

---

# **Integrating geological uncertainty and dynamic data into modelling procedures for fractured reservoirs**

---

BASTIAN STEFFENS

*A thesis submitted in fulfilment of the requirements  
for the degree of Doctor of Philosophy*

Institute of GeoEnergy Engineering  
School of Energy, Geoscience, Infrastructure and Society  
Heriot-Watt University

July 2022

The copyright in this thesis is owned by the author. Any quotation from the thesis or use of any of the information contained in it must acknowledge this thesis as the source of the quotation or information.

# *Abstract*

Modelling, simulating and characterising flow through naturally fractured reservoirs is a multi-disciplinary effort. The scarcity of data combined with the additional layer of complexity that fractures add to a reservoir makes an efficient integration of all available data fundamental. However, the vast range of data types to be considered and the multitude of disciplines giving their input often results in communication barriers and individuals working within their comfort area, creating further challenges for uncertainty propagation. It is however critical for decision-making to develop geologically consistent reservoir models that recognise the challenges of simulating flow through systems with high permeability and scale contrasts and address the need for an ensemble of reservoir models to sufficiently cover geological uncertainties and their impact on fluid flow.

In this work I developed several workflows for naturally fractured reservoir modelling that invite cross-disciplinary thinking by integrating geological uncertainties and dynamic data into the modelling procedure and foster ensemble modelling from the start. The workflows are tested on a synthetic field that is based upon a conceptual model for fold-related fracture distributions. The first workflow involves the use of multiple-point statistics to efficiently model reservoir-scale fracture distribution by upscaling discrete fracture networks and converting them into training images. To cover the impact of fracture-related geological uncertainties on fluid flow efficiently, flow diagnostics were used to screen and afterwards cluster and select training images according to their flow response for further reservoir modelling. The second workflow proposes a novel reservoir modelling technique that considers both static and dynamic data and utilises entropy to generate a diverse ensemble of reservoir models that all match an outset objective. Finally, an agent-based reservoir modelling workflow is introduced, where within a reservoir model, independent but interacting agents follow a set of rules to generate reservoir models that take into account geological prior information and expected dynamic flow responses to drive modelling efforts.

Overall, we demonstrated that combining approaches from various disciplines into cross-disciplinary workflows provides great potential for subsurface characterisation. What workflow to adopt within a project, depends on various boundary conditions. The availability of data and time, the confidence in the understanding of the reservoir and the ultimate goal behind the modelling exercise. These factors can impact whether moving along with the simpler, more parametric multiple-point statistics workflow, the entropy-driven workflow that utilises static and dynamic data or the more data-driven agent-based modelling workflow is the right choice.

# *Acknowledgements*

Many people's help, guidance, and support went into this thesis. Without them this work would not have been possible. I want to express my gratitude towards my supervisors Vasily Demyanov, Dan Arnold and Helen Lewis for taking me on as their PhD student. Thank you Vasily for your stewardship and advice where needed and for always being supportive and giving me plenty of opportunities to explore. Dan, thank you for being filled with a sheer endless amount of fantastic ideas and substantially improving my writing and presentation skills. Helen, it was great having you in my supervisory team and making sure that there was always someone with a *geologists hat* looking at this work. Thank you also to you and Gary Couples for inviting me to your Geomechanics course in Grenoble at the very beginning of my PhD.

I would also like to thank everyone in the energy academy for making it a melting pot of different backgrounds and cultures where you can get caught up talking about all sorts of exciting topics and learn new things daily. There was something for everyone, from discovering Mexican or Russian sweets (both very intriguing) to discussions about different cuisines, worldviews, and politics with perspectives from all over the planet. I would especially like to mention Daniel, Jorge, Fran, Christine, Victoria and Mark for the good times.

A big shout out goes to all current and former members of the GeoDataScience Group. We went on some memorable trips to Serbia and London and group dinners turned out to always be loads of fun. Especially I would like to thank Rhona for quickly becoming a really good friend and for making my start in Scotland and within the group super enjoyable. I wish you all the best with your new job where I am sure you will not fail to impress. Tom for being a great dinner host and helping me out with Petrel a lot initially, all under the condition of being mentioned here ;). Quentin for plenty of fun and successful projects that we were and still are working on together. Despite them being only marginally related to our PhDs, they kept things exciting and I learned loads from it. I am looking forward to continuing this. Athos for some vivid discussions about potential startup ideas from your PhD - do not forget the 10% ;). And Chao for the final stint of my PhD where I learned plenty about the world of tea and got to appreciate your company in the still empty office.

Being part of the CDT for Oil & Gas, put together by John Underhill was also a pleasure. Thank you Anna for organising fantastic field trips and training courses that allowed me to see so many different corners of the UK and Europe. I truly enjoy the community that came with the CDT and it is great to have this network of friends all over the UK and beyond. I particularly want to mention Louis and Eduardo for becoming great friends of mine.

A special thank you also goes to Alan, the one-person IGE IT department. Thanks for ensuring that the working from home phase went so well and for always being helpful and solving problems almost immediately during regular times.

On the Provence field trip, I learned a tremendous amount about naturally fractured reservoirs within a short period of time whilst also enjoying great company, food, and scenery. Thank you Sebastian Geiger and the Carbonates Group for inviting me.

I would also like to thank Nathanael for being a great squash, swim, cycle, hill and croissant run buddy. We've definitely experienced some adventures together and it is always great chatting to you.

I want to thank all my friends from back home in Germany. It is incredible to have such a strong and supportive group of people around you who welcome you warmly and with open arms after so many months of not seeing each other.

Second to last I want to say thank you to my family for supporting me throughout all of my academic endeavours for the past couple of years. It is always good to know that someone out there has your back.

Finally, I would like to thank Katie for being the most wonderful partner to me and enduring me during the height of lockdowns or PhD-meltdowns. Every day with you is an inspiration, and I am looking forward to our next big adventure.

This PhD study was undertaken as part of the Natural Environmental Research Council (NERC) Centre for Doctoral Training (CDT) in Oil Gas. It is sponsored via a James Watt Scholarship provided by Heriot Watt University (HWU) whose support is gratefully acknowledged.

Edinburgh, February 2022

## Research Thesis Submission

Please note this form should be bound into the submitted thesis.

Name:	Bastian Steffens		
School:	EGIS		
Version: <i>(i.e. First, Resubmission, Final)</i>	Final	Degree Sought:	Doctor of Philosophy

### Declaration

In accordance with the appropriate regulations I hereby submit my thesis and I declare that:

1. The thesis embodies the results of my own work and has been composed by myself
2. Where appropriate, I have made acknowledgement of the work of others
3. The thesis is the correct version for submission and is the same version as any electronic versions submitted\*.
4. My thesis for the award referred to, deposited in the Heriot-Watt University Library, should be made available for loan or photocopying and be available via the Institutional Repository, subject to such conditions as the Librarian may require
5. I understand that as a student of the University I am required to abide by the Regulations of the University and to conform to its discipline.
6. I confirm that the thesis has been verified against plagiarism via an approved plagiarism detection application e.g. Turnitin.

### ONLY for submissions including published works

Please note you are only required to complete the Inclusion of Published Works Form (page 2) if your thesis contains published works)

7. Where the thesis contains published outputs under Regulation 6 (9.1.2) or Regulation 43 (9) these are accompanied by a critical review which accurately describes my contribution to the research and, for multi-author outputs, a signed declaration indicating the contribution of each author (complete)
8. Inclusion of published outputs under Regulation 6 (9.1.2) or Regulation 43 (9) shall not constitute plagiarism.

\* Please note that it is the responsibility of the candidate to ensure that the correct version of the thesis is submitted.

Signature of Candidate:		Date:	01/07/2022
-------------------------	---	-------	------------

### Submission

Submitted By ( <i>name in capitals</i> ):	BASTIAN STEFFENS
Signature of Individual Submitting:	
Date Submitted:	01/07/2022

### For Completion in the Student Service Centre (SSC)

Limited Access	Requested	Yes	No	Approved	Yes	No
<i>E-thesis Submitted (mandatory for final theses)</i>						
Received in the SSC by ( <i>name in capitals</i> ):				Date:		

### Inclusion of Published Works

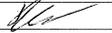
Please note you are only required to complete the Inclusion of Published Works Form if your thesis contains published works under Regulation 6 (9.1.2)

#### Declaration

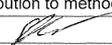
This thesis contains one or more multi-author published works. In accordance with Regulation 6 (9.1.2) I hereby declare that the contributions of each author to these publications is as follows:

Citation details	Steffens, B., Demyanov, V., Couples, G., Arnold, D., and Lewis, H. (2019). Flow Through Fractured Reservoirs Under Geological and Geomechanical Uncertainty. In 81 <sup>st</sup> EAGE Conference and Exhibition 2019, London, United Kingdom.
Author 1	Contribution to method development in discussions, code implementation, wrote manuscript
Author 2	Supervision, contribution to method development in discussions
Author 3	Supervision, contribution to method development in discussions
Author 4	Supervision, contribution to method development in discussions
Author 5	Supervision, contribution to method development in discussions
Signature:	
Date:	01/07/2022

Citation details	Steffens, B., Demyanov, V., Arnold, D., and Lewis, H. (2020). A Workflow for Dynamic Screening Assisted, Automated Fractured Reservoir Modelling. In 82 <sup>nd</sup> EAGE Annual Conference and Exhibition Online 2020.
Author 1	Contribution to method development in discussions, code implementation, wrote manuscript
Author 2	Supervision, contribution to method development in discussions
Author 3	Supervision, contribution to method development in discussions
Author 4	Supervision, contribution to method development in discussions
Signature:	
Date:	01/07/2022

Citation details	Steffens, B., Demyanov, V., Arnold, D., and Lewis, H. (2021). Entropy-driven particle swarm optimization for reservoir modelling under geological uncertainty – application to a fractured reservoir. In 83 <sup>rd</sup> EAGE Annual Conference and Exhibition 2021, Amsterdam, Netherlands.
Author 1	Contribution to method development in discussions, code implementation, wrote manuscript
Author 2	Supervision, contribution to method development in discussions
Author 3	Supervision, contribution to method development in discussions
Author 4	Supervision, contribution to method development in discussions
Signature:	
Date:	01/07/2022

Citation details	Steffens, B., Demyanov, V., Arnold, D., and Lewis, H. (2021). Can agent-based modelling help to update conceptual geological models? - A fractured reservoir example. In 2021 RING Meeting. ASGA
Author 1	Contribution to method development in discussions, code implementation, wrote manuscript
Author 2	Supervision, contribution to method development in discussions

Author 3	Supervision, contribution to method development in discussions
Author 4	Supervision, contribution to method development in discussions
Signature:	
Date:	01/07/2022

Citation details	Steffens, B., Corlay, Q., Suurmeyer, N., Noglows, J., Arnold, D., and Demyanov, V.(2022). Can agents model hydrocarbon migration for petroleum system analysis? A fast-screening tool to de-risk hydrocarbon prospects.Energies, 15(3):902.
Author 1	Contribution to method development in discussions, code implementation, wrote manuscript
Author 2	Contribution to method development in discussions, code implementation, wrote manuscript
Author 3	Supervision, contribution to method development in discussions
Author 4	Supervision, contribution to method development in discussions
Author 5	Supervision, contribution to method development in discussions
Author 6	Supervision, contribution to method development in discussions
Signature:	
Date:	01/07/2022

# Contents

<b>Abstract</b>	<b>i</b>
<b>Acknowledgements</b>	<b>ii</b>
<b>Contents</b>	<b>vi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Chapter objectives and thesis outline . . . . .	4
<b>2 Naturally fractured reservoirs and how to model them</b>	<b>8</b>
2.1 What makes a reservoir a naturally fractured reservoir . . . . .	9
2.2 How and where do natural fractures occur? . . . . .	13
2.3 Modelling and flow simulation of naturally fractured reservoirs . . . . .	17
2.4 Setup of synthetic naturally fractured reservoir . . . . .	22
<b>3 Methodology</b>	<b>26</b>
3.1 Multiple-point statistics . . . . .	26
3.1.1 The training image . . . . .	28
3.1.2 Different multiple-point statistics algorithms . . . . .	29
3.2 Flow diagnostics . . . . .	32
3.3 Meta-heuristic optimisation . . . . .	35
3.3.1 The particle swarm optimiser (PSO) . . . . .	37
3.4 Entropy . . . . .	44
3.5 Dimension reduction . . . . .	46
3.6 Clustering . . . . .	50
3.7 Agent-based modelling . . . . .	56
<b>4 Modelling uncertainty in naturally fractured reservoirs with multiple-point statistics</b>	<b>58</b>
4.1 Introduction . . . . .	58
4.2 Previous applications of MPS to naturally fractured reservoir modelling . . . . .	59
4.3 Methodology . . . . .	64
4.3.1 Step 1: Geological analysis . . . . .	65
4.3.2 Step 2: Monte-Carlo DFN simulation . . . . .	66
4.3.3 Step 3: Upscaling . . . . .	66

4.3.4	Step 4: Grid cell classification / rock typing . . . . .	67
4.3.5	Step 5: Run flow diagnostics . . . . .	67
4.3.6	Step 6: Calculate training image entropy . . . . .	68
4.3.7	Step 7: Dimension reduction . . . . .	70
4.3.8	Step 8: Clustering . . . . .	70
4.3.9	Step 9: Modelling and flow simulations . . . . .	70
4.4	Results . . . . .	71
4.4.1	Application to a synthetic reservoir . . . . .	71
4.4.2	Step 1: Geological analysis . . . . .	71
4.4.3	Step 2: Monte-Carlo DFN simulation . . . . .	72
4.4.4	Step 3 + 4: Upscaling & rock typing . . . . .	74
4.4.5	Step 5: Running flow diagnostics . . . . .	76
4.4.6	Step 6: Training image entropy . . . . .	76
4.4.7	Step 7 + 8: Dimension reduction + clustering . . . . .	79
4.4.8	Step 9: Modelling and flow simulations . . . . .	81
4.5	Discussion . . . . .	83
<b>5</b>	<b>Entropy-guided and dynamic response driven modelling of naturally fractured reservoirs</b>	<b>88</b>
5.1	Introduction . . . . .	88
5.2	Methodology . . . . .	89
5.2.1	Geological model building . . . . .	93
5.2.2	Screening models for dynamic response - Flow diagnostics . . . . .	94
5.2.3	Optimisation . . . . .	94
5.2.4	Post-processing . . . . .	103
5.3	Results . . . . .	106
5.3.1	Application to synthetic case study . . . . .	106
5.3.2	Standard PSO - local search . . . . .	107
5.3.3	Entropy-PSO . . . . .	122
5.4	Discussion . . . . .	136
<b>6</b>	<b>Agent-based modelling for naturally fractured reservoirs</b>	<b>145</b>
6.1	Introduction . . . . .	145
6.2	How agent-based modelling can improve conceptual geological models . . . . .	147
6.3	Methodology . . . . .	148
6.3.1	Agent-based model design . . . . .	148
6.3.2	ABM workflow summary . . . . .	154
6.3.3	Validation of implemented rules . . . . .	154
6.4	Results . . . . .	158
6.4.1	Application to synthetic case study . . . . .	158
6.4.2	Geological concept and expected dynamic flow behaviour . . . . .	160
6.4.3	ABM setup and objective . . . . .	160
6.4.4	ABM simulation results . . . . .	161
6.5	Discussion . . . . .	163

<b>7</b>	<b>Summary, conclusion and future work</b>	<b>167</b>
7.1	Summary . . . . .	167
7.2	Conclusions . . . . .	169
7.3	Recommendations for future work . . . . .	171
	<b>References</b>	<b>175</b>

# List of Publications

This thesis contains excerpts from the following papers:

**Steffens, B.**, Demyanov, V., Couples, G., Arnold, D., and Lewis, H. (2019). Flow Through Fractured Reservoirs Under Geological and Geomechanical Uncertainty. *In 81st EAGE Conference and Exhibition 2019, London, United Kingdom.*

**Steffens, B.**, Demyanov, V., Arnold, D., and Lewis, H. (2020). A Workflow for Dynamic Screening Assisted, Automated Fractured Reservoir Modelling. *In 82nd EAGE Annual Conference and Exhibition Online 2020.*

**Steffens, B.**, Demyanov, V., Arnold, D., and Lewis, H. (2021). Entropy-driven particle swarm optimization for reservoir modelling under geological uncertainty – application to a fractured reservoir. *In 83rd EAGE Annual Conference and Exhibition 2021, Amsterdam, Netherlands.*

**Steffens, B.**, Demyanov, V., Arnold, D., and Lewis, H. (2021). Can agent-based modelling help to update conceptual geological models? - A fractured reservoir example. *In 2021 RING Meeting. ASGA.*

The following paper was published during this thesis and helped in its development:

**Steffens, B.**, Corlay, Q., Suurmeyer, N., Noglows, J., Arnold, D., and Demyanov, V. (2022). Can agents model hydrocarbon migration for petroleum system analysis? A fast-screening tool to de-risk hydrocarbon prospects. *Energies*, 15(3):902. <https://doi.org/10.3390/en15030902>.

# Chapter 1

## Introduction

On a daily basis, we all come into contact with products made from natural resources extracted from the Earth in some way. Yet, based on personal experiences, the general public has little knowledge about where these resources come from, how their production and accumulation can be managed sustainably and efficiently and how things look beneath the surface we live. A reason for this lack of knowledge or high degree of uncertainty is that the subsurface domain is far from our everyday life and therefore a rather abstract concept. Nevertheless, understanding the subsurface is as essential now as it has been in the past. With the initiation of the Paris Agreement in 2016, the energy transition is gradually starting to take shape. There are several pillars of the energy transition where knowledge about the subsurface is fundamental for their success. For example, a potential solution for the large-scale temporary storage of excess power produced from wind farms or solar parks would be a power-to-gas system. Excess power is converted to synthetic gas and stored in salt caverns for later usage. Other examples would be the energy extraction from geothermal systems or capturing carbon dioxide directly from heavy emitters such as cement producers and storing it in the subsurface in its liquid or solid state. Every example listed above requires substantial upfront financial investments, and, if executed incorrectly, uninformed decisions can have significant negative impacts on future generations. To prevent these from happening, we must understand the subsurface, minimise risk and allow us to make informed decisions under our uncertain view of the subsurface and lack of understanding to the needed level of detail.

To allow for a smooth energy transition with minimal impact on society's other needs, such as energy security and standard of living, a functioning and reliable supply of energy sources such as hydrocarbons is critical. To ensure that the extraction of hydrocarbons from oil or gas reservoirs is done as sustainably, economically and dependably as possible, it is once more important to understand subsurface workings. However, like the general public, we - the geoscientists, reservoir engineers and subsurface experts - only have very little information on how exactly the subsurface - our area of operation - looks. Most knowledge is based upon indirect measurements, lab experiments, analogue data, outcrops and conceptual understandings. Therefore, we mainly know what the subsurface *could* look like but can never be entirely sure. Thus any decision regarding the extraction of hydrocarbons from reservoirs entails large degrees of uncertainty.

To integrate all available information about a reservoir and better understand and characterise it, geoscientists generate 3D geological models of the reservoir. With the help of different modelling techniques, geological structures, bodies and their associated properties (e.g. rock type, porosity and permeability) are generated. To the dismay of many geoscientists however, it should be noted that these modelling techniques can capture the complexity of nature only to some extent. Reservoir engineers then apply time-consuming numerical methods on the generated reservoir models that simulate how fluids would flow through the interconnected pore space of the reservoir rocks. This information combined can then be used to predict the reservoirs' flow performance and the distribution of hydrocarbons within the reservoir. As mentioned beforehand, we only have a limited amount of information to characterise the reservoir. Therefore, basing predictions about reservoir performance on a single reservoir model would not adequately account for the uncertainty we encounter in the subsurface. Ideally, we would have a large ensemble of reservoir models covering a wide range of encountered uncertainties and thereby identify their impact on fluid flow behaviour. This would leave us with hundreds of reservoir models on which to perform flow simulations, taking a substantial amount of time and being unfeasible in practice. Instead, we must choose a representative subset from the ensemble to take forward to the flow simulation step. Often time constraints or difficulties with handling several models simultaneously and making predictions based upon the outcomes from the ensemble lead to the collapse of uncertainty into a single base case. However, relying

on a single base case for decision-making under uncertainty is not ideal as the predictive capabilities of a single model are questionable [Bentley and Smith, 2008, Ringrose and Bentley, 2021].

These issues are especially true for naturally fractured reservoirs (NFRs). They hold around 60% of the world's proven oil and 40% of gas reserves but are inherently difficult to manage successfully [Schlumberger, 2008]. They tend to have short production lifespans, with high initial production attributed to only a subgroup of wells, making the best part of the drilled wells somewhat obsolete. Often, several essential producers reach their shut-in pressure or show signs of early water breakthrough, resulting in a fast decline in production. One reason for this behaviour is that fluids can rapidly be channelled through the network of natural fractures and bypass the majority of the reservoir. Overall, this leads to a low final recovery of less than 20% for more than 60% of fractured reservoirs [Allen and Sun, 2003]. The main driver for these moderate results is the lack of understanding of how fractured reservoirs work. Static and dynamic data to describe fracture distributions and their behaviour in the subsurface are scarce and hard to obtain [Bratton et al., 2006]. Their comparably small size and quasi-planar dimension make them difficult to detect with seismic, rendering it hard to predict their field-wide distribution/density and inter-connectivity. Altogether, this makes building geologically consistent models of naturally fractured reservoirs difficult and leads to an extra layer of uncertainty that needs to be considered in development decisions.

Additional problems arise from the fact that understanding and modelling NFRs and simulating flow through NFRs is a task of multi-disciplinary nature. Whilst this is also true for conventional reservoirs, the added layer of uncertainty and fewer constraining points make this even more problematic for NFRs. As pointed out in the previous section, data on fracture distributions and characteristics are scarce, and therefore all information available should be integrated into the reservoir model. This could for example be outcrop data and a conceptual understanding of fracture distributions in particular geological settings. The geoscientist then uses this information to generate the reservoir model and aims for it to be as geological realistic as possible. When speaking of geological realism in this thesis, I do not necessarily refer to reservoir models that could be a snapshot of what we notice in outcrops. The main priority is capturing the effect of the geological

observations on fluid flow on a reservoir scale. Reservoir engineers then have to balance between mimicking the actual flow rates observed in the wells and selecting appropriate, not overly time-consuming, fluid flow simulation tools. The problem is that the strong permeability contrasts of up to several magnitudes between fractures and the rock matrix make simulating flow rather complex. Often simplifications at the cost of the geological input have to be made, which can lead to the generation of geologically inconsistent or unrealistic models that cannot make reliable predictions [Arnold, 2008, Carter et al., 2006]. Finally, an expert for uncertainty quantification or a data scientist applies their tool-set to the generated reservoir models to quantify uncertainty. All these modelling steps implied workflows that are often silo-ed within the expert domain and do not communicate very well between one another especially handling uncertainty propagation. This creates a challenge to come up with consistent modelling workflows that would combine data, knowledge and expertise across these silos. In current workflows, the sheer range of different data types of variable degrees of resolution, scale and certainty and the multitude of disciplines giving their input encourages individuals to operate within their silo.

This thesis therefore focuses on developing decision-oriented reservoir modelling workflows applied to naturally fractured reservoirs that from the beginning require multi-disciplinary input, ideally leading to a *de-silofication* of thinking and better cross-disciplinary communication between geoscience, engineering and data science. From the beginning the workflows promote the integration of dynamic data and geological uncertainties, leading to the generation of ensembles of geologically sound reservoir models, where uncertainty is not collapsed but rather spread over different possible scenarios.

## **1.1 Chapter objectives and thesis outline**

This thesis aims to address some of the challenges inherent when dealing with naturally fractured reservoirs. Specifically, it focuses on assembling workflows that carry geological uncertainties around fracture distributions and information about dynamic reservoir behaviour throughout the entire modelling, simulation, and decision-making cycle of a project. For reasons of practicality, this has to be done efficiently with a manageable ensemble of reservoir models that robustly quantify a wide range of uncertainties and

realistic geological scenarios for naturally fractured reservoirs. To achieve this, a common ground between the disciplines of geoscience, reservoir engineering and uncertainty quantification and data science has to be established. Traditional modelling approaches follow a linear procedure, where each discipline gives their input at a different stage of a project, naturally suppressing an interdisciplinary working culture. Therefore, the workflows developed here require all involved disciplines to provide input from the beginning of a project. This obliges different disciplines to work together, gradually leading to the diffusion of domain knowledge between the different disciplines, eventually leading to a better understanding of the needs and uncertainties that each discipline addresses. The thesis is divided into seven chapters and structured as follows:

### **Chapter 2: Naturally fractured reservoirs and how to model them**

This chapter gives a brief introduction to naturally fractured reservoirs and the impact that fractures have on fluid flow. The chapter will cover different types of fractures, fracture distributions and methods for modelling and simulating fluid flow. The chapter closes with the setup of a synthetic but geologically consistent and realistic case study for a naturally fractured reservoir that will be used throughout this work.

---

### **Chapter 3: Methodology**

This chapter gives a brief overview of the methods across geomodelling, engineering and computer science domain used throughout the thesis. First, the concept of multiple-point statistics and its advantages over other spatial property modelling techniques is outlined. Flow diagnostics, a proxy model for fluid flow simulations, is introduced next. This is followed by entropy, a measure from information theory used to evaluate the model ensemble diversity. Then the particle swarm optimiser (PSO) - an optimisation technique used to guide geological model building is presented, followed by dimension reduction and clustering techniques used to reduce the model ensemble for manageable uncertainty quantification. The chapter ends with a brief discussion about the concept of agent-based modelling that is used to enhance non-parametric geological model tuning with geologically consistent rule-based guidance .

#### **Chapter 4: Modelling uncertainty in naturally fractured reservoirs with multiple-point statistics**

This chapter introduces a method that allows modelling distributions of naturally fractures on reservoir scale with the help of multiple-point statistics. Discrete fracture networks are upscaled and turned into training images that can then capture the impact that geological uncertainties around fracture distributions have on fluid flow. The developed method is tested on the synthetic case study introduced in chapter 2. The methods for training image generation developed in this chapter also build the foundation for the work carried out in the upcoming chapters.

---

#### **Chapter 5: Entropy-guided and dynamic response driven modelling of naturally fractured reservoirs**

This chapter demonstrates two novel geological modelling workflows capable of generating reservoir models that are preconditioned on dynamic data and specific geological concepts with the help of flow diagnostics and the particle swarm optimisation algorithm. Entropy is used to steer the model generation and ensure that an ensemble of geologically diverse reservoir models that capture the impact of geological uncertainties on flow dynamics is generated. With the help of several post-processing steps, maps for future decision-making are also introduced and based upon the information obtained from the modelling workflows. The workflows are then tested on the synthetic case study presented in chapter 2.

---

#### **Chapter 6: Agent-based modelling for naturally fractured reservoirs**

In this chapter an agent-based approach for dynamically driven reservoir modelling is introduced. By combining agent-based modelling (ABM) with flow diagnostics, reservoir models are iteratively updated and allow for the adaptation of newly incoming data. The versatility of ABMs allows addressing a wide bandwidth of different questions with only minor modifications to the initial setup of the ABM. Here the ABM is used to update prior conceptual geological models in uncertainty quantification studies.

---

## **Chapter 7: Conclusions, summary and future work**

This chapter covers the main findings and workflows developed in this thesis and sets them into relation to the aim of the thesis. It also presents an outlook and points out several considerations for future investigations and potential improvements.

## **Chapter 2**

# **Naturally fractured reservoirs and how to model them**

Reservoirs can be classified as naturally fractured reservoirs (NFRs), when the flow response of the reservoir is different to which can typically be expected for this type of reservoir and this difference in response can be accredited to fractures with high confidence [Couples, 2013]. For example, when well test permeability deviates remarkably from measured core plug permeability. The reason for this deviation can mainly be associated to the scale of observation. Whilst a core plug only measures centimeters at most and only measures matrix permeability, a well test measures the permeability for a much larger area. If this area is now heavily fractured, likely enhancing permeability, well test permeability will strongly deviate from that of the core plug. This chapter will first introduce the concept of a naturally fractured reservoir and the impact that fractures have on fluid flow. The different types of fractures and how and where they exist will be highlighted next. This is followed by a summary of methods used to model NFRs and close this chapter by introducing a synthetic case study for a naturally fractured reservoir utilised throughout this thesis.

## 2.1 What makes a reservoir a naturally fractured reservoir

The majority of rocks experience natural fracturing. Under the temperature and stress state of the earth's upper lithosphere, rocks achieve large strains by brittle failure, initiating fractures if the stress to which the rock is exposed lies beyond its elastic limits [Scholz, 2002]. In consequence, nearly all reservoirs are, to an extent, reservoirs with fractures. This leads to the statement by Narr [2006] that *"all reservoirs should be considered fractured until proven otherwise"*. Yet not all reservoirs necessarily need to be treated with the added level of complexity that NFRs bring. There are numerous definitions for what makes a naturally fractured reservoir, but according to Nelson [2001], *"A fractured reservoir is defined as a reservoir in which naturally occurring fractures either have, or are predicted to have, a significant effect on reservoir fluid flow either in the form of increased reservoir permeability and/or reserves or increased permeability anisotropy."* Therefore, considering fractures and their effect on reservoir fluid flow can potentially be invaluable to future reservoir management decisions. As in the early stages of reservoir development it is difficult to predict whether the impact of fractures on the reservoir's behaviour is positive, neutral or negative, the presence of fractures should always be closely monitored [Aguilera, 1987, Aliverti et al., 2003, Berkowitz, 2002, Dean and Lo, 1988, Gilman, 2003, Huskey and Crawford, 1967, Narr, 2006, Nelson, 2001, Pirson, 1953, Spence et al., 2014]. According to Nelson [2001], there are four different types of naturally fractured reservoirs, depending on the relative proportion of flow and storage capacity between matrix and fractures (figure 2.1). They range from type I reservoirs, where flow and storage capacity are dominated by fractures (e.g. fractured basement reservoirs), up to type IV reservoirs, where fractures only have a supportive function to the matrix that dominates flow and storage capacity. Fractured carbonate reservoirs normally lie in-between those two extremes (type II and type III), where flow and storage are controlled by both, matrix and fractures and the transfer between the two.

Countless laboratory experiments have demonstrated that if a rock is loaded, it will initially respond to the load by elastically deforming. If the applied load is below the rock's bulk yield point, the stored elastic straining energy is recoverable and unloading the rock will

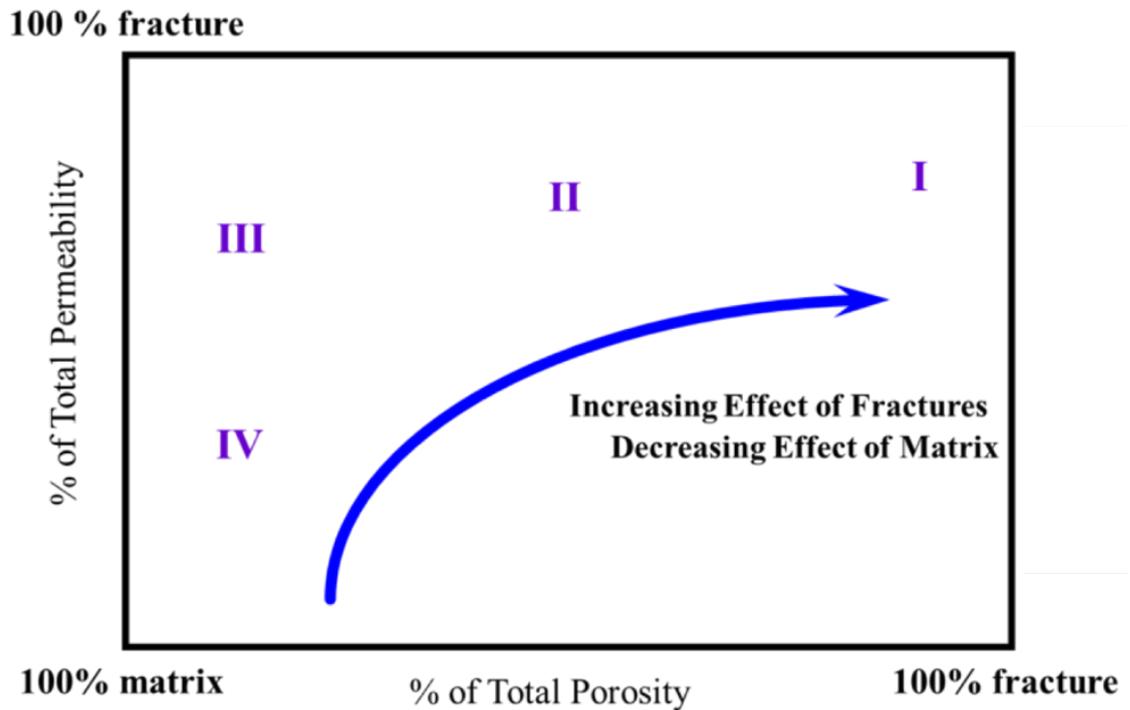


FIGURE 2.1: Classification of naturally fractured reservoirs after Nelson [2001].

allow it to go back into its initial state, leaving no permanent strain . The rock’s material properties are not changed. If the rock is loaded beyond its yield point, it will respond by developing irrecoverable volumetric plastic strain. The rocks material properties change. This permanent damage to the rock often expresses itself by creating localised deformation zones or fracturing with dilatational and compactional components as well as shear offset [Lewis et al., 2009, Stearns and Friedman, 1972, Tiab and Donaldson, 2004, van der Pluijm and Marshack, 2004]. By trying to understand better how fractures form, Griffith [1921] assumed that every material contains preexisting and unseen tiny flaws or "Griffith cracks". Schematically the cracks are elliptical but are also considered sharp at their tips. As the cracks have free surfaces, no shearing can occur across them. Thus principle stresses around the cracks are orientated perpendicular and parallel to the surface of the cracks. The stress state around the tips of the cracks are amplified, in theory to infinity, allowing for cracks to propagate and connect up to larger cracks even at small stresses if the stress the rock experiences exceeds its strength at the crack tips. Three end members describe the mode of fracturing a rock experiences (figure 2.2). Mode I fractures are opening/tensile mode fractures or joints with only little displacement. These fractures form parallel to

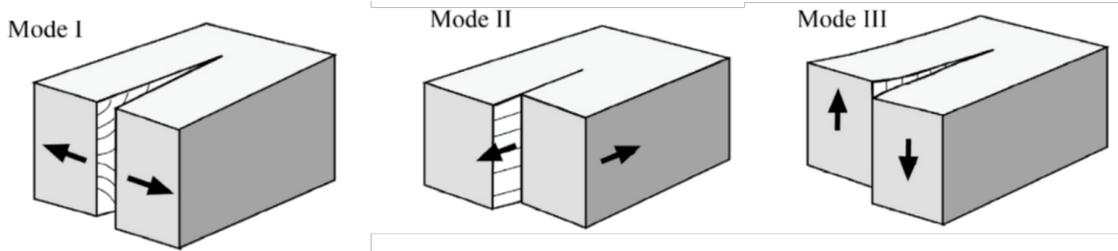


FIGURE 2.2: Model showing the three different modes of fracturing. Mode I being of opening mode and mode II and III of shearing mode (modified after Gudmundsson [2011]).

the maximum principal stress direction and perpendicular to the minimum principal stress direction [Zoback, 2007]. Mode II (sliding mode) and III (tearing mode) fractures can show fault-like offset patterns and show displacement along their plane. They form at an angle to the maximum principal stress directions.

Considering a fractures influence on fluid flow is vital for a better understanding of naturally fractured reservoirs. The cubic law can approximate fluid flow through a single fracture. The assumption is that a fracture’s surfaces can be idealised by two parallel planar plates separated by a constant distance and laminar flow conditions are present [Witherspoon et al., 1980] (figure 2.3 left). Under these conditions, fracture permeability can be described with equation 2.1:

$$k_f = \frac{a^2}{12} \quad (2.1)$$

where  $k_f$  is fracture permeability and  $a$  the fracture aperture [Witherspoon et al., 1980]. This highlights the importance of fracture aperture and its impact on flow through fractures. Consequently, fractures are inherently highly conductive compared to the surrounding rock matrix, resulting in the potential for channelised, high flow rates [Bourbiaux, 2010, Wennberg et al., 2016]. Deviations from this ideal case (planar parallel plates) lead to the reduction of flow and can significantly affect the hydro-mechanical behaviour of natural fractures (figure 2.3 centre). The so-called Joint–Roughness Coefficient (JRC), which alters effective fracture permeability with increasing fracture plane irregularity or roughness, can for example be used to account for this [Barton et al., 1985]. With hydro-thermal fluids flowing through initial open fractures, the growth of crystals can turn these fractures, into fully or partly sealing veins [Virgo et al., 2014] (figure 2.3 right). Fluid flow

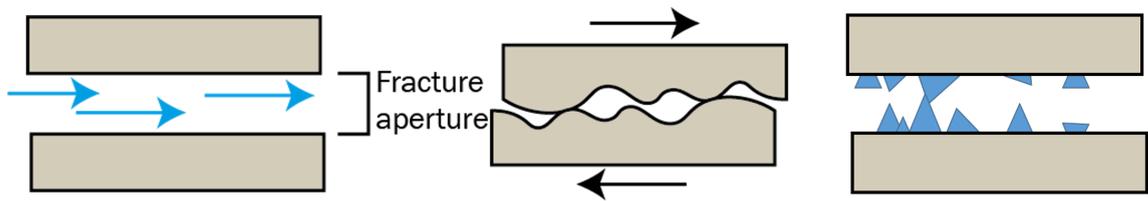


FIGURE 2.3: Left: Idealised fracture representation by two planar plates; centre: A more realistic, rough fracture surfaces negatively impacts fluid flow, but if the fracture experienced shearing, it can also prevent the fracture from completely closing under changing stress conditions; right: Partially sealed fracture with crystals precipitating on the fracture surface. These can also prevent fractures from closing.

along completely sealed fractures is non-existent and can cause fractures to act as barriers to flow. This has a strong impact on reservoir performance as it reduces fluid movement in some parts of the reservoir but can also be beneficial in certain production scenarios. If the process of sealing remains incomplete, for example caused by differential growth speeds of crystals along the fracture plane or partial cementation of gouge in shear fractures, the porosity within the fracture can be partly preserved [Hilgers and Urai, 2002]. These bridges, that can also be found in rough fractures, increase morphological variations along fracture planes and therefore the chance to maintain some fracture dilatancy when stress field orientation changes. For example, during the depletion of a reservoir the interaction and coupling between the fluid and the solid, make fracture aperture a dynamic and highly anisotropic fracture property that needs to be acknowledged in naturally fractured reservoirs [Baghbanan and Jing, 2008, Bisdom et al., 2017, Couples, 2013, Min et al., 2004].

At larger scales, the connectivity of individual fractures to fracture networks also plays a fundamental role in fluid flow in NFRs [Bonnet et al., 2001, Phillips et al., 2020]. Outcrop observations for example show that fractures can be grouped into so-called fracture sets with a defined hierarchical order. A fracture set consists of fractures showing similar orientations and geometric characteristics and most likely formed under similar stress conditions relative to their orientation [Odling et al., 1999, Van Golf-Racht, 1982]. Considering a host rock that only contains a single fracture set, intersections between individual fractures are scarce. However, a host rock with numerous fracture sets will have countless intersections between fractures and substantially increase the connectivity and flow behaviour of a reservoir or storage site.

We can summarise that fractures or fracture networks can affect the economic viability of a hydrocarbon or geothermal reservoir and the feasibility of storage sites in various ways [Aliverti et al., 2003]. Mineralised fractures turned into veins, deformation bands, stylolites or gouge-filled shear fractures can act as baffles or seals and cause heavy compartmentalisation. This has the potential to turn reservoir recovery uneconomic. Open or partially open fractures substantially enhance reservoir conductivity and can make a reservoir economically viable in the first place. However, they also introduce an additional layer of material complexity by heavily localising flow. This can lead to substantial performance contrasts of different wells, with a fraction of all wells responsible for the majority of the production. The consequences are steep production spikes and early water breakthroughs along streaks of high fracture permeability. This has the potential to leave large proportions of the reservoir unswept behind the water front [Allen and Sun, 2003, Narr, 2006]. Additionally, production induced pressure alterations change the effective stress fractures experience and can lead to modifications of fracture aperture and flow potential [Mäkel, 2007]. Here, partly sealed veins and fractures with fracture planes with substantial morphological variations can help propping up openings, despite changes in the stress field, resulting in better reservoir recovery [Mäkel, 2007]. Independent, of what the effect of natural fractures on the investigated reservoir turns out to be, their potential impact on reservoir behaviour and its commercial viability, show the importance of the quote from Narr [2006] at the beginning of this section regarding naturally fractured reservoirs. It is fundamentally important to identify the impact of existing fractures on the reservoir's dynamic behaviour.

## **2.2 How and where do natural fractures occur?**

After concluding that fractures can significantly impact reservoir performance, we need to understand how to characterise fractures based on their key attributes, the presence of several fracture networks and their potential distribution in the subsurface. Fractures can be observed at all scales, ranging from tiny fissures only visible under the microscope, fractures at core and outcrop scales, and even reservoir to basin scale [Bonnet et al., 2001]. Given their multi-scale nature, numerous direct and indirect measuring techniques have to be compiled to understand fracture distributions in the subsurface. Examples of

potential fracture data sources are borehole images, outcrop data, numerical simulations, conceptual models, laboratory experiments, production logging tools, drill stem tests, and seismic data (e.g. [Bisdom, 2016, Couples and Lewis, 1999, Couples et al., 1998, Forbes Inskip et al., 2018, Hall et al., 2007, Hardebol et al., 2015, Lewis and Couples, 1993, Maerten and Maerten, 2006, Mäkel, 2007, Nelson, 2001, Odling et al., 1999, Virgo et al., 2014, Watkins et al., 2015, Wilson et al., 2011]). The observations made from all these combined sources allow us to recognise that fractures and fracture distributions follow specific trends and patterns. Fractures are either associated to localised deformational events (e.g. associated with faults and folds) or regional deformational events (e.g. related to plate tectonics). Examples for both are highlighted in the following:

### **Fractures related to regional events**

Fractures that formed due to large-scale events (e.g. host rock burial or uplift) that change the regional stress regime represent the most common fracture type. Due to their origin from large-scale, tectonic events, the resulting fracture patterns remain constant over large areas [Lorenz et al., 1991]. An example can be seen in the right image of figure 2.4. Here, most fractures follow a similar orientation and only a subset of fractures developed at an acute angle to these. They tend to show little to no offset and extend perpendicular to the rock bedding (e.g. bottom left image in figure 2.4). They can be meters to 100s of meters long, whilst their height is often bound to stratigraphic units. According to Odling et al. [1999] this is because different units or layers can be mechanically decoupled from each other if laterally continuous bedding planes or thin shale horizons occur (left sketch figure 2.4). Another possibility is that layers show a mechanical strength greater than the stresses the rock experienced, causing them not to fracture and hinder further fracture propagation. Additionally, different stratigraphical units can have different mechanical properties, leading to different fracture patterns developing in each unit (figure 2.4 sketch in centre) [Ericsson et al., 1998, Fay-Gomord et al., 2018, Gauthier et al., 2002, Zahm and Hennings, 2009].

### **Fractures related to localised events**

Fractures of this origin can be assigned to localised deformational events and are therefore spatially much more limited than fractures associated with regional events. If present,

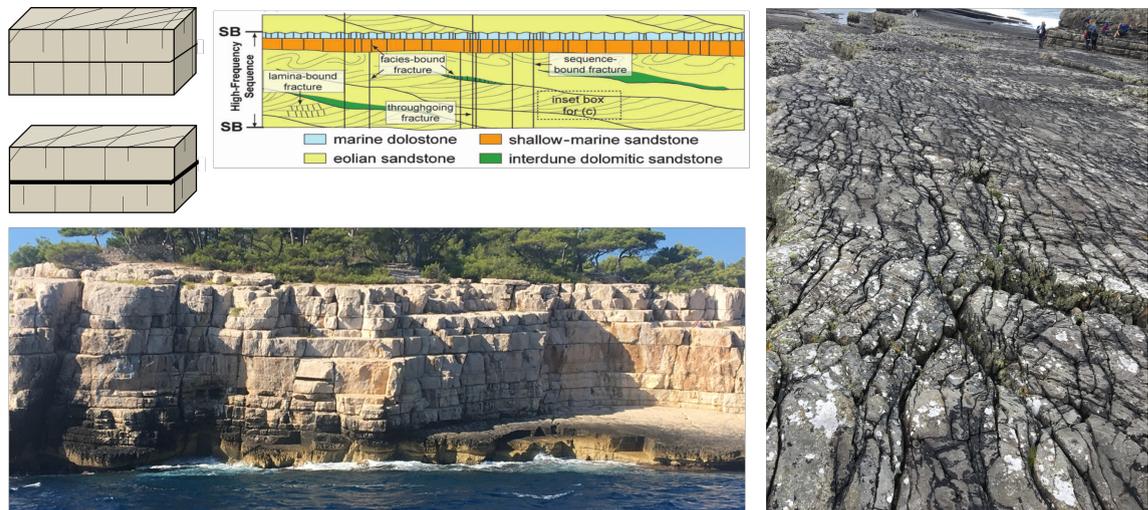


FIGURE 2.4: Top left: Schematic illustration showcasing how a thin shale layer (thick black line) can inhibit fracture propagation. Top centre: Schematic illustration of fracture hierarchy within different lithologies after Zahm and Hennings [2009]. Bottom left: Fractured carbonates with bed-bound fractures as well as fractures cutting through the entire outcrop section (Cassis, France). Right: regional fractures in Ross formation, SW Ireland.

they have a substantial impact on the flow behaviour of a reservoir. Examples of localised deformational events are folds and faults.

*Fault related fractures:* Fault related fractures typically form parallel to the fault and are mostly contained to the damage zone surrounding the fault ( e.g. figure 2.5 top left sketch). The density of fracturing away from the fault has been recorded to decline logarithmically, exponentially or linearly with distance [Caine et al., 1996, Faulkner et al., 2010, Vermilye and Scholz, 1998], but is also strongly dependent on the mechanical properties of the surrounding lithologies as well as the lateral and vertical extent and the displacement of the faulting event. This leads to fault related fracturing developing away from the fault ranging from centimeters to 100s of meters. Two examples for this are shown in figure 2.5. The fault in the bottom left is a normal fault with substantial amounts of deformation materialising in its hanging wall (right side of fault). The majority of the formations in the foot wall (left side of fault) are very rich in clay and show little signs of fracturing. Apart from several thin sandstone beds, the mechanical properties of the lithologies prevent any brittle deformation. A different observation can be made in the the right image of figure 2.5. In the carbonate rocks present here, we can see heavy fracturing within close proximity to the fault as well as a damage zone extending several meters around the fault. When



FIGURE 2.5: Top left: schematic illustration of a fault and the surrounding, heavily fractured damage zone; Bottom left: Normal fault with little deformation in the foot wall and intensive deformation in the hanging wall, Billefjorden, Svalbard. Right: Fontaine-de-Vaucluse fault zone in southern France. Intensive fracturing of the carbonate rocks can be observed in close proximity to the fault.

comparing the high fracture intensity in this zone with that of the surrounding lithologies, the localised impact of these deformation events on the host rock becomes apparent.

*Fold related fractures:* Many studies have been undertaken to understand how folding affects fracture development, and numerous conceptual models try to capture fold related fracturing [Lewis and Couples, 1993, Price, 1966, Stearns, 1964, Watkins et al., 2015]. Two examples for conceptual models for fold related fracturing are shown in figure 2.6. The fracture distributions expected strongly depend on the folding the rocks experience and the lithologies present within the fold. In general, however, horizontal shortening causes the fold's crestal or hinge zone to experience a considerable proportion of the fold related straining the rocks experience. This results in a large proportion of fracturing happening in this region. Fracture intensity is high, and fractures tend to be sub-parallel to the fold axis. On the flanks and away from hinges, fracture distributions are less predictable, generally lower and can be expected to be more localised with shear fractures present to accommodate the straining. In figure 2.6 we can see an example of a relatively small fold that experiences the high intensity of fracturing in the crestal region of the fold.

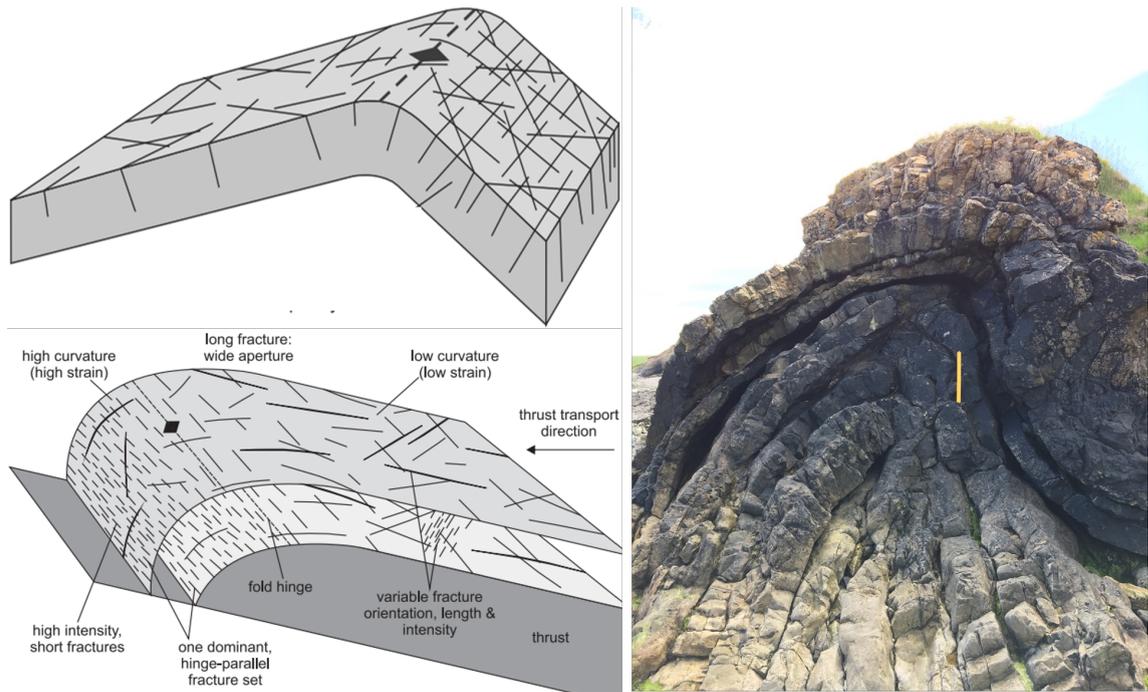


FIGURE 2.6: Left: Conceptual models by Lewis and Couples [1993] (top) and Watkins et al. [2015] (bottom) trying to explain fold related fracturing. Right: Example of a fold and associated fracturing, Spittal Beach, Berwick-upon-Tweed.

*Fracture corridors:* Fracture corridors, also called fracture swarms, are tabular, very localised zones (meters) that show a very high intensity of fracturing compared to the surrounding rock [de Joussineau and Petit, 2021, Questiaux et al., 2010]. They tend to be sub-parallel and have little displacement along individual fractures (figure 2.7). Their origin is occasionally linked to local structural or sedimentary heterogeneity, and it is thought that they can sometimes present the most efficient way to accommodate strain. Nevertheless, fracture corridors remain hard to explain, let-alone predict.

## 2.3 Modelling and flow simulation of naturally fractured reservoirs

Over the last decades, numerous modelling and simulation approaches have been developed for NFRs. A fundamental challenge for modelling naturally fractured reservoirs is the contrast between fluid behaviours in the matrix and the fractures. In open fracture networks, fluid flow in fractures is usually orders of magnitude higher than in the matrix, whilst the vast majority of the storage space is kept within the matrix and not the fractures [Matthäi

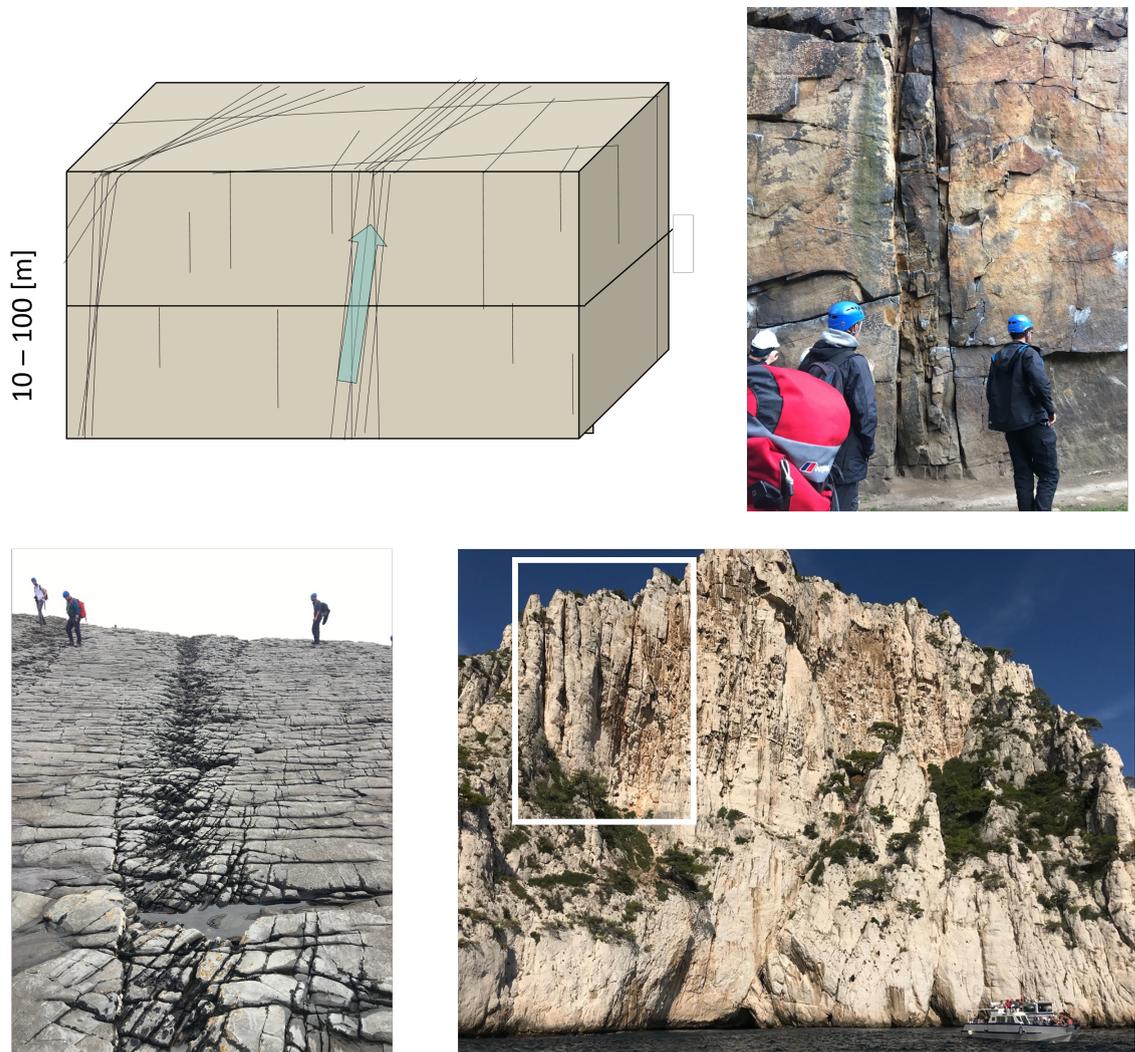


FIGURE 2.7: Schematic sketch of fracture corridor (top left) and different examples for fracture corridors as observed in different outcrop settings.

and Belayneh, 2004]. Modelling approaches that tackle this challenge can loosely be split into methods where fractures are either represented implicitly by an effective continuum or explicitly where fractures are explicitly represented as geometric features within the model [Berre et al., 2019]. Explicit methods are Discrete Fracture Network (DFN) and Discrete Fracture Matrix (DFM) models. One of the downsides of explicit methods is that they take significant amounts of time to build (e.g. data gathering) and simulate (e.g. meshing) [Ringrose and Bentley, 2021]. In implicit methods, fracture networks are upscaled to a continuous medium representing the underlying fracture properties. Upscaling intends to have a continuous medium that shows the same macro-scale behaviour one would expect from the explicitly modelled fractures but for a fraction of the computational cost [Sævik and Nixon, 2017]. Implicit methods can be subdivided into single porosity, dual-porosity

and dual-porosity-dual-permeability models. A brief summary of the methods mentioned above and upscaling methods used to turn explicit fracture representation into implicit ones will be given in the following.

In the subsurface, fractures can be immeasurably abundant, making reliable predictions on fracture distributions a challenging task. However, outcrop observations show that a large proportion of fractures can be grouped into so-called fracture sets that all show similar orientations and geometric characteristics. This makes describing fractures easier as we can assign an adequate range of lengths, heights, orientations, dispersions and apertures to fractures within each fracture set. Additionally, a fracture intensity (P32) is assigned to each fracture set and describes the area of fractures per unit volume of rock mass [Dershowitz and Herda, 1992]. Reliable sources required for the accurate description of fracture sets on the other hand are (i) frequently sparse because of the high expense for data collection and (ii) the data collection itself already carries a significant degree of uncertainty within itself [Santos et al., 2015]. Therefore, statistical approaches usually describe fracture networks in the subsurface. If the flow in a reservoir is dominated by the fracture system and the matrix permeabilities contribution to flow is insignificant, naturally fractured reservoirs can be modelled with the help of so-called Discrete Fracture Network (DFN) models [Dershowitz et al., 2004, Hyman et al., 2015]. DFNs represent stochastic realisations of fracture networks based upon input parameter ranges used to describe fracture sets. Simulating flow through an immeasurable number of fractures on reservoir scale turns into a computational challenge and therefore the application of DFNs is usually restricted to smaller scales. DFNs are usually upscaled to equivalent continuous media for reservoir scale simulations, but more on this later.

Another approach for explicitly modelling fractures in naturally fractured reservoirs is the Discrete Fracture Matrix (DFM) approach [Geiger and Matthai, 2014, Kim and Deo, 2000, Obeysekara et al., 2018]. In contrast to the DFN approach, fluid flow through both the fractures and the matrix is taken into account. A benefit of the DFM approach is that fluid transfer between fracture-fracture, matrix-fracture and matrix-matrix systems is modelled without the need for transfer functions that attempt to compensate for this missing interaction. The geometric nature of fractures will however in most cases require DFMs to be modelled with unstructured meshes which comes with a substantial computational

overhead [Berre et al., 2019]. Therefore, this method is not suitable for reservoir scale simulations but can be used to calibrate continuum models [Lemonnier and Bourbiaux, 2010].

The advantage of implicit continuum methods over the above described explicit methods is that fractures are not directly modelled, making them computationally more efficient. This makes them particularly attractive for practitioners required to model and simulate fractures on the reservoir scale. Additionally, the implicit representation prevents them from having to define exact fracture locations. Instead, fracture networks are upscaled to continuous media with the help of upscaling techniques. Before describing the continuum methods used for modelling naturally fractured reservoirs, I will give a brief overview of the most popular methods for upscaling fracture permeabilities: the Oda method (analytical) and flow-based upscaling (numerical).

Flow-based upscaling is a numerical approach where a fluid injection under various possible boundary conditions (e.g. linear, constant or episodic pressure) is simulated through the generated fracture network [Durlofsky, 1991]. The pressure differential between inlet and outlet in comparison to the flow rate of the injected fluid then allows us to calculate the effective permeability of the fracture network with equation 2.2:

$$K_{eff} \iiint_{\Omega} \nabla p dV = \iiint_{\Omega} K_m \nabla p dV + \iint_{\Gamma} T \nabla p ds \quad (2.2)$$

where  $K_{eff}$  and  $K_m$  represent the effective and matrix permeability, respectively,  $T$  the fracture transmissivity,  $\nabla p$  the fluid pressure differential between the fluid inlet and outlet,  $\iiint_{\Omega} dV$  represents the integration over the grid block volume and  $\iint_{\Gamma} ds$  the integration of every fracture surface contained within the grid block [Durlofsky, 1991, Sævik and Nixon, 2017]. However, solving this problem numerically on reservoir scale is too time-consuming and impractical in reality [Ahmed Elfeel and Geiger, 2012].

Therefore, Oda [1985] developed a method that by introducing several simplifying assumptions, tries to find a permeability tensor that represents the fracture network analytically. The assumptions are that matrix permeability is negligible, fractures are perfectly planar

and extend to infinity [Sævik and Nixon, 2017]. With these assumptions equation 2.2 can be reduced to equation 2.3:

$$K_{eff} = \frac{1}{V} \sum_{i=1}^N T_i A_i (I - n_i n_i) \quad (2.3)$$

where  $N$  represents the sum of all fractures  $i$  within the grid block with volume  $V$  and  $T_i$ ,  $A_i$  and  $n_i$  are properties of fracture  $i$ .  $T_i$  is the fracture transmissivity,  $A_i$  the fracture area,  $n_i$  the unit normal vector for the fracture tangential plane.  $I$  represents the identity matrix and  $n_i n_i$  denotes an outer tensor (dyadic tensor) [Oda, 1985, Sævik and Nixon, 2017]. However, the assumption that fractures extend to infinity has the consequence that nearly all fractures are connected, independent of their length or position in the grid block, which can lead to overestimations of fracture permeability. The Oda Gold approach, an extension to the Oda approach, tries to assess fracture connectivity within the grid block and take it into account during the upscaling procedure [Elfeel et al., 2013, Golder-Associates, 2018]. Here, fractures below a connectivity threshold are not considered in the permeability calculations.

The simplest implicit continuum method is the single porosity approach, where both, the matrix and the upscaled fracture network are lumped together to a pseudo-matrix [Berre et al., 2019]. This means that all properties, such as porosity and permeability, are shared among fractures and the matrix. The main advantage of the single-porosity approach is its computational efficiency that comes from neglecting small scale details, which allow for reservoir scale simulations. However, the simplicity of this approach also comes at the cost of potentially representing flow in naturally fractured reservoirs with lower accuracy than representing them explicitly. This is mainly because continuum models' underlying requirement is to have a representative elementary volume (REV). In other words this means a volume above which the properties of the volume under investigation are stationary. For the REV requirement to hold, there has to be a scale at which individual features can be ignored and averaged to a continuum. This may not be valid for fracture networks due to their potential multi-scale nature and must be evaluated case-by-case [Bear, 1972, Berkowitz, 2002, Elfeel et al., 2013, La Pointe et al., 1996, Odling et al., 1999].

The most common continuum approach for modelling naturally fractured reservoirs is the dual-porosity approach [Barenblatt et al., 1960, Warren and Root, 1963]. Here, fractures are not lumped together with the matrix but are represented as a separate porous continuum. The primary matrix function is to hold a large volume of storage capacity and a nominal flow capacity. On the other hand, the fractures have an insignificant storage capacity but a tremendous flow capacity. The majority of the flow, therefore, happens within the fractured system that interacts with the matrix system via transfer functions that attempt take into account the mass exchange between the matrix and the fracture system [Kazemi et al., 1976]. The matrix blocks themselves can however not react with each other. In cases where matrix-matrix interactions also become important, for example if transport processes in the matrix play a more substantial role, dual-porosity models have also been extended to dual-porosity-dual-permeability models [Lemonnier and Bourbiaux, 2010]. This increase in complexity for both methods compared to the single-porosity continuum approach of course comes at the expense of computational efficiency and additional required preparational work.

In this work I will be using the single-porosity continuum approach to represent fractures and matrix and perform upscaling with the of the Oda-gold method. Further details follow in the upcoming chapters.

## **2.4 Setup of synthetic naturally fractured reservoir**

The previous sections identified the added complexities that differentiate a naturally fractured reservoir from a conventional reservoir and have to be considered for successful reservoir management. I outlined how fractures might be distributed in the subsurface and the reasons for these distributions and how this can help us make fracture distribution predictions. We also covered different approaches that allow us to capture the behaviour of naturally fractured reservoirs in geological models and flow simulations. As the objective behind this work is to come up with novel workflows that in-cooperate geological, fracture-related uncertainties and their impact on fluid flow into modelling procedures, we decided to set up a synthetic case study of a naturally fractured reservoir that we will be testing our developed workflows on throughout the thesis.

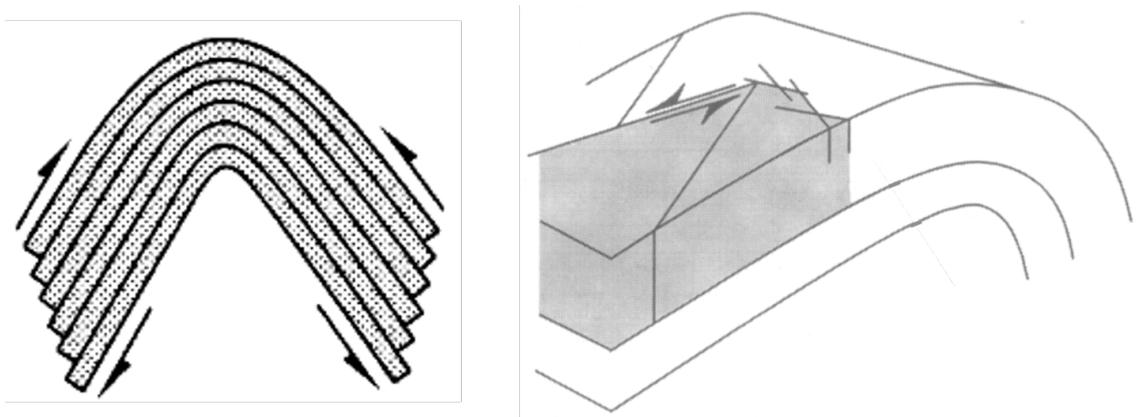


FIGURE 2.8: Left: Schematic sketch showing flexural slip folding after Ramsay and Huber [1987]. Right: Schematic illustration showcasing localised slip of mechanical units resulting in fracturing in fold flanks after Couples et al. [1998].

We decided to set up a case study that depicts an anticlinal fold with fracturing inspired by the conceptual geological model for the Teton anticline, Sawtooth Range, Montana, USA by Stearns [1964]. This early work and the work by Price [1966] have been the basis for geologists to describe fold related fracturing over the past decades [Burberry et al., 2019, Couples et al., 1998, Ghosh and Mitra, 2009, Stearns and Friedman, 1972]. According to one of the most recent investigations by Burberry et al. [2019] the geometric relationship and the development of the fracture sets occurred over at least four separate events and is linked to thrust-related folding. The first fracturing stage is associated with far-field stresses and produces two perpendicular fracture sets to the axis of the anticline that is yet to be formed. The second and third stage of fracture development is then associated with the actual formation of the anticlinal fold. Fractures formed in these stages form parallel to the fold axis in the crestal part and parallel to the fractures created during the first stage, linking them up along the flanks. The fourth stage of fracturing happens during the further tightening of the fold. It causes the formation of different longitudinal fractures but also reactivation movement of favourable orientated faults and fractures on the flanks.

Another concept for fracture distributions we want to consider in our case study is based on flexural-slip folding and the work by Couples et al. [1998]. Unlike the work by Burberry et al. [2019] that relates tectonic events to different consecutive fracturing events, Couples et al. [1998] address the question why certain types of fractures form under certain circumstances and use this knowledge to make predictions about fracture distributions. To get a good understanding of what is meant by flexural slip folding, we first move away from

rocks and simplify the matter by looking at a deck of cards placed on a table. If we start pushing the deck of cards together from both sides, we will (i) see the cards bulging or folding upwards and (ii) also observe slip occurring between individual cards. Something comparable can also be observed in nature within folded rock sequence [Geoff Tanner, 1989] ( e.g. figure 2.8 left). During the folding event, rock layers get divided into mechanical units, partitioned by active slipping surfaces like for example the interface between two lithologies. This partitioning event also causes bending related strain, which expresses itself through fracturing, to be bound to individual mechanical units [Couples et al., 1998]. Couples et al. [1998] also recognised that bedding-plane slip events often occur in a more localised or "patchy" fashion rather than happening uniformly (figure 2.8 right). Depending on the structural position within the fold, the rocks respond to straining with different types of fracturing. Fractures related to the bending stress state within each mechanical unit are relatively short, widely distributed, bound to mechanical units and are in an angle of 30 – 60 to the boundary of the mechanical unit and are often referred to as type 3a and 3b fractures (pink fractures in figure 2.9). Another type of fracturing is also related to the bending the mechanical unit experiences but is more dominant in the crestal zone and develops fractures parallel to the fold axis (purple fractures in figure 2.9). They are also rather short and form perpendicular to the mechanical unit. The last set of fractures is more dominant on the flanks and can be associated to the localised, slipping events along boundaries of mechanical units (blue fractures figure 2.9). They are substantially longer than the other fractures, normal to bed boundaries and greater angle to the fold axes. Fracture corridors are occasionally also associated with this type of fracturing. Couples et al. [1998] also mention the successive development of new slip surface and mechanical units throughout the folding process leading to further subdivision of previous mechanical units and the overprinting of existing fracture patterns and overall more complex fracture networks.

The model used in this study is inspired by the Teton anticline model developed by Stearns [1964] and Burberry et al. [2019] as well as the observations made by Couples et al. [1998] on fracturing related to flexural-slip folding. The model measures 2000m x 1000m x 35m and comprises low-quality carbonate rocks (low porosity and permeability) and seven mechanical units (each roughly five meters in height) with bed-bounding fractures that are

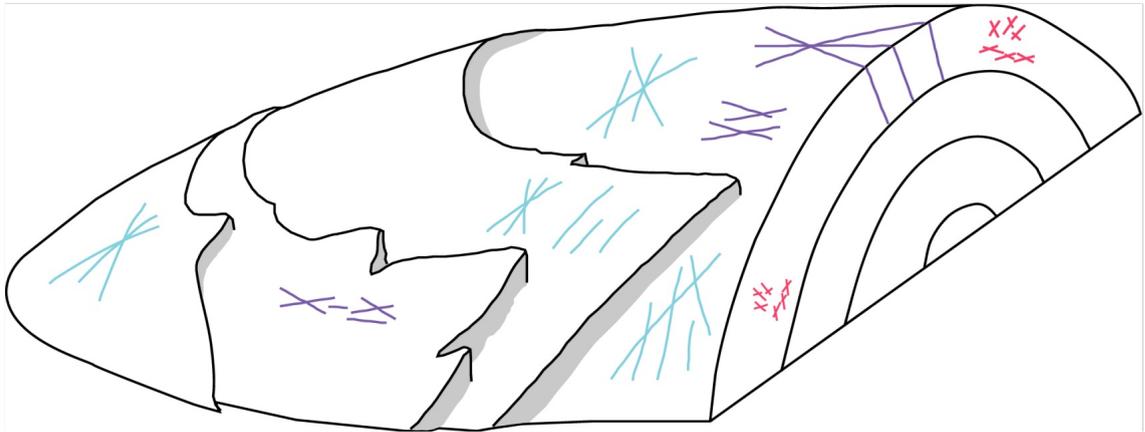


FIGURE 2.9: Fracture model for the Teton anticline after Stearns [1964].

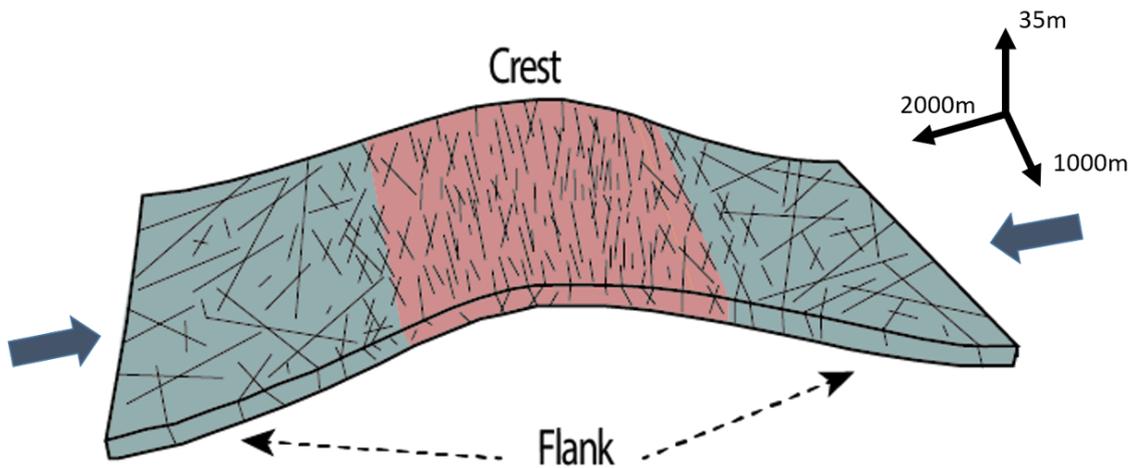


FIGURE 2.10: The conceptual fold model based on the Teton anticline model for fold-related fracturing. In the red zone around the fold centre, crestal fractures are short and parallel to the fold axis. In the green zone, fractures are longer, more dispersed and oblique to the fold axis. Note that this represents one of seven mechanical units with bed-bound fractures.

separated by slip planes (figure 2.10). The model only consists of the horizontal-extension dominated upper part of the Teton anticline model and shows a minimum of 2 distinctive folding related fracture patterns. In the fold crest, fractures in alignment with the fold axis are of short length but high intensity and low dispersion. More extended reactivation fractures characterise the flank orientated perpendicular to the fold axis and show a higher degree of disorder and lower intensity. Throughout the fold, background fracture sets can also be present related to the regional pre-folding events. It should be noted, that the synthetic case study developed here, is a simplification of potential fracturing histories that can be observed in real naturally fractured reservoirs but will be sufficient and geologically realistic enough to test out the workflows develop within this work.

# Chapter 3

## Methodology

This chapter gives a brief overview of the methods used throughout the thesis. I will first introduce the concept of multiple-point statistics and its advantages over other spatial property modelling techniques. Next, I will discuss flow diagnostics, a proxy model for fluid flow simulations. Entropy, a measure from information theory used to evaluate the model ensemble diversity, is introduced next. Then I present the particle swarm optimiser (PSO) - an optimisation technique used to guide geological model building. This is followed by dimension reduction and clustering techniques that are used to reduce the model ensemble for manageable uncertainty quantification. The chapter ends with a brief discussion about the concept of agent-based modelling.

### 3.1 Multiple-point statistics

In many cases, the spatial complexity of geological features that need to be modelled can be captured only to a limited extent by object-based modelling techniques [Journel et al., 1998] or conventional variogram-based dual point statistics, such as sequential indicator simulation [Cosentino, 2001].

Variogram-based geostatistical methods were first introduced by Matheron [1963] and came to the petroleum industry from the mining industry, where they are applied with great success to populate porous media properties over a volume of rock. However, one difference between the industries is the abundances of data. It is common to drill wells in a

grid-like pattern to mark out the boundaries of ore bodies to be processed in mining. This is not the case in the petroleum industry, leaving geomodellers only with a small selection of core and log data from drilled wells and "soft-data" (e.g. seismic) to build reservoir models representing the subsurface. Another difference is the purpose of modelling. The mining industry is interested in estimating the grade and limits of ore bodies to be mined. In contrast, the purpose of a reservoir model is to capture geological features that dictate the flow response of the reservoir [Caers and Zhang, 2002]. This becomes problematic when curvilinear features such as channel-bodies are supposed to be modelled with a technique that describes the correlation between only two spatial locations. In particular geobody connectivity, which is of great importance for fluid flow, might not get captured. Alternative modelling techniques, such as object-based modelling could be utilized here, but these are also limited by the simplicity of the generated shapes [Cosentino, 2001]. Consequently, the dynamic flow behaviour of the reservoir can be strongly disturbed and lead to costly decision-making for future reservoir development.

A more recent development in texture-based geostatistical modelling is multiple-point statistics (MPS), which better preserves geological patterns [Caers and Zhang, 2002, Strebelle, 2002]. Briefly, MPS algorithms can be described as scanning so-called training images for textures that are then used to populate the reservoir model with properties such as facies type and conditioned the populated patterns to the observed data. These training images represent a way to integrate more realistic and complex spatial correlation in geological patterns into the model, while a variogram limits spatial correlation to two-point statistics. Training images are essentially conditioned to the field data and can be obtained from outcrops, modern-day analogues, even conceptual and physics based model that preserve key spatial patterns, connectivity and relationships between geobodies [Strebelle, 2002]. MPS first entered the stage with the work by Guardiano and Srivastava [1993] and Journel [1993]. Due to computational bottlenecks MPS remained a theoretical concept without practical application throughout the 1990s and was limited to small 2D cases. The first MPS algorithm that applied to real-world problems and is still widely in use is *snesim* by Strebelle [2002]. Today it is one of many MPS algorithms designed to create realistic geological models of the subsurface [Arpat and Caers, 2005, Guardiano and Srivastava, 1993, Tahmasebi et al., 2012, Zhang et al., 2006].

### 3.1.1 The training image

One of the biggest challenges for the successful application of MPS is the training image. The best MPS algorithm will always be limited by the quality of the training image that it samples from. Apart from existing conditioning data, all essential information on the underlying geology of the reservoir of interest has to be stored in the training image. It acts as a data silo for geological patterns that the MPS algorithm then samples from with the advantage of relying on multi-point statistics instead of variogram-based two-point statistics to describe geological heterogeneity [Caers and Zhang, 2002]. This makes it possible to generate reservoir models with more complex geological features such as curvilinear channel systems. Additionally, sketching outcrops and conceptual geological models are a fundamental part of every geoscientist's training. It therefore comes more naturally to any geoscientist or geomodeller to translate their understanding of the subsurface into a training image than into a variogram - based geostatistical method. They can immediately see which geological patterns a reservoir model will contain. However, this also comes at the price of being less of a statistical representation of the given data and opens the space up for a more subjective interpretation of all available data [Mariethoz and Caers, 2014], which can be a blessing or a curse. Other issues that can arise are artifacts and conditioning issues when there is a conflict between a training image and existing conditioning data. It is therefore useful to have a set of differing training images available.

Training images can be derived in many ways. Off the shelf training image portfolios for fluvial and carbonate environments can be found online [Jung et al., 2010, Mariethoz and Caers, 2014]. If a training image needs to be generated from scratch, outcrop data, like photographic images can also make good training images if it is possible appropriately capture the geobody of interest [Bruna et al., 2019]. They represent the geological ground truth and can be helpful to understand the spatial architecture and connectivity of different features in two dimensions [Tahmasebi, 2018]. Object-based or Boolean modelling can also be used to generate training images [Damsleth et al., 1992, Deutsch and Wang, 1996]. Predefined shapes (e.g. channels) are iteratively distributed in a 3D volume and can be controlled with different geometric parameters, such as width, sinuosity, orientation or thickness. They represent an easy to implement yet powerful

solution to generate training images. Another possibility is to use process-based modelling [Koltermann and Gorelick, 1992, Lancaster and Bras, 2002, Lopez et al., 2009, Mullins et al., 2021]. Geological process modelling tries to mimic the physical processes that control the deposition or growth of porous media. While generating realistic models, these simulations are computationally demanding. They require parameter fine-tuning (e.g. sediment influx rate, sea-level fluctuations) to generate a training image that complies with the modellers' expectations. If available, high resolution seismic data can also be used as training images [Mariethoz and Caers, 2014]. To avoid redundancy effects, it is however important that the seismic data differs from the conditioning data. Independent of the technique used, training images should not be an exact, detailed representation of the particular subsurface but should only capture the essence of the geology conceptually to contain a sufficient amount of training pattern replicas to derive pattern statistics from it. Otherwise, too many specific data events will cause the MPS algorithm to generate reservoir models with little variability or unrealistic patterns [Strebelle, 2002].

### **3.1.2 Different multiple-point statistics algorithms**

As with every problem, there is no one size fits it all MPS algorithm. Depending on the issue at hand, different algorithms have their advantages and disadvantages. They all try to capture and replicate the features of a training image. Some do this in a probabilistic way, others in a more pattern-based style. The three most popular MPS algorithms will be summarised in the following passages.

#### **3.1.2.1 Probabilistic MPS algorithm: Snesim**

As already mentioned in section 3.1, the *Single Normal Equation Simulation* - or *snesim* algorithm developed by Strebelle [2002] was the first truly applicable MPS algorithm. It solved the problem that made the predecessor algorithm by Guardiano and Srivastava [1993] impractical. In the earlier version for every new iteration, their algorithm repeatedly inspected the entire training image, causing a too high CPU overhead. This problem is solved by storing relevant data from the training image (e.g. the number of occurrences of a data event/ data template) in a search tree structure that can be revisited every iteration

[Knuth, 1997, Strebelle, 2000]. This circumvents scanning the entire training image more than once.

Similar to sequential indicator simulation, the *snesim* algorithm is a sequential algorithm [Cosentino, 2001]. Following a predefined random path, it sequentially visits each unknown grid cell and assigns a value. This value is worked out by first searching for conditional data in a predefined neighbourhood template around the unknown value. Then the conditional probability of each property occurring at the unknown location is calculated based upon the data templates stored in the search tree as well as conditional well or seismic data. Finally, the algorithm samples from that conditional probability density distribution to assign a value to the cell. Each newly generated grid cell turns into conditioning data for the remaining unknown cells.

It should also be noted that *snesim* was developed only to simulate binary or categorical properties like geological facies distributions. It is therefore unsuitable for the modelling of continuous petrophysical properties. Another downside of *snesim* is that it tends to lose information on larger scales if hard conditioning data, such as well data are included [Arpat and Caers, 2005]. The reason is that *snesim* will go through the entire search tree structure to find events that exactly match the hard data. This can lead to the dropping of hard conditioned grid cells, causing the loss of large-scale information and the generation of artificial features. To adequately counteract this occurrence and capture all elements present in the training image, it is recommended to be twice the size of the largest occurring characteristics [Arpat and Caers, 2005, Caers and Zhang, 2002, Zhang et al., 2006].

In figure 3.1 we can see what an unconditional *snesim* realisation (right) from a channel system training image (left) looks like. Curvi-linear meandering channels are well preserved, crosscut each other and the channel connectivity is comparable to that of the training image.

### **3.1.2.2 Pattern-based MPS algorithms: Simpat and Filtersim**

Building up on the work from Strebelle [2002], on probabilistic MPS algorithms, a group of more pattern-based MPS algorithms such as *simpat* and *filtersim* emerged. *Simpat* and *filtersim* try to enhance the replication accuracy of patterns found on training images [Arpat

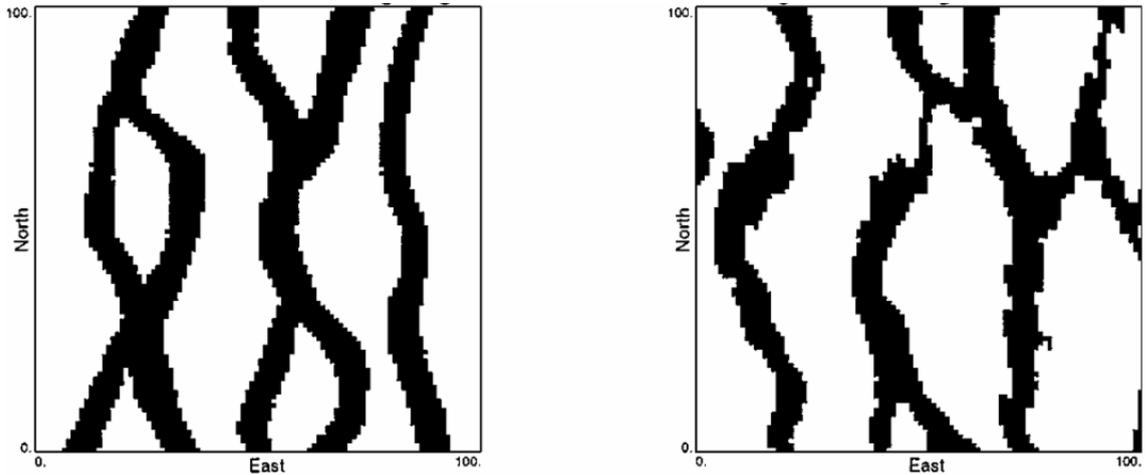


FIGURE 3.1: Left: Channel system training image; Right: Unconditional *snesim* realisation [Strebelle, 2000].

and Caers, 2005, Zhang, 2006]. They do not try to sample individual data events from conditional probability distributions but scan the training image with a predefined template for patterns of similarity that are then grouped to create a pattern database [Palmer, 1999]. With the help of filters (e.g. linear combinations of pixel values, representing statistical properties such as mean, gradient or curvature) the number of patterns to select from each class is reduced to prevent repetitions. Each class is then represented by a prototype pattern that represents that class well.

After completing this preprocessing step, the algorithms sequentially follow a random path, visiting each empty grid cell. A random pattern from the prototype group that best fits the surrounding conditioning data is chosen and pasted onto the model at each cell. A simple comparison would be to describe *"the process as being similar to constructing a jigsaw puzzle drawing from stacks of similar-looking pieces."* [Remy et al., 2009]. Therefore, a variogram model is not reproduced, but the pattern extracted directly from the training image represents the multiple-point statistics [Zhang et al., 2006]. If this pattern is non-stationary or unique, the algorithms also replicate these features on the simulation grid, slightly reducing the assumption of stationarity as is the case with *snesim* [Honarkhah and Caers, 2010].

Unlike with *snesim*, pattern-based MPS algorithms do not perform any kind of morphing, rotation or filtering on the extracted patterns to make sure that the replication accuracy to the training image is high. This however also comes at a cost. By anchoring the algorithm

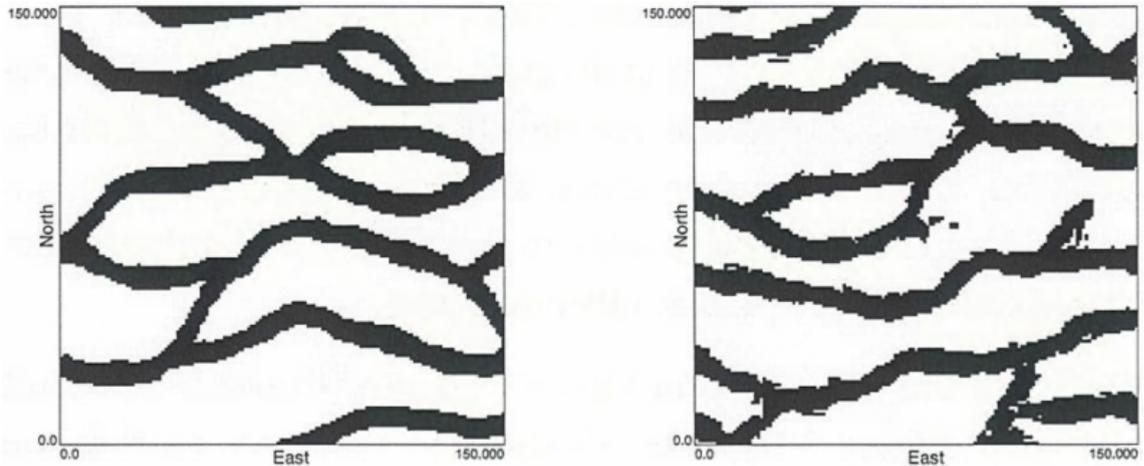


FIGURE 3.2: Left: Channel system training image; Right: Unconditional *filtersim* realisation [Zhang et al., 2006].

to the training image, variability is only introduced by randomly sampling each pattern class. Suppose the training image itself only contains a limited number of unique patterns. In that case, the generated models could look very similar, representing a limitation in quantifying geological uncertainties [Honarkhah and Caers, 2010].

In figure 3.2 we can see what an unconditional *filtersim* realisation (right) from a channel system training image (left) looks like. Similar to *snesim*, on visual inspection this MPS algorithm is capable of replicating curvilinear features from the training image and preserve overall channel connectivity.

It should be mentioned that in this work the choice of algorithm fell onto the *snesim*, due to its immediate availability.

## 3.2 Flow diagnostics

A common denominator between the subsequent chapters is that capturing the impact of geological uncertainties on fluid flow requires building hundreds of reservoir models and checking their fluid flow behaviour. This makes using computationally heavy full physics fluid flow simulation for evaluation unfeasible. Thus, simplified methods are required as a way to quickly approximate the flow behaviour of a reservoir model. It was therefore decided to use flow diagnostics for this evaluation step instead. Flow diagnostics are simplified, computationally inexpensive numerical experiments that are run on the

reservoir model and give a quantitative approximation for the dynamic behaviour of a reservoir model [Møyner et al., 2015]. Flow diagnostics allow us to screen reservoir models for their fluid flow fields. Measuring the heterogeneity of flow fields within a reservoir model is a suitable way to approximate the flow behaviour of a reservoir. This knowledge is important to get an understanding about which areas of a reservoir would be swept or remain unswept by an invading fluid (e.g. water injection) [Watson et al., 2021]. These measures can then be used as inputs for metrics to describe reservoir heterogeneity, such as the dynamic Lorenz coefficient or flow-capacity storage-capacity diagrams that are also often linked to streamline simulation [Datta-Gupta and King, 2007, Shahvali et al., 2012, Shook and Mitchell, 2009, Watson et al., 2021].

The flow or velocity fields required for the calculation of flow diagnostics can cost-effectively be calculated by solving for a single phase, in-compressible fluid flow equation. A reservoirs velocity field for single, in-compressible flow can be described with equation 3.1:

$$\nabla * \vec{v} = q, \vec{v} = -K\nabla p \quad (3.1)$$

where  $\vec{v}$  represents fluid velocity,  $q$  a source term,  $K$  permeability and  $p$  the fluid pressure. If we assume that  $\vec{v}$  is constant and divergence free, there will through each individual grid cell  $x$  in the reservoir be a unique curve, also called a streamline  $\psi$ , that is tangential to the velocity field. The streamline starts at the nearest source term (e.g. injector well) and ends at the nearest sink term (e.g. producer well) and can be parameterized by its arc length  $r$ . To understand how the heterogeneous velocity field impacts flow, usually the so-called time-of-flight is calculated. Time-of-flight is a spatial coordinate that measures the time it takes an imaginary particle to travel a certain distance along a streamline and is calculated with equation 3.2:

$$\tau = \int_0^r \frac{\phi}{|\vec{v}|} dr \quad (3.2)$$

where  $\tau$  is the time-of-flight,  $\phi$  the porosity and  $\vec{v}$  the velocity along the streamline with an arc length  $r$ . The travel time for an imaginary particle from the injector well to a given

point in the reservoir is called forward time-of-flight  $\tau_f$ . The travel time of that imaginary particle from that point to the producer is called backwards time-of-flight  $\tau_b$ . The total travel time  $\tau_t$  is a combination of  $\tau_f$  and  $\tau_b$ . All three measures can be used as indicators for swept (low tof) or unswept reservoir areas (high tof). It is noteworthy to iterate that, while velocity  $\vec{v}$  varies along a streamline, it remains constant within each grid cell  $x$ .

Whilst for conventional streamline simulations, the time-of-flight  $\tau$  is computed along the streamline and varies along the arc length  $r$  of the streamline, the time-of-flight can also be directly computed for each grid cell  $x$  within a reservoir. This can be achieved by solving equation 3.3 with a standard finite volume method:

$$\vec{v} * \nabla \tau = \phi \quad (3.3)$$

For each grid cell  $x$  we now have a volume-averaged value for  $\tau$  across all streamlines segments that pass through that grid cell. The advantage with the grid-based values for  $\tau$  is that they represent the piston-like propagation front in a reservoir and could be calculated by any flow simulator.

Conventional, static measures that try to describe reservoir heterogeneity based on petrophysical properties can also be calculated with the flow diagnostic values described above, but here, we take into account dynamic properties such as flow path heterogeneity [Shook and Mitchell, 2009]. Metrics used in this work are flow-capacity-storage-capacity diagrams, the dynamic Lorenz coefficient and the sweep efficiency [Møyner et al., 2015]. Flow-capacity  $F$  is the cumulative flow in order of increasing travel time  $\tau_t$  and storage-capacity  $\Phi$  is based upon the swept pore volume. By ordering the local pore volume of each grid cell at position  $\vec{x}$ ,  $F$  and  $\Phi$  can be calculated based on the local value for  $\tau_t$  with equation 3.4:

$$\Phi = \int_0^\tau \phi(\vec{x}) d\tau_t, F = \int_0^\tau \phi\left(\frac{\vec{x}}{\tau_t}\right) d\tau_t \quad (3.4)$$

The dynamic Lorenz coefficient  $L_c$  is a measure for the difference in displacement between completely homogeneous displacement and the actual displacement in a heterogeneous medium and can be calculated with equation 3.5:

$$L_c = 2 \int_0^1 (F(\Phi) - \Phi) d\Phi \quad (3.5)$$

where complete homogeneous displacement equates to a  $L_c$  of 0 and completely heterogeneous displacement to a  $L_c$  of 1. Sweep efficiency  $E_v$  is a measure for the volume of in-place fluid contacted by the injected fluid at time  $t$  and can be calculated with equation 3.6:

$$E_v = \Phi + (1 - F) \frac{d\Phi}{dF} \quad (3.6)$$

Several applications in the past demonstrated that time-of-flight-based flow diagnostics are a suitable screening tool for decision making and reservoir management under uncertainty [Møyner et al., 2015, Spooner et al., 2019, Watson et al., 2021] and will therefore be used throughout this work.

It should be mentioned, that any implementation of flow diagnostics used in this work, is based upon the MATLAB library Matlab Reservoir Simulation Toolbox (MRST) developed by Lie et al. [2012].

### 3.3 Meta-heuristic optimisation

Optimisation techniques try to minimise or maximise the output of a given objective function by selecting from a set of predefined input parameters that all have constraining upper and lower bounds for input values [Venter, 2010]. There are many ways to group optimisation algorithms. One possibility is differentiating between algorithms that solve deterministic or stochastic optimisation problems. Deterministic optimisation algorithms are often gradient-based and work well when applied to problems with few unknowns. The input data for the given problem are accurately known and the algorithms tend to converge towards an optimum solution quickly [Cavazzuti, 2013]. However, they run into issues

with many real-world applications where optimisation is performed under uncertainty. There could for example be (i) no analytical description of the function available (e.g. fluid flow simulations), (ii) the objective function itself is discontinuous and has regions with invalid solutions, (iii) the objective function is noisy and has multiple local optimum solutions, (iv) the problem constitutes a mixture of continuous and discrete variables or (v) the problems at hand are highly non-linear and complexity of the optimisation problem rises exponentially thereby making discrete optimisers inefficient when tasked with finding an optimal solution.

As the subsurface domain inherently entails uncertainty, optimisation problems here are generally ill-posed and more suitable for stochastic optimisation algorithms [Hajizadeh et al., 2010, Mohamed et al., 2010, Sun, 2013]. Many of these algorithms get their inspiration from processes and behaviours observed in nature, such as the flocking of birds, human-human interactions, the behaviour of a school of fish, or the social interactions in bee and ant colonies [Abdel-Basset et al., 2018, Bonabeau et al., 1999, Eberhart et al., 2001, Holland, 1998, McCall, 2005, Parsopoulos and Vrahatis, 2002a, Schutte and Groenwold, 2005, Sousa et al., 2004, Ursem and Vadstrup, 2004]. They represent a group of more robust and generic algorithms designed for complex optimisation problems by injecting a random component into the searching operating and sampling the parameter space more often. From a mathematical point of view they are more intuitive to understand and proved themselves in a variety of problems across industries [Abraham et al., 2006, Bianchi et al., 2009, Chang et al., 2004]. Most of these algorithms are population-based algorithms, where an initial set of samples evolves towards convergence. The natural model depicted determines the evolution principles, which typically involve some element of perturbation, updating and unpredictability. Swarm intelligence (SI) refers to these population-based algorithms that emulate the collective behaviour of self-organised natural systems [Cavazuti, 2013]. The cooperative behaviour of individual, unsophisticated "agents" on a local level and their interaction with the environment can lead to the emergence of intelligent behavioural patterns for the collective of individuals. Difficulties with these algorithms stem from parameter tuning to achieve an optimum trade-off between the exploratory and exploitative behaviour of the algorithm that can lead to significantly different optimisation outcomes. Due to its suitability for global optimisation problems, its capability to

achieve good results quickly, its lower computational cost in comparison to other stochastic algorithms (e.g. genetic algorithms), ease of implementation, configuration and the low number of hyper-parameters, the particle swarm optimiser (PSO) will be used in the scope of this study [Matott et al., 2006, Mohamed et al., 2010, Mouser and Dunn, 2005].

### **3.3.1 The particle swarm optimiser (PSO)**

The particle swarm optimiser (PSO) is a powerful tool for stochastic optimisation problems proposed by Kennedy and Eberhart [1995] and has a proven track record for history matching problems of reservoir models [Fernández Martínez et al., 2012, Kathrada and Carter, 2010, Kato et al., 2014, Mohamed et al., 2010, Rwechungura et al., 2011, Tolstukhin et al., 2012, Vazquez et al., 2015], well placement and drilling optimisation [Irgens and Lavenue, 2007, Onwunalu and Durlofsky, 2010, Self et al., 2016]. It is a population-based, iterative search algorithm inspired by the flocking of birds [Kennedy and Eberhart, 1995, Shi and Eberhart, 1998], where each particle explores the multidimensional parameter space, trying to optimize the match to a given objective function. Their movements are determined by (i) following their own paths towards an optimum solution (cognitive component) and (ii) the actions taken by either the entirety of the swarm or a subgroup of it (social component) to alter their movements. Since PSO was first proposed, numerous variants of the original algorithm with modifications to particle topology, parameter search space, boundary conditions, constraints and objectives have been developed. Depending on the type of PSO and parameter configurations selected, the PSO has the ability to either converge quickly towards satisfactory solutions or to conduct global exploration. For decades the PSO has been successfully deployed to tackle optimisation problems from a vast range fields, such as neural network training [Kennedy and Eberhart, 1995] static and dynamic function optimisation [Blackwell and Branke, 2006, Blackwell, 2005, Hu and Eberhart, 2001, Shi and Eberhart, 1998, 1999], multi-modal and multi-objective optimisation [Brits et al., 2002, Coello Coello and Lechuga, 2002, Mason et al., 2017, Parsopoulos and Vrahatis, 2002b], connectivity enhancement for the internet of things [Hasan and Al-Rizzo, 2019] and data clustering [Cohen and De Castro, 2006]. In this section I will give an introduction to the algorithm and its workings.

As mentioned before, PSO got its inspiration from the flocking behaviour of birds [Eberhart et al., 2001]. The algorithm captures this by representing each bird as a particle that contributes to a swarm of particles. Unlike most deterministic optimisation algorithms, the PSO setup allows it not to use gradient-based methods for optimisation, which removes the necessity for the objective function to be differentiable. Each particles' movement through the  $n$ -dimensional search space is dictated by a velocity vector derived from an equation taking into account (i) the past experience of the particle, (ii) the past experience of the swarm or a subset of the swarm and optionally (iii) an inertia component giving further control over particle velocity. Each particle has a cognitive component allowing it to remember its personal best position ever visited and a social component that informs it about the best global position of the swarm or that a subset of the swarm ever visited. Each particle's velocity is updated according to the most recent information with every iteration. These components allow a swarm of many "unsophisticated" particles to develop intelligent behavioural patterns when searching through the parameter space in a trade-off between exploitation (move towards the current optimal solution) and exploration (investigate surrounding parameter space for superior solutions).

Mathematically the PSO, its particles and their interactions can be described with the following set of definitions and equations. A population consists of  $n$  particles in a  $d$ -dimensional parameter space, each at position

$$X_i = (x_{i1}, x_{i2}, \dots, x_{ij}), \quad i \in [1, n], \quad j \in [1, d] \quad (3.7)$$

At the initialisation phase, all particles are randomly placed within the parameter space. The next location that a particle visits is determined by the velocity vector  $V_i$  for each particle:

$$V_i = (v_{i1}, v_{i2}, \dots, v_{ij}), \quad i \in [1, n], \quad j \in [1, d] \quad (3.8)$$

Note here that this vector should be capped to prevent overshooting particles. According to benchmarking studies by Eberhart and Shi [2000] a good rule of thumb for maximum velocity  $V_{max}$  is to pick half the range of each variable in the  $d$ -dimensional search space.

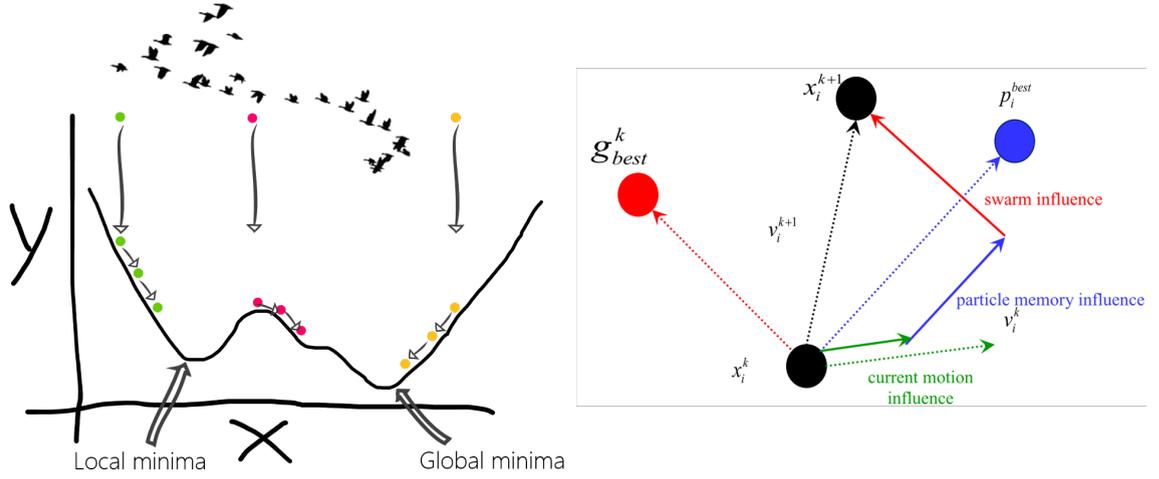


FIGURE 3.3: Left: Example for the particle movement of a standard pso within a 1-dimensional parameter space. Right: Particle velocity vector composition [Mohamed, 2011].

The cognitive component of each particle allows them to remember the best position  $P_{best}$  visited by each particle:

$$P_{best_i} = (p_{best_{i1}}, p_{best_{i2}}, \dots, p_{best_{ij}}), \quad i \in [1, n], \quad j \in [1, d] \quad (3.9)$$

The social component, which is the global best position within the swarm is denoted as  $G_{best}$ . At each iteration these information are then used to calculate the particles velocity vector  $V_i$  with equation 3.10 and illustrated in figure 3.3:

$$V_i(t+1) = wv_i(t) + c_1R_1(P_{i,pbest}(t) - P_i(t)) + c_2R_2(P_{gbest}(t) - P_i(t)) \quad (3.10)$$

where  $v_i(t)$  and  $P_i(t)$  represent velocity and position of *particle*<sub>*i*</sub> at iteration *t*,  $P_{i,pbest}$  represents the personal best of *particle*<sub>*i*</sub>,  $P_{gbest}$  the best current global position, *w* is an inertia weight and  $R_1$  and  $R_2$  are both random vectors, usually in the range of 0 - 1.  $c_1$  represents the cognitive acceleration coefficient and  $c_2$  the social acceleration coefficient [Shi and Eberhart, 1998]. The new position  $X_i$  of a particle at  $t + 1$  is calculated with equation 3.11:

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad (3.11)$$

A schematic overview of the workings of the standard PSO algorithm is summarised in algorithm 1 and figure 3.3.

---

**Algorithm 1:** Pseudo-code for standard particle swarm optimiser

---

**–Initialisation phase–**

```

for each particle  $i$  do
  for each dimension  $d$  do
    Initialise particle position  $x_{id}$  randomly within permissible range
    Initialise particle velocity  $v_{id}$  randomly within permissible range
  end
  Calculate particle fitness  $fitness_i$ 
   $personal_{best} = fitness_i$  // set current pos to personal best pos
  if  $fitness_i > global_{best}$  then
    |  $global_{best} = fitness_i$  // set current pos to global best pos
  end
end

```

**–Optimisation phase–**

```

while maximum iterations or minimum misfit criteria unattained do
  for each particle  $i$  do
    for each dimension  $d$  do
      Calculate particle velocity  $v_{id}$  according to eq. 3.10
      Update particle position  $x_{id}$  according to eq. 3.11
    ;
    end
    Calculate particle fitness  $fitness_i$ 
    if  $fitness_i > personal_{best}$  then
      |  $personal_{best} = fitness_i$ 
    end
    if  $fitness_i > global_{best}$  then
      |  $global_{best} = fitness_i$ 
    end
  end
end

```

---

The selection of both acceleration coefficients and the inertia weight has been shown to have the most significant impact on the algorithm’s performance and allows to manipulate the PSOs exploration-exploitation trade-off [Engelbrecht, 2005, Kennedy, 1997, 1998, Suganthan, 1999, Trelea, 2003]. A parameter setting that allows for good exploration capabilities while being able to perform fine local searches would be desirable. An optimiser that can escape local minima would be even better. The work of Trelea [2003] closely examines how particles in a standard PSO behave and converge under different settings for these parameters. In their work inertia weight is called  $a$  and the cognitive and

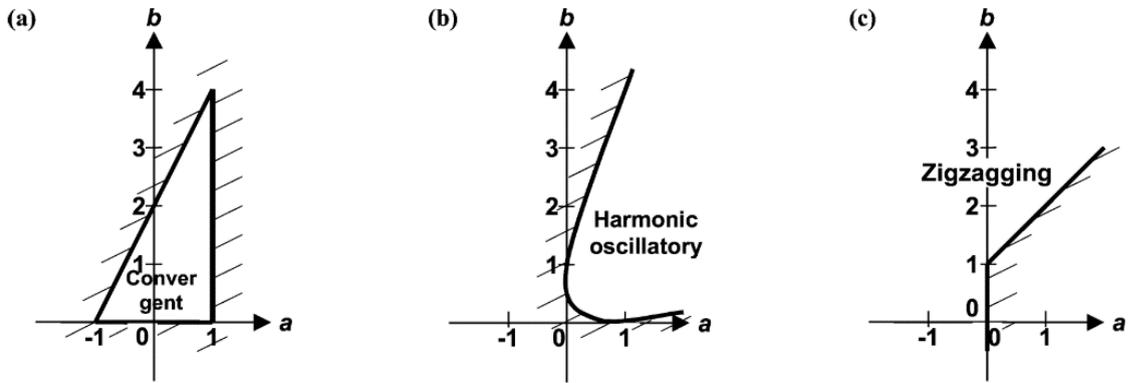


FIGURE 3.4: Particle behaviour domains for different PSO parameter settings showing (a) convergent behaviour, (b) harmonic oscillating behaviour or (c) zig-zagging behaviour [Trelea, 2003].

social component  $c_1$  and  $c_2$ , respectively, are simplified to  $b$  with equation 3.12:

$$b = \frac{c_1 + c_2}{2} \quad (3.12)$$

Applying a PSO to a set of five standard benchmark functions with different parameter settings for  $a$  and  $b$  can lead to either (i) converging, (ii) harmonic oscillating and (iii) zig-zagging particle behaviour (figure 3.4). Figure 3.5 shows examples for different parameter settings and the resulting behaviours of an individual particle. As can be seen by this set of experiments, choosing an appropriate set of parameters for the PSO is crucial. Therefore, throughout this work, the setting of all weightings follow the suggestions by Trelea [2003], where  $c_1 = c_2 = 1.498$  and  $w = 0.729$ .

Another aspect that can profoundly impact the PSO outcomes is the neighbourhood topology. Suppose all particles are connected and form one giant swarm (figure 3.6 b). In that case, there is the potential that the explorations capabilities are hindered and cause premature convergence to local optima [Engelbrecht, 2013, Shi and Eberhart, 1999]. This of course also depends on the complexity of the objective at hand. An alternative to the global neighbourhood are swarms with numerous local neighbourhoods where only parts of the swarm are in communication [Kennedy, 1999]. Here, each particle is solely affected by particles in its local neighbourhood and makes decisions based only upon the behaviour of the local community of particles. This type of topology takes much longer to converge in general, but it has a much stronger exploratory characteristic. Typical local

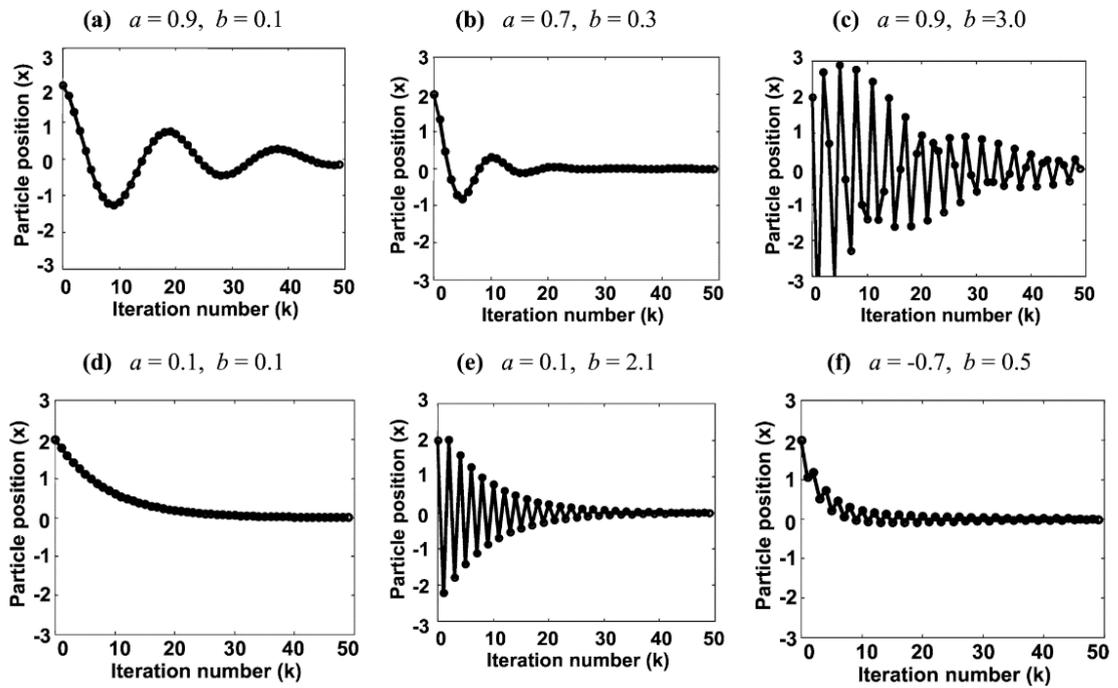


FIGURE 3.5: For various values of parameters  $a$  and  $b$ , examples of the dynamic behaviour of a single particle are shown. (a) Slowly convergent harmonic oscillations. (b) Quickly convergent harmonic oscillations. (c) Zigzagging harmonic vibrations. (d) Convergence that is not oscillatory. (e) Zigzagging in a symmetrical pattern. (f) Zigzagging in an asymmetrical pattern [Trelea, 2003].

neighbourhood typologies are the ring, wheel, random, Von Neumann and star topology (figure 3.6 a,c-f). In this work a ring topology is used for the PSO, where particles are connected with their  $k$ -nearest neighbours, a topology often used for local-neighbourhood PSOs.

Particles with velocities outside of the predetermined velocity range  $V_{min}$  and  $V_{max}$  or particles with new positions outside the parameter search space have to be handled by either adjusting their velocity or re-positioning them with the help of boundary handlers [Helwig and Wanka, 2008, Helwig et al., 2009, Huang and Mohan, 2005, Schutte et al., 2004, Zhang et al., 2004]. The selection of an adequate boundary handler can be essential when optimum solutions are situated close to the boundaries. Common methods used for velocity handling can set the velocity to zero, invert the velocity by a factor  $-z$  or adjust the velocity to be the distance between the current and the previous position. Methods for handling particles that move outside the predefined parameter space are random, reflecting, damping and absorbing methods (figure 3.7). The random method resets the particle to a new random position within the parameter range. The reflecting method re-spawns the

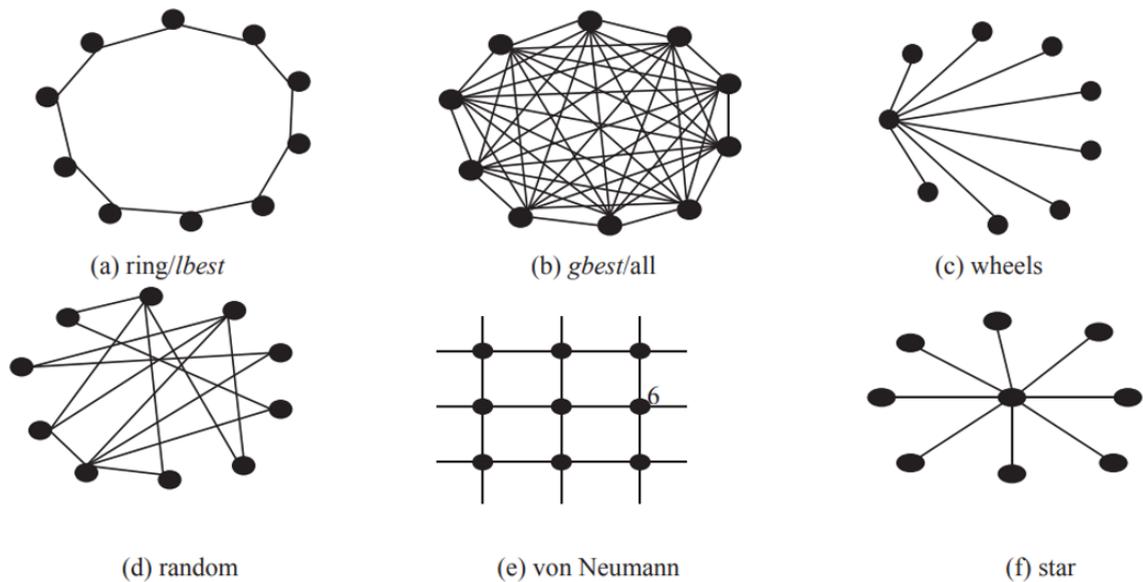


FIGURE 3.6: Population topologies in particle swarms [Lynn et al., 2018]

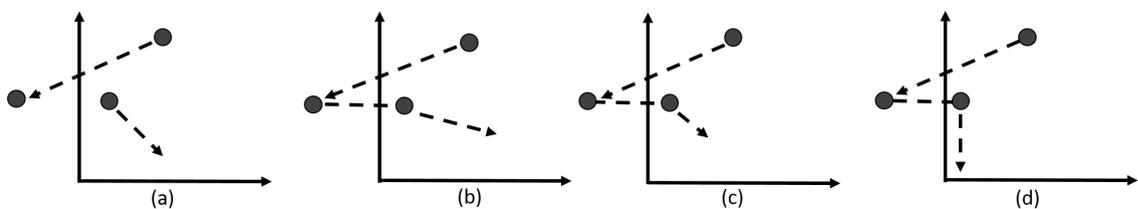


FIGURE 3.7: Different boundary strategies for handling particles: (a) random, (b) reflecting, (c) damping, (d) absorbing (illustration modified after Mohamed [2011]).

particle at the boundary of the parameter range and inverts the velocity vector, directing the particle away from the boundary. The damping method shrinks the velocity vector and by a random factor between 0 and  $V_{max}$  and directs it into the opposite direction. Finally, the absorbing method resets the particle to the boundary and velocity is set to zero. In this work, unless noted otherwise, particles exceeding their bounds are reflected off the boundary and velocities are inverted.

It should be mentioned, that any implementation of the particle swarm optimiser developed in this work, is based upon the python package PySwarms developed by James V. Miranda [2018] and represents the core of the workflow developed in chapter 5.

### 3.4 Entropy

Another important building block utilised throughout this thesis is a way to preserve geological diversity in the generated reservoir models. One cannot quantify geological uncertainty adequately by setting up a workflow that will over and over build the same geological model given a set of input parameters and output objectives. This would underestimate uncertainty biased to the repetitive model representations. To make sure that we can have better control over geological diversity, I borrowed something from the field of information theory called Shannon entropy. Shannon entropy measures the disorder that exists in a system and allows quantifying the amount of information of an outcome [Hansen, 2021, Reza, 1994, Shannon, 1948] and is defined by equation 3.13 as:

$$H = - \sum_{i=1}^N p_i \log p_i \quad (3.13)$$

where  $H$  depicts the entropy of a system and  $p_i$  the probability of each possible outcome to occur. It allows us to quantify how surprised we are by the occurrence of an outcome (i.e. an outcomes's probability). Outcomes that rarely occur carry a high load of information, and outcomes that occur more frequently carry a low information content. In this work, the primary use case for entropy will be to compare different reservoir model realisations. If the modelled geology and, therefore, the fluid flow behaviour is identical, adding a new model similar to already generated models adds no new information/knowledge to the ensemble. For example, the case of the simple model of a fluvial reservoir shown in figure 3.8. On the left side we can see how the entropy of ensemble A develops with every additional model realisation and on the right side how the entropy of ensemble B develops. We can see that for ensemble B every additional realisation is identical to the previous one, given every model a probability of 1.0 to occur. After generating the first model, no additional information is added to the ensemble that we do not already know about. This is also reflected by the entropy of ensemble B. Plugging the probabilities assigned to each model into equation 3.13 results in an entropy of 0.0, independent of ensemble size. if we however look at ensemble A, we can see that every added realisation shows a slightly different potential channel body geometry and therefore carries additional

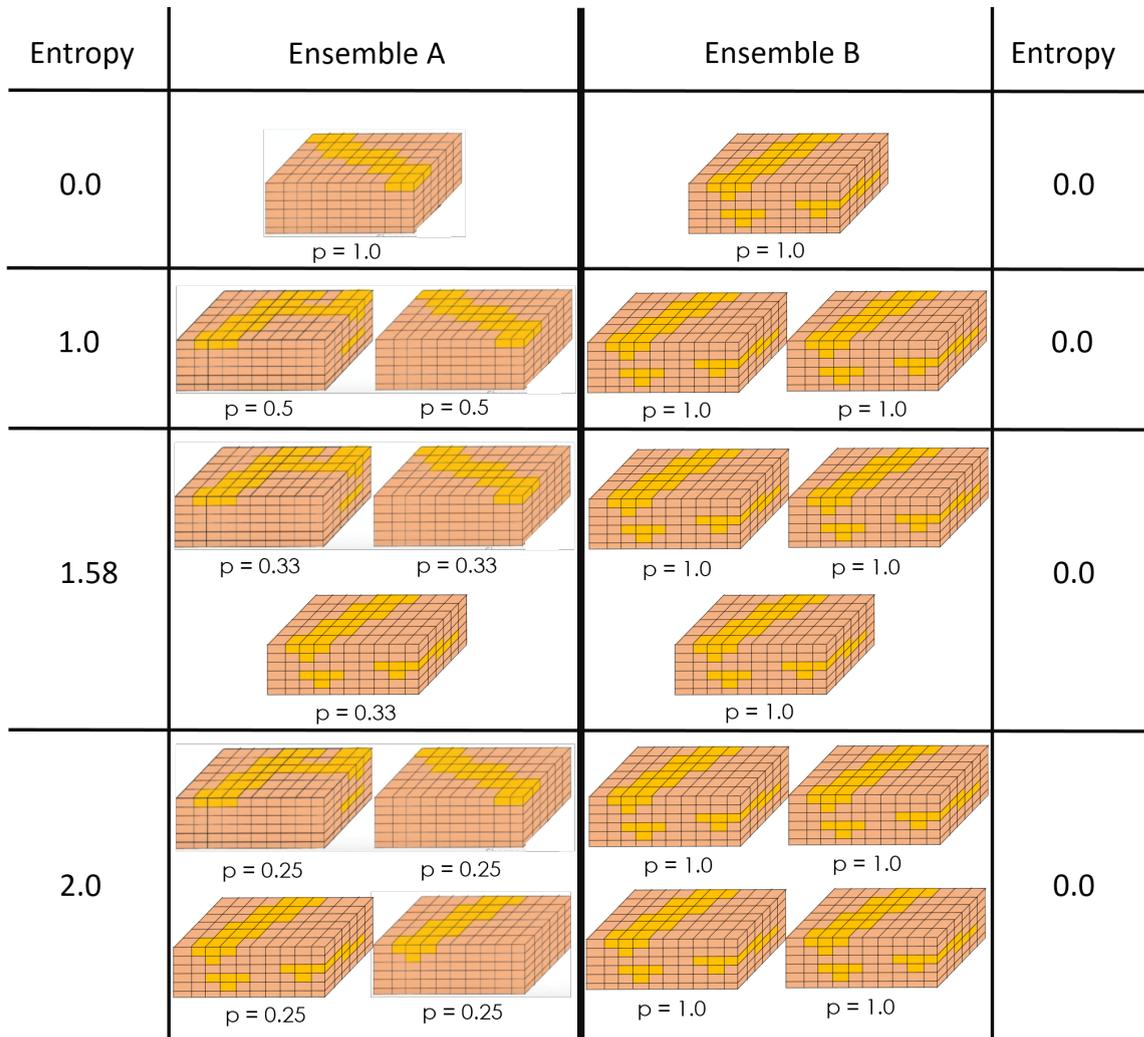


FIGURE 3.8: Fluvial reservoir model with channel (yellow) and background (brown) facies. Right: Ensemble of identical model realisations, giving every model a probability of 1.0 to occur and an entropy of 0.0 for the entire ensemble. In ensemble B models carry a low load of information. Left: Ensemble with varying model realisations added at every step, leading to an entropy of 2.0 for the ensemble with four models. In ensemble A every model carries a high load of information.

information than the other realisations. This is reflected in a probability for each model and also in the entropy of the ensemble, that increases with every additional model. With this simple example, I showcased how we could potentially use entropy in this work to get a quantitative measure for model diversity. It should however also be mentioned that this is just an illustrative example for the use of entropy. Here, only changing the value in a single cell of the models shown in ensemble B would assign the same amount of information to ensemble B as ensemble A. By visual inspection alone, this notion can be discarded as inadequate. In upcoming chapters I deploy entropy in a range of different ways, but in general measure entropy on a cell-by-cell basis and not a model basis.

### 3.5 Dimension reduction

When trying to manage geological uncertainties around a reservoir, modellers can produce hundreds of stochastic reservoir model realisations based upon different concepts and input parameters. However, carrying every single model through conventional simulation and optimisation workflows is impractical. Ensemble filter methods for example can reach ensemble sizes of 100s – 1000s of models [Liu and Oliver, 2005]. It is more sensible to move forward with a more manageable subset of models that cover a range of uncertainty comparable to that of the entire set of models. Suppose we now project parameters that can describe each reservoir model realisation (e.g. input parameters used to build the model or property distributions within the model) into a higher-dimensional parameter space. In that case, we can assume that models that are projected next to each other (neighbours) will show a dynamic response that is more comparable than that of models that have a greater distance between each other. Suzuki and Caers [2006] for example, showed that by projecting the static properties of a model into a higher dimensional parameter space, models that are close neighbours have a comparable flow response to each other. Many other studies out there demonstrate the usefulness of comparing reservoir models by projecting them into a higher-dimensional parameter space [Arnold et al., 2016, Rongier et al., 2016, Scheidt and Caers, 2013, Scheidt et al., 2015]. In higher dimensions, identifying which models or data points are neighbours can become challenging and was coined as the *curse of dimensionality* by Bellman [1966]. The reason is the increase in space that comes along with every extra dimension. As the dimensionality of a parameter space increases, individual data points (reservoir models in our case) become so spread out that the idea of distance and density that we are familiar with and can imagine in one to perhaps four dimensions becomes less effective. With every additional dimension, distances between points become more and more equidistant. This becomes a problem when using algorithms that rely on distance or similarity, like clustering techniques (more on this in the next section). Additionally, visualisations and interpretations of data in higher dimensions are also challenging. Valuable techniques that address this problem are dimension reduction techniques, where we project a higher-dimensional parameter space into a lower-dimensional space (figure 3.9).

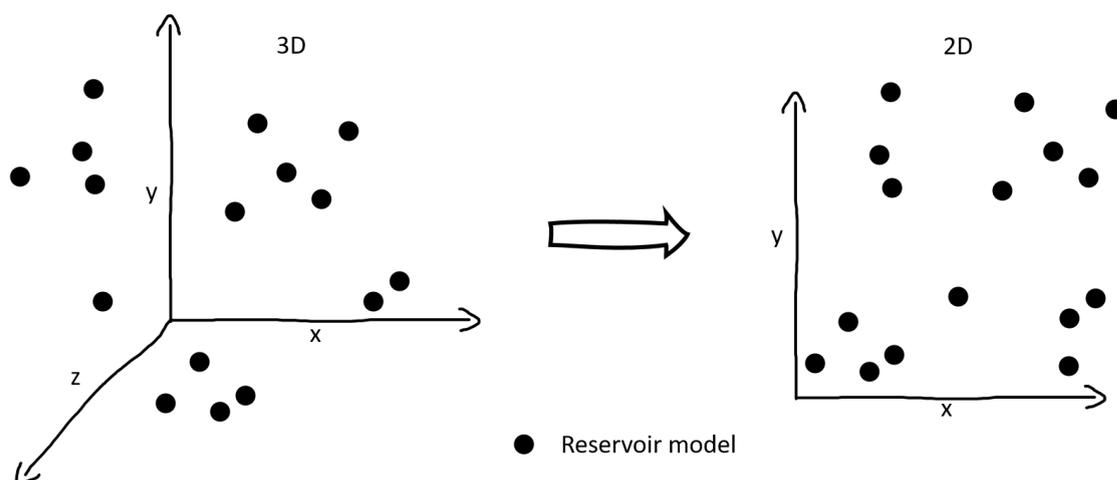


FIGURE 3.9: Schematic example for a projection of a three-dimensional representation  $(x,y,z)$  of a reservoir model into a two-dimensional representation  $(x,y)$ .

Examples of dimension reduction techniques are principal component analysis (PCA) which is a matrix factorisation technique, t-distributed stochastic neighbour embedding (t-SNE) and Uniform Manifold Approximation and Projection (UMAP), which are both graph layout techniques. [Jolliffe and Cadima, 2016, Laurens van der Maaten and Hinton, 2008, McInnes, 2018]. In this work, I exclusively used UMAP and will briefly introduce that technique and PCA, a commonly used method.

The objective of the PCA is to find the most *meaningful* directions in a data set and discard the directions of lower significance, thereby reducing the dimensionality of the data set. This is done by representing the standardised data by its principal components or eigenvectors, the data set's directions that carry the most variance and information. Ordering the eigenvectors according to their eigenvalues (coefficients of eigenvectors) allows us to pick eigenvectors that carry the highest load of information. By adding the  $n$  most significant eigenvectors to a matrix and multiplying the original data set with that matrix, we can project our data into a lower,  $n$ -dimensional space whilst preserving most of the information. However, a problem with the PCA is that it relies on the assumption of linear correlation among different parameters in the data set. In reality, there is often no linear correlation among parameters which leads to the loss of large amounts of information. Additionally, the PCA relates variance to information, which is not necessarily the case with real data. Lastly, if there is merely little correlation among variables in the data set, it will require many eigenvectors (and therefore dimensions) to

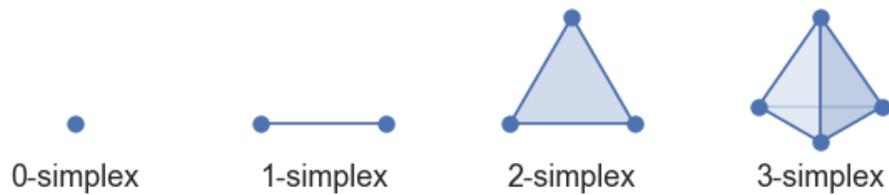


FIGURE 3.10: Examples for 0-simplex, 1-simplex, 2-simplex and 3-simplex (after McInnes [2018]).

represent the data set adequately.

UMAP on the other hand has no difficulty handling non-linear data sets. There are broadly two steps that UMAP applies to the given data set. It first sets up a connected graph representation of the data set in the original dimension and then tries to reproduce the graph representation in lower dimensions [McInnes, 2018]. Both steps consist of several sub-steps that I will outline below, but for a more detailed description, I refer to [McInnes et al., 2018] and the extensive UMAP documentation.

To allow for the construction of connected graphs in higher dimensions, I will first introduce the notion of a simplex. A simplex describes the simplest possible connection between  $n$  points, where a 0-simplex is simply a point, a 1-simplex a line, a 2-simplex a triangle, etc. (figure 3.10). The simplexes can now be used to build out a connected topological representation of our data set. Initially, each point in our data set is equivalent to a 0-simplex. Radially expanding a circle (in 2D) outwards from each point by a radius  $r$ , will eventually lead to the overlapping of circles from different points that can then be turned into 1, 2, or  $n$ -dimensional simplexes (figure 3.11).

An issue that already becomes apparent in the simple data set presented here is selecting an adequate radius  $r$ . The varying distances among data points will lead to only locally connected networks (small  $r$ ) or a network, where every point is connected (large  $r$ ). Neither is desirable, as they fail to capture the data set's underlying structure adequately. To circumvent this issue, it is assumed that in the higher-dimensional parameter space the data is embedded in, the notion of distance between points shrinks or expands, depending on the data set's local density. This means that a different radius  $r$  can be selected for each

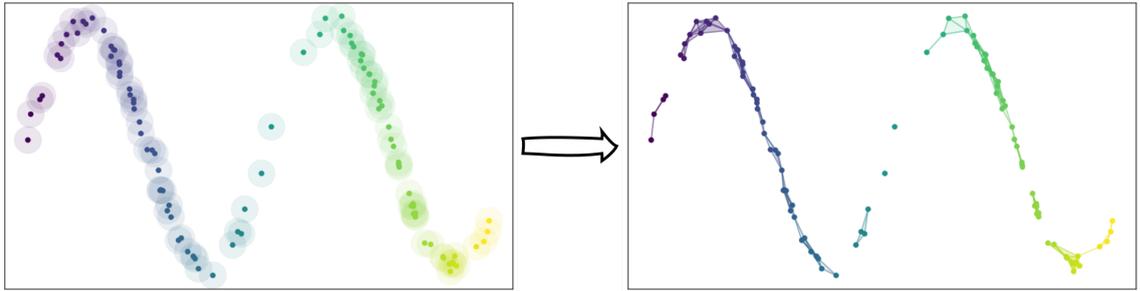


FIGURE 3.11: Sinuous example data set with circles of constant radius  $r$  drawn around each point (left) allowing for the representation of the data with simplexes (right). It should be noted, that not the entire data set is connected in this representation (after McInnes [2018]).

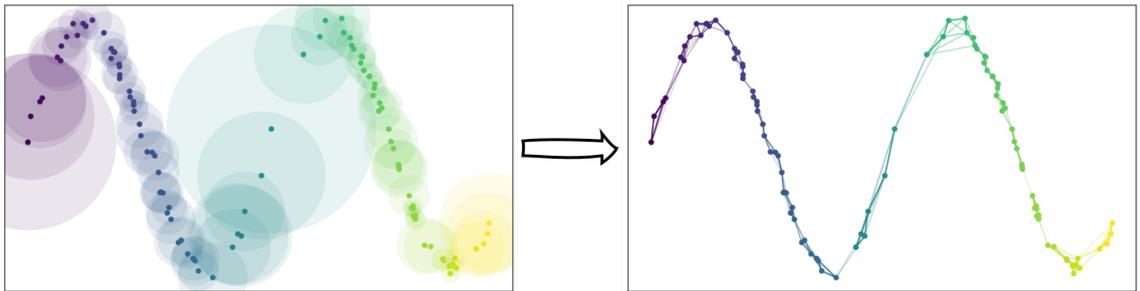


FIGURE 3.12: Sinuous example data set with varying circles of radius  $r$  drawn around each point (left), until  $K$  nearest neighbours are found. Simplexes can now represent the data fully connected (after McInnes [2018]).

point, allowing it to be large in sparse areas and smaller in densely populated areas (figure 3.12 left). Whether an area of the data set is sparsely or densely populated is determined by the distance to a points  $K$ -nearest-neighbours. However, varying the radius  $r$  for each data point leads to another problem as multiple points can now have numerous connections (edges) based upon different notions of distance. The *strength* of each connection is now determined by the distance between points, with points being closer to each other having a *stronger* connection and therefore a higher probability to be connected (weighted edges). Also, each point must be connected to at least its closest neighbour to ensure that the graph is fully connected (figure 3.12 right).

The second step is to embed this connected graph into the targeted lower dimensional parameter space by finding a topology that matches the original. Here, optimisation techniques are employed to find lower-dimensional representations of the graph that honour aspects like the original edge weighting.

The advantages of UMAP are that it is based upon an advanced theoretical foundation,

allowing it to efficiently and robustly work with data sets in higher dimensions. Compared to t-SNE, a similar dimension reduction technique, it is substantially faster when applied to large data sets. Additionally, by trying to preserve the higher dimensional connected graph in a lower dimension, UMAP can give a well-balanced representation of global and local data structures. A disadvantage of UMAP and many other non-linear dimension reduction techniques compared to linear techniques such as PCA is the lack of interpretability in the reduced dimension results [McInnes et al., 2018]. Unlike PCA, the dimensions that the data are projected into are not related to the highest information load (variance) directions but carry no particular importance.

### 3.6 Clustering

As pointed out in the previous section, selecting a manageable subset of reservoir models that cover a similar range of uncertainty compared to an initial set of models is vital for efficient uncertainty management procedures. Long run times for full physics simulations prevent us from carrying all reservoir models through decision-making processes. The selection of a manageable subset is made with the support of clustering algorithms. Clustering algorithms aim to identify similar sets of data points that can be grouped and are referred to as a cluster [Likas et al., 2003] (figure 3.13). To ensure that data are presented in a favourable way for unsupervised clustering algorithms to excel, I performed the dimension reduction step explained in the previous section. The next step is to determine how to group individual reservoir models and select a subset from each group of models whilst covering as much of the geological uncertainty as possible. There are numerous clustering techniques available with a proven track record. I will be focusing on two algorithms: K-means clustering [James et al., 2013, Steinhaus, 1956], a well-known, highly utilised method and Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) developed by Campello et al. [2013]. HDBSCAN will be used throughout this thesis and the details of its python implementation can be found at Leland McInnes et al. [2016].

K-means clustering represents one of the simplest forms of clustering and is a fast, iterative partitioning algorithm that is first mentioned by Steinhaus [1956] and MacQueen [1967]. Given a data set, the algorithm starts with  $K$  random starting points that will be the seeds

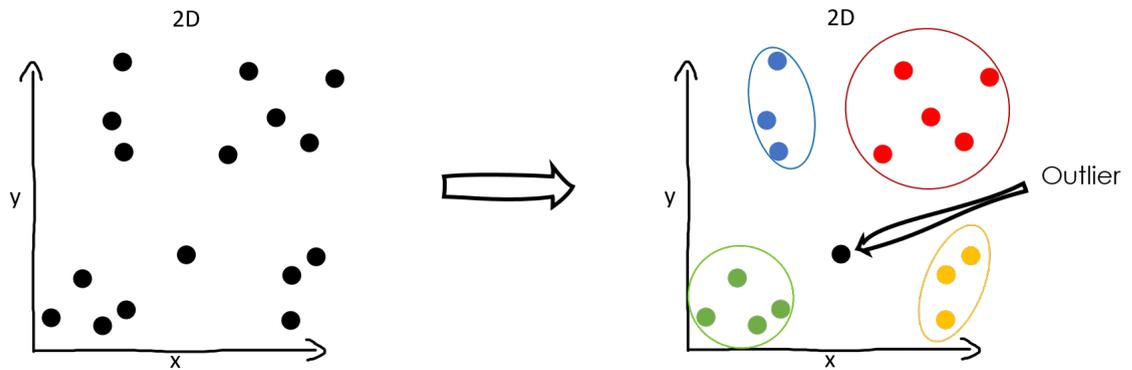


FIGURE 3.13: Data points situated close to each other should ideally be picked up by a clustering algorithm and grouped (colour-coded). Points that do not belong to a cluster will ideally be excluded from the selection process.

for the  $K$  clusters. It is noteworthy that the user defines  $K$ . The Euclidean distance between the remaining points and the  $K$  cluster seeds is now calculated, and every point is assigned to the closest cluster seed. The entire data set is now partitioned into  $K$  clusters. These are however heavily dependent on the random seed placement. Therefore, in the next step, the mean, which is equivalent to the cluster centroid, is calculated for each cluster. These now represent the updated  $K$  seed points for each cluster. The Euclidean distance between all points and the  $K$  cluster seeds is recalculated and every point is assigned to the closest cluster seed. These steps are iterated until (i) either a certain number of iterations is reached or (ii) the cluster mean does not change anymore and reaches a steady state. As already pointed out before, the K-means clustering algorithm strongly depends on each cluster's initial random seed points. As K-means clustering is computationally efficient, it is common to rerun the algorithm numerous times with different starting seeds. I then calculated the cost  $C$  for every run, the sum of the final squared Euclidean distance between each point and its corresponding centroid. The run with the lowest  $C$ , i.e.  $K$  centroids closest to the data points, is then selected as the best initiation point. However, we still need to know or define the number of clusters  $K$  manually. If we do not know how many clusters there are in a data set, it is common to iteratively go through numerous cluster sizes  $K$  and calculate the cost  $C$  for each  $K$ . The user can then select a  $K$  that balances the cost  $C$ . This can for example be done with the elbow method [Kodinariya, T.M and Makwana, P.R, 2013].

Besides its speed and ease of understanding, K-means clustering also comes with several

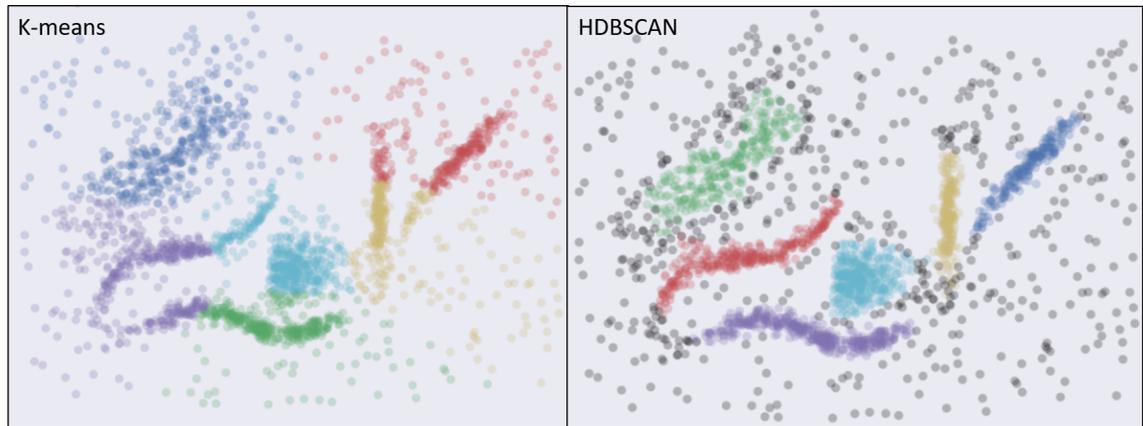


FIGURE 3.14: Left: K-means clustering with  $K = 6$  on 2-dimensional data set. Right: Clustering of same data set with HDBSCAN (after Leland McInnes et al. [2016]).

downsides. K-means has difficulties outlining non-spherical clusters. The requirement of personal judgment for selecting  $K$  also makes it less objective. As K-means is a partitioning algorithm, every data set point has to be assigned to a cluster. This means that outliers that the human eye can easily pick up will be assigned to a cluster. And lastly, as already pointed out beforehand, it relies heavily on the initialisation of the  $K$  seeds. An example of how k-means performs on a 2D data set with noise and non-spherical clusters can be seen on the left of figure 3.14. One can identify six clusters by visual inspection, which is also the information fed into K-means. However, the results do not hold up to what one would recognise as clusters by visual inspection. Another algorithm that is based upon k-means algorithm is k-medoids [Kaufman and Rousseeuw, 2009]. Here data points themselves are selected as cluster centers, therefore giving it a better interpretability over k-means. Additionally, it uses different measures for dissimilarities, making it more robust against outliers. It is nevertheless a partitioning algorithm that will struggle with clusters of different shapes and densities.

HDBSCAN has fewer problems with many of the issues that K-means faces. The algorithm can be broken down into several sub-steps. The first step is a preprocessing step that makes sure that noise within the data is not integrated into clusters. Simply put, the data set is transformed to push data points that qualify as noise further apart than initially. This is done by a local approximation of density. An expanding circle of radius  $r$  is drawn around each data set point until a minimum of  $n$  points is within the circle (figure 3.15). The smaller  $r$ , the denser the area around is populated. Note that  $n$  has to be defined by the

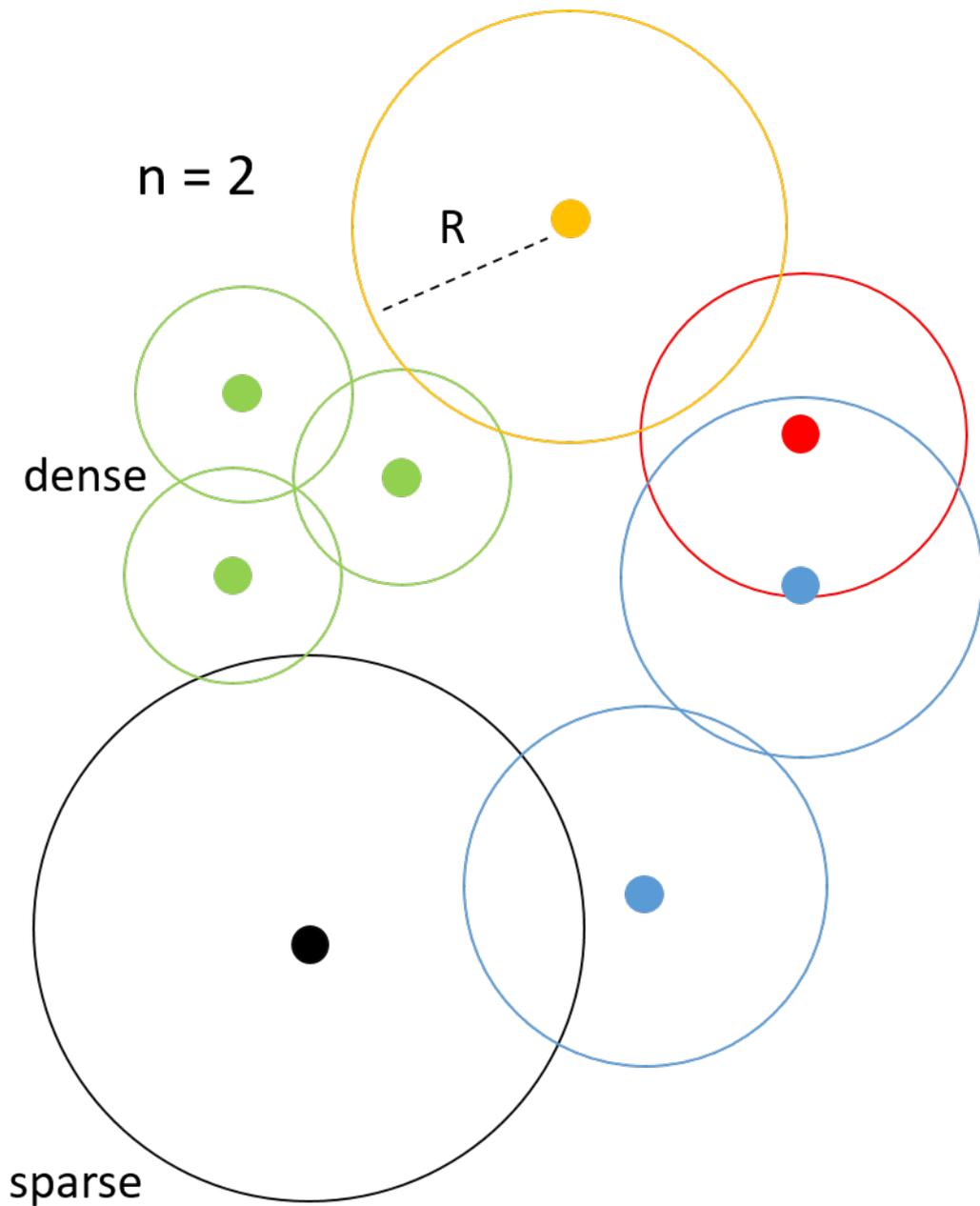


FIGURE 3.15: Local density estimation for data set by gradually expanding a circle around each point until  $n$  points are within circle.

user. A point is now classified as sparse if it is either not close enough to the other point or not in the core distance  $r$  of the point. Data points classified as sparse are now artificially pushed further apart.

In the next step, we gradually increase the radius  $r$  of the circle of each point (figure 3.15). Initially each point is its own cluster. Circles that touch each other merge into new clusters. If points from two different clusters touch, they become a single cluster.

Eventually, we would end up with one big cluster and a hierarchy of cluster development which at its bottom consists of a single cluster for each data point. This can be visualised in the form of a dendrogram, where with increasing  $r$ , more and more points merge to fewer branches until everything is connected to one tree (figure 3.16). A visual example would be the gradual drop-down of sea level. Initially, we would only have sea ( $r = 0$ ), but as the sea level is lowered (increase in  $r$ ), islands (clusters) start popping up. A further drop-down of sea level causes land bridges to merge islands (clusters merging) until there is only land left. With these steps alone, we would already be able to cluster data in a superior way to K-means. By selecting an adequate radius  $r$ , noisy data is discarded, and unlike k-means, density-based clustering is indifferent to the shape of a cluster. The steps up until now are comparable to the DBSCAN algorithm, on which HDBSCAN is based [Ester et al., 1996]. DBSCAN however requires the manual definition of  $r$ , which is a sensitive parameter capable of distorting the clustering outcomes. DBSCAN assumes a global  $r$  that does not differentiate between clusters with differing point densities. Here the hierarchical component of HDBSCAN comes into play. In HDBSCAN,  $r$  gradually decreases (rising sea level), causing a single cluster to shrink and either split up into a new cluster or points are declared as noise. Whether points are declared as noise is defined by the minimum number of points  $n$  required for a set of points to be defined as a cluster. Points that fall below that requirement are noise and not further considered, which mainly happens. Points above that requirement cause the formation of a new cluster. In the last step, HDBSCAN now decides which clusters to keep together or represent as individual clusters. This is done by calculating the stability of an initial cluster in comparison to the stability of both newly formed clusters. For each cluster we can define at what  $r_{init}$  value it was first initiated. For each point in the cluster, we can also identify an  $r_{point}$  value at which it would be discarded as noise. By taking the difference between  $r_{point}$  and  $r_{init}$  and summing the difference up for every point in the cluster, we get a stability value for each cluster. If the combined stability value of the two newly formed clusters is greater than the stability value of the old cluster, the new clusters persist. Otherwise, they stay combined in the initial cluster. This allows for the introduction of clusters with varying densities into HDBSCAN. Performing HDBSCAN clustering on the same data set as before gives much better results (figure 3.14 right).

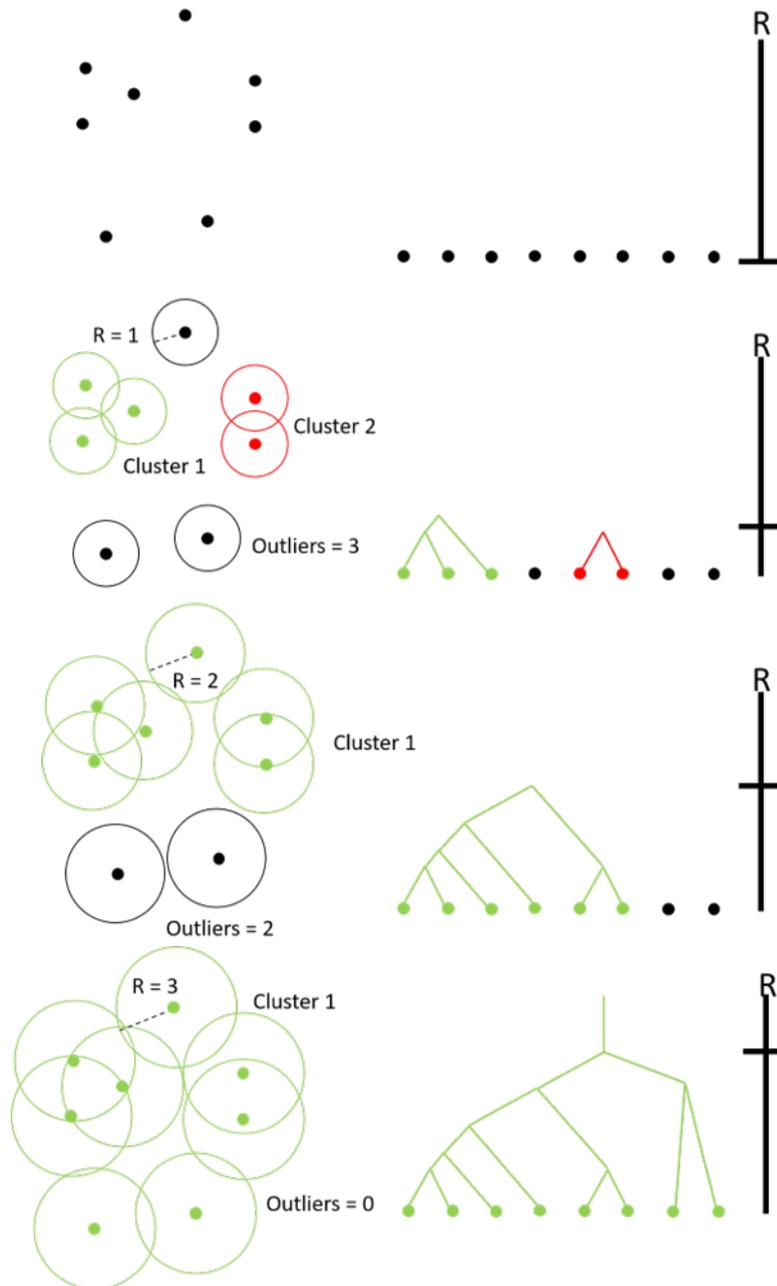


FIGURE 3.16: Gradually increasing radius  $r$  causing the generation and merging of new clusters. This can also be visualised with the help of a dendrogram.

### **3.7 Agent-based modelling**

Agent-based modelling is a class of algorithms that originated from the development of cellular automata and complexity theory which allowed the modelling of complex systems with the help of simple neighbourhood algorithms [Cervelle and Formenti, 2008, von Neumann, 1966, Wolfram, 1994]. Agent-based modelling describes a modelling style where the workings of a system are captured by the elementary units that make up the whole of the system [Bonabeau, 2002]. This is achieved by representing each elementary unit as an autonomous agent that follows a simple set of rules. The rules do not require intricate mathematical equations or expert knowledge on the subject matter [Wilensky and Rand, 2015]. A big advantage of ABMs over conventional equation-based models is that they are much easier to understand for individuals without domain expertise. This allows for improved communication and understanding across all stakeholders involved in the modelling study. The so-called "glass-box" style of modelling adds to the verifiability of a model by allowing everyone concerned to dissect the model down to its basic components [Tisue and Wilensky, 2004, Wilensky and Rand, 2015]. Despite their simplicity, throughout simulation of the agent-based models, these rules allow the collective of agents to self-organise and recreate the workings and emergent behaviours of complex phenomena observed in reality [Macal and North, 2010]. Emergent phenomena are a characteristic of complex systems [Wilensky and Rand, 2015]. Patterns that appear to be highly complicated following simple rules. This emergent behaviour occurs when the entirety of the system develops unforeseeable attributes on the scale of the systems that are not attached to the individual agents making up the system. Typically an ABM would have a set of agents with specific characteristics and inter-agent relationships. The agents exist in an environment that they can also interact with. The environment can be as simple as just providing a reference for the spatial position the agents occupy, but it can also carry more complex information that can influence an agents' behaviour [Wilensky and Rand, 2015]. Simulating the agent-agent and agent-environment interactions for a large set of agents iteratively is deemed to be the most effective method [Macal and North, 2010]. Agent-based models are particularly effective when there is a medium number (tens to millions) of interacting agents present [Casti, 1996].

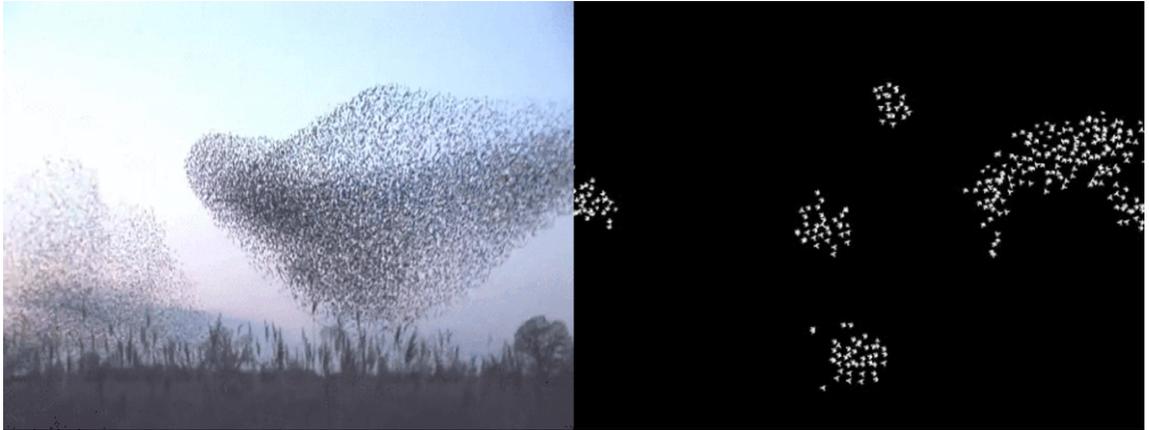


FIGURE 3.17: The flocking of starlings (left) is an often demonstrated example for the usage of ABMs trying to mimic complex behaviours with a set of simplistic rules (right, generated with Netlogo [Wilensky, 1999]).

Agent-based modelling is widely used in other industries and fields of science, such as financial market modelling, traffic analysis, the spread of infectious diseases, ecology, cognitive sciences, sociology, sports simulation, video animations and policy analysis, but little application for subsurface modelling [Arthur, 2021, Bianchi and Squazzoni, 2015, Castro et al., 2020, Damiano et al., 2013, DeAngelis and Diaz, 2019, Farmer and Foley, 2009, Ghadai et al., 2016, Guo and Sprague, 2016, Iori and Porter, 2012, Madsen et al., 2019, North and Macal, 2007, Perez and Dragicevic, 2009, Zuparic et al., 2017]. One example usage of ABMs from ecology would be mimicking the flocking of starlings shown in figure 3.17.

It should be mentioned, that any ABM implementation used in this work, got its inspiration from the python package Mesa [Mesa, 2015] and the code developed for Steffens et al. [2022].

# Chapter 4

## Modelling uncertainty in naturally fractured reservoirs with multiple-point statistics

### 4.1 Introduction

As already outlined in chapter 2, the underlying geology of naturally fractured reservoirs brings many modelling challenges. Modelling techniques can fail to depict geological realism, are time-demanding to build or are not feasible for reservoir scale in the first place.

This chapter aims to develop a method that allows us to model distributions of natural fractures on the reservoir scale quickly and efficiently, whilst not overly compromising geological realism. This is achieved by generating training images from discrete fracture networks that are then used in multiple-point statistics to populate reservoir models, as previously shown in the works of Jung et al. [2013] and Arnold et al. [2016]. The method developed in this chapter also aims to honour measured data and integrate conceptual knowledge and outcrop information throughout the modelling process. As there is a great deal of geological uncertainty associated with naturally fractured reservoirs, this method aims to capture these uncertainties within an ensemble of reservoir models to evaluate their impact on the fluid flow response of the reservoir. The methods for training image

generation developed in this chapter also build the foundation for the work carried out in the upcoming chapters. It should be noted, that investigating the uncertainties introduced by the choice modelling techniques (e.g. the representation of fractures with DFNs, upscaling of DFNs to continuous media, the selected grid cell size for reservoir modelling and simulation), would be beyond the scope of this work.

The chapter is structured as follows. Section 4.2 gives an overview of previous applications of MPS to naturally fractured reservoirs. Section 4.3 presents the developed workflow for training image generation and selection. Finally, section 4.4 applies the workflow on the synthetic case study we setup in chapter 2.

## **4.2 Previous applications of MPS to naturally fractured reservoir modelling**

Various works using MPS to characterise naturally fractured reservoirs have been done over the past two decades and can be organised in two fields. They either used MPS to simulate discrete realisation of natural fracture networks or discrete fracture networks were upscaled to continuous media property fields that are then used as training images for MPS.

The groundwork in the area of discrete fracture modelling with the help of MPS was laid by Liu et al. [2002] and Liu et al. [2009] focuses on capturing fracture statistics from outcrops, such as spacing and fracture density as well as fracture connectivity, a vital parameter for fluid movement through naturally fractured reservoirs. An outcrop image is scanned with a gridded template superposed which distinguishes between grid cells that contain fractures and those that do not. The different patterns that the template detects are then stored in a histogram that captures the occurrence and distribution of each pattern event. Based on this information and areal fracture proportion maps, the MPS algorithm subsequently seeds fracture initiation points on an empty grid. The multi-point conditional probability obtained in the previous step is used sequentially to grow fractures from the initiation points. In figure 4.1 the training data (left) are depicted together with a realisation from that training image (right). From a purely visual inspection, it appears that at this

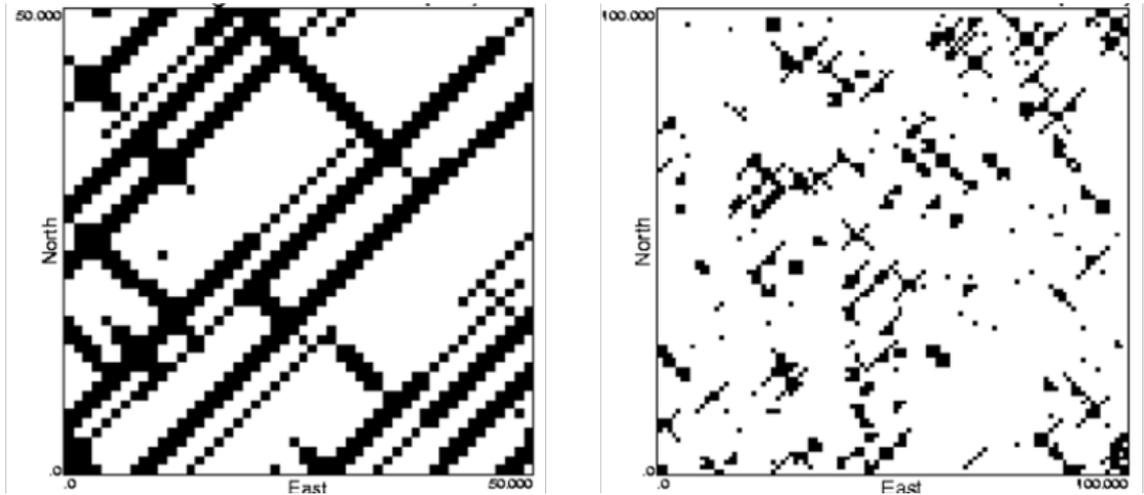


FIGURE 4.1: Left: Image that is scanned to obtain multi-point conditional probability. Fractures are depicted in black and matrix in white. The proportion of fractures is 50%; Right: MPS realisation that is constrained to well data and a fracture proportion of 20% [Liu et al., 2002].

overall fracture orientation was preserved. Fracture connectivity however is not, which could lead to very different fluid flow behaviours in both systems.

After nearly two decades of stagnation, Bruna et al. [2019] and Chandna and Srinivasan [2019] are the first to make advances on the work of Liu et al. [2002] to use MPS for discrete fracture modelling and capture fracture network variability as observed in outcrops on multiple scales. To do so, they made use of the direct sampling method [Mariethoz et al., 2010, Straubhaar, 2017]. Similar to the different MPS algorithms summarised in section 3.1.2, this method iteratively visits unknown nodes following a random path. It then compares the unknown nodes' neighbourhood with patterns found on the training image. One should note that this algorithm does not store patterns found on the training image beforehand but randomly scans the training image for matching patterns every iteration. The decision-process on what value to assign to the unknown node then follows two ways. Either the algorithm scans the training image and finds a candidate that falls below a predefined dissimilarity threshold. Or, if the proportion of scanned nodes on the training image exceeds a point, the so far found best value is assigned to the unknown node. This procedure is repeated until there are no more unknown nodes left. To account for geological realism and to include non-stationarity into their workflow, different grid areas are assigned to different training image zones, allowing for the usage of multiple training images on a single grid. A considerable advantage of this method is that

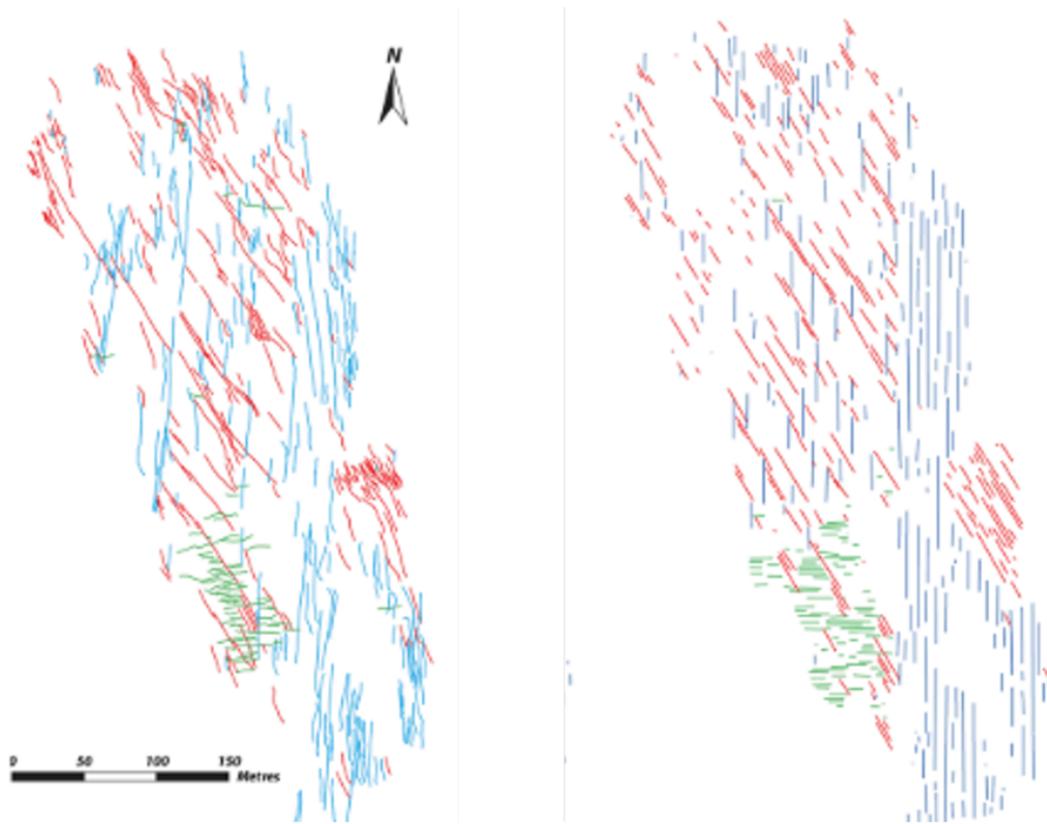


FIGURE 4.2: Left: Outcrop based reference case from which multiple training images are extracted for different fracturing zones; Right: Unconstrained realisation that captures characteristics of the fracture network [Bruna et al., 2019].

it will enable the geologist to use conceptual sketches as training image input. This gives the user complete control over the fracture statistics and fracture network configurations [Bruna et al., 2019]. From a purely visual inspection, it appears that overall fracture orientation and fracture network statistics are preserved to a sufficient degree (figure 4.2). It however seems that parameters controlling the MPS performance have a significant impact on the results, requiring a lot of tuning if numerous types of fracture sets are to be modelled in an uncertainty quantification driven workflow. Additionally, the overall 2D fracture connectivity of the unconstrained realisation is significantly lower than that of the reference outcrop.

In general, MPS algorithms generating discrete representations of fracture networks are sufficiently capable of capturing the underlying fracture geology. However, these methods are still not feasible for a reservoir-scale uncertainty quantification workflow involving fluid flow simulations. Running hundreds of realisations is too time demanding. This is enabled by the works of Jung et al. [2013], Arnold et al. [2016] and Looi et al. [2017]

who try representing fracture networks as continuous media properties for training images that can be used in MPS algorithms. The work done in this chapter very much ties into their findings. The aim for Jung et al. [2013] was to come up with a method for the rapid generating of a range of training images for naturally fractured reservoirs that cover the geological uncertainty associated with the fractures. Their focus is not on recreating completely accurate representations of the subsurface fracture distribution but more on building a set of large scale reservoir models that capture the fracture-related uncertainty on fluid flow response in a meaningful way. In the first step, they identify the uncertainty range associated with the potential subsurface fracture distributions. This uncertainty is then represented by generating an extensive set of discrete fracture networks (DFN) that represents the underlying fracture uncertainties.

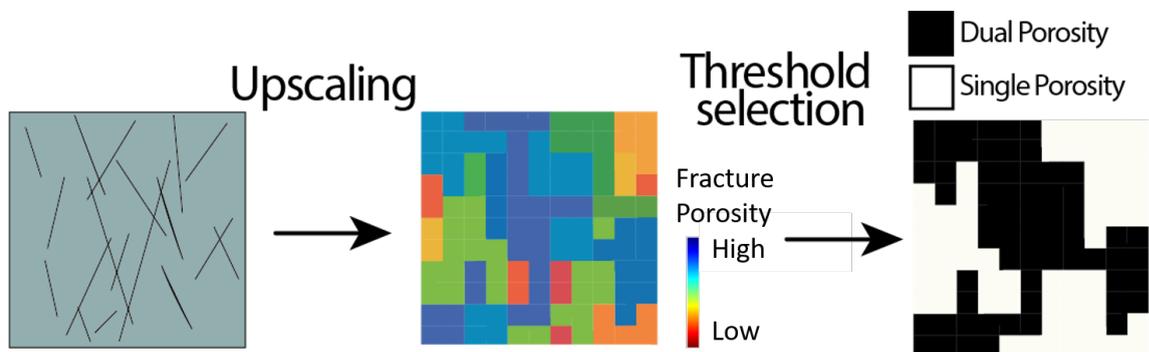


FIGURE 4.3: Conceptual DFN model (left) is upscaled to continuous media properties (porosity is pictured in the middle). Subsequently, a threshold on porosity is applied to determine how to model flow through every single cell (single or dual-porosity formulation).

The key to this technique lies in the upscaling of the DFNs to continuous media properties that can be used as training images for MPS (figure 4.3). When upscaling a DFN, every grid cell of the resulting upscaled medium will contain a different intensity of fractures. Based on a cut-off value related to fracture porosity, they determined whether a cell should be described as a dual-medium (matrix and fracture porosity modelled separately) or a single medium system (assuming that matrix and fractures can be lumped together into a single porosity model, without significant loss of information). The emerging binary patterns were then used as training images for MPS to populate the reservoir model with fracture distributions (figure 4.3). The underlying assumption is that fracture sets have a somewhat repetitive nature that can be captured and distributed over a larger area by the training image. The spatial variation of these fracture sets as reflected in the fracture connectivity

is caught within the generated binary patterns. In addition, one should be aware of the errors that are introduced by upscaling the DFNs, as pointed out by Elfeel et al. [2013]. To capture the underlying fracture uncertainty adequately without the need to use every single training image for reservoir modelling, Jung et al. [2013] applied k-medoid clustering to the binary patterns observed in the training images and selected a representative pattern from every emerging cluster to take forward to the reservoir modelling step (figure 4.4) [Dubuisson and Jain, 1994].

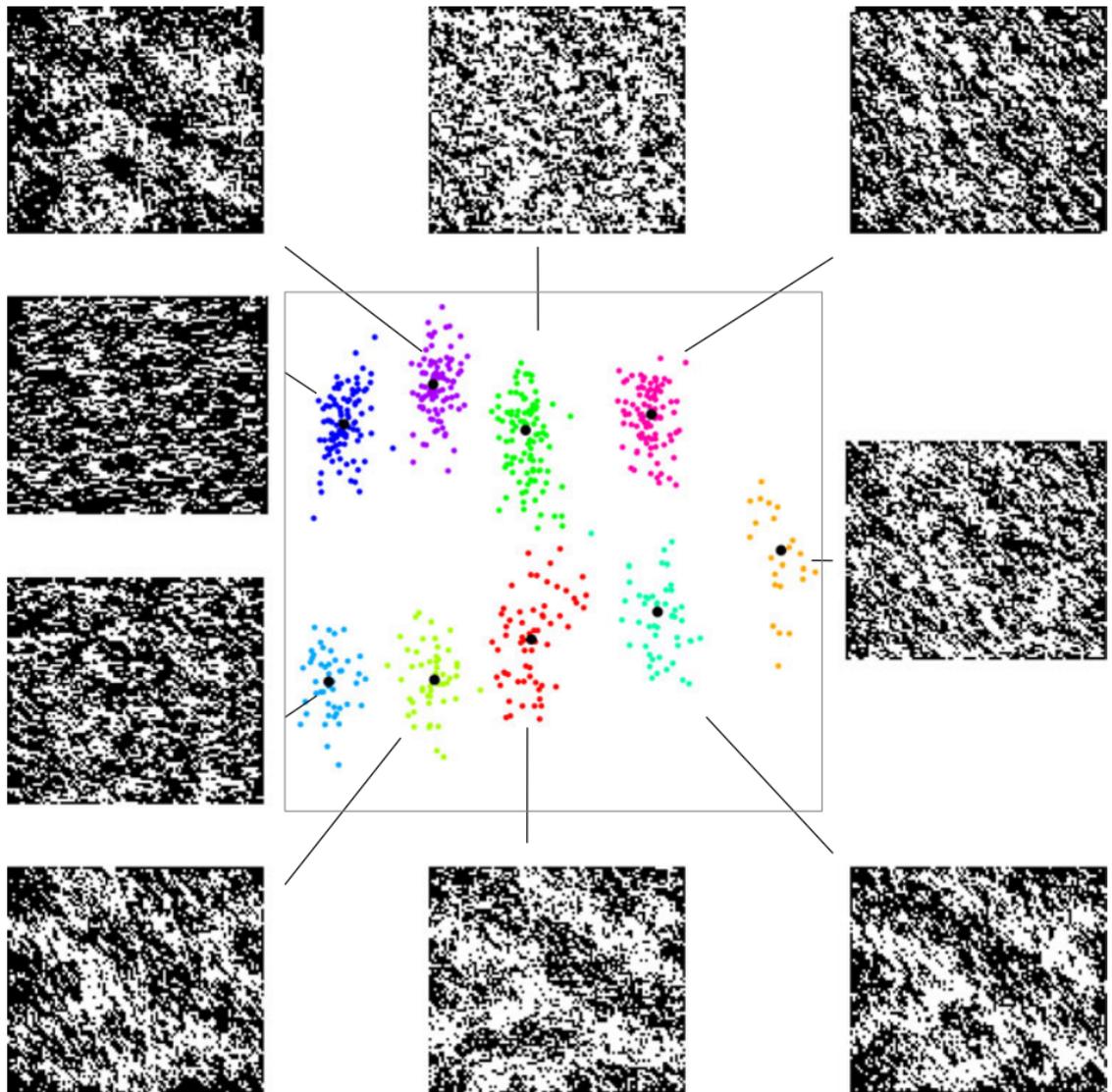


FIGURE 4.4: Visualisation of the modified Hausdorff distance of patterns from 624 binary training images projected into metric space with the help of multidimensional scaling. With the use of the k-medoids method training image clusters could be identified, and representative training images from each cluster can be selected (highlighted) [Jung et al., 2013]

## 4.3 Methodology

As mentioned in the previous section, the methodology for modelling naturally fractured reservoirs with the help of multiple-point statistics developed here is closely related to the work by Jung et al. [2013] and an extension of the work of Steffens et al. [2019]. The intention is not to model a set of exact depictions of fracture networks but to grasp the impact that fracture-related uncertainties can have on large scale fluid flow responses whilst still preserving a degree of geological realism. This is currently only possible when representing fractures as continuous media. The workflow developed in this chapter can be subdivided into nine steps that are sketched out in figure 4.5 and will be broken down further in the rest of this section:

1. Identify different zones of fracturing within the reservoir and assess uncertainty regarding fracture distributions within each zone.
2. Perform Monte-Carlo simulation to generate DFN models for each zone to capture uncertainty around fracture distributions.
3. Upscale DFN models to training images with continuous media properties (porosity and permeability).
4. Apply cut-off to decide which cells show matrix or a fracture dominated flow.
5. Run flow diagnostics for each training image with various well configurations.
6. Calculate time-of-flight-based entropy of 2x2 moving window for each training image
7. Apply dimension reduction (UMAP) on time-of-flight values of each cell for each zone.
8. Apply density-based clustering algorithm HDBSCAN to preprocessed data and select a representative set of training images from each cluster.
9. Use a combination of sampled training images from each zone to generate a set of diverse reservoir models that capture a similar range of fracture-related uncertainty as would be the case for building reservoir models from all training image combinations.

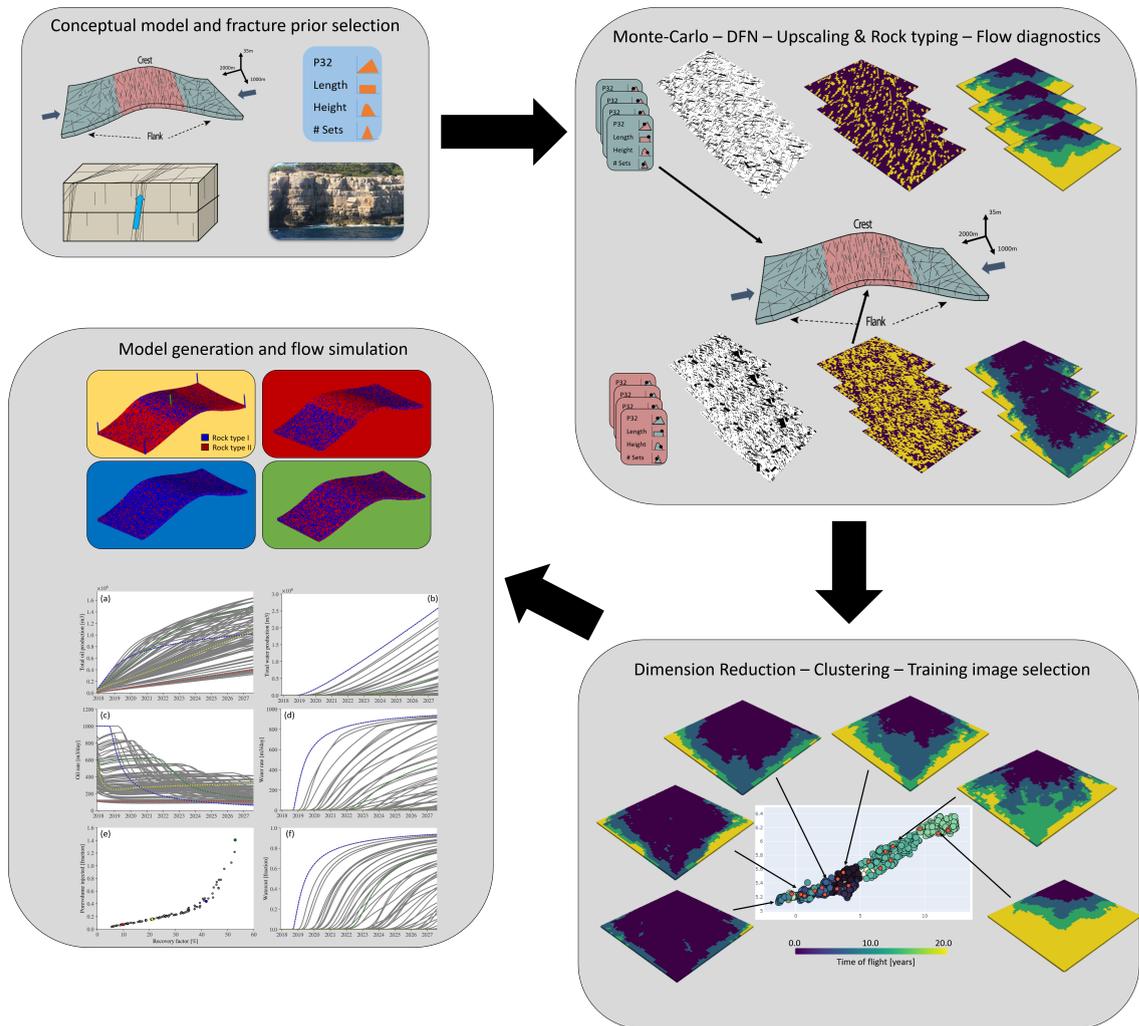


FIGURE 4.5: Overview of the proposed workflow using multiple-point statistics to model naturally fractured reservoirs.

### 4.3.1 Step 1: Geological analysis

The first step in this workflow does not differ from any standard geological analysis that aims to understand a fracture system better. As data usually are only sparsely available, a combination of different data types should be leveraged to shape a conceptual picture of the geology at hand. Potential data sources include literature, well data, analogue fields, outcrop data, lab and computer-based experiments, and a fundamental geological understanding to inform on geologically reasonable assemblies. As we will be using training images to capture fracture network distributions, we have to take the non-stationary nature of fracture networks into account. Therefore, the reservoir has to be divided into sub-regions in which we can assume stationary fracture networks. For each zone or sub-region, we then have to identify plausible fracture distributions that are aligned with

the underlying conceptual model and the associated input parameters required for DFN modelling (e.g. the number of fracture sets, fracture intensity, length, height, orientation, dip, azimuth and concentration).

### **4.3.2 Step 2: Monte-Carlo DFN simulation**

The geological input parameters required for DFN modelling are associated with a significant degree of uncertainty that can impact the fluid flow behaviour through fracture networks. To capture the impact of these uncertainties on the output, we decided to use Monte-Carlo simulation to generate hundreds of training images per sub-region. Automating the DFN generation process and keeping the DFN size within a range that is not too computationally demanding (here up to km scale), allows for the generation of a single model within minutes. Monte-Carlo simulation is a well-established methodology across multiple disciplines (finance, engineering and sciences) to randomly sample from a probability distribution of every uncertain input parameter [Metropolis et al., 1949]. This step is repeated for every identified sub-region in the reservoir.

### **4.3.3 Step 3: Upscaling**

Running fluid flow simulations for discrete fracture networks would be too time-consuming to compute within an uncertainty quantification framework on a reservoir scale. Therefore, the generated DFNs have to be upscaled to continuous media properties in this step. As pointed out by Ahmed Elfeel and Geiger [2012], Elfeel et al. [2013] and Karatalov et al. [2017], upscaling errors will be introduced during the DFN upscaling procedure and should be taken into consideration. A DFN can either be upscaled analytically or with the help of numerical, flow-based methods [Ahmed Elfeel and Geiger, 2012]. As flow-based processes are time-consuming to run, the analytical, more computational efficient Oda-Gold approach was chosen for upscaling in this work [Elfeel et al., 2013]. The Oda Gold method is based upon the Oda method [Oda, 1985]. Oda Gold represents an improvement on the initial Oda method, which assumes that fractures are of infinite length. This assumption of infinite fracture length consequently led to an overestimation of fracture network connectivity and therefore permeability. According to Elfeel et al. [2013] the Oda Gold method takes into account fracture length and delivers results comparable to

flow-based methods within a fraction of the time. Further details on upscaling techniques for fractures to continuous media can be found in chapter 3.

#### **4.3.4 Step 4: Grid cell classification / rock typing**

In this step, the upscaled models are simplified to training images with binary patterns consisting of grid cells. We either have grid cells with fracture-dominated flow or matrix-dominated flow (figure 4.6). When upscaling DFNs, the fracture intensity of each grid cell varies. At low intensities fluid flow through a grid cell is governed by the matrix properties as the fractures do not have a sufficient level of connectivity. With intermediate fracture intensities fractures start supporting flow through the matrix. At high intensities fractures built up an adequate degree of connectivity to channelise flow and turn into highways for fluid flow. We need to define a uniform criterion that allows differentiation between the two rock types - grid cells with fracture-dominated flow or grid cells with matrix-dominated flow. Based upon our investigations and the work done by Jung et al. [2013], fracture porosity is a good indicator for this differentiation. Grid cells with a fracture porosity greater than  $1.0e-05$  get assigned with rock type 1 (fracture dominated flow) and grid cells with a fracture porosity lower than  $1.0e-05$  get set with rock type 2 (matrix dominated flow). For simplicity, the permeability in x, y and z-direction for rock type 1 with fracture dominated flow equates to the mean fracture permeability in x, y and z-direction from all grid cells of rock type 1 of a training image. Grid cells with rock type 2 will also get a uniform permeability assigned to them that depends on the matrix properties. This classification allows highlighting of the effect that fracture networks have on fluid flow.

#### **4.3.5 Step 5: Run flow diagnostics**

In this workflow we try to understand what the impact of fracture-related uncertainties on fluid flow is. Therefore, when selecting a suitable subset of training images to represent this uncertainty, it is good to know the flow behaviour of every training image before performing the multiple-point statistics modelling step. We use flow diagnostics to quickly get a good indication of the range of fluid flow behaviour covered by all training images. Fractures are highly anisotropic, tabular features. This causes them to focus fluid flow and have the

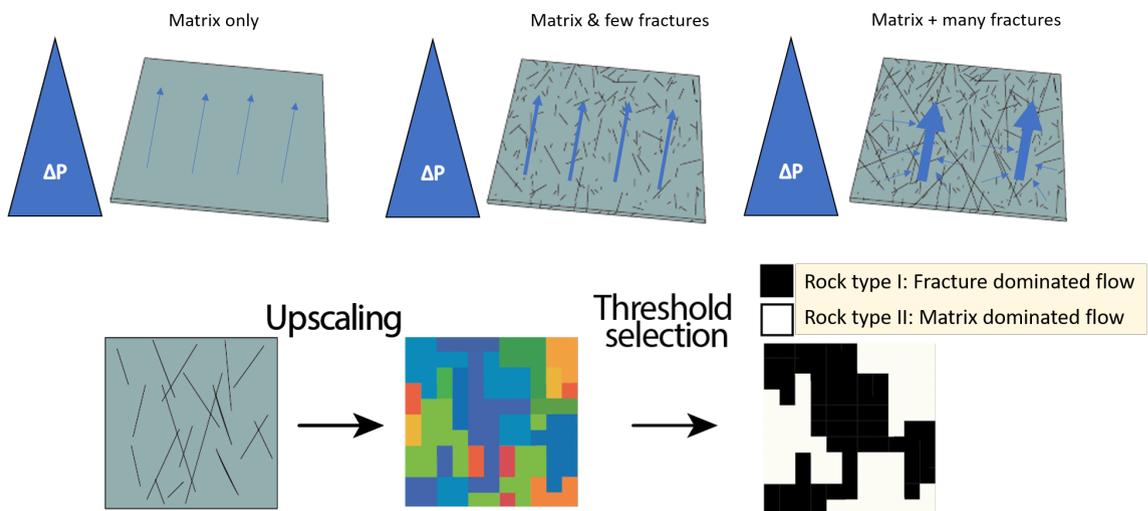


FIGURE 4.6: Top: From left to right: By increasing fracture intensity, initially matrix dominated flow that eventually gets supported by fracture permeability will ultimately turn into more localised, fracture dominated flow. Bottom: Upscaling of DFN to continuous media properties with subsequent differentiation between cells with matrix dominated and fracture dominated flow.

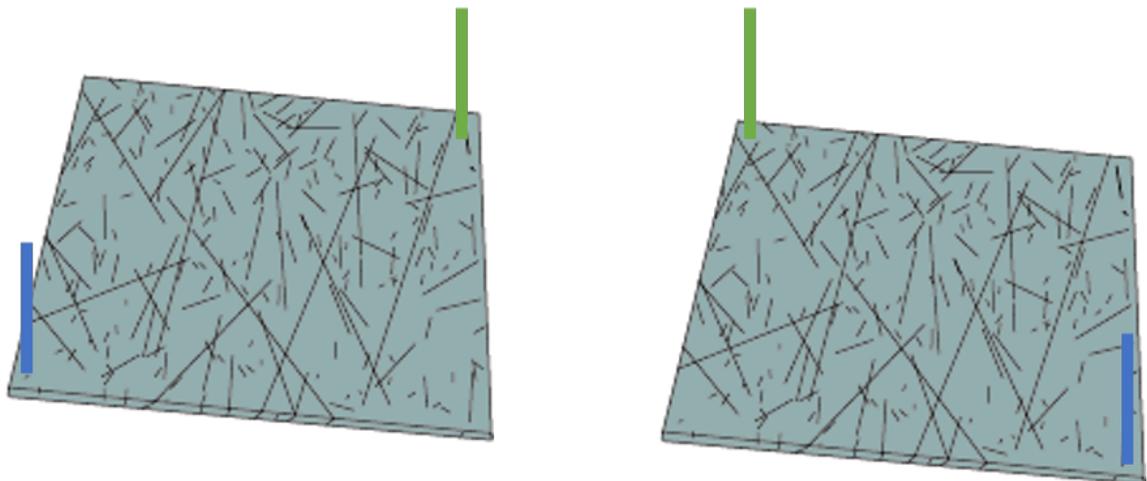


FIGURE 4.7: Flow diagnostics are run with multiple producer well (green) and injector well (blue) configurations per training image to capture the impact of fracture anisotropy on fluid flow.

potential to make it highly directional. To capture this direction dependency and reduce the impact of a single well configuration, numerous well configurations are tested for each training image (e.g. figure 4.7).

### 4.3.6 Step 6: Calculate training image entropy

In chapter 3 section 3.4 we already mentioned that Shannon entropy is a good metric to quantify diversity [Shannon, 1948]. Here we make use of spatial entropy as it was

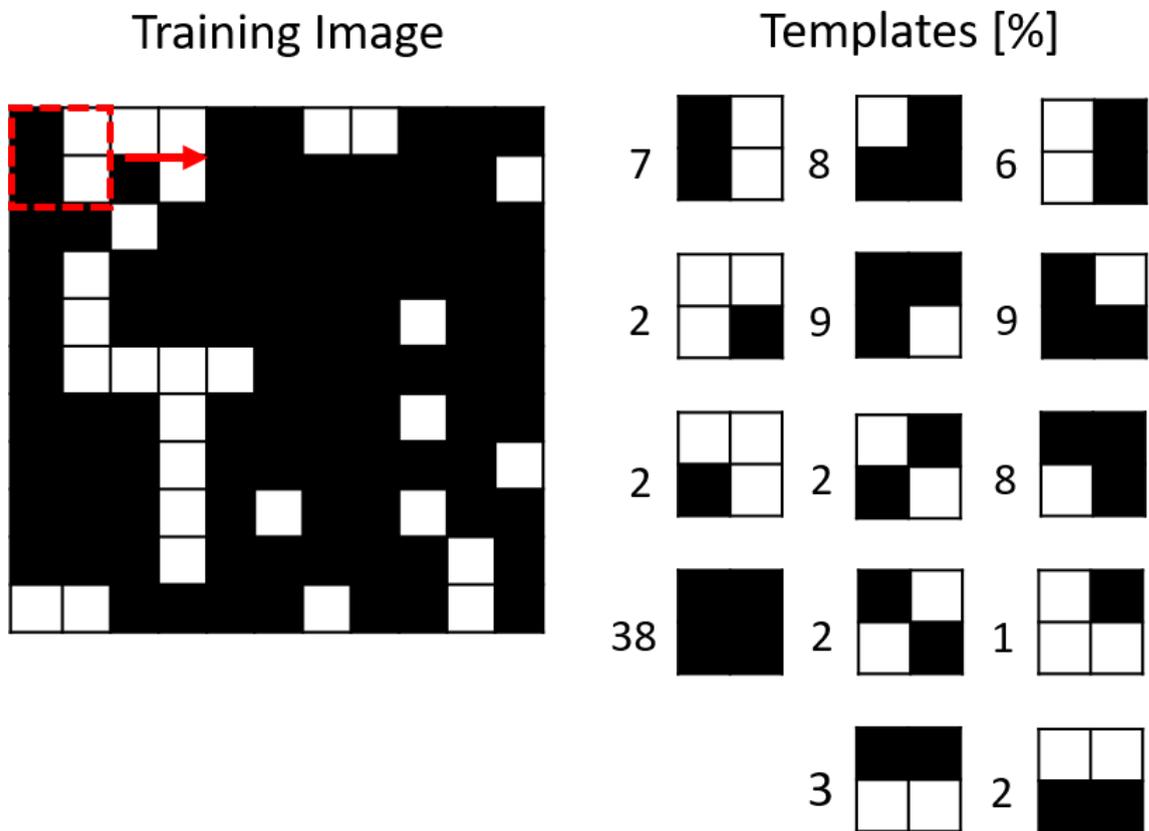


FIGURE 4.8: A schematic binary training image that is scanned with a 2x2 moving window (red). The moving window detects unique templates that make up the training image and also counts their frequency. These data can then be used to calculate a training images entropy.

proposed by Journel and Deutsch [1993]. With the help of a 2x2 moving window, we scan through a training image and count the number of unique templates and their frequency that would be required to rebuild the training image (figure 4.8). This information is then used to calculate the spatial entropy for each training image. Instead of calculating the entropy for the static training image properties (e.g. porosity and permeability), this work makes use of the time-of-flight data we calculated in step 5. The time-of-flight data is binned into four predefined bins, so each cell of a training image is categorised as either experiencing, low, medium, high or very high sweep. Without binning the number of unique templates required to rebuild the training image’s sweeping pattern would become overwhelming and the calculated entropy meaningless. This step is repeated for every well configuration tested out in step 5.

### **4.3.7 Step 7: Dimension reduction**

Next, a clustering algorithm will be applied to the time-of-flight data of each training image. To enhance the effectiveness of the clustering, we first apply a dimension reduction technique to all the time-of-flight data of the training images. We decided to use UMAP (Universal Manifold Approximation and Projection) as it balances the preservation of local versus global data structures and runs quickly (see chapter 3 section 3.5 for further details). It has also been demonstrated that if deployed together with the clustering algorithm we selected the final results are of better quality than without applying dimension reduction [Allaoui et al., 2020, Pealat et al., 2021].

### **4.3.8 Step 8: Clustering**

This step aims to select a representative subset of training images that covers the outlined geological uncertainties and their impact on fluid flow. The time-of-flight data preprocessed in step 7 and the entropy data calculated in step 6 will now be clustered with the help of HDBSCAN (Hierarchical Density-based Spatial Clustering of Applications with Noise, see chapter 3 section 3.6 for further details). The great thing about HDBSCAN is that we are not required to predefine the number of clusters we expect to see in our data and the shape of our cluster is also unimportant. From each individual cluster we can now select  $n$  training images to be used in the final step.

### **4.3.9 Step 9: Modelling and flow simulations**

With a representative subset of training images for each zone that capture the uncertainty around fracture distributions and their impact on fluid flow the final step is the generation of reservoir models. Depending on how many models are feasible to run, there are two possible options to proceed. The time-consuming step would be to go through all possible training image combinations for each zone and then use multiple-point statistics to generate the reservoir models. Another option would be to decide on a maximum limit of reservoir models to generate and then randomly sample a training image from each zone, use multiple-point statistics for the modelling and generate the desired number of models. The generated ensemble of reservoir models should now efficiently capture the

uncertainties related to fracture distributions. Full physics flow simulations can now be deployed for the next steps.

## **4.4 Results**

### **4.4.1 Application to a synthetic reservoir**

In this section we will now apply the workflow described in the previous section to the synthetic reservoir that we introduced in chapter 3 section 3.7. The situation is the following: We would like to assess the potential fluid flow behaviour of a reservoir but are not aware of the property distributions within the reservoir. However, we have information on likely ranges for fracture distributions, different trends, rock type properties based on outcrop data, analogues, geomechanical simulations etc. This allows us to (i) outline areas where we assume different types of fracturing and (b) quantify the range of possible fracture distributions within each zone. This allows us to follow each step of the previously described workflow to eventually capture the impact of fracture-related uncertainties on fluid flow and check if this range is in line with the actual response of the synthetic reservoir.

### **4.4.2 Step 1: Geological analysis**

Based upon the information given in chapter 3 section 3.7 it was found that the anticlinal reservoir (2000m x 1000m x 35m) consists of carbonate rocks of low reservoir quality (low porosity and permeability) and has seven mechanical units with bed-bound fractures. The reservoir can be subdivided into two different zones of distinct fracturing. The crestal zone and both flanks (figure 4.9). In both zones it is assumed that the fracture distributions are stationary. To successfully perform DFN modelling we now have to define a set of parameters that characterise the fractures to be generated. The parameters are fracture density (P32), fracture length (scale and shape), maximum fracture length and fracture orientation (dip, azimuth and concentration). Fracture aperture is kept constant at 0.0003m. This characterisation has to be done for every fracture set and for fractures specific to the crest, the flank and generic background fractures that are not associated

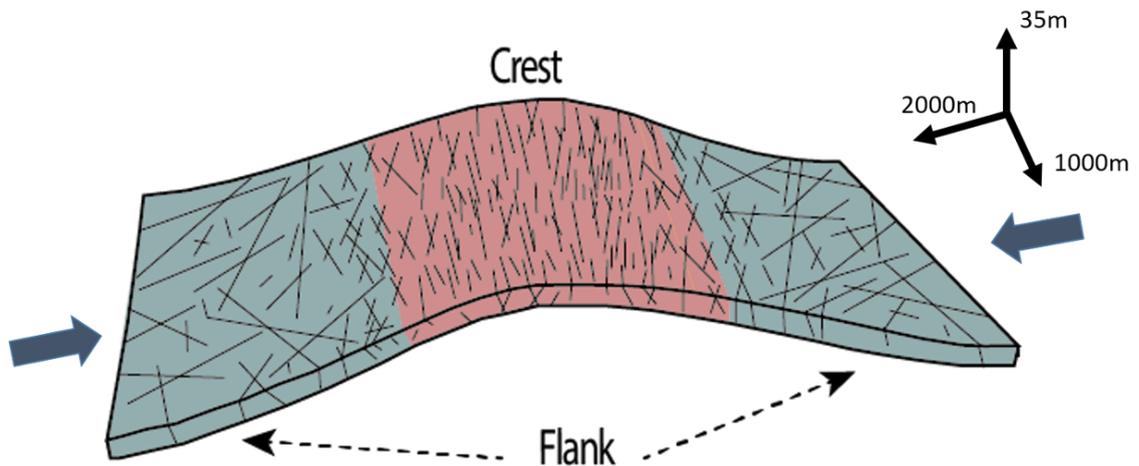


FIGURE 4.9: The conceptual fold model based on the Teton anticline model for fold-related fracturing. In the red zone around the fold centre, crestal fractures are short and parallel to the fold axis. In the green zone, fractures are longer, more dispersed and oblique to the fold axis.

with the folding event. A summary of the ranges of possible parameters is given in table 4.1.

#### 4.4.3 Step 2: Monte-Carlo DFN simulation

For the Monte-Carlo simulation we have to acknowledge that there will always be a trade-off between computational time available and how well we are able to cover the impact that the uncertainties around fracture distributions given in table 4.1 have on fluid flow. By tracking how the standard deviation of the dynamic Lorenz coefficient changes with every additional generated training image, we get a good measure on how much extra coverage we get with additional training image. As can be seen in the example run in figure 4.10, after around 300 generated training images the standard deviation of the dynamic Lorenz coefficient reaches a plateau of around 0.042. It was therefore decided to go with a total of 600 iterations to account for a sufficient margin of error, whilst not over straining the computational time available. To make sure that the maximum fracture length of each potential fracture set (table 4.1) can be realised within the training image, whilst not costing too much time to generate, each training image measures  $600m \times 600m \times 5m$ . For each DFN training image we sample from either crest and the flank distributions and the background fracture distributions as shown in table 4.1. The most impactful parameters are the fracture density P32 and the number of fractures sets. Together they control how

### DFN Modelling Priors

Flank									
P32	# Sets	Set #	Fracture length			Orientation			
			shape	scale	maxlength	dip	azimuth	dispersion	
<b>Sampling Range</b>	uniform 0.01 - 0.1	uniform 1 - 4	1	uniform 2.1 - 2.5	uniform 5 - 25	uniform 50 - 500	uniform 75 - 90	uniform 150 - 180	uniform 0 - 50
<b>Sampling Range</b>			2	uniform 2.1 - 2.5	uniform 5 - 25	uniform 50 - 500	uniform 75 - 90	uniform 330 - 360	uniform 0 - 50
<b>Sampling Range</b>			3	uniform 2.1 - 2.5	uniform 5 - 25	uniform 50 - 500	uniform 85 - 90	uniform 170 - 190	uniform 0 - 50
<b>Sampling Range</b>			4	uniform 2.1 - 2.5	uniform 5 - 25	uniform 50 - 500	uniform 85 - 90	uniform 170 - 350	uniform 0 - 50
Crest									
<b>Sampling Range</b>	uniform 0.01 - 0.1	uniform 1 - 4	1	uniform 2.1 - 2.5	uniform 5 - 25	uniform 20 - 500	uniform 45 - 70	uniform 255 - 285	uniform 5 - 50
<b>Sampling Range</b>			2	uniform 2.1 - 2.5	uniform 5 - 25	uniform 20 - 500	uniform 45 - 70	uniform 75 - 105	uniform 5 - 50
<b>Sampling Range</b>			3	uniform 2.1 - 2.5	uniform 5 - 25	uniform 20 - 500	uniform 85 - 90	uniform 270 - 300	uniform 5 - 50
<b>Sampling Range</b>			4	uniform 2.1 - 2.5	uniform 5 - 25	uniform 20 - 500	uniform 85 - 90	uniform 240 - 270	uniform 5 - 50
Background									
<b>Sampling Range</b>	uniform 0.01 - 0.1	uniform 1 - 4	1	uniform 2.1 - 2.5	uniform 5 - 25	uniform 50 - 500	uniform 75 - 90	uniform 150 - 180	uniform 0 - 50
<b>Sampling Range</b>			2	uniform 2.1 - 2.5	uniform 5 - 25	uniform 50 - 500	uniform 0 - 90	uniform 0 - 360	uniform 0 - 50
<b>Sampling Range</b>			3	uniform 2.1 - 2.5	uniform 5 - 25	uniform 50 - 500	uniform 0 - 90	uniform 0 - 360	uniform 0 - 50
<b>Sampling Range</b>			4	uniform 2.1 - 2.5	uniform 5 - 25	uniform 50 - 500	uniform 0 - 90	uniform 0 - 360	uniform 0 - 50

TABLE 4.1: Overview of prior distributions used for the DFN modelling for fractures in the flank, the crest and overall background.

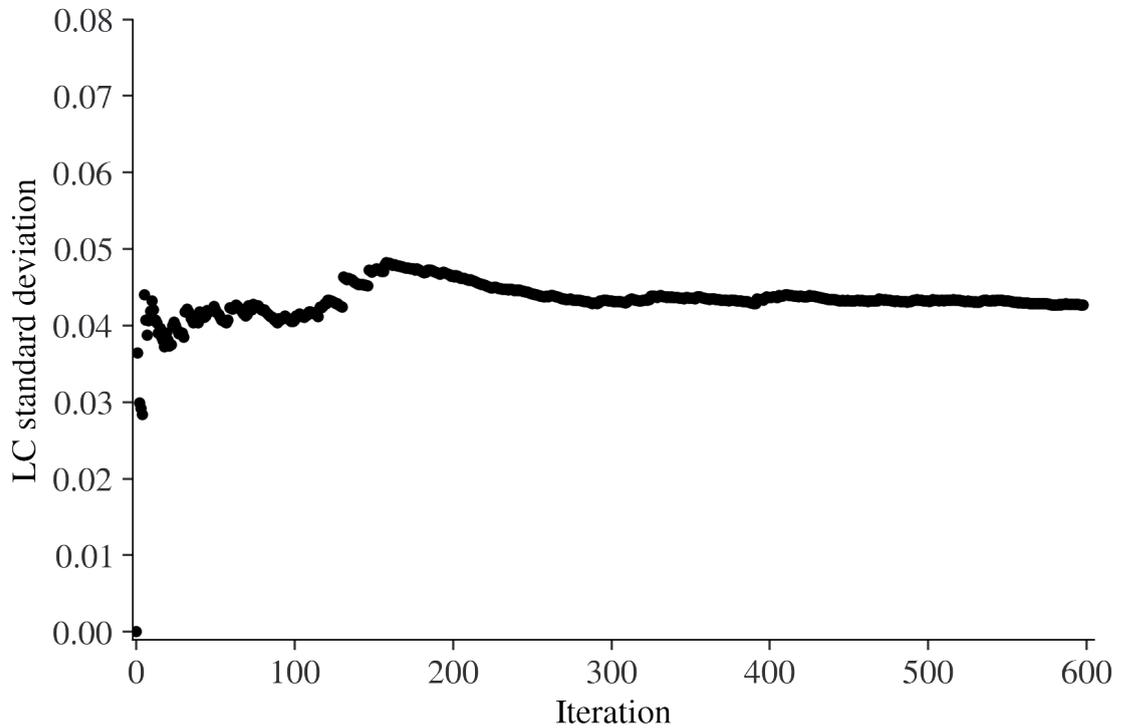


FIGURE 4.10: Development of standard deviation for of the dynamic Lorenz coefficient over iteration time. It can be seen that a plateau is reached after around 300 iterations and stays constant.

many fractures there are (P32) and how interconnected they are (number of fracture sets). In the left column of figure 4.11 we can see an exemplary collection of randomly selected DFN training images from the crestal zone and the flanks. Based on this small sample set alone we can see that there is already an extensive coverage of fracture-related uncertainties given.

#### 4.4.4 Step 3 + 4: Upscaling & rock typing

The DFN training images are now upscaled and a rock type is assigned to each cell. For simplicity, fracture porosity will be used as the differentiation criterion. Grid cells with porosity greater than  $1.0e-5$  will be of rock type 1 (fracture dominated flow) and grid cells with a lower fracture porosity will be of rock type 2 (matrix dominated flow). It should be noted, that an alternative parameter suitable for rock typing would have been fracture permeability. It takes into account fracture connectivity and fracture orientation. It however also adds complexity to this step, as fracture permeability is a tensor. This means that the rock type for example in X direction could be different to the rock type of

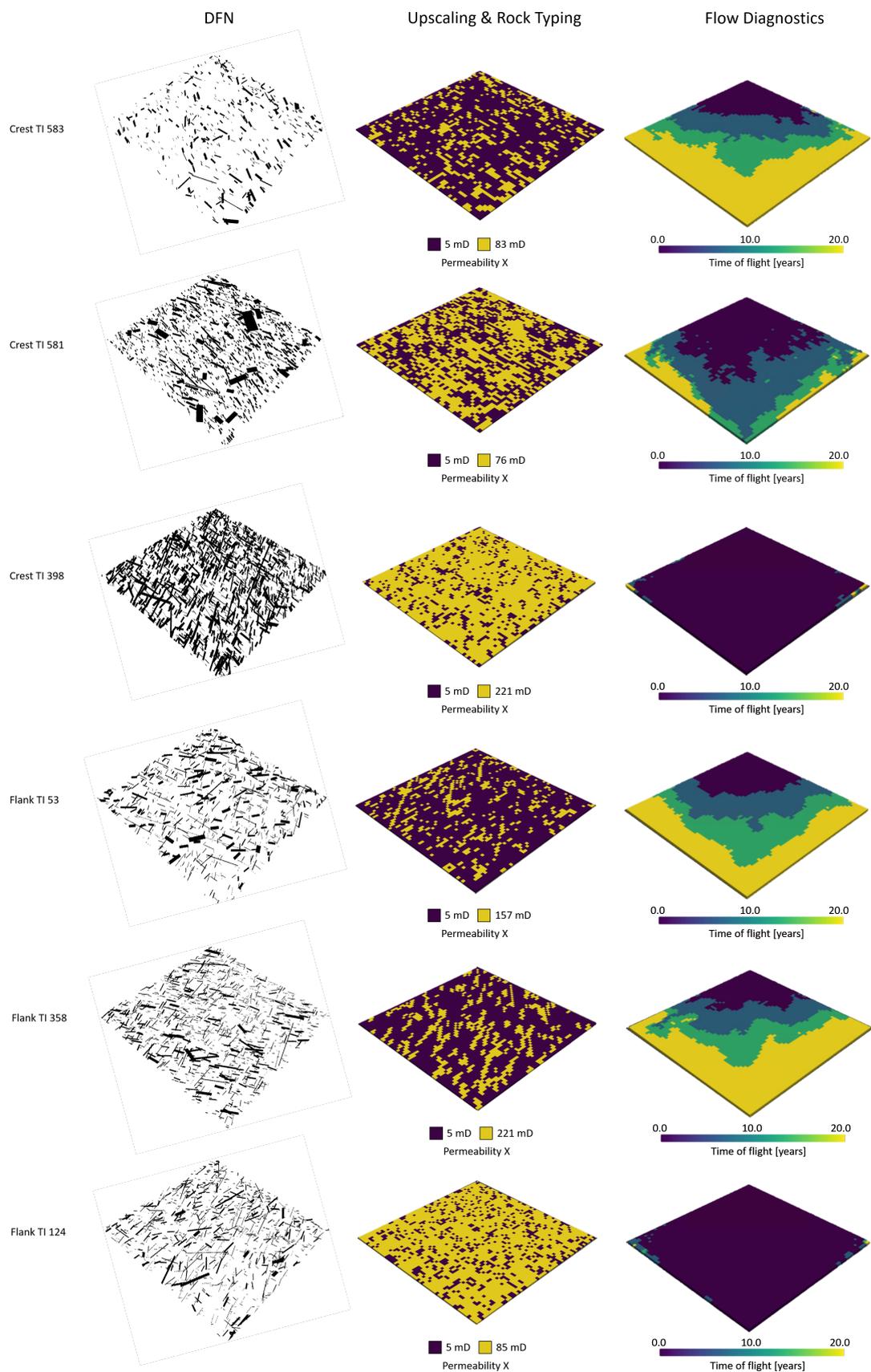


FIGURE 4.11: An exemplary set of training images from the crestal zone and the flanks. The left column shows the initial DFN that is then upscaled and transformed into different rock types (centre column). The right column shows the binned backwards time-of-flight distribution for a producer (top corner) - injector (bottom corner) pair.

the same cell in Y or Z direction. Based upon information given in Step 1 and chapter 3 section 3.7, It was decided that a constant, uniform permeability of 5 mD is sensible for rock type 2. In the centre column of figure 4.11 we can see how the distribution of rock type 1 vs rock type 2 varies with different fracture distributions. In the examples here only the permeability in the X-direction is depicted.

#### **4.4.5 Step 5: Running flow diagnostics**

We performed the flow diagnostic test with two well settings to better capture the directional effects that fractures can have on the fluid flow. The first test was run with the injector in the bottom corner and the producer in the top corner. The second test was run with the injector in the right corner and the producer in the left corner. In figure 4.11 in the right column the backwards time-of-flight for the first testing scenario is visualised. For visualisation and clustering purposes later on, the backwards time-of-flight data are organised into four bins (0-5 years, 5-10 years, 10-15 years and 15-20+ years of residence time). Based on this exemplary set of data, we can recognise that the effect of different initial fracture distributions on fluid flow one would expect from the DFNs is preserved during the upscaling and rock typing step. Training images with a high fracture intensity and interconnection show time-of-flight travel times that differ significantly from training images with low fracture intensity and interconnection leading to a vast range of different sweeping patterns for training images of the flank and the crestal zone.

#### **4.4.6 Step 6: Training image entropy**

To understand how the calculated time-of-flight-based entropy differs from training image to training image it is plotted against the proportion of rock type 1 (fracture dominated flow) per training image. This can be seen in figure 4.12 for the flank training images and figure 4.13 for the crestal training images. The colour-coding of each point will be explained later. What becomes clear is that both plots resemble a downwards facing parabola. Going from left to right on the X-axis, we can see a gradual, linear increase of the time-of-flight-based entropy with an increasing fraction of rock type 1. This increase hits a peak entropy of around 3.5, where we have a proportion of about 0.42 of rock type 1 in the training images. Up until this point the correlation between entropy and rock

type 1 proportion follows a very narrow band. The second limb of the parabola shows the inverted development. With a further increase of the ratio of rock type 1, the entropy starts to decrease, especially in the beginning of the second limb. It also shows a much wider spread of negative correlation. Towards the end, with very high proportions of rock type 1 this spread starts to decrease again, meaning that the entire training image is very much swept homogeneously. This indicates that we can be confident about how fractures will impact the heterogeneity of fluid flow in our reservoir below a certain threshold. In this case up to a proportion of around 0.42 of rock type 1, fractures seem to only have a supporting role for fluid flow within a training image. If we further increase the proportion of rock type 1, fractures start forming connected networks and we have more uncertainty about entropy and, therefore, how sweeping patterns will look like with a given fraction of rock type 1. This increases the difficulty of making good predictions on fluid flow. It will therefore be critical that the subset of training images that will be used for further modelling will contain training images from the second limb. This phenomenon is independent of well configuration and can be observed with both well settings. An explanation for this behaviour could potentially be found in the field of percolation theory [Broadbent and Hammersley, 1956, Essam, 1980]. Percolation theory can be used to understand the connectivity of specific entities within a system and what impact this connectivity has on the overall behaviour of the system. Of particular interest is the critical area where the so-called percolation threshold is overcome. This is the point where the specific entities form a network spanning the entire system. This causes the system to transition to a distinctively different behaviour before reaching the percolation threshold. This sharp transition point can be observed in various fields, such as biology, virology, ecology, traffic modelling, geology and multi-layer interdependent networks [Berkowitz, 1995, Boswell et al., 1998, Brunk and Twarock, 2021, Buldyrev et al., 2010, King, 1990, Lee et al., 2017, Li et al., 2015, Robinson, 1983, Thovert et al., 2017]. In our case it could be that flow through the training images switches from being controlled by cells of rock type 2 (matrix dominated flow) to cells of rock type 1 (fracture dominated flow). While increasing the proportion of rock type 1 from 0.06 to around 0.42, the overall sweep is gradually enhanced, leading to more diverse sweeping patterns picked up by the entropy. It does however not matter too much what the exact properties of rock type 1 are. They merely act as enhancers of fluid flow. At a proportion of around 0.42, this behaviour switches over to a more unstable, rock

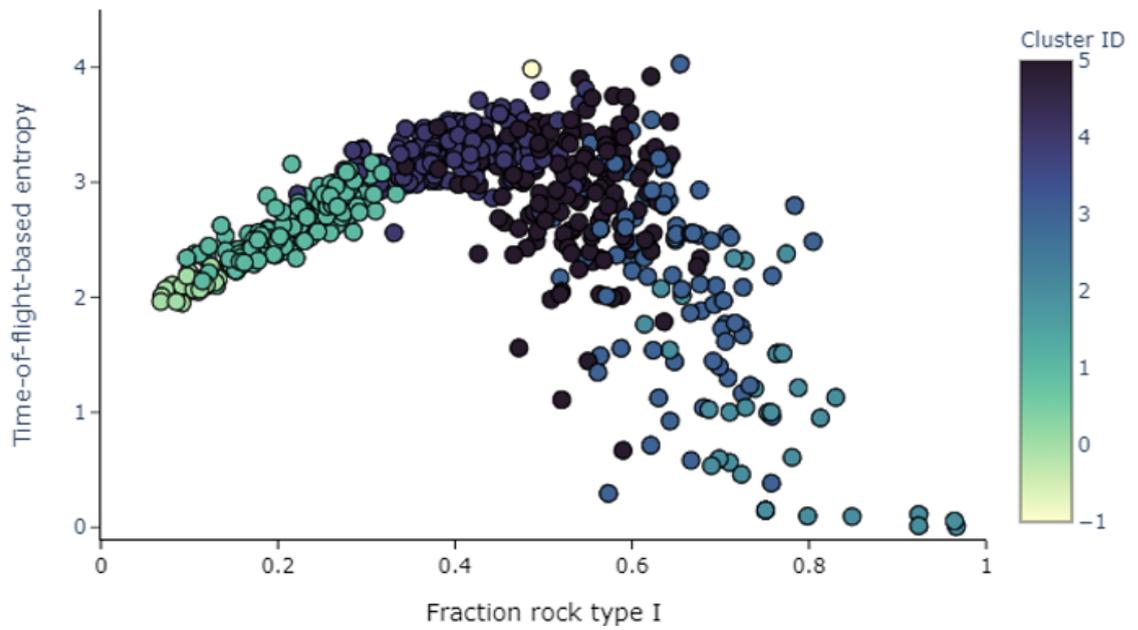


FIGURE 4.12: Time-of-flight-based entropy plotted against the proportion of rock type 1 for each training image of the flank. Colours match the clustering results from step 8.

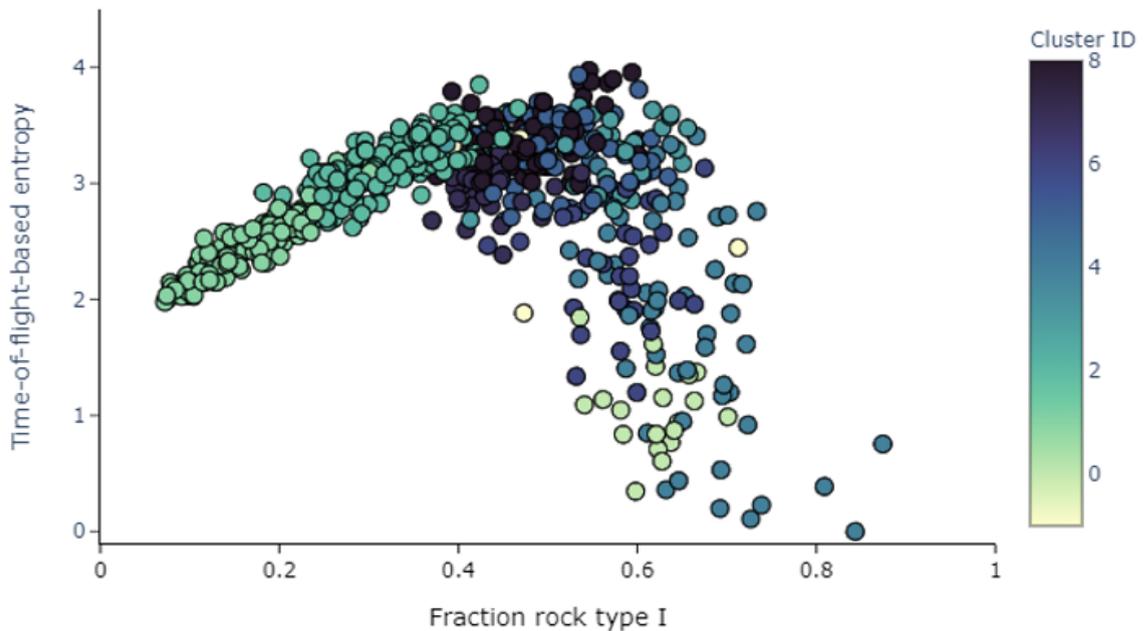


FIGURE 4.13: Time-of-flight-based entropy plotted against the proportion of rock type 1 for each training image of the crest. Colours match the clustering results from step 8.

type 1 dominated flow. Fracture properties such as the anisotropy of fracture permeability become more critical and cause the broader spread of entropy for training images with a similar fraction of rock type 1. In future work, percolation theory could potentially also be used to distinguish between the rock type assigned to a cell.

#### 4.4.7 Step 7 + 8: Dimension reduction + clustering

As already highlighted in the previous step, there are training images that show very similar characteristics regarding how they get swept. Using multiple of those training images for reservoir modelling would be of minor use to quantify the impact that fracture distribution uncertainty has on fluid flow behaviour. Therefore, in this combined step of dimension reduction with UMAP and clustering with HDBSCAN of the training images based on their time-of-flight distributions for both well settings and the time-of-flight-based entropy is performed. The hyper-parameters modified from their default for UMAP were:  $n\_neighbors = 20$ ,  $min\_dist = 0$ ,  $n\_components = 5$  and for HDBSCAN:  $min\_cluster\_size = 20$ ,  $min\_samples = 1$  and  $allow\_single\_cluster = False$ . In figure 4.14 and figure 4.15 one can see the clustering results for all training images from the flank and the crest respectively. It should be noted that repeating the clustering and dimension reduction steps numerous times always revealed comparable outcomes. In both cases the final results from the dimension reduction resemble elongated bodies that clustered into individual snippets. This continuity of these bodies most likely stems from the fact that we use a Monte-Carlo sampling approach for the training image generation and, therefore, covered the entire band-width of potential training images. However, with the help of clustering, we can now identify the training images that get swept similarly and, therefore, do not need to test them all in the reservoir modelling step. This is highlighted by the samples taken from each cluster for both the flank and the crest in figure 4.14 and figure 4.15 respectively. Here it also becomes clear that the clusters are arranged so that clusters with similar sweeping patterns are adjacent to each other. Interestingly, when colour coding the plots of the time-of-flight-based entropy vs proportion of rock type 1 (figure 4.12 and figure 4.13) with the clusters IDs that were picked up in this step, there is a very close resemblance to the results of this step. Apart from the training images with a very high fraction of rock type 1, training images of the same cluster are grouped together. The same also holds when considering the neighbourhood. Clusters next to each are also situated next to each other on the entropy vs proportion of rock type 1 figures. Picking up similar characteristics with both techniques gives us more confidence about selecting an appropriate subset of training images for the final modelling and flow simulation step. A total of 20 training images out of the 600 training images were sampled for each training image zone. We sampled

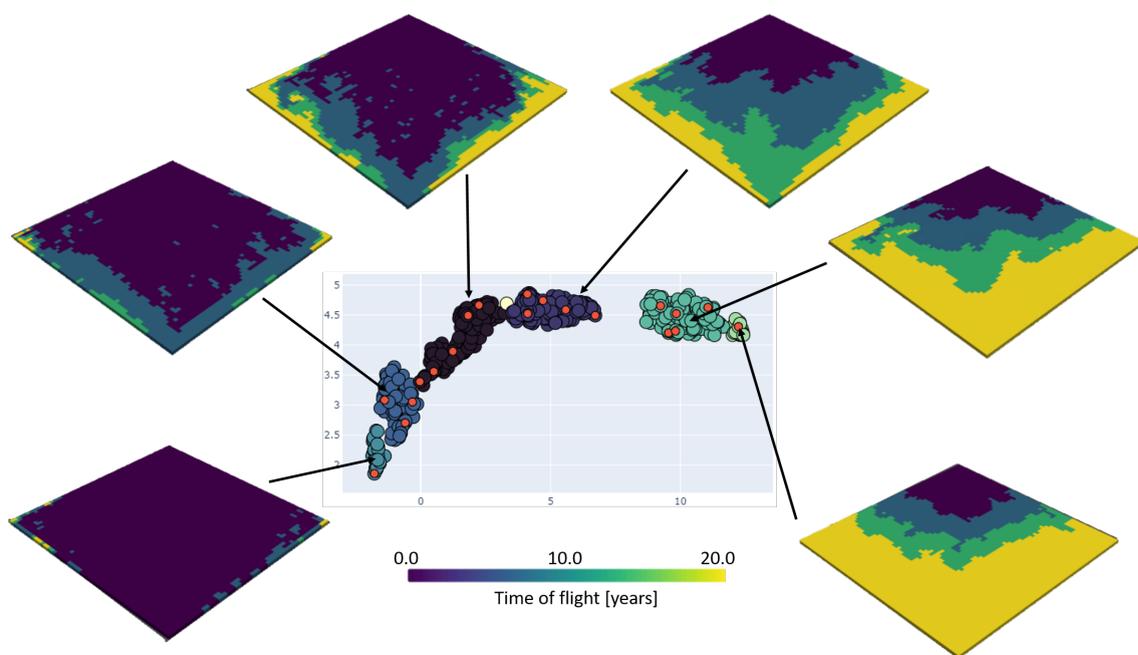


FIGURE 4.14: Time-of-flight-based clustering of each training image of the flank. Colours match the associated cluster.

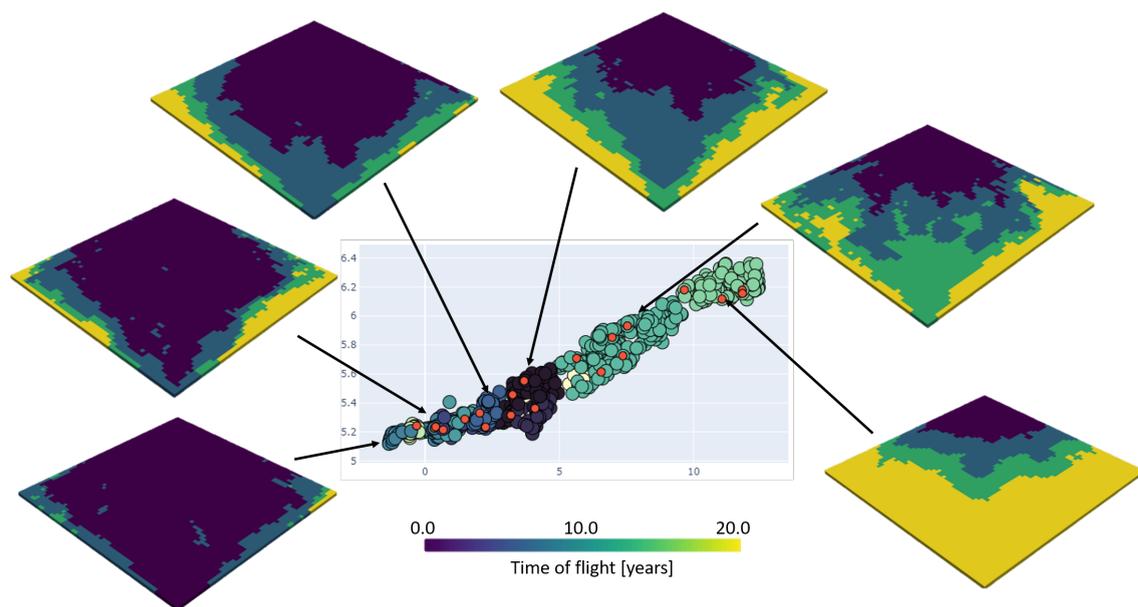


FIGURE 4.15: Time-of-flight-based clustering of each training image of the crest. Colours match the associated cluster.

randomly from each cluster, proportional to their size. The training images that we picked are highlighted in red on figure 4.14 and 4.15.

#### 4.4.8 Step 9: Modelling and flow simulations

With this subset of training images we can now build the reservoir models. The reservoir models measure 2.000m x 1.000m x 35m and are modelled with 200 x 100 x 7 grid cells that measure 10m x 10m x 5m. That results in a total of 140.000 grid cells. Due to the restrictions of the tNavigator academic license, higher resolution models are not possible. We decided to build a total of 100 models to capture the fracture-related uncertainties. An additional layer of uncertainty is the exact extent of the crestal and the flank zones and the width of the transitional area between both zones. In the transitional zones properties can either be generated with the selected crestal or flank training images. We sampled every reservoir model from the parameters defined in table 4.2, with an example given in figure 4.16. The simulations will be run under black oil settings, a five-spot production pattern with one injector well in each corner and a producer well in the centre of the crest over a total of ten years.

<b>Zone extend priors [cell no]</b>				
	<b>Crest start</b>	<b>Crest end</b>	<b>Transition start</b>	<b>Transition end</b>
<b>Sampling</b>	uniform	uniform	uniform	uniform
<b>Range</b>	65 - 75	125 - 135	1 - 10	1 - 10

TABLE 4.2: Overview of prior distributions used to define the crestal and flank zone extends in the generated reservoir models.

For comparison we also generated a "true" sample reference case by randomly sampling from the 600 training images generated with the Monte-Carlo simulation for crest and flank. We also randomly selected the extent of the crest and the flank as well as the transition zones. The reference case is sampled randomly from the initial training images to verify if this behaviour could be captured by the subset of models we generated. It should be noted that it is not reasonable to go through all possible combinations of training images for both zones as this would equate to at least 360.000 different models - not taking into account the uncertainty around the extend of each zone. The results of all flow simulations are summarised in figure 4.17. In each plot of figure 4.17, the green, blue, red and yellow colour corresponds to the three different production profiles and the reference case. The total oil production shown in figure 4.17a over the ten years ranges

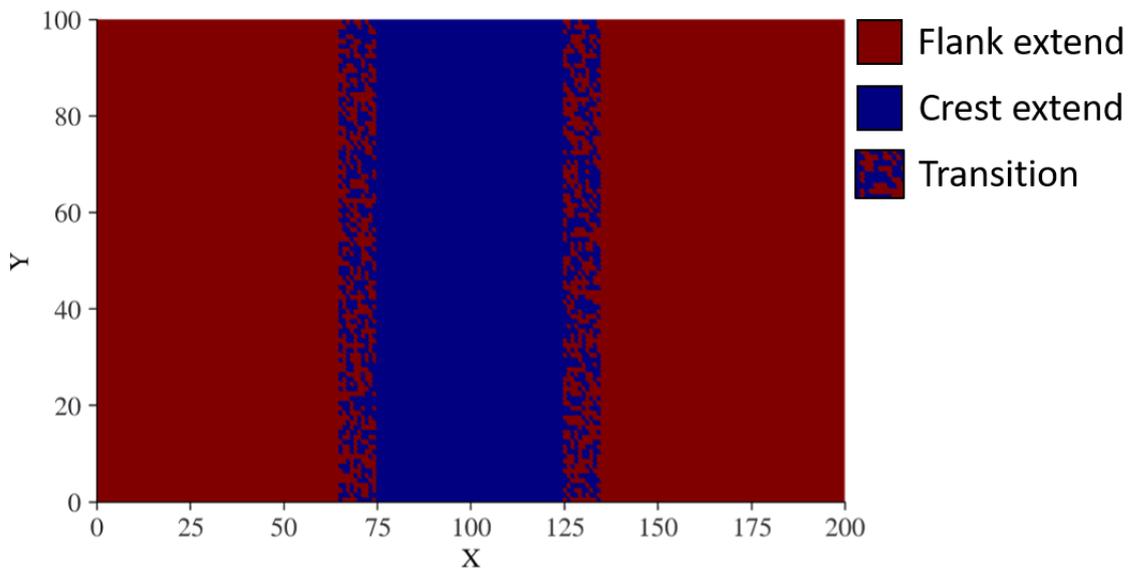


FIGURE 4.16: Example for the extent of the flank and the crestal zone and the extent of the transitional zone, in which area training images of both zones can be deployed.

from  $0.2 - 1.6 \times 10^6 \text{m}^3$ , equating to 5 – 55% oil recovery factors. It is interesting to see that there are at least three distinct patterns of how the reservoirs behave under production. The first pattern is highlighted in blue on figure 4.17 and figure 4.18. Reservoir models of this type show an initial very high oil production at rates of up to  $1000 \text{m}^3$  per day which quickly declines and is replaced by a very high production of water. The very early water breakthrough in these reservoir models is likely localised through the highly connected system of cells with rock type 1 where fractures dominate flow in both flanks as well as the crest (figure 4.18). The second pattern, to which the reference case (yellow) as well as to some degree the green case fit start off at intermediate production levels that quickly drop off but can then be maintained or slightly increased over time. Models that depict this pattern correspond to models with the highest total oil production and showcase how the injector wells support a more homogeneous and constant sweep of the reservoir with no or only later stage water breakthrough. Indicative for reservoir models with this production pattern are models that have a high proportion of rock type 1 cells in the crest that cause the initial burst in production, but only have a moderate proportion of rock type 1 cells in the flanks (figure 4.18) that prevents an early water breakthrough to occur. The last observed pattern (red) is that of a constant, low oil production rate and low total oil recovery. A characteristic of these models is that they have an overall low proportion of rock type 1 cells and therefore much lower overall permeability (figure 4.18). Water breakthroughs

can also not be observed in this model type. Figure 4.17e depicts the recovery factor of every reservoir vs the overall pore volume of water injected. It is interesting to see that from a recovery factor of 5 – 35% there is a constant, linear increase of the pore volume injected with every increase of recovery factor. Beyond that point, with around 0.45 pore volumes of water injected the increase in the recovery factor increases slower with a maximum recovery factor of around 50% at 1.45 pore volumes of water injected.

Overall, it seems like the subset of reservoir models produced with this workflow can cover an extensive range of fracture induced uncertainty in flow behaviour. Based on the initial fracture related uncertainty, it allows for the classification of the reservoir into three potential scenarios that will all have different development decisions. It is noteworthy that these three different types of possible scenarios were also picked up in the step where we had a closer look at the time-of-flight-based entropy of all training images (see figure 4.12 and figure 4.13).

## 4.5 Discussion

In this chapter, we presented a fast and efficient workflow to assess the impact that fracture-related uncertainty has on fluid flow. The aim was to preserve geological realism in NFRs, propagate geological uncertainties throughout the entire modelling workflow, and capture this uncertainty within a feasible set of reservoir models. Therefore, the initially generated DFNs are upscaled to continuous media properties that are subsequently screened for their dynamic flow behaviour with the help of flow diagnostics. The resulting time-of-flight distributions and the underlying entropy of each training image are clustered to group training images with a similar dynamic behaviour together. This allows us to sample from each cluster to generate a smaller subset of training images for multiple-point statistics that still cover the full range of dynamic responses we recorded. With these training images we can then create a set of reservoir models that efficiently capture the impact of fracture-related uncertainty on fluid flow on reservoir scale. By using multiple-point statistics instead of DFNs for the fracture modelling on reservoir scale, reservoir models can be generated within a fraction of the time it would take with DFN modelling. Whilst not being as geologically realistic as a DFN realisation might be, it makes this approach a

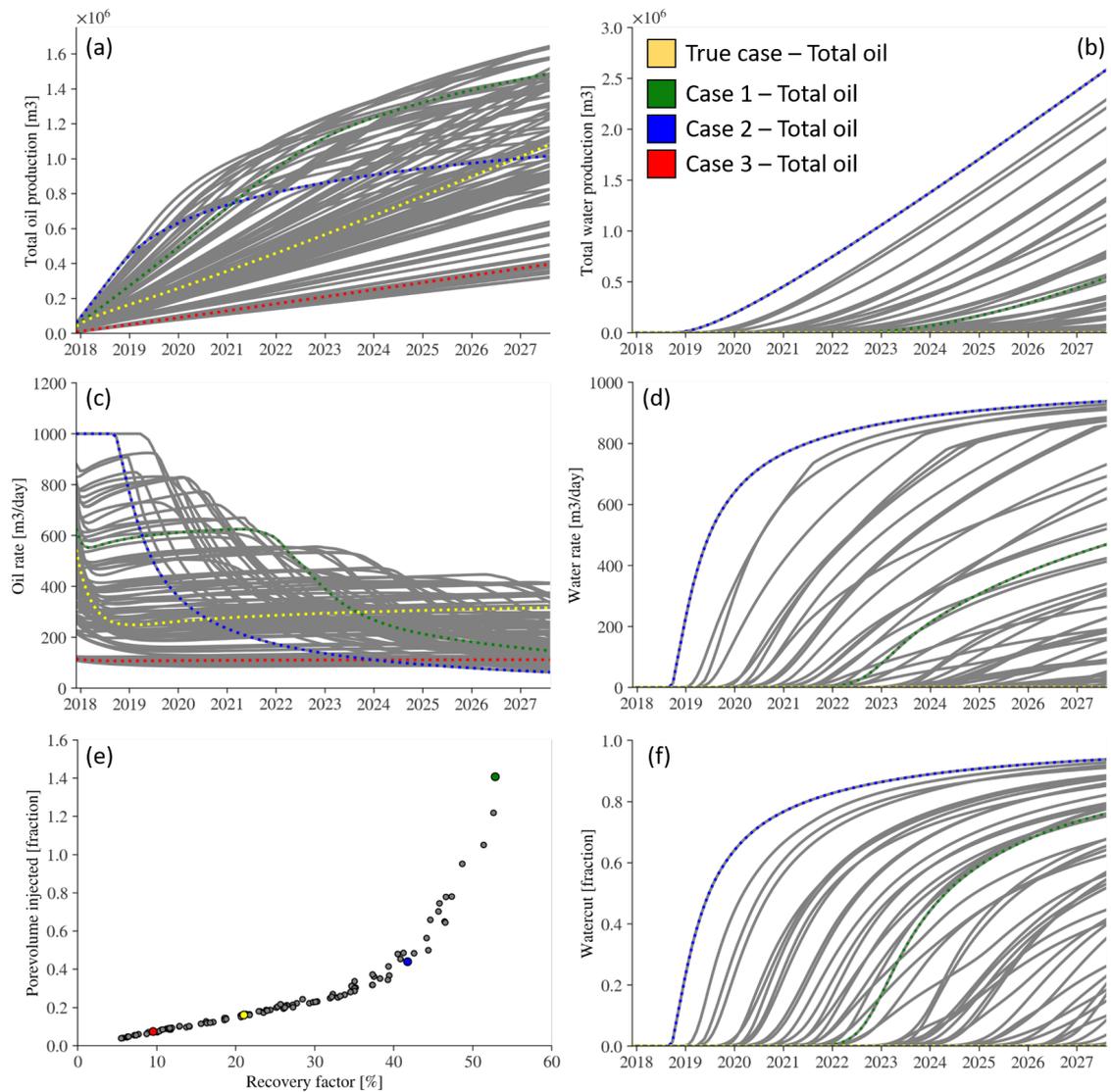


FIGURE 4.17: Results of flow simulation for 100 models displaying total oil production (a), total water production (b), oil production rate (c), water production rate (d), pore volume of water injected vs oil recovery factor (e) and water cut of the producer well (f).

lot more feasible for practitioners to capture a wide range of fracture-related uncertainty with a large set of reservoir models.

An assumption that is critical for the validity of the combined DFN-MPS approach is the assumption that fracture networks are stationary or have a representative elementary volume (REV) within each defined zone. The REV describes the minimum scale at which the distribution of discrete, heterogeneous properties (individual fractures in this case) becomes statistically homogeneous [Hill, 1963] and is the underlying anchor point for continuum modelling techniques [Bear, 1972]. Whether a concept of REV exists for fracture network or not is, however, still an unresolved issue open for discussion [Bonnet

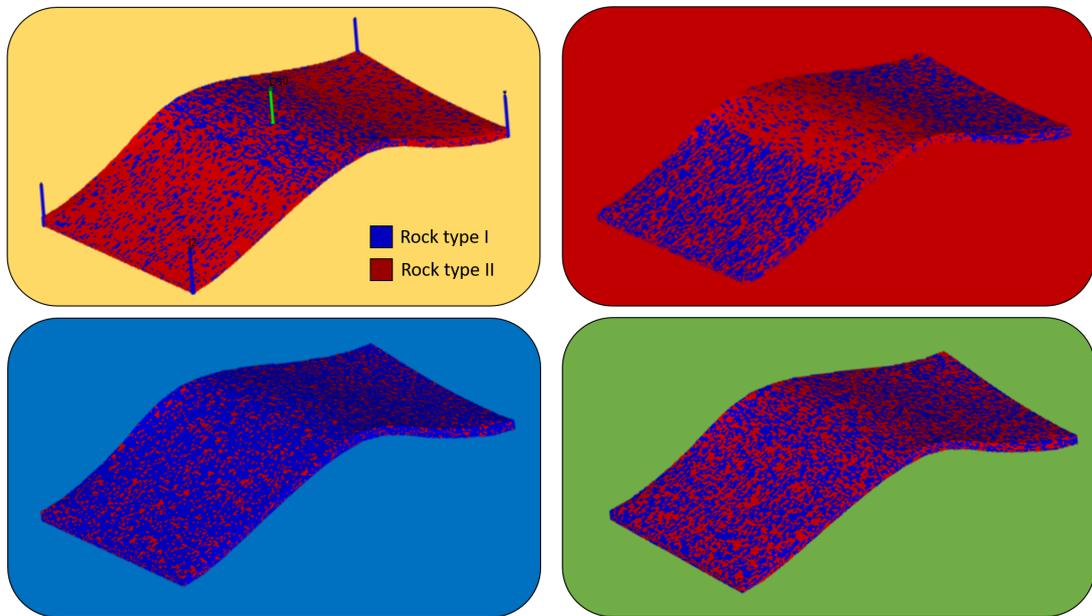


FIGURE 4.18: Examples of models generated. The background colours refer to the reference case (yellow), and the three distinct patterns described in the results (green, blue, red). Rock type I (fracture dominated flow) and rock type II (matrix dominated flow) are depicted for each model. In the reference case, the producer well (green) and the injector wells (blue) are also shown.

et al., 2001, Manocchi et al., 1999, Min et al., 2004, Neuman, 1988, Odling et al., 1999, Seers, 2015]. The argument is made that whether or not a REV exists, strongly depends on the characteristics of the fracture network under investigation [La Pointe et al., 1996]. In reality, even if it can be established that an REV exists, defining the minimum REV is still challenging but mandatory for DFN modelling and the generation of training images [Cheng-Haw et al., 2007]. The MPS algorithm used in this work requires for training images to be stationary and would therefore fail if there was no REV. In this work the assumption is therefore made that fracture sets investigated here develop a somewhat repetitive nature that can be captured and distributed over a larger area by the upscaled training images that are generated from the DFNs.

It could also be argued that the training images and the resulting reservoir models do not represent fractures in a realistic geological way. However, if we consider that fractures by nature shear fractures and joint systems are highly discontinuous, sometimes follow very abrupt trends, or occur in dense swarms, the training images become a lot more feasible. Therefore, the same approach would only work to a limited degree in more continuous

systems where flow is not dominated by fracture networks, like for example in channelised fluvial system without fracturing.

When using the output data from the flow diagnostics to perform the clustering, we manage to translate the effect that uncertainties around static properties (fractures) have on the dynamic fluid flow behaviour of the system. We can neglect static uncertainties that have only a minor impact on fluid flow and focus on the most critical uncertainties. By investigating the relationship between the proportion of rock type one present in each training image with the time-of-flight based entropy, we were able to detect the training images that are passed the percolation threshold and represent a tipping point after which the intensity of fracturing together with the anisotropic nature of fractures have a substantial effect on a fluid flow. Here it becomes more challenging to predict fluid flow which makes uncertainties around fracture distributions more impactful. Including the time-of-flight-based entropy into the clustering process as an additional metric allows us to be more confident that these critical training images are represented in the final subset of training images. Clustering presents a good method to manage uncertainty by reducing the number of required training images to cover the uncertainty range related to the fracture distribution with a more manageable subset of training images. However, it should also be clear that the trade-off from clustering is that the robustness of the uncertainty assessment in decision making is reduced. However, if the alternative is to either go with a single base case and not take the uncertainty into account or spend too much time on the modelling part, then this is an acceptable trade-off.

Despite capturing the fracture-related uncertainties well, a downside of this overall modelling workflow is that it lacks the capability to significantly change the underlying conceptual geological model. We are bound by our initial assumptions about the extent of the crest zone and the flank zone. Adapting these, based, for example, on a dynamic flow response we expect to see from the reservoir is not possible with this workflow and we are very much bound to our initial assumptions about the conceptual geological model. Another issue is that the coupling of dynamic and static geological data only happens at the training image level. If we already have an idea about the potential flow behaviour of our reservoir model, it is currently not possible to take this into account with this workflow. The model itself does not understand the most sensible combinations of training images

and extents of the different zones, given the desired flow response. These, however, are problems that will be addressed in the upcoming chapters. And here the training images generated with the help of this workflow will be the fundamental building blocks.

# Chapter 5

## Entropy-guided and dynamic response driven modelling of naturally fractured reservoirs

### 5.1 Introduction

Some of the shortcomings of the workflow for training image generating and modelling described in chapter 4 are (i) the lack of flexibility when it comes to defining the extent of different reservoir zones within the model and (ii) the inability to integrate dynamic data directly into the reservoir modelling workflow. Most conventional reservoir modelling workflows face similar problems (figure 5.1): With the help of seismic, well and analogue data we get an idea about what kind of geology to expect in the subsurface, which allows us to develop a range of geological concepts to assess interpretational uncertainty. Based upon this initial assessment, we can then build hundreds of reservoir models that cover a large share of the geological uncertainty across multiple conceptual interpretations. Unfortunately that leaves us with thousands of reservoir models that are far too expensive to run flow simulations on and we need to downscale. The down-scaling step can, for example, be achieved with clustering techniques, where similar models get grouped together. This allows us to select a smaller subset of models that adequately cover the uncertainty range that the entire set of models covered. This is where we then run

flow simulations, condition the models to production data and try to match the models' performance with that of the true dynamic response of the reservoir. This is ultimately a stiff inverse problem that can lead to multiple solutions. Unfortunately, this is also the first time we either manually or with the help of an optimizer get control of the dynamic behaviour of our reservoir models. Many working hours are spent generating a range of reservoir models that try to capture geological uncertainties until this point. However, until the final stage of the workflow, the impact of these geological uncertainties on fluid flow is not assessed.

This chapter demonstrates two novel geological modelling workflows capable of generating reservoir models with a desired range or type of dynamic response quickly based on the particle swarm optimisation algorithm (PSO) and flow diagnostics. The new *agent-based* workflows aim to produce an ensemble of reservoir models that capture a desired production outcome (either a set of similar models with a specific production response or a diverse ensemble of production outcomes) efficiently, highlighting important geological features of the model and/or capturing the impact of geological uncertainties on the flow dynamics.

The chapter is organised as follows. In section 5.2 we propose the overall methodology of our workflow and present two variants of how entropy is embedded into it. One uses a local particle swarm optimiser with the entropy measure as a *restarting* criterion and the other includes entropy into each particle's decision-making process. In section 5.3 we again apply the workflow to the synthetic case study setup in chapter 2. Parts of the work carried out in this chapter are extensions of the work presented in Steffens et al. [2020] and Steffens et al. [2021b].

## 5.2 Methodology

The workflow developed in this chapter consists of three components: (i) parameterised geological model building, (ii) flow simulation and (iii) model updating/ optimisation (figure 5.2). Conventional reservoir modelling workflows are often sequential in their setup and the geological modelling step can get disconnected from the flow simulation step.

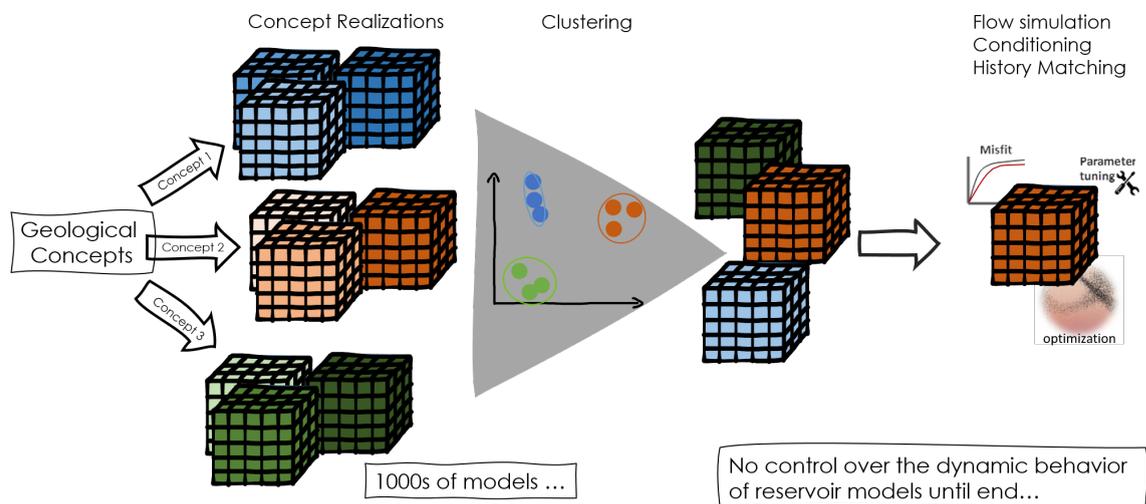


FIGURE 5.1: Example for a conventional reservoir modelling workflow.

In this work each component of the proposed workflow is interconnected and gets revisited iteratively. This allows constant feedback and optimisation of the geological model before history matching happens and enables the generation of dynamically conditioned geological reservoir models.

The great thing about this iterative reservoir modelling workflow is that the information about the dynamic behaviour of the reservoir model is integrated into the generation process of the reservoir models. This allows the building of geological reservoir models that either all have a similar dynamic production response but different underlying geology or have similar underlying geology but a diverse set of production outcomes - all depending on what type of uncertainty is to be investigated. By modifying the optimisation criteria of the particle swarm optimiser, the conditioned generation of reservoir models can be steered and controlled from the beginning onward and allow for a more focused investigation of the impact of geological uncertainties on fluid flow.

The idea of the workflow is now exemplified with the help of a simplistic illustration shown in figure 5.3. The aim is to create a model that reproduces the dynamic response of the synthetic "truth" case. It consists of fine sand in the upper half and coarse sand in the lower half. The coarse sand shows an order of magnitude higher permeability than the fine sand. One of the output metrics from flow diagnostics is the dynamic Lorenz coefficient (a measure of dynamic heterogeneity), which has a value of 0.32 for the synthetic "truth" case. In reality, we would not know the dynamic response of our reservoir, but depending

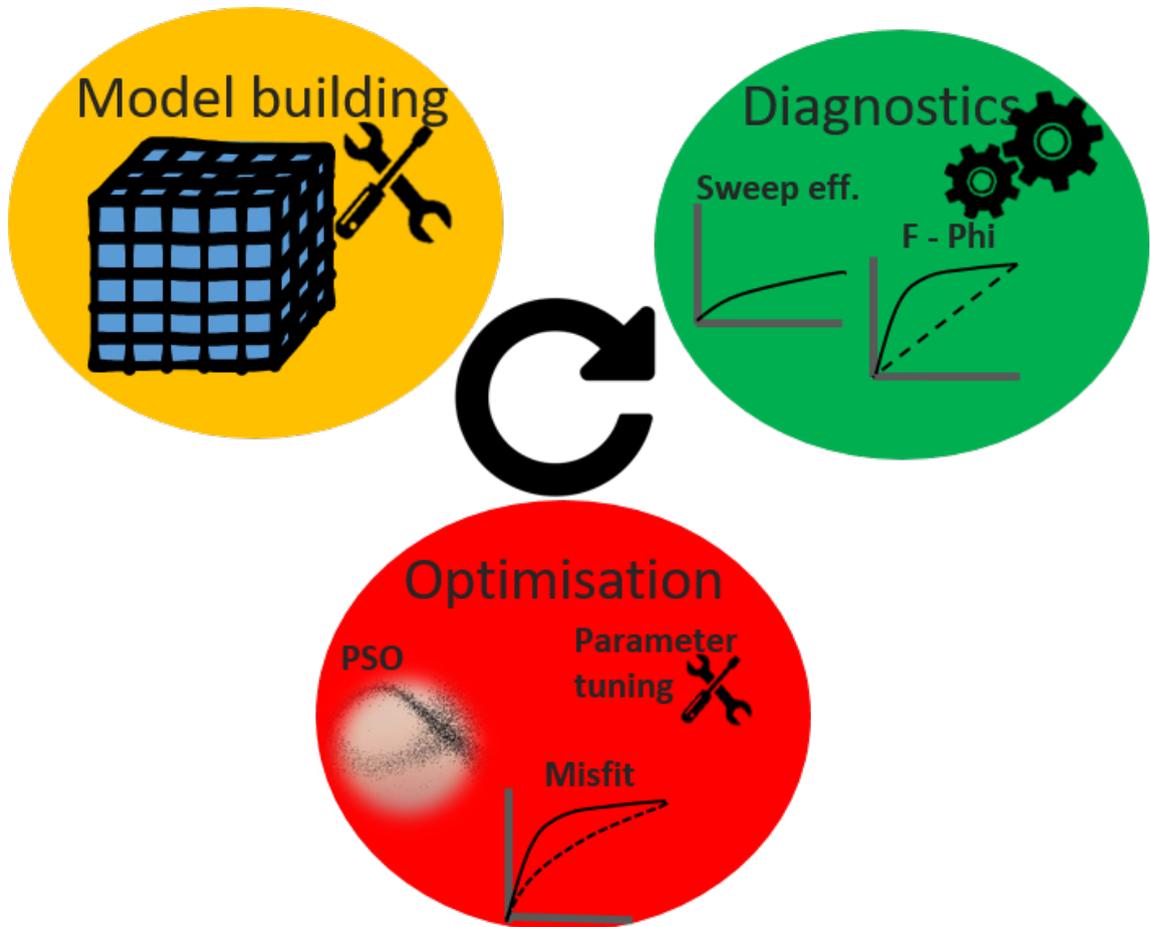


FIGURE 5.2: Schematic overview of proposed modelling workflow

on the data one has available, one could make a reasonable estimation of this response. Therefore, the first task is to think about what parameters you would need to modify to get a model that assembles the dynamic response of the “truth model”. In this case, fine to coarse sand distribution would be a good parameter to control.

The workflow now proceeds as follows: In the initialisation phase, the PSO selects a random value for fine to coarse sand distribution for each of the four particles representing a single model (figure 5.4). These models are then automatically built with the geomodelling workflow, and their dynamic response is calculated with flow diagnostics. This response is then compared to the objective of the optimisation algorithm, in our case a dynamic Lorenz coefficient of 0.3. The PSO then drives (combination of personal best, global best and random component) the particles towards the parameter configuration (fine and coarse sand distribution) of the model with the best fit to the objective. The models based on the new parameter configuration are then automatically built and tested on their dynamic

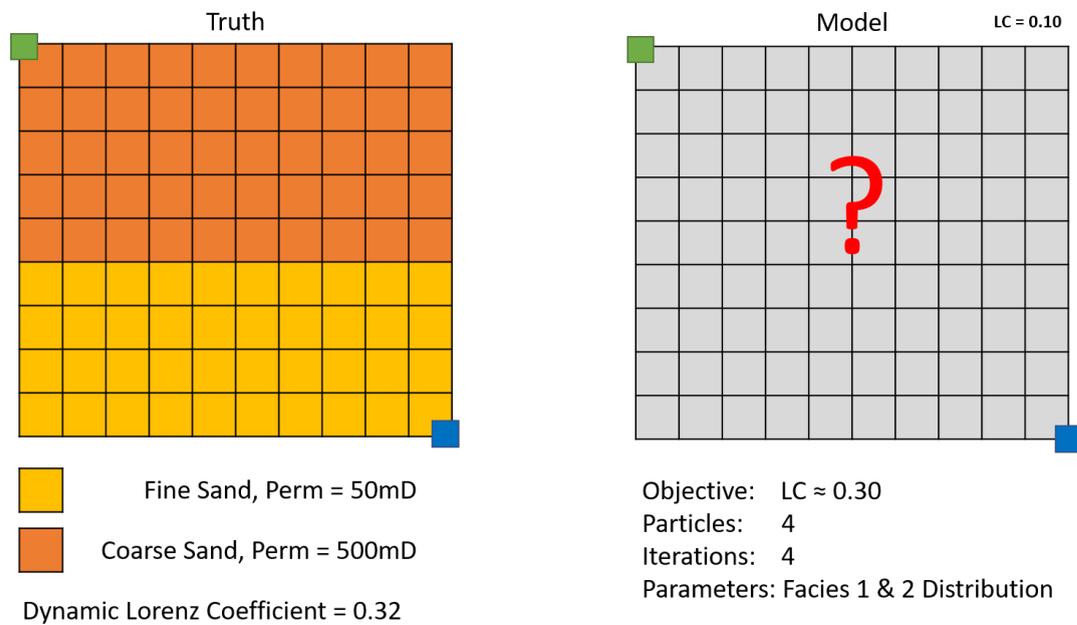


FIGURE 5.3: Left: Truth case; Right: modelling objective with the particle swarm optimisation settings.

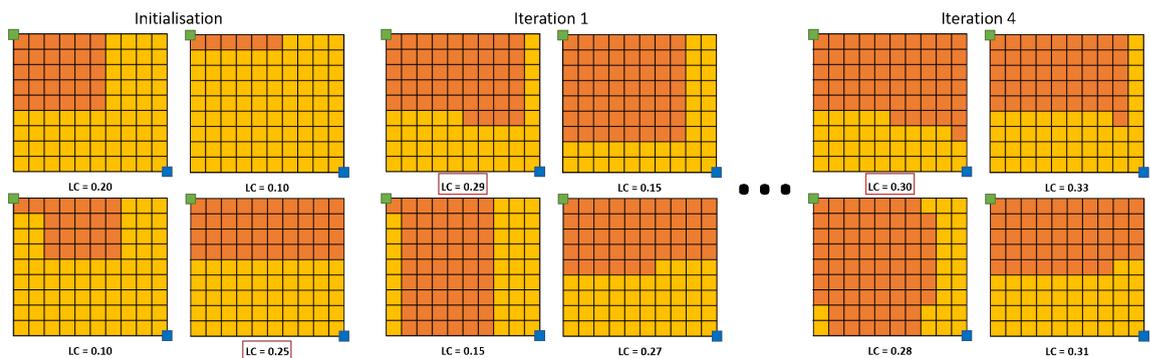


FIGURE 5.4: Model realisations for each particle after initialisation, iteration one and the final iteration.

response in the next iteration. This iterative optimisation process is continued for a set number of iterations, four in our case. From all iterations, an ensemble of diverse models with the best fit to the desired dynamic Lorenz coefficient can then be selected for full physics flow simulations used for future decision making.

The core of this chapter is dedicated to the integration and comparison of two PSO variants within this workflow. First, we will give a quick overview of the geological model building component, then introduce the dynamic screening component and finally introduce the two PSO variants used here. The last step is to perform several post-processing steps that will allow us to generate maps for future decisions based upon the information from all

models generated. Similar to the previous chapter, we will be using a combination of flow diagnostics and Shannon entropy to compare the outcomes from both workflows.

### 5.2.1 Geological model building

This step must efficiently integrate the known geological prior information into the reservoir modelling workflow. In this work we decided to use Petrel from Schlumberger for geological modelling. This part of the workflow is entirely interchangeable. It could for example also be done within an open-source geomodelling environment by using a combination of GemPy and PyKrige within python [Murphy Benjamin, 2021, Murphy Benjamin et al., 2021, Varga et al., 2019, 2021], SGeMS [Remy et al., 2009] or the Geology Designer from Rock Flow Dynamics. The key challenge here is to translate the prior knowledge into something that can be understood by the modelling tool. We therefore came up with a geological modelling workflow that allows for the parameterisation of geological input variables. An example for such a modelling workflow can be seen in the schematic algorithm 2.

---

**Algorithm 2:** Example for a parameterised geological modelling workflow

---

**+++ Sampling phase +++**

**for** each geological modelling variable  $i$  (e.g. training image or zone extend) **do**

    | Take random (or predefined) sample from prior distribution of variable  $i$

**end**

**+++ Modelling phase +++**

1. Define reservoir zones according to input variables
2. Populate each zone with facies properties according to selected geostatistical algorithm and specified geological modelling variables
3. Calculate petrophysical properties according to predefined rules and specified geological modelling variables      // e.g. selected training image

**+++ Exporting phase +++**

Export generated reservoir model and generate simulation file

---

In Petrel, a workflow similar to the one shown in the schematic algorithm is executed and builds the reservoir models according to the selected geological variables. The important thing here is that we need to generate and export new reservoir models without human interference. Therefore, the overall workflow first samples from the geological variables,

then generates a batch file containing the geological variables' values and opens Petrel. The Petrel workflow exports the reservoir models, ready for flow simulation in the final step. Throughout the overall workflow, this geological modelling step is automatically executed repeatedly.

### **5.2.2 Screening models for dynamic response - Flow diagnostics**

Within this workflow, thousands of reservoir models will be generated and tested on their dynamic flow response. Running full physics flow simulations on every model would be too time-consuming. We therefore make use of flow diagnostics (described in chapter 3 section 3.2) to get a quick indication about the dynamic flow behaviour of the generated reservoir models. This allows us to screen all reservoir models within a fraction of the time otherwise needed. The output metrics of particular interest within this workflow are the dynamic Lorenz coefficient and the backwards time-of-flight. Both give us an indication about how the reservoir got swept. It would also be possible to replace flow diagnostics with other tools that give a quick indication for flow response, such as streamline simulations or reduced physics surrogate models [Jin et al., 2020, Thiele et al., 2010, Wilson and Durlofsky, 2012].

### **5.2.3 Optimisation**

The optimisation component represents the cornerstone of this workflow. The main aim is to help generate geological reservoir models in line with the expected geology and the observed or desired dynamic flow response. Here we rely on the particle swarm optimiser (PSO) covered in chapter 3 section 3.1. As outlined before, we will use two PSO variants for the optimisation part and compare the outcomes. The objective behind this is to address one of the flaws with the PSO: the premature convergence to a local optima and getting stuck there. This is often associated with the exploration capacity of the PSO. Reservoir modelling or history matching present ill-posed inverse problems with a large degree of uncertainty. We can therefore be certain to have many reservoir models that, when under production, give a similar dynamic response to what is observed in the true field, but do not represent the actual reservoir. This will cause problems once such a model is used to make predictions, forecast, and plan new well placements. We therefore ideally use an optimiser

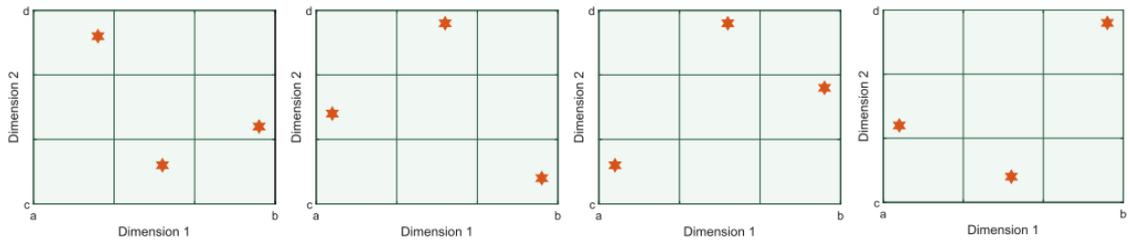


FIGURE 5.5: Exemplary arrangements of sample points in a two-dimensional parameter space with the help of LHS. The sample points are sparsely distributed to prevent over aggregation in different areas (figure after Li et al. [2020]).

that is very exploratory and finds a multitude of optimum solutions or, even better an optimiser especially robust against getting stuck in local minima in the first place. As we are using a proxy for flow simulations that only takes a fraction of the time conventional full physics flow simulations would take, we can afford to use a more exploratory search as it is feasible to run numerous iterations to allow for the algorithm to converge. And that is what we are trying to achieve with the different versions of the PSO developed here.

Usually meta-heuristic algorithms, such as the PSO are initiated randomly [Li et al., 2020]. The random initialisation can however have a significant impact on the optimiser’s performance. Therefore, in this work any meta-heuristic algorithm that we deploy will be initiated with the help of latin hypercube sampling (LHS) [McKay, 1992, McKay et al., 1979]. With this method the search space is first gridded into segments of equal intervals. We can now uniformly draw random samples from this stratified search space to generate our starting points. According to Deutsch and Deutsch [2012] this will prevent oversampling of specific areas (computationally inefficient) as we are often constrained by the number of realisations we can run, therefore missing out on sampling several regions of the feature space. With LHS, sample points are distributed sparsely to avoid oversampling issues and effectively cover the entire search space (e.g. figure 5.5). There is a tendency for the standard LHS to become less efficient for high dimensional search space problems. In this work we therefore use a LHS method that puts emphasise onto sampling multidimensional parameter spaces uniformly (LHSMDU) by Deutsch and Deutsch [2012]. This is accomplished by generating a large number of realisations using Monte-Carlo simulations and sequentially deleting realisations close to each other in the multidimensional space. Afterwards, the distributed realisations are post-processed to ensure uniformity of parameter distributions in the multidimensional space.

### 5.2.3.1 Standard PSO

The first PSO that we will be using is the standard PSO with a local setting and an option to reduce the inertia weight  $w$  gradually. As mentioned before, the local best PSO has a ring topology where particles are connected only to their  $k$ -nearest neighbours. This causes every particle to be attracted to a different local best position. On the other hand, the standard global PSO has a star topology, where every particle is connected and only one global best position exists that attracts all particles. The difference in topology gives a stronger exploratory component to the local best PSO [Engelbrecht, 2013]. We are trying to find a diverse set of potential reservoir models to capture the inherent geological uncertainty. Therefore the natural choice is to go with the local best PSO over the standard global best PSO.

The optimisation objective is to build geological reservoir models that fulfil a defined flow response from running flow diagnostics. To achieve this, the PSO can select from a range of geological modelling parameters, build the reservoir models, and compare the flow response to the optimum flow response. The flow diagnostic metric we want to match across different models is the flow capacity - storage capacity diagram [Shook and Mitchell, 2009] (e.g. figure 5.6). We take into account the dynamic Lorenz coefficient and the shape of the flow capacity - storage capacity curves (F-Phi curves). We take two things into account: (i) the dynamic Lorenz coefficient, (ii) the shape of the F-Phi curve. What we then try to minimise is (i) the absolute difference between the expected dynamic Lorenz coefficient and the measured dynamic Lorenz coefficient and (ii) the root mean square error (RMSE) between the expected (black) and measured (blue) shape of the F-Phi curve ( $d0$ ) as well as its first ( $d1$ ) and second ( $d2$ ) order derivative:

$$misfit = |LC_m - LC_e| + \sqrt{\frac{1}{n} \sum_{i=1}^n (d0_i - f0_i)^2} + \sqrt{\frac{1}{n} \sum_{i=1}^n (d1_i - f1_i)^2} + \sqrt{\frac{1}{n} \sum_{i=1}^n (d2_i - f2_i)^2} \quad (5.1)$$

Over iteration time, we expect to build geological reservoir models that have a dynamic response that resembles the expected shape of the F-Phi curve and the dynamic Lorenz coefficient. We decided to use the RMSE as the evaluation metric, because its focus is on

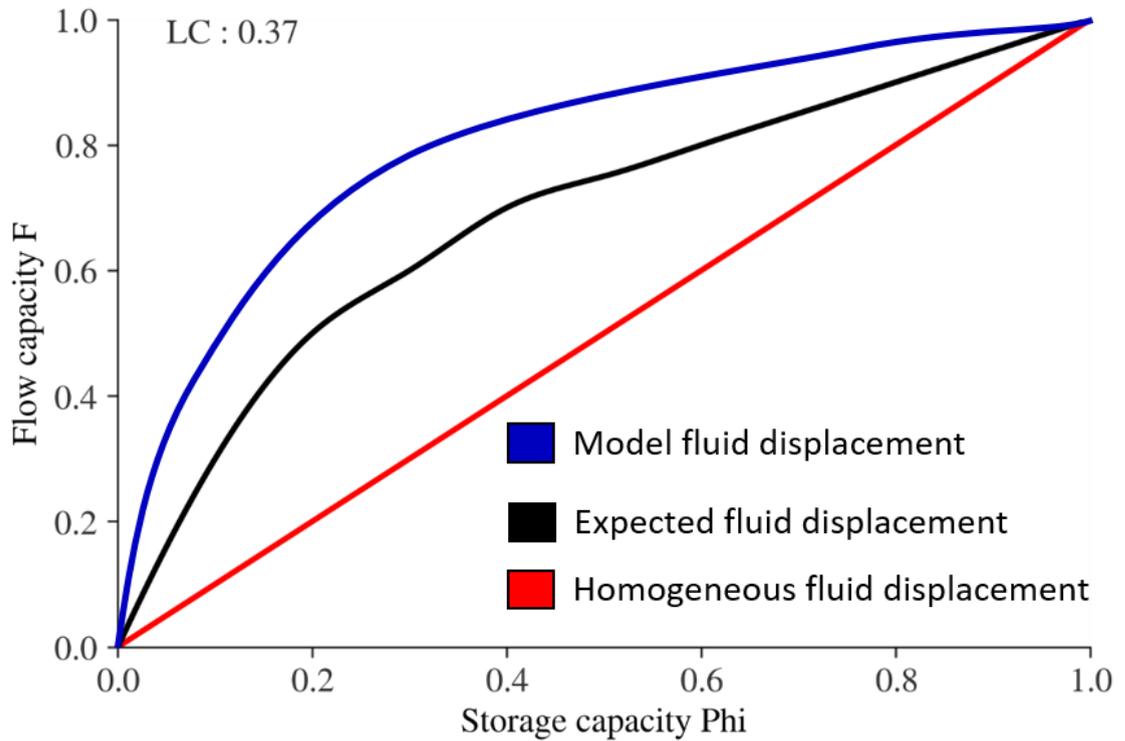


FIGURE 5.6: Flow capacity - storage capacity diagram showing the expected flow behaviour of the reservoir (black line) with an expected Lorenz coefficient of 0.37. The blue line represents one exemplary output model from the PSO that has the objective to match the expected flow behaviour curve. The red line indicates a case of homogeneous sweep and a Lorenz coefficient of 0.0.

penalising outliers heavier than small errors. This makes sense in this workflow, as we are not interested in obtaining a single precise solution, but many solutions that approximately fit our objective. Depending on the objective, other error measures such as for example the mean squared error (MSE) can easily be implemented.

As stated above, the intention of optimisation here is two-fold. We want to (i) have a diverse set of reservoir models that (ii) all fit our objective function for the flow response. One problem with the PSO could be that once a particle generates a reservoir model that has a good fit to the objective, other particles will be attracted to the position already discovered. We will only generate reservoir models similar to the ones already built. Therefore, there would be no novelty in this and no point in continuing the optimisation procedure that leads to generating more models alike. It is of lesser importance for us to get a single model that perfectly fits our objective, as we are only using a proxy model to get an indication about the flow response in the first place. We need models within an acceptable range of the objective function but with a diverse set of different models.

Furthermore, we are also not necessarily interested in the diversity of the input variables or the distribution of the static properties within the model. We are interested in observing diversity of the flow response and in how the reservoir models get swept. A proxy for swept and unswept regions in the reservoir models is the time-of-flight that we calculate with flow diagnostics. The backward time-of-flight tells us how long it would take an imaginary particle to get from a cell in the reservoir model to the producer well. A proxy basically for swept and unswept reservoir regions. High time-of-flight values mean that it would take ages for the particle to reach the producer and vice versa. We can now look at the time-of-flight in a single cell for all the reservoir models that give an acceptable fit to the objective function (figure 5.7). This tells us how many different ways that cell got swept, based on the underlying geology. We can now again make use of Shannon entropy in a similar way to Wellmann and Regenauer-Lieb [2012]. For the time-of-flight, we can calculate the entropy  $H$  for a single cell in a model with equation 5.2:

$$H(x) = - \sum_{i=1}^m p_i(x) \log p_i(x) \quad (5.2)$$

where  $x$  denotes the cell location and  $m$  the number of possible time-of-flight values that the cell can have, and  $p_i$  the probability of each time-of-flight value to occur within that cell. For simplicity the time-of-flight values are binned into  $m$  equal sections. Throughout this work  $m$  will equate to 40, unless specified otherwise. What determines the entropy is how many different time-of-flight values we have in that cell and the distribution of those values. The more differing time-of-flight values the higher the entropy for that cell. Suppose we now perform this calculation for every cell of the models that match our objective and assume independence of these cells. In that case, we can get an idea about the spatial entropy distribution of those models. Adding these values up will give us a joint entropy value for the entire system of models that match our objective as shown in equation 5.3:

$$H_{system} = - \sum_{i=1}^n H_i(x) \quad (5.3)$$

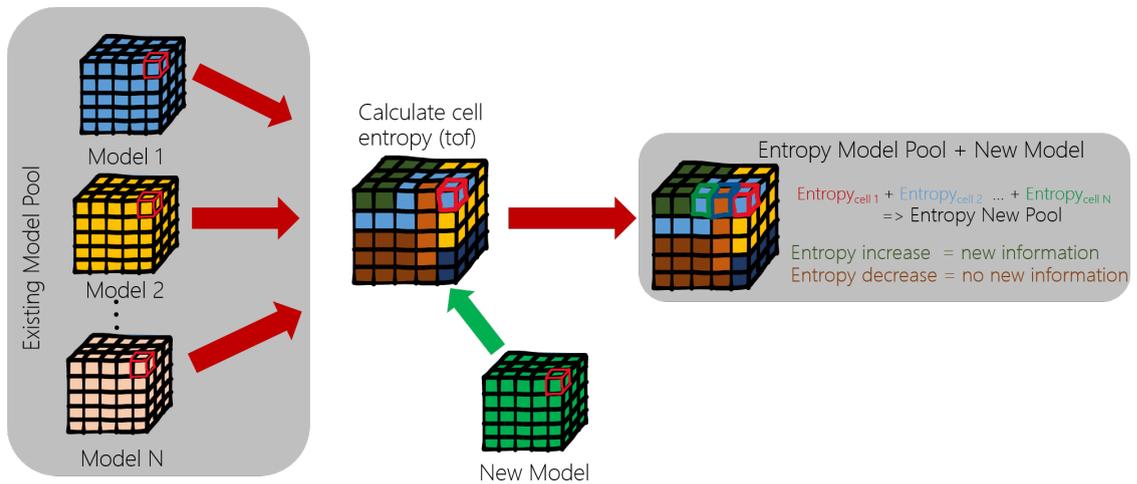


FIGURE 5.7: Example for the calculation of the entropy for system of best models. After  $n$  iterations there are  $N$  models that have a satisfactory fit to the objective function. After  $n + 1$  iterations another models satisfies these conditions. By calculating the time-of-flight based entropy of each cell for all these models we get can assess the diversity or information that is held by this pool of models. If the entropy increases with the new model included, then new information was added. If entropy decreases, no new information are added to the system.

This metric allows to control and influence how our models evolve over the PSO search. The higher the entropy of the system  $H_{system}$ , the more diverse the sweeping of our reservoir model ensemble is. Under the current settings, with  $m = 40$  and the reservoir model consisting of 140.000 cells, the maximum time of flight theoretically achievable is 744.800. This however does not take into account that for example in close proximity to the producer well, time-of-flight values will always be low. Additionally, only models that fit the outset objective (matching dynamic response and increasing the entropy of the system) get included into the calculation of  $H_{system}$ , further reducing the pool of potential models. Therefore the 744.800 should only be regarded as the absolute top  $H_{system}$  could theoretically achieve. It should also be noted, that the assumption made above about the independence of each cell is a very strong assumption. An alternative measure from information theory that could also be introduced here is that of mutual information. Given two random variables  $a$  and  $b$ , we can express how much information we get about  $b$ , given  $a$ , in other words, how much shared or mutual information do both variables have [McEliece, 2012]. Monitoring the value of  $H_{system}$  over the PSO iterations gives us an indication about the behaviour of the PSO. If  $H_{system}$  increases, we are generating novel models that add information to our set of already generated and flow simulated models (figure 5.8). If  $H_{system}$  decreases, we are generating models with sweeping patterns we

have already observed and, therefore, are a waste of simulation time. We can in-cooperate this into the guidance of the PSO and use it as a restart or stopping criterion:

$$PSO_{restart} = \begin{cases} False & \text{if } slope_{entropy} \geq 0 \\ True, & \text{else} \end{cases} \quad (5.4)$$

where we look at the slope that we would get by fitting a line through the entropy of the system for the  $n$  last iterations. By focusing on the slope of the entropy development over  $n$  iterations, we make sure not to restart or stop the PSO every time it does not find matching new models. This is only done if over a large number of iterations no novel models are found and the particles are probably homing closer on the already discovered models (figure 5.8). By stopping the PSO at this point, we prevent wasting further simulation time. Throughout this work  $n$  will equate to 5, unless specified otherwise. If we decide to restart the PSO, the particles will be initiated at a different position with the help of the latin hypercube optimised for uniform sampling in multidimensional parameter space (LHSM DU). This frees the particles from the previously reached local optima and allows them to find new optima to the objective function and find novel reservoir models that will increase the system's entropy.

### 5.2.3.2 Entropy-guided PSO

In this section we develop a modified PSO inspired by Guo et al. [2019]. It allows particles to use entropy to guide their decision-making and find novel reservoir models. One of the drawbacks of the PSO described in the previous section is that entropy is only used to control the entire PSO by either restarting or terminating it. Entropy, however, can be used to further inform the particle behaviour itself and make their search for more diverse models more effective. New, undiscovered models within an acceptable range of the objective function will initially contribute significantly to the system's entropy. When the particles are guided by historical information about their personal and local best only, more particles are attracted to these positions. Particles arriving at similar locations to the previous best location at later iteration times will generate similar models. This will eventually stop adding new information to the system and decrease the system's entropy

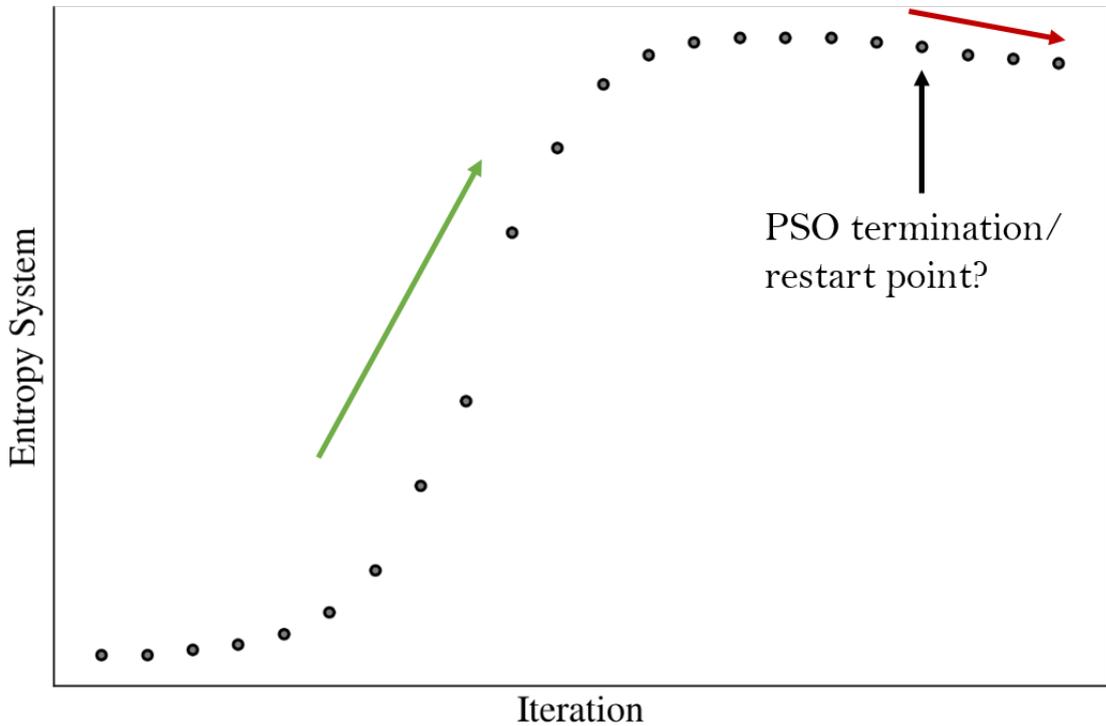


FIGURE 5.8: Example for how the entropy of the system develops over iteration time. The initial boost indicates that the PSO finds many new and novel reservoir models. In contrast, the decrease in entropy at the end shows that no new reservoir models are located and the PSO should either be terminated or restarted.

over time. This is where the particle's historical knowledge becomes a limitation. We will now elaborate on a new PSO that allows particles to use entropy to make decisions about their movement without getting attracted to already discovered positions.

The entropy-PSO differs from the standard PSO by (i) using a particle quality score, a combination of particle entropy contribution and particle misfit, to evaluate a particle and (ii) not taking into account historical particle information. The particle quality score can be calculated with equation 5.5

$$particle\_quality_i = W_1 * H_{contribution\_scaled_i} + W_2 * misfit\_scaled_i \quad (5.5)$$

where  $W_1$  and  $W_2$  are weighting factors that allow for the control of the exploratory and exploitative behaviour of the entropy-PSO respectively. The  $misfit\_scaled_i$  is equivalent to the scaled misfit used in the standard PSO as described in equation 5.1:

$$misfit\_scaled_i = \frac{misfit_i - misfit_{min}}{misfit_{max} - misfit_{min}} \quad (5.6)$$

where  $misfit_{min}$  and  $misfit_{max}$  represent the lowest and highest misfit of the current iteration respectively.  $H\_contribution_i$  is calculated with equation 5.7:

$$H_{contribution_i} = \begin{cases} H_{system+i} - H_{system} & \text{if } misfit_i \leq misfit_{threshold} \\ 0, & \text{else} \end{cases} \quad (5.7)$$

where  $misfit_{threshold}$  is a predefined value that determines an acceptable  $misfit_i$  and  $H_{system+i}$  is the entropy of the system of existing best models together with the new model generated by particle  $i$ . This is then used to calculate  $H_{contribution\_scaled_i}$  with:

$$H_{contribution\_scaled_i} = 1 - \frac{H_{contribution_i} - H_{contribution_{min}}}{H_{contribution_{max}} - H_{contribution_{min}}} \quad (5.8)$$

where  $H_{contribution_{min}}$  and  $H_{contribution_{max}}$  represent the lowest and highest entropy contribution of the current iteration respectively.

High particle quality scores are assigned to particles with a low  $misfit$  and a high entropy contribution value. Particles with a low quality score show high misfits and do not contribute to the entropy of the existing system. They can also have a  $misfit$  that is smaller than the  $misfit_{threshold}$  but do not carry additional information, leading to a negative entropy contribution.

Ranking all particles from the current iteration according to their particle quality score gives us the  $global_{best}$  position. Next, we define a neighbourhood for the particles to compare themselves to their  $n$  neighbours and evaluate their  $local_{best}$ . Throughout this work  $n$  will equate to 5, unless specified otherwise.

This information is now used to calculate the velocity  $v$  for each particle with equation 5.9:

$$V_i(t+1) = wv_i(t) + c_1R_1(P_{g_{best}}(t) - P_i(t)) + c_2R_2(P_{l_{best}}(t) - P_i(t)) \quad (5.9)$$

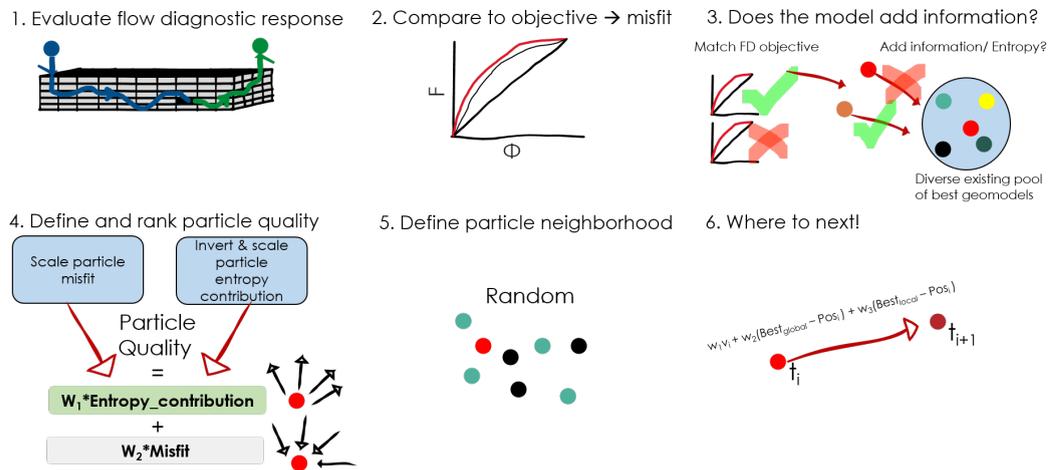


FIGURE 5.9: Schematic overview of steps that each particle in entropy-PSO undergoes:(1) calculate the flow diagnostic response (2) compare the response with the objective set out to match (3) If particle misfit is below predefined misfit threshold, calculate entropy contribution of the particle to the system of existing best models. If the particle is above a predefined misfit threshold, the entropy contribution is set to 0 (4) calculate the particle quality for every particle in the swarm at the current iteration (5) define a random neighbourhood for each particle (6) calculate the particle velocity only with information from the current iteration. No historical data are used.

It should be reiterated that in the entropy-PSO, we do not make use of the previous particle position history as entropy is included in the ranking of the particles. Here movement is controlled by the current local best particle and the current global best particle. In comparison: the standard PSO looks at the historic personal best and the historic local or global best to make decisions. For the entropy-PSO particles search the parameter space for novel and good matching parameter configurations. Once they visited these configurations and do not learn anything new, they are pushed out of this space to explore elsewhere. This is often not the case with the standard PSO, where particles get trapped in local optima due to their historical memory component. Additionally, the entropy-PSO considers the output space to dictate particle movement, not the diversity of the input space. Novel models are therefore only models where the underlying geology influences the reservoirs flow behaviour in a different way to previous models. The procedure that each particle undergoes every iteration is also summarised in figure 5.9.

## 5.2.4 Post-processing

With the help of the above-described workflows, we can produce a large amount of data and hundreds of reservoir models that fulfil the objective that we set out. However, to

make use of all available data sensibly, it is required to perform several post-processing steps. One big issue is that many models will have a very similar underlying geology and dynamic response. Taking these models forward within a history matching workflow is time-consuming and not feasible in practice. We therefore have to select a subset from these models that adequately represents the entity of best models. The method is briefly described in section 5.2.4.1 on dimension reduction and clustering. In section 5.2.4.2 we develop a technique that helps us use the entire set of best models by generating time-of-flight-based risk maps that could potentially be used for the planning of infill wells.

#### **5.2.4.1 Dimension reduction and clustering**

The dimension reduction and clustering techniques used here are similar to those applied within the training image generation workflow in chapter 4. To get a representative subset from all the generated models that fulfil the outlined objective, we first apply a dimension reduction technique to the best models' time-of-flight data. We use UMAP (Universal Manifold Approximation and Projection, see chapter 3 section 3.5 for further details) in this step for the same reasons outlined previously: it balances the preservation of local versus global data structures and runs in a short time. After the dimension reduction, the data are clustered with the help of HDBSCAN (Hierarchical Density-based Spatial Clustering of Applications with Noise, see chapter 3 section 3.6 for further details). HDBSCAN is insensitive to cluster shape and defining the number of clusters is not required. Pealat et al. [2021] and Allaoui et al. [2020] have shown that using UMAP in combination with HDBSCAN significantly improves the clustering outcomes. From each cluster generated within this step, we can select  $n$  reservoir models that can be taken forward to the next steps.

#### **5.2.4.2 Time-of-flight based risk maps**

As already outlined before, working with hundreds of reservoir models is not feasible. However, there is still useful information within the entire data set of best matching models. In figure 5.7 we demonstrate how to calculate the spatial time-of-flight-based entropy distribution. As shown in equation 5.2, the time-of-flight value in a single grid

cell of all the best matching reservoir models is used to calculate the entropy of that grid cell. This informs us in how many ways the grid cell got swept. The more diverse the time-of-flight values, the higher the entropy. Performing this calculation for every grid cell gives us an idea about the spatial time-of-flight-based entropy distribution. For cells with a high entropy we can, therefore, assure that we generated a wide range of models that caused a cell to be swept in numerous different ways. Cells with low entropy always get swept in a similar way - indifferent to the underlying geology. This means that the prior assumptions made for the underlying conceptual geological model together with the current well positioning and the expected flow behaviour do not significantly affect the sweep these cells experience. Either cells will always be swept, for example cells in close proximity to a well. Or it will always take a long time before cells will get swept and could therefore represent a good opportunity for potential infill wells. By calculating the mean or median time-of-flight for every cell of the set of best matching models we can get a good indication about the sweeping behaviour. Using a combination of both properties allows us to (i) identify areas that show a high potential for remaining hydrocarbons in place and (ii) verify that the models that we base these findings on adequately cover the geological uncertainty we assume for the area. We therefore came up with two properties that consider these information and the risk of a cell being swept. In the first case, for every cell we calculate:

$$RO_i = T_i * (1 - H_i) \quad (5.10)$$

where  $T_i$  represents the median time-of-flight value of a cell and  $H_i$  represents the entropy of a cell based on all models matching the objective function. Another, more subjective way to calculate the potential of a cell being swept is done with:

$$R1_i = \begin{cases} 3 & \text{if } T_i \geq T_{threshold} \text{ and } H_i \leq H_{threshold} \\ 2 & \text{if } T_i \geq T_{threshold} \text{ and } H_i \geq H_{threshold} \\ 1 & \text{if } T_i \leq T_{threshold} \text{ and } H_i \geq H_{threshold} \\ 0 & \text{if } T_i \leq T_{threshold} \text{ and } H_i \leq H_{threshold} \end{cases} \quad (5.11)$$

where  $T_{threshold}$  and  $H_{threshold}$  represent a median time-of-flight and entropy threshold to be selected by the user. The assigned values between 0 to 3 represent the chance of a cell being unswept in increasing order. With equation 5.10 and equation 5.11 we can now build spatial maps of the reservoir to identify regions with the highest chance of being unswept.

## 5.3 Results

### 5.3.1 Application to synthetic case study

Now we will apply both workflows developed in the previous section to the synthetic reservoir that we described in chapter 2 section 2.4. The idea for this case study is the following: Fractures are below seismic resolution and, therefore, add a high degree of geological uncertainty regarding their presence or absence. Additionally, they can act as highways for fluid flow that causes a reservoir to be swept inhomogeneously, potentially leaving behind large patches of oil within the reservoir. We would like to locate potential target positions for infill wells with the highest chance of hitting these unswept areas. To capture the reservoir's geological uncertainty, we generate a geologically diverse ensemble of reservoir models that all demonstrate a similar dynamic flow behaviour but get swept in different ways. Capturing the reservoir's dynamic behaviour will be part of the objective for the PSO. It is captured by the flow-capacity storage capacity diagram based on a 5-spot production pattern with a producer in the center of the folds crest and injector wells in all corners of the flank shown in figure 5.10. The reservoir models measure 2.000m x 1.000m x 35m and are modelled with 200 x 100 x 7 grid cells that measure 10m x 10m x 5m. That results in a total of 140.000 grid cells. Due to the restrictions of the academic tNavigator license used for flow simulations, higher resolution models are not possible. The geological uncertainty regarding fracture distributions within the crest and the flank respectively was captured by the subset of training images generated in chapter 4. Multiple-Point statistics are therefore used within the geomodelling part of the workflow to generate the reservoir models and distribute petrophysical properties. The parameter space that the particle swarm optimizer can work with consists of a range of training images that are used to model the fracture distributions in the crest and the flanks, variability for the extent of the crestal as well as the flank zone and differing permeabilities for the cells with

matrix dominated flow. It should be noted, that training images are arranged by increasing fraction of rock type 1 (fracture dominated flow). An overview for the parameter space is given in table 5.1:

**Parameter space for geomodelling with PSO**

<b>Parameter</b>	<b>Minimum</b>	<b>Maximum</b>
Training image crest - TI1	1	20
Crest-patch extend I-direction min	1	200
Crest-patch extend I-direction max	1	200
Crest-patch extend J-direction min	1	100
Crest-patch extend J-direction max	1	100
Training image flank - TI3	1	20
Flank-patch extend I-direction min	1	200
Flank-patch extend I-direction max	1	200
Flank-patch extend J-direction min	1	100
Flank-patch extend J-direction max	1	100
Crest extend min	50	150
Crest extend max	50	150
Rock-type II permeability (mD)	0.01	10

TABLE 5.1: Parameter space used for geomodelling within both PSO workflows

### 5.3.2 Standard PSO - local search

The standard PSO was run with 36 particles over 50 iterations, a local neighborhood setting of 5 particles, which according to Trelea [2003] gives a good balance between simulation speed and convergence. The social and cognitive weighting was set to a value of 1.49618, an inertia weight of 0.9, a damping factor of 0.99 and a reflective strategy for handling particles with boundary issues. Models that show a  $RMSE < 0.15$  to the set objective are considered an acceptable match and are included in the calculation of the time-of-flight-based entropy that will determine whether the PSO will restart.

#### 5.3.2.1 PSO simulation results

In figure 5.11 a summary on the misfit development and the dynamic response of the generated models is given. In the top left of figure 5.11 we can see how the  $RMSE$  gets gradually reduced for a subset of particles over iteration time. Particles highlighted in pink generated reservoir models that show an acceptable match ( $RMSE < 0.15$ ) to the

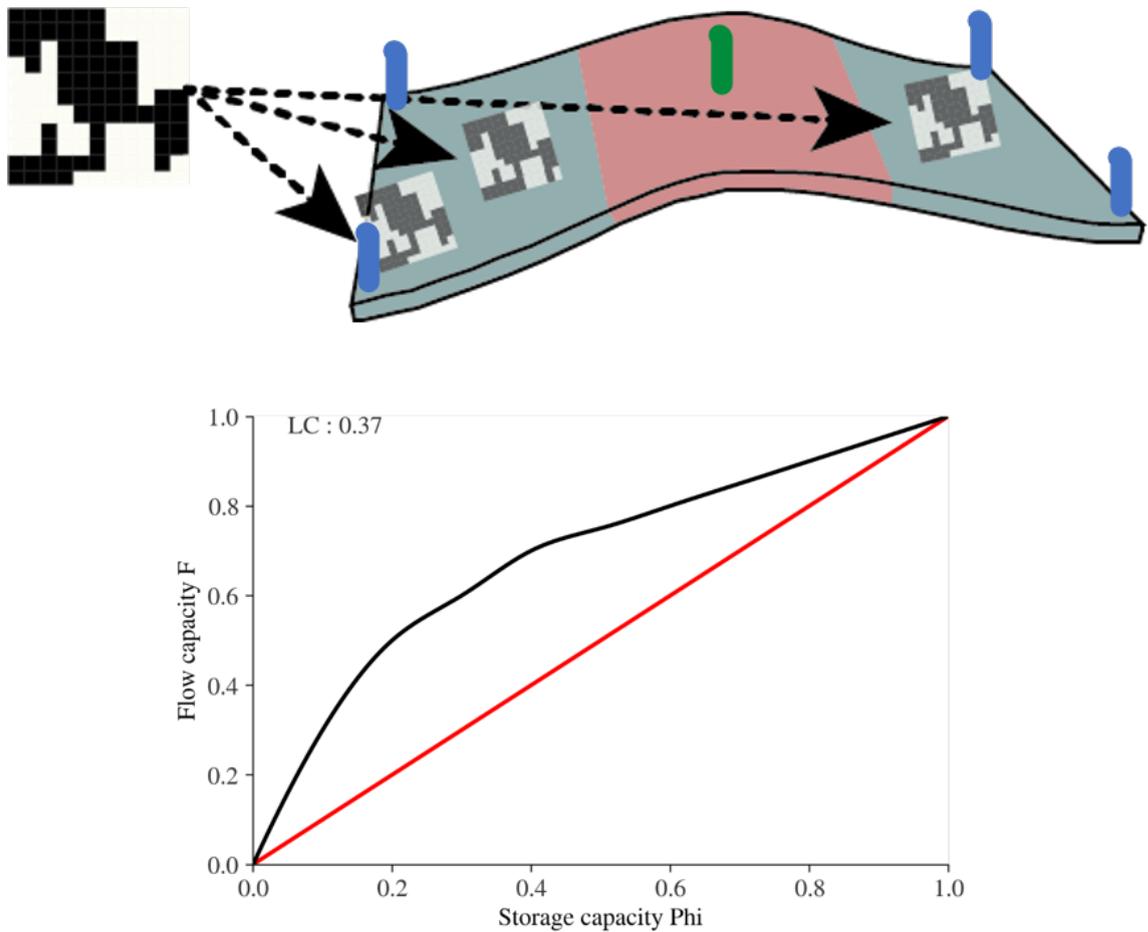


FIGURE 5.10: Top: Reservoir showing the crestal zone (red) and the flank zone (turtle green) with 5 spot production pattern. Injectors (blue) are position in the corners of the reservoir and the producer (green) in the center of the crestal zone. The reservoir is modelled with the help of an MPS, that can select from a range of training images for both the flank and the crest. Bottom: The objective of the PSO is to match the depicted F-Phi diagram and a dynamic Lorenz coefficient of 0.37.

objective function. At the same time there is still a high share of particles that do not show a great fit to the outset objective. This is indicative of a locally searching, PSO setup. After 24 iterations (indicated by the red arrow) the PSO automatically resets which causes a repeat in the previously observed behaviour: particles gradually reduce their misfit to the objective function whilst the overall swarm still maintains a high exploratory component. The reason for the PSO reset can be observed in figure 5.12 where the entropy development of the models that are within the acceptable range to the objective function is plotted against iteration time. Initially no matching models are found by the PSO and therefore the entropy stays flat at 0. This is followed by a rapid increase of entropy when the first matching models are found. Within 10 iterations an entropy of more than 350.000

is reached. However, the entropy plateaus around this level and decreases slightly when matching models are generated that have already been discarded by the PSO in previous iterations. This novelty decline causes the PSO to restart after 24 iterations. The restart of the PSO allows for the discovery of new matching reservoir models that cause a further gradual increase of entropy, reaching a plateau at around 450.000. This represents a further 28% increase in entropy compared to the previously reached plateau. After 50 iterations the PSO generated 1.800 reservoir models with 322 models within an acceptable range to the objective function. The exploratory component of the PSO can also be observed when looking at the flow diagnostic response shown in figure 5.11. The red line in the top right and bottom left plot of figure 5.11 represent the dynamic Lorenz coefficient and the corresponding F-Phi curve set out as an objective to be matched. Pink dots and lines highlight models that represent an acceptable match to the objective function. Grey dots and lines represent models that did not fit within this range. The broad flow responses highlight the wide range of different outlier reservoir models generated by the PSO. It however also becomes clear that for the dynamic Lorenz coefficient plot (top right figure 5.11) and the F-Phi graph (bottom left figure 5.11) no reservoir models were generated that either show dynamic Lorenz coefficients greater than what is deemed acceptable or showed F-Phi graphs that indicated a greater, more extreme curvature with higher F values at lower Phi values. The same holds true for the sweep efficiency plot (figure 5.11 bottom right) as the models that show an acceptable match to the objective also represent the lower bound for what sweep models experienced. Despite the exploratory setting of the locally searching PSO, it failed to generate models that thoroughly explored the boundaries for good matching models.

This lack of exploration is less evident when looking at the parameter space sampled by the PSO. In figure 5.13 each histogram shows how the PSO sampled the parameter space throughout the entire simulation. The PSO manages to sufficiently sample the predefined ranges of each parameter and shows sampling spikes for specific ranges of each parameter (e.g. training images for the crest - TI1 were mainly in the range of 5 - 10, with rock type 1 fractions in the range of 0.18–0.34). Figure 5.15 gives a similar impression. With the help of box-plots it shows how the particles explore each parameter at every iteration. Overall, trends are visible with the interquartile range (boxes) becoming narrower and converging

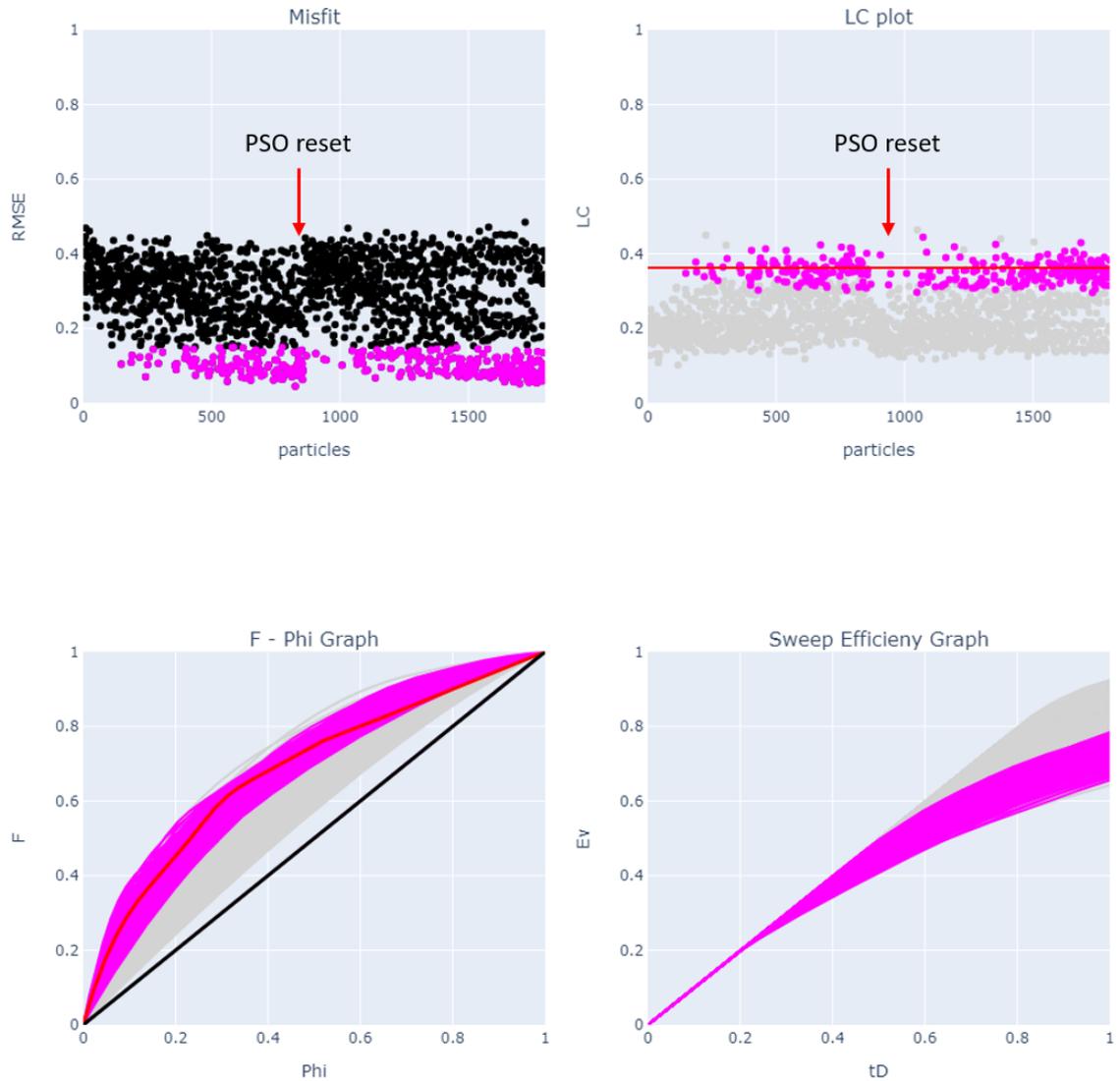


FIGURE 5.11: Simulation results from the standard PSO. Models with an acceptable match to the objective function ( $RMSE < 0.15$ ) are shown in pink. Grey dots and lines represent models that did not give an acceptable match. Top left:  $RMSE$  development of each particle over simulation time. The red arrow indicates the point in the simulation where the decrease in entropy forced the PSO to be restarted; Top right: Development of Lorenz coefficient (LC) for each particle over simulation time. The red horizontal line indicates a Lorenz coefficient of 0.37 which is the PSO objective; Bottom left: F-Phi diagrams for all generated models, whereas the red line indicates the F-Phi curve set as the PSO objective; Bottom right: Sweep efficiency diagram of all generated models.

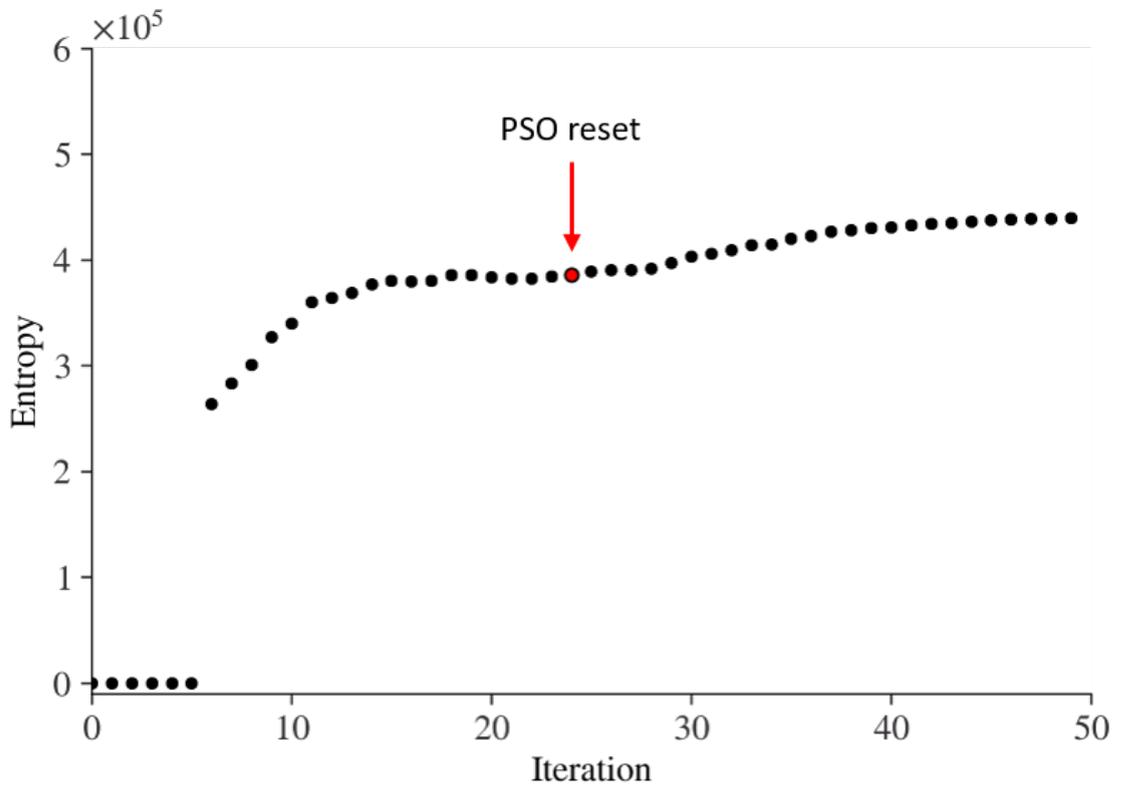


FIGURE 5.12: Development of time-of-flight-based entropy of all models that have an acceptable match to the objective function throughout simulation time. Entropy plateaus quickly but continues rising after the automatic reset.

towards a more limited range ( e.g. *Hinge\_min* converging towards values around 90). Yet many particles are still actively exploring the parameter space, which becomes evident by the long *whiskers* and outliers present for most parameters through all iterations (e.g. *Matrix\_perm* and *F1\_I\_MIN*). In figure 5.14 another histogram representation of the sampled parameter space is given. However, only the parameter values of models that provide an acceptable match to the objective function are shown this time. Here, much more concentration of the parameters is evident. For example, 150 of the 322 models that give an acceptable match to the objective function were generated with crestal training image 8 and matrix permeabilities in the range of 3 – 5mD. This indicates that the PSO, despite the evidence for a highly exploratory behaviour, was only able to find models that match the objective function based on a limited number of parameter ranges.



FIGURE 5.13: Histogram distribution for parameter space sampled by the PSO. All models generated are included in this plot.



FIGURE 5.14: Histogram distribution for the parameters sampled by the PSO. Only models that are within an acceptable range of the objective function are included.

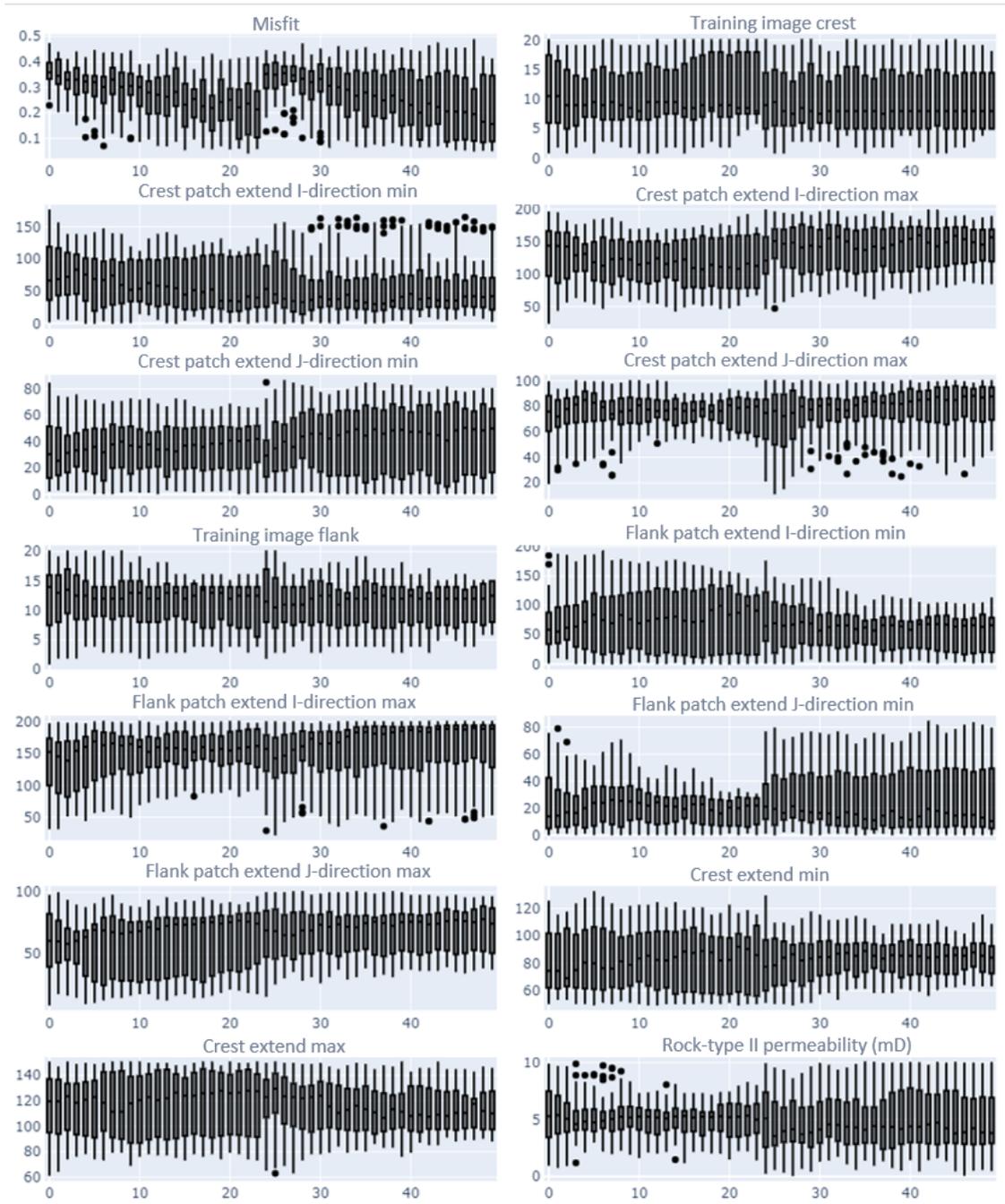


FIGURE 5.15: Box-plots showing the development of every parameter sampled by the PSO through simulation time. The exploratory nature of the locally searching PSO becomes evident here as a large part of the parameter space is covered for most of the simulation time.

### 5.3.2.2 Time-of-flight-based risk maps

Taking forward the 320 out of 1.800 reservoir models that give an acceptable match to the objective function allows us to generate the risk maps that can be used to support decision-making for potential infill well target zones. Figure 5.16 shows the folded reservoir model's flattened representation with the median time-of-flight (left) and the spatial time-of-flight-based entropy (right) of the best models. The median time-of-flight distribution indicates that the radial area around the producer in the centre of the model with the given reservoir models always show very low time-of-flight values. As is to be expected, the direct pathways between the producer well and the injector wells (corners) also show low median time-of-flight values. When moving away from the centre of the crest towards the flanks, time-of-flight values increase and show less uniform distributions. High and low time-of-flight values more often show up close to each other. The time-of-flight is the highest on both flanks, between the injector wells and therefore not within the direct pathways between the producer well and the injector wells. Similar patterns can be picked up for the time-of-flight-based spatial entropy distribution (figure 5.16 right). The radial area around the producer well in the centre of the crest has low entropy values, indicating that the time-of-flight is unaffected by the underlying geology and always shows similar values, i.e. the proximity of the producer is always swept in the same way. However, entropy values are high over the transition between the crestal and the flank zones and the direct pathways between the injector and producer wells. This high information content indicates that here the underlying geology has a significant impact on the way that the reservoir gets swept. The areas that already showed very high median time-of-flight values on the outside of the flanks, in-between the injector wells, also indicate a low entropy. Again, this suggests that with the prior assumptions made about the underlying geology and the expected dynamic response, these cells all experience a similar sweep, indifferent to the underlying geology.

In figure 5.17 the time-of-flight based risk maps are generated with the help of equation 5.10 (right) and equation 5.11 (left) and the spatial entropy and median time-of-flight data given in figure 5.16. Both represent flattened visualisations of the reservoir model and can be used to identify potentially unswept areas in the reservoir by combining sweep data (tof) with the given information content (entropy) for each cell of the model. In equation 5.11 we applied a tof threshold of 10 and an entropy threshold of 2.66 (maximum possible

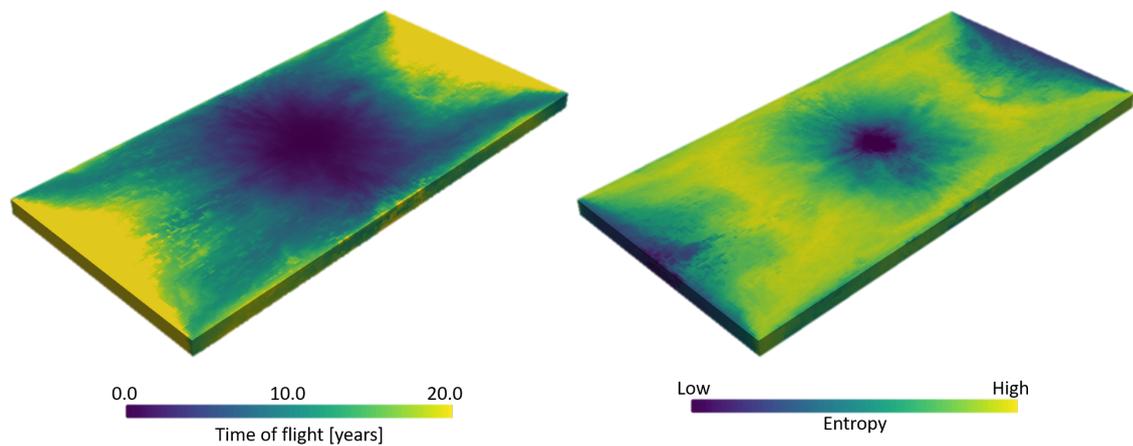


FIGURE 5.16: Flattened visualisation of the reservoir model. Left: Median time-of-flight values are shown for all reservoir models within the acceptable range of the objective function; Right: time-of-flight-based entropy distribution of all reservoir models within the acceptable range of the objective function.

entropy is 5.32) to generate the left plot. Dark blue zones indicate areas with entropy and time-of-flight values below the set threshold. This zone is present in the centre of the crest, where the producer well is placed. The chances for unswept areas in this zone are the lowest. Turquoise zones represent areas with a high entropy but the time-of-flight values lower than the set threshold. This zone is mainly present in the crestal zone, surrounding the dark blue zone and the direct pathways between the producer and the injector wells. Despite having high entropy values here, the time-of-flight values are lower than the set threshold and indicate that most areas are probably swept. The chances of finding unswept locations for potential infill wells are low. The orange zone shows entropy and time-of-flight values above the set threshold. Again, the high entropy here indicates that the underlying geology heavily impacts how this zone gets swept. However, this time time-of-flight values are high, indicating a higher chance of finding unswept areas. The dark red zone indicates entropy values below the set threshold and time-of-flight values above the set threshold. This combination gives us the highest chance of finding unswept reservoir areas and the most significant potential for infill wells. The underlying geology and expected flow response are unlikely to allow these areas to be swept with models generated with the standard PSO workflow and the current well setting. Based on the results shown here, which enables us to condense the information given from all 320 best models, the highest chance of finding unswept areas is in the dark red zone. The right risk map in figure 5.17 created with the help of equation 5.10 shows similar results. However,

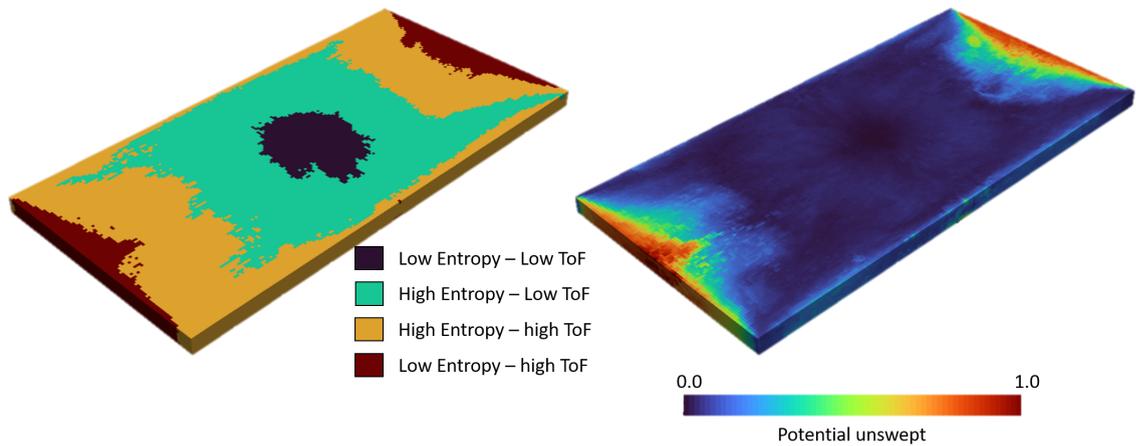


FIGURE 5.17: Flattened visualisation of the reservoir model. Time-of-flight based risk maps generated by combining the spatial entropy and median distribution of time-of-flight values of the models within the acceptable range of the objective function shown in figure 5.16; Left: Segmentation of the reservoir model into sections of low medium and high chance of finding unswept areas according to equation 5.11; Right: Segmentation of reservoir model into potentially unswept zones according to equation 5.10.

the majority of the reservoir model is considered to have a very low potential for finding unswept areas. The areas on the flank between the injector wells are again the ones with the most considerable potential. The results are similar to those presented before and as indicated in figure 5.16. Unlike with the previous subjective and categorical method, here we visualise the same input data objectively and their continuous nature allows for more refined observations. Both maps indicate that the highest chance for finding unswept reservoir regions are on the flanks in-between the positions of the infill wells.

### 5.3.2.3 Dimension reduction and clustering

320 best models that are within the acceptable range of the objective function are still too many models to take through all steps of conventional simulation and optimisation workflows. To work with a more manageable ensemble of models, dimension reduction and clustering with the help of UMAP and HDBSCAN is applied to the time-of-flight data of the 320 best models that are within the acceptable range of the objective function. In this case, each model contains 140.000 cells which equates to the dimensions of the parameter space. This step aims to preserve the information obtained from the 320 best models to an acceptable degree. The settings for UMAP were kept at default values apart from  $n\_neighbours = 30$ ,  $min\_dist = 0$  and  $n\_components = 30$ . The settings for HDBSCAN

were also kept at default values apart from  $min\_samples = 1$ ,  $allow\_single\_cluster = True$  and  $min\_cluster\_size = 10$ . The 320 models are separated into 10 separate clusters, as shown in figure 5.18. One should acknowledge that the aim of our dimension reduction is to pre-process our data for clustering - not for visualisation purposes. An advantage of UMAP is that we can select to how many dimensions we would like to reduce our data set to. For visualisation purposes however, in figure 5.18 we only look at 2 out of 30 dimensions (dimensions 0 and 1) that we projected our data into. Therefore, it appears as if there is a substantial overlap between some clusters. Figure 5.19 showcases the same data as in figure 5.18, but this time with different 2D representations (top: dimensions 8 and 9; bottom: dimensions 5 and 6). Focusing for example on cluster 6 (dark blue), we can see that in the bottom projection, cluster 6 is close to overlapping with clusters 2 (light-green) and 3 (green). However, the top projection of cluster 6 is distinctively separated from clusters 2 and 3. Therefore, one should not focus too much on the two-dimensional representation of the data. The time-of-flight data from representative models from a selection of clusters is also displayed in figure 5.18. The visualisations show a flattened representation of the folded reservoir model. It confirms that models from different clusters get swept differently. For example, models from cluster 0 (eggshell white, bottom right corner) show a rather uniform distribution of low time-of-flight values. Models from cluster 4 (green, top left corner) indicate a less uniform time-of-flight distribution, with low values mainly present in the crestal zone and the pathway from the producer well to the injector well in the bottom right corner. All clusters in the bottom left corner indicate low time-of-flight values in the crest and along the borders of the models going from the crest towards the injector wells. The clusters in the bottom right corner differ in the extent of the radial pattern of low time-of-flight values around the producer well in the centre of the crest. The low time-of-flight values range from being confined to the crest to almost extending over the entirety of both flanks.

A total of 20 models are now sampled from the 10 clusters. Each cluster is sampled in proportion to its size while allowing for each cluster to be sampled at least once. The entropy of the 20 models aggregates to 427.430 and is comparable to the entropy achieved with the 320 models. The distributions for the median time-of-flight (figure 5.20 left) and

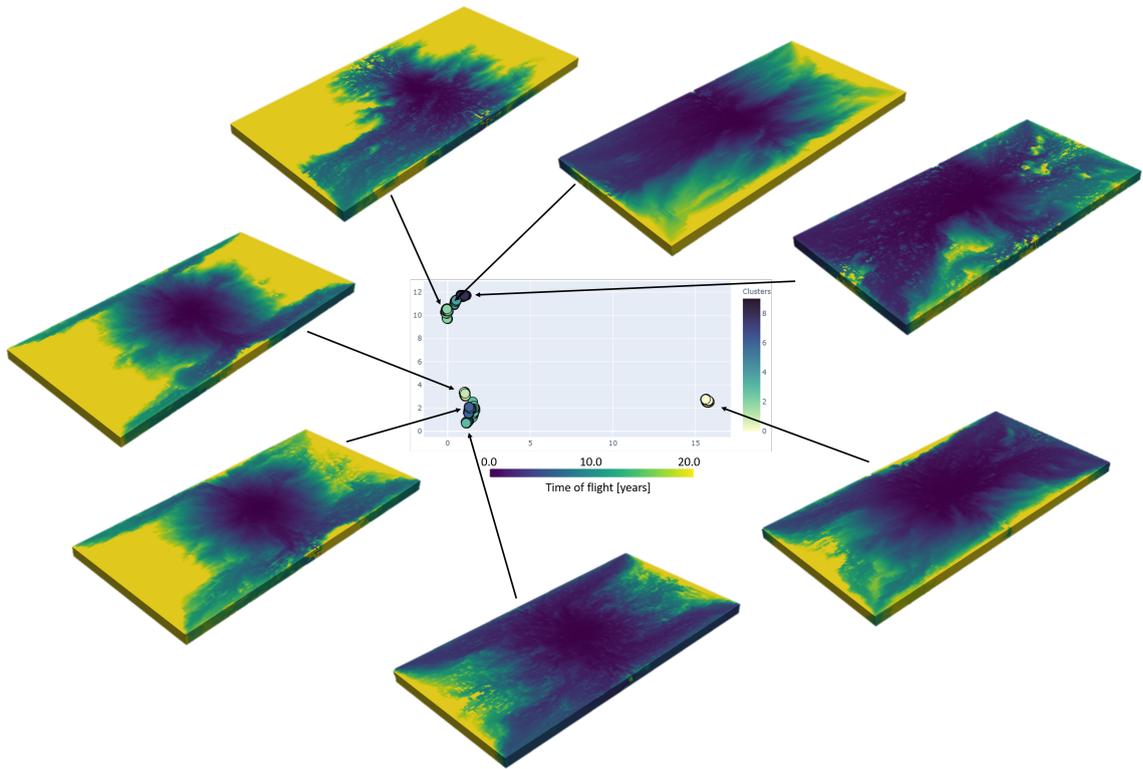


FIGURE 5.18: Time-of-flight-based clustering of models within the acceptable range of the objective function of the standard PSO. Each point represents a reservoir model's lower-dimensional representation (30 dimensions). Colours match the associated cluster. By presenting the models in two dimensions in this plot, one gets the impression of substantial overlap of different clusters. Exemplary flattened reservoir models are sampled from a subset of the clusters showing the spatial time-of-flight distribution.

the spatial time-of-flight-based entropy (figure 5.20 right) as well as the time-of-flight-based risk maps (figure 5.21) generated in section 5.3.2.2 are now reproduced with the subset of 20 models. From a first visual inspection, the results look very similar to those generated with the 320 models. The median time-of-flight (figure 5.20 left) shows that the areas with time-of-flight values greater or equal to 20 years on both flanks are more significant than the results from the 320 models indicated. A similar observation can be made for the spatial time-of-flight-based entropy distribution (figure 5.20 right). The entropy for individual cells overall reaches lower highs. The area on both flanks between the injector wells with very low entropy values is more extensive than with 320 models. These observations also pass through to the generated time-of-flight-based risk maps. The same threshold values for equation 5.11 were applied to create the left risk map in figure 5.21 (tof threshold = 10 and entropy threshold = 2.66). The map generated shows similar trends to the one generated with 320 models. However, the zone of high entropy and low

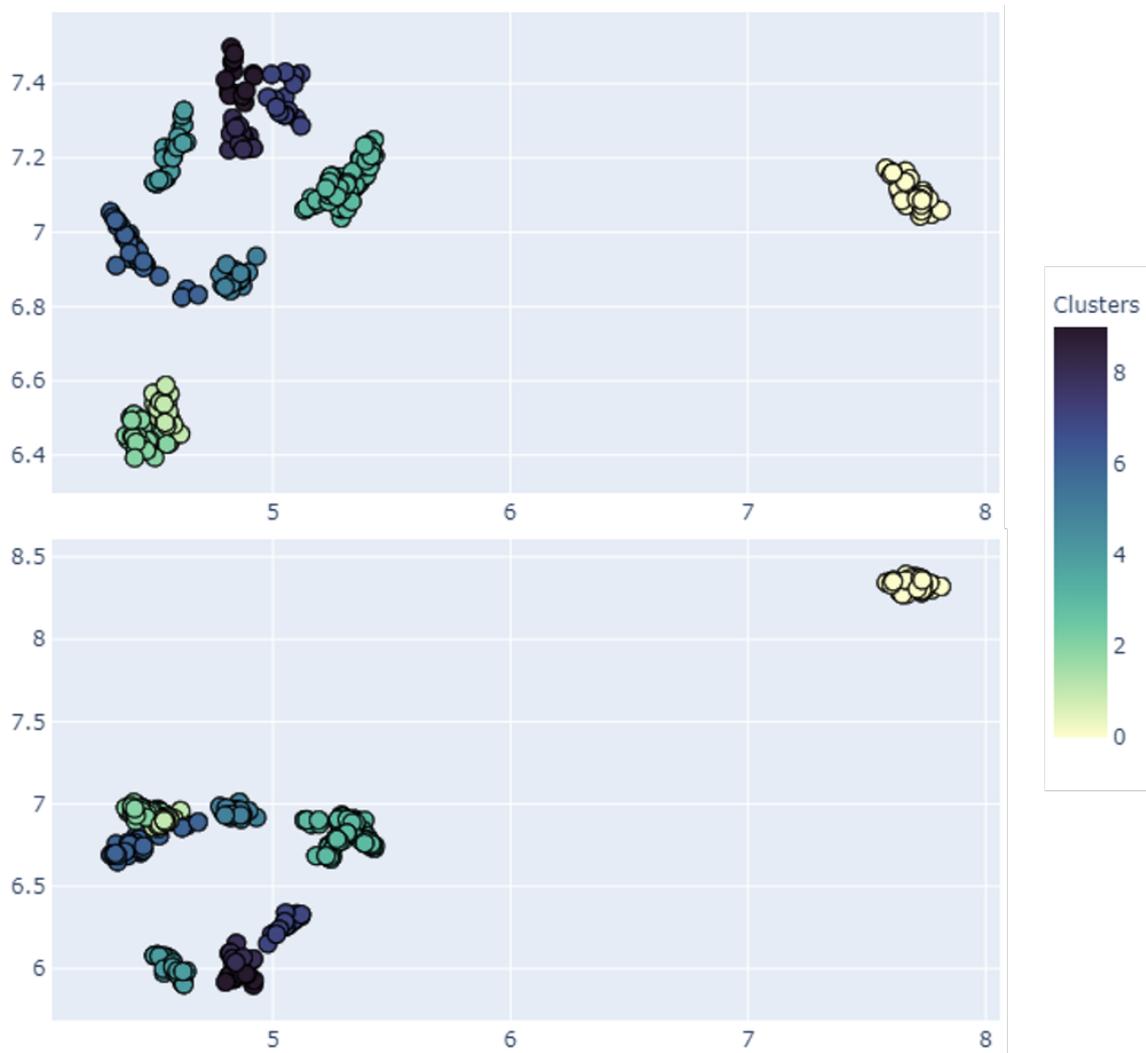


FIGURE 5.19: Two alternative representations of the time-of-flight-based clustering performed in figure 5.18. Dimensions selected for visualisation here (top: dimensions 8 and 9; bottom: dimensions 5 and 6) differ from the ones chosen in figure 5.18 (dimensions 0 and 1). Depending on the dimensions selected for visualisation, different clusters give the false impression of overlapping, whereas being apart from each other in the original 30-dimensional space chosen for clustering.

time-of-flight is limited to a turquoise halo around the producer well that does not extend beyond the crestal area. The orange zone of high entropy and high time-of-flight takes up more space within the model. These are the zones where we would expect to have a higher chance of finding potentially unswept areas. The dark red zone with low entropy and high time-of-flight values is also larger than with 320 models. The location is the same. However, the extend towards the crest expanded. We would deem these areas to have a very low chance of being swept with the current well positioning, the underlying geology and the expected flow response. It is worth mentioning, that for the risk maps

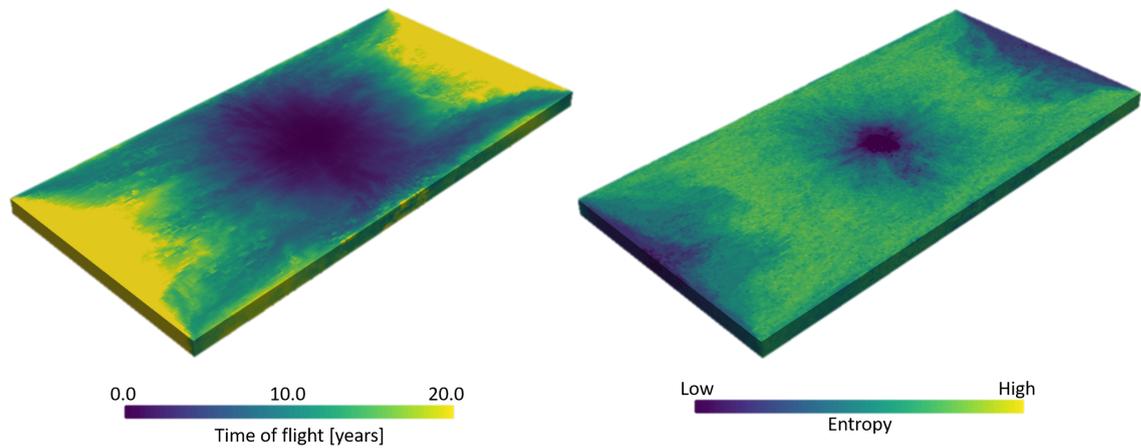


FIGURE 5.20: Flattened visualisation of the reservoir model. Left: Median time-of-flight values shown for 20 representative models sampled from each cluster shown in figure 5.18; Right: time-of-flight based entropy distribution shown for 20 representative models sampled from each cluster shown in figure 5.18.

generated with the sub-set of 20 models, the proportion of cells with low entropy and high time-of-flight values almost doubled to 15% in comparison to the risk maps generated with all 320 best models. This effect does not come up as strong in the risk map generated with equation 5.10. The zones that indicate a high chance of finding potentially unswept areas, especially on the flanks between the injector wells, are greater than those with the 320 models. Additionally, the overall potential of individual cells to be unswept also appears higher than with the 320 models. Conclusively, the evaluation of the 20 models tells a similar story as the 320 models together. The overall entropy in both cases is similar and shows that most of the information was captured during the clustering and downsampling step. However, a distinctive difference is that the results from the 20 models indicate a lot more overall potential for unswept areas and are therefore more *optimistic*. Working with an ensemble of only 20 models makes future decision making more efficient and manageable, but comes with a loss of information.

#### 5.3.2.4 Flow simulations and comparison to reference case

On the subset of 20 reservoir models, we will now run flow simulations to check if they capture the behaviour of the reference case set up in chapter 4 section 4.4. The simulations will run under black oil settings, and the same five-spot production pattern as before (one injector well in each corner and a producer well in the centre of the crest) is simulated for ten years. The results of all flow simulations are summarised in figure 5.22. In each

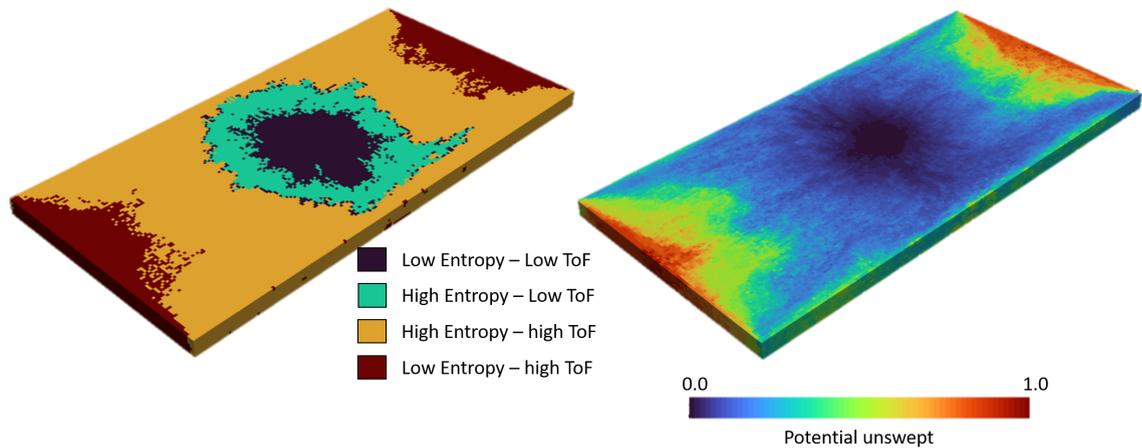


FIGURE 5.21: Flattened visualisation of the reservoir model. Time-of-flight based risk maps generated by combining the spatial entropy and median distribution of time-of-flight values for 20 representative models sampled from each cluster shown in figure 5.18; Left: Segmentation of the reservoir model into sections of low medium and high chance of finding unswept areas according to equation 5.11; Right: Segmentation of reservoir model into potentially unswept zones according to equation 5.10.

plot of figure 5.22, the green, blue, red and yellow colour corresponds to three distinct production profiles and the reference case. The total oil production shown in figure 5.22 a over the ten years ranges from  $0.4 - 1.3 \times 10^6 \text{m}^3$ , equating to 15 – 32% oil recovery factors. Despite only working with 20 models, it is interesting that we can observe three different production patterns. For the first pattern, the case depicted in green is a good representation (figure 5.22). Models that show this pattern correspond to models with the highest total oil production at production rates of up to  $650 \text{m}^3$  of oil per day which quickly declines, and oil production is gradually replaced by an early water breakthrough and high rates of water production. Models that show this sweeping pattern are associated with cluster 0 in figure 5.18 (eggshell white). They indicate a high proportion of rock type 1 (fracture dominated flow). The very early water breakthrough in these reservoir models is likely associated with localised flow through the highly connected system of cells with rock type 1, where fractures dominate flow in particular along the flanks. A good representation for the second production pattern observed is the case shown in blue and to some degree, the reference case (yellow). Models that depict this pattern correspond to intermediate total oil production models and demonstrate how the injector wells support a more homogeneous and constant reservoir sweep without late-stage water breakthrough. Models that show this sweeping pattern are associated with the clusters seen in the bottom left of figure 5.18. Models with no or late-stage water breakthrough have a high proportion

of rock type 0 (matrix dominated flow). The higher the ratio of rock type 1, the more likely an early water breakthrough occurs. The last observed pattern is a constant, low oil production rate coupled with a very early water breakthrough (figure 5.22). The third case (red) is a good representation of this pattern and a characteristic of these models is that they have a high proportion of rock type 1 with very bad overall reservoir connectivity. This leads to preferential production from localised parts of the reservoir. Models that show this sweeping pattern are associated with the clusters seen in the top left of figure 5.18. Figure 5.22 e depicts the recovery factor of every reservoir model vs the overall pore volume of water injected. Interestingly, the models have a very narrow range of recovery factors and pore volumes injected within proximity to the reference case. The recovery factor varies between 15 – 35%, with 0.16 – 0.41 pore volumes of water injected.

Overall, the subset of 20 reservoir models produced in the standard PSO workflow capture the dynamic response seen in the reference case. The workflow also generated models swept in different ways whilst honouring the underlying assumptions about the geology and fracture distributions. It however seems like under the current input guidance and the workings of the standard PSO, only three major trends with minor abbreviations for sweeping the reservoir model were found. Nevertheless, these findings from these models and the time-of-flight-based risk maps can now be used to make informed decisions under uncertainty about potential locations for infill wells.

### **5.3.3 Entropy-PSO**

The entropy-PSO was run with 36 particles over 50 iterations, a global neighbourhood setting connecting all particles, a social and cognitive weighting value of 1.49618, an inertia weight of 0.9, a damping factor of 0.99 and a reflective strategy for handling particles with boundary issues. Models that show a  $RMSE < 0.15$  to the set objective are considered an acceptable match and included in calculating the time-of-flight-based entropy.

#### **5.3.3.1 PSO simulation results**

In figure 5.23 a summary of the misfit development and the dynamic response of the generated models is given. In the top left of figure, 5.23 we can see how the misfit of

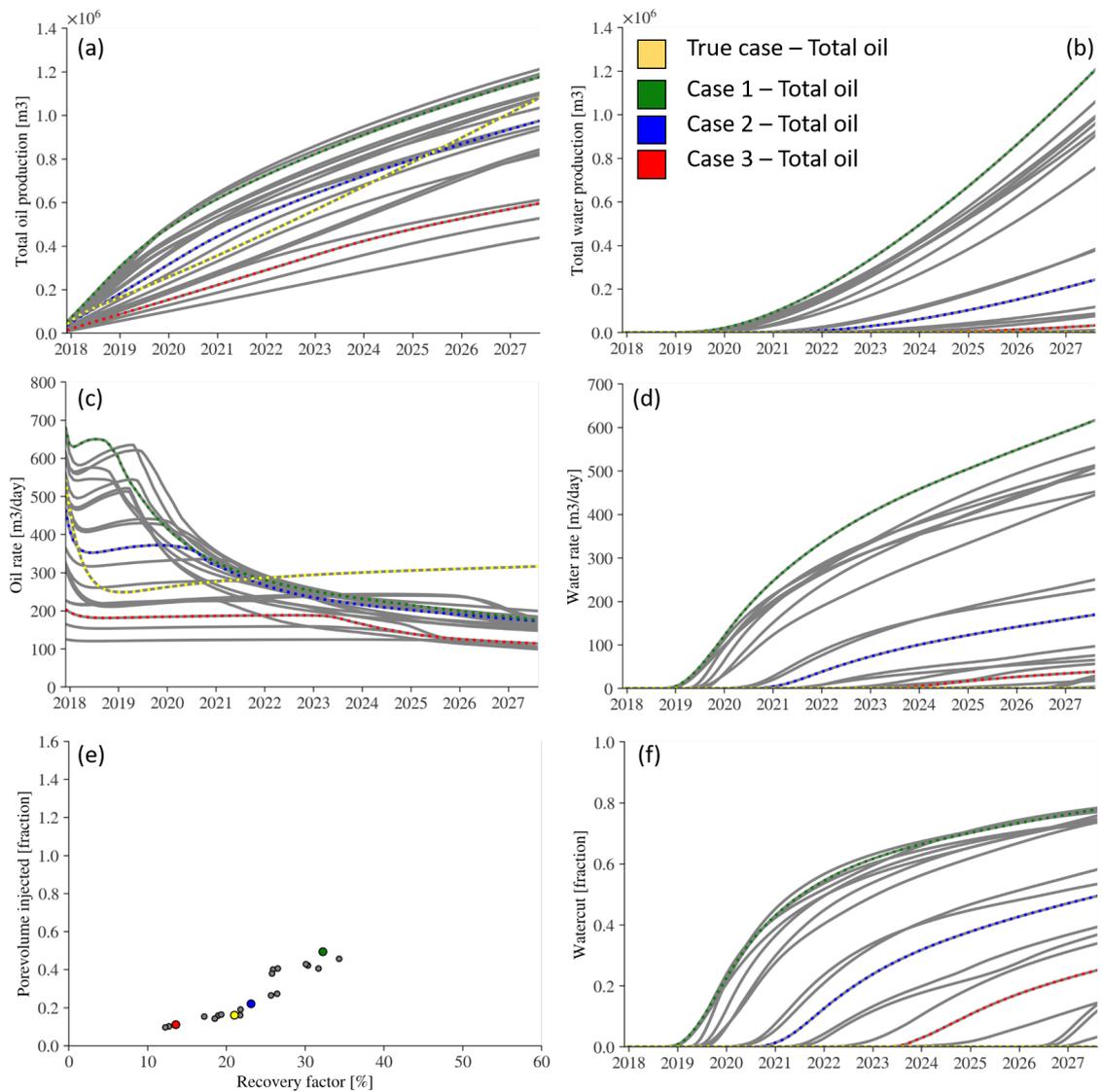


FIGURE 5.22: Results of flow simulation for 20 representative models sampled from each cluster shown in figure 5.18 displaying total oil production (a), total water production (b), oil production rate (c), water production rate (d), pore volume of water injected vs oil recovery factor (e) and water cut of the producer well (f).

particles to the objective function reduces gradually over iteration time. It also becomes apparent that initially it takes the particles several iterations before particles start moving towards an acceptable fit to the objective function. Once this initial hindrance is overcome, most particles generate models that give an acceptable match to the objective function or are very close to passing the *RMSE* threshold of 0.15. Throughout the simulation, however, we are also able to maintain a share of particles that produce reservoir models that do not improve their fit to the objective function. This is indicative of the exploratory setting of the entropy-PSO. Over 50 iterations the entropy-PSO generated 1.800 models, with 392

models that give an acceptable match to the objective function. In figure 5.24 we can see how the entropy of the pool of best models develops over iteration time (black dots). For comparison, the entropy development of the best models generated with the standard PSO is shown by the red dots. Interestingly, it takes the entropy-PSO 12 iterations to generate models that give a good match to the underlying objective. This is almost twice as long as it took the standard PSO to generate the first matching reservoir models. For the following 13 iterations, the system's entropy increases linearly up to an entropy of around 520.000 and surpasses the maximum entropy achieved by the standard PSO. Until iteration 50, the entropy of the system generated with the entropy-PSO continuously increases until it reaches an entropy of 575.000. Note how there is no single iteration where the entropy decreases compared to the previous iteration. The exploratory component of the entropy-PSO can also be observed when looking at the flow diagnostic responses of the generated models shown in figure 5.23. The red line in the top right and bottom left plot of figure 5.23 represent the dynamic Lorenz coefficient and the corresponding F-Phi curve set out as an objective to be matched. Pink dots and lines highlight models that represent an acceptable match to the objective function. Grey dots and lines represent models that did not fit within this range. Again, the broad range of different flow responses shows the diversity in the generated reservoir models and speaks for the exploratory component of the entropy-PSO. Unlike the standard-PSO, here we also notice that the entropy-PSO generated reservoir models that show dynamic Lorenz coefficients (figure 5.23 top right) higher than the set objective for a dynamic Lorenz coefficient of 0.37, reaching extremes of up to 0.6. The same behaviour can be seen in the F-phi graph (figure 5.23 bottom left) and the sweep efficiency graph (figure 5.23 bottom right). Both graphs indicate that the entropy-PSO generated reservoir models that show more extreme dynamic flow behaviours than the models generated by the standard PSO. This allowed the entropy-PSO to create models that more sufficiently explored the boundaries for good matching models.

When looking at how the entropy-PSO sampled the parameter space, this exploratory behaviour initially seems less evident. In figure 5.25 each histogram shows how the entropy-PSO sampled the parameter space throughout the simulation. In contrast to the standard PSO, many parameters (e.g. *TI\_1*, *F1\_I\_MAX* and *Matrix\_perm*) show histograms following normal or skewed normal distributions. However, we can also

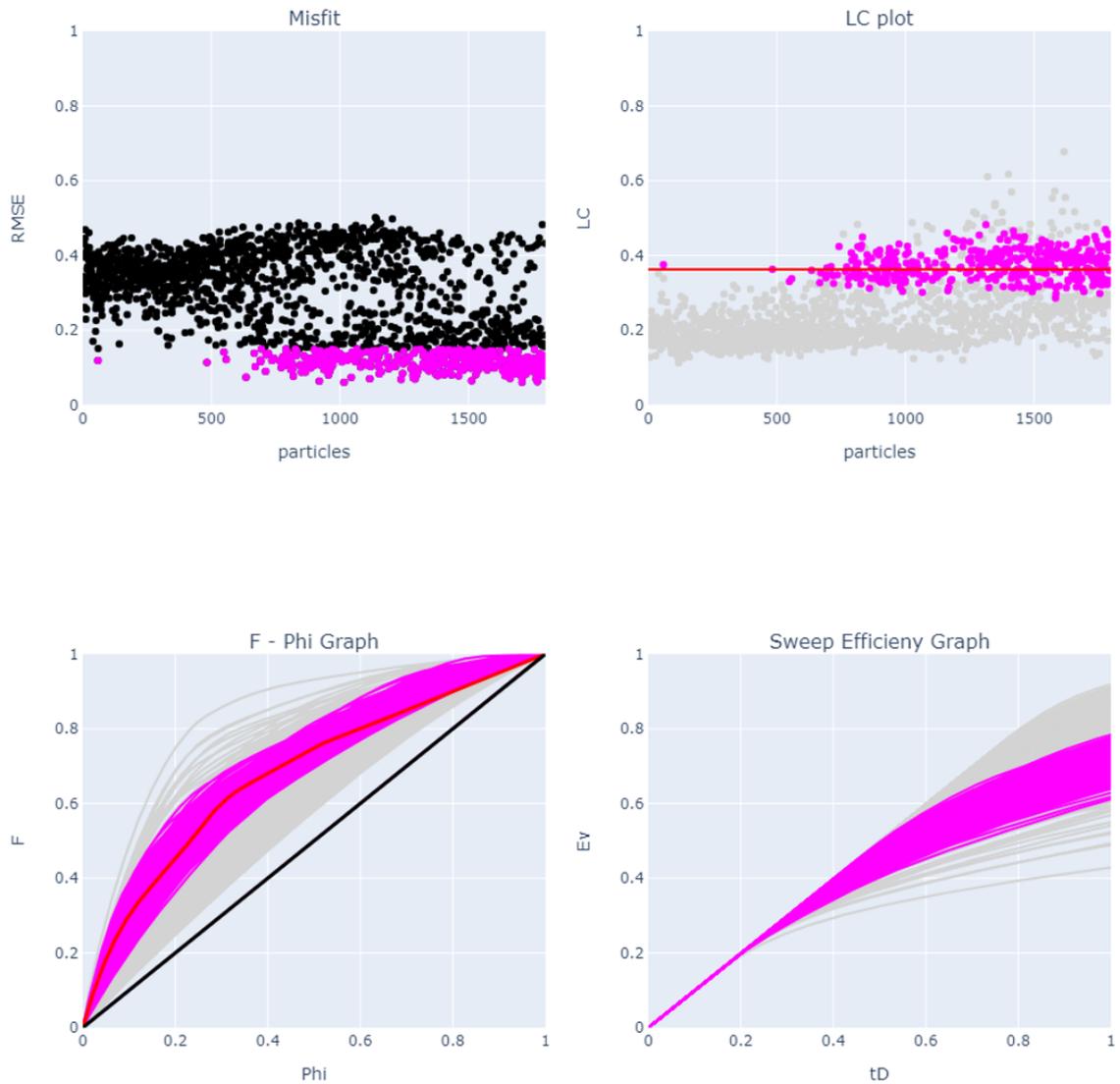


FIGURE 5.23: Simulation results from the entropy-PSO. Models with an acceptable match to the objective function ( $RMSE < 0.15$ ) are shown in pink. Grey dots and lines represent models that did not give an acceptable match. Top left:  $RMSE$  development of each particle over simulation time. Top right: Development of the dynamic Lorenz coefficient (LC) for each particle over simulation time. The red horizontal line indicates a dynamic Lorenz coefficient of 0.37 which is the PSO objective; Bottom left: F-Phi diagrams for all generated models, whereas the red line indicates the F-Phi curve set as the PSO objective; Bottom right: Sweep efficiency diagram of all generated models.

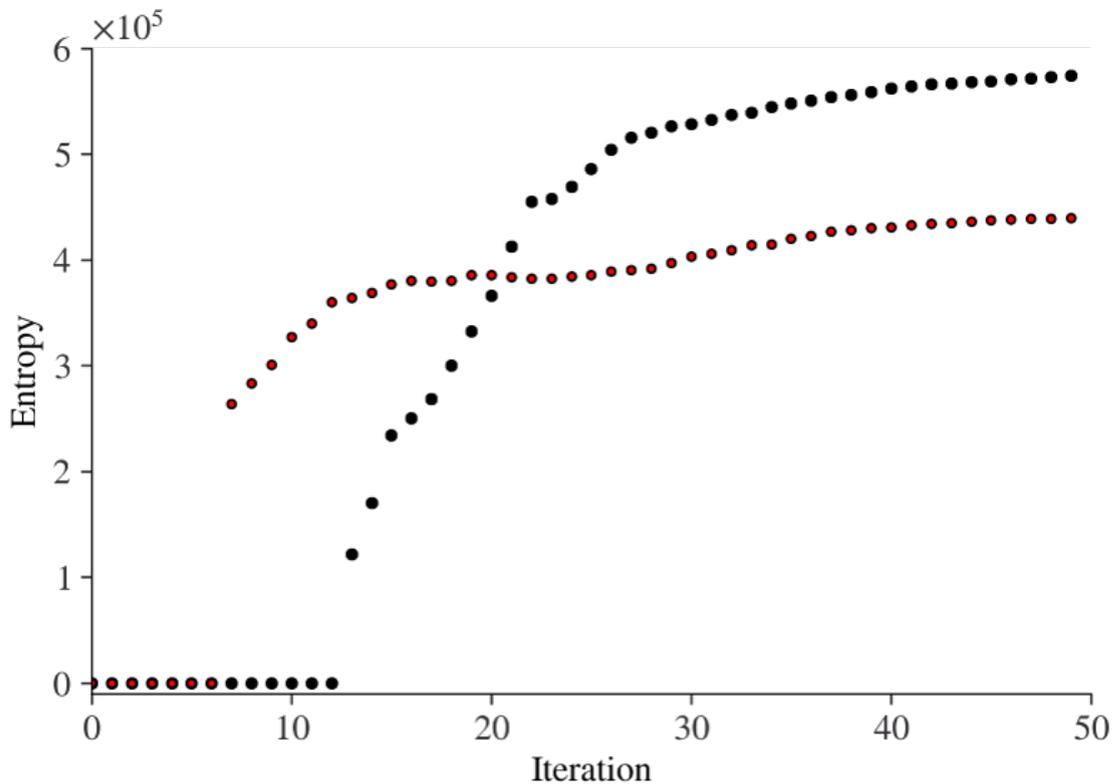


FIGURE 5.24: Development of time-of-flight-based entropy of all models that have an acceptable match to the objective function throughout simulation time. Red dots represent the entropy development for the standard PSO and black dots represent the entropy development of the entropy-PSO.

observe sampling spikes for specific ranges of some parameters (e.g.  $F1\_J\_MAX$ ). In figure 5.26 another histogram representation of the sampled parameter space is given. This time only the parameter values of models that provide an acceptable match to the objective function are shown. Here, much more concentration of the parameters is evident. For example, all 392 models that give an acceptable match to the objective function were generated with crestal training image 15. This indicates that the entropy-PSO, despite the evidence for a highly exploratory behaviour seen in the dynamic flow behaviour (figure 5.23) and the entropy development of the system (figure 5.24), conducts a much more focused search through the parameter space. Figure 5.27 gives a similar impression. With the help of box-plots it shows how the particles explore each parameter at every iteration. Overall, the box-plots cover a much narrower range at every iteration than the box-plots shown for the standard PSO (figure 5.15), indicating a more focused search through the parameter space. We can, however, also see that the box-plots cover a large area of the parameter space throughout the simulation time and are also capable of quickly changing

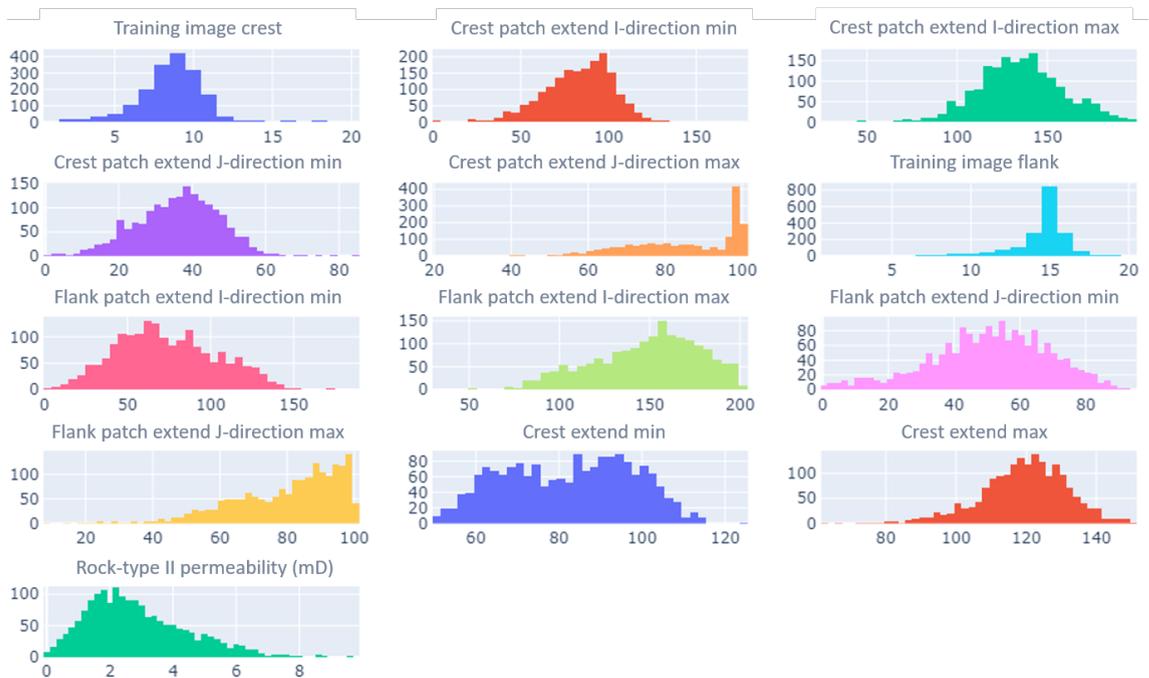


FIGURE 5.25: Histogram distribution for parameter space sampled by the entropy-PSO. All models generated are included in this plot.

the search direction (e.g.  $F3\_I\_MIN$  and  $F3\_J\_MIN$ ).

### 5.3.3.2 Time-of-flight-based risk maps

Taking forward the 392 reservoir models that give an acceptable match to the objective function allows us to generate the risk maps that can be used to support decision-making, for example, for potential infill target zones. In figure 5.28 a flattened representation of the folded reservoir model shows the median time-of-flight (left) and the spatial time-of-flight-based entropy (right) of the best models. The median time-of-flight distribution indicates that with the given set of reservoir models the radial area around the producer in the center of the model always show very low time-of-flight values. As to be expected, the direct pathways between the producer well and the injector wells (corners) also show low median time-of-flight values. When moving away from the centre of the crest towards the flanks, time-of-flight values increase and show less uniform distributions. High and low time-of-flight values more often show up close to each other. This however is not expressed to the same extent on both flanks. The left flank shows overall high time-of-flight values except for an area on the right side of the flank with median time-of-flight values frequently reaching values of more than 20 years. The area between both injector wells



FIGURE 5.26: Histogram distribution for the parameters sampled by the entropy-PSO. Only models that are within an acceptable range of the objective function are included.

that showed the largest area of high time-of-flight values in the standard PSO workflow shown in figure 5.16 also shows a small area of higher time-of-flight values. On the right flank, the distribution of time-of-flight values is a lot more patchy and the area between the two injectors shows overall higher time-of-flight values than the same area on the left flank. However, there is not as clear of a distinction between areas with low and high time-of-flight values as we saw for the left flank. Similar patterns can be picked up for the time-of-flight-based spatial entropy distribution (figure 5.28 right). A radial area around the producer well in the centre of the crest has low entropy values, indicating that the time-of-flight is unaffected by the underlying geology. For the rest of the reservoir, high entropy values are dominant. Again, this is differently expressed on both flanks, similar to the median time-of-flight data. Apart from a few stringers with lower entropy, the right flank shows a homogeneous distribution of high entropy values. Entropy on the left flank is also high but slightly lower and more patchy than on the right flank. This becomes especially apparent for the direct path between the producer and injector well in the top left corner. The general high information content throughout the entire model indicates that the underlying geology impacts the way the reservoir gets swept for most of the reservoir. In comparison to the entropy distribution for the standard PSO shown in figure 5.16, there is not a clear distinction between low and high entropy values when transitioning from the

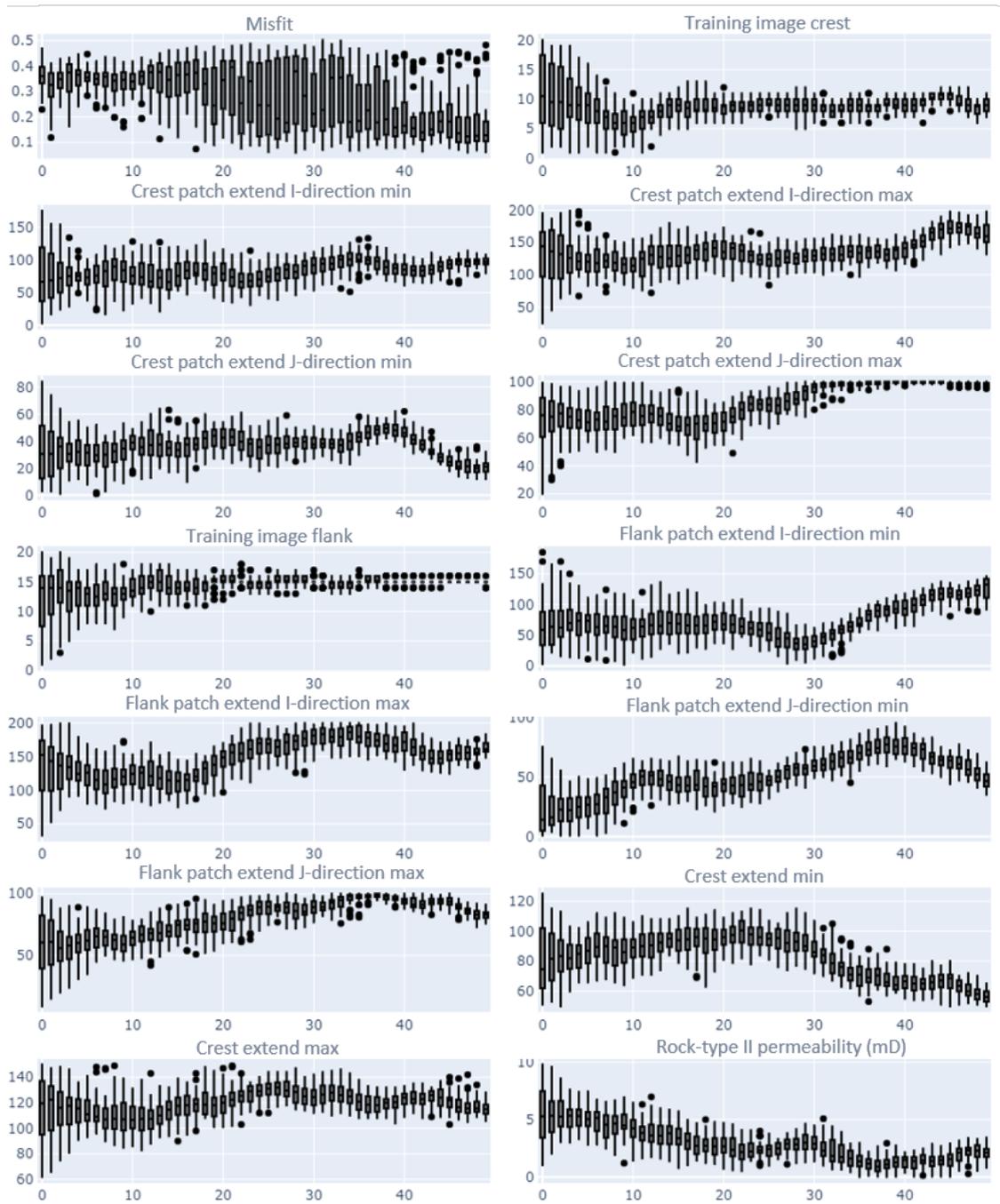


FIGURE 5.27: Box-plots showing the development of every parameter sampled by the entropy-PSO through simulation time. Evolutionary behaviour over iteration time becomes evident, as entropy dictates particles which areas to visit or not to visit.

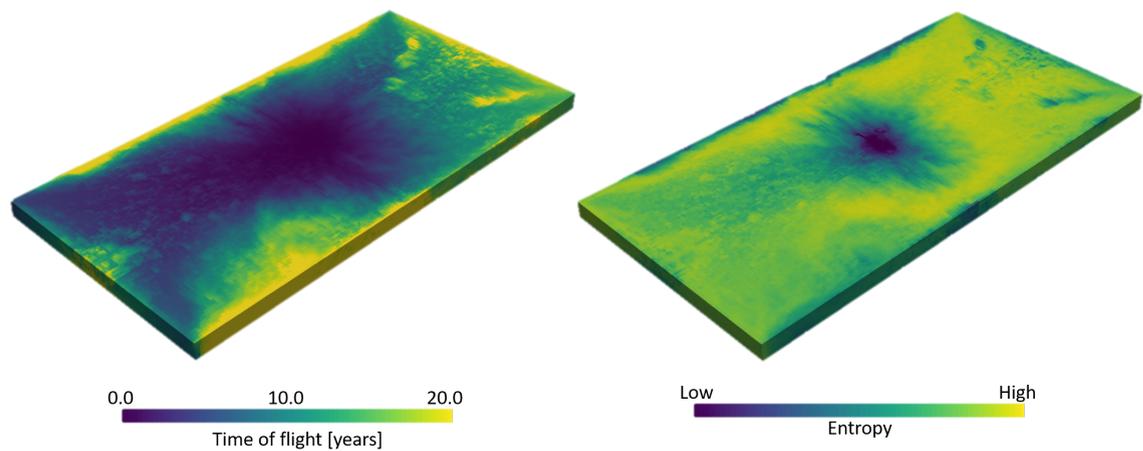


FIGURE 5.28: Flattened visualisation of the reservoir model. Left: Median time-of-flight values shown for all reservoir models that are within the acceptable range of the objective function; Right: time-of-flight based entropy distribution of all reservoir models that are within the acceptable range of the objective function.

crest to the flank.

In figure 5.29 the time-of-flight based risk maps are generated with the spatial entropy and median time-of-flight data given in figure 5.28 and equation 5.10 (right) and equation 5.11 (left). Both represent flattened visualisations of the reservoir model and can be used to identify potentially unswept areas in the reservoir by combining sweeping data (tof) with the given information content (entropy) for each cell of the model. In equation 5.11 we applied a tof threshold of 10 and an entropy threshold of 2.66 (maximum possible entropy is 5.32) to generate the left plot. Dark blue zones indicate areas with entropy and time-of-flight values below the set threshold. This zone can only be found in the centre of the crest around the producer well. The chances for unswept areas in this zone are the lowest of the entire model. Turquoise zones represent areas with a high entropy but time-of-flight values lower than the set threshold. This zone is the most dominant in the model and covers the majority of the crest and the left flank and the pathway between the producer and the left injector well. Despite having a high degree of entropy here, the time-of-flight values lower than the set threshold indicate that most areas are probably swept and the chances for finding unswept areas for potential infill wells are low. The orange zone indicates entropy and time-of-flight values above the set threshold. Again, the high entropy here suggests that the underlying geology heavily impacts how this zone gets swept. However, this time, time-of-flight values are high, indicating a higher chance of finding unswept areas. This zone makes up the majority of the remaining reservoir.

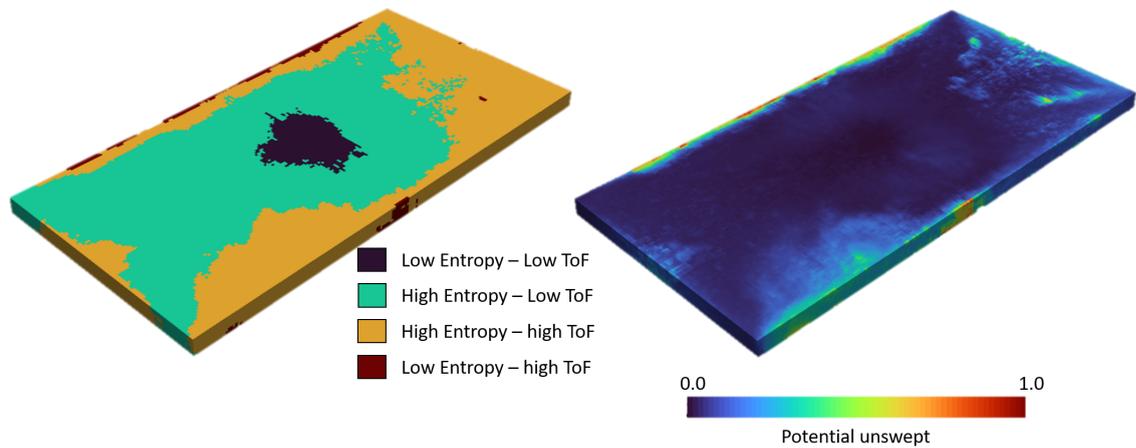


FIGURE 5.29: Flattened visualisation of the reservoir model. Time-of-flight based risk maps generated by combining the spatial entropy and median distribution of time-of-flight values of the models within the acceptable range of the objective function shown in figure 5.28; Left: Segmentation of the reservoir model into sections of low medium and high chance of finding unswept areas according to equation 5.11; Right: Segmentation of reservoir model into potentially unswept zones according to equation 5.10.

It is particularly interesting to see that on the left flank the area that previously showed time-of-flight values of 20 years or more is now also within the orange zone. The dark red zone indicates entropy values below the set threshold and time-of-flight values above the set threshold. This combination gives us the highest chance of finding unswept reservoir areas and the greatest potential for infill wells. Apart from several small shows on the sides of the reservoir model, this zone is not present here. The right risk map in figure 5.29 created with the help of equation 5.10 suggests that there is a low potential for the majority of the reservoir to find unswept areas and targets for infill well. There are several stringers on the right flank and the zone on the left flank that previously already showed high median time-of-flight values. The areas on both flanks between the injector wells could also be potential targets. Both maps together indicate that overall, there is a low chance for successful infill wells. The highest chance of finding unswept reservoir regions would be between the injector wells on the flanks or on the left flank in the area that was already indicative of high median time-of-flight values.

### 5.3.3.3 Dimension reduction and clustering

To work with a manageable ensemble of models, dimension reduction and clustering with the help of UMAP and HDBSCAN is applied to the time-of-flight data of the 392 best models that are within the acceptable range of the objective function. This step

intends to preserve the information obtained from the 392 best models to an acceptable degree. The settings for UMAP were kept at default values apart from  $n\_neighbours = 30$ ,  $min\_dist = 0$  and  $n\_components = 30$ . The settings for HDBSCAN were also kept at default values apart from  $min\_samples = 1$ ,  $allow\_single\_cluster = True$  and  $min\_cluster\_size = 10$ . As can be seen in figure 5.30, the 392 models separate into 12 separate clusters with 11 outlier models not being assigned to any cluster. The time-of-flight data from representative models of a selection of clusters is also displayed in figure 5.30. The visualisations show a flattened representation of the folded reservoir model. The visualisations confirm that models from different clusters get swept differently. Overall there seems to be a strong divide between models from clusters seen on the left side of the diagram compared to models from clusters on the right side of the diagram. On the right side, models show more uniform, low time-of-flight value distributions throughout the reservoir model. On the left side, the distribution of time-of-flight values is more heterogeneous, and we often observe that only one of the flanks experiences low time-of-flight values. The other flank then shows overall higher time-of-flight values with stringers of low time-of-flight values occasionally connecting the producer well with an injector well. Overall, in comparison to the models generated with the standard PSO workflow, there is more diversity in the different distributions of time-of-flight values, which are good indicators for how a reservoir would be swept. This is not immediately apparent, as it seems like the entropy-PSO samples a less diverse range of parameters (see histograms in figure 5.20).

A total of 20 models is now sampled from the 12 clusters. Each cluster is sampled in proportion to its size, allowing each cluster to be sampled at least once. The entropy of the 20 models aggregates to 448.403, which is significantly lower than the entropy achieved with the 392 models, but still higher than the entropy achieved with the 320 models generated with the standard PSO workflow. The distributions for the median time-of-flight data (figure 5.28 left) and the time-of-flight-based entropy data (figure 5.28 right), as well as the time-of-flight-based risk maps (figure 5.29) generated in section 5.3.3.2 are now reproduced with the subset of 20 models. A visual inspection of the median time-of-flight distribution (figure 5.31 left) suggests that the results shown here are almost identical to the median time-of-flight distributions shown in figure 5.28 with 392 models. The

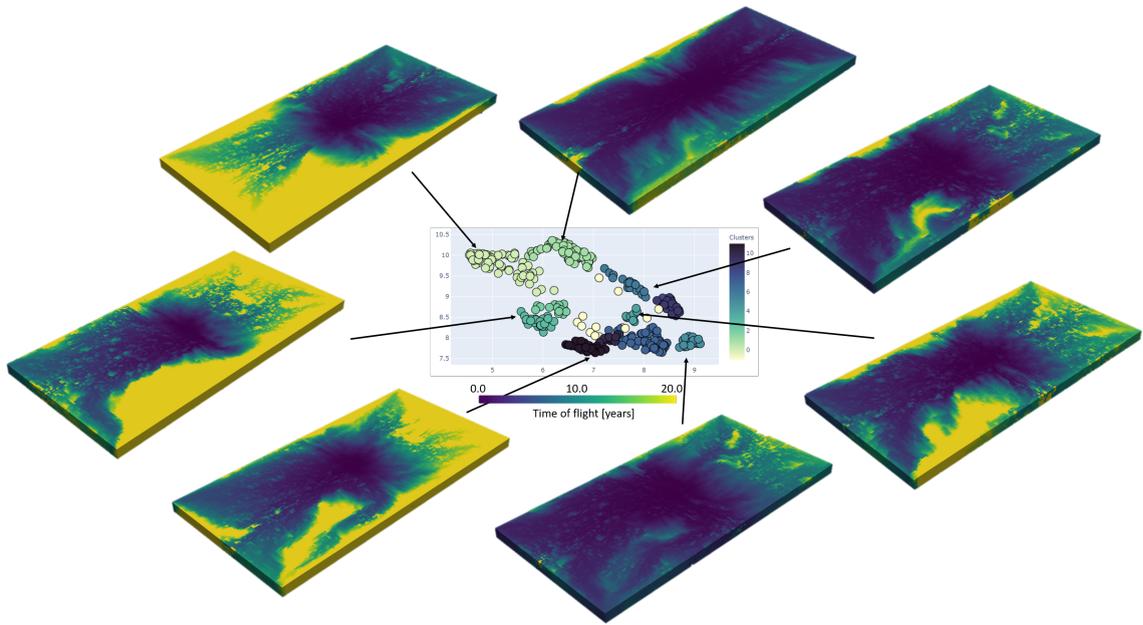


FIGURE 5.30: Time-of-flight-based clustering of models within the acceptable range of the objective function of the entropy-PSO. Each point represents a reservoir model's lower-dimensional representation (30 dimensions). Colours match the associated cluster. Exemplary flattened reservoir models are sampled from a subset of the clusters showing the spatial time-of-flight distribution.

time-of-flight-based entropy distributions (figure 5.31 right) is also almost identical to the time-of-flight-based entropy distributions shown in figure 5.28, but have overall lower highs. The lower entropy values also show when generating the time-of-flight-based risk maps. The same threshold values for equation 5.11 were applied to create the left risk map in figure 5.21 (tof threshold = 10 and entropy threshold = 2.66). The map generated here shows similar trends to the one generated with 392 models. However, the zone of high entropy and low time-of-flight is limited to a turquoise halo around the producer well and a thinner band on the left flank that attempts to connect up with both injector wells. The orange zones of high entropy and high time-of-flight now covers most of the reservoir. This is the zones where we would expect to have a higher chance of finding potentially unswept areas. The dark red zone with low entropy and high time-of-flight values is slightly more present, especially on the right side of the left flank, but overall still has a very limited presence. Overall, the risk map generated with equation 5.10 (figure 5.32 right) shows more potentially unswept areas than the same map generated with the 392 models. Especially between the injector wells on both flanks and the area on the right side on the left flank where we observed high median time-of-flight values. Again, the overall potential of individual cells to be unswept also appears higher than with the 392

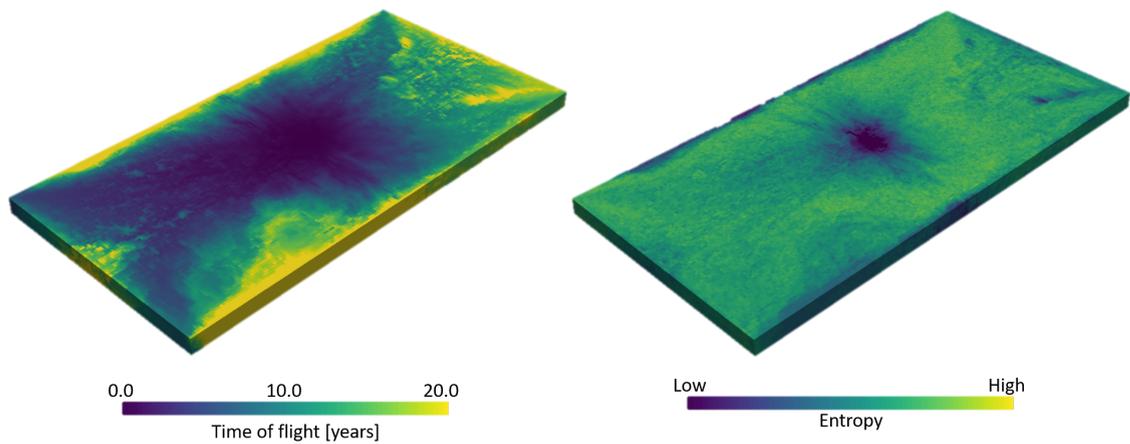


FIGURE 5.31: Flattened visualisation of the reservoir model. Left: Median time-of-flight values shown for 20 representative models sampled from each cluster shown in figure 5.29; Right: time-of-flight based entropy distribution shown for 20 representative models sampled from each cluster shown in figure 5.29.

models.

The evaluation of the 20 models reveals a similar image to what we observed with the 392 models. The time-of-flight-based entropy is high throughout the majority of the reservoir model, indicating that a large degree of information was captured during the clustering and down sampling procedure. However, a distinctive difference is that the results from the 20 models indicate a lot more overall potential for unswept areas and is more *optimistic*. Working with an ensemble of only 20 models makes future decision making more efficient and manageable, but comes with a loss of information.

#### 5.3.3.4 Flow simulations and comparison to reference case

On the subset of 20 reservoir models, we will now run flow simulations to check if they could capture the behaviour of the reference case set up in section 4.4. The simulations will be run under black oil settings, and the same five-spot production pattern as before (one injector well in each corner and a producer well in the centre of the crest) is simulated over ten years. The results of all flow simulations are summarised in figure 5.33. In each plot of figure 5.33, the green, blue, red and yellow lines correspond to a selection of representative patterns and the reference case. The total oil production shown in figure 5.22 a over the ten years ranges from  $0.4 - 1.25 \times 10^6 m^3$ , equating to 12 – 36% oil recovery factors. Despite only working with 20 models, there is are numerous patterns on how

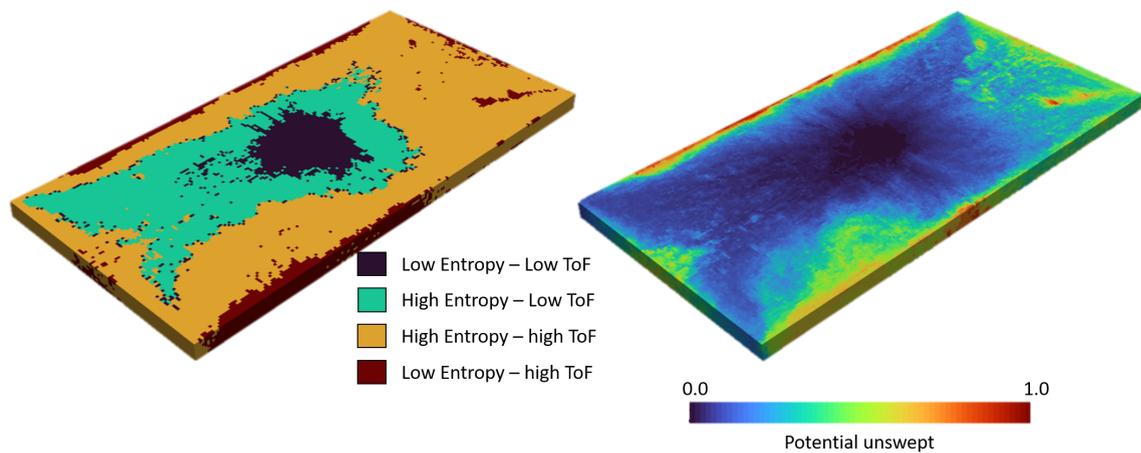


FIGURE 5.32: Flattened visualisation of the reservoir model. Time-of-flight based risk maps generated by combining the spatial entropy and median distribution of time-of-flight values for 20 representative models sampled from each cluster shown in figure 5.29; Left: Segmentation of the reservoir model into sections of low medium and high chance of finding unswept areas according to equation 5.11; Right: Segmentation of reservoir model into potentially unswept zones according to equation 5.10.

the reservoir models behave under production, covering a larger range of responses than we observed with the standard-PSO workflow. With the entropy-PSO, each of the 12 clusters we sampled from shows a more unique production pattern. The same holds for the observed oil production rates ranging between  $120 - 680m^3$  of oil per day. Some models show constant low production rates between  $120 - 220m^3$  of oil per day, whilst other models start with high production rates (between  $400 - 680m^3$  of oil per day) for around 1.5 years that then gradually level down to rates of  $200 - 250m^3$  of oil per day. Similar variations in ranges can be observed for water production. Models with the highest water production tend to also be the models with the earliest water breakthrough (after around one year of oil production) and highest water production (e.g. the third case in green with a total water production of  $1.3 \times 10^6 m^3$ ). These models indicate a high proportion of rock type 1 (fracture dominated). In contrast, models that show initial water production at later stages also tend to have lower overall oil production rates (e.g. case three in red). These models indicate a high proportion of rock type 0 (matrix dominated). Figure 5.22 e depicts the recovery factor of every reservoir vs the overall pore volume of water injected. Interestingly, there is a very narrow range of recovery factors and pore volumes injected that are all within proximity to the reference case. The recovery factor varies between 12 – 35% with 0.16 – 0.54 pore volumes of water injected.

Overall, the subset of 20 reservoir models produced in this workflow with the entropy PSO capture the dynamic response seen in the reference case. The workflow also managed to generate models swept in different ways whilst honouring the underlying assumptions about the geology and fracture distributions. It however seems like the reference case has much more steady and continuous oil production and does not show any signs of water breakthrough so far. Nevertheless, these findings from these models and the time-of-flight-based risk maps can now be used to make informed decisions under uncertainty about potential locations for infill wells. A detailed comparison between the findings of this workflow and the previously introduced workflow is given in the discussion.

## 5.4 Discussion

In this chapter we introduced two new geological modelling workflows that allow for the generation of dynamically driven and geologically constrained reservoir models. The benefit of integrating dynamic data into the reservoir modelling process before the history matching procedure is that we can control the generated reservoir models' dynamic behaviour from the beginning. We can for example create a set of models that all get swept in a very similar way, show similar dynamic responses but are based upon differing underlying geology. We can also create a set of models based upon similar underlying geology but they then get swept in various ways. Creating either only depends on the setup of the geological modelling workflow and is not limited to the exemplary case presented here. This allows us to further delve into those uncertainties with the highest impact on a set objective and better understand how static data affects dynamic behaviour. For example, if we look at the recovery factor versus the pore volume injected shown in figure 5.22 e for the standard PSO workflow and figure 5.33 e for the entropy-PSO workflow compared to the outcomes shown in chapter 4.4 in figure 4.18. In chapter 4.4 we did not have any prior information about the dynamic flow behaviour of the reservoir to take into account. We only looked at the prior geological uncertainties. This chapter included prior information about the dynamic reservoir behaviour and prior geological information into the model generation procedure. The results indicate that by having the dynamic prior information, we were able to generate a much more representative set of models that show similar behaviours to that of the reference case. By focusing more on models with a similar

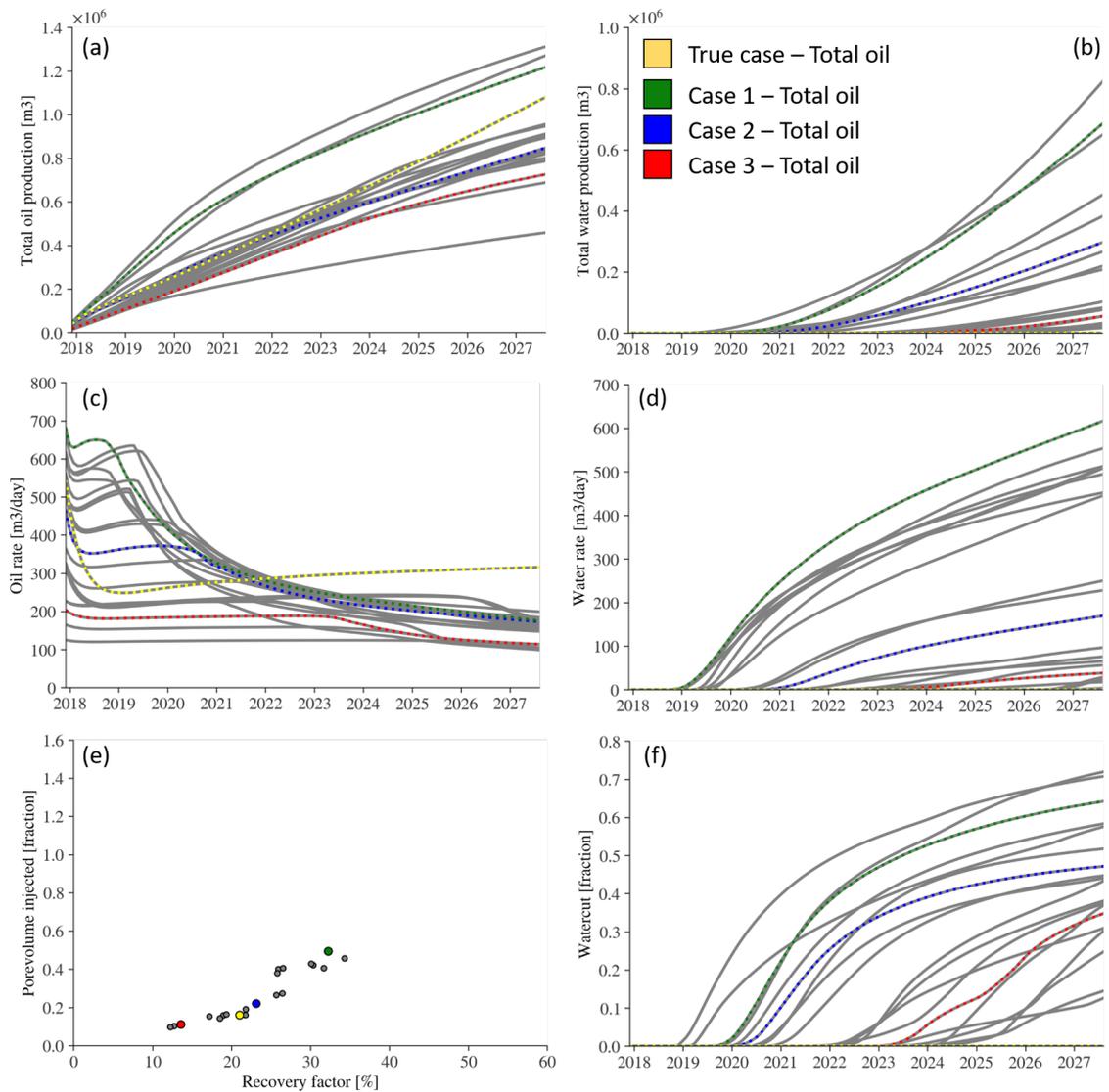


FIGURE 5.33: Results of flow simulation for 20 representative models sampled from each cluster shown in figure 5.29 displaying total oil production (a), total water production (b), oil production rate (c), water production rate (d), pore volume of water injected vs oil recovery factor (e) and water cut of the producer well (f).

dynamic response, we were also able to explore the uncertainty about how the underlying geology together with the given dynamic response, could still lead to numerous different ways the reservoir gets swept.

An important building block in this workflow is using flow diagnostics instead of full physics flow simulations. For example, for the case study described above, a single reservoir model was automatically generated, checked on its flow behaviour with flow diagnostics and evaluated on its fits with the outset objectives within 2 minutes on a standard workstation. This allows for the generation and evaluation of thousands of

models within a short time frame, which would not be possible with full physics flow simulations. Another essential building block used in both workflows is the time-of-flight-based entropy. With entropy, we now have a measure that objectively informs us about the diversity of the reservoir models we generated. We use it either as a restarting criterion (standard PSO) that tells us when no more novel, matching models are explored or it is directly involved in particle movement and model generation (entropy PSO). By calculating the entropy for a dynamic property, we can also translate geological uncertainties to their impact on dynamic flow behaviour. An improvement on the current calculation for the systems entropy to combine the entropy approaches of Wellmann and Regenauer-Lieb [2012] used here and Journel and Deutsch [1993] used in chapter 4. Instead of adding up the entropy values of single cells to calculate the systems entropy, we could bin the entropy values and have a moving window or cube screen the through the model. By counting the number of unique templates and their frequency that would be required to rebuild the model we can calculate the systems spatial entropy. We would not be able to visualise this entropy, but it would be a useful measure for the diversity of the system as it better captures the spatial component. Finally, we can select a representative and manageable subset of reservoir models to take forward by applying dimension reduction (UMAP) and clustering techniques (HDBSCAN) to the dynamically preconditioned models that fit with the outset objectives. These can now be used within history matching workflows and future decision-making procedures. The time-of-flight-based entropy distributions also indicate how much information was preserved during the down-sampling step and what the trade-off for a more manageable subset of models costs us in information. We can also use the time-of-flight-based entropy distributions to monitor areas that experienced the most significant loss of information during the down-sampling step. There are however also shortcomings with fully automated workflows. The extended flexibility provided by automated workflows makes it even more important to select a sensible prior geological parameter range. Otherwise, the workflow tends to build models that fit the outset objective but do not make geological sense. Therefore, it is always required to have a geoscientist or geomodeller inspect the generated models to verify their realism.

When comparing the results of the standard PSO workflow with the entropy-PSO workflow, we observed that in both cases, it takes several iterations before the entropy for the

system of best models increases. Once this happens, we observed a rapid expansion of the time-of-flight-based entropy with every additional iteration of the workflow. After reaching a plateau, subsequent improvements in entropy are substantially smaller. This makes sense, as initially every new model that has an acceptable match to the objective function will extensively increase the information we have on how the potential time-of-flight distribution could look. Additional models will only incrementally increase the overall information. It was also interesting to see how this initial rapid increase in entropy lasted longer for the entropy-PSO. This shows how including entropy into the decision-making process of each particle allows capturing a more extensive scope of uncertainty. The standard PSO can also run into the problem of just circling local minima, constantly generating very similar models. On the other hand, the entropy-PSO uses entropy to actively push particles out of these local minima once explored and look for new, undiscovered minima. In comparison to the standard PSO, this allowed the entropy-PSO to produce 72 additional models that give an acceptable match to the objective function whilst showing different sweeping behaviours and production profiles. This became evident during the dimension reduction and clustering step. Both workflows generated over 300 matching models grouped into several clusters. For the standard PSO these clusters still fell into three overarching groups, reiterating the circling around local minima thesis. This overarched hierarchical grouping was not possible for the entropy-PSO. Unlike the standard PSO, the entropy-PSO was also able to achieve higher entropy values throughout the model. Therefore, areas that appeared to have a relatively low risk when analysed with the standard PSO, seemed to have less potential when checked with the entropy-PSO (figure 5.34). This is also verified when running the flow simulation of the reference case for 20 years. The majority of the reservoir gets swept with the current well setup. The time-of-flight-based risk maps indicate that the entropy-PSO does not find outstanding areas for potential infill wells. However, we also observed that more diverse cases are discovered with the entropy-PSOs current setup. Still, there is also a risk of not exploring some areas of the parameter space at all, at least if iteration time is not increased. Evidence for this behaviour can be found in figure 5.28 and figure 5.29. We can observe that the left flank has overall lower median time-of-flight values and lower potential to find unswept reservoir regions than the right flank. As we do not differentiate between either flank when we set up our geological priors explored by the

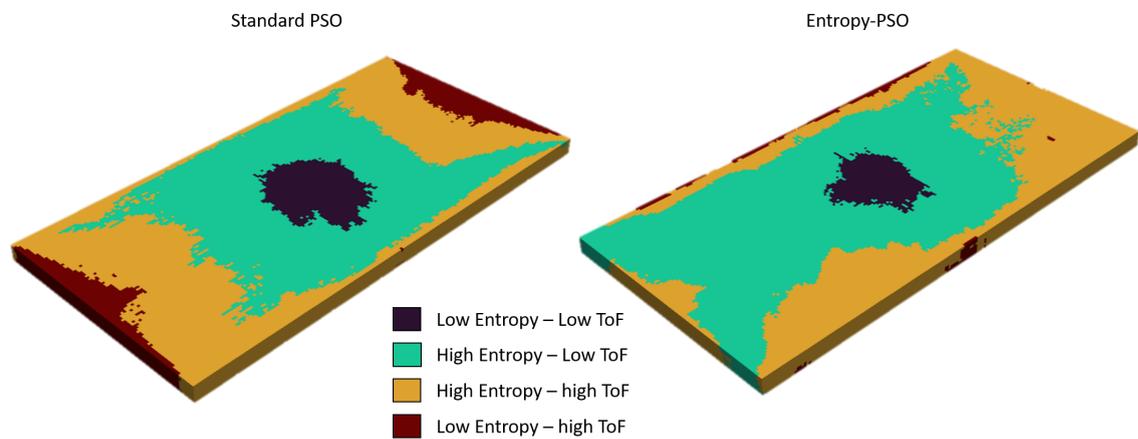


FIGURE 5.34: Flattened visualisation of the reservoir. Comparison of the time-of-flight based risk maps for the standard PSO (left) and the entropy-PSO (right).

entropy-PSO, we would expect a more consistent behaviour on both flanks, as observed in the standard PSO. Here exploration is more uniform, which is again expressed in figure 5.28 and figure 5.29, showing similar distributions for the median time-of-flight values and the chance of finding unswept areas on both flanks. However, as we already discussed, the standard PSO also misses out on plenty of information and does not explore the full range of uncertainty. A potential solution to the entropy-PSOs under-exploration in some areas would be to initiate multiple starting populations on different simulation runs to ensure that the parameter space is explored more thoroughly. Another alternative could be splitting up the swarm into several sub-swarms that explore the parameter space more independently. Not all information is shared amongst particles from different sub-swarms.

The time-of-flight-based risk maps demonstrated that they are a good tool to use the information from the entire ensemble of models with an acceptable match, without the need to move forward with every single model. They are also a good way to identify where we lost most information during the down-sampling step. By combining entropy and the median time-of-flight, we can now identify the areas that are most likely to be unswept based on all the available information. The risk maps generated with equation 5.11 offer the opportunity to introduce additional, subjective selection criteria to narrow down the areas of interest. This however also comes at the cost of potentially introducing our own human biases when evaluating a reservoir by selecting different thresholds for the median time-of-flight and entropy. This should always be considered for this method as results can easily be maneuvered in either direction. This problem does not occur when

working the more objective risk maps based on equation 5.10. They however do not allow for the inclusion of additional knowledge or information. Therefore it is recommended to always work with both methods in conjunction when evaluating potential targets for infill wells. In the synthetic field studied in this chapter, the regions with the highest chance for being unswept were between the injector wells on both flanks. This seems intuitive for the simplistic case studied here and would probably be obvious for most reservoir engineers even without the generated risk maps. This might however not be this easy in more complex scenarios. Here, these maps can be used to quantitatively identify areas that have the lowest likelihood of being swept, whilst taking into account the geological prior information together with information about the dynamic behaviour of the reservoir. We observed that the risk maps generated with the subset of 20 models are more *optimistic* for potential unswept reservoir zones. The explanation for this effect can be traced back to the time-of-flight-based entropy. As shown in equation 5.2, each cell's time-of-flight values are binned into  $m = 40$  equal sections. This means that maximum entropy for a cell is achieved when it has a uniform distribution of 40 different time-of-flight values. When working with a subset of only 20 models, achieving maximum entropy with  $m = 40$  is impossible. We observed that this does not represent too much of a problem for models generated with the standard PSO workflow. The entropy of the entire system of matching models calculated with equation 5.3 was similar for all models and the subset of 20 models. For the entropy-PSO workflow, the loss of information during the down-scaling step is more significant and ideally more than 20 models would be included in the subset of models to take forward. This means that the amount of information or uncertainty we are capable to capture with our models, also dictate what size of ensemble is at least required to capture a comparable amount of information. What is also worth highlighting is that the evaluations performed here depend on the time interval considered. In the example presented, water-breakthrough in the producer wells has already occurred which means that there is a lot more information available about the sweeping behaviour and the movement of the water front in the reservoir. A different case would be a reservoir prior to water-breakthrough where uncertainty would be significantly larger as water is not picked up in the wells and more uncertainty exists about the water front. In this case entropy would probably be a lot higher as there are more possible configurations without this additional piece of information (water-breakthrough in producer wells). What is only

partially covered with an approach focusing solely on the time-of-flight based entropy to steer model development is the underlying geology. This method only looks at the impact that the underlying geology has on fluid behaviour, but information about the geology itself is not directly considered for reservoir model steering. If, for example, two similar sweep patterns are generated by different underlying geological settings, the workflow developed here will not differentiate between the two settings and potentially discard one geological scenario. An example where this could become problematic would be when the current reservoir acts as an analogue model for a newly explored reservoir. A possible solution to this issue for future approaches could be to also calculate the entropy of the underlying static reservoir model and use a combination of this static *geological* entropy and the time-of-flight based entropy for the steering of model development.

Another topic that needs addressing is that in the current workflow setups, each of the 20 models from the subset is assumed to be equally probable, which is very unlikely to be true [Hutahaeen, 2017]. In a real setting, we would not know the probability of each model but, we can assign a likelihood to each model, based on the discrepancy between observed and simulated data. Other approaches are to weight each of the 20 models based upon relative cluster size [Scheidt and Caers, 2009] or the sum of posterior probabilities of each cluster [Hutahaeen et al., 2019]. A more geologically driven approach would be a weighting of the input data based on the modellers underlying conceptual understanding of structural geology. Implementing either approach into the current workflow would make it more reliable for uncertainty prediction.

A final note has to be made and how the concept of maximum entropy shines through within the entropy-PSO workflow. Maximum entropy is associated with the idea of maximising information entropy by finding the smoothest possible probability distribution that is compatible with the given constraints expressed as prior knowledge [Conrad, 2004, Edwin T. Jaynes, 1982, Guiasu and Shenitzer, 1985, Jaynes, 1957, Shore and Johnson, 1980, Wu, 2012]. Applications for the maximum entropy theory range across numerous disciplines, such as improving image resolution in medicine or forensics [Skilling and Bryan, 1984], inferring input values for gaps in economic data [Golan et al., 1994], improving estimates of climate envelopes [Hijmans and Graham, 2006], predicting linkage distributions in

nodes networks [Wang et al., 2007] and predicting the shape of static and dynamic micro-scale patterns in ecology [Harte, 2011, Phillips et al., 2006]. A fundamental element for choosing probability distributions is based on entropy. Jaynes [1957] contend that the maximum entropy distribution is *"uniquely determined as the one which is maximally noncommittal with regard to missing information, in that it agrees with what is known, but expresses maximum uncertainty with respect to all other matters"*. Therefore, an entirely unconstrained problem would lead to a uniform distribution, meaning that we are selecting from a uniform prior without constraining the parameter space with any biases. If we however sample from a distribution in a different way, we implicitly make assumptions about the system that we previously stated not to have any information about [Edwin T. Jaynes, 1982]. If we know the mean and the standard deviation of a distribution, the maximum entropy would be given with a normal distribution that reflects this prior information. In complex systems, the maximum entropy can therefore be used to measure macro-scale developments to infer micro-scale distributions. In our case, the macro-scale entropy is the overall entropy measurement for the collection of best models that all match our outset objective and maximise entropy. The micro-scale in this case would be how to sample the geological parameter space under the given boundary conditions. In our case, we did not apply any constraints on how the entropy-PSO should sample the parameter space. The only restrictions we gave were a desired flow diagnostic response, a tolerance range around this response and the prerequisite for new models to increase the entropy of the existing system of matching models. In the histograms of figure 5.26, we showed the parameter space sampling for all models with an acceptable match to the outset objective function. The objective function and the error tolerance we predefined now act as prior constraints. This is reflected by the sampling distributions in the parameter space. Many of the sampled parameters show distributions that from visual inspection show characteristics of a normal distribution. This allows us to infer, that achieving maximum entropy with the constraints on the dynamic output space, led to a sampling of the input space that would suggest we do have information about the static input space. We can now observe normal distributions with a given mean and standard deviation for some of the parameters that provide us with the maximum information under the given constraints. For other parameters, like for example *Hinge\_min*, this is not the case. Here it appears that we have two normal distributions overlapping. It implies that at least two configuration modes for

this parameter allow us to generate matching reservoir models. There are three possible explanations for Parameters that do not show normal distributions: (i) we either did not explore the parameter space for long enough, (ii) the parameter has no impact on the output space and is therefore neglectable or (iii) there is an overall larger degree of uncertainty around these parameters that cannot currently be captured and would require the entropy-PSO to run for more iterations. For a large majority of the parameter space, we obtained a normal distribution with a given mean and standard deviation that captures the uncertainty for the output space under the given constraints. Much work has been done to preserve diversity in particle movement within the parameter space [Blackwell and Bentley, 2002, Blackwell, 2005, Higashi and Iba, 2003, Hu et al., 2013, Jin and Branke, 2005, Li, 2010, Liang et al., 2006, Ling et al., 2008, Siarry et al., 2002, Sun et al., 2006, Wang et al., 2013, Zhan et al., 2009]. We were more interested in preserving the spatial diversity of a constrained output space. This constraint on the output space can now be translated to a measure for uncertainty of the input space.

# Chapter 6

## Agent-based modelling for naturally fractured reservoirs

### 6.1 Introduction

In the previous chapter we made use of a particle swarm optimizer to guide the evolution and update of reservoir models to match a particular dynamic response. Within a swarm, every particle is in control of a set of parameters that determine the characteristics of the reservoir model to be generated. The reservoir models developed are controlled by the behaviour of each particle, but also the swarm as an entity. The models are, however, restricted by the predefined prior parameter space that particles can roam across in search of models that minimise the objective function. A subset of these parameters are for example responsible for the setup of a geostatistical modelling algorithm that populates the reservoir model with petrophysical properties. Here, setting up an appropriate parameter space can become challenging when the models aim to capture complex problems or take into account dynamic data. This is because geostatistical tools tend to be rather static, whereas the reality is often highly variable and non-stationary. For example Wu et al. [2007] and Rojas [2013] introduced the region concept for MPS modelling to take into account non-stationarity within a reservoir model. They make the argument that a reservoir can be subdivided into sub-zones, where each sub-zone is so different to other sub-zones, that it requires for the use of different training images and parameters. This

could for example be during the transition between different depositional environments or environments that contain strongly varying geobodies such as deltas (e.g. distributor channels and mouth bars). Overall, using parametric geostatistical tools to model this non-stationary behaviour with the help of different trends or region concepts can quickly become extremely parametric and overly complex.

Instead of using linear deterministic parametric models to introduce structural trends or sub-zones, one could also use a non-parametric adaptive algorithm to define the trend model and even update it to match dynamic data. Therefore, in this chapter we introduce an agent-based approach for dynamically driven reservoir modelling. In agent-based modelling (ABM), autonomous agents simulate complex adaptive systems by following a set of simple predefined rules when interacting with each other and their environment. Each agent's behaviour can be goal-directed, allowing that agent to compare the outcome of its behaviours against its goals and adjust its responses accordingly. An ABMs setup allows it to evolve over simulation time and generate outcomes that are less confined by a parameter space that we defined. Instead, it can evolve over time with the help of the rules, interaction with the environment and agent-agent interactions. This freedom however also comes at the costs of not being in full control of the model development anymore. In this chapter we combine agent-based modelling with flow diagnostics, to iteratively update the generate reservoir models and for example fit the model to newly incoming production data. The versatility of ABMs allows us to address a wide bandwidth of different questions with only minor modifications of the initial setup of the ABM. Here we apply the ABM to assist with updating prior conceptual geological models in uncertainty quantification studies.

The chapter is organised as follows. In section 6.2 we emphasise the importance of geological conceptual models and where agent-based modelling can make its contribution to improve geological conceptual models. In section 6.3 we propose the overall methodology of our ABM as designed to be used for updating prior conceptual geological models. In section 6.4 we again apply the workflow to the synthetic case study setup in chapter 2, but this time try to update the underlying conceptual model according to our objectives. Parts of the work carried out in this chapter are based on the work presented in Steffens et al. [2021a] and also took inspiration from Steffens et al. [2022].

## **6.2 How agent-based modelling can improve conceptual geological models**

Building numerical reservoir models of the subsurface heavily relies on geological conceptual models and domain knowledge like prior geological information [Loayza et al., 2016]. This is because acquiring subsurface data such as seismic, well and core data is expensive, time consuming and their overall availability is sparse. The geological conceptual models allow geoscientists to use the sparse data and integrate them with their understanding of the subsurface to make predictions on likely spatial distributions of facies, deformation structures and associated reservoir properties. Sources for conceptual models are analogue outcrops, process-based models, sequence stratigraphic frameworks, physical experiments, geomechanical simulations, analogue fields and modern-day equivalent depositional environments [Couples and Lewis, 1999, Franzel et al., 2021, Grammer et al., 2004, Howell et al., 2014, Lewis et al., 2009, Mullins et al., 2021, Van Wagoner et al., 1990]. It is however important to remember that conceptual models are only based upon scarce data, analogues and geological understanding. Even if the overall concept that a geologist has in mind is correct, the spatial distribution and exact location of different facies or rock properties can vary substantially. This could be because vital information are not recorded within the sparse data. For example the presence of a fault is not captured or the expected extension of a specific facies type is lower than what the conceptual model suggests. It can also be that additional data, such as production data, become available over time that invalidate the current conceptual model or make other concepts more likely. The new data can then also be used to test and perhaps reassess the existing conceptual model. Conclusively, decision making for ongoing development or new development of a reservoir is heavily reliant on conceptual geological models.

Agent-based modelling describes a class of algorithms that is well suited to refine and test how well a conceptual model fits with an expected or recorder reservoir flow response. Part of the agents' objective in this study is to evaluate how well the conceptual model at an agents current position matches with the predefined and the expected flow diagnostic response. The potential for hundreds of iterations and agent movements makes using computationally heavy full physics fluid flow simulations to evaluate the flow response

of the reservoir infeasible. Therefore, agents use flow diagnostics to assess the goodness of their position. Apart from a few recent studies [Azam et al., 2015, De Kemp, 2021, Jaxa-Rozen et al., 2019, Steffens et al., 2022] however, ABMs have little application in the subsurface modelling domain. Therefore, in this work we attempt to use agent-based modelling to update conceptual geological models used for reservoir modelling. It should be reiterated that this is only one possible application area of this particular ABM as it could also be utilised to generate an ensemble of reservoir models that try to capture geological uncertainties, similar to the studies demonstrated in the previous chapters. First we will introduce the setup of the proposed agent-based model, what defines the environment and what rules dictate the agent's behaviour. Then the workings of the implemented rules are validated.

## **6.3 Methodology**

### **6.3.1 Agent-based model design**

The objective of the ABM is to build a reservoir model that balances the expected, predefined flow diagnostic response of the reservoir model with the underlying conceptual geological model. If this balancing act is not resolved sufficiently, the ABM rules allow the agents to update the underlying conceptual geological model until an acceptable balance is achieved. In this section we will describe the ABM design that was chosen for this study. The first part defines what constitutes the environment and then the agents and their behaviour and how they can update the conceptual model are described in more detail. This is followed by a quick validation of the functionality of every rule.

#### **6.3.1.1 Environment**

The aim of the agent-based model developed for this work is to test and potentially update a conceptual geological model for a subsurface reservoir model. For simplicity, the conceptual model is represented as a two-dimensional grid (figure 6.1). The grid represents the environment of the ABM in which the agents can interact and move around. A three-dimensional environment implementation is also possible but would increase the computational costs and the model complexity. Also, a two-dimensional conceptual

model aligns well with the case study showcased below, where lateral rather than vertical variability of fracture distributions are investigated. Each cell of the grid gets a zone assigned to it that represents the underlying conceptual model. For example, in figure 6.1 the cells overlying the channel system (yellow) in the conceptual model will be assigned to the channel system zone.

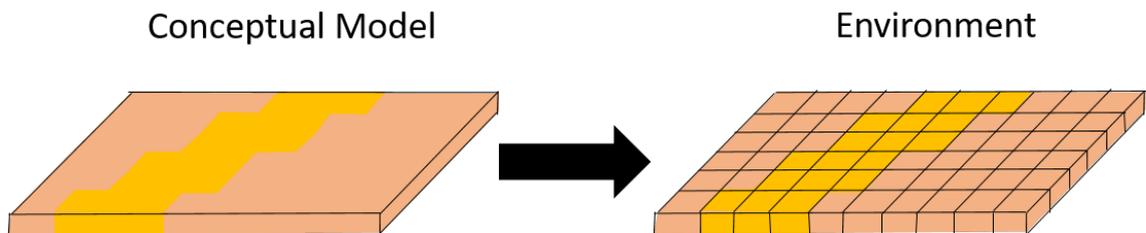


FIGURE 6.1: Transformation of the conceptual model (left) into the gridded agent environment (right). For simplicity the environment is a 2D representation of the conceptual model.

### 6.3.1.2 Agent behaviour

The way agents move and interact with the environment and with one another is the core of the ABM algorithm implementation. In our work the agents are positioned within the environment and then compete against each other for space (figure 6.2 a). Each grid cell of the environment is assigned to an agent. This is done by dividing the environment between the agents with the help of Voronoi tessellation. Voronoi tessellation allows partitioning of an n-dimensional space - in this case a 2D surface - into convex polygons closest to a set of points (agents) on the surface [Dobrin, 2005] (figure 6.2 b). According to the selected distance metric (e.g. Euclidean or Manhattan), each agent becomes the centre of a Voronoi polygon and while the agents move around, the Voronoi tessellation adjusts accordingly. Polygon boundaries are formed where there is no single nearest point (agent). Voronoi tessellations are also called Voronoi decomposition, diagrams or Dirichlet tessellations, abundantly occur in natural patterns and have applicability in various fields of science, technology and arts [Du et al., 1999, Ju L. et al., 2011, Okabe et al., 2009]

Now each agent and its associated Voronoi polygon are assigned to a conceptual model zone that is captured by the underlying environment. This is done by checking what conceptual model zone each grid cell within the polygon belongs to. A majority vote then assigns the dominating conceptual model zone to the entire polygon (figure 6.2 c). In

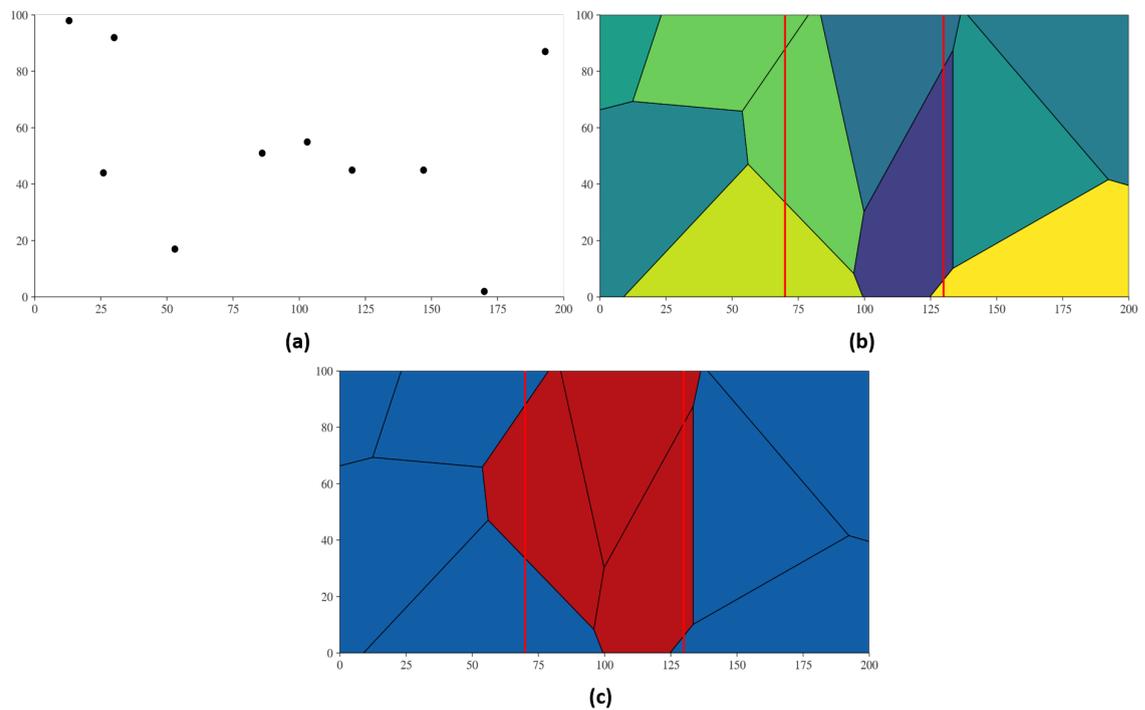


FIGURE 6.2: (a) Random placement of agents (points) within the environment. (b) Partition of the environment with the help of voronoi tessellation. Randomly colour-coded. (c) Assignment of each polygon to the occupied grid cell characteristics of the underlying environment. Polygons are colour-coded by the underlying conceptual model that the majority of the polygon is placed in. In which zone the agent itself is placed, does not play a role here. The red lines indicate a transition from one conceptual model zone (blue polygons) to the next (red polygons).

figure 6.2 b and 6.2 c the red horizontal lines indicate a transition from one conceptual model zone to the next. With this method the complexity or resolution of the conceptual model is controlled by the number of active agents and increases with more agents.

So far, every point of the 2D environment is represented by a unique agent, the corresponding polygon that include the agent location and has a zone of the conceptual model attached. For simplicity the 2D information of the conceptual model is now projected onto the 3D reservoir model. This means that the assumption is made that the 2D environment can simply be extended vertically into 3D without any alterations. From a geological perspective this means that assumptions made about the underlying geology are constant in vertical direction (z direction) and only vary horizontally (x and y direction). The next step is to use this information to populate a reservoir model with petrophysical properties. There are many options to do this for example, picking a standard geostatistical method such as sequential indicator simulation (SIS) that considers the information from the steps

mentioned previously [Journel et al., 1998]. In this work we decided to use multiple-point statistics (MPS) to perform the property modelling [Strebelle, 2002]. Depending on which zone of the conceptual model a polygon reflects, the agent gets to pick from a set of different training images that take into account the underlying geological concept and related geological uncertainties. Agents therefore represent the link that transfers geological prior knowledge onto the reservoir model. MPS is then used to populate the polygon based on the selected training image. The process of choosing and picking a training image is explained in more detail below. After every single agent has populated their surrounding polygon with the selected training image and a complete reservoir model is created, flow diagnostics are run on the reservoir model. It should be noted that at the ABMs initiation point, the reservoir model will not be geologically consistent as agents randomly select from numerous training images to build the reservoir model. Geological consistency will be achieved over ABM simulation time. The results from the flow diagnostic run are then used to calculate quantitative metrics such as flow capacity- storage capacity diagrams or the dynamic Lorenz coefficient. The misfit of these metrics to the predefined or the expected flow diagnostic response is then calculated.

### **6.3.1.3 Agent movement**

The agents are randomly placed within the 2D environment. An improvement to random agent placement could be a more clustered approach for placement, where agents will have a higher probability to spawn in areas of higher uncertainty, for example in the transition between different zones. Other prior information, such as seismic data could also be used to dictate agent placement. Essentially the random agent placement reflects an uninformative prior - any concept or agent configuration is possible. If the agents are not randomly placed they can be linked to an informative prior or concept. It is possible to keep the number of agents constant throughout the ABM simulation or to add more agents iteratively. At each iteration, every agent has to decide which grid cell to move to and which training image to choose. The objective that drives the agents decision process is to optimise the match to a set of predefined agent governing rules. The rules are explained in the next section. To limit computational time and introduce a random component into the simulation, agents can not test every training image combination in every grid cell. Instead, they are forced to pick from a predefined number of possible movements. Each possibility represents a

random draw from a selection of training images and a randomly selected grid cell within a set radius around the grid cell the agent is currently present in (figure 6.3). Agents also have the option to stay at their current position or remove themselves entirely from the environment. If an agent already occupies a grid cell, another agent can not move to the same grid cell.

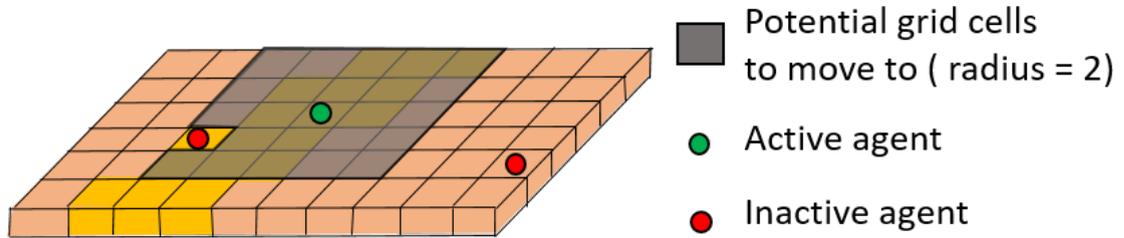


FIGURE 6.3: Possible grid cells for the active agent to move to with radius set to 2. Notice how the active agent can not move into cells occupied by inactive agents. For each cell the agent can also pick from a selection of petrophysical properties or training images.

#### 6.3.1.4 Agent rules

After defining what possibilities for movement agents have and how this movement defines the generated reservoir model and the flow diagnostic response, we now look at the rules that drive the agents' decision-making process: which position on the grid to move to and which training image to select. The agents selection process for the best position is driven by three rules. The process can be described with equation 6.1:

$$\mathbf{Position}_{\text{best}} = \min(\mathbf{w}_1\mathbf{F}_i + \mathbf{w}_2\mathbf{C}_i + \mathbf{w}_3\mathbf{T}_i). \quad (6.1)$$

where  $\mathbf{F}_i$  is the reservoir models fit with the flow diagnostics rule at agent position  $i$ ,  $\mathbf{C}_i$  the reservoir models fit with the conceptual model boundary rule at agent position  $i$  and  $\mathbf{T}_i$  the reservoir models fit with the training image preference rule at agent position  $i$ . The weighting factors  $\mathbf{w}_x$  allow for the adjustment of the importance of every rule.

The flow diagnostics rule compares the expected flow diagnostic response with the flow diagnostic response we get from the reservoir model with the agent at position  $i$ . In our case we are interested in minimising the misfit to the shape of predefined flow capacity-storage capacity diagrams as well as the resulting dynamic Lorenz coefficient [Shook and

Mitchell, 2009]. This is done by calculating the root mean squared error (RMSE) between the expected and the actual flow capacity- storage capacity diagram, its first and second-order derivative, and the dynamic Lorenz coefficient as already shown in equation 5.1 in chapter 5. The resulting outcomes of all potential positions are then scaled between 0 and 1. With this rule we intend to inform the agents about dynamic data that we have about the reservoir and include them in the agents' decision making and model building process.

The conceptual model boundary rule preferentially directs agents towards the areas where the conceptual model zones change. This allows for a more accurate determination of the TI region boundaries with the higher density of agents along these boundaries and therefore a better refinement of these areas (figure 6.2 c). The rule measures the absolute distance between the boundaries and each potential position the agent tests out. The outcomes from all possible agent positions are then scaled between 0 and 1.

The training image preference rule measures the quality or "goodness" of the training image options the agent can pick from and enforces geological consistency. The quality score for each training image varies between 0 and 3 and is based upon its coherence with the training images of the surrounding agents and the underlying conceptual model the agent is positioned in. The lower the score, the better the training image. If the training image under investigation matches the conceptual model assigned to the agent and is identical to the most common training image of the surrounding agents, a quality score of 0 is given to the training image. If it only matches the conceptual model assigned to the agent, it gets a score of 1. If it matches the most common training image of the surrounding agents but does not match the conceptual model assigned to the agent, the training image receives a score of 2. If the training image does not fit any selection criteria, it is given a quality score of 3. The quality scores of all potential training images are then scaled between 0 and 1. This rule intends to (i) reduce the models' complexity by sticking with as few training images as possible and (ii) reward the agents for honouring the underlying conceptual geological model. The rule penalises agents if they try to select from training images that do not match the underlying conceptual model. Therefore agents only pick training images that diverge from the underlying conceptual model if a sufficient balance with the other two rules can not be achieved.

### 6.3.2 ABM workflow summary

The agent-based modelling workflow developed in this work can be broken down into an initiation phase and an iterative phase that is repeated  $n$  times and summarised in figure 6.4 and pseudo algorithm 3 :

---

**Algorithm 3:** Pseudo-code for agent-based reservoir modelling

---

**–Initialisation phase–**

Project geological conceptual model onto environment

**for each agent  $i$  do**

    | Random placement in environment

**end**

Partition of environment among agents with Voronoi tessellation

**for each agent  $i$  do**

    | Assign conceptual model zone based on prior information

    | Randomly select training image from assigned conceptual model zone

**end**

**–Optimisation phase–**

**while** *maximum iterations or minimum misfit criteria not reached* **do**

**for each agent  $i$  do**

        | Check surrounding free grid cells and randomly pick  $k$  positions together  
        | with a sampled training image

**for each position  $k$  do**

            | Update Voronoi tessellation

            | Update reservoir model

            | Run flow diagnostics

**end**

        | Move agent to position  $k$  that optimises objective of implemented agent rules

**end**

**end**

---

### 6.3.3 Validation of implemented rules

ABMs are designed to develop complex behaviours with only a few simplistic rules implemented. To make sure we fully comprehend the agents movement a validation case is set up. Here, each of the three rules is tested out individually by setting the weighting factors  $w_x$  of the untested rules in equation 6.1 to 0. This makes them irrelevant to the agents' decision process. The conceptual model used in the validation case is similar to the one introduced in the case study section below. It is projected onto a 2D grid that measures

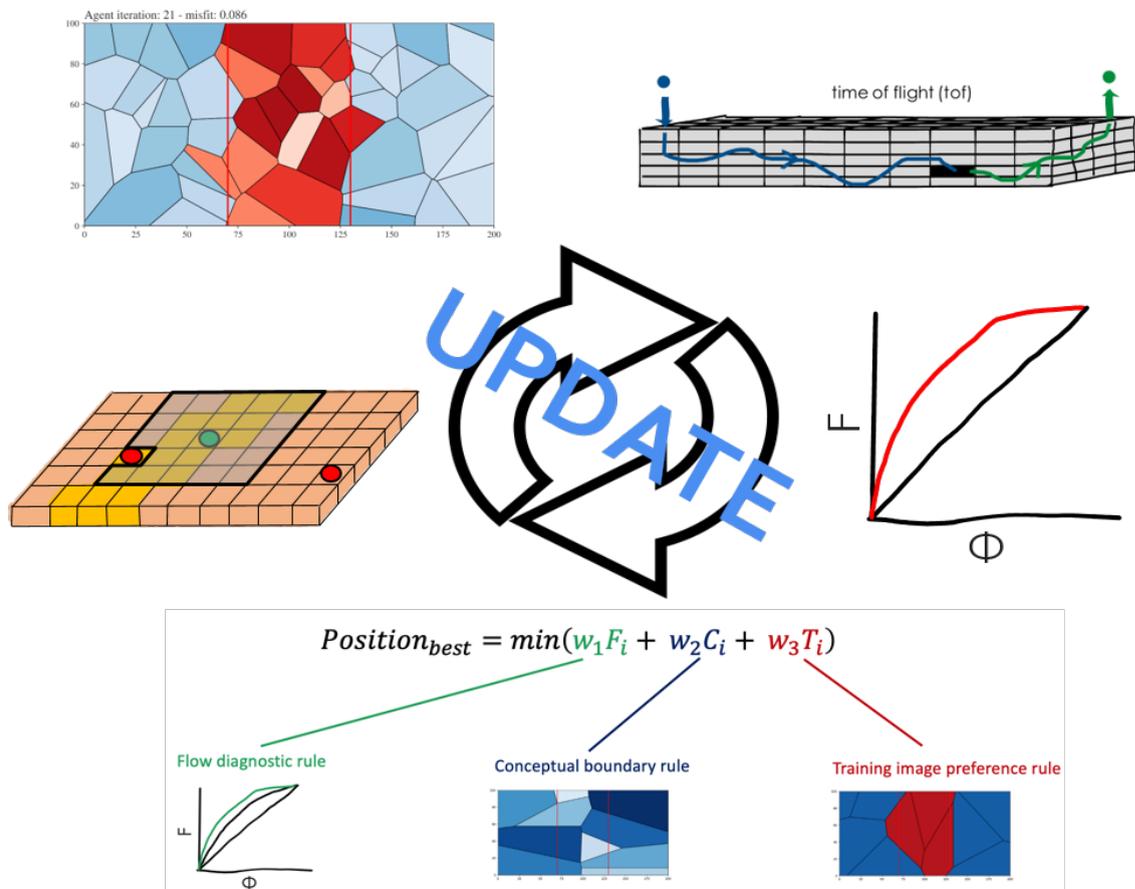


FIGURE 6.4: Overview of the iterative updating mechanism each agent undergoes each iteration. The stating point is from the middle-left: (i) check surrounding positions and available training images (ii) redo voronoi tessellation according to new agent position (iii) build reservoir model (iv) run flow diagnostics (v) determine best new position for agent based on a combination of agent-governing rules. Repeat for next agent.

200x100 cells. For the validation this setup is sufficient to describe the conceptual model as consisting of a red zone that extends from cell 70-130 on the x-axis and spans the entire y-axis (figure 6.5). The rest of the conceptual model consists of the blue zone. In this setup 25 agents will over 25 iterations try to optimise for one of the three rules.

### 6.3.3.1 Flow diagnostic rule

The metric to pay attention to for the flow diagnostic rule is the misfit between the predefined flow-capacity storage capacity diagram and the measured models achieved flow-capacity storage capacity diagram. In the top of figure 6.6 we can see that within the first iterations the ABM manages to reduce its misfit for every agent from around 0.23 to 0.08. The remaining iterations gradually decrease the misfit to a minimum of around 0.05 after 25 iterations. Increasing the number of agents and the number of iterations

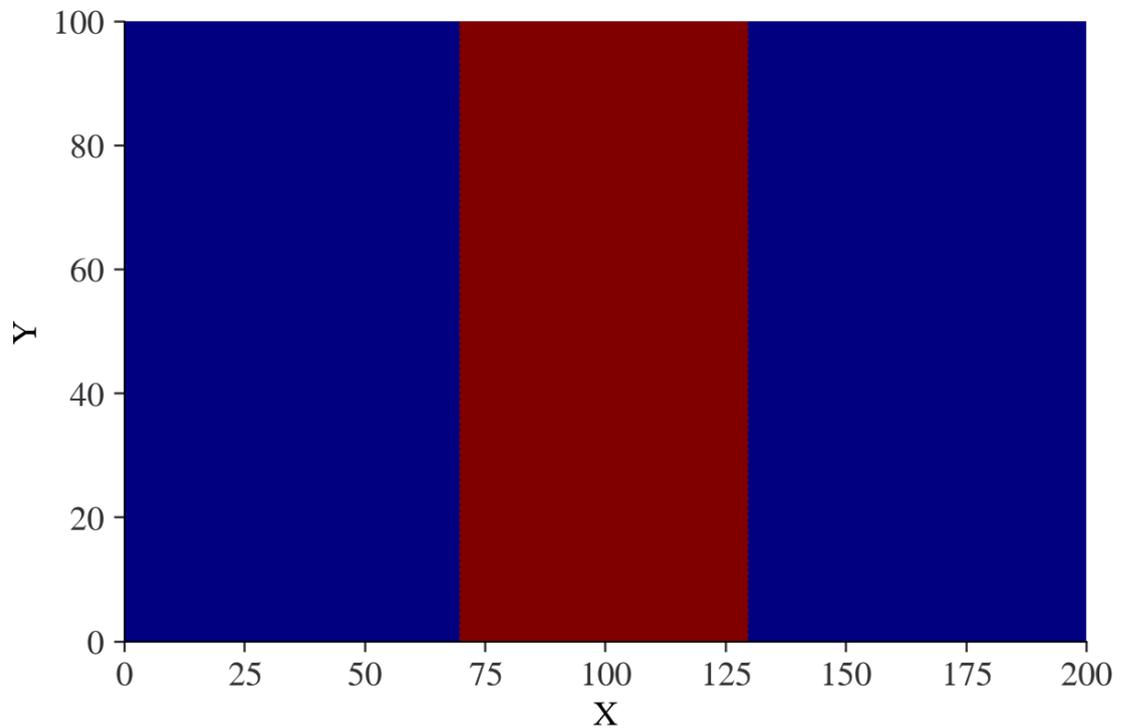


FIGURE 6.5: Conceptual model for validation purposes representing only a red and a blue zone.

would probably improve the overall development of the misfit to even lower levels. For validation purposes, this result is however satisfying and demonstrates the workings of the flow diagnostic rule.

### 6.3.3.2 Conceptual model boundary rule

For the conceptual model boundary rule it is best to look at the movement of the agents and the associated polygons. Over the iteration time the agents should move towards the border between both conceptual model zones (figure 6.4 a). As shown in figure 6.7, the agents and the surrounding polygons move towards the border between both conceptual models zones (horizontal red lines) throughout the iteration time. The conceptual model boundary rule is therefore also validated.

### 6.3.3.3 Training image preference rule

For the training image preference rule the conceptual model zone affiliation of each agent and the associated polygons is important. With the setting of the validation study, all

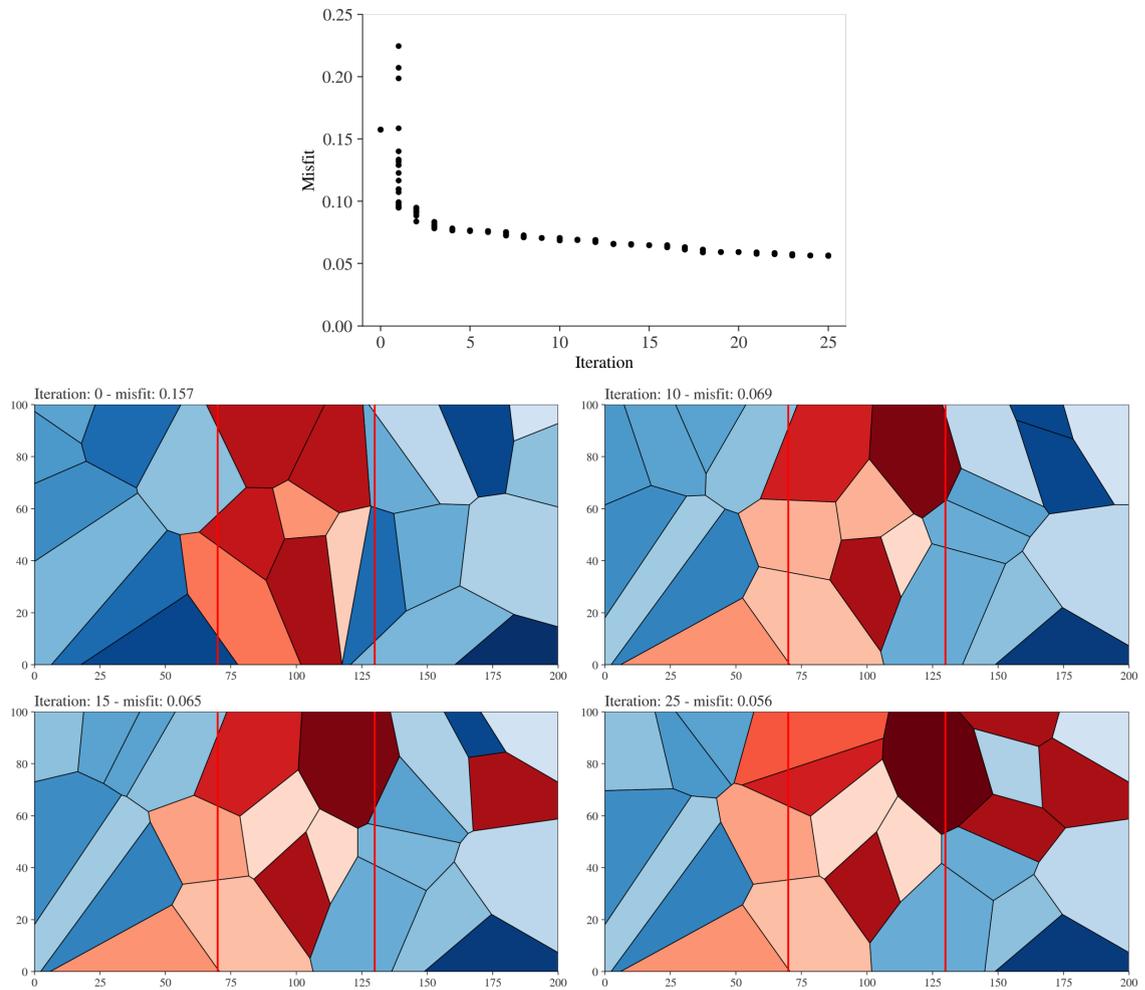


FIGURE 6.6: Top: Validation of the flow diagnostic rule. Over the iteration time of every individual agent the misfit to the set flow diagnostic objective gradually reduces for every agent. Bottom: Polygon distribution at iteration 0, 10, 15 and 25 is shown respectively. The different shades of blue for each polygon represent different training images from the blue zone. The red polygons represent different training images from the red zone. The red lines indicate the boundary between the red and the blue zone projected onto the environment. As only the flow diagnostic rule is considered in this case, agents do not honour the other rules in their decision-making process.

agents and polygons that are dominantly within the red zone (between red lines) should pick training images that are associated with the red zone. All agents and polygons that are dominantly within the blue zone should select a training image from the blue zone throughout the whole simulation time. As shown in figure 6.8, the simulation starts by randomly assigning a training image from either zone to the agents. Throughout the simulation the agents however quickly adapt to the underlying rules and only select training images from the zone that they are positioned in and therefore the training image preference rule is also validated.

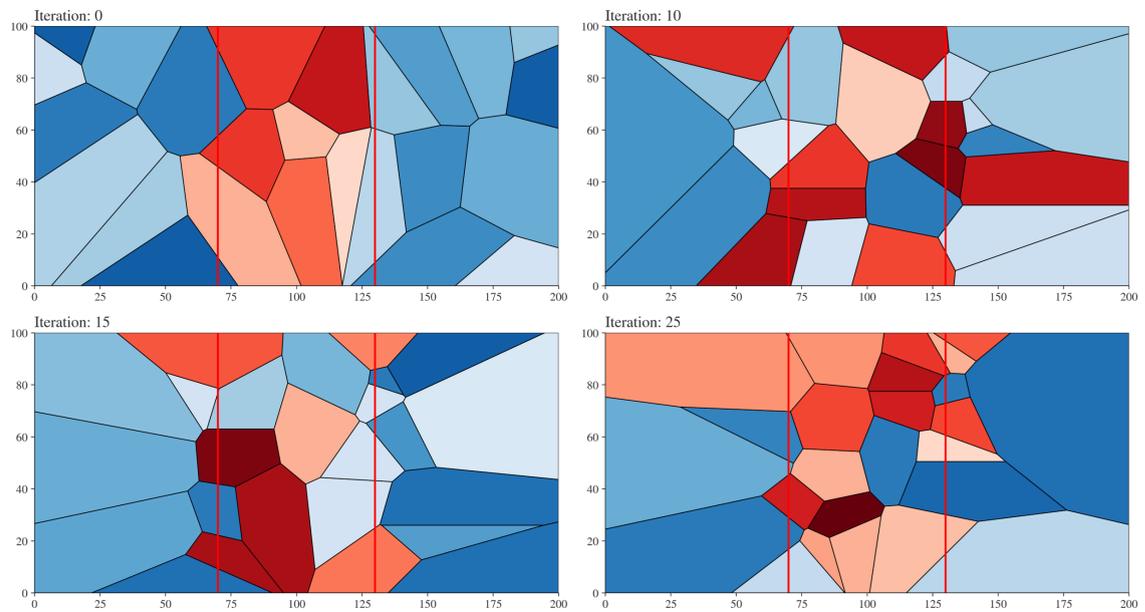


FIGURE 6.7: Validation of the conceptual model boundary rule. Polygon distribution at iteration 0, 10, 15 and 25 is shown respectively. The different shades of blue for each polygon represent different training images from the blue zone. The red polygons represent different training images from the red zone. The red lines indicate the boundary between the red and the blue zone projected onto the environment. As only the conceptual model boundary rule is considered in this case, agents agents and their surrounding polygons constantly move closer to the conceptual zone boundaries. The other rules are not honoured in the agents decision-making process.

## 6.4 Results

### 6.4.1 Application to synthetic case study

The ABM method developed and validated in the previous section is now tested on the synthetic case study that we described in chapter 2 section 2.4. The scenario here is the following: A naturally fractured reservoir is discovered and the geoscientists working the field come up with a geological conceptual model that is comparable with the Teton anticline model [Stearns, 1964] (figure 6.9 a). Based on this concept, a reservoir engineer comes up with an indication of the dynamic flow behaviour of the reservoir and tries to capture that in a flow capacity- storage capacity diagram (figure 6.9 b). The aim of the ABM now is to test the geological and the dynamic assumptions made above. Is it possible for the ABM to generate a reservoir model that satisfactorily fits both concepts? If not, is the ABM capable of highlighting areas in both concepts that are flawed and require reworking? The rest of the case study is set up as follows: First, a brief introduction to the

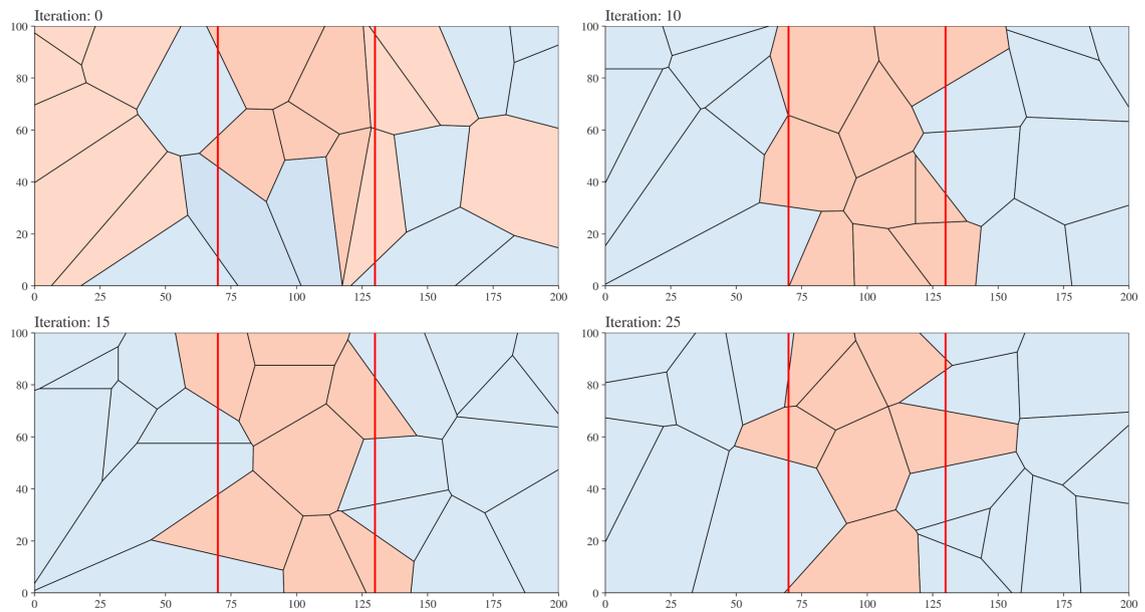


FIGURE 6.8: Validation of the training image preference rule. Polygon distribution at iteration 0, 10, 15 and 25 is shown respectively. Blue polygons represent training images from the blue zone (flank). Red polygons represent training images from the red zone (crest). The red lines indicate the boundary between the red and the blue zone projected onto the environment. As only the training image preference rule is considered the polygons that take up the area between the red lines pick the training images from the red zone. Polygons outside the red lines pick training images from the blue zone. The other rules are not honoured in the agents decision-making process.

geological conceptual model and the thoughts behind the flow capacity- storage capacity diagrams is given. In the second step an ABM setup to test out the concepts is introduced. After running the ABM, the new information are analysed to resolve if rethinking of the concepts is required.

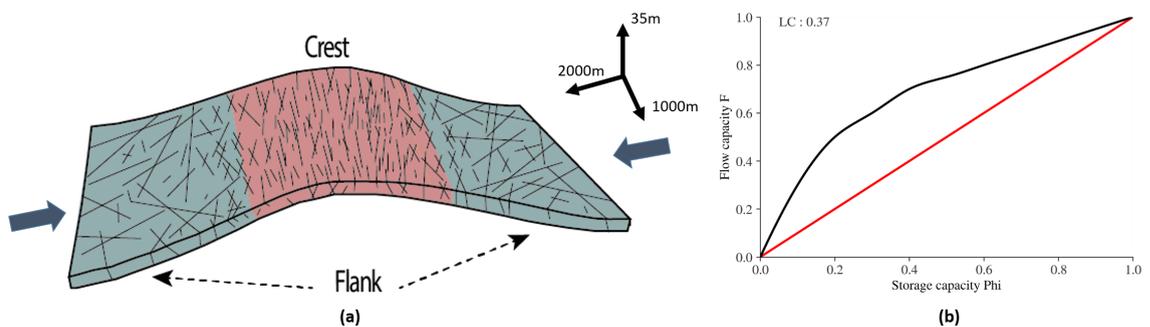


FIGURE 6.9: (a) Conceptual model based on Teton anticline model for fold related fracturing. In the red zone around the fold crest fractures are short and parallel to the fold axis. In the green zone fractures are longer, more dispersed and oblique to the fold axis. (b) Flow capacity - storage capacity diagram showing the expected flow behaviour of the reservoir (black line) with an expected Lorenz coefficient of 0.37. The red line indicates a case of homogeneous sweep and a Lorenz coefficient of 0.0.

## **6.4.2 Geological concept and expected dynamic flow behaviour**

As already stated above, the conceptual geological model for the reservoir is based on the Teton anticline model by Stearns [1964] and was already introduced in chapter 2. The model used in this study measures 2000m x 1000m x 35m and comprises low-quality carbonate rocks (low porosity and permeability) and seven mechanical units (each roughly five meters in height) with bed-bounding fractures that are separated by slip planes (figure 6.8 a). The model only consists of the horizontal-extension-dominated upper part of the Teton anticline model and shows a minimum of 2 distinctive folding related fracture patterns. In the fold crest, fractures are in alignment with the fold axis, are of short length but high intensity and low dispersion. The flank is characterised by longer reactivation fractures that are orientated perpendicular to the fold axis and show a higher degree of disorder and lower intensity.

Based on the information above, the assumption is made that a significant degree of fluid flow will occur within the fracture network, whilst the carbonate rock matrix will predominantly act as the hydrocarbon storage system and flow through the matrix will be of subordinate importance. This translates to a flow capacity - storage capacity diagram shown in figure 6.9 b and an expected dynamic Lorenz coefficient of 0.37.

## **6.4.3 ABM setup and objective**

As stated previously, in this work the ABM is set up to perform property modelling with the help of multiple-point statistics. Based on the geological conceptual model, we can expect that in a carbonate rock of low reservoir quality, the driving force behind fluid flow will be the presence or absence of fractures. We therefore generate training images that are based on a method developed in Steffens et al. [2019] and were already showcased in chapter 4. The uncertainty around the fracture patterns and parameters is captured by a catalogue of 20 different training images for the crestal area and the flanks. For every agent's training image selection procedure, it is essential to arrange the training images in a meaningful manner. Therefore training images are arranged by increasing fraction of rock type 1 (fracture dominated flow). The ABM simulation is run for a total of 50 iterations and initiated with 50 starting agents. For simplicity the spawning of new agents

and the removal of agents from the environment is not permitted. The number of positions each agent can test is set to 10 and the search radius for each agent is 10 grid cells. The ABM can investigate and focus on different objectives depending on the weighting given to the rules explained above. As this work is about trying to understand the validity of the conceptual model, the training image preference rule is weighted with 1.0, the flow diagnostic rule with 0.7 and the conceptual model boundary rule with 0.2. The simulation time for this set-up on a standard workstation will be approximately three days.

#### **6.4.4 ABM simulation results**

In the top of figure 6.10 we can see how the misfit to the expected flow diagnostic response developed over simulation time. Notice how at every iteration the misfit for all 50 active agents is plotted together. A maximum misfit of 0.36 is reached at the beginning of the simulation which gradually moves down towards a band-width of 0.10 to 0.20 within which it oscillate. The minimum misfit of 0.05 is achieved after 50 iterations. What we also observe are these irregular spikes in the misfit (.e.g iteration 14, 28 and 36), leading to intermediate higher misfits of up to 0.23. The reason for these spikes is the interaction between the different rules, in particular the training image preference rule. If the agents (polygons) surrounding the agent about to move gradually changed for example to a new majority training image, this can have as a consequence, that the training image currently used by the agents about to move is not in the majority anymore. It might therefore be out of favour and another training image will deliver a better overall match. This is not the case if the flow diagnostic rule is looked at on its own. However, these spikes are quickly compensated for by the other agents during their turns, trying to optimise their positioning.

The bottom half of figure 6.10 displays the distribution of agents in the environment and the training images assigned to each agent at different iterations. The different shades of blue for individual polygons indicate training images that belong to the flank training image catalogue. Red polygons indicate training images that are from the crestal training image catalogue. At the initiation at iteration 0 there is as expected no coherence among the training images, no movement of agents towards the boundaries, the initial conceptual model is still held up and a high misfit of 0.261. Already after 10 iterations we can see significant improvements with a misfit of 0.079, which is at the lower end of misfits

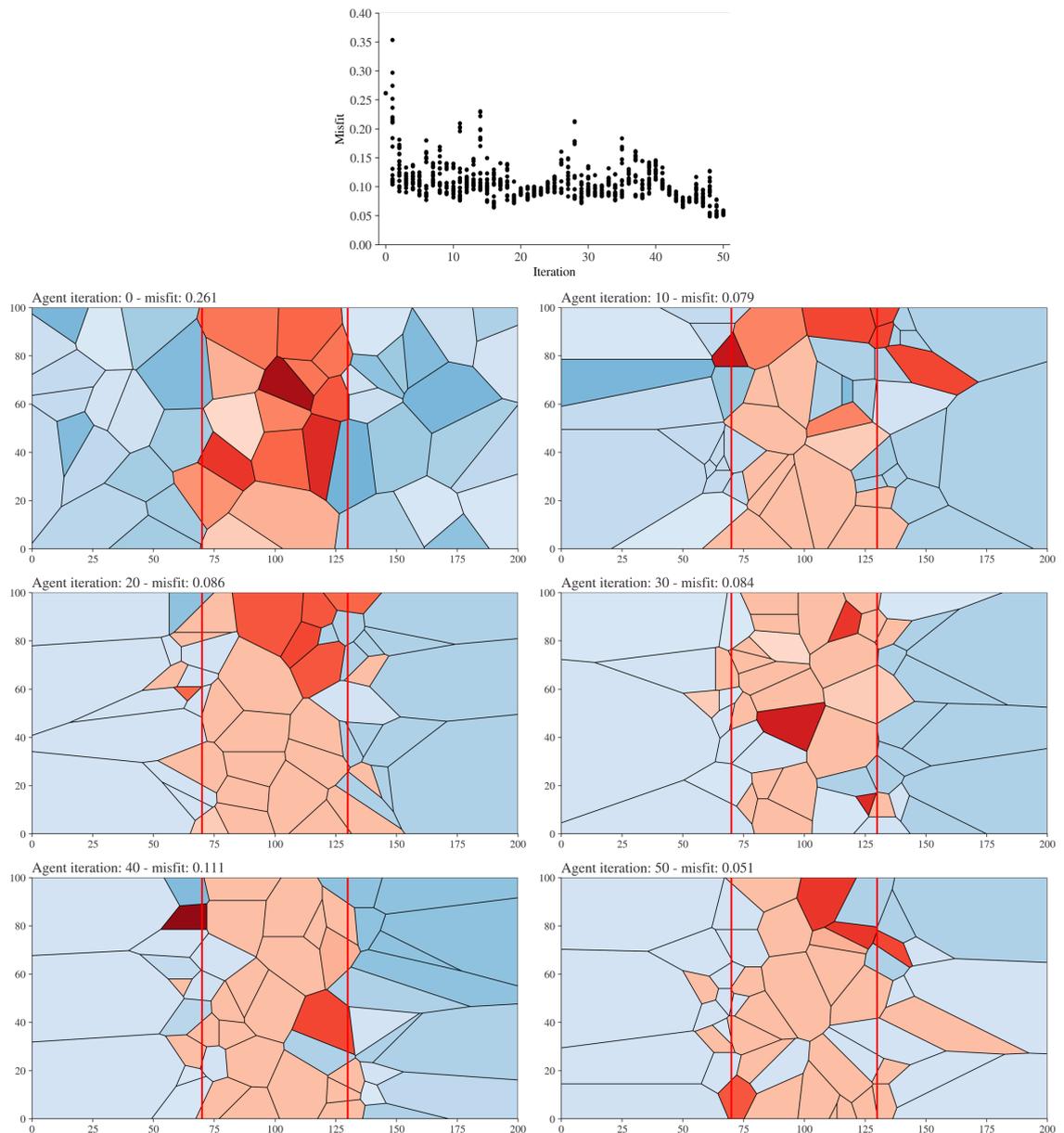


FIGURE 6.10: Top: Misfit of the model to the flow diagnostic objective throughout iteration time. Notice how at every iteration the misfit for all 50 active agents is plotted. Bottom: Distribution of agents in the environment at different stages of the simulation. The different shades of blue for each polygon represent different training images from the flank training image catalogue. The red polygons represent different training images from the crestal training image catalogue. The red lines indicate the boundary between the flank and the conceptual model's crestal zone projected onto the environment.

achieved throughout simulation time. Agents are moving towards the boundary zones (red lines) between the crestal and the flank zones and training images on the right flank and the crest gradually start becoming more uniform. The lack of coherence will however lead to low quality scores for the majority of agents. Many agents (polygons) represent a different training image - despite being from the correct training image catalogue. As this

is heavily penalised by the ABM it will interchange the incoherent training images with more coherent training images and force the agents to adapt throughout simulation time, fully unfolding all agent rules. Another interesting observation is that we are starting to see these incisions within the crestal zone (between red lines), where we now have flank training images (blue polygons) replacing crestal training images. Throughout the next iterations, the training image distribution becomes more and more coherent, whilst the misfit manages to remain at low levels. The observation regarding incisions within the crestal zone, where agents replace crestal training images (red) with flank training images (blue) also continues. This development could indicate three things. The selection of training images within each catalogue is not adequate and the prior assumptions made by the geoscientist do not fully reflect the uncertainties around fold related fracturing correctly. The assumptions made by the reservoir engineer about the dynamic flow behaviour are incorrect and require further refinement. It could also be that the conceptual geological model itself is correct, but the proportions of the different zones are not chosen correctly and require a reassessment.

## **6.5 Discussion**

The adaptation of agent-based modelling is happening in many industries but has yet to play a significant role in the subsurface [Steffens et al., 2022]. In this chapter, we gave an example of how agent-based modelling can streamline and improve the testing of conceptual geological models and integrate static, conceptual and dynamic data into the validation process that autonomous agents carry out. The idea was to develop a tool that complements a geoscientist's task of creating a geological conceptual model and provides for immediate refinement or complete reassessment and remodelling of the conceptual model and the prior assumptions about property distributions or the dynamic response. The ABM presented here integrates conceptual and static data alongside dynamic data to verify if the underlying geological conceptual assumptions are plausible or not. Instead of following a trial and error approach, adding and weighting different rules that dictate the agents' movement, we can screen a conceptual model from various angles. As already emphasised in chapter 5, flow diagnostics are a fundamental building block in this ABM framework. Here, redoing the Voronoi tessellation, building a reservoir model and running

flow diagnostics and evaluating the agents best position runs within seconds for each tested position. In the case described above, that equates to less than a minute per agent per iteration on a standard workstation. This allows for the ABM to quickly detect flaws within the geological concept and channel further resources towards improving the conceptual model. In the synthetic case study for example we could see that initially the crestal area of the fold was possibly outlined too wide and narrowing the zones width could improve the conceptual models fit with the suggested dynamic response, as shown in figure 6.11. Note that this is only one of many possible scenarios for how the conceptual model could look like. It could also be that the training images we selected did not fully cover the uncertainties that are related to the fracture distributions in the flank and the crest. Or perhaps our assumptions about the dynamic behaviour were incorrect and need refinement. Fractures in nature behave in a highly nonlinear fashion and complex non-stationary distributions are possible. For example, these incisions we observed in the crest and flank could be indications for fracture corridors as shown in figure 6.11. With conventional geostatistical methods modelling this with trends would be challenging and it would be difficult to identify the potential for the occurrence of phenomena that are very difficult to predict, such as fracture corridors [Bisdom et al., 2014, de Jossineau and Petit, 2021, Questiaux et al., 2010]. The independent interactions of individual agents that we have in ABMs however, where agents constantly try to get into the best possible state for themselves, will constantly lead to the development of emergent behaviours [DeAngelis and Diaz, 2019]. Therefore, illuminating the potential for cases like fracture corridors is one of the core strength of the ABM approach.

Furthermore, the ABM also allows for the testing and setup of multiple conceptual scenarios. By running a full simulation with the same setting for various scenarios, the geological uncertainty can be reduced by promoting and further enhancing an ensemble of well-matching scenarios and discarding unsuitable scenarios. The outcomes of the proposed method can then be used to either build or update reservoir models in a more accurate and dynamic and static data-driven way. The benefits of following a data-driven route rather than a manual trial and error approach when it comes to, for example selecting training images for MPS or selecting different parameters to match static data with dynamic observations, have been highlighted in Caeiro et al. [2014].

Thanks to how ABMs are set up, this tool is flexible in its application and can easily be adapted to a modified task. The set of predefined rules guides each agents decision-making process. Adding additional rules, exchanging the rules or modifying the environment to change an agents behaviour only requires minor adjustments to the code. Therefore, increasing the ABMs complexity to make the outcomes more realistic whilst having a comparable simulation run-time is easily achievable. It is also possible to frame the initial question that the ABM is supposed to answer differently and for example increase the agents' focus on matching the dynamic response. As mentioned beforehand, the aim of the ABM lies in supporting a geoscientists decision process and interpretations in a data-driven way. The geoscientist can therefore be comfortable with testing out new ideas and interpretations, whilst making sure they conform with the known data and assumptions made. We tested the ABM approach on a simple linear example, but in reality, structural trends can be more complex, following geological features such as clinoforms or pinch-outs in three dimensions. This is where a non-parametric, adaptive method has big advantages to classic parametric models.

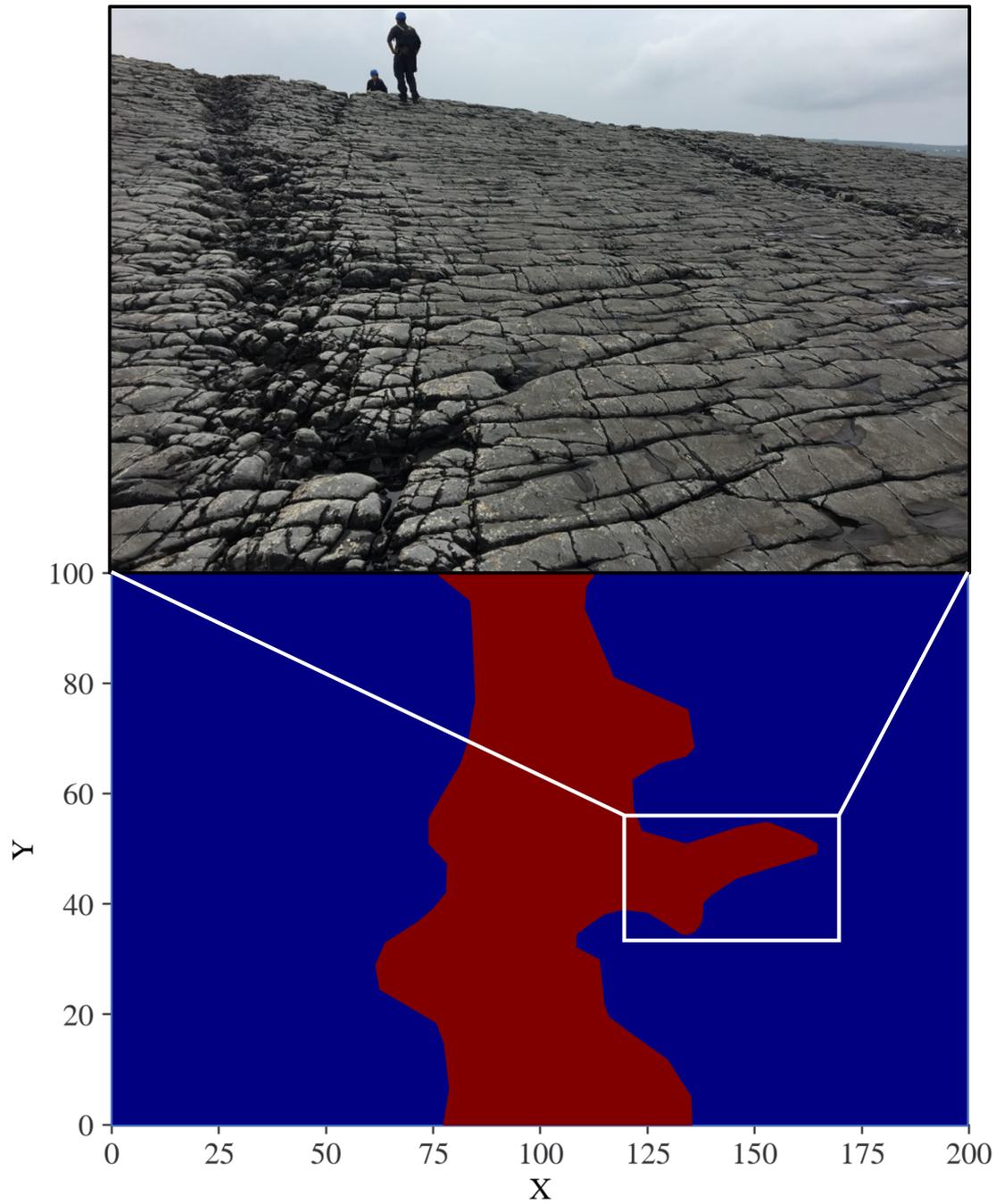


FIGURE 6.11: Top: Example for a fracture corridor in carbonate rocks (Ross formation, County Clare, Ireland). Bottom: Updated conceptual model with potential fracture corridor on right flank.

# Chapter 7

## Summary, conclusion and future work

### 7.1 Summary

Effectively moving ahead with the energy transition will require us to understand the workings of naturally fractured rocks and reservoirs. They can often be the host rock for hydrocarbons or geothermal systems, potential storage sites for carbon dioxide sequestration, and a reliable groundwater source. Each of the above-described areas is associated with heavy upfront capital expenditures and decision-making under uncertainty. If carried out wrong, uninformed judgments might have severe consequences for future generations. To avoid this from happening, we must understand the subsurface, minimise risk, and make informed judgments in the face of uncertainty.

The cross-disciplinary nature of characterising, modelling and simulating flow through naturally fractured reservoirs whilst only working with scarce data on fracture distributions etc., makes efficient integration and preservation of all data vital. Generating geologically consistent reservoir models that acknowledge the complexity of simulating flow through systems with strong permeability and scale contrasts (matrix flow vs fracture flow) whilst also considering the need for an ensemble of reservoir models to cover geological uncertainties and their impact on fluid flow is fundamental for decision-making.

This thesis aimed to develop cross-disciplinary workflows for reservoir modelling of naturally fractured reservoirs that allow for the integration of geological uncertainties

and dynamic data into the modelling procedure, promote ensemble modelling, and bring the disciplines involved in the modelling of naturally fractured reservoirs closer together. We therefore started setting the scene by introducing the concept of naturally fractured reservoirs and generated a synthetic case study to test the workflows (figure 7.1 top left). This was followed by a broad overview of methods from different fields such as data science, information theory and heuristic optimisation that will be utilised within the workflows developed in the remainder of the thesis. A characteristic of the developed workflows is that they become less parametric and constrained by user inputs and more driven by the underlying data with every chapter. In chapter 4 we developed a method that uses training images generated from upscaled discrete fracture networks to distribute natural fractures on reservoir scale quickly and efficiently whilst also considering geological realism of the structural geology imprint with the help of multiple-point statistics (figure 7.1 right side). To acknowledge the impact that fracture-related geological uncertainties have on fluid flow, we used flow diagnostics to screen and then cluster and select training images according to their flow response. The generated training images then served as major building blocks in workflows developed subsequently. The first set of workflows developed in chapter 5 generate models to an expected dynamic flow response and underlying geological prior information (figure 7.1 bottom centre). The aim here was to generate a diverse ensemble of reservoir models with an optimiser's help and use entropy as a measure of diversity that controls the optimiser's model development. We also showcase how this ensemble of models can be utilised to generate decision-making maps for example, for potential infill wells targets. In chapter 6 we take things a step further and propose an agent-based modelling methodology, where within a reservoir model, independent but interacting agents follow a set of rules to generate reservoir models that take into account geological prior information and expected dynamic flow responses (figure 7.1 bottom left). We used the methodology to refine an underlying prior conceptual geological model in the case shown here. The nature and setup of the agent-based model would also allow for us to investigate various other areas of interest with only minor modifications to the existing codebase.

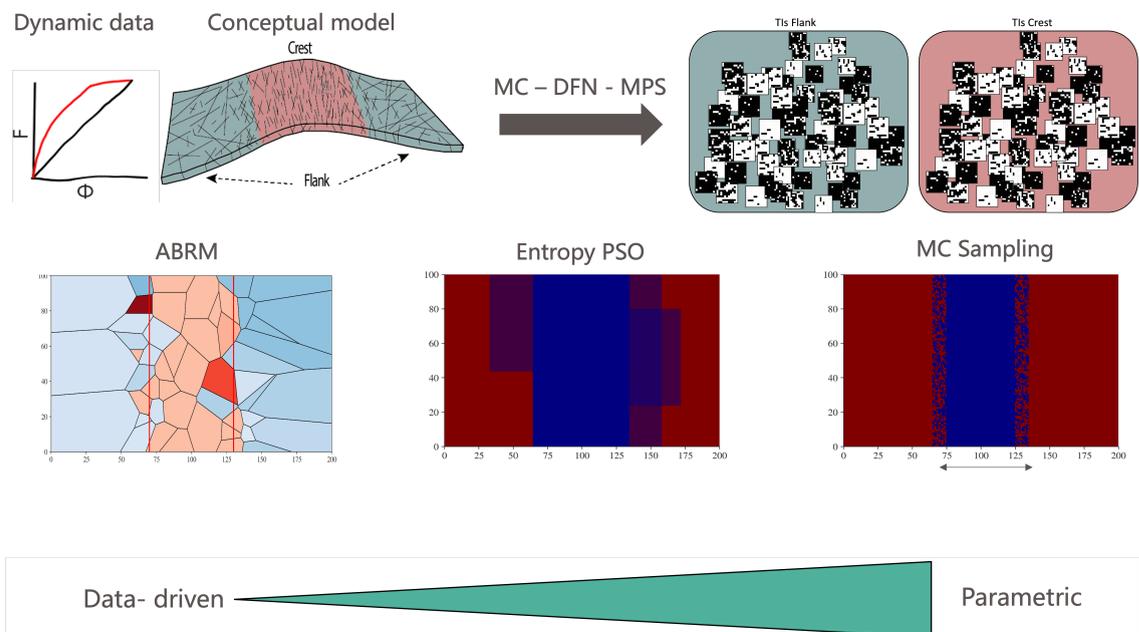


FIGURE 7.1: Conceptual summary of the workflows developed in this thesis. With the help of either conceptual geological information or dynamic data and conceptual geological information a suite of training images is generated that can be used within three workflows. The workflows move from being parametric (Monte-Carlo Sampling) to more data-driven approaches (Entropy PSO and Agent-Based-Reservoir Modelling). It should also be highlighted that the complexity of setting up these workflows increases with the more data driven approaches, therefore making the selection of the right workflow to utilize dependent on a projects' objective.

## 7.2 Conclusions

Based on the proposed workflows and the results presented in preceding chapters, the following conclusions can be drawn from this thesis:

1. With the help of training images generated for multiple-point statistics, fracture-related geological uncertainties and their impact on fluid flow can be propagated throughout an entire modelling workflow for fractured reservoirs fast and efficiently. By initially setting up a Monte-Carlo sampling approach for training image generation, we cover the impact of fracture-related uncertainties on fluid flow. Screening each training image on its dynamic flow behaviour with the help of flow diagnostics and clustering training images with a similar flow response allows us to only take forward a subset of training images that cover a comparable range of geological uncertainty and its impact on fluid flow. This subset of training images can then be used to generate a set of reservoir models that effectively capture fracture-related

uncertainties and their effects on reservoir scale fluid flow. Using multiple-point statistics instead of discrete fracture network modelling on reservoir scale allows the construction of reservoir models within a fraction of the time, making it suitable for practitioners to capture fracture-related uncertainties quickly.

2. The advantage of workflows that allow for the generation of dynamically driven and geologically constrained ensembles of reservoir models over conventional reservoir modelling workflows is that we control the dynamic behaviour of the generated reservoir models from the beginning. Either generating a series of models that all get swept in the same way but are based on different assumptions about the underlying geology, or building models based on the same underlying geology but get swept in different ways only depends on the setup of the workflow proposed in chapter 5. By integrating measures for reservoir model diversity, such as the time-of-flight-based entropy, we can ensure that the optimisers used within the workflow explore novel, undiscovered models that also fit with the initial outset objective and allow for the generation of an ensemble of models that most adequately represent all possibilities to be robust towards subsurface development risk. The time-of-flight-based entropy is also a valuable tool to generate risk maps for decision making based upon the information given from a large set of reservoir models and visualise spatial uncertainty.
3. Where conventional geostatistical methods struggle and quickly become overly parametric, agent-based modelling can represent an attractive modelling alternative. Agent-agent and agent-environment interactions can lead to evolution of emergent behaviours that would be hard for geostatistical methods to capture. This lends itself well to geological features such as highly nonlinear and non-stationary fracture distributions. The agent-based model developed in this thesis can integrate conceptual geological data alongside static and dynamic data and, for example, test if these data are aligned with assumptions made about the geological model. This allows it to be used as a data-driven tool to support geoscientists with their decision processes and interpretations. The agent-based-model setup makes it a tool that can be easily adapted to a modified task by adding or removing rules that guide each agent's decision-making process with only minor adjustments to the codebase.

4. As the level of implementation complexity varies between the developed workflows, the decision for which of the workflows to use depends on a projects' objective as well as the availability of data, time and confidence in the own understanding of the reservoir. If there is little uncertainty regarding the conceptual geological model, it is sufficient to work with the MPS workflow with Monte-Carlo sampling described in chapter 4. It is the most parametric workflow, with the majority of uncertainties being predefined in a parametric way. It is also the simplest workflow to implement. If it is deemed important to (i) include dynamic data into the modelling procedure and (ii) allow for an overall higher degree of freedom for model building, the entropy-PSO introduced in chapter 5 is the most suitable approach. The largest degree of freedom as well as the most data-driven technique is the agent-based-reservoir-modelling (ABRM) approach presented in chapter 6. This method requires a significant amount of upfront work to be setup correctly, but it also is the workflow most detached from any disadvantages that come with conventional geostatistical modelling techniques. It is therefore suitable to challenge existing geological concepts quantitatively and stimulate the subsurface teams' creativity.
5. Overall, taking advantage of established approaches from various disciplines and bundling them into cross-disciplinary workflows provides a lot of potential for the subsurface domain. For example, using flow diagnostics instead of full physics flow simulations to quickly screen a reservoir model's dynamic response permits generating and screening thousands of models. Combining these with unsupervised learning and clustering approaches or agent-based modelling are only two of many potential applications that arise from cross-disciplinary thinking that will shape the future role that the subsurface will play within the energy transition.

### **7.3 Recommendations for future work**

Several options to improve or extend the research discussed in this thesis exist. Some only require using different testing data sets or swapping out components, whereas others require further codebase development. Some potential scenarios for future research are:

1. The first and most apparent extension to this work would be to test the workflows developed here on actual field data from a naturally fractured reservoir with a team of subsurface experts. Real fields are (i) substantially more complex and (ii) often only have little constraining data available. It would for example be interesting to document how a field before water-breakthrough with significantly higher uncertainty (e.g. waterfront location) would impact the development of the time-of-flight-based entropy and the associated risk maps. Applying the methods to real field data would also allow for better verification of the results and conclusions made in this thesis and highlight the true value of data-driven, cross-disciplinary workflows.
2. Further improvements to this work would be to replace the single-porosity flow diagnostics methods utilised here with the method developed by Spooner et al. [2021]. By introducing a retardation factor into their calculations, they take into account matrix-fracture interactions and their impact on water breakthrough and extend flow diagnostics to dual-porosity systems. This would however also require the consideration of additional complexity that is added to several steps performed throughout this work. For example, instead of upscaling fractures and matrix to a single-porosity continuum, dual-porosity upscaling would be required. This again requires a modification to the current training image generation process that only takes into account two different rock types and does not account for a dual-porosity descriptions of the system. To increase the geological realism, it could also be considered to replace or assist the DFN upscaling step for training image generation with computationally more costly Embedded Discrete Fracture Models (EDFM) as proposed by Wong [2019]. In this hybrid approach thresholds can be defined that differentiate between the fractures that can be represented implicitly as continuous media and the fractures that have to be represented explicitly to not overly abstract flow dynamics. The selection of an adequate threshold is however a subjective choice defined by the modeller. This introduces further uncertainties that would need to be investigated. Additionally, the threshold selection is often based upon DFM model simulation that require substantial upfront time to develop and simulate. Managing hundreds of these efficiently to seamlessly integrate with

the existing modelling workflows described in this work, would also significantly increase workflow complexity. An advantage of the EDFM method is that it is compatible with conventional reservoir simulators such as tNavigator or Eclipse, lowering the barrier for integration into the workflows developed in this thesis, once the calibration phase is successfully designed.

3. We also mentioned several potential improvements that can either be derived from or integrated relatively easy into the existing entropy-guided modelling workflows of chapter 5. More sophisticated sampling approaches can be applied instead of generating a subset of models by randomly sampling from the clusters developed in both workflows and assuming equal probability. We could take into account attributes such as relative cluster size or the discrepancy between observed and simulated behaviours of each model. In its current setup, the entropy-PSO initiates as a single swarm. To prevent potential under-exploration of some areas, one could consider multiple starting populations or a subdivision of the swarm into several sub-swarms that explore the parameter space more independently. Additional use cases for entropy within the workflows of chapter 5 are also possible. Monitoring how the spatial time-of-flight-based entropy maps develop over the PSOs simulation time would allow us to identify how we get new information in specific areas with each iteration of the PSO and how this is linked with the exploration of the parameter space. When calculating the system's entropy, we could also use a moving window or cube to calculate how entropy develops to get a better spatial component into the system's entropy.
4. The most significant potential for further improvements lies with the agent-based modelling (ABM) approach developed in chapter 6 as to date there have only been few applications of ABM in the subsurface domain. Integrating seismic attributes as an environment for the agents to move around, complimenting a predefined conceptual model would allow additional data integration. An additional rule that could easily be implemented to impact the agents' decision-making would be considering the size of each agent's Voronoi polygon as a weighting factor. Additionally, agents could be preferentially placed in areas where a higher resolution is required, for example, inferred from seismic data. Also, giving agents the capability to evolve

and learn from their behaviour and the group of agents they interact with could be an interesting addition [Wilensky and Rand, 2015]. This could lead to a scenario where several agents start competing against each other. *Geological Agents* could try to update the underlying geological model to the given dynamic and static data, whilst *Engineering Agents* can move around wells or modify production rates to decrease the match to the reservoir model, eventually leading to an equilibrium state between both agent groups. Another idea would be to make use of entropy within this modelling approach to (i) highlight zones within a model with the most changes or uncertainty throughout simulation time, leading to the attraction of additional agents and (ii) start numerous ABMs in parallel and use entropy to ensure diversity among the ABM runs. Finally, one could test out the ABM approach on different reservoir types and different modelling techniques. For example, as proposed by Arnold [2008], to update channel geometries or add and remove channels within a fluvial reservoir setting modelled with object-based methods.

# References

- M. Abdel-Basset, L. Abdel-Fatah, and A. K. Sangaiah. Metaheuristic algorithms: A comprehensive review. In *Intelligent Data-Centric Systems, Computational Intelligence for Multimedia Big Data on the Cloud with Engineering Applications*, pages 185–231. Academic Press, 2018.
- A. Abraham, H. Guo, and H. Liu. Swarm intelligence: Foundations, perspectives and applications. *Studies in Computational Intelligence*, 26:3–25, 2006.
- R. Aguilera. Well Test Analysis of Naturally Fractured Reservoirs. *SPE Formation Evaluation*, 2:239–252, 1987.
- M. Ahmed Elfeel and S. Geiger. Static and Dynamic Assessment of DFN Permeability Upscaling. *Proceedings of SPE Europe/EAGE Annual Conference*, (June), 2012.
- E. Aliverti, M. Biron, A. Francesconi, D. Mattiello, S. Nardon, and C. Peduzzi. Data analysis, processing and 3D fracture network simulation at wellbore scale for fractured reservoir description. *Geological Society Special Publication*, 209(1):27–37, 2003.
- M. Allaoui, M. L. Kherfi, and A. Cheriet. Considerably Improving Clustering Algorithms Using UMAP Dimensionality Reduction Technique: A Comparative Study. In A. El Moataz, D. Mammass, A. Mansouri, and F. Nouboud, editors, *Image and Signal Processing. ICISP 2020. Lecture Notes in Computer Science*, pages 317–325. Springer International Publishing, 2020.
- J. Allen and Q. S. Sun. Controls on Recovery Factor in Fractured Reservoirs: Lessons Learned from 100 Fractured Fields. In *Proceedings of SPE Annual Technical Conference and Exhibition*. OnePetro, 2003.

- D. Arnold, V. Demyanov, M. Christie, A. Bakay, and K. Gopa. Optimisation of decision making under uncertainty throughout field lifetime: A fractured reservoir example. *Computers & Geosciences*, 95:123–139, 2016.
- D. P. Arnold. *Geological Parameterisation of Petroleum Reservoir Models for Improved Uncertainty Quantification*. PhD thesis, Heriot-Watt University, 2008.
- G. B. Arpat and J. Caers. A Multiple-scale, Pattern-based Approach to Sequential Simulation. In *Geostatistics Banff 2004*, number 1, pages 255–264. 2005.
- W. B. Arthur. Foundations of complexity economics. *Nature Reviews Physics*, 3(2): 136–145, 2021.
- F. Azam, M. Sharif, and S. Mohsin. Multi Agent Based Model for Earthquake Intensity Prediction. *Journal of Computational and Theoretical Nanoscience*, 12(12):5765–5777, 2015.
- A. Baghbanan and L. Jing. Stress effects on permeability in a fractured rock mass with correlated fracture length and aperture. *International Journal of Rock Mechanics and Mining Sciences*, 45(8):1320–1334, 2008.
- G. I. Barenblatt, I. P. Zheltov, and I. N. Kochina. Basic concepts in the theory of seepage of homogeneous liquids in fissured rocks [strata]. *Journal of Applied Mathematics and Mechanics*, 24(5):1286–1303, 1960.
- N. Barton, S. Bandis, and K. Bakhtar. Strength, deformation and conductivity coupling of rock joints. *International Journal of Rock Mechanics and Mining Sciences*, 22(3): 121–140, 1985.
- J. Bear. *Dynamics of fluids in porous media*. Courier corporation, 1972.
- R. Bellman. Dynamic Programming. *Science*, 153(3731):34–37, 1966.
- M. Bentley and S. Smith. Scenario-based reservoir modelling: The need for more determinism and less anchoring. *Geological Society Special Publication*, 309:145–159, 2008. ISSN 03058719. doi: 10.1144/SP309.11.
- B. Berkowitz. Analysis of fracture network connectivity using percolation theory. *Mathematical geology*, 27(4):467–483, 1995.

- B. Berkowitz. Characterizing flow and transport in fractured geological media: A review. *Advances in Water Resources*, 25(8-12):861–884, 2002.
- I. Berre, F. Doster, and E. Keilegavlen. Flow in fractured porous media: A review of conceptual models and discretization approaches. *Transport in Porous Media*, 130: 215–236, 2019.
- F. Bianchi and F. Squazzoni. Agent-based models in sociology. *Wiley Interdisciplinary Reviews: Computational Statistics*, 7:284–306, 2015.
- L. Bianchi, M. Dorigo, L. M. Gambardella, and W. J. Gutjahr. A survey on metaheuristics for stochastic combinatorial optimization. *Natural Computing*, 8(2):239–287, 2009.
- K. Bisdom. *Burial-related fracturing in sub-horizontal and folded reservoirs. Geometry, geomechanics and impact on permeability*. PhD thesis, Delft University, 2016.
- K. Bisdom, B. D. Gauthier, G. Bertotti, and N. J. Hardebol. Calibrating discrete fracture-network models with a carbonate three-dimensional outcrop fracture network: Implications for naturally fractured reservoir modeling. *AAPG Bulletin*, 98(7):1351–1376, 2014.
- K. Bisdom, H. M. Nick, and G. Bertotti. An integrated workflow for stress and flow modelling using outcrop-derived discrete fracture networks. *Computers and Geosciences*, 103:21–35, 2017.
- T. Blackwell and P. Bentley. Dynamic Search With Charged Swarms. In *Proceedings of the 4th annual conference on genetic and evolutionary computation*, pages 19–26, 2002.
- T. Blackwell and J. Branke. Multiswarms, Exclusion, and Anti-Convergence in Dynamic Environments. *IEEE Transactions on Evolutionary Computation*, 10(4):459–472, 2006.
- T. M. Blackwell. Particle swarms and population diversity. *Soft Computing*, 9(11): 793–802, 2005.
- E. Bonabeau. Agent-based modeling: Methods and techniques for simulating human systems. *PNAS*, 99(3):7280–7287, 2002.
- E. Bonabeau, G. Theraulaz, and M. Dorigo. *Swarm intelligence: from natural to artificial systems*. Oxford university press, 1999.

- E. Bonnet, O. Bour, N. E. Odling, P. Davy, I. Main, P. Cowie, and B. Berkowitz. Scaling of Fracture Systems in Geological Media. *Reviews of Geophysics*, 39(3):347–383, 2001.
- G. P. Boswell, N. F. Britton, and N. R. Franks. Habitat fragmentation, percolation theory and the conservation of a keystone species. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 265(1409):1921–1925, 1998.
- B. Bourbiaux. Fractured Reservoir Simulation: a Challenging and Rewarding Issue. *Oil and Gas Science and Technology*, 65(2):227–238, 2010.
- T. Bratton, D. V. Canh, N. Van Que, N. V. Duc, P. Gillespie, D. Hunt, B. Li, R. Marcinew, S. Ray, B. Montaron, R. Nelson, D. Schoderbek, and L. Sonneland. The nature of naturally fractured reservoirs. *Oilfield Review*, 18(2):4–23, 2006.
- R. Brits, A. Engelbrecht, and F. Van Den Bergh. A niching particle swarm optimiser. In *Proceedings of the 4th Asia-Pacific conference on simulated evolution and learning*, 2: 692–696, 2002.
- S. R. Broadbent and J. M. Hammersley. Percolation processes. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 53, pages 629–641. Cambridge university press, 1956.
- P. O. Bruna, J. Straubhaar, R. Prabhakaran, G. Bertotti, K. Bisdom, G. Mariethoz, and M. Meda. A new methodology to train fracture network simulation using multiple-point statistics. *Solid Earth*, 10:537–559, 2019.
- N. E. Brunk and R. Twarock. Percolation Theory Reveals Biophysical Properties of Virus-like Particles. *ACS Nano*, 15(8):12988–12995, 2021.
- S. V. Buldyrev, R. Parshani, G. Paul, H. E. Stanley, and S. Havlin. Catastrophic cascade of failures in interdependent networks. *Nature*, 464(7291):1025–1028, 2010.
- C. M. Burberry, D. L. Cannon, J. W. Cosgrove, and T. Engelder. Fracture patterns associated with the evolution of the Teton anticline , Sawtooth Range , Montana , USA. *Geological Society, London, Special Publications*, 487(1):229–261, 2019.

- M. H. Caeiro, A. Soares, V. Demyanov, and M. Christie. Optimized History Matching with Stochastic Image Transforming of a Deltaic Reservoir. In *Mathematics of Planet Earth*, pages 571–574. Springer Berlin Heidelberg, 2014.
- J. Caers and T. Zhang. Multiple-point geostatistics : a quantitative vehicle for integrating geologic analogs into Multiple Reservoir Models. 2002.
- J. S. Caine, J. P. Evans, and C. B. Forster. Fault zone architecture and permeability structure. *Geology*, 24(11):1025–1028, 1996.
- R. J. G. B. Campello, D. Moulavi, and J. Sander. Density-Based Clustering Based on Hierarchical Density Estimates. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 160–172, 2013.
- J. N. Carter, P. J. Ballester, Z. Tavassoli, and P. R. King. Our calibrated model has poor predictive value: An example from the petroleum industry. *Reliability Engineering and System Safety*, 91:1373–1381, 2006.
- J. L. Casti. Seeing the Light at El Farol: A Look at the Most Important Problem in Complex Systems Theory. *Complexity*, 1(5):7–10, 1996.
- J. Castro, S. Drews, F. Exadaktylos, J. Foramitti, F. Klein, T. Konc, I. Savin, and J. van den Bergh. A review of agent-based modeling of climate-energy policy. *WIREs Climate Change*, 11:1–26, 2020.
- M. Cavazzuti. *Optimization Methods: From Theory to Design - Scientific and Technological Aspects in Mechanics*. Springer, 2013.
- J. Cervelle and E. Formenti. Algorithmic Complexity and Cellular Automata. In *Mathematical Theory of Cellular Automata, Encyclopedia of Complexity and System Sciences*. Springer-Verlag, 2008.
- A. Chandna and S. Srinivasan. Modeling natural fracture networks using improved geostatistical inferences. *Energy Procedia*, 158:6073–6078, 2019.
- B. C. Chang, A. Ratnaweera, S. K. Halgamuge, and H. C. Watson. Particle swarm optimisation for protein Motif discovery. *Genetic Programming and Evolvable Machines*, 5(2):203–214, 2004.

- L. Cheng-Haw, L. Chen-Chang, and L. Bih-Shan. The estimation of dispersion behavior in discrete fractured networks of andesite in Lan-Yu Island, Taiwan. *Environmental Geology*, 52:1297–1306, 2007.
- C. A. Coello Coello and M. S. Lechuga. MOPSO: A proposal for multiple objective particle swarm optimization. In *Proceedings of the 2002 Congress on Evolutionary Computation, CEC 2002*, volume 2, pages 1051–1056. IEEE, 2002.
- S. C. Cohen and L. N. De Castro. Data clustering with particle swarms. In *2006 IEEE Congress on Evolutionary Computation, CEC 2006*, pages 1792–1798. IEEE, 2006.
- K. Conrad. Probability Distributions and Maximum Entropy. *Entropy*, 6(452), 2004.
- L. Cosentino. *Integrated Reservoir Studies*. Editions Technip, 2001.
- G. D. Couples. Geomechanical impacts on flow in fractured reservoirs. *Geological Society, London, Special Publications*, 374(1):145–172, 2013.
- G. D. Couples and H. Lewis. Effects of interlayer slip in model forced folds. *Geological Society, London, Special Publications*, 169(1):129–144, 1999.
- G. D. Couples, H. Lewis, and P. W. Geoff Tanner. Strain partitioning during flexural-slip folding. *Geological Society, London, Special Publications*, 127:149–165, 1998.
- R. Damiano, V. Lombardo, and F. Nunnari. Virtual agents for the production of linear animations. *Entertainment Computing*, 4:187–194, 2013.
- E. Damsleth, C. B. Tjolsen, H. Omre, and H. H. Haldorsen. Two-stage stochastic model applied to a North Sea reservoir. *Journal of Petroleum Technology*, 44(4):402–408, 1992.
- A. Datta-Gupta and M. J. King. *Streamline simulation : theory and practice*. 2007.
- G. de Joussineau and J. P. Petit. Mechanical insights into the development of fracture corridors in layered rocks. *Journal of Structural Geology*, 144(104278), 2021.
- E. A. De Kemp. Spatial agents for geological surface modelling. *Geoscientific Model Development*, 14(11):6661–6680, 2021.

- R. H. Dean and L. L. Lo. Simulations of Naturally Fractured Reservoirs. *SPE Reservoir Engineering*, 3(2):638–648, 1988.
- D. L. DeAngelis and S. G. Diaz. Decision-Making in Agent-Based Modeling: A Current Review and Future Prospectus. *Frontiers in Ecology and Evolution*, 6(237), 2019.
- W. Dershowitz, P. L. Pointe, and T. Doe. Advances in discrete fracture network modeling. pages 882–894, 2004.
- W. S. Dershowitz and H. H. Herda. Interpretation of fracture spacing and intensity. In *33rd U.S. Symposium on Rock Mechanics (USRMS), Santa Fe, New Mexico*, pages 757–766, 1992.
- C. V. Deutsch and L. Wang. Hierarchical Object-Based Geostatistical Modeling of Fluvial Reservoir. *SPE Annual Technical Conference and Exhibition, Denver, Colorado*, pages 221–235, 1996.
- J. L. Deutsch and C. V. Deutsch. Latin hypercube sampling with multidimensional uniformity. *Journal of Statistical Planning and Inference*, 142(3):763–772, 2012.
- A. Dobrin. A review of properties and variations of Voronoi diagrams. *Whitman College*, 10(1.453):9156, 2005.
- Q. Du, V. Faber, and M. Gunzburger. Centroidal Voronoi tessellations: Applications and algorithms. *SIAM Review*, 41(4):637–676, 1999.
- M.-P. Dubuisson and A. Jain. A modified Hausdorff distance for object matching. In *Proceedings of 12th international conference on pattern recognition*, pages 566–568. IEEE, 1994.
- L. J. Durlofsky. Numerical calculation of equivalent grid block permeability tensors for heterogeneous porous media. *Water Resources Research*, 27(5):699–708, 1991.
- R. Eberhart and Y. Shi. Comparing nonlinear inertia weights and constriction factors in particle swarm optimization. In *Proceedings of the 2000 congress on evolutionary computation*, volume 1, pages 84–88. IEEE, 2000.
- R. C. Eberhart, Y. Shi, and J. Kennedy. *Swarm intelligence*. Elsevier, 2001.

- Edwin T. Jaynes. On The Rationale of Maximum-Entropy Methods. *Proceedings of The IEEE*, 70(9):939–952, 1982.
- M. Elfeel, S. Jamal, C. Enemanna, D. Arnold, and S. Geiger. Effect of DFN upscaling on history matching and prediction of naturally fractured reservoirs. In *75th European Association of Geoscientists and Engineers Conference and Exhibition 2013 Incorporating SPE EUROPEC 2013: Changing Frontiers*, 2013.
- A. P. Engelbrecht. *Fundamentals of computational swarm intelligence*. John Wiley & Sons, Inc., 2005.
- A. P. Engelbrecht. Particle swarm optimization: Global best or local best? In *2013 BRICS congress on computational intelligence and 11th Brazilian congress on computational intelligence*, pages 124–135. IEEE, 2013.
- J. B. Ericsson, H. C. McKean, and R. J. Hooper. Facies and curvature controlled 3D fracture models in a Cretaceous carbonate reservoir, Arabian Gulf. *Geological Society, London, Special Publications*, 147:299–312, 1998.
- J. W. Essam. Percolation theory. *Reports on Progress in Physics*, 43(7):833–912, 1980.
- M. Ester, H.-P. Kriegel, J. Sander, and X. X. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD-96 Proceedings*, pages 226–231, 1996.
- J. D. Farmer and D. Foley. The economy needs agent-based modelling. *Nature*, 460(7256):685–686, 2009.
- D. R. Faulkner, C. A. Jackson, R. J. Lunn, R. W. Schlische, Z. K. Shipton, C. A. Wibberley, and M. O. Withjack. A review of recent developments concerning the structure, mechanics and fluid flow properties of fault zones. *Journal of Structural Geology*, 32(11):1557–1575, 2010.
- O. Fay-Gomord, M. Verbiest, E. Lasseur, B. Caline, C. Allanic, F. Descamps, S. Vandycke, and R. Swennen. Geological and mechanical study of argillaceous North Sea chalk: Implications for the characterisation of fractured reservoirs. *Marine and Petroleum Geology*, 92:962–978, 2018.

- J. L. Fernández Martínez, T. Mukerji, E. García Gonzalo, and A. Suman. Reservoir characterization and inversion uncertainty via a family of particle swarm optimizers. *Geophysics*, 77(1), 2012.
- N. D. Forbes Inskip, P. G. Meredith, M. R. Chandler, and A. Gudmundsson. Fracture Properties of Nash Point Shale as a Function of Orientation to Bedding. *Journal of Geophysical Research: Solid Earth*, 123:8428–8444, 2018.
- M. Franzel, S. J. Jones, N. Meadows, M. B. Allen, and K. Mccaffrey. The interplay of climatic and eustatic controls on syn-rift fluvial architecture : A high-resolution study from the Triassic Upper Buntsandstein , NE Spain. *Basin Research*, 33:1–25, 2021.
- B. Gauthier, V. Auzias, M. Garcia, and E. Chiapello. Static and Dynamic Characterization of Fracture Pattern in the Upper Jurassic Reservoirs of an Offshore Abu Dhabi Field: From Well Data to Full Field Modeling. In *Abu Dhabi International Petroleum Exhibition and Conference*. OnePetro, 2002.
- S. Geiger and S. Matthai. What can we learn from high-resolution numerical simulations of single- and multi-phase fluid flow in fractured outcrop analogues? *Geological Society, London, Special Publications*, 374(1):125–144, 2014.
- P. W. Geoff Tanner. The flexural-slip mechanism. *Journal of Structural Geology*, 11(6): 635–655, 1 1989.
- P. Ghadai, L. P. Shree, L. Chhatria, and R. Prasad. A Study on Agent Based Modelling for Traffic Simulation. *International journal of Computer Science and Information Technologies*, 7(2):932–936, 2016.
- K. Ghosh and S. Mitra. Structural controls of fracture orientations, intensity, and connectivity, Teton anticline, Sawtooth Range, Montana. *AAPG Bulletin*, 93(8):995–1014, 2009.
- J. R. Gilman. Practical aspects of simulation of fractured reservoirs. In *International Forum on Reservoir Simulation, Buhl, Baden-Baden, Germany*, pages 23–27, 2003.
- A. Golan, G. Judge, and S. Robinson. Recovering Information from Incomplete or Partial Multisectoral Economic Data. *The Review of Economics and Statistics*, 76(3):541–549, 1994.

- Golder-Assoicates. FracMan 7.7, 2018.
- G. M. Grammer, P. M. Harris, and P. Eberli. Integration of outcrop and modern analogs in reservoir modeling: Overview with examples from the Bahamas. *AAPG Memoir*, 80: 1–22, 2004.
- A. A. Griffith. The Phenomena of Rupture and Flow in Solids. *Philosophical transactions of the royal society of london. Series A, containing papers of a mathematical or physical character*, 221(582-593):163–198, 1921.
- F. B. Guardiano and R. M. Srivastava. Multivariate geostatistics: beyond bivariate moments. In *Geostatistics Troia '92. Vol. 1*, pages 133–144. Springer, 1993.
- A. Gudmundsson. *Rock fractures in geological processes*. Cambridge University Press, 2011.
- S. Guiasu and A. Shenitzer. The principle of maximum entropy. *The Mathematical Intelligencer*, 7(1):42–48, 1985.
- R. J. Guo and K. Sprague. Replication of human operators' situation assessment and decision making for simulated area reconnaissance in wargames. *Journal of Defense Modeling and Simulation*, 13(2):213–225, 2016.
- W. Guo, L. Zhu, L. Wang, Q. Wu, and F. Kong. An Entropy-Assisted Particle Swarm Optimizer for Large-Scale Optimization Problem. *Mathematics*, 7:414, 2019.
- Y. Hajizadeh, M. Christie, and V. Demyanov. History matching with differential evolution approach; a look at new search strategies. In *72nd European Association of Geoscientists and Engineers Conference and Exhibition 2010: A New Spring for Geoscience. Incorporating SPE EUROPEC 2010*, 2010.
- S. a. Hall, H. Lewis, and X. Macle. Improved seismic identification of inter-fault damage via a linked geomechanics-seismic approach. *Geological Society, London, Special Publications*, 289(1):187–207, 2007.
- T. M. Hansen. Entropy and Information Content of Geostatistical Models. *Mathematical Geosciences*, 53:163–184, 2021.

- N. J. Hardebol, C. Maier, H. Nick, S. Geiger, G. Bertotti, and H. Boro. Multiscale fracture network characterization and impact on flow: A case study on the Latemar carbonate platform. *Journal of Geophysical Research: Solid Earth*, 120:8197–8222, 2015.
- J. Harte. *Maximum entropy and ecology: a theory of abundance, distribution, and energetics*. 2011.
- M. Z. Hasan and H. Al-Rizzo. Optimization of Sensor Deployment for Industrial Internet of Things Using a Multiswarm Algorithm. *IEEE Internet of Things Journal*, 6(6): 10344–10362, 2019.
- S. Helwig and R. Wanka. Theoretical analysis of initial particle swarm behavior. In *International conference on parallel problem solving from nature*, pages 889–898. Springer, 2008.
- S. Helwig, F. Neumann, and R. Wanka. Particle swarm optimization with velocity adaptation. In *Proceedings of the 2009 International Conference on Adaptive and Intelligent Systems, ICAIS 2009*, pages 146–151. IEEE, 2009.
- N. Higashi and H. Iba. Particle swarm optimization with Gaussian mutation. In *Proceedings of the 2003 IEEE Swarm Intelligence Symposium*, pages 72–79. IEEE, 2003.
- R. J. Hijmans and C. H. Graham. The ability of climate envelope models to predict the effect of climate change on species distributions. *Global Change Biology*, 12(12): 2272–2281, 2006.
- C. Hilgers and J. L. Urai. Experimental study of syntaxial vein growth during lateral fluid flow in transmitted light: First results. *Journal of Structural Geology*, 24(6-7): 1029–1043, 2002.
- R. Hill. Elastic properties of reinforced solids: some theoretical principles. *Journal of the Mechanics and Physics of Solids*, 11(5):357–372, 1963.
- J. H. Holland. *Emergence: From chaos to order*. Oxford, UK: Oxford University Press, 1998.
- M. Honarkhah and J. Caers. Stochastic simulation of patterns using distance-based pattern modeling. *Mathematical Geosciences*, 42(5):487–517, 2010.

- J. A. Howell, A. W. Martinius, and T. R. Good. The application of outcrop analogues in geological modelling: a review, present status and future outlook. *Geological Society, London, Special Publications*, 387:1–25, 2014.
- M. Hu, T. Wu, and J. D. Weir. An adaptive particle swarm optimization with multiple adaptive methods. *IEEE Transactions on Evolutionary Computation*, 17(5):705–720, 2013.
- X. Hu and R. C. Eberhart. Tracking dynamic systems with PSO: where’s the cheese. In *Proceedings of the workshop on particle swarm optimization*, pages 80–83, 2001.
- T. Huang and A. S. Mohan. A hybrid boundary condition for robust particle swarm optimization. *IEEE Antennas and Wireless Propagation Letters*, 4:112–117, 2005.
- W. L. Huskey and P. B. Crawford. Performance of Petroleum Reservoirs Containing Vertical Fractures in the Matrix. *Society of Petroleum Engineers Journal*, pages 221–228, 1967.
- J. Hutahaeen. *Multi-Objective Methods for History Matching, Uncertainty Prediction and Optimisation in Reservoir Modelling*. PhD thesis, Heriot-Watt University, 2017.
- J. Hutahaeen, V. Demyanov, and M. Christie. Reservoir development optimization under uncertainty for infill well placement in brownfield redevelopment. *Journal of Petroleum Science and Engineering*, 175:444–464, 2019.
- J. D. Hyman, S. Karra, N. Makedonska, C. W. Gable, S. L. Painter, and H. S. Viswanathan. DfnWorks: A discrete fracture network framework for modeling subsurface flow and transport. *Computers and Geosciences*, 84:10–19, 2015.
- G. Iori and J. Porter. *Agent-Based Modelling for Financial Markets*. London, UK: Department of Economics, City University London, 2012.
- M. Irgens and W. L. Lavenue. Use of advanced optimization techniques to manage a complex drilling schedule. In *Proceedings - SPE Annual Technical Conference and Exhibition*. OnePetro, 2007.
- G. James, D. Witten, T. Hastie, and R. Tibshirani. *An Introduction to Statistical Learning*. 2013.

- L. James V. Miranda. PySwarms: a research toolkit for Particle Swarm Optimization in Python. *The Journal of Open Source Software*, 3(21):433, 2018.
- M. Jaxa-Rozen, J. H. Kwakkel, and M. Bloemendal. A coupled simulation architecture for agent-based/geohydrological modelling with NetLogo and MODFLOW. *Environmental Modelling and Software*, 115:19–37, 2019.
- E. Jaynes. Information Theory and Statistical Mechanics. *Physical Reviews*, 106(4): 620–630, 1957.
- Y. Jin and J. Branke. Evolutionary optimization in uncertain environments - A survey. *IEEE Transactions on Evolutionary Computation*, 9(3):303–317, 2005.
- Z. L. Jin, Y. Liu, and L. J. Durlofsky. Deep-learning-based surrogate model for reservoir simulation with time-varying well controls. *Journal of Petroleum Science and Engineering*, 192:107273, 2020.
- I. T. Jolliffe and J. Cadima. Principal component analysis: A review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374, 2016.
- A. G. Journel. Geostatistics: roadblocks and challenges. In *Geostatistics Troia'92*, pages 213–224. Springer, 1993.
- A. G. Journel and C. V. Deutsch. Entropy and spatial disorder. *Mathematical Geology*, 25(3):329–355, 1993.
- A. G. Journel, R. Gunderso, E. Gringarten, and T. Yao. Stochastic modelling of a fluvial reservoir : a comparative review of algorithms. *Journal of Petroleum Science and Engineering*, 21:95–121, 1998.
- Ju L., R. T., and G. M. Voronoi Tessellations and Their Application to Climate and Global Modeling. In *Numerical Techniques for Global Atmospheric Models. Lecture Notes in Computational Science and Engineering*, pages 313–342. Springer, Berlin, Heidelberg, 2011.
- A. Jung, T. Aigner, D. Palermo, S. Nardon, and M. Pontiggia. A hierarchical database on carbonate geobodies and its application to reservoir modelling using multi-point

- statistics. In *72nd European Association of Geoscientists and Engineers Conference and Exhibition 2010: A New Spring for Geoscience. Incorporating SPE EUROPEC 2010*, 2010.
- A. Jung, D. H. Fenwick, and J. Caers. Training image-based scenario modeling of fractured reservoirs for flow uncertainty quantification. *Computational Geosciences*, 17:1015–1031, 2013.
- N. Karatalov, A. Stefaniak, N. Caspian, O. Company, and L. Vaughan. DFN Modeling Aided Reservoir Characterization. In *Abu Dhabi International Petroleum Conference and Exhibition*. OnePetro, 2017.
- M. Kathrada and J. N. Carter. Case studies of successfully history matched reservoir simulation models using a powerful optimiser being of limited predictive value. In *Society of Petroleum Engineers - 14th Abu Dhabi International Petroleum Exhibition and Conference 2010, ADIPEC 2010*, 2010.
- Y. Kato, H. Okano, and S. Takahashi. Application of PSO and MOPSO to gas injection analysis. In *Society of Petroleum Engineers - International Petroleum Technology Conference 2014, IPTC 2014 - Innovation and Collaboration: Keys to Affordable Energy*, 2014.
- L. Kaufman and P. J. Rousseeuw. *Finding groups in data: an introduction to cluster analysis*. John Wiley & Sons, 2009.
- H. Kazemi, L. Merrill, K. Porterfield, and P. Zeman. Numerical Simulation of Water-Oil Flow in Naturally Fractured Reservoirs. *Society of Petroleum Engineers Journal*, 16: 317–326, 1976.
- J. Kennedy. The particle swarm : Social adaptation of knowledge. In *Proceedings of 1997 IEEE International Conference on Evolutionary Computation*, pages 303–308. IEEE, 1997.
- J. Kennedy. The behaviour of particles. In *Evolutionary Programming VII. EP 1998. Lecture Notes in Computer Science, vol 1447*, pages 579–589. Springer Berlin Heidelberg, 1998.

- J. Kennedy. Small worlds and mega-minds: Effects of neighborhood topology on particle swarm performance. In *Proceedings of the 1999 Congress on Evolutionary Computation, CEC 1999*, volume 3, pages 1931–1938. IEEE, 1999.
- J. Kennedy and R. Eberhart. Particle Swarm Optimization. In *Proceeding of the IEEE International Conference on Neural Networks.—Perth: IEEE Service Center. 1995.*, pages 1942–1948. IEEE, 1995.
- J. G. Kim and M. D. Deo. Finite element, discrete-fracture model for multiphase flow in porous media. *AIChE Journal*, 46(6):1120–1130, 2000.
- P. R. King. The connectivity and conductivity of overlapping sand bodies. In *North sea oil and gas reservoirs—II*, pages 353–362. Springer, Dordrecht, 1990.
- D. E. Knuth. *The Art of Computer Programming*, 1997.
- Kodinariya, T.M and Makwana, P.R. Review on determining number of Cluster in K-Means Clustering. *International Journal of Advance Research in Computer Science and Management Studies*, 1(6):90–95, 2013.
- C. E. Koltermann and S. M. Gorelick. Paleoclimatic signature in terrestrial flood deposits. *Science*, 256:1775–1782, 1992.
- P. L. La Pointe, P. C. Wallmann, and S. Follin. Continuum modelling of fractured rock masses: Is it useful? In *ISRM International Symposium - EUROCK 1996*, pages 343–350, 1996.
- S. T. Lancaster and R. L. Bras. A simple model of river meandering and its comparison to natural channels. *Hydrological Processes*, 16:1–26, 2002.
- Laurens van der Maaten and G. Hinton. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.
- L. S. Lee, N. Brunk, D. G. Haywood, D. Keifer, E. Pierson, P. Kondylis, J. C. Y. Wang, S. C. Jacobson, M. F. Jarrold, and A. Zlotnick. A molecular breadboard: Removal and replacement of subunits in a hepatitis B virus capsid. *Protein Science*, 26:2170–2180, 2017.

- Leland McInnes, J. Healy, and S. A. Revision. The hdbscan Clustering Library, 2016. URL [https://hdbscan.readthedocs.io/en/latest/how\\_hdbscan\\_works.html](https://hdbscan.readthedocs.io/en/latest/how_hdbscan_works.html).
- P. Lemonnier and B. Bourbiaux. Simulation of Naturally Fractured Reservoirs. State of the Art Part 2. *Oil and Gas Science and Technology*, 65(2):263–286, 2010.
- H. Lewis and G. D. Couples. Production evidence for geological heterogeneities in the Anschutz Ranch East Field, western USA. *Geological Society Special Publication*, 73: 321–338, 1993.
- H. Lewis, S. A. Hall, and G. D. Couples. Geomechanical simulation to predict open subsurface fractures. *Geophysical Prospecting*, 57(2):285–299, 2009.
- D. Li, B. Fu, Y. Wang, G. Lu, Y. Berezin, H. E. Stanley, and S. Havlin. Percolation transition in dynamical traffic network with evolving critical bottlenecks. *PNAS*, 112 (3):669–672, 2015.
- Q. Li, S. Y. Liu, and X. S. Yang. Influence of initialization on the performance of metaheuristic optimizers. *Applied Soft Computing*, 91:106193, 2020.
- X. Li. Niching without niching parameters: Particle swarm optimization using a ring topology. *IEEE Transactions on Evolutionary Computation*, 14(1):150–169, 2010.
- J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Transactions on Evolutionary Computation*, 10(3):281–295, 2006.
- K. A. Lie, S. Krogstad, I. S. Ligaarden, J. R. Natvig, H. M. Nilsen, and B. Skaflestad. Open-source MATLAB implementation of consistent discretisations on complex grids. *Computational Geosciences*, 16:297–322, 2012.
- A. Likas, N. Vlassis, and J. J. Verbeek. The global k-means clustering algorithm. *Pattern Recognition*, 36:451–461, 2003.
- S. H. Ling, H. H. Iu, K. Y. Chan, H. K. Lam, B. C. Yeung, and F. H. Leung. Hybrid particle swarm optimization with wavelet mutation and its industrial applications. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 38(3):743–763, 2008.

- N. Liu and D. S. Oliver. Critical evaluation of the ensemble Kalman filter on history matching of geologic facies. *SPE Reservoir Evaluation and Engineering*, 8(6):470–477, 2005.
- X. Liu, S. Srinivasan, and D. Wong. Geological Characterization of Naturally Fractured Reservoirs Using Multiple Point Geostatistics. In *Proceedings - SPE Symposium on Improved Oil Recovery*, pages 1057–1068. OnePetro, 2002.
- X. Liu, C. Zhang, Q. Liu, and J. Birkholzer. Multiple-point statistical prediction on fracture networks at Yucca Mountain. *Environmental Geology*, 57(6):1361–1370, 2009.
- J. C. Loayza, N. Orellana, I. Yemez, and V. Singh. Importance of conceptual geological models in 3D reservoir modelling. *First Break*, 34:39–49, 2016.
- L. Looi, D. Arnold, V. Demyanov, H. Lewis, and A. Anupam. Modelling Uncertainties of Non-stationary Fractured Reservoirs. In *79th EAGE Conference & Exhibition 2017*, 2017.
- S. Lopez, I. Cojan, J. Rivoirard, and A. & Galli. Process-based stochastic modelling: meandering channelized reservoirs. In *Analogue and Numerical Modelling of Sedimentary Systems: From Understanding to Prediction*, pages 139–144. Wiley, Oxford, 2009.
- J. C. Lorenz, L. W. Teufel, and N. R. Warpinski. Regional fractures I: a mechanism for the formation of regional fractures at depth in flat-lying reservoirs. *AAPG bulletin*, 75(11):1714–1737, 1991.
- N. Lynn, M. Z. Ali, and P. N. Suganthan. Population topologies for particle swarm optimization and differential evolution. *Swarm and Evolutionary Computation*, 39:24–35, 2018.
- C. M. Macal and M. J. North. Tutorial on agent-based modelling and simulation. *Journal of Simulation*, 4:151–162, 2010.
- J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297, 1967.

- J. K. Madsen, R. Bailey, E. Carrella, and P. Koralus. Analytic Versus Computational Cognitive Models: Agent-Based Modeling as a Tool in Cognitive Sciences. *Current Directions in Psychological Science*, 28(3):299–305, 2019.
- L. Maerten and F. Maerten. Chronologic modeling of faulted and fractured reservoirs using geomechanically based restoration: Technique and industry applications. *AAPG Bulletin*, 90(8):1201–1226, 8 2006.
- G. H. Mäkel. The modelling of fractured reservoirs: constraints and potential for fracture network geometry and hydraulics analysis. *Geological Society, London, Special Publications*, 292:375–403, 2007.
- T. Manzocchi, J. J. Walsh, P. Nell, and G. Yielding. Fault transmissibility multipliers for flow simulation models. *Petroleum Geoscience*, 5(1):53–63, 1999. ISSN 13540793. doi: 10.1144/petgeo.5.1.53.
- G. Mariethoz and J. Caers. *Multiple-Point Geostatistics Stochastic modeling with training images*. Wiley Blackwell, 2014.
- G. Mariethoz, P. Renard, and J. Straubhaar. The direct sampling method to perform multiple-point geostatistical simulations. *Water Resources Research*, 46, 2010.
- K. Mason, J. Duggan, and E. Howley. Multi-objective dynamic economic emission dispatch using particle swarm optimisation variants. *Neurocomputing*, 270:188–197, 2017.
- G. Matheron. Principles of geostatistics. *Economic geology*, 58(8):1246–1266, 1963.
- L. S. Matott, A. J. Rabideau, and J. R. Craig. Pump-and-treat optimization using analytic element method flow models. *Advances in Water Resources*, 29:760–775, 2006.
- S. K. Matthäi and M. Belayneh. Fluid flow partitioning between fractures and a permeable rock matrix. *Geophysical Research Letters*, 31, 2004.
- J. McCall. Genetic algorithms for modelling and optimisation. *Journal of Computational and Applied Mathematics*, 184:205–222, 2005.
- R. J. McEliece. Entropy and mutual information, 2012.

- L. McInnes. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction, 2018. URL <https://umap-learn.readthedocs.io/en/latest/index.html>.
- L. McInnes, J. Healy, and J. Melville. UMAP : Uniform Manifold Approximation and Projection for Dimension Reduction. *arxiv*, 2018.
- M. D. McKay. Latin hypercube sampling as a tool in uncertainty analysis of computer models. In *Proceedings of the 24th Winter Simulation Conference*, pages 557–564, 1992.
- M. D. McKay, R. J. Beckman, and W. J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):239–245, 1979.
- Mesa. Mesa: Agent-based modeling in Python 3+, 2015. URL <https://mesa.readthedocs.io/en/latest/index.html>.
- N. Metropolis, S. Ulam, S. Journal, A. Statistical, and N. Sep. The Monte Carlo Method. *Journal of the American Statistical Association*, 44(247):335–341, 1949.
- K. B. Min, L. Jing, and O. Stephansson. Determining the equivalent permeability tensor for fractured rock masses using a stochastic REV approach: Method and application to the field data from Sellafield, UK. *Hydrogeology Journal*, 12:497–510, 2004.
- L. Mohamed. *Novel Sampling Techniques for Reservoir History Matching Optimisation and Uncertainty Quantification in Flow Prediction*. PhD thesis, Heriot-Watt University, 2011.
- L. Mohamed, M. Christie, and V. Demyanov. Comparison of Stochastic Sampling Algorithms for Uncertainty Quantification. *SPE Journal*, 15:31–38, 2010.
- C. R. Mouser and S. A. Dunn. Comparing genetic algorithms and particle swarm optimisation for an inverse problem exercise There is a long history of development and updating of dynamic finite element. *ANZIAM Journal*, 46:89–101, 2005.
- O. Møyner, S. Krogstad, and K.-A. Lie. The application of flow diagnostics for reservoir management. *SPE Journal*, 20(2):306–323, 2015.

- J. Mullins, H. van der Vegt, and J. Howell. Combining Multiple Point Geostatistics and Process-based Models for Improved Reservoir Modelling. *Petroleum Geoscience*, 27, 2021.
- Murphy Benjamin. PyKrige, 2021. URL <https://github.com/GeoStat-Framework/PyKrige>.
- Murphy Benjamin, R. Yurchak, and S. Mueller. GeoStat-Framework/PyKrige: v1.6.0 (Version v1.6.1), 2021.
- W. Narr. *Characterization of Naturally Fractured Reservoirs*. Society of Petroleum Engineers., 2006.
- R. Nelson. *Geologic Analysis of Naturally Fractured Reservoirs*. Number 2. Gulf Professional Publishing, 2001.
- S. Neuman. Stochastic continuum representation of fractured rock permeability as an alternative to the REV and fracture network concepts. In *Groundwater Flow and Quality Modelling*, pages 331–362. Springer, 1988.
- M. J. North and C. M. Macal. *Managing business complexity: discovering strategic solutions with agent-based modeling and simulation*. Oxford University Press, 2007.
- A. Obeyesekera, Q. Lei, P. Salinas, D. Pavlidis, J. Xiang, J. P. Latham, and C. C. Pain. Modelling stress-dependent single and multi-phase flows in fractured porous media based on an immersed-body method with mesh adaptivity. *Computers and Geotechnics*, 103:229–241, 2018.
- M. Oda. Permeability Tensor for Discontinuous Rock Masses. *Géotechnique*, 35(4): 483–495, 1985.
- N. E. Odling, P. Gillespie, B. Bourgin, C. Castaing, J. P. Chiles, N. P. Christensen, E. Fillion, A. Genter, C. Olsen, L. Thrane, R. Trice, E. Aarseth, J. J. Walsh, and J. Watterson. Variations in fracture system geometry and their implications for fluid flow in fractures hydrocarbon reservoirs. *Petroleum Geoscience*, 5:373–384, 1999.
- A. Okabe, B. Boots, K. Sugihara, and S. N. Chiu. *Spatial tessellations: concepts and applications of Voronoi diagrams*. John Wiley & Sons, 2009.

- J. E. Onwunalu and L. J. Durlofsky. Application of a particle swarm optimization algorithm for determining optimum well location and type. *Computational Geosciences*, 14(1): 183–198, 2010.
- S. E. Palmer. *Vision Science: Photons to Phenomenology*. MIT Press, Cambridge, MA, 1999.
- K. Parsopoulos and M. Vrahatis. Recent approaches to global optimization problems through Particle Swarm Optimization. *Natural computing*, 1(2):235–306, 2002a.
- K. E. Parsopoulos and M. N. Vrahatis. Particle swarm optimization method in multiobjective problems. In *Proceedings of the 2002 ACM symposium on Applied computing*, pages 603–607, 2002b.
- C. Pealat, G. Bouleux, V. Cheutet, C. Pealat, G. Bouleux, and V. Cheutet. Improved Time-Series Clustering with UMAP dimension reduction method. In *ICPR 2020 - 25th International conference in Pattern Recognition, Jan 2021, Milano, 2021*.
- L. Perez and S. Dragicevic. An agent-based approach for modeling dynamics of contagious disease spread. *International Journal of Health Geographics*, 8:50, 2009.
- S. J. Phillips, R. P. Anderson, and R. E. Schapire. Maximum entropy modeling of species geographic distributions. *Ecological Modelling*, 190:231–259, 2006.
- T. Phillips, N. Kampman, K. Bisdorn, N. D. Forbes Inskip, S. A. den Hartog, V. Cnudde, and A. Busch. Controls on the intrinsic flow properties of mudrock fractures: A review of their importance in subsurface storage. *Earth-Science Reviews*, 211:103390, 2020.
- S. J. Pirson. Performance of Fractured Oil Reservoirs. *AAPG Bulletin*, 37(2):232–244, 1953.
- N. J. Price. *Fault and Joint Development in Brittle and Semi-brittle Rock*. Pergamon Press, 1966.
- J. M. Questiaux, G. D. Couples, and N. Ruby. Fractured reservoirs with fracture corridors. *Geophysical Prospecting*, 58(2):279–295, 2010.
- J. G. Ramsay and M. I. Huber. *Modern Structural Geology*, volume 2. Academic Press, 1987.

- N. Remy, A. Boucher, and J. Wu. *Applied Geostatistics with SGeMS: A Users' Guide*. Technical report, 2009.
- F. M. Reza. *An introduction to information theory*. Courier Corporation, 1994.
- P. Ringrose and M. Bentley. *Reservoir model design*, volume 2. Springer, 2021.
- P. C. Robinson. Connectivity of fracture systems—a percolation theory approach. *Journal of Physics A: Mathematical and General*, 16:605–614, 1983.
- T. S. Rojas. *Controlling Realism and Uncertainty in Reservoir Models using Intelligent Sedimentological Prior Information*. PhD thesis, Heriot-Watt University, 2013.
- G. Rongier, P. Collon, P. Renard, J. Straubhaar, and J. Sausse. Comparing connected structures in ensemble of random fields. *Advances in Water Resources*, 96:145–169, 2016.
- R. Rwechungura, M. Dadashpour, and J. Kleppe. Application of particle swarm optimization for parameter estimation integrating production and time lapse seismic data. In *Society of Petroleum Engineers - Offshore Europe Oil and Gas Conference and Exhibition 2011, OE 2011*, 2011.
- P. N. Sævik and C. W. Nixon. Inclusion of Topological Measurements into Analytic Estimates of Effective Permeability in Fractured Media. *Water Resources Research*, 53: 9424–9443, 2017.
- R. F. Santos, T. S. Miranda, J. A. Barbosa, I. F. Gomes, G. C. Matos, J. F. Gale, V. H. Neumann, and L. J. Guimaraes. Characterization of natural fracture systems: Analysis of uncertainty effects in linear scanline results. *AAPG Bulletin*, 99(12):2203–2219, 2015.
- C. Scheidt and J. Caers. Representing spatial uncertainty using distances and kernels. *Mathematical Geosciences*, 41:397–419, 2009.
- C. Scheidt and J. Caers. Uncertainty Quantification in Reservoir Performance Using Distances and Kernel Methods—Application to a West Africa Deepwater Turbidite Reservoir. *SPE Journal*, 14:680–692, 2013.

- C. Scheidt, C. Jeong, T. Mukerji, and J. Caers. Probabilistic falsification of prior geologic uncertainty with seismic amplitude data: Application to a turbidite reservoir case. *Geophysics*, 80(5):89–100, 2015.
- Schlumberger. Characterization of Fractured Reservoirs: Reliable, predictive models to optimize carbonate reservoir performance. Technical report, 2008.
- C. H. Scholz. *The Mechanics of Earthquakes and Faulting*. Cambridge University Press, 2 edition, 2002.
- J. F. Schutte and A. A. Groenwold. A study of global optimization using particle swarms. *Journal of Global Optimization*, 31(1):93–108, 2005.
- J. F. Schutte, J. A. Reinbolt, B. J. Fregly, R. T. Haftka, and A. D. George. Parallel global optimization with the particle swarm algorithm. *International Journal for Numerical Methods in Engineering*, 61(13):2296–2315, 2004.
- T. D. Seers. *Image Based Characterisation of Structural Heterogeneity within Clastic Reservoir Analogues*. PhD thesis, University of Manchester, 2015.
- R. V. Self, A. Atashnezhad, and G. Hareland. Use of a swarm algorithm to reduce the drilling time through measurable improvement in rate of penetration. In *50th US Rock Mechanics / Geomechanics Symposium 2016*. OnePetro, 2016.
- M. Shahvali, B. Mallison, K. Wei, and H. G. Gross. An alternative to streamlines for flow diagnostics on structured and unstructured grids. *SPE Journal*, 17(3):768–778, 2012.
- C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423, 1948.
- Y. Shi and R. Eberhart. Modified particle swarm optimizer. In *Proceedings of the IEEE Conference on Evolutionary Computation, ICEC*, 1998.
- Y. Shi and R. C. Eberhart. Empirical study of particle swarm optimization. In *Proceedings of the 1999 Congress on Evolutionary Computation, CEC 1999*, pages 1945–1950, 1999.
- G. M. Shook and K. M. Mitchell. A Robust Measure of Heterogeneity for Ranking Earth Models: The F PHI Curve and Dynamic Lorenz Coefficient. In *SPE Annual Technical Conference and Exhibition*,. OnePetro, 2009.

- J. E. Shore and R. W. Johnson. Axiomatic Derivation of the Principle of Maximum Entropy and the Principle of Minimum Cross-Entropy. *IEEE Transactions on Information Theory*, 26(1):26–37, 1980.
- P. Siarry, A. Pétrowski, and M. Bessaou. A multipopulation genetic algorithm aimed at multimodal optimization. *Advances in Engineering Software*, 33(4):207–213, 2002.
- J. Skilling and R. K. Bryan. Maximum entropy image reconstruction: general algorithm. *Mon Not R astr. Soc*, 211:111 – 124, 1984.
- T. Sousa, A. Silva, and A. Neves. Particle Swarm based Data Mining Algorithms for classification tasks. *Parallel Computing*, 30(5-6):767–783, 2004.
- G. H. Spence, G. D. Couples, T. G. Bevan, R. Aguilera, J. W. Cosgrove, J.-M. Daniel, and J. Redfern. Advances in the study of naturally fractured hydrocarbon reservoirs: a broad integrated interdisciplinary applied topic. *Geological Society, London, Special Publications*, 374:1–22, 2014.
- V. E. Spooner, S. Geiger, and D. Arnold. Ranking Fractured Reservoir Models Using Flow Diagnostics. In *SPE Reservoir Simulation Conference*. OnePetro, 2019.
- V. E. Spooner, S. Geiger, and D. Arnold. Dual-Porosity Flow Diagnostics for Spontaneous Imbibition in Naturally Fractured Reservoirs. *Water Resources Research*, 57, 2021.
- D. W. Stearns. Macrofracture patterns on Teton Anticline, northwest Montana. *Transactions of the American Geophysical Union*, 45:107–108, 1964.
- D. W. Stearns and M. Friedman. Reservoirs in Fractured Rock: Geologic Exploration Methods. *AAPG Special Volumes*, pages 82–106, 1972.
- B. Steffens, V. Demyanov, G. Couples, D. Arnold, and H. Lewis. Flow Through Fractured Reservoirs Under Geological and Geomechanical Uncertainty. In *81st EAGE Conference and Exhibition 2019*. EAGE, 2019.
- B. Steffens, V. Demyanov, D. Arnold, and H. Lewis. A Workflow for Dynamic Screening Assisted, Automated Fractured Reservoir Modelling. In *EAGE 2020 Annual Conference & Exhibition online*. EAGE, 2020.

- B. Steffens, V. Demyanov, D. Arnold, and H. Lewis. Can agent-based modelling help to update conceptual geological models ? A fractured reservoir example. In *2021 Ring Meeting*. ASGA, 2021a.
- B. Steffens, V. Demyanov, D. Arnold, and H. Lewis. Entropy-driven particle swarm optimization for reservoir modelling under geological uncertainty – application to a fractured reservoir. In *EAGE 2021 Annual Conference & Exhibition*. EAGE, 2021b.
- B. Steffens, Q. Corlay, N. Suurmeyer, J. Noglows, D. Arnold, and V. Demyanov. Can agents model hydrocarbon migration for petroleum system analysis? A fast-screening tool to de-risk hydrocarbon prospects. *Energies*, 15:902, 2022.
- H. Steinhaus. Sur la division des corps matériels en parties. *Bull. Acad. Polon. Sci*, 1 (804):801, 1956.
- J. Straubhaar. Deesse user’s guide, 2017.
- S. Strebelle. *Sequential simulation drawing structures from training images*. PhD thesis, Stanford University, 2000.
- S. Strebelle. Conditional simulation of complex geological structures using multiple-point statistics. *Mathematical Geology*, 34(1), 2002.
- P. N. Suganthan. Particle swarm optimiser with neighbourhood operator. In *Proceedings of the 1999 Congress on Evolutionary Computation, CEC 1999*, pages 1958–1962, 1999.
- J. Sun, W. Xu, and W. Fang. A diversity-guided quantum-behaved particle swarm optimization algorithm. In *Asia-Pacific Conference on Simulated Evolution and Learning*, pages 497–504. Springer Berlin Heidelberg, 2006.
- N. Z. Sun. *Inverse problems in groundwater modeling*. Springer Science & Business Media, 6 edition, 2013.
- S. Suzuki and J. Caers. History matching with an uncertain geological scenario. In *SPE Annual Technical Conference and Exhibition*, pages 1118–1134. OnePetro, 2006.
- P. Tahmasebi. Multiple Point Statistics: A Review. In *Handbook of Mathematical Geosciences: Fifty Years of IAMG*, pages 613–643. 2018.

- P. Tahmasebi, A. Hezarkhani, and M. Sahimi. Multiple-point geostatistical modeling based on the cross-correlation functions. *Computational Geosciences*, 16:779–797, 2012.
- M. R. Thiele, R. P. Batycky, and D. H. Fenwick. Streamline simulation for modern reservoir-engineering workflows. *Journal of Petroleum Technology*, 62:64–70, 2010.
- J. F. Thovert, V. V. Mourzenko, and P. M. Adler. Percolation in three-dimensional fracture networks for arbitrary size and shape distributions. *Physical Review E*, 95(4), 2017.
- D. Tiab and E. C. Donaldson. *Petrophysics*. Gulf Professional Publishing, 2 edition, 2004.
- S. Tisue and U. Wilensky. NetLogo: Design and implementation of a multi-agent modeling environment. In *Proceedings of Agent 2004*, 2004.
- E. Tolstukhin, B. Lyngnes, and H. H. Sudan. Ekofisk 4D seismic-seismic history matching workflow. In *SPE Europec/EAGE Annual Conference*. OnePetro, 2012.
- I. C. Trelea. The particle swarm optimization algorithm: Convergence analysis and parameter selection. *Information Processing Letters*, 85:317–325, 2003.
- R. K. Ursem and P. Vadstrup. Parameter identification of induction motors using stochastic optimization algorithms. *Applied Soft Computing*, 4(1):49–64, 2004.
- B. A. van der Pluijm and S. Marshack. *Earth structure: An introduction to structural geology and tectonics*. W. W. Norton and Company, Inc., 2004.
- T. Van Golf-Racht. *Fundamentals of fractured reservoir engineering*. Elsevier, 1982.
- J. C. Van Wagoner, R. M. Mitchum, K. M. Campion, and V. D. Rahmanian. *Siliciclastic Sequence Stratigraphy in Well Logs, Cores, and Outcrops: Concepts for High-Resolution Correlation of Time and Facies*. AAPG Methods in Exploration Series, 1990.
- M. D. Varga, A. Schaaf, and F. Wellmann. GemPy 1.0: open-source stochastic geological modeling and inversion. *Geoscientific Model Development*, 12:1–32, 2019.
- M. D. Varga, A. Schaaf, and F. Wellmann. Gempy, 2021. URL <https://www.gempy.org/>.

- O. Vazquez, C. Young, V. Demyanov, D. Arnold, A. Fisher, A. Macmillan, and M. Christie. Produced-water-chemistry history matching in the Janice field. *SPE Reservoir Evaluation and Engineering*, 18(4):564–576, 2015.
- G. Venter. Review of Optimization Techniques. *Encyclopedia of Aerospace Engineering*, 2010.
- J. M. Vermilye and C. H. Scholz. The process zone: A microstructural view of fault growth. *Journal of Geophysical Research*, 103(B6):1223–12237, 1998.
- S. Virgo, S. Abe, and J. L. Urai. The evolution of crack seal vein and fracture networks in an evolving stress field: Insights from Discrete Element Models of fracture sealing. *Journal of Geophysical Research: Solid Earth*, 119(12):8708–8727, 2014.
- J. von Neumann. *Theory of Self-Reproducing Automata*. University of Illinois Press, 1966.
- C. Wang, V. Satuluri, and S. Parthasarathy. Local probabilistic models for link prediction. In *Seventh IEEE international conference on data mining*, pages 322–331. IEEE, 2007.
- H. Wang, H. Wang, H. Sun, C. Li, S. Rahnamayan, and J.-s. Pan. Diversity enhanced particle swarm optimization with neighborhood search Diversity enhanced particle swarm optimization with neighborhood search. *Information Sciences*, 223:119–135, 2013.
- J. E. Warren and P. J. Root. The Behavior of Naturally Fractured Reservoirs. *Society of Petroleum Engineers*, 3(03):245–255, 1963.
- H. Watkins, R. W. Butler, C. E. Bond, and D. Healy. Influence of structural position on fracture networks in the Torridon Group, Achnashellach fold and thrust belt, NW Scotland. *Journal of Structural Geology*, 74:64–80, 2015.
- F. Watson, S. Krogstad, and K. A. Lie. The use of flow diagnostics to rank model ensembles. *Computational Geosciences*, 2021.
- J. F. Wellmann and K. Regenauer-Lieb. Uncertainties have a meaning: Information entropy as a quality measure for 3-D geological models. *Tectonophysics*, 526-529: 207–216, 2012.

- O. P. Wennberg, G. Casini, S. Jonoud, and D. C. Peacock. The characteristics of open fractures in carbonate reservoirs and their impact on fluid flow: A discussion. *Petroleum Geoscience*, 22:91–104, 2016.
- U. Wilensky. NetLogo, 1999.
- U. Wilensky and W. Rand. *An introduction to agent-based modeling: modeling natural, social, and engineered complex systems with NetLogo*. The MIT Press, 2015.
- C. E. Wilson, A. Aydin, M. Karimi-Fard, L. J. Durlofsky, A. Sagy, E. E. Brodsky, O. Kreylos, and L. H. Kellogg. From outcrop to flow simulation: Constructing discrete fracture models from a LIDAR survey. *AAPG Bulletin*, 95(11):1883–1906, 2011.
- K. C. Wilson and L. J. Durlofsky. Computational optimization of shale resource development using reduced-physics surrogate models. In *Society of Petroleum Engineers Western Regional Meeting 2012*, pages 126–139, 2012.
- P. A. Witherspoon, J. S. Y. Wang, K. Iwai, and J. E. Gale. Validity of cubic law for fluid-flow in a deformable rock fracture. *Water Resources Research*, 16(6):1016 – 1024, 1980.
- S. Wolfram. *Cellular Automata*. CRC press, 1994.
- D. L. Y. Wong. *Explicit- Implicit representation of Naturally Fractured Reservoirs*. PhD thesis, 2019.
- J. Wu, T. Zhang, and A. Boucher. Non-stationary Multiple-point Geostatistical Simulations with Region Concept. In *Proceedings of the 20th SCRF meeting, Stanford, CA, USA*, 2007.
- N. Wu. *The maximum entropy method*. Springer Science & Business Media, 32 edition, 2012.
- C. K. Zahm and P. H. Hennings. Complex fracture development related to stratigraphic architecture: Challenges for structural deformation prediction, Tensleep Sandstone at the Alcova anticline, Wyoming. *AAPG Bulletin*, 93(11):1427–1446, 2009.

- Z. H. Zhan, J. Zhang, Y. Li, and H. S. Chung. Adaptive particle swarm optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 39(6):1362–1381, 2009.
- T. Zhang, P. Switzer, and A. Journel. Filter-based classification of training image patterns for spatial simulation. *Mathematical Geology*, 38(1):63–80, 2006.
- T. F. Zhang. *Filter-Based Training Pattern Classification For Spatial Simulation*. PhD thesis, Stanford University, 2006.
- W. J. Zhang, X. F. Xie, and D. C. Bi. Handling boundary constraints for numerical optimization by particle swarm flying in periodic search space. In *Proceedings of the 2004 Congress on Evolutionary Computation, CEC2004*, pages 2307–2311. IEEE, 2004.
- M. D. Zoback. *Reservoir Geomechanics*. 2007.
- M. Zuparic, V. Jauregui, M. Prokopenko, and Y. Yue. Quantifying the impact of communication on performance in multi-agent teams. *Artificial Life and Robotics*, 22:357–373, 2017.