);
        Check convergence;
    end
end
Return the optimal parameters θ
```

Although DNNs have shown promising and impressive performance in approximating the models with high-dimensionality and non-linearity, it is still very challenging to accurately predict the position of spatially discontinuous outputs, such as the shapes captured in binary images, as considered in this paper. It has been indicated in the literature that, in such cases, a two-stage (TS) training strategy through alternatively minimizing regression

and segmentation loss functions is likely to improve the performance [35]. Taking into account that the aim of our proposed cRRDB-U-Net surrogate model is to accurately predict binary image data, we construct a combined loss function where a small weight is used to regularize the MSE loss rather than the BCE loss as suggested in [35] where an accurate prediction of spatially continuous grid-based fluid saturations was the final target. In our network training process, we adopt a similar TS strategy. A hybridization of the MSE and BCE loss functions with a predefined weighting coefficient ω, i.e., can be defined as follows

$$J(\boldsymbol{\theta}) = L_{BCE}(\boldsymbol{\theta}) + \omega L_{MSE}(\boldsymbol{\theta}) \tag{7.9}$$

The procedure of iteratively updating the neural network parameters using the TS training strategy is summarized in Algorithm 5. In each iteration, a subset of the training samples is randomly chosen from the full dataset, and then the tunable network parameters $\boldsymbol{\theta}$ of cRRDB-U-Net model are adaptively adjusted twice in a consecutive manner. In the first stage, the gradient of regression loss $L_{MSE}(\boldsymbol{\theta})$ is used to compute a preliminary update of the parameters. Then, in the second stage, the gradient of combined loss $J(\boldsymbol{\theta})$ (i.e., Eq.7.9) is used to further adapt the network parameters. These two training stages will be consecutively implemented for each data minibatch to update the parameters until the training process reaches convergence. In numerical experiments presented later, a comparative study of the trained cRRDB-U-Net using the TS (i.e., Algorithm 5) and OS (i.e., Algorithm 4) training strategies will be conducted.

Advantages of TS training strategy can be expected from two contributions: (1) Through ingesting the spatial continuities of state variables in the first training stage, the intrinsic physical principles are partially considered, which can help facilitate the network training by incorporating physical constrains. (2) Incorporating continuous state variable data acts as an effective means to augment the training sample size, which in turn enables partial mitigation of the overfitting problem and helps generalizing the CRRDB-U-Net surrogate model to generic models.

Algorithm 5: Procedure of iteratively optimizing the neural network parameters $\boldsymbol{\theta}$ using the two-stage (TS) training strategy. The MSE loss weighting coefficient ω is set to 0.01.

Set an initial trainable network parameters $\boldsymbol{\theta}^0$;
while $epoch < n_{epoch}$ **do**
  while $minibatch < N_s \times N_t$ **do**
    Stage (1) : Calculate the gradient of $\bigtriangledown L_{MSE}(\boldsymbol{\theta})$ using AD tool;
    Update the parameters $\boldsymbol{\theta}$ using **Adam**$(\boldsymbol{\theta}) \rightarrow \boldsymbol{\theta}$;
    Stage (2) : Calculate the gradient $\bigtriangledown J(\boldsymbol{\theta})$ $(\boldsymbol{\theta})$ using AD tool;
    Update the parameters $\boldsymbol{\theta}$ using **Adam**$(\boldsymbol{\theta}) \rightarrow \boldsymbol{\theta}$;
    Check convergence;
  end
end
Return the optimal parameters $\boldsymbol{\theta}$

Once the neural network is trained using either the one-stage or two-stage training strategy, online prediction is straightforward. Given an arbitrary input **m**, iterative implementation of Eq.7.3 is then used to predict outputs for all N_t time instances. Specifi-

cally, each saturation output $\hat{\mathbf{y}}^n$ or binarized output $\hat{\mathbf{S}}^n$ at the n-th time instance is sequentially predicted by providing the geological parameters $\mathbf{m}$ and the time index t^n as inputs. This procedure is computationally efficient as it does not involve any high-fidelity model simulations. After training the cRRDB-U-Net surrogate model successfully, we can apply the cRRDB-U-Net surrogate model within other workflows. Here we will consider its use for estimation of the uncertain parameters $\mathbf{m}$ given binary image data using the ES-MDA workflow.

7.4. EXPERIMENTS AND RESULTS

In this section, the proposed deep-learning hybrid framework will be applied to two example cases representing subsurface flow model parameter estimation problems in which 2D and 3D seismic images are available as measured data respectively. Both cases consider spatially heterogeneous reservoirs with immiscible two-phase (oil and water) flow dynamics.

7.4.1. DESCRIPTION OF THE EXAMPLE CASES

The 2D reservoir model of Case 1 was created by [45] and consists of a single rock layer representing a fluvial depositional setting containing high-permeable channels (river deposits) and a low-permeable background (clay or fine sand deposits). The high-permeable and low-permeable channels represent two faices, which are indicated by binarized value 0 and 1, respectively. The value of log-permeability for these two facies has a large contrast. The permeability of clay facies and sand facies is 20 mD and 2000 mD, respectively. Given the facies indicators $\mathbf{m}$, we compute the permeability value for each grid using a transformation function ($20e^{log100\mathbf{m}}$). Fig.7.4 illustrates the distribution of facies indicators in the high-fidelity model realization used to generate synthetic measurements, as well as the locations of 3 vertical liquid producer well and 6 vertical water injection wells labeled as P_1 to P_3, and I_1 to I_6 that are drilled into the reservoir. Note that the permeability values follow a non-Gaussian distribution containing two modes with nearly constant values, which is generally considered very challenging for parameter estimation methods.

The second example case (Case 2) is a frequently used 3D benchmark model used in the SAIGUP project [38] with a realistic structure based on existing North Sea oil fields. The 3D SAIGUP benchmark model contains nine producers and nine injectors. The reference model, model settings can be found as Fig.5.1 and Table.5.3 in Chapter 5 and Table.6.1 in Chapter 6.

In our numerical experiments, the open-source simulator *Flow* from the Open Porous Media (OPM) project for reservoir modeling and simulation [46], is used to run the high-fidelity (HF) model simulations and generate the training samples. The result of the simulations are pressure and saturation grids at the times of seismic repeat surveys and time series of bottom-hole pressure (BHP) and flow rates of both oil and water in all wells. In this study we will use the saturation grids (2D and 3D images for the two examples cases respectively) simulated with the synthetic truth models as measurements. Some details about reservoir geometry, rock properties, fluid properties, and well-control settings for the 2D case are shown in Table 7.2.

7.4.2. TRAINING DATA GENERATION

The prior uncertainty in the gridblock values of permeability is captured by an en-

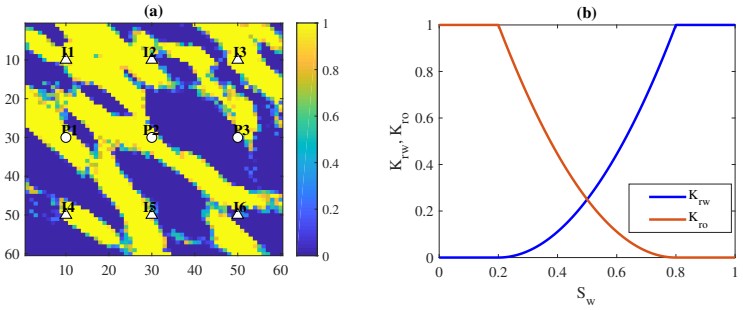


Figure 7.4: The illustration of geological realization and well placement for the 2D non-Gaussian facies model. (a) the true model. The high-permeable and low-permeable channels represent two faices, which are indicated by binarized value 0 and 1, respectively; (b) the relative permeability curves of this water-oil two-phase flooding system. The triangles and circles denote the injectors and producers, respectively.

Table 7.2: Experiment settings using OPM for the 2D non-Gaussian facies model.

Dimension, $N_x \times N_y \times N_z$	60×60×1
Number of injectors and producers	6, 3
Water/oil density	1014 kg/m^3, 859 kg/m^3
Water/oil viscosity	0.4 mP·s, 2 mP·s
Initial oil/water saturation	S_o=0.80, S_w=0.20
Bottom-hole pressure for producers	25 MPa
Bottom-hole pressure for injectors	40 MPa
Historical production time	1800 days
Pre-defined threshold value S_{con}	0.35

semble of random realizations of the permeability field. For the 2D non-Gaussian facies model, we use the 2000 facies realizations made available by [45]. For the 3D SAIGUP benchmark model we generate Gaussian-distributed realizations of log-permeability using the Stanford geo-statistical modelling software (*SGeMS*) [47]. An optimization-based principle component analysis (O-PCA) proposed in [45] and conventional PCA are applied to re-parameterize the parameter fields for these two models, respectively. 70 and 304 PCA coefficients are preserved to represent the original parameter fields in the two cases respectively and then used to generate the training and validation samples. We should note that O-PCA is particularly useful to preserve the non-Gaussian properties. It reformulated the PCA as a bound-constrained optimization problem and introduced a regularization term to generate binary or bi-modal parameter fields. More information can be referred to [45]. We randomly generate an ensemble of the Gaussian-distributed PCA coefficients and then get the facies models using O-PCA. Although we suggest that the facies indicators should be binary 0,1, the generated realizations by O-PCA contain multiple colors (around the channel edges) and not all grid cells are classified as either 0 or 1, e.g., see Fig.7.4-7.5.

Fig.7.5 and Fig.7.6 show the log-permeability (left) and simulation results at 2 times for four realizations of the Case 1 and Case 2 models respectively. The simulations results are the saturation images from which binarized images are derived based on a saturation threshold value of S_{con}=0.35. The model realizations shown in Fig.7.5(a) and Fig.7.6(a) are chosen to be the reference (synthetic truth) models for these two cases. It can be seen

that, in both cases, the evolution of the fluid facies varies strongly among the different geological model realizations, resulting in high variability in the training dataset. We should note that while we use the LHDC as innovations in the history matching step, we will be showing the binarized fluid facies maps in subsequent figures, because they can be more easily interpreted.

The simulation period in the cases are 1800 days and 5400 days respectively, and the training sample data are collected at N_t=10 intervals of 180 days for Case 1, and N_t=10 intervals of 540 days for Case 2. After reorganizing the dataset, 3000, 5000, 8000, 10000 and 15000 training images are created corresponding to 300, 500, 800, 1000, and 1500 simulation runs respectively for the 2D non-Gaussian facies model.

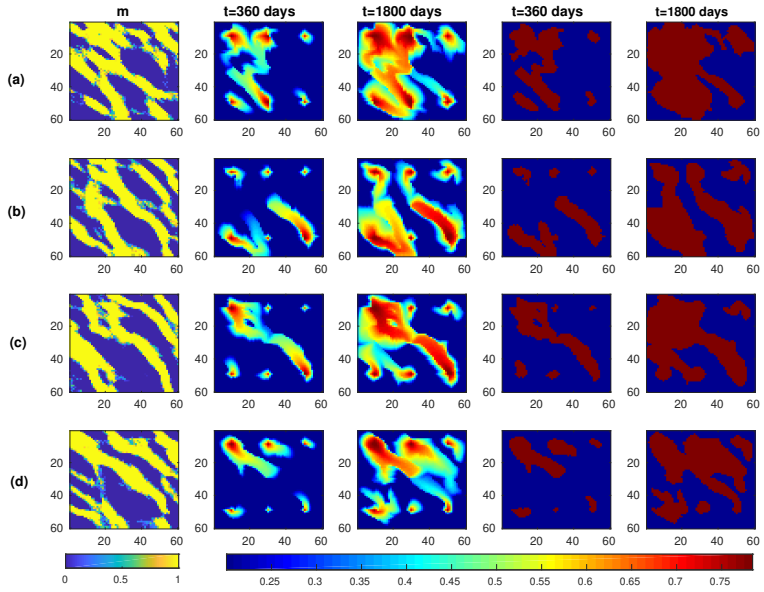


Figure 7.5: Predictions of the time-varying saturation maps and binarized contour maps corresponding to four non-Gaussian realizations for the 2D synthetic non-Gaussian facies model. The contour maps are obtained through applying a front threshold value of 0.35. Subfigures (a) - (d) represent four different model realizations and their predictions at day 360 and day 1800.

7.4.3. PERFORMANCE METRICS

To evaluate the quality of cRRDB-U-Net surrogate model with respect to the number of training samples, N_{test} independent model simulations based on the HF and surrogate models are performed. We define an evaluation metric γ_s^n to represent the pixel-wise mismatch between two binarized images at timestep n evaluated over N_{test} validation samples is defined as

$$\gamma_s^n = \frac{1}{N_{test} N_{\mathbf{m}}} \sum_{i=1}^{N_{test}} \sum_{j=1}^{N_{\mathbf{m}}} \| \hat{F}_j^{i,n} - F_j^{i,n} \|, \quad n = 1, \ldots, N_t. \tag{7.10}$$

and can be interpreted as the fraction of incorrectly labeled grid cells (pixels) in the grid

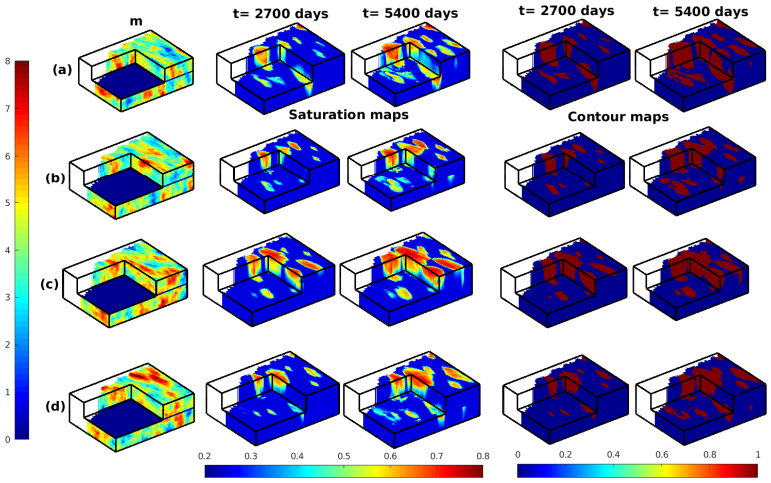


Figure 7.6: Predictions of the time-varying saturation maps and binary contour maps corresponding to four Gaussian realizations of logarithmic permeability for the 3D SAIGUP benchmark model. The contour maps are obtained through applying a front threshold value of 0.35. Subfigures (a) - (d) represent four different model realizations and their predictions at day 2700 and day 5400.

(image). $\hat{\mathbf{F}}_j^{i,n}$ and $\mathbf{F}_j^{i,n}$ denote the binarized fluid facies maps predicted from the high-fidelity model (HFM) and cRRDB-U-Net surrogate model respectively for validation sample i, gridblock j and timestep n. If the two images are equal, γ_s^n will attain it minimum value of 0, if no two values in the images are identical, γ_s^n=0. Note that differences between the binary images can be related to errors in the location of the fluid front that separates the two fluid phases in the reservoir. The overall field-average values over all N_t time instances, denoted as γ_s, is given by

$$\gamma_s = \frac{1}{N_t}\sum_{n=1}^{N_t}\gamma_s^n. \tag{7.11}$$

The binary facies indicators or log-permeability values are the only uncertain parameters and they are calibrated using the proposed ES-MDA framework using cRRDB-U-Net surrogate model. With the aim of analyzing the history matching results, we introduce two error metrics measured on data misfits e_{obs} and parameter misfits $e_{\mathbf{m}}$ as follows,

$$e_{obs} = \sqrt{\frac{\sum_{i=1}^{N_d}\sum_{j=1}^{N_t}(\mathbf{d}_{obs}^{i,j} - \mathbf{d}_{upt}^{i,j})^2}{N_d N_t}}, \quad e_{\mathbf{m}} = \sqrt{\frac{\sum_{i=1}^{N_m}(\mathbf{m}_{true}^{i} - \mathbf{m}_{upt}^{i})^2}{N_m}}. \tag{7.12}$$

where, $\mathbf{d}_{obs}^{i,j}$ and $\mathbf{d}_{upt}^{i,j}$ represent the measurements and simulated data using the updated model, respectively. $\mathbf{m}_{true}^{i}$ and $\mathbf{m}_{upt}^{i}$ denote the binary facies indicators or logarithmic permeability values from the 'true' model and updated model, respectively.

7.4.4. TRAINING AND VALIDATION OF THE SURROGATE

The parameter settings for training the cRRDB-U-Net surrogate are summarized in Table 7.3. The training parameter settings for the 3D SAIGUP model can be found as Table.6.1 in Chapter 6. These parameters were used used to train the surrogate models for

all listed training set and batch sizes. Taking the 2D synthetic model as an example, during the training process, 100 training samples are randomly selected from the entire, e.g., 20000, training dataset to optimize the neural network parameters in each iteration. In the following we will discuss the quality of predictions (i.e. the fluid facies maps or images) generated with the trained surrogate.

Table 7.3: Training parameters settings for training cRRDB-U-Net model corresponding to the 2D case.

Training size (N_s)	300,500,800,1000,1500
Re-organized training size ($N_s \times N_t$)	2000,6000,10000,16000,20000
Testing size (N_{test})	200
Initial learning rate	0.005
Optimizer	*Adam*
Batch size	100
Number of epochs	100

Fig.7.7 and Fig.7.8 show the fluid facies maps predicted by the HF and cRRDB-U-Net surrogate models for the reference realizations of Case 1 and Case 2 respectively. It can be seen in Fig.7.7 for Case 1 that the trained surrogate model is capable of predicting accurate fluid distributions. For instance, the presence of single or multiple fluid fronts at different times is correctly captured as seen in Fig.7.7. Small errors are noticeable, however, which are associated with small errors in predicted front locations. These errors are seen to decrease with increasing number of training samples N_s. The impact of the number of training samples is particularly clear at early times where multiple isolated fluid fronts are developing. Fig.7.8 displays analogous results at day 2700 and day 5400 for Case 2. in this case the possibility of vertical flow results in somewhat larger regions of error for small training set size.

We further assess the quality of proposed surrogate model based on the γ_s^n and γ_s metrics which quantify the quality of image reconstruction, particularly taking into account the structural similarity of edges inside the images. In order to verify the robustness of the surrogate model, the values of γ_s^n corresponding to $N_{test} = 200$ independent validation samples are plotted in Fig.7.9. Increasing the number of training samples progressively improves the values of γ_s^n metric. Values are relatively high at early times because the fluid fronts have not yet expanded very strongly such that most of the domain will still contain only the initial fluid phase (oil in this case) in all test cases. The values are gradually decreasing with time as the injected fluid phase is replacing the initially present phase, leading to expanding fluid fronts, and correspondingly higher chances of differences between the true and predicted front locations. The surrogate is obtaining relatively lower γ_s^n for Case 2 than for Case 1. The overall field-average error γ_s is 4.24% for 1000 training samples, suggesting a relatively high degree of accuracy.

7.4.5. HISTORY MATCHING RESULTS - CASE 1

In the previous section we verified the applicability of our proposed surrogate model for predicting dynamic fronts. We now apply the surrogate model within a history matching (i.e. data assimilation) workflow, where the image-oriented distance parameterization is used to characterize differences between observed and simulated images, and the ES-MDA method is used to minimize these differences by updating the underlying model

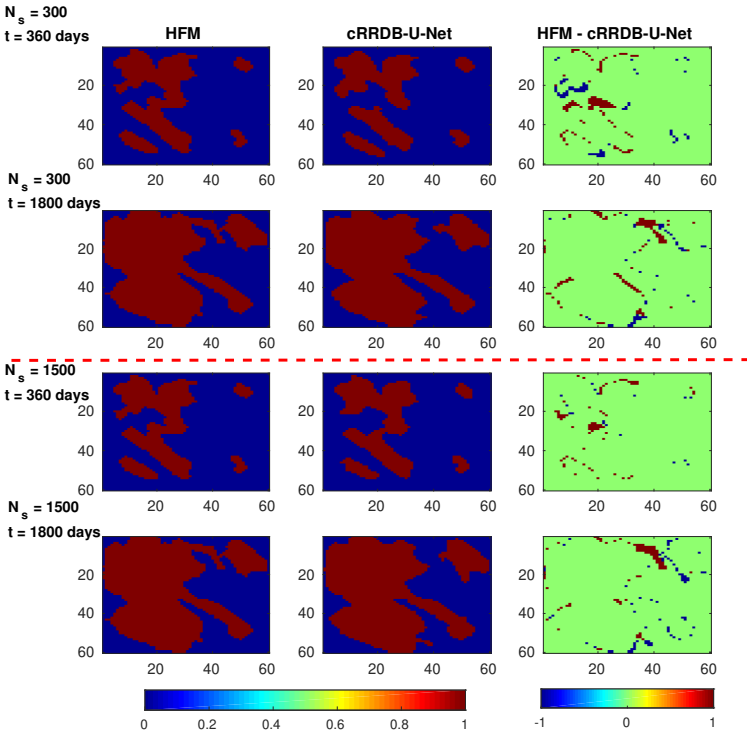


Figure 7.7: Comparisons of time-varying fluid facies maps predicted from HFM, cRRDB-U-Net and their absolute errors at day 360 and day 1800 for the 2D synthetic non-Gaussian facies model. The cRRDB-U-Net models are trained using N_s = 300 and 1500 samples.

parameters. In the examples these parameters are the properties of the grid-cells (specifically, the permeability) of the HF model. While the total number of active grid cells are 3600 for Case 1 and 78720 for Case 2, the PCA parameterzation has reduced this to 70 and 304 coefficients respectively (See Section 4.1). The observations correspond to the LHDC metric derived from applying a threshold value of S_{con} = 0.35 to the saturation maps at day 360 or at day 1800 resulting from simulation of the HF reference (synthetic truth) model, and the corresponding maps from a surrogate model simulation. The standard deviation of the uncorrelated measurement errors is assumed to be 30 m, which is close to the length of one grid block. Results from the hybrid workflow will be compared to results obtained by using an ensemble of HF model realizations instead of the trained surrogate. We choose N_a = 15 iterations and an ensemble size of N_e = 500 as standard values for the ES-MDA workflow and we compare results obtained with the hybrid workflow for a range of training set sizes.

FIXED COMPUTATIONAL BUDGET

We first compare results from the hybrid workflow against results from the HF model workflow for (a) a fixed number of HF model simulations (1500), and (b) a fixed number of iterations (15). Fig.7.10 compares the data misfits for the prior and posterior ensemble of realizations obtained with the DNN trained with N_s = 1500, and the posterior HF mode

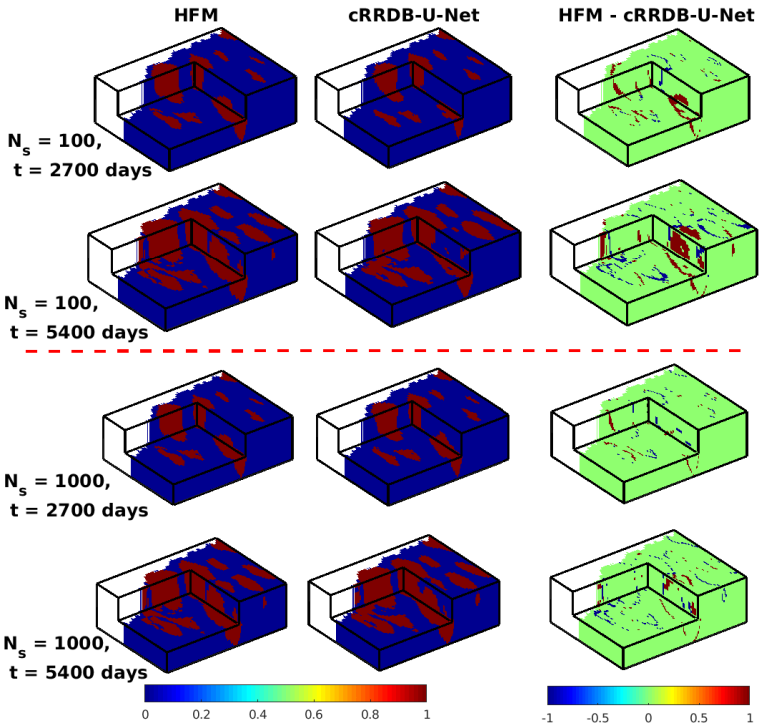
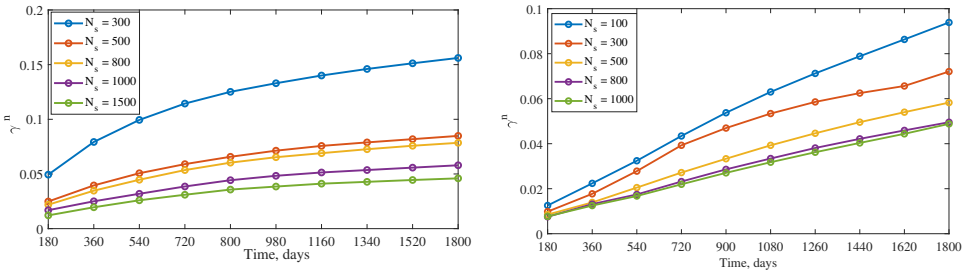


Figure 7.8: Comparisons of time-varying fluid facies maps predicted from HFM, cRRDB-U-Net and their absolute errors at day 2700 and day 5400 for the 3D SAIGUP model. The cRRDB-U-Net models are trained using N_s = 100 and 1000 samples.



(a) 2D non-Gaussian facies model

(b) 3D Gaussian SAIGUP model

Figure 7.9: The plots of γ_s^n values of cRDDB-U-Net with respect to the number of training samples N_s = 300, 500, 800, 1000, and 1500 samples for the . (a) 2D non-Gaussian facies model; (b) 3D Gaussian SAIGUP model.

realizations resulting from N_a = 15 iteration (HFM(15)) and N_a = 3 iterations (HFM(3)). Note that HFM(3) requires 3 × 500 = 1500 HF model simulations, which is the equal to the 1500 simulations used to train the DNN, while HFM(15) requires 15 × 500 = 7500 HF model simulations. Results are presented for images obtained at 360 days and 1800 days separately. The DNN-based data misfits are larger than those obtained with 15 ES-MDA

iterations with the HF model, but smaller than those obtained with 3 iterations. These results indicate that a significant reduction of computational cost should be feasible for a given desired quality of the result. In this study, the total number of HFM simulations is taken as an indicator of the computational cost, since the GPU time for running the DNN surrogate model is negligible compared to the CPU time for running the HF model. The runtime for training the DNN with data from $N_s = 1000$ samples is about 25 minutes, which is equivalent to running 300 HF model simulations. In other word, the computational cost of the off-line training stage is equivalent to about 1300 HF model runs, while the cost of HFM(15) is equal to 7500 HFM simulations. The use of cRRDB-U-Net as a surrogate model reduces the computational time by a factor 5.8 for the 2D synthetic model of Case 1. Note that the computational saving of our proposed deep-learning method will increase linearly with the ensemble size.

Fig.7.11 displays the average effective number of data as a function of iterations. Preprocessing the imaging-type data using distance-based parameterization can drastically decrease the number of data, for example from original 3600 to 580 at day 1800. A small effective number of data indicates a high degree of similarity of two images. The effective number of data gradually decreases as the iterations, revealing that the predicted water fronts from the posterior models become closer to the observed water fronts and the model uncertainty is hence reduced.

The ensemble mean and standard deviations of the updated ensemble of permeability before and after history matching at 360 days and 1800 days are displayed in Fig.7.12. These statistics are calculated as the per-grid-block averages and standard deviations over the 500 realizations in the ensemble $\mathbf{m}_1^{N_a},\ldots,\mathbf{m}_{N_e}^{N_a}$, where $N_e = 500$ and $N_a = 15$. Values are compared to the corresponding statistics calculated for the prior ensemble (iteration 0 instead of N_a). The channel structures can be reconstructed almost perfectly from both the 360 and 1800-day images. While the large standard deviations in the prior standard deviation suggest that in the initial ensemble the locations of channel boundary positions are strongly varying, a comparison of the posterior standard deviation maps and the true parameter map indicates that the channel boundaries are consistently aligned with the truth. Larger variability is mostly found in this bands a few grid blocks wide along the true channel boundaries. The field-average posterior ensemble standard deviation has approximately decreased from 0.45 to 0.16, which indicates a significant reduction of model uncertainty. We also should note that since the original realizations conditioned to permeability values at the well locations, the majority of realizations, e.g., the prior ensemble mean, already seem to have channels at approximately the right locations.

EFFECTS OF TRAINING SAMPLE SIZE

We repeat the workflow for a series of increasing training sizes N_s = 300,500,800,1000 and 1500. Fig.7.13 shows the posterior parameter misfits as a function of the number of training samples. It clearly can be seen that the accuracy of the hybrid workflow results gradually improves as the number of training samples increases, especially for data gathered at day 360. Results do not improve much for $N_s > 800$. For data gathered at 1800 days, the best results results are obtained, somewhat surprisingly, for $N_s = 500$. Since the different training scenarios rely on different random parameter realizations, the results of each training stage could be impacted by the samples included in the training set. ES-MDA is a statistical method and results could also vary slightly if the ensemble statistics are affected by these results as well. One could, in principle, repeat each experiment with training sets

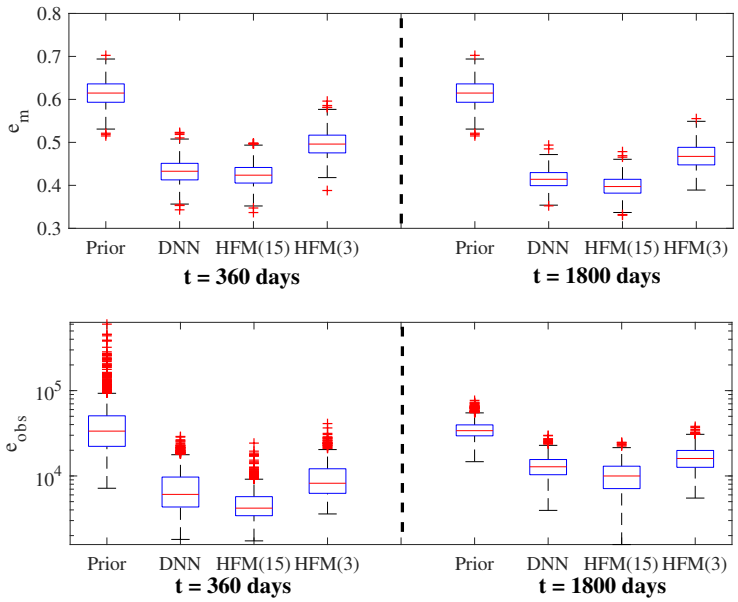


Figure 7.10: Boxplots of data misfits and parameter errors using cRRDB-U-Net surrogate model and high-fidelity models for the 2D non-Gaussian facies model. The cRRDB-U-Net surrogate model is trained by 1500 training samples. HFM(3) and HFM(15) are the abbreviations of ES-MDA-HFM with 3 and 15 iterations, respectively.

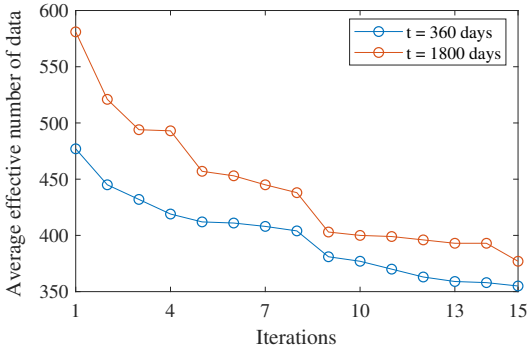


Figure 7.11: The over-all effective number of data during the iteration process.

containing different random samples, to quantify the impact of the random sample selection on the results, but we have not done that here. The overall trends in the results, however, clearly indicate that the accuracy of the surrogate model and its corresponding history matching results will improve as the number of training samples increases.

Fig.7.14 shows the posterior ensemble mean and standard deviation (STD) for the different N_S scenarios. It is evident that the main structure of binary channels can be successfully reconstructed in all cases, also for the smaller values of N_s, and that the posterior ensemble mean gradually becomes much closer to the true model as the number of train-

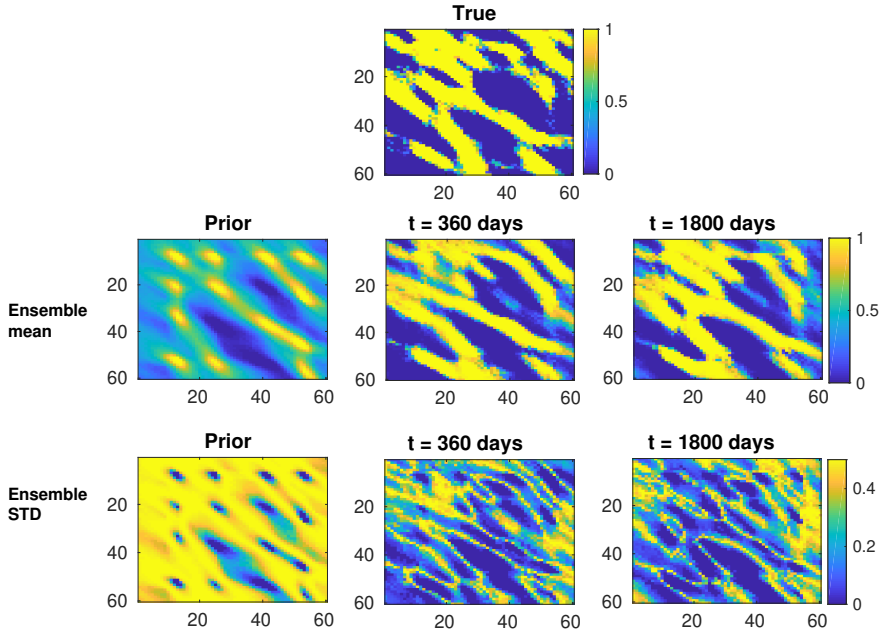


Figure 7.12: Ensemble mean and standard deviation (STD) of binary facies indicators for the prior models and posterior models at day 360 and day 1800.

ing samples increases. Based on Fig.7.13 Fig.7.14 it can be concluded that results with similar quality as the HF model workflow requiring 7500 simulations can be obtained using the hybrid surrogate-supported workflow with 500-800 simulations. Overall, we can collect more informative measurements at day 1800, and therefore obtain relatively better results than that of day 360.

We summarize the posterior parameter misfits $e_{\mathbf{m}}$ and computational cost indicated by the number of HFM simulations. It can be obviously seen that the proposed cRRDB-U-Net surrogate model maintains a satisfactory accuracy even using a small number of training samples. For example, the cRRDB-U-Net model using $N_s = 500$ training samples is still capable of generating satisfactory posterior models which are very similar to the true model. In terms of computational efficiency, it clearly reveals that our proposed history matching framework only needs a relatively small number of high-fidelity model simulations, e.g., 500, in contrast to 7500 HFM simulations required by the conventional ES-MDA method.

COMPARATIVE STUDY OF ONE-STAGE AND TWO-STAGE TRAINING STRATEGIES

The results presented so far were obtained using the two-stage training strategy detailed in section 3.3.3. We repeated the training procedure for $N_s = 300, 500, 800, 1000, 1500$ in order to compare two-stage strategy with the one-stage strategy, which only considers the BCE loss function which is commonly used for image segmentation tasks. Results from the one-stage training procedure are illustrated in Fig.7.15. When comparing with the results shown in Fig.7.9(a), it is clear that the two-stage strategy achieves lower validation

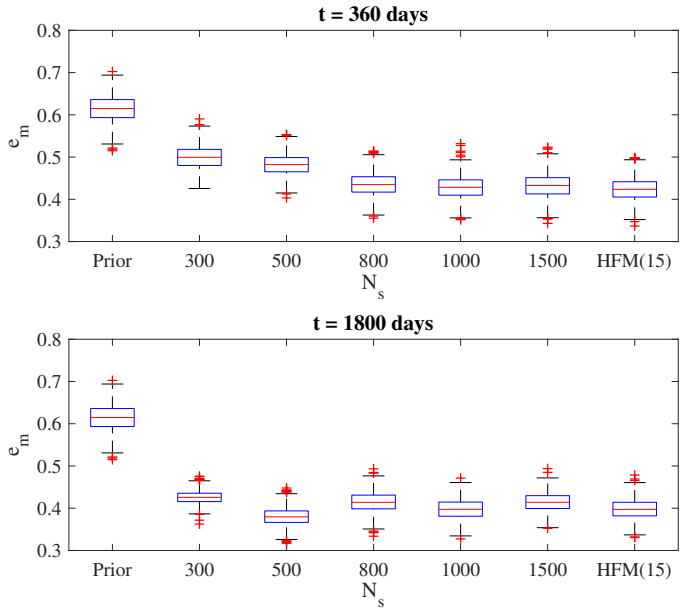


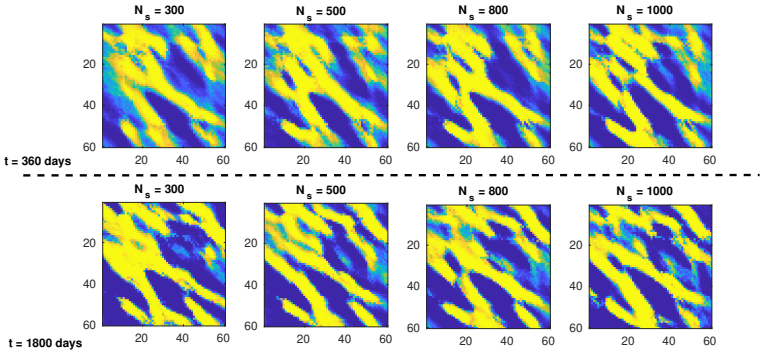
Figure 7.13: Boxplots of parameter misfits $e_{\mathbf{m}}$ using cRRDB-U-Net surrogate model as a function of the number of training sample N_s. The two rows are corresponding to day 360 and day 1800, respectively.

errors than the one-stage strategy. Fig.7.16 shows a comparison of the posterior parameter misfits for these two training strategies for Ns = 300 and N_s = 1500. When we train the surrogate models with N_s=300 training samples, the one-stage strategy achieves γ values of 9.85% and 18.82% at day 360 and day 1800, which are larger than the 7.12% and 15.12% errors from two-stage strategy. However, the differences between these two training strategies decrease as the number of training samples increases. For example, when we train the surrogate model using 1500 samples, the one-stage strategy achieves γ_s values of 3.01% and 5.18% at day 360 and day 1800, respectively, which are only slightly larger than the values of 2.25% and 4.89% from the two-stage strategy. Overall, two-stage training is found to improve the predictions, particularly when small numbers of training samples are available.

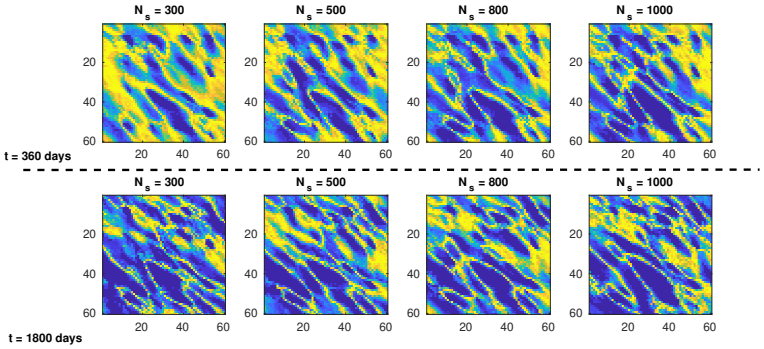
We remind the reader that the first training aims to minimize the MSE loss expressed in terms of continuous saturation values. In this study we have set the MSE loss weight ω to 0.01. In general, a lager weight ω will lead to a better approximation of the spatially continuous saturation. However, the aim of our proposed surrogate modelling and data assimilation workflow is to accurately predict binary-type image data. Some limited experimentation with alternative values of $omega$ suggested that larger values will lead to less accurate reproduction of fluid facies. Based on visual inspection of results for different values of omega, an value of 0.01 appeared to be a good choice, but that the optimal value should be subject to further investigation.

COMPARISON WITH CONVENTIONAL RESIDUAL U-NET

To verify the effectiveness of the RRDB structure of the deep CNN, a comparative study of cR-U-Net and cRRDB-U-Net was performed. Fig.7.17 show the parameter error met-



(a) Posterior ensemble mean



(b) Posterior ensemble standard deviation (STD)

Figure 7.14: Posterior ensemble mean and standard deviation (STD) of parameter estimates obtained for 300,500,800,1000 and 1500 training samples. (a) Posterior ensemble mean; (b) posterior ensemble STD.

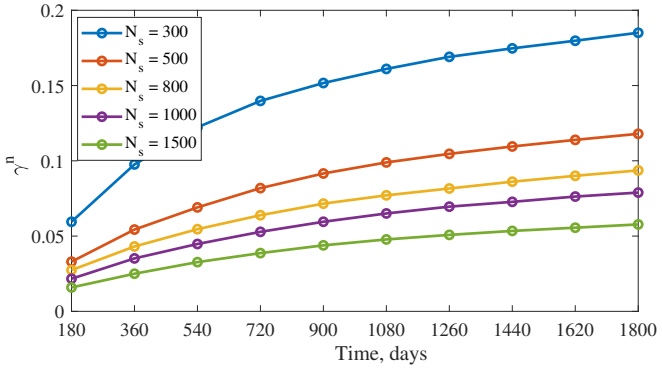


Figure 7.15: The γ_s^n values of cRDDB-U-Net with respect to the number of training samples N_s = 300, 500, 800, 1000, and 1500, respectively. The results correspond to cRRDB-U-Net surrogate model using **OS** training strategy.

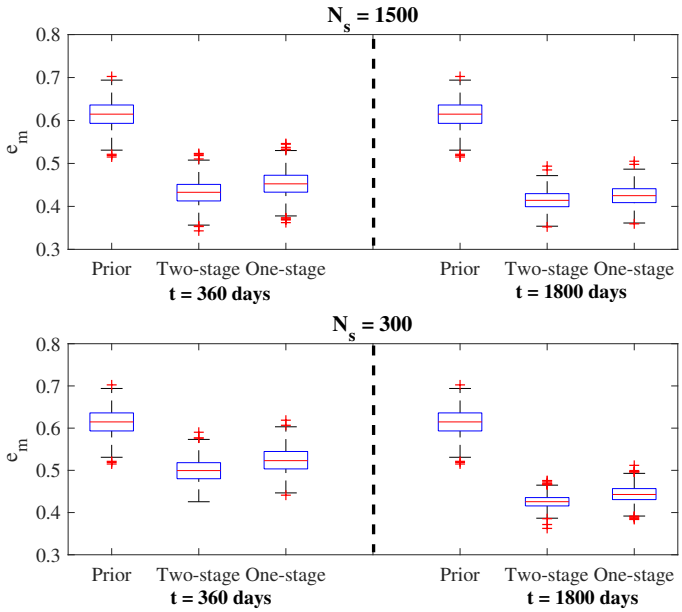


Figure 7.16: Boxplots of parameter misfits $e_{\mathbf{m}}$ for the **TS** and **OS** training strategy. The two rows are corresponding to N_s=1500 and N_s=300 training samples, respectively.

ric γ_s for the cR-U-Net surrogate model. Comparing to the results in Fig.7.9(a), the improvements from the RRDB structure are clearly indicated lower field-average relative error γ_s values, especially for the larger training set sizes. When we train the deep-learning surrogate models using 300 samples, the cR-U-Net and cRRDB-U-Net obtain comparable results with γ_s values around 18%. Although RRDB enables us to train deeper neural networks for better approximations of spatially discontinuous fluid-fronts, more parameters are introduced by the RRDB structure that need to be trained as well. Training the cRRDB-U-Net surrogate model with a small number of samples might cause overfitting problems, and hence may not achieve desirable improvements in comparison to cR-U-Net. As the number of training samples increases, however, the improvements in quality of the cRRDB-U-Net surrogate model are much larger than that of cR-U-Net. For example, when we train these two surrogate models using 1500 samples, cR-U-Net and cRRDB-U-Net obtain γ_s values of 12.5% and 4.89% at day 1800, respectively. Overall, the RRDB structure significantly improves the surrogate model's ability to predict the positions of fluid fronts, especially for large training set sizes. Fig.7.18 shows a comparison of the posterior parameter misfits for the cRRDB-U-Net and cR-U-Net surrogate models. It clearly can be seen that the RRDB structure outperforms standard U-Net in terms of the posterior parameters errors.

7.4.6. HISTORY MATCHING RESULTS - CASE 2

In this section we present a more realistic application of the proposed surrogate model that involves a more complex 3D reservoir model with uncertain permeability in all 78720 active grid cells. The dimension of the uncertainty space is reduced by capturing the spatial relationships between individual grid block values in a total of 304 coefficients for cor-

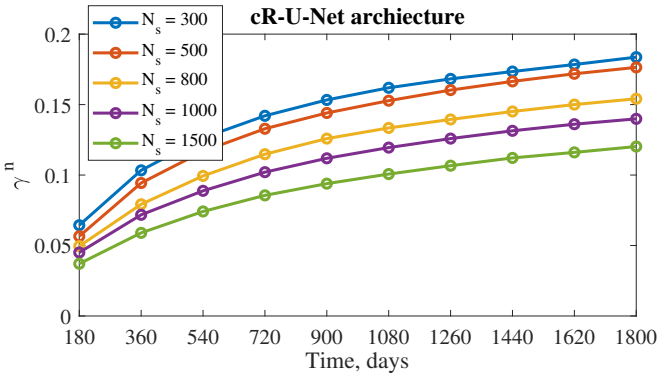


Figure 7.17: The γ_s^n values of cR-U-Net with respect to the number of training samples N_s = 300, 500, 800, 1000, and 1500 samples, respectively.

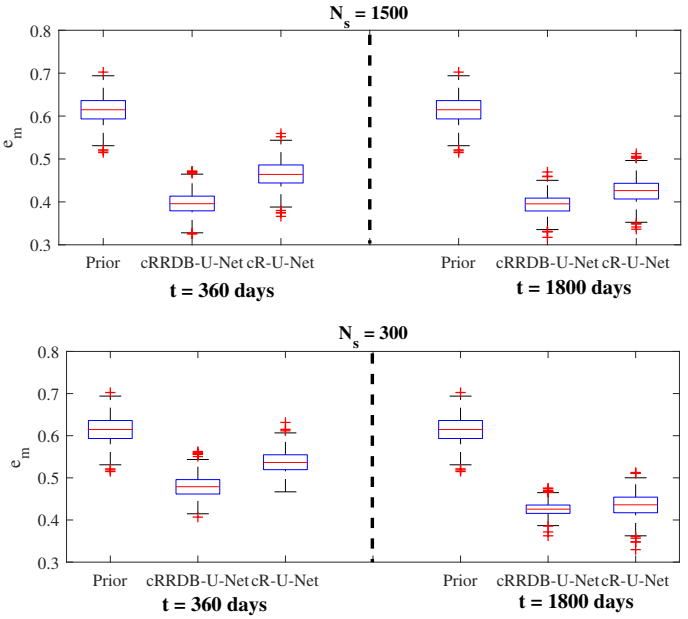


Figure 7.18: Boxplots of parameter misfits $e_{\mathbf{m}}$ using cR-U-Net and cRRDB-U-Net surrogate models. The two rows are corresponding to N_s=1500 and N_s=300 training samples, respectively.

responding 3D patterns obtained by Principle Orthogonal Analysis of a large number of plausible log-permeability realizations. The PCA coefficients are the only uncertain parameters, and are updated using the proposed hybrid workflow. We assume that saturation images are available at either 2700 days or 5400 days, after inversion of time-lapse seismic data. We furthermore assume that the data has sufficiently high resolution to enable estimation of saturation in all 22 layers of the reservoir. We use an ensemble size of N_e = 500 and a fixed number of N_s = 10 iterations for the ES-MDA algorithm.

Fig.7.19 shows the parameter misfits $e_{\mathbf{m}}$ before and after history matching for training

set sizes N_s= 100, 300, 500, 800 and 1000. Fairly consistent improvements in the accuracy of the results can be observed with increasing training set size. In this case study, the surrogate models trained with 100 and 300 samples obtain similar history matching results. Similar results are also observed for the surrogate models trained with 500 and 800 samples. A possible explanation is that a random sampling strategy might generate some ineffective training data, which have no significant contribution to the improvement of the history matching results. One could imagine that procedures in which additional training samples are generated guided by intermediate history matching results could be more effective, but we have not investigated this idea further.

Our proposed ES-MDA-DNN is able to generate history matching results comparable to ES-MDA-HFM. However, to achieve the same accuracy, ES-MDA-DNN and ES-MDA-HFM require respectively1000 and 5000 high-fidelity model simulations. The use of cRRDB-U-Net as a surrogate model reduces the computational time by a factor of 5 for the problem defined in this case. We should emphasize that the computational saving will become more substantial when a larger ensemble sizes are used. The ES-MDA-DNN hybrid worfklow requires a relatively small number of high-fidelity model runs at the offline training stage and the computational cost does not grow with the ensemble size. Compared to the HFM(10) results, the hybrid DNN workflow is generally performing a little bit worse but at a lower cost. The use of 500 initial realizations might be much larger than that would be used in applications to complex fields, where an ensemble size of 200 is often considered as the realistic maximum. Fig.7.20 shows the posterior ensemble parameter errors $e_{\mathbf{m}}$ as a function of ensemble size and number of ES-MDA iterations. It can be clearly seen in Fig.7.20(a) that our proposed ES-MDA-DNN obtains a better solution that is at present practically feasible at lower computational cost (e.g., 1000 simulations). By contrast, ES-MDA-HFM would lead to a total number of 2000 HFM simulations over 10 iterations. Furthermore, the obtained results with an HF model ensemble size of 100 suggest ensemble collapse to some degree. As illustrated in Fig.7.20(b), for a fixed computational budget (1000 simulations in this example), the hybrid surrogate-assisted workflow delivers more accurate results than ES-MDA-HFM.

Fig.7.21(a) shows the fluid-front positions before and after history matching at day 2700 and day 5400. The fluid-fronts for the prior models are significantly different from the observed front positions. There are nine completely discrete front contours around the nine injectors at day 2700. After the history matching, the fluid-fronts almost match the observed ones. Although the front positions are relatively more complex at day 5400 than at day 2700, a very good result still can be obtained. When the cRRDB-U-Net surrogate models are trained with a small sample size, e.g., N_s = 300 or N_s = 500, several fluid-front positions near injectors I1, I2 and I6 (the lower left part of the reservoir model) are not well matched, and the permeability around these positions cannot be significantly improved. In contrast, the fluid-front positions near the injector I9, I4 and I7 (the upper right part of the reservoir) are matched very well, which leads to good calibrations of the reservoir models around this area. Fig.7.21(b) shows the average effective number of data as a function of iterations. Comparing to the above 2D synthetic model, the use of distance-based parameterization obtains a much larger reduction of effective data, e.g., from original 157,440 to 17,095 after history matching, which quantifies the reduction of model uncertainty. This result also demonstrates a high scalability of our proposed hybrid workflow to practical applications.

Fig.7.22 shows the posterior ensemble mean and standard deviation corresponding to

different training sample sizes N_s=100, 300, 500, 800 and 1000, respectively. It clearly can be seen that the posterior ensemble means are very close to the true model. The geological parameters, e.g. highly permeable zones, are successfully reconstructed using even a small number of training samples, e.g. N_s = 300. The true model is gradually reproduced as the number of training samples increases. The significant reduction in the standard deviation of the spread of parameter estimates, e.g., from approximately 2.5 in the prior to 0.5 in the posterior ensemble, further indicates a high accuracy of the history matching result. In order to further verify the reliability of the posterior models from our proposed surrogate-based history matching approach, we compare predictions of the well water injection rate (WWIR) and well water-cut (WWCT) for nine injectors and nine producers, see Fig.7.23. These quantities were not used in the history matching procedure and can therefore be viewed as independent validation data. In order to generate the predictions, the permeability estimates obtained with the surrogate-assisted hybrid workflow are used as input for simulations with the HF model simulator. Although the predictions of the initial models are significantly different from the true model, after history matching to the binary image data, the mean and spread in rate predictions from the updated models are much more consistent with the rates generated with the HF truth model.

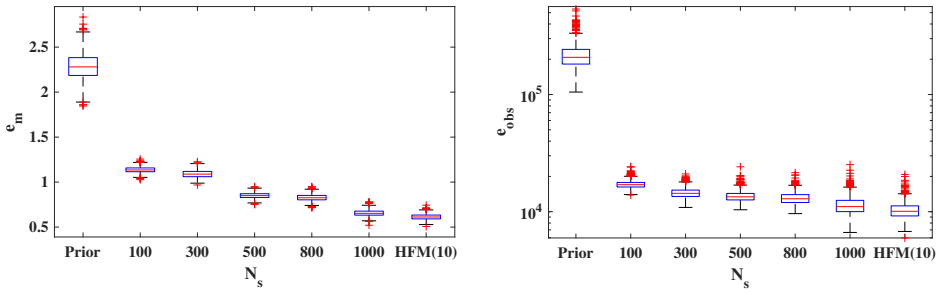


Figure 7.19: Boxplots of data misfits and parameter misfits using cRRDB-U-Net surrogate model and high-fidelity model for the 3D benchmark SAIGUP model. The cRRDB-U-Net models are trained with N_s=100, 300, 500, 8000 and 1000 samples, respectively.

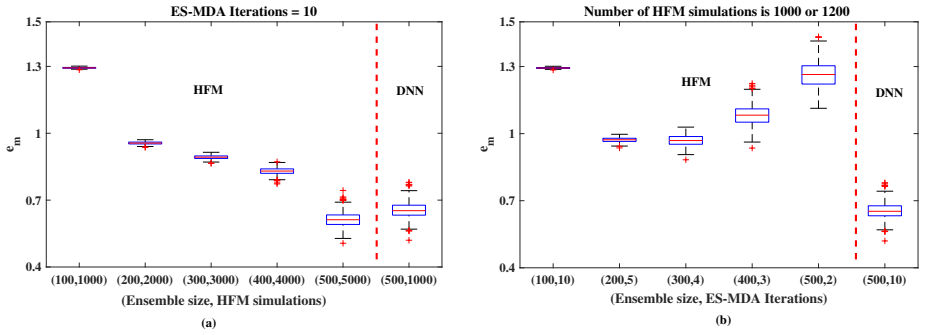
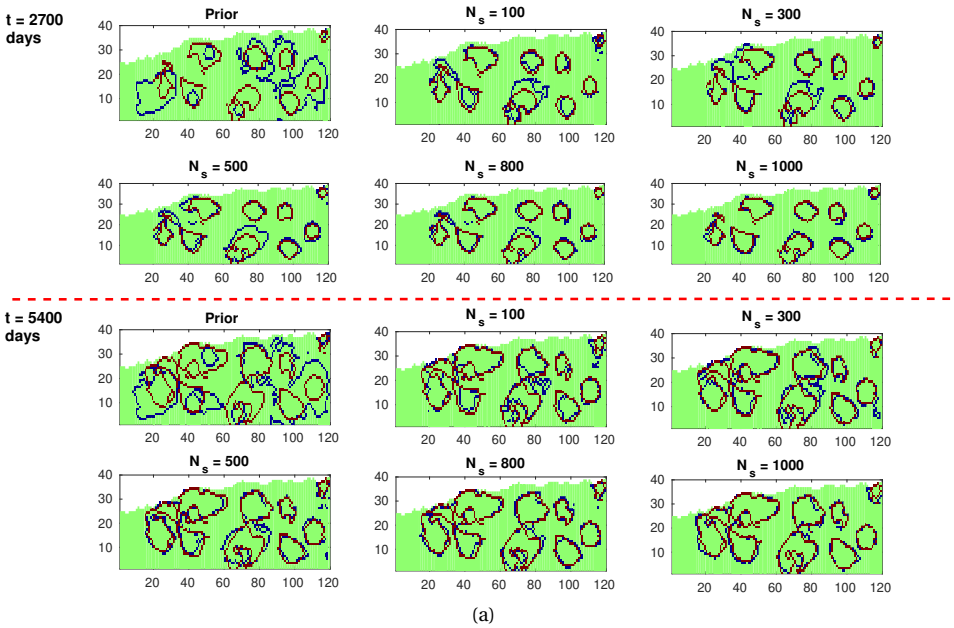
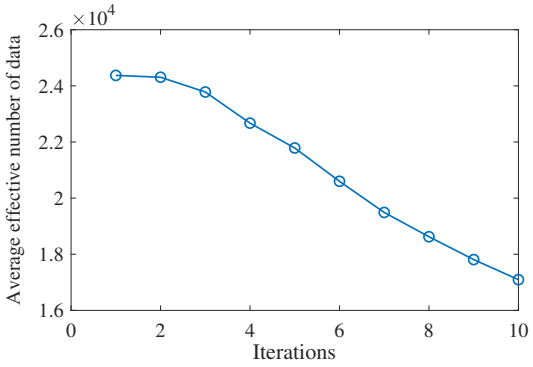


Figure 7.20: Boxplots of posterior ensemble parameter errors $e_{\mathbf{m}}$ as a function of ensemble size N_s and ES-MDA iterations. (a) The ES-MDA iteration number is 10 and the ensemble size ranges from 100 to 500; (b) The number of HFM simulations is about 1000 or 1200.



(a)

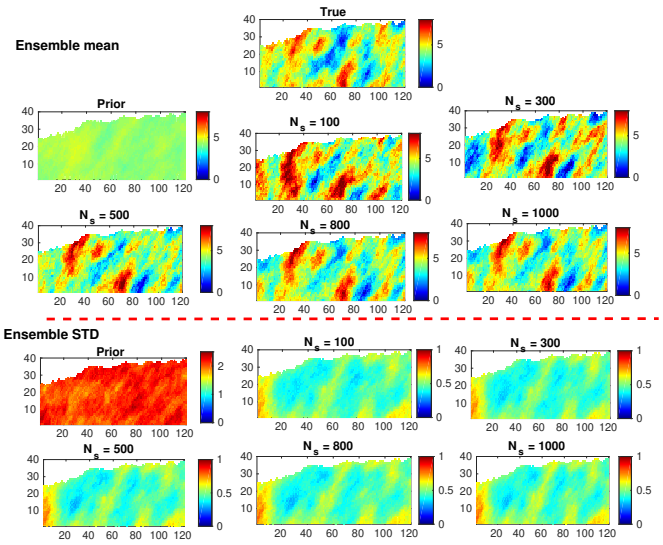


(b)

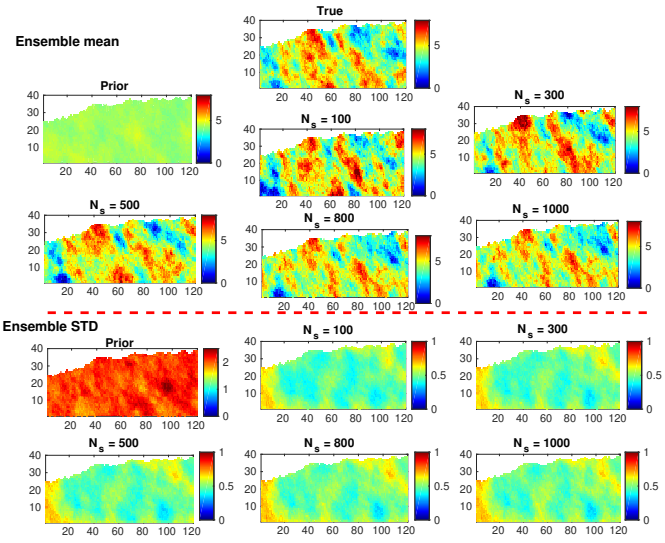
Figure 7.21: Characteristics of the ensemble of innovations used by LHDC for 3D model. (a) Simulated water-fronts after 2700 days and 5400 days in one posterior model realization before and after history matching using the cRRDB-U-Net surrogate model. The observed and simulated fronts are denoted in blue and red lines, respectively. This figure displays the water-fronts of the 11th vertical layer. (b) The over-all effective number of data during the iteration process.

7.4.7. COMPUTATIONAL COST

In this section, we will briefly discuss the aspects of computation cost of the proposed workflow for parameter estimation using cRRDB-U-Net surrogate modelling. We denote the cost of a high-fidelity model simulation and a surrogate model simulation as C_{HFM} and C_S respectively, the number of training data N_s, the ensemble size N_e and the number of iterations N_a, the total computational cost of the overall workflow based on HF model



(a) The 1st vertical layer



(b) The 11th vertical layer

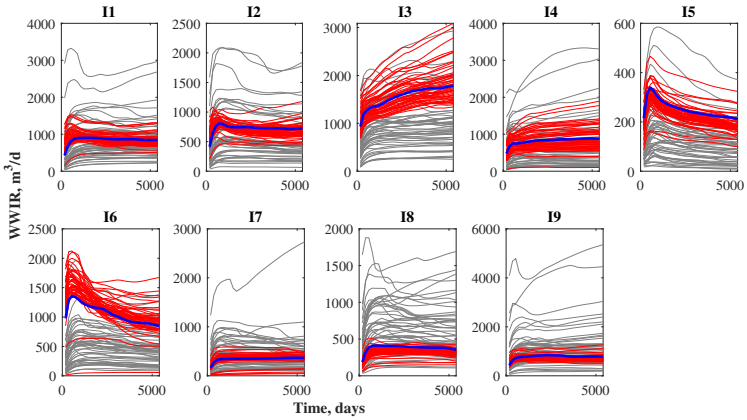
Figure 7.22: Comparison of the updated ensemble posterior realizations using cRDDB-U-Net surrogate model with respect to the number of training samples N_s = 100, 300, 500, 800 and 1000, respectively. (a) The 1st vertical layer; (b) The 11th vertical layer.

simulations only is

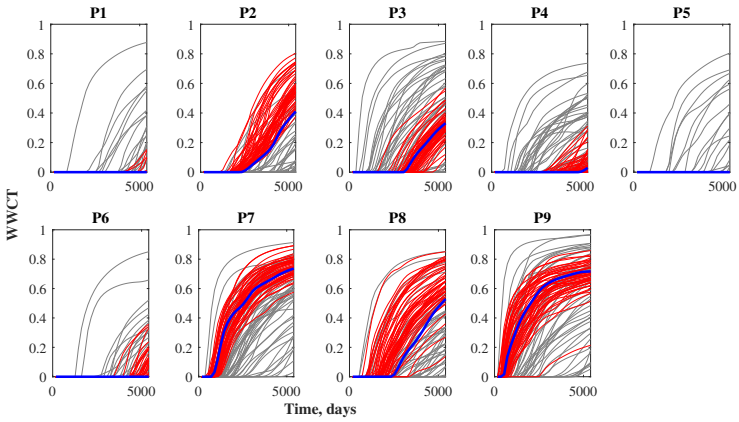
$$T_{HFM} = N_a \cdot N_e \cdot C_{HFM}. \tag{7.13}$$

For the surrogate-assisted hybrid workflow, the total cost is

$$T_S = N_s \cdot C_{HFM} + C_{train} + N_a \cdot N_e \cdot C_S. \tag{7.14}$$



(a) Well water injection rate (WWIR)



(b) Well watercut (WWCT)

Figure 7.23: Predictions of water injection rate and watercut for the nine injectors and nine producers. The gray lines-initial models, blue lines-reference model, red lines-updated models from the cRRDB-U-Net surrogate trained with $N_s = 1000$ samples.

The runtimes of a single HF model simulation for the 2D model of Case 1 and the 3D model of Case 2 are about 5.0 s and 250.0 seconds respectively on a machine with i5-4690 Intel CPUs (4 cores, 3.5GHz) and 24 GB memory. In comparison, the runtime of the cRRDB-U-Net surrogate model is about 0.1 seconds for both cases, so it is a factor of $10^2 - 10^3$ smaller than the runtime for the HF model. The cRRDB-U-Net surrogate model is trained on a NVIDIA GeForce GTX 745 GPU and separately takes about 25 minutes and 31.4 hours using 1000 training samples for the 2-D and 3-D case respectively, which are equivalent to run approximately 300 and 450 HFM simulations, respectively. For the comparison, we set $C_S \approx 10^{-3} C_{HFM}$ and $C_{train} \approx 10^{-1} \cdot N_s \cdot C_{HFM}$ and determine cost reduction T_S / T_{HFM} expressed in terms of HF model simulations as a function of the parameters N_s, N_e and

N_a,

$$\frac{T_S}{T_{HFM}} = \frac{N_s + 10^{-1}N_s + 10^{-3}N_aN_e}{N_aN_e} \approx \frac{Ns}{N_aN_e}. \tag{7.15}$$

For typical values N_s = 1000, N_e = 200 and N_a = 10, the cost ratio is about 0.5. For an ensemble size of 500 this is about 0.2, and if the training size can be reduced to 500 as well, the ratio becomes 0.1. We note that in the surrogate-assisted workflow the number of iterations could be increased significantly without incurring higher computational cost since the cost of simulating the proxy is almost negligible relative to simulating the HF model. We may expect that a higher number of iterations can contribute to improved performance of the ES-MDA algorithm to a certain extend.

REFERENCES

[1] P. Suetens, *Fundamentals of medical imaging* (Cambridge university press, 2017).

[2] T. Lopez, A. Al Bitar, S. Biancamaria, A. Güntner, and A. Jäggi, *On the use of satellite remote sensing to detect floods and droughts at large scales,* Surveys in Geophysics **41**, 1461 (2020).

[3] P. S. Mahapatra, S. P. Puppala, B. Adhikary, K. L. Shrestha, D. P. Dawadi, S. P. Paudel, and A. K. Panday, *Air quality trends of the kathmandu valley: A satellite, observation and modeling perspective,* Atmospheric Environment **201**, 334 (2019).

[4] J.-F. Ma, L. Li, H.-F. Wang, M.-Y. Tan, S.-L. Cui, Y.-Y. Zhang, Z.-P. Qu, L.-Y. Jia, and S.-H. Zhang, *Geophysical monitoring technology for co 2 sequestration,* Applied Geophysics **13**, 288 (2016).

[5] G. Fu, H.-X. Lin, A. Heemink, A. Segers, F. Prata, and S. Lu, *Satellite data assimilation to improve forecasts of volcanic ash concentrations,* Atmos. Chem. Phys. Discuss , 1 (2016).

[6] J. Jin, A. Segers, A. Heemink, M. Yoshida, W. Han, and H.-X. Lin, *Dust emission inversion using himawari-8 aods over east asia: An extreme dust event in may 2017,* Journal of Advances in Modeling Earth Systems **11**, 446 (2019).

[7] M. Rüttgers, S. Lee, S. Jeon, and D. You, *Prediction of a typhoon track using a generative adversarial network and satellite images,* Scientific reports **9**, 1 (2019).

[8] R. Trautz, T. Daley, D. Miller, M. Robertson, G. Koperna Jr, and D. Riestenberg, *Geophysical monitoring using active seismic techniques at the citronelle alabama co2 storage demonstration site,* International Journal of Greenhouse Gas Control **99**, 103084 (2020).

[9] L. Adam, K. van Wijk, T. Otheim, and M. Batzle, *Changes in elastic wave velocity and rock microstructure due to basalt-co2-water reactions,* Journal of Geophysical Research: Solid Earth **118**, 4039 (2013).

[10] A. A. Emerick and A. C. Reynolds, *History matching time-lapse seismic data using the ensemble kalman filter with multiple data assimilations,* Computational Geosciences **16**, 639 (2012).

[11] T. Mannseth and K. Fossum, *Assimilating spatially dense data for subsurface applications—balancing information and degrees of freedom,* Computational Geosciences **22**, 1323 (2018).

[12] X. Luo, T. Bhakta, G. Naevdal, *et al., Correlation-based adaptive localization with applications to ensemble-based 4d-seismic history matching,* SPE Journal **23**, 396 (2018).

[13] M. Liu and D. Grana, *Ensemble-based seismic history matching with data reparameterization using convolutional autoencoder,* in *SEG Technical Program Expanded Abstracts 2018* (Society of Exploration Geophysicists, 2018) pp. 3156–3160.

[14] O. Leeuwenburgh and R. Arts, *Distance parameterization for efficient seismic history matching with the ensemble kalman filter,* Computational Geosciences **18**, 535 (2014).

[15] Y. Zhang and O. Leeuwenburgh, *Image-oriented distance parameterization for ensemble-based seismic history matching,* Computational Geosciences **21**, 713 (2017).

[16] M. Trani, R. Arts, O. Leeuwenburgh, *et al.*, *Seismic history matching of fluid fronts using the ensemble kalman filter,* SPE Journal **18**, 159 (2012).

[17] A. Abadpour, P. Bergey, R. Piasecki, *et al.*, *4d seismic history matching with ensemble kalman filter-assimilation on hausdorff distance to saturation front,* in *SPE Reservoir Simulation Symposium* (Society of Petroleum Engineers, 2013).

[18] E. Tillier, S. Da Veiga, and R. Derfoul, *Appropriate formulation of the objective function for the history matching of seismic attributes,* Computers & Geosciences **51**, 64 (2013).

[19] 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).

[20] H. Ates, A. Bahar, S. E.-A. Salem, M. Charfeddine, *et al.*, *Ranking and upscaling of geostatistical reservoir models using streamline simulation: A field case study,* SPE Reservoir Evaluation & Engineering **8**, 22 (2005).

[21] C. Xiao, O. Leeuwenburgh, H. X. Lin, and A. Heemink, *Non-intrusive subdomain pod-tpwl for reservoir history matching,* Computational Geosciences **23**, 537 (2019).

[22] R. K. Tripathy and I. Bilionis, *Deep uq: Learning deep neural network surrogate models for high dimensional uncertainty quantification,* Journal of computational physics **375**, 565 (2018).

[23] S. W. Cheung, E. T. Chung, Y. Efendiev, E. Gildin, Y. Wang, and J. Zhang, *Deep global model reduction learning in porous media flow simulation,* Computational Geosciences **24**, 261 (2020).

[24] S. Mo, N. Zabaras, X. Shi, and J. Wu, *Deep autoregressive neural networks for high-dimensional inverse problems in groundwater contaminant source identification,* Water Resources Research **55**, 3856 (2019).

[25] Z. Zhong, A. Y. Sun, and H. Jeong, *Predicting co2 plume migration in heterogeneous formations using conditional deep convolutional generative adversarial network,* Water Resources Research **55**, 5830 (2019).

[26] Y. Zhu and N. Zabaras, *Bayesian deep convolutional encoder–decoder networks for surrogate modeling and uncertainty quantification,* Journal of Computational Physics **366**, 415 (2018).

[27] Y. J. Heo, S. J. Kim, D. Kim, K. Lee, and W. K. Chung, *Super-high-purity seed sorter using low-latency image-recognition based on deep learning,* IEEE Robotics and Automation Letters **3**, 3035 (2018).

[28] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, *et al.*, *Tensorflow: Large-scale machine learning on heterogeneous distributed systems,* arXiv preprint arXiv:1603.04467 (2016).

[29] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, *Pytorch: An imperative style, high-performance deep learning library,* in *Advances in neural information processing systems* (2019) pp. 8026–8037.

[30] M. J. Kumar, D. G. R. Kumar, and R. V. K. Reddy, *Review on image segmentation techniques,* International Journal of Scientific Research Engineering & Technology (IJSRET), ISSN , 2278 (2014).

[31] D. L. Pham, C. Xu, and J. L. Prince, *Current methods in medical image segmentation,* Annual review of biomedical engineering **2**, 315 (2000).

[32] Y. Wang, Y. Haichao, D. Gao, and J. Wang, *Image segmentation and object detection using fully convolutional neural network,* (2019), uS Patent 10,304,193.

[33] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee, *Enhanced deep residual networks for single image super-resolution,* in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (2017) pp. 136–144.

[34] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, *Residual dense network for image restoration,* IEEE Transactions on Pattern Analysis and Machine Intelligence (2020).

[35] S. Mo, N. Zabaras, X. Shi, and J. Wu, *Integration of adversarial autoencoders with residual dense convolutional networks for estimation of non-gaussian hydraulic conductivities,* Water Resources Research **56**, e2019WR026082 (2020).

[36] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, *Residual dense network for image super-resolution,* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2018) pp. 2472–2481.

[37] O. Ronneberger, P. Fischer, and T. Brox, *U-net: Convolutional networks for biomedical image segmentation,* in *International Conference on Medical image computing and computer-assisted intervention* (Springer, 2015) pp. 234–241.

[38] J. D. Matthews, J. N. Carter, K. D. Stephen, R. W. Zimmerman, A. Skorstad, T. Manzocchi, and J. A. Howell, *Assessing the effect of geological uncertainty on recovery estimates in shallow-marine reservoirs: the application of reservoir engineering to the saigup project,* Petroleum Geoscience **14**, 35 (2008).

[39] D. Obidegwu, R. Chassagne, C. MacBeth, *et al.*, *Seismic assisted history matching using binary image matching,* in *EUROPEC 2015* (Society of Petroleum Engineers, 2015).

[40] Z. L. Jin, Y. Liu, and L. J. Durlofsky, *Deep-learning-based reduced-order modeling for subsurface flow simulation,* arXiv preprint arXiv:1906.03729 (2019).

[41] M. Tang, Y. Liu, and L. J. Durlofsky, *A deep-learning-based surrogate model for data assimilation in dynamic subsurface flow problems,* Journal of Computational Physics , 109456 (2020).

[42] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, *Convolutional lstm network: A machine learning approach for precipitation nowcasting,* Advances in neural information processing systems **28**, 802 (2015).

[43] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, *Densely connected convolutional networks,* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017) pp. 4700–4708.

[44] K. He, X. Zhang, S. Ren, and J. Sun, *Identity mappings in deep residual networks,* in *European conference on computer vision* (Springer, 2016) pp. 630–645.

[45] H. X. Vo and L. J. Durlofsky, *A new differentiable parameterization based on principal component analysis for the low-dimensional representation of complex geological models,* Mathematical Geosciences **46**, 775 (2014).

[46] A. F. Rasmussen, T. H. Sandve, K. Bao, A. Lauser, J. Hove, B. Skaflestad, R. Klöfkorn, M. Blatt, A. B. Rustad, O. Sævareid, *et al.*, *The open porous media flow reservoir simulator,* Computers & Mathematics with Applications (2020).

[47] C. V. Deutsch, A. G. Journel, *et al.*, *Geostatistical software library and user's guide,* New York **119** (1992).

8

CONCLUSION AND FUTURE WORK

8.1. CONCLUSION

In this dissertation, we mainly focus on investigating two types of surrogate models, i.e., projection-based subdomain POD-TPWL and deep neural network surrogate, for efficiently estimating geological parameters in oil reservoir models. The key findings and main conclusions of this work corresponding to the five research questions described in Chapter 1, section 1.3, can be summarized as follows:

1. **Research question 1:** The commonly used reduced-order modelling aims primarily at approximating the dynamic response of the full model as accurately as possible. Can the approximated gradients estimated by non-intrusive reduced-order modelling be used to efficiently estimate geological parameters of reservoir models with sufficient accuracy?

   Geological parameters reduction and proper orthogonal decomposition techniques are able to drastically reduce the reservoir model. The use of domain decomposition further enables the large-scale applications since the number of required full model simulations depends primarily on the number of the globally reduced parameters and not on the dimension of the underlying full-order model. In addition, the use of radial-basis function interpolation eases the construction of surrogate model by avoiding a code intrusion problem. For the synthetic cases studied in Chapter 3, the adjoint model of the original high-dimensional non-linear model can be effectively replaced by the proposed subdomain reduced-order linear model for the gradient-based history matching problem. The number of full-order model simulations required for history matching is roughly 2-3 times the number of global parameter patterns in contrast to 4-5 times without applying domain decomposition. Experiment results show that our proposed subdomain POD-TPWL approach achieves similar accuracy as the classic finite-difference based history matching, but at much lower computational cost.

2. **Research question 2:** From a computational point of view, decomposing the global parameters in each subdomain individually should be very attractive. Can local pa-

rameterization by subdomain-based decomposition help to improve the computational efficiency of ROM-based parameter estimation for large-scale problems?

A local parameterization method through combining principle component analysis and domain decomposition is proposed to decompose the high-dimensional solution space for the spatial parameter field into lower-order parameter subspaces associated with individual spatial subdomains. The use of local parameterization enables the application of subdomain POD-TPWL to large-scale problems since the number of full-order model simulations depends primarily on the number of the local parameters in each subdomain. The number of full-order model runs can be limited by using a larger number of subdomains in which a smaller number of local parameters will need to be retained. For the cases studied in Chapter 4, the number of full-order model simulations required was roughly 2 times the maximum number of local parameter patterns among all subdomains. The method demonstrates a very attractive scalability for a model with a much larger domain and when using the same size of the subdomains the number of full-order model simulations will hardly increase. The performance of our proposed subdomain POD-TPWL with local parameterization is very sensitive to the domain decompositions, and we should decompose the model domain with much cares.

3. **Research question 3:** The reduced-order modelling approach is shown to be applicable for certain problems. How can alternative surrogate modelling approaches inspired by recent advances in deep learning be used to speed up the history matching process?

   We have introduced an efficient deep-learning inversion framework where the original high-fidelity model is replaced by a time-conditioning residual U-Net (cR-U-Net) surrogate model to speedup gradient-based seismic history matching. The proposed surrogate model is used to estimate the unknown permeability fields of a 3D benchmark reservoir model by assimilating synthetic measurements. The main contribution of this work is on using the DNN surrogate to approximate the objective function gradient by auto-differentiation. The inherent computational saving stems from the use of GPU units, which is an absolute essential but also a major benefit derived from the deep learning packages. The preliminary numerical results show that the proposed methodology is highly efficient since the required number of high-fidelity model runs for training the cR-U-Net surrogate model is relatively small. In addition, the proposed deep-learning inversion algorithm with stochastic gradient optimizer demonstrates an effective convergence performance against the model and data noise for the considered high-dimensional seismic history matching problem with a large number of measurements and parameters.

4. **Research question 4:** Since both deep-learning surrogate model and reduced-order model are intentionally used for the gradient-based history matching, what are the relative benefits and disadvantages of deep-learning and reduced order surrogate modelling approaches for gradient-based history matching?

   Both methods have their own advantages and disadvantages. The most important step for successfully implementing subdomain POD-TPWL method is to determine the optimal domain decomposition scheme. Although our proposed adaptive strategy could partially circumvent the negative impacts of inappropriate domain de-

composition schemes, the final solution is completely restricted by our predefined dictionary of domain decompositions, and therefore the performance of this adaptive strategy strongly depends on the design of this dictionary. The choice of a DNN architecture is relatively subjective (although flexible). The applications of cR-U-Net have the underlying advantage to make full use of the high-performance computing units, such as the GPU cards, since the exiting deep learning packages can efficiently use them. The ability of a software program to fully utilize GPUs is a big advantage.

Subdomain POD-TPWL constructs reduced-order linear model around a specific trajectory using the first-order Taylor expansion. If the testing model is far away from this trajectory, the accuracy inevitably will deteriorate. This leads to the necessity in frequently reconstructing the subdomain POD-TPWL once the new testing models have large discrepancies from the current reduced-order linear model. In contrast, the cR-U-Net actually constructs a global surrogate model based on the globally distributed training samples, which are contrasted with the construction of the linear model using local training samples, and hence retraining cR-U-Net surrogate model is not strictly required. Definitely, filling the entire parameter space as much as possible using a large number of training samples, e.g., 1000 in this dissertation, substantially improves the accuracy.

There is a trade-off between accuracy and computational complexity. For small sample sizes the physics based subdomain POD-TPWL is more efficient and that for larger sample sizes the error of subdomain POD-TPWL at some point does not improve any more for a fixed domain decomposition strategy. The DNN is less efficient for small sample sizes, but gets better and better with larger samples sizes. A direct application of the proposed DNN surrogates to field-scale models with a large degree of freedom definitely poses huge computational challenges and memory requirements for training them. This drawback seemingly makes the DNN approach less attractive than that of the subdomain POD-TPWL. In order to generate multiple posterior realizations for quantifying the model uncertainty, the subdomain POD-TPWL has to be independently implemented for each ensemble member, which requires us to repeatedly update the reduced-order models with several additional high-fidelity model simulations. By contrast, the cR-U-Net surrogate model obtains multiple posterior realizations through starting from different initial models, which does not involves additional high-fidelity model runs. The computational saving will increase linearly with the ensemble size. Overall, the cR-U-Net will become much more efficient than that of subdomain POD-TPWL for generating an ensemble of posterior realizations.

5. **Research question 5:** As an alternative to gradient-based seismic history matching, how can ensemble-based history matching approaches benefit from the use of deep-learning surrogate models, especially for the challenging application of large-scale seismic history matching?

   We present an efficient stochastic inversion framework where a deep convolutional neural network is designed to fully replace the high-fidelity model, and then is integrated with an image-oriented distance parameterization to deal with the challenging ensemble-based seismic history matching problem. An architecture of time-conditioning Residual-In-Residual Dense Block U-Net (cRRDB-U-Net) consisting of encoding, transition and decoding units, is trained to predict the spatially discon-

tinuous water fronts. The proposed deep-learning based surrogate model is used to calibrate the unknown permeability fields of a small 2D synthetic non-Gaussian model and of a modified version of 3D benchmark model. For the first case, our methodology accurately reconstructs the geological structures, e.g., binary channels, of the non-Gaussian models and shows comparable results with the high-fidelity model. For the second case, training cRRDB-U-Net surrogate model with only 1000 high-fidelity models generates satisfactory results. Since the GPU time for running cRRDB-U-Net surrogate model is negligible compared to that for running high-fidelity model, the total number of high-fidelity model simulations is taken as an indicator of the computational cost. In order to obtain comparable history matching accuracy, the use of cRRDB-U-Net as a surrogate model specifically reduces the computational times to by a factor 5 for the problem defined in Chapter 7. We should point out that the overall computational saving is case-dependent and the data generation and training time required to create a surrogate model seems to be the major limitation of the method, especially for the large-scale models.

8.2. FUTURE WORK

There are some other aspects of the proposed methodology that could possibly be improved. In the following, a few potential research directions are listed for future investigation:

1. It has been observed that sampling strategy has to be chosen with care to obtain an efficient implementation of the projection-based subdomain POD-TPWL. Some diagnostics could possibly be devised to determine if and how many additional sampling points need to be generated. All examples have shown that the choice of domain decomposition strategy has a significant impact on the history matching results. Although an adaptive strategy has been proposed, it also may be beneficial to design a prior dictionary of domain decomposition schemes based on information about the either the scales of variability of the parameter field or of the main dynamical patterns. Since in reservoir applications these patterns are strongly affected by the placement of producers and injectors the subdomain decomposition could possibly be designed using the well lay-out. We have chosen somewhat arbitrary decomposition of the global domain into rectangular subdomains. In this thesis the distribution of the measurements was almost uniform across all subdomains, and hence an uniform choice of the number of local PCA patterns in each subdomain is appropriate. To improve efficiency of the approach, the number of local parameters in each subdomain could possibly also be determined by taking into account the number of available measurements.

2. Although some preliminary results have been obtained, it deserves to improve our insights into how the DNN surrogate works for gradient-based inversion, e.g., when it will work (weak- or strong non-linear) and the relation between accuracy and computational complexity. The degree of model non-linearity dominates the convergence performance, e.g., local or global minima, of the gradient-based optimization. Nevertheless, whether the non-linearity of the high-fidelity model is correctly captured by the DNN surrogate model is not very clear. In contrast, derivative-free global optimization methods, e.g., particle-swarm optimization, genetic algorithm

and evolutionary algorithm, could search solutions in the global space at the expense of numerous model runs. The high computational-efficiency of our proposed deep-learning models definitely could facilitate the application of these global optimization methods.

3. The currently proposed DNN surrogate model is built on a set of detailed full physics simulations at the training stage, which is completely independent on the history matching process. The history matching has the potential of instructing the construction of DNN surrogate model, during which a set of locally accurate surrogates are adaptively refined. The surrogate model could be trained with a small number of initial training samples, and then be progressively enriched with many additional training samples (i.e., re-sampled simulation runs) close to the target distribution at the history matching process. Under this situation, the DNN surrogate can perhaps be retrained relatively quickly if we only add one or several new data points and just continue from the previously trained model. In order to prevent the model from overfitting to these new training points, the fine-tuning or transfer learning strategy should be useful [1, 2].

4. Incorporating the deep-learning surrogate model in history matching procedure inevitably introduces additional sources of uncertainty due to approximation errors [3]. Investigating the influences of surrogate approximation errors on the history matching process could contribute to achieve more accurate and reliable results [4]. In the current work, the proposed methods were tested on synthetic or simplified benchmark problems. Their potential use in practical applications and other complex systems deserves further exploration due to their data-driven and non-intrusive nature.

5. The complementary advantage of projection-based ROM and DNN opens up another opportunity of research and therefore deserves to be explored. A direct application of DNN surrogates to real-world models with a large number of degrees of freedom poses huge computational challenges for training them. We may expect that a smart combination of ROM and DNN has the potential of enlarging the validity of ROM and hence to improve the accuracy through a DNN-based error correction. In particular, based on our preliminary results in [5] we can substantially improve the accuracy of ROM by quantifying the discrepancy between the high-fidelity model and the ROM outputs using a DNN. Or we can project the high-dimensional model into the reduced-subspace as commonly done in ROM and then construct DNN surrogate in the subspace, which results in a large reduction of computational cost and memory requirement [6–8].

REFERENCES

[1] S. De, J. Britton, M. Reynolds, R. Skinner, K. Jansen, and A. Doostan, *On transfer learning of neural networks using bi-fidelity data for uncertainty propagation,* arXiv preprint arXiv:2002.04495 (2020).

[2] L. Yan and T. Zhou, *An adaptive surrogate modeling based on deep neural networks for large-scale bayesian inverse problems,* arXiv preprint arXiv:1911.08926 (2019).

[3] J. Zhang, Q. Zheng, D. Chen, L. Wu, and L. Zeng, *Surrogate-based bayesian inverse modeling of the hydrological system: An adaptive approach considering surrogate approximation error,* Water Resources Research **56**, e2019WR025721 (2020).

[4] T. Butler and T. Wildey, *Utilizing adjoint-based error estimates for surrogate models to accurately predict probabilities of events,* International Journal for Uncertainty Quantification **8** (2018).

[5] J. Huang, *Machine learning based error modeling for surrogate model in oil reservoir problem,* (2019).

[6] A. T. Mohan and D. V. Gaitonde, *A deep learning based approach to reduced order modeling for turbulent flow control using lstm neural networks,* arXiv preprint arXiv:1804.09269 (2018).

[7] M. Wang, S. W. Cheung, W. T. Leung, E. T. Chung, Y. Efendiev, and M. Wheeler, *Reduced-order deep learning for flow dynamics. the interplay between deep learning and model reduction,* Journal of Computational Physics **401**, 108939 (2020).

[8] P. Wu, J. Sun, X. Chang, W. Zhang, R. Arcucci, Y. Guo, and C. C. Pain, *Data-driven reduced order model with temporal convolutional neural network,* Computer Methods in Applied Mechanics and Engineering **360**, 112766 (2020).

ACKNOWLEDGEMENT

This PhD project was fully completed at Delft University of Technology, the Netherlands. I sincerely thank the China Scholarship Council (CSC) for providing research funding. The achievements of PhD dissertation result from support and cooperation from many people.

To begin with, I would like to gratefully acknowledge my promotors Prof. Arnold Heemink and Prof. Hai-Xiang Lin. At the first year of my PhD study, everything was almost new to me and hence I was not familiar with my project and did not know how to independently manage my research progress in the Netherlands. Thank both of you for giving me full freedom to explore anything that I am curious about. During this self-studying process, I have learnt how to independently build up knowledge and come up with new research ideas efficiently. They also provided me opportunities to share my research progress on various conferences and to exchange ideas with other researchers. Besides, I am very grateful to Arnold and Hai-Xiang for their long-standing patience to my personal language problem. I can image that sometimes they did not easily understand what I have said, however they still kept smiling and encourage me to speak slowly, which give me great courage. Their understanding has always supported me in completing my PhD thesis smoothly.

During the past four years, I am always feeling so happy and comfortable to cooperate with Dr.Olwijn Leeuwenburgh. He always kept patient to supervise and assist me resolve relevant technical problems. Thank you for teaching me how to use the OPM-$flow$ reservoir simulator and providing me the benchmark SAIGUP case data which has been fully used in my oil reservoir data assimilation project. Especially, you invite me to visit TNO regularly, which supports me a nice experience and opportunity to discuss with the oil industry engineers. It helps me to build up my connections with experts on both modeling, measurements and assimilation fields. I am very grateful for these experiences, which will help much to my future research.

I would like to give my most sincere thanks to my group colleagues. Evelyn Sharabi and Dorothée Engering always try their best to help me when I encounter any life and study problems. I am a totally new user for the Linux system. Whenever I get stuck with some technique issues in my computer, Kees Lemmens was always very patient to help me resolve the problem. His sufficinet experience in the field of computer programming made me so impressive. I also want to thanks our data-assimialtion learning group. The regular Friday Data Assimilation meeting resembles us together to discuss and share some new ideas in the community of data assimilation. I feel so happy to continuously learn new knowledge with all of you. I also want to thank my friends Guangliang Fu, Sha Lu, Jianbin Jin, Senlei Wang, Xiaohui Wang, Yuan Chen, Xiao Deng, Tuo Deng, Xiao Li, Jie Liu, Xiuxiu Zhan, Bowen Fan, Haopeng Wang, Xuan Tao et al. The life time I spent with everyone is always so beautiful, and the mutual encouragement made me no longer lonely on the way to pursue my research dream. I also want to thank Dr.Chengquan Men and my friend Yongjiu Wei who accept my requirements to be the guarantors when I apply for the financial support from China Scholarship Council.

I want to give my most sincere thanks to my girlfriend, Mingjin Liu. We have been together for almost ten years. Thank you for always giving me unlimited support during my PhD study in Netherlands. When I hesitate to apply CSC program, you encourage me to pursue my research dream. You always company with me whenever I feel depressed due to the disappointed results from my PhD project. Without your everlasting care, unconditional support and enormous love, this would have never been possible. Thank you for all the sacrifices you made without any complaints or hesitation! Thank you for always being there when I needed you the most. Our long distance relationship were a most welcome distraction from the daily hassle of life and always made me realize what is truly important in life. You are my wonder and I hope that we can be together for the rest of our life.

Finally, I would also like to thank my family for their unlimited support and understanding during my PhD study. They have always been together with me and give me infinite encouragements.

Cong Xiao

Delft, September 2020

LIST OF PUBLICATIONS

JOURNALS

4. **Xiao, C.**, Leeuwenburgh, O., Lin, H.X. and Heemink, A., 2020. *Surrogate-Assisted Gradient-Based Seismic History Matching: Comparative Study between Projection-Based Reduced-Order Modeling and Deep Neural Network.* Journal of Petroleum Science and Engineering. (Revised)

3. **Xiao, C.**, Lin, H.X., Leeuwenburgh, O. and Heemink, A., 2020. *Conditioning of Deep-Learning Surrogate Models to Image Data with Application to Reservoir Characterization.* Knowledge-Based Systems. (Revised)

2. **Xiao, C.**, Leeuwenburgh, O., Lin, H.X. and Heemink, A., 2021. *Efficient estimation of space varying parameters in numerical models using non-intrusive subdomain reduced order modeling.* Journal of Computational Physics, 424, p.109867.

1. **Xiao, C.**, Leeuwenburgh, O., Lin, H.X. and Heemink, A., 2019. *Non-intrusive subdomain POD-TPWL for reservoir history matching.* Computational Geosciences, 23(3), pp.537-565.

CONFERENCES

4. **Xiao, C.**, Heemink, A., Lin, H.X. and Leeuwenburgh, O., *Efficient Deep-Learning Inversion for Big-Data Assimilation: Application to Seismic History Matching*, Presented at ECMOR XVII, Edinburgh, United Kingdom, 14-17 September, 2020.

3. **Xiao, C.**, Heemink, A., Lin, H.X. and Leeuwenburgh, O., *Subdomain Reduced-Order Modelling with Smooth Local Parameterization for Large-Scale Inversion Problem*, Presented at ENUMATH 2019 conference, The Netherlands, 30 September - 4 October, 2019.

2. **Xiao, C.**, Heemink, A., Lin, H.X. and Leeuwenburgh, O., *A machine-learning Based Subdomain POD-TPWL for Large-Scale Inversion Problems*, Presented at InterPore2019, Valencia, Spain, 6 -10 May, 2019.

1. **Xiao, C.**, Heemink, A., Lin, H.X. and Leeuwenburgh, O., *Subdomain Adjoint-Based Variational Data Assimilation for Reservoir History Matching*, Presented at 13th International EnKF Workshop. Bergen, Norway, 28 - 30, May, 2018.

SIDE WORKS

1. **Xiao, C.**, Tian, L., Zhang, L., Wang, G. and Deng, Y., 2020. *Distributed Gauss-Newton Optimization with Smooth Local Parameterization for Large-Scale History-Matching Problems.* SPE Journal, 25(01), pp.56-80.