

"Process-based machine learning for multi-physics and chemistry
in geothermal applications: Parameter estimation and
optimization under uncertainties"

Von der Fakultät für Georessourcen und Materialtechnik der
Rheinisch-Westfälischen Technischen Hochschule Aachen

zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften

genehmigte Dissertation

vorgelegt von

Ryan Kurniawan Santoso, M. Sc.

Berichter: Univ.-Prof. Florian Wellmann, Ph. D.
Prof. Dr. Martin O. Saar
Univ.-Prof. Dr. sc. Julia Kowalski
Dr. Nikolaos Prasianakis

Tag der mündlichen Prüfung: 21.11.2024

RWTH AACHEN UNIVERSITY

&

ETH ZURICH

DOCTORAL THESIS

**Process-based machine learning for
multi-physics and chemistry in
geothermal applications: Parameter
estimation and optimization under
uncertainties**

Author:

Ryan Santoso, M.Sc.
(from Indonesia)

Supervisors:

Prof. Florian Wellmann, Ph.D.
Prof. Martin O. Saar, Ph.D.

Examination date:

21.11.2024

Examiners:

Prof. Dr. Julia Kowalski
Dr. Nikolaos Prasianakis

*A thesis submitted in fulfillment of the requirements
for the Joint Doctoral degree*

in the

Faculty of Georesources and Materials Engineering, RWTH Aachen University

&

Department of Earth and Planetary Sciences (D-EAPS), ETH Zurich

500000

Fakultät für Georessourcen und Materialtechnik

S. Frenzel-Gumlich

Intzestr. 1

52056 Aachen

Fax: 0241-80-92370

E-Mail: promotionen@fb5.rwth-aachen.de

Eidesstattliche Erklärung

1.)

Hiermit versichere ich eidesstattlich, dass ich die Dissertation selbstständig verfasst und alle in Anspruch genommenen Hilfen in der Dissertation angegeben habe. ☒

Unterschrift

Ort, Datum

2.) Sofern die Dissertation in einer Einrichtung außerhalb der RWTH entstanden ist:

Hiermit erkläre ich, dass mit Veröffentlichung der Dissertation keine bestehenden Betriebsgeheimnisse verletzt werden. ☐

Unterschrift

Ort, Datum

*“To my parents, with their unwavering support through my endeavors,
Agus and Koestantie.”*

“The doctoral research has been carried out in the context of an agreement on joint doctoral supervision between RWTH Aachen, Germany, and ETH Zurich, Switzerland.”

RWTH AACHEN UNIVERSITY

&

ETH ZURICH

Kurzzusammenfassung

Fakultät für Georessourcen und Materialtechnik, RWTH Aachen

&

Departement der Erd- und Planetenwissenschaften (D-EAPS), ETH Zurich

Joint Doctoral degree

Prozessbasiertes maschinelles Lernen für multiphysikalische und chemische Prozesse in geothermischen Anwendungen: Parameterstudien und Optimierung unter Berücksichtigung von Unsicherheiten

von Ryan Santoso

Geothermale Energie, die Wärme aus der Erde, bietet eine saubere und nachhaltige Alternative, um sich von fossilbrennstoffbasierter Energie abzuwenden und dem Klimawandel entgegenzuwirken. Da diese Energiequelle aus dem Untergrund stammt und nur wenige Messdaten für ihre Charakterisierung verfügbar sind, muss bei der Gestaltung eines effizienten und sicheren Extraktionsprozesses auf numerische Simulationen zurückgegriffen werden. Diese numerischen Simulationen sollten auch Unsicherheiten adressieren, die durch ungenaue Charakterisierung aufgrund knapper Messdaten entstehen, um Robustheit und Zuverlässigkeit zu gewährleisten. Dies stellt eine bedeutende rechnerische Herausforderung dar, die zahlreiche Modellläufe erfordert, um Inversions- und Optimierungsprobleme zu lösen. In dieser Dissertation stellen wir eine effiziente Methode vor, um diese inversen und Optimierungsprobleme unter einem multiphysikalischen und chemischen Rahmen zu ermöglichen, was mit den derzeitigen modernsten numerischen Methoden wie der Finite-Elemente- und Finite-Volumen-Methode nicht machbar ist.

In letzter Zeit wurde Künstliche Intelligenz und Maschinelles Lernen wie Neuronale Netze häufig eingesetzt, um numerische Simulationen zur Durchführung kostspieliger inverser und Optimierungsaufgaben in geothermischen Anwendungen zu beschleunigen. Ein Modell mit Künstlicher Intelligenz stellt ein numerisches Modell dar, das es eine Beziehung zwischen Ein- und Ausgabe auf datengesteuerte Weise lernt. Da datengesteuerte maschinelle Lernmodelle jedoch nicht auf Physik- und Chemiegesetzen basieren, können sie oft ungenaue Vorhersagen liefern. Folglich erfordert die Erzielung höherer Vorhersagegenauigkeit eine große Anzahl von Trainingsdaten, was zu einer umfangreichen Generierung von Trainingsdaten führt. Daher ist es entscheidend, das Wissen über physikalische und chemische Prozesse in den Aufbau von maschinellen Lernmodellen zu integrieren, insbesondere durch prozessbasierte maschinelle Lernmethoden, um genaue Vorhersagen und die Effizienz von maschinellen Lernmodellen zu gewährleisten.

Es gibt verschiedene prozessbasierte maschinelle Lernmethoden, wie die nichtinvasive reduzierte Basismethode und das physikbasierte Neuronale Netzwerk. Nicht alle prozessbasierten maschinellen Lernmethoden sind jedoch für den Umgang mit

den mehrphysikalischen und chemischen Prozessen in geothermischen Anwendungen geeignet. Anhand von zwei mehrphysikalischen Problemen, hydro-mechanischen (HM) Benchmark-Problemen, die in geothermischen Anwendungen häufig verwendet werden, vergleichen wir die nichtintrusive reduzierte Basismethode, das physikbasierte neuronale Netzwerk und datengesteuerte maschinelle Lernmethoden. Unsere Ergebnisse zeigen, dass die nichtintrusive reduzierte Basismethode die genauesten Näherungen mit der geringsten Anzahl von Trainingsdaten liefert. Die nichtintrusive reduzierte Basismethode zeigt auch eine signifikante Beschleunigung im Vergleich sowohl zum physikbasierten Neuronalen Netzwerk als auch zu datengesteuerten maschinellen Lernmethoden. Diese Erkenntnis ist entscheidend für die Bearbeitung realer geothermischer Systeme, bei denen numerische Modelle oft hochdimensional und nichtlinear sind und erhebliche Rechenressourcen für die Generierung von Trainingsdaten erfordern. Dies brachte uns dazu, die nichtintrusive reduzierte Basismethode für reale, mehrphysikalische geothermische Systeme, wie den Fall von Den Haag, weiter zu entwickeln.

Wir führen eine Bewertung des Wärmeentzugsprozesses in Den Haag, Niederlande, unter Verwendung der nichtintrusiven reduzierten Basismethode durch. Dies ermöglicht uns, eine Unsicherheitsquantifizierung effizient durchzuführen, um die Platzierung der Produktionsbohrlöcher zu untersuchen, unser Verständnis des transienten thermo-hydraulischen (TH) Prozesses in Den Haag aus der Perspektive der Entropieproduktion zu verbessern und optimale Betriebsbedingungen unter Berücksichtigung von Unsicherheiten zu entwerfen. Bemerkenswert ist, dass die nichtintrusive reduzierte Basismethode eine beeindruckende Beschleunigung um sechs bis sieben Größenordnungen im Vergleich zur Verwendung der Finite-Elemente-Methode zeigt.

Darüber hinaus markiert unsere Arbeit die erste Implementierung der nichtintrusiven reduzierten Basismethode zur Bewältigung von Chemieproblemen, wie reaktive Transportprobleme und chemische Gleichgewichtsberechnungen. Diese Probleme sind für ihre Nichtlinearität und das Vorhandensein von 'Schock'-Merkmale bekannt. Die Berücksichtigung dieser 'Schock'-Merkmale stellt eine erhebliche Herausforderung dar, wenn datengesteuerte maschinelle Lernmethoden verwendet werden, welche die bisher am häufigsten verwendeten Ansätze in der geochemischen Forschungsgemeinschaft sind. Wir zeigen, dass die nichtintrusive reduzierte Basismethode die "Schock"-Merkmale mit einer geringeren Anzahl von Trainingsdaten erfassen kann, verglichen mit datengesteuerten maschinellen Lernmethoden. Dieses Ergebnis eröffnet Möglichkeiten für zukünftige Inversions- und Optimierungsprobleme, die Chemie von echten, geologischen Fallbeispielen betreffen, welche sonst einen hohen Rechenaufwand erfordern.

RWTH AACHEN UNIVERSITY

&

ETH ZURICH

Abstract

Faculty of Georesources and Materials Engineering, RWTH Aachen University

&

Department of Earth and Planetary Sciences (D-EAPS), ETH Zurich

Joint Doctoral degree

Process-based machine learning for multi-physics and chemistry in geothermal applications: Parameter estimation and optimization under uncertainties

by Ryan Santoso

Geothermal energy, the heat from the Earth, offers a clean and sustainable alternative for transitioning away from fossil-fuel based energy and addressing climate change. Given that this energy resource is located underground and sparse measurement data are available for its characterization, designing an efficient and safe extraction process must rely on numerical simulations. The numerical simulations should also address uncertainties associated with inaccurate characterization caused by sparse measurement data, ensuring robustness and reliability. This poses a significant computational challenge, demanding numerous model runs to solve inverse and optimization problems. In this dissertation, I present an efficient method to enable these inverse and optimization problems under multi-physics and chemistry settings, which is not feasible using the current state-of-the-art numerical methods, such as finite element and finite volume methods.

Recently, machine learning methods, such as Neural Networks, have been widely employed to accelerate numerical simulations for performing costly inverse and optimization tasks in geothermal applications. A machine learning model represents a numerical model by learning its input-output relationship, a data-driven approach. However, since data-driven machine learning models are not inherently based on physics and chemistry laws, they can often produce inaccurate predictions. Consequently, achieving higher prediction accuracy requires a large number of training samples, leading to exhaustive sample generation. Therefore, integrating the knowledge of physical and chemical processes into the construction of machine learning models, namely process-based machine learning methods, is essential to ensure accurate predictions and the efficiency of machine learning models.

There are several process-based machine learning methods, such as the non-intrusive reduced-basis method and the Physics-Informed Neural Network. However, not all process-based machine learning methods are suitable for dealing with the multi-physics and chemistry settings in geothermal applications. Using two multi-physics problems, hydro-mechanical (HM) benchmark problems used in geothermal applications, I compare the non-intrusive reduced-basis method, the Physics-Informed Neural Network, and data-driven machine learning methods. The results indicate that the non-intrusive reduced-basis method provides the most accurate approximations with the smallest number of training samples. The non-intrusive reduced-basis method

also demonstrates a more substantial speedup compared to both the Physics-Informed Neural Network and data-driven machine learning methods. This finding is crucial for addressing real-world geothermal cases, where numerical models are often high-dimensional and non-linear, requiring significant computational resources for generating training samples. This convinced me to implement the non-intrusive reduced-basis method for real-world, multi-physics geothermal cases, such as the The Hague case.

I conduct an assessment of the heat extraction process in The Hague, Netherlands, utilizing the non-intrusive reduced-basis method. This allows me to efficiently perform uncertainty quantification to address producer placement, enhancing the understanding of the transient thermo-hydraulic (TH) process in The Hague from the perspective of entropy production, and designing optimum operating conditions while considering uncertainties and safety. Notably, the non-intrusive reduced-basis method demonstrates an impressive six to seven orders-of-magnitude speedup compared to the use of the finite element method.

Furthermore, this work marks the first implementation of the non-intrusive reduced-basis method for addressing chemistry (C) problems, such as reactive transport problems and chemical equilibrium calculations. These problems are known for their non linearity and the presence of 'shock' features. Capturing these 'shock' features poses a significant challenge when utilizing data-driven machine learning methods, which are, up to now, the most commonly used approaches within the geochemistry communities. I show that the non-intrusive reduced-basis method can capture the "shock" features with a smaller number of training samples, compared to the data-driven machine learning methods. This result opens up possibilities for future inverse and optimization involving chemistry in real-world geothermal cases, which are computationally expensive.

Acknowledgements

The completion of this dissertation has been made possible through both direct and indirect support from many people. My profound gratitude towards them cannot be justly reflected in this note.

I am very grateful to my advisors, Prof. Florian Wellmann and Prof. Martin Saar, for their guidance and support during my doctoral studies. I particularly appreciate the fruitful and motivating discussions. I am also thankful for the opportunities they provided, allowing me to participate in numerous conferences and workshops. These experiences have significantly contributed to strengthening my knowledge in model order reduction, geothermal, and reactive transport.

I extend my special thanks to Dr. Denise Degen and Dr. Allan Leal for their unwavering day-to-day support and countless discussions throughout my doctoral studies. Their encouragement to enhance my knowledge has led to a profound understanding of every aspect of my dissertation. In particular, I am grateful for their dedication to spending time with me at least once a week to answer my confusions, providing invaluable inspiration for solving my problems.

I also want to thank Dr. Renate Pechinig and Dominique Knapp for their support during my stay in Geophysica. They provide the data of The Hague, Netherlands which becomes the center of this dissertation. A special thanks goes to Dr. Nora Koltzer for her invaluable guidance during my stay at Fraunhofer IEG, particularly for her assistance in understanding the coaxial borehole heat exchanger. Additionally, I extend my thanks to Dr. Mauro Cacace for our collaborative work and his support in troubleshooting for GOLEM.

Special thanks to the CG3 group members—Kaifeng Gao, Nils Chudalla, Elisa Heim, Jian Yang, Carlos Colombo, Jan von Harten, Dr. Christin Bobe—and the GEG group members—Kevin Hau, Serhat Kucuk, Anna Kotsova, Nicolas Jurado, Dr. Powei Huang, Dr. Tsubasa Onishi, Dr. Anna Jentsch, Dr. Xiangzhao Kong—for their companionship, good humor in many stressful days, and inspiring scientific coffee time.

My experience in RWTH Aachen University and ETH Zurich would not have been so enjoyable without so many good friends. I especially thank for Sofia Brisson, Emilie Lemaire, Deyan Tian, and Dima Shamsedine for our friendship and emotional support during my doctoral studies and Lily Suherlina for her support during my stay in Zurich.

I also acknowledge EASYGO program, funded by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 956965. This program has been instrumental in facilitating the attainment of a double PhD degree from both RWTH Aachen University and ETH Zurich. Special thanks go to Dr. Maren Brehme and Dr. Paromita Deb for their leadership in this program and EASYGO PhD comrades for their support during my doctoral studies. Additionally, I express my gratitude to the IRTG MIP program for providing opportunities to participate in various beneficial trainings and seminars.

Contents

Kurzzusammenfassung	vii
Abstract	ix
Acknowledgements	xi
1 Introduction	1
1.1 Motivation	1
1.2 Machine learning meets small data	3
1.3 Understanding a complex system under uncertainties	5
1.4 Aligning models and data remains challenging	5
1.5 Finding an optimum under uncertainties	6
1.6 Scientific aims	6
1.7 Structure of the dissertation	9
1.8 Publications and conferences	10
2 Materials and methods	13
2.1 Non-intrusive reduced-basis method	13
2.1.1 Generation of basis function	16
2.1.2 Calculation of reduced-basis coefficients	16
2.2 Parameter estimation studies	19
2.2.1 Sensitivity analysis	19
2.2.2 Uncertainty quantification	22
2.3 Optimization under uncertainties studies	22
3 Formulation of process-based machine learning	25
3.1 Surrogate modeling using machine learning	25
3.1.1 Data-driven machine learning methods	25
Neural Networks (NN)	26
3.1.2 Process-based machine learning methods	27
Physics-Informed Neural Network (PINN)	28
3.2 Comparison strategy	29
3.3 Benchmarks	31
3.3.1 Mandel's problem	31
3.3.2 Oedometer problem	33
3.4 Results	36
3.4.1 Parameter-estimation setting	37
3.4.2 State-estimation setting	39
3.4.3 Offline and online time	40
3.5 Discussions	41

4	Application of NI-RB method for well targeting	45
4.1	The Hague, Netherlands	45
4.1.1	Regional geological model	45
4.1.2	Reservoir model	50
4.2	Steady-state thermal process	52
4.3	Workflow	53
4.4	Results	55
4.4.1	Surrogate model	55
4.4.2	Global sensitivity analysis	56
4.4.3	Uncertainty quantification	59
4.5	Discussions	68
5	Application of NI-RB method for optimizing operating conditions	71
5.1	Transient thermo-hydraulic problem	71
5.1.1	Synthetic model setup	72
5.1.2	The Hague model setup	74
5.2	Optimization formulation	78
5.2.1	Optimization for the synthetic case	79
	Scenario 1	80
	Scenario 2	80
	Scenario 3	81
	Scenario 4	81
5.2.2	Optimization for the The Hague case	82
	Mean-properties formulation	83
	Mean-responses formulation	83
	Reliability-Based Robust Design Optimization formulation	83
5.3	Workflow	84
5.4	Results	85
5.4.1	Synthetic case result	86
	Surrogate model	86
	Global sensitivity analysis	87
	Optimization	88
5.4.2	The Hague case result	91
	Surrogate model	92
	Global sensitivity analysis	93
	Optimization	96
5.5	Discussions	97
6	Entropy production for global sensitivity analysis	101
6.1	Transient thermo-hydraulic problem	101
6.2	Entropy production	102
6.3	Variations for the global sensitivity analyses	103
6.4	Results	104
6.4.1	Performances of surrogate models	104
6.4.2	The calculation of entropy production	105
6.4.3	Influence of different objective functions before injection period	106
6.4.4	Influence of different objective functions during injection period	108
6.5	Discussions	111

7	Application of NI-RB method for reactive transport simulations	113
7.1	Reactive transport problems	113
7.2	Case study	114
7.3	Results	117
7.4	Discussion	127
8	Application of NI-RB method for chemical equilibrium calculations	129
8.1	Chemical equilibrium formulations	129
8.2	Non-intrusive reduced-basis method formulation	132
8.2.1	Generation of basis functions	132
8.3	Comparison strategy	133
8.4	Case study	134
8.4.1	Proposed sampling strategy	136
8.5	Results	137
8.5.1	Impact of sampling strategy	138
8.5.2	Impact of number of training samples	141
8.5.3	Offline and online time	141
8.6	Discussion	144
9	Closure	147
9.1	Conclusions	147
9.2	Potential future extensions	151
A	fast: A Python library for the NI-RB implementation	xxi
A.1	Structure of the library	xxi
A.2	Installations	xxiii
A.3	Functions	xxiii
A.4	Example	xxvi
	Bibliography	xxix

List of Figures

1.1	Interplay between the amount of data and the amount of physics/-chemistry knowledge for the implementation of process-based modeling, data-driven machine learning, and process-based machine learning. This figure is adopted from Karniadakis et al. (2021) and modified.	3
1.2	Structure of this dissertation.	9
2.1	Construction of the NI-RB model where basis functions are obtained with the POD approach before performing machine learning (in this case, a Neural Network model) training. The time-independent formulation is in (a), where the basis functions are generated from a steady-state thermal simulation on the regional model of The Hague, and the time-dependent formulation is in (b), where the basis functions are generated from a transient thermo-hydraulic simulation on the reservoir model of The Hague.	15
3.1	A data-driven NN model, constructed by minimizing a data loss function.	26
3.2	A PINN model, constructed by minimizing both data loss and physics loss functions.	28
3.3	Comparison strategy for different process-based and data-driven machine learning methods: (a) parameter estimation setting and (b) state estimation setting.	30
3.4	Computational domain and boundary conditions of the Mandel's problem. L_x is the maximum extent in x-direction, L_y is the maximum extent in y-direction.	32
3.5	Computational domain and boundary conditions of the Oedometer problem. L_x is the maximum extent in x-direction, L_y is the maximum extent in y-direction, and L_z is the maximum extent in z-direction. For the flow component, all faces have no-flow boundary condition towards its normal vector direction: $\frac{\partial P_f}{\partial x} = 0$ at left and right, $\frac{\partial P_f}{\partial y} = 0$ at front and back, and $\frac{\partial P_f}{\partial z} = 0$ at top and bottom. For the solid component, all faces have Dirichlet-type boundary condition (set to be zero) at their associated directions (except the right face): $u_x = 0$ at left, $u_y = 0$ at front and back, and $u_z = 0$ at top and bottom. The negative value at right boundary denotes compression.	35
3.6	All comparisons among different data-driven machine learning methods and process-based machine learning methods under multiple settings for both Mandel's and Oedometer problem. The number inside the box denotes the number of training samples used for each method used in the comparison.	36

3.7	Comparison among surrogate models constructed by the NN-P and the NI-RB in terms of (a) $u_x(t)$, (b) $P_f(x = 0\ m, t)$, (c) relative error for $u_x(t)$, and (d) relative error for $P_f(x = 0\ m, t)$, utilizing Mandel's problem. The dashed lines "--" denote the NN-P predictions, and the solid lines denote the NI-RB predictions.	38
3.8	Comparison among surrogate models constructed by the NN-P and the NI-RB in terms of (a) $P_f(t)$ and (b) the relative error, utilizing the Oedometer problem. The dashed lines "--" denote the NN-P predictions, and the solid lines denote the NI-RB predictions.	38
3.9	Comparison among surrogate models constructed by the NN-S and the PINN in terms of (a) $u_x(t)$, (b) $P_f(x = 0\ m, t)$, (c) relative error for $u_x(t)$, and (d) relative error for $P_f(x = 0\ m, t)$, utilizing Mandel's problem. The dashed lines "--" denote the NN-S predictions and the solid lines denote the PINN predictions.	39
3.10	Comparison among surrogate models constructed by the NN-S, the PINN, the NN-P, and the NI-RB in terms of (a) $u_x(t)$, (b) $P_f(x = 0\ m, t)$, (c) relative error for $u_x(t)$, and (d) relative error for $P_f(x = 0\ m, t)$, utilizing the whole FE nodal solutions of Mandel's problem as training samples. The ones with orange and yellow color are constructed for parameter-estimation setting with the following linestyle nomenclatures: the dashed lines "--" denote the NN-P predictions and the solid lines denote the NI-RB predictions.	40
3.11	Offline and online time for different surrogate models utilizing (a) Mandel's problem and (b) the Oedometer problem.	41
3.12	The impact of different weights towards the relative error of (a) $u_x(t)$ and (b) $P_f(x = 0\ m, t)$, utilizing the Mandel's problem. w_p denotes the weights for the physics loss in the loss function; PINN-A denotes the use of PINN with an adaptive weight's algorithm (?).	43
4.1	The case study of The Hague, Netherlands. (a) shows the location of The Hague region in West Netherlands Basin, marked with a red dot. The figure is adopted from Willems et al. (2020) and modified by adding the location of The Hague. (b) shows my regional geological model coverage and geological layer description. The purple star denotes the proposed drilling location and the black dots are well data used for constructing the regional geological model. The figure is adopted from Mottaghy, Pechinig, and Vogt (2011) and modified. In the model, the positive x-axis corresponds to the east direction, and the positive y-axis corresponds to the north direction.	46
4.2	Measurement data within the regional geological model. (a) shows the acquisition years, (b) shows the location of measurement points in xy-axis, (c) shows the location of measurement points in xz-axis, and (d) shows the measurement data for each well at different depth and their values. The well names associated with different well numbers are in Table 4.2.	49
4.3	The reservoir model of The Hague, Netherlands. The image illustrates the layout of the reservoir model within the regional model and a detailed view of the reservoir model itself. The box with the purple line is the reservoir model location. The image is adopted from Mottaghy, Pechinig, and Vogt (2011) and modified.	51

4.4	Workflow for conducting uncertainty quantification of The Hague, Netherlands. This workflow is based on the works of Degen, Veroy, and Wellmann (2022) and Degen et al. (2021a).	53
4.5	Offline and online time of the surrogate model for The Hague regional model. The value 2.9×10^7 denotes the computational speed-up achieved by using the NI-RB surrogate model relative to the FEM simulation. . .	56
4.6	Scenarios used for global sensitivity analysis in The Hague case study. .	56
4.7	Global sensitivity analysis result using The Hague regional model: (a) using sum-of-temperature objective function, (b) using no-weight scenario in measurement-focused function, (c) using expert-weight scenario in measurement-focused function, and (d) using cluster-weight scenario in measurement-focused function.	58
4.8	Uncertainty quantification results: (a), (c), (e), (g), (i), (k), (m) consecutively are posterior of $\lambda_N, \lambda_{CK}, \lambda_{KN}, \lambda_S, \lambda_{AT}, \lambda_{RB}, Q$ obtained with different scenarios; (b), (d), (f), (h), (j), (l), (n) consecutively are q-q plot of $\lambda_N, \lambda_{CK}, \lambda_{KN}, \lambda_S, \lambda_{AT}, \lambda_{RB}, Q$ obtained with different scenarios. Note that, in the no-weight scenario, λ_{RB} is not significant, hence, it does not appear in (k) and (l).	61
4.9	Convergence criteria for each parameter involved in uncertainty quantification: (a), (d), (g), (j), (m), (p), (s) are Geweke criterion consecutively applied on $\lambda_N, \lambda_{CK}, \lambda_{KN}, \lambda_S, \lambda_{AT}, \lambda_{RB}, Q$ obtained with different scenarios; (b), (e), (h), (k), (n), (q), (t) consecutively are autocorrelation of $\lambda_N, \lambda_{CK}, \lambda_{KN}, \lambda_S, \lambda_{AT}, \lambda_{RB}, Q$ obtained with different scenarios; (c), (f), (i), (l), (o), (r), (u) consecutively are traces of $\lambda_N, \lambda_{CK}, \lambda_{KN}, \lambda_S, \lambda_{AT}, \lambda_{RB}, Q$ obtained with different scenarios. Note that, in the no-weight scenario, λ_{RB} is not significant, hence, it does not appear in (p), (q), and (r).	65
4.10	Uncertainty quantification results for The Hague case study: (a), (b), (c), and (d) are the 0.05 quantile; (e), (f), (g), and (h) are the mean; (i), (j), (k), and (l) are the 0.95 quantile. (a), (e), and (i) use uncalibrated scenario; (b), (f), and (j) use no-weight scenario; (c), (g), and (k) use expert-weight scenario; (d), (h), and (l) use cluster-weight scenario.	67
5.1	Numerical simulation setup for optimizing operating conditions of a synthetic case.	72
5.2	The spatio-temporal solution of the thermo-hydraulic simulation on the synthetic case: (a) temperature and (b) pressure. The red color denotes responses obtained using the lower bound values, blue color the mean values, and green color the upper bound values.	73
5.3	Numerical simulation setup for optimizing operating conditions in The Hague, Netherlands.	76
5.4	The variation in (a) temperature at producer from the lower bound to upper bound values and (b) pressure at injector and producer from the lower bound to upper bound values. The dashed lines in (b) denote pressure at injector, while the solid lines pressure at producer.	77
5.5	Different optimization formulations for synthetic case.	80
5.6	Workflow for conducting optimization under uncertainties. This workflow is inspired by the works of Degen, Veroy, and Wellmann (2022) and Degen et al. (2021a).	85

5.7	Global sensitivity analysis results utilizing synthetic case with (a) total depletion time, (b) pressure at the left side at $t = 1.0$ days, (c) pressure at the left side at $t = 5.0$ days, and (d) pressure at the left side at $t = 10.0$ days.	87
5.8	Optimization results with the synthetic case using (a) scenario 1, (b) scenario 2, (c) scenario 3, (d) scenario 4, and (e) scenario 5. In (d) and (e), the blue dots denote the objective function and the orange dots the safety constraint.	89
5.9	The distribution of $(5 - P_f(x = 0, t))$ is depicted for (a) Scenario 2 and (b) Scenario 3. The black dashed line represents the 5 MPa pressure limit, and the red dashed line indicates all realizations exceeding 5 MPa.	91
5.10	Global sensitivity analysis for The Hague case: (a), (b), (c) are with thermal power objective function for 100, 300, and 600 months consecutively; (d), (e), (f) are with pressure at injector for 100, 300, and 600 months consecutively; (g), (h), (i) are with pressure at producer for 100, 300, and 600 months consecutively.	95
5.11	Optimization for heat extraction in The Hague, Netherlands considering uncertainties and safety: (a) the contour plot of the optimization results and (b) the thermal output power over time. In (a), the yellow lines denote the thermal output power and the purple lines the safe probability.	97
6.1	Analysis phases and objective functions used in the global sensitivity analysis for understanding the thermo-hydraulic process in The Hague, Netherlands.	103
6.2	Offline and online time for the calculation of entropy production. FEM denotes the finite element model, FD the calculation of entropy production with given pressure and temperature surrogate models, SM the calculation of entropy production using the entropy production surrogate model.	106
6.3	Global sensitivity analysis results with different objective functions before injection period: (a) total sensitivity index, and (b) first-order sensitivity index. The sensitivity results using the total thermal energy, the total kinetic energy, the total thermal and kinetic energy, and the total entropy production as objective functions are denoted by thermal, kinetic, combined, and entropy consecutively. The pink boxes highlight the significant parameters.	107
6.4	Global sensitivity analysis results with different objective functions during injection period: (a) total sensitivity index at $t = 100$ months, (b) first-order sensitivity index at $t = 100$ months, (c) total sensitivity index at $t = 300$ months, (d) first-order sensitivity index at $t = 300$ months, (e) total sensitivity index at $t = 600$ months, and (f) first-order sensitivity index at $t = 600$ months. The sensitivity results using the total thermal energy, the total kinetic energy, the total thermal and kinetic energy, and the total entropy production as objective functions are denoted by thermal, kinetic, combined, and entropy consecutively. The pink boxes highlight the significant parameters.	109

6.5	Analysis on governing heat transfer process using Rayleigh and Peclet number: (a) entropy production as a function of Rayleigh and Peclet number, (b) a 2-D view of (a) showing entropy production as a function of only Rayleigh number, and (c) a 2-D view of (a) showing entropy production as a function of only Peclet number. Ra_{cr} denotes the critical Rayleigh number.	111
7.1	The computational domain of the dolomitization process.	115
7.2	The evolution of the number of basis functions in NI-RB surrogate models with different numbers of training samples.	117
7.3	Comparison between the data-driven NN (blue color) and the NI-RB (orange color) surrogate model in predicting the calcite response with different numbers of training samples, utilizing blind case 1. (a) and (b) show a comparison in terms of calcite response and relative error consecutively for 5000 training samples. (c) and (d) show results for 10000 training samples, (e) and (f) for 50000 training samples.	119
7.4	Comparison between data-driven NN (blue color) and NI-RB (orange color) surrogate model in predicting the dolomite response with different numbers of training samples, utilizing blind case 1. (a) and (b) show the comparison in terms of dolomite response and relative error consecutively for 5000 training samples. (c) and (d) show results for 10000 training samples, Figure (e) and (f) for 50000 training samples.	120
7.5	Comparison between data-driven NN (blue color) and NI-RB (orange color) surrogate model in predicting the $CO_{2(aq)}$ response with different numbers of training samples, utilizing blind case 1. (a) and (b) show the comparison in terms of $CO_{2(aq)}$ response and relative error consecutively for 5000 training samples. (c) and (d) show results for 10000 training samples, (e) and (f) for 50000 training samples.	121
7.6	Comparison between data-driven NN (blue color) and NI-RB (orange color) surrogate model in predicting the pH response with different numbers of training samples, utilizing blind case 1. (a) and (b) show the comparison in terms of pH response and relative error consecutively for 5000 training samples. (c) and (d) show results for 10000 training samples, (e) and (f) for 50000 training samples.	122
7.7	Comparison between data-driven NN (blue color) and NI-RB (orange color) surrogate model in predicting the calcite response with different numbers of training samples, utilizing blind case 2. (a) and (b) show the comparison in terms of calcite response and relative error consecutively for 5000 training samples. (c) and (d) show results for 10000 training samples, (e) and (f) for 50000 training samples.	123
7.8	Comparison between data-driven NN (blue color) and NI-RB (orange color) surrogate model in predicting the dolomite response with different numbers of training samples, utilizing blind case 2. (a) and (b) show the comparison in terms of dolomite response and relative error consecutively for 5000 training samples. (c) and (d) show results for 10000 training samples, (e) and (f) for 50000 training samples.	124

7.9	Comparison between data-driven NN (blue color) and NI-RB (orange color) surrogate model in predicting the $\text{CO}_{2(\text{aq})}$ response with different numbers of training samples, utilizing blind case 2. (a) and (b) show the comparison in terms of $\text{CO}_{2(\text{aq})}$ response and relative error consecutively for 5000 training samples. (c) and (d) show results for 10000 training samples, (e) and (f) for 50000 training samples.	125
7.10	Comparison between data-driven NN (blue color) and NI-RB (orange color) surrogate model in predicting the pH response with different numbers of training samples, utilizing blind case 2. (a) and (b) show the comparison in terms of pH response and relative error consecutively for 5000 training samples. (c) and (d) show results for 10000 training samples, (e) and (f) for 50000 training samples. The black dashed-line in the relative error plot denotes the desired accuracy.	126
7.11	The offline and the online time for both the data-driven NN and the NI-RB surrogate models.	127
8.1	The amount of $\text{CO}_{2(\text{g})}$ gas liberated from a $\text{CO}_2\text{-NaCl-CaCO}_3$ solution at 50°C , 1×10^{-4} molal CaCO_3 , and at (a) 5 bar, (b) 10 bar, (c) 30 bar, and (d) 70 bar. Different colors denote different NaCl concentrations. The dots denote the CO_2 solubility, a condition at which the first gas bubble appears. I use 1×10^{-6} mol of gas as the criterion to define solubility. In inverse chemical equilibrium problems, the task is to accurately capture the dots.	131
8.2	Scenario used for constructing the surrogate models.	134
8.3	Illustration of the case study.	134
8.4	Illustration of sampling strategy used for the case study. The red dot refers to a single realization. The six plots at the left corner denote 500 samples of pressure, temperature, initial c_{NaCl} , and initial c_{CaCO_3} , generated using the Latin Hypercube sampling method. These 500 samples are plotted in two-dimensional format (from top to bottom, then left to right): temperature vs pressure, temperature vs $\log_{10} c_{\text{NaCl}}$, pressure vs $\log_{10} c_{\text{NaCl}}$, temperature vs $\log_{10} c_{\text{CaCO}_3}$, pressure vs $\log_{10} c_{\text{CaCO}_3}$, $\log_{10} c_{\text{NaCl}}$ vs $\log_{10} c_{\text{CaCO}_3}$. Within the red dot, ten samples of initial CO_2 concentration are generated following the enumeration at the right corner figure. Number 1 (green dot) is associated with CO_2 solubility, Number 2 is a 1.25 value of Number 1, Number 3 is a 0.75 value of Number 1, Number 4 is the upper bound, Number 5 is the lower bound, Number 6 is $\frac{1}{2}(\text{value of Number 3} + \text{value of Number 5})$, Number 7 is $\frac{1}{2}(\text{value of Number 3} + \text{value of Number 6})$, Number 8 is $\frac{1}{2}(\text{value of Number 3} + \text{value of Number 7})$, Number 9 is $\frac{1}{2}(\text{value of Number 2} + \text{value of Number 4})$, and Number 10 is $\frac{1}{2}(\text{value of Number 2} + \text{value of Number 9})$. The total number of samples above is $500 \times 10 = 5000$	137
8.5	Relative error distribution for 500 prediction samples obtained using (a) POD-NN trained with 50,000 training samples, generated via proposed sampling strategy in Section 8.4.1, (b) data-driven NN trained with 50,000 training samples, generated via proposed sampling strategy in Section 8.4.1, (c) POD-NN trained with 50,000 training samples, generated via Latin Hypercube sampling, and (d) data-driven NN trained with 50,000 training samples, generated via Latin Hypercube sampling.	139

8.6	Comparison between POD-NN (solid lines) and data-driven NN (dashed lines) in predicting the CO ₂ solubility at different pressure, temperature, and salinity. Figure (a), (c), (e), and (g) are produced by training both POD-NN and data-driven NN with 50,000 training samples generated via my proposed sampling strategy. Figure (b), (d), (f), and (h) are produced by training both POD-NN and data-driven NN with 50,000 training samples generated via the Latin Hypercube sampling method. Figure (a) and (b) are in 5 bar pressure, (c) and (d) are in 10 bar pressure, (e) and (f) are in 30 bar pressure, and (g) and (h) are in 70 bar pressure.	140
8.7	Relative error distribution for 500 prediction samples obtained using (a) POD-NN trained with 10,000 training samples, (b) data-driven NN trained with 10,000 training samples, (c) POD-NN trained with 25,000 training samples, (d) data-driven NN trained with 25,000 training samples, (e) POD-NN trained with 50,000 training samples, and (f) data-driven NN trained with 50,000 training samples.	142
8.8	Comparison between POD-NN (solid lines) and data-driven NN (dashed lines), trained with different numbers of training samples generated via my proposed sampling strategy, in predicting CO ₂ solubility at different pressure, temperature, and salinity. Figure (a), (d), (g), and (j) are trained with 10,000 training samples. Figure (b), (e), (h), and (k) are trained with 25,000 training samples. Figure (c), (f), (i), and (l) are trained with 50,000 training samples. Figure (a), (b), and (c) are in 5 bar pressure, Figure (d), (e), and (f) are in 10 bar pressure, Figure (g), (h), and (i) are in 30 bar pressure, and Figure (j), (k), and (l) are in 70 bar pressure.	143
8.9	Comparison of offline and online time using Reaktoro, data-driven NN, and POD-NN.	144
A.1	The structure of the fast Python library. This structure is inspired from the code development conducted by Ivagnes, Demo, and Rozza (2023) and O’Leary-Roseberry et al. (2022a).	xxi
A.2	General workflow for constructing a surrogate model with the fast Python library. This workflow is inspired by the works of Ivagnes, Demo, and Rozza (2023) and O’Leary-Roseberry et al. (2022a).	xxiii

List of Tables

3.1	Physical parameters ranges for Mandel's problem.	33
3.2	Physical parameters ranges for the Oedometer problem.	35
4.1	Geological description and rock properties of The Hague regional model.	47
4.2	Well names for different well numbers in Figure 4.2d.	50
4.3	Geological description and rock properties of The Hague reservoir model.	52
4.4	Prior distribution for each parameter used in uncertainty quantification of The Hague, Netherlands.	54
4.5	Tolerance and hyperparameters used in the surrogate construction for temperature state.	55
5.1	Prior distribution for each parameter used in the optimization of the synthetic case.	74
5.2	Prior distribution for each parameter used in the optimization of The Hague, Netherlands.	78
5.3	The performance of surrogate models used in the synthetic case. The approximation error is calculated using test samples. The values de- noted with a red font color represent the total time needed to generate the training and test samples, while the black color represents the time needed to construct a surrogate model. The values in the brackets de- note the time needed for a trained surrogate model to predict a new case with 31 time steps and the ones outside the brackets denote the time needed to predict a new case for a single time step.	86
5.4	Architectures and hyperparameters of surrogate models used in the syn- thetic case.	86
5.5	Performance of the surrogate models used in the The Hague case. The approximation error is calculated using test samples. The values de- noted with a red font color represent the total time needed to generate the training and test samples, while the black color represents the time needed to construct a surrogate model. The values in the brackets de- note the time needed for a trained surrogate model to predict a new case with 302 time steps and the ones outside the brackets denote the time needed to predict a new case for a single time step.	92
5.6	Architectures and hyperparameters of surrogate models used in the The Hague case.	92

6.1	Performance of the surrogate models used in the global sensitivity analysis. The approximation error is calculated using test samples. The values denoted with a red font color represent the total time needed to generate the training and test samples, while the black color represents the time needed to construct a surrogate model. The values in the brackets denote the time needed for a trained surrogate model to predict a new case with 302 time steps and the ones outside the brackets denote the time needed to predict a new case for a single time step. . .	105
6.2	Architectures and hyperparameters of surrogate models used in global sensitivity analysis.	105
7.1	Chemical systems for the dolomitization process.	115
7.2	Physical and chemical parameter ranges for the case study. Blind case 1 and 2 are used for testing the performance of the surrogate models. .	116
7.3	User-defined accuracy for each response for constructing basis functions.	117
7.4	The Optimum hyperparameters for both the data-driven NN and the NI-RB surrogate model used for calculating offline and online time. . .	127
8.1	Chemical systems for the case study.	135
8.2	Physical and chemical parameter ranges for the case study.	135
8.3	POD-NN and data-driven NN architecture, trained with 50,000 training samples generated via my proposed sampling strategy, for the calculation of offline and online time.	141

List of Abbreviations

AI	Artificial Intelligence
GSA	Global Sensitivity Analysis
I-RB	Intrusive Reduced Basis
MCMC	Markov Chain Monte Carlo
MC	Monte Carlo
NI-RB	Non-Intrusive Reduced Basis
NN	Neural Network
PDE	Partial Differential Equation
PINN	Physics-Informed Neural Network
QoI	Quantity of Interest
SA	Sensitivity Analysis
UQ	Uncertainty Quantification

Nomenclature

Symbol Name

B biases of a Neural Network model

d vector of measurement data

D operation-related parameters, such as injection rate and injection temperature

$f(\cdot)$ a quantity of interest function

$H[\cdot]$ entropy

l_r learning rate in a Neural Network model

N_b batch size in a Neural Network model

N_e number of epochs in a Neural Network model

Symbol Name

N_l number of layers in a Neural Network model

N_n number of neurons in a Neural Network model

N_{op} number of operation-related parameters

N_o number of optimization steps

N_p number of parameters

N_{tr} number of training samples

N_{ts} number of test samples

N_t number of time steps

N_x number of spatial discretizations

Symbol Name

S	sensitivity index
S^{local}	local sensitivity index
t	time
$\mathbf{W}$	weights of a Neural Network model
w	weights in the loss function of Physics-Informed Neural Network model
$\mathbf{x}$	spatial coordinates
$\mathbf{z}(\cdot)$	a vector of physical states
$\mathcal{P}(\cdot)$	probability distribution function
$\mathbb{P}$	a set
$\mathbb{R}$	real number space
ϵ	user-defined tolerance

Symbol Name

ω	a weight for either parameters or function in local sensitivity analysis
θ	hyperparameters of a Neural Network model
ξ	physical parameters (random variables)
ζ	relative error

Chapter 1

Introduction

*"What we observe is not nature itself
but nature exposed to our method of questioning."*

— Werner HEISENBERG (1958)

1.1 Motivation

Sustainable and environmentally friendly energy sources are essential to meet current energy demands while addressing the challenges of climate change. Geothermal energy emerges as a viable option, tapping into Earth's internal heat, located in subsurface structures (Jolie et al., 2021; Moeck, 2014). Understanding the processes within Earth's subsurface structures is important for the further development of safe and efficient heat extraction strategies.

Many geophysical methods, such as magnetotelluric (Di Paolo et al., 2020; Samrock et al., 2018) and gravity methods (Guglielmetti and Moscariello, 2021; Atef et al., 2016), and borehole measurements (Verweij et al., 2012; Mottaghy, Pechnig, and Vogt, 2011) are used to gain a direct understanding of subsurface processes. However, due to economic constraints, these direct measurements are usually sparse and cannot provide detailed predictions needed for the sizing of power plants or the design of optimal field development strategies (Franco and Vaccaro, 2014). A detailed understanding of subsurface processes can only be achieved through numerical simulations and analytical solutions. Employing analytical solutions is favorable because they are computationally inexpensive and have few degrees of freedom (Birdsell et al., 2024; Birdsell, Adams, and Saar, 2021). However, analytical solutions are based on simplified assumptions, making them less suitable for solving complex subsurface structures (Birdsell et al., 2024). Therefore, numerical simulations with process-based models are essential.

Numerical simulations solve non-linear partial differential equations (PDEs) related to fluid flow in porous media, mass transport, heat transport, chemical processes, and mechanical processes (Van Zelst et al., 2022; Kolditz et al., 2016; Kolditz et al., 2012). They require an accurate characterization of Earth's subsurface properties, e.g., permeability, porosity, chemical reaction rates, and elastic modulus, to deliver reliable predictions (Degen et al., 2023b; Degen, Veroy, and Wellmann, 2022; Schulte et al., 2020). Due to limited measurement data, primarily caused by economic constraints, and incomplete knowledge of physical/chemical processes, accurate characterizations are often unachievable, yielding uncertainties regarding the models' predictions (Degen et al., 2023b; Degen, Veroy, and Wellmann, 2022; Degen et al., 2021a; Schulte et al., 2020). Nonetheless, we still need to design efficient and safe heat extraction strategies (Schulte et al., 2020). This task is particularly challenging, as the

design must be optimal while accounting for all possibilities arising from the uncertainties.

In order to achieve an optimal design under uncertainties, optimization under uncertainties and parameter estimation studies, including uncertainty quantification and global sensitivity analyzes, are required. These studies provide risk estimations and optimal designs that are needed for the decision-making process. Both optimization under uncertainties and parameter estimation studies employ probabilistic approaches, such as the Monte Carlo (MC)-based method, which requires numerous model runs (Degen et al., 2021a; Degen et al., 2021b; Moustapha and Sudret, 2019).

Subsurface numerical models are often high-dimensional due to the use of fine meshes to capture complex geometries and to achieve numerical stability when solving nonlinear multiphysics PDEs (Degen et al., 2023b; Degen, Cacace, and Wellmann, 2022). Therefore, performing optimization studies under uncertainties and parameter estimation directly with these high-dimensional models is challenging. A possible solution is to construct a surrogate model to reduce the computational cost of model runs. The surrogate model should only reduce the computational cost in a mathematical sense, not by simplifying the physics, the geometries, or the parameterizations (Degen et al., 2021a). A simplification of the physics, the geometries, or the parameterizations would further introduce uncertainties into the model's predictions, therefore, driving us further away from the original problems. Reduction in a mathematical sense implies that the use of a surrogate model should not change the statistical behavior of the original problem, as could be identified through techniques such as global sensitivity analyzes (Degen et al., 2023b; Degen et al., 2021a).

Recently, machine learning methods have been employed to construct surrogate models to reduce the computational cost of running high-dimensional subsurface numerical models (Bauer et al., 2021; Willard et al., 2020; Bergen et al., 2019; Reichstein et al., 2019). The machine learning model is constructed by looking at the input-output relationship in a data-driven manner, namely *data-driven machine learning* (Willard et al., 2020; LeCun, Bengio, and Hinton, 2015). However, the drawbacks of data-driven machine learning methods are their lack of interpretability (Willcox, Ghattas, and Heimbach, 2021; Bauer et al., 2021) and the need for a large number of training samples (Tsai et al., 2021). These limitations highlight the need for a hybrid approach, combining physics and data-driven machine learning methods, referred to as *physics-based machine learning methods* (Willcox, Ghattas, and Heimbach, 2021; Karniadakis et al., 2021; Bauer et al., 2021; Willard et al., 2020).

Willard et al. (2020) discuss several recent physics-based machine learning methods, e.g., incorporating physics into the loss function of a machine learning model or incorporating physics in the form of a basis function and using a machine learning model to determine the weight for each basis function. However, not every physics-based machine learning method guarantees physically consistent predictions and efficient construction for geothermal applications.

Furthermore, most studies have focused mainly on multiphysics problems (Degen et al., 2023b; Willcox, Ghattas, and Heimbach, 2021; Karniadakis et al., 2021; Bauer et al., 2021; Willard et al., 2020), excluding the aspect of chemistry. In this dissertation, I incorporate both physical and chemical principles in the development of machine learning models. I refer to this approach as *process-based machine learning methods*. This term further emphasizes that the physics-based machine learning method (as discussed in many literature contributions, such as in Degen et al. (2023b) and Willcox, Ghattas, and Heimbach (2021)) can be extended and used for multiphysics and chemistry applications.

1.2 Machine learning meets small data

The notion of adopting machine learning for engineering and scientific computing has gained huge interest from the scientific community, especially with the rise of big data (Willcox, Ghattas, and Heimbach, 2021; Karniadakis et al., 2021). However, before implementing any machine learning method specifically for multi-physics and chemistry settings in geothermal applications, it is essential to understand the interplay between the amount of data and knowledge in physics/chemistry to successfully construct a machine learning model. Three distinct regimes can be distinguished by observing this interplay, as depicted in Figure 1.1:

- Lots of physics/chemistry knowledge & small data: a regime where the collection of measurement data is expensive and most physics / chemistry is known with incomplete characterization of material properties, geometries, boundary conditions, and initial conditions. The measurement data are sparse (Degen et al., 2023b; Karniadakis et al., 2021; Raissi, Perdikaris, and Karniadakis, 2019),
- Some physics/chemistry knowledge & some data: a regime where more data are offered to compensate for the incomplete characterization of material properties, geometries, boundary conditions, and initial conditions (Karniadakis et al., 2021; Raissi, Perdikaris, and Karniadakis, 2019),
- No physics/chemistry knowledge & big data: a regime where enormous amounts of data exist without any knowledge of the underlying physics/chemistry (Karniadakis et al., 2021).

Geothermal applications are situated in the regime of *lots of physics/chemistry knowledge & small data* (Degen et al., 2023b; Degen, Veroy, and Wellmann, 2022; Wellmann and Caumon, 2018; Caers, 2011), which is in the left part of Figure 1.1. This situation is notably different from recent successful implementations in Solid Earth applications, where most are situated in the *no physics/chemistry knowledge & big data* regime (see the right part of Figure 1.1). In these cases, data-driven machine learning models are used to identify and learn patterns, such as picking seismic time arrival (Zhu and Beroza, 2019; Bergen et al., 2019), identifying seismic events (Titos et al., 2018; Wiszniowski, Plesiewicz, and Trojanowski, 2014; Esposito et al., 2006; Bergen et al., 2019), and identifying lithologies (Kuhn, Cracknell, and Reading, 2018).

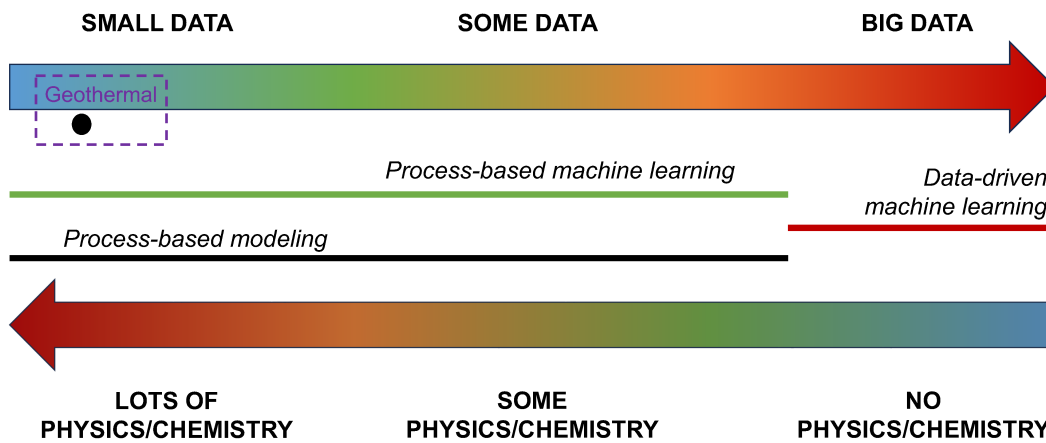


FIGURE 1.1: Interplay between the amount of data and the amount of physics/chemistry knowledge for the implementation of process-based modeling, data-driven machine learning, and process-based machine learning. This figure is adopted from Karniadakis et al. (2021) and modified.

The logical consequences for the sparse data condition are as follows:

- the measurement data should not be used to train a machine learning model, instead they should be used for validating and calibrating the physics and chemistry,
- the machine learning model should be interpretable for causal reasoning of parameter estimation or optimization results,
- the machine learning model should adhere the physical and chemical laws and produce physically/chemically consistent predictions,
- the construction of the machine learning model should be efficient, considering the expensive cost of running geothermal subsurface model.

For these reasons, process-based machine learning methods are developed by incorporating physics and chemistry laws into machine learning models.

There are several process-based machine learning methods: the Physics-Informed Neural Network (PINN) method, where physics is incorporated into the loss function of a Neural Network (NN) model in the form of soft constraints (Karniadakis et al., 2021; Raissi, Yazdani, and Karniadakis, 2020; Raissi, Perdikaris, and Karniadakis, 2019; Raissi and Karniadakis, 2018); the non-intrusive reduced basis (NI-RB) method, where the structure of physics is maintained in basis functions and a machine learning model is used to determine the weight for each basis function (Wang, Hesthaven, and Ray, 2019; Swischuk et al., 2019; Hesthaven and Ubbiali, 2018); the Fourier Neural Operator method and the Physics-Encoded Neural Network (PeNN) method, where the physics is introduced within the architecture of an NN model, e.g. in the form of a Fourier transform (Faroughi et al., 2022; Li et al., 2020). In this dissertation, I will only focus on the PINN method, which is the most popular physics-based machine learning method (Degen et al., 2023b; Karniadakis et al., 2021), and the NI-RB method, which is rigorously developed in the mathematics community and guarantees physical consistency in its predictions (Wang, Hesthaven, and Ray, 2019; Swischuk et al., 2019; Hesthaven and Ubbiali, 2018).

The Physics-Informed Neural Network (PINN) method has been implemented in many fluid mechanics applications (Raissi, Yazdani, and Karniadakis, 2020; Mao, Jagtap, and Karniadakis, 2020; Raissi and Karniadakis, 2018) and in Solid Earth applications (Amini, Haghighat, and Juanes, 2023; Amini, Haghighat, and Juanes, 2022; Haghighat, Amini, and Juanes, 2022; Yeung, Barajas-Solano, and Tartakovsky, 2022; Waheed et al., 2021). The concept involves encoding the physical laws in the loss function of an NN model, eliminating the need to develop a physics-based model (Karniadakis et al., 2021; Raissi, Yazdani, and Karniadakis, 2020; Raissi, Perdikaris, and Karniadakis, 2019; Raissi and Karniadakis, 2018). However, there have been reports of failure of the PINN method, which produces physically inconsistent predictions due to gradient pathology problems (Latrach et al., 2023; Santoso et al., 2022; Chuang and Barba, 2022; Krishnapriyan et al., 2021; Wang, Teng, and Perdikaris, 2021; Beuclet et al., 2021; Márquez-Neila, Salzmann, and Fua, 2017). Various approaches have been proposed to solve this problem, leading to a complex training process (Latrach et al., 2023; Amini, Haghighat, and Juanes, 2023; Amini, Haghighat, and Juanes, 2022; Haghighat, Amini, and Juanes, 2022; Krishnapriyan et al., 2021).

In contrast, process-based modeling remains an undoubtedly effective framework for accurately capturing physical and chemical processes, explaining causalities, and providing reliable predictions (Willcox, Ghattas, and Heimbach, 2021). Process-based modeling can be performed with *small* or *some data*, as indicated by the black line

in Figure 1.1. The NI-RB method is based on process-based modeling, maintaining the structure of the process-based model (Degen et al., 2023b; Willcox, Ghattas, and Heimbach, 2021; Hesthaven and Ubbiali, 2018). However, the NI-RB method is fairly unknown in Solid Earth applications (Degen et al., 2023b). There are several recent implementations for Solid Earth applications (Santoso et al., 2023a; Santoso et al., 2023b; Degen, Cacace, and Wellmann, 2022).

1.3 Understanding a complex system under uncertainties

Before performing any uncertainty quantification or optimization, it is crucial to understand the physical and chemical behaviors described by numerical models within given parameter ranges. This understanding is required to analyze whether the responses of the numerical models make physical and chemical sense and are descriptive with respect to the measurement data (Saltelli, Tarantola, and Campolongo, 2000; Box, 1979). This can be achieved through sensitivity analyses (Wainwright et al., 2014; Saltelli, Tarantola, and Campolongo, 2000). Since I am dealing with multi-physics and chemistry, a global sensitivity analysis is required (Saltelli et al., 2019; Sobol, 2001).

Performing a global sensitivity analysis for system understanding presents two challenges. First, I have to select a comprehensive objective function that captures the physical and chemical processes across the entire spatial and temporal domain (Saltelli et al., 2019; Wainwright et al., 2014). While I always have access to model states, such as pressure and temperature in transient thermo-hydraulic processes, analyzing complex processes solely based on these states can be difficult. Second, achieving a robust global sensitivity analysis demands intensive computation, particularly in terms of the cost associated with high-dimensional model runs and the calculation of certain objective functions. For example, if an objective function incorporates derivative terms, the computational cost of computing gradients must be considered. In this dissertation, I propose a thermodynamics-based objective function to enhance the understanding of complex systems and leverage the non-intrusive reduced-basis method, a process-based machine learning method, to alleviate the computational burden associated with numerous calculations of the objective function for high-dimensional models.

1.4 Aligning models and data remains challenging

Aligning numerical models with measurement data to ensure meaningful predictions involves two primary considerations. Firstly, I have to address the challenges posed by limited and sparse measurement data, often resulting from clustered measurement locations and unavailable data for specific areas (Degen et al., 2021b). Secondly, reducing the number of physical and chemical parameters involved in uncertainty quantification is essential (Degen, Veroy, and Wellmann, 2022; Degen et al., 2021b; Wainwright et al., 2014).

In this dissertation, I adopt a global sensitivity analysis (GSA)-driven uncertainty quantification strategy, as outlined in the work of Degen, Veroy, and Wellmann (2022) and Degen et al. (2021b). This strategy involves identifying biases caused by clustered measurement locations or limited data through global sensitivity analysis, followed by applying appropriate treatment to the measurement data, such as weighting (Degen, Veroy, and Wellmann, 2022; Degen et al., 2021b). Subsequently, I select the significant parameters based on the results of the global sensitivity analysis and proceed

with uncertainty quantification (Degen, Veroy, and Wellmann, 2022; Degen et al., 2021b). Furthermore, this approach necessitates numerous model runs, where I can leverage the non-intrusive reduced-basis method, a process-based machine learning technique, to reduce computational costs.

1.5 Finding an optimum under uncertainties

The primary objective of every geothermal project is to achieve an efficient and safe heat extraction process, which requires the design of safe and efficient operating conditions. However, because of the limited and sparse measurement data, the characterization of subsurface material properties is often inaccurate. Consequently, it becomes essential to account for uncertainties in all aspects of the optimization process, including the objective function and constraints, to ensure a robust and reliable design. The challenges are twofold: first, determining how to formulate objective functions and constraints and second, dealing with the computational cost of solving the optimization problem, given that sampling is involved in every component of the problem.

In this dissertation, I propose a Reliability-based Robust Design Optimization (RRDO) formulation to accommodate for uncertainties in both the objective function and the constraints (Bogoclu, Roos, and Nestorović, 2021; Moustapha et al., 2016). The idea is to solve the optimization with the mean response of the objective function and the probabilistic constraints (Bogoclu, Roos, and Nestorović, 2021). However, this formulation requires sampling in both objective function and constraints calculation. Here, I use the non-intrusive reduced-basis method, a process-based machine learning technique, to enable such numerous calculations.

1.6 Scientific aims

To safely and efficiently extract geothermal energy, it is crucial to determine optimal operating conditions, including fluid injection rate and fluid injection temperature. This process must take into account uncertainties in subsurface properties, such as thermal conductivity, permeability, porosity, and heat flow, due to limited measurement data and incomplete knowledge of the involved physics and chemistry. For this purpose, I raise three research questions:

- **Which process-based machine learning method is suitable for optimization under uncertainties and parameter estimation studies in geothermal applications?**

There are two related studies that compare different physics-based machine learning methods:

- Willard et al. (2020) present a comprehensive review of various physics-based machine learning methods, such as the physics-guided loss function approach, referring to the PINN method, and the hybrid physics-ML approach, including the NI-RB method. However, numerical proofs for comparison in Solid Earth or geothermal applications, where dealing with a *small data* situation is crucial, are notably absent. It is important to note that their objective is not to converge to a single best approach for domain-specific applications.

- Degen et al. (2023b) present a comparison between the PINN method and the NI-RB method for Solid Earth applications, including geothermal applications. They address scalability, interpretability, generalizability, and robustness of each method. Although there are numerical examples for the NI-RB method, a detailed numerical comparison between the PINN method and the NI-RB method, specifically for parameter estimation and optimization, is still lacking.

In this section, I contribute by offering a detailed numerical comparison between the PINN method and the NI-RB method, specifically dedicated to gain insights into potential large-scale parameter estimation and optimization implementations in geothermal applications. This study, connected to the other chapters, is driven by my necessity to identify an efficient and robust process-based machine learning method for my subsequent large-scale, real-case parameter estimation and optimization under uncertainties studies.

- **How an uncertainty quantification of the entire physical state is used for well targeting?**

Within a geothermal field development, determining the next well to be drilled, called *well targeting*, for either gathering more measurement data to reduce uncertainties or production/injection purpose is crucial. Leveraging the fast predictive capabilities of the NI-RB method allows us to efficiently generate uncertainty quantification for the entire physical state, contributing valuable insights to the decision-making process. Here, I quantify the uncertainties associated with drilling a producer in The Hague, expecting a temperature of 75 - 76 ° C at a target depth of around 2 km.

- **What is the optimal operating conditions for injection process, considering subsurface uncertainties and safety?**

Finding an optimal operating conditions is crucial to achieve safe and efficient heat extraction process. This optimization should consider both subsurface uncertainties and safety. The key point for this research question is how to impose a safety condition in which it is affected by subsurface uncertainties. Moreover, the use of the NI-RB method is useful to enable performing this optimization with a high-dimensional model such as the The Hague case. This is the first implementation of NI-RB in the context of geothermal applications.

Related to the third research question, it is essential to understand the physical processes occurring in The Hague before performing any optimization. This understanding should account for epistemic uncertainties due to the lack of measurement data that characterize both thermal and hydraulic properties. Hence, I also investigate:

- **Is entropy production a suitable objective function for a global sensitivity analysis of transient thermo-hydraulic processes?**

Considering two physical states, pressure and temperature, obtained from solving coupled thermo-hydraulic PDEs, finding an objective function that encapsulates both states remains a challenge. The objective function here should, in

essence, assign proper weights to each state, reflecting a certain concrete physical quantity. Börsing et al. (2017) investigate the use of entropy production for the analysis of conductive and convective systems. Building upon this research, I posit that entropy production can serve as a suitable objective function for Global Sensitivity Analysis (GSA) of transient thermo-hydraulic processes. This is the first investigation of using entropy production as an objective function in variance-based GSA. In connection with the other chapters, this study employs NI-RB to enable a GSA with an entropy production objective function in a real case, The Hague.

Another aspect to be considered when designing a safe and efficient heat extraction process is chemistry. Incorporating the chemistry aspect into multiphysics problems, such as in reactive transport simulations, increases computational costs. Therefore, performing parameter estimation or optimization involving chemistry poses significant computational challenges. In light of this, two additional research questions are raised:

- **Can the non-intrusive reduced basis (NI-RB) method be used to accelerate the entire reactive transport simulations?**

The reactive transport simulation plays a crucial role in predicting the transport of chemical species, chemical reactions, and mineral precipitation in geothermal applications, potentially impacting heat extraction and field operations. The simulation involves solving highly non-linear PDEs, producing "shock" features in the predictions that pose a challenge for surrogate modeling (Laloy and Jacques, 2022; Kyas et al., 2022; Fotherby et al., 2022; Sprocati and Rolle, 2021; Jagtap, Mudunuru, and Nakshatralla, 2021; Leal et al., 2020; Laloy and Jacques, 2019; Jatnieks et al., 2016; Stockmann et al., 2017; Steefel, DePaolo, and Lichtner, 2005). Given the success of the implementation of the NI-RB method in highly non-linear fluid dynamics applications (Pichi et al., 2023; Tonicello et al., 2022; Wang, Hesthaven, and Ray, 2019), I posit that the NI-RB method is promising for surrogate modeling in reactive transport simulations. This is the first implementation of NI-RB in this context. This study, connected to the other chapters, further demonstrates the ability of the NI-RB method to handle nonlinear equations before implementing it for large-scale, real-case parameter estimation studies.

- **How the non-intrusive reduced basis (NI-RB) be used to accelerate chemical equilibrium calculations?**

Within a reactive-transport simulation, chemical equilibrium is performed to predict chemical products as a result of several possible reactions, considering minimum Gibbs energy and mass conservation (Bethke, 2022; Leal et al., 2017; Leal, Kulik, and Kosakowski, 2016; Leal, Blunt, and LaForce, 2014; Beyer, Nagel, and Shao, 2014). Notably, in some applications that employ reactive transport simulation, the most computationally expensive task often involves the calculation of chemical equilibrium, particularly when numerous chemical species are involved (Laloy and Jacques, 2022; Kyas et al., 2022; Leal et al., 2020; Laloy and Jacques, 2019; Jatnieks et al., 2016). Efforts have been made to accelerate chemical equilibrium calculations using data-driven machine learning methods (Collard et al., 2023; De Lucia and Kühn, 2021; Laloy and Jacques, 2019; Jatnieks et al., 2016).

However, challenges persist, including low predictability and mass imbalance, the need for a large number of training samples, and the lack of interpretability in data-driven machine learning methods (Collard et al., 2023; Laloy and Jacques, 2022; De Lucia and Kühn, 2021; Leal et al., 2020). Moreover, given the success of the implementation of the intrusive reduced-basis (I-RB) method on PDE-constrained optimization problems (Pichi et al., 2022; Zhang, Bader, and Veroy, 2014; Negri et al., 2013), which has a form similar to the chemical equilibrium formulation, I propose adopting the projection formulation and combining it with a machine learning model to form the NI-RB method. This is the first implementation in this context. This study, which is specifically related to Chapter 7, further demonstrates the reliability of the NI-RB method for dealing with chemically related problems in geothermal applications.

1.7 Structure of the dissertation

The structure of this dissertation (see Figure 1.2) is as follows: I show the relevance of using process-based machine learning for parameter estimation and optimization under uncertainties in geothermal applications in Chapter 1. Chapter 2 provides a comprehensive explanation of the non-intrusive reduced-basis (NI-RB) method used throughout this dissertation, alongside an overview of studies addressing parameter estimation and optimization under uncertainties. Chapter 3 delves into the selection of a suitable process-based machine learning method for multiphysics parameter estimation and optimization under uncertainties in geothermal applications. The chosen method, the NI-RB method, becomes the primary focus and is subsequently implemented across different applications in several chapters. In Chapter 4, I begin by introducing the The Hague case study, which serves as the main case study of this dissertation. Then, I show a parameter estimation study for The Hague case. In Chapter 5, I further perform an optimization under uncertainties for the case study of The Hague, Netherlands. Chapter 6 aims to gain system understanding through the lens of entropy production, while Chapters 7 and 8 address chemistry problems.

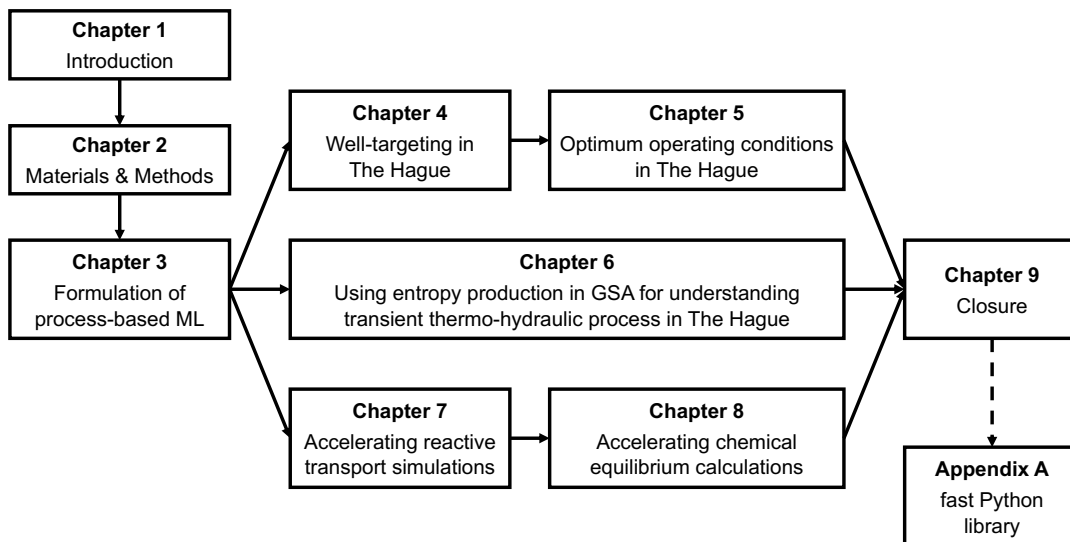


FIGURE 1.2: Structure of this dissertation.

In Chapter 4, I employ the NI-RB method to enable uncertainty quantification of the entire temperature state of The Hague model to evaluate the uncertainties in

finding the temperature spot of 75 - 76 ° C. Chapter 5 describes the use of the NI-RB method to find the optimal operating conditions, considering uncertainties and safety, in the case of The Hague, Netherlands.

In Chapter 6, I propose the use of entropy production as an objective function in the global sensitivity analysis to investigate the transient thermo-hydraulic process in The Hague, Netherlands. Chapter 6 also complements Chapter 4 and Chapter 5.

In Chapter 7, I show the first implementation of the NI-RB method to accelerate reactive transport simulations. In addition, I demonstrate the successful implementation of the NI-RB method for reactive transport simulation by accelerating chemical equilibrium calculation in Chapter 8. I present my conclusions and some recommendations for future research in Chapter 9. In Appendix A, I provide a tutorial on how to use the fast Python library to implement the NI-RB method.

1.8 Publications and conferences

The publications and conferences covering some Chapters in this dissertation are as follows:

The comparison among the data-driven machine learning method, the Physics-Informed Neural Network (PINN) method, and the non-intrusive reduced-basis (NI-RB) method using two hydro-mechanical benchmark cases (Chapter 3) exists in the form of an extended abstract and is presented in the following:

- Santoso, R. K., Degen, D., Cacace, M., and Wellmann, F.: State-of-the-art physics-based machine learning for hydro-mechanical simulation in geothermal applications, European Geothermal Congress, Berlin, Germany, 17 - 21 October (2022).

It is prepared for a journal publication as follows:

- **Santoso, R. K., Degen, D., Cacace, M., and Wellmann, F.: A physics-based machine learning formulation for enabling efficient uncertainty quantification in Solid Earth applications, Computers & Geosciences (in preparation, a).**

The optimization for the The Hague case, considering uncertainties and safety (Chapter 5) is prepared for a journal publication:

- **Santoso, R. K., Degen, D., Knapp, D., Pechnig, R., and Wellmann, F.: Optimum operating conditions with uncertainties and safety: A case study from The Hague, Netherlands (in preparation, b).**

The first implementation of entropy production as an objective function in variance-based global sensitivity analyses, assisted by a surrogate model constructed using the NI-RB method (Chapter 6) exists in the form of an extended abstract and is presented in

- Santoso, R. K., Degen, D., Knapp, D., Pechnig, R., and Wellmann, F.: Entropy production as a comprehensive indicator to address epistemic uncertainties – A case study of The Hague, Netherlands, 49th Workshop on Geothermal Reservoir Engineering, Stanford, California, 12 - 14 February (2024).

The first implementation of the NI-RB method in reactive transport simulations utilizing a benchmark case of the dolomitization problem (Chapter 7) is prepared for a journal publication:

- Santoso, R. K., Leal, A. M. M., Degen, D., Wellmann, F., and Saar, M. O.: **Process-based machine learning for reactive transport simulations in Solid Earth applications**, ACS Omega (in preparation, c).

The first implementation of the NI-RB method for accelerating the chemical equilibrium calculation (Chapter 8) is prepared for a journal publication:

- Santoso, R. K., Leal, A. M. M., Degen, D., Wellmann, F., and Saar, M. O.: **Chemical-based machine learning for forward and inverse chemical equilibrium reaction modeling**, Chemical Engineering Science (in preparation, d).

Chapter 2

Materials and methods

*"Science is not about making predictions or performing experiments.
Science is about explaining."*
— Bill GAEDE (2008)

This chapter introduces the concept of the non-intrusive reduced-basis (NI-RB) method, utilized extensively in this dissertation. My own developed Python library, namely **fast**, used for the implementation of the NI-RB method is presented later in Appendix A. I also present an overview of parameter estimation studies, including sensitivity analysis and uncertainty quantification and optimization under uncertainties.

2.1 Non-intrusive reduced-basis method

The projection-based model order reduction (MOR) methods aim to create a low-dimensional representation (a surrogate model) of a physics-based model by reducing its degree of freedom (Degen, Cacace, and Wellmann, 2022; Wang, Hesthaven, and Ray, 2019; Swischuk et al., 2019; Hesthaven and Ubbiali, 2018; Benner, Gugercin, and Willcox, 2015). These methods exploit the structure of the physics-based model to ensure physical consistency in the surrogate's predictions (Willcox, Ghattas, and Heimbach, 2021; Swischuk et al., 2019; Benner, Gugercin, and Willcox, 2015). Common projection-based MOR methods include the balanced truncation (BT) method (Benner, Gugercin, and Willcox, 2015; Antoulas, 2005) and the intrusive reduced basis (I-RB) method (Degen, Veroy, and Wellmann, 2020; Hesthaven, Rozza, Stamm, et al., 2016; Benner, Gugercin, and Willcox, 2015). The NI-RB method, which is the focus of this dissertation, is a modification of the intrusive reduced-basis method (I-RB) (Wang, Hesthaven, and Ray, 2019; Hesthaven and Ubbiali, 2018). The NI-RB method has been developed to deal with complex non-linear problems where the I-RB method has limitations (Wang, Hesthaven, and Ray, 2019; Hesthaven and Ubbiali, 2018). The reader is encouraged to consult Hesthaven, Rozza, Stamm, et al. (2016) for more explanation on the I-RB method.

The main idea of the NI-RB method is to approximate a physical solution in a linear combination of basis functions and coefficients. For time-independent problems, it is expressed as follows: (Wang, Hesthaven, and Ray, 2019; Swischuk et al., 2019; Hesthaven and Ubbiali, 2018)

$$\hat{\mathbf{z}}(\mathbf{x}, \boldsymbol{\xi}) = \hat{\mathbf{z}}(\boldsymbol{\xi}; \theta) = \sum_{k=1}^r \alpha_k(\boldsymbol{\xi}; \theta) V_k(\mathbf{x}), \quad (2.1)$$

and for time-dependent problems below (Wang, Hesthaven, and Ray, 2019; Swischuk et al., 2019; Hesthaven and Ubbiali, 2018)

$$\hat{\mathbf{z}}(\mathbf{x}, t, \boldsymbol{\xi}) = \hat{\mathbf{z}}(\boldsymbol{\xi}, t; \theta) = \sum_{k=1}^r \alpha_k(\boldsymbol{\xi}, t; \theta) V_k(\mathbf{x}), \quad (2.2)$$

where r is the reduced dimension, $\mathbf{V}(\mathbf{x})$ are the basis functions, $\alpha(\boldsymbol{\xi})$ or $\alpha(\boldsymbol{\xi}, t)$ are the reduced-basis coefficients, $\boldsymbol{\xi}$ are the parameters, and θ are the hyperparameters. The illustration is in Figure 2.1a for time-independent problems and 2.1b for time-dependent problems. Note that for time-dependent problems, it is necessary to store snapshots at different time steps, which requires a larger memory capacity.

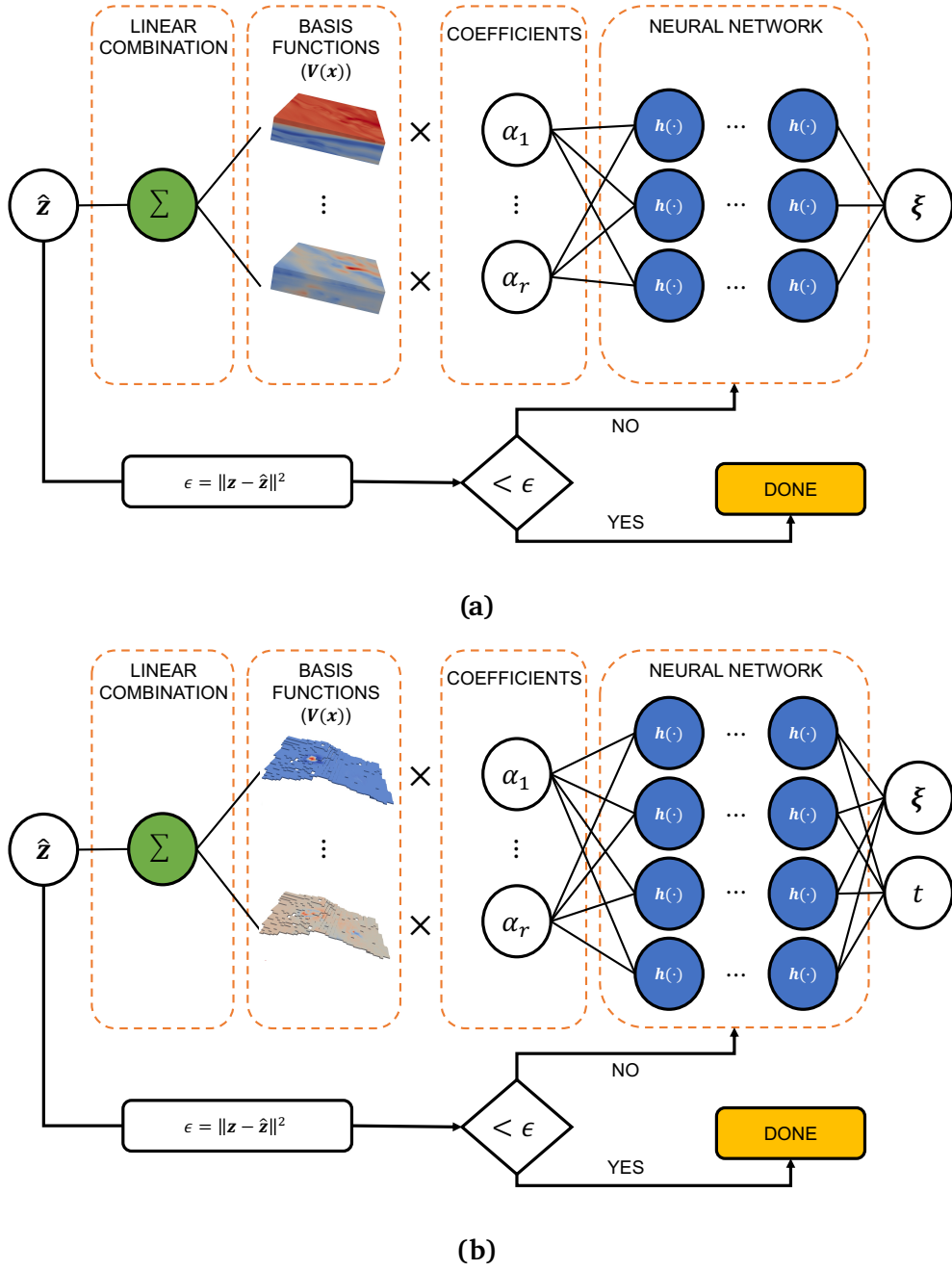


FIGURE 2.1: Construction of the NI-RB model where basis functions are obtained with the POD approach before performing machine learning (in this case, a Neural Network model) training. The time-independent formulation is in (a), where the basis functions are generated from a steady-state thermal simulation on the regional model of The Hague, and the time-dependent formulation is in (b), where the basis functions are generated from a transient thermo-hydraulic simulation on the reservoir model of The Hague.

There are two important steps within the NI-RB method: offline and online phase. The offline phase aims to construct the surrogate model, while the online phase is used to use the surrogate model for predictions or evaluations. The construction of the surrogate model consists of two steps: the calculation of the basis functions and the coefficients.

2.1.1 Generation of basis function

The basis functions are constructed by applying the Proper Orthogonal Decomposition (POD) to the training set (Wang, Hesthaven, and Ray, 2019; Swischuk et al., 2019; Hesthaven and Ubbiali, 2018). The POD performs a singular value decomposition (SVD) to obtain the eigenvector and the eigenvalue matrix of the snapshot matrix. The dimension of the eigenvector matrix is then reduced to form basis functions. This reduction is based on the "energy" term (utilizing eigenvalues), following the Schmidt-Eckart-Young theorem (Guo and Hesthaven, 2019). For time-independent problems, it is expressed as follows: (Guo and Hesthaven, 2019)

$$\frac{\sum_{k=1}^r \lambda_k^2}{\sum_{k=1}^{N_{tr}} \lambda_k^2} \leq \epsilon, \quad (2.3)$$

and for time-dependent problems, it is shown below (Guo and Hesthaven, 2019)

$$\frac{\sum_{k=1}^r \lambda_k^2}{\sum_{k=1}^{N_{tr} \times N_t} \lambda_k^2} \leq \epsilon, \quad (2.4)$$

where λ is the eigenvalue, r is the reduced dimension, and ϵ is the user-defined tolerance. For time-dependent problems, the two-step POD approach is used (Wang, Hesthaven, and Ray, 2019). The POD procedure maintains the structure of the physics-based model in the basis functions. The detailed steps are in Algorithm 1.

Algorithm 1: Generation of basis function in the non-intrusive reduced-basis (NI-RB) method.

input : $\mathbb{P}_{tr}$: training set with size N_{tr} , ϵ : user-defined tolerance.

output: basis function $V(\mathbf{x})$

if time-independent **then**

1. Perform SVD on $\mathbb{P}_{tr}$ to obtain left singular matrix V_h with the size $N_{tr} \times N_x$, eigenvalue (diagonal) matrix Σ with the size $N_{tr} \times N_{tr}$, and right singular matrix Q with the size $N_{tr} \times N_{tr}$.
2. Reduce V_h to V with the size $r \times N_x$ using Equation 2.3.

else

1. Perform two-step SVD on $\mathbb{P}_{tr}$ to obtain left singular matrix V_h with the size $N_{tr} \times N_t \times N_x$, eigenvalue (diagonal) matrix Σ with the size $(N_{tr} \times N_t) \times (N_{tr} \times N_t)$, and right singular matrix Q with the size $(N_{tr} \times N_t) \times (N_{tr} \times N_t)$.
2. Reduce V_h to V with the size $r \times N_x$ using Equation 2.4.

end

3. The basis functions $V(\mathbf{x})$ is obtained.
-

2.1.2 Calculation of reduced-basis coefficients

In contrast to the Galerkin projection used in the I-RB method to obtain coefficients, the NI-RB method employs a machine learning model. The machine learning model provides a map from physical parameters and time to reduced-basis coefficients (Wang, Hesthaven, and Ray, 2019; Swischuk et al., 2019; Hesthaven and Ubbiali, 2018). The machine learning model can be Gaussian Process Regression (GPR), k-Nearest Neighbor (KNN), polynomials, and Neural Network (NN) (Swischuk et al., 2019; Hesthaven

and Ubbiali, 2018). Since we are dealing with a multiphysics setting (nonlinear), I consider NN models to predict reduced-basis coefficients (Hesthaven and Ubbiali, 2018).

NN models explore input-output relationships through a series of multiplications with weighting matrices and activation functions that mimic the structure of biological neurons (LeCun, Bengio, and Hinton, 2015; Kriesel, 2007). The activation functions, for instance, hyperbolic tangent (tanh) function, sigmoid function, rectified linear unit (ReLU) function, etc., bring non-linear effect to the system to capture non-linear input-output relationship (LeCun, Bengio, and Hinton, 2015).

The approximate solution via an NN model can be expressed as shown below: (LeCun, Bengio, and Hinton, 2015; Kriesel, 2007).

$$\hat{\alpha}(\xi) \text{ or } \hat{\alpha}(\xi, t) = \hat{\alpha}(\cdot; \mathbf{W}, \mathbf{B}, \theta) = h(\dots h(\mathbf{W}(\cdot) + \mathbf{B})), \quad (2.5)$$

where $\hat{\alpha}$ represents approximate solutions, $\mathbf{W}$ is the weight, $\mathbf{B}$ is the bias, θ are the hyperparameters, and h is the activation function. The size of $\mathbf{W}$, $\mathbf{B}$, and h is influenced by the complexity of the architecture. The hyperparameters considered are the number of layers, the number of neurons in each layer, the learning rate, the number of epochs, and the batch size.

The training of the machine learning model utilizes a two-loop approach, presented in Algorithm 2.

Algorithm 2: Calculation of reduced-basis coefficients in the non-intrusive reduced-basis (NI-RB) method.

input : N_o : number of optimization steps, basis function $V(\mathbf{x})$, $\mathbb{P}_{tr}$: training set with size N_{tr} , $\mathbb{P}_{ts}$: test set with size N_{ts}
output: NeuralNetwork($\mathbf{W}$, $\mathbf{B}$, θ)

```

1. Initialize  $\mathbf{W}$ ,  $\mathbf{B}$ ,  $\theta$ 
for  $i \leftarrow 1$  to  $N_o$  do
    2. Equation 2.6 or 2.7  $\leftarrow$  NeuralNetwork( $\mathbb{P}_{tr}, \mathbf{W}_{i-1}, \mathbf{B}_{i-1}, \theta_{i-1}$ )
    3. Find optimum  $\mathbf{W}_i, \mathbf{B}_i$ 
    4. Equation 2.10 or 2.11  $\leftarrow$  NeuralNetwork( $\mathbb{P}_{ts}, \mathbf{W}_i, \mathbf{B}_i, \theta_{i-1}$ )
    5.  $\theta_i \leftarrow$  HypEvaluator( $\theta_{i-1}, \epsilon_{ts}$ )
end
6. Use the trained surrogate model for predicting new cases.
```

Note that: HypEvaluator contains any optimization algorithm to find optimum hyperparameters.

The inner loop minimizes the discrepancy between predicted reduced-basis coefficients and true reduced-basis coefficients (from training samples) to tune weights and biases under a set of hyperparameters (number of layers, number of neurons in each layer, number of epochs, learning rate, and batch size). I use the mean square error term to express the discrepancy between predicted reduced-basis coefficients and true reduced-basis coefficients. For time-independent problems, it is expressed as follows:

$$\frac{1}{N_{tr}} \sum_{k=1}^{N_{tr}} \left\| \alpha_k(\xi) - \hat{\alpha}_k(\xi; \mathbf{W}, \mathbf{B}, \theta) \right\|_{tr}^2, \quad (2.6)$$

and for time-dependent problems, it is as follows:

$$\frac{1}{N_{tr}} \sum_{k=1}^{N_{tr}} \left\| \alpha_k(\boldsymbol{\xi}, t) - \hat{\alpha}_k(\boldsymbol{\xi}, t; \mathbf{W}, \mathbf{B}, \theta) \right\|_{tr}^2, \quad (2.7)$$

Equation 2.6 or 2.7 is solved using the gradient descent approach. The reduced-basis coefficients for solving Equation 2.6 are obtained by the following calculation: (Swischuk et al., 2019; Benner, Gugercin, and Willcox, 2015)

$$\boldsymbol{\alpha}(\boldsymbol{\xi}) = \mathbf{V}^T \mathbf{z}(\mathbf{x}, \boldsymbol{\xi}), \quad (2.8)$$

and for Equation 2.7 are the following:

$$\boldsymbol{\alpha}(\boldsymbol{\xi}, t) = \mathbf{V}^T \mathbf{z}(\mathbf{x}, t, \boldsymbol{\xi}), \quad (2.9)$$

Note that I do not directly use the physical solution $\mathbf{z}$ in the minimization in Equation 2.6 or 2.7 to ensure efficiency, especially when dealing with large-scale models. One can imagine how many resources are needed if the physical solutions coming from a large-scale model are used, for instance, with the size of >1 million, to find optimum weights and biases (they might be on the order of millions) of a machine learning model. The outer loop minimizes the discrepancy between predicted physical solutions and true physical solutions (from test samples) to tune the hyperparameters. The mean squared error term is used to express the discrepancy. For time-independent problems, it is written as follows:

$$\frac{1}{N_{ts}} \sum_{k=1}^{N_{ts}} \left\| z_k(\mathbf{x}; \boldsymbol{\xi}) - \sum_{i=1}^r \alpha_i(\boldsymbol{\xi}; \theta) V_i(\mathbf{x}) \right\|_{ts}^2, \quad (2.10)$$

and for time-dependent problems the following:

$$\frac{1}{N_{ts}} \sum_{k=1}^{N_{ts}} \left\| z_k(\mathbf{x}, t; \boldsymbol{\xi}) - \sum_{i=1}^r \alpha_i(\boldsymbol{\xi}, t; \theta) V_i(\mathbf{x}) \right\|_{ts}^2, \quad (2.11)$$

Now I use the physical solution in minimization in Equation 2.10 or 2.11. With this two-loop approach, I reduce the amount of resources for training the machine learning model. Hence, I have an efficient approach to construct the NI-RB surrogate model. I use test samples for the evaluation of equation 2.10 or 2.11 to ensure the good predictability of the surrogate model. Equation 2.10 or 2.11, therefore, becomes the error estimator of the NI-RB model.

Training and test samples are constructed by collecting snapshots of physical solutions. In line with the previous idea for constructing training samples, training samples should be representative of the entire input parameter range and small. Therefore, I again use the Latin Hypercube sampling strategy (Iman and Conover, 1980). The test samples are constructed with a random sampling strategy and used to evaluate the predictability of the surrogate model. The test samples are as before guaranteed to not be within the training samples. The Latin Hypercube algorithm is provided by the SALib library (Herman and Usher, 2017).

Another important consideration is the preprocessing of the training data. The preprocessing aims to ensure training convergence (Misra, Li, and He, 2020), therefore, it helps produce a reliable surrogate model. Scales the samples into an equivalent range. There are several ways to perform preprocessing, for instance, the normal scaling or min-max scaling approach (Pedregosa et al., 2011). The normal scaling approach

scales the training samples to have a mean of zero and a standard deviation of one (Pedregosa et al., 2011). The min-max scaling approach scales each training sample in the $[0, 1]$ range with the maximum and minimum of all training samples (Pedregosa et al., 2011). I use a normal-scaling approach for the input and a min-max-scaling approach for the output.

In addition to the selection of appropriate training samples and the preprocessing of the data, another important point is the selection of test samples. The test samples here are used to measure the predictability of the surrogate model (Zhang et al., 2021). The test samples are randomly selected. They should not be in the training set. I use a random sampling approach for the selection of the test samples to ensure that the error produced from the evaluation with the test samples is representative of the error in the later prediction phase.

I use my Python library, *fast*, for the implementation of NI-RB method. The *fast* library consists of the Tensorflow Python library for constructing the NN model and performing the inner loop optimization process (Abadi et al., 2015), and the BOHB-HPO Python library for the implementation of the BOHB algorithm (Karakaşlı, 2021). The Bayesian optimization with hyperband (BOHB) algorithm developed by (Falkner, Klein, and Hutter, 2018) is used to perform the outer loop optimization process. This method is a modification of the traditional Bayesian optimization method. It enables parallelization and requires fewer optimization iterations than the traditional Bayesian optimization method due to combining it with the Hyperband method (Falkner, Klein, and Hutter, 2018). Please refer to Falkner, Klein, and Hutter (2018) for a detailed discussion of this method.

2.2 Parameter estimation studies

A parameter estimation study aims to understand the impact of uncertain parameters on model predictions, understand the correlation between parameters, and align the model with measurement data (Gábor and Banga, 2015; North, Pyle, and Zhang, 2014). This study includes sensitivity analysis (SA) and uncertainty quantification (UQ) (Aster, Borchers, and Thurber, 2018; Tarantola, 2005).

2.2.1 Sensitivity analysis

The idea of sensitivity analysis (SA) is to find the relative importance of different physical parameters in the predictions of the physics-based model (Saltelli et al., 2004). This basic idea can be extended to three specific objectives: (Degen et al., 2021a; Saltelli et al., 2019; Wainwright et al., 2014)

- to improve system understanding,
- to check the sufficiency of existing measurement data to determine the values of physical parameters under the uncertainties of other parameters,
- to identify significant physical parameters to be estimated or calibrated (involved in UQ).

SAs can be divided into local and global (Degen et al., 2021a; Wainwright et al., 2014; Saltelli et al., 2004).

A local SA is performed by calculating a scaled partial derivative of a quantity of interest (QoI) in a physics-based model with respect to a physical parameter. It is

mathematically expressed as follows: (Wainwright et al., 2014)

$$\begin{aligned} S_i^{local} &= \frac{\omega_{\xi,i}}{\omega_f} \frac{\partial f(\mathbf{z}(\mathbf{x}, t), \mathbf{x}, t, \mathbf{D}; \boldsymbol{\xi})}{\partial \xi_i} \bigg|_{\xi_i^o}, \\ &= \frac{\omega_{\xi,i}}{\omega_f} \frac{f(\xi_1^o, \dots, \xi_i^o + \Delta\xi, \dots, \xi_{N_p}^o) - f(\xi_1^o, \dots, \xi_{N_p}^o)}{\Delta\xi_i}, \end{aligned} \quad (2.12)$$

where S_i^{local} is the local sensitivity index of i -parameter, $\omega_{\xi,i}$ is a scaling factor of physical parameter (it can be the standard deviation or range of the parameter that represents the parameter variability), ω_f is a scaling factor of QoI function, $f(\mathbf{z}(\mathbf{x}, t), \mathbf{x}, t, \mathbf{D}; \boldsymbol{\xi})$ is a QoI as a function of physical states $\mathbf{z}(\mathbf{x}, t) \in \mathbb{R}^{N_t \times N_x}$, spatial coordinates $\mathbf{x} \in \mathbb{R}^{N_x}$, time $t \in \mathbb{R}^{N_t}$, operation-related parameters $\mathbf{D} \in \mathbb{R}^{N_{op}}$, and physical parameter $\boldsymbol{\xi} \in \mathbb{R}^{N_p}$, ξ_i^o is a pre-defined reference point. N_x is the number of spatial discretizations, N_t is the number of time discretizations, and N_p is the number of parameters. Local SAs only focus on the vicinity of a parameter, do not consider parameter correlations under a defined maximum variation, and are for linear problems (Degen et al., 2021a; Saltelli et al., 2019; Wainwright et al., 2014; Saltelli et al., 2004). Performing local SA is computationally cheaper than global sensitivity analysis (GSA), since it requires only two model runs to evaluate a sensitivity index (Wainwright et al., 2014). However, for geothermal applications, it is tricky to determine the reference point (e.g., "best" knowledge). An incorrect identification of this reference point will introduce an error in the sensitivity analysis, thus producing misleading estimates of parameter sensitivities and later producing unreliable UQ (Degen et al., 2021a). Furthermore, in geothermal applications, most problems are nonlinear, characterized by hyperbolic partial differential equations (PDEs) that describe multi-physics conditions (Degen et al., 2023b; Degen, Cacace, and Wellmann, 2022; Saltelli et al., 2019). Taking into account these limitations explored in previous studies by Degen et al. (2021a), Saltelli et al. (2019), and Wainwright et al. (2014), I only consider the use of GSA throughout the dissertation.

A global sensitivity analysis (GSA) addresses the importance of a physical parameter in a QoI by exploring the entire space of physical parameters, considering nonlinearities and correlations among parameters (Wainwright et al., 2014). It does not need a predefined reference point, avoiding the introduction of an error in the sensitivity calculation (Degen et al., 2021a). The commonly used method for GSA is the variance-based method, such as the Sobol method (Wainwright et al., 2014). The sensitivity index is calculated using a function decomposition, assuming that the QoI function is square-integrable (Sobol, 2001). There are several sensitivity indexes: the total sensitivity index that accounts for the total contribution of a physical parameter, including its correlation with other parameters, to the model; the first-order sensitivity index that accounts for the influence of a physical parameter on the model itself; the second-order sensitivity index that accounts the correlation between two parameters, and the high-order sensitivity index for multiple parameters (Sobol, 2001). The calculation of sensitivity indices follows the algorithm developed by Glen and Isaacs (2012). The idea behind the Glen and Isaacs (2012)'s algorithm is as follows:

- create two matrices $\mathcal{A}$ and $\mathcal{B}$ where each of them has the size of $N_s \times N_p$ and contains N_s realizations sampled using MC method,
- perform model runs with all realizations within matrix $\mathcal{A}$ and $\mathcal{B}$ to get vector a and b consecutively where each of them has the size of $N_s \times 1$,

- create a matrix $\mathcal{C}$ to calculate sensitivity index for each parameters. For example, if I want to calculate sensitivity index of parameter 1, I copy all values in matrix $\mathcal{B}$ to matrix $\mathcal{C}$ and replace the first column in matrix $\mathcal{C}$ with the first column in matrix $\mathcal{A}$, written as follows $\mathcal{C} = [\mathcal{A}[:, 1], \mathcal{B}[:, 2 : N_p]]$,
- perform model runs with all realizations in matrix $\mathcal{C}$ to get vector c_i (in this case c_1 for parameter 1) where the size is $N_s \times 1$,
- repeat step 3 and 4 for different parameters. If I want to calculate high-order index, let us say sensitivity of parameter 1 and 2, I again copy all values in matrix $\mathcal{B}$ to matrix $\mathcal{C}$ and replace the first and second columns in matrix $\mathcal{C}$ with the first and second column in matrix $\mathcal{A}$, written as follows $\mathcal{C} = [\mathcal{A}[:, 1], \mathcal{A}[:, 2], \mathcal{B}[:, 3 : N_p]]$.

Looking at the steps above, the total model runs are $N_s \times (N_p + 2)$, where N_s is the number of realizations and N_p is the number of parameters. I can mathematically express the first-order sensitivity index as follows:

$$S_i = \frac{1}{\text{Var}[f(\mathbf{z}(\mathbf{x}, t), \mathbf{x}, t, \mathbf{D}; \boldsymbol{\xi})]} \frac{1}{N_s - 1} \sum_{w=1}^{N_s} (a_w - \mathbb{E}[f(\mathbf{z}(\mathbf{x}, t), \mathbf{x}, t, \mathbf{D}; \boldsymbol{\xi})]) (c_{i,w} - \mathbb{E}[f(\mathbf{z}(\mathbf{x}, t), \mathbf{x}, t, \mathbf{D}; \boldsymbol{\xi})]), \quad (2.13)$$

where S is the sensitivity index, a is the element of the matrix $\mathcal{A}$, c is the element of the matrix $\mathcal{C}$, $\text{Var}[\cdot]$ is the variance and $\mathbb{E}[\cdot]$ is the mean. The high-order sensitivity index can be expressed as follows:

$$S_{ij..k} = \frac{1}{\text{Var}[f(\mathbf{z}(\mathbf{x}, t), \mathbf{x}, t, \mathbf{D}; \boldsymbol{\xi})]} \frac{1}{N_s - 1} \sum_{w=1}^{N_s} (a_w - \mathbb{E}[f(\mathbf{z}(\mathbf{x}, t), \mathbf{x}, t, \mathbf{D}; \boldsymbol{\xi})]) (c_{ij..k,w} - \mathbb{E}[f(\mathbf{z}(\mathbf{x}, t), \mathbf{x}, t, \mathbf{D}; \boldsymbol{\xi})]), \quad (2.14)$$

The total sensitivity index S^T is expressed below:

$$S_{ij..k}^T = \frac{1}{\text{Var}[f(\mathbf{z}(\mathbf{x}, t), \mathbf{x}, t, \mathbf{D}; \boldsymbol{\xi})]} \frac{1}{N_s - 1} \sum_{w=1}^{N_s} (c_{ij..k,w} - \mathbb{E}[f(\mathbf{z}(\mathbf{x}, t), \mathbf{x}, t, \mathbf{D}; \boldsymbol{\xi})]) (b_w - \mathbb{E}[f(\mathbf{z}(\mathbf{x}, t), \mathbf{x}, t, \mathbf{D}; \boldsymbol{\xi})]), \quad (2.15)$$

where b is the element of the matrix $\mathcal{B}$.

To construct the matrix $\mathcal{A}$ and $\mathcal{B}$, Saltelli et al. (2008) develop an algorithm based on Monte Carlo (MC) sampling. The convergence rate of the MC sampling method is $\frac{1}{\sqrt{N_s}}$ where N_s is the number of realizations, following the law of large numbers (Lapeyre, 2007). Thus, a robust GSA requires a large number of samples, along with numerous model runs. In this context, a surrogate model is necessary to accelerate the run of a model and enable a robust GSA. In this dissertation, I will only use the variance-based global sensitivity analysis.

2.2.2 Uncertainty quantification

An uncertainty quantification aims to systematically assess and characterize uncertainties associated with a physics-based model on model predictions, e.g., uncertainties in physical parameters, model geometries, boundary conditions and initial conditions. They are achieved by aligning the model predictions with some measurement data (Sullivan, 2015). The commonly used method is the Markov Chain Monte Carlo (MCMC) (Stuart, 2010) which is based on the Bayes theorem, expressed as follows:

$$\mathcal{P}(\boldsymbol{\xi}|\mathbf{d}) \propto \mathcal{P}(\mathbf{d}|\boldsymbol{\xi}) \mathcal{P}(\boldsymbol{\xi}) \quad (2.16)$$

where $\mathcal{P}(\boldsymbol{\xi}|\mathbf{d})$ is the posterior distribution, which describes the updated knowledge of the parameters $\boldsymbol{\xi}$ given the measurement data $\mathbf{d}$, $\mathcal{P}(\mathbf{d}|\boldsymbol{\xi})$ is the likelihood, and $\mathcal{P}(\boldsymbol{\xi})$ is the prior distribution. For sampling the posterior distribution, there are various algorithms, such as Metropolis-Hastings and Adaptive Metropolis. I will only use the Metropolis-Hastings algorithm to perform MCMC since it is not the scope of this dissertation to explore the other algorithms.

For the implementation of the MCMC method, it is important to select the optimum hyperparameters to ensure convergence. Hyperparameters are number of iterations, burn-in numbers, thinning, and proposal distribution scale (Haario, Saksman, and Tamminen, 2001). The convergence criteria are autocorrelation (Raftery and Lewis, 1995), z scores (Geweke, 1991), and goodness-of-fit (Brooks, Catchpole, and Morgan, 2000). The autocorrelation of all samples should be around zero (Raftery and Lewis, 1995), the z scores should be between +2 and -2 (Geweke, 1991), and the predictions of the model should be close to the measurement data (Brooks, Catchpole, and Morgan, 2000).

To meet convergence criteria, a large number of realizations are often required, typically in the order of millions. Therefore, the use of surrogate model is crucial to represent the physics-based model, then, enable this large number of realizations.

Another important consideration is the number of parameters involved in uncertainty quantification. If there is a large number of parameters, the method will suffer from a curse of dimensionality, necessitating an extremely large number of realizations that are not computationally affordable even with the use of surrogate model. Furthermore, when statistically insignificant parameters are part of uncertainty quantification, achieving convergence becomes challenging as estimating the values of these parameters becomes difficult. Therefore, it is essential to perform GSA before performing any uncertainty quantification (Degen, Veroy, and Wellmann, 2022; Degen et al., 2021a).

2.3 Optimization under uncertainties studies

Once a model is aligned with the measurement data, optimization can be performed to identify the optimum operating conditions. The optimization process involves three important components: (Bogoclu, Roos, and Nestorović, 2021; Moustapha, Marelli, and Sudret, 2019)

- Robust objective function(s) for optimizing operating conditions: The objective function(s) may comprise multiple functions, resulting in a multi-objective optimization. It is essential that the objective function(s) account for the uncertainties of physical parameters to achieve a robust operating condition.

- Constraint(s) for safety: One of the safety constraints must quantify the failure probabilities associated with the selected operating condition. Note that the safety constraint(s) is also influenced by uncertain physical parameters.
- Constraint(s) for operating conditions: One of the constraints should define the bounds for the operating conditions, taking into account economic constraints related to the efficiency of heat extraction and the size of the power plant.

By incorporating these components, the optimization process ensures a comprehensive consideration of uncertainties and safety, which ultimately leads to the determination of robust and efficient operating conditions. These three components lead to the optimization formulations of Reliability-Based Design Optimization (RBDO) and Reliability-Based Robust Design Optimization (RRDO) (Bogoclu, Roos, and Nestorović, 2021; Moustapha, Marelli, and Sudret, 2019; Tang, Chen, and Wei, 2012). These formulations are widely used in the civil and mechanical engineering domain (Moustapha, Marelli, and Sudret, 2019; Tang, Chen, and Wei, 2012; Zang, Friswell, and Motterhead, 2005), not in geothermal applications.

In geothermal applications, the commonly used approach to perform optimization under uncertainties involves optimizing the mean response of the objective function while taking into account uncertainties that arise from physical parameters, called Robust Design Optimization (RDO) (Bogoclu, Roos, and Nestorović, 2021). Safety constraints are often implicitly incorporated (and are not quantified) by selecting specific ranges of operating conditions that mitigate the risk of unsafe situations (Qin et al., 2023; Yan, Gudala, and Sun, 2023; Wang et al., 2023; Schulte et al., 2020; Arnold et al., 2016; Chen, Li, and Reynolds, 2012). For example, in an injection process, one might narrow the range of possible injection rates to prevent the creation of fractures or the reactivation of existing faults, thus addressing the issue of induced seismicity.

However, this approach can lead to a conservative and expensive design or even to an incorrect one, given incomplete knowledge of physical processes and limited data. In addition, I face challenges in quantifying the associated risks associated with our chosen operating conditions. Therefore, a more reliable strategy involves explicitly imposing safety constraints that account for uncertainties in their calculation, as is done in Reliability-Based Design Optimization (RBDO) or Robust and Reliable Design Optimization (RRDO) formulations.

The RBDO aims to find optimal operation-related parameters $\mathbf{D} \in \mathbb{R}^{N_{op}}$ under uncertainty in several physical parameters. This is mathematically expressed as follows: (Bogoclu, Roos, and Nestorović, 2021)

$$\mathbf{D}^* = \operatorname{argmin} f^{opt}(\mathbf{D}) \quad \text{subject to} \quad \begin{cases} f_j^{soft}(\mathbf{D}) \leq 0, & j = 1, \dots, \\ \mathcal{P}(\min_k (f_k^{hard}(\mathbf{D}, \boldsymbol{\xi})) \leq 0) \leq \mathcal{P}_k^{user}, & k = 1, \dots, \end{cases} \quad (2.17)$$

Here, $\mathbf{D}^*$ represents the optimal operation-related parameters, $f^{opt}(\cdot)$ is the Quantity of Interest (QoI) to be optimized, such as power output in geothermal applications, $f^{soft}(\cdot)$ is a QoI serving as a soft constraint, for example, constraint for operating condition, $f^{hard}(\cdot)$ is a QoI serving as a hard constraint, such as maintaining the pressure below the fracture limit, and $\mathcal{P}^{user}$ is a user-defined threshold probability. An issue with this formulation is that it does not account for uncertainties within the objective function. Incorporating uncertainties into the objective function is crucial in geothermal applications, such as optimizing power output, which depends on uncertain subsurface properties.

An extension of RBDO to accommodate uncertainties within the objective function is the RRDO. It is mathematically expressed as follows: (Bogoclu, Roos, and

Nestorović, 2021)

$$\begin{aligned} \mathbf{D}^* = \operatorname{argmin} \quad & \mathbb{E}[f^{opt}(\mathbf{D}, \boldsymbol{\xi})] \\ \text{subject to} \quad & \begin{cases} f_j^{soft}(\mathbf{D}) \leq 0, & j = 1, 2, \dots, \\ \mathcal{P}(\min_k (f_k^{hard}(\mathbf{D}, \boldsymbol{\xi})) \leq 0) \leq \mathcal{P}_k^{user}, & k = 1, 2, \dots, \end{cases} \end{aligned} \quad (2.18)$$

I follow the strategy Bogoclu, Roos, and Nestorović (2021) to solve Equation 2.18. They use Monte Carlo method to calculate the probability of hard constraints $f^{hard}(\cdot)$ and Latin Hypercube sampling method to aid in the calculation of the objective function(s). The optimization is performed using the gradient-free NSGA-II algorithm, a modification of the genetic algorithm that allows fast convergence with a relatively small number of iteration steps and efficient handling of multi-objective functions (Deb et al., 2002).

A challenge in solving the RRDO problem is that it requires numerous runs of models to solve both the optimization and the probabilistic part (Bogoclu, Roos, and Nestorović, 2021; Moustapha et al., 2016). A surrogate model is needed to represent the physics-based model to enable optimization under uncertainties.

Chapter 3

A suitable process-based machine learning method for parameter estimation and optimization under uncertainties studies in geothermal applications

"Extraordinary claims require extraordinary evidence."

— Carl SAGAN (1979)

Before performing any parameter estimation and optimization under uncertainties studies in a real case study, it is crucial to identify an efficient and robust method for constructing a surrogate model. This is particularly important given the common use of high-dimensional models in geothermal applications. Performing either parameter estimation or optimization under uncertainties directly on these models is computationally challenging. I compare three different methods for constructing a surrogate model: the data-driven machine learning method, the Physics-Informed Neural Network (PINN) method, and the non-intrusive reduced-basis (NI-RB) method. Two hydro-mechanical (HM) benchmark cases, namely the 2-D Mandel's and 3-D Oedometer problem, are employed for the comparison. This chapter is covered in Santoso et al. (2021, 2022) and Santoso et al. (in preparation, a).

3.1 Surrogate modeling using machine learning

In this Section, I provide a review about machine learning methods for surrogate modeling applications. Specifically, I focus on data-driven machine learning methods and the Physics-Informed Neural Network (PINN) method. Note that the NI-RB method has already been explained in Chapter 2. In this chapter, I use the time-dependent NI-RB formulation, employing a Neural Network model for calculating the reduced-basis coefficients.

3.1.1 Data-driven machine learning methods

A data-driven machine learning method constructs surrogate models that map physical/chemical parameters ξ or spatio-temporal coordinates $\mathbf{x}, t$ to solutions $z(\mathbf{x}, t; \xi)$. Initially, the mapping from physical/chemical parameters ξ and spatio-temporal coordinates $\mathbf{x}, t$ into solutions $z(\mathbf{x}, t; \xi)$ is described through a set of partial differential

equations (PDEs), so-called process-based model, which is often computationally expensive to solve and requires a huge amount of resources, for example, in high performance computing (HPC) infrastructures. This is due to the use of fine meshes to stabilize the numerical scheme when dealing with complex geometries and wide-range model parameters (Kolditz et al., 2016). The data-driven machine learning method treats responses from the process-based model as data. Thus, the training process minimizes the discrepancy between the predictions of the machine learning model and the responses of the process-based model. Numerous machine learning models, such as Gaussian Process Regression (GPR), Neural Networks (NN), Autoencoder (AE), etc., exist. Two commonly used machine learning models in geoscience are GPR and NN (Morikawa et al., 2021; Gudmundsdottir and Horne, 2020; Zhu and Beroza, 2019; Elsheikh, Wheeler, and Hoteit, 2013). GPR models explore input-output relationships of a given training samples through a kernel (e.g. exponential kernel, etc.) (Wang, 2020; Rasmussen and Williams, 2006; Bishop and Nasrabadi, 2006) and are usually deployed for linear cases (Raissi, Babaee, and Karniadakis, 2019; Raissi, Perdikaris, and Karniadakis, 2017), while NN models are for non-linear cases (Swischuk et al., 2019; Hesthaven and Ubbiali, 2018). In this dissertation, I am dealing with nonlinear multiphysics problems, therefore, I only consider the use of NN models.

Neural Networks (NN)

The data-driven NN model approximates a solution as shown below: (LeCun, Bengio, and Hinton, 2015; Kriesel, 2007)

$$\hat{\mathbf{z}}(\mathbf{x}, t; \boldsymbol{\xi}) = \hat{\mathbf{z}}(\cdot; \mathbf{W}, \mathbf{B}, \theta) = h(\dots h(\mathbf{W}(\cdot) + \mathbf{B})), \quad (3.1)$$

where $\hat{\mathbf{z}}$ represents approximate solutions, $\mathbf{W}$ is the weight, $\mathbf{B}$ is the bias, θ are the hyperparameters, and h is the activation function. The size of $\mathbf{W}$, $\mathbf{B}$, and h is influenced by the complexity of the architecture. The hyperparameters considered are the number of layers, the number of neurons in each layer, the learning rate, the number of epochs, and the batch size. A schematic representation of the data-driven NN model is shown in Figure 3.1.

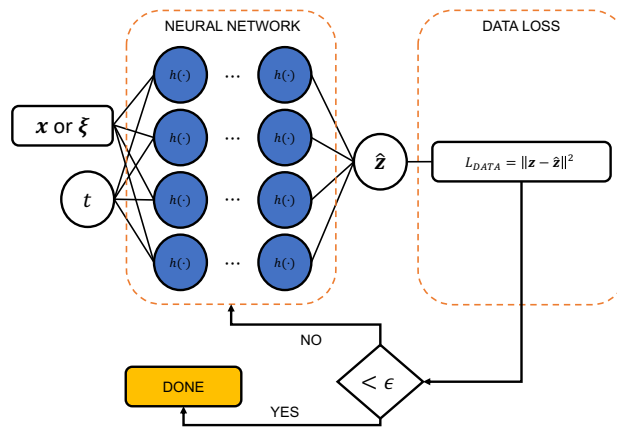


FIGURE 3.1: A data-driven NN model, constructed by minimizing a data loss function.

As mentioned in Section 2.1.2, having representative samples and performing the preprocessing of training and test samples are critical to ensure the efficiency and good predictability of the surrogate model.

I perform the training process in two loops as shown in Algorithm 3 (Goodfellow, Bengio, and Courville, 2016). The inner loop aims to tune the weights and biases of an NN model under a set of hyperparameters. This step minimizes the error between the NN model predictions and the process-based model solutions, utilizing a training set. The trained NN model is then passed to the outer loop for further hyperparameter tuning. In the outer loop, I try to find an optimum set of hyperparameters for the next inner loop iteration to ensure good predictability of the NN model. The step minimizes the error between the NN model predictions and the process-based model solutions, utilizing a set of tests. I iterate these inner and outer loop steps until I reach the desired accuracy.

Algorithm 3: Training steps for Neural Network models

input : N_o : number of optimization steps, $\mathbb{P}_{tr}$: training set with size N_{tr} , $\mathbb{P}_{ts}$: test set with size N_{ts}

output: NeuralNetwork($\mathbf{W}$, $\mathbf{B}$, θ)

Initialize $\mathbf{W}$, $\mathbf{B}$, θ : number of layers N_l , number of neurons N_n , learning rate l_r , number of epochs N_e , batch size N_b

for $i \leftarrow 1$ **to** N_o **do**

for $j \leftarrow 1$ **to** $N_{e,i-1}$ **do**

$$\epsilon_{tr} = \frac{1}{N_{tr}} \sum_{k=1}^{N_{tr}} \left\| z_k(\mathbf{x}, t; \boldsymbol{\xi}) - \hat{z}_k(\cdot; \mathbf{W}_{i-1}, \mathbf{B}_{i-1}, \theta_{i-1}) \right\|_{tr} \leftarrow$$

 NeuralNetwork($\mathbb{P}_{tr}, \mathbf{W}_{i-1}, \mathbf{B}_{i-1}, \theta_{i-1}$)

 find optimum $\mathbf{W}_i, \mathbf{B}_i$

end

$$\epsilon_{ts} = \frac{1}{N_{ts}} \sum_{k=1}^{N_{ts}} \left\| z_k(\mathbf{x}, t; \boldsymbol{\xi}) - \hat{z}_k(\cdot; \mathbf{W}_i, \mathbf{B}_i, \theta_{i-1}) \right\|_{ts} \leftarrow$$

 NeuralNetwork($\mathbb{P}_{ts}, \mathbf{W}_i, \mathbf{B}_i, \theta_{i-1}$)

$\theta_i \leftarrow \text{HypEvaluator}(\theta_{i-1}, \epsilon_{ts})$

end

Note that: HypEvaluator contains any optimization algorithm to find optimum hyperparameters. ϵ_{tr} is error evaluated on training samples and ϵ_{ts} is error evaluated on test samples.

I use the same Python libraries as used in Section 2.1.2. Both training and test set construction and machine learning training are conducted using a single core on an Apple MacBook Pro with an M1 Chip and 8 GB RAM.

3.1.2 Process-based machine learning methods

Process-based machine learning methods incorporate physics/chemistry into the construction of machine learning models to improve their interpretability and reliability (Willcox, Ghattas, and Heimbach, 2021; Willard et al., 2020). Recently, the PINN method is widely used for surrogate modeling in geoscience. On the other hand, the NI-RB method has been rigorously developed within mathematics communities, however, it is fairly unknown for geoscience communities. Hence, I consider the PINN method and the NI-RB method in my comparative study. In this Section, I only present an overview of the PINN method as Section 2.1 already provides an overview about the NI-RB method.

Physics-Informed Neural Network (PINN)

The main idea of the PINN method is to add residual PDEs of physical laws, initial conditions, and boundary conditions to the loss function of the NN model L , mathematically expressed as follows: (Raissi, Perdikaris, and Karniadakis, 2019; Raissi and Karniadakis, 2018)

$$\begin{aligned}
 L = & w_d \left\| \mathbf{z}(\mathbf{x}, t, \boldsymbol{\xi}) - \hat{\mathbf{z}}(\mathbf{x}, t; \mathbf{W}, \mathbf{B}, \theta) \right\|^2 + \\
 & w_{p,res} \left\| \mathcal{L}(\hat{\mathbf{z}}(\mathbf{x}, t; \mathbf{W}, \mathbf{B}, \theta), \boldsymbol{\xi}) - 0 \right\|^2 + \\
 & w_{p,IC} \left\| \mathbf{z}_{IC}(\mathbf{x}, \boldsymbol{\xi}) - \hat{\mathbf{z}}(\mathbf{x}, t = 0; \mathbf{W}, \mathbf{B}, \theta) \right\|^2 + \\
 & w_{p,BC_D} \left\| \mathbf{z}_D(\mathbf{x}, t, \boldsymbol{\xi}) - \hat{\mathbf{z}}(\mathbf{x} \in \partial\Omega_D, t; \mathbf{W}, \mathbf{B}, \theta) \right\|^2 + \\
 & w_{p,BC_N} \left\| \mathbf{z}_N(\mathbf{x}, t, \boldsymbol{\xi}) - \hat{\mathbf{z}}(\mathbf{x} \in \partial\Omega_N, t; \mathbf{W}, \mathbf{B}, \theta) \right\|^2,
 \end{aligned} \tag{3.2}$$

where w_d is the weight for prediction-data error term, $w_{p,res}$ is the weight for PDEs residual, $w_{p,IC}$ is the weight for initial conditions, w_{p,BC_D} is the weight for Dirichlet boundary conditions, and w_{p,BC_N} is the weight for Neumann boundary conditions. Again, different problems might have different initial conditions and boundary conditions (such as Robin boundary condition), not restricted with the expression in Equation 3.2. The PINN method relies on automatic differentiation to calculate the loss function (Raissi, Perdikaris, and Karniadakis, 2019; Raissi and Karniadakis, 2018). Its illustration is in Figure 3.2.

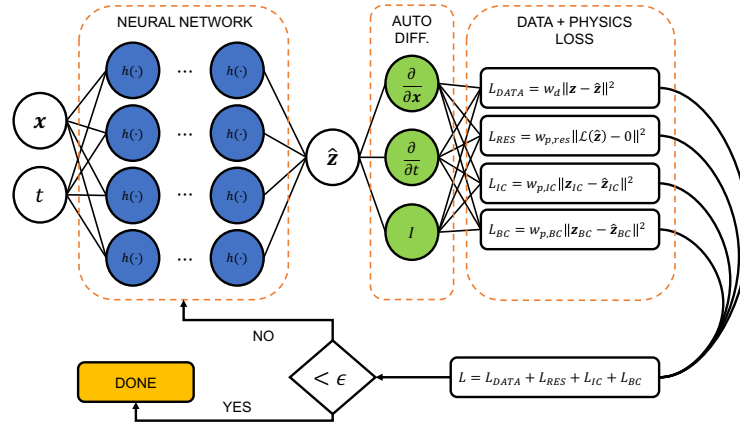


FIGURE 3.2: A PINN model, constructed by minimizing both data loss and physics loss functions.

In addition to the loss function of the NN model, the PINN method does not change any internal architecture. Therefore, the mathematical expression of the predictions from the PINN model is the same as that from the NN model:

$$\hat{\mathbf{z}}(\mathbf{x}, t; \boldsymbol{\mu}) = \hat{\mathbf{z}}(\mathbf{x}, t; \mathbf{W}, \mathbf{B}, \theta) = h(\dots h(\mathbf{W}(\mathbf{x}, t) + \mathbf{B})), \tag{3.3}$$

The PINN method is associated with a gradient pathology issue due to having multiple terms in its loss function (Chuang and Barba, 2022; Haghighat, Amini, and Juanes,

2021). The challenge is to balance the impact of each term in the training process by adjusting its weight (Chen et al., 2018). An imbalance in the impact of each term will result in an inaccurate learning process (Chen et al., 2018). I present this challenge with my case studies in Section 3.5.

The preprocessing in the PINN method is slightly different from that of the data-driven methods. The preprocessing of training samples here must be accompanied by the scaling of each term in the loss function (Amini, Haghighat, and Juanes, 2022; Haghighat, Amini, and Juanes, 2021). It is to ensure rapid convergence during the training process (Amini, Haghighat, and Juanes, 2022; Haghighat, Amini, and Juanes, 2021). An approach to scale terms in the loss function is presented in Amini, Haghighat, and Juanes (2022) and Haghighat, Amini, and Juanes (2021), following a dimensional analysis routine in the field of computational fluid dynamics. The steps are as follows (Amini, Haghighat, and Juanes, 2022; Haghighat, Amini, and Juanes, 2021; Ruzicka, 2008)

1. Define dimensionless term of input variables t and $\mathbf{x}$, and output variable $\mathbf{z}$
2. Scale the equation, starting from either boundary conditions or initial conditions
3. Find the scaling term for each variable and use it to perform preprocessing

For the construction of the PINN model, I use the SciANN library (Haghighat and Juanes, 2021).

3.2 Comparison strategy

In many geothermal applications, uncertainties arise from an inaccurate characterization of subsurface properties due to limited measurement data and incomplete knowledge of physical and chemical processes (Degen et al., 2023b; Degen, Cacace, and Wellmann, 2022; Degen, Veroy, and Wellmann, 2022; Schulte et al., 2020; Wellmann and Caumon, 2018). Parameter estimation studies are then essential to address the uncertainties caused by these properties towards the model's predictions (Degen et al., 2023b; Degen et al., 2021b). I introduce a *parameter estimation setting*, referring to a situation when I have a surrogate model that takes physical/chemical parameters and time as input and delivers solutions at specific locations (based on the discretization scheme), see Figure 3.3a. In this setting, I compare the NI-RB method as a process-based machine learning method and the NN-P method (an NN with parameter estimation setup) as a data-driven method.

Recently, the PINN method is widely used for Solid Earth applications (Amini, Haghighat, and Juanes, 2022; Haghighat, Amini, and Juanes, 2021; Waheed et al., 2021; Tartakovsky et al., 2020; He et al., 2020). If we look at those implementations, the PINN method is designed for state estimation purposes, not for parameter estimation purposes. Hence, I extend my comparison by introducing a *state estimation setting*, referring to a situation when I have a surrogate model that takes spatio-temporal coordinates as inputs and delivers solutions at any locations with a fixed set of physical/chemical parameters, see Figure 3.3b. The idea is based on the following premise: If I can efficiently construct the PINN model under a state estimation setting, it will also be efficient to extend it for a parameter estimation setting. Here, I compare the PINN method as the process-based machine learning method and the NN-S method (an NN with state estimation setup) as the data-driven method.

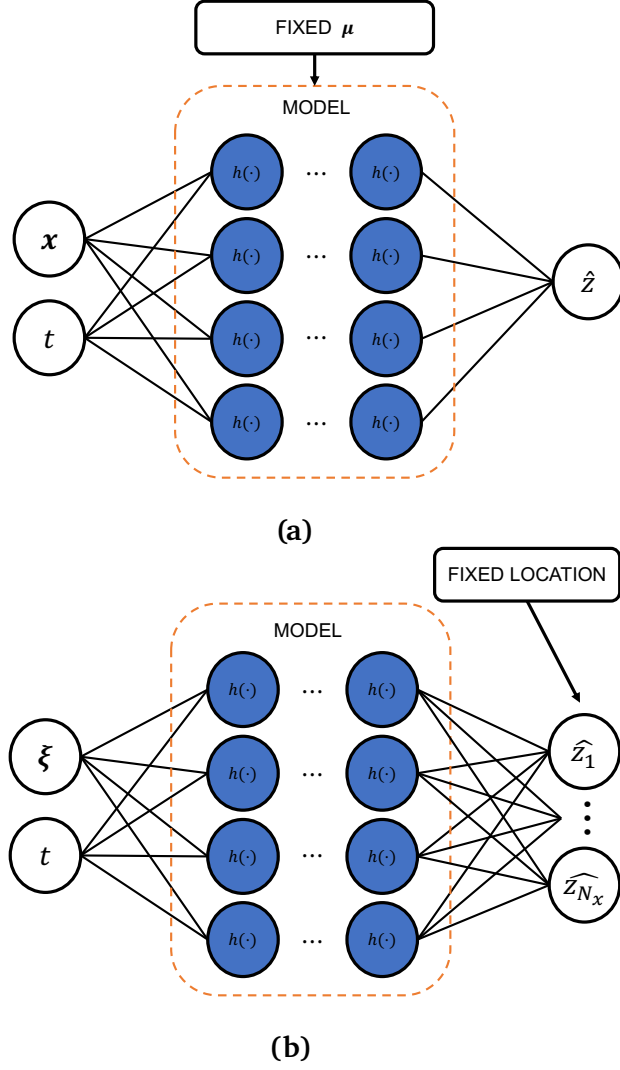


FIGURE 3.3: Comparison strategy for different process-based and data-driven machine learning methods: (a) parameter estimation setting and (b) state estimation setting.

A requirement before performing parameter estimation and optimization under uncertainties studies is to ensure that the surrogate model produces physically/chemically consistent predictions and preserves the original structure of the process-based model (Degen et al., 2023b). Here, I perform an additional comparison in which I use all FE nodal solutions as training samples and check whether the surrogate model can reproduce the physics or not. The idea is based on the premise for supporting the state-estimation-setting comparison: if any state-estimation-setting surrogate model, trained with all FE nodal solutions, can accurately reproduce the solutions, I can then be sure to extend it for a parameter estimation setting.

I compare different methods based on efficiency during the surrogate model construction, speedup at the prediction phase, and accuracy at the prediction phase. In addition, the relative error ζ is introduced to highlight the difference between the prediction of the surrogate model $\hat{z}(\mathbf{x}, t)$ and the FE solution $\mathbf{z}(\mathbf{x}, t)$, mathematically

written as follows:

$$\zeta = \frac{\|\mathbf{z}(\mathbf{x}, t, \xi) - \hat{\mathbf{z}}(\mathbf{x}, t, \xi)\|}{\|\mathbf{z}(\mathbf{x}, t, \xi)\|}, \quad (3.4)$$

3.3 Benchmarks

I consider two hydro-mechanical benchmark problems for comparing data-driven and process-based machine learning methods: the Mandel's problem and the Oedometer problem. These two benchmark problems are widely used to test the convergence and performance of hydro-mechanical solvers and algorithms, in addition to having rich physics that highlights tight coupling of hydraulic and mechanical processes (Haghighat, Amini, and Juanes, 2021; Cacace and Jacquy, 2017; Rutqvist, 2017; Jha and Juanes, 2014).

3.3.1 Mandel's problem

As the first case, I consider the Mandel problem that solves the fluid flow component (single phase and slightly compressible) in porous media, solid displacement due to loading and changes in pore pressure, and the strain-displacement relationship to describe a poroelasticity phenomenon (Haghighat, Amini, and Juanes, 2021; Borregales et al., 2019). The flow of fluid in porous media is in Equation 3.5], while the solid displacement with the strain-displacement relationship is in Equations 3.6 and 3.7 (Haghighat, Amini, and Juanes, 2021; Borregales et al., 2019).

$$\frac{1}{M} \frac{\partial P_f}{\partial t} + b \frac{\partial \varepsilon_v}{\partial t} - \nabla \cdot \left(\frac{\kappa}{\eta} \nabla P_f \right) = 0 \quad \text{in } \Omega \times I, \quad (3.5)$$

$$K_{dr} \nabla \varepsilon_v + \frac{1}{2} v^* K_{dr} \nabla (\nabla \cdot \mathbf{u}) + \frac{3}{2} v^* K_{dr} \nabla \cdot (\nabla \mathbf{u}) - b \nabla P_f = 0 \quad \text{in } \Omega \times I, \quad (3.6)$$

$$\varepsilon_v - \nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega \times I, \quad (3.7)$$

where M is the Biot bulk modulus, ϕ is the porosity of porous media, K_f is the fluid modulus, K_s is the solid modulus, b is the Biot coefficient, ε_v is the volumetric elastic strain, K_{dr} is the drained bulk modulus, v is the Poisson ratio, Υ is the Lamé's modulus, G is the shear modulus, κ is the permeability of porous media, η is the fluid viscosity, P_f is the pore pressure, and $\mathbf{u}$ is the solid displacement (for the Mandel's problem, I only solve for displacement in x-direction u_x and displacement in y-direction u_y). Some parameters are calculated as follows: (Haghighat, Amini, and Juanes, 2021; Borregales et al., 2019)

$$\frac{1}{M} = \frac{\phi}{K_f} + \frac{\phi - b}{K_s}, \quad (3.8)$$

$$b = 1 - \frac{K_{dr}}{K_s}, \quad (3.9)$$

$$v^* = \frac{1 - 2v}{1 + v}, \quad (3.10)$$

$$v = \frac{\Upsilon}{2(\Upsilon + G)}, \quad (3.11)$$

The computational domain and boundary conditions are illustrated in Figure 3.4.

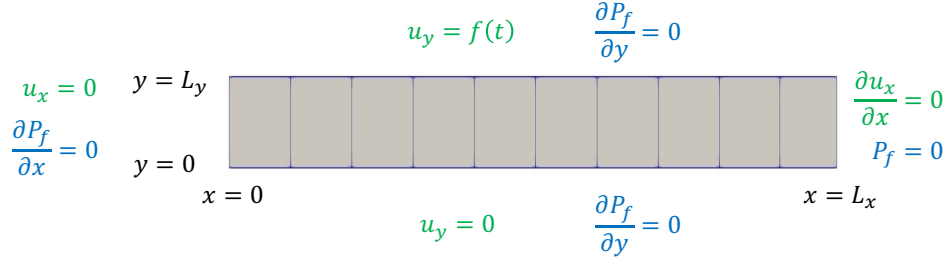


FIGURE 3.4: Computational domain and boundary conditions of the Mandel's problem. L_x is the maximum extent in x-direction, L_y is the maximum extent in y-direction.

The initial condition should satisfy the criterion below: (Borregales et al., 2019)

$$\frac{1}{M}P_f + b\nabla \cdot \mathbf{u} = 0, \quad (3.12)$$

The loading function at the top boundary is determined from the analytical solution of Mandel's problem itself (Haghighat, Amini, and Juanes, 2021; Borregales et al., 2019; Cacace and Jacquy, 2017).

$$f(t) = \frac{\sigma_0 L_y}{G} \left[\frac{-(1-v)}{2} + (1-v_u) \sum_{k=1}^{\infty} \frac{\sin \omega_k \cos \omega_k}{\omega_k - \sin \omega_k \cos \omega_k} \exp \left(\frac{-\omega_k c_r t}{L_x^2} \right) \right], \quad (3.13)$$

$$\tan \omega_k = \frac{1-v}{v_u - v} \omega_k, \quad (3.14)$$

$$v_u = \frac{3v + bC(1-2v)}{3 - bC(1-2v)}, \quad (3.15)$$

$$C = \frac{bM}{K_{dr} + b^2 M}, \quad (3.16)$$

$$c_r = \frac{\kappa}{\eta \left(\frac{1}{M} + \frac{b^2}{K_{dr}} \right)}, \quad (3.17)$$

where σ_0 is the initial stress and ω_k is the k-th solution of Equation 3.13. Using the Finite Element (FE) scheme, the physical model attempts to solve $\mathbf{u}(\mathbf{x}, t)$ and P_f . To solve this problem, I use the open source GOLEM simulator (?).

The Mandel's problem here has 2,112 degrees of freedom. The computational domain spans $[0.00, 1.00]$ m in the x-direction, $[0.00, 0.10]$ m and the y-direction, and $[0.00, 0.70]$ s in time. For the remainder of the discussion, I will refer to $t = 0 - 0.2$ s as the early time period and $t = 0.2 - 0.7$ s as the late time period. The model has ten elements in the x-direction, one element in the y-direction, and thirty timesteps. All parameters are listed in Table 3.1.

TABLE 3.1: Physical parameters ranges for Mandel's problem.

Input parameters			Lower bound	Upper bound	Blind case
Lame's modulus	Υ	[GPa]	0.40	0.60	0.49
Shear modulus G		[GPa]	0.70	0.80	0.71
Permeability κ		[m ²]	0.5×10^{-3}	2.0×10^{-3}	1.7×10^{-3}
Fluid's viscosity η		[Pa s]	0.5×10^{-3}	1.5×10^{-3}	0.7×10^{-3}
Porosity ϕ		[-]	0.05	0.15	0.12
Solid modulus K_s		[GPa]	2.00	3.00	2.29
Fluid modulus K_f		[GPa]	6.00	9.00	7.35

There is a non-linearity in Mandel's problem due to a strong coupling between fluid flow and solid mechanics. The most important non-linear feature is in the early-time period where a "jump", known as the Mandel-Cryer effect, exists in the pressure response (Jha and Juanes, 2014). For further discussion of Mandel's problem, the reader is encouraged to visit (Jha and Juanes, 2014). Based on the literature (Haghighat, Amini, and Juanes, 2021; Borregales et al., 2019; Cacace and Jacquety, 2017), the benchmark responses are the spatio-temporal pressure response and the temporal horizontal displacement response in $x = 1.0 \text{ m}$ and $y = 0.1 \text{ m}$. I then use these responses for my comparison.

3.3.2 Oedometer problem

As the second case, I consider the Oedometer problem that describes a 3-D poro-elastoplastic phenomenon with a single-phase and slightly compressible fluid flow in porous media (Cacace and Jacquety, 2017). It solves a similar set of equations like in Mandel's problem with additional plasticity terms. The fluid flow component with additional plasticity terms is written as follows (Cacace and Jacquety, 2017).

$$\frac{1}{M} \frac{\partial P_f}{\partial t} + b \frac{\partial \varepsilon_v}{\partial t} + (1-b) \frac{\partial \varepsilon_v^p}{\partial t} - \nabla \cdot \left(\frac{\kappa}{\eta} \nabla P_f \right) = 0 \quad \text{in } \Omega \times I, \quad (3.18)$$

$$K_{dr} \nabla \varepsilon_v + \frac{1}{2} v^* K_{dr} \nabla (\nabla \cdot \mathbf{u}) + \frac{3}{2} v^* K_{dr} \nabla \cdot (\nabla \mathbf{u}) - b \nabla P_f = 0 \quad \text{in } \Omega \times I, \quad (3.19)$$

$$\varepsilon_v - \nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega \times I, \quad (3.20)$$

$$\frac{\partial \varepsilon^p}{\partial t} = \dot{\gamma} \frac{\partial \mathcal{Q}}{\partial \boldsymbol{\sigma}'} \quad \text{in } \Omega \times I, \quad (3.21)$$

$$\varepsilon_v^p \approx \varepsilon_{xx}^p + \varepsilon_{yy}^p + \varepsilon_{zz}^p \quad \text{in } \Omega \times I, \quad (3.22)$$

where ε_v^p is the volumetric plastic strain, ε^p is the vector of the plastic strain, σ' is the vector of effective stress, $\dot{\gamma} = \dot{\gamma}(\sigma', \kappa)$ is the plastic multiplier that satisfies the classical Kuhn-Tucker conditions ($\dot{\gamma} \geq 0, \mathcal{F} \leq 0, \dot{\gamma}\mathcal{F} = 0$) with $\mathcal{F}(\sigma', \kappa)$ being the yield surface (Cacace and Jacquey, 2017). $\mathcal{Q} = \mathcal{Q}(\sigma', \kappa)$ is the plastic potential function giving the direction of the plastic strain increment and κ is the vector of internal variables that affect the evolution of the yield surface during the loading and unloading process (Cacace and Jacquey, 2017). Equation 3.21 is solved with the return map algorithm (Algorithm 4), considering the Drucker-Prager frictional plastic yield criterion (Cacace and Jacquey, 2017).

Algorithm 4: Return-map algorithm

1. At $k = 1$, solve the following equations for timestep $n + 1$ for the stress increment, internal parameter, and plastic multiplier

$$\begin{bmatrix} \left(\mathbb{C}^{-1} + \Delta\gamma_{n+1}^{(k)} \frac{\partial^2 \mathcal{Q}_{n+1}^{(k)}}{\partial \sigma'^2} \right) & \Delta\gamma_{n+1}^{(k)} \frac{\partial^2 \mathcal{Q}_{n+1}^{(k)}}{\partial \sigma' \partial \kappa} & \frac{\partial \mathcal{Q}_{n+1}^{(k)}}{\partial \sigma'} \\ \Delta\gamma_{n+1}^{(k)} \frac{\partial \mathcal{H}_{n+1}^{(k)}}{\partial \sigma'} & \left(\mathbb{K} + \Delta\gamma_{n+1}^{(k)} \frac{\partial \mathcal{H}_{n+1}^{(k)}}{\partial \kappa} \right) & \mathcal{H}_{n+1}^{(k)} \\ \frac{\partial \mathcal{F}_{n+1}^{(k)}}{\partial \sigma'} & \frac{\partial \mathcal{F}_{n+1}^{(k)}}{\partial \kappa} & 0 \end{bmatrix} \begin{bmatrix} \Delta\sigma'_{n+1}^{(k)} \\ \Delta\kappa_{n+1}^{(k)} \\ \Delta^2\gamma_{n+1}^{(k)} \end{bmatrix} = \begin{bmatrix} -\Delta\varepsilon_{n+1}^p + \Delta\gamma \frac{\partial \mathcal{Q}_{n+1}^{(k)}}{\partial \sigma'} \\ \Delta\kappa_{n+1} - \Delta\gamma \mathcal{H} \\ -\mathcal{F}(\sigma', \kappa)_{n+1}^{(k)} \end{bmatrix}$$

where $\mathbb{C} = \Upsilon \delta_{ij} \delta_{kl} + 2G \delta_{ik} \delta_{jl}$ is the rank-four elastic stiffness tensor, $\mathcal{H}$ is the hardening flow rule for κ , subscript n refers to t_n , subscript $n + 1$ refers to t_{n+1} , superscript k refers to k -th iteration

2. Update the solutions for $k + 1$ iteration step

$$\begin{aligned} \sigma'_{n+1}^{(k+1)} &= \sigma'_{n+1}^{(k)} + \Delta\sigma'_{n+1}^{(k)} \\ \varepsilon_{n+1}^p{}^{(k+1)} &= \varepsilon_{n+1}^p{}^{(k)} - \mathbb{C}^{-1} : \Delta\sigma'_{n+1}^{(k)} \\ \Delta\gamma_{n+1}^{(k+1)} &= \Delta\gamma_{n+1}^{(k)} + \Delta^2\gamma_{n+1}^{(k)} \end{aligned}$$

3. Repeat step 1 and 2 for $k + 1$ iteration step until the right-hand-side matrix in step 1 reaches minimum threshold values.
-

Within the Drucker-Prager yield criterion, $\mathcal{F}$ also depends on the Mohr-Coulomb friction angle Φ and cohesion C_o (Cacace and Jacquey, 2017). While $\mathcal{Q}$ is also influenced by the dilation angle ψ by implementing the non-associated form of Drucker-Prager (Cacace and Jacquey, 2017). Subscription xx denotes principal x-direction, yy principal y-direction, and zz principal z-direction. In this case, I solve for the pore pressure P_f , displacement in the x direction u_x , displacement in the y direction u_y and displacement in the z direction u_z . These variables are set to zero at the initial timestep. The computational domain and boundary conditions are illustrated in Figure 3.5.

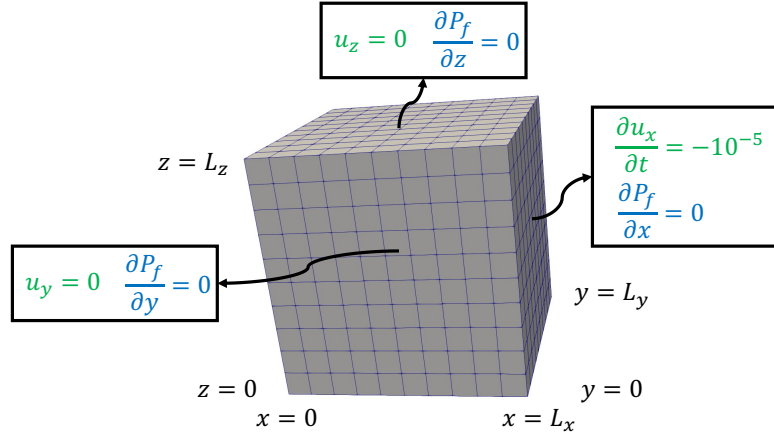


FIGURE 3.5: Computational domain and boundary conditions of the Oedometer problem. L_x is the maximum extent in x-direction, L_y is the maximum extent in y-direction, and L_z is the maximum extent in z-direction. For the flow component, all faces have no-flow boundary condition towards its normal vector direction: $\frac{\partial P_f}{\partial x} = 0$ at left and right, $\frac{\partial P_f}{\partial y} = 0$ at front and back, and $\frac{\partial P_f}{\partial z} = 0$ at top and bottom. For the solid component, all faces have Dirichlet-type boundary condition (set to be zero) at their associated directions (except the right face): $u_x = 0$ at left, $u_y = 0$ at front and back, and $u_z = 0$ at top and bottom. The negative value at right boundary denotes compression.

The Oedometer problem has 5,324 degrees of freedom. The computational domain spans $[0.0, 1.0]$ m in the x direction, $[0.0, 1.0]$ m in the y direction, $[0.0, 1.0]$ m in the z direction, and $[0.0, 200.0]$ s in time. The model has ten elements in the x direction, ten elements in the y direction, ten elements in the z direction, and fifty timesteps. All parameters are listed in Table 3.2. To solve this problem, I use the open source GOLEM simulator (?).

TABLE 3.2: Physical parameters ranges for the Oedometer problem.

Input parameters	Lower bound	Upper bound	Blind case
Biot bulk modulus M [GPa]	3.50	4.50	3.76
Shear modulus G [GPa]	3.50	4.00	3.75
Permeability κ [m ²]	1.0×10^{-15}	1.0×10^{-13}	7.8×10^{-14}
Porosity ϕ [-]	0.10	0.20	0.15
Solid modulus K_s [GPa]	6.00	8.00	7.81
Cohesion C_o [GPa]	1.0×10^{-3}	1.2×10^{-3}	1.0×10^{-3}
Friction angle Φ [°]	20	30	23
Dilation angle ψ [°]	0	10	10

A nonlinear and iterative plastic strain calculation (reflected from the use of return-map algorithm) brings more complexity to solve the Oedometer problem, compared to Mandel's problem. For further discussion on the Oedometer problem, readers are encouraged to visit (Cacace and Jacquey, 2017). Based on the literature (Cacace and Jacquey, 2017), the benchmark responses are spatio-temporal horizontal displacement and pressure response. However, since the horizontal displacement response is constant with changing physical parameter values, I do not use it in my comparison. Only the temporal pressure response at $x = 0.5 \text{ m}$, $y = 0.5 \text{ m}$, and $z = 0.5 \text{ m}$ is used for my comparison, since the pressure response is constant over space.

3.4 Results

In this section, I present the challenges of constructing the surrogate model and compare the performance of different surrogate models. The discussion begins by comparing a data-driven machine learning method (NN-P) and a process-based machine learning method (NI-RB) in the parameter-estimation setting, emphasizing the importance of this setting for uncertainty quantification study in geothermal applications. Since the PINN method is not designed for parameter-estimation setting (Raissi, Perdikaris, and Karniadakis, 2019; Raissi and Karniadakis, 2018), its implementation for uncertainty quantification study is questionable. To prove this point, I continue with a state-estimation setting to extend the comparison for the PINN method. I will also discuss in more detail potential solutions of the PINN method for parameter-estimation setting in Section 3.5. All comparisons utilize the blind case in Table 3.1 (the Mandel's problem) and in Table 3.2 (the Oedometer problem). I present a diagram in Figure 3.6 showing all analyses performed with both Mandel's and Oedometer problem for a better overview.

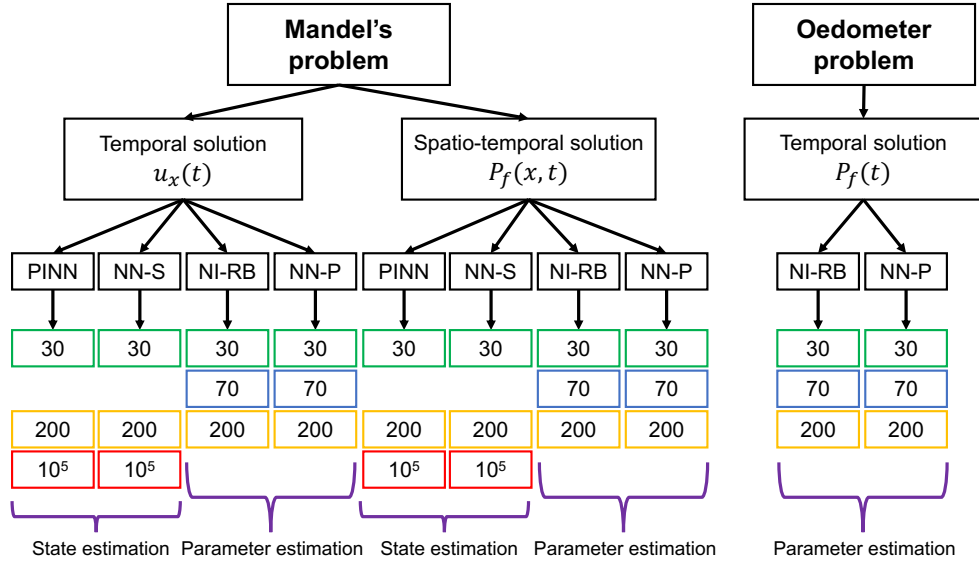


FIGURE 3.6: All comparisons among different data-driven machine learning methods and process-based machine learning methods under multiple settings for both Mandel's and Oedometer problem. The number inside the box denotes the number of training samples used for each method used in the comparison.

In the parameter-estimation setting, I compare the NN-P method and the NI-RB method with an increasing number of training samples, from 30 to 200 samples for

both Mandel's and Oedometer problem. I specifically show the comparison on 30, 70 and 200 samples. Thirty training samples can be considered small for benchmark problems having analytical solutions, while two hundred training samples are considered large enough, and seventy training samples are moderate. Note that the bounds for the number of training samples are smaller than the degree of freedom of both Mandel's and Oedometer problem. In the state-estimation setting, I have to extend the number of training samples to 100,000 samples since the PINN method has not delivered accurate predictions yet, and I reach the RAM memory limit for an Apple Macbook Pro. I need a bigger computing infrastructure to go beyond 100,000 samples for the PINN model training. Note that 100,000 samples are already larger than the degree of freedom of both Mandel's and Oedometer problems. The prediction accuracy of the NN-S method and the PINN method is compared to an increasing number of training samples, from 30 to 100,000 samples. I here show the comparison at 30, 200, and 100,000 samples for only Mandel's problem. I do not construct a surrogate model for the Oedometer problem since it is extremely complex to include the return-map algorithm with the Drucker-Prager criterion (Algorithm 2) in the loss function. This algorithm requires an iterative procedure and cannot be easily assigned in the loss function (the same issue is observed in solid mechanics with plastic deformation in (Haghighat et al., 2021)). A further approach is needed, which is not the scope of this dissertation. Here, we can directly see the impracticality of the PINN method. I also acknowledge that I can treat this complex return-map algorithm as "hidden/missing physics" so that I can run the PINN method (Raissi, Perdikaris, and Karniadakis, 2019; Raissi and Karniadakis, 2018). The terms can be omitted from the loss functions and compensated for by adding more training samples. However, it is again not the scope of this dissertation and it is beyond the RAM memory limit of an Apple Macbook Pro. For the running-time comparison, I show the offline time (time needed to construct the surrogate model) and the online time (time needed by the surrogate model to predict a new case).

3.4.1 Parameter-estimation setting

The NI-RB surrogate model provides more accurate predictions than the NN-P surrogate model for temporal responses with a small number of training samples. It is proven for both Mandel's and Oedometer problems in Figure 3.7a, 3.7c, 3.8a, and 3.8b. In a small number of training samples ($N_s = 30$), non-physical artifacts (in the form of small-scale perturbations) appear in the prediction of the NN-P surrogate model (caused by frequent overshooting and undershooting), while the NI-RB surrogate model can capture the solution with a slight deviation (Figure 3.7a). Furthermore, with increasing number of training samples, the accuracy of both surrogate models increases and their predictions are on top of each other (Figures 3.7a and 3.8a). However, the NI-RB model has always higher accuracy than the NN-P model, see the relative error curves in Figure 3.7c and 3.8b.

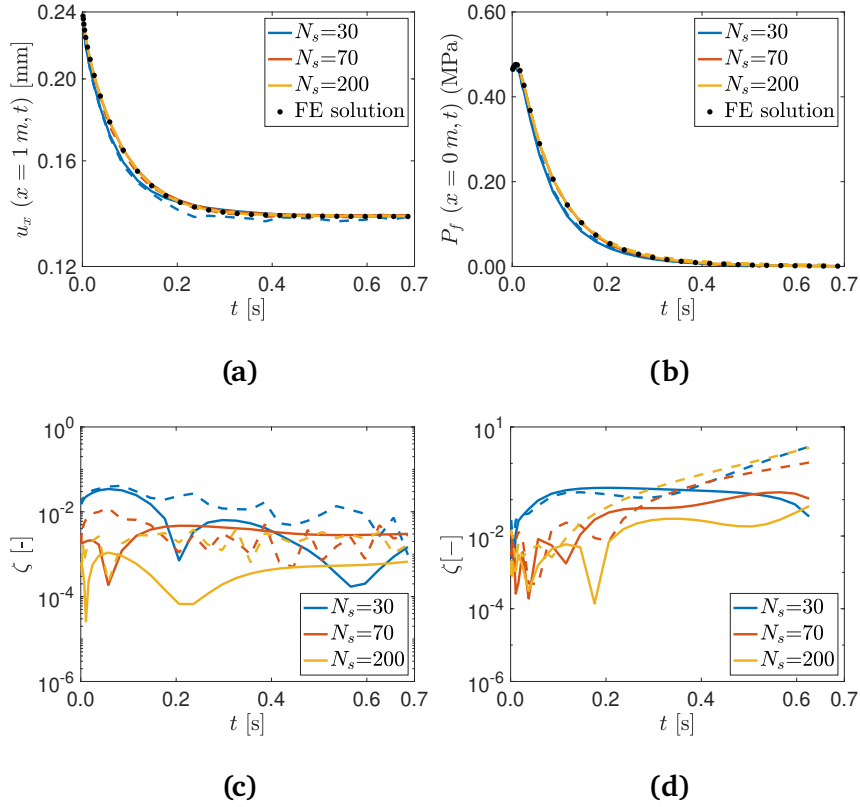


FIGURE 3.7: Comparison among surrogate models constructed by the NN-P and the NI-RB in terms of (a) $u_x(t)$, (b) $P_f(x=0\text{ m}, t)$, (c) relative error for $u_x(t)$, and (d) relative error for $P_f(x=0\text{ m}, t)$, utilizing Mandel's problem. The dashed lines "—" denote the NN-P predictions, and the solid lines denote the NI-RB predictions.

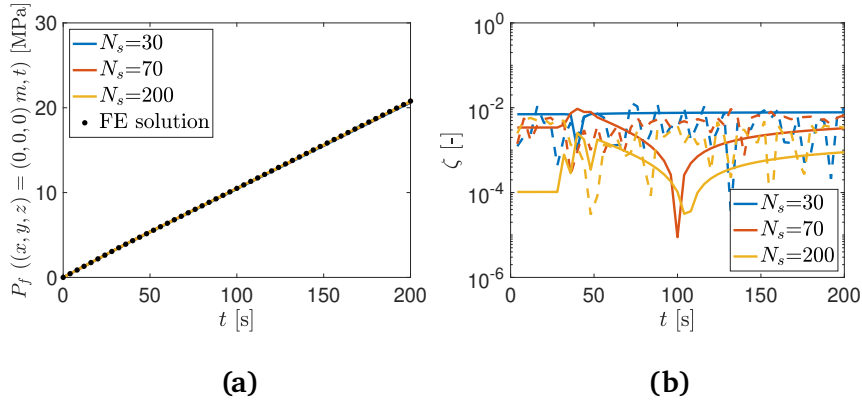


FIGURE 3.8: Comparison among surrogate models constructed by the NN-P and the NI-RB in terms of (a) $P_f(t)$ and (b) the relative error, utilizing the Oedometer problem. The dashed lines "—" denote the NN-P predictions, and the solid lines denote the NI-RB predictions.

For spatio-temporal responses, the NI-RB surrogate model is also more effective than the NN-P surrogate model. The NI-RB surrogate model can already capture the "jump" in the early time period without losing accuracy in the late time period with a small number of training samples (Figure 3.7b). However, the NN-P surrogate model loses precision in either the early time period or the late time period with a small number of training samples (Figure 3.7d). As the number of training samples increases,

both models' accuracy increases. However, the NN-P surrogate model always has an accuracy problem in the late time period (Figure 3.7d).

3.4.2 State-estimation setting

The PINN method requires an extremely large number of training samples, larger than the degrees of freedom of Mandel's problem, to produce accurate predictions for temporal responses. At a small number of training samples ($N_s = 30$), the PINN surrogate model can capture the shape of the FE solution (even though with low accuracy), while the prediction of the NN-S model is completely off. However, the PINN method requires at least $N_s = 10^5$ to start producing accurate solutions. We can clearly observe this behavior in Figure 3.9a and 3.9c. The same number of training samples is also needed for the NN-S surrogate model to achieve high accuracy, see Figure 3.9a and 3.9c.

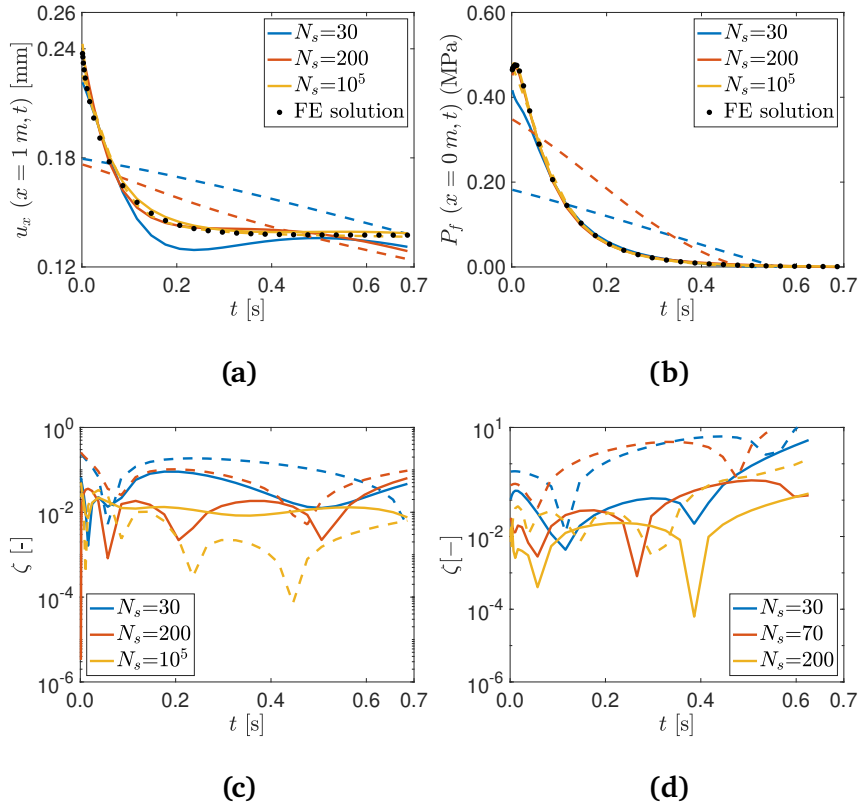


FIGURE 3.9: Comparison among surrogate models constructed by the NN-S and the PINN in terms of (a) $u_x(t)$, (b) $P_f(x = 0 m, t)$, (c) relative error for $u_x(t)$, and (d) relative error for $P_f(x = 0 m, t)$, utilizing Mandel's problem. The dashed lines "—" denote the NN-S predictions and the solid lines denote the PINN predictions.

For spatio-temporal responses, similar behavior like in the temporal responses is observed for both the PINN and the NN-S surrogate models. In a small number of training samples, the PINN model can capture the shape of the solution, while the NN-S model is completely off. The PINN model can capture the "jump" in the early-time period (Figure 3.7b). However, both the PINN and NN-S surrogate models need at least $N_s = 10^5$, greater than the degree of freedom of Mandel's problem, to be able to accurately capture the response.

Since I see in the previous results that the PINN method requires an extremely large number of training samples, bigger than the degree of freedom of Mandel's problem, to deliver accurate predictions, I would like to further investigate whether the PINN surrogate model can reproduce solutions of Mandel's problem with the solutions themselves (the whole FE nodal solutions). I can observe in Figure 3.10 that the PINN surrogate model has the lowest accuracy. It shows that the PINN method requires more training samples than the whole FE nodal solutions (the degree of freedom) to be able to accurately reproduce the solution. On the other hand, the NI-RB surrogate model produces accurate predictions and maintains the structure of Mandel's problem.

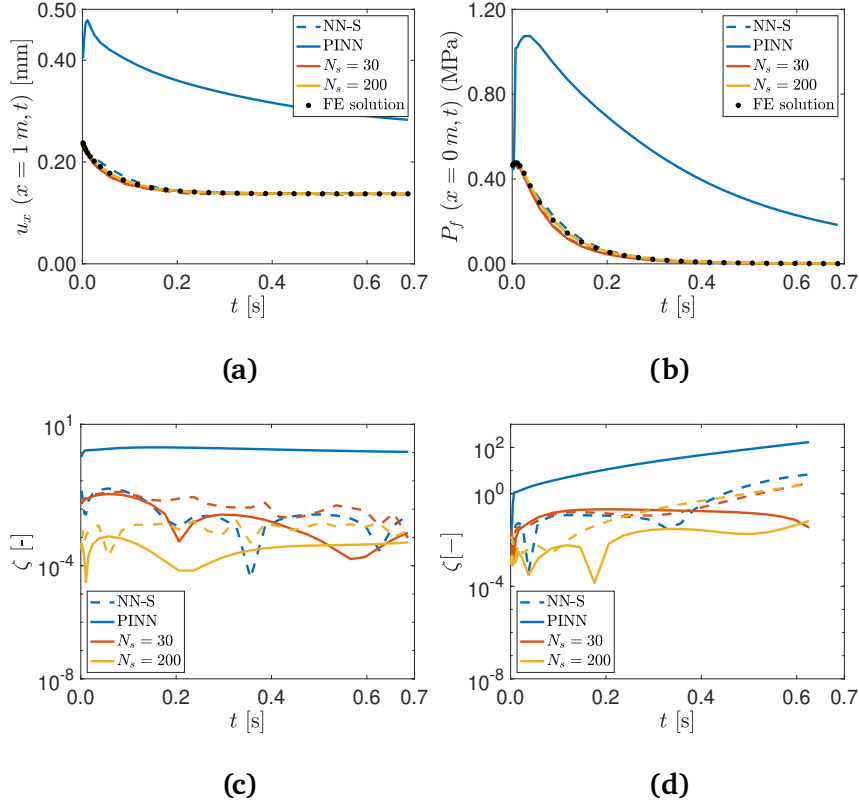


FIGURE 3.10: Comparison among surrogate models constructed by the NN-S, the PINN, the NN-P, and the NI-RB in terms of (a) $u_x(t)$, (b) $P_f(x = 0 m, t)$, (c) relative error for $u_x(t)$, and (d) relative error for $P_f(x = 0 m, t)$, utilizing the whole FE nodal solutions of Mandel's problem as training samples. The ones with orange and yellow color are constructed for parameter-estimation setting with the following linestyle nomenclatures: the dashed lines "—" denote the NN-P predictions and the solid lines denote the NI-RB predictions.

3.4.3 Offline and online time

The PINN method has the longest offline time (time required to train a NN model with an optimum set of hyperparameters), as can be seen in Figure 3.11a. This long offline time is because the PINN method requires a significant amount of time to compute its complex loss function during the back-propagation process (huge computational graph loads caused by the implementation of automatic differentiation and weight-tuning process for each term in the loss function) (Chuang and Barba, 2022). However, the NN-S, NN-P and NI-RB methods have similar offline time (Figure 3.11a and 3.11b), depending on the optimum combination of hyperparameters.

All methods have similar online time (time needed to predict a new case with a trained NN model), with the PINN method being the slowest, see Figures 3.11a and 3.11b. The PINN method must perform the most matrix multiplication operations because it has the deepest and biggest network architecture. On the other hand, the NI-RB method performs the least matrix multiplication operations.

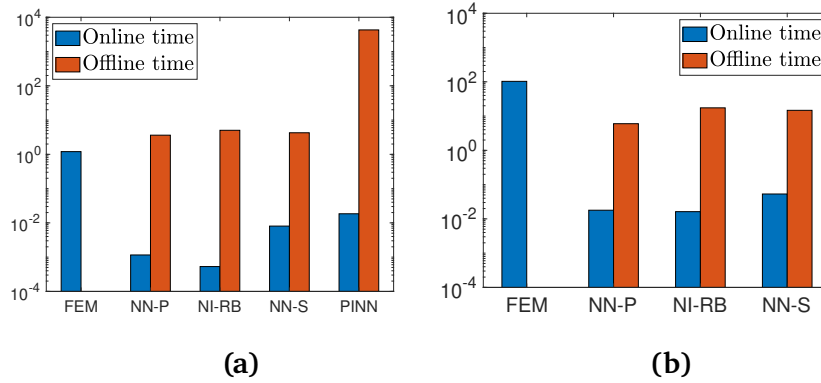


FIGURE 3.11: Offline and online time for different surrogate models utilizing (a) Mandel's problem and (b) the Oedometer problem.

3.5 Discussions

The NI-RB method is suitable for parameter estimation and optimization under uncertainties studies in geothermal applications as it produces accurate predictions with a small number of training samples, maintains the original structure of process-based models, and provides significant speed-up for predicting a new case. It is proven using both Mandel's and Oedometer problem. There are also several studies to further prove the reliability and robustness of the NI-RB method in geothermal applications: uncertainty quantification for the thermo-hydro-mechanical (THM) process during a hydrofracture treatment in Groß Schönebeck, Germany (Degen, Cacace, and Wellmann, 2022), capture of convection (thermo-hydro process) in geothermal reservoirs (Degen et al., 2023b), modeling salt dome development with geodynamics simulation (thermo-mechanical process) (Degen et al., 2023b). Several studies with the intrusive version of the NI-RB method (I-RB method) can also demonstrate its robustness (NI-RB and I-RB methods are fundamentally similar; the difference lies in the calculation of the coefficients: the NI-RB method utilizes a machine learning model, while the I-RB method utilizes the Galerkin projection): uncertainty quantification for thermal process in Upper Rhine Graben, Germany (Degen et al., 2021a), Alpine region (Degen et al., 2021b), and Berlin-Brandenburg, Germany (Degen, Veroy, and Wellmann, 2022). Compared to the data-driven method, the NI-RB method is more efficient, since the data-driven method demands a large number of training samples to achieve high accuracy. The data-driven method produces non-physical artifacts in its predictions in a small number of training samples due to overshooting and undershooting.

The PINN method, originally designed for state estimation problems, is not suitable for parameter estimation and optimization under uncertainties studies in Solid Earth applications. I acknowledge several approaches to tune the PINN surrogate model for parameter estimation problems, such as embedding the physical parameters into the loss function and updating the parameters throughout the training process (Zhang et al., 2022; Yin et al., 2021; Cai et al., 2021; Zhang, Yin, and Karniadakis, 2020; Chen

et al., 2020), modifying the PINN surrogate model into the Bayesian PINN (B-PINN) surrogate model (Psaros et al., 2023; Linka et al., 2022; Yang, Meng, and Karniadakis, 2021), modifying the NN architecture within the PINN method into the Generative Adversarial Network (GAN) architecture (Psaros et al., 2023; Meng et al., 2022; Yang, Zhang, and Karniadakis, 2020). However, these modified PINN approaches carry a huge computational burden (Yang, Meng, and Karniadakis, 2021) because they need to train the NN model for every new set of parameters/hyperparameters during the uncertainty quantification process. Furthermore, they also demand an extremely large number of training samples (Yang, Meng, and Karniadakis, 2021) to ensure that every trained NN model produces physically/chemically consistent predictions (Chuang and Barba, 2023), so that the NN model error (primarily caused by insufficient training samples) will not spill into the uncertainty measure. I further show these points in my results under a state-estimation setting. I show that the PINN method requires an extremely large number of training samples, larger than the degree of freedom of a finite element formulation (in my case, the Mandel problem), to achieve accurate predictions. The PINN method also requires a long offline phase, as it has to perform automatic differentiation to calculate its complex loss function and adjust the weights for each term in its loss function (Chuang and Barba, 2022). If the use of PINN surrogate model for parameter estimation problems is based on its original design for state estimation problems, any parameter estimation and optimization under uncertainties studies with the PINN surrogate model will not be efficient.

Another problem that arises during the construction of the PINN surrogate model is how to optimally tune the weights for each term in the loss function. I further illustrate this issue in Figure 3.12 using Mandel's problem. Suboptimal weight adjustment in the loss function can result in a local minimum during training (Chuang and Barba, 2023; Chuang and Barba, 2022; Beucler et al., 2021; Márquez-Neila, Salzmann, and Fua, 2017). For example, to achieve a small global error, the solid displacement component might be weighted more than the fluid flow component. This weighting scheme causes significant errors in the pressure response, while achieving high accuracy in the displacement response. We can notice that changing these weights has a different effect on the predictions $u_x(t)$ and $P_f(x = 0 \text{ m}, t)$. For example, applying $w_p = 0.5$ improves the accuracy of the prediction of $u_x(t)$ (Figure 13a) but lowers the accuracy of $P_f(x = 0 \text{ m}, t)$ (Figure 13b). The adaptive weights algorithm (Haghighat and Juanes, 2021) does not produce accurate predictions either (it only produces accurate prediction on $u_x(t)$ but inaccurate prediction on $P_f(x = 0 \text{ m}, t)$ at late-time period). I believe that a complex tuning scheme is required in future studies to adapt the PINN method for nonlinear multiphysics problems. Considering this issue, the PINN method is rendered impractical and inefficient.

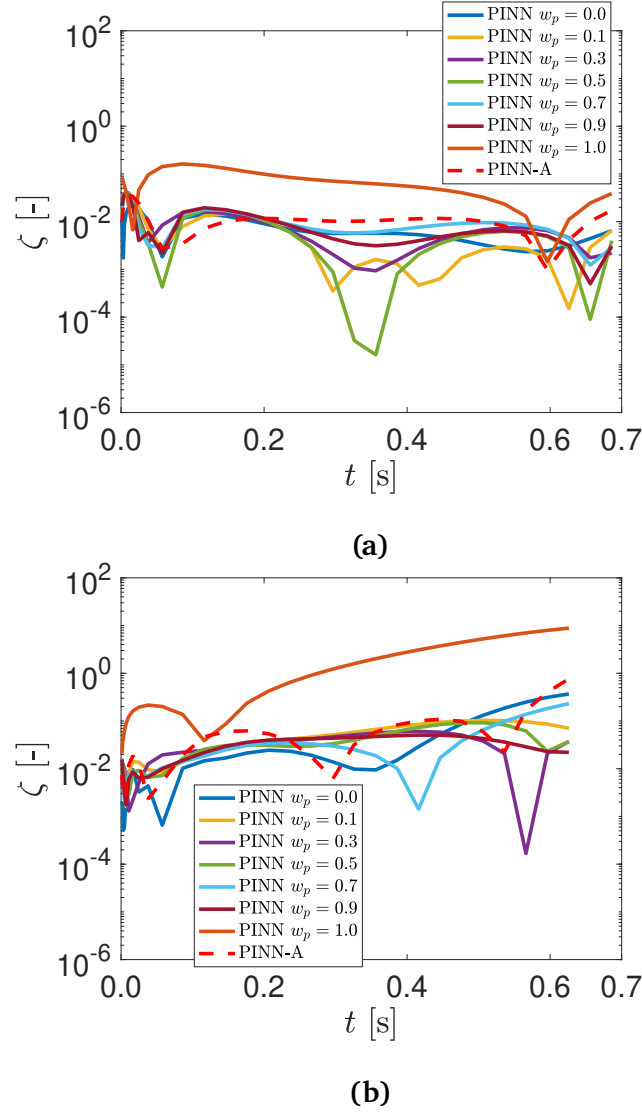


FIGURE 3.12: The impact of different weights towards the relative error of (a) $u_x(t)$ and (b) $P_f(x=0 m, t)$, utilizing the Mandel's problem. w_p denotes the weights for the physics loss in the loss function; PINN-A denotes the use of PINN with an adaptive weight's algorithm (?).

Compared to the data-driven method, the PINN method shows better performance in a small number of training samples. The PINN surrogate model can capture the shape of the physical solution (including "jump" features), while the data-driven surrogate model is completely off.

When comparing all data-driven and process-based machine learning methods in the context of parameter estimation and optimization under uncertainties studies in geothermal applications, the most suitable method is the NI-RB method. The NI-RB method requires a small number of training samples to produce accurate predictions and maintains the original structure of the process-based model. This further helps us reduce the computational cost of running high-dimensional process-based models, which are commonly used in geothermal applications. This reduction in computational cost cannot be achieved with data-driven machine learning methods and the PINN method. The NI-RB method also provides a significant speed-up in predicting a new case that is beneficial for enabling large-scale parameter estimation and optimization under uncertainties studies.

Chapter 4

The non-intrusive reduced basis method for well targeting in geothermal applications - Insights from The Hague, Netherlands

*"As far as the laws of mathematics refer to reality, they are not certain.
And as far as they are certain, they do not refer to reality."*
— Albert EINSTEIN (1922)

The objective of this chapter is to evaluate the potential of the proposed producer (targetted well) location for heat extraction processes in The Hague, Netherlands. This assessment involves quantifying the temperature range at the producer under a steady-state thermal process, considering uncertainties in thermal rock properties due to sparse measurement data. Parameter estimation studies are performed, including global sensitivity analysis (GSA) and uncertainty quantification (UQ), using the regional geological model of The Hague, Netherlands. Here, I showcase the application of the non-intrusive reduced-basis (NI-RB) method to facilitate these large-scale parameter estimation studies. Additionally, this chapter serves as a preliminary test of the NI-RB method before its further implementation in a transient thermo-hydraulic setting in The Hague model. This chapter is covered in Santoso et al. (2023a).

4.1 The Hague, Netherlands

My central case study is located in The Hague, Netherlands, focusing on a deep geothermal installation project that aims to heat 6000 houses in the Den Haag Zuidwest district (Mottaghy, Pechnig, and Vogt, 2011; Vogt et al., 2010). The objective is to optimize operating conditions, taking into account uncertainties and safety measures, to achieve the target power output of 5 MWt (Mottaghy, Pechnig, and Vogt, 2011). The regional geological model and the reservoir model used in my study were developed in collaboration with Geophysica Beratungsgesellschaft mbH, Germany (Mottaghy, Pechnig, and Vogt, 2011).

4.1.1 Regional geological model

The Hague region is located in the West Netherlands Basin (Willems et al., 2020; Mijndieff, 2020), as illustrated in Figure 4.1a. The regional geological model of The Hague spans 22.5 km east-west, 24.3 km north-south, and has a thickness of 5 km (Mottaghy,

Pechnig, and Vogt, 2011). Geological layers and characteristics are identified by Simmelink, Vandeweyer, and Ramaekers (2007), while rock properties are determined by well logs, cuttings, and borehole breakouts by Mottaghy, Pechnig, and Vogt (2011).

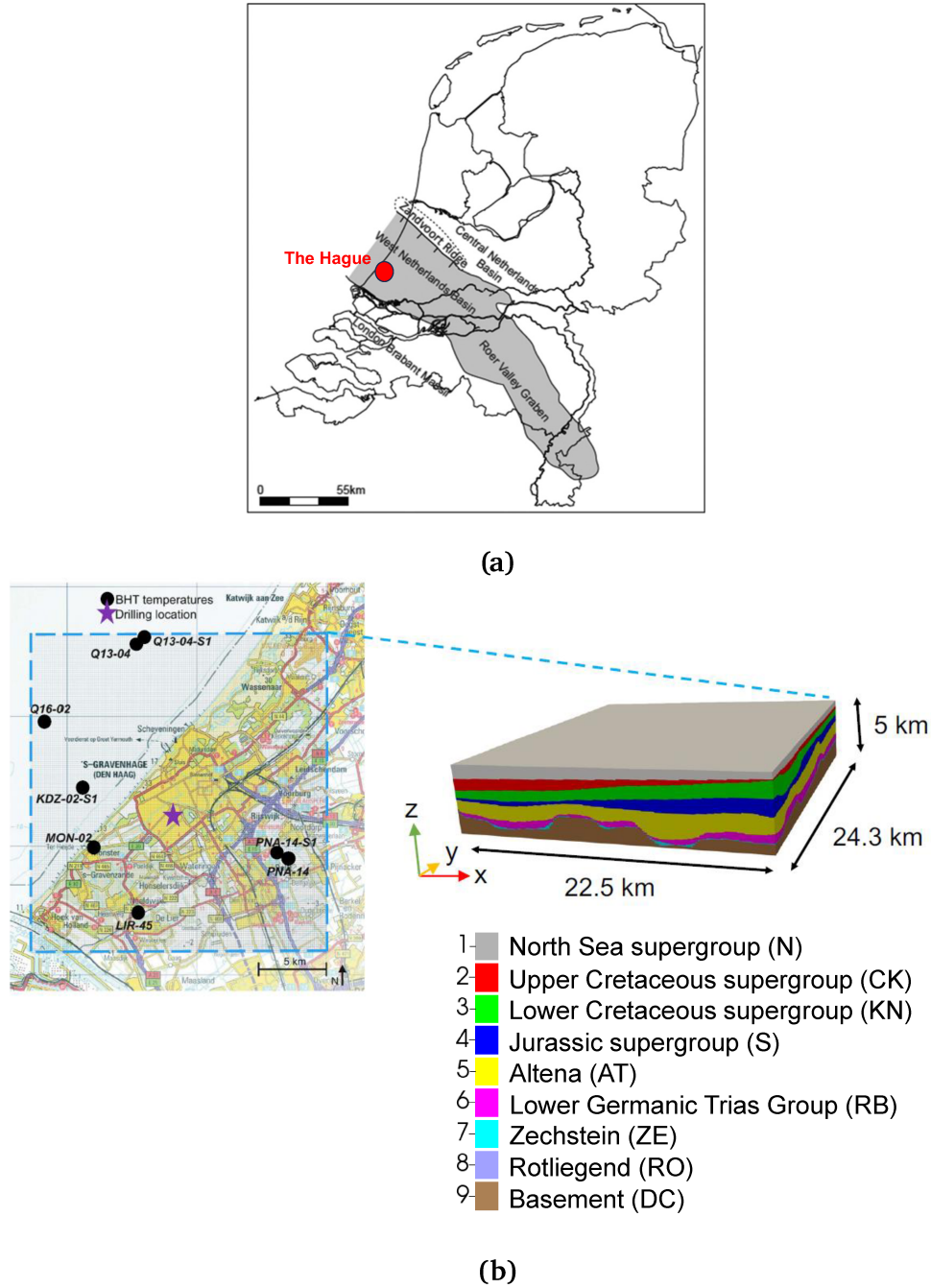


FIGURE 4.1: The case study of The Hague, Netherlands. (a) shows the location of The Hague region in West Netherlands Basin, marked with a red dot. The figure is adopted from Willems et al. (2020) and modified by adding the location of The Hague. (b) shows my regional geological model coverage and geological layer description. The purple star denotes the proposed drilling location and the black dots are well data used for constructing the regional geological model. The figure is adopted from Mottaghy, Pechnig, and Vogt (2011) and modified. In the model, the positive x-axis corresponds to the east direction, and the positive y-axis corresponds to the north direction.

The regional geological model has 3,704,659 nodes and 3,402,000 elements with

hexahedral grid (HEX8). The positive x-axis of the model corresponds to the east direction, the positive y-axis corresponds to the north direction, and the positive z-axis points to the surface. The geological description and rock properties for each layer are in Table 4.1.

The Hague region is originally conductive before any heat extraction activities (Mottaghy, Pechnig, and Vogt, 2011; Vogt et al., 2010). Hence, the uncertain parameters considered on this scale are solid thermal conductivities and heatflow in the regional model (Mottaghy, Pechnig, and Vogt, 2011). The heatflow is in the range of $0.06 - 0.07 \text{ W/m}^2$ (Vogt et al., 2010). The region was originally fully saturated with water, having $1,000 \text{ kg/m}^3$ density, $0.65 \text{ W/(m} \cdot \text{K)}$ thermal conductivity, and $4,193.5 \text{ J/(kg} \cdot \text{K)}$ heat capacity (Mottaghy, Pechnig, and Vogt, 2011).

TABLE 4.1: Geological description and rock properties of The Hague regional model.

Layers (top to bottom)	Solid density [kg/m^3]	Solid heat capacity [$\text{J/(kg} \cdot \text{K)}$]	Solid thermal conductivity [$\text{W/(m} \cdot \text{K)}$]	Porosity [-]	Permeability [m^2]
North Sea Supergroup (N)	2,570	960	1.8 - 4.0	1×10^{-3}	1×10^{-17}
Upper Cretaceous Supergroup (CK)	2,570	960	1.7 - 3.0	1×10^{-3}	1×10^{-17}
Lower Cretaceous Supergroup (KN)	2,570	960	1.8 - 3.7	1×10^{-3}	1×10^{-17}
Jurassic Supergroup (S)	2,570	960	2.3 - 4.9	1×10^{-3}	1×10^{-17}
Altena (AT)	2,570	960	1.6 - 2.7	1×10^{-3}	1×10^{-17}
Lower Germanic Trias Group (RB)	2,570	960	1.5 - 4.7	1×10^{-3}	1×10^{-17}
Zechstein (ZE)	2,570	960	1.7 - 5.0	1×10^{-3}	1×10^{-17}
Rotliegend (RO)	2,570	960	2.0 - 4.8	1×10^{-3}	1×10^{-17}
Basement (DC)	2,570	960	1.5 - 3.9	1×10^{-3}	1×10^{-17}

As depicted in Figure 4.1b, Mottaghy, Pechnig, and Vogt (2011) used six wells, consisting of 10 measurement points, to calibrate the regional models. The data pertain to the original (unperturbed) temperature within the region (Bonté, Van Wees, and Verweij, 2012). I supplement this calibration effort by collecting additional measurement data (in total 32 wells, consisting of 105 measurement points), which will be

used for calibration in Chapter 4. Figure 4.2 presents all the measurement data within the coverage of the regional model along with their acquisition years, adopted from Bonté, Van Wees, and Verweij (2012). The measurement error reported by Bonté, Van Wees, and Verweij (2012) is $\pm 5 - 10$ °C, reflecting corrections for the flow in the wellbore during measurement and extrapolations to the original temperature using the Horner method.

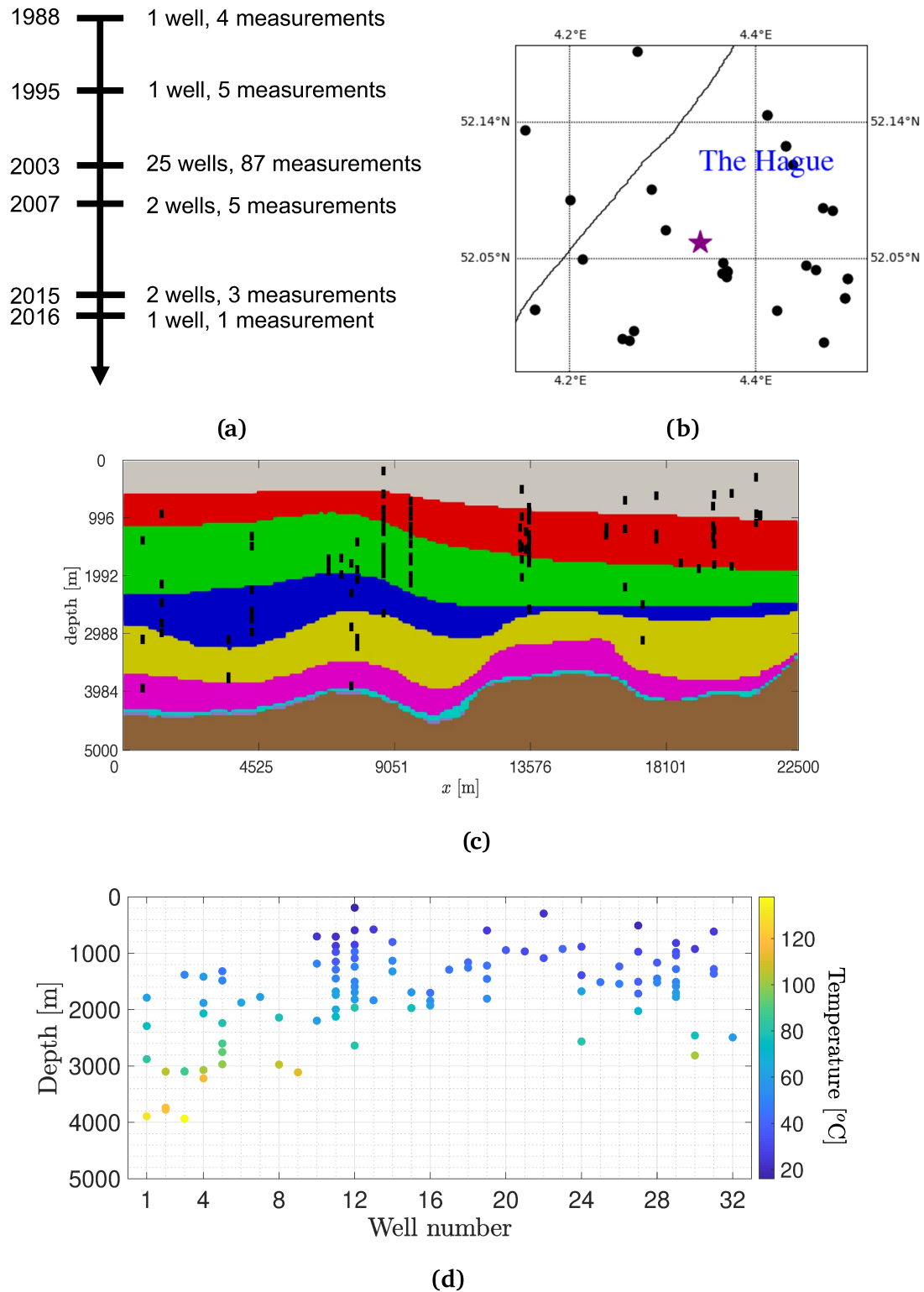


FIGURE 4.2: Measurement data within the regional geological model. (a) shows the acquisition years, (b) shows the location of measurement points in xy-axis, (c) shows the location of measurement points in xz-axis, and (d) shows the measurement data for each well at different depth and their values. The well names associated with different well numbers are in Table 4.2.

TABLE 4.2: Well names for different well numbers in Figure 4.2d.

Well number	Well name	Well number	Well name
1	LIR-45	17	MED-01
2	KDZ-02/S1	18	MED-02
3	Q16-02	19	OLE-01
4	Q13-04/S1	20	PNA-GT-01
5	MON-02/S1	21	PNA-GT-02
6	PNA-14	22	PNA-GT-03
7	PNA-04/S1	23	PNA-GT-04
8	SGZ-01/S1	24	RWK-01
9	WAS-23/S2	25	RWK-02
10	DEL-03	26	RWK-03
11	HAG-01	27	RWK-04
12	HAG-02	28	RWK-05
13	LED-01	29	RWK-06
14	LED-02	30	SGZ-01/S1
15	LIR-01/S3	31	WAS-01
16	LIR-03	32	WAS-23/S2

4.1.2 Reservoir model

The reservoir model is within the regional model, N82°E (Mottaghy, Pechnig, and Vogt, 2011). It has a size of 5.5 km $\times$ 3.5 km $\times$ 1.1 km, with 21,275 nodes using the hexahedral grid (HEX8). The illustration is in Figure 4.3.

The reservoir model comprises five layers, as illustrated in Figure 4.3. The top layer, depicted in pink, represents a cut of the Layer Lower Cretaceous Supergroup (KN), corresponding to the green layer in Figure 4.1b. In contrast, the bottom layer, shown in red, is a cut of Layer Altena (AT), corresponding to the yellow layer in Figure 4.1b. The second layer from the bottom, shown in gray, is a cut of Layer Jurassic Supergroup (S), corresponding to the blue layer in Figure 4.1b. The remaining two layers, depicted in purple and orange in Figure 4.3, are Rodenrijs Claystone (RC) and Delft Sandstone (DS), identified in Willems et al. (2020) and Mijnlief (2020).

as located in the transition between the Lower Cretaceous and Jurassic layers of the West Netherlands Basin. The Delft Sandstone (DS) serves as a target for geothermal installation, while the Rodenrijs Claystone acts as the reservoir caprock (Mottaghy, Pechnig, and Vogt, 2011).

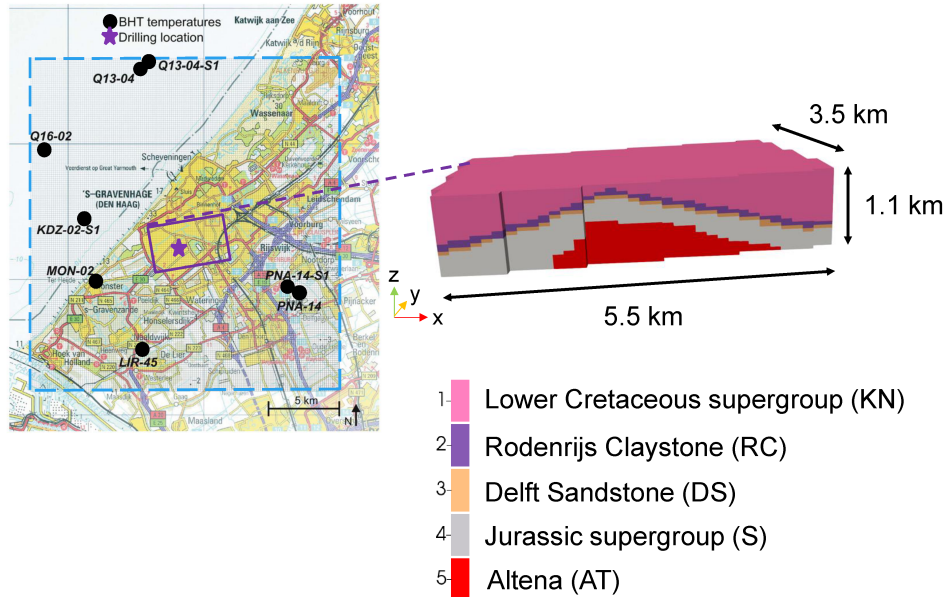


FIGURE 4.3: The reservoir model of The Hague, Netherlands. The image illustrates the layout of the reservoir model within the regional model and a detailed view of the reservoir model itself. The box with the purple line is the reservoir model location. The image is adopted from Mottaghy, Pechnig, and Vogt (2011) and modified.

The geological description and rock properties for each layer are detailed in Table 4.3. The reservoir model is used to simulate the injection-production process, with injection rates ranging from 41.67 to 45.83 L/s and an injection temperature between 35 and 40°C (Veldkamp et al., 2016; Mottaghy, Pechnig, and Vogt, 2011). The uncertain parameters considered are the thermal conductivity of the layers adjacent to the Delft Sandstone and the permeability of the Delft Sandstone layer itself. There are no pressure or temperature measurement data during injection-production activities or well testing.

TABLE 4.3: Geological description and rock properties of The Hague reservoir model.

Layers (top to bottom)	Solid density [kg/m ³]	Solid heat capacity [J/(kg · K)]	Solid thermal conductivity [W/(m · K)]	Porosity [-]	Permeability [m ²]
Lower Cretaceous Super-group (KN)	2,570	960	1.8 - 3.7	1×10^{-3}	1×10^{-17}
Rodenrijs Claystone (RC)	2,570	960	3.5	1×10^{-3}	1×10^{-17}
Delft Sandstone (DS)	2,570	960	5.6	0.15	3×10^{-14} - 2×10^{-12}
Jurassic Super-group (S)	2,570	960	2.3 - 4.9	1×10^{-3}	1×10^{-17}
Altena (AT)	2,570	960	1.6 - 2.7	1×10^{-3}	1×10^{-17}

4.2 Steady-state thermal process

The Hague region is a conductive basin, where the primary mechanism that governs the temperature distribution is heat conduction and fluid-mediated processes are less relevant (Degen and Cacace, 2021; Mottaghy, Pechnig, and Vogt, 2011; Vogt et al., 2010). Since I am only interested in investigating the initial temperature distribution in the region, neglecting the effect of tectonism, paleoclimate, and surface temperature variations, I use a steady-state condition. A steady-state conductive thermal process solves the energy balance equation, mathematically expressed below: (Bartels, Kühn, and Clauser, 2003)

$$\nabla \cdot ([\lambda_s^{1-\phi}(T) + \lambda_f] \nabla T) = 0, \quad (4.1)$$

where T is temperature [$^{\circ}\text{C}$], $\lambda_s(T)$ is temperature-dependent solid thermal conductivity [$\text{W}/(\text{m} \cdot ^{\circ}\text{C})$], λ_f is fluid thermal conductivity [$\text{W}/(\text{m} \cdot ^{\circ}\text{C})$], and ϕ is porosity. The use of temperature-dependent conditions for solid thermal conductivity is based on laboratory measurements performed on cutting samples (Mottaghy, Pechnig, and Vogt, 2011). The temperature-dependent thermal conductivity is calculated as follows: (Bartels, Kühn, and Clauser, 2003)

$$\lambda_s(T) = \begin{cases} \frac{770}{350+T[^{\circ}\text{C}]} + 0.7 [\text{W}/(\text{m}^{\circ}\text{C})], & T > 800^{\circ}\text{C}, \\ \left(\frac{770}{350+T[^{\circ}\text{C}]} + 0.7 \right) \left[\frac{\lambda_s^{ref}}{2.781} \times \left(1 - \frac{T[^{\circ}\text{C}]-20}{800-20} \right) + \left(\frac{T[^{\circ}\text{C}]-20}{800-20} \right) \right] [\text{W}/(\text{m}^{\circ}\text{C})], & T < 800^{\circ}\text{C}, \end{cases} \quad (4.2)$$

where λ_s^{ref} is the reference solid thermal conductivity value [$\text{W}/(\text{m} \cdot ^{\circ}\text{C})$]. I use the The Hague regional model in Figure 4.1c. The top boundary condition is Dirichlet-type, reflecting the average annual ground surface temperature, expressed as follows:

$$T_{top} = 11^{\circ}\text{C}, \quad (4.3)$$

while, the bottom boundary condition is Neumann type, reflecting the heatflux going into the region, expressed as follows:

$$[\lambda_s^{1-\phi}(T) + \lambda_f^\phi] \frac{\partial T}{\partial z} = Q \text{ [W/m}^2\text{]}, \quad (4.4)$$

where Q is the heatflux. The values for the rock properties, the fluid properties and the heat flux are in Section 4.1.1.

I use GOLEM, an open source multiphysics simulator based on MOOSE framework, to obtain the temperature state by solving Equations 4.1 - 4.4 (Cacace and Jacquey, 2017). GOLEM utilizes the finite element (FE) approach to obtain the solutions (Permann et al., 2020; Cacace and Jacquey, 2017). Since the degree of freedom of this problem is huge, I use 48 cores with 192 GB RAM in CLAIX High-Performance Computing (HPC) infrastructure. The CLAIX High-Performance Computing (HPC) infrastructure is equipped with $2 \times$ Intel Xeon Platinum 8160 Processors “SkyLake” (2.1 GHz, 24 cores each). The average computational time for each case is 10 minutes.

4.3 Workflow

The objective of this Chapter is to characterize the uncertainty of the entire temperature state of The Hague region, used for well-targeting. Figure 4.4 shows the overall steps based on the workflow of Degen, Veroy, and Wellmann (2022) and Degen et al. (2021a).

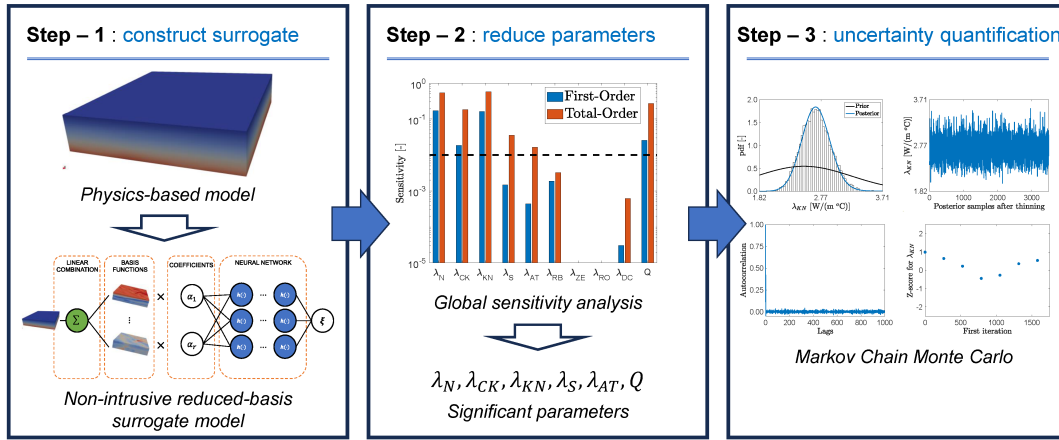


FIGURE 4.4: Workflow for conducting uncertainty quantification of The Hague, Netherlands. This workflow is based on the works of Degen, Veroy, and Wellmann (2022) and Degen et al. (2021a).

The detailed steps are the following:

- **construct a surrogate model using the non-intrusive reduced-basis (NI-RB) method**, see Section 2.1. The surrogate model provides a map from uncertain physical parameters to the temperature state. The construction of a surrogate model is required to reduce the computational cost, allowing large-scale uncertainty quantification. Following the study of Mottaghy, Pechnig, and Vogt (2011), the uncertain physical parameters here are the solid thermal conductivity of each layer and heatflow $\lambda_N, \lambda_{CK}, \lambda_{KN}, \lambda_S, \lambda_{AT}, \lambda_{RB}, \lambda_{ZE}, \lambda_{RO}, \lambda_{DC}, Q$. The prior distribution for each parameter is given in Table 4.4.

- **select significant parameters.** I use variance-based Global Sensitivity Analysis (GSA), see Section 2.2.1, to identify significant parameters. Selecting significant parameters is critical to reduce the number of parameters involved in uncertainty quantification, thus improving the robustness of the analysis (Degen, Veroy, and Wellmann, 2022; Degen et al., 2021a). Note that nonsignificant parameters are not excluded from the analysis; instead, they are assigned their mean values. Furthermore, by analyzing GSA results, we can identify biases in my model due to insufficient measurement data and localized measurement locations (Degen et al., 2021b).
- **uncertainty quantification.** I use the Markov Chain Monte Carlo (MCMC) method, employing the Metropolis-Hasting algorithm, to perform uncertainty quantification (Iglesias and Stuart, 2014), see Section 2.2.2. An important point is to optimally select hyperparameters such that the convergence criteria such as the Geweke criterion and autocorrelation are fulfilled.

TABLE 4.4: Prior distribution for each parameter used in uncertainty quantification of The Hague. Netherlands.

Parameters	Distribution	Mean	Standard deviation	Lower bound	Upper bound
λ_N [W/(m°C)]	Truncated normal	2.80	0.73	1.80	4.00
λ_{CK} [W/(m°C)]	Truncated normal	2.26	0.44	1.73	2.99
λ_{KN} [W/(m°C)]	Truncated normal	2.50	0.73	1.82	3.71
λ_S [W/(m°C)]	Truncated normal	3.75	0.89	2.30	4.89
λ_{AT} [W/(m°C)]	Truncated normal	2.15	0.35	1.64	2.72
λ_{RB} [W/(m°C)]	Truncated normal	2.96	1.03	1.47	4.66
λ_{ZE} [W/(m°C)]	Truncated normal	3.58	1.14	1.71	4.96
λ_{RO} [W/(m°C)]	Truncated normal	3.90	1.18	1.96	4.81
λ_{DC} [W/(m°C)]	Truncated normal	2.12	1.07	1.47	3.88
Q [W/m ²]	Uniform	-	-	0.06	0.07

4.4 Results

I present my results as follows: Section 4.4.1 displays the surrogate performance. In Section 4.4.2, I show the significant parameters identified through the global sensitivity analysis. Furthermore, I analyze biases caused by insufficient measurement data and localized measurement locations using the concept of process-oriented and measurement-focused objective functions, used in Degen et al. (2021b). Finally, I perform uncertainty quantification with the significant parameters identified in Section 4.4.3.

4.4.1 Surrogate model

I utilize 500 training and test samples to construct my surrogate model, considering the non-linearity introduced by temperature-dependent thermal conductivity. Table 4.5 presents details regarding the number of basis functions, tolerance, and hyperparameters utilized in the surrogate model for the temperature state. The L_2 error estimator is 1.2×10^{-1} °C, which is lower than the measurement error range of 5 - 10 °C. This ensures that the approximation error will not influence the later global sensitivity analysis and uncertainty quantification results.

TABLE 4.5: Tolerance and hyperparameters used in the surrogate construction for temperature state.

Time-trajectory tolerance	Parameter tolerance	Number of basis functions
-	1×10^{-4}	23
Hyperparameters		Values
Number of neurons in hidden layer 1 N_{n,l_1}		52
Number of neurons in hidden layer 2 N_{n,l_2}		140
Number of epochs N_e		46768
Learning rate l_r		1.5×10^{-4}
Batch size N_b		14

I show the offline and online time of the surrogate model in Figure 4.5. The offline time is defined as the time needed to train a surrogate model without considering the time needed to generate the training and test set. The online time is defined as the time that a surrogate model needs to predict a new case. As depicted in Figure 4.5, I gain a more than 7 orders of magnitude acceleration in predicting a new case. The offline time is 22 minutes. To be more precise, if I include the generation of the training and test set, the offline time becomes $22 + 4.8 \times 10^5$ minutes.

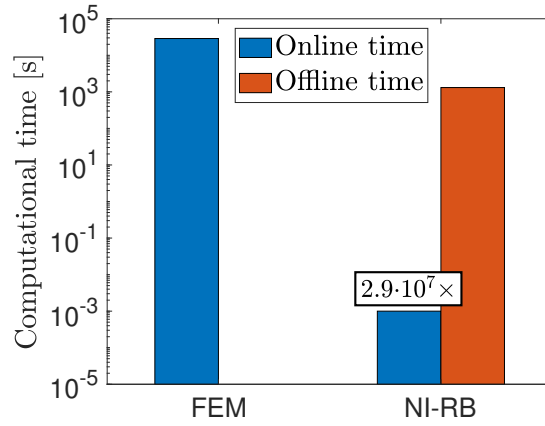


FIGURE 4.5: Offline and online time of the surrogate model for The Hague regional model. The value 2.9×10^7 denotes the computational speed-up achieved by using the NI-RB surrogate model relative to the FEM simulation.

4.4.2 Global sensitivity analysis

Besides reducing the number of parameters involved in the subsequent quantification of the uncertainty, a preliminary analysis using global sensitivity analysis (GSA) is beneficial for identifying biases in the model resulting from limited and clustered measurement data (Degen et al., 2021b; Degen et al., 2021a). This analysis guides the approach to address these biases during later stages of uncertainty quantification, e.g., assigning weights to clustered measurement data (Degen et al., 2021a). I perform several analyses with process-focused and measurement-focused objective functions to investigate biases (Degen et al., 2021b). The overall analyses are in Figure 4.6. Given that the governing physical process in The Hague, Netherlands is heat conduction, I employ the sum-of-temperature function as the process-focused objective function. For the measurement-focused functions, I implement three weighting scenarios that account for clustered measurement locations and the absence of measurement data at depths deeper than 4 km, as shown in Figure 4.2b and 4.2c. The scenarios are: 1) no weight, indicating no weights assigned to the measurement data, 2) expert weight, based on the selected measurement data used in Mottaghy, Pechnig, and Vogt (2011), and 3) cluster weight, determined by the density of measurement data points existing in a certain layer, which means that each measurement data will be weighted less in a densely measured layer and more in sparsely measured layers.

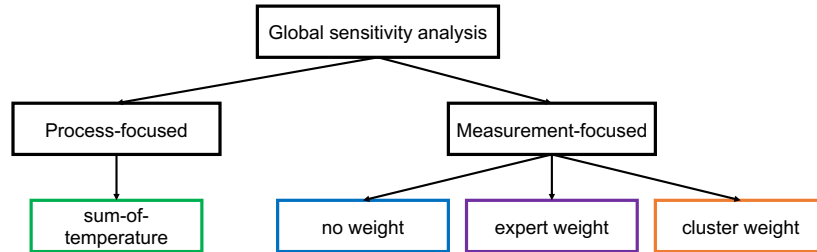


FIGURE 4.6: Scenarios used for global sensitivity analysis in The Hague case study.

Figure 4.7 shows the results of the global sensitivity analysis using all the scenarios in Figure 4.6. I use 100,000 realizations per parameter with the Saltelli sampling method to avoid statistical error (Degen et al., 2021b; Saltelli et al., 2010). Using a surrogate model, the global sensitivity analysis for each scenario is done in 6.5 minutes with a single core of Apple Macbook Pro with M1 chip and 8 GB RAM.

In Figure 4.7a, I identify $\lambda_N, \lambda_{CK}, \lambda_{KN}, \lambda_S, \lambda_{AT}, \lambda_{RB}, \lambda_{DC}, Q$ as significant parameters. In particular, λ_N, λ_{DC} , and Q play a crucial role in regulating the inflow and outflow of energy within the region. This is expected considering their impact on the overall flow of energy. Furthermore, $\lambda_{CK}, \lambda_{KN}, \lambda_S, \lambda_{AT}$, and λ_{RB} emerge as significant parameters, given the substantial thickness of the corresponding layers. These thick layers contribute to changes in the temperature distribution in the region. It is particularly noteworthy that λ_{KN}, λ_S , and λ_{AT} are essential, as these layers contain the reservoir model, indicating the importance of extracting heat from the reservoir. $\lambda_{ZE}, \lambda_{RO}$ are not significant since those layers are thin.

Theoretically, if I have sufficient measurement data and they are well distributed, global sensitivity analysis with a measurement-focused function yields the same result as with a process-focused function, describing the governing physical processes (Degen et al., 2021b). However, given the available measurement data, see Figure 4.2b and 4.2c, I can only characterize layer N, CK, KN, S, AT, and heat flow, as they are indicated as significant. Notably, there is an absence of measurement data for layer DC (indicated as nonsignificant), and there are insufficient data for layer RB (indicated as nonsignificant despite some available data). Moreover, Figure 4.7b reveals that layers S and AT are considered important based on their total-order sensitivity indices, not in conjunction with their first-order sensitivity indices. Despite the presence of some measurement data points in these layers, the influence of dense data, particularly in layer KN, contributes to this outcome. Consequently, if I would like to exclude layers S and AT, the other parameters correlated with these two layers will not be properly characterized. Hence, I include λ_S, λ_{AT} in uncertainty quantification.

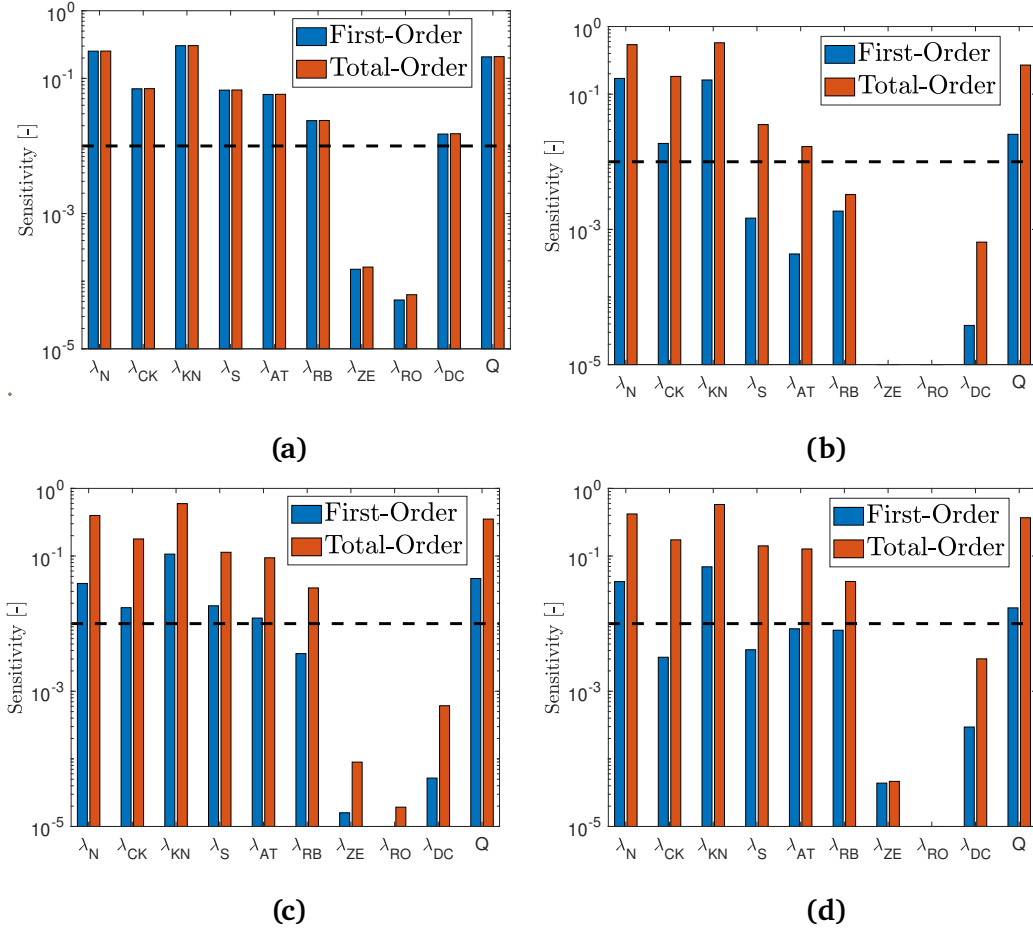


FIGURE 4.7: Global sensitivity analysis result using The Hague regional model: (a) using sum-of-temperature objective function, (b) using no-weight scenario in measurement-focused function, (c) using expert-weight scenario in measurement-focused function, and (d) using cluster-weight scenario in measurement-focused function.

Following the results in Figure 4.7a and 4.7b, I try to de-cluster the data by assigning weights (Degen et al., 2021a). The aim is to obtain an improved uncertainty characterization for all layers and heatflow. The first approach to de-cluster the measurement data here is through expert judgment by selecting representative measurement locations as is done in Mottaghy, Pechinig, and Vogt (2011). Figure 4.1b shows the selected wells and Figure 4.7c presents the results of the global sensitivity analysis. The result is close to the one with a process-focused function. However, layer DC still cannot be characterized due to the absence of measurement data. I can also see that layer RB is significant only based on its total-order sensitivity index.

In Figure 4.7d, I implement a cluster-weight scenario (the second approach) within the measurement-focused function, producing results similar to those obtained with the expert-weight scenario. However, I observe that the first-order sensitivity indices of λ_{CK} , λ_S , λ_{AT} , and λ_{RB} fall below the cut-off line, indicating their reduced significance.

Through these analyses, I observe that a de-clustering process (by assigning weights to the measurement data) changes the significance parameters considered in the uncertainty quantification process. It gives us insights that more measurement data are needed to confirm the significance of certain parameters, such as λ_{RB} in this case study. This is crucial as it directly influences the results of uncertainty quantification, and consequently our decision-making process on whether to proceed with the heat extraction process on the proposed location or not.

4.4.3 Uncertainty quantification

I employ Markov Chain Monte Carlo (MCMC) with three scenarios—no weight, expert weight, and cluster weight—in alignment with the results of the global sensitivity analysis. For no-weight scenario, I perform MCMC with $\lambda_N, \lambda_{CK}, \lambda_{KN}, \lambda_S, \lambda_{AT}, Q$. For both expert-weight and cluster-weight scenarios, the parameters are $\lambda_N, \lambda_{CK}, \lambda_{KN}, \lambda_S, \lambda_{AT}, \lambda_{RB}, Q$. Note that even though the expert-weight and cluster-weight scenarios use the same parameters, they have a different likelihood function in the MCMC due to a different weighting scheme. It is also important to note that uncertainty quantification involves assimilating our model with available measurement data. Consequently, it is not conducted based on the global sensitivity result obtained with a process-focused function.

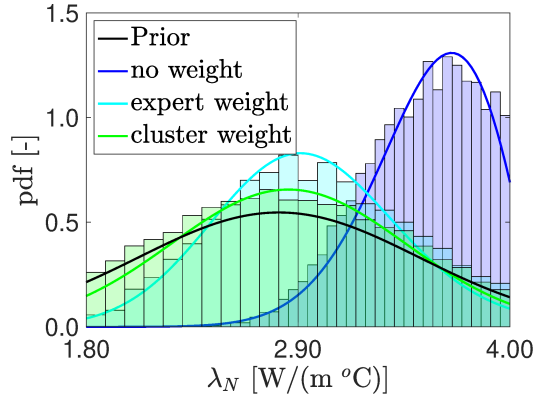
Figure 4.8 shows the posterior and distribution analysis for each parameter involved in MCMC. All scenarios are executed with 1,000,000 steps, 300,000 burn-in, and 100 thinning. Using a surrogate model, the uncertainty quantification for each scenario is done in 3.5 hours with a single core of Apple Macbook Pro with M1 chip and 8 GB RAM. The convergence results are in Figure 4.9. For all parameters, the Geweke criterion stays with +2 and -2, the autocorrelation shows the robustness by having values close to zero, and the traces plots indicate sufficient mixing.

In the no-weight scenario, a significant reduction in uncertainty can be observed for each parameter, particularly for layer KN, where dense measurement data exist. The quantification of all parameters results in non-Gaussian posterior distributions, indicating nonlinearity within the system. However, in this scenario, I am unable to characterize layer RB as the measurement data cluster around layer KN. The characterization of Q is also challenging, as I lack direct measurement data or deep measurement points.

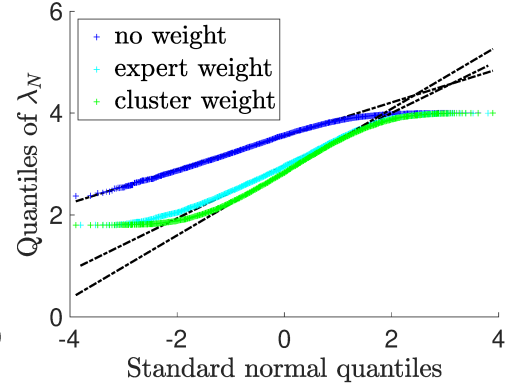
In the cluster-weight scenario, layer RB can now be characterized, as illustrated in Figure 4.8k. However, unlike the no-weight scenario, there is no significant reduction in uncertainty for each parameter. The challenge remains to characterize Q due to the absence of direct measurement data or deep measurement points. Additionally, all posterior distributions are non-Gaussian.

In the expert-weight scenario, the use of fewer measurement data (only 10 measurement data points, as reported by Mottaghy, Pechinig, and Vogt (2011)) also results in a less significant uncertainty reduction compared to the no-weight scenario, which employs 105 measurement data points. As depicted in Figure 4.8, the posterior distributions closely resemble the prior distributions. However, a poor characterization is observed in layer S, where the posterior distribution is substantially shifted to the left.

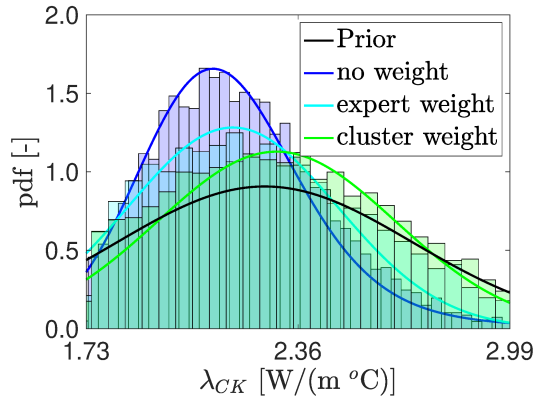
Figure 4.10 illustrates the uncertainty quantification results for various scenarios. To better show the non-Gaussian distribution observed in all results, I present the 0.05 quantile, mean, and the 0.95 quantile. The no-weight scenario exhibits the most significant reduction in uncertainties compared to the uncalibrated one. It is supported by a significant change in the distribution of all parameters, as depicted in Figure 4.8. The expert-weight scenario also show a reduction in uncertainties, although not as pronounced as the no-weight scenario. In contrast, the cluster-weight scenario yields results similar to the uncalibrated one.



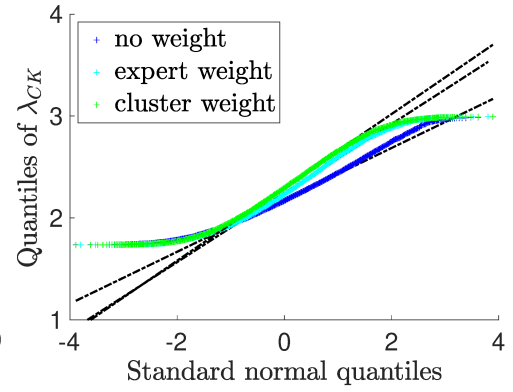
(a)



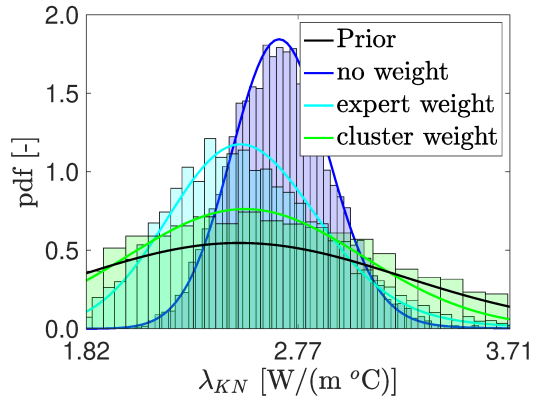
(b)



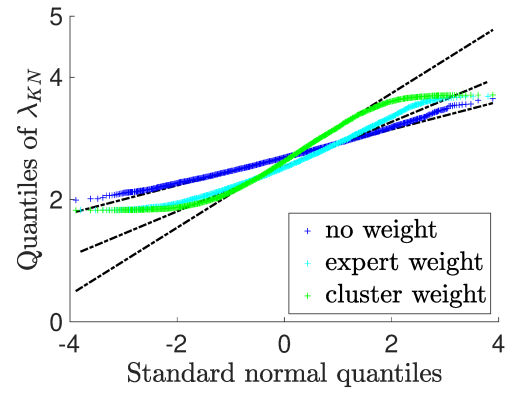
(c)



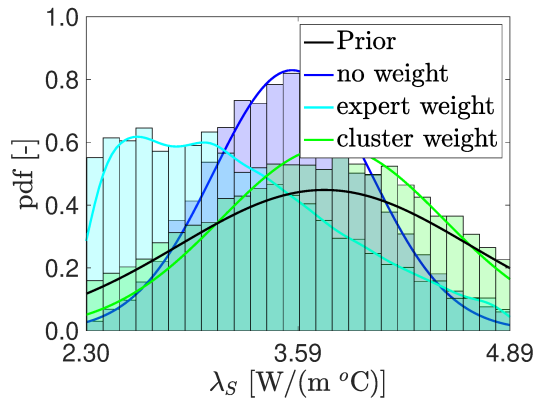
(d)



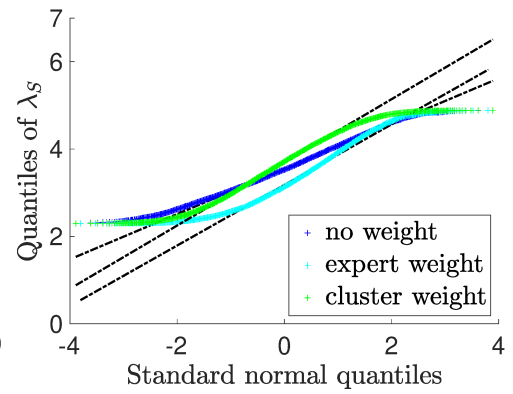
(e)



(f)



(g)



(h)

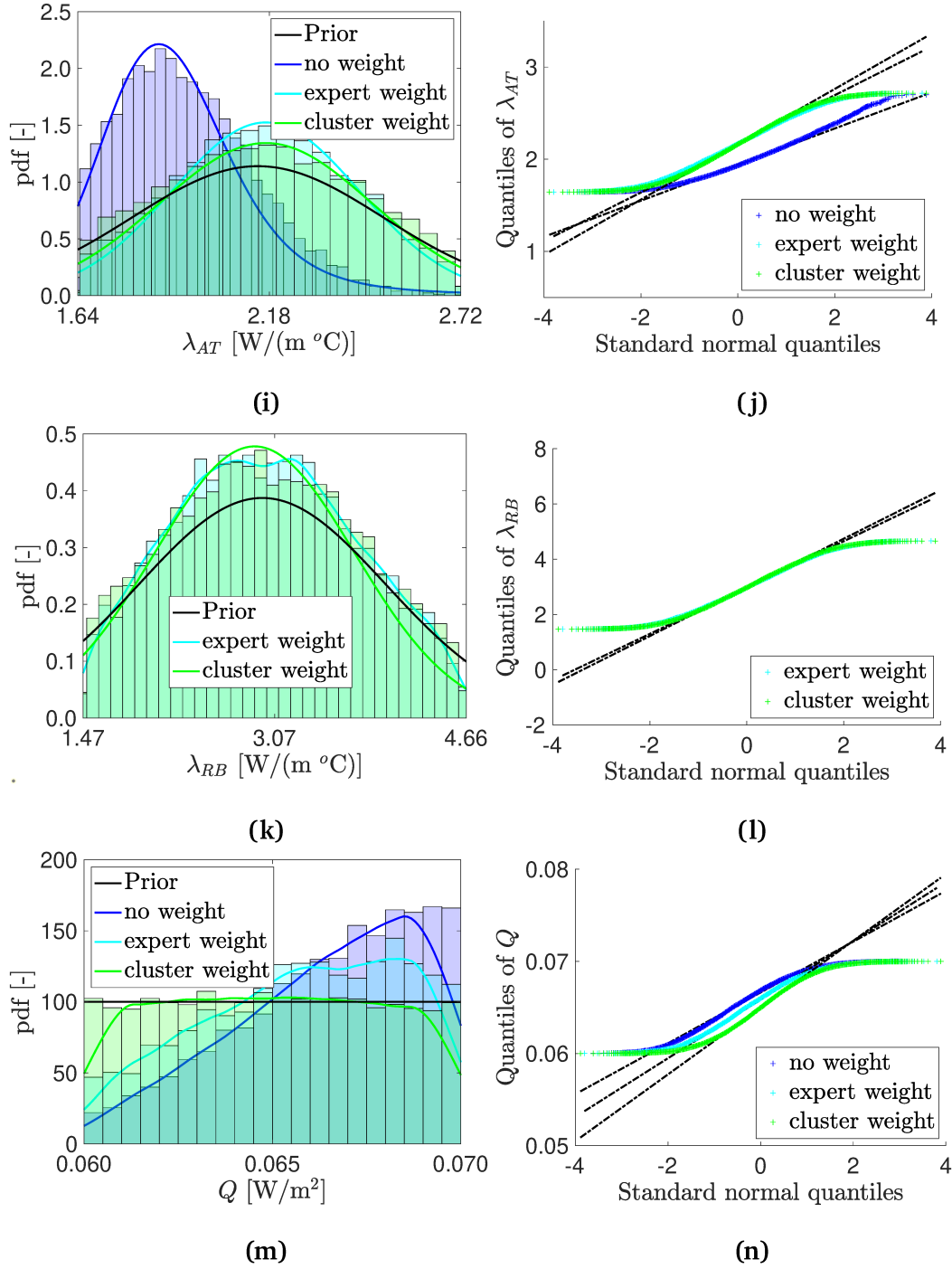
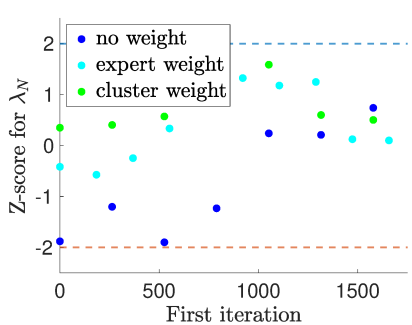
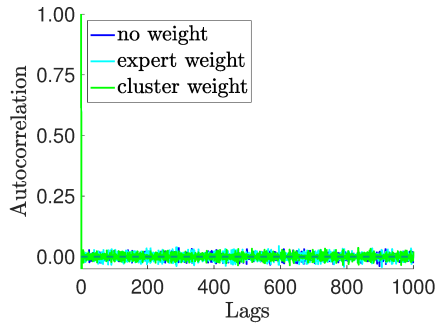


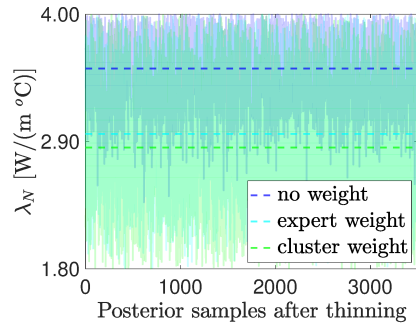
FIGURE 4.8: Uncertainty quantification results: (a), (c), (e), (g), (i), (k), (m) consecutively are posterior of $\lambda_N, \lambda_{CK}, \lambda_{KN}, \lambda_S, \lambda_{AT}, \lambda_{RB}, Q$ obtained with different scenarios; (b), (d), (f), (h), (j), (l), (n) consecutively are q-q plot of $\lambda_N, \lambda_{CK}, \lambda_{KN}, \lambda_S, \lambda_{AT}, \lambda_{RB}, Q$ obtained with different scenarios. Note that, in the no-weight scenario, λ_{RB} is not significant, hence, it does not appear in (k) and (l).



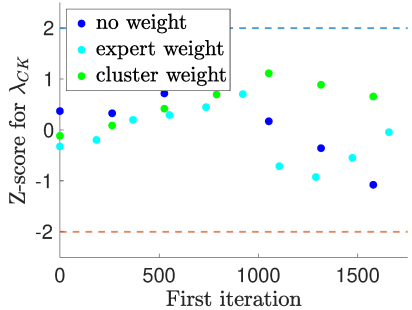
(a)



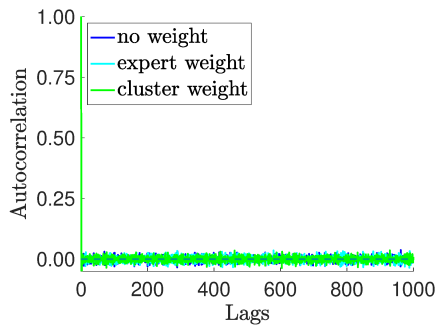
(b)



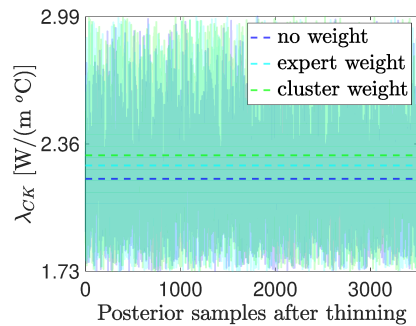
(c)



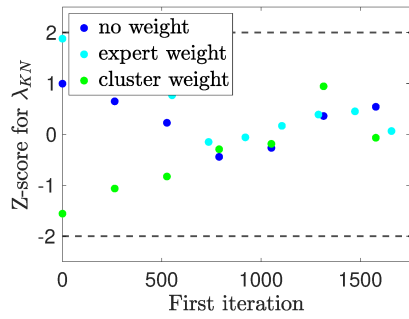
(d)



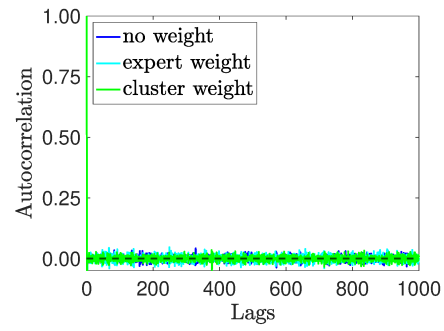
(e)



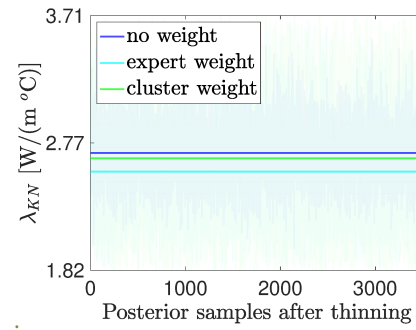
(f)



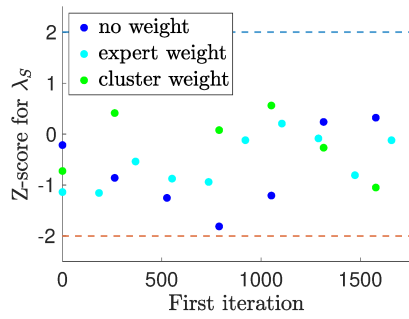
(g)



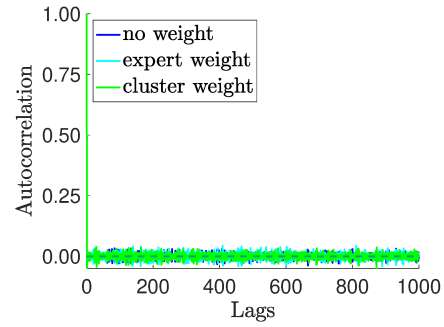
(h)



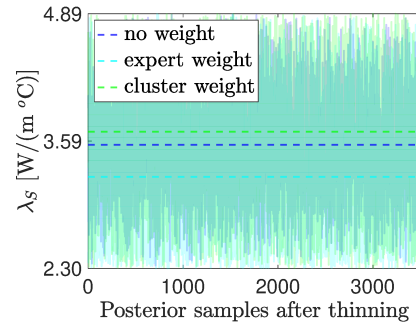
(i)



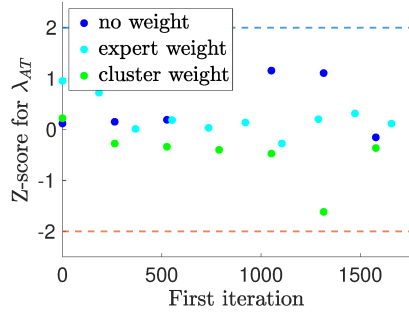
(j)



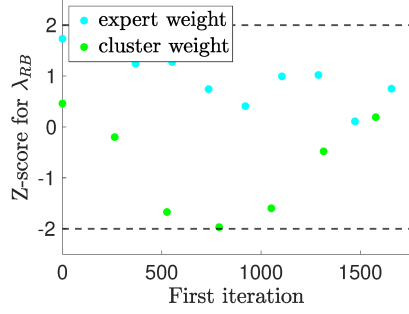
(k)



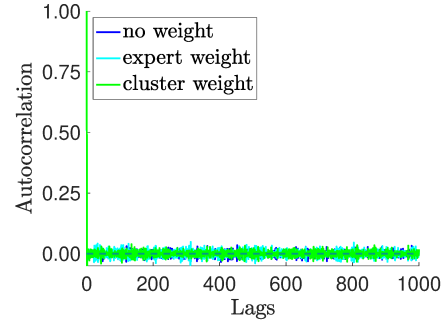
(l)



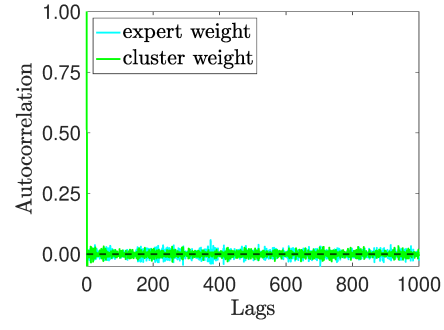
(m)



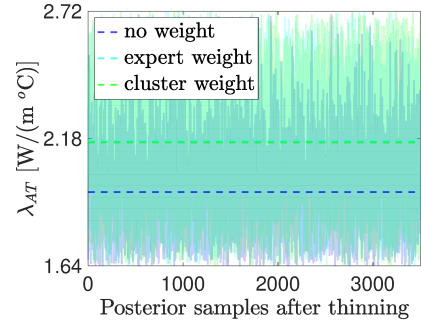
(p)



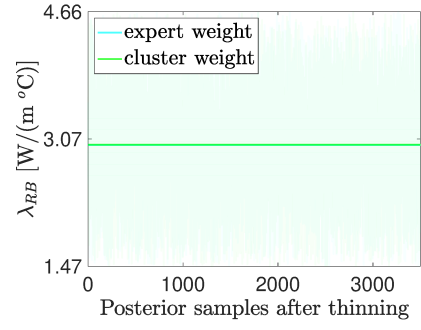
(n)



(q)



(o)



(r)

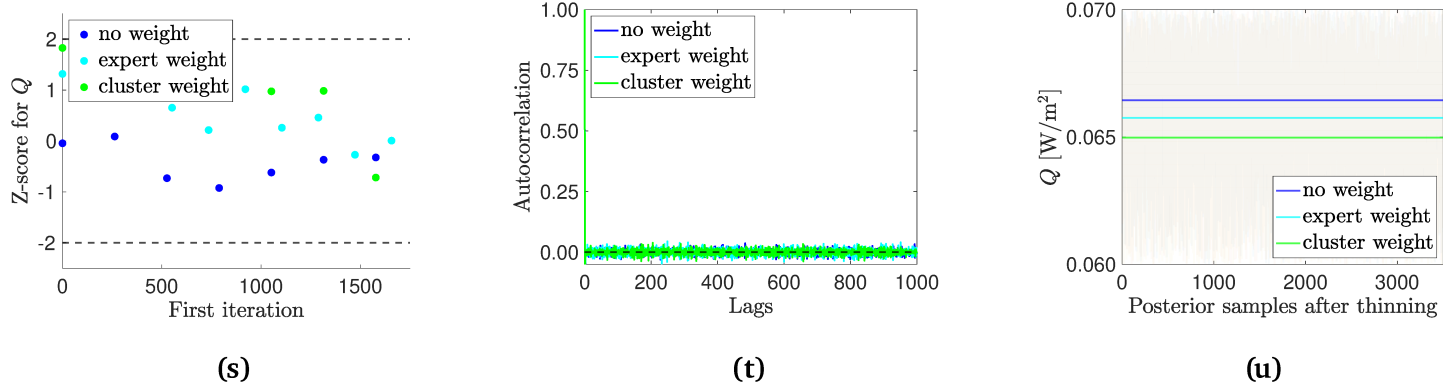
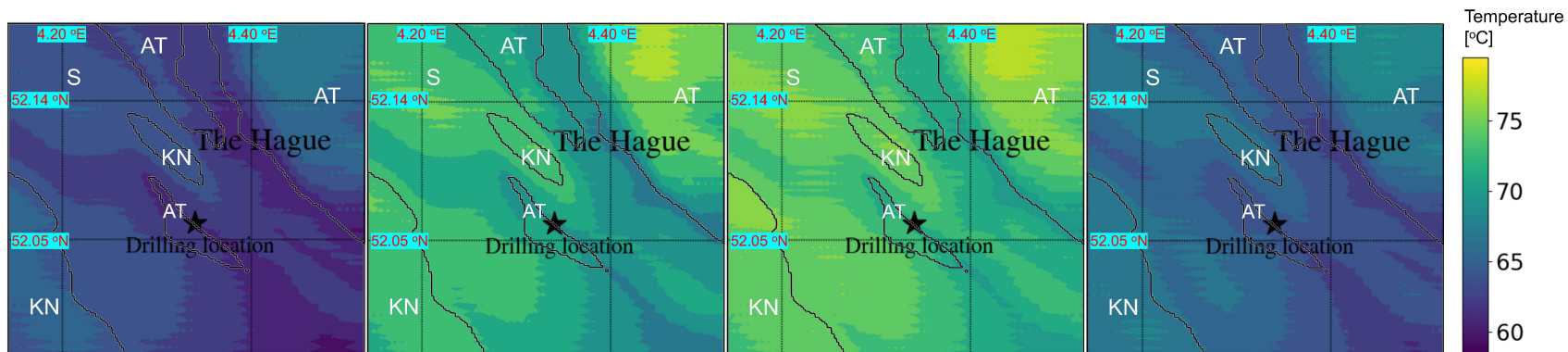


FIGURE 4.9: Convergence criteria for each parameter involved in uncertainty quantification: (a), (d), (g), (j), (m), (p), (s) are Geweke criterion consecutively applied on $\lambda_N, \lambda_{CK}, \lambda_{KN}, \lambda_S, \lambda_{AT}, \lambda_{RB}, Q$ obtained with different scenarios; (b), (e), (h), (k), (n), (q), (t) consecutively are autocorrelation of $\lambda_N, \lambda_{CK}, \lambda_{KN}, \lambda_S, \lambda_{AT}, \lambda_{RB}, Q$ obtained with different scenarios; (c), (f), (i), (l), (o), (r), (u) consecutively are traces of $\lambda_N, \lambda_{CK}, \lambda_{KN}, \lambda_S, \lambda_{AT}, \lambda_{RB}, Q$ obtained with different scenarios. Note that, in the no-weight scenario, λ_{RB} is not significant, hence, it does not appear in (p), (q), and (r).

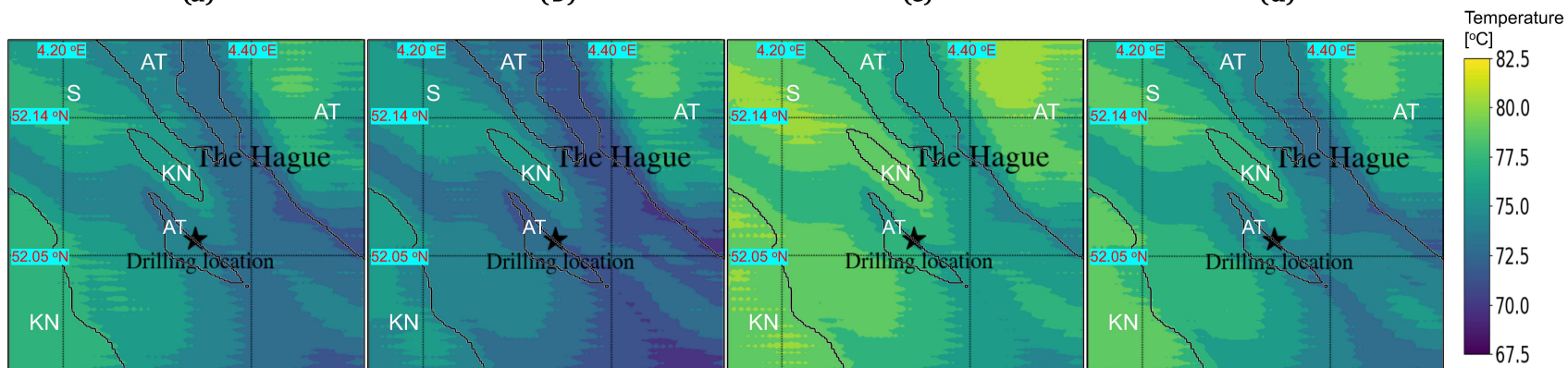


(a)

(b)

(c)

(d)



(e)

(f)

(g)

(h)

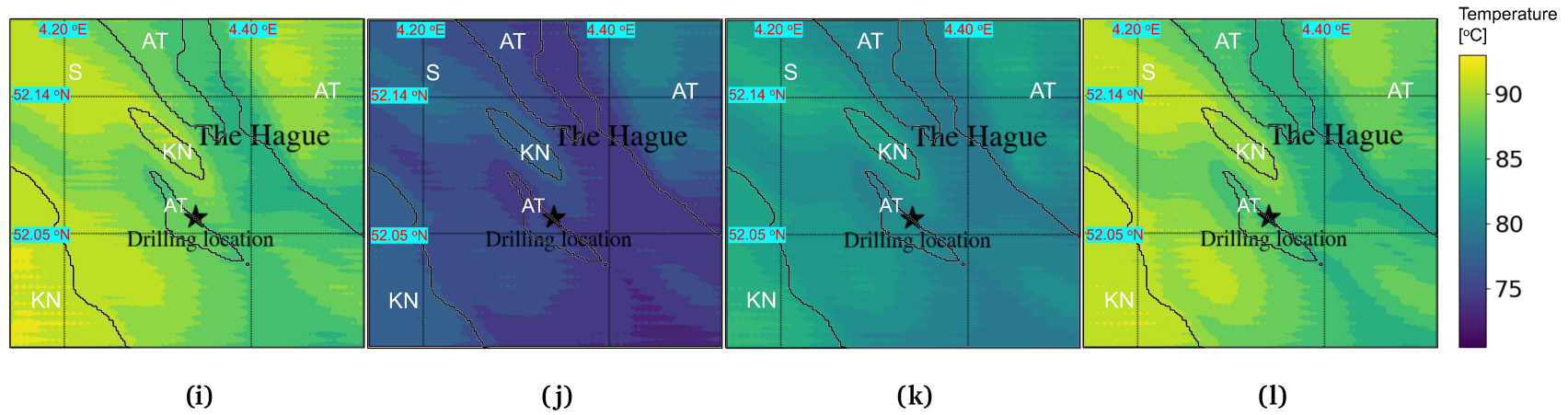


FIGURE 4.10: Uncertainty quantification results for The Hague case study: (a), (b), (c), and (d) are the 0.05 quantile; (e), (f), (g), and (h) are the mean; (i), (j), (k), and (l) are the 0.95 quantile. (a), (e), and (i) use uncalibrated scenario; (b), (f), and (j) use no-weight scenario; (c), (g), and (k) use expert-weight scenario; (d), (h), and (l) use cluster-weight scenario.

Through these analyses, I observe that the de-clustering process, whether utilizing expert weighting or a cluster weighting scheme, enables the characterization of the Lower Germanic Trias Group (RB) layer and removes the clustering effect. However, it yields a result similar to that of the uncalibrated scheme. However, in the no-weighting scheme, I acknowledge the presence of biases resulting from clustered measurement locations, which are unavoidable. It is evident that my measurement data primarily focus on characterizing the layers above the Lower Germanic Trias Group (RB) layer. Therefore, if my focus were to shift solely to characterizing the layers above the Lower Germanic Trias Group (RB) layer, my measurement data would be descriptive enough.

4.5 Discussions

With the use of non-intrusive reduced-basis (NI-RB) surrogate model, I can perform the global sensitivity analysis on a large-scale The Hague model in average 3 minutes and uncertainty quantification in average 3 hours. It is attributed to a 7 orders of magnitude speed-up provided during the online phase by the surrogate model. The NI-RB method proves to be efficient, requiring only small training samples to accurately capture the entire temperature state of The Hague.

Mottaghy, Pechnig, and Vogt (2011) uses a forward Monte Carlo approach to calibrate the The Hague model and assumes a Gaussian distribution for all posterior distributions. In contrast, my work employs a Markov Chain Monte Carlo approach (a Bayesian inference approach) that allows for non-Gaussian posterior distributions. Schaeffner et al. (2021) compare these two approaches and demonstrate that the posterior distributions of the forward Monte Carlo approach are often inconsistent with the Gaussian assumption. Moreover, the forward Monte Carlo approach does not account for correlations among parameters, leading to a non-robust characterization of posterior distributions.

Although the mean of my temperature state has the same range as the one in Figure 9 of Mottaghy, Pechnig, and Vogt (2011), aligned with the observations of Schaeffner et al. (2021), I notice that using this selected data (expert-weight scenario) causes a non-robust posterior characterization of λ_S , see Figure 4.8g. This is completely different from the posterior distribution of λ_S in Mottaghy, Pechnig, and Vogt (2011). Therefore, I cannot rely on the results from this scenario.

Moreover, the implementation of the cluster-weight scenario does not result in a reduction in uncertainties. Here, I can characterize layer RB. In the context of my study, observing the results from both the expert-weight and cluster-weight scenarios collectively shows that assigning weights to de-cluster the measurement data is unnecessary. Therefore, I proceed with the results from the no-weight scenario as it already provides robust characterization for the posterior distributions. It is also important to note that, due to the absence of direct measurements of the heatflow and deep measurement data, achieving an accurate posterior characterization of the heatflow is challenging, as illustrated in Figure 4.8m.

Relying on the no-weight scenario result, I cannot characterize layer RB which is denoted as significant in the global sensitivity analysis in Figure 4.7a. If I would like to characterize this layer, more measurement data in this layer are needed, together with deep measurement data. However, based on Mottaghy, Pechnig, and Vogt (2011), the main interest is to only characterize the layers around the reservoir for heat extraction evaluation: layer CK, KN, S, and AT. Therefore, my result from the no-weight scenario remains sufficient for fulfilling this purpose.

Given the proposed drilling location, marked with a purple star in Figure 4.1b, and utilizing the uncertainty quantification result with the no-weight scenario, the uncertainty range of the temperature (from the 0.05 quantile to the 0.95 quantile) is 69°C to 77°C. In contrast, without calibration, the uncertainty range expands to 61°C to 86°C. Consequently, with the expected temperature of around 75°C (Mottaghy, Pechinig, and Vogt, 2011), this narrower uncertainty range enhances confidence to drill a well in the proposed drilling location.

Chapter 5

Optimal operating conditions, considering subsurface uncertainties and safety - Insight from The Hague, Netherlands

"Every object is a mixture of lots of things, so we can deal with it only as a series of approximations and idealizations."

— Richard FEYNMAN (1963)

From the evaluation using the steady-state thermal process, I now confirm that the proposed drilling location is suitable for the purpose described in Mottaghy, Pechinig, and Vogt (2011). To assess heat extraction, another calibration is needed using pressure and temperature data measured in flowing wells. Unfortunately, such measurement data are not available (Mottaghy, Pechinig, and Vogt, 2011). Hence, in this chapter, I conduct the heat extraction evaluation, optimizing operating conditions while considering all possible uncertainties and emphasizing safety. The evaluation from the previous chapter gives insight into determining the lower and upper bounds of the operating conditions, which will be discussed in this chapter. This chapter is covered in Santoso et al. (in preparation, b).

5.1 Transient thermo-hydraulic problem

For addressing the heat extraction in The Hague, Netherlands, I solve a transient thermo-hydraulic problem. A transient thermo-hydraulic problem consists of mass balance, Darcy law, and heat transfer equations. The mass balance equation for the fluid phase is expressed as follows: (Cacace and Jacquey, 2017)

$$\frac{\partial \phi \rho_f}{\partial t} + \nabla \cdot (\rho_f \mathbf{u}_f) = R, \quad (5.1)$$

where ϕ is the porosity of the porous medium [-], ρ_f is the density of the fluid phase [kg/m³], t is time [months], $\mathbf{u}_f$ is the Darcy velocity [m/s], and R is the sink or source term [kg/(m³ · s)]. The Darcy law is expressed as below: (Cacace and Jacquey, 2017)

$$\mathbf{u}_f = \frac{\mathbf{k}}{\mu_f} (\nabla P_f - \rho_f g \mathbf{z}), \quad (5.2)$$

where $\mathbf{k}$ is the permeability of the porous medium [m²], μ_f is the viscosity of fluid phase [Pa · s], P_f is the pore pressure [MPa], g is the gravitational acceleration, and

$\mathbf{z}$ denotes the direction on the z axis where the gravitational force works. The heat transfer equation assuming local thermal equilibrium between the fluid and the rock phase is expressed as follows: (Cacace and Jacquey, 2017)

$$\frac{\partial}{\partial t}([\phi\rho_f c_f + (1 - \phi)\rho_s c_s] T) + \nabla \cdot (\rho_f c_f \mathbf{u}_f T - [\lambda_s^{1-\phi}(T) + \lambda_f^\phi] \nabla T) = \dot{H}, \quad (5.3)$$

where c_f is the fluid specific heat [J/(kg °C)], ρ_s is the solid density [kg/m³], c_s is the solid specific heat [J/(kg °C)], T is the temperature [°C], $\lambda_s(T)$ is temperature-dependent solid thermal conductivity [W/(m · °C)], and λ_f is fluid thermal conductivity [W/(m · °C)]. The temperature-dependent thermal conductivity is calculated as follows: (Bartels, Kühn, and Clauser, 2003)

$$\lambda_s(T) = \begin{cases} \frac{770}{350+T[^\circ\text{C}]} + 0.7 \text{ [W/(m}^\circ\text{C)]}, & T > 800^\circ\text{C}, \\ \left(\frac{770}{350+T[^\circ\text{C}]} + 0.7\right) \left[\frac{\lambda_s^{ref}}{2.781} \times \left(1 - \frac{T[^\circ\text{C}]-20}{800-20}\right) + \left(\frac{T[^\circ\text{C}]-20}{800-20}\right)\right] \text{ [W/(m}^\circ\text{C)]}, & T < 800^\circ\text{C}, \end{cases} \quad (5.4)$$

where λ_s^{ref} is the reference solid thermal conductivity value [W/(m · °C)]. I employ a finite element (FE) approach to solve for variables P_f and T in my study. Numerical simulations are performed using GOLEM, an open source multiphysics simulator based on the MOOSE framework (Cacace and Jacquey, 2017).

5.1.1 Synthetic model setup

To gain a further understanding of optimization under uncertainties with a safety consideration, I use a synthetic case: 1-D cold water injection into a hot horizontal core. It is based on the transient thermo-hydraulic process. In this synthetic case, I do not use a temperature-dependent thermal conductivity setup, therefore, Equation 5.3 becomes the following:

$$\frac{\partial}{\partial t}([\phi\rho_f c_f + (1 - \phi)\rho_s c_s] T) + \nabla \cdot (\rho_f c_f \mathbf{u}_f T - [(1 - \phi)\lambda_s + \phi\lambda_f] \nabla T) = \dot{H}, \quad (5.5)$$

I solve Equation 5.1, 5.2, and 5.5 for variables P_f and T using GOLEM (Cacace and Jacquey, 2017). The computational domain, the boundary conditions, and the initial conditions are shown in Figure 5.1.

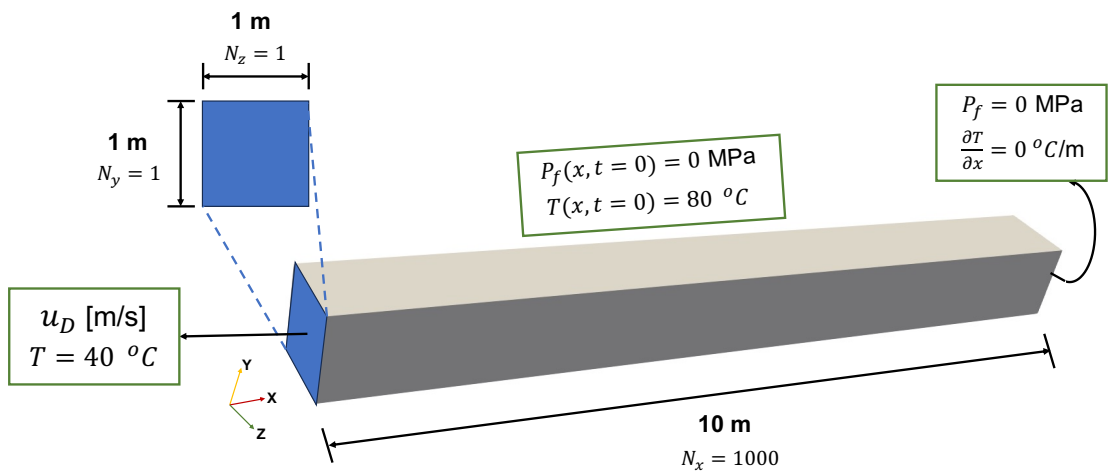


FIGURE 5.1: Numerical simulation setup for optimizing operating conditions of a synthetic case.

The initial condition for pressure is zero everywhere:

$$P_f(x, t = 0) = 0[\text{MPa}], \quad (5.6)$$

While the initial condition for temperature is 80 °C

$$T(x, t = 0) = 80[^\circ\text{C}], \quad (5.7)$$

On the left side of the domain (the inlet side), the pressure boundary condition reflects the injection scenario, mathematically expressed as follows:

$$\frac{\partial P_f}{\partial x} = \frac{\mathbf{k}}{\mu_f} u_D [\text{MPa/m}], \quad (5.8)$$

where u_D is the injection velocity in m/s. The boundary condition for temperature is of Dirichlet type:

$$T(x = 0, t) = 40[^\circ\text{C}], \quad (5.9)$$

On the right side of the domain (the outlet side), the boundary condition for pressure is Dirichlet-type:

$$P_f(x = 10[\text{m}], t) = 0[\text{MPa}], \quad (5.10)$$

While the boundary condition for temperature is Neumann type:

$$\frac{\partial T}{\partial x} = 0[^\circ\text{C/m}], \quad (5.11)$$

The domain has 1,000 discretization along the x-axis and 101 time steps in days (from 0 to 10.0 days). Hence, the total degree of freedom is 202,000. Executing this simulation setup requires 3 minutes with a single core in an Apple Macbook Pro with an M1 chip and 8 GB RAM.

I consider permeability as the uncertain parameter and injection velocity u_D as the operation-related parameters. Their value ranges and distributions are in Table 5.1. I show the temperature and pressure response at both the injector and the producer using the upper, mean, and lower bound values in Figure 5.2.

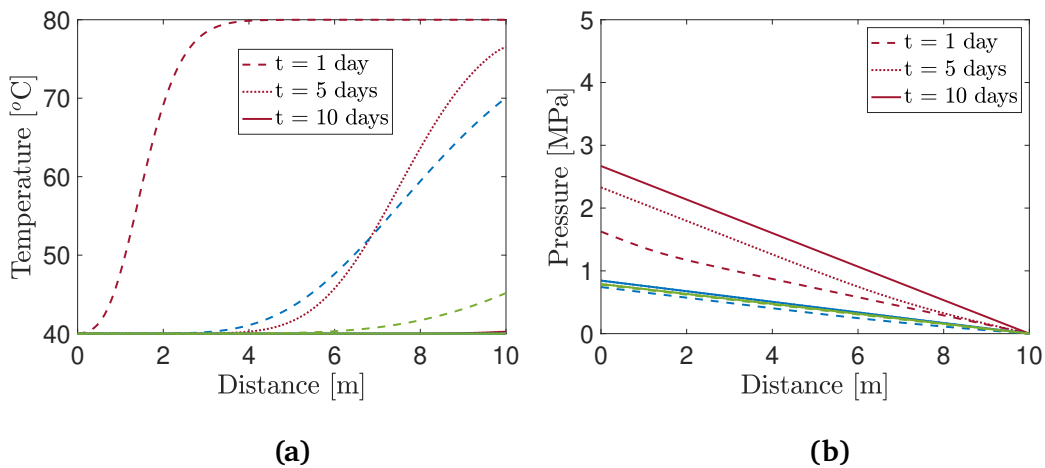


FIGURE 5.2: The spatio-temporal solution of the thermo-hydraulic simulation on the synthetic case: (a) temperature and (b) pressure. The red color denotes responses obtained using the lower bound values, blue color the mean values, and green color the upper bound values.

From Figure 5.2a, I have different temperature values by varying the uncertain

material parameters and the operation-related parameters from their lower bounds to their upper bounds. For the temperature response, I am interested in the total depletion time in which the temperature in the whole computational domain is 40 °C. This total depletion can be achieved for less than ten days, depending on the injection velocity. The injection velocity is set to vary from 1.0 to 10 m/s such that the total depletion time can vary from two to nearly nine days and create a significant pressure increase on the left side.

In Figure 5.2b, the pressure linearly decreases from the left side to the right side of the domain. The increase in pressure in the entire domain over time is mainly caused by the increase in pressure in the left side. The increasing trend of pressure over time varies depending on the permeability of the core. I am interested in the increase of pressure over time at the left side, and this pressure should not go beyond 5 MPa. This 5 MPa is intentionally selected to create 50% failure situation in the optimization process with the given permeability range in Table 5.1.

TABLE 5.1: Prior distribution for each parameter used in the optimization of the synthetic case.

Parameters	Distribution	Mean	Standard deviation	Lower bound	Upper bound
$k \text{ m}^2]$	Uniform	-	-	2.96×10^{-14}	1.00×10^{-12}
$u_D \text{ [m/s]}$	Uniform	-	-	1.00	10.0

5.1.2 The Hague model setup

As presented in Chapter 2, there are two geological models of The Hague, Netherlands: the regional model and the reservoir model. For the evaluation of heat extraction, my primary interest lies in the reservoir model rather than the entire region (Mottaghy, Pechinig, and Vogt, 2011). While the separation of these models helps to reduce computational costs, it introduces challenges in assigning boundary conditions for the reservoir model (Degen et al., 2022). The presence of uncertainties in the rock properties of the layers within the regional model, particularly at the boundaries with the reservoir model, complicates the assignment of proper boundary conditions. Maintaining consistency in the physical response between the reservoir and regional models becomes challenging under these circumstances. However, attempting to merge both models poses challenges, as insufficient data hinders the extrapolation of reservoir boundaries into the layers of the regional model. Furthermore, this attempt will further increase the uncertainties.

To reduce error and maintain consistency in the physical response between the reservoir and the regional model, I implement the transfer solution approach. The detailed steps are the following:

- At $t = 0$, perform a steady-state thermal simulation with temperature-dependent thermal conductivity setup as in Chapter 4 on the regional model. The top boundary condition of the regional model is a Dirichlet type, reflecting the average annual ground surface temperature, expressed as follows:

$$T_{top}^{\text{reg}} = 11^\circ\text{C}, \quad (5.12)$$

while, the bottom boundary condition is a Neumann type, reflecting the heatflux going into the region, expressed as follows:

$$[\lambda_s^{1-\phi}(T^{\text{reg}}) + \lambda_f^\phi] \frac{\partial T^{\text{reg}}}{\partial z} = Q^{\text{reg}} \text{ [W/m}^2\text{]}, \quad (5.13)$$

where Q is the heatflux going into the region. The superscript reg denotes the regional model. The values for the rock properties, the fluid properties and the heatflux are in Section 4.1.1.

- At $t = 0$, transfer the temperature state associated with the reservoir area to the reservoir model.
- At $t > 0$, perform a transient thermo-hydraulic with temperature-dependent thermal conductivity setup on the reservoir model. The top boundary condition of the reservoir model for thermal process is a heterogeneous Dirichlet type from the temperature state of the regional model, expressed as follows:

$$T_{\text{top}}^{\text{res}} = T^{\text{reg}}(x_{\text{top}}^{\text{res}}, y_{\text{top}}^{\text{res}}, z_{\text{top}}^{\text{res}})^\circ\text{C}, \quad (5.14)$$

while, the top boundary condition of the reservoir model for hydraulic process is also a heterogeneous Dirichlet type from the hydraulic gradient of the regional model, expressed as follows:

$$P_{f,\text{top}}^{\text{res}} = P_f^{\text{reg}}(x_{\text{top}}^{\text{res}}, y_{\text{top}}^{\text{res}}, z_{\text{top}}^{\text{res}}) \text{ MPa}, \quad (5.15)$$

where the superscript res denotes the reservoir model. The bottom boundary condition for thermal process is a Neumann type, averaging the heatflux of each cell in the regional model associated with bottom reservoir boundary, expressed as follows:

$$[\lambda_s^{1-\phi}(T^{\text{res}}) + \lambda_f^\phi] \frac{\partial T^{\text{res}}}{\partial z} = \overline{Q^{\text{reg}}(x_{\text{bottom}}^{\text{res}}, y_{\text{bottom}}^{\text{res}}, z_{\text{bottom}}^{\text{res}})} \text{ [W/m}^2\text{]}, \quad (5.16)$$

while, it is no-flow boundary condition for hydraulic process for the remaining boundaries. The values for the rock properties, the fluid properties and the allowed operating ranges are given in Section 4.1.2.

The illustrations of these steps are shown in Figure 5.3.

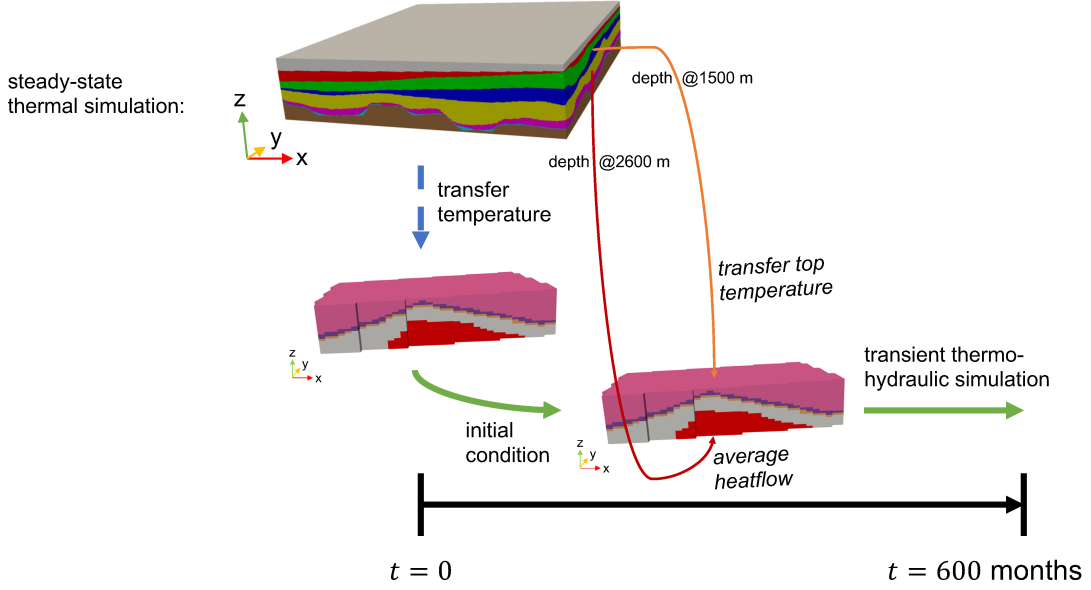


FIGURE 5.3: Numerical simulation setup for optimizing operating conditions in The Hague, Netherlands.

The steady-state thermal simulation in the regional model involves solving T with a total of 3,402,000 degrees of freedom. On the other hand, the transient thermo-hydraulic simulation for the reservoir model has 12,850,100 degrees of freedom, solving both P_f and T over 302 time steps (with the final time step set at 50 years (Motaghay, Pechinig, and Vogt, 2011)). The use of 302 time steps (in months) is essential to achieve numerical convergence within the finite element (FE) approach.

Executing this simulation setup demands the utilization of 48 cores along with 192 GB RAM on the CLAIX High-Performance Computing (HPC) infrastructure. This infrastructure is equipped with $2 \times$ Intel Xeon Platinum 8160 Processors "SkyLake", each running at 2.1 GHz and featuring 24 cores. The average computational time for each case is approximately 6 hours.

I consider 11 uncertain material parameters, consisting of thermal conductivity of each layer as in Table 4.4, heatflow, and permeability of the Delft sandstone, and 2 operation-related parameters, consisting of injection rate R_{inj} and injection temperature T_{inj} . The prior distributions for the parameters are in Table 5.2. The temperature and pressure response at both the injector and the producer is shown using the upper, mean, and lower bound values in Figure 5.4.

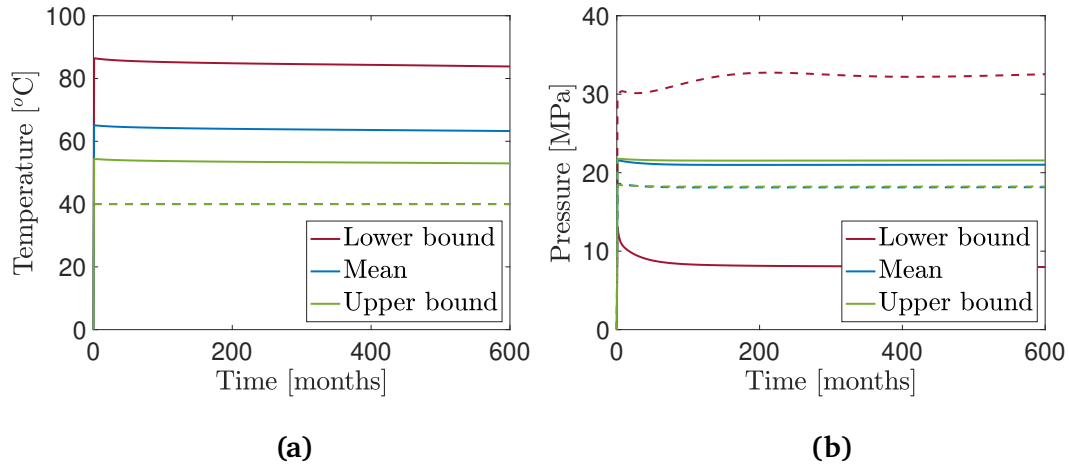


FIGURE 5.4: The variation in (a) temperature at producer from the lower bound to upper bound values and (b) pressure at injector and producer from the lower bound to upper bound values. The dashed lines in (b) denote pressure at injector, while the solid lines pressure at producer.

In Figure 5.4a, I have different temperatures in the producer by varying the uncertain material parameters and operation-related parameters from their lower bounds to their upper bounds. However, all responses exhibit a similar trend, a small decline over a span of 50 years with a spike in the early years. The pressure responses at the producer and injector follow a similar trend with the variation of uncertain material parameters and operation-related parameters from their means to upper bounds, see Figure 5.4b. However, using lower-bound values, the pressure response at the injector shows a different trend and exceeds 25 MPa, the breakdown pressure from the leak-off test (Willems et al., 2020). This situation poses a risk of unsafe operation since it could trigger the creation of fractures or reactivation of existing faults and fractures, causing induced seismicity (Mignan et al., 2015). However, using the lower bound values, the pressure response in the producer also exhibits a different trend compared to the use of the mean and upper bound values, and it remains below 20 MPa, which is the static water pressure in the producer calculated with hydraulic gradient. If the pressure at the producer cannot exceed 20 MPa, there will be no flow, which will result in no heat extraction.

TABLE 5.2: Prior distribution for each parameter used in the optimization of The Hague, Netherlands.

Parameters	Distribution	Mean	Standard deviation	Lower bound	Upper bound
λ_N [W/(m°C)]	Truncated normal	2.80	0.73	1.80	4.00
λ_{CK} [W/(m°C)]	Truncated normal	2.26	0.44	1.73	2.99
λ_{KN} [W/(m°C)]	Truncated normal	2.50	0.73	1.82	3.71
λ_S [W/(m°C)]	Truncated normal	3.75	0.89	2.30	4.89
λ_{AT} [W/(m°C)]	Truncated normal	2.15	0.35	1.64	2.72
λ_{RB} [W/(m°C)]	Truncated normal	2.96	1.03	1.47	4.66
λ_{ZE} [W/(m°C)]	Truncated normal	3.58	1.14	1.71	4.96
λ_{RO} [W/(m°C)]	Truncated normal	3.90	1.18	1.96	4.81
λ_{DC} [W/(m°C)]	Truncated normal	2.12	1.07	1.47	3.88
Q [W/m ²]	Uniform	-	-	0.06	0.07
k [m ²]	Uniform	-	-	2.96×10^{-14}	1.97×10^{-12}
R_{inj} [L/s]	Uniform	-	-	41.67	45.83
T_{inj} [°C]	Uniform	-	-	35	40

5.2 Optimization formulation

The aim of this chapter is to find optimum operating conditions considering uncertainties and safety. Hence, in this section, I present different optimization formulations for both the synthetic case and the The Hague case that might be suitable to achieve my goal. It is important to note that I primarily utilize the synthetic case to explore most of the formulations and gain understanding of each possible approach. The key considerations within the formulation should be:

- the objective function is subject to uncertainties. The primary question here refers to which statistical moment or formulation is used to represent the objective function in the optimization formulation.
- some constraints are subject to uncertainties. The primary question here concerns how to integrate these constraints into the optimization formulation. Possible options may include using the worst-case value of the constraint, utilizing the mean value of the constraint, or employing a reliability-based approach (Bogoclu, Roos, and Nestorović, 2021; Moustapha et al., 2016).

There are many ways to formulate an optimization problem that takes into account the two key factors mentioned above. In Section 2.3, I already discussed both implicit and explicit methods, along with their respective consequences. I now focus on the explicit formulation, where the safety constraint, in particular, is incorporated directly into the optimization formulation.

Furthermore, in terms of the formulation of the objective function, I adopt the mean objective function within the optimization formulation, a common approach found in many studies (Qin et al., 2023; Yan, Gudala, and Sun, 2023; Wang et al., 2023; Bogoclu, Roos, and Nestorović, 2021; Schulte et al., 2020; Arnold et al., 2016; Chen, Li, and Reynolds, 2012). Furthermore, I acknowledge the recent development of a formulation that uses the superquantile concept to further account for risk within the objective function, known as the "risk-averse formulation" (Luo et al., 2023; Chaudhuri et al., 2022; Rockafellar, Uryasev, et al., 2000). This formulation emphasizes the importance of the tail of the objective function distribution (Luo et al., 2023; Chaudhuri et al., 2022; Rockafellar, Uryasev, et al., 2000). Although this formulation is not addressed in this dissertation, it warrants further investigation in future research.

5.2.1 Optimization for the synthetic case

In this synthetic case, my goal is to minimize the total depletion time obtained by solving an inverse problem, seeking the time at which the overall temperature across the computational domain is 40 °C. The expected upper limit of the total depletion time is 3 days.

There are two constraints: operation-related and safety constraints. The constraint for injection velocity (operation-related) is

$$1.0[\text{m/s}] \leq u_D \leq 10.0[\text{m/s}], \quad (5.17)$$

While the safety constraint is the following:

$$\max(P_f(x = 0, t; u_D, \mathbf{k})) < 5[\text{MPa}], \quad (5.18)$$

For this synthetic case, I explore four possible optimization formulations, shown in Figure 5.5. The detailed formulation is as follows:

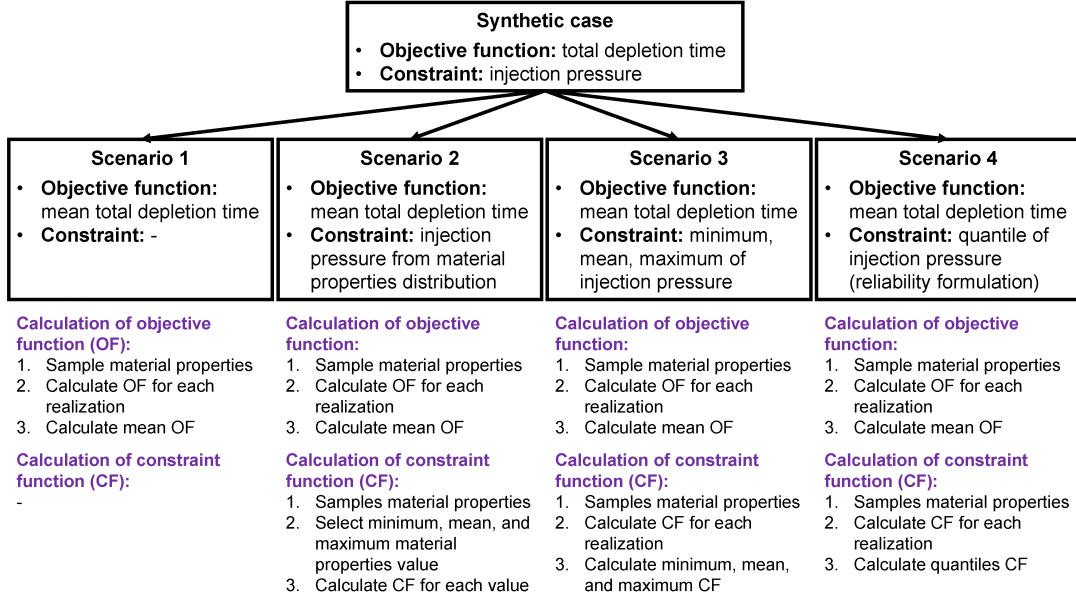


FIGURE 5.5: Different optimization formulations for synthetic case.

Scenario 1

Scenario 1 represents a constrained stochastic optimization problem, where the constraints in Equation 5.18 are ignored, see Figure 5.5. The formulation is expressed as follows:

$$\begin{aligned} \max_{u_D} \quad & \overline{t_{\text{depletion}}(u_D, \mathbf{k})} \text{ [days]} \\ \text{subject to} \quad & 1.0[\text{m/s}] \leq u_D \leq 10.0[\text{m/s}] \end{aligned} \quad (5.19)$$

where the overline denotes the mean.

Scenario 2

Up to this point, I still have difficulty in calculating the constraint in Equation 5.18 due to its dependence on uncertain permeability. In Scenario 2, I propose to calculate this constraint using the minimum, mean or maximum value of permeability, as illustrated in Figure 5.5. Meanwhile, the objective function continues to be calculated using the mean response from multiple realizations of permeability. The formulation utilizing the minimum permeability value is expressed as follows:

$$\begin{aligned} \max_{u_D} \quad & \overline{t_{\text{depletion}}(u_D, \mathbf{k})} \text{ [days]} \\ \text{subject to} \quad & 1.0[\text{m/s}] \leq u_D \leq 10.0[\text{m/s}] \\ & \max(P_f(x=0, t; u_D, \min(\mathbf{k}))) < 5 \text{ [MPa]} \end{aligned} \quad (5.20)$$

The formulation utilizing the mean permeability value is expressed as follows:

$$\begin{aligned} \max_{u_D} \quad & \overline{t_{\text{depletion}}(u_D, \mathbf{k})} \text{ [days]} \\ \text{subject to} \quad & 1.0[\text{m/s}] \leq u_D \leq 10.0[\text{m/s}] \\ & \max(P_f(x=0, t; u_D, \text{mean}(\mathbf{k}))) < 5 \text{ [MPa]} \end{aligned} \quad (5.21)$$

The formulation utilizing the maximum permeability value is expressed as follows:

$$\begin{aligned} & \max_{u_D} \quad \overline{t_{\text{depletion}}(u_D, \mathbf{k})} \text{ [days]} \\ & \text{subject to} \quad 1.0[\text{m/s}] \leq u_D \leq 10.0[\text{m/s}] \\ & \quad \quad \quad \max(P_f(x=0, t; u_D, \max(\mathbf{k}))) < 5 \text{ [MPa]} \end{aligned} \quad (5.22)$$

Scenario 3

In Scenario 3, I propose calculating the constraint in Equation 5.18 using the minimum, mean, or maximum injection pressure, as illustrated in Figure 5.5. Meanwhile, the objective function continues to be calculated using the mean response from multiple realizations of permeability. The formulation utilizing the minimum injection pressure is expressed as follows:

$$\begin{aligned} & \max_{u_D} \quad \overline{t_{\text{depletion}}(u_D, \mathbf{k})} \text{ [days]} \\ & \text{subject to} \quad 1.0[\text{m/s}] \leq u_D \leq 10.0[\text{m/s}] \\ & \quad \quad \quad \min(\max(P_f(x=0, t; u_D, \mathbf{k}))) < 5 \text{ [MPa]} \end{aligned} \quad (5.23)$$

The formulation utilizing the mean injection pressure is expressed as follows:

$$\begin{aligned} & \max_{u_D} \quad \overline{t_{\text{depletion}}(u_D, \mathbf{k})} \text{ [days]} \\ & \text{subject to} \quad 1.0[\text{m/s}] \leq u_D \leq 10.0[\text{m/s}] \\ & \quad \quad \quad \text{mean}(\max(P_f(x=0, t; u_D, \mathbf{k}))) < 5 \text{ [MPa]} \end{aligned} \quad (5.24)$$

The formulation utilizing the maximum injection pressure is expressed as follows:

$$\begin{aligned} & \max_{u_D} \quad \overline{t_{\text{depletion}}(u_D, \mathbf{k})} \text{ [days]} \\ & \text{subject to} \quad 1.0[\text{m/s}] \leq u_D \leq 10.0[\text{m/s}] \\ & \quad \quad \quad \max(\max(P_f(x=0, t; u_D, \mathbf{k}))) < 5 \text{ [MPa]} \end{aligned} \quad (5.25)$$

Scenario 4

Scenario 4 is the Reliability-Based Robust Design Optimization (RRDO) formulation, see Equation 2.18. The objective function is from the mean response and the constraint in Equation 5.18 is calculated using quantile concept. The constraint in Equation 5.18 can be re-written in quantile as follows:

$$\text{Quantile}_{\mathcal{P}_k^{user}}(\max(P_f(x=0, t; u_D, \mathbf{k}))) < 5 \text{ [MPa]}, \quad (5.26)$$

Based on Equation 5.26, the safety conditions are quantified by allowing $\mathcal{P}_k^{user}$ quantile to violate the safety limits, for example, the maximum allowable pressure. The formal optimization formulation is as follows:

$$\begin{aligned} & \max_{u_D} \quad \overline{t_{\text{depletion}}(u_D, \mathbf{k})} \text{ [days]} \\ & \text{subject to} \quad 1.0[\text{m/s}] \leq u_D \leq 10.0[\text{m/s}] \\ & \quad \quad \quad \Pr(\max(P_f(x=0, t; u_D, \mathbf{k})) > 5 \text{ [MPa]}) < \mathcal{P}^{user} \end{aligned} \quad (5.27)$$

where $\Pr$ denotes the probability.

To further understand the consequence of implementing Equation 5.26 as a constraint, I introduce **Scenario 5**, where I solve Equation 5.27 with permeability range 5×10^{-13} to 1×10^{-12} m². I use both the algorithm and the duquo Python library provided by Bogoclu, Roos, and Nestorović (2021).

A detailed procedure for solving the RRDO formulation is as follows: (Bogoclu, Roos, and Nestorović, 2021; Moustapha and Sudret, 2019)

1. Given an initial design u_D ,
2. Check whether u_D honor the constraint $1.0[\text{m/s}] \leq u_D \leq 10.0[\text{m/s}]$ or not,
3. If the constraint in Step 2 is violated, **the optimization is terminated**,
4. Calculate the safety constraint $\Pr(\max(P_f(x = 0, t; u_D, \mathbf{k}))[\text{MPa}])$ using the Monte Carlo approach and calculate its quantile $\mathcal{P}^{user}$,
5. Check whether the quantile in Step 4 is less than 5 MPa or not,
6. If the constraint in Step 4 is violated, **the optimization is terminated**,
7. Calculate the objective function $\overline{t_{\text{depletion}}(u_D, \mathbf{k})}$ [days],
8. If the maximum number of optimization step is achieved (in this case 300 steps), **the optimization is terminated**,
9. Update the design u_D and do Step 2.

Optimization is performed using the Genetic Algorithm approach (Bogoclu, Roos, and Nestorović, 2021).

5.2.2 Optimization for the The Hague case

For the The Hague case, my goal is to optimize the thermal power output using a doublet configuration. Therefore, I propose three different formulations: 1) the mean-properties formulation, 2) the mean-response formulation, and 3) the Reliability-Based Robust Design Optimization (RRDO) formulation. The mathematical expression for the objective function in all formulations is as follows:

$$P_{\text{th}} = \rho_f \times R_{\text{inj}} \times c_f \times (T_{\text{prod}}(\xi) - T_{\text{inj}}) \times 1 \times 10^{-6}, \quad (5.28)$$

where P_{th} is the thermal power output [MWth], R_{inj} is the injection rate [m³/s], $T_{\text{prod}}(\xi)$ is the temperature at producer [°C] (it is a function of uncertain material properties), T_{inj} is the injection temperature [°C], and 1×10^{-6} is a constant for the conversion from watt to megawatt.

There are two constraints for operating conditions in all formulations: injection rate and injection temperature constraints. The injection rate constraint is the following:

$$41.67[\text{L/s}] \leq R_{\text{inj}} \leq 45.83[\text{L/s}], \quad (5.29)$$

The injection temperature constraint is as follows:

$$35[^\circ\text{C}] \leq T_{\text{inj}} \leq 40[^\circ\text{C}], \quad (5.30)$$

The safety constraints include injection pressure and production pressure constraints. The injection pressure is the following:

$$P_{f,\text{inj}}(\xi) < 25[\text{MPa}], \quad (5.31)$$

The production pressure constraint is written below:

$$P_{f,prod}(\xi) > 20[\text{MPa}], \quad (5.32)$$

where both the injection and production pressure are a function of uncertain material properties.

Mean-properties formulation

I refer to the mean-properties formulation as the use of the mean/most likely value of each material property to represent uncertainties in the optimization formulation. It is similar to Equation 5.21 in Scenario 2 of the synthetic case. Several studies for subsurface flow optimization, including geothermal applications, use this formulation (Juliussen and Bjornsson, 2021; Wang, Li, and Reynolds, 2009; Sarma et al., 2008; Brouwer and Jansen, 2004). The optimization formulation for the The Hague case is the following:

$$\begin{aligned} \max_{R_{inj}, T_{inj}} \quad & \overline{\rho_f \times R_{inj} \times c_f \times (T_{prod}(\xi) - T_{inj}) \times 1 \times 10^{-6}} [MW] \\ \text{subject to} \quad & 41.67[\text{L/s}] \leq R_{inj} \leq 45.83[\text{L/s}] \\ & 35[^\circ\text{C}] \leq T_{inj} \leq 40[^\circ\text{C}] \\ & P_{f,inj}(\bar{\xi}) < 25[\text{MPa}] \\ & P_{f,prod}(\bar{\xi}) > 20[\text{MPa}], \end{aligned} \quad (5.33)$$

Mean-responses formulation

In this formulation, I use the mean injection pressure and production pressure as constraints in the optimization problem. It is similar to Equation 5.24 in Scenario 3 of the synthetic case. The formulation is as follows:

$$\begin{aligned} \max_{R_{inj}, T_{inj}} \quad & \overline{\rho_f \times R_{inj} \times c_f \times (T_{prod}(\xi) - T_{inj}) \times 1 \times 10^{-6}} [MW] \\ \text{subject to} \quad & 41.67[\text{L/s}] \leq R_{inj} \leq 45.83[\text{L/s}] \\ & 35[^\circ\text{C}] \leq T_{inj} \leq 40[^\circ\text{C}] \\ & \overline{P_{f,inj}(\xi)} < 25[\text{MPa}] \\ & \overline{P_{f,prod}(\xi)} > 20[\text{MPa}], \end{aligned} \quad (5.34)$$

Reliability-Based Robust Design Optimization formulation

I here use the Reliability-Based Robust Design Optimization (RRDO) formulation for the The Hague case. It is similar to Equation 5.27 in Scenario 4 of the synthetic case and the formulation in Section 2.3. The formulation is mathematically expressed as follows:

$$\begin{aligned} \max_{R_{inj}, T_{inj}} \quad & \overline{\rho_f \times R_{inj} \times c_f \times (T_{prod}(\xi) - T_{inj}) \times 1 \times 10^{-6}} [MW] \\ \text{subject to} \quad & 41.67[\text{L/s}] \leq R_{inj} \leq 45.83[\text{L/s}] \\ & 35[^\circ\text{C}] \leq T_{inj} \leq 40[^\circ\text{C}] \\ & \Pr((P_{f,inj}(\xi) > 25) \& (P_{f,prod}(\xi) < 20)) < \mathcal{P}^{user} [\text{MPa}], \end{aligned} \quad (5.35)$$

I use both the algorithm and the duqo Python library provided by Bogoclu, Roos, and Nestorović (2021).

A detailed procedure for solving the RRDO formulation for the The Hague case is as follows: (Bogoclu, Roos, and Nestorović, 2021; Moustapha and Sudret, 2019)

1. Given an initial design R_{inj}, T_{inj} ,
2. Check whether R_{inj} honor the constraint $41.67[\text{L/s}] \leq R_{inj} \leq 45.83[\text{L/s}]$ or not,
3. If the constraint in Step 2 is violated, **the optimization is terminated**,
4. Check whether T_{inj} honor the constraint $35[^\circ\text{C}] \leq T_{inj} \leq 40[^\circ\text{C}]$ or not,
5. If the constraint in Step 4 is violated, **the optimization is terminated**,
6. Calculate the safety constraint $\Pr(P_{f,inj}(\xi))$ and $\Pr(P_{f,prod}(\xi))$ using the Monte Carlo approach and calculate their quantile $\mathcal{P}^{user}$,
7. Check whether the quantiles in Step 6 is less than 25 MPa and more than 20 MPa consecutively or not,
8. If the constraint in Step 6 is violated, **the optimization is terminated**,
9. Calculate the objective function $\rho_f \times R_{inj} \times c_f \times (T_{prod}(\xi) - T_{inj}) \times 1 \times 10^{-6}$ [MW],
10. If the maximum number of optimization step is achieved (in this case 300 steps), **the optimization is terminated**,
11. Update the design R_{inj}, T_{inj} and do Step 2.

The optimization is performed using the Genetic Algorithm approach (Bogoclu, Roos, and Nestorović, 2021).

5.3 Workflow

Figure 5.6 shows the workflow for executing every optimization formulation presented in Equation 5.19 to 5.35. It is inspired by the works of Degen, Veroy, and Wellmann (2022) and Degen et al. (2021a).

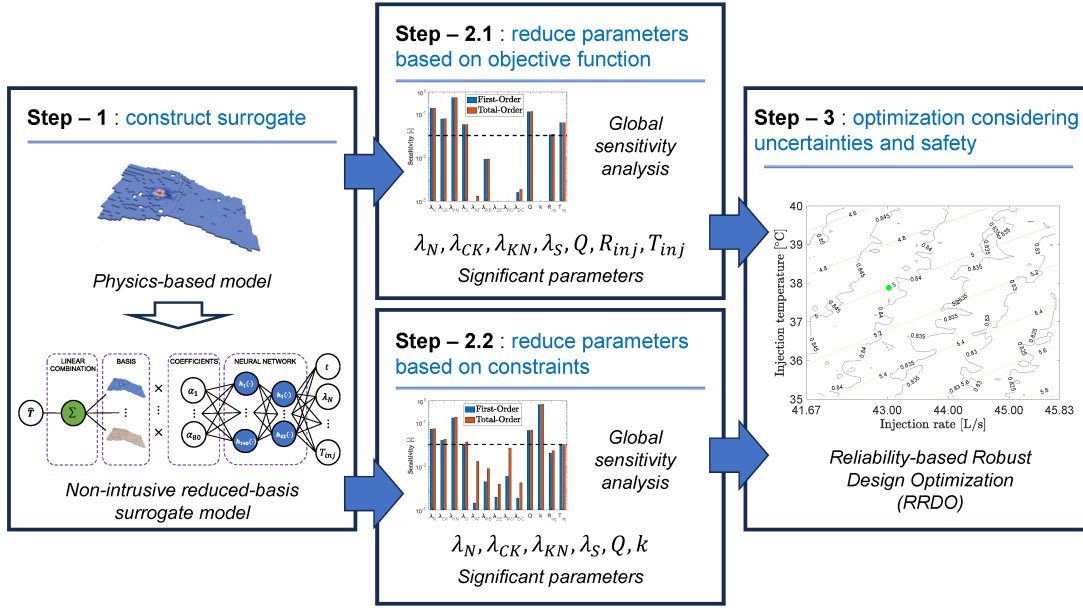


FIGURE 5.6: Workflow for conducting optimization under uncertainties. This workflow is inspired by the works of Degen, Veroy, and Wellmann (2022) and Degen et al. (2021a).

The detailed steps are the following:

- **construct a surrogate model using the non-intrusive reduced-basis (NI-RB) method**, see Section 2.1. The surrogate model provides a map from uncertain physical and operation-related parameters to the temperature and pressure state. The construction of surrogate models is required to reduce computational cost, enabling large-scale optimization under uncertainties.
- **select significant parameters based on the objective function (Step 2.1)**. I use variance-based Global Sensitivity Analysis (GSA), see Section 2.2.1, to identify significant parameters based on the objective function. Selecting significant parameters is critical to ensure robustness in the mean calculation in the objective function.
- **select significant parameters based on constraints (Step 2.2)**. I again use variance-based Global Sensitivity Analysis (GSA), see Section 2.2.1, to identify significant parameters based on the constraints. It is critical to ensure robust sampling for the safety constraints calculation.
- **optimization**. I solve different optimization formulations with significant parameters identified through GSA in the previous step. The use of surrogate models here is essential, as there will be more than a million model runs during the optimization process. I use the Scipy Python library (Virtanen et al., 2020) and the duquo Python library to perform the optimization (Bogoclu, Roos, and Nestorović, 2021).

5.4 Results

In this section, I begin by presenting the optimization results obtained from the five scenarios proposed for the synthetic case. These optimization results are intended

to enhance my understanding of the reliability constraint in the Reliability-Based Robust Design Optimization (RRDO) formulation. Following this, I discuss the results obtained from the three different optimization schemes in the The Hague case.

5.4.1 Synthetic case result

In the synthetic case optimization, scenarios 1-4 aim to understand the impact of different formulations of the safety constraint on optimization results (I will show the graphical solutions of these scenarios), while scenarios 4-5 aim to assess the impact of material properties ranges on the significance of the safety constraint.

Surrogate model

I utilize 50 training and test samples, each consisting of 101 time steps, to construct my surrogate model. Table 5.3 presents details on the number of basis functions, tolerance, L_2 error estimator, offline time, and online time of the surrogate model for the temperature and pressure state. The approximation errors of both the temperature and pressure surrogate models, presented in Table 5.3, are sufficient to not influence the later global sensitivity analysis and optimization results. Table 5.4 shows the architecture and hyperparameters of the surrogate models.

TABLE 5.3: The performance of surrogate models used in the synthetic case. The approximation error is calculated using test samples. The values denoted with a red font color represent the total time needed to generate the training and test samples, while the black color represents the time needed to construct a surrogate model. The values in the brackets denote the time needed for a trained surrogate model to predict a new case with 31 time steps and the ones outside the brackets denote the time needed to predict a new case for a single time step.

Surrogate model	Tolerance	Number of basis functions	L_2 approximation error	Offline time [s]	Online time [s]
Temperature	p: 1×10^{-4}	14	9.1×10^{-3}	$3.2 \times 10^3 +$	2.5×10^{-4}
	t: 1×10^{-4}			1.8×10^4	(7.8×10^{-3})
Pressure	p: 1×10^{-4}	10	1.7×10^{-2}	$2.5 \times 10^3 +$	2.2×10^{-4}
	t: 1×10^{-4}			1.8×10^4	(6.8×10^{-3})

Note: p denotes parameter-cut tolerance, t denotes time-trajectory-cut tolerance.

TABLE 5.4: Architectures and hyperparameters of surrogate models used in the synthetic case.

Surrogate model	Neurons in layer 1	Neurons in layer 2	Learning rate	Number of epochs	Batch size
Temperature	52	140	1.5×10^{-4}	46,768	41
Pressure	52	140	1.5×10^{-5}	46,768	41

The offline and online time of the surrogate model is shown in Table 5.3. The offline time is defined as the time needed to train a surrogate model, considering the time needed to generate the training and test set. The online time is defined as the time that a surrogate model needs to predict a new case. Solving for pressure and temperature for a single time step using the finite element method takes 348.39 seconds. Consequently, I achieve more than 5 orders of magnitude acceleration in predicting the pressure and temperature states of a new case.

Global sensitivity analysis

I perform two global sensitivity analyses using total depletion time (objective function in the optimization process) and pressure at the left side (safety constraint in the optimization process).

Figure 5.7a shows the global sensitivity analysis with total depletion time as the objective function. I can see that only injection velocity u_D is the significant parameter. It makes sense since the permeability only affects the pressure build-up at the left side. Hence, for the mean calculation in the optimization process, I will only use injection velocity u_D .

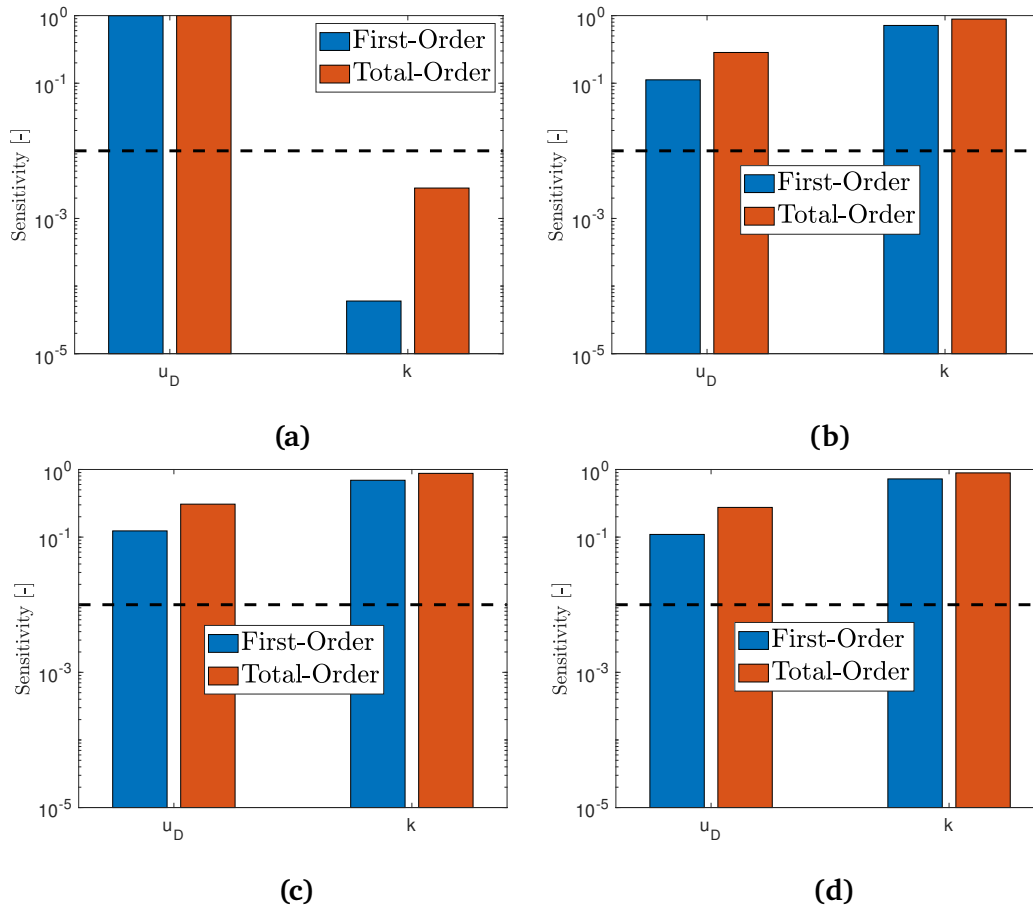


FIGURE 5.7: Global sensitivity analysis results utilizing synthetic case with (a) total depletion time, (b) pressure at the left side at $t = 1.0$ days, (c) pressure at the left side at $t = 5.0$ days, and (d) pressure at the left side at $t = 10.0$ days.

Figure 5.7b, c, and d show the global sensitivity analysis with pressure on the left side as the objective function at different time steps. Both permeability k and

injection velocity u_D are significant. For the calculation of the safety constraint in the optimization process, I will use these two parameters.

Optimization

Figure 5.8 shows the optimization results using Scenario 1 - 5. For Scenario 1 - 3, optimization is conducted using a trust region method in the Scipy library with 100 realizations for calculating the mean of the objective function and a maximum of 300 iterations. The non-intrusive reduced-basis (NI-RB) surrogate model enables solving this optimization problem in 15 seconds. While, for Scenario 4 - 5, I employ 1,000 initial guesses to solve the Reliability-based Robust Design (RRDO) formulation (Equation 5.27) with a maximum of 300 iterations. I use 100 realizations to calculate the mean of the objective function and 1,000,000 realizations for calculating the safety constraints. With the NI-RB surrogate model, I can solve this optimization problem in 21.37 minutes. All optimizations are performed on a single core on an Apple MacBook Pro, equipped with an M1 chip and 8 GB RAM.

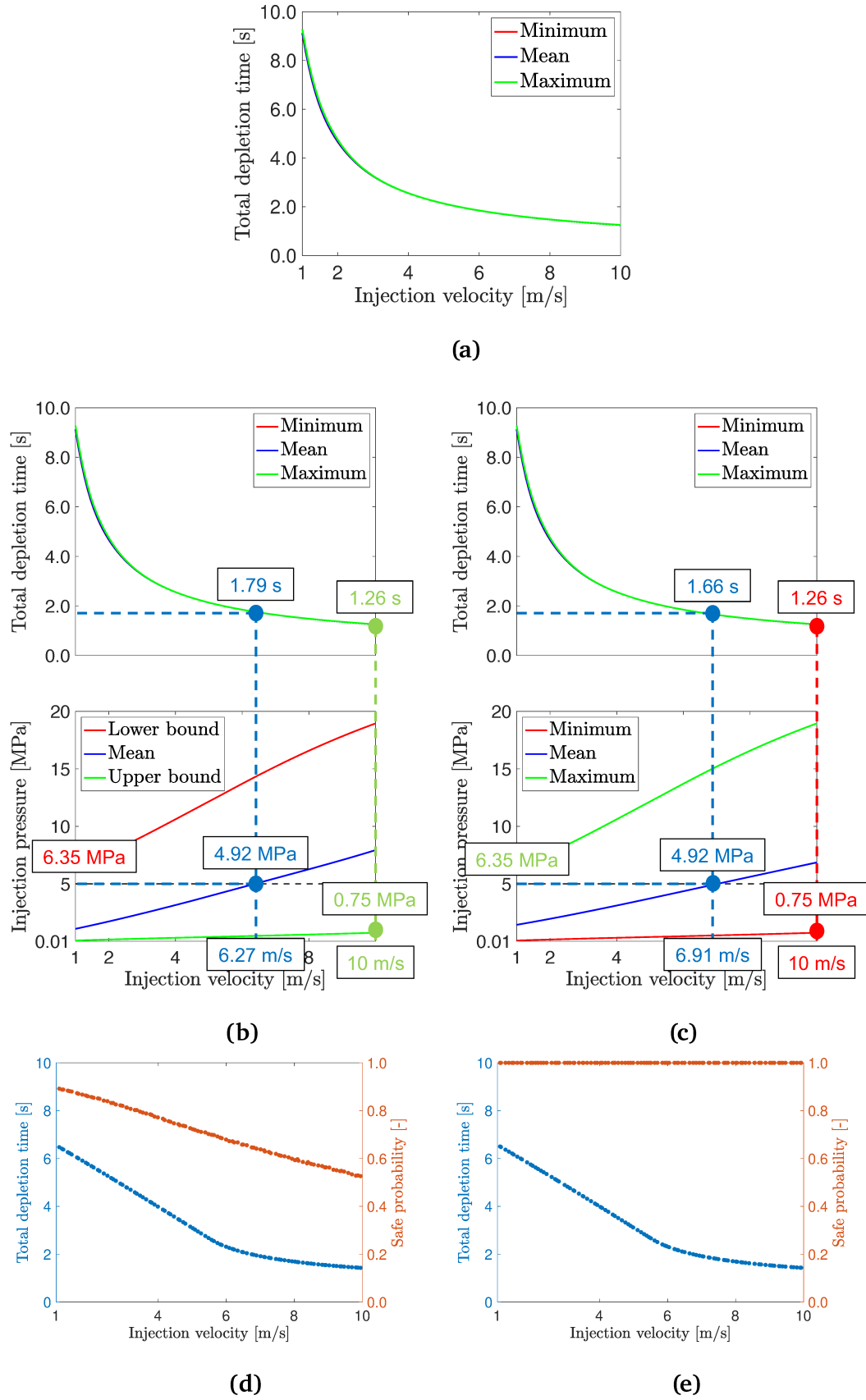


FIGURE 5.8: Optimization results with the synthetic case using (a) scenario 1, (b) scenario 2, (c) scenario 3, (d) scenario 4, and (e) scenario 5. In (d) and (e), the blue dots denote the objective function and the orange dots the safety constraint.

If we observe the spread of objective function responses under different realizations in Figure 5.8a - c, we confirm the global sensitivity analysis result presented in Figure 5.7a, indicating that the objective function is not sensitive to the variation in permeability. Here, I can infer that my objective function behaves like a deterministic function. In contrast, the safety constraint (the injection pressure) exhibit a wide spread when considering varying injection velocity and permeability. This observation aligns with the results of the global sensitivity analysis depicted in Figures 5.7b - d.

In Scenario 1 shown in Figure 5.8a, I remove the injection pressure constraint and keep the velocity constraint. This yields a minimum objective function of 1.26 seconds, corresponding to an injection velocity of 10 m/s. Although this allows us to achieve the fastest depletion time, it comes at the cost of disregarding safety measures, thereby accepting a significant probability of inducing fractures and creating unsafe operating conditions.

In Scenario 2 and 3 shown in Figure 5.8b and c consecutively, I focus on how to properly impose the safety constraint which is sensitive to variations in permeability and injection velocity. In these scenarios, I try to impose the safety constraint in a deterministic way.

In Scenario 2 shown in Figure 5.8b, I use the lower bound, mean, and upper bound permeability values to calculate the safety constraint, alongside the dynamically varying injection velocity during the optimization process. When using the lower bound permeability value, the injection pressure always exceeds the 5 MPa limit for any injection velocity. Consequently, there is no optimization solution within this injection velocity range, emphasizing the need for an injection velocity significantly below 1 m/s to ensure safe operating conditions. When using the mean permeability value to calculate the safety constraint, my injection velocity is limited to 6.27 m/s, resulting in a depletion time of 1.79 seconds. Depleting faster is not feasible, as the injection pressure is already close to the 5 MPa limit. With the upper bound permeability value, all injection velocity values will not create the injection pressure exceed the 5 MPa limit. Hence, I can achieve the fastest depletion time of 1.26 seconds with a 10 m/s injection velocity. The key consideration now is determining which solution to implement in my operations, given the absence of specific safety measures for each solution. The intuitive choice would be the mean-permeability solution, a question arising: Is this solution representative of the entire variation in injection pressure?

An intuitive next step (Scenario 3) involves utilizing the minimum, mean, and maximum values of the injection pressure response to determine the safety constraint. The minimum and maximum injection pressure responses provide the same bounds for the injection pressure response as in Figure 5.8b. An interesting observation pertains to the mean response. Using the mean injection pressure response as a safety constraint allows for a higher injection velocity, consequently achieving a faster depletion time. When comparing the utilization of mean permeability value and mean injection pressure for calculating the safety constraint, questions arise: even though both scenarios result in the same injection pressure value, which scenario is safer?

I further analyze this safety measure by investigating the distribution of the injection pressure response produced in Scenario 2 and 3. I generate 1,000,000 realizations using the Latin Hypercube sampling method. I analyze the distribution of (5 – injection pressure) for Scenarios 2 and 3, as illustrated in Figure 5.9, as the pressure limit is 5 MPa. In Scenario 2, over 34% of my realizations exceed the 5 MPa pressure limit, whereas in Scenario 3, it is more than 36%. Consequently, in this synthetic case study, utilizing the mean permeability value results in safer operating conditions.

However, across Scenario 1 - 3, it remains unclear how to quantitatively impose a specific safety level in the optimization process. This ambiguity arises because the use of the mean permeability value or the mean injection pressure response does not universally apply to any distribution shape. My aim is to establish a consistent safety measure that can be explicitly incorporated into the optimization process, reflecting an accepted level of risk. For instance, in a Gaussian distribution, a mean value is associated with 50% realization. The use of quantiles offers a consistent approach, independent of distribution shape (Witte and Witte, 2017). This concept is implemented in the RRDO formulation in the RRDO formulation in Scenario 4 - 5.

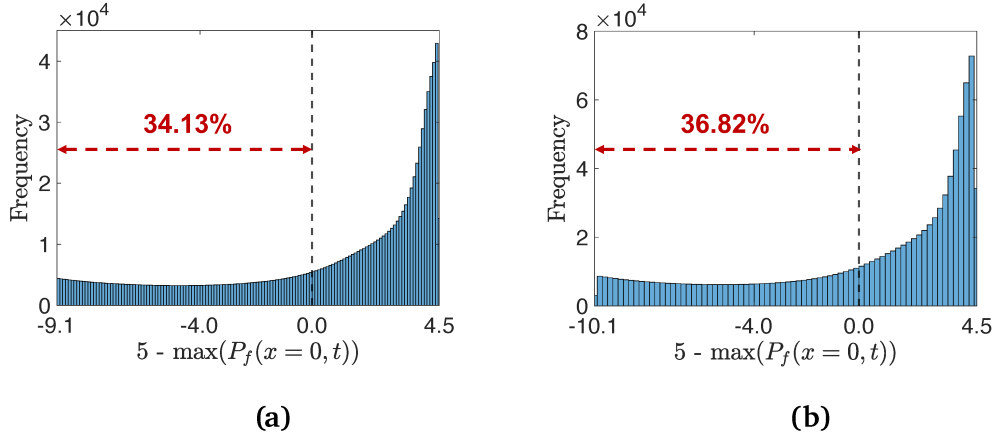


FIGURE 5.9: The distribution of $(5 - P_f(x = 0, t))$ is depicted for (a) Scenario 2 and (b) Scenario 3. The black dashed line represents the 5 MPa pressure limit, and the red dashed line indicates all realizations exceeding 5 MPa.

Figure 5.8d (Scenario 4) shows the optimization result using the full range of permeability and injection velocity. We can see that as the injection velocity decreases, the safety probability increases. In addition, a reduction in injection velocity results in an increase in the total depletion time, indicating the need for a considerably longer time to extract all the heat from the domain. With this result, I can then now select a suitable injection velocity that gives an acceptable depletion time and risk.

Scenario 5 illustrated in Figure 5.8e represents an extreme condition characterized by a highly permeable reservoir, allowing safe injection at any velocity within the specified range. In such a case, the safety constraint becomes negligible, allowing for a wide range of injection velocities.

If we now look at Scenario 3 - 5, those scenarios utilize the injection pressure responses to calculate the safety constraint. This differs from Scenario 2, where a specific permeability value is assumed in the calculation. Due to limited measurement data, the permeability characterization is often inaccurate (Degen, Veroy, and Wellmann, 2022; Schulte et al., 2020). Therefore, assuming a particular permeability value in the calculation of the safety constraint introduces bias to the safety constraint.

5.4.2 The Hague case result

From Figure 5.4a, we can see that the temperature at the producer is relatively constant until 600 months. Hence, I only need to perform the optimization in the last time step.

Surrogate model

I employ 300 training samples and 75 test samples, each consisting of 302 time steps, to construct my surrogate model. Table 5.5 presents details on the number of basis functions, tolerance, L_2 error estimator, offline time, and online time of the surrogate model for the temperature and pressure state. The approximation error for both temperature (looking at the approximation error in Chapter 4, the accuracy of the temperature surrogate model here is acceptable) and pressure surrogate model are acceptable, guaranteeing not to influence the later global sensitivity analysis and optimization results. In Table 5.6, I present the architecture and hyperparameters of the surrogate models.

TABLE 5.5: Performance of the surrogate models used in the The Hague case. The approximation error is calculated using test samples. The values denoted with a red font color represent the total time needed to generate the training and test samples, while the black color represents the time needed to construct a surrogate model. The values in the brackets denote the time needed for a trained surrogate model to predict a new case with 302 time steps and the ones outside the brackets denote the time needed to predict a new case for a single time step.

Surrogate model	Tolerance	Number of basis functions	L_2 approximation error	Offline time [s]	Online time [s]
Temperature	p: 1×10^{-4} t: 1×10^{-4}	80	3.4×10^{-2}	$8.4 \times 10^3 +$ 6.8×10^6	5.9×10^{-4} (1.7×10^{-1})
Pressure	p: 1×10^{-4} t: 1×10^{-4}	31	3.4×10^{-2}	$4.6 \times 10^3 +$ 6.8×10^6	6.3×10^{-4} (1.9×10^{-1})

Note: p denotes parameter-cut tolerance, t denotes time-trajectory-cut tolerance.

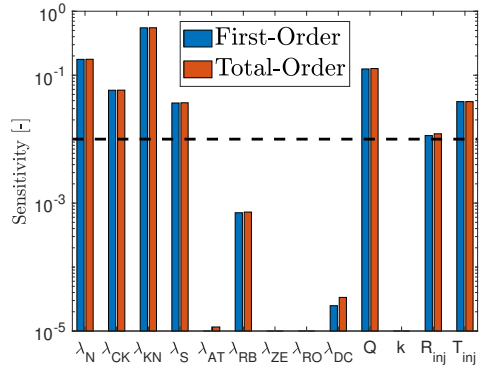
TABLE 5.6: Architectures and hyperparameters of surrogate models used in the The Hague case.

Surrogate model	Neurons in layer 1	Neurons in layer 2	Learning rate	Number of epochs	Batch size
Temperature	52	140	1.5×10^{-4}	46,768	2,330
Pressure	92	119	2.9×10^{-5}	46,459	5,337

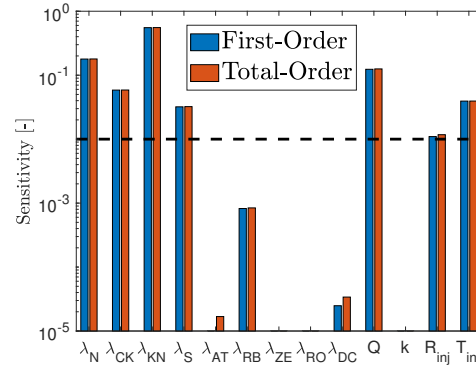
The offline and online time of the surrogate model is shown in Table 5.4. Solving for pressure and temperature for a single time step using the finite element method takes 60 seconds with 48 cores. Consequently, I achieve more than 6 orders of magnitude acceleration in predicting the pressure and temperature states of a new case.

Global sensitivity analysis

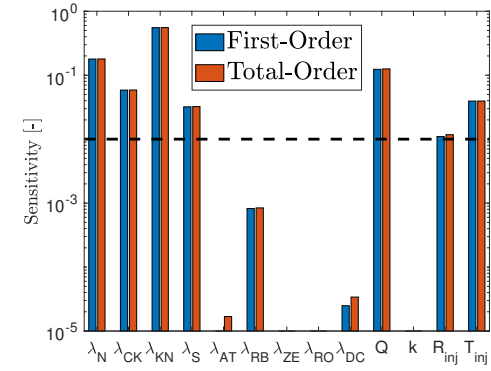
I perform three global sensitivity analyses using thermal power (objective function in the optimization process), pressure at injector, and pressure at producer (safety constraints in the optimization process). I choose 100, 300, and 600 months in Figure 5.10 representing early, mid, and final time steps. Note that I cannot produce power and pressure sensitivity results at $t = 0$ because there is no flow.



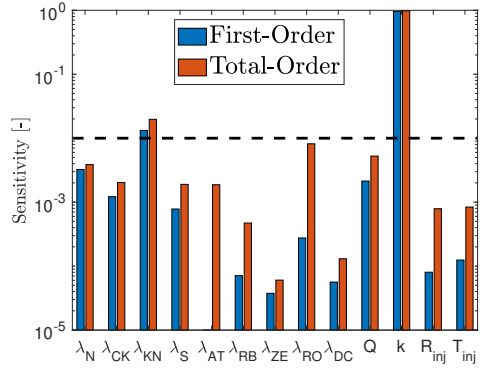
(a)



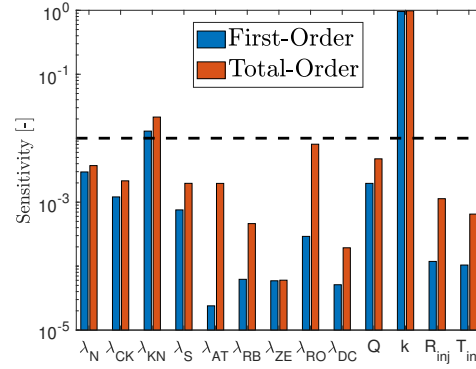
(b)



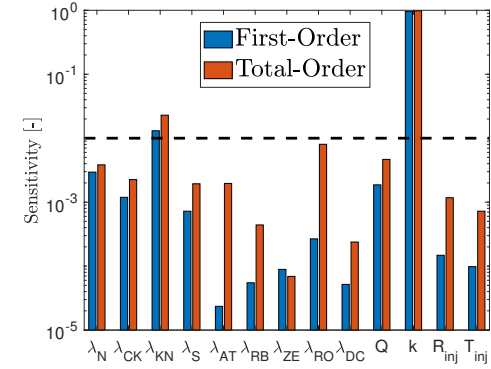
(c)



(d)



(e)



(f)

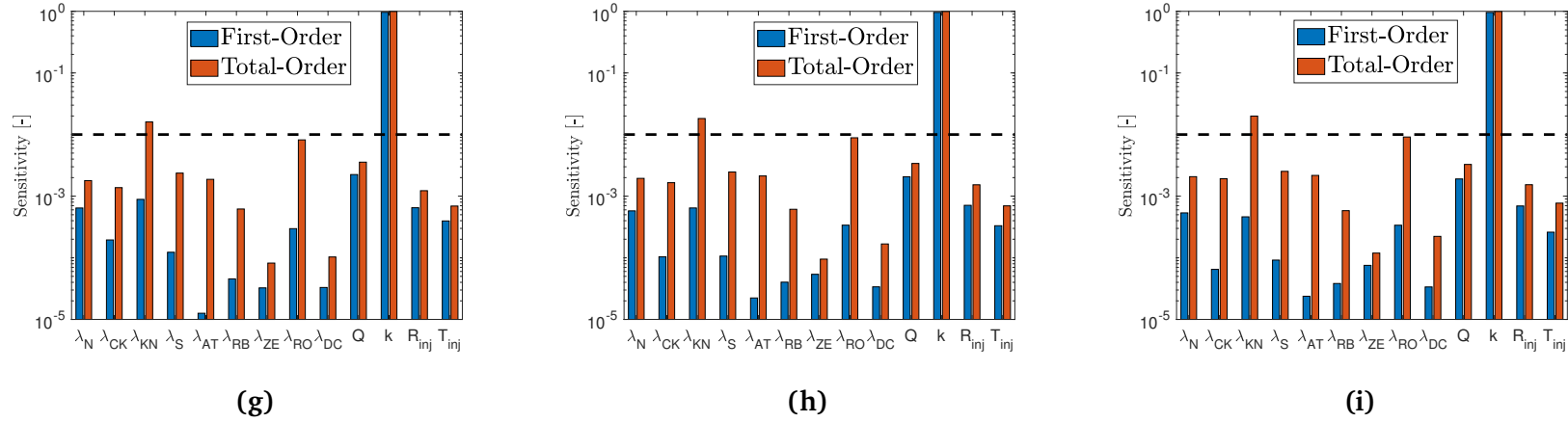


FIGURE 5.10: Global sensitivity analysis for The Hague case: (a), (b), (c) are with thermal power objective function for 100, 300, and 600 months consecutively; (d), (e), (f) are with pressure at injector for 100, 300, and 600 months consecutively; (g), (h), (i) are with pressure at producer for 100, 300, and 600 months consecutively.

In Figure 5.10, I observe that there is no change in the sensitivity values for all objective functions at all time steps. This indicates that there is no change in physical process occurring in the system, and it validates my approach to only perform optimization at the final time step.

In Figure 5.10a - c, the significant parameters are λ_N , λ_{CK} , λ_{KN} , λ_S , Q , R_{inj} , and T_{inj} . The significance of λ_N , λ_{CK} , λ_{KN} , λ_S , and Q appears due to the inclusion of the production temperature in the calculation of thermal power. Since R_{inj} and T_{inj} appear explicitly in the thermal power calculation, they are also significant. Note that it is important that we have significant R_{inj} and T_{inj} identified through global sensitivity analysis because we will optimize the mean response of the optimization objective function by changing these two parameters. If the optimization objective function is not significant towards the operation-related parameters, it leads to an unreasonable situation in which the optimum operating conditions could be any value and the heat extraction operation is solely dependent on reservoir properties. For the mean calculation in the optimization process, I will only sample λ_N , λ_{CK} , λ_{KN} , λ_S , Q , R_{inj} , and T_{inj} .

In Figure 5.10d - f and Figure 5.10g - i, we can see that both pressure at the producer and at the injector are only sensitive to λ_{KN} and k . This is because only layer KN is permeable and the injection-production process occurs only in that layer. For the calculation of the safety constraints, I will only sample λ_{KN} and k .

Optimization

I use 1,000 initial guesses to solve the Reliability-based Robust Design (RRDO) formulation (Equation 5.35) with a maximum of 300 iterations, 100 realizations for calculating the mean of the objective function, and 1,000,000 realizations for calculating the safety constraints. With the use of the non-intrusive reduced-basis (NI-RB) surrogate model, I can solve this optimization problem in 33.33 hours. While, for solving optimization problem in Equation 5.33 and 5.34, I use the trust region algorithm with a maximum of 300 iterations. I solve each of these problems in 10.35 minutes using the NI-RB surrogate model. These optimizations are performed with a single core on an Apple Macbook Pro, equipped with an M1 chip and 8 GB RAM.

Figure 5.11a shows the thermal output power and safe probability in the contour plot and the dots are the optimization solutions. Mottaghy, Pechnig, and Vogt (2011) propose an injection rate of 41.67 L/s and injection temperature of 40 °C to extract heat from the Delft Sandstone (the black dot in Figure 5.11a). If I put this solution on the power and safe-probability contours, it is a safe design with a success probability of more than 85%, the safest among the other optimization results. However, employing this solution would only enable the fulfillment of the 5 MW power output target in the early years (less than 5 years), and in the subsequent years until year 50, the reservoir would only be capable of delivering around 4.6 - 4.8 MW (see the black line in Figure 5.11b).

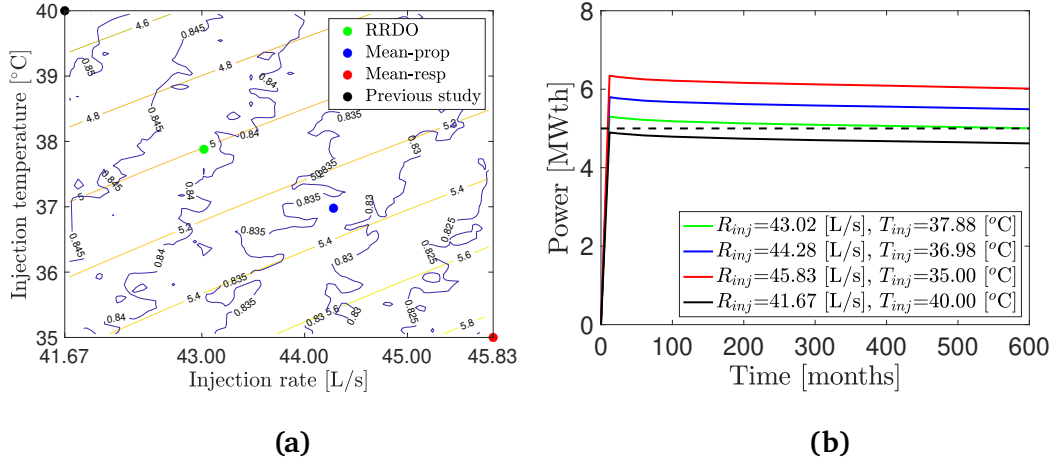


FIGURE 5.11: Optimization for heat extraction in The Hague, Netherlands considering uncertainties and safety: (a) the contour plot of the optimization results and (b) the thermal output power over time. In (a), the yellow lines denote the thermal output power and the purple lines the safe probability.

The blue dot in Figure 5.11a represents the optimization solution using the mean-properties formulation (Equation 5.33). It is important to note that employing mean properties values for directly computing the safety constraints is different from using several realizations of properties to compute the safety constraints, then taking their mean responses. This is because the safety constraints distributions are skewed. Fortunately, utilizing mean properties values here yields safer conditions compared to using mean responses, albeit at the cost of a lower thermal power. However, it is important to note that relying on mean properties values does not consistently guarantee a safe condition, as we lack a direct measure of safety conditions.

If I observe Equation 5.34 and 5.35, they are fundamentally the same. If I transform the safety constraints in Equation 5.34 into quantile formulation, they are associated with the allowed failure probability (62% safe probability). Consequently, the red dot in Figure 5.11a (the solution to mean-responses formulation) should not exist. However, because of the bounds, the solution takes the highest injection rate and lowest injection temperature such that high thermal power is produced, violating the safety constraints. Finally, for The Hague setup, utilizing the mean-responses formulation yields optimum operating conditions at a 62% safe probability.

Using Equation 5.35, I can now control the safety condition based on the quantile. To meet the 5 MW target while ensuring a high safe probability, the optimization formulation yields the solution represented by the green dot in Figure 5.11a. This selection is associated with more than 84% safe probability.

5.5 Discussions

The five optimization scenarios in the synthetic case illustrate how different constraint formulations influence the optimization outcome. Scenario 1, without any safety constraint, selects the maximum injection velocity, which leads to injection pressures exceeding 5 MPa and therefore achieves the fastest depletion time. In a real-world situation, such an unconstrained strategy could have catastrophic consequences.

Scenarios 2 and 3 replace the stochastic constraint with deterministic ones to reduce computational effort. From Equation 5.2, pressure varies inversely with the permeability for a given velocity. Consequently, computing the constraint using the

minimum permeability in Scenario 2 is equivalent to using the maximum pressure in Scenario 3, and similarly for maximum permeability and minimum pressure. The primary difference arises from the use of the mean: Scenario 2 uses the mean permeability, while Scenario 3 uses the mean pressure. Since the mean pressure (in Scenario 3) is proportional to $\text{mean}(1/k)$, this formulation results in a looser constraint, allowing higher injection velocities and therefore shorter depletion times compared to Scenario 2 (which uses $1/\text{mean}(k)$). However, both scenarios neglect the tail of the pressure distribution (Figure 5.9) (the pressure distribution is non-Gaussian), leading to a higher injection velocity and a less safe probability. Compared with Scenario 4, which accounts for the full pressure distribution, both Scenario 2 and 3 exhibit lower safe probabilities despite their computational simplicity (common formulations for the optimization problem).

Scenario 4 evaluates the safety constraint using the full distribution of pressure rather than a deterministic one, making it the baseline for comparison. In this synthetic case, the optimal control computed with the mean-permeability formulation (Scenario 2) achieves approximately 62% compliance with the 5 MPa limit (38% violation), while the mean-pressure formulation (Scenario 3) achieves about 59% compliance (41% violation). Thus, Scenario 2 is marginally less risky than Scenario 3. However, both scenarios fail to provide explicit control over tail risk, whereas Scenario 4 allows the user to enforce a specified violation probability. For example, if we require at least 80% compliance (allowing 20% violation), the maximum injection velocity must be limited to 3.2 m/s, resulting in a depletion time of 4.7 seconds — longer and more conservative than both Scenario 2 and 3.

In real-world applications where permeability is uncertain, applying Scenarios 2 or 3 can be hazardous. Localized low-permeability zones can cause pressures to exceed safe limits, potentially inducing fractures or seismicity. Scenario 4 offers a more robust alternative by allowing the operator to specify an acceptable safety limit. In contrast, cases such as Scenario 5, where the reservoir is extremely permeable, are rare and generally unrealistic.

The results of both the synthetic case and the The Hague case demonstrate that incorporating the quantile concept into the safety constraint offers a quantitative measure of the safety level, regardless of the distribution shape of the responses (Witte and Witte, 2017). The key point here is that the safety constraint carries epistemic uncertainties. Hence, calculating this safety constraint with a deterministic setup by assuming particular material properties values as is done in Figure 5.8b and mean-properties scheme in Figure 5.11 introduces a bias. Moreover, neglecting this safety constraint may lead to either 1) successful operations if extreme conditions exist, e.g., extremely permeable reservoir, see Figure 5.8e, or 2) unsuccessful operations due to miscalculated system failure (there exists a significant probability for system failure), see Figure 5.8a, minimum and maximum scenario in Figure 5.8c, and mean-responses scenario in Figure 5.11.

Among the three formulations applied to the The Hague case study, the Reliability-based Robust Design Optimization (RRDO) formulation emerges as the preferred method for ensuring a consistent measure of safety conditions and facilitating the optimal selection of operating conditions based on acceptable safety risk and thermal power targets. In contrast, the use of mean-properties and mean-responses formulations does not guarantee consistent safety conditions, as their results are influenced by the skewness of the safety constraint distribution. A non-robust formulation may result in the selection of operating conditions that could endanger heat extraction operations in The Hague, potentially leading to induced seismicity. Finally, for The Hague case, the

optimum operating conditions are 43.02 L/s injection rate and 37.88 °C, guaranteed with more than 84% safe probability.

A challenge to solve the Reliability-based Robust Design Optimization (RRDO) formulation is the computational cost. RRDO requires millions to billions of realizations to robustly compute the failure probabilities and the mean of the objective function. With the use of surrogate models, constructed via the non-intrusive reduced-basis (NI-RB) method, I can reduce the computational cost of performing model runs without introducing physical inconsistency in the predictions, hence, enabling efficient handling of millions to billions of computations. Moreover, the use of global sensitivity analysis further reduces the computational cost related to the number of realizations needed to achieve a statistically robust calculation of failure probabilities and the mean of the objective function.

If we look at Figure 5.11a, we can see that we can achieve 5 MW thermal output with any injection rate and injection temperature between 37 - 40 °C. Alternatively, it is possible to achieve nearly 5.4 MW of thermal power while maintaining a safe probability of above 84%. However, in order to make a more informed decision in this scenario, additional economic constraints or considerations related to power plant design should be taken into account.

To increase my confidence in selecting operating conditions related to the safety risk, I require more measurement data, specifically transient pressure and temperature measurement data. This data helps to calibrate the uncertain reservoir properties. A more accurate characterization of the rock properties through this calibration process will enable us to make well-founded decisions about optimum operating conditions.

Recent studies, such as those by Luo et al. (2024) and Chaudhuri et al. (2022), introduce a more robust and reliable framework for optimization under uncertainty based on the concept of Conditional Value-at-Risk (CVaR). This approach ensures an appropriate level of conservativeness, thereby safeguarding against catastrophic failure events. Such a method could be particularly beneficial for geothermal applications and warrants further investigation.

Chapter 6

Entropy production as a comprehensive objective function for global sensitivity analysis of transient thermo-hydraulic processes - Insight from The Hague, Netherlands

*"Life requires all the pieces,
not just the ones that we tend to like."*

— Craig LOUNSBROUGH

In connection with Chapter 5, it is crucial to understand the transient thermo-hydraulic behavior of the The Hague reservoir before undertaking any calibration or optimization. This becomes particularly significant since I have only initial temperature measurement data, lacking transient pressure and temperature data (see Chapter 2 and Chapter 5). Consequently, the relevant parameters that should be addressed in the model calibration or optimization process could be neglected. To address this, I introduce the use of entropy production, offering a comprehensive insight into the thermo-hydraulic process based on thermodynamics. Additionally, this chapter also highlights the use of the non-intrusive reduced-basis (NI-RB) method, facilitating large-scale global sensitivity analysis. This chapter is covered in Santoso et al. (2024).

6.1 Transient thermo-hydraulic problem

For understanding the transient thermo-hydraulic behavior in The Hague, Netherlands, I solve coupled fluid flow and heat transfer partial differential equations (PDEs), consisting of mass balance, Darcy law, and energy balance equations. The mass balance equation for the fluid phase is expressed as follows: (Cacace and Jacquey, 2017)

$$\frac{\partial \phi \rho_f}{\partial t} + \nabla \cdot (\rho_f \mathbf{u}_f) = R, \quad (6.1)$$

where ϕ is the porosity of porous medium [-], ρ_f is the density of fluid phase [kg/m³], t is time [months], $\mathbf{u}_f$ is the Darcy velocity [m/s], and R is the sink or source term

[kg/(m³ · s)]. The Darcy law is expressed as below: (Cacace and Jacquey, 2017)

$$\mathbf{u}_f = \frac{\mathbf{k}}{\eta_f} (\nabla P_f - \rho_f g \mathbf{z}), \quad (6.2)$$

where $\mathbf{k}$ is the permeability of the porous medium [m²], μ_f is the viscosity of fluid phase [Pa · s], P_f is the pore pressure [MPa], g is the gravitational acceleration, and $\mathbf{z}$ denotes the direction on the z axis where the gravitational force works. The energy balance equation assuming local thermal equilibrium between the fluid and the rock phase is expressed as follows: (Cacace and Jacquey, 2017)

$$\frac{\partial}{\partial t} ([\phi \rho_f c_f + (1 - \phi) \rho_s c_s] T) + \nabla \cdot (\rho_f c_f \mathbf{u}_f T - [\lambda_s^{1-\phi}(T) + \lambda_f^\phi] \nabla T) = \dot{H}, \quad (6.3)$$

where c_f is the fluid specific heat [J/(kg · °C)], ρ_s is the solid density [kg/m³], c_s is the solid specific heat [J/(kg · °C)], T is the temperature [°C], $\lambda_s(T)$ is temperature-dependent solid thermal conductivity [W/(m · °C)], and λ_f is fluid thermal conductivity [W/(m · °C)]. The temperature-dependent thermal conductivity is calculated as follows: (Bartels, Kühn, and Clauser, 2003)

$$\lambda_s(T) = \begin{cases} \frac{770}{350+T[^\circ\text{C}]} + 0.7 \text{ [W/(m}^\circ\text{C)]}, & T > 800^\circ\text{C}, \\ \left(\frac{770}{350+T[^\circ\text{C}]} + 0.7 \right) \left[\frac{\lambda_s^{ref}}{2.781} \times \left(1 - \frac{T[^\circ\text{C}]-20}{800-20} \right) + \left(\frac{T[^\circ\text{C}]-20}{800-20} \right) \right] \text{ [W/(m}^\circ\text{C)]}, & T < 800^\circ\text{C}, \end{cases} \quad (6.4)$$

where λ_s^{ref} is the reference solid thermal conductivity value [W/(m · °C)]. I employ a finite element (FE) approach to solve for variables P_f and T in my study. The numerical simulations are conducted using GOLEM, an open-source multiphysics simulator based on the MOOSE framework (Cacace and Jacquey, 2017).

Here, I use the same setup as in Section 5.1.2. Please refer to Section 5.1.2 for the detailed parameters and their distributions. Executing this simulation setup demands the utilization of 48 cores along with 192 GB RAM on the CLAIX High-Performance Computing (HPC) infrastructure. This infrastructure is equipped with 2 × Intel Xeon Platinum 8160 Processors "SkyLake", each running at 2.1 GHz and featuring 24 cores. The average computational time for each case is approximately 6 hours.

6.2 Entropy production

According to the Second Law of Thermodynamics, entropy is a measure of molecular disorder or randomness within a system (Kern and Weisbrod, 1967). Entropy production, on the other hand, quantifies the increase in entropy, characterizing the thermodynamic state and irreversibility of a system (Börsing et al., 2017; Bejan, 2013; Regenauer-Lieb et al., 2010). This concept serves as an abstraction to describe multiphysics processes, including thermo-hydraulic processes in geothermal applications. Bejan (2013) further formulates the volumetric rate of total entropy production $\dot{S}$ in a saturated porous medium, expressed as a linear combination of the irreversible heat transfer $\dot{S}_{\text{therm}}$ and fluid flow friction $\dot{S}_{\text{visc}}$ contributions:

$$\dot{S} = \dot{S}_{\text{therm}} + \dot{S}_{\text{visc}} = \frac{\lambda_s^{1-\phi}(T) + \lambda_f^\phi}{\left(\frac{T_{\text{top}} + T_{\text{bottom}}}{2} \right)^2} (\nabla T)^2 + \frac{\eta_f}{k_0 \left(\frac{T_{\text{top}} + T_{\text{bottom}}}{2} \right)} \mathbf{u}_f^2, \quad (6.5)$$

where T_{top} is the temperature at the top boundary, T_{bottom} is the temperature at bottom boundary, and k_0 is the absolute permeability. The formulation in Equation 6.5 is used

because the system considered in this study is isotropic.

Huang and Wellmann (2021) and Börsing et al. (2017) utilize the entropy production to identify instabilities in hydrothermal systems, discern a shift from conductive to convective processes, and indicate variations in geometries and heterogeneity of the permeability. They relate the concept of entropy production with the Rayleigh and Nusselt numbers. Niederau, Wellmann, and Börsing (2019) and Wellmann and Regenauer-Lieb (2012) further implement the concept of entropy production to analyze the impact of heterogeneous permeability and structural uncertainties on the heat transfer process in the Perth Basin, Western Australia. Considering the success of these works, I further use the entropy production as an objective function in global sensitivity analyses to improve my understanding of the transient thermo-hydraulic process in The Hague, Netherlands. This understanding is important to ensure safe and efficient injection-production operations in this region.

6.3 Variations for the global sensitivity analyses

I aim to investigate the influence of various objective functions for the global sensitivity analysis (the formulation is given in Section 2.2.1) to improve the understanding of thermo-hydraulic processes. In Figure 6.1, I illustrate all performed global sensitivity analyses.

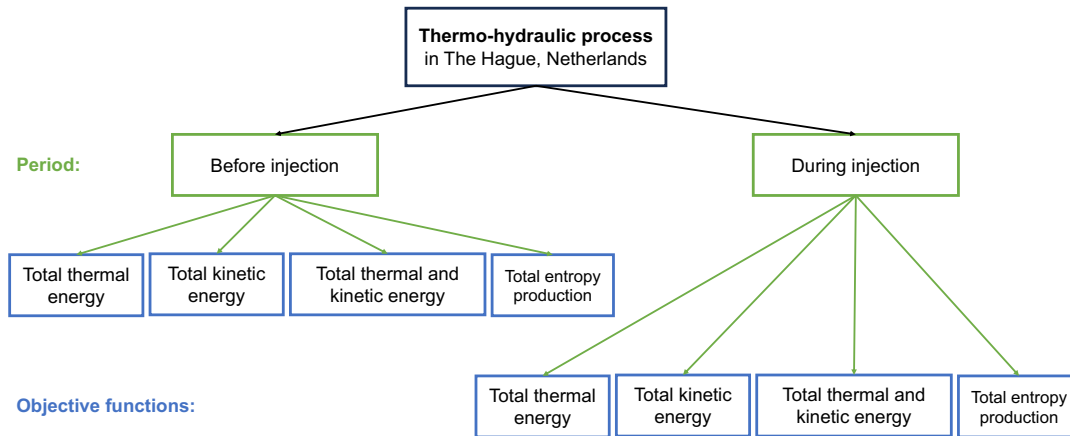


FIGURE 6.1: Analysis phases and objective functions used in the global sensitivity analysis for understanding the thermo-hydraulic process in The Hague, Netherlands.

I split the period of analysis into two phases:

- **Before injection:** in this period, since the The Hague region is conductive, there is no flow (Mottaghy, Pechnig, and Vogt, 2011). The main heat transfer mechanism here is heat conduction.
- **During injection:** in this period, there is a flow due to the injection and production process in the doublet setting.

For each period, I perform the global sensitivity analysis with four different objective functions:

- **Total thermal energy:** The total thermal energy is calculated as the sum of the temperature of the entire model. This objective function produces a sensitivity analysis that illustrates changes in the overall temperature distribution of the reservoir model, treating each region equally.

- **Total kinetic energy:** This objective function is computed as the sum of the pressure of the entire model. It generates a sensitivity analysis that outlines changes in the overall pressure distribution of the reservoir model, with equal consideration given to every region.
- **Total thermal and kinetic energy:** I propose this objective function to capture simultaneous changes in both the pressure and the temperature, and this objective function assigns equal weights to the temperature and pressure responses across the entire reservoir model. It is calculated as the sum of the temperature and pressure of the entire model.
- **Total entropy production:** Inspired by the derivation in Section 6.2, this objective function has a similar purpose as the "total thermal and kinetic energy" objective function. However, for the total entropy production, the gradient of the temperature and pressure is used instead of the temperature and pressure themselves.

I utilize the SALib Python library to perform the global sensitivity analyzes, employing the Saltelli sampling method (Herman and Usher, 2017). To reduce statistical error, 100,000 realizations per parameter are used, resulting in 2,800,000 total realizations for the entire global sensitivity analysis.

6.4 Results

I present the results of my global sensitivity analyzes, leveraging the NI-RB surrogate models to significantly reduce computational costs. To begin, I present the performance metrics of the surrogate models utilized in the global sensitivity analysis, along with an efficient approach for calculating the entropy production. Subsequently, I dive into the results of the global sensitivity analysis, consisting of before injection and during the injection period.

6.4.1 Performances of surrogate models

Before performing any global sensitivity analysis, it is imperative to ensure the accuracy of the surrogate models, which are used to reduce computational cost. In this context, I approximate the pressure, temperature, and entropy production, consecutively, using the NI-RB method. The entropy production is a quantity derived from the pressure and temperature states. The entropy production surrogate model is constructed to facilitate a later comparison of the efficiency between calculating the entropy production function using both the pressure and temperature surrogate models and using the dedicated entropy production surrogate model. The performance metrics for each surrogate model are presented in Table 6.1 and the corresponding architectures are shown in Table 6.2. The offline time is defined as the time needed to construct a surrogate model, including the time for the generation of training and test samples. The online time is defined as the time spent to predict a new case. The training of the surrogate model and the online phase are conducted using a single core on an Apple Macbook Pro with a M1-chip and 8 GB RAM. Based on the performance presented in Table 6.1, I can now confidently confirm that my surrogate models have acceptable accuracy for use in future global sensitivity analysis.

TABLE 6.1: Performance of the surrogate models used in the global sensitivity analysis. The approximation error is calculated using test samples. The values denoted with a red font color represent the total time needed to generate the training and test samples, while the black color represents the time needed to construct a surrogate model. The values in the brackets denote the time needed for a trained surrogate model to predict a new case with 302 time steps and the ones outside the brackets denote the time needed to predict a new case for a single time step.

Surrogate model	Tolerance	Number of basis functions	L_2 approximation error	Offline time [s]	Online time [s]
Temperature	p: 1×10^{-4} t: 1×10^{-4}	80	3.4×10^{-2}	$8.4 \times 10^3 + 6.8 \times 10^6$	5.9×10^{-4} (1.7×10^{-1})
Pressure	p: 1×10^{-4} t: 1×10^{-4}	31	3.4×10^{-2}	$4.6 \times 10^3 + 6.8 \times 10^6$	6.3×10^{-4} (1.9×10^{-1})
Entropy production	p: 1×10^{-4} t: 1×10^{-4}	380	2.9×10^{-3}	$3.4 \times 10^4 + 6.8 \times 10^6$	1.3×10^{-3} (4.0×10^{-1})

Note: p denotes parameter-cut tolerance, t denotes time-trajectory-cut tolerance.

TABLE 6.2: Architectures and hyperparameters of surrogate models used in global sensitivity analysis.

Surrogate model	Neurons in layer 1	Neurons in layer 2	Learning rate	Number of epochs	Batch size
Temperature	52	140	1.5×10^{-4}	46,768	2,330
Pressure	92	119	2.9×10^{-5}	46,459	5,337
Entropy production	48	124	3.8×10^{-5}	41,506	31,613

6.4.2 The calculation of entropy production

The challenge of performing a global sensitivity analysis with entropy production as an objective function is the calculation of the entropy production itself. It requires the calculation of temperature and pressure gradients. Figure 6.2 shows the offline and online time for calculating the entropy production. The offline time is defined as the time needed to generate training and test samples and training surrogate models for the calculation of entropy production. The online time is defined as the time needed to predict a new entropy production solution with 302 time steps. Note that for the online time calculation, I use 302 time steps to align with the finite element method's calculation, which employs 302 time steps.

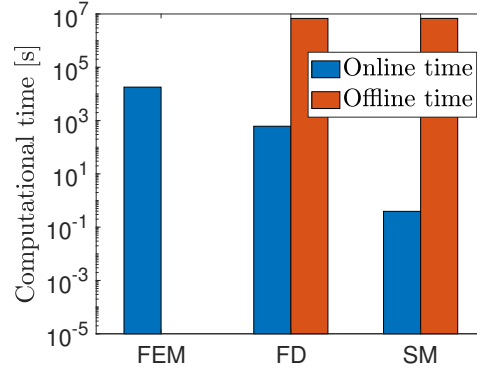


FIGURE 6.2: Offline and online time for the calculation of entropy production. FEM denotes the finite element model, FD the calculation of entropy production with given pressure and temperature surrogate models, SM the calculation of entropy production using the entropy production surrogate model.

If I employ a finite difference scheme to compute entropy production using given pressure and temperature surrogate models, I can predict a new entropy production solution in 10.2 minutes for 302 time steps (2.03 seconds per time step). This scheme necessitates the generation of training and test samples and the training of pressure and temperature surrogate models, totaling 1,893 computing hours. In contrast, utilizing an entropy production surrogate model allows us to predict a new entropy production solution in 0.4 seconds for 302 time steps (1.3 milliseconds per time step). The generation of training and test samples, along with the training of the entropy production surrogate model, takes 1,898 computing hours. Consequently, employing the entropy production surrogate model for entropy production calculations proves to be more efficient, offering faster online time while requiring a similar amount of offline time. The use of an entropy production surrogate results in a speedup of more than four orders of magnitude compared to the finite element model.

6.4.3 Influence of different objective functions before injection period

In Figure 6.3, I show the total sensitivity and first-order sensitivity indices obtained with different objective functions. There are no significant differences between the first-order values and their associated total sensitivity values, indicating minimal correlations. It is essential to note that during this period there is no fluid flow and heat conduction is the governing process that regulates the temperature throughout the reservoir. Since there is no flow before the injection period, I cannot obtain the sensitivity result with the total kinetic energy objective function.

From Figure 6.3a and Figure 6.3b, using the total thermal energy as an objective function, I identify $\lambda_N, \lambda_{CK}, \lambda_{KN}, \lambda_S, Q$ as significant parameters. Specifically, λ_N, λ_{CK} characterize the heat loss from the reservoir, Q describes the heat entering the reservoir, and λ_S controls the heat influx to the reservoir. Looking at the reservoir model in Figure 4.3, the Altona layer (depicted in red in Figure 4.3) and the Jurassic layer (depicted in blue in Figure 4.3) serve as the bottom boundary. Intuitively, the Altona layer should have been significant from the thermal perspective, controlling heat influx to the reservoir. However, since the Altona layer is surrounded by the Jurassic layer, making the Jurassic layer the primary contributor controlling the heat influx to the reservoir. Consequently, only the Jurassic layer emerges as a significant parameter.

The total sensitivity and first-order sensitivity indices depicted in Figure 6.3 yield the same results for both the total thermal energy and the objective thermal and kinetic energy functions. This is because the sensitivity result using the total thermal

and kinetic energy objective function (in this case, I assign equal weights assigned to temperature and pressure responses) is dominated by the temperature response. Hence, I here identify the same significant parameters as those identified using the total thermal energy objective function.

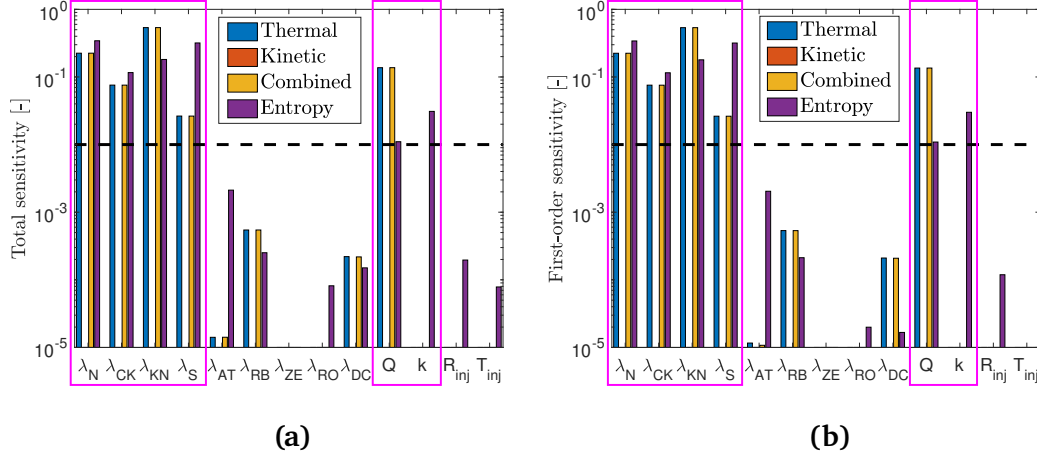


FIGURE 6.3: Global sensitivity analysis results with different objective functions before injection period: (a) total sensitivity index, and (b) first-order sensitivity index. The sensitivity results using the total thermal energy, the total kinetic energy, the total thermal and kinetic energy, and the total entropy production as objective functions are denoted by thermal, kinetic, combined, and entropy consecutively. The pink boxes highlight the significant parameters.

Employing the total entropy production as the objective function, a similar ranking is obtained compared to the total thermal energy, and thermal and kinetic energy objective functions, with the addition of the significant parameter k . Notably, even in the absence of fluid flow before the injection period, a pressure gradient of static water exists within the reservoir. In the entropy production formulation (Equation 6.5), permeability appears explicitly, and given the pressure gradient of static water, permeability becomes a significant factor, impacting the entropy production.

I now compare the use of the total entropy production and thermal energy objective functions with the total sensitivity values. Note that I do not compare the result using the total entropy production and the thermal and kinetic energy objective function since the result using total thermal and kinetic energy objective function is the same as the result using total thermal energy objective function. The total sensitivity value of the heat flow Q with the total entropy production as the objective function decreases, while λ_S increases, compared to the use of the total thermal energy as the objective function. It is attributed to the calculation of T_{bottom} in Equation 6.5, which uses the temperature at the bottom boundary, consisting of the Jurassic and Altena layers, rather than the basement (note that the reservoir spans from the Lower Cretaceous layer to the Altena layer). Despite this change, both the Jurassic layer and the heat flow Q remain significant, as the temperature at the Jurassic and Altena layers is strongly influenced by the heat flow Q .

The total sensitivity value of λ_N , λ_{CK} increases with the use of the total entropy production objective function compared to the use of total thermal energy objective function. This is because the calculation for T_{top} in Equation 6.5 utilizes the temperature at the top boundary, at the intersection between the Upper Cretaceous and Lower Cretaceous layers. This intersection is directly connected to the North Sea layer that controls heat loss. Consequently, the North Sea layer also emerges as a significant parameter.

Looking at Delft sandstone (part of the Lower Cretaceous layer), it has the highest permeability of all layers and its permeability values exhibit a certain range. Consequently, I anticipate the contribution of irreversible thermal and fluid flow friction in the entropy production. This contrasts with other layers, such as the Upper Cretaceous, Jurassic, and Altena layers, where entropy production is primarily influenced by an irreversible thermal contribution because of their low permeability. As the contribution of fluid flow appears, the contribution of thermal energy decreases, resulting in a lower sensitivity value of λ_{KN} than that derived from the objective function of total thermal energy, and the importance of permeability k .

In summary, employing total thermal energy, thermal and kinetic energy, and entropy production as objective functions in global sensitivity analyzes captures significant parameters contributing to the conductive thermal process (before the injection period). However, the use of total kinetic energy as an objective function does not identify these significant parameters, as it is designed specifically to capture fluid flow.

6.4.4 Influence of different objective functions during injection period

I observe no change in the sensitivity results over time for all objective functions, see Figure 6.4. This implies that there is no significant change in the physical processes captured by all objective functions over time. Furthermore, there are no significant correlations among the parameters, as evidenced by the absence of significant differences between first-order sensitivity values and their associated total sensitivity values (see Figure 6.4). Moreover, the global sensitivity analyzes conducted with the total thermal energy, as well as the total thermal and kinetic energy objective functions during the injection period, yield the same sensitivity results to those obtained before the injection period. This indicates that the system undergoes the same physical processes both before and during the injection period.

When utilizing the total kinetic energy objective function, I identify the same significant parameters as those identified using the total thermal energy objective function, with the addition of significant permeability k . Note that the calculation of pressure requires fluid density, which is temperature-dependent. Consequently, λ_N , λ_{CK} , λ_{KN} , λ_S , and Q emerge as significant in this context. However, since these parameters do not directly influence the pressure, both the total sensitivity and first-order values are lower than those obtained with the total thermal energy objective function. In Darcy's law, permeability appears explicitly in the pressure calculation, which makes the significance of permeability k the highest among the identified parameters.

When employing the total entropy production objective function, I observe the same significant parameters as those identified using the total kinetic energy objective function. However, their total and first-order sensitivity values differ. This difference arises because entropy production considers contributions from both irreversible thermal and fluid flow friction, whereas the total kinetic energy solely accounts for fluid flow. This distinction is evident in the sensitivity values of λ_{KN} and k . During this period, fluid flow and heat transfer occur at the same time. The total thermal energy, designed specifically to capture thermal processes, produces a higher λ_{KN} value than when using the total entropy production, with permeability k not considered significant. Conversely, the total kinetic energy, tailored for capturing fluid flow components, identifies permeability k as the most significant parameter. Since entropy production encompasses both thermal and fluid flow components, it identifies λ_{KN} and k as significant parameters, with a λ_{KN} value smaller than that obtained from the use of total thermal energy and a permeability k value smaller than that from using total kinetic energy.

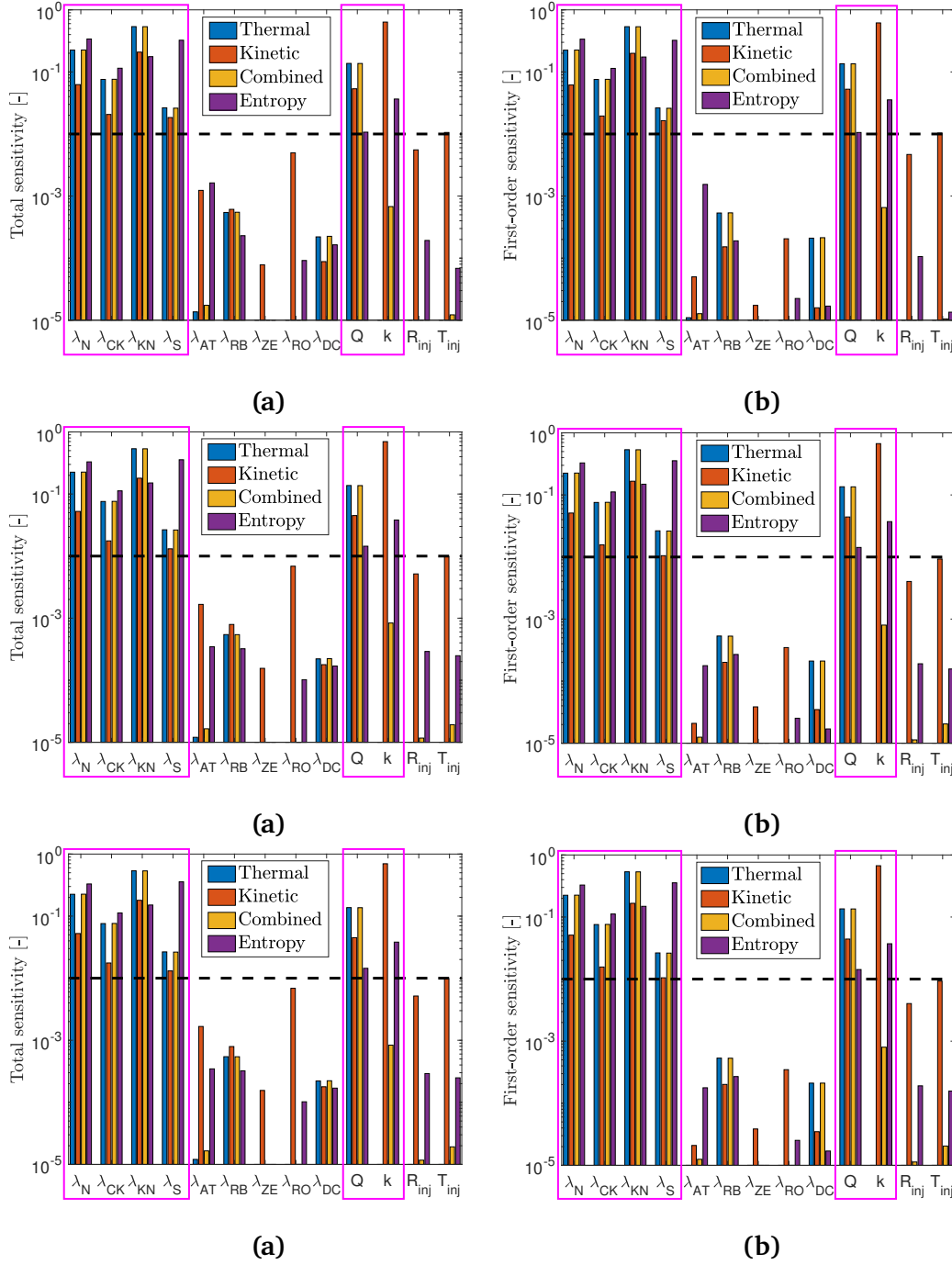


FIGURE 6.4: Global sensitivity analysis results with different objective functions during injection period: (a) total sensitivity index at $t = 100$ months, (b) first-order sensitivity index at $t = 100$ months, (c) total sensitivity index at $t = 300$ months, (d) first-order sensitivity index at $t = 300$ months, (e) total sensitivity index at $t = 600$ months, and (f) first-order sensitivity index at $t = 600$ months. The sensitivity results using the total thermal energy, the total kinetic energy, the total thermal and kinetic energy, and the total entropy production as objective functions are denoted by thermal, kinetic, combined, and entropy consecutively. The pink boxes highlight the significant parameters.

I further employ Rayleigh and Peclet numbers to determine the governing heat

transfer process in the Delft Sandstone. The Rayleigh number is mathematically expressed as follows: (Turcotte and Schubert, 2002)

$$Ra = \frac{\beta_f g \rho_f^2 c_f \mathbf{k} H (T_{\text{bottom}} - T_{\text{top}})}{\eta_f (\lambda_s^{1-\phi}(T) + \lambda_f^\phi)}, \quad (6.6)$$

where β_f is the thermal expansion of the fluid and H is the thickness of the layer (in my case the Delft Sandstone). The Peclet number is mathematically expressed below: (Turcotte and Schubert, 2002)

$$Pe = \frac{\rho_f c_f \mathbf{u}_f l_c}{(\lambda_s^{1-\phi}(T) + \lambda_f^\phi)}, \quad (6.7)$$

where l_c is the characteristic length. I perform uniform sampling on all parameters in Table 5.2 and create 1,000 realization. Figure 6.5 illustrates the entropy production as a function of the Rayleigh and Peclet numbers. Subsequently, Figure 6.5b reveals that no realization triggers convection, as all values fall below the critical Rayleigh number (Börsing et al., 2017; Bejan, 2013; Turcotte and Schubert, 2002). Meanwhile, Figure 6.5c indicates high Peclet numbers for all realizations, indicating the dominance of the heat advection process (Turcotte and Schubert, 2002). Consequently, I conclude that forced convection governs the heat transfer process (Turcotte and Schubert, 2002). This analysis further confirms the importance of permeability k and λ_{KN} as important factors in a forced convection process in Delft Sandstone.

Additionally, with the use of non-intrusive reduced-basis (NI-RB) surrogate models, I can conduct the global sensitivity analysis for one time step in approximately 10 minutes.

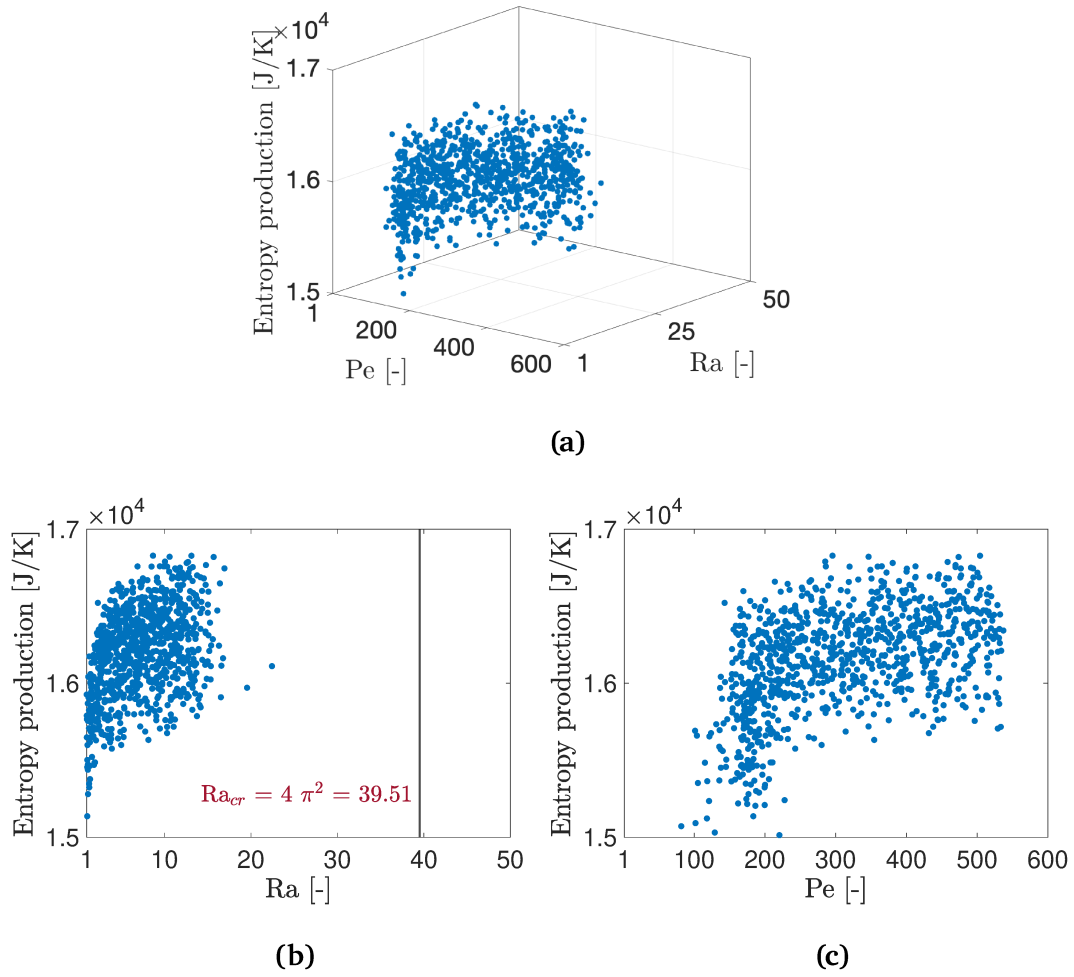


FIGURE 6.5: Analysis on governing heat transfer process using Rayleigh and Peclet number: (a) entropy production as a function of Rayleigh and Peclet number, (b) a 2-D view of (a) showing entropy production as a function of only Rayleigh number, and (c) a 2-D view of (a) showing entropy production as a function of only Peclet number. Ra_{cr} denotes the critical Rayleigh number.

6.5 Discussions

From the global sensitivity analysis in Figure 6.3 and Figure 6.4, supported by the Rayleigh-Peclet analysis in Figure 6.5, I can conclude that entropy production provides a comprehensive insight into the transient thermo-hydraulic process, since it describes the state of a system based on the thermodynamics law. It combines the information from the pressure and the temperature state in a thermodynamical way, not just simply adding the pressure and temperature state, as is done in the combined objective function. Hence, the use of the entropy production as an objective function in global sensitivity analysis provides more information. The sensitivity results obtained with the total entropy production objective function fall between those of the total thermal energy and the kinetic energy objectives. With the total thermal energy objective function, I can only identify λ_N , λ_{CK} , λ_{KN} , λ_S , and Q as significant parameters, strongly related to the heat conduction process and losing the fluid flow component. With the total kinetic energy objective function, I additionally identify k as a significant parameter. However, with this function, I cannot obtain any result before the injection period,

since there is no flow. This objective function can only highlight the contribution of the fluid flow. However, the target function for the total entropy production captures both the heat conduction and the fluid flow contributions simultaneously. Therefore, the use of entropy production as an objective function in global sensitivity analyses is suitable for capturing the transient thermo-hydraulic process.

The primary challenge in employing the entropy production as an objective function for global sensitivity analyses lies in the calculation of the gradients. As depicted in Equation 6.5, the entropy production requires the calculation of the pressure and temperature gradients. If I calculate the entropy production using the pressure and the temperature solutions directly, the global sensitivity analysis becomes computationally intensive. This is because, in each realization, I must employ a finite-difference scheme on the spatio-temporal pressure and temperature solutions, leading to a long computational time. Alternatively, I could create a surrogate model using pre-calculated entropy production values. In this approach, entropy production is computed during the generation of the training samples. It is worth noting that the entropy production solution can exhibit significant spatial and temporal variations spanning almost four orders of magnitude, necessitating a log transformation. However, achieving a substantial reduction with the Proper Orthogonal Decomposition (POD) becomes challenging with this log transformation, resulting in an increased offline time. In my case, utilizing an entropy production surrogate model proves to be a more efficient strategy for global sensitivity analyses, especially considering the need for a considerable number of realizations to avoid statistical errors.

The Delft Sandstone experiences a forced convection process, indicated by the significant contribution of λ_{KN} and k from the thermodynamic perspective through global sensitivity analysis and Rayleigh-Peclet analysis. It undergoes the same physical process (involving contributions from both irreversible thermal and fluid flow friction components) until 600 months, as indicated by the constant sensitivity index values across all time steps. If injection-production operations were to be conducted in this layer, I can anticipate no significant changes in the reservoir due to these operations.

Chapter 7

The non-intrusive reduced basis method for reactive transport simulations

*"If a machine is expected to be infallible,
it cannot also be intelligent."*

— Alan TURING (1947)

In the context of the efficiency and safety of heat extraction processes from geothermal reservoirs, an understanding of chemical transport and reactions is crucial. For example, scale build-up around the wellbore can result in injectivity problems, leading to an unsafe and inefficient injection process. This understanding requires numerical simulation, specifically through reactive transport simulation. In this chapter, I employ the non-intrusive reduced-basis (NI-RB) method, for the first time, to accelerate reactive transport simulation, enabling large-scale parameter-estimation and optimization-under-uncertainties studies with it. This chapter is covered in Santoso et al. (2023b) and Santoso et al. (in preparation, c).

7.1 Reactive transport problems

Reactive transport problems solve multicomponent chemical reactions, multispecies chemical transport, and fluid flow (Steefel, DePaolo, and Lichtner, 2005; Steefel and Lasaga, 1994). There are multiple formulations for solving reactive transport problems where the coupling between the transport of chemical species and their chemical reactions is established either through mass action equations (Steefel, DePaolo, and Lichtner, 2005; Saaltink, Ayora, and Carrera, 1998; Steefel and Lasaga, 1994; Rubin, 1990; Reeves and Kirkner, 1988; Kirkner and Reeves, 1988) or through Gibbs energy minimization (GEM) (Leal et al., 2017; Leal, Kulik, and Kosakowski, 2016). I follow the GEM-based formulation from Leal, Kulik, and Kosakowski (2016), in which the transport equations are formulated in terms of the mass conservation of elements (that is, chemical elements and electric charge): (Kyas et al., 2022; Leal et al., 2020; Leal, Kulik, and Kosakowski, 2016).

$$\frac{\partial(b_i^f + b_i^s)}{\partial t} + \nabla \cdot \left(\frac{\mathbf{u}}{\phi} b_i^f - D \nabla b_i^f \right) = 0, \quad i = 1, \dots, N_e \quad (7.1)$$

where b is the bulk concentration [mol / m³], t is the time [s], $\mathbf{u}$ is the Darcy velocity [m / s], ϕ is porosity [-], D is the diffusion coefficient [m²/s], and N_e is number of elements. The subscript i is the enumeration of chemical elements, the superscript f

denotes the fluid, and the superscript s denotes the solid contribution. For chemical species, I assume a local chemical equilibrium between fluid species and solid minerals (Kyas et al., 2022). The chemical equilibrium is formulated through a Gibbs energy minimization calculation: (Kyas et al., 2022; Leal et al., 2020; Leal, Kulik, and Kosakowski, 2016)

$$\min_n G(T, P, n) \text{ subject to } \begin{cases} An = b \\ n \geq 0 \end{cases}, \quad (7.2)$$

$$G(T, P, n) = \sum_i^{N_s} (\mu_i^\circ + RT \ln(a_i(T, P, n))) n_i, \quad (7.3)$$

where $G(T, P, n)$ is the Gibbs energy [J/mol] of the chemical system, T is temperature [K], P is pressure [Pa], N_s is the number of chemical species, n_i is the molar amount of chemical species with index i , A is the stoichiometric coefficient in matrix form, μ° is the standard chemical potential [J/mol], R is the universal gas constant [J/(mol · K)], and a_i is the activity of the chemical species with index i . Equations 7.1 - 7.3 are referred to as the *physical-chemistry-based model* for the remainder of the Chapter. The physical-chemistry-based model solves for spatio-temporal physical and chemical states $\mathbf{z} = [b_1^f(\mathbf{x}, t), \dots, b_{N_e}^f(\mathbf{x}, t), b_1^s(\mathbf{x}, t), \dots, b_{N_e}^s(\mathbf{x}, t)]$. In my study, I use the Finite Volume (FV) method to solve the reactive transport problem and Reaktoro, an open-source C++/Python framework, to perform chemical equilibrium calculations (Leal, 2015).

There are two main challenges in solving Equations 7.1 - 7.3: 1) solving the chemical equilibrium problem (Equation 7.2 - 7.3) requires an optimization scheme that is computationally expensive if numerous chemical species are involved and it is performed for every grid cell and every time step (Jatnieks et al., 2016) and 2) small time steps and fine meshes are required in Solid Earth applications to capture complex geometries and achieve numerical stability, leading to a high-dimensional problem (Degen, Cacace, and Wellmann, 2022). Hence, High-Performance Computing (HPC) infrastructures are often needed to solve these equations (Sprocati and Rolle, 2021; Jatnieks et al., 2016).

Given the computational cost of performing reactive transport simulations, conducting large-scale parameter-estimation or optimization-under-uncertainties studies poses a considerable challenge. Consequently, I rely on the surrogate model, building on the concepts discussed in previous Chapters. For the surrogate modeling method, I compare the data-driven and physics-based machine learning methods. The data-driven machine learning method is the Neural Network, while the physics-based machine learning method is the non-intrusive reduced-basis (NI-RB) method. The surrogate model provides a map from the physical and chemical parameters ξ to the physical and chemical states.

7.2 Case study

I use 1-D reactive transport of the dolomitization process to conduct the comparison. It is a benchmark problem used in Kyas et al. (2022) and Leal et al. (2020) inspired by the benchmark proposed in Beyer, Nagel, and Shao (2014). The problem solves Equations 7.1 - 7.3 with the chemical systems in Table 7.1.

TABLE 7.1: Chemical systems for the dolomitization process.

Elements:	C, Ca, Cl, H, Mg, Na, O, Si, Z*			
Phases:	Aqueous, Calcite, Dolomite, Quartz			
Species:	CO _{2(aq)}	Cl ⁻ _(aq)	H ₂ O _{2(aq)}	MgCO _{3(aq)}
	CO ₃ ²⁻ _(aq)	ClO ⁻ _(aq)	HCO ₃ ⁻ _(aq)	MgCl ⁺ _(aq)
	Ca(HCO ₃) ⁺ _(aq)	ClO ₂ ⁻ _(aq)	HCl _(aq)	MgOH ⁺ _(aq)
	Ca ₂ ⁺ _(aq)	ClO ₃ ⁻ _(aq)	HClO _(aq)	Na ⁺ _(aq)
	CaCO _{3(aq)}	ClO ₄ ⁻ _(aq)	HClO _{2(aq)}	NaCl _(aq)
	CaCl ⁺ _(aq)	H ⁺ _(aq)	HO ₂ ⁻ _(aq)	NaOH _(aq)
	CaCl _{2(aq)}	H _{2(aq)}	Mg(HCO ₃) ⁺ _(aq)	O _{2(aq)}
	CaOH ⁺ _(aq)	H ₂ O _(l)	Mg ₂ ⁺ _(aq)	OH ⁻ _(aq)
	Calcite	Dolomite	Quartz	

Z* represents an electric charge

The computational domain is illustrated in Figure 7.1, and is discretized into 100 cells ($N_x = 100$).

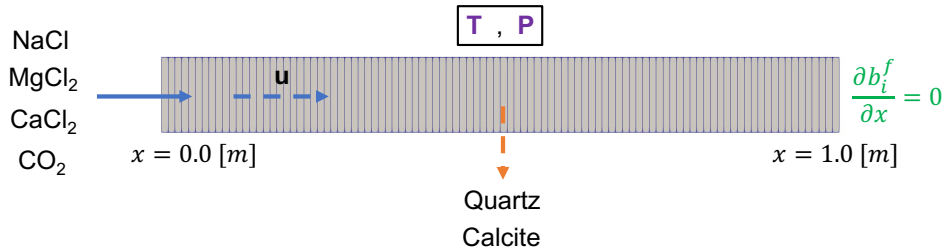


FIGURE 7.1: The computational domain of the dolomitization process.

I consider a porous medium composed of quartz and calcite, which is initially saturated with NaCl. A brine containing NaCl, MgCl₂, CaCl₂, and CO₂ is injected at $x = 0.0$ [m] with a velocity $\mathbf{u}$. At $x = L_x = 1.0$ [m], the composition of the output fluid is observed; therefore, a no-flow boundary condition is applied.

I further define the dimensionless time t_D and dimensionless distance x_D as follows:

$$t_D = \frac{\mathbf{u} \cdot t}{L_x} \quad (7.4)$$

$$x_D = \frac{x}{L_x} \quad (7.5)$$

where L_x is the maximum distance [m]. Time spans $[0.0, 2.0]$ are discretized into 100 time steps ($N_t = 100$). This results in 290,000 degrees of freedom (with 29 unknown variables, including the Lagrange multipliers associated with the Gibbs energy minimization calculations).

I select the following nine parameters that are of interest for benchmarking reactive transport simulators (Kyas et al., 2022; Jatnieks et al., 2016; Beyer, Nagel, and Shao, 2014):

- $b_{\text{NaCl},i}$: initial NaCl concentration within the domain
- $b_{\text{NaCl},b}$: injected NaCl concentration at $x = 0.0$ [m]
- $b_{\text{MgCl}_2,b}$: injected MgCl_2 concentration at $x = 0.0$ [m]
- $b_{\text{CaCl}_2,b}$: injected CaCl_2 concentration at $x = 0.0$ [m]
- $b_{\text{CO}_2,b}$: injected CO_2 concentration at $x = 0.0$ [m]
- ϕ : porosity of the medium
- V_{quartz} : volumetric quartz composition of the medium
- T : temperature
- $\mathbf{u}$: injection Darcy velocity

Their value ranges are shown in Table 7.2. Note that I also pick two randomly chosen blind cases and ensure that they are neither in the training nor the test set. Blind cases 1 and 2 are selected to test the performance of the surrogate model in predicting a moderate and an extreme case (extremely high salinity as an initial condition) consecutively.

TABLE 7.2: Physical and chemical parameter ranges for the case study. Blind case 1 and 2 are used for testing the performance of the surrogate models.

Input parameters	Lower bound	Upper bound	Blind case 1	Blind case 2
$b_{\text{NaCl},i}$ [molal]	0.050	5.000	1.224	4.000
$b_{\text{NaCl},b}$ [molal]	0.050	1.200	0.588	0.625
$b_{\text{MgCl}_2,b}$ [molal]	0.010	0.100	0.057	0.055
$b_{\text{CaCl}_2,b}$ [molal]	0.010	0.050	0.043	0.030
$b_{\text{CO}_2,b}$ [molal]	0.050	1.500	0.926	0.775
ϕ [-]	0.100	0.250	0.128	0.175
V_{quartz} [-]	0.950	0.980	0.969	0.965
T [°C]	50.00	90.00	53.29	70.00
$\mathbf{u}$ [m/d]	0.50	2.00	1.24	1.25

With the given ranges, I could identify "shock" features that reflect a nonlinearity in the physical and chemical states of the dolomitization problem. The "shock" features exist because of a combined effect between the transport of chemical species to initiate the reactions, and the dissolution and precipitation of calcite and dolomite (Kyas et al., 2022; Leal et al., 2020). The interest is on the spatio-temporal solution of the calcite concentration, the dolomite concentration, the pH, and the aqueous carbon dioxide concentration (Kyas et al., 2022; Leal et al., 2020).

7.3 Results

In this section, I present a comparison between the data-driven and the physics-based machine learning methods in terms of accuracy, efficiency, and challenges during the training and prediction phase. The comparison utilizes two blind cases, blind case 1 and blind case 2 as shown in Table 7.2. I create three training sets, containing 50, 100, and 500 samples. Each realization contains 100 time steps. In this study, the number of samples in a test set is the same as in its associated training set. Both training set generation and machine learning training are performed on an Apple MacBook Pro with 8 GB RAM and an M1 chip.

Figure 7.2 shows the evolution of the number of basis functions with different numbers of training samples for different responses of interest. The number of basis functions for all responses is relatively stable for training set sizes exceeding 10000 samples. This proves that 10000 training samples are sufficient to reach convergence. The stabilized number of basis functions is different for each response, depending on the desired accuracy. For example, I require a larger number of basis functions for the calcite response than for the CO₂ response, since I need to capture the "shock" features (Figure 7.3) of the calcite response. The user-defined accuracy for each response is presented in Table 7.3

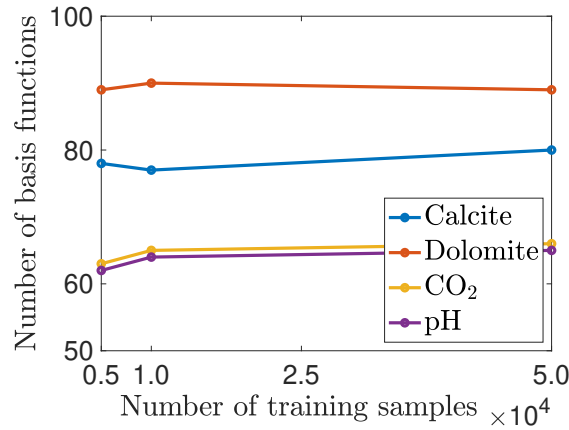


FIGURE 7.2: The evolution of the number of basis functions in NI-RB surrogate models with different numbers of training samples.

TABLE 7.3: User-defined accuracy for each response for constructing basis functions.

Input parameters	Time-trajectory tolerance	Parameter tolerance
Calcite	5×10^{-6}	1×10^{-7}
Dolomite	5×10^{-5}	1×10^{-6}
CO ₂	1×10^{-5}	1×10^{-6}
pH	1×10^{-3}	1×10^{-3}

Figures 7.3, 7.4, 7.5, and 7.6 compare the performance of the data-driven Neural Network and the NI-RB method in capturing the responses of calcite, dolomite, CO₂, and pH, respectively, for different numbers of training samples under blind case 1.

For both the calcite (Figure 7.3) and dolomite (Figure 7.4) responses, I only show the responses until $x_D = 0.50$ because after $x_D = 0.50$, the responses are just constant lines on top of each other for different time steps and they are perfectly predicted by both surrogate models. In general, we can observe that, by increasing the number of training samples, the predictions of both surrogate models are becoming more accurate. Five thousand training samples are insufficient for either surrogate model to capture the responses, especially the ones with "shock" features. However, with this number of training samples, the NI-RB surrogate model can already capture the shape of the responses and has less deviations compared to the data-driven NN surrogate model. The NI-RB surrogate model can then accurately capture all responses with a minimum amount of 10000 training samples. On the other hand, data-driven NN surrogate models require more training samples (> 10000 training samples) to capture the responses. Even with 50000 training samples, data-driven NN surrogate models cannot capture the "shock" features of the calcite and the dolomite responses, whereas the NI-RB surrogate models can capture "shock" features with only 10000 training samples.

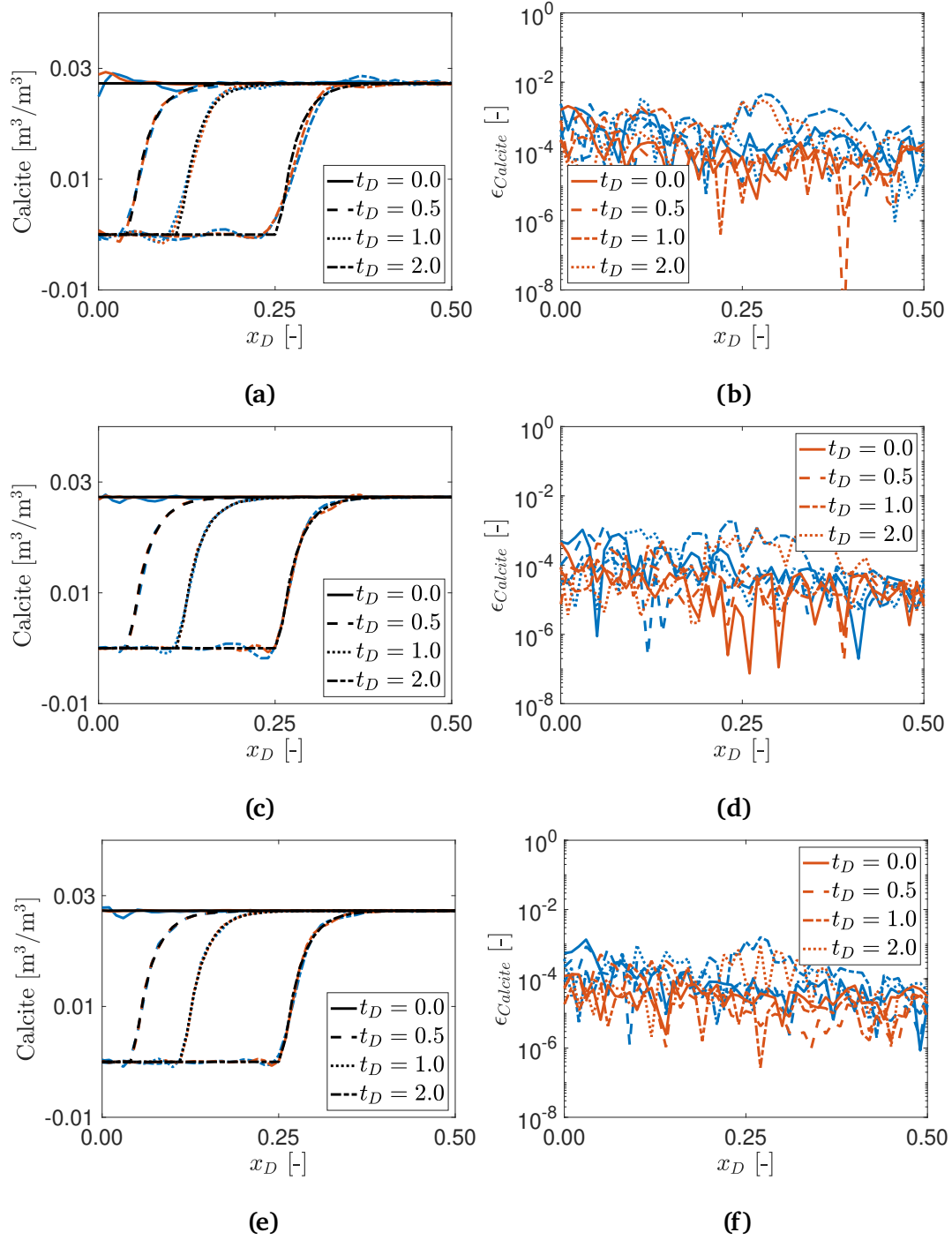


FIGURE 7.3: Comparison between the data-driven NN (blue color) and the NI-RB (orange color) surrogate model in predicting the calcite response with different numbers of training samples, utilizing blind case 1. (a) and (b) show a comparison in terms of calcite response and relative error consecutively for 5000 training samples. (c) and (d) show results for 10000 training samples, (e) and (f) for 50000 training samples.

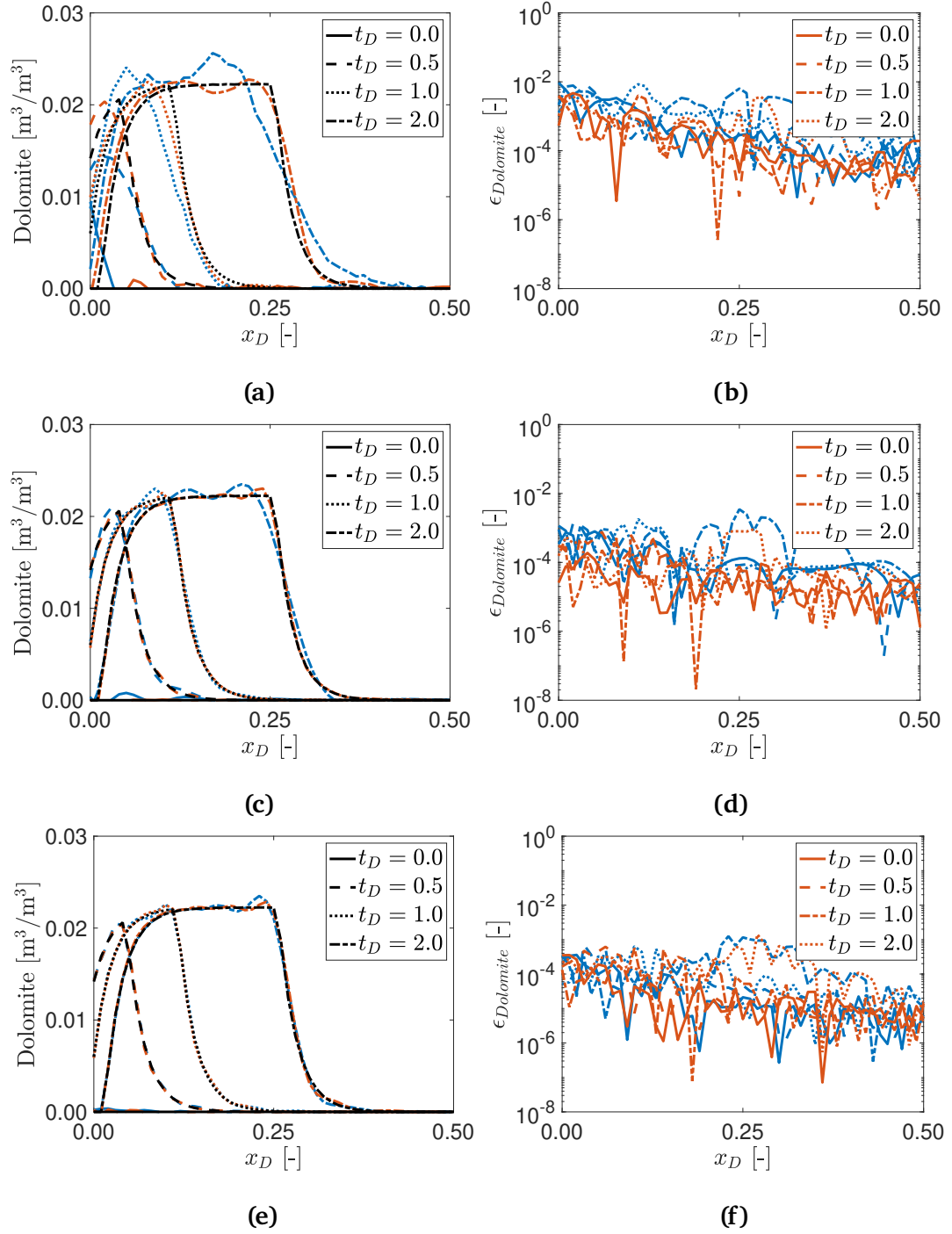


FIGURE 7.4: Comparison between data-driven NN (blue color) and NI-RB (orange color) surrogate model in predicting the dolomite response with different numbers of training samples, utilizing blind case 1. (a) and (b) show the comparison in terms of dolomite response and relative error consecutively for 5000 training samples. (c) and (d) show results for 10000 training samples, Figure (e) and (f) for 50000 training samples.

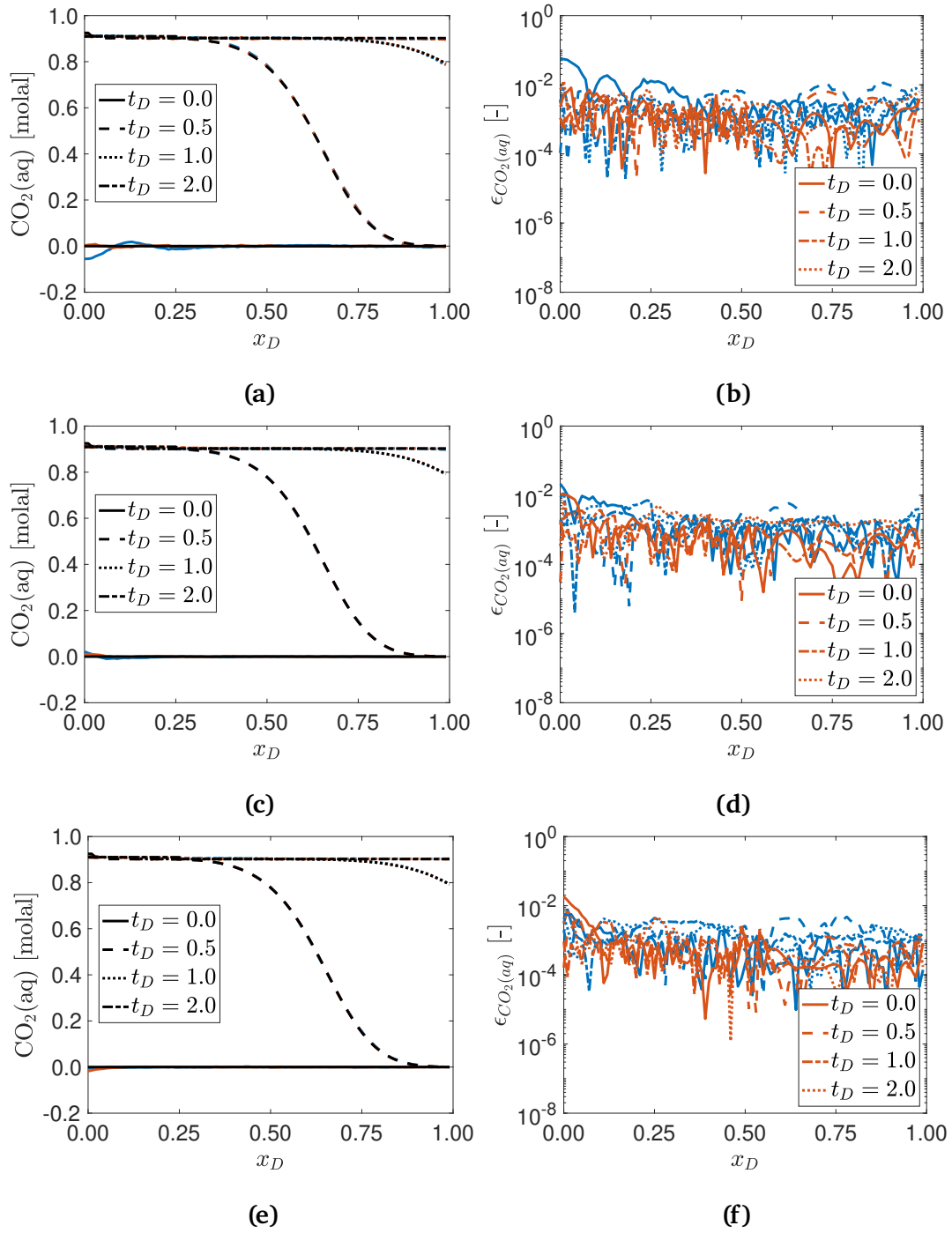


FIGURE 7.5: Comparison between data-driven NN (blue color) and NI-RB (orange color) surrogate model in predicting the $\text{CO}_2(\text{aq})$ response with different numbers of training samples, utilizing blind case 1. (a) and (b) show the comparison in terms of $\text{CO}_2(\text{aq})$ response and relative error consecutively for 5000 training samples. (c) and (d) show results for 10000 training samples, (e) and (f) for 50000 training samples.

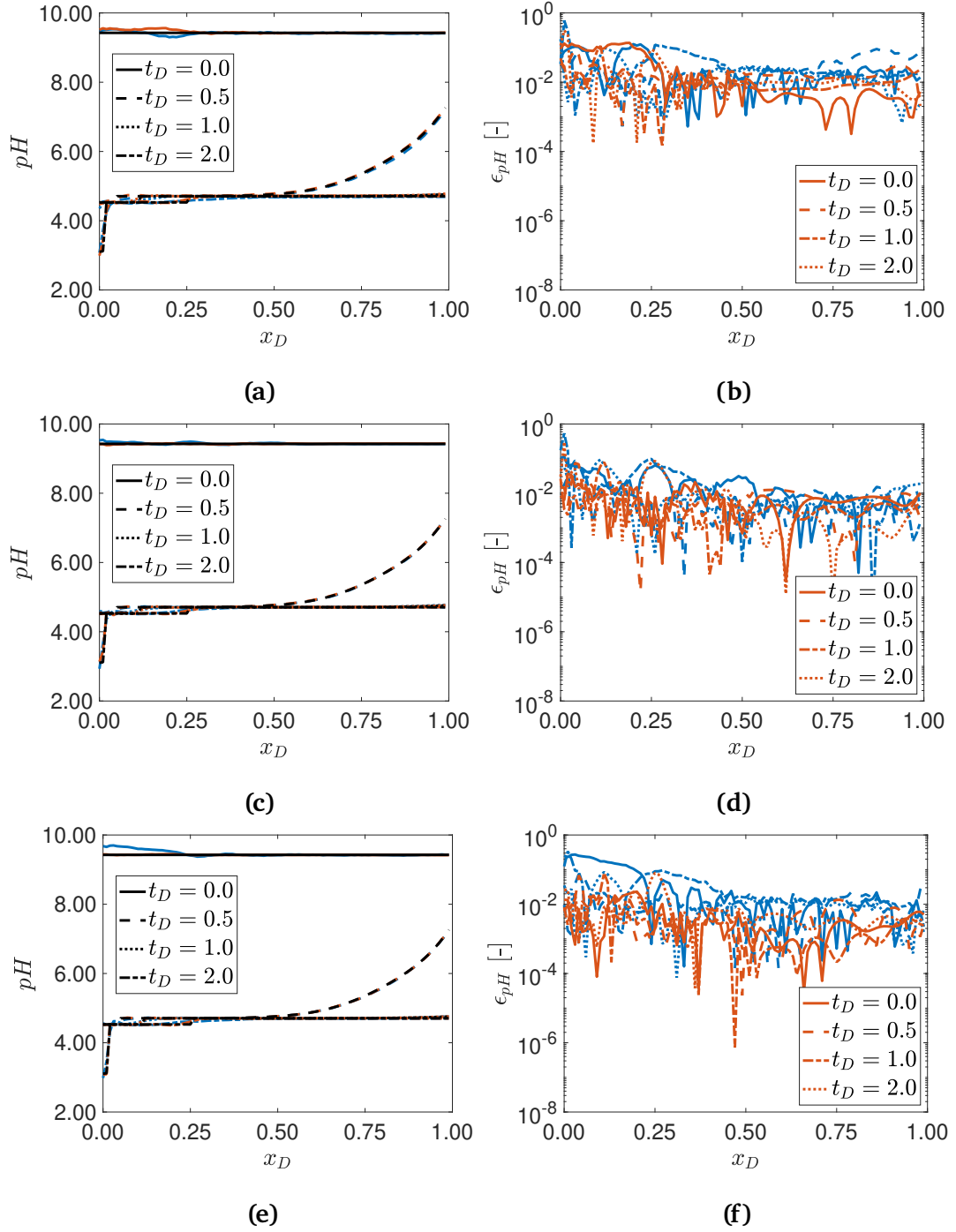


FIGURE 7.6: Comparison between data-driven NN (blue color) and NI-RB (orange color) surrogate model in predicting the pH response with different terms numbers of training samples, utilizing blind case 1. (a) and (b) show the comparison in terms of pH response and relative error consecutively for 5000 training samples. (c) and (d) show results for 10000 training samples, (e) and (f) for 50000 training samples.

I further demonstrate the higher accuracy achieved by using the NI-RB surrogate model instead of the data-driven NN surrogate model with an extreme case (blind case 2). This blind case is illustrated in Figures 7.7 to 7.10. For both the calcite (Figure 7.7) and dolomite (Figure 7.8) responses, I again only show the responses until $x_D = 0.50$ for the reasons mentioned above. The NI-RB surrogate model in blind case 2 has the same behavior as in blind case 1. The data-driven NN surrogate model in blind case

2 also shows the same behavior as in blind case 1. The NI-RB surrogate model can capture the responses with only 10000 training samples, while the data-driven NN surrogate model cannot capture the responses even with 50000 training samples.

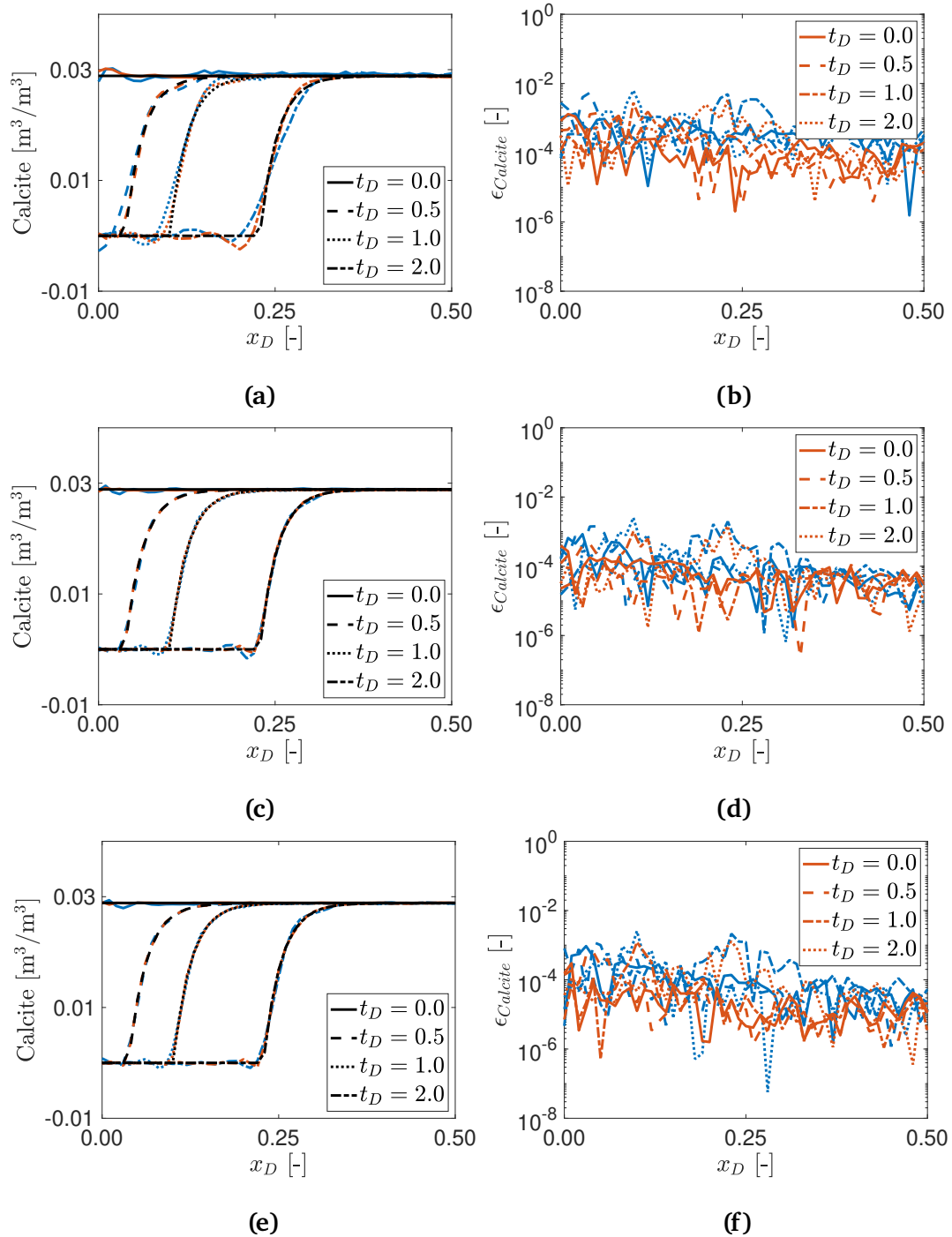


FIGURE 7.7: Comparison between data-driven NN (blue color) and NI-RB (orange color) surrogate model in predicting the calcite response with different numbers of training samples, utilizing blind case 2. (a) and (b) show the comparison in terms of calcite response and relative error consecutively for 5000 training samples. (c) and (d) show results for 10000 training samples, (e) and (f) for 50000 training samples.

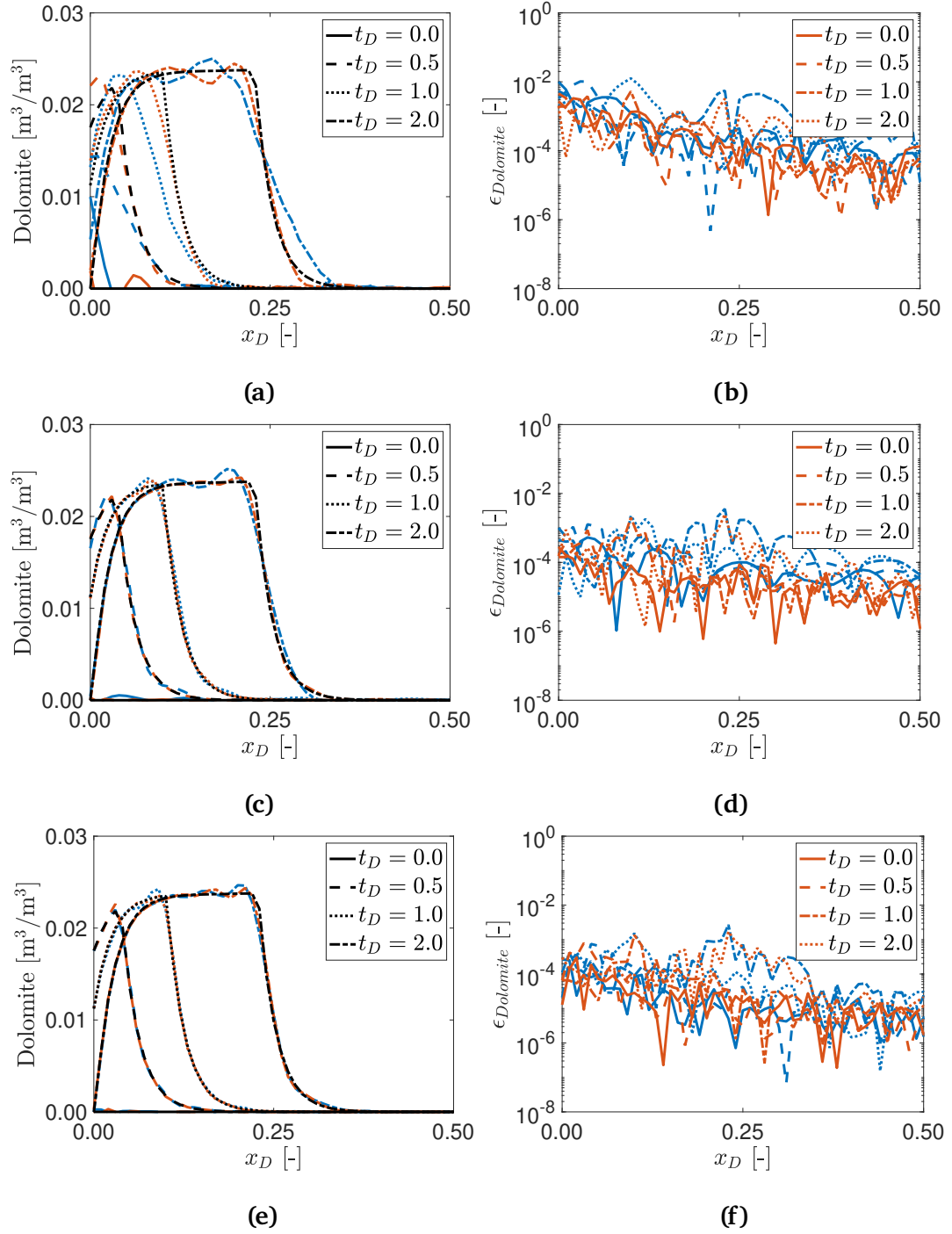


FIGURE 7.8: Comparison between data-driven NN (blue color) and NI-RB (orange color) surrogate model in predicting the dolomite response with different numbers of training samples, utilizing blind case 2. (a) and (b) show the comparison in terms of dolomite response and relative error consecutively for 5000 training samples. (c) and (d) show results for 10000 training samples, (e) and (f) for 50000 training samples.

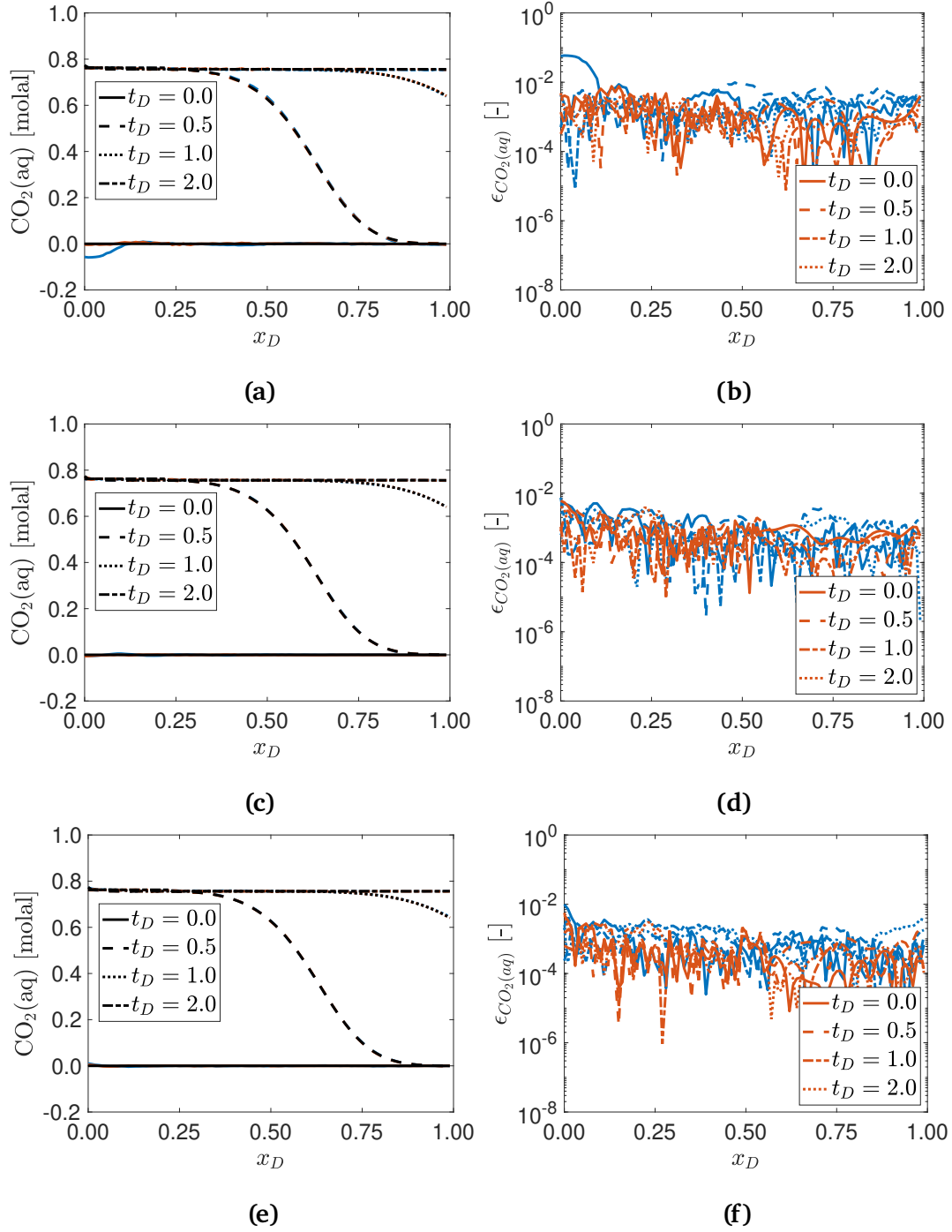


FIGURE 7.9: Comparison between data-driven NN (blue color) and NI-RB (orange color) surrogate model in predicting the $\text{CO}_2(\text{aq})$ response with different numbers of training samples, utilizing blind case 2. (a) and (b) show the comparison in terms of $\text{CO}_2(\text{aq})$ response and relative error consecutively for 5000 training samples. (c) and (d) show results for 10000 training samples, (e) and (f) for 50000 training samples.

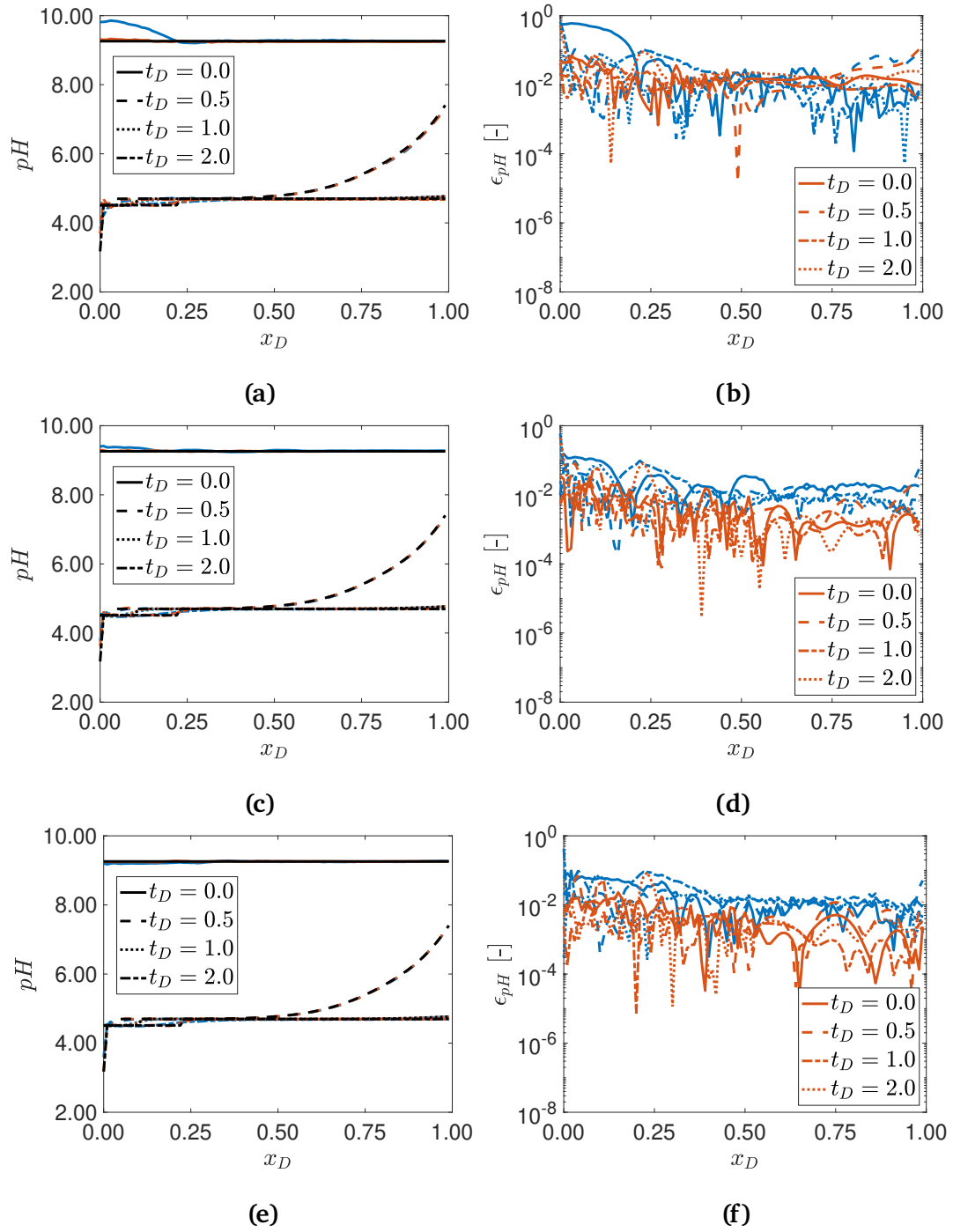


FIGURE 7.10: Comparison between data-driven NN (blue color) and NI-RB (orange color) surrogate model in predicting the pH response with different numbers of training samples, utilizing blind case 2. (a) and (b) show the comparison in terms of pH response and relative error consecutively for 5000 training samples. (c) and (d) show results for 10000 training samples, (e) and (f) for 50000 training samples. The black dashed-line in the relative error plot denotes the desired accuracy.

Figure 7.11 shows the offline and online time for both the data-driven NN and NI-RB surrogate models. The offline time shows the time needed to train a surrogate model with optimum hyperparameters, without considering the time to generate the training and the test set. The online time shows the time needed for a trained surrogate model to predict a new case. The offline time for both data-driven NN and

NI-RB surrogate models are similar. The online time for the NI-RB surrogate model is slightly shorter than for the data-driven NN surrogate model. This is because the NI-RB surrogate model has a simpler architecture than the data-driven NN model. The data-driven NN surrogate model requires a slightly larger number of neurons, associated with more weights, than the NI-RB surrogate model to capture the nonlinear response. The hyperparameters for the data-driven NN and the NI-RB surrogate model are shown in Table 7.4.

TABLE 7.4: The Optimum hyperparameters for both the data-driven NN and the NI-RB surrogate model used for calculating offline and online time.

Hyperparameters	Data-driven NN	NI-RB
Number of neurons in hidden layer 1 N_{n,l_1}	150	52
Number of neurons in hidden layer 2 N_{n,l_2}	60	140
Number of epochs N_e	4567	46768
Learning rate l_r	5.7×10^{-4}	1.5×10^{-4}
Batch size N_b	5820	1299

Both the data-driven NN and the NI-RB surrogate models give nearly 3 orders of magnitude speed-up in predicting a new case, compared to the use of the FV model.

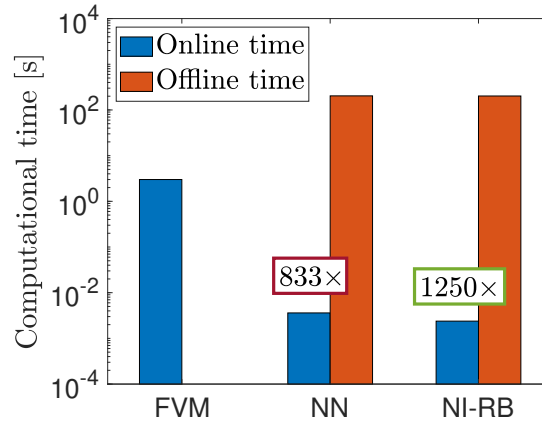


FIGURE 7.11: The offline and the online time for both the data-driven NN and the NI-RB surrogate models.

7.4 Discussion

Throughout the comparison, the NI-RB method can deliver accurate predictions with a smaller number of training samples than the data-driven method. Even with a large number of training samples, a data-driven NN model could not capture the "shock" features. This is because NI-RB surrogate models preserve the structure of the physical-chemistry-based model (containing these "shock" features) in the basis functions (Willcox, Ghattas, and Heimbach, 2021).

The offline time for both the data-driven NN and the NI-RB surrogate models are similar. The offline time is mainly determined by the optimum selection of hyperparameters from the Bayesian Optimization with Hyperband (BOHB) algorithm.

Comparing the data-driven and the NI-RB method for the context of scalable and reliable uncertainty quantification in Solid Earth applications, involving chemistry, the most suitable method is the NI-RB method. The NI-RB method has several advantages:

- It produces physically and chemically consistent predictions (Bauer et al., 2021),
- It is efficient, such that it does not require a large number of training samples for its construction, considering the amount of resources and time needed to run high-dimensional Solid Earth models (Degen et al., 2023b; Bauer et al., 2021; Santoso et al., 2022),
- It is fast in predicting new cases, considering numerous runs needed for uncertainty quantification (Degen et al., 2023b; Willcox, Ghattas, and Heimbach, 2021; Bauer et al., 2021),

For an extremely large number of physical and chemical parameters, such as in the cement system (Laloy and Jacques, 2022), the NI-RB method will suffer the curse of dimensionality, which will require a large number of training samples. Hence, the use of the NI-RB method will be inefficient. A possible solution is to reduce the dimensionality of physical and chemical parameters using the active subspaces or the Karhunen-Loeve method (Wu et al., 2023; O’Leary-Roseberry et al., 2022a). See Wu et al. (2023) and O’Leary-Roseberry et al. (2022a) for detailed implementations. I plan to investigate this issue in my future study.

Chapter 8

Accelerating chemical equilibrium calculation with the non-intrusive reduced basis (NI-RB) method

*"I wish to establish some sort of system not guided by chance
but by some sort of definite and exact principle."*

— Dmitri MENDELEEV (1891)

In connection with Chapter 7, the computational cost of performing chemical equilibrium calculations can be expensive, especially when dealing with complex chemical systems and reactions. It poses a challenge, particularly in executing a single run of reactive transport simulation. Hence, in this Chapter, I employ, for the first time, the non-intrusive reduced-basis (NI-RB) method to accelerate chemical equilibrium calculations. This chapter is covered in Santos et al. (in preparation, d).

8.1 Chemical equilibrium formulations

Chemical equilibrium refers to a condition when the forward and reverse rates of all possible chemical reactions are balanced, corresponding to a state of minimum Gibbs energy from a thermodynamics perspective (Leal et al., 2017; Smith, 1982). A chemical equilibrium calculation is used to calculate changes in the amount of chemical species resulting from chemical reactions, disregarding kinetics considerations (Leal et al., 2017; Smith, 1982). There are two different mathematical formulations for chemical equilibrium problems, namely the law of mass-action (LMA formulation) (Lichtner, 1985; Reed, 1982; Smith, 1982; Morel and Morgan, 1972) and the Gibbs energy minimization (GEM) formulation (Leal et al., 2017; Leal, Kulik, and Kosakowski, 2016). Both formulations are fundamentally the same, yet pose advantages and disadvantages in terms of database usage and numerical stability (Leal et al., 2017). Here I follow the GEM formulation, expressed as: (Kyas et al., 2022; Leal et al., 2020; Leal, Kulik, and Kosakowski, 2016)

$$\min_n G(n; \xi) = \sum_{i=1}^{N_s} \mu_i(n; \xi) n_i \quad \text{subject to} \quad \begin{cases} An = b(\xi), \\ n \geq 0, \end{cases} \quad (8.1)$$

where $G(n; \xi)$ is the Gibbs energy [J/mol] of the chemical system, A is the matrix stoichiometric coefficient, n is the molar amount of chemical species $[n_1, \dots, n_{N_s}]$, N_s is the number of species, μ is the chemical potential and b is the element concentration [mol/m³]. ξ contains physical and chemical parameters, such as temperature, pressure, and initial concentration of chemical species.

I formulate Equation 8.1 under Karush-Kuhn-Tucker (KKT) conditions to robustly solve it, expressed as: (Leal, Kulik, and Kosakowski, 2016)

$$\text{find } (n, y, z) \text{ such that: } \begin{cases} \frac{\partial G(n; \xi)}{\partial n} - A^T y - z = 0, \\ An - b(\xi) = 0, \\ n_i z_i = 0 \ (i = 1, \dots, N_s), \end{cases} \quad (8.2)$$

where y is a Lagrange multiplier and z_i is a slack variable, introduced to ensure non-negativity (Leal et al., 2017; Leal, Kulik, and Kosakowski, 2016). For a detailed derivation of this, I refer to Leal, Kulik, and Kosakowski (2016). The Newton method is then used to solve equation 8.2. During the execution of the Newton method, I solve the following linear systems: (Leal, Kulik, and Kosakowski, 2016)

$$\begin{bmatrix} H(n; \xi) & -A^T & -I \\ A & 0 & 0 \\ Z & 0 & N \end{bmatrix} \begin{bmatrix} \Delta n \\ \Delta y \\ \Delta z \end{bmatrix} = - \begin{bmatrix} \mu(n; \xi) - A^T y - z \\ An - b(\xi) \\ NZe - \tau e \end{bmatrix}, \quad (8.3)$$

during the k th Newton iteration and update (n, y, z) for the next $(k + 1)$ Newton iteration, employing

$$n^{k+1} = n^k + \beta_n^k \Delta n^k, \quad (8.4)$$

$$y^{k+1} = y^k + \Delta y^k, \quad (8.5)$$

$$z^{k+1} = z^k + \beta_z^k \Delta z^k, \quad (8.6)$$

where H is the Hessian matrix of the Gibbs energy function, normalized by RT , where R is the universal gas constant and T is temperature. The mathematical expression of the Hessian matrix is given as: (Leal, Kulik, and Kosakowski, 2016)

$$H := \frac{\partial^2 G}{\partial n^2} = \frac{\partial \mu}{\partial n} = \frac{\partial \ln a}{\partial n}, \quad (8.7)$$

where $N = \text{diag}(n_1, \dots, n_{N_s})$ is a diagonal matrix of n , $Z = \text{diag}(z_1, \dots, z_{N_s})$ is a diagonal matrix of z , $e = [1, 1, \dots, 1]^T$, a is the activity of a chemical species, and τ is a perturbation constant ($\tau > 0$ and is sufficiently small). Furthermore, β_n and β_z are calculated as follows to ensure non-negativity (Leal, Kulik, and Kosakowski, 2016):

$$\beta_v = \min_i \beta_{v_i}, \quad \beta_{v_i} := \begin{cases} -\delta \frac{v_i}{\Delta v_i} & \text{if } v_i + \Delta v_i < 0, \\ 1 & \text{otherwise,} \end{cases} \quad (8.8)$$

where $\delta = 0.9999$ and v is a vector n or z . At every Newton iteration, I check the convergence of the chemical equilibrium calculation through: (Leal, Kulik, and Kosakowski, 2016)

$$\|f(n, y, z)\|_\infty < \epsilon_{tol}, \quad (8.9)$$

where $\|\cdot\|_\infty$ is the infinity norm, defined as $\|\cdot\|_\infty = \max |\cdot|$, and ϵ_{tol} is a tolerance for convergence. I use Reaktoro (Leal, 2015), an open-source C++/Python framework, to perform the chemical equilibrium calculations by solving Equation 8.1 - 8.9.

A challenge when performing reactive transport simulations is the computational cost of solving Equation 8.3, together with the entire iterative Newton procedure, when dealing with numerous complex chemical reactions, often requiring significant

computational resources. Moreover, each chemical equilibrium calculation has to be performed for every mesh cell at every time step. It is thus essential to reduce the cost of these calculations to reduce the overall computational cost of performing reactive transport simulations, particularly when performed in three spatial dimensions, over long durations, at high spatial and/or temporal resolutions, and when involving many reactive species.

Another challenge is solving inverse chemical equilibrium problems, such as titrations. Inverse chemical equilibrium problems require multiple forward chemical equilibrium runs, while performing optimization (Leal and Smith, 2022). This type of calculation can be expensive, especially when dealing with complex and numerous chemical species. The computation also involves a root-finding process for highly non-linear responses, which often exhibit the so-called "shock" features (Leal and Smith, 2022). I illustrate such a "shock" feature in Figure 8.1, showing the CO_2 gas concentration as CO_2 is added to a NaCl - CaCO_3 solution.

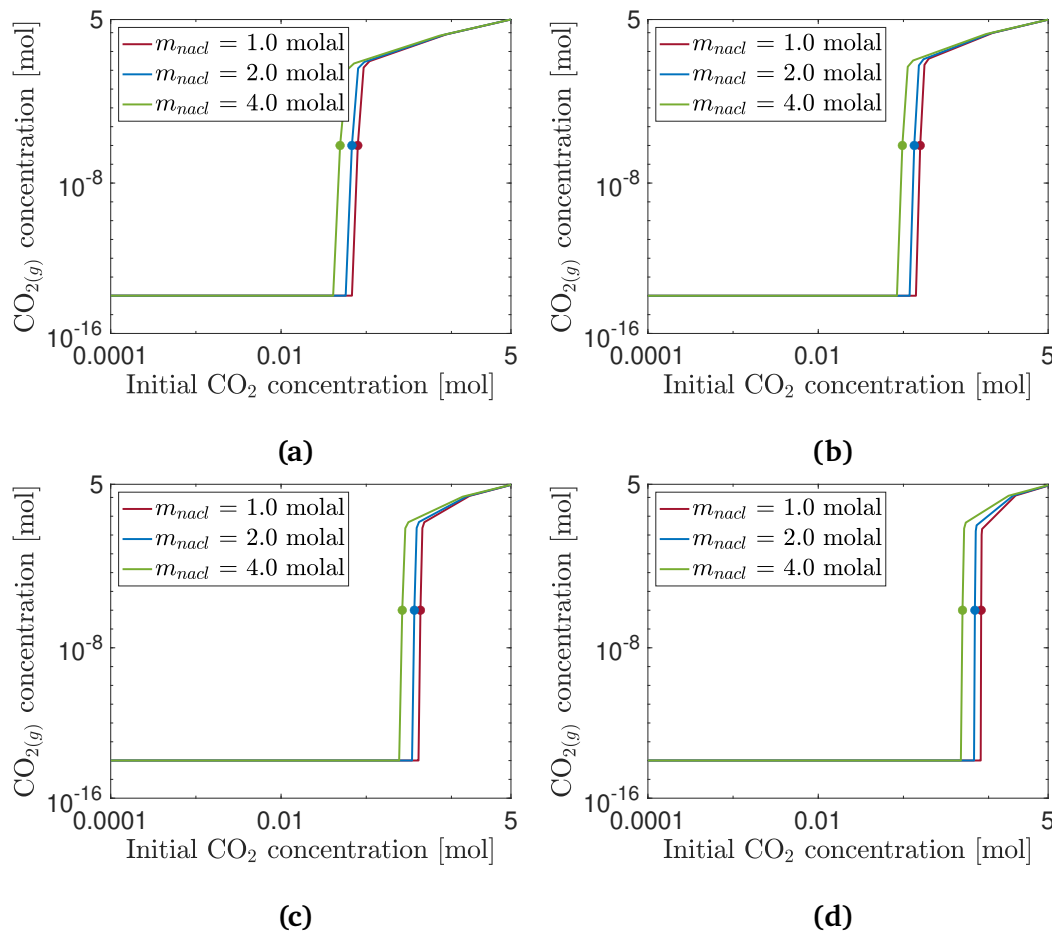


FIGURE 8.1: The amount of $\text{CO}_2(\text{g})$ gas liberated from a CO_2 - NaCl - CaCO_3 solution at 50°C , 1×10^{-4} molal CaCO_3 , and at (a) 5 bar, (b) 10 bar, (c) 30 bar, and (d) 70 bar. Different colors denote different NaCl concentrations. The dots denote the CO_2 solubility, a condition at which the first gas bubble appears. I use 1×10^{-6} mol of gas as the criterion to define solubility. In inverse chemical equilibrium problems, the task is to accurately capture the dots.

8.2 Non-intrusive reduced-basis method formulation

The non-intrusive reduced-basis (NI-RB) method expresses a solution $\hat{z}(\mathbf{x}; \boldsymbol{\xi})$ as a linear combination of basis functions $V(\mathbf{x})$ and coefficients $\alpha(\boldsymbol{\xi})$, written as follows: (Swischuk et al., 2019; Hesthaven and Ubbiali, 2018)

$$\hat{z}(\mathbf{x}; \boldsymbol{\xi}) = \sum_{k=1}^r \alpha_k(\boldsymbol{\xi}) V_k(\mathbf{x}), \quad (8.10)$$

where r is the reduced dimension, $\mathbf{x}$ is the spatial or temporal coordinates, and $\boldsymbol{\xi}$ is the input parameters. k is for the enumeration. The NI-RB method is an extension of the intrusive reduced-basis (I-RB) method, designed to address the limitations of the I-RB method when dealing with hyperbolic or nonlinear partial differential equations (PDEs) (Wang, Hesthaven, and Ray, 2019; Hesthaven and Ubbiali, 2018). The construction of basis functions in the NI-RB method is the same as in the I-RB method, while the Galerkin projection part is replaced by the use of Neural Networks (Wang, Hesthaven, and Ray, 2019; Hesthaven and Ubbiali, 2018). Consequently, I can say that

- Given any matrix system $A(\boldsymbol{\xi})p(\boldsymbol{\xi}) = b(\boldsymbol{\xi})$ either from linear or non-linear equations, as long as we can guarantee that the matrix system is well-posed and has a unique solution for all input parameters $\boldsymbol{\xi}$ within a certain range, we can use the reduced-basis method to generate the basis functions and perform reduction (Hesthaven, Pagliantini, and Ripamonti, 2023; Khan, Shahebul Hasan, and Ng, 2020; Stabile and Rozza, 2018; Hesthaven and Ubbiali, 2018; Hesthaven, Rozza, Stamm, et al., 2016; Negri et al., 2013),
- Even in the presence of non-linear behavior within the system $A(\boldsymbol{\xi})p(\boldsymbol{\xi}) = b(\boldsymbol{\xi})$ (e.g. dependencies among solutions), the construction of basis functions commences by treating $A(\boldsymbol{\xi})p(\boldsymbol{\xi}) = b(\boldsymbol{\xi})$ as a matrix system constructed from elliptic or parabolic PDEs. Non-linearities are then accommodated in the reduced-basis coefficients using a non-linear interpolator such as Neural Networks (Wang, Hesthaven, and Ray, 2019; Swischuk et al., 2019; Mücke et al., 2019; Hesthaven and Ubbiali, 2018; Negri et al., 2013),
- Despite the construction of basis functions using the idea of $A(\boldsymbol{\xi})p(\boldsymbol{\xi}) = b(\boldsymbol{\xi})$ as a matrix system constructed from elliptic or parabolic PDEs, the reduction and surrogate model construction still utilizes the snapshots generated by truly solving $A(\boldsymbol{\xi})p(\boldsymbol{\xi}) = b(\boldsymbol{\xi})$ as a non-linear system, including iterative procedures (Negri et al., 2013).

Since the workflow for constructing the NI-RB surrogate model is already explained in Section 2.1, in this Section, I will only discuss the construction of basis functions. It is crucial since the treatment for constructing the basis functions is different from that in Section 2.1.

8.2.1 Generation of basis functions

Before generating the basis functions, I first write the first-order optimality condition of the chemical equilibrium problem as follows:

$$\text{find } (n, y) \text{ such that: } \begin{cases} \frac{\partial G(n; \boldsymbol{\xi})}{\partial n} - A^T y = 0, \\ An - b(\boldsymbol{\xi}) = 0. \end{cases} \quad (8.11)$$

I can rewrite Equation 8.11 in a matrix system as follows:

$$\begin{bmatrix} \frac{\partial G^*(n; \xi)}{\partial n} & -A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} n \\ y \end{bmatrix} = \begin{bmatrix} G^*(n; \xi) \\ b(\xi) \end{bmatrix}, \quad (8.12)$$

where $\frac{\partial G^*(n; \xi)}{\partial n}$ is the left-hand side component of $\frac{\partial G(n; \xi)}{\partial n}$ and $G^*(n; \xi)$ is the right-hand side component of $\frac{\partial G(n; \xi)}{\partial n}$. The above matrix system can be approximated as a saddle-point problem, following the idea of (Negri et al., 2013; Brezzi and Fortin, 2012).

Negri et al. (2013) construct an intrusive reduced basis (I-RB) approximation of the elliptic optimal control problem by exploiting first-order optimality condition, see Equation 8.11 and Equation 8.12, and using the *optimize-then-discretize-then-reduce* scheme, supported by Brezzi theorem within the saddle-point framework. For detailed proofs and derivations, see Negri et al. (2013). Building upon this approach, I implement Negri et al. (2013)'s approach for constructing basis functions of the KKT problem in the context of the chemical equilibrium problem. The basis functions are constructed from an aggregated space of n and y : $z = [n_1, \dots, n_{N_s}, y_1, \dots, y_{N_e}]$ (Negri et al., 2013). Now I see n and y as x in Equation 8.10. Then a Proper Orthogonal Decomposition (POD) is used to generate basis functions from given training snapshots $Q = [\mathbf{q}_1, \dots, \mathbf{q}_{N_{tr}}] \in \mathbb{R}^{N_{tr} \times (N_s + N_e)}$ where N_{tr} is the number of training samples and N_e is the number of elements (Swischuk et al., 2019; Hesthaven and Ubbiali, 2018).

8.3 Comparison strategy

To check the suitability of using the non-intrusive reduced-basis (NI-RB) method for constructing a surrogate model for the chemical equilibrium problem, I present two setups:

- forward chemical equilibrium problem: I solve chemical equilibrium with a given temperature, pressure, and initial concentration of reactants to predict the concentration of chemical products,
- inverse chemical equilibrium problem: I solve the forward chemical equilibrium problem multiple times to find a value of an input parameter that induces a certain phenomenon, e.g. titration problem.

I also use the data-driven machine learning method to construct another surrogate model, providing a comparison for the NI-RB method. For each setup, I compare the performance of the NI-RB and data-driven machine learning method with the scenarios in Figure 8.2.

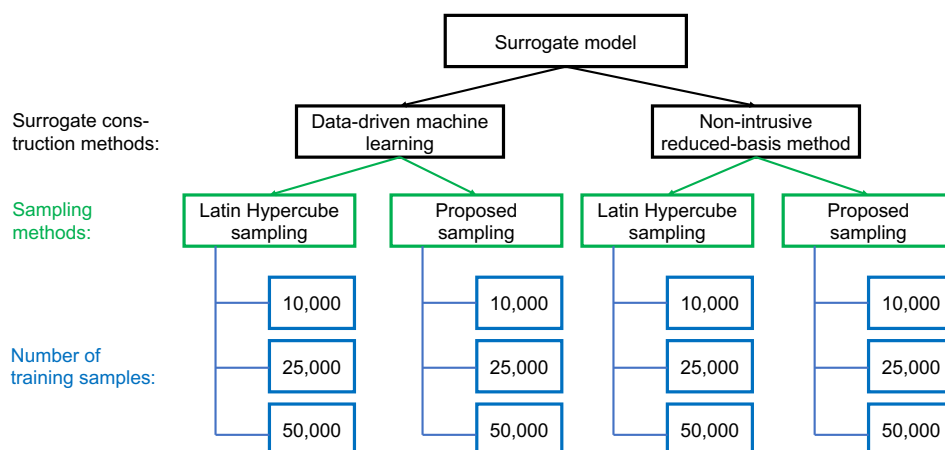


FIGURE 8.2: Scenario used for constructing the surrogate models.

I present two sampling strategies for generating training samples used for constructing the surrogate model:

- Latin Hypercube sampling method: as explained in Section 2.1.2, The Latin Hypercube sampling provides space-filling samples and assumes uniform distribution for each input parameter. For this sampling method, I will construct the surrogate model with 10000, 25000, and 50000 training and test samples.
- Proposed sampling method: The idea of the proposed sampling method is to provide fine samples around the "shock" features. It is problem-dependent; hence, the details will be in Section 8.4. For this sampling method, I will construct the surrogate model with 10000, 25000, and 50000 training and test samples.

8.4 Case study

I use a case study of mixing CO_2 - CaCO_3 - NaCl in H_2O , see Figure 8.3.

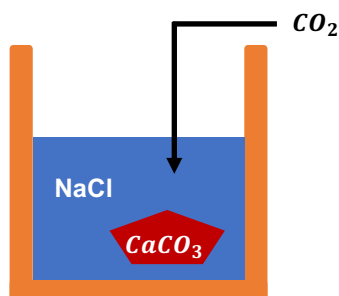


FIGURE 8.3: Illustration of the case study.

For the conditions chosen for training, three phases can coexist in equilibrium: aqueous solution, solid CaCO_3 , and a CO_2 -rich gaseous/supercritical phase with CO_2 and H_2O . The phases and chemical species considered are given in Table 8.1. I select the following five parameters, which are varied for training purposes:

- T : temperature
- P : pressure

- c_{NaCl} : initial NaCl concentration
- c_{CO_2} : initial CO_2 concentration
- c_{CaCO_3} : initial CaCO_3 concentration

Their value ranges are shown in Table 8.2, covering possible extreme conditions. I have 18 degrees of freedom as the solution to the chemical equilibrium problem, containing 6 Lagrange multipliers and 12 concentrations.

TABLE 8.1: Chemical systems for the case study.

Elements:	C, Ca, Cl, H, Na, O, Z*		
Phases:	Aqueous, Calcite, CO_2 gas, H_2O gas		
Species:	$\text{CO}_{2(\text{aq})}$	$\text{Cl}^-_{(\text{aq})}$	$\text{H}_2\text{O}_{2(\text{aq})}$
	$\text{CO}_3^{2-}_{(\text{aq})}$	$\text{ClO}^-_{(\text{aq})}$	$\text{HCO}_3^-_{(\text{aq})}$
	$\text{Ca}(\text{HCO}_3)^+_{(\text{aq})}$	$\text{ClO}_2^-_{(\text{aq})}$	$\text{HCl}_{(\text{aq})}$
	$\text{Ca}_2^+_{(\text{aq})}$	$\text{ClO}_3^-_{(\text{aq})}$	$\text{HClO}_{(\text{aq})}$
	$\text{Na}^+_{(\text{aq})}$	$\text{NaCl}_{(\text{aq})}$	$\text{NaOH}_{(\text{aq})}$
	$\text{CaCO}_{3(\text{aq})}$	$\text{ClO}_4^-_{(\text{aq})}$	$\text{HClO}_{2(\text{aq})}$
	$\text{CaCl}^+_{(\text{aq})}$	$\text{H}^+_{(\text{aq})}$	$\text{HO}_2^-_{(\text{aq})}$
	$\text{CaCl}_{2(\text{aq})}$	$\text{H}_{2(\text{aq})}$	$\text{O}_{2(\text{aq})}$
	$\text{CaOH}^+_{(\text{aq})}$	$\text{H}_2\text{O}_{(\text{l})}$	$\text{OH}^-_{(\text{aq})}$
	$\text{CO}_{2(\text{g})}$	$\text{H}_2\text{O}_{(\text{g})}$	Calcite

Z* represents an electric charge

TABLE 8.2: Physical and chemical parameter ranges for the case study.

Input parameters	Lower bound	Upper bound
T [$^{\circ}\text{C}$]	25.00	90.00
P [bar]	1.00	100.0
c_{NaCl} [molal]	1×10^{-4}	5.000
c_{CO_2} [molal]	1×10^{-4}	5.000
c_{CaCO_3} [molal]	1×10^{-4}	2.000

I already show in Figure 8.1 the main challenge in the selected case study. The "shock feature" (i.e., when the solution becomes saturated with respect to the gas or mineral) poses a challenge in predicting with machine learning models given the sudden change in smoothness of the concentrations.

8.4.1 Proposed sampling strategy

To generate a high-quality training set, the following steps should be followed, which I found to be very important, particularly the logarithmic scaling of variables and identifying where phase transitions occur so that more training samples are considered in these regions:

- Uses log scaling to parameters having more than two or three orders of magnitude ranges: If we look at Table 8.2, we can identify three chemical parameters, c_{NaCl} , c_{CO_2} , and c_{CaCO_3} , having nearly five orders of magnitude range. A log scaling narrows these ranges, hence, guarantees representative sampling.
- Identifies shock features on chemical concentration profiles: The highlight in my case study is the concentration of CO_2 gas as it experiences phase transition. If we look at Figure 8.1, the concentration of CO_2 gas has a shock feature. If I increase either pressure, temperature, or other chemical parameters, these shock features have similar patterns and they just shift either to left or right.
- Performs sampling on parameters, that do not create any shock features on either chemical concentration profiles or Lagrange multipliers, using the Latin Hypercube method: Since it is only CO_2 gas concentration has a shock feature and is only affected by initial CO_2 concentration, pressure, temperature, initial c_{NaCl} , and initial c_{CaCO_3} are sampled using the Latin Hypercube method.
- Calculate CO_2 solubility for each realization: This calculation provides the first sample (green dot in Figure 8.4) and it is the basis to generate the other points to capture the shock feature
- performs the following sampling for each realization:
 - Take the CO_2 solubility as the first sample
 - Generate the second and third sample by adding and subtracting 25% of the value of Sample 1. The selection of 25%-value is empirical and it can vary depending on the problem. However, it is critical to place the second and third at the two ends of the shock feature
 - Generate Sample 6 that equally divides the interval between Sample 3 and 5
 - Generate Sample 7 that equally divides the interval between Sample 3 and 6
 - Generate Sample 8 that equally divides the interval between Sample 3 and 7
 - Generate Sample 9 that equally divides the interval between Sample 2 and Sample 4
 - Generate Sample 10 that equally divides the interval between Sample 2 and Sample 9
- Redo all the steps for other training sets with different number of samples
- Perform all these steps to generate test samples.

In my case study, the number of training samples equals the number of test samples. Ten samples of the initial concentration CO_2 are kept for each realization obtained by Latin hypercube sampling. This number proves sufficient to capture the shock feature

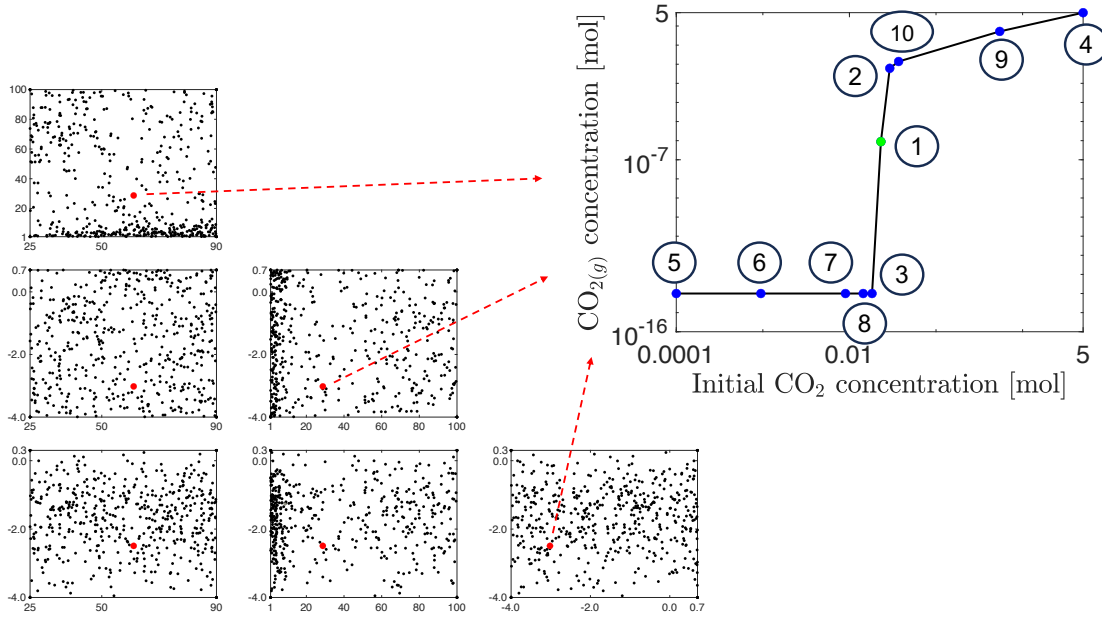


FIGURE 8.4: Illustration of sampling strategy used for the case study. The red dot refers to a single realization. The six plots at the left corner denote 500 samples of pressure, temperature, initial c_{NaCl} , and initial c_{CaCO_3} , generated using the Latin Hypercube sampling method. These 500 samples are plotted in two-dimensional format (from top to bottom, then left to right): temperature vs pressure, temperature vs $\log_{10} c_{\text{NaCl}}$, pressure vs $\log_{10} c_{\text{NaCl}}$, temperature vs $\log_{10} c_{\text{CaCO}_3}$, pressure vs $\log_{10} c_{\text{CaCO}_3}$, $\log_{10} c_{\text{NaCl}}$ vs $\log_{10} c_{\text{CaCO}_3}$. Within the red dot, ten samples of initial CO_2 concentration are generated following the enumeration at the right corner figure. Number 1 (green dot) is associated with CO_2 solubility, Number 2 is a 1.25 value of Number 1, Number 3 is a 0.75 value of Number 1, Number 4 is the upper bound, Number 5 is the lower bound, Number 6 is $\frac{1}{2}$ (value of Number 3 + value of Number 5), Number 7 is $\frac{1}{2}$ (value of Number 3 + value of Number 6), Number 8 is $\frac{1}{2}$ (value of Number 3 + value of Number 7), Number 9 is $\frac{1}{2}$ (value of Number 2 + value of Number 4), and Number 10 is $\frac{1}{2}$ (value of Number 2 + value of Number 9). The total number of samples above is $500 \times 10 = 5000$.

and ensures good coverage for other parameters. For example, to obtain 5000 samples (see Figure 8.4), I use the Latin Hypercube sampling method 500 times for pressure, temperature, initial c_{NaCl} , and initial c_{CaCO_3} . Subsequently, we generate 10 samples of initial CO_2 concentration for each realization of pressure, temperature, initial c_{NaCl} , and initial c_{CaCO_3} .

8.5 Results

In this section, I compare the non-intrusive reduced-basis (NI-RB) method (POD-NN) described in Section 2.1 with a purely data-driven machine learning method (NN) when applied for the case study in Section 8.4. My goal is to demonstrate how different sampling strategies impact the predictive capabilities of both physics-based and data-driven machine learning models for chemical equilibrium calculations. In particular, I investigate how accurately these surrogate models are when estimating saturation states of aqueous solutions with respect to gases and minerals (i.e., solubility calculations), when phases become thermodynamically stable and cause changes in the smoothness of concentration profiles. Furthermore, I investigate the influence of varying the number of training samples on the accuracy of predictions for both approaches.

Lastly, I evaluate the efficiency of both physics-based and data-driven machine learning methods.

The mass imbalance phenomenon, denoted in Kyas et al. (2022) and Leal et al. (2020), arises from inaccurate predictions of the concentration of chemical species at the equilibrium condition. This discrepancy results in the total mass of these predictions not being equivalent to the initial mass. In this study, I am more restrictive by ensuring accurate predictions for each concentration of chemical species. Therefore, I introduce an additional relative-error metric $e_{relative}$ to assess the accuracy of physics-based and data-driven machine learning model predictions, expressed mathematically as follows:

$$e_{relative} = \frac{\|\mathbf{z}(\mathbf{x}; \boldsymbol{\xi}) - \hat{\mathbf{z}}(\boldsymbol{\xi}; \mathbf{W}, \mathbf{B})\|_{pr}}{\|\mathbf{z}(\mathbf{x}; \boldsymbol{\xi})\|_{pr}} \quad (8.13)$$

where pr denotes the prediction/blind set that is ensured not to be within both the training and test set. I perform all training and training set generations using a single core on an Apple Macbook Pro with 8 GB RAM.

8.5.1 Impact of sampling strategy

In Figure 8.5, relative error profiles for concentrations of various chemical species are presented, comparing the predictions obtained from both physics-based and data-driven machine learning methods. Five hundred prediction samples are used that are not in the training and test sets. Notably, with 50,000 training samples generated using my proposed sampling strategy, the POD-NN surrogate model delivers more accurate predictions compared to the data-driven NN surrogate model. The data-driven NN model specifically exhibits lower accuracy in predicting the concentrations of chemical species undergoing phase transitions, such as $\text{H}_2\text{O}_{(g)}$ and $\text{CO}_{2(g)}$. Moreover, when both the POD-NN and data-driven NN surrogate models are trained with 50,000 samples generated by the Latin hypercube sampling method, they produce inaccurate predictions for certain aqueous species and those undergoing phase transitions.

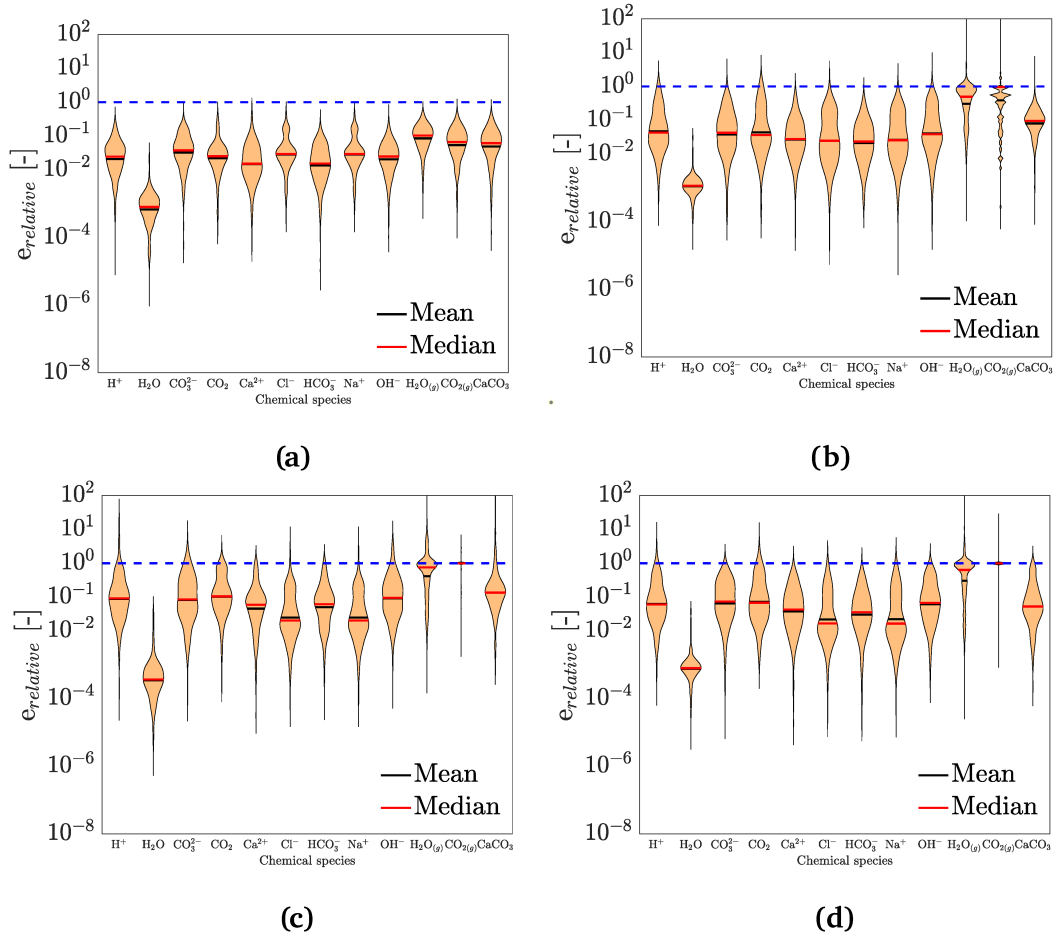


FIGURE 8.5: Relative error distribution for 500 prediction samples obtained using (a) POD-NN trained with 50,000 training samples, generated via proposed sampling strategy in Section 8.4.1, (b) data-driven NN trained with 50,000 training samples, generated via proposed sampling strategy in Section 8.4.1, (c) POD-NN trained with 50,000 training samples, generated via Latin Hypercube sampling, and (d) data-driven NN trained with 50,000 training samples, generated via Latin Hypercube sampling.

Figure 8.6 presents a comparison between POD-NN and data-driven NN methods in solving the inverse chemical problem, specifically predicting CO_2 solubility. The POD-NN surrogate model, trained with 50,000 samples generated by my proposed sampling strategy, successfully predicts CO_2 solubility in varying pressure, temperature, and salinity conditions. In contrast, the data-driven NN surrogate model deviates significantly from the data. Both the POD-NN and the data-driven NN surrogate models, trained with 50,000 samples generated by the Latin Hypercube sampling method, exhibit inaccurate predictions in all pressure, temperature, and salinity scenarios. In conclusion, the sampling strategy significantly affects the accuracy of both POD-NN and data-driven NN models.

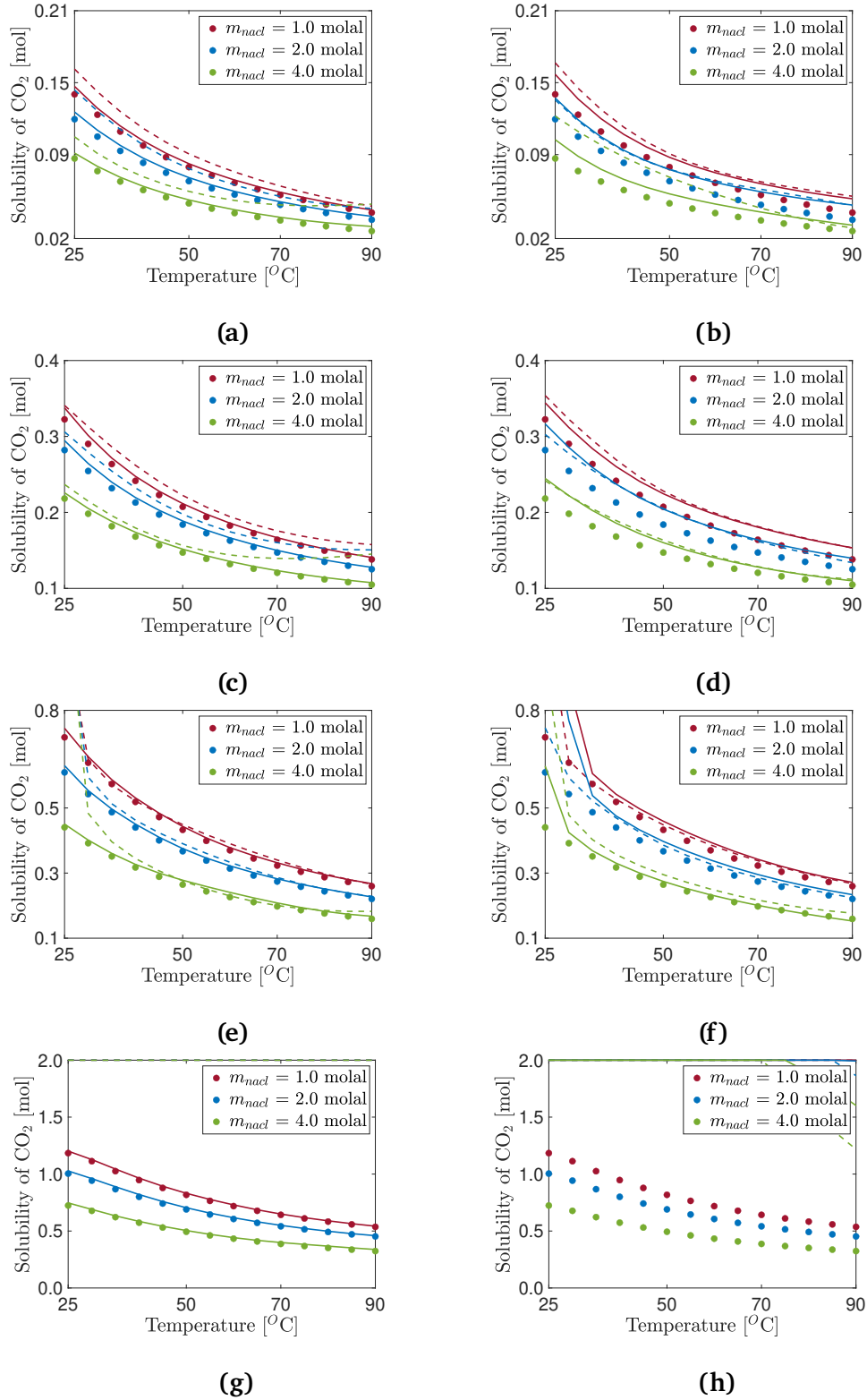


FIGURE 8.6: Comparison between POD-NN (solid lines) and data-driven NN (dashed lines) in predicting the CO_2 solubility at different pressure, temperature, and salinity. Figure (a), (c), (e), and (g) are produced by training both POD-NN and data-driven NN with 50,000 training samples generated via my proposed sampling strategy. Figure (b), (d), (f), and (h) are produced by training both POD-NN and data-driven NN with 50,000 training samples generated via the Latin Hypercube sampling method. Figure (a) and (b) are in 5 bar pressure, (c) and (d) are in 10 bar pressure, (e) and (f) are in 30 bar pressure, and (g) and (h) are in 70 bar pressure.

8.5.2 Impact of number of training samples

In Figure 8.7, I compare the POD-NN and data-driven NN surrogate models trained with varying numbers of samples generated through my proposed sampling strategy. As the number of training samples increases, both the POD-NN and data-driven NN surrogate models exhibit improved prediction accuracy. Notably, the predictions of POD-NN model consistently surpass those of the data-driven NN model. Even with 50,000 training samples, the data-driven NN model struggles to accurately predict concentrations, particularly for chemical species undergoing phase transitions.

Figure 8.8 compares the performance of the POD-NN and data-driven NN surrogate models, trained with varying numbers of training samples, in predicting CO₂ solubility. As the number of training samples increases, the accuracy of the POD-NN model's predictions improves consistently. In contrast, the accuracy of the data-driven NN fluctuates; although it improves at low pressures (5 bar and 10 bar) with more training samples, it exhibits variability at different pressures, compensating for high accuracy at these low pressures. Even with 50,000 training samples, the data-driven NN surrogate model fails to accurately capture CO₂ solubility.

8.5.3 Offline and online time

In Table 8.3, the architectures of the POD-NN and the data-driven NN surrogate models, employed to calculate offline and online times, are presented. The offline time is defined as the time needed to construct a surrogate model either using a physics-based or data-driven machine learning method, disregarding the time needed to generate training and test samples. The online time is the time needed for a trained surrogate model to predict a new case. The table shows that the POD-NN model has fewer neurons in its layers compared to the data-driven NN model.

TABLE 8.3: POD-NN and data-driven NN architecture, trained with 50,000 training samples generated via my proposed sampling strategy, for the calculation of offline and online time.

Surrogate model	Number basis	Neurons in Layer 1	Neurons in Layer 2	Learning rate	Number epochs	Batch size
POD-NN	16	87	40	6.5×10^{-4}	40,657	2,620
NN	-	155	155	8.0×10^{-4}	46,807	18,971

In Figure 8.9, aligned with the information in Table 8.3, it is shown that the data-driven NN method exhibits a longer offline time than the POD-NN method due to its more complex architecture. Consequently, the data-driven NN surrogate model is slightly slower in predicting a new case compared to the POD-NN surrogate model. Despite these differences, both the data-driven NN and the POD-NN surrogate models offer a substantial speedup of nearly three orders of magnitude compared to Reaktoro.

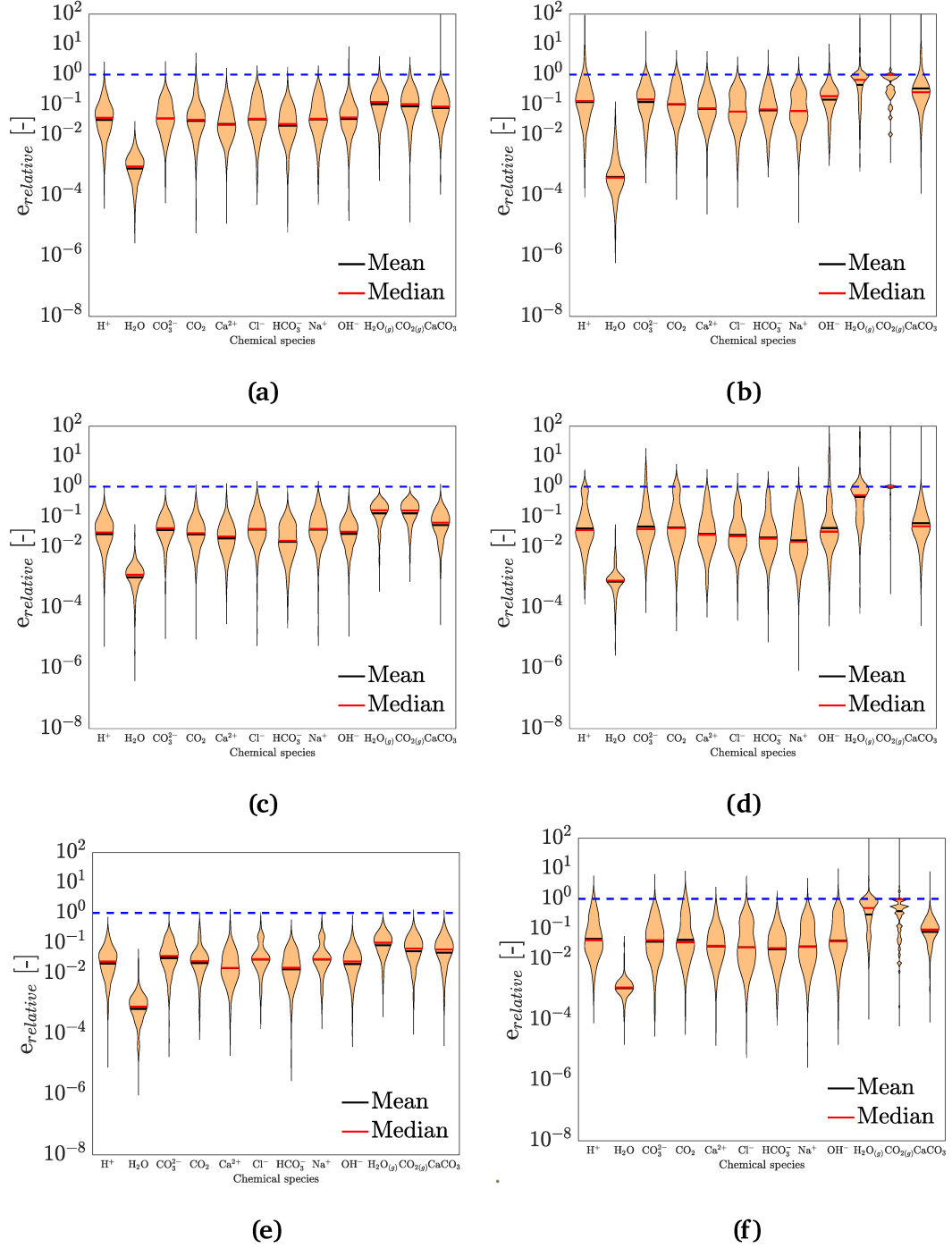


FIGURE 8.7: Relative error distribution for 500 prediction samples obtained using (a) POD-NN trained with 10,000 training samples, (b) data-driven NN trained with 10,000 training samples, (c) POD-NN trained with 25,000 training samples, (d) data-driven NN trained with 25,000 training samples, (e) POD-NN trained with 50,000 training samples, and (f) data-driven NN trained with 50,000 training samples.

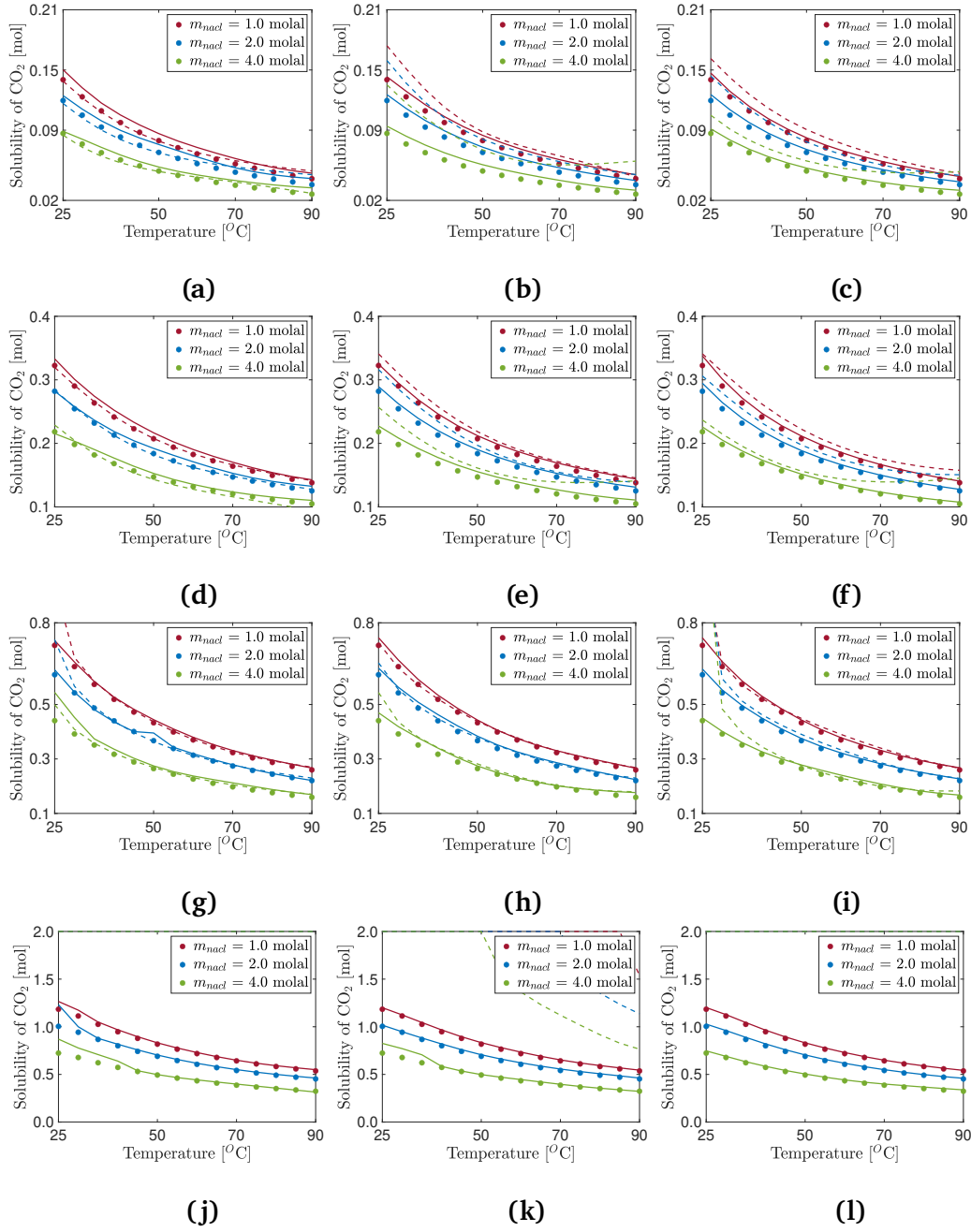


FIGURE 8.8: Comparison between POD-NN (solid lines) and data-driven NN (dashed lines), trained with different numbers of training samples generated via my proposed sampling strategy, in predicting CO_2 solubility at different pressure, temperature, and salinity. Figure (a), (d), (g), and (j) are trained with 10,000 training samples. Figure (b), (e), (h), and (k) are trained with 25,000 training samples. Figure (c), (f), (i), and (l) are trained with 50,000 training samples. Figure (a), (b), and (c) are in 5 bar pressure, Figure (d), (e), and (f) are in 10 bar pressure, Figure (g), (h), and (i) are in 30 bar pressure, and Figure (j), (k), and (l) are in 70 bar pressure.

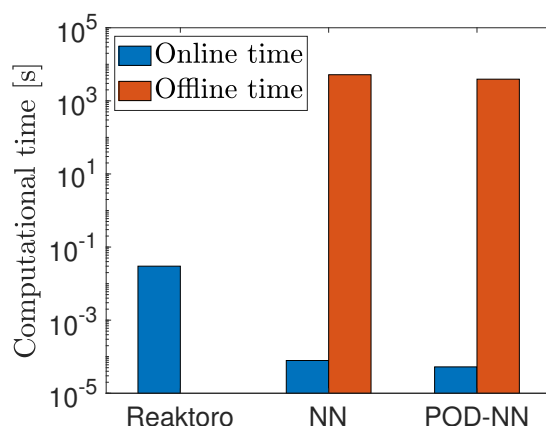


FIGURE 8.9: Comparison of offline and online time using Reaktoro, data-driven NN, and POD-NN.

8.6 Discussion

Constructing a high-quality dataset for the training of non-intrusive reduced-basis surrogate models or data-driven machine-learning models is crucial to ensuring accurate predictions. Figure 8.6 illustrates the substantial increase in the accuracy of both POD-NN and data-driven NN model predictions when employing a training set generated through my proposed sampling strategy. Given that the solution we aim to capture with either a physics-based or data-driven machine-learning model is highly non-linear and contains a shock feature, it is imperative to encompass the entire shape of this solution in the training set. This inclusion ensures that a physics-based or data-driven machine learning model recognizes such distinctive features. Note that any surrogate model will not inherently capture nonlinearity, such as a shock feature, without encoding them in the training samples (Pichi et al., 2023; Fu et al., 2023; Pichi et al., 2022; Strazzullo et al., 2018; Negri et al., 2013). Finally, the POD-NN model trained with a training set generated via my proposed sampling strategy can deliver accurate predictions for both forward and inverse chemical equilibrium problems.

From Figure 8.7 and 8.8, it is evident that the POD-NN model, with a maximum of 50,000 training samples generated through my proposed sampling strategy, achieves accurate predictions for both forward and inverse chemical equilibrium problems. In contrast, even with the same number of training samples, the data-driven NN model predictions are significantly inaccurate. This discrepancy arises because the POD-NN method maintains the structure of the chemical equilibrium problem, including shock features, within its basis functions. Consequently, the POD-NN surrogate model can achieve accurate predictions with a significantly smaller number of training samples (Willcox, Ghattas, and Heimbach, 2021; Wang, Hesthaven, and Ray, 2019; Hesthaven and Ubbiali, 2018; Strazzullo et al., 2018; Negri et al., 2013). Therefore, compared to the data-driven NN method, the POD-NN method requires substantially fewer training samples to produce accurate predictions for both forward and inverse chemical equilibrium problems.

The POD-NN method is more efficient than the data-driven NN method, requiring a shorter offline time and providing nearly three orders of magnitude improvement compared to Reaktoro. This speedup is critical for addressing both forward and inverse

chemical equilibrium problems involving complex chemical reactions or large-scale reactive transport simulations.

Looking at Figure 8.9 and considering the construction of a POD-NN model with 50,000 training samples, achieving accurate predictions involves 100,000 runs of the Reaktoro model (50,000 training samples and 50,000 test samples), in addition to the offline time. Considering the overall time needed to generate samples and construct a surrogate model, this effort is paid back after performing 230,700 forward chemical equilibrium calculations (meaning that it is useful for large-scale/high-dimensional reactive transport simulations). This amount of resources spent on the construction of the POD-NN model remains relatively large considering the recent success of the implementation of the POD-NN method for non-linear high-dimensional problems (Cao et al., 2023; Wu et al., 2023; Degen, Cacace, and Wellmann, 2022; Swischuk et al., 2020; Wang, Hesthaven, and Ray, 2019). Future research should focus on further reducing offline time through hyper-reduction approaches to generate basis functions, such as utilizing the Autoencoder and its variants (Ivagnes, Demo, and Rozza, 2023; Cicci, Fresca, and Manzoni, 2022; Romor, Stabile, and Rozza, 2022; Kadeethum et al., 2022; Kadeethum and Yoon, 2022).

In Solid Earth applications, performing reactive transport simulations is expensive, considering the high dimensionality of subsurface numerical models and complex chemical reactions (Kyas et al., 2022; Leal et al., 2020; Berkowitz et al., 2016; Steefel, DePaolo, and Lichtner, 2005). The use of the POD-NN surrogate model to replace geochemical calculations reduces computational costs, therefore, enables realistic large-scale reactive transport simulations.

Chapter 9

Closure

*"Words are combined into sentences,
this combination answering to that of ideas into thoughts."*
— Henry SWEET (1900)

9.1 Conclusions

I present the conclusions of my studies by addressing the scientific questions outlined in Section 1.6:

1. **Which process-based machine learning method is suitable for optimization under uncertainties and parameter estimation studies in geothermal applications?**

Throughout this dissertation, I demonstrate that the non-intrusive reduced-basis (NI-RB) method, a process-based machine learning method, is very promising for parameter estimation and optimization under uncertainties in geothermal applications. These applications are characterized by limited measurement data, high-dimensional numerical models, and nonlinear multiphysics and chemistry.

The term *process-based machine learning method* emphasizes that the non-intrusive reduced-basis (NI-RB) method, a physics-based machine learning method discussed in the literature (Degen et al., 2023b; Willcox, Ghattas, and Heimbach, 2021), can be extended to address both multiphysics and chemistry processes in geothermal applications.

The NI-RB method yields physically/chemically consistent predictions and requires fewer training samples compared to the Physics-Informed Neural Network (PINN) method and the data-driven machine learning method for achieving similar predictive accuracy (see Chapter 3). Its effectiveness is validated in addressing high-dimensional and nonlinear transient thermo-hydraulic problems, as well as steady-state thermal problems, in the The Hague, Netherlands, case study (see Chapters 4, 5, and 6). The use of the NI-RB method results in more than six orders-of-magnitude speedup compared to the finite element model, while maintaining high accuracy. This speedup facilitates large-scale global sensitivity analyses, uncertainty quantification, and optimization under uncertainties to determine optimal operating conditions for heat extraction in The Hague.

For problems involving chemistry (see Chapters 7 and 8), the NI-RB surrogate model successfully captures "shock" features, ensuring high accuracy. Additionally, it provides a three to four orders-of-magnitude speedup compared to the

finite volume method, which constitutes a nonlinear constrained optimization problem (the Gibbs Energy Minimization for calculating chemical equilibria).

Although the NI-RB method is effective for spatial and temporal reductions, it is inefficient in dealing with high-dimensional parameters. The existence of high-dimensional parameters significantly increases the amount of resources and time required during the offline phase, owing to the increased number of physics-based model runs. This poses a challenge during the offline phase. A possible solution is to use the Active Subspaces (AS) approach to reduce the parameter space (Wu et al., 2023; O’Leary-Roseberry et al., 2022b; O’Leary-Roseberry et al., 2022a; Romor et al., 2022; Constantine, 2015). I am aware of the necessity to address this problem in future research.

Another challenge arises when dealing with highly nonlinear problems, such as the presence of distinct ‘shock’ features for various parameter realizations, as observed in chemistry problems. This may result in less efficient reductions, leading to long offline times, particularly during the training of the machine learning model. To address this issue, several solutions can be considered: utilizing the Autoencoder (AE) approach for spatial and temporal reduction (Pichi, Moya, and Hesthaven, 2024; Romor, Stabile, and Rozza, 2022; Wu et al., 2021; Lee and Carlberg, 2020) or employing the domain decomposition method to reduce problem complexity by dividing it into several sub-problems (Discacciati and Hesthaven, 2024; Kapteyn et al., 2022).

2. How an uncertainty quantification of the entire physical state can be used for well targeting?

A benefit of using the non-intrusive reduced-basis (NI-RB) method is that I can efficiently perform uncertainty quantification (UQ) of the entire state of a nonlinear and high-dimensional problem, not restricted to its quantity of interest or observation space. For the steady-state thermal process in The Hague, Netherlands (see Chapter 4), I gain more than seven orders-of-magnitude speedup by using the NI-RB surrogate model, thereby enabling UQ of the entire state requiring only 3.5 hours employing a single core on an Apple Macbook Pro, equipped with an M1 chip and 8 GB of RAM.

Here, I use global sensitivity analysis (GSA)-driven UQ to ensure the robustness of the process and reduce the computational cost of performing model runs (Degen, Veroy, and Wellmann, 2022; Degen and Cacace, 2021). It is crucial to note that, despite the reduction of the computational cost, facilitated by GSA, the sheer number of model runs remains substantial. Direct execution with the high-dimensional process-based model is not feasible, necessitating the use of the NI-RB surrogate model (Degen, Veroy, and Wellmann, 2022; Degen and Cacace, 2021). Based on the UQ results, I obtain the temperature variation around the proposed producer location ranging from 69 °C to 77 °C. The expected temperature is around 75 °C to fulfill 5 MW_{th} target (Mottaghy, Pechnig, and Vogt, 2011). Therefore, it is recommended to place a producer at the proposed location.

Furthermore, the temperature variation range identified (69°C to 77°C) is utilized in the optimization process detailed in Chapter 5. This information helps establish the bounds for the fluid injection rate and injection temperature.

3. What are the optimal operating conditions for the injection process, considering subsurface uncertainties and safety?

The optimal operating conditions are obtained by solving the Reliability-based Robust Design (RRDO) optimization, considering subsurface parameter uncertainties and safety. This optimization formulation requires an extremely large number of model runs because it has to calculate probabilistic constraints while performing stochastic optimization. In this case, the use of the non-intrusive reduced-basis (NI-RB) method is essential to reduce the computational cost. Thereafter, enable solving the RRDO for a non-linear and high-dimensional model. I gain more than six orders-of-magnitude speedup when calculating a new transient thermo-hydraulic solution for The Hague, Netherlands. Hence, I can solve the RRDO formulation for The Hague, Netherlands, in 33.3 hours with a single core on an Apple Macbook Pro, equipped with an M1 chip and 8 GB of RAM.

The RRDO formulation accounts for the full distribution of the constraint response, enabling an explicit specification of a quantitative safety level. This approach differs from the mean-constraint response and the mean-properties formulation (deterministic constraint formulation), which typically leads to less conservative and potentially unsafe results when aiming to achieve a specific objective function target.

Through the RRDO formulation, I obtain an injection rate of 43.02 L/s and an injection temperature of 37.88 °C with more than 84% probability of safety as the optimum operating conditions for the heat extraction process in The Hague, Netherlands. In contrast, alternative formulations, such as mean-responses and mean-properties, yield higher thermal output rates but with elevated injection rates, lacking a strong safety guarantee. It is worth noting that the previous study by Mottaghy, Pechinig, and Vogt (2011) suggests a more conservative injection rate and temperature, leading to an inability to meet the 5 MW_{th} target.

In connection with the third research question, I added an additional research question to improve my understanding of the transient thermo-hydraulic process in The Hague. The conclusion is as follows:

- **Is entropy production a suitable objective function for a global sensitivity analysis of transient thermo-hydraulic processes?**

The use of entropy production as an objective function in the global sensitivity analysis (GSA) improves our understanding of the transient thermo-hydraulic process under epistemic uncertainties in The Hague, Netherlands. This understanding can be achieved since the entropy production is based on the Second Law of Thermodynamics, accounting for the contribution of irreversibility of thermal processes and fluid flow friction. However, this understanding cannot be achieved using pressure and temperature states as objective functions, since they are designed to focus solely on either a thermal process or on fluid flow. Additionally, the use of non-intrusive reduced-basis (NI-RB) methods allows conducting the GSA in approximately 10 minutes with a single core on an Apple Macbook Pro, equipped with an M1 chip and 8 GB of RAM.

In relation to the extended implementation of the non-intrusive reduced-basis method for chemistry problems, I raised two additional research questions. The conclusions are as follows:

- **Can the non-intrusive reduced basis (NI-RB) method be used to accelerate entire reactive transport simulations?**

The non-intrusive reduced-basis (NI-RB) method is very promising for surrogate

modeling in reactive transport simulations. To the best of my knowledge, I provide here the first implementation of the NI-RB method for reactive transport simulations. The NI-RB surrogate model delivers accurate predictions, capturing "shock" features, with a smaller number of training samples compared to the use of data-driven machine learning models. I gain nearly three orders-of-magnitude speedup compared to the use of finite-volume methods for dealing with a dolomitization problem, a benchmark problem for reactive transport simulations. This provides a path for further use in parameter estimation and optimization under uncertainties for large-scale reactive transport problems.

- **How can the non-intrusive reduced basis (NI-RB) method be used to accelerate chemical equilibrium calculations?**

Inspired by the implementation of the intrusive reduced-basis (I-RB) method for optimal elliptic control problems by Negri et al. (2013), I show the implementation of the non-intrusive reduced-basis (NI-RB) method for chemical equilibrium calculations. Accelerating chemical equilibrium calculations are crucial when dealing with complex chemical reactions, as this computation must be performed for every cell at each time step in a reactive transport simulation. In addition, a new sampling technique is introduced to address phase transitions during chemical equilibrium calculations. The NI-RB surrogate model demonstrates the ability to provide accurate predictions for both forward and inverse chemical equilibrium problems with a smaller number of training samples compared to using a data-driven machine learning model.

Furthermore, my approach achieves nearly a three-orders-of-magnitude speedup compared to employing the Gibbs Energy Minimization (GEM) solver from Reaktoro (Leal, 2015) for chemical equilibrium calculations involving phase transitions. Notably, this marks the first implementation of the NI-RB method for solving chemical equilibrium problems.

However, considering the successful implementation of the NI-RB method in Chapter 3 to 7, the number of training samples used here is still too large, leading to a long offline time. This results from bypassing a constrained optimization problem that is highly nonlinear. This problem can be solved by using an Autoencoder (AE) approach to achieve better reductions than the Proper Orthogonal Decomposition (POD) approach (Pichi, Moya, and Hesthaven, 2024; Romor, Stabile, and Rozza, 2022; Wu et al., 2021; Lee and Carlberg, 2020). I plan to address this problem in future research.

The implementation of a non-intrusive reduced-basis (NI-RB) method in this dissertation uses our own Python library, namely fast. It offers flexibility in selecting different machine learning models for the calculation of reduced-basis coefficients, depending on the complexity of a problem, and in implementing, if needed, reductions in parameter space. I prioritize maintaining the user-friendliness of the library without compromising its scientific integrity. In the future, I plan to add more features, such as an Autoencoder approach for performing reductions of highly non-linear problems.

In conclusion, the use of non-intrusive reduced basis methods is very promising for high-dimensional and large-scale parameter estimation and optimization under uncertainties in multiphysics and chemistry problems in geothermal applications. Furthermore, in the field of chemistry, this opens new opportunities to enable large-scale reactive transport simulations and fast chemical computation for complex chemical reactions.

9.2 Potential future extensions

Throughout this dissertation, I have demonstrated that the use of the non-intrusive reduced-basis (NI-RB) method is highly promising for parameter estimation and optimization under uncertainties in geothermal applications, particularly in addressing large-scale and high-dimensional multiphysics and chemistry problems. However, several limitations in the NI-RB method, as shown in this dissertation, deserve further investigation. Some suggested directions for future research include the following:

- *Reduction in parameter space:* In this dissertation, I do not use a large number of parameters, as it demands a large number of model runs, resulting in an extremely long offline time and exhaustive generation of training samples. Reduction in parameter space is crucial when dealing with reactive transport simulations (Kyas et al., 2022; Leal et al., 2020; Jatnieks et al., 2016; Leal, Blunt, and LaForce, 2015) and heterogeneous media (Wu et al., 2023; O’Leary-Roseberry et al., 2022a; O’Leary-Roseberry et al., 2022b). A solution is to use the Active Subspaces approach (O’Leary-Roseberry et al., 2022a; O’Leary-Roseberry et al., 2022b). The idea is to reduce the parameter space by looking at its significant contribution to the variation of a state (O’Leary-Roseberry et al., 2022a; O’Leary-Roseberry et al., 2022b). Hence, the input for the machine learning model will be a reduced number of parameters, making the machine learning model as small as possible and reducing the number of machine learning weights and biases to be trained.
- *Hyper-reduction for highly nonlinear problems:* When dealing with highly nonlinear problems, such as those related to chemical equilibria, the use of Proper Orthogonal Decomposition (POD) for spatial and temporal reductions may not be efficient. Inefficient spatial and temporal reductions necessitate a larger number of training samples and a more complex architecture for the machine learning model, resulting in a long offline time. A possible solution is to adopt an Autoencoder (AE) approach as an alternative to POD (Pichi, Moya, and Hesthaven, 2024; Romor, Stabile, and Rozza, 2022; Thuerey et al., 2021; Wu et al., 2021; Lee and Carlberg, 2020). The idea is to train an AE model, consisting of encoding and decoding components, to effectively perform spatial and temporal reductions (Pichi, Moya, and Hesthaven, 2024; Romor, Stabile, and Rozza, 2022; Thuerey et al., 2021; Wu et al., 2021; Lee and Carlberg, 2020). Since training an AE model is computationally expensive, a careful investigation of a problem is needed before employing a non-intrusive reduced-basis (NI-RB) method with AE, e.g. investigating the eigenvalue decay and using principal component analysis (PCA) vs t-Distributed Stochastic Neighbor Embedding (t-SNE) for visualizing the reduced subspaces (Kadeethum et al., 2022; Kadeethum and Yoon, 2022).

From an application perspective, I suggest the following research directions:

- *Parametrization of random fields:* The use of a random field is crucial for describing the spatial heterogeneity of material properties (Mariethoz and Caers, 2014). In this dissertation, I have not yet introduced spatial heterogeneity in material properties in any of the case studies. The challenge lies in parameterizing this random field to enable its incorporation into the non-intrusive reduced-basis (NI-RB) method. As mentioned earlier, incorporating the entire random field as

input parameters into the NI-RB surrogate model is not feasible, as it would significantly expand the number of training samples required to construct the surrogate model. The investigation could compare the effectiveness of the Karhunen-Loeve (KL), Active Subspaces (AS), and Autoencoder (AE) approaches.

- *Accelerating chemical kinetics*: Following the development of model order reduction for dynamical system (Benner, Gugercin, and Willcox, 2015; Baur, Benner, and Feng, 2014), it would be interesting to implement the non-intrusive reduced-basis (NI-RB) method for chemical kinetics calculations. Note that chemical kinetics problems can be thought as dynamical systems. In this case, an efficient approach to perform the reduction needs to be investigated. The use of Proper Orthogonal Decomposition (POD) and Balanced Truncation (BT) could be compared (Benner, Gugercin, and Willcox, 2015; Baur, Benner, and Feng, 2014).
- *Real-world reactive transport problems*: In this dissertation, I implement the non-intrusive reduced-basis (NI-RB) method to benchmark cases for reactive transport simulations. It would be interesting to extend this implementation to a real-world reactive transport problem to further validate its effectiveness.
- *Entropy production for thermo-hydro-mechano-chemical (THMC) process*: Entropy production has been used for understanding thermo-hydro-mechano-chemical (THMC) processes in Solid Earth applications (Siddiqui, Regenauer-Lieb, and Roshan, 2024). It is of interest to further use entropy production as an objective function in global sensitivity analyses (GSA) for THMC system understanding under epistemic uncertainties.

Appendix A

fast: A Python library for the NI-RB implementation

This appendix contains an explanation about how to use my fast Python library for the implementation of the non-intrusive reduced-basis (NI-RB) method. This library is developed as an extension of the codes from Degen et al. (2023a). It will be an open source library and can be downloaded from Github. I will first discuss the structure of the library in Section A.1 and how to install the library in Section A.2. Then it is followed by an explanation for each function within the library in Section A.3 and an example of my chemistry problem in Section A.4.

A.1 Structure of the library

Figure A.1 shows the structure of the fast Python library. The fast Python library consists of two major components: core functions and utilities. The core functions contains all functions needed to implement the NI-RB method, while the utilities contains supporting functions. This structure is used for flexible development in the future. The fast Python library is built using the following external libraries: Tensorflow (Abadi et al., 2015), Numpy (Harris et al., 2020), Scikit-learn (Pedregosa et al., 2011), Scipy (Virtanen et al., 2020), h5py, and athena-mathlab (Romor, Tezzele, and Rozza, 2021).

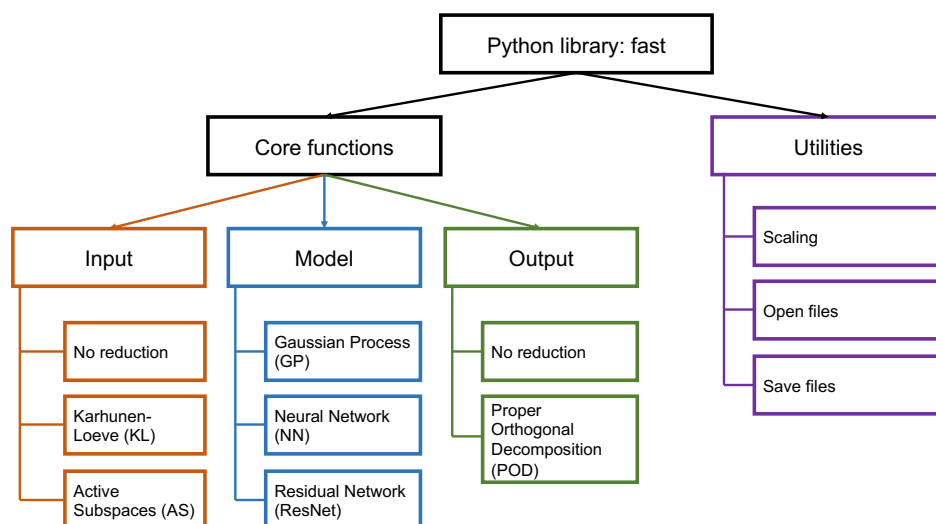


FIGURE A.1: The structure of the fast Python library. This structure is inspired from the code development conducted by Ivagnes, Demo, and Rozza (2023) and O’Leary-Roseberry et al. (2022a).

The organization of the core functions component draws inspiration from the code development conducted by Ivagnes, Demo, and Rozza (2023) and O’Leary-Roseberry et al. (2022a). The core-functions component contains the following functions:

- **Input:** This group of functions is dedicated to performing parameter space reduction. Currently, it comprises two approaches: the *Karhunen-Loeve* (KL) and *Active Subspaces* (AS) functions. The Karhunen-Loeve (KL) approach focuses solely on parameter space variance for reduction, while the Active Subspaces (AS) approach considers state variation resulting from parameter variations, utilizing gradients to inform the reduction process (Romor et al., 2022). Consequently, the computational cost of implementing the AS approach is more expensive than that of the KL approach. If dealing with a high-dimensional parameter space, it is recommended to opt for the AS approach for parameter space reduction (O’Leary-Roseberry et al., 2022b; O’Leary-Roseberry et al., 2022a; Romor et al., 2022). Moreover, users also have the flexibility to skip the parameter reduction step if deemed unnecessary. **In this dissertation, both the AS and the KLE approaches within this group are not deployed as my case studies do not require reducing the parameter space.**
- **Model:** This group comprises various machine learning models employed for computing reduced-basis coefficients. Currently, the available machine learning models include the Gaussian Process (GP), the Neural Network (NN), and the Residual Network (ResNet). The Gaussian Process (GP) is primarily utilized for addressing linear problems (Raissi, Perdikaris, and Karniadakis, 2017), while the Neural Network (NN) and the Residual Network (ResNet) are employed for handling non-linear problems (O’Leary-Roseberry et al., 2022a; O’Leary-Roseberry et al., 2022b; Wang, Hesthaven, and Ray, 2019; Hesthaven and Ubiali, 2018). Consequently, the computational cost associated with training the NN and the ResNet is higher compared to the GP. Additionally, due to its more complex architecture, the ResNet requires more computational resources than the NN. **In this dissertation, I do not use the GP model due to the nonlinear nature of my case studies. Furthermore, the ResNet model is also not used, as the use of a Neural Network model is already sufficient.**
- **Output:** This group of functions aims at performing spatio-temporal reduction for the state. Currently, there is only one function to perform this reduction: Proper Orthogonal Decomposition (POD), following the formulation in Chapter 2. If this function is skipped, the construction of a surrogate model becomes data-driven, see Chapter 2.

The utilities component includes several functions that support scaling operations, file opening, and file saving. I offer two types of scaling: min-max scaling and standard scaling, providing flexibility for different normalization needs. The supported file formats for operations related to files include .h5, .mat, .npz, .txt, and .pickle. These formats are chosen based on their relevance and common usage within the context of my application.

With this structure, anyone can contribute to implementing more efficient and accurate approaches for parameter space reduction, scaling, spatio-temporal state reduction, or training machine learning models based on the defined groups.

A.2 Installations

I provide a readme file in my later Github, explaining how to install the library. Currently, the easiest way to install is through conda. The installation steps are the following:

1. Install anaconda in your machine, see <https://docs.anaconda.com/free/anaconda/install/index.html>
2. Once conda environment installed, install devenv: `conda install devenv`
3. Clone fast
4. Go to fast folder
5. Enter in your terminal: `conda devenv`. This process will create a new environment called (fast)
6. Activate fast environment: `conda activate fast`
7. Build fast library: `python setup.py bdist_wheel`
8. Install build: `pip install -e .`
9. Enjoy the library

A.3 Functions

The fast Python library offers great flexibility in combining various approaches in constructing an NI-RB surrogate model. The structures outlined in Figure A.2 are crucial components that should exist during the construction of a surrogate model.

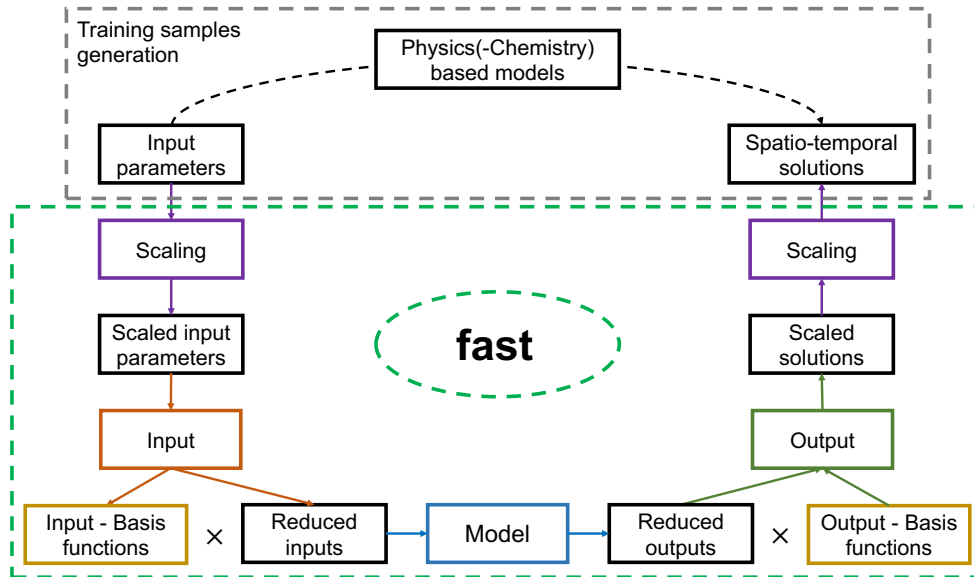


FIGURE A.2: General workflow for constructing a surrogate model with the fast Python library. This workflow is inspired by the works of Ivagnes, Demo, and Rozza (2023) and O’Leary-Roseberry et al. (2022a).

The scaling group of functions (denoted in the purple box in Figure A.2) aims to normalize the input parameters or states. The detailed functions are as follows:

- `fast.utilities.scale(X).minmax()`: this function applies minmax scaling to a given input array $\mathbf{X}$. The array $\mathbf{X}$ can be an input-parameter array with the size $N_{tr} \times N_p$ or snapshots with size $N_{tr} \times N_x$ or spatio-temporal snapshots $N_{tr} \times N_t \times N_x$. The maximum and minimum values used in the scaling are obtained directly from the array $\mathbf{X}$.
- `fast.utilities.scale(X).customized_minmax(y_lower,y_upper)`: this function applies minmax scaling to a given input array $\mathbf{X}$. The array $\mathbf{X}$ can be an input-parameter array with the size $N_{tr} \times N_p$ or snapshots with size $N_{tr} \times N_x$ or spatio-temporal snapshots $N_{tr} \times N_t \times N_x$. The maximum values `y_upper` and minimum values `y_lower` used in the scaling are user-defined.
- `fast.utilities.scale(X).standard()`: this function applies standard scaling to a given input array $\mathbf{X}$. The array $\mathbf{X}$ can be an input-parameter array with the size $N_{tr} \times N_p$ or snapshots with size $N_{tr} \times N_x$ or spatio-temporal snapshots $N_{tr} \times N_t \times N_x$. The mean and variance values used in the scaling are obtained directly from the array $\mathbf{X}$.
- `fast.utilities.scale(X).customized_standard(y_mean,y_var)`: this function applies standard scaling to a given input array $\mathbf{X}$. The array $\mathbf{X}$ can be an input-parameter array with the size $N_{tr} \times N_p$ or snapshots with size $N_{tr} \times N_x$ or spatio-temporal snapshots $N_{tr} \times N_t \times N_x$. The mean values `y_mean` and variance values `y_var` used in the scaling are user-defined.
- `fast.utilities.scale(X).euclidean(x_lower,x_upper,y_lower,y_upper)`: this function applies linear scaling to a given input array $\mathbf{X}$. The array $\mathbf{X}$ can be an input-parameter array with the size $N_{tr} \times N_p$ or snapshots with size $N_{tr} \times N_x$ or spatio-temporal snapshots $N_{tr} \times N_t \times N_x$. The parameters `x_lower` and `x_upper` correspond to the user-defined minimum and maximum values in the array $\mathbf{X}$, respectively, while `y_lower` and `y_upper` correspond to the expected minimum and maximum scaled values, respectively.

The input group of functions (denoted in the orange box in Figure A.2) aims to reduce the parameter space. The detailed functions are as follows:

- `fast.input.AS(X, Y, cut_off)`: this function uses the Active Subspaces approach to reduce the parameter space. Users need to enter the input-parameter array $\mathbf{X}$ with the size of $N_{tr} \times N_p$, the scaled state $\mathbf{Y}$ with the $N_{tr} \times N_x$ or $N_{tr} \times N_t \times N_x$, and the size of reduced parameters (`cut_off`). The outputs of this function are the reduced $\mathbf{X}$ and the parameter-space basis functions.
- `fast.input.KLE(X, cut_off)`: this function uses the Karhunen-Loeve approach to reduce the parameter space. The users need to enter the input-parameter array $\mathbf{X}$ with the size of $N_{tr} \times N_p$, and error tolerance (`cut_off`). The outputs of this function are the reduced $\mathbf{X}$ and the parameter-space basis functions.

The model group of functions (denoted in the blue box in Figure A.2) contains the machine learning model used for calculating the reduced-basis coefficients. The detailed functions are as follows:

- `fast.model.gpr(X, Y, length_scale, file_name, is_save).train()`: this function contains a Gaussian Process (GP) model, built based on Scikit-learn library. Users need to provide the reduced and scaled input parameters $\mathbf{X}$, the scaled and reduced state $\mathbf{Y}$, the GP's hyperparameters: the `length_scale`. The "length_scale"

requires a list of values for each scale with the size of N_p . The "file_name" parameter requires a string input for the name of a saved trained GP model if "is_save" is activated. The "is_save" parameter is a boolean type. The outputs of this function are the trained model and the summary of hyperparameters.

- `fast.model.neural_network(X, Y, neuron_in_l1, neuron_in_l2, epochs, learning_rates, batches, activation, loss_function, verbose, file_name, is_save).train()`: this function contains a Neural Network (NN) model, built based on Tensorflow library. Users need to provide the reduced scaled input parameters X , the scaled and reduced state Y , the NN's hyperparameters: the number of neurons in layer 1 (`neuron_in_l1`), the number of neurons in layer 2 (`neuron_in_l2`), number of epochs (`epochs`), learning rate (`learning_rates`), batch size (`batches`), activation functions (`activation`), and loss function (`loss_function`). The "activation" parameter requires a string input, such as 'sigmoid', 'relu', following the structure in Tensorflow library. The "loss_function" parameter requires a string input, such as 'mae', 'mse', following the structure in Tensorflow library. The "verbose" parameter is a boolean parameter to show the training process. The "file_name" parameter requires a string input for the name of a saved trained NN model if "is_save" is activated. The "is_save" parameter is a boolean type. The outputs of this function are the trained model and the summary of hyperparameters.
- `fast.model.resnet(X, Y, neuron_in_l1, neuron_in_l2, epochs, learning_rates, batches, activation, loss_function, verbose, file_name, is_save).train()`: this function contains a Residual Network (ResNet) model, built based on Tensorflow library. The structure of this function is similar to the `neural_network` function. Users need to provide the reduced scaled input parameters X , the scaled and reduced state Y , the NN's hyperparameters: the number of neurons in layer 1 (`neuron_in_l1`), the number of neurons in layer 2 (`neuron_in_l2`), number of epochs (`epochs`), learning rate (`learning_rates`), batch size (`batches`), activation functions (`activation`), and loss function (`loss_function`). The "activation" parameter requires a string input, such as 'sigmoid', 'relu', following the structure in Tensorflow library. The "loss_function" parameter requires a string input, such as 'mae', 'mse', following the structure in Tensorflow library. The "verbose" parameter is a boolean parameter for showing the training process. The "file_name" parameter requires a string input for the name of a saved trained ResNet model if "is_save" is activated. The "is_save" parameter is a boolean type. The outputs of this function are the trained model and the summary of hyperparameters.

The output group of functions (denoted in the green box in Figure A.2) performs a reduction for the spatio-temporal snapshots. The detailed functions are as follows:

- `fast.output.POD(Y, parameter_accuracy, time_accuracy, is_spatiotemporal).get_basis(get_spectra)`: this function is dedicated for performing reduction for the spatio-temporal snapshots Y using the Proper Orthogonal Decomposition (POD) approach. Users need to provide the snapshots Y , the error tolerance for parameter trajectory (`parameter_accuracy`), and the error tolerance for time trajectory (`time_accuracy`) if we have spatio-temporal snapshots and `is_spatiotemporal` is activated. The "get_spectra" parameter is a boolean type that confirms whether the eigenvalue spectra are shown as the output of this function or not. The primary output of this function is the basis functions for the state. Within this function, an additional function used for displaying the projection errors: `fast.output.POD(Y, parameter_accuracy, time_accuracy, is_spatiotemporal).check_error(Y_val, scaling, is_minmax)`. The extra input needed from users is the scaled validation set

Y_{val} , the scaling factor (scaling). If the scaling is minmax, users should provide a list of minimum and maximum values with the size of N_x for a steady-state problem or $N_t \times N_x$ for a transient problem, and activate the "is_minmax" parameter. The "is_minmax" parameter is a boolean type.

There are several supporting functions used for opening and saving files. The functions are the following:

- `fast.utilities.open(file_name, keywords)`: this function is used for opening a file given the path of the file and keywords to open the file. The supported formats are .h5 opened by adding `.from_h5()`, .mat opened by adding `.from_mat()`, .txt opened by adding `.from_text()`, .npz opened by adding `.from_npy()`, and .pickle opened by adding `.from_pickle()`.
- `fast.utilities.save(file, file_name, keywords)`: this function is used for saving a file given the array (file), the filename and the keywords to save the file. The supported formats are .h5 opened by adding `.as_h5()`, .mat opened by adding `.as_mat()`, .txt opened by adding `.as_text()`, .npz opened by adding `.as_npy()`, and .pickle opened by adding `.as_pickle()`.

In the future, I will allow researchers from around the world to contribute to the advancement of this library by following the structure in Figure A.1 and A.2 to add new functionalities.

A.4 Example

I show here an example using my chemical equilibrium case study. Note that a further improvement in the Neural Network model hyperparameters is needed to achieve the same accuracy as that of Chapter 8. The objective of using this example is to demonstrate how the *fast* Python library works.

In this example, an array of input parameters used for training X_{train} with the size of $N_{tr} \times N_p$, an array of input parameters used for validation X_{test} with the size of $N_{ts} \times N_p$, an array of snapshots used for training Y_{train} with the size of $N_{tr} \times N_x$, and an array of snapshots used for validation Y_{test} with the size of $N_{ts} \times N_x$ are given. I save these files in a folder named "data". The steps to construct an NI-RB surrogate model using the *fast* Python library are the following:

1. Import the fast Python library

```
1 from fast.utilities import *
2 from fast.output import *
3 from fast.model import neural_network
```

2. Open the input-parameter arrays and snapshot arrays using fast.utilities

```
1 #Path
2 Xtrain_file = "data/Xtrain.mat"
3 Xval_file   = "data/Xval.mat"
4 Ytrain_file = "data/Ytrain.h5"
5 Yval_file   = "data/Yval.h5"
6
7 #Open Xtrain, Xval, Ytrain, Yval with fast library
8 Xtrain_unscaled = open(Xtrain_file, "X_real").from_mat()
9 Xval_unscaled   = open(Xval_file, "X_real").from_mat()
10 Ytrain_unscaled = open(Ytrain_file, "transformed_state").from_h5()
11 Yval_unscaled   = open(Yval_file, "transformed_state").from_h5()
```

3. Scale the snapshots using fast.utilities

```

1 #Step 1: before applying POD to physical states , we need to scale
   them --> result: scaled solutions
2 #The training set should be scaled using min max approach
3 Ytrain_scaled, output_scaling_factor = scale(Ytrain_unscaled,
   is_spatiotemporal = False).minmax()
4
5 #The validation set should be scaled using min max approach , with
   scaling factor from training set (row 0: lower bound, row 1: upper
   bound)
6 Yval_scaled = scale(Yval_unscaled).customized_minmax(
   output_scaling_factor[0,:], output_scaling_factor[1,:])

```

4. Apply the POD approach to the scaled snapshots using fast.output.POD

```

1 #Step 2: applying POD on training set only --> result: basis
2 #The POD uses scaled training set; we obtain basis functions and
   spectra decay
3 basis, spectra = POD(Ytrain_scaled, threshold_params, time_accuracy =
   None, is_spatiotemporal = False).get_basis(get_spectra = True)
4
5 #We need to check the projection error using validation set; enter
   unscaled solution + scaling factor in array format where row 0:
   lower bound and row 1: upper bound
6 L1_error, L2_error = POD(Ytrain_scaled, threshold_params,
   time_accuracy = None, is_spatiotemporal = False).check_error(
   Yval_unscaled, output_scaling_factor, is_minmax = True)

```

5. Scale the input parameters using fast.utilities. In this example, I do not apply reduction to the parameter space since it is not high dimensional.

```

1 #Step 3: scaling input for machine learning construction --> result:
   scaled input
2 #The input is scaled using standard scaling
3 Xtrain_scaled, input_scaling_factor = scale(Xtrain_unscaled,
   is_spatiotemporal = False).standard()
4
5 #The input validation set is scaled using standard scaling , with
   scaling factor from training set
6 Xval_scaled = scale(Xval_unscaled).customized_standard(
   input_scaling_factor[0,:], input_scaling_factor[1,:])

```

6. Generate reduced-basis coefficients for training the machine learning model with the help of numpy library

```

1 import numpy
2
3 #Step 4: generating alpha for machine learning construction -->
   result: alpha training
4 #Remember, in NI-RB, the machine learning model is trained using
   reduced-basis coefficients
5 alpha_train = numpy.matmul(basis, Ytrain_scaled.T).T

```

7. Setup and train the Neural Network model using fast.model.neural_network

```

1 #Step 5: training machine learning --> trained ML
2 #Remember based on universal approximation theorem, NN model with 2
   layers can approximate most functions
3 network, network_params = neural_network(
4     inputs = Xtrain_scaled,

```

```

5     outputs      = alpha_train ,
6     neuron_in_l1 = 91,
7     neuron_in_l2 = 56,
8     epochs       = 35446,
9     learning_rates = 0.000915000600168507,
10    batchs       = 4062,
11    activation    = 'sigmoid',
12    loss_function = 'mae',
13    verbose      = 0,
14    file_name     = None,
15    is_save      = False
16 ).train()

```

8. Predict the reduced-basis coefficients of the test case using the trained machine learning model

```

1 #Step 6: prediction --> alpha_val
2 #The network variable inherits tf structure, so it can use predict
  function from tf
3 alpha_val = network(Xval_scaled, training=False)

```

9. Calculate the full solution of the test case with the help of numpy library

```

1 #Step 7: projection back --> predicting Yval
2 Yval_predict_scaled = numpy.dot(alpha_val, basis)
3
4 Yval_predict_unscaled = scale(Yval_predict_scaled).euclidean(numpy.
  zeros( (1,Yval_predict_scaled.shape[1]) ).flatten(), numpy.ones(
  (1,Yval_predict_scaled.shape[1]) ).flatten(),
  output_scaling_factor[0,:], output_scaling_factor[1,:])

```

10. Calculate the approximation error using Scikit-learn library

```

1 #Step 8: check approximation quality
2 L2_approximation_error = mean_squared_error(Yval_unscaled,
  Yval_predict_unscaled, multioutput='uniform_average', squared=
  False)
3 L1_approximation_error = mean_absolute_error(Yval_unscaled,
  Yval_predict_unscaled, multioutput='uniform_average')
4 print('L1-approximation_error: ' + str(L1_approximation_error))
5 print('L2-approximation_error: ' + str(L2_approximation_error))

```


Bibliography

- Abadi, Martín et al. (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. URL: <https://www.tensorflow.org/>.
- Amini, Danial, Ehsan Haghighat, and Ruben Juanes (2022). “Physics-Informed Neural Network Solution of Thermo–Hydro–Mechanical Processes in Porous Media”. In: *Journal of Engineering Mechanics* 148.11, p. 04022070.
- (2023). “Inverse modeling of nonisothermal multiphase poromechanics using physics-informed neural networks”. In: *Journal of Computational Physics* 490, p. 112323.
- Antoulas, Athanasios C (2005). *Approximation of large-scale dynamical systems*. SIAM.
- Arnold, Dan et al. (2016). “Optimisation of decision making under uncertainty throughout field lifetime: A fractured reservoir example”. In: *Computers & Geosciences* 95, pp. 123–139.
- Aster, Richard C, Brian Borchers, and Clifford H Thurber (2018). *Parameter estimation and inverse problems*. Elsevier.
- Atef, H et al. (2016). “The contribution of gravity method in geothermal exploration of southern part of the Gulf of Suez–Sinai region, Egypt”. In: *NRIAG Journal of Astronomy and Geophysics* 5.1, pp. 173–185.
- Bartels, Jörn, Michael Kühn, and Christoph Clauser (2003). “Numerical simulation of reactive flow using SHEMAT”. In: *Numerical Simulation of Reactive Flow in Hot Aquifers: SHEMAT and Processing SHEMAT*, pp. 5–74.
- Bauer, Peter et al. (2021). “The digital revolution of Earth-system science”. In: *Nature Computational Science* 1.2, pp. 104–113.
- Baur, Ulrike, Peter Benner, and Lihong Feng (2014). “Model order reduction for linear and nonlinear systems: a system-theoretic perspective”. In: *Archives of Computational Methods in Engineering* 21.4, pp. 331–358.
- Bejan, Adrian (2013). *Convection heat transfer*. John Wiley & sons.
- Benner, Peter, Serkan Gugercin, and Karen Willcox (2015). “A survey of projection-based model reduction methods for parametric dynamical systems”. In: *SIAM review* 57.4, pp. 483–531.
- Bergen, Karianne J et al. (2019). “Machine learning for data-driven discovery in solid Earth geoscience”. In: *Science* 363.6433, eaau0323.
- Berkowitz, Brian et al. (2016). “Measurements and models of reactive transport in geological media”. In: *Reviews of Geophysics* 54.4, pp. 930–986.
- Bethke, Craig M (2022). *Geochemical and biogeochemical reaction modeling*. Cambridge university press.
- Beucler, Tom et al. (2021). “Enforcing analytic constraints in neural networks emulating physical systems”. In: *Physical Review Letters* 126.9, p. 098302.
- Beyer, Christof, Thomas Nagel, and Haibing Shao (2014). “Reactive Transport”. In: *Thermo-Hydro-Mechanical-Chemical Processes in Fractured Porous Media: Modelling and Benchmarking: Closed-Form Solutions*. Springer, pp. 255–275.
- Birdsell, Daniel T, Benjamin M Adams, and Martin O Saar (2021). “Minimum transmissivity and optimal well spacing and flow rate for high-temperature aquifer thermal energy storage”. In: *Applied Energy* 289, p. 116658.

- Birdsell, Daniel T et al. (2024). “Analytical solutions to evaluate the geothermal energy generation potential from sedimentary-basin reservoirs”. In: *Geothermics* 116, p. 102843.
- Bishop, Christopher M and Nasser M Nasrabadi (2006). *Pattern recognition and machine learning*. Vol. 4. 4. Springer.
- Bogoclu, Can, Dirk Roos, and Tamara Nestorović (2021). “Local Latin hypercube refinement for multi-objective design uncertainty optimization”. In: *Applied Soft Computing* 112, p. 107807.
- Bonté, Damien, J-D Van Wees, and JM Verweij (2012). “Subsurface temperature of the onshore Netherlands: new temperature dataset and modelling”. In: *Netherlands Journal of Geosciences* 91.4, pp. 491–515.
- Borregales, Manuel et al. (2019). “A partially parallel-in-time fixed-stress splitting method for Biot’s consolidation model”. In: *Computers & Mathematics with Applications* 77.6, pp. 1466–1478.
- Börising, Nele et al. (2017). “Entropy production in a box: analysis of instabilities in confined hydrothermal systems”. In: *Water Resources Research* 53.9, pp. 7716–7739.
- Box, George EP (1979). “Robustness in the strategy of scientific model building”. In: *Robustness in statistics*. Elsevier, pp. 201–236.
- Brezzi, Franco and Michel Fortin (2012). *Mixed and hybrid finite element methods*. Vol. 15. Springer Science & Business Media.
- Brooks, Stephen P, Edward A Catchpole, and Byron JT Morgan (2000). “Bayesian animal survival estimation”. In: *Statistical Science* 15.4, pp. 357–376.
- Brouwer, Dirk Roelof and J-D Jansen (2004). “Dynamic optimization of waterflooding with smart wells using optimal control theory”. In: *SPE journal* 9.04, pp. 391–402.
- Cacace, Mauro and Antoine B Jacquey (2017). “Flexible parallel implicit modelling of coupled thermal–hydraulic–mechanical processes in fractured rocks”. In: *Solid Earth* 8.5, pp. 921–941.
- Caers, Jef (2011). *Modeling uncertainty in the earth sciences*. John Wiley & Sons.
- Cai, Shengze et al. (2021). “Artificial intelligence velocimetry and microaneurysm-on-a-chip for three-dimensional analysis of blood flow in physiology and disease”. In: *Proceedings of the National Academy of Sciences* 118.13, e2100697118.
- Cao, Lianghao et al. (2023). “Bayesian model calibration for diblock copolymer thin film self-assembly using power spectrum of microscopy data and machine learning surrogate”. In: *Computer Methods in Applied Mechanics and Engineering* 417, p. 116349.
- Chaudhuri, Anirban et al. (2022). “Certifiable risk-based engineering design optimization”. In: *AIAA Journal* 60.2, pp. 551–565.
- Chen, Chaohui, Gaoming Li, and AC C Reynolds (2012). “Robust constrained optimization of short-and long-term net present value for closed-loop reservoir management”. In: *SPE Journal* 17.03, pp. 849–864.
- Chen, Yuyao et al. (2020). “Physics-informed neural networks for inverse problems in nano-optics and metamaterials”. In: *Optics express* 28.8, pp. 11618–11633.
- Chen, Zhao et al. (2018). “Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks”. In: *International conference on machine learning*. PMLR, pp. 794–803.
- Chuang, Pi-Yueh and Lorena A. Barba (2022). *Experience report of physics-informed neural networks in fluid simulations: pitfalls and frustration*. DOI: 10.48550/ARXIV.2205.14249. URL: <https://arxiv.org/abs/2205.14249>.
- Chuang, Pi-Yueh and Lorena A Barba (2023). “Predictive Limitations of Physics-Informed Neural Networks in Vortex Shedding”. In: *arXiv preprint arXiv:2306.00230*.

- Cicci, Ludovica, Stefania Fresca, and Andrea Manzoni (2022). “Deep-HyROMnet: A deep learning-based operator approximation for hyper-reduction of nonlinear parametrized PDEs”. In: *Journal of Scientific Computing* 93.2, p. 57.
- Collard, Nathalie et al. (2023). “A neural-network-based hydrothermal dolomitization model for future coupling with basin modeling in the energy transition context”. In: *Goldschmidt 2023 Conference*. GOLDSCHMIDT.
- Constantine, Paul G (2015). *Active subspaces: Emerging ideas for dimension reduction in parameter studies*. SIAM.
- De Lucia, Marco and Michael Kühn (2021). “DecTree v1. 0—chemistry speedup in reactive transport simulations: purely data-driven and physics-based surrogates”. In: *Geoscientific Model Development Discussions* 2021, pp. 1–26.
- Deb, Kalyanmoy et al. (2002). “A fast and elitist multiobjective genetic algorithm: NSGA-II”. In: *IEEE transactions on evolutionary computation* 6.2, pp. 182–197.
- Degen, Denise and Mauro Cacace (2021). “Effects of transient processes for thermal simulations of the Central European Basin”. In: *Geoscientific Model Development* 14.3, pp. 1699–1719.
- Degen, Denise, Mauro Cacace, and Florian Wellmann (2022). “3D multi-physics uncertainty quantification using physics-based machine learning”. In: *Scientific Reports* 12.1, p. 17491.
- Degen, Denise, Karen Veroy, and Florian Wellmann (2020). “Certified reduced basis method in geosciences: Addressing the challenge of high-dimensional problems”. In: *Computational Geosciences* 24, pp. 241–259.
- (2022). “Uncertainty quantification for basin-scale geothermal conduction models”. In: *Scientific Reports* 12.1, p. 4246.
- Degen, Denise et al. (2021a). “Global sensitivity analysis to optimize basin-scale conductive model calibration—A case study from the Upper Rhine Graben”. In: *Geothermics* 95, p. 102143.
- Degen, Denise et al. (2021b). “How biased are our models?—a case study of the alpine region”. In: *Geoscientific Model Development* 14.11, pp. 7133–7153.
- Degen, Denise et al. (2022). “Crustal-scale thermal models: Revisiting the influence of deep boundary conditions”. In: *Environmental Earth Sciences* 81.3, p. 88.
- Degen, Denise et al. (Sept. 2023a). *Non-Intrusive Reduced Basis Code for Geothermal, Geodynamic, and Hydrology Benchmarks*. DOI: 10.5281/zenodo.8369108. URL: <https://doi.org/10.5281/zenodo.8369108>.
- Degen, Denise et al. (2023b). “Perspectives of Physics-Based Machine Learning for Geoscientific Applications Governed by Partial Differential Equations”. In: *Geoscientific Model Development Discussions* 2023, pp. 1–50.
- Di Paolo, Federico et al. (2020). “La Palma island (Spain) geothermal system revealed by 3D magnetotelluric data inversion”. In: *Scientific reports* 10.1, p. 18181.
- Discacciati, Niccolò and Jan S Hesthaven (2024). “Model reduction of coupled systems based on non-intrusive approximations of the boundary response maps”. In: *Computer Methods in Applied Mechanics and Engineering* 420, p. 116770.
- Elsheikh, Ahmed H, Mary F Wheeler, and Ibrahim Hoteit (2013). “Nested sampling algorithm for subsurface flow model selection, uncertainty quantification, and non-linear calibration”. In: *Water Resources Research* 49.12, pp. 8383–8399.
- Esposito, AM et al. (2006). “Automatic discrimination among landslide, explosion-quake, and microtremor seismic signals at Stromboli volcano using neural networks”. In: *Bulletin of the Seismological Society of America* 96.4A, pp. 1230–1240.
- Falkner, Stefan, Aaron Klein, and Frank Hutter (2018). “BOHB: Robust and efficient hyperparameter optimization at scale”. In: *International Conference on Machine Learning*. PMLR, pp. 1437–1446.

- Faroughi, Salah A et al. (2022). “Physics-guided, physics-informed, and physics-encoded neural networks in scientific computing”. In: *arXiv preprint arXiv:2211.07377*.
- Fotherby, Angus et al. (2022). “An emulation-based approach for interrogating reactive transport models”. In: *EGUsphere*, pp. 1–36.
- Franco, Alessandro and Maurizio Vaccaro (2014). “Numerical simulation of geothermal reservoirs for the sustainable design of energy plants: A review”. In: *Renewable and Sustainable Energy Reviews* 30, pp. 987–1002.
- Fu, Yangyang et al. (2023). “Exploring the effects of training samples on the accuracy of crop mapping with machine learning algorithm”. In: *Science of Remote Sensing* 7, p. 100081.
- Gábor, Attila and Julio R Banga (2015). “Robust and efficient parameter estimation in dynamic models of biological systems”. In: *BMC systems biology* 9.1, pp. 1–25.
- Geweke, John (1991). *Evaluating the accuracy of sampling-based approaches to the calculation of posterior moments*. Tech. rep. Federal Reserve Bank of Minneapolis.
- Glen, Graham and Kristin Isaacs (2012). “Estimating Sobol sensitivity indices using correlations”. In: *Environmental Modelling & Software* 37, pp. 157–166.
- Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016). *Deep learning*. MIT press.
- Gudmundsdottir, Halldora and Roland N Horne (2020). “Prediction modeling for geothermal reservoirs using deep learning”. In: *45th workshop on geothermal reservoir engineering*. Stanford, California: Stanford University, pp. 1–12.
- Guglielmetti, Luca and Andrea Moscariello (2021). “On the use of gravity data in delineating geologic features of interest for geothermal exploration in the Geneva Basin (Switzerland): prospects and limitations”. In: *Swiss Journal of Geosciences* 114.1, pp. 1–20.
- Guo, Mengwu and Jan S Hesthaven (2019). “Data-driven reduced order modeling for time-dependent problems”. In: *Computer methods in applied mechanics and engineering* 345, pp. 75–99.
- Haario, Heikki, Eero Saksman, and Johanna Tamminen (2001). “An adaptive Metropolis algorithm”. In: *Bernoulli*, pp. 223–242.
- Haghighat, Ehsan, Danial Amini, and Ruben Juanes (2021). “Physics-informed neural network simulation of multiphase poroelasticity using stress-split sequential training”. In: *arXiv preprint arXiv:2110.03049*.
- (2022). “Physics-informed neural network simulation of multiphase poroelasticity using stress-split sequential training”. In: *Computer Methods in Applied Mechanics and Engineering* 397, p. 115141.
- Haghighat, Ehsan and Ruben Juanes (2021). “SciANN: A Keras/TensorFlow wrapper for scientific computations and physics-informed deep learning using artificial neural networks”. In: *Computer Methods in Applied Mechanics and Engineering* 373, p. 113552.
- Haghighat, Ehsan et al. (2021). “A nonlocal physics-informed deep learning framework using the peridynamic differential operator”. In: *Computer Methods in Applied Mechanics and Engineering* 385, p. 114012.
- Harris, Charles R. et al. (Sept. 2020). “Array programming with NumPy”. In: *Nature* 585.7825, pp. 357–362. DOI: 10.1038/s41586-020-2649-2. URL: <https://doi.org/10.1038/s41586-020-2649-2>.
- He, QiZhi et al. (2020). “Physics-informed neural networks for multiphysics data assimilation with application to subsurface transport”. In: *Advances in Water Resources* 141, p. 103610.
- Herman, Jon and Will Usher (2017). “SALib: An open-source Python library for Sensitivity Analysis”. In: *The Journal of Open Source Software* 2.9. DOI: 10.21105/joss.00097. URL: <https://doi.org/10.21105/joss.00097>.

- Hesthaven, Jan, Cecilia Pagliantini, and Nicolò Ripamonti (2023). “Adaptive symplectic model order reduction of parametric particle-based Vlasov–Poisson equation”. In: *Mathematics of Computation*.
- Hesthaven, Jan S, Gianluigi Rozza, Benjamin Stamm, et al. (2016). *Certified reduced basis methods for parametrized partial differential equations*. Vol. 590. Springer.
- Hesthaven, Jan S and Stefano Ubbiali (2018). “Non-intrusive reduced order modeling of nonlinear problems using neural networks”. In: *Journal of Computational Physics* 363, pp. 55–78.
- Huang, Po-Wei and Florian Wellmann (2021). “An explanation to the Nusselt–Rayleigh discrepancy in naturally convected porous media”. In: *Transport in Porous Media* 137, pp. 195–214.
- Iglesias, Marco and Andrew M Stuart (2014). “Inverse problems and uncertainty quantification”. In: *SIAM News* 20, pp. 2–3.
- Iman, Ronald L and WJ Conover (1980). “Small sample sensitivity analysis techniques for computer models. with an application to risk assessment”. In: *Communications in statistics-theory and methods* 9.17, pp. 1749–1842.
- Ivagnes, Anna, Nicola Demo, and Gianluigi Rozza (2023). “Towards a machine learning pipeline in reduced order modelling for inverse problems: neural networks for boundary parametrization, dimensionality reduction and solution manifold approximation”. In: *Journal of Scientific Computing* 95.1, p. 23.
- Jagtap, NV, Maruti K Mudunuru, and KB Nakshatrala (2021). “A deep learning modeling framework to capture mixing patterns in reactive-transport systems”. In: *arXiv preprint arXiv:2101.04227*.
- Jatnieks, Janis et al. (2016). “Data-driven surrogate model approach for improving the performance of reactive transport simulations”. In: *Energy Procedia* 97, pp. 447–453.
- Jha, Birendra and Ruben Juanes (2014). “Coupled multiphase flow and poromechanics: A computational model of pore pressure effects on fault slip and earthquake triggering”. In: *Water Resources Research* 50.5, pp. 3776–3808.
- Jolie, Egbert et al. (2021). “Geological controls on geothermal resources for power generation”. In: *Nature Reviews Earth & Environment* 2.5, pp. 324–339.
- Juliussøn, Egill and Sigurdur Bjornsson (2021). “Optimizing production strategies for geothermal resources”. In: *Geothermics* 94, p. 102091.
- Kadeethum, Teeratorn and Hongkyu Yoon (2022). *Reduced order modeling with Barlow Twins self-supervised learning: Navigating the space between linear and nonlinear solution manifolds*. Tech. rep. Sandia National Lab.(SNL-NM), Albuquerque, NM (United States).
- Kadeethum, Teeratorn et al. (2022). “Non-intrusive reduced order modeling of natural convection in porous media using convolutional autoencoders: Comparison with linear subspace techniques”. In: *Advances in Water Resources* 160, p. 104098.
- Kapteyn, Michael G et al. (2022). “Data-driven physics-based digital twins via a library of component-based reduced-order models”. In: *International Journal for Numerical Methods in Engineering* 123.13, pp. 2986–3003.
- Karakaşlı, G. (2021). *boh-b-hpo*. <https://github.com/goktug97/boh-b-hp>.
- Karniadakis, George Em et al. (2021). “Physics-informed machine learning”. In: *Nature Reviews Physics* 3.6, pp. 422–440.
- Kern, Raymond and Alain Weisbrod (1967). “Thermodynamics for geologists”. In: *Current Directions in Biomedical Engineering* 6.3, pp. 107–110.

- Kirkner, David J and Howard Reeves (1988). "Multicomponent mass transport with homogeneous and heterogeneous chemical reactions: effect of the chemistry on the choice of numerical algorithm: 1. Theory". In: *Water Resources Research* 24.10, pp. 1719–1729.
- Kolditz, Olaf et al. (2012). "OpenGeoSys: an open-source initiative for numerical simulation of thermo-hydro-mechanical/chemical (THM/C) processes in porous media". In: *Environmental Earth Sciences* 67, pp. 589–599.
- Kolditz, Olaf et al. (2016). *Thermo-hydro-mechanical chemical processes in fractured porous media: modelling and benchmarking*. Vol. 25. Springer.
- Kriesel, David (2007). *A Brief Introduction to Neural Networks*. URL: available at <http://www.dkriesel.com>.
- Krishnapriyan, Aditi et al. (2021). "Characterizing possible failure modes in physics-informed neural networks". In: *Advances in Neural Information Processing Systems* 34.
- Kuhn, Stephen, Matthew J Cracknell, and Anya M Reading (2018). "Lithologic mapping using Random Forests applied to geophysical and remote-sensing data: A demonstration study from the Eastern Goldfields of Australia". In: *Geophysics* 83.4, B183–B193.
- Kyas, Svetlana et al. (2022). "Accelerated reactive transport simulations in heterogeneous porous media using Reaktoro and Firedrake". In: *Computational Geosciences* 26.2, pp. 295–327.
- Laloy, Eric and Diederik Jacques (2019). "Emulation of CPU-demanding reactive transport models: a comparison of Gaussian processes, polynomial chaos expansion, and deep neural networks". In: *Computational Geosciences* 23, pp. 1193–1215.
- (2022). "Speeding up reactive transport simulations in cement systems by surrogate geochemical modeling: deep neural networks and k-nearest neighbors". In: *Transport in Porous Media* 143.2, pp. 433–462.
- Lapeyre, Bernard (2007). "Introduction to Monte-Carlo Methods". In: *Lecture, Halmstad, Sweden*, pp. 2–4.
- Latrach, Abdeldjalil et al. (2023). "A Critical Review of Physics-Informed Machine Learning Applications in Subsurface Energy Systems". In: *arXiv preprint arXiv:2308.04457*.
- Leal, Allan MM, Martin J Blunt, and Tara C LaForce (2014). "Efficient chemical equilibrium calculations for geochemical speciation and reactive transport modelling". In: *Geochimica et Cosmochimica Acta* 131, pp. 301–322.
- (2015). "A chemical kinetics algorithm for geochemical modelling". In: *Applied Geochemistry* 55, pp. 46–61.
- Leal, Allan MM, Dmitrii A Kulik, and Georg Kosakowski (2016). "Computational methods for reactive transport modeling: A Gibbs energy minimization approach for multiphase equilibrium calculations". In: *Advances in Water Resources* 88, pp. 231–240.
- Leal, Allan MM and William R Smith (2022). "Inverse chemical equilibrium problems: General formulation and algorithm". In: *Chemical Engineering Science* 252, p. 117162.
- Leal, Allan MM et al. (2017). "An overview of computational methods for chemical equilibrium and kinetic calculations for geochemical and reactive transport modeling". In: *Pure and Applied Chemistry* 89.5, pp. 597–643.
- Leal, Allan MM et al. (2020). "Accelerating reactive transport modeling: on-demand machine learning algorithm for chemical equilibrium calculations". In: *Transport in Porous Media* 133.2, pp. 161–204.
- Leal, AMM (2015). *Reaktoro: An open-source unified framework for modeling chemically reactive systems*.

- LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton (2015). “Deep learning”. In: *nature* 521.7553, pp. 436–444.
- Lee, Kookjin and Kevin T Carlberg (2020). “Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders”. In: *Journal of Computational Physics* 404, p. 108973.
- Li, Zongyi et al. (2020). “Fourier neural operator for parametric partial differential equations”. In: *arXiv preprint arXiv:2010.08895*.
- Lichtner, Peter C (1985). “Continuum model for simultaneous chemical reactions and mass transport in hydrothermal systems”. In: *Geochimica et Cosmochimica Acta* 49.3, pp. 779–800.
- Linka, Kevin et al. (2022). “Bayesian Physics Informed Neural Networks for real-world nonlinear dynamical systems”. In: *Computer Methods in Applied Mechanics and Engineering* 402, p. 115346.
- Luo, Dingcheng et al. (2023). “Efficient PDE-constrained optimization under high-dimensional uncertainty using derivative-informed neural operators”. In: *arXiv preprint arXiv:2305.20053*.
- Luo, Dingcheng et al. (2024). “SOUPy: Stochastic PDE-constrained optimization under high-dimensional uncertainty in Python”. In: *Journal of Open Source Software* 9.99, p. 6101.
- Mao, Zhiping, Ameya D Jagtap, and George Em Karniadakis (2020). “Physics-informed neural networks for high-speed flows”. In: *Computer Methods in Applied Mechanics and Engineering* 360, p. 112789.
- Mariethoz, Gregoire and Jef Caers (2014). *Multiple-point geostatistics: stochastic modeling with training images*. John Wiley & Sons.
- Márquez-Neila, Pablo, Mathieu Salzmann, and Pascal Fua (2017). “Imposing hard constraints on deep networks: Promises and limitations”. In: *arXiv preprint arXiv:1706.02025*.
- Meng, Xuhui et al. (2022). “Learning functional priors and posteriors from data and physics”. In: *Journal of Computational Physics* 457, p. 111073.
- Mignan, Arnaud et al. (2015). “Induced seismicity risk analysis of the 2006 Basel, Switzerland, Enhanced Geothermal System project: Influence of uncertainties on risk mitigation”. In: *Geothermics* 53, pp. 133–146.
- Mijnlieff, Harmen F (2020). “Introduction to the geothermal play and reservoir geology of the Netherlands”. In: *Netherlands Journal of Geosciences* 99, e2.
- Misra, S, H Li, and J He (2020). “Robust geomechanical characterization by analyzing the performance of shallow-learning regression methods using unsupervised clustering methods”. In: *Machine Learning for Subsurface Characterization*, pp. 129–155.
- Moeck, Inga S (2014). “Catalog of geothermal play types based on geologic controls”. In: *Renewable and Sustainable Energy Reviews* 37, pp. 867–882.
- Morel, Francois and James Morgan (1972). “Numerical method for computing equilibria in aqueous chemical systems”. In: *Environmental Science & Technology* 6.1, pp. 58–67.
- Morikawa, Kosuke et al. (2021). “Forecasting temporal variation of aftershocks immediately after a main shock using Gaussian process regression”. In: *Geophysical Journal International* 226.2, pp. 1018–1035.
- Mottaghy, Darius, Renate Pechinig, and Christian Vogt (2011). “The geothermal project Den Haag: 3D numerical models for temperature prediction and reservoir simulation”. In: *Geothermics* 40.3, pp. 199–210.
- Moustapha, M, S Marelli, and B Sudret (2019). *UQLab user manual—Reliability-based design optimization*. Tech. rep. Technical report, Chair of Risk, Safety & Uncertainty Quantification, ETH ...

- Moustapha, Maliki and Bruno Sudret (2019). “Surrogate-assisted reliability-based design optimization: a survey and a unified modular framework”. In: *Structural and Multidisciplinary Optimization* 60, pp. 2157–2176.
- Moustapha, Maliki et al. (2016). “Quantile-based optimization under uncertainties using adaptive Kriging surrogate models”. In: *Structural and multidisciplinary optimization* 54, pp. 1403–1421.
- Mücke, Nikolaj Takata et al. (2019). “Reduced order modeling for nonlinear pde-constrained optimization using neural networks”. In: *2019 IEEE 58th Conference on Decision and Control (CDC)*. IEEE, pp. 4267–4272.
- Negri, Federico et al. (2013). “Reduced basis method for parametrized elliptic optimal control problems”. In: *SIAM Journal on Scientific Computing* 35.5, A2316–A2340.
- Niederau, Jan, J Florian Wellmann, and Nele Börsing (2019). “Analyzing the influence of correlation length in permeability on convective systems in heterogeneous aquifers using entropy production”. In: *Geothermal Energy* 7.1, pp. 1–27.
- North, Gerald R, John A Pyle, and Fuqing Zhang (2014). *Encyclopedia of atmospheric sciences*. Vol. 1. Elsevier.
- O’Leary-Roseberry, Thomas et al. (2022a). “Derivative-informed projected neural networks for high-dimensional parametric maps governed by PDEs”. In: *Computer Methods in Applied Mechanics and Engineering* 388, p. 114199.
- O’Leary-Roseberry, Thomas et al. (2022b). “Learning high-dimensional parametric maps via reduced basis adaptive residual networks”. In: *Computer Methods in Applied Mechanics and Engineering* 402, p. 115730.
- Pedregosa, F. et al. (2011). “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12, pp. 2825–2830.
- Permann, Cody J et al. (2020). “MOOSE: Enabling massively parallel multiphysics simulation”. In: *SoftwareX* 11, p. 100430.
- Pichi, Federico, Beatriz Moya, and Jan S Hesthaven (2024). “A graph convolutional autoencoder approach to model order reduction for parametrized PDEs”. In: *Journal of Computational Physics*, p. 112762.
- Pichi, Federico et al. (2022). “Driving bifurcating parametrized nonlinear PDEs by optimal control strategies: application to Navier–Stokes equations with model order reduction”. In: *ESAIM: Mathematical Modelling and Numerical Analysis* 56.4, pp. 1361–1400.
- Pichi, Federico et al. (2023). “An artificial neural network approach to bifurcating phenomena in computational fluid dynamics”. In: *Computers & Fluids* 254, p. 105813.
- Psaros, Apostolos F et al. (2023). “Uncertainty quantification in scientific machine learning: Methods, metrics, and comparisons”. In: *Journal of Computational Physics*, p. 111902.
- Qin, Zhen et al. (2023). “Efficient Optimization of Energy Recovery From Geothermal Reservoirs With Recurrent Neural Network Predictive Models”. In: *Water Resources Research* 59.3, e2022WR032653.
- Raftery, Adrian E and Steven M Lewis (1995). “The number of iterations, convergence diagnostics and generic Metropolis algorithms”. In: *Practical Markov Chain Monte Carlo* 7.98, pp. 763–773.
- Raissi, Maziar, Hessam Babaei, and George Em Karniadakis (2019). “Parametric Gaussian process regression for big data”. In: *Computational Mechanics* 64.2, pp. 409–416.
- Raissi, Maziar and George Em Karniadakis (2018). “Hidden physics models: Machine learning of nonlinear partial differential equations”. In: *Journal of Computational Physics* 357, pp. 125–141.

- Raissi, Maziar, Paris Perdikaris, and George E Karniadakis (2019). “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations”. In: *Journal of Computational physics* 378, pp. 686–707.
- Raissi, Maziar, Paris Perdikaris, and George Em Karniadakis (2017). “Machine learning of linear differential equations using Gaussian processes”. In: *Journal of Computational Physics* 348, pp. 683–693.
- Raissi, Maziar, Alireza Yazdani, and George Em Karniadakis (2020). “Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations”. In: *Science* 367.6481, pp. 1026–1030.
- Rasmussen, CE and C Williams (2006). “Gaussian processes for machine learning., (MIT Press: Cambridge, MA)”. In.
- Reed, Mark H (1982). “Calculation of multicomponent chemical equilibria and reaction processes in systems involving minerals, gases and an aqueous phase”. In: *Geochimica et Cosmochimica Acta* 46.4, pp. 513–528.
- Reeves, Howard and David J Kirkner (1988). “Multicomponent mass transport with homogeneous and heterogeneous chemical reactions: effect of the chemistry on the choice of numerical algorithm: 2. Numerical results”. In: *Water Resources Research* 24.10, pp. 1730–1739.
- Regenauer-Lieb, Klaus et al. (2010). “Time-dependent, irreversible entropy production and geodynamics”. In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 368.1910, pp. 285–300.
- Reichstein, Markus et al. (2019). “Deep learning and process understanding for data-driven Earth system science”. In: *Nature* 566.7743, pp. 195–204.
- Rockafellar, R Tyrrell, Stanislav Uryasev, et al. (2000). “Optimization of conditional value-at-risk”. In: *Journal of risk* 2, pp. 21–42.
- Romor, Francesco, Giovanni Stabile, and Gianluigi Rozza (2022). “Non-linear manifold rom with convolutional autoencoders and reduced over-collocation method”. In: *arXiv preprint arXiv:2203.00360*.
- Romor, Francesco, Marco Tezzele, and Gianluigi Rozza (2021). “ATHENA: Advanced Techniques for High dimensional parameter spaces to Enhance Numerical Analysis”. In: *Software Impacts* 10, p. 100133. DOI: 10.1016/j.simpa.2021.100133.
- Romor, Francesco et al. (2022). “Kernel-based active subspaces with application to computational fluid dynamics parametric problems using the discontinuous Galerkin method”. In: *International Journal for Numerical Methods in Engineering* 123.23, pp. 6000–6027.
- Rubin, Jacob (1990). “Solute transport with multisegment, equilibrium-controlled reactions: A feed forward simulation method”. In: *Water Resources Research* 26.9, pp. 2029–2055.
- Rutqvist, Jonny (2017). “An overview of TOUGH-based geomechanics models”. In: *Computers & Geosciences* 108, pp. 56–63.
- Ruzicka, MC (2008). “On dimensionless numbers”. In: *Chemical Engineering Research and Design* 86.8, pp. 835–868.
- Saaltink, Maarten W, Carlos Ayora, and Jesús Carrera (1998). “A mathematical formulation for reactive transport that eliminates mineral concentrations”. In: *Water resources research* 34.7, pp. 1649–1656.
- Saltelli, Andrea, Stefano Tarantola, and Francesca Campolongo (2000). “Sensitivity analysis as an ingredient of modeling”. In: *Statistical science*, pp. 377–395.
- Saltelli, Andrea et al. (2004). *Sensitivity analysis in practice: a guide to assessing scientific models*. Vol. 1. Wiley Online Library.
- Saltelli, Andrea et al. (2008). *Global sensitivity analysis: the primer*. John Wiley & Sons.

- Saltelli, Andrea et al. (2010). "Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index". In: *Computer physics communications* 181.2, pp. 259–270.
- Saltelli, Andrea et al. (2019). "Why so many published sensitivity analyses are false: A systematic review of sensitivity analysis practices". In: *Environmental modelling & software* 114, pp. 29–39.
- Samrock, Friedemann et al. (2018). "Magnetotelluric image of transcrustal magmatic system beneath the Tulu Moya geothermal prospect in the Ethiopian Rift". In: *Geophysical Research Letters* 45.23, pp. 12–847.
- Santos, Ryan et al. (2022). "State-of-the-art physics-based machine learning for hydro-mechanical simulation in geothermal applications". In: *Proceedings European Geothermal Congress*.
- Santos, Ryan Kurniawan et al. (2023a). "Physics-based machine learning for reactive transport simulations". In: *Goldschmidt 2023 Conference*. GOLDSCHMIDT.
- Santos, Ryan Kurniawan et al. (2023b). *Uncertainty quantification with a physics-based machine learning method for geothermal-well targeting: A case study of The Hague, Netherlands*. Tech. rep. Copernicus Meetings.
- Sarma, Pallav et al. (2008). "Production optimization with adjoint models under non-linear control-state path inequality constraints". In: *SPE Reservoir Evaluation & Engineering* 11.02, pp. 326–339.
- Schaeffner, Maximilian et al. (2021). "Forward vs. Bayesian Inference Parameter Calibration: Two Approaches for Non-deterministic Parameter Calibration of a Beam-Column Model". In: *International Conference on Uncertainty in Mechanical Engineering*. Springer, pp. 173–190.
- Schulte, Daniel O et al. (2020). "Multi-objective optimization under uncertainty of geothermal reservoirs using experimental design-based proxy models". In: *Geothermics* 86, p. 101792.
- Siddiqui, Mohammed Abdul Qadeer, Klaus Regenauer-Lieb, and Hamid Roshan (2024). "Thermo-Hydro-Chemo-Mechanical (THCM) Continuum Modeling of Subsurface Rocks: A Focus on Thermodynamics-Based Constitutive Models". In: *Applied Mechanics Reviews* 76.3, p. 031001.
- Simmeling, HJ, V Vandeweyer, and J Ramaekers (2007). "Geologisch locatie-specifiek onderzoek voor het Business Plan Geothermie Den Haag Zuid-West". In: *TNO-report*. www.geothermie.nl.
- Smith, William Robert (1982). "Chemical reaction equilibrium analysis". In: *Theory and Algorithms*.
- Sobol, Ilya M (2001). "Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates". In: *Mathematics and computers in simulation* 55.1–3, pp. 271–280.
- Sprocati, R and M Rolle (2021). "Integrating process-based reactive transport modeling and machine learning for electrokinetic remediation of contaminated groundwater". In: *Water Resources Research* 57.8, e2021WR029959.
- Stabile, Giovanni and Gianluigi Rozza (2018). "Finite volume POD-Galerkin stabilised reduced order methods for the parametrised incompressible Navier–Stokes equations". In: *Computers & Fluids* 173, pp. 273–284.
- Steeffel, Carl I, Donald J DePaolo, and Peter C Lichtner (2005). "Reactive transport modeling: An essential tool and a new research approach for the Earth sciences". In: *Earth and Planetary Science Letters* 240.3–4, pp. 539–558.
- Steeffel, Carl I and Antonio C Lasaga (1994). "A coupled model for transport of multiple chemical species and kinetic precipitation/dissolution reactions with application to

- reactive flow in single phase hydrothermal systems". In: *American Journal of science* 294.5, pp. 529–592.
- Stockmann, M et al. (2017). "Smart Kd-values, their uncertainties and sensitivities- Applying a new approach for realistic distribution coefficients in geochemical modeling of complex systems". In: *Chemosphere* 187, pp. 277–285.
- Strazzullo, Maria et al. (2018). "Model reduction for parametrized optimal control problems in environmental marine sciences and engineering". In: *SIAM Journal on Scientific Computing* 40.4, B1055–B1079.
- Stuart, Andrew M (2010). "Inverse problems: a Bayesian perspective". In: *Acta numerica* 19, pp. 451–559.
- Sullivan, Timothy John (2015). *Introduction to uncertainty quantification*. Vol. 63. Springer.
- Swischuk, Renee et al. (2019). "Projection-based model reduction: Formulations for physics-based machine learning". In: *Computers & Fluids* 179, pp. 704–717.
- Swischuk, Renee et al. (2020). "Learning physics-based reduced-order models for a single-injector combustion process". In: *AIAA Journal* 58.6, pp. 2658–2672.
- Tang, Yuanfu, Jianqiao Chen, and Junhong Wei (2012). "A sequential algorithm for reliability-based robust design optimization under epistemic uncertainty". In: *Reliability Engineering & System Safety* 110, pp. 1–11.
- Tarantola, Albert (2005). *Inverse problem theory and methods for model parameter estimation*. SIAM.
- Tartakovsky, Alexandre M et al. (2020). "Physics-informed deep neural networks for learning parameters and constitutive relationships in subsurface flow problems". In: *Water Resources Research* 56.5, e2019WR026731.
- Thurey, Nils et al. (2021). "Physics-based deep learning". In: *arXiv preprint arXiv:2109.05237*.
- Titos, Manuel et al. (2018). "A deep neural networks approach to automatic recognition systems for volcano-seismic events". In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 11.5, pp. 1533–1544.
- Tonicello, Niccolò et al. (2022). "Non-intrusive reduced order models for the accurate prediction of bifurcating phenomena in compressible fluid dynamics". In: *arXiv preprint arXiv:2212.10198*.
- Tsai, Wen-Ping et al. (2021). "From calibration to parameter learning: Harnessing the scaling effects of big data in geoscientific modeling". In: *Nature communications* 12.1, p. 5988.
- Turcotte, Donald L and Gerald Schubert (2002). *Geodynamics*. Cambridge university press.
- Van Zelst, Iris et al. (2022). "101 geodynamic modelling: how to design, interpret, and communicate numerical studies of the solid Earth". In: *Solid Earth* 13.3.
- Veldkamp, JG et al. (2016). *Thermal fracturing due to low injection temperatures in geothermal doublets*. Utrecht: TNO.
- Verweij, JM et al. (2012). "Pressure and fluid dynamic characterisation of the Dutch subsurface". In: *Netherlands Journal of Geosciences* 91.4, pp. 465–490.
- Virtanen, Pauli et al. (2020). "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". In: *Nature Methods* 17, pp. 261–272. doi: 10.1038/s41592-019-0686-2.
- Vogt, C et al. (2010). "Reducing temperature uncertainties by stochastic geothermal reservoir modelling". In: *Geophysical Journal International* 181.1, pp. 321–333.
- Waheed, Umair bin et al. (2021). "PINNeik: Eikonal solution using physics-informed neural networks". In: *Computers & Geosciences* 155, p. 104833.
- Wainwright, Haruko M et al. (2014). "Making sense of global sensitivity analyses". In: *Computers & Geosciences* 65, pp. 84–94.

- Wang, Chunhong, Gaoming Li, and Albert C Reynolds (2009). "Production optimization in closed-loop reservoir management". In: *SPE journal* 14.03, pp. 506–523.
- Wang, Jie (2020). "An intuitive tutorial to Gaussian processes regression". In: *arXiv preprint arXiv:2009.10862*.
- Wang, Nanzhe et al. (2023). "Deep learning based closed-loop well control optimization of geothermal reservoir with uncertain permeability". In: *Renewable Energy* 211, pp. 379–394.
- Wang, Qian, Jan S Hesthaven, and Deep Ray (2019). "Non-intrusive reduced order modeling of unsteady flows using artificial neural networks with application to a combustion problem". In: *Journal of computational physics* 384, pp. 289–307.
- Wang, Sifan, Yujun Teng, and Paris Perdikaris (2021). "Understanding and mitigating gradient flow pathologies in physics-informed neural networks". In: *SIAM Journal on Scientific Computing* 43.5, A3055–A3081.
- Wellmann, Florian and Guillaume Caumon (2018). "3-D Structural geological models: Concepts, methods, and uncertainties". In: *Advances in geophysics*. Vol. 59. Elsevier, pp. 1–121.
- Wellmann, J Florian and Klaus Regenauer-Lieb (2012). "Effect of geological data quality on uncertainties in geological models and subsurface flow fields". In: *Thirty-Seventh Workshop on Geothermal Reservoir Engineering*.
- Willard, Jared et al. (2020). "Integrating scientific knowledge with machine learning for engineering and environmental systems". In: *arXiv preprint arXiv:2003.04919*.
- Willcox, Karen E, Omar Ghattas, and Patrick Heimbach (2021). "The imperative of physics-based modeling and inverse theory in computational science". In: *Nature Computational Science* 1.3, pp. 166–168.
- Willems, Cees JL et al. (2020). "Geology of the Upper Jurassic to Lower Cretaceous geothermal aquifers in the West Netherlands Basin—an overview". In: *Netherlands Journal of Geosciences* 99, e1.
- Wiszniowski, Jan, Beata M Plesiewicz, and Jacek Trojanowski (2014). "Application of real time recurrent neural network for detection of small natural earthquakes in Poland". In: *Acta Geophysica* 62, pp. 469–485.
- Witte, Robert S and John S Witte (2017). *Statistics*. John Wiley & Sons.
- Wu, Keyi et al. (2023). "Large-scale Bayesian optimal experimental design with derivative-informed projected neural network". In: *Journal of Scientific Computing* 95.1, p. 30.
- Wu, Pin et al. (2021). "Reduced order model using convolutional auto-encoder with self-attention". In: *Physics of Fluids* 33.7.
- Yan, Bicheng, Manojkumar Gudala, and Shuyu Sun (2023). "Robust optimization of geothermal recovery based on a generalized thermal decline model and deep learning". In: *Energy Conversion and Management* 286, p. 117033.
- Yang, Liu, Xuhui Meng, and George Em Karniadakis (2021). "B-PINNs: Bayesian physics-informed neural networks for forward and inverse PDE problems with noisy data". In: *Journal of Computational Physics* 425, p. 109913.
- Yang, Liu, Dongkun Zhang, and George Em Karniadakis (2020). "Physics-informed generative adversarial networks for stochastic differential equations". In: *SIAM Journal on Scientific Computing* 42.1, A292–A317.
- Yeung, Yu-Hong, David A Barajas-Solano, and Alexandre M Tartakovsky (2022). "Physics-Informed Machine Learning Method for Large-Scale Data Assimilation Problems". In: *Water Resources Research* 58.5, e2021WR031023.
- Yin, Minglang et al. (2021). "Non-invasive inference of thrombus material properties with physics-informed neural networks". In: *Computer Methods in Applied Mechanics and Engineering* 375, p. 113603.

- Zang, C, MI Friswell, and JE Mottershead (2005). “A review of robust optimal design and its application in dynamics”. In: *Computers & structures* 83.4-5, pp. 315–326.
- Zhang, Chiyuan et al. (2021). “Understanding deep learning (still) requires rethinking generalization”. In: *Communications of the ACM* 64.3, pp. 107–115.
- Zhang, Enrui, Minglang Yin, and George Em Karniadakis (2020). “Physics-informed neural networks for nonhomogeneous material identification in elasticity imaging”. In: *arXiv preprint arXiv:2009.04525*.
- Zhang, Enrui et al. (2022). “Analyses of internal structures and defects in materials using physics-informed neural networks”. In: *Science advances* 8.7, eabk0644.
- Zhang, Zhenying, Eduard Bader, and Karen Veroy (2014). “A duality approach to error estimation for variational inequalities”. In: *arXiv preprint arXiv:1410.2095*.
- Zhu, Weiqiang and Gregory C Beroza (2019). “PhaseNet: A deep-neural-network-based seismic arrival-time picking method”. In: *Geophysical Journal International* 216.1, pp. 261–273.