Impact of structural uncertainty on the assessment of geothermal reservoirs

Zur Erlangung des akademischen Grades eines

DOKTORS DER NATURWISSENSCHAFTEN (Dr. rer. nat.)

von der KIT-Fakultät für
Bauingenieur-, Geo- und Umweltwissenschaften
des Karlsruher Instituts für Technologie (KIT) genehmigte

DISSERTATION

von
M.Sc. Ali Dashti

Tag der mündlichen Prüfung: 18.11.2024
Referent: Prof. Dr.-Ing. Thomas Kohl
Korreferent: Prof. Dr. Florian Wellmann

Kalsruhe 2025

Abstract

Renewable energy sources are the future of sustainable energy production, supplying heat and electricity while reducing dependence on fossil fuels. Geothermal energy is supposed to be one of the key players in the energy transition due to its cascade application and availability. Despite being omnipresent, geothermal energy remains a largely untapped resource because of the existing risks and complexity of the subsurface. Besides the high initial investment, exploration, and development risks are the main limitations related to geothermal resources. In spite of their high cost, geological models are imperfect representations of the subsurface due to inherent uncertainties. Reliable knowledge about the underground is necessary for a wide variety of applications. This thesis is a contribution to a more detailed understanding of the impact of structural uncertainties on the behaviour of geothermal reservoirs. Probabilistic structural models, rather than deterministic representations, are applied for further numerical analyses. Individual studies are developed to capture the relationship between geological features and simulation results.

However, incorporating stochastic structural models into numerical simulations necessitates robust and automated meshing techniques to bridge the gap between the geological model and the numerical solvers. Preserving the intricate geometry of the real underground environment is crucial for accurate simulations. While manual meshing offers complete control over complex geometries, its time-consuming nature requires the development of automated workflows. To account for the inherent uncertainty in these models, numerous simulations with diverse structural representations are required. This significantly increases the workload associated with manual mesh generation.

The first study of the thesis (Chapter 3) presents a synthetic show-case to quantify the impact of structural uncertainty on a tracer test. Different structural models for a simplified Enhanced Geothermal System (EGS) are generated via GemPy. A Python library, GeoMeshPy, is developed to process the outputs of the GemPy and return readable input for the mesh generator (Gmsh). Besides maintaining the geometry of the complex geological models, the mesh for as many as required scenarios is accomplished automatically in the Python ecosystem without any manual intervention. In EGS settings, fractures typically dominate the flow regime, hence their location is important. Dramatic changes are observed

in the tracer recovery of the EGS setting due to a slight change in the location of the smallest fracture. A two-week delay in peak arrival time, along with a 30 % reduction in peak value, constitutes strong evidence highlighting the importance of structural uncertainty. This uncertainty can significantly influence the expectations of tracer tests.

The impact of geological uncertainty on heat storage applications located in the vicinity of urban areas is evaluated in the second study of the thesis (Chapter 4). Solar and wind's mismatch with the energy demand necessitates large-scale storage. Geothermal plants also have summer surpluses. While Aquifer Thermal Energy Storage (ATES) boasts the highest storage capacity, the subsurface uncertainty remains a challenge. Chapter 4 uses an example from the Greater Geneva Basin to assess how sensitive temperature and water flow patterns are to the reservoir geometry. Even when varying the flat planes into rough surfaces by moving the reservoir boundaries ±10 m and ±15 m, the results remained unchanged. This insensitivity suggests that thermal regimes in thick reservoirs are localized and expensive 3D seismic exploration might be unnecessary in some cases. In another example (DeepStor), a hypothetical sealing fault is introduced to the sand layers of the Oligocene Meletta beds, in the Upper Rhine Graben (URG), to delimit the 10 m thick reservoir. The hypothetical fault is relocated 16 times, i.e. different spacing from the injection well, in the geological model and proved to increase the pressure value up to 7 % in comparison to fault-free (base case) realization. The impact of the fault has been even less significant on the thermal recovery (c. 1.5 %).

Chapter 3 explores the impact of structural uncertainty using 50 scenarios, highlighting the need for more simulations to reach a robust conclusion. In Chapter 5 of this thesis machine learning (ML) is leveraged to obtain a more comprehensive uncertainty quantification scheme. Three ML models are employed: decision tree regression (DTR), random forest regression (RFR), and gradient boosting regression (GBR). DTR and RFR predict the entire breakthrough curve (BTC) but risk overfitting or underfitting. GBR addresses this by predicting each time step as a separate target variable, capturing potential correlations. This approach is implemented as a chain of regression models. The chapter utilizes two examples. The first, similar to the simplified reservoir in Chapter 3, uses coordinates of the only perturbed fracture as input features for the ML model. The second example of this chapter simulated the tracer test in an EGS reservoir with a more complex fracture network. For the second example, the location and dipping angle of two fracture surfaces are perturbed. In both examples, the model predicts the BTC as a time series representing the simulated concentration of the tracer. While time series prediction is a well-established field, our model differs by not relying solely on historical data to predict the future. Unlike traditional methods, our ML model directly correlates the entire BTC with the input features, bypassing

the need for time windows in future predictions. The acceptable increase in RMSE from train to test data confirms the ability of the chain model to capture both the general trend and small-scale variations. In particular, using the ML model reduces computational time by six orders of magnitude compared to traditional solvers. This efficiency allows for calculating the BTC for a vast number of geological realizations.

The last chapter of the thesis (Chapter 6) addresses the possible risks concerning the operational style of a classic geothermal doublet and a storage triplet for the faulted Malm aquifer in the north of Munich. The possibility of fault reactivation and changes in operation-induced displacement at the borehole location is compared within two proposed concepts. Due to the high transmissivity of the reservoir, the hydraulic impact of the operation is negligible and the thermal signal in the underground has been detected to be the dominant factor in defining the dynamic behaviour of the reservoir. In the storage triplet, the perturbed area around the cold well is larger than the hot well. For the geothermal doublet, the area is even bigger because of the continuous cold fluid injection at the injection well. This effect directly increases the stressed area and subsequently the displacement around the well and the slip tendency of the fault surface. Therefore, the storage concept can cover the seasonal mismatch between heat production from the geothermal doublet and also operate with a lower risk of fault reactivation. For the storage concept also different well locations are tested to see the sensitivity of the fault reactivation to the well placement (especially the cold one). A distance farther than 145 m can guarantee a safe storage operation for at least 10 years in the Malm aquifer.

Abstract

Erneuerbare Energiequellen sind die Zukunft der nachhaltigen Energieerzeugung, die Wärme und Strom liefern und gleichzeitig die Abhängigkeit von fossilen Brennstoffen verringern. Die Geothermie soll aufgrund ihrer Kaskadenanwendung und Verfügbarkeit einer der Hauptakteure der Energiewende sein. Obwohl sie allgegenwärtig ist, bleibt die geothermische Energie aufgrund der bestehenden Risiken und der Komplexität des Untergrunds eine weitgehend ungenutzte Ressource. Neben den hohen Anfangsinvestitionen sind Explorations- und Erschließungsrisiken die wichtigsten Einschränkungen im Zusammenhang mit geothermischen Ressourcen. Trotz ihrer hohen Kosten sind geologische Modelle aufgrund der ihnen innewohnenden Unsicherheiten nur unvollkommene Darstellungen des Untergrunds. Verlässliche Kenntnisse über den Untergrund sind für eine Vielzahl von Anwendungen notwendig. Diese Arbeit ist ein Beitrag zu einem genaueren Verständnis der Auswirkungen struktureller Unsicherheiten auf das Verhalten geothermischer Reservoire. Für weitere numerische Analysen werden probabilistische Strukturmodelle anstelle von deterministischen Darstellungen verwendet. Es werden individuelle Studien entwickelt, um die Beziehung zwischen geologischen Merkmalen und Simulationsergebnissen zu erfassen.

Die Einbeziehung stochastischer Strukturmodelle in numerische Simulationen erfordert jedoch robuste und automatisierte Meshing-Verfahren, um die Lücke zwischen dem geologischen Modell und den numerischen Solvern zu schließen. Die Beibehaltung der komplizierten Geometrie der realen unterirdischen Umgebung ist für genaue Simulationen entscheidend. Die manuelle Vernetzung bietet zwar die vollständige Kontrolle über komplexe Geometrien, ist aber so zeitaufwändig, dass die Entwicklung automatisierter Arbeitsabläufe erforderlich ist. Um die inhärenten Unsicherheiten in diesen Modellen zu berücksichtigen, sind zahlreiche Simulationen mit unterschiedlichen strukturellen Darstellungen erforderlich. Dies erhöht den mit der manuellen Mesh-Generierung verbundenen Arbeitsaufwand erheblich.

In der ersten Studie dieser Arbeit (Kapitel 3) wird ein synthetischer Fall vorgestellt, um die Auswirkungen der strukturellen Unsicherheit auf einen Tracerversuch zu quantifizieren. Verschiedene Strukturmodelle für ein vereinfachtes Enhanced Geothermal System (EGS) werden mit GemPy generiert. Eine Python-Bibliothek mit dem Namen GeoMeshPy verarbeitet die Ausgaben von GemPy und liefert lesbaren Input für den Mesh-Generator (Gmsh).

Neben der Beibehaltung der Geometrie der komplexen geologischen Modelle wird das Netz für beliebig viele Szenarien automatisch im Python-System ohne manuelle Eingriffe erstellt. In EGS-Umgebungen dominieren typischerweise Klüfte das Strömungsregime, daher ist ihre Lage wichtig. Dramatische Veränderungen werden bei der Tracer-Gewinnung in EGS-Umgebungen aufgrund einer geringfügigen Änderung der Lage des kleinsten Bruches beobachtet. Eine zweiwöchige Verzögerung der Ankunftszeit des Spitzenwerts zusammen mit einer Verringerung des Spitzenwerts um 30 % ist ein deutlicher Hinweis auf die Bedeutung der strukturellen Unsicherheit. Diese Unsicherheit kann die Erwartungen an die Tracerversuche erheblich beeinflussen.

In der zweiten Studie dieser Arbeit (Kapitel 4) werden die Auswirkungen der geologischen Unsicherheit bei Wärmespeicheranwendungen in der Nähe von städtischen Gebieten bewertet. Das Ungleichgewicht zwischen Solar/Windenergie und der Energienachfrage macht eine Speicherung in großem Maßstab erforderlich. Auch geothermische Anlagen haben im Sommer Überschüsse. Aquifer Thermal Energy Storage (ATES) bietet zwar die höchste Speicherkapazität, aber die Unsicherheit im Untergrund bleibt eine Herausforderung. In Kapitel 4 wird anhand eines Beispiels aus dem Greater Geneva Basin untersucht, wie empfindlich Temperatur- und Strömungsmuster auf die Oberflächenform des Reservoirs reagieren. Selbst wenn man die flachen Ebenen in raue Oberflächen umwandelt, indem man die Grenzen des Reservoirs um ±10 m und ±15 m verschiebt, bleiben die Ergebnisse unverändert. Diese Unempfindlichkeit deutet darauf hin, dass die thermischen Verhältnisse in dicken Lagerstätten lokalisiert sind und dass teure seismische 3D-Untersuchungen in einigen Fällen unnötig sind. In einem anderen Beispiel (DeepStor) wird eine hypothetische abdichtende Störung in die Sandschichten der Oligocene Meletta-Schichten eingeführt, um die 10 m dicke Lagerstätte abzugrenzen. Die hypothetische Störung wurde im geologischen Modell 16-mal verlegt, d. h. in verschiedenen Abständen von der Injektionsbohrung, und es hat sich gezeigt, dass sie den Druckwert im Vergleich zu einer Störungsfreien (Basisfall) Realisierung nur um bis zu 7 % erhöht. Der Einfluss der Störung auf die thermische Ausbeute war sogar noch unbedeutender (ca. 1,5 %).

In Kapitel 3 werden die Auswirkungen der strukturellen Unsicherheit anhand von 50 Szenarien untersucht, wobei deutlich wird, dass für solide Schlussfolgerungen mehr Simulationen erforderlich sind. Kapitel 5 dieser Arbeit nutzt maschinelles Lernen (ML) für ein robusteres Quantifizierungsschema der Unsicherheit. Es werden drei Modelle verwendet: decision tree regression (DTR), Random-Forest-Regression (RFR) und Gradient-Boosting-Regression (GBR). DTR und RFR sagen die gesamte Durchbruchskurve (BTC) voraus, riskieren aber eine Über- oder Unteranpassung. GBR löst dieses Problem, indem es jeden Zeitschritt als separate Zielvariable vorhersagt und so potenzielle Korrelationen erfasst. Dieser Ansatz wird als eine Kette von Regressionsmodellen implementiert. In diesem Kapitel werden zwei Beispiele verwendet. Im ersten Beispiel, das dem vereinfachten Reservoir in Kapitel 3 ähnelt, werden die Koordinaten der einzigen gestörten Störungen als Eingangsmerkmale für das ML-Modell verwendet. Das zweite Beispiel dieses Kapitels simuliert den Tracertest in einem EGS-Reservoir mit einem komplexeren Kluftnetzwerk. Im zweiten Beispiel werden die Lage und der Neigungswinkel von zwei Bruchflächen gestört. In beiden Beispielen sagt das Modell die Tracer-Durchbruchskurve (BTC) voraus, eine Zeitreihe, die die simulierte Konzentration des Tracers darstellt. Während die Zeitreihenvorhersage ein gut etabliertes Feld ist, unterscheidet sich unser Modell dadurch, dass es sich nicht ausschließlich auf historische Daten stützt, um die Zukunft vorherzusagen. Im Gegensatz zu herkömmlichen Methoden korreliert unser ML-Modell direkt die gesamte BTC mit den Eingangsmerkmalen und umgeht so die Notwendigkeit von Zeitfenstern für zukünftige Vorhersagen. Der akzeptable Anstieg des RMSE von den Trainings- zu den Testdaten bestätigt die Fähigkeit des Kettenmodells, sowohl den allgemeinen Trend als auch kleinräumige Schwankungen zu erfassen. Besonders hervorzuheben ist, dass die Störung des ML-Modells die Berechnungszeit im Vergleich zu herkömmlichen Solvern um sechs Größenordnungen reduziert. Diese Effizienz ermöglicht die Berechnung der BTC für eine große Anzahl von geologischen Realisierungen.

Das letzte Kapitel der Arbeit (Kapitel 6) befasst sich mit den möglichen Risiken hinsichtlich der Betriebsweise Eines klassischen geothermischen Doublets und eines Speichertripletts für den gestörten Malm-Aquifer im Norden von München. Die Möglichkeit der Störungsreaktivierung und die Veränderungen der betriebsbedingt ausgelösten Verschiebungen am Bohrlochstandort werden innerhalb der beiden vorgeschlagenen Konzepte verglichen. Aufgrund der hohen Transmissivität des Reservoirs ist der hydraulische Einfluss des Betriebs vernachlässigbar und das thermische Signal im Untergrund wurde als dominierender Faktor für das dynamische Verhalten des Reservoirs ermittelt. Bei dem Speicher-Triplett ist der gestörte Bereich um die kalte Bohrung größer als um die heiße Bohrung. Bei dem geothermischen Doublet ist der Bereich sogar noch größer, da an der Injektionsbohrung kontinuierlich kaltes Fluid reingepumpt wird. Dieser Effekt vergrößert direkt die beanspruchte Fläche und in der Folge die Verschiebung um die Bohrung und die Rutschtendenz der Bruchfläche. Daher kann das Speicherkonzept die jahreszeitliche zwischen der Wärmeproduktion des geothermischen Doublets ausgleichen und auch mit einem geringeren Risiko der Störungsreaktivierung arbeiten. Für das Speicherkonzept werden auch verschiedene Bohrungsstandorte getestet, um zu sehen, wie empfindlich die Störungsreaktivierung auf die Bohrungsplazierung (insbesondere die Kalte) reagiert. Ein Abstand von mehr als 150 m kann im Malm-Aquifer einen sicheren Speicherbetrieb für mindestens 10 Jahre gewährleisten.

Acknowledgements

Lots of people supported me during my PhD and its my pleasure to appreciate their contribution to my work. I would like to thank all my colleagues working at Geothermal Energy and Reservoir Technology department. I enjoyed a lot working with each of you. Lunch plus coffee breaks and also our weekly beer drinking events were always joyful for me. Before mentioning the name of some specific people, I should apologize from anyone whose name is missed here.

First, I would like to appreciate Thomas Kohl for his supervision and support. I must appreciate him wholeheartedly because his vast knowledge and wisdom which have always been a source of inspiration for me.

To start my PhD, Thomas suggested me to find a financial support for our ideas and the German Academic Exchange Service (Deutscher Akademischer Austauschdienst: DAAD) agreed to cover my expenses for a PhD in Germany as the Research Grants- Doctoral program in Germany 2019/20. I appreciate this organization for giving the opportunity to researchers. Similar to me, millions of other researchers have been able to follow their dreams thanks to DAAD as the world's largest funding organisation for the international exchange of students and researchers.

Pleasure is all mine to express the deepest gratitude to Dr. Maziar Gholami-Korzani who supervised me in the start of my PhD. His support helped me to find my way for shaping up my research.

Many thanks to Christophe Geuzaine from the University of Liège for his invaluable developments in Gmsh. Christophe's help saved a huge amount of time for me. Looking into the history of the commits in the Gmsh GitLab repo can prove how much time a highly prestigious professor like Christophe Geuzaine is dedicating to open source.

I appreciate every committee member of my PhD. Thanks for dedicating time to my work. Your comments will be a highly valuable source of information for me to improve the thesis.

A huge thanks to Silke because of her continuous and unconditional support with administrative issues related to the (well-known) KIT bureaucracy.

Last but not least, I would like to thank my wife (Nasim) who was always with me throughout the long journey of my PhD. Dear Nasim, thanks for your patience.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

Since the Kyoto Protocol launched in 1997, there has been a growing push to reduce greenhouse gas (GHG) emissions across industries, particularly in energy. This force is driven by the growing demand for cleaner energy sources, as evidenced in recent United Nations (UN) Climate Change Conferences (COP26 and COP27) [1]. A notable share of the current proved resources of fossil fuels must remain untouched until 2050 to limit the GHG production [2–4]. Climate change is known to be the greatest environmental threat to the planet. Meanwhile, anthropogenic GHG emissions are the driving force for global warming. Carbon dioxide (CO_2) emissions account for approximately 80% of the total anthropogenic GHG and showed a direct impact on temperature rise. Therefore, policymaking and target settings must be based on the available CO_2 emission data [5]. The main share of CO_2 emissions stems from energy consumption [6]. An obvious steady rise in the level of CO_2 in the atmosphere exists. Because of the COVID-19 pandemic and consequent limitations on outdoor activities, a 5.8% reduction of the CO_2 emissions in 2020 has resulted. However, the trend started to rise again in 2022. The curve plotted in Fig. 1.1 shows the trend of CO_2 emission from 2000 to now [7]. The green shaded area shows the range of pathways that fulfill the requirement of the 1.5°C increase in the global temperature as requested by the Intergovernmental Panel on Climate Change (IPCC) [8]. A net zero CO_2 in 2050 is requested by IPCC [8]. Following the 2021 Glasgow meeting, countries formulated this 2050 net zero pathway.

The objective for holding the global temperature rise to 1.5°C above pre-industrial levels requires a balance between GHG production and removal. Such a goal can only be achieved by fast emissions reductions across any energy consuming sectors [9]. Activities like consuming fossil fuels and deforestation significantly increase the atmospheric concentrations of GHGs and hinder achieving net zero 2050. Meanwhile, accurate monitoring of global emissions, particularly at the national level, is challenging due to weak reporting and accountability mechanisms. Significant informational gaps exist when it comes to tracking emissions

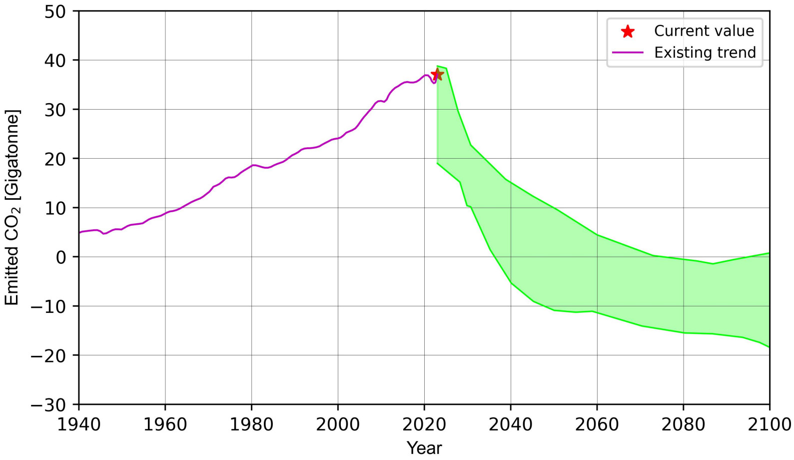

Fig. 1.1 The trend of the CO_2 emission and possible ranges of solutions that allow for following the 1.5°C increase of the global temperature [8].

from sectors such as military activities, shipping, and aviation, especially from less developed countries. In light of net zero 2050, human beings all over the world must increasingly incorporate carbon neutrality targets in domestic and international laws to undertake carbon neutrality commitments.

In 2022, the Earth had already exceeded 1.2°C of the average global temperature since 1870 and consumed the carbon budget of the next five years to limit the global temperature increase to 1.5°C. Since the start of the global record of temperature in 1880, summer 2023 proved to be the hottest summer period, i.e. June, July, and August, based on the data released by NASA [10]. The statistics from the mentioned source confirmed that summer 2023 has been 0.23°C warmer than previous summers.

In case of following this drastic trend, worldwide catastrophic events can inevitably happen [11]. Experts caution that the potential collapse of the Greenland and West Antarctic ice sheets will become a real possibility at the 1.5°C threshold, along with the risk of reaching other critical tipping points even before global heating reaches 2°C [12]. In recent decades, there has been a notable rise in the frequency of natural disasters like urban flooding linked to catastrophic precipitations, devastating heat waves, extreme winds, frequent drought periods, glacial melting, and so on [13, 14]. As an example, marine ecosystems as the largest producer of oxygen (O_2) and the largest absorber of CO_2 can be severely endangered by global warming [15]. Such effects are interconnected and can create a cascade of impacts that influence multiple aspects of the Earth's systems [16]. Another study revealed a strong direct relation between maximum daily temperature and the frequency of violent crimes occurring outdoors [17]. Therefore, the rising temperatures are not only associated with wildfires but also with the incidence of aggressive conflicts in the urban context.

In response to the concerning increase in CO_2 emissions, positive actions have been implemented since 2015. The trend of investment in resources is shown in Fig. 1.2. A report by the IEA [7] says that countries are putting more money into clean energy sources than fossil fuels. This is because clean energy is becoming cheaper and crises such as the war in Ukraine has made people worried about getting enough oil and gas. Clean energy investment is growing rapidly (24% increase expected between 2021 and 2023) but still lags behind the total investment in fossil fuels, which have a long history in the energy sector. Even though fossil fuels have downsides, there are challenges (technical and political) to overcome before we can fully rely on clean energy sources. Fossil fuels pack a lot of energy for their price. In the case of comparing the output energy to the energy used to find and extract fossil fuels, they are almost cheap [18]. Fortunately, the cost of extracting fossil fuels is growing, while renewable energy sources are becoming more affordable [19]. Renewable energy has massive potential, however it's not yet readily available to meet all the needs of society. Developing efficient systems that combine different renewable sources could be key to making the switch to clean energy affordable and realistic [20]. In order to establish an eco-environment, the transition of the primary energy source from fossil fuels to clean, and renewable ones is imperative. Several factors can boost a country's use of renewable energy sources. These include strong democratic institutions, effective governance, high oil prices (which can incentivize a switch from oil), and a wealthier population [21]. Consequently, an escalated adoption of renewable energy over time diminishes air pollution and aids in ensuring a sustainable energy supply for future energy demands.

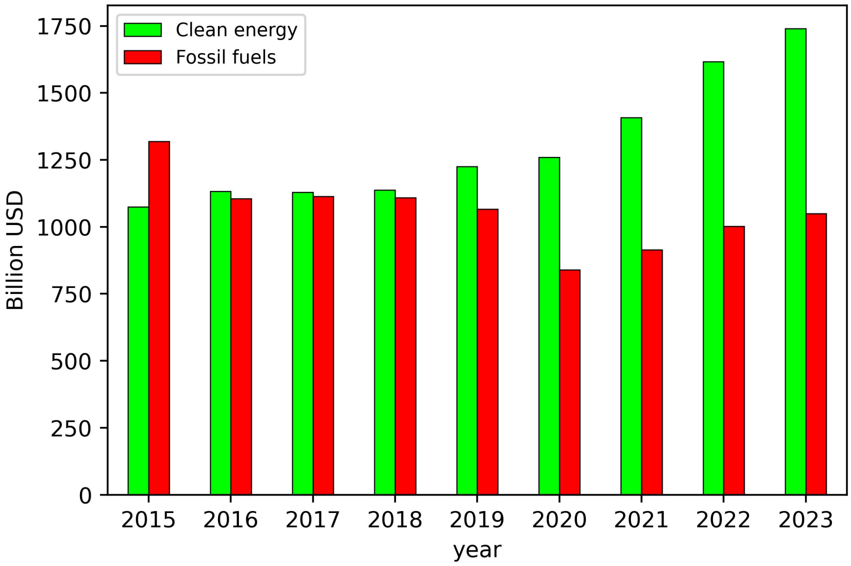

Fig. 1.2 Changing trend in the investments for clean sources of energy versus fossil fuels [7].

Renewable energy resources are supposed to be the main contributor to the net zero 2050 roadmap [22]. International Energy Agency (IEA) defines "renewable energy" as "energy that is derived from natural processes that are replenished constantly". In order to attain carbon neutrality, the global proportion of renewable energy is anticipated to surge from 14% in 2018 to approximately 74% by 2050, necessitating a continuous annual escalation. The establishment of renewable energy sources by 2030, at an annual cost of $4 trillion, is imperative to accomplish net zero emissions by 2050. The initial expenditure may pose a challenge for many countries, prompting the need for financial and technical support to facilitate the energy transition. However, the achievement of net zero carbon emissions can yield profitability for renewable energy investments. Furthermore, renewable energy exhibits greater resilience to market disruptions and enhances energy security for each nation. Recognizing the significance of widespread implementation of renewable energy across all countries (to reduce dependence on fossil fuels) will assist policymakers in advocating the utilization of such environmentally friendly sources [23]. Costa Rica, as a developing country, has become a world champion for using renewable energy sources. Their switch to clean energy is a great example of how a country can successfully move towards more environmentally friendly energy resources [24].

According to existing literature, renewable energy technologies can be categorized as follows. Firstly, there is solar energy, also referred to as photovoltaic energy [25]. Secondly, wind energy involves harnessing wind power to generate electricity [26]. While wind and solar energies are considered clean sources of energy, it is important to note that they are reliant on weather conditions and critical mining resources. The third category is geothermal energy, which is derived from the Earth's heat [25]. Geothermal energy has the potential to provide both electricity and heat on an industrial scale. The fourth renewable energy supplier is hydropower, which involves converting the potential energy of water into kinetic energy. Hydroelectricity is generated by constructing dams on rivers, where the water's higher altitude creates a force that drives turbines to generate electricity. Hydropower reigns supreme as the world's largest renewable energy source, boasting a global capacity of around 1150 gigawatts [27, 23]. Biomass energy, as the fifth category, comes from organic materials like wood, crop leftovers, animal manure, and other plant and animal sources. Biomass energy is produced by burning these materials. These wastes can be incinerated or undergo anaerobic digestion to produce heat and/or electricity [27].

Geothermal energy boasts a long history as one of the first renewable resources harnessed by humans. Its applications are incredibly diverse, from keeping us warm (heating buildings and pools) to aiding in agriculture (drying crops and heating greenhouses). It can even be used to create comfortable spaces for fish farming (aquaculture) [28]. The indirect application involves harnessing the extracted heat to generate electricity. The adoption of geothermal resources for direct utilization has seen substantial growth [29]. With the growing energy needs, geothermal power is crucial for both electricity and heating. The UN acknowledges geothermal as a frontrunner among renewable resources. Its potential is enormous, estimated at five times that of solar, wind, biomass, and hydro combined. Geothermal could provide 5000 EJ/yr, compared to solar's 1575 EJ/yr, wind's 640 EJ/yr, biomass's 276 EJ/yr, and hydro's 50 EJ/yr [30]. Geothermal energy stands out among renewables for its reliable power generation. Studies show it has a higher capacity factor (actual compared to potential output) than other sources [31]. This makes it a strong supplier to meet the future energy demands of our society. Projections estimate geothermal power generation will reach 25'000 megawatts by 2030 [32], potentially reducing the environmental harm caused by fossil fuels. Unlike wind and solar, geothermal isn't affected by weather or time, adding to its appeal as a clean base load. As an insightful example presented by Gutiérrez-Negrín (2024) [32], a 10 MW geothermal power plant with a 90% annual availability slightly outperforms a 28 MW onshore wind farm operating with a 32% annual availability. With these advantages, geothermal is becoming a key player in the fight for a low-carbon future.

1.1 Motivation

Looking up, even the moon with a 384'400 km distance can be seen. However, looking down can only provide us with direct observations from the surface of the Earth. Derived interpretations for the underground come out of indirect and error-prone measurements. Every geological data set has limitations in terms of spatiality and resolution. Meanwhile, such data are (objectively) interpreted which enhances the uncertainty of the gained knowledge. As we face an extreme lack of direct observation and omnipresent uncertainty, we should be highly cautious in any underground engineering or exploitation plan. Assessing the uncertainty of the underground geometry can be even more important than assessing uncertainty on rock properties because the material property distribution also depends on the geometry [33]. In the first step of every planning for underground applications, the possible geometry should be defined. The thickness of the formations can make a huge difference for volume-dependent applications like hydrocarbon production or hot fluids storage. The importance of errors increases for geography-dependent applications e.g. influence of a fracture's location on flow. Structural uncertainties have often been recognized as the most important source of uncertainty [34]. Geological 3-D models are constructed to represent the underground. The reliability of a model stands tightly on the accuracy of the input data and as mentioned is sensitive to uncertainty. As a famous quote by George E. P. Box:

"Essentially, all models are wrong, but some are useful."

The sources of uncertainties are manifold and interconnected, e.g. measurement errors, biased interpretations, sparse data, and mathematical formulations. Developing multiple geological realizations in a scenario modelling approach can be considered the most widely used approach to include the structural uncertainty of the conceptual geological models [35]. An intuitive example presented by Bond et al. (2010) [36] confirms that relying on the best model is equivalent to hiring a novice geoscientist while an expert can present multiple realizations.

My planned work in this PhD thesis adds a random source of uncertainty to the input data to evaluate its impact on the numerical simulation. At first sight, random noise may look irrelevant for real cases, but the presented chapters will clarify how the most simplistic sources of structural uncertainty can contribute to geothermal applications. Inevitable uncertainty of the subsurface geometry that may result in low heat extraction rates is one of the major risks for geothermal projects and this thesis covers some possible uncertainties. Fig. 1.3 shows the trend of the developed workflow. After generating each structural model, it should be prepared in a way to be readable by numerical solvers which can take a considerable amount of time. The color pattern in Fig. 1.3 can be assumed as simulated conduction or fluid flow in the underground. The permeable part of the model is the reservoir that is sandwiched between the upper and lower units. The same physics should be applied to isolate the impact structures, otherwise, the result can be a mixture of structural and parametric variations.

A valid reason for ignoring structural uncertainty is the tyranny of the meshing process. Most numerical solvers can yield reliable results only if we solve the problem in a domain that is well discretized. The 3-D geological models can act as frameworks for numerical simulations but their space must be discretized via meshing before being fed into the solvers. The meshing process is the bottleneck for structural uncertainty analyses. It requires a huge amount of time to prepare a water-tight mesh even for only a single geological scenario. Perturbing the model and making the mesh for each realization will of course add up to the cumbersomeness of the meshing process. A very typical uncertainty study or in a better word a sensitivity analysis establishes one single geological model, makes a mesh for it, and tries different material properties. However, in this study, different meshes are generated and imported to the numerical solvers to simulate the underground processes and see how the change in the geometry can modify the distribution of the temperature, pressure, stress, and solute fields.

Variations in geological model geometry, as represented by different meshes, can significantly influence the distribution of the variables in numerical simulations. This hypothesis is grounded in the idea that structural uncertainty (changes in geometry) plays a critical role in subsurface modelling, and it seeks to test this by generating and analyzing different meshes for numerical solvers.

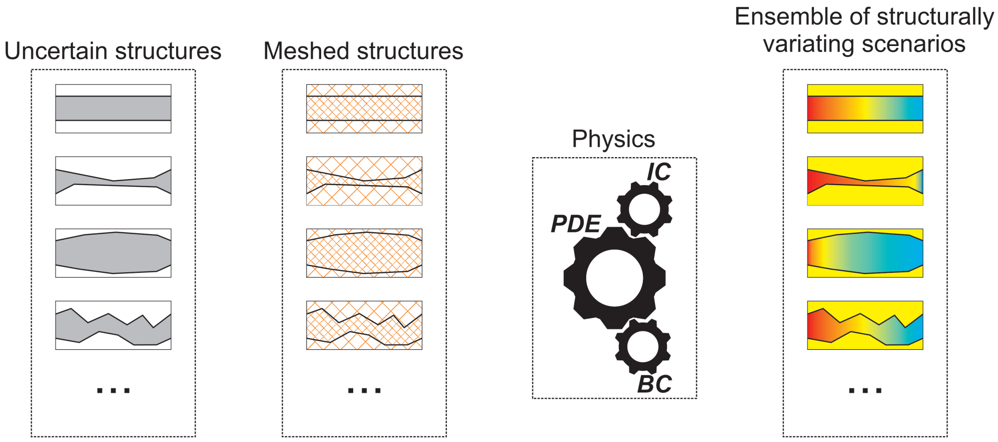

Fig. 1.3 A schematic of the developed workflow in this PhD thesis. The mesh should resemble the initial model that allows the projection of the different structures in the ensemble of simulations.

Thanks to the development provided by researchers from RWTH Aachen (Prof. Florian Wellmann and his group), making several geological realizations is possible via a tool named GemPy [37]. However, to include different geological realizations in simulations, it is impossible to make mesh for each realization manually. GemPy can generate a large number of geological realizations very fast and efficiently but its output should be adjusted and meshed. Developments done in this thesis allow for converting outputs of GemPy (as many as required realizations) automatically to readable inputs for a mesh generator called Gmsh [38]. A converter between GemPy and Gmsh is developed in the frame of this PhD that allows for including stochastic geological models in numerical simulations.

1.2 Thesis structure

Chapters in this thesis are arranged in a way to show the importance of geological modelling and uncertainty analysis. Different geothermal applications are subjected to structural uncertainty and their responses are analyzed here. Including the geological uncertainties in the numerical simulations and assessing their impact has been my main goal in this thesis.

In Chapter 2, a general overview of how geological models are constructed and can contain uncertainty is presented. Outputs of typical geological modeling tools are introduced in this chapter and required adjustments for preparing them as readable inputs for the meshing process are elaborated. A geological modeling tool called GemPy is introduced briefly and functionalities of the recently developed Python library, GeoMeshPy, are presented showing how outputs of GemPy can be used in a mesh generator called Gmsh. The new surface interpolation function of Gmsh is also introduced in this chapter. Several visualizations are provided in this chapter to deliver the concepts behind the automation of the mesh generation and surface interpolation of Gmsh.

In Chapter 3 one of the most important geothermal applications is designed to test and quantify the impact of structural uncertainty. An understanding of the underground flow regime in the medium is a pivotal requirement in any long-term subsurface application. Fluids can move through both the matrix and fracture but in most cases, the latter one dominates the flow. Therefore, fracture zones are of high importance for designing exploitation concepts like geothermal reservoirs. A synthetic tracer test is designed in a simplified enhanced geothermal systems (EGS) to evaluate the flow condition. This chapter deciphers the high sensitivity of tracer flow to the perturbation of structural elements of the geological model. Experts usually have a base idea or knowledge about underground structural situations and then analyze available data on the basis of their understanding. These possibly varying geological scenarios should be considered to design a reliable and optimum tracer test. Each model is discretized exactly based on its geological features (layers and faults) before being fed into the numerical solver. Therefore, geology has a twofold application: making the framework for numerical modelling and being used to interpret available data. Tracer measurements are typically fed into analytical solvers to extract some parameters from the medium that the tracer has passed through. The expert may consider a fixed length for a conduit and then try to back-calculate the transmissivity of the conduit based on the tracer recovery curve. The transmissivity value will be tuned in a way that fits the analytical solution to the measured data, independent of the length or shape of the conduit. The designed model in this study contains only three faults but still can be a perfect unsolvable enigma for analytical solvers. This chapter is an already published paper by Dashti et al. (2023) [39] in Geothermics. As the first and corresponding author of this publication, I have contributed to conceptualization, methodology, simulation, validation, code development; writing– original draft, and editing.

In Chapter 4 impacts of the structural uncertainty on the high-temperature heat storage applications in urban areas are quantified. The success of an HT-ATES design strictly depends on the underground characteristics while surface facilities can be engineered in a way to increase the efficiency. Therefore, a visualization (both conceptual and numerical) is necessary for consolidating an energy supply roadmap. In the case of conceptual models, underground reservoirs are usually considered as uniform huge boxes at km scale. Available studies are more dedicated to documenting the sensitivity of the system to operational and petrophysical parameters. The reason behind such strategies is that deep heat storage studies still follow the shallow (low temperature) storage literature while ignoring the real geological setting in depths > 1 km. Two showcases are introduced in this chapter. The Geneva case shows how numerical simulations can help to design a more efficient exploration campaign. The top and bottom surfaces of the reservoir are perturbed in this model and the thermohydraulic performance of the different scenarios are evaluated to check for the sensitivity of the storage system to surface perturbations. The other storage case is the DeepStor where depleted sand layers of Meletta beds in the Upper Rhine Graben are considered to act as the main reservoir for heat storage. An unpredicted sealing sub-seismic fault can be one of the worst cases that can happen to a storage project in this thin sand layer. Hence, a synthetic sealing fault has been added to the DeepStor model to assess its impact on temperature and pressure fields. This part of the work also considers the real geometry of the reservoir layer and introduces a synthetic sealing fault to the storage model to forecast the behaviour of the reservoir. In the DeepStor showcase a predictor is formulated to forecast the underground geology based on the measured pressure value during the testing phase of the borehole. Chapter 4 of this thesis is another published paper by Dashti et al. (2024) [40] in Geoscientific Model Development. As the first and corresponding author of this publication, I have contributed to conceptualization, methodology, simulation, validation, code development; writing– original draft, and editing.

In Chapter 5 I extend the idea of uncertainty quantification by deploying ML methods. This study again focuses on embedding structural uncertainties into flow characterization. In Chapter 3, only 50 numerically simulated structural realizations are introduced to investigate the impact of structural geology on underground tracer flow. In each realization, the position of one structural element in the model was perturbed to observe the sensitivity of tracer flow to the structural model. This chapter contains the results related to ML models that enable a more comprehensive uncertainty quantification. The target ML model is trained to map geometric data to tracer concentration. The position of the variating structural elements is used as the input features of the ML model. Specifically, the fault is represented as a planar surface, and the coordinates of its four corners are applied as input features. Then, the entire tracer recovery curve is chosen as the target variable. The target is to replace the computationally demanding solver with speedy ML models that promote uncertainty quantification dramatically. State-of-the-art ML methods like decision tree regression (DTR), random forest regression (RFR), and gradient boosting regression (GBR) allowed for generating thousands of tracer recovery curves rather than just 50. This chapter is an already published paper by Dashti et al. (2024) [41] in Geothermics. As the first and corresponding author of this publication, I have contributed to conceptualization, methodology, simulation, validation, code development; writing– original draft, and editing.

Chapter 6 presents the ongoing research collaboration between KIT and Stadtwerke Munich on the design of an optimum operational style in the Malm aquifer. Fully coupled thermo-hydro-mechanical (THM) models for two different operational styles including a classic geothermal doublet and a high temperature storage triplet are developed. In this chapter, I evaluate the (geo)mechanical risks related to the operation of a classic geothermal doublet and a storage triplet in the faulted Malm aquifer. While the 530-m thick reservoir geometry is simplistic in this study, a complex and realistic fault surface is embedded into the model to evaluate the possible risk of reactivation. The complex fault surface is meshed successfully and shows a wide variety of slip tendency changes due to its dip and strike. While thermohydraulic performance of the two concepts can be almost similar, the mechanical behaviour is expected to vary in both the near and far field properties of the model. This impact results in different displacement and slip tendency variations merely caused by two different operational styles. After finding the concept with less fault reactivation risk, a sensitivity analysis was also carried out to find out the possible relationship between the well locations and a fault reactivation potential. This chapter is going to be submitted to a scientific journal and I will be the first and also corresponding author of it.

Chapter 2

Automating the data flow from structural models to numerical solvers

This chapter delves into the 3-D subsurface structural models and the inherent uncertainties they entail. These models serve as frameworks for conducting sophisticated simulations in underground environments. However, the available data collected from the subsurface are scarce and susceptible to uncertainties. In this PhD, I begin by incorporating complex structural models into the simulations and subsequently introduce uncertainties to the initially deterministic model. Different geothermal applications are designed to evaluate the impact of the geological model on the underlying processes.

2.1 3-D modelling

Geometrically, a geological model consists of surfaces (in 3-D) or lines (in 2-D) that bound volumes or surfaces. Different processes (like creation, deposition, transportation, erosion, alteration, etc.) happen during various spans of time (up to millions of years) to achieve such geometries within the Earth. Structural models represent the steady-state geometry of the underground system [42]. Each surface or line of the model stems from different geological events. Therefore, it is of high importance to generate models carefully to avoid artifacts that do not match nature. The 3-D models are constructed in a way to be geometrically correct (as smooth and accurate surfaces as possible), topologically consistent (watertight and correct contacts), structurally coherent, and geologically realistic [43, 44].

2.1.1 Importance

Traditionally, geological models are shown by 2-D maps, e.g., well sections, geologic profiles, and plans, as what has been produced by William Smith and others about 200 years ago. However, a 3-D visualization can provide us with an intuitive feel and a higher degree of data consistency. Three-dimensional (3-D) geologic models are the best representation of the existing features at depth [45]. They are crucial in understanding the subsurface and are increasingly used as a basis in a wide variety of applications [46–48]. A reliable representation of the subsurface is vital to different spectra of human activities, ranging from hydrocarbon exploration and production to environmental engineering. The model can be designed in a way to answer various business questions but typically, building for fit-for-purpose models is the most efficient way [49]. Using the 3-D models, decision-makers can get a full insight into the subsurface using virtual cross-sections, allowing critical plans to be made before any real site exploitation. Numerical simulation of the underground behaviour has been routinely made on the basis of structural frameworks since the early 1970s to plan primary, secondary, or tertiary development in hydrocarbon fields [50]. Therefore, 3-D geological modelling especially in complex settings is an important research emphasis.

The following sections encompass the uncertainty, required data, interpolation techniques, and finally the output of the geological models.

2.1.2 Uncertainty

A geological model is constructed using both subjective interpretation and mathematical interpolation/extrapolation techniques to serve as a simplified representation of the complex, real-world natural system. Meanwhile, geological knowledge is gained based on different data sets that involve uncertainties stemming from the resolution, technical measurement errors, etc. Given the inherent limitations in our understanding of the natural system, multiple plausible geological models can be formulated. These models are characterized by their compatibility with available data and existing knowledge, making them valid within the constraints of our current understanding [35]. In applications where geological conditions significantly impact outcomes, uncertainties can introduce risks.

Structural uncertainty impacts a wide variety of decisions that should be made in different life spans (exploration, development, production, and drilling) of an underground application.

Therefore, merely constructing the most accurate geological model is insufficient; it is equally crucial to incorporate geological uncertainties into the analysis. Wellmann and Caumon (2018) [51] and Wellmann et al. (2010) [52] delved into different aspects of the geological uncertainty. Several methods are available in the literature to estimate and quantify uncertainty [52, 51, 53–55].

Thore et al. (2002) [55] presented a comprehensive approach to determine, quantify, and handle the structural uncertainties for the models generated by seismic data. They examined both the magnitude and direction of the different sources of uncertainty. They concluded that in the seismic processing chain usually only migration, interpretation (e.g., tying seismic markers and synthetic seismogram, picking horizons, fault detection, and building of a complete structural framework), and time-to-depth conversion uncertainties dominate. However, this fact may vary from case to case as other steps like acquisition can be very uncertainty-prone in land seismic surveys.

Cherpeau et al. (2010) [56] delved into the different approaches of embedding the uncertainty in the structural model: topological, explicit, and implicit geometrical perturbations. This work combined an implicit representation of the surfaces with a stochastic binary tree to represent the spatial regions to account for the uncertainty of the fault surfaces as they play a key role in subsurface flow.

Wellmann and Regenauer-Lieb (2012) [34] applied the concept of information entropy introduced by Shannon (1948) [57]. In their insightful study, they added new data and information, e.g. different geological hypotheses, to a 3-D model to observe the uncertainty change at each step. Information entropy has been their quantitative indicator of uncertainty.

The developed methodology in this PhD enables any source of realistic geological uncertainty. While the history of the discussions around uncertainty is very long [56, 51], incorporating them in numerical simulations has been procrastinated due to the difficulty of generating numerous structural models and meshing them.

Another reason for neglecting the structural uncertainty is that changes in the structural model imply changes in the facies and petrophysical models and in the mesh [58]. As highlighted in this thesis, automating the generation of the structural model and mesh is the only practical way to handle the uncertainty in 3-D simulations. Therefore, it is my contribution to generate uncertain scenarios in such a way that the structural model and the mesh are generated automatically for each realization.

Here the source of uncertainty remains generic and random to avoid any bias. The main reason is that incorporating the structural uncertainty in numerical simulations has been chosen as the main focus. The randomness of the method in cases results in highly complex structures that are successfully meshed and used as inputs for advanced numerical simulations.

2.1.3 Input data

Building geological models involves a wide variety of data. In a simplistic view, some depth measurements can be fed into an interpolation algorithm to extract surfaces. However, the importance of 3-D geological models resulted in new paradigms in data acquisition. All the existing 3-D geological models rely heavily on large numbers of measurements done on the surface, boreholes, and cross-sections. To create truly realistic and accurate models of the underground, geological expertise is needed to interpret the data and avoid creating misleading results, i.e. artifacts [59]. Geological knowledge connects all the discrete data points to present a more comprehensive picture. The knowledge may look redundant due to its qualitative nature but it can increase the reliability of outputs returned from mathematical algorithms. Raw data including geophysical and/or geological data are intrinsically ambiguous and sparse. Therefore, the most realistic models can be tailored via data coming from different sources accompanied by a previous geological interpretation. There are different geological modelling tools like Petrel, Leapfrog, OpendTect, GeoModeller, GemPy, etc. and each tool requires different data sets to construct a 3D geological model. For example, Petrel only requires point data representing the contact of two layers while in Geomodeller or GemPy the user must provide the orientation measurements besides at least two points of each existing surface in the model. Orientation data are defined by azimuth (angle deviation from north) and inclination (angle deviation from the horizontal plan). An example in the following will clarify how the orientation data can add geological knowledge to the results of the modelers.

Fig. 2.1-a shows the most naive way to generate underground surfaces in a 2-D example. It is assumed that only the depth value is recorded and the algorithm can generate the line (and surface in 3-D) without other constraints. The figure only shows five possible scenarios while an infinite number of mathematical outcomes can be achieved. As the next step, more geological data can decrease the number of possibilities, i.e. increase the reliability of the model. In Fig. 2.1-b orientation data (visualized as arrows) are also included and force the line to be perpendicular to them. This way, only two scenarios remain which can represent either a faulted setting or a gentle anticline and syncline fold. It is assumed that geological knowledge can be incorporated to add fault and/or fold structures to the measured data. As the next step, the properties of the fault can be concatenated to the input data to consider it as a stand-alone geological feature. In this study GemPy [37] is used for several reasons. Firstly, GemPy is an open-source Python code and the developing team is supportive of adding new features upon the requests from the community. Secondly, it is a well-optimized code for making uncertainty analysis and generating a myriad of geological realizations. Thirdly, the code allows to include geological knowledge in the modelling process. The relation between layers and faults in GemPy can be defined in a way that faults will only displace some specific layers, based on the user's choice. Readers are highly encouraged to see the available documentation on this useful code (https://github.com/gempy-project/gempy).

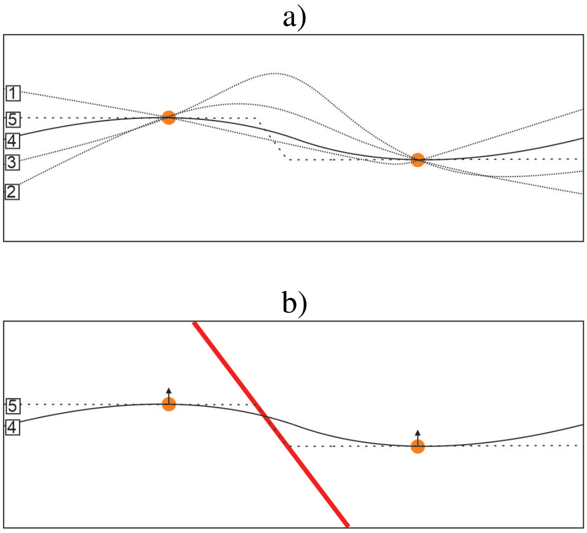

Fig. 2.1 a) Two orange circles represent measured depths for a geological surface. Five possible scenarios can be generated by connecting two points. b- Orientation data shown as arrows will force the modeler to only make lines that are perpendicular to the direction. This way only two scenarios remain and the correct one can be chosen based on the geological knowledge, e.g. existence of a fault can confirm scenario 5 while an observed fold structure in seismic data can support scenario 4.

2.1.4 Methods

Interpolation is a mathematical technique traditionally used to fill the gaps between scattered measurements. In the context of generating 3-D geological models, a variety of interpolation methods are deployed to create a continuous representation of the subsurface based on sparse 1-D data [60]. Each of the available tools for geological modelling uses its preferred algorithm. Methods like inverse-distance weighting interpolation [61], kriging [62], triangulation with linear interpolation [63], and cubic spline interpolation [64] are discussed with details in the literature. Herein only details of the kriging are explained as it is implemented in the GemPy that was used as the geological modeller in this study.

Kriging

Kriging is based on the principles of spatial autocorrelation stating that nearby locations are more likely to have similar values than those that are farther apart. Kriging leverages this concept to estimate values at unobserved locations by analyzing the spatial relationship between data points and assigning weights to them based on their proximity. In simple words, it determines the expected variances with respect to the distances between points rather than the absolute measured value at each point. The whole interpolation process in Kriging can be split into two parts elaborated in the following:

Variogram analysis

A variogram quantifies the relationship between differences in pairs of measurements and the distance of the pairs [65, 66]. The following equation generates the dissimilarity value between pairs:

$$\gamma(h) = \frac{1}{2N(h)} \sum_{i=1}^{N(h)} (Z(x_i) - Z(x_i + h))^2, \tag{2.1}$$

where $\gamma(h)$ represents the variogram value, $N(h)$ is the number of possible pairs, $Z(x_i) - Z(x_i + h)$ represents the difference between two values of a pair and h is the distance between two pairs known as the lag.

As a rule of thumb, if we repeat a measurement, the value should not change. For example, repeating the measurement of the concentration of one element in a fixed location should give the same results as the location has not changed. This rule introduces the idea behind the “nugget” value in variogram. Another general rule is that the similarity between points decreases with distances and after some distances (called “range”) the values of two points will get independent of each other. In other words, the pairs whose distance is more than the range are uncorrelated. Another parameter is called “sill” and represents the total variance contribution or the maximum variability between pairs of points. Variogram parameters are introduced in Fig. 2.2. After calculating the $\gamma(h)$ for all possible pairs of a data set, some functions are used to fit a function to the data points. This way, a theoretical/model variogram will be calculated from the experimental one. These functions typically use the h array and tune sill, nugget, and range parameters in a way to achieve the best fit between the experimental data and the theoretical/model variogram. In the following, the spherical model is presented:

$$\gamma = n + C_0 \times \left(1.5 \times \frac{h}{r} - 0.5 \times \left(\frac{h}{r}\right)^3\right), \tag{2.2}$$

where γ represents the theoretical/model variogram value, n is the nugget, C_0 is the sill, r is the range, and h is again the distance between two pairs known as the lag. In the spherical model the variogram value for the pairs whose distance is more than the range ($h > r$) is calculated merely based on the nugget and sill because this is where the γ values reach a plateau:

$$\gamma = b + C_0. \tag{2.3}$$

Other methods such as exponential, gaussian and so on are fully explored in the literature [67, 68].

Weight optimization

Up to now, variogram analysis provided us with a value-independent relation between data points. In fact, it captures the correlation between data based on their spatial relations rather than absolute values. A linear combination of the measured locations multiplied by some weights is the gist of estimation in Kriging:

$$\hat{Z}(x_0) = \sum_{i=1}^{N} \omega_i Z(x_i) + \varepsilon, \tag{2.4}$$

where $\hat{Z}(x_0)$ is the prediction at unknown location, ω is some weight values and ε is the error. The following equation is the simplest form to predict the weights:

$$A \times \omega = b. \tag{2.5}$$

where ω is again the unknown weight vector while A and b are known matrices and vectors. Then, the weight vector can be calculated by replacing the A and b with some spatial information. Weights can be later used in the estimator (Eq. 2.7). To do so, Kriging converts the generated theoretical variogram into a covariance. By flipping the diagram of the variogram, the covariogram curve is the result (see Fig. 2.2).

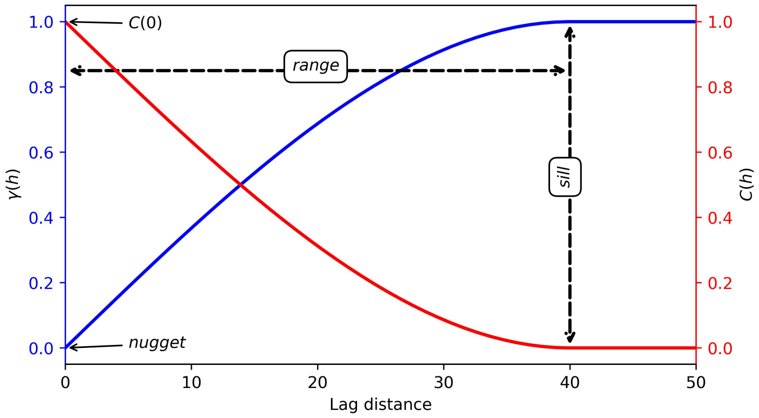

Fig. 2.2 The variogram plotted in blue measures the difference of pairs of points with respect to their distance. The covariance, plotted in red, is a metric to measure the similarity between pairs. Three main variogram parameters including sill, nugget, and range are annotated to grasp their concepts graphically.

Universal Co-Kriging

Co-Kriging solves for the weights of the primary and secondary variables using a system of linear equations, similar to Kriging, but with the added complexity of accounting for both spatial autocorrelation (within the primary variable) and cross-correlation (between the primary and secondary variables). In the case of GemPy, depth, and orientation data are used to build the matrices of the Co-Kriging method. Universal Co-Kriging is an extension of Co-Kriging that accounts for the presence of a trend (non-stationarity) in the primary and/or secondary variables. While simple (Co-)Kriging assumes that the mean of the variables is constant (stationary), Universal Co-Kriging incorporates a drift or spatially varying trend that may depend on known explanatory variables (e.g., elevation, distance to a source, etc.). Here a brief introduction is presented and for more details see [69, 70, 37]

Similar to Kriging, Co-Kriging predicts an unknown value at an unknown location ($\hat{Z}_1(x_0)$) by using a weighted combination of the known values of both the primary and secondary variables. The equation for Co-Kriging is:

$$\hat{Z}_1(x_0) = \sum_{i=1}^{n_1} \omega_i^{(1)} Z_1(x_i) + \sum_{i=1}^{n_2} \omega_j^{(2)} Z_2(x_j), \tag{2.6}$$

where $\hat{Z}_1(x_0)$ is the predicted value of the primary variable at location x_0, $Z_1(x_i)$ represents known values of the primary variable at locations x_i, $Z_2(x_j)$ represents known values of the secondary variable at locations x_j, $\omega_i^{(1)}$ is the weights for the primary variable, $\omega_j^{(2)}$ is the weights for the secondary variable, n_1 and n_2 also represent the number of known points for the primary and secondary variables, respectively.

The weights are chosen to minimize the estimation error (variance). In Co-Kriging, the calculations become more complex because it accounts for both the spatial autocorrelation of the primary variable (how similar values are as a function of distance), and the cross-correlation between the primary and secondary variables (how the two variables are related across space). Additionally, to ensure the prediction is unbiased with respect to the trend, additional constraints are imposed as

$$\sum_{i=1}^{n_1} \omega_i^{(1)} f_k(x_i) + \sum_{i=1}^{n_2} \omega_j^{(2)} g_k(x_j) = f_k(x_0), \tag{2.7}$$

where f_k, and g_k are basis functions used to model the trend (or drift) in Universal Co-Kriging.

2.1.5 Output

As the output, geological modelling tools are well-known for making eye-pleasing colorful visualizations. However, depending on the software, different data can be exported. To be applicable for later numerical simulations, the exported geological model should be spatially discretized. Most of the modelling tools provide basic gridding schemes which can be used for very simple calculations but are unable to be used for simulating coupled and complicated processes like heat convection and so on.

Uncertainty analysis is the focal point of this study. Therefore, several 3-D models had to be generated and exported. One of the reasons for choosing GemPy was its efficiency in perturbing the base case models and generating uncertain scenarios. Meanwhile, it is highly time-consuming to export each scenario separately and handle its data. Thanks to the efficiency of Python in handling big data, all the modelling results were exported as arrays stored in a single file. For the purpose of this study, vertices of all the features (layers and faults) were exported from GemPy to be used for making the mesh.

2.2 Mesh generation

Most of the numerical methods need to be performed on discretized spaces. Typically big domains are first generated and later on, the space in each one is discretized via mesh generators. The mesh must accurately capture the geometry of the input domain with high-quality (well-shaped) cells. From a numerical point of view, the mesh should be fine enough in the adjacency of the important parts of the domain. Meanwhile, the number of elements in the mesh also should stay within a reasonable range for running the simulations. In a routine meshing workflow, some points in the space can be generated to create lines, surfaces, and volumes out of them. After creating the basic geometrical elements, a mesh can be created for the desired geometry. Different methods and software are available for making mesh. In this study, I used Gmsh [38] which is a fast and efficient meshing tool. Gmsh is also freely available in different programming languages like Python, Julia, etc., and also has its scripting method (generating .geo files) and a user-friendly graphical user interface (GUI).

Surface interpolation

For the sake of this PhD, new functionalities are added to Gmsh that allow for including geological uncertainty in numerical simulations. The new functionality fits surfaces into any arbitrary shape of points. Generating surfaces matching scattered point clouds is a challenging task even for 3-D geological modelling tools. Researchers in the field of computational geometry, computer graphics and computer aided design (CAD) dedicated a considerable amount of time to achieve robust solutions for this problem [71]. The challenge is reconstructing a surface that fits the point cloud while there is no unique solution to this problem. The point cloud can also be scattered, not evenly distributed, very large, and may have discontinuities e.g., faults that make the surface reconstruction more complicated.

The new features of Gmsh are linked to B-spline surface interpolation. Many data analysis tasks involve finding a smooth pattern that underlies data points. Researchers often use B-spline functions of varying complexity to achieve this in one and two dimensions, either exactly matching the data (interpolation) or providing a close fit (approximation) [72, 73]. What makes the B-Spline surfaces special is that not only do they join smoothly, but they also have smooth curves and bends where they meet. For instance, a "cubic" B-spline surface is particularly smooth (C^0 continuity), ensuring not just continuous connection but also continuous changes in slope (C^1 continuity) and curvature (C^2 continuity) across its boundaries [74]. A B-spline surface is constructed by applying the tensor product technique to the (B-spline) curves [75]. The following expression calculates the function for the B-spline curve passing through control points in 1-D:

$$S(t) = \sum_{i=0}^{n} N_{i,k}(t) P_i, \tag{2.8}$$

where n defines the number of control points, k is the degree of the curve fitted to the points, e.g. 3 for a cubic polynomial, $N_{i,k}(t)$ is the basis function, t is the global parameter making the knot vector (T) and P_i represents the coordinate of the corresponding control point. Then, the following rule defines the value of the basis function for each control point:

$$N_{i,0}(t) = \begin{cases} 1 & \text{if } t_i \leq t < t_{i+1} \\ 0 & \text{otherwise.} \end{cases}$$

The above-mentioned rule acts like a flag for turning the basis functions on and off. $N_{i,0}(t)$ is a B-spline of degree 0, i.e. a piece-wise constant polynomial. The Cox–de Boor recursion formula [76] can be exerted to define higher degree B-Spline which all root in lower degree ones until reaching degree 0:

$$N_{i,j}(t) = \frac{t - t_i}{t_{i+j} - t_i} N_{i,j-1}(t) + \frac{t_{i+j+1} - t}{t_{i+j+1} - t_{i+1}} N_{i+1,j-1}(t). \tag{2.9}$$

The following diagram shows the idea of the Cox–de Boor recursion schematically:

$$\begin{tikzcd}
N_{0,0}(t) \arrow[r] & N_{0,1}(t) \arrow[r] & N_{0,k-1}(t) \arrow[r] & N_{0,k}(t) \\
N_{1,0}(t) \arrow[r] \arrow[ur] & N_{1,1}(t) \arrow[r] \arrow[ur] & N_{1,k-1}(t) \arrow[ur] & \\
N_{n-1,0}(t) \arrow[r] \arrow[ur] & N_{n-1,1}(t) \arrow[ur] & & \\
N_{n,0}(t) \arrow[ur] & & &
\end{tikzcd}$$

For example, we may look for a cubic B-spline ($k = 3$) passing through 4 control points ($n = 3$, because it starts from 0 in the Eq. 2.6). For the sake of simplification, we consider a uniform distribution for the knot vector to be $T = [0,1,2,3,4]$. Then, the following equation should be solved for this case:

$$S(t) = \sum_{i=0}^{3} N_{i,3}(t) P_i. \tag{2.10}$$

The term i iterates over the number of control points. Then, for the first control ($i = 0$ and P_0) the basis function will be $N_{0,3}(t)$. The Cox–de Boor recursion for this $N_{0,3}(t)$ is presented in the following:

$$\begin{tikzcd}
N_{0,0}(t) \arrow[r] & N_{0,1}(t) \arrow[r] & N_{0,2}(t) \arrow[r] & N_{0,3}(t) \\
N_{1,0}(t) \arrow[r] \arrow[ur] & N_{1,1}(t) \arrow[r] \arrow[ur] & N_{1,2}(t) \arrow[ur] & \\
N_{2,0}(t) \arrow[r] \arrow[ur] & N_{2,1}(t) \arrow[ur] & & \\
N_{3,0}(t) \arrow[ur] & & &
\end{tikzcd}$$

In the equation form also $N_{0,3}(t)$ equals:

$$= \frac{t - t_0}{t_3 - t_0} N_{0,2}(t) + \frac{t_4 - t}{t_4 - t_1} N_{1,2}(t). \tag{2.11}$$

Note that the $N_{0,2}(t)$ and $N_{1,2}(t)$ from equation 2.9 need to be calculated as we did for $N_{0,3}(t)$:

$$N_{0,2}(t) = \frac{t - t_0}{t_2 - t_0} N_{0,1}(t) + \frac{t_3 - t}{t_3 - t_1} N_{1,1}(t), \tag{2.12}$$

$$N_{1,2}(t) = \frac{t - t_1}{t_3 - t_1} N_{1,1}(t) + \frac{t_4 - t}{t_4 - t_2} N_{2,1}(t). \tag{2.13}$$

The schematic shows how accumulation of $N_{0,2}(t)$ and $N_{1,2}(t)$ yields in $N_{0,3}(t)$ basis function. But now, $N_{0,1}(t)$, $N_{1,1}(t)$, and $N_{2,1}(t)$ should be written in a way to reach the lowest level of the algorithm (left-hand side of the schematic: $N_{0,0}(t)$, $N_{1,0}(t)$, $N_{2,0}(t)$, and $N_{3,0}(t)$):

$$N_{0,1}(t) = \frac{t - t_0}{t_1 - t_0} N_{0,0}(t) + \frac{t_2 - t}{t_2 - t_1} N_{1,0}(t), \tag{2.14}$$

$$N_{1,1}(t) = \frac{t - t_1}{t_2 - t_1} N_{1,0}(t) + \frac{t_3 - t}{t_3 - t_2} N_{2,0}(t), \tag{2.15}$$

$$N_{2,1}(t) = \frac{t - t_2}{t_3 - t_2} N_{2,0}(t) + \frac{t_4 - t}{t_4 - t_3} N_{3,0}(t). \tag{2.16}$$

Now, Eq 2.10 to 2.14 can be substituted in Eq 2.9. However, we already know that if $t_i \leq t < t_{i+1}$, the basis function equals 0 because for the $N_{0,3}(t)$ the i value is 0. Therefore, only $N_{0,0}(t)$ matters in calculation of $N_{0,3}(t)$:

$$N_{0,3}(t) = \frac{(t-t_0)^3}{(t_3-t_0)(t_2-t_0)(t_1-t_0)} N_{0,0}(t). \tag{2.17}$$

By substituting the t values from the vector T we will have

$$N_{0,3}(t) = \frac{t^3}{6} N_{0,0}(t). \tag{2.18}$$

Eq. 2.16 holds for the interpolation of the curve between the first two knot values, i.e. 0 and 1. The same should be done also when the t is between 1 and 2. After marching through the knot vector, the same process should repeat to calculate the basis functions related to the other control points: $N_{1,3}(t)$, $N_{2,3}(t)$, and $N_{3,3}(t)$.

While $S(t)$ approximates the function of a line, $S(u,v)$ in the following equation yields a B-spline surface patch constructed by the tensor product to the B-spline curves in the x and y directions. The linear combinations of B-spline basis functions allow for generating the surface from two curves:

$$S(u,v) = \sum_{i=0}^{m} \sum_{j=0}^{n} N_i^k(u) N_j^l(v) P_{i,j}, \tag{2.19}$$

where u and v are global parameters which act like t for the B-spline curve, i and j are indices of the m and n number of control points in x and y directions, respectively, k and l are degree of the B-spline curves in x and y directions, respectively, and $P_{i,j}$ is the coordinate of the control points. The process for making B-spline curves happens here but in two directions and at the end a grid point is generated to represent the interpolated surface. The following simple one line of code in the Python API of Gmsh allows for making a B-spline surface that fits into any shape of point cloud:

Gmsh.model.occ.addSurfaceFilling()

It is necessary to first provide the function with the line numbers that create a boundary around the point cloud. Therefore, I first create the related lines surrounding the point cloud and feed their numbers into the function. The surrounding lines of the point cloud i.e. polygon can have any arbitrary shape. As the next argument, the function asks for the point numbers that the surface should fit into. Under the hood, Gmsh leverages the functionalities of the OpenCASCADE Kernel for constructing the B-spline surface. The OpenCASCADE is an open-source full-scale 3-D geometry library. Several examples containing detailed descriptions and informative visualizations are provided in my GitHub repository (https://github.com/Ali1990dashti) to guide people who are interested in making meshes for complex geological models.

In the frame of this PhD, geological layers can now be defined by fitting a B-spline surface going through each set of data points. Gmsh optimizes the fitting process in a way to achieve the highest accuracy. Points of complex surfaces can be imported into Gmsh and then it will automatically fit surfaces to them. Fig. 2.3-a shows a geological model constructed in GemPy. The model contains 4 surfaces that cut the whole volume into five ones. To recreate these surfaces in the mesh, Gmsh only needs to know the coordinates of the points. After importing the coordinates into Gmsh, first, a line passing through the border of the point cloud should be created. Then, Gmsh fits a surface based on the created borderline and also existing points. Four created surfaces in Fig. 2.3-b match what has been generated in GemPy. Fig. 2.3-c also provides a view inside the meshed model to present the similarity between surfaces. The surface fitting function has been called four times in this model to create four surfaces. After generating these four surfaces, the whole domain has been cut based on them (via Boolean operations) to generate five regions, i.e. layers.

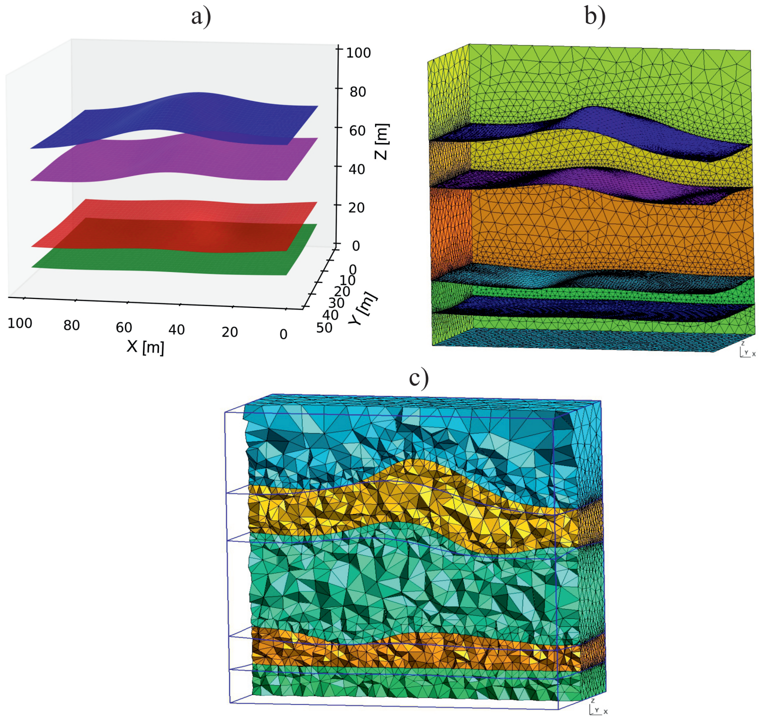
a)
b)
c)

Fig. 2.3 a) Four surfaces coming out of a complex geological model built by GemPy. Surfaces are big enough to touch the model's boundaries. b) Similar surfaces have been generated in Gmsh with B-spline surface fitting. c) A view inside the model to visualize the high quality of the generated mesh.

Meshing automation

As the next step for automatizing, several Python scripts are prepared to create water-tight meshes for as many geological realizations as required. To make the mesh applicable in numerical solvers, it must be well prepared. In the following a simple geometry is presented to discuss the mesh preparation steps. Fig. 2.4-a shows the generated mesh in the first step. It is a 2-D surface with a 1-D line in the middle. The surface can be considered as a metal plate and the line can be a fracture. Both the surface and line have some nodes shown by red dots and edges (lines connecting the nodes). The main problem of this mesh is that nodes of the 1-D line are not connected to the 2-D plane (background). If we put a high temperature boundary condition on the fracture, the temperature can not propagate from the fracture to the plate because their elements do not share any node or element. The fracture should be embedded into the background plate. Fig. 2.4-b shows how the mesh for the simple geometry looks like after the embedment.

Another point is that mesh should be refined correctly before the simulations. In parts of the domain, a sharp change in the properties may happen. Therefore, the solver should have enough calculation spots (nodes) in that area. In our simplified model it is assumed that a temperature boundary condition may be deployed on top of the fracture. The higher density of nodes in Fig. 2.4-c spotlights this part of geometry. More nodes are required to enable a better estimation for the solver. The majority of the solvers are sensitive to high gradients of values. A sharp change of temperature between two nodes of the same edge can lead to numerical oscillations while dividing the same length into more nodes and edges can mitigate the error.

Another requirement for mesh is that it should contain some information about the created geometry. Most of the mesh generators make the mesh based on existing elements (points, lines, surfaces, and volumes). In the end, each element contains entities like nodes, edges, faces, and cells. For example, a point can only have nodes while a line can have both nodes and edges or a surface can contain nodes, edges, and faces. In our simplified mesh the line has some nodes and edges. In real world, this line (fracture) has some physical properties that need to be included in the simulation. Therefore, the mesh should be prepared in a way that the material properties can be assigned to the edges and nodes of the line. In Gmsh physical properties can be used to achieve this goal. All the entities of the line can be grouped and called a fracture. Then, in the numerical solver, the related material properties can be attributed to the fracture and not every node and edge of the line.

In this study, the whole process from importing the geometry to exporting the mesh is automated. With one single Python script, thousands of points are imported into Gmsh and the surfaces are generated based on the imported point clouds. In the next step, the embedment process is done automatically based on the existing elements in the geometry.

Other steps like refinement and physical properties assignment are also done without any need for manual work. This is the only way to make the mesh for several geological realizations. In the end, each geological realization will have its mesh. Please refer to the Github repository of Ali Dashti (https://github.com/Ali1990dashti) to access the available examples.

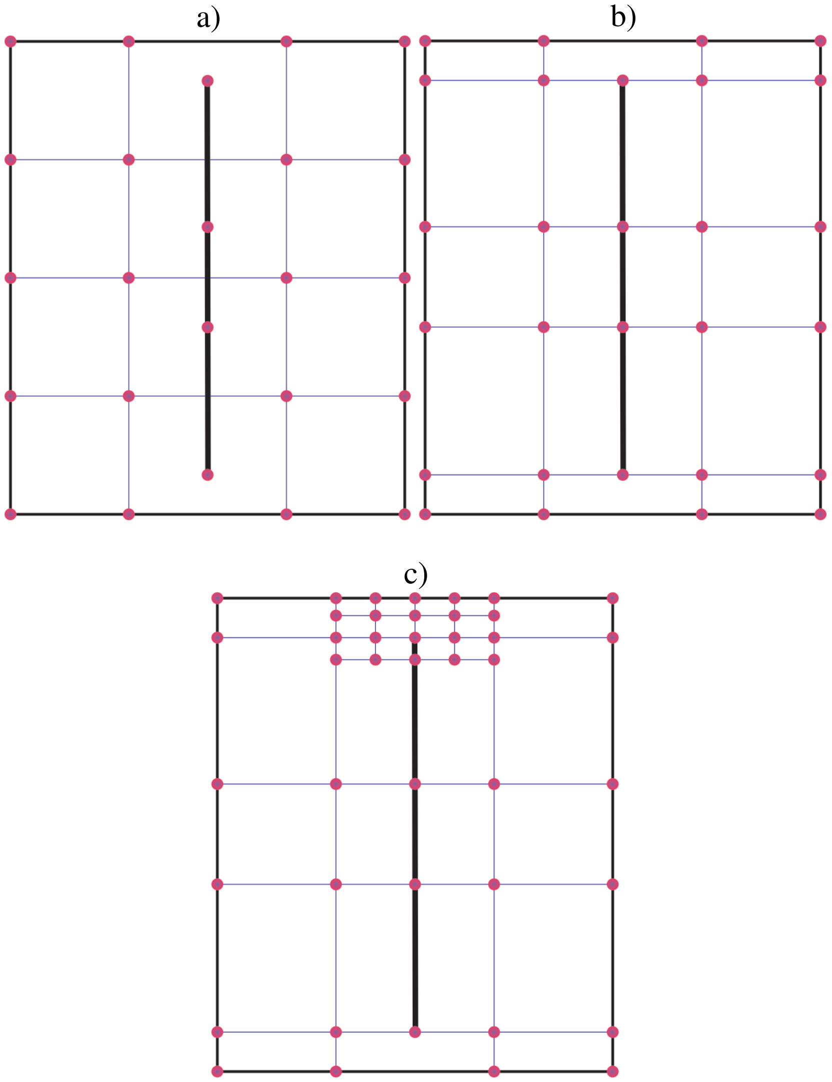

Fig. 2.4 A simple 2-D example is presented to explain the required properties of a finite element mesh. a) The domain is composed of a 2-D plane and one line inside. The primary mesh contains nodes represented by dots and elements shown by lines. b) The elements of the line are correctly embedded inside the plane. In this step the line and plane are sharing nodes. c) The upper part of the line is supposed to be a hot spot in the numerical simulation, hence more nodes and elements are generated around this point.

2.3 GeoMeshPy

To be able to run simulations of a myriad of geological scenarios, there is no other way than automating the mesh generation. In this study, outputs of GemPy are prepared and adjusted in a way to be readable inputs for Gmsh. The whole adjustment process from GemPy to Gmsh is published as a Python library called GeoMeshPy. GemPy can return points representing the existing geological features like surfaces and faults. Fig. 2.5-a shows a geological model coming out of GemPy. The model contains two normal faults cutting a dome-shaped structure. From a visual point of view, the model looks fine. However, Fig. 2.5-b gives a better view of the results generated in GemPy. In Fig. 2.5-b vertices of the surface (contact of layers) are plotted as black dots and faults are represented via two surfaces. In fact, the surface between the layers (blue surface in Fig. 2.5-a) is passing through the exact point shown as black dots in Fig. 2.5-b. In GemPy, fault surfaces are also generated by vertices but here surfaces are visualized to keep the figure simple.

The first notable issue is that the vertices of the layers are making an irregular grid with some redundant points (see the black dots and their distribution in Fig. 2.5-b). GemPy generates a regular grid of vertices plus some minor irregular ones that should be removed. GeoMeshPy looks for these irregular vertices and removes them. Sorting a regular grid based on the x and y values is far easier and faster than an irregular one.

The more problematic issue is that the fault surfaces in GemPy do not cut the contact of the layers properly. Black dots are stretching over two fault planes in Fig. 2.5-b confirming this fact. The graphic visualization (Fig. 2.5-a) shows the offset but the exported data (Fig. 2.5-b) depict this problem. Therefore, it was the second goal of GeoMeshPy to cut the surfaces in geological models with faults. To do so, in locations where the vertices of the layers and faults were meeting each other, the vertices of layers were removed. A threshold is defined that will remove the overlapping vertices of the layers and faults. This way, the single point cloud coming out of GemPy will be split into three patches: one in the right-hand side, one in the middle of two faults, and one on the left-hand side. Colorful dots in Fig. 2.6 show these three patches.

This removal of the redundant points is useful but still, we need to guarantee that contact of the layers will pass through fault surfaces. After making the three patches, GeoMeshPy looks at the relation of each patch with the existing fault and adds extra points to them. For example, the green patch is terminated with a fault surface, hence some new point will be added in a way to pass the patch through the left fault. For the middle patch (orange one) both sides are facing faults and new points need to be generated in both directions. The red patch in Fig. 2.6 is facing a fault on its left-hand side and a new set of points should be generated in that part of this point cloud. Based on these three patches and two faults, the

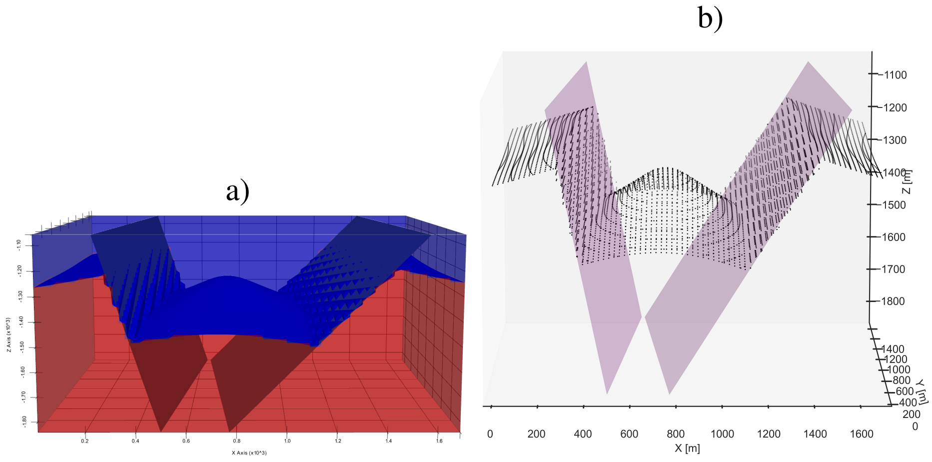

Fig. 2.5 a) A geological model with two normal faults and a dome-like layer created in GemPy. The visualization shows the displacement of the layers via two faults. b) Exported vertices of the layers are plotted as black dots and two fault surfaces are visualized as surfaces. The black dots are continuous in the domain which confirms that the faults are displacing the layers without cutting them.

whole domain is split into six regions. Blue and red stars in Fig. 2.6 show each region of the model. Later, the coordinates of the stars can be used to assign proper physical properties to each part of the mesh. Gmsh will create 6 volumes for such a model while from a geological point of view, the model has only two formations.

Based on the recently generated patches, the domain should be split into 6 volumes. The six stars shown in Fig. 2.6 are later on used in Gmsh to assign the correct physical properties in each part of the model. The red stars represent the lower formation while the blue ones are extracted to show the upper formation. These representative points are detected by GeoMeshPy based on the generated model in GemPy (available examples and documentation at https://github.com/Ali1990dashti/GeoMeshPy). The final result of meshing is presented in Fig. 2.7. In this example, the surface fitting function has been called three times to create patches.

It should be noted that GeoMeshPy can handle as many as generated models in GemPy. Through running one single block of code in Python, all the geological realizations can be prepared as a readable input file for Gmsh.

Fig. 2.8 summarizes the workflow I designed in this PhD thesis. While GemPy has been available for years for making uncertainty analysis, involving the uncertainty in numerical simulations has not been accomplished due to the meshing process. GeoMeshPy is developed specifically to enable the connection of GemPy with Gmsh. However, GeoMeshPy suffers from limitations regarding the type of structural models that can be generated in GemPy.

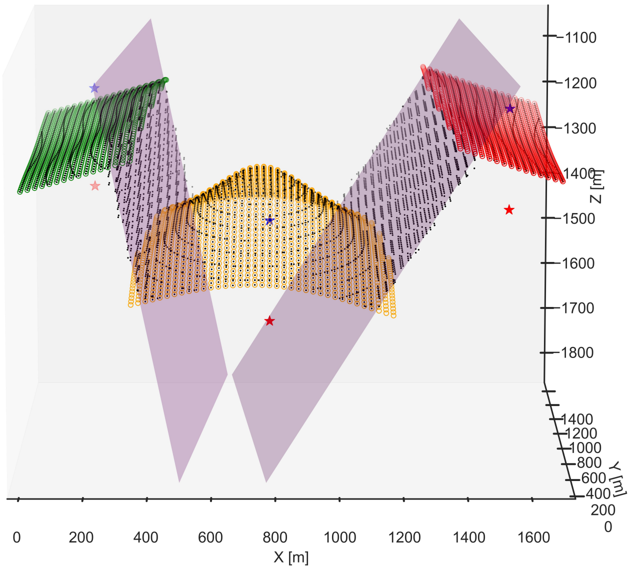

Fig. 2.6 The continuous layer coming from GemPy (black dots) is split into three patches based on the existing faults. This approach divides the domain into six regions that can be identified by stars.

While GemPy can generate a wide variety of geological relations, e.g. erosion, fault, onlap, etc. GeoMeshPy can only deal with erosional and specific types of fault relations. the main issue is the unpredictability of the generated vertices in GemPy. If these two faults of the model presented in Fig. 2.6 intersect, the distribution of their vertices will also be different. GeoMeshPy can handle any number of parallel faults but in the case of intersecting faults, it can deal with only three of them. GemPy allows the user to explicitly tailor the relationship between faults and layers that affect the vertices and GeoMeshPy will fail if it does not recognize the relation properly.

2.3.1 Applications

Fluid flow

EGS applications try to engineer a reservoir that facilitates the circulation of the fluids. The success of such projects is tightly linked to the location and orientation of the thin fracture zones. Due to the omnipresent uncertainty, it is safer to consider the measured properties as probability distributions rather than deterministic values. Then, based on the set of values drawn from these distributions, the knowledge updates. In Chapter 3 of this thesis, GeoMeshPy is used to quantify the effect of uncertainties of the location and orientation of a fault on flow. Fig. 2.9 presents a highly simplified fractured reservoir of a possible EGS site. The big fault surfaces where injection and production happen are connected via a smaller

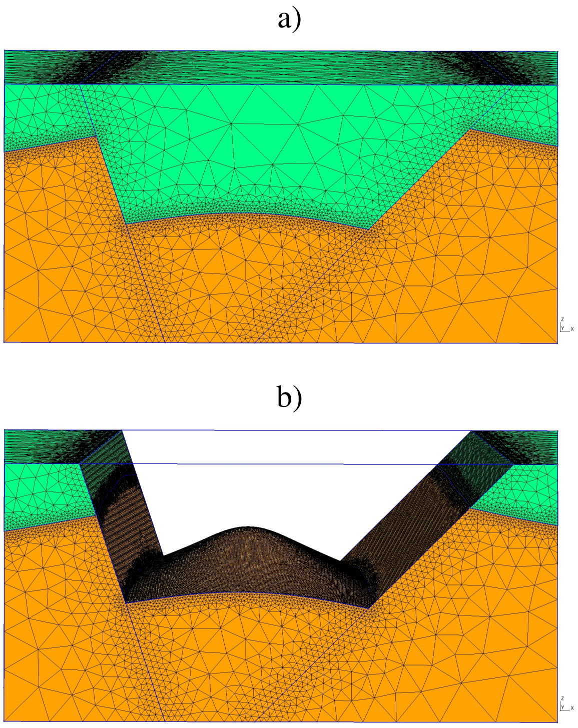

Fig. 2.7 Two views on the generated mesh based on the geological model. The top formation is shown as green blocks while the lower formation is in orange. The water-tight mesh follows the exact topology of the geological model.

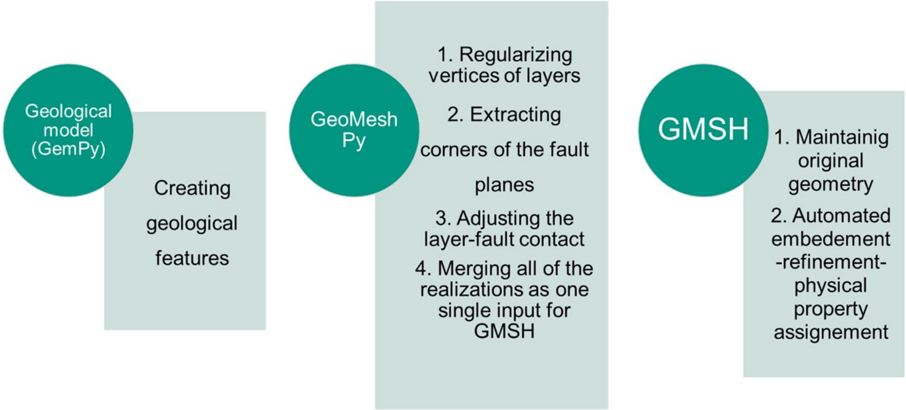

Fig. 2.8 The workflow developed based on GemPy and Gmsh codes.

fracture i.e. fault. In the figure, only two possible scenarios are plotted. In one case maybe the orange fracture may represent the real underground situation while in another case we may encounter the blue fracture. The flow regime differs dramatically from one case to another. Therefore, including each scenario is the only solution. In the case of Fig. 2.9 we need two geological models and meshes but in reality, up to thousands of realizations should be included to cover a range of possible uncertainty. GemPy allows for making as much as required geological models. Then, all the generated models can be imported to GeoMeshPy to prepare them as readable inputs for mesh generators like Gmsh.

In this thesis, the tracer flow is simulated using an open source finite element (FE) application called TIGER (THMC sImulator for GEoscientific Research) [77]. TIGER is developed at Karlsruhe Institute of Technology (KIT) and inherits the functionalities of MOOSE (Multiphysics Object-Oriented Simulation Environment) framework [78]. MOOSE is a flexible and parallel framework that solves multi-physics and -scale problems in a fully coupled implicit manner. MOOSE wraps the state-of-the-art libraries, such as PETSc (a suite of data structures and routines developed to calculate scalable parallel solutions) [79] and libMesh [80]. In TIGER mass and momentum balances are combined to calculate the pore pressure field:

$$S_m \frac{\partial P}{\partial t} + \nabla \cdot q = Q, \tag{2.20}$$

where S_m is the specific storage coefficient of the (mixture) liquid and solid phases, P is the pore pressure, t represents time, q is the fluid or Darcy velocity vector, and Q denotes the sink/source term for injection and production. The Darcy velocity is also calculated based on

$$q = \frac{k}{\mu}(\nabla P + \rho_f g), \tag{2.21}$$

where k is the permeability tensor, μ is the fluid dynamic viscosity, ρ_f represents fluid density, and g is the gravitational acceleration vector.

After solving the hydraulic field, the tracer concentration spatial and temporal changes (as a conservative and non-reactive solute) are solved via advection-diffusion-dispersion equation:

$$\phi \frac{\partial C}{\partial t} + (-\nabla \cdot D_m \nabla C) + \nabla \cdot C = Q, \tag{2.22}$$

where ϕ is the porosity, C is the solute concentration, and D_m is the sum of molecular diffusion and dispersion.

Heat storage

To store large amounts of thermal energy over long time periods underground, the storage concept seems a promising technology [81] but requires an accurate characterization. The

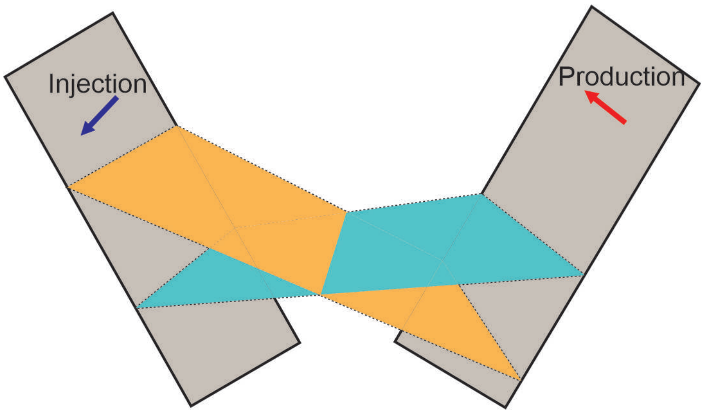

Fig. 2.9 A simplified schematic to show two possible ways of connecting two big fractures.

existence of a porous and permeable reservoir is the first requirement. Chapter 4 of this study first evaluates the uncertainties related to the geometry and volume of the reservoir candidates for a storage application. A simple base case where the reservoir was a simple box was perturbed to see the effect of the reservoir's geometry on the thermohydraulic performance of the system. A big difference between the expected volume and the real one is a potential risk for storage. Gmsh with its new functionalities is able to fit surfaces to any arbitrary shape of points.

TIGER is again used for running transient numerical simulations. The hydraulic field is solved as mentioned before and the advection-diffusion equation is used to solve for heat transport, i.e. temperature field:

$$\rho c_p \frac{\partial T}{\partial t} - \lambda \nabla^2 T + (\rho c_p)_f q \nabla T = 0, \tag{2.23}$$

where ρc_p and λ represent heat capacity and thermal conductivity of the in equilibrium liquid and solid phases, T is the temperature, and $(\rho c_p)_f$ is the heat capacity of the liquid phase.

Fault reactivation risk

Despite being a promising source of renewable energy, geothermal activities face a common potential hurdle: induced seismicity. Research by Seithel et al. (2019) [82] and Kraft et al. (2008) [83] suggests this risk can even extends to hydrothermal systems, which rely on porous rock formations. As explored in the final chapter of this thesis, the Munich area exemplifies this concern. A complex fault plane is embedded in the Malm aquifer to see the possible impact of the operation on this fault.

To properly calculate the possibility of the fault surface reactivation, coupled THM simulations are carried out. The PorousFlow module of the MOOSE framework is used as a simulator for THM simulations. All the mathematical background of the PorousFlow is discussed by Wilkins et al. (2021) [84] and a short introduction is also presented here. The home page of the MOOSE (https://mooseframework.inl.gov/index.html) also contains great details on the different aspects of the MOOSE and each module including PorousFlow.

To solve the hydraulic field in a THM simulation, PorousFlow uses mass conservation which is described by the continuity equation:

$$\frac{\partial m}{\partial t} + m\nabla \cdot v_s + \nabla \cdot F - q = 0, \tag{2.24}$$

where m is the fluid mass, v_s is the velocity of the solid skeleton, F represents the flux, and q is the source term. PorousFlow can handle multi-phase and multi-component flows which is elaborated by Wilkins et al. (2021) [84]. The flux term F is composed of the advective flux and diffusive-and-dispersive flux:

$$F = F_{Advection} + F_{Diffusion+Dispersion}. \tag{2.25}$$

Due to the single-phase nature of our problem, the dispersive and diffusive fluxes are ignored and the advective flux of the fluid is solved by Darcy's law.

PorousFlow assumes a local thermodynamic equilibrium for the liquids and solid phases. The energy conservation for heat is described by the continuity equation

$$\frac{\partial e}{\partial t} + e\nabla \cdot v_s + \nabla \cdot F^T - q^T = 0. \tag{2.26}$$

where e the heat energy, F^T represents the heat fluid flux, and q^T is a heat source. Rock grain density, the specific heat capacity of the rock, and the internal energy of the fluid phase are used to calculate the heat energy (e). The heat flux is a sum of heat conduction and convection with the fluid:

$$F^T = -\lambda \nabla T + F_{Advection}, \tag{2.27}$$

where λ is the thermal conductivity tensor, which in PorousFlow is a function of the thermal conductivities of the fluid and rock phases.

PorousFlow leverages the available functionalities from other MOOSE module in order to handle solid mechanics and the coupling. The evolution of porosity with pressure, strain, and temperature is fundamental to the coupling between flows and solid mechanics. The effective stress tensor is calculated as [84]

$$\sigma_{ij}^{eff} = \sigma_{ij}^{tot} + \alpha \delta_{ij} P, \tag{2.28}$$

where σ^{eff} represents the effective stress tensor, σ^{tot} is the total stress tensor, α is the Biot coefficient, δ is the Kronecker delta and P is the pressure. Then, the elastic constitutive law [85] links the deformation of the porous solid to the effective stress tensor:

$$\sigma_{ij}^{eff} = E_{ijkl}(\varepsilon_{kl}^{elastic} - \delta_{kl}\alpha_T T), \tag{2.29}$$

where E_{ijkl} is the elasticity tensor of the solid skeleton (drained version) which is inverse of the drained compliance tensor, $\varepsilon^{elastic}$ is the strain tensor, α_T is the volumetric drained thermal expansion coefficient, and T is the temperature.

Chapter 3

Impact of structural uncertainty on tracer test design in faulted geothermal reservoirs

This chapter is published in Geothermics: https://doi.org/10.1016/j.geothermics.2022.102607

Abstract

Evaluation of underground processes requires numerical modelling based on sophisticated and reliable meshing. Our new *GeoMeshPy* library focuses on the discretization of probabilistic geological structures. This study presents a synthetic show-case for the capacity of this library to quantify the impact of structural uncertainty. Here, 50 models were developed taking advantage of the computational efficiency of *GeoMeshPy*. Assuming a geothermal doublet system embedded in a faulted reservoir with an unclear structure, the recovery time and magnitude of a tracer breakthrough were calculated. Even small angular variations up to ±15° in one of the faults yield differences of up to 26 and 30 percent for peak arrival time and magnitude, respectively. An additional inversion scheme of each of the 50 curves allows quantifying the impact on the Péclet number varying from 3.4 to 4.3 due to structural variability. Analytically calculated dispersion coefficients are almost one order of magnitude higher than values used for simulations. Besides this mismatch, calculated dispersion coefficients (ranging from 125.6 and 129.4 m) are unable to represent the structural uncertainty.

3.1 Introduction

An understanding of the underground flow regime is a pivotal necessity for any long-term subsurface application. Flow characterization through tracer tests is vital in a wide variety of reservoir engineering. Applications are in the fields of geothermal [86, 87], conventional hydrocarbons [88, 89], shale gas [90], CO_2 storage [91, 92], nuclear or toxic waste disposals [93, 94] and so on. For safety issues, these tests are of high importance since they provide robust underground knowledge on the field scale. Furthermore, legal regulations necessitate a reliable forecasting system on a millennial timescale e.g. for CO_2 storage. The success of such projects (CO_2 storage) hinges on various interconnected factors but adding the underground geometry makes the assessment more integrated and reliable.

Since the fluid flow in fractures typically dominates the total regime [95–97], characterization of these zones is crucial for designing exploitation concepts [98]. However, the bias between structural information and flow regime can lead to oversimplifications for reservoir production schemes. Even in permeable fractures, the nature of flow is a problematic and active subject matter [99–102].

To quantify the influence of faults on the performance of the reservoir, detailed understanding of their geometry is essential but rarely available [103]a . Depending on the interpreter's experience, faults are detected from visually recognizable lineaments of discontinuities in reflection patterns on 2/3D seismic data. Besides the interpretation uncertainty, variations of wave velocity or acquisition quality can introduce errors in the structural definition of a fault system [104–106]. This may even apply to state-of-the-art fault detection methods deploying automated machine learning algorithms [107–109]. On the other hand, constrained borehole data can accurately access the depth location of faults. The evaluation of advanced logging techniques can be biased due to local heterogeneities and interpreters' eyesight [110]. In addition, borehole data provide rather punctual information covering a very small fraction of target volume [111].

With the structural information from seismic or borehole data resulting in a conceptual model, hydraulic field tests will further visualize the underground setting. The tracer test, designed in the early 1900s, by itself is a well-documented technology [112, 113] and applied for hydraulic characterization in both porous and fractured media. Tracer test curves quantify the rate and direction of flow but are unable to provide unique solutions for the location and geometry of faults that control the flow regime.

Wellmann and Regenauer-Lieb (2012) [34] highlighted that lack of data, measurement errors, external disturbances, and human factors lead to ubiquitous uncertainty in geological models. Today, it is highly recommended to include uncertainty estimations in the engineering of underground applications [114, 115, 52, 116].

Given the possible scale of uncertainties, from pore space to the basin structural scale, this study investigates the structural uncertainty rather than sensitivity analyses or addressing specific material sets existing on a once elaborated mesh. To encapsulate the uncertainty, one approach is to present the values of the measured properties as probability distributions.

Typically structural uncertainties are evaluated by attributing different material properties i.e. material uncertainty in numerical schemes. Reconstructing geological features (e.g., faults and contact of layers) is a time-consuming and computationally demanding process. The open-source, Python-based implicit modelling package called Gempy [37] is used to generate probabilistic structural models. Herein we establish a novel and fast workflow in Python ecosystem that includes this structural uncertainty directly in the meshing procedure. The developed Python library is called *GeoMeshPy* and converts Gempy outputs into inputs for the mesh generator called GMSH [38].

This study aims to assess the uncertainty of structural information using synthetic tracer test data. The presented numerical scheme should lead to an optimum experimental tracer test design that allows for defining test duration or required tracer mass. The structural uncertainty is included in designing the field tracer experiments and its effect on flow and transport properties is quantified. The manuscript first elaborates the probabilistic setting of the numerical scheme using 50 geological realizations to cover a range of structural possibilities in a hydraulic solute (HS) modelling framework. Then, we quantify how these variations can affect both the tracer test design and interpretations derived from this test.

3.2 Methodology

3.2.1 Overall approach

Tracer experiments result in breakthrough curves (BTCs) typically monitored from borehole locations. There are different analytical or numerical models to assess the BTC with different degrees of sophistication matching the complexities of geological models [117, 118]. Available analytical solutions are unable to provide reliable information for underground settings due to simplifications. In the inversion scheme of such methods, characteristics like flow condition and dispersion coefficients are tuned in a way to fit the continuous BTCs on sparse measured data [119, 120]. They converge to matches without including the possible 3D geometry of the underground.

From the numerical modelling perspective, usually a deterministic geological realization is constructed and parameters are again tuned to fit the field data to results. It is highly time consuming to generate several structural models and later mesh each one. Therefore, the gap between geological models and numerical solvers hampers including structural uncertainty in numerical simulation. Here, we present an automated pipeline to construct these meshes, providing still a large complexity of geological information. Gempy, as the modelling tool, is connected to the mesh generator, GMSH, to create meshes based on the variating structural model. The developed package, *GeoMeshPy*, is able to handle a variety of existing functionalities in Gempy. Available examples can visualize the capacities of the developed package. Therefore, 50 different structural realizations with their unique geometries are meshed and fed into the simulator. Each model is discretized based on its geological features e.g. layers and faults before being fed into a numerical solver. BTCs are simulated for several perturbed geological realizations with fixed small-scale parameters (e.g., porosity, permeability, diffusion, and dispersion coefficients). In the next step, generated BTCs are assessed analytically to quantify the effect of the structural uncertainty on extracted parameters. Then, extracted parameters are compared to the input data of simulations regarding the structural model. Completing this loop can help to quantify the effect of structural uncertainty on the flow characteristics of the underground. The following bullet points summarize the general scheme of this study:

- Generating 50 different structural realizations
- Converting outputs of geomodelling tool into useable inputs for the mesh generator
- Meshing each realization based on its geometry
- Simulating tracer test for each realization with fixed operational input parameters
- Inverting a bandwidth of analytically calculated outputs from simulation results

3.2.2 Conceptual model

In this study, a generic doublet system is designed to investigate the impacts of structural uncertainty on flow-related parameters. The overall scheme of the model follows Soultz-sous-Forêts reservoir which is one of the most well-known Enhanced Geothermal Systems (EGS) [121]. The presented doublet system inherits the basic parameters of the mentioned EGS but with less geological complexity. Several authors [122–129] discussed details of the most probable flow and fracture systems in Soultz-sous-Forêts. In this study, a simplified model of this EGS is utilized to understand the impact of the structural uncertainty rather than providing deterministic and complicated solutions.

The synthetic doublet system includes two deviated wells (injection and production) penetrating a granite reservoir. Stratigraphically, the geological model contains two different layers: sedimentary cap rock and granitic reservoir. Three faults are also embedded in the doublet system (Fig. 3.1). The structural model resembles the rift settings, which act as the host for most of the geothermal applications worldwide. Two major big faults called injection and production faults are detected through drilled wells and seismic data. These two faults are abbreviated as Fault_Inj and Fault_Pro in Fig. 3.1. The third one, called connecting fault (abbreviated as Fault_Con in Fig. 3.1), is only observed in seismic data making the geological model uncertain. This fault can be assumed as a karsified/fractured zone with uncertain depth information. There are 50 green lines in this figure representing a 2D view of the connecting fault/any kind of conduit in the middle of the model. Each green line makes a different geological realization. The geological model includes two certain (!) faults (Fault_Inj and Fault_Pro) cut by wells and the uncertain conduit named here Fault_Con. The central point of this uncertain fault can modulate in a 60-meter depth window to cover a range of possibilities.

Fig. 3.1 also includes a stereonet showing the considered uncertainty for this connecting fault under the stress field of Soultz. The dipping angle of this fault can vary up to ±15°. Purely unbiased random numbers from a uniform distribution are generated to perturb the orientation and dipping angle of this connecting fault. Regarding the high uncertainty of seismic data for connecting fault, a surface with zero dip and strike angles (000°/00°) is chosen as the starting point and later perturbed. As an outcome, probabilistic geological models are generated to be fed into numerical solvers rather than sticking to one single deterministic model. The depth, dipping angle, and strike of the Fault_Con can be variated up to values that the fault does not intersect the contact between the granite and cap rock. In this case a new relation between the vertices of the Fault_Con and the layer will be generated that breaks the automatic data adjustment and mesh generation. GemPy is able to create any level of complexity but GeoMeshPy can handle specific configurations of the vertices, as the show-case presented in Fig. 3.1. The extension of the model in x, y and z directions is 1200, 1000, and 700 m, respectively.

3.2.3 Modelling

Geological modelling and vertice conversion

In this study, Gempy, an open source code developed in Python ecosystem, is used for probabilistic geological modelling. The code is highly efficient in generating and visualizing probabilistic structural models based on the potential-field interpolation method [37]. As inputs, Gempy requires depth points and orientation of the included features (layers and faults) in the model. The model is then discretized based on the 3-D matrices of lithology and/or fault IDs (Figs 2 and 3 in Schaaf et al. (2020) [130]). As output, it gives vertices representing all the existing features. Gempy uses the potential-field method for interpolations in which faults are unable to cut contacts of formations.

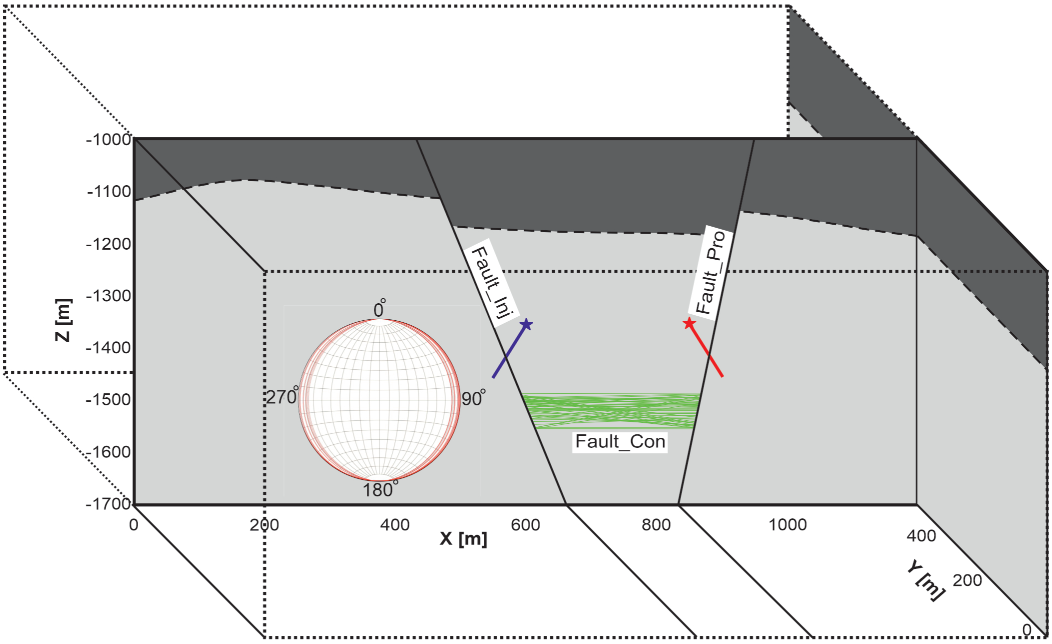

Fig. 3.1 A 3D schematic view on the synthetic model and designed doublet system. Injection fault (Fault_Inj) is cut by the injection well and production fault (Fault_Pro) is cut by the production well. These two certain faults are shown as continuous black lines and thinner green lines show the trace of the uncertain connecting fault (Fault_Con) in the middle (y=500) of the model. Each green trace is making a unique geological realization. A stereonet showing the different traces of the uncertain connecting fault is also embedded in the figure. Dipping values varying up to ±15°.

In this study, vertices of one contact of layers and three faults are separated and later refined to be used as inputs for the mesh generator. Contact of layers separates the Granite reservoir from the cap rock. Initially, as a data cleanup step, the redundant points in the vicinity of existing cutting faults (Fault_Inj and Fault_Pro) and contact of formations are removed. Then, the remaining points will be clustered into three groups based on their relationship with two cutting faults. Fig. 3.2-a shows how Fault_Inj and Fault_Pro can divide vertices of the layers' contact into three patches.

There are three main issues regarding outputs of Gempy that *GeoMeshPy* solves. Firstly, vertices have some redundant points. Areas A-I, A-II, and A-III are annotated in the Fig. 3.2-a to highlight these extra points. After removing these redundancies, the remaining vertices will make a regular grid. *GeoMeshPy* can regularize the vertices of as many exiting layers in the model. Secondly, the contact of the layers is stretched over two cutting faults (areas annotated with B in Fig. 3.2-a). In order to make vertices applicable for later surface generation in GMSH, it is necessary to remove extra points of the existing faults. This way, the continuous output of Gempy will split into three patches (colorful circles in the figure). After removing the extra points, some new regular points are added to the refined ones in order to make sure that the vertices will pass through fault surfaces. Hollow colorful circles in Fig. 3.2-a are obviously passing through two cutting faults while the generated geometry in Gempy is also preserved. Then, it is possible to fit a surface to each patch separately. Thirdly, in the case of connecting fault (Fault_Con), Gempy gives the vertices in all the domain (visualized by letter C in Fig. 3.2-a). This fault is not making any offset but may have effects on the flow. Therefore, its contact points with two other faults are detected in *GeoMeshPy* to recreate the small red surface of Fig. 3.2-a in GMSH.

Fig. 3.2-b shows the final spatial discretization for one of the geological realizations. Besides maintaining the geometry of the model, the mesh generation is accomplished automatically in Python ecosystem. Vertices of Gempy are processed in *GeoMeshPy* and later fed into the Python API of the GMSH. Manual mesh generation for 50 geological realizations is a highly time demanding and error-prone process.

Mesh generation

Mesh generation is the primary step in any discretization method applied to solve a physical phenomenon numerically [131]. Amongst the commercial and open-source packages that can generate suitable meshes for geological modelling [132], the open-source GMSH [38] is selected because of its high efficiency, integrated geometrical modeler, and Python API. GMSH has been adopted specifically to create geometrical surfaces passing through an arbitrary set of points, and to intersect them with other CAD entities. Coupled with its ability

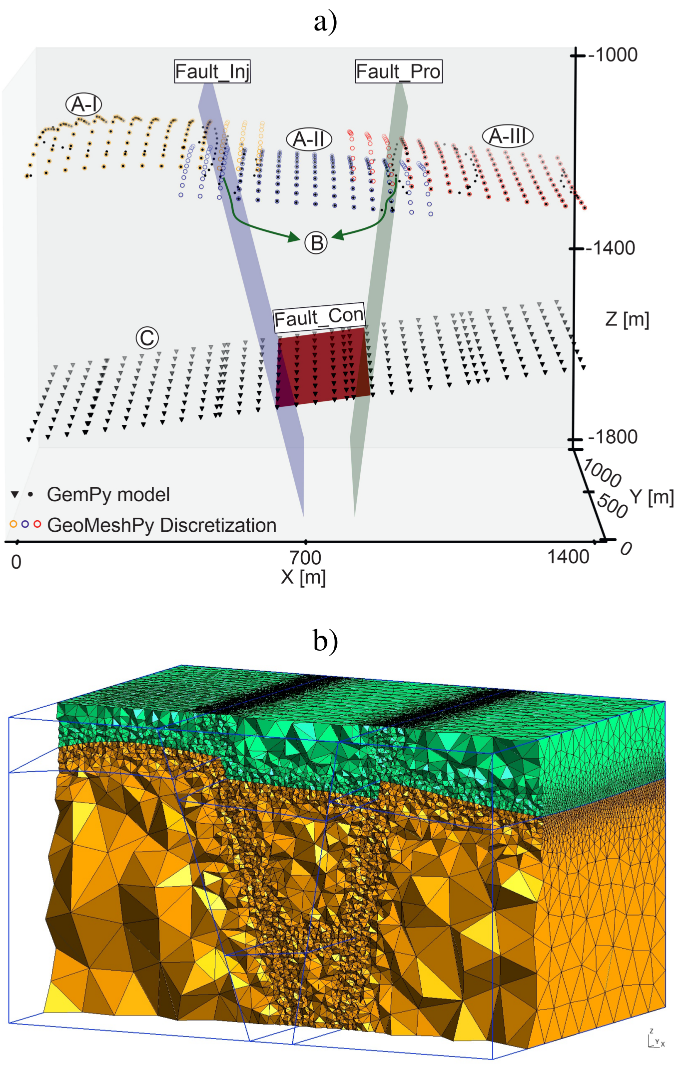

Fig. 3.2 a) Raw output of Gempy for the contact of layers and connecting fault (Fault_Con) is shown as black dots and triangles, respectively. Colorful circles (orange, blue and red) represent refined vertices separated into three patches in *GeoMeshPy*. Fault_Inj and Fault_Pro cut the only exiting contact of layers. The raw output of Gempy has also some redundancies that make the grid irregular in area A-I, A-II and A-III. Area B also shows the continuity of the contact of layers on two faults. *GeoMeshPy* also reduces the larger Gempy output for the Fault_Con. Contacts of this fault with two other big ones is detected. The red surfaces is made using four calculated corners of connecting fault. b) The generated mesh includes two volumes representing the geological layers and three faults.

to generate adapted meshes on non-manifold geometries, this allows to generate conformal meshes that can directly inherit geometries from an advanced modeler like Gempy.

In this study outputs of Gempy are modified using GeoMeshPy and then imported into GMSH. Fig. 3.2-b visualizes a cross section of the discretized scheme achieved in GMSH. The new surface fitting functionality in GMSH can be also used for vertices coming from any other geomodelling tool. It only requires a regular grid of vertices and also the indices where vertices should be separated. For example, in the presented case study vertices of the layers' contact is divided into three patches. Therefore, two separation points should be defined for GMSH. Otherwise, there is no other way for the software to distinguish each patch. For more details please see the available examples. Thanks to the robust and versatile mesh refinement functionality in GMSH, after recreating all the geological features, the mesh size is automatically refined in the vicinity of important features (Fig. 3.2-b).

Numerical simulation

An open source finite element (FE) application called TIGER (Thermo-Hydro-Chemical simulator for Geoscience Research) [77] is used to simulate a tracer test. TIGER is developed on the top of MOOSE (Multiphysics Object-Oriented Simulation Environment) framework [133]. MOOSE is a fully implicit coupled FE toolkit inherits all the capabilities of PETSc (a suite of data structures and routines for the scalable parallel solution of scientific application) and libMesh libraries. Inheriting from the state-of-the-art FE solver, TIGER can quantify Thermo-Hydro-Chemical processes in geothermal domains. TIGER solves the hydraulic field of the pore pressure by combining mass and momentum balances including Darcy's law. Details of the TIGER's mathematical background are addressed in Gholami Korzani et al. (2020) [77]. With TIGER the user have access to most of available solvers and preconditioning schemes. In this study Newton based nonlinear solver that uses a basic line search is used as the nonlinear solver. Hypre's algebraic multigrid (AMG) package (BoomerAMG) is selected as the linear preconditioner. Hypre is a set of solvers/preconditioners from Lawrence Livermore National Laboratory. PETSc and MOOSE documentations elaborated mathematical backgrounds of the selected solvers. In this study the problem is solved fully implicitly in the TIGER. Streamline upwinding schemes of MOOSE were used to simulate model regions of high advection [134, 135].

In TIGER, fractures can be considered as lower-dimensional elements, which leads to a mixed-dimensional problem formulation. The important note is that lower dimensional elements should be properly connected using the embedment functionality of GMSH with higher dimensional ones. Otherwise, isolated elements will be unable to be coupled with higher ones. For example, fluid i.e. tracer is unable to flow from an isolated well represented by 1D lines toward faults (2D surfaces) or matrix (3D volume).

Tracer transport natural phenomena are also solved based on advection-diffusion-dispersion equations. Both space and time changes are included in the hydraulic kernel of TIGER:

$$\frac{\varphi \partial C}{\partial t} + (-\nabla \cdot D_m + q\nabla \cdot C) = Q, \tag{3.1}$$

where φ represents the porosity; C is the solute concentration; t is time; D_m is the sum of molecular diffusion and dispersion, q is the darcy velocity and Q is the source/sink term for injection/production of tracer. Required mathematical details of the tracer transport equations are discussed in literature e.g. Egert et al. (2020) [77], and Gholami Korzani et al. (2020) [77]. Table 3.1 represents the parametrization used for all the simulations of the tracer test.

Table 3.1 Parameters selected to design a generic tracer test. a: [123], b: [136], c: [137]

Parameters		**Values**
Flow rate [l s^{-1}]	Injection	45
	Production	45
Permeability [m^2]	Matrix (reservoir)	1.0×10^{-16} [a]
Porosity [-]	Faults	1 [b]
	Matrix (reservoir)	0.05 [b]
Aperture [m]	Injection Fault	0.2 [a]
	Production Fault	0.15 [a]
	Connecting Fault	0.5 [a]
Transmissivity [m^2 s^{-1}] at 25 °C	Injection Fault	1.2×10^{-6}
	Production Fault	3.3×10^{-7}
	Connecting Fault	5.6×10^{-7}
Porosity [-]	Faults	1 [b]
	Matrix (reservoir)	0.05 [b]
Faults' dispersion coefficients [m]	Longitudinal	15 [c]
	Transverse	5 [c]
Solute diffusion [m^2 s^{-1}]		4.0×10^{-10} [b]
Simulation time [day]		366

3.3 Results and discussion

3.3.1 Mesh sensitivity analysis

To investigate numerical results dependency with respect to spatial discretization, different customized refinements are applied. Field option in GMSH is an efficient functionality allowing for refining meshes close to points, lines, or surfaces while gradually coarsening from a given distance. This way, many adequately fine elements will be generated in a user-defined area around important parts of the model. In tracer transport modelling, 1D wellbores and 2D faults are controlling factors due to high hydraulic and tracer gradient variations. Accordingly, the connection of faults and wellbores with 3D elements has been considered as another influential point of the model.

Regarding these remarks, defining proper mesh sizes is challenging and results revealed that mesh size on fault surfaces plays the most important role. Hence, different mesh sizes on fault surfaces were tried and Fig. 3.3-a shows the relative error versus the number of elements in one of the realizations. The independent solution was achieved when the mesh size is 0.25 m around the wells (a radius of < 3 m) and it is 5 m on the fault surfaces (up to 25 m away from the fault surfaces). Total number of elements in this mesh is 1.35×10^7. However, a coarser mesh with fewer elements (6.6×10^6 versus 1.35×10^7) was used which has about 2% relative error. This reduction in the number of elements can save more than 90 minutes for the computation time running on 24 cores of a High-Performance Computing cluster. The difference between meshes with 1.02×10^7 and 1.35×10^7 elements is also negligible, i.e. less than 0.5%. Therefore, it is decided to choose the required refinement level having 6.6×10^6 elements. Fig. 3.3-b also represents the BTCs of each mesh. As the figure shows, the mesh with 6.6×10^6 yields enough accuracy and the difference with finer meshes is negligible.

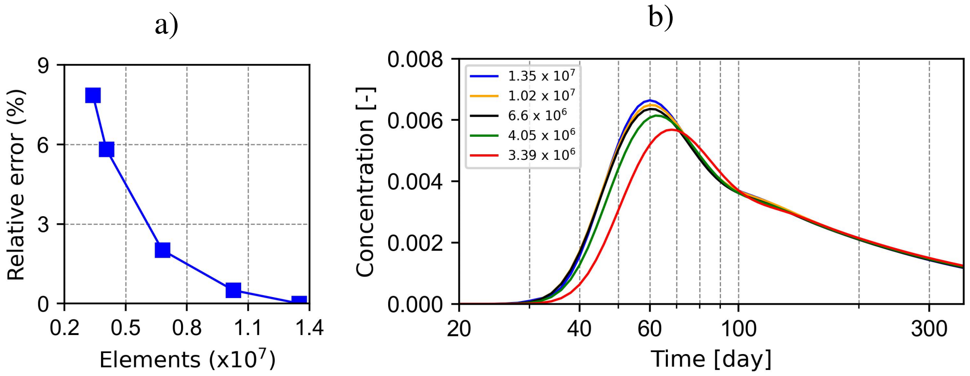

Fig. 3.3 a) The relative error in tracer retrieval peak values versus the number of elements. b) Corresponding BTCs of each mesh. After the required level of refinement (6.6×10^6 elements), BTCs are almost similar.

3.3.2 Tracer flow simulations

To quantify the impacts of structural uncertainty, 50 different geological realizations are designed to simulate a tracer test experiment with the same hydraulic and tracer parameters

(Table 3.1). The tracer is injected 24 hours during day 8 while the response is monitored for a one-year period in the production well. Fig. 3.4 shows BTCs plotted in a logarithmic time scale and peak concentration values marked with red crosses. The peak arrival time varies between days 54 and 68. Two weeks of delay in arrival time, as well as a 30% reduction in peak magnitude, are clear shreds of evidence to highlight the importance of structural uncertainty that can influence expectations from such an important field test.

In the simulations, the produced fluid at the production well is reinjected into the injection well. Therefore, the circulating fluid has concentrations of the tracer for the duration of the simulation even after day 9 when the tracer injection stopped. The observed second peaks between days 100 and 150 of simulation in Fig. 3.4 are caused by this reinjection rather than multiple flow paths or stagnation zones. Approaching the end of simulation time, all BTCs are converging toward similar values while the reinjection again causes the difference. For example, the difference in BTCs on day 300 is far less than on day 60. After the second peak no further rises are observed. This can be interpreted as the low concentrations of tracers are fading out due to dispersion and diffusion rather than advective transport within the fault networks.

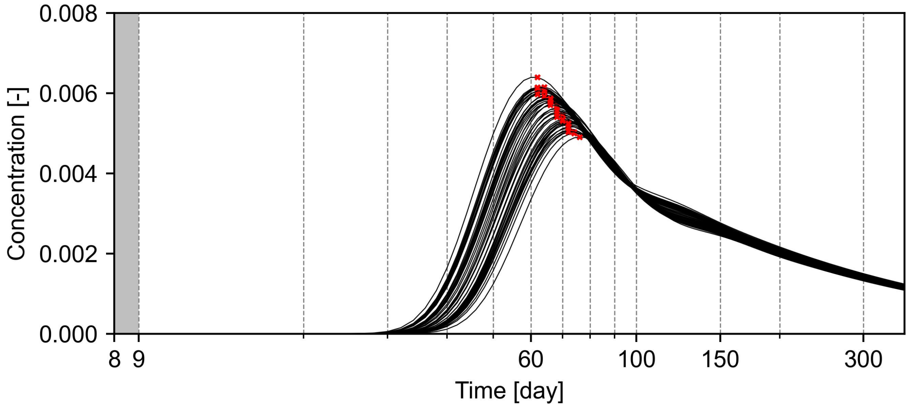

Fig. 3.4 Extracted solute concentration at the production well for 50 different geological realizations. Values for tracer concentration are unitless and the tracer is injected with concentration 1 on day 8 (greyed out).

The pressure field is also another important factor, especially in geothermal applications. Flow rate and underground structures mainly control the pressure field. Fig. 3.5 shows the evolution of the pore pressure field in both injection (blue) and production (red) wells. As shown in the figure, compared to solute concentration the pore pressure is less sensitive to structural uncertainty. But, it can be clearly seen that the connecting fault still affects the pore pressure field, particularly at the production well. The pressure drop at the production well is varying about 8% while at injection well it is less than 1.2%. The difference in the aperture and permeability of faults makes this evolution curve asymmetric. Production fault has less aperture and permeability compared to injection fault (Table 3.1) which makes the pressure drop more severe than the pressure rise. Transmissivity variations of three faults caused such pressure fields in injection and production wells.

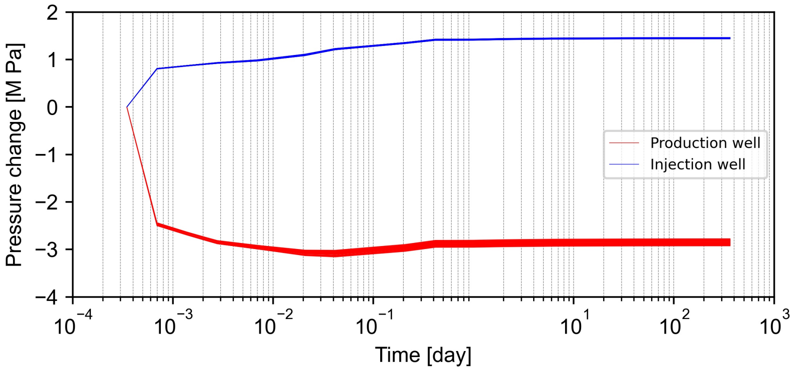

Fig. 3.5 Pressure changes during one year of 45 l/s injection and production for 50 different geological realizations.

3.3.3 Controlling mechanisms of the tracer concentration

In the designed model, the tracer is passing through three different conduits (the injection, connecting and production faults) and also the matrix. It is usually assumed that each medium contributes to the flow based on its transmissivity. For example, three conductive faults are mainly moving the tracer from injection toward the production point through advection. Meanwhile, tracer diffusion into the almost impermeable matrix can increase the long-term concentration results but with a lowered peak. Fig. 3.6 visualizes the idea of sequential multi-flow features deployed in this study. The figure shows the tracer distribution on day 54 of tracer injection for the maximum peak realization. Tracer transport within the fault area is mainly advection-dominated while it diffuses into the surrounding matrix. Therefore, both the small and structural scale properties of the main conduits (three faults) are controlling the tracer movement. As this figure shows, the tracer plume moves from the injection point toward the production. Besides the internal properties of the faults, their structural impact on the tracer plume is investigated in the following.

Annotations on Fig. 3.6 provide a schematic view of the designed structural model in order to demonstrate some possibly controlling geometrical elements of the model. Connections of the wells and the certain (as explained before) injection and production faults are fixed for all 50 realizations. Two filled stars represent where the injection and the connecting faults (P1) and also the production and the connecting faults (P2) meet each other. Accordingly, different pathway lengths, names on the as L_{Inj}, L_{Con} and L_{Pro} stem from changes in the location of the connecting fault. In the figure, wells are extended in the x-z plane in the middle of the model (y=500 m), and L_{Inj}, L_{Con} and L_{Pro} are also measured on the same plane. This figure demonstrates one realization (out of 50) where the connecting fault dips upward from the injection well toward the production well and the depth of the P1 is higher than P2. In contrast, downward dipping is when the connecting fault intersects the production fault deeper than the injection fault, i.e. the P1 will be shallower than P2. In case of having a of depth difference less than 10 m between two points (P1 and P2), the connecting fault can be assumed as a horizontal plane.

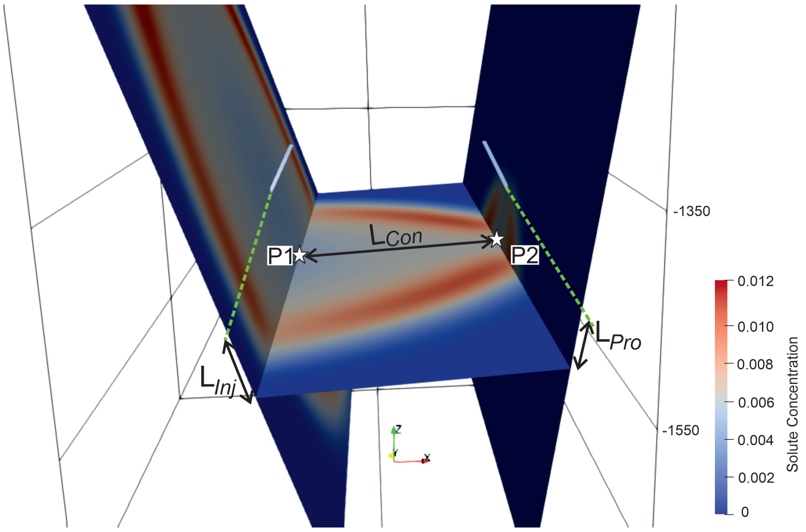

Fig. 3.6 The concentration of the tracer in the peak time (day 54) of the realization with the highest concentrations magnitude. Tracer plume is observable as a red ring. The most important geometrical parameters are annotated. The length of injection (L_{Inj}) measures the distance from the connection of the injection well and fault to the intersection of injection and connecting faults (P1). The length of production (L_{Pro}) represents the distance from the connection of the production well and fault to the intersection of production and connecting faults (P2). The length of connection (L_{Con}) is the length of the connecting faults intersected with the other two faults.

The length of the connecting fault (L_{Con}) is expected to be a controlling geometry. Fig. 3.7 shows L_{Con} and the total length ($L_{Inj} + L_{Con} + L_{Pro}$) versus normalized (based on the median) peak values of all 50 realizations. Surprisingly, Fig. 3.7-a demonstrates a direct relationship between the peak concentration values and the length of the connecting fault due to the overall structure of the model. As shown in Fig. 3.6, ignoring the dipping angle, if the connecting fault goes deeper, its length decreases but the total pathway for tracer transport will be longer and deeper. Because of the model's geometry, less L_{Con} results in higher total lengths. In the case of the total length (Fig. 3.7-b), the concentration decreases by increasing the length (a reverse relationship).

To further investigate the impacts of the connecting faults, all peaks are marked based on the dipping angles of the connecting fault. It is clearly observable in Fig. 3.7-a that upward dipping realizations resulted in higher peak values. This trend suggests that the length of production (L_{Pro}) should be investigated because when the connecting fault is dipping upward, L_{Pro} decreases. Additionally, these upward realizations (and horizontal ones) show a very good linear relationship. As the next step, the impact of the total length of the flow path is investigated in Fig. 3.7-b. The total lengths in most upward realizations are shorter than the median length as its marker signs (crosses) in Fig. 3.7-b are mainly clustered on the left-hand side of the scatter plot. The best linear correlation is found in the realizations with almost horizontal dipping as expected. The relatively weak linear correlation (R^2=0.83) necessitates finding more influential parameters on tracer concentration.

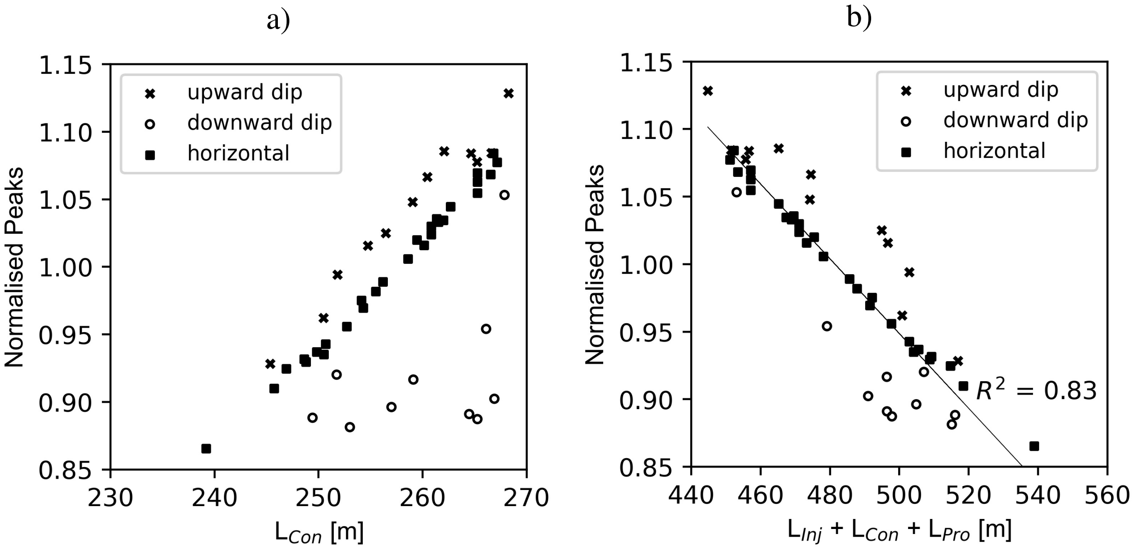
a) b)

Fig. 3.7 a) Normalized peak concentration values plotted versus the connecting fault lengths. b) Normalized peak concentration values plotted versus the total length of the pathways.

Fig. 3.8-a visualizes the normalized peak concentrations versus the production length (L_{Pro}, refer to Fig. 3.6). This figure is colored and clustered based on the recorded peak arrival times. The first clear conclusion is that the less the L_{Pro}, the higher the peak and the earlier the recorded time. Secondly, the correlation has increased and reached a reasonable value (R^2 equals 0.95). L_{Pro} can be considered as the most sensitive geometry factor. In fact, the dipping angle of the connecting faults playswhose a major role because it can control both the total length of the pathway and also length of production.

However, there is still a 5% mismatch between L_{Pro} prediction and simulation results. For example, the green dots cluster representing BTCs whose peaks recorded on day 60 are highly scattered. Minor impact of the other two distances (L_{Inj} and L_{Con}) can cause such complexity in different realizations. Fig. 3.8-b shows cross-sections of the highly scattered (green) and concentrated (orange) clusters. This figure also includes two realizations with the maximum (red line) and the minimum (blue) peak of tracer concentration. In the green cluster, the angle and the direction of the connecting fault are highly diverse while in the orange cluster, the connecting faults are almost horizontal. As expected, dipping angle variations modify both the total length and L_{Pro} that is detected to be a leading parameter. The correlation coefficient can get closer to 1 by excluding the green cluster. This cluster highlights the fact that only one single parameter (L_{Pro}) is unable to predict the tracer peak values with 100% accuracy. Several geometrical parameters are superposed on each other to make such flow regimes and tracer recovery.

3.3.4 Tracer tests design and data inversion

In tracer campaigns, operators usually wait for a predefined amount of time to detect the concentrations of injected tracer. However, a small structural perturbation in the underground can have unexpected impacts on detection time and concentration. Ignoring uncertainty and unrealistic expectations can result in an early shutdown of tests and considering conduits as barrier zones. In reverse, an obscure connecting fault can be misrepresented as an efficient conduit, i.e. a structural bias leads to overestimations for the flow. However, a computationally affordable uncertainty analysis can avoid such hazardous and expensive failures.

BTCs characterize the flow regime [138, 139], and are usually analyzed based on the first detection and the arrival peak time of the tracer concentration in a production well. Tailing in BTCs indicates the presence of stagnation zones. Peak arrival time provides information to calculate the transmissivity of the main flow conduit. Furthermore, longitudinal and transversal dispersion coefficients are estimated based on the time difference of the initial arrival and peak times and bandwidth of the peak. To extract the mentioned information, an inversion process is required where BTCs are usually fitted to measured concentration values based on velocity, length of flow paths, dispersion coefficients and so on. A myriad of analytical methods are proposed in the literature [140, 141, 120] to carry out inversion.

Fig. 3.9 shows two simulated BTCs (out of 50) in this study and their corresponding fitted curves using the Multi-Flow Inversion of Tracer (MFIT) tool [140]. The Single-Fracture Dispersion Model (SFDM) [142] module of MFIT enabling solute transport as advection-dispersion in fractures and diffusion in the matrix, is applied to make these fits. BTCs with

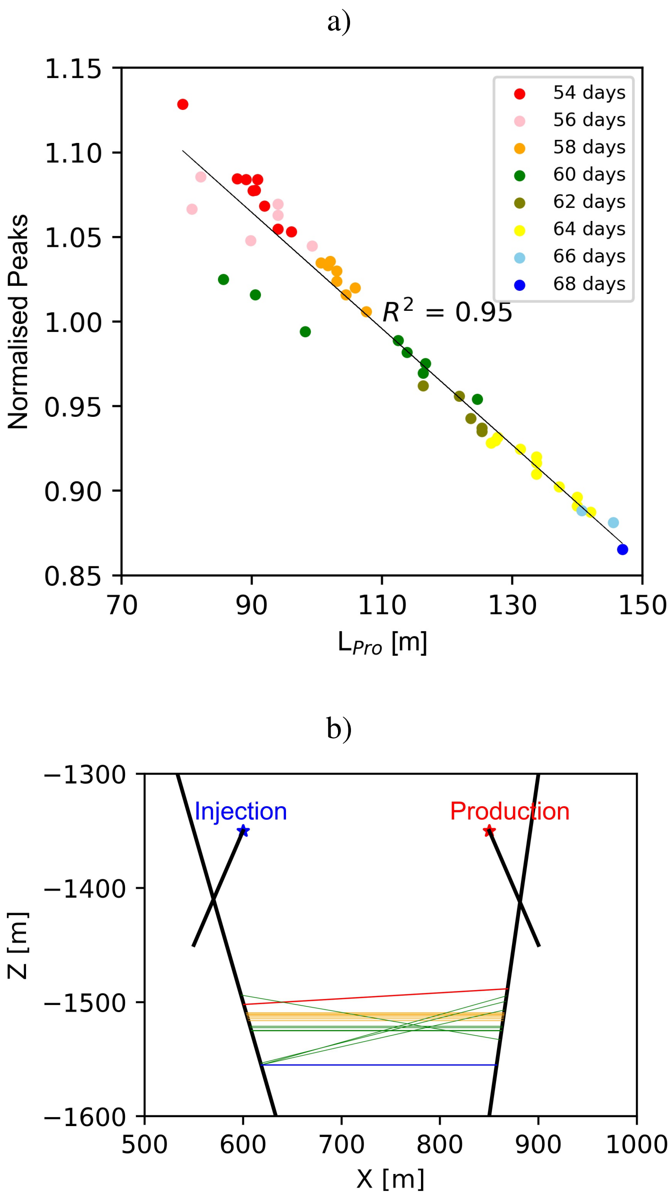

Fig. 3.8 a) Normalized peak concentration values plotted versus the length of the connection in the production fault and colored by the peak arrival times. b- Green lines represent the realizations where the connecting fault orientation is highly scattered (day 60). Orange lines represent the realizations where the connecting fault orientation is the most concentrated (day 58). Red and blue lines also represent the realizations with the highest and lowest peak concentrations, respectively.

the highest and lowest peaks are deliberately selected as the most critical simulations. For both BTCs, the difference between fitted curves and data points is effortlessly observable. The second mode of tracer concentrations (between days 100 to 150) is missing in both fitted curves. Apart from peak magnitudes, the other difference is in the initial arrival time. As discussed in detail, the effect of dip direction in conjunction with the geometry in different realizations controls the flow. However, fitted curves are proposing faster initial arrival times due to ignoring the complexity of the flow path as available analytical approaches miss it.

On the other hand, dispersion coefficients can be simply considered as the standard deviation of a BTC [143]. Fitted curves propose higher standard deviations due to their early initial arrival times. The smaller peak magnitudes of the fits can also reveal that dispersion coefficients are exaggerated in the analytical solver. After day 170 of the simulation, two fitted curves converge toward the same values. This means that the analytical solver considered diffusion as the only process transferring the tracer after this time. However, the absolute values of the tail in the fitted curves are less than half of the simulated BTCs due to overestimating the diffusion. It is noteworthy that the area below each BTC is equal to its corresponding fitted curve.

Providing reliable characteristics of the complicated pathways using analytical approaches can be cumbersome. Mathematically, several curves can be fitted to a bunch of points. Physically, obtaining and measuring coefficients and flow parameters (dispersion coefficients or Péclet number [144]) can be error-prone. For example, two fitted curves in Fig. 3.9 have different Péclet numbers that are set to be 3.4 and 4.3 for fitted curves of maximum and minimum peak concentration BTCs, respectively. The total distance parameter (L_{Inj} + L_{Con} + L_{Pro}) is the reason behind the different Péclet numbers in these two realizations. The different distances arise from structural uncertainty that is often ignored. The following equation correlates Péclet number (*Pe*) with the dispersion coefficient:

$$Pe = \frac{u\ L}{D_L}, \tag{3.2}$$

where, $u\ [m\ s^{-1}]$ is velocity, $L\ [m]$ is total distance ($L_{Inj} + L_{Con} + L_{Pro}$) in flow direction and $D_L\ [m^2\ s^{-1}]$ represents longitudinal hydrodynamic dispersion coefficient for fracture network [145]. In this study, molecular diffusion is assumed to be negligible (solute diffusion equals $4.0 \times 10^{-10}\ m^2\ s^{-1}$ in Table 3.1). The longitudinal hydrodynamic dispersion coefficient is calculated using

$$D_L = u\ a_L, \tag{3.3}$$

where $a_L\ [m]$ is the longitudinal dispersion coefficient. Combining Eq. 3.2 and 3.3 results in

$$Pe = \frac{L}{a_L}, \tag{3.4}$$

where *Pe* is calculated merely based on the advection and dispersion, ignoring the diffusion. Routinely, it is assumed that the total length of the pathway (L: L_{Inj} + L_{Con} + L_{Pro}) is known i.e. fixed. But length can have a major role in changing the Péclet number. Ignoring L of the maximum and minimum realizations resulted in the overestimation of dispersion coefficients in the maximum realization. The analytically calculated longitudinal dispersion coefficients (using Eq. 3) of minimum and maximum peak realizations are 125.6 and 129.4 m, respectively. These two numbers are much higher than the used value in the simulations (15 m as per Table 3.1). The high dispersivity of the fitted curves stems from ignoring the complicated flow direction from injection toward the production well. Higher dispersion of the maximum realization is also due to its lower Péclet number (3.4). Fitted curves propose different Péclet numbers and dispersion coefficients that are caused by structural geology (L) rather than material properties (a_L). However, these analytically calculated variables are unable to reflect the existing uncertainty in the geological model.

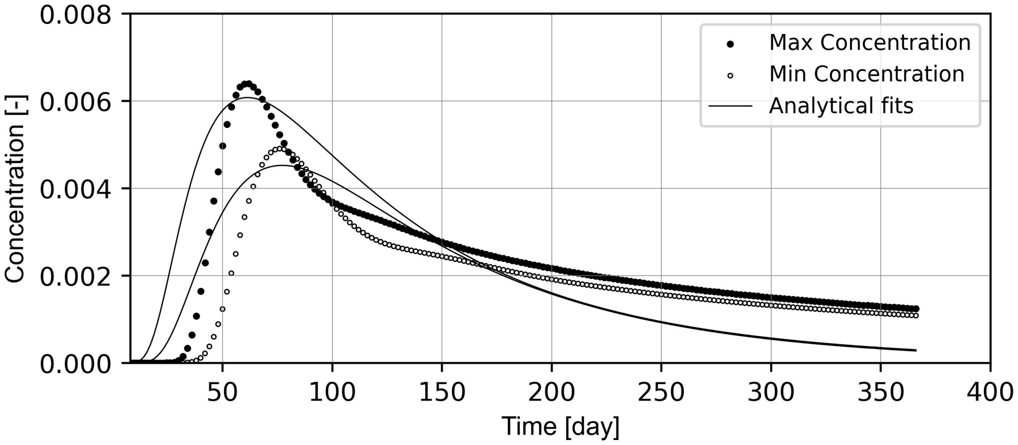

Fig. 3.9 Realizations with the highest and lowest peak magnitudes are shown as black dots and circles, respectively. Their corresponding fitted analytical solutions are represented as continuous black lines.

3.4 Conclusion

In this study an application of the new tool, *GeoMeshPy* is described on the example of a tracer injection in a fractured system. *GeoMeshPy* is at the interface between GemPy and GMSH allowing to automate the generation of customized meshes accounting for well-refined geological structures. This tool can transfer the structural uncertainty of GemPy into a numerical mesh. In parallel, GMSH, has been updated with the new functionality to automatically emulate structures created in geomodelling tools.

The case study was selected carefully to quantify the impact of probabilistic geometries on tracer breakthrough curves, BTCs, in multi-fractured systems. These BTCs represent a typical setting for underground structures explored by doublet borehole systems. Our *GeoMeshPy* based simulations demonstrate that BTCs are strongly affected when varying the orientation of a connecting fault, assumed as a key structural element in the geological model. In contrast to parametric sensitivity analyses, our models were fully remeshed to represent the real geological uncertainty with the highest possible numerical accuracy. The stochastic procedure allowed us to attribute a key importance to the length of the production well (L_{Pro}). The generated, 50 geometries – following a normal distribution – are affecting the length of the total flow circulation, either by shortening or extending. Assuming fixed operational parameters, the pressure values also fluctuated due to variations in the structural model. Both the arrival peak time and concentration magnitude are affected by the introduced uncertainty. Realizations, in which the connecting fault dips upward toward the production, yield BTCs with higher tracer peak concentrations and earlier arrival times due to its shorter pathway on the production fault.

In the next step, the numerically simulated BTCs were inverted for estimating the overall dispersion coefficient, based on standard analytical methods. It is shown that such analytic calculation schemes are unable to reproduce realistic dispersion coefficients as well as the applied structural setting. The calculated coefficients are considerably higher than the simulated values by as much as one order of magnitude. This overestimation is caused by ignoring the complex flow path through sequentially connected fractures. Even the rather simple model setting applied here may pose an unsolvable puzzle for analytical solvers.

The possibilities of applying *GeoMeshPy* in the future are abundant. Although being applied herein in a synthetic case study on BTC, future applications will result in quantifying the geometrical uncertainty for different real concepts like nuclear waste disposal or storage. Furthermore, such structural uncertainty analyses allow to design of field tests like schedule and required mass of tracer.

Code availability

GeoMeshPy is a free, open-source Python library licensed under the MIT License. It is hosted on the GitHub repository https://github.com/Ali1990dashti/GeoMeshPy. The package is also available as a standard python package (https://pypi.org/project/geomeshconv/).

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

Ali Dashti is receiving the financial support from The German Academic Exchange Service (Deutscher Akademischer Austauschdienst: DAAD) to do his PhD in Germany as the Research Grants- Doctoral programmes in Germany 2019/20. This organization is appreciated for giving the opportunity to researchers. The study is also part of the Helmholtz portfolio project Geoenergy. The support from the program “Renewable Energies”, under the topic “Geothermal Energy Systems”, is gratefully acknowledged. Dr. Jacques Bodin is appreciated due to his help on using MFIT and informative comments.

Chapter 4

Developing meshing workflows in GMSH v4.11 for Geologic Uncertainty Assessment of the High-Temperature Aquifer Thermal Energy Storage

This chapter is published in Geoscientific Model Development (GMD): https://doi.org/10.1016/j.geothermics.2022.102607

Abstract

Evaluating uncertainties of geological features on fluid temperature and pressure changes in the reservoir plays a crucial role in the safe and sustainable operation of the High-Temperature Aquifer Thermal Energy Storage (HT-ATES). This study introduces a new automated surface fitting function in the Python API of GMSH (v. 4.11) to simulate the impacts of structural barriers and variations of the reservoir geometries on thermohydraulic behaviour in heat storage applications. These structural features cannot always be detected by geophysical exploration but can be present due to geological complexities. A Python workflow is developed to implement an automated mesh generation routine for varying geological scenarios. This way, complex geological models and their inherent uncertainties are transferred into reservoir simulations. Developed meshing workflow is applied to two case studies: 1) Greater Geneva Basin with the Upper Jurassic ("Malm") limestone reservoir and 2) the 5° eastward tilted DeepStor sandstone reservoir in the Upper Rhine Graben with a uniform thickness of 10 m. In the Greater Geneva Basin example, the top and bottom surfaces of the reservoir are randomly varied ± 10 m and ± 15 m, generating a total variation of up to 25 % from the initially assumed 100 m reservoir thickness. The injected heat plume in this limestone reservoir is independent of the reservoir geometry variation, indicating the limited propagation of the induced thermal signal. In the DeepStor reservoir, a vertical sub-seismic fault juxtaposing the permeable sandstone layers against low permeable clay-marl units is added to the base case model. The fault is located in distances varying from 4 m to 118 m of

the well to quantify the possible thermohydraulic response within the model. The variation of the distance between the fault and the well resulted in an insignificant change in the thermal recovery (1.5 %) but up to a 10.0 % pressure increase for the (shortest) distance of 4 m from the injection well. Modelling the pressure and temperature distribution in the 5° tilted reservoir, with a well placed in the center of the model, reveals that heat tends to accumulate in the updip direction while pressure increases in the downdip direction.

4.1 Introduction

Aquifer Thermal Energy Storage (ATES) yields the highest storage capacities compared to other energy storage solutions [146]. Based on the injection temperature and application, ATES falls into two categories: I) High-Temperature (>50 °C) Aquifer Thermal Energy Storage (HT-ATES; e.g. [147] and II) Low-Temperature Aquifer Thermal Energy Storage (LT-ATES; e.g. [148].

Seasonal storage constitutes a low risk in terms of time, budget and performance [149]. The typically applied "huff-puff" concept of HT-ATES foresees the horizontal transport of large volumes of fluid within an aquifer. Push-pull operation requires a single well for the injection and production [81]. Hence, it is more efficient than the "flow-through" concept, especially in the test phase [150]. HT-ATES provides a significant advantage in its reduced site dependence compared to conventional deep geothermal utilization. It exploits suitable aquifers that can be encountered in the deeper subsurface of major populated urban areas [151, 152]. Appropriate reservoir conditions for heat storage are widely distributed in the uppermost 2 km of the continental crust [153–155, 149, 156]. Suitable reservoirs for thermal energy storage can even exist in thick successions of fractured rocks (e.g. [157]). Another advantage of HT-ATES is its minimal surface area requirement, making it an attractive option in densely populated urban areas [158].

Development of HT-ATES hinges on appropriate petrophysical properties of the deep aquifer that can be used as a reservoir. Such design requires conceptual geological and numerical models. Most HT-ATES studies describe reservoir geometries as homogeneous kilometer scale box shaped volumes. The sensitivity of these volumes to relevant parameters (e.g., well configuration, transmissivity, flow rate, and conductivity) has been extensively studied [159–161, 149, 162]. The conceptual designs of both, HT- and LT-ATES, typically apply box shaped reservoir simulations while disregarding natural geometries and the impact of geological uncertainties.

Establishing HT-ATES in previously exploited oil fields leverages the data and experiences gained from past exploration and production activities. Some depleted hydrocarbon reservoirs are re-used for natural gas storage to meet increased demand during the winter season. Compared to CO_2 [163] or H_2 [164] storage, these depleted reservoirs are yet less commonly used for heat. This scarcity of experience necessitates the development of numerical modelling approaches.

Subsurface data inherently encompass varying degrees of uncertainty originating from measurement errors, biased extrapolations and interpretations, heterogeneities, and simplifications [114, 34, 52]. In this study, we focus on the impact of structural and geometrical uncertainties on pressure and temperature distribution and their spatio-temporal development in heat storage reservoirs. These uncertainties comprise varying morphology of the reservoir roof and floor surfaces and vertical sub-seismic faults that laterally delimit the reservoir, but cannot be predicted from surface measurements. These impacts are often simplified or ignored due to the complexities of re-meshing. Prognostic geological models cannot cope with the uncertainties of the subsurface. Uncertainty analysis highlights the necessity of applying stochastic geological models rather than a deterministic geometrical representation. This study expands the application presented in Dashti et al. (2023) [165] by introducing an automated workflow that generates meshes for complex structural models, enabling the quantification of relevant processes in HT-ATES.

In this study, two potential HT-ATES sites in the vicinity of populated areas are evaluated: 1) the Greater Geneva Basin (GGB) next to Geneva (SW Switzerland) and 2) the designated DeepStor site, located at the campus of Karlsruhe Institute of Technology (KIT; SW Germany). These two locations exhibit significant differences in reservoir geometry, lithology, petrophysical properties, and thicknesses. To assess the impact of structural uncertainties on both the Geneva and DeepStor HT-ATES cases, we designed different scenarios. Quantification of the uncertainty included thickness and geometry variations by adapting a fast, specific meshing workflow. Different scenarios with identical material properties but variating meshes (geologies) are run for each HT-ATES case. The meshing routine generates surfaces from discrete point clouds to create arbitrarily shaped volumes. This automated meshing procedure allows to establish various stochastic numerical models that account for the resolution of the data and even can include an additional vertical fault. Consequently, meshing routines represent the basis for advanced thermohydraulic analyses from arbitrarily inserted faults into the model.

4.2 Uncertainty and Numerical model developments

4.2.1 Greater Geneva Basin

The HT-ATES system proposed for the outskirts of Geneva is situated within the GGB and is designed to store the excess thermal energy, up to 35 GWh, from a nearby power plant [166]. For details on the geology of the GGB refer to Kuhlemann and Kempf (2002) [167]. Two formations are recognized as potential heat storage reservoirs: Upper Jurassic Malm limestones and sand rich layers in the Cenozoic Molasse sediments [168]. The geothermal gradient for the GGB is equal to 25-30 K km^{-1} [169, 168]. The 2530 m deep geothermal well (Thonex-01) intersected >900 m thick Malm limestones and marl succession with a bottom hole temperature of 88 °C and low flow rates of <0.5 l s^{-1} [170]. The gradient is not very promising for geothermal heat production from the reservoir, but heat storage can efficiently support the higher heat demand during the winter season. The flow rate has also been low due to the reservoir's characteristics in that specific location.

Collignon et al. (2020) [166] conducted a local parametric sensitivity analysis on the Molasse and Malm limestone reservoirs of the HT-ATES. The proposed target Malm limestones consist of patch reefs with high porosities (Chevalier et al., 2010; Rybach, 1992). In their scope study, Collignon et al. (2020) [166] assumed a box shaped reservoir with flat surfaces at the depths of -1100 and -1200 m. assumed a box shaped reservoir with flat top and bottom surfaces at -1100 m and -1200 m depths, respectively. Our study simulates the pressure and temperature fields in the geometrically variating Malm reservoirs while the material properties are fixed and identical. We investigate the impact of the geological uncertainty caused by the carbonate reservoir. Such uncertainties typically stem from the exploration of a reservoir structure that can be based on earlier seismic data acquisition [171, 104]. The sources of error comprise data acquisition, preprocessing, stacking, migration, availability of well data for depth calibration, quality of velocity models for time-depth conversion, and ambient noise level [172, 55].

To perturb the geological model, a randomized error is superimposed on the top and bottom surfaces of the initial box shaped reservoir. This error is introduced randomly due to the lack of any real geologic model. This study follows the work performed on a generic box with flat surfaces by Collignon et al. (2020) [166]; consequently, the considered uncertainty also remains generic and random numbers are chosen as the error values to avoid any bias. However, statistically independent geometry points can result in an underestimation of uncertainty [173]. The points can vary independently which results in high variance but the lack of spatial correlation reduces the difference between the average of each scenario [173, 70].

For the top surface, a range of ± 10 m arbitrary error is imposed on the original flat plane. For the bottom surface, the range of perturbation is increased to ± 15 m due to the decrease in the quality of seismic data with depth. The reasoning behind these arbitrary values of 10 m and 15 m, as well as their increase with depth, is elaborated by Lüschen et al. (2011) [174] and Stamm et al. (2019) [115], respectively. The availability of the well data allowed for well-to-seismic tie which increases the accuracy. In the geological model, it is assumed that at intersections of the wells with the top (-1100 m) and bottom (-1200 m) surfaces of the reservoir, the depth value is a certain data. A simplified 2D schematic is presented in Fig. 4.1a to visualize the process of assigning generic uncertainty to the depth data of the GGB. As shown in the figure, the base case assumes the simplest geometry and all scenarios must pass through the four certain points.

For the Malm limestone reservoir, a grid of discrete points in x, y, and z coordinates of a 3D space (representing surfaces) is generated. The regular grid consists of 41×26 nodes in x and y directions, respectively with a fixed 20 m distance. The perturbed model is a purely generic example where at each grid point the random error is added to its vertical coordinate similar to the 2D example in Fig. 4.1a. For the grid points representing the top surface, any value from -10 to +10 has been generated and added to their initial vertical coordinates, i.e., -1100 m. The same process applied for the bottom surface but with a bigger range of error (-15 to +15). In realistic cases, geological surfaces may be subjected to other sources of uncertainty. For instance, a function could be defined to establish a direct relationship between the error value and the distance from the wells, addressing spatial correlation. However, this approach could lead to generating a reservoir with concave or convex surfaces, while meshing highly complex surfaces is one of the contributions of this study. Fig. 4.1-b presents a scenario with two perturbed surfaces of the Malm limestone layer. The irregularity of the reservoir's undulating surfaces is observable in this figure. The entire discretized model includes basement, reservoir, and caprock as lower, middle and upper units, respectively.

4.2.2 DeepStor

The proposed DeepStor site is located in the Cenozoic sediments of the Upper Rhine Graben (URG) and aims to use an abandoned and depleted oil field for thermal storage in the sand layers of the Oligocene Meletta beds. For details on the geology and stratigraphy of the URG, refer to Grimmer et al. (2017) [175], Dèzes et al. (2004) [176] and Schumacher (2002) [177] and references therein. Fig. 4.2 highlights the abundance of N-S striking normal faults in the URG that – if suitably oriented in the stress field – facilitate convective fluid flow in fractured Permo-Mesozoic and crystalline basement rocks. Convection in fractured Permo-Mesozoic

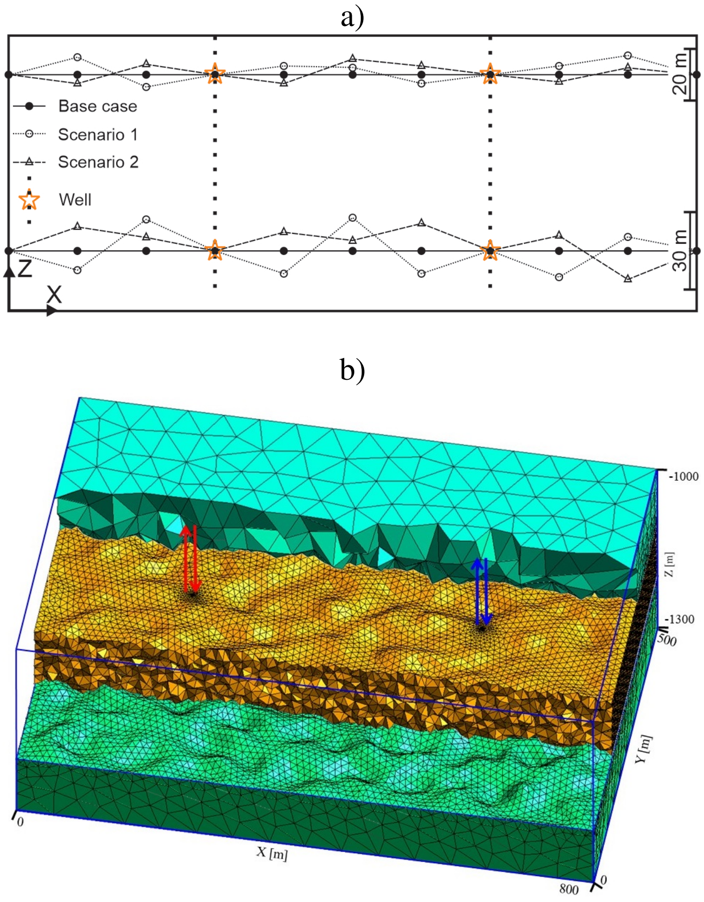

Fig. 4.1 a) The solid line passing through black dots represents the base case. In each of the three scenarios, the geometry of the reservoir is different but all the lines pass through the orange stars which highlight the contact points of the wells and reservoir. b) The entire discretized model of a perturbed scenario. The reservoir layer in the middle is sandwiched by the basement and caprock units. Red arrows represent the injection and production operations in the hot well whereas the cold well is shown with blue arrows.

rocks creates positive thermal anomalies in the Cenozoic graben filling generating locally geothermal gradients of up to 100 K km^{-1} [178–180]. DeepStor designated HT-ATES aims to utilize the Oligocene Meletta sandstones that were exploited for oil from 1957 to 1986 [181] in the footwall of the sealing Leopoldshafen fault where oil and some gas accumulated updip [182, 183].

The DeepStor model in this study encompasses a volume with 1000×1000 m^2 area and 250 m height (see Fig. 4.3-a with the sand layers of the Meletta beds). Due to the inherent uncertainties, sub-seismic faults characterized by offsets <20 m cannot be accurately identified using either 3D seismic or well data. These faults can laterally delimit reservoir layers and impact heat storage potentials and operations [184]. Mathematical models have been developed to characterize these faults because of their abundance and importance [185–188, 51]. While sub-seismic faults are expected to exist, their location in the subsurface remains largely unknown.

To evaluate the impact of sub-seismic faults on HT-ATES operation, a hypothetical N-S striking fault is introduced in different parts of the base geological model. The strike of this vertical fault is parallel with Stutensee and Leopoldshafen faults (Fig. 4.2). The fault remains as a planar 2D surface due to the lack of any information about the hypothetical sub-seismic fault. In this study, the uniform 15 m vertical displacement of the fault exceeds the thickness of the reservoir. This pessimistic assumption enables the prediction of the worst case scenarios for the storage in which a fault completely blocks the reservoir by juxtaposing it against the impermeable matrix. If the offset is reduced and some contacts between the reservoir on either side of the fault are permitted, the effect of the fault diminishes. Our modelling results are also applicable for faults with larger dip-slip displacements.

The single test well (a hot one) is positioned in the center of the model (Fig. 4.3). This arrangement aligns with real storage cases where a test well allows for an optimal design. Data from this well are subsequently processed to establish a potential relationship between measured pressure values and the location of a fault. This study evaluates the impact on reservoir temperature and pressure through thermohydraulic simulations for 16 different fault locations. In total, 17 scenarios are considered in which the parameterization scheme remains the same but the geology (mesh) varies:

- Fourteen scenarios with a fault varying from 4 m to 112 m distances east of the well
- Two scenarios with a fault in the west of the well at 8 m and 48 m distances
- One fault free base case

The 4 m to 112 m range is chosen to evaluate the effect of the fault on the heat propagation and also examine the possible impact of the fault distance on the pressure response at the well location. Fig. 4.3b depicts a scenario with an arbitrary fault located 98 m in the east of the well.

To evaluate the impact of the fault zones, Konrad et al. (2021) [103] considered an arbitrary parameter combination in two scenarios: an aquifer with a geothermal well with and without an inclined fault zone. There are three main differences between the mentioned study and the developed study for DeepStor: 1) here the material properties are fixed, 2) we changed the location of the fault rather than a 0 or 1 flag to include or exclude the fault from the conceptual model, and 3) the fault here is embedded in a way to block the flow rather than increasing the productivity.

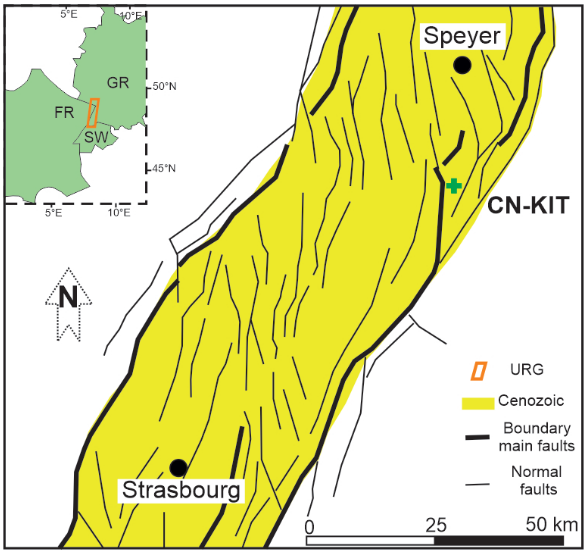

Fig. 4.2 A tectonic overview of the URG and its surrounding area. The green plus symbol indicates the proposed location for HT-ATES in the north of Karlsruhe. Bold lines mark major faults of the rift boundary fault system. DeepStor site is located between Leopoldshafen (in west) and Stutensee (in east) normal faults (modified from Grimmer et al. (2017) [175]).

A simplified example in Fig. 4.4 illustrates how the fault embedding is developed for the DeepStor model. Fig. 4.4a depicts two surfaces with different colours representing the simplified top and bottom surfaces of the DeepStor reservoir. For a better visualization, surfaces are divided into patches and grid points are labelled with numbers ranging from 1 to 36. In reality, a single surface is generated that fits the grid points of the upper surface (18 black dots) and the same for the lower surface (18 black triangles) (Fig. 4.4a). The fault displaces the reservoir layer as shown in Fig. 4.4b. The outline of the fault in the model is represented by thick red lines passing through points 10, 11, and 12 on the top surface and points 28, 29, and 30 on the bottom surface. The first limitation of the workflow is that the fault can only be placed at existing grid points within the geological model. The workflow is developed to incorporate only N-S striking faults that is its second limitation. Another limitation is the dip angle of the arbitrary fault. The script also simplifies faults to be vertical, neglecting the possibility of inclined faults.

The well in the simplified example indicates the certain depths of the top and bottom surfaces in the model. In the faulted example, the top surface will be divided into two splits: the first split includes point numbers 1 to 12 (left hand side of the fault), and the second one with point numbers 10 to 18 (right hand side of the fault). The left hand side split of the fault does not move and only the right one is displaced downward by the amount of the offset. This approach is used in this example because the well is located within the left hand side

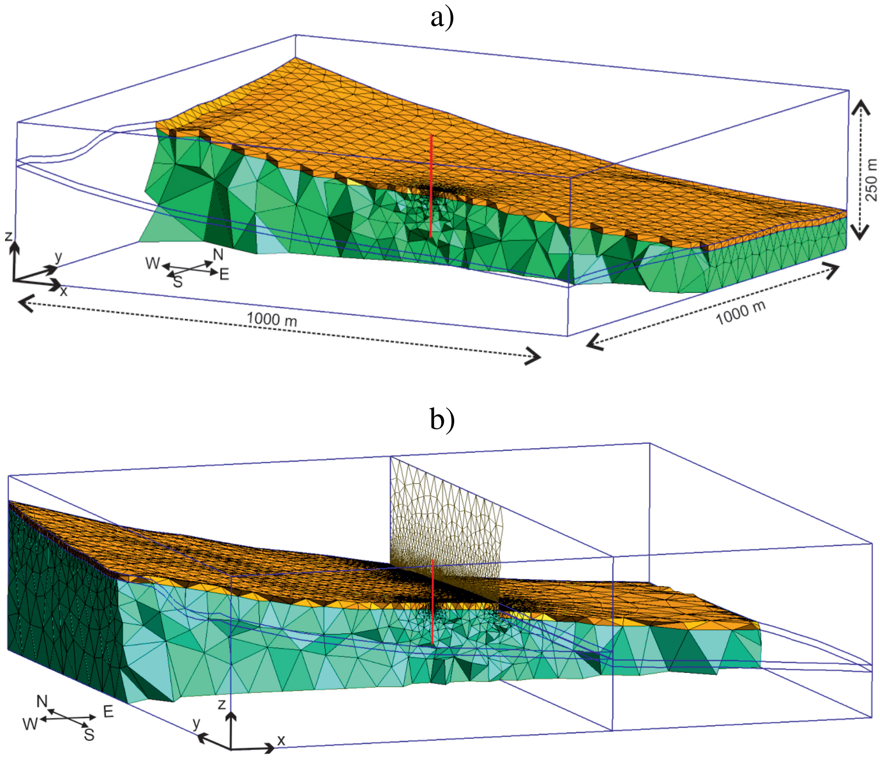

Fig. 4.3 a) A section across the permeable reservoir layer (orange) and basement (green) of the DeepStor base case. Impermeable clay caprock is not shown to have a better view of the discretized model and morphology of the reservoir. b) A fault is introduced in the model. The dimension of the faulted model remains the same as the base case (1000×1000×250). The fault surface of this example is located 98 m east of the well. In both subplots, the well location is shown via a red line.

split. For each split, an extra set of points is also considered to ensure that the split is properly intersected by the fault plane. In the first split of the top surface, point numbers 13, 14, and 15 are added. One single surface fitting to point numbers from 1 to 15 will be generated for this split. Hashed patches in Fig. 4b show how the extra points are allowing the first split of the top surface to extend toward the fault plane. For the second split of the top surface, points 7, 8, and 9 are additionally included. The second split of the top surface passes through 12 black dots numbered from 7 to 18. This surface generation process is repeated for the bottom surface, whose points are represented by black triangles. Finally, the fault plane will also be generated that intersects each split of the top and bottom surfaces. The extra patches and their corresponding points and lines can be deleted after generating the correct geometry. The explained process allows displacing the grid points of the DeepStor base case or GGB.

All the explained steps are implemented and fully elaborated in an example (see Code and data availability section).

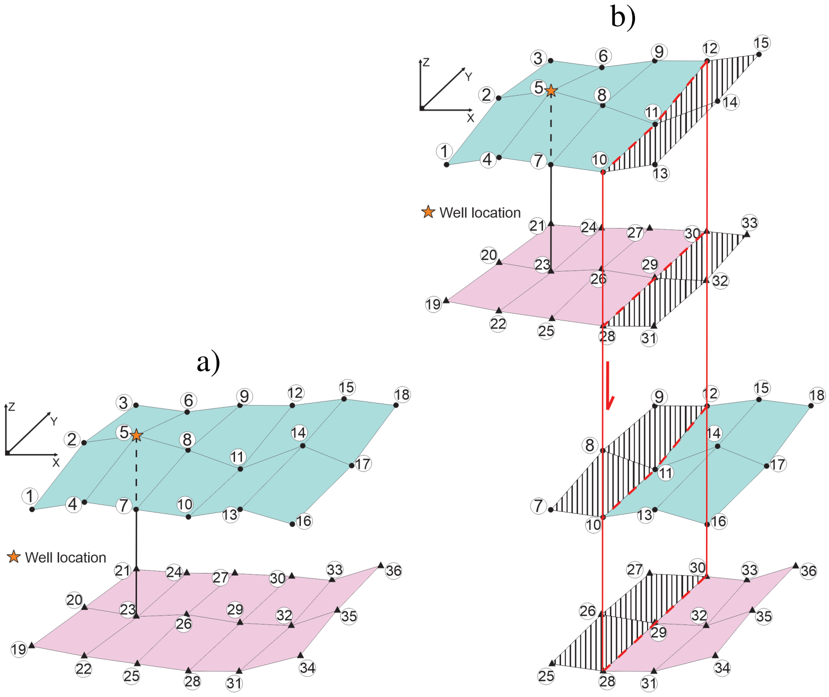

Fig. 4.4 a) The top and bottom surfaces of the simplified reservoir layer are represented via blue and pink patched surfaces, respectively. Black dots represent the grid points of the top surface while the bottom surface passes through the black triangles. The well location and trajectory are shown via an orange star and a black line, respectively. b) A normal fault with an arbitrary offset is displacing the hanging wall (right hand side splits) downward. Hashed patches are the extra ones added to each split.

4.2.3 Tool developments based on GMSH

The open source finite element mesh generator GMSH [38] is used to generate the required high quality spatial discretization. GMSH recently gained the ability to create geometrical surfaces passing through arbitrary sets of points and to combine these surfaces with other geometrical entities (curves, surfaces or volumes) through Boolean operations thanks to the built-in OpenCASCADE geometry kernel (Open CASCADE Technology). The new features linked to B-Spline surface interpolation and non-manifold meshing are available in the latest stable version of GMSH (v. 4.11). This allows to preserve the geological topology of the layers and enables the generation of high quality, adapted finite element meshes for complex geometries like modified Malm limestone reservoir surfaces (Fig. 4.1b) or tilted Meletta beds

(Fig. 4.3). While the model of Dashti et al. [165] lacks complicated geometries, the recently added functionality of GMSH is tested in this study by implementing complex geometries. The overall workflow for the spatial discretization is based on the following steps that are implemented in fully elaborated scripts using the Python API of GMSH (see Code and data availability section):

- The global outline of the domain of study is defined by adding a single (solid) volume – usually a parallelepiped;
- Geological layers are defined by fitting, through numerical optimization, a B-Spline surface going through each set of grid points defining a geological interface. The grid point cloud can come from any modelling tool and the only requirement is that they should make a regular grid. Simplified schematics like Fig. 4 show how the input point cloud can look like. GMSH only requires the x, y, and z values of each point. Default parameters for the B-Spline degree and the tolerances for the fitting ensure a smooth surface with reasonable local curvature changes;
- Sources and wells (or other zero- or one-dimensional features) are defined as additional points and curves in the model;
- All the geometrical entities are intersected globally in order to produce a conforming boundary representation of the complete model, possibly with non-manifold features (points and curves “embedded” in surfaces and/or volumes);
- Mesh size fields are automatically defined to refine the mesh when approaching the boundaries of the reservoir, as well as when approaching the wells and/or the sources;

The global unstructured mesh is then generated automatically. The mesh is made of tetrahedra inside volumes, triangles on the interfaces, lines on the wells and points on the sources. This mesh is conforming, i.e., the elements are arranged in such a way that if two of them intersect, they do so along a face, an edge or a node, and never otherwise. It is necessary to first generate the desired number of scenarios for uncertainty analysis and later on one single block of code in Python will yield the same number of meshes.

In the GGB cases and also the base case of the DeepStor, only two surfaces representing the top and bottom of the reservoir are generated in the mesh. In the faulted cases of the DeepStor (Fig. 4.4b), the grid points making the top and bottom surfaces of the reservoir are discontinuous due to the presence of the fault. Therefore, GMSH should make four different surfaces to reconstruct the faulted scenarios. As visualized in Fig. 4.4b, each split is extended to intersect the fault surface, resulting in some additional small patches. These extra parts can be removed in GMSH before meshing. Fully elaborated Jupyter notebooks are provided (see the Code and data availability section) to detail the meshing process for both the DeepStor and GGB cases.

Multi-level mesh refinement is implemented in both models using various functions available in GMSH. In the GGB case, Distance and Threshold fields enable a gradual mesh size increase from 2 m to 75 m, starting from the wells and extending towards the model boundaries. Additionally, the mesh size is set to 15 m near the top and bottom surfaces of the reservoir and gradually increases to 75 m. On average, GGB meshes contain approximately 35’000 nodes and 210’000 elements. The average is presented due to the variations in the mesh caused by geometrical differences. The fast and automated workflow facilitates the generation of meshes for each complex geological model, such as the perturbed GGB scenarios, within 80 seconds on a Core i7 laptop. Notably, the running time encompasses the entire process from importing data into GMSH to exporting a refined conforming mesh.

DeepStor employs the same refinement strategies but with different mesh sizes. The minimum mesh size is set to 0.5 m near the single well and gradually increases to 125 m. The model also includes a large 2D fault plane. Distance and Threshold fields are introduced for the fault plane, forcing the mesh size to be 3 m near the fault. The DeepStor base case contains 9’026 nodes and 62’317 elements. The mesh is generated in 45 seconds for this fault free case. For the 16 scenarios with the sub-seismic fault, the number of nodes and elements increases to 37’000 and 250’000, respectively. To achieve the specified mesh sizes in both GGB and DeepStor cases, a mesh sensitivity analysis was conducted to ensure the independence of simulation results (temperature and pressure fields) from the mesh size.

4.2.4 Numerical modelling

The open source finite element application TIGER (Thermo-Hydro-Chemical sImulator for Geoscience Research) [77] is used to simulate the heat storage processes for GGB and DeepStor cases. TIGER is developed on top of the MOOSE (Multiphysics Object-Oriented Simulation Environment) framework. As a general purpose PDE environment, the MOOSE framework is fully coupled and encompasses a wide variety of completely implicit solvers [78, 133]. It inherits functionalities from PETSc which is a suite of data structures and routines applied for scalable parallel solution and libMesh that allows for generating and also reading spatial discretization. In our study, the coupled thermal and hydraulic kernels of TIGER are deployed to obtain the evolution of temperature and pressure. To reproduce the results, other MOOSE based applications like GOLEM [189] or available modules of MOOSE, e.g., Porous Flow [84], can be used. In TIGER, the mass transport equation (given by mass balance along with the Darcy velocity) is used to simulate the hydraulic behaviour of the system. For heat transport, TIGER uses the advection-diffusion equation [77]. TIGER simplifies the meshing by enabling a mixed-dimensional problem formulation. Therefore, we considered the wells and faults in the mesh as 1D lines and 2D surfaces, respectively.

Used thermal and petrophysical data for the simulation of both cases are directly obtained from the literature. Table 4.1 contains the values selected for the required parameters in our simulations. Considering homogenous petrophysical properties for patch reefs is highly idealized, but we adhere to the available published data in this instance. Otherwise, a wide range of uncertainty/heterogeneity can be considered for each input parameter. Collignon et al. (2020) [166] used MATLAB Reservoir Simulation Toolbox to simulate the thermohydraulic processes. In this study, simulation results (heat plume propagation and recovery) are compared and benchmarked against their work.

The GGB model includes a doublet system simulated over 10 years. The loading, unloading, and resting phases of the model follow the strategy introduced by Collignon et al. (2020) [166]. Each annual cycle comprises four months of loading, two months of rest, four months of unloading, and two months of rest. The loading phase corresponds to the injection of hot water via the hot well when the cold well is in production mode. Temperatures for hot and cold fluid injection are set to be 90 °C and 39 °C, respectively. Both wells have a fixed flow rate of 10 $\mathrm{l\ s^{-1}}$ but in different directions. The MOOSE control system dynamically updates the temperature boundary condition (BC) during the simulation. In the injection phase, the temperature BC is applied to the corresponding nodes in the model, either set to 90 °C or 39 °C. During the production phase, the temperature BC is deactivated. The time stepping for 10 years of simulation is divided into 10 loading, 10 unloading and 20 rest phases. The piecewise linear function of MOOSE is used to increase the time steps in each phase to have a more efficient numerical convergence. During the first cycle (four months of injecting hot fluid into the hot well and producing from the cold well), the time step size increases from one hour to 10 days. Subsequently, the time step size decreases to one hour at the beginning of the rest cycle and gradually increases to 20 days at the end. At the start of the next four month cycle (producing from the hot well and injecting cold fluid into the cold well), the time step size is forced to be one hour and increases to 10 days. For GGB, the simulation runtime is approximately 3 hours on 12 cores of a high performance computing (HPC) cluster with 62 gigabytes of random-access memory (RAM).

Stricker et al. (2020) [159] introduced the properties of the reservoir for DeepStor in a generic model and we used the data of their reference case (Table 4.1). In our simulations, the geology and consequently the mesh is the major difference to the model of Stricker et al. (2020) [159] while the parameterization scheme remains the same. Rather than the doublet model described by Stricker et al. (2020) [159], a single "push-pull" well is demonstrated in our study. Herein we focus on the thermohydraulic impacts in the near field of a single well in a model with a fault plane. In our meshing procedure, faults (as 2D planes) are integrated only for displacing the 3D elements. They do not have any significance for the MOOSE simulation and can be considered as being only a virtual plane without any physical property because they control the hydraulic behaviour of the model by juxtaposing the reservoir layer against the impermeable matrix. The simulation time is set to 10 years. Hot fluid with a temperature of 140 °C is injected in a six month period, followed by six months of production operation. The MOOSE control system is again applied to switch the temperature BC between injection and production cycles. The flow rate is fixed at 2 l s^{-1} in both the injection and production phases. The time discretization follows the six month cycles and consists of 20 temporal frames for the whole simulation time. Time steps increase from 10 minutes to 10 days in each cycle. Time steps at the start point of each cycle are considered to be shorter in the DeepStor simulations compared to GGB due to the lower thickness of the reservoir and higher complexity of the model. Almost 74’000 degrees of freedom in the faulted scenarios demand an average of 4 hours of computation time on 12 cores of an HPC cluster with 62 gigabytes of RAM. Simulations in the faulted scenarios of the DeepStor are computationally more demanding compared to GGB due to the complexity of the model.

For both the GGB and DeepStor cases, similar approaches are applied for defining boundary and initial conditions. After running a steady state thermohydraulic simulation for each scenario, the results have been applied as the initial condition for transient simulation of that specific case. In both the steady state and transient simulations, two Dirichlet BCs are also applied for the temperature at the top and bottom surfaces of each model. By introducing a function that represents the temperature gradient, MOOSE allows for assigning the correct temperature values to the model. The depth-dependent temperature function is mentioned in the following:

$$T = T_{\text{surface}} + Z \times GT \tag{4.1}$$

where z denotes the depth (in m) and GT the geothermal gradient (in K $km^{-}1$). In the case of pressure, one Dirichlet BC is defined on the bottom surface of the model based on the following function for both the steady state and transient simulations (assuming hydrostatic equilibrium):

$$P = (Z - WT) \times \rho \times g \tag{4.2}$$

where WT represents the water table depth (in m), *rho* is the density (in kg m^{-3}) and g is the gravitational acceleration (set as 9.81 m s^{-2}).

Table 4.1 Parameters selected as inputs from the numerical simulations of two case studies

Parameter		Case studies	
		GGB [166]	DeepStor [159]
Reservoir	Thickness [m]	∼100	∼10
	Permeability [m^2]	9.8×10^{-15}	6.6×10^{-14}
	Porosity [-]	0.15	0.15
	Thermal Conductivity [W/mK]	1.8	2.5
Caprock	Thickness [m]	∼100	∼100
	Permeability [m^2]	9.8×10^{-19}	10^{-18}
	Porosity [-]	0.05	0.15
	Thermal Conductivity [W/mK]	1.4	1.4
Flow rate [l/s]		10	2
Geothermal gradient [K/km]		26	50
Water table [m]		10	10

Neither temperature nor pressure BCs are set on the side faces, hence they follow the gradient. No flow BCs are considered for the side faces of the models. The sizes of the models are also big enough to avoid any interaction between the pressure and temperature values of the boundaries and injection/production operation.

4.3 Results

4.3.1 GGB

The upper and lower contacts of the reservoir are perturbed to investigate their possible effect/s on the heat and pressure distributions in the HT-ATES. The heat recovery of the system has remained unaffected due to its dependence on local temperature values. Despite changing the geometry of the reservoir, propagation of the heat also appears the same for the three presented scenarios of GGB in Fig. 4.5. Temperature values of the highlighted traces in Fig. 4.5 are extracted to visualize the heat plume propagation. The uppermost scenario in Fig. 4.5 is the base case (a box shaped reservoir with flat planes) while the two next ones are named scenarios 1 and 2 in Fig. 4.6. Even after 10 years the heat is still locally propagated, 40 m, around the hot well for the base case and the other two perturbed scenarios (Fig. 4.6). The overlap of all three curves confirms the independence of the temperature field from the introduced geometrical perturbation of the thick reservoir layer. In addition to the three scenarios presented in the study, eight other geometries are meshed and simulated. The results indicate that the storage capacity (production temperature) remains consistent across the simulated scenarios. For further analysis, 101 different geometries are generated and uploaded (refer to the Code and data availability section).

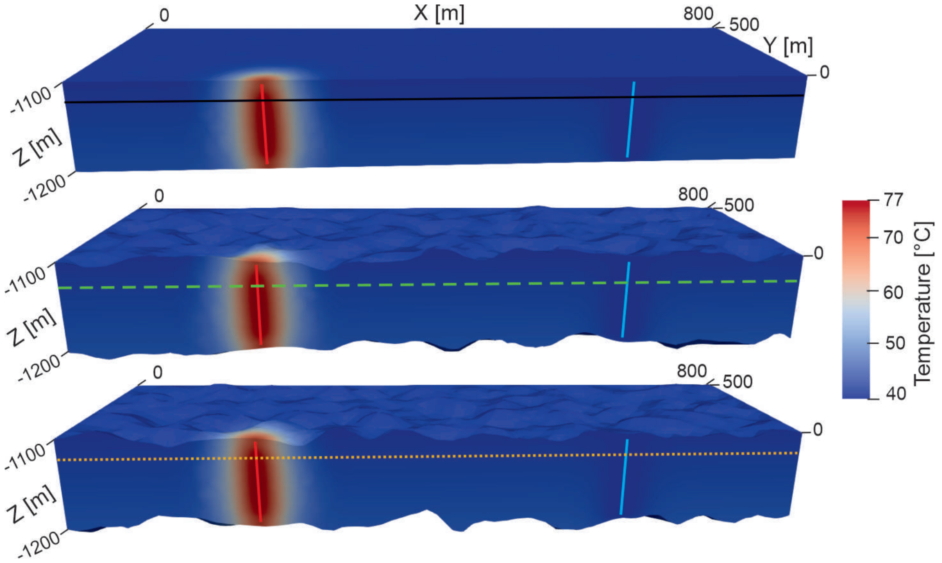

Fig. 4.5 Heat distribution after 10 years of storage in the Malm limestone reservoir of the GBB. Red and blue lines represent hot and cold wells, respectively. The upper scenario with a uniform box shaped reservoir is considered as the base case while contacts of the reservoir in the middle and lower scenarios are perturbed. Solid black, dashed green and dotted orange traces are used in Fig. 4.6 for plotting the temperature values.

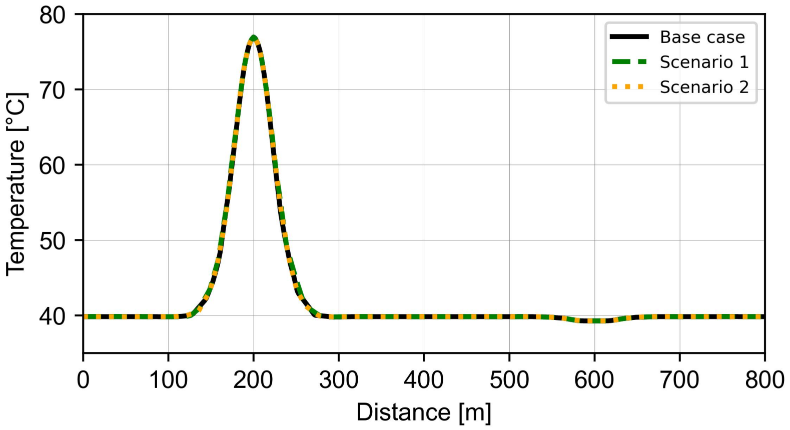

Fig. 4.6 Temperature distribution curves of the values coming from the base case and two perturbed scenarios after 10 years for the GGB. Hot and cold wells are located at 200 m and 600 m points of the x axis. To find the location of the plotted traces, refer to Fig. 4.5. The extension of the model in x direction (Distance) ranges from 0 m to 800 m.

4.3.2 DeepStor

Despite incorporating the reservoir's real geology into this study, the recovery and heat plume radius (45 m) of the base case are similar to what is presented by Stricker et al. (2020) [159] for their reference case. The recovery rate is calculated as the ratio between extracted and injected thermal energy at the top of the well's openhole section. Therefore, this parameter only covers the data from one single point of the 3D model and is unable to see the difference between complex and simple reservoir structural models. Fig. 4.7 shows an increase in heat recovery from 67 % to over 82 % between the first and tenth years. The difference between 17 simulated cases is insignificant (1.5 %). Cases with the highest difference, i.e., extremes, are plotted in Fig. 4.7 to keep the figure readable. The recovery difference between scenarios increases over time, as evidenced by the divergence of the three recovery curves. Despite the negligible difference, the case with a fault located 48 m in the west of the well has the best recovery while the case with a fault at 4 m distance in the east is the worst. For the best recovery, the reason is linked to the total volume of the reservoir and upward movement of the low density hot fluid. The reservoir is tilted and hot fluid moves to the updip direction due to the density effect. Then, a barrier in the updip (west) side of the reservoir can block the movement of the hot fluid and make a more efficient heat storage reservoir. The reason behind the worst recovery is that heat loss happens through the reservoir and matrix contact.

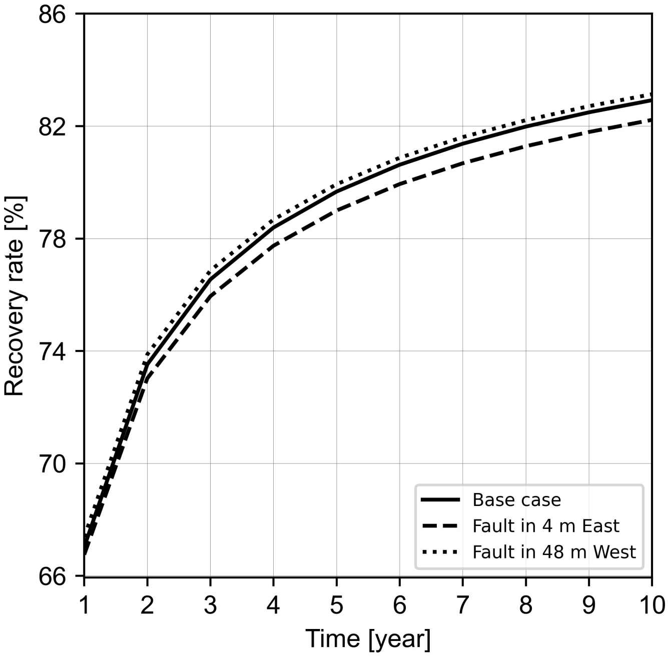

Fig. 4.7 Heat recovery in three scenarios of the DeepStor model. Only two extremes and the base case are plotted to keep the plot more readable.

Fig. 4.8 shows the heat accumulation in four distinct simulated scenarios. In the base case (Fig. 4.8a), the radius and temperature of the heat plume corroborate the results of Stricker et al. (2020) [159]. The heat plume extends approximately 45 m in the x and y directions. The primary distinction is that the heat plume's slope aligns with the tilted reservoir in this instance. The angle between the vertical well and tilted heat plumes in Fig. 4.8 indicates this 5° inclination. The heat plume is most severely affected in the case where the fault is located 4 m in the east of the well (Fig. 4.8b). When the fault is moved to the edge of the plume (45 m in the east: Fig. 4.8c), the heat plume appears nearly identical to that of the base case. The resemblance between Fig. 4.8a and c suggests that the impact of the fault on the heat plume diminishes. The heat plume is getting slightly warmer when the fault is assumed to be 48 m on the west side of the well (Fig. 4.8d). Recovery curves also confirmed the higher efficiency of this scenario. After injecting hot water, it flows toward the updip direction of the reservoir due to its lower density. Over a 10 year simulation time, such localization of the reservoir can increase the recovery but in a longer period, these barriers reduce the available storage capacity of the reservoir.

Fig. 4.9a and b show a 2D section of the model and the total pressure (hydrostatic plus operation induced pressure) values across the sand reservoir after the first injection cycle. Ten injection (and production) cycles are included in the simulation and the maximum pressure increase is observed at the end of the first injection cycle. The plotted trace of the pressure curves in Fig. 4.9b is shown as a dashed line in the cross section view (Fig. 4.9a). The pressure curves illustrate the data of five cases and the initial condition of the trace passing through the reservoir. The pressure increase of the base case at the well location from the initial condition to the end of the first injection cycle is approximately 10 % (11.52 MPa to 12.61 MPa). The initial condition of the model shows that pressure values are distributed asymmetrically in the reservoir. This distribution confirms the role of the reservoir's inclination on pressure in the model. The eastern part of the reservoir layer is dipping downward and under higher hydrostatic pressure. Therefore, in the majority of the faulted scenarios (14 out of 16), the hypothetical fault is located on the eastern side of the well to present the worst case scenarios and enable a better assessment of the maximum potential pressure increase. Even in the worst case scenario (with the fault at 4 m in the east of the well) the pressure value at the fault is only 7 % higher than the value in the same location of the base case. The total pressure at the fault location of the worst case is 13.1 MPa while in the base case, it is 12.25 MPa. Figure 9b also suggests a relation between the pressure increase and the location of the fault.

The impact of the fault on the temperature and pressure fields of one case from DeepStor is presented in Fig. 10. This figure depicts a small slice from the center of the model, spanning

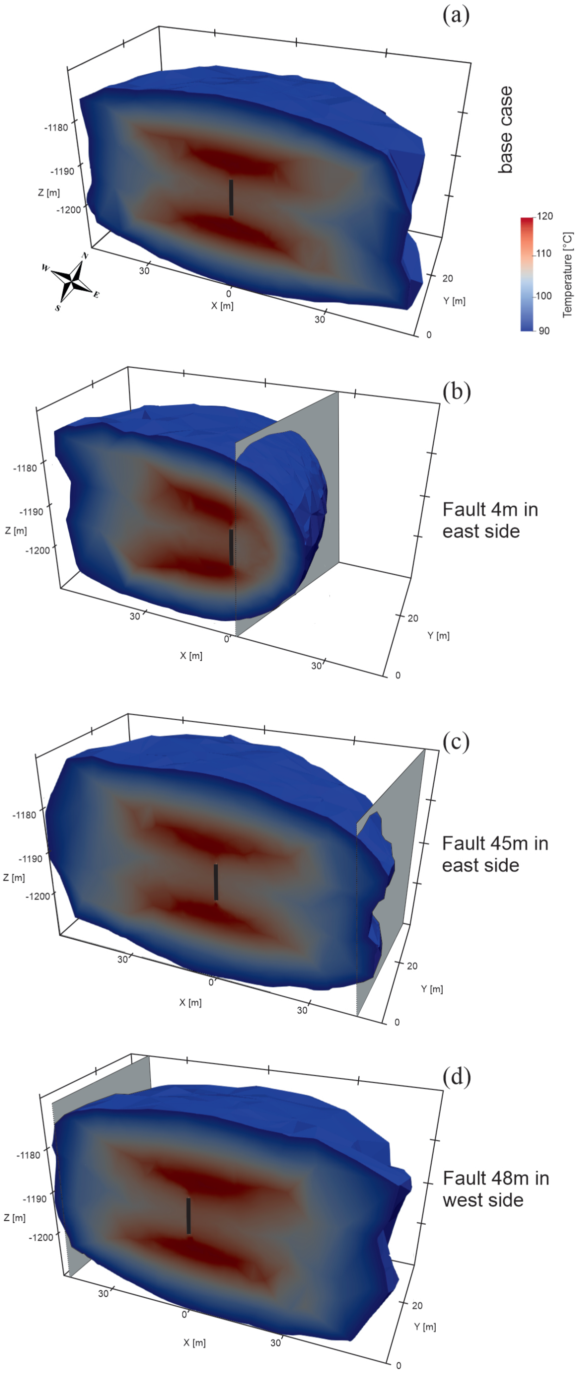

Fig. 4.8 Heat accumulation in four different scenarios of the DeepStor model at the end of the last production cycle. The planned well is shown as a solid black line. Subplots from (a) to (d) represent different scenarios including base case, arbitrary fault (shown with a grey surface) in 4 and 45 m in the east of the well and 48 m in the west. The temperature scale is also the same and shown only once in subplot (a) to avoid repetition.

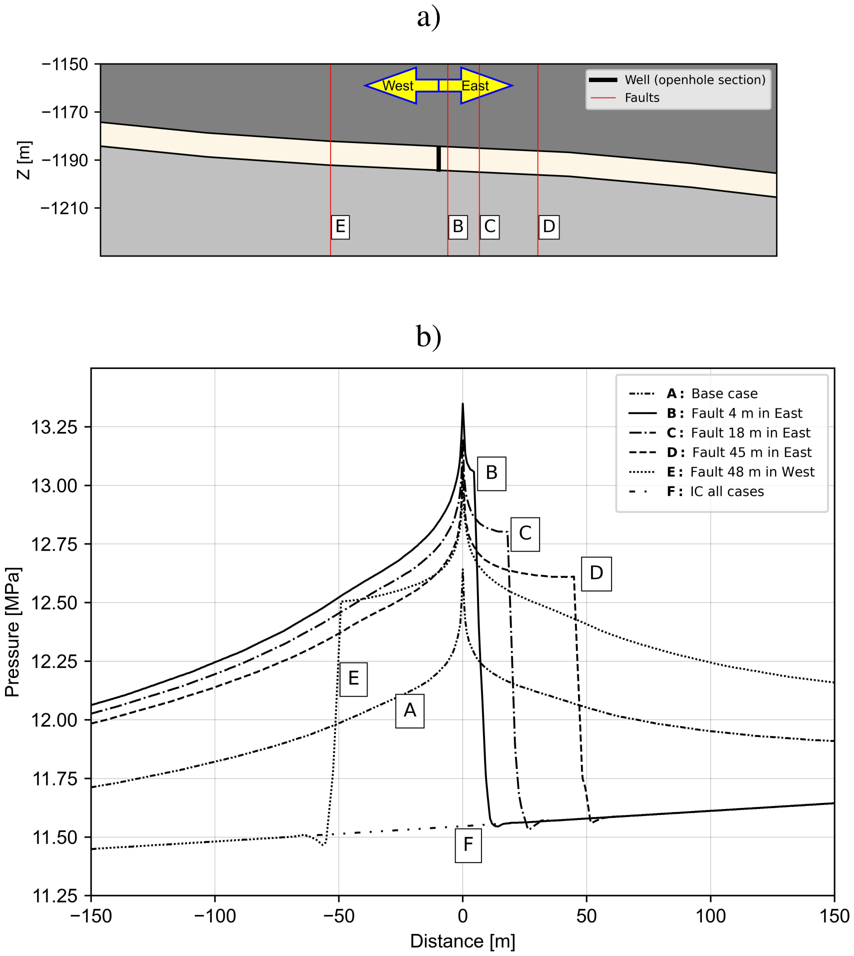

Fig. 4.9 a) The cross section indicates the position of the traces used for plotting the pressure data of five different scenarios and the initial condition (IC). b) Total pressure increase of five simulated cases at the end of the first injection cycle. Negative values for distance represent the western side of the well. To make the curves more readable, scenarios are labelled as A, B, C, D, E, and the initial condition as F.

an area of 600 m by 250 m in x and z directions, respectively. The results demonstrate that the embedded fault effectively creates a barrier very close to the well by introducing a substantial offset. Nevertheless, parts of the injected heat diffused from the reservoir into the matrix, as evident in Fig. 10a.

Fig. 4.11 is a contour plot of the total pressure distribution within the reservoir layer. A surface parallel to the tilted reservoir layer is chosen to create this plot. The trace line shown in Fig. 4.9a is extended in the y direction to transform it from a line to a surface, making it applicable to the contour plots. In both plots, the well is located in the center with 0.0 and 0.0 coordinates. The first notable point is that pressure is accumulating alongside the contact of

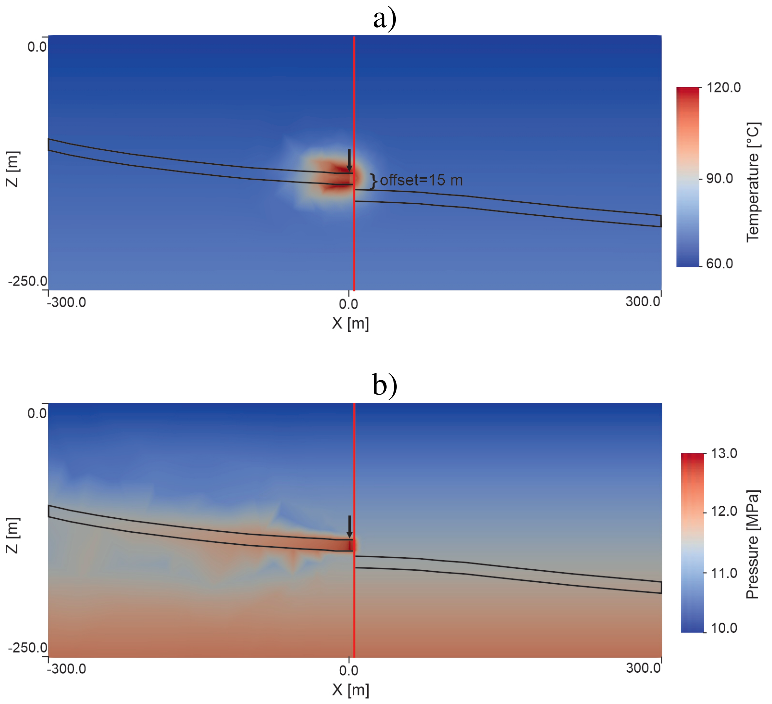

Fig. 4.10 a) Temperature changes in a cross section of the DeepStor model at the end of the last production cycle. b) Pressure regime in the model after the first injection cycle (6 months). In both subplots location of the well is highlighted by a black arrow in the middle of the model. The fault is represented as a continuous thick red line which locates 4 m in the east of the well and has a fixed 15 m offset. The thick black line also represents the boundaries of the reservoir layer.

the reservoir with the matrix. Instead of spherical pressure plumes, contour lines propose an elliptical high pressure regime with a major axis perpendicular to the fault surface. Despite the negligible difference in the fault distance between Fig. 4.11a and b, the pressure values are higher in the case with the fault on eastern side of the well.

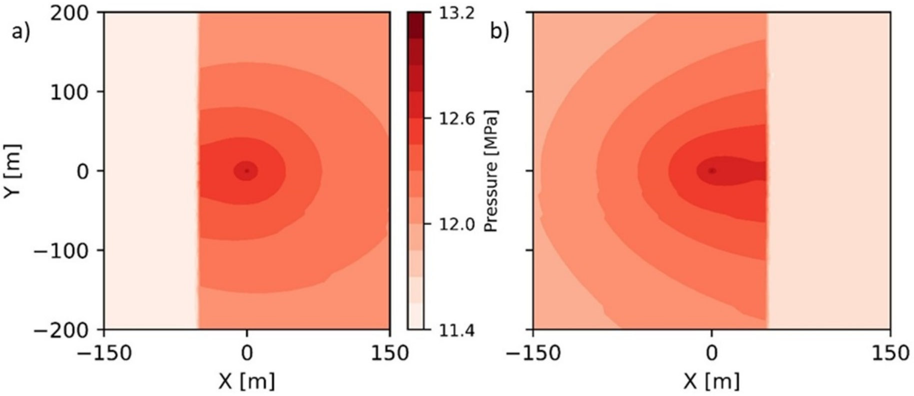

Fig. 4.11 Total pressure changes after the first injection cycle in two scenarios. The well position is in the center of both plots (coordinates=0.0 and 0.0). Fault position is easily distinguishable by the sharp change in the pressure data: 48 m in the west of well (a) and 45 m in the east (b). Negative and positive values for x and y axis are relative to the position of the well.

The presence of the arbitrary fault in the DeepStor model can be identified in the calculated pressure values from the top of the well. Fig. 4.12 shows the history of the total pressure values on the openhole section during the first year of the HT-ATES operation. The location of the fault, either in the east or west, impacts the pressure. The fault distance in the two scenarios is the same (8 m) but in different directions from the well. Due to the pressure accumulation in the downdip direction, a fault with the same distance on the eastern side of the well can increase pressure more than the same one on the western side. The slight difference between the solid black and dashed red curves is detectable in Fig. 4.12.

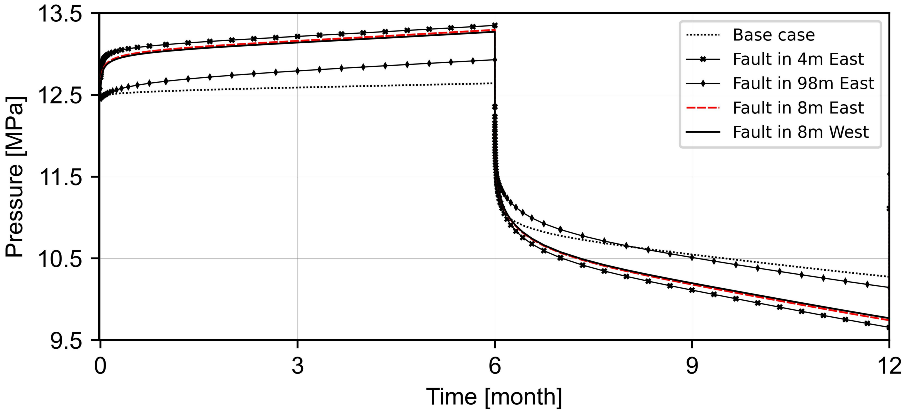

Fig. 4.12 Total pressure evolution in the well during the first injection and production phase. Higher pressure accumulation in the east of the well can be observed by the slight difference between dashed red (Fault in 8m East) and solid black (Fault in 8m West) curves.

4.4 Discussion

The developed meshing workflow streamlines the incorporation of geological models and their uncertainties into numerical simulations. This study used generic initial models and introduced arbitrary uncertainties but the same strategy can be applied to real world cases. The discussion section first addresses the existing limitations of the workflow. The included geological uncertainty is later discussed to be applied in both the exploration and development phases.

4.4.1 Limitations of the Workflow

While offering a starting point for automating the meshing process, the developed workflow has limitations. In terms of surface complexity, the mesh size size should refined enough in the adjacency of the complex surface to build it properly. Meanwhile, the refinement around the surfaces can generate a huge number of elements that increase the computational cost of the numerical simulations. Our current workflow for the DeepStor is limited to creating vertical faults, whereas they can be inclined in reality. To investigate the impact of the dip angle, a new scenario with a 67° dipping fault was compared to the existing vertical fault case. The most extreme case with the fault located 4 m east of the well was chosen for this comparison. As Fig. 4.13 shows the temperature distribution and well temperature profiles are identical for both cases confirming the insensitivity of the simulation results to the considered variations in dip angle. This conclusion applies solely to the DeepStor model with its specific configuration.

The workflow can include only N-S striking faults. Similar to the special case shown in Fig. 4.13, a fault plane with a 5° deviation has been tested but results remained similar to the case with 0° deviation. This insensitivity to fault dip and orientation is specific to the DeepStor model and in other applications and settings, e.g., tracer flow in a multi-fractured reservoir [165], results can be highly sensitive to them. Another limitation is the workflow's inability to place faults beyond existing grid points (vertices) in the geological model. While increasing resolution can expand available locations, faults still remain confined to predefined locations.

Despite these constraints, the chosen scenarios effectively assessed DeepStor's performance risks. The study revealed negligible impact on thermal performance and a maximum pressure increase of 10 % at the injection well for the closest fault location (4 m). Moreover, increasing the upper distance (112 m) further diminishes the impact of the fault.

While constructing the geological model in advanced tools like Petrel, GemPy, or Leapfrog can provide more scenarios for uncertainty analysis, integrating them with mesh

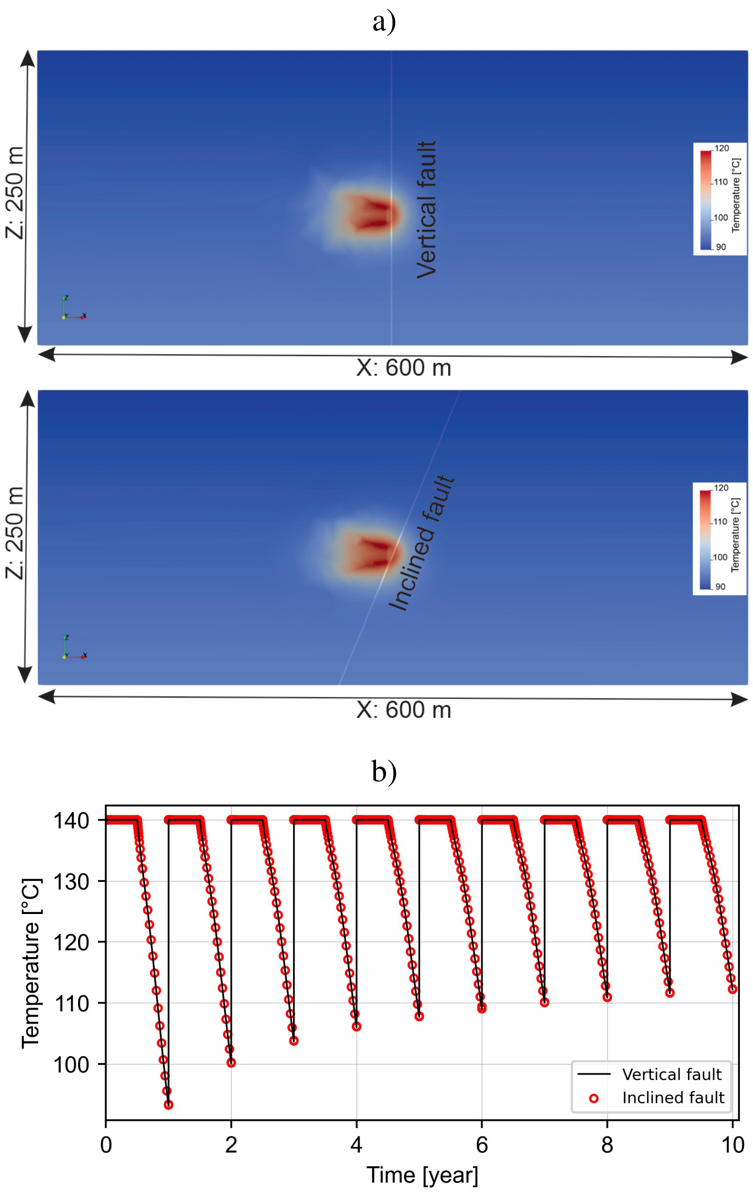

Fig. 4.13 a) Temperature change after the last production cycle in the DeepStor model with a vertical (top) and inclined (bottom) fault. b) Calculated temperature at the top of well over 10 years of simulation.

generators is cumbersome. MeshIt [190], as a mesh generation tool, also aims to address geological models but still requires manual intervention for complex surface creation.

4.4.2 Exploration campaign design

GGB was presented in this study to detect the possible impacts of geometrical uncertainty on the HT-ATES's thermal performance. While all material properties and BCs in our simulations are fixed and derived from the base case of a published document, the geological model, i.e., the mesh, varies. For the chosen parametrization, the heat plume radius even after 10 years of continuous injection and production is still about 40 m around the hot well. Introduced geometrical uncertainty to the GGB case is generic but the proposed workflow is applicable for any real case with its unique complexity/uncertainty. The complex top and bottom surfaces of the reservoir also hardly play any role in the heat distribution of the Malm reservoir. In the case of thinner reservoirs (<20 m) a ± 10 m shift can increase/decrease the volume of the reservoir up to 50 %, but the thermal performance of the Malm reservoir in GGB remained independent of such small scale thickness variations. This fact confirms the unnecessity of complex exploration methods for such specific cases like GGB. Dedicating huge efforts to preliminary steps discourages policymakers from investing in renewable solutions like HT-ATES in settings similar to what has been assumed for GGB in this study. In some cases, existing 2D seismic slices of oilfields can bring enough accuracy to generate reliable forecasts. Computationally affordable geological scenario based analyses of the reservoir can save the time dedicated to exploration.

4.4.3 Field development plan

Based on the presented results for DeepStor, the distribution of both the heat and pressure are tightly linked to the inclination of the thin reservoir. Therefore, incorporating the real geology into the planning process can be a critical factor in optimizing the placement of the second well. As the next step, perturbing the depth, inclination, and thickness of the layer can provide us with a range of possible depths that can be expected during the drilling of the second well.

Within the URG, the majority of hydrocarbons are accumulated thanks to the existence of sealing faults. Therefore, DeepStor can also encounter these structural features. Thermo-hydraulic simulations revealed that only faults located within distances less than the heat plume radius (45 m) can have negative impacts on storage performance. Considering the size of the heat plume, it is highly unlikely to see any effect from or on the Leopoldshafen or Stutensee faults regarding the thermal performance of the system in a 10 year time frame. The target sand layer is very thin and in the case of thicker formations, the impact of faults can be even less important and observable.

The existing trend in Fig. 4.9 and 4.12 enables a primary forecast of fault distance (in case of having any) merely based on the recorded well pressures. The pressure difference between day five of injection and the initial condition versus the distance of the fault to the well is used to formulate the forecast. It is assumed that on day five of injection, the initial reservoir condition and injection operation have reached equilibrium. This pressure value can also be measured through a hydraulic test conducted on the well. In the base case of DeepStor, the maximum total pressure reaches from the initial 11.5 MPa to 13.3 MPa, representing a 15 % increase at the end of the first injection cycle. Notably, over half of this increase (11.5 MPa to 12.5 MPa) is observed by day five of the simulation. Fig. 4.14 shows the relation of these two variables where the fault distance from the well versus the pressure increase after five days is plotted. All the 14 black dots represent the scenarios in which the fault is located in the east of the well. For comparison, the case with a fault in 8 m distance in the west of the well is also plotted as a circle to present the pressure accumulation in the downdip direction. To address the worst case scenarios and be as pessimistic as possible, the forecast has been founded only on the base of the faults located on the east of the well. A simple exponential function with three degrees of freedom provides an acceptable level of accuracy (RMSE=0.013 MPa) for the prediction. More simulations can strengthen the presented forecast scheme. Due to the discussed limitations and lack of enough scenarios, we here present the possibility of formulating such a simple forecast for a complex reservoir. In the case of generating more simulations, advanced methods like machine learning can also be used. Once developed, other arbitrary distances can be fed into the predictor and the pressure value on day five of injection will be returned without making meshes and running the numerical simulations. As a limitation of our meshing workflow, the fault has been located only at specific distances while the proposed predictor can work for any distance.

After conducting the test phase in reality and measuring the pressure value on day five of injection, the data can be inserted into the predictor to back-calculate the distance of the fault (if present). In the case of finding discrepancies between prior assumptions about the fault location and the output of the predictor, the geologic model can be updated. However, the validity of this inversion scheme strongly depends on the accuracy of the chosen modelling assumptions like the material properties used (Table 4.1) and including only one fault. Otherwise, the difference between measured and calculated pressures can originate from any other sources like petrophysical properties. Global sensitivity analyses shed light on the effect of each parameter on the response of the system. In the case of measuring material properties with error levels less than the sensitive range of the system, the proposed forecast scheme can be more reliable for predicting the underground structural model and performing independently of the parametrization.

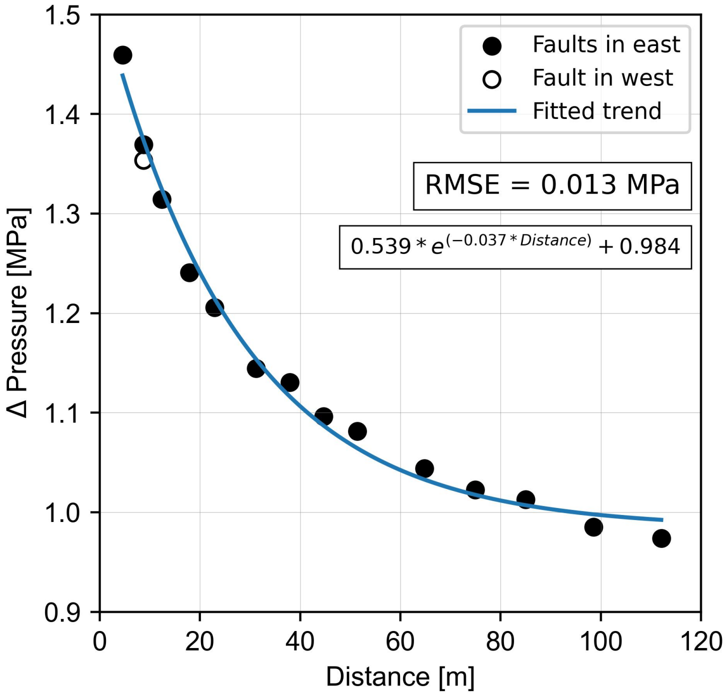

Fig. 4.14 Difference between the well pressure on day five of injection and initial condition (Δ Pressure) versus the distance of the arbitrary fault to the well. The continuous line represents an exponential function with three degrees of freedom.

4.5 Conclusion

In the framework of uncertainty quantification, we have developed a tool applicable to complex geological structures. This study demonstrates a geological scenario based analysis of HT-ATES in two showcases. A new implementation in GMSH provided us with the possibility to automate the generation of complex geological surfaces to overcome the time demanding process. The developed automated workflow in Python brings the possibility to make several meshes composed of surfaces with arbitrary shapes. This workflow also enables a fast generation of finite element meshes using one single block of code in Python without manual adjustments. Generated meshes will link the geological uncertainty of the models to numerical simulators. We used the geological uncertainty as a key input for decision making in different phases (exploration to development) of the HT-ATES.

A HT-ATES is simulated for Geneva as the second most populated city in Switzerland. In GGB, randomly generated geological surfaces are used to assess the sensitivity of results to the geometry of the reservoir rather than the material properties of the model. The GGB model confirms the independence of the temperature from the geometry of the Malm reservoir. The rough structure of the Malm layer can be detected even through 2D seismic slices. Therefore, surveys for finding the exact morphology of the top and bottom surfaces with higher accuracies are unnecessary for such cases. This study highlights the necessity of running computationally affordable simulations before any exploration campaign.

The porous sand layers existing within Meletta beds beneath KIT campus are also promising storage space. For DeepStor adding one more level of complexity (a vertical sub-seismic fault) to interpreted data expresses the performance risks such as possible significant heat losses and/or pressure increase. With the assumed material properties, the presented evaluation on DeepStor proved that only in cases where the fault is closer than 45 m to the well, the thermal performance of the system can be negatively affected. The effect on the thermal recovery of the well is hardly observable but the overall dimension of the heat plume can change due to such faults in the vicinity (<45 m). Numerically calculated pressure values at the well location can decipher the faults even in 118 m distances assuming the fixed and certain petrophysical properties. The relation between pressure changes and the location of the introduced fault is used in this study to establish a case specific forecasting scheme for detecting possible locations of the barriers in the DeepStor model.

The adjacency of the proposed site to oil depleted reservoirs is a big advantage but the real experience of HT-ATES in such locations is still immature, hence first order estimates from risk analyses need to be conducted. Further studies are required to address also the challenges associated with DeepStor including the geochemical interaction or the impact of residual hydrocarbons in the formation. Adding new functionalities to the developed Python script of the DeepStor model can also enable a more comprehensive uncertainty analysis by perturbing the strike and dipping angle of the sub-seismic fault. Integrating geomodelling tools with mesh generators also offers a promising approach to expand the scope of uncertainty analysis beyond solely varying the fault location, allowing for the inclusion of additional degrees of freedom.

Code and data availability

GMSH can be accessed via the published releases on the official GitLab repository at https://gitlab.onelab.info/gmsh/gmsh. Required data and developed workflows for running the model for both of the showcases are fully documented and available in GitHub (https://github.com/Ali1990dashti/GeoMeshPy/tree/main/Examples/Storage_Models) and Zenodo (https://zenodo.org/records/10256834) repositories of the first author.

Author contributions

AD: Conceptualization; Methodology; Simulation; Validation; Code development; Writing–original draft. JCG: Conceptualization; Geological modelling; Supervision; review & editing. CG: Code development; Writing – review & editing. FB: Geological modelling; TK: Conceptualization; Supervision; Writing – review & editing.

Competing interests

The authors declare that they have no conflict of interest.

Acknowledgment

Ali Dashti received the financial support from the German Academic Exchange Service (Deutscher Akademischer Austauschdienst: DAAD) to do his PhD in Germany as the Research Grants-Doctoral programmes in Germany 2019/20. This organization is appreciated for giving the opportunity to researchers. This study is part of the subtopic "Geoenergy" in the program "MTET—Materials and Technologies for the Energy Transition" of the Helmholtz Association. The authors are grateful to Dr. Guido Blöcher, Prof. Guillaume Caumon, and Prof. Florian Wellmann for their insightful reviews and comments that significantly improved the quality of this manuscript. Dr. Mauro Cacace is acknowledged for his fast and efficient editorial handling. Authors appreciate the support of Prof. Eva Schill for the data availability and geological model of the DeepStor. Dr. Denise Degen is appreciated due to her support and constructive comments. Fruitful comments of Kai R. Stricker regarding the numerical modelling section are wholeheartedly acknowledged.

Chapter 5

Machine learning for robust structural uncertainty quantification in fractured reservoirs

This chapter is published in Geothermics: https://doi.org/10.1016/j .geothermics.2024.103012

Abstract

Including uncertainty is essential for accurate decision-making in underground applications. We propose a novel approach to consider structural uncertainty in two enhanced geothermal systems (EGSs) using machine learning (ML) models. The results of numerical simulations show that a small change in the structural model can cause a significant variation in the tracer breakthrough curves (BTCs). To develop a more robust method for including structural uncertainty, we train three different ML models: decision tree regression (DTR), random forest regression (RFR), and gradient boosting regression (GBR). DTR and RFR predict the entire BTC at once, but they are susceptible to overfitting and underfitting. In contrast, GBR predicts each time step of the BTC as a separate target variable, considering the possible correlation between consecutive time steps. This approach is implemented using a chain of regression models. The chain model achieves an acceptable increase in RMSE from train to test data, confirming its ability to capture both the general trend and small-scale heterogeneities of the BTCs. Additionally, using the ML model instead of the numerical solver reduces the computational time by six orders of magnitude. This time efficiency allows us to calculate BTCs for 2'000 different reservoir models, enabling a more comprehensive structural uncertainty quantification for EGS cases. The chain model is particularly promising, as it is robust to overfitting and underfitting and can generate BTCs for a large number of structural models efficiently.

5.1 Introduction

Numerical simulations of physical systems described by differential equations are essential in engineering. Advancements in hardware have enabled computing units to solve coupled nonlinear differential equations, encompassing a wide range of phenomena, from weather forecasting [191] to blood circulation in living bodies [192]. However, these methods are computationally intensive and highly sensitive to specific cases. Besides the huge energy consumption of these computational infrastructures [193], their availability is also limited. Furthermore, parameter tuning, sensitivity analysis [194], and uncertainty quantification [195, 196] demand up to millions of simulations.

Machine learning (ML) methods have gained significant traction across various fields [197, 198], including geothermal applications [199]. In this context, data-driven and physics-informed ML (physics-informed neural network, PINN) techniques are of great interest [200, 201]. PINNs and their diverse descendants are ceaselessly flourishing to replace numerical solvers [202–205]; however, their accuracy and time-efficiency for solving complex problems is still a subject of development [206].

One of the challenges in geothermal applications is characterizing fluid flow through complex underground networks. While the geometry of a fracture can define the general direction of flow, the local variation of petrophysical properties impacts the specific pathways [207]. The enhanced geothermal system (EGS), as an engineered underground reservoir, strongly relies on high flow rate circulation through the impermeable matrix. To enhance the reservoir's permeability, the cold fracturing fluid is injected to create new fractures or reopen the pre-existing ones (e.g. Kohl and Mégel (2007) [126]). Hence, a complex underground fracture/flow pattern can be observed in any EGS example like the model presented by Egert et al. (2020) [123]. Integrating local data coming from wells with field measurements like tracer tests [112] can provide insights into the EGS situation. Tracer test campaigns usually yield breakthrough curves (BTCs), which are widely used to extract properties of the porous media and fracture network. However, each measuring method is error-prone resulting in inherent uncertainty [172, 52]. Therefore, incorporating structural uncertainties in numerical simulations in EGS settings makes the flow forecast more realistic [208].

This study proposes to replace computationally demanding simulations with speedy ML models to quantify structural uncertainty estimations derived from tracer data in two different EGS settings. By state-of-the-art ML methods like decision tree regression (DTR), random forest regression (RFR), and gradient boosting regression (GBR), multifold BTCs are generated on top of pure time-consuming numerical simulations. We train reliable ML models to map geometric data from the uncertain fractures of the EGS reservoir to the simulated BTC. The position of the variating structural elements is used as the input feature, and the entire BTC is chosen as the target variable. The proposed ML model correlates the entire BTC with input features, rather than using a time window to predict the future.

5.2 Methodology

5.2.1 Tracer models

Tracer flow in two different cases are applied in this study. The conceptual model introduced by Dashti et al. (2023) [165] is used here as the first case. The model for the first case is called the 'simple case' because it is a highly simplified version of an EGS with a doublet configuration. The conceptual model contains two main transmissive/open faults that are connected to an injection and production well. There is also an additional sub-horizontal fault/fracture structure making a connection between the major faults at greater depth. However, data related to this structure are subject to uncertainty since this fault is located far from the drilling trajectory, and its existence as a conduit is confirmed only by additional geophysical surveys or hydraulic testing. Fig. 5.1 provides a schematic view of the model, where two sub-vertical faults intersect with the injection and production wells and are labeled as Fault_Inj and Fault_Pro, respectively. The sub-horizontal fault, referred to as Fault_Con in the figure, is represented by thin green lines, as it connects the two major faults. Dashti et al. (2023) introduced a range of structural scenarios and perturbed the location of the sub-horizontal fault 50 times to investigate the impact of structural uncertainty on flow.

To comprehensively evaluate the performance of ML methods, a second, more intricate fracture network model (named as complex case) was developed (Fig. 5.2). The 'complex case' incorporates seven fractures, with two designated as uncertain. The impact of varying these two fractures' depth and dip angle on tracer flow was assessed through 100 scenarios. All scenarios shared identical material properties, while the uncertain fractures' dip and depth were varied. The modelling assumptions of the complex case are similar to the simple case which is already addressed in Dashti et al. (2023) [165].

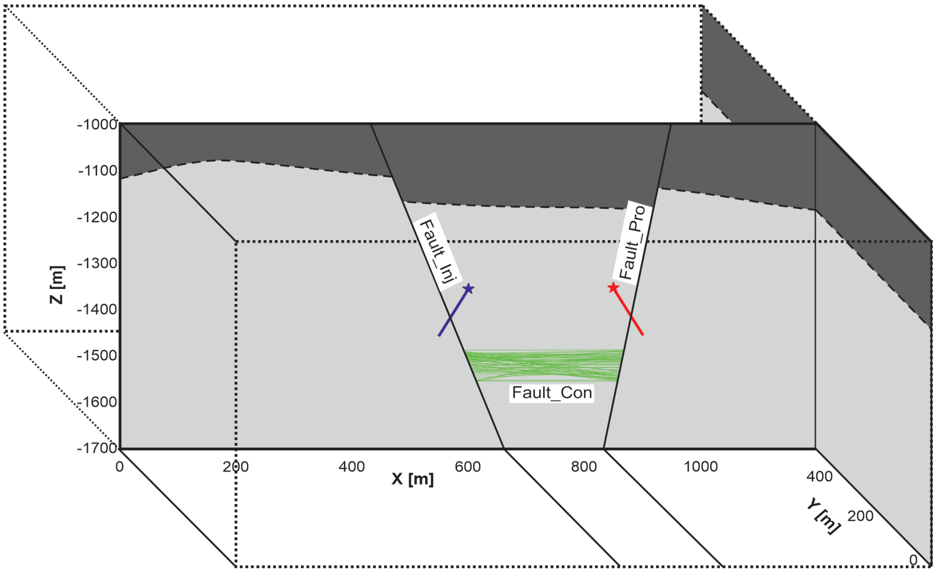

Fig. 5.1 A schematic view of the simple case. The two certain sub-vertical faults (Fault_Inj and Fault_Pro) are shown as continuous black lines and the thinner green lines show traces of the uncertain sub-vertical fault (Fault_Con). Each green trace makes a unique structural scenario.

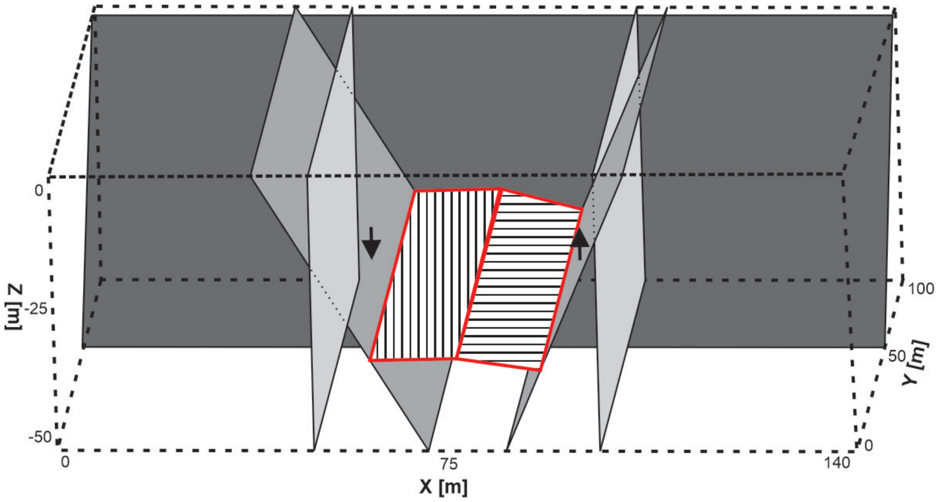

Fig. 5.2 A complex EGS setting with seven fractures. Certain (five) fractures are shown as grey surfaces with varying shades and solid black borders while the two uncertain fractures are highlighted via the thick red border and hashed infill. Two arrows show the location of the injection and production wells.

5.2.2 Machine learning model

The ML model in this study predicts the tracer concentrations over time, i.e. the BTCs for two cases. Time series estimation for different applications is a well-documented topic [209, 210]. For example, Alakeely and Horne (2020) [211] introduced a recurrent neural network to predict the future by incorporating historical data. Such methods predict the system's long-term performance based on a moderate duration of the monitoring data. However, our study predicts the entire time series making the ML models applicable for cases without any historical data. Due to the nature of the problem, two different strategies are developed.

- Strategy 1: Two ML models, DTR and RFR, are trained to independently predict the tracer concentration values. Both models predict the entire time steps of the BTC, using the input features. In this study DTR and RFR correlate structural information of the geological model with the tracer concentration. While in DTR a single tree is trained to capture the relation between the input features and target variable, RFR cultivates several trees in parallel (bagging). DTR is simple to implement and interpret, but it can be prone to overfitting. Therefore, the more complex RFR is also included in this study. The mathematical foundations of DTR and RFR are well-documented in the literature e.g. Kotsiantis (2013) [212], Liu et al. (2012) [213] and XU et al. (2005) [214].
- Strategy 2: A GBR model is used to predict the concentration value at each time step by correlating it with the previous prediction. The GBR is an ensemble method that combines multiple simple and weak learners sequentially (bagging) to improve the overall performance of the model. This approach, denoted as the chain model, requires GBR to be executed for each time step of the BTC. Details of this approach are elaborated in the following.

Decision tree and random forest

A DTR model is a predictive modeling approach that partitions the feature space into a set of regions using a tree structure. The objective is to predict a continuous target variable by learning simple decision rules inferred from the data features [215]. Each node in the tree represents a decision based on a feature threshold, and each leaf node represents a predicted value, typically the mean of the target variable in that region. The algorithm starts with the entire dataset at the root node and iteratively splits the data into subsets to minimize prediction error. The splitting is based on a feature and a threshold value, chosen to optimize a specific criterion, usually the reduction in variance or Mean Squared Error (MSE).

Random Forest regression is an ensemble learning method that combines multiple decision trees to improve prediction accuracy and control overfitting [216]. The basic idea is to create a "forest" of decision trees, each trained on a random subset of the data, and aggregate their predictions.

Chain GBR model

Fig. 5.3 provides an overview of the chain regression model for the simple case. A BTC, serving as the target variable, is presented in Fig. 5.3-a. The input features are composed of the structural geometric information from the reservoir model with the coordinates of four corners of the uncertain sub-horizontal fracture (P1, P2, P3, and P4 in Fig. 5.3-b). The model correlates the x/z coordinates with the BTC concentration values, i.e. the y-coordinate data remain fixed across all scenarios for the sake of simplicity. All the governing equations and modelling assumptions behind the calculation of the BTCs are fully addressed in Dashti et al. (2023) [165]. For the complex case, coordinates of the two uncertain fracture surfaces are used as the input feature while the BTC data are target variables.

The chain model predicts the BTC concentration values in a sequential manner. It starts by predicting the concentration for the first time step (C1) based on the input features (Figure 3-b and c). For the second and following time steps (C2), the model uses the previous values, i.e. C1, along with the input features. Some errors can exist in the predicted C1 by GBR. However, to predict C2, the input feature list still contains 8 coordinate values than have a higher impact compared to the recently predicted C1. This gradual addition of the predicted values can help the chain model to adjust the weight of added features, i.e. previously predicted concentrations. Figure 3-c illustrates how concentration values from previous steps concatenate in the input features' list. To predict the first concentration value (C1) in the GBR chain model, the input feature list initially contains eight values. To predict the concentration for the last time step of the simple case (C169), the input feature list contains eight coordinates and 168 previously predicted concentration values. In the complex case, the BTC includes 140 concentration values. The input feature list of the DTR and RFR models remains fixed, because these two methods predict all the time steps of the BTC merely based on the coordinates of the fractures.

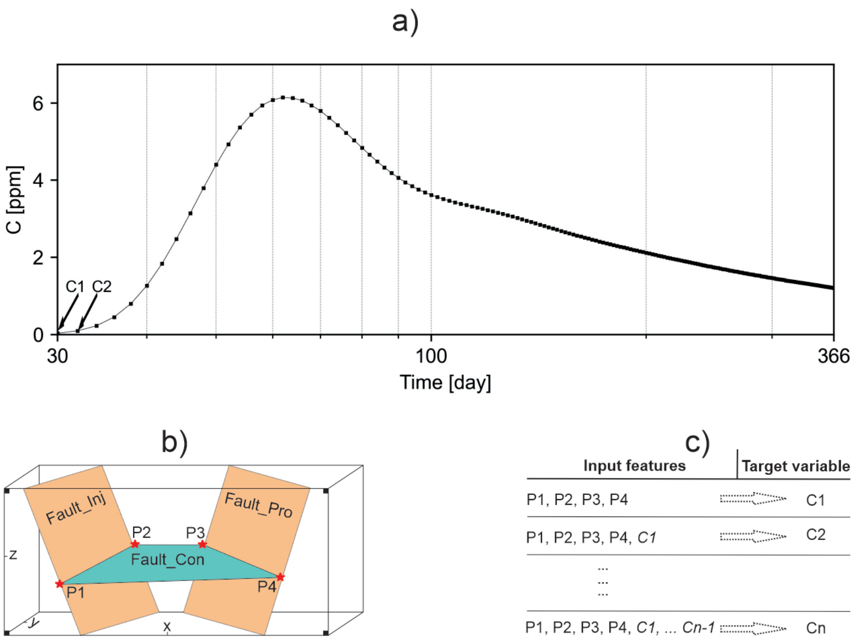

Fig. 5.3 Workflow developed for chain GBR model a) A BTC representing concentration values, C, versus logarithmic time scale. b) Four corners of the sub-horizontal uncertain fault, P1, P2, P3, and P4, are used in the ML model to predict the first concentration value (C1) for the simple case. c) To predict the second concentration value (C2), the first predicted value (C1) is also included besides the coordinates of four corners. In each time step, the previous values are added up to the list of input features.

The GBR algorithm [217] is selected due to its simplicity, bagging nature, and efficiency as a predictor for the chain model. Like other supervised ML algorithms [218], GBR learns a function that maps the input feature/s (x) to target variable/s f(x) with the minimum loss

$$F(x) = \operatorname{argmin} L(f(x), \hat{f}(x)), \tag{5.1}$$

where L is the loss function and $\hat{F_m}(x)$ represents the prediction. The loss function is chosen based on the type of learning (e.g., regression, classification) and the type of the target variable (e.g., discrete, continuous). Squared error (L_2) loss [219] is a simple and efficient loss function when outliers are not expected and is hence chosen as

$$L = \sum_{i=1}^{n} \frac{1}{2}(f(x) - \hat{f}(x))^2. \tag{5.2}$$

ML methods generating an ensemble of predicting models in parallel (bagging methods like RFR) or sequential (boosting methods like GBR) are more reliable than models consisting of a single strong predictive model (like DTR) [220, 221]. Boosting methods like GBR can have a better performance for working on small data sets compared to bagging methods that distribute the data set between different predictors. GBR starts with a very simple model ($F_0(x)$), trying to fit a straight horizontal line (average of target variable). In fact, the derivation of the loss function with respect to the predictions establishes the average value as the best guess for the first tree. In the next round, the GBR algorithm maps the input features to the residuals (remaining errors) of the previous tree, a process that can be interpreted as performing gradient descent on the negative derivative of the difference between prediction and target variable w.r.t. the prediction [222]. The use of residuals rather than absolute values is another reason for choosing GBR. This allows for the inclusion of residuals contributed by recently error-prone predicted concentration values into the model. In subsequent rounds, new decision trees are trained based on the accumulated residuals of the whole ensemble [223]

$$\hat{F_m}(x) = F_{m-1}(x) + \alpha_m(\hat{f}_m)(x), \tag{5.3}$$

where $\hat{F_m}(x)$ represents the final general function that connects input features to the target variable, $F_{m-1}(x)$ contains the information from all previous tress, α is the learning rate that avoids overfitting and $\hat{f_m}(x)$represents the last tree that is correlating remaining residuals and the input features. Low learning rates decrease the impact of each tree, i.e., more trees will be needed but the model also will be more generalized. GBR minimizes the error of each tree and uses the remaining errors as the target variable of the next tree. In this way, the model is trained based on its minimized errors and aggregates several trees with decreasing errors. He et al. (2022) [224] delved into the details of the GBR.

ML model optimization and quality control

Each ML model has two types of arguments: 1) inputs that include hyperparameters (parameters related to the model's architecture) and features selected by the user for predicting the target variable/s, and 2) output arguments that consist of internal weights and the target variable/s. The ML model is trained to minimize the error by tuning its input arguments, allowing the learning algorithm to optimize the output arguments and achieve better scores on the withheld test set [225, 226]. This iterative process, known as hyperparameter tuning [227] involves optimization of parameters such as the learning rate, number of trees, maximum depth of trees, etc. to decrease the error. Determining the optimal number of trees poses a challenge due to the bias-variance trade-off [228, 229]. Another hyperparameter, the maximum depth of a tree, is defined as the longest path between the root node (first node) and the leaf node (last node).

Grid search is a hyperparameter tuning method that allows input arguments to be defined as a range rather than a single value. It performs an exhaustive search over all possible combinations of values to identify the model with the lowest error i.e. highest score. For the RFR model, the number of trees and maximum depth is considered as arrays with 20 and 10 elements, respectively that result in 200 combinations. For the DTR model also maximum depth of each tree, the minimum number of samples in a leaf node and the minimum number of samples for splitting an internal node are tuned. In total, an ensemble of 540 models has been calculated using hyperparameter tuning for the DTR method. In the GBR algorithm of the chain model default values are used. Conventionally, higher score values are preferred over lower ones, and therefore we also tried to find out the combination with the maximum negated MSE using the grid search.

To evaluate the model's performance, k-fold cross-validation [230] has been employed. Rather than splitting the input data into train and test, it randomly splits them arbitrarily into k number of "splits". Then, the ML model will keep one split as the test and all others as the train sets. In the case of splitting data into five splits, the same number of models will be run and in each run, splits will be shuffled. This five-run procedure will be performed for all the assumed 200 combinations of hyperparameters in the grid search for the RFR method. Therefore, it finally creates 1'000 ML models – each of them being an ensemble of individual trees – and the ensemble with the highest score will be used for the final prediction. In this study, we follow the recommendations in the literature [231, 232] and use five splits for cross-validation for all three methods. Training (online) time for the 1,000 ML models of the RFR model on a Core i7 laptop is approximately 10 seconds. For DTR, with an ensemble of 2,700 ML models, the online time remains to be around 10 seconds. The simplicity of DTR compared to RFR results in faster computation. The chain model proves to be the most time-consuming approach, taking around 70 seconds for training without any hyperparameter optimization. Several hyperparameters were tested for the chain model, but the online time only increased without improving the model's accuracy. Therefore, default values were chosen for the chain model. For both the simple and complex cases several values have been tried for the learning rate in hyperparameter tuning but in the end, the default one (0.1) has been used. The required time for predicting a new solution with the trained models (offline time) remains in the range of milliseconds. To access the input data and trained ML models of two cases, please refer to the code and data availability section.

Fig. 5.4-a and b show the distribution of the negative MSE scoring metric in train and test splits, focusing on two tuned hyperparameters of the RFR model in the simple case. The average of the MSE in the four train splits is presented in Fig. 5.4-a. The distribution of the average scoring metric in the train splits is influenced by both the number and maximum

depth of trees. Based on Fig. 5.4-a, the accuracy of the model increases as both the maximum depth of trees and the number of trees increase. However, the score distribution in the test split (Fig. 5.4-b) is more complicated. The scores in the test split are generally lower than those in the train splits (-0.04 to -0.004 versus -0.018 to -0.003). While the score distribution for the train split promises high accuracy of the model by increasing the two hyperparameters, the heterogeneous distribution in Figure 4-b raises doubts on this conclusion. The presented example in Figure 4 concludes that determining the optimal combination even for only two hyperparameters is not a straightforward task. Going to higher dimensions can make the situation more complicated and unsolvable. Therefore, methods like grid search identify the best combination of tuned hyperparameters.

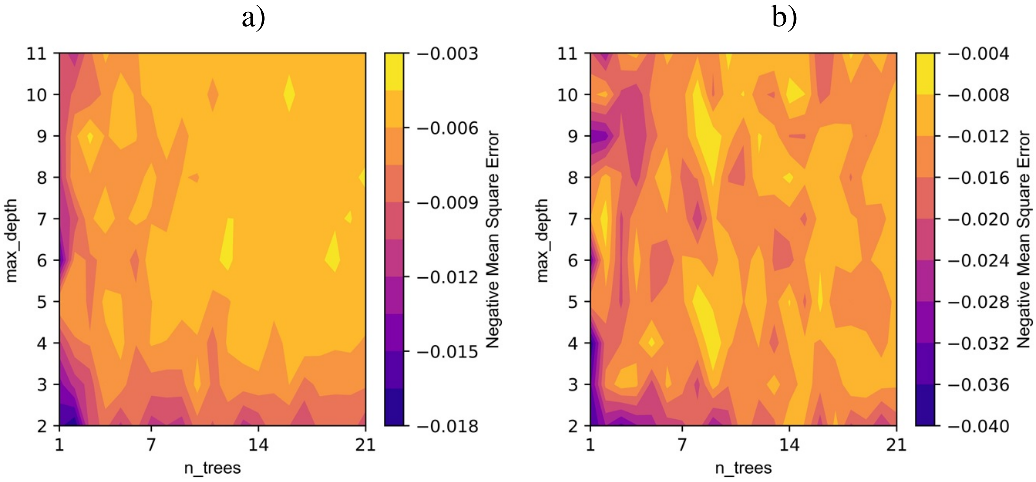

Fig. 5.4 Change in the accuracy of the ML model with respect to different combinations of two hyperparameters of the RFR model on the train (a) and test (b) splits. The accuracy distribution in the train split (a) is smooth and higher accuracies can be achieved by increasing the number of trees and maximum depth of each tree. Subplot b depicts the more patchy and anisotropic behaviour of the accuracy with respect to the hyperparameters.

5.3 Results

5.3.1 Simple case

Dashti et al. (2023) [165] employed numerical simulations to assess the effects of uncertainty in structural models using 50 different structural scenarios in a simplified EGS setting. In these synthetic models, a 24-hour tracer injection on day eight of the simulation was assumed and monitored along a one-year time span in the production well (see Fig. 5.5-a with e.g. peak concentration time varying between days 54 and 68). To better present the variations, a box plot (Fig. 5.5-b) is generated by extracting the highest concentration value from each BTC and normalizing them based on their median. The variation of the tracer peak concentration time, as well as a 25% fluctuation in peak magnitude, emphasize the significance of structural uncertainty, which can introduce unexpected deviations in the results of important field tests. The appearance of a second peak between days 100 and 150 in Fig. 5.5-a is due to the reinjection of the tracer, not multiple flow paths or stagnation zones. The first 30 days of the simulation are disregarded due to negligible concentration (almost zero) of the tracer in the production well during that period.

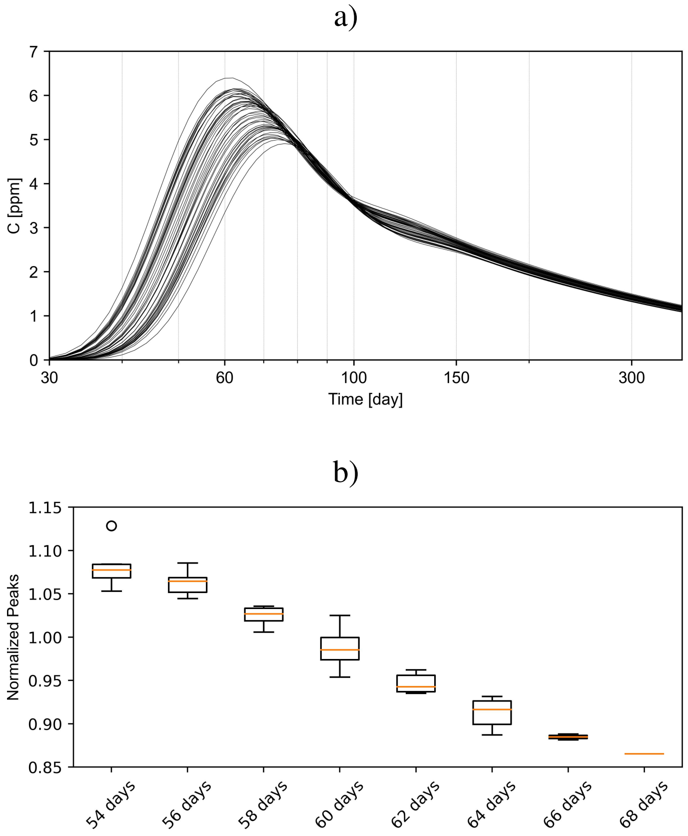

Fig. 5.5 a) Unique BTCs simulated using the finite element solver and used as target variables for the ML models. BTCs are different from each other due to changing structural models. b) A box plot visualizing the normalized peak concentration values versus the time of the calculated peak (analysis based on Dashti et al. (2023) [165]

Results of the k-fold cross-validation in Figure 6 show how RMSE varies in five splits of the three ML methods. The average RMSE of the chain model is lower than the DTR and RFR. Apart from the higher absolute accuracy, the homogeneity of the model's performance is another important factor to consider. Based on Fig. 5.6, RMSE values in the DTR model show higher standard deviations. The higher standard deviation of RMSE for the DTR model suggests that it is overfitting the training data. Overfitting occurs when a model learns the training data too well and is unable to generalize to new data. In the case of the DTR model, this may be due to the fact that it is a single-tree model. Hence it is more likely to memorize the training data than an ensemble model like the RFR or chain model. In this study, the simplicity of the DTR model is the main factor leading to overfitting issues. The RFR model mitigates overfitting by initiating multiple parallel trees that distribute the input data. The chain model also incorporates several sequential models that consistently outperform a single model. Overall, the chain model is the most accurate and robust ML model for predicting BTCs in cases without any historical data. It has a lower average RMSE and a lower standard deviation of RMSE than the RFR and DTR models.

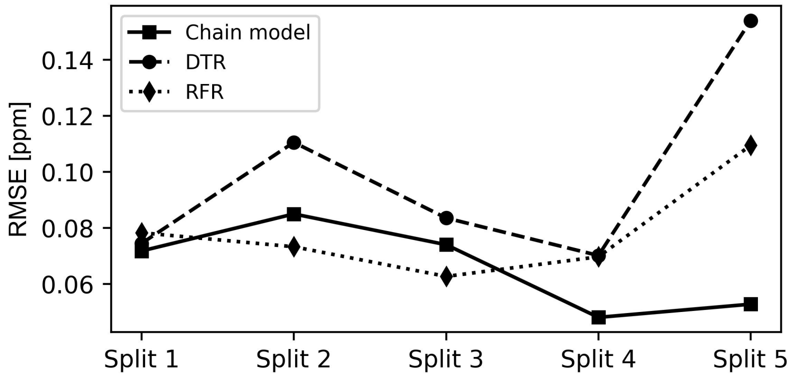

Fig. 5.6 Accuracy distribution of the three designed ML models within their splits. RMSE values are represented as accuracy parameters.

To better assess the trained models and prevent information leakage, two additional scenarios are imported into the three ML models. The trained ML models are then utilized to predict the BTCs of these two new test scenarios . Table 5.1 presents the accumulated RMSEs of these two test scenarios (test set) and models' input data (train set). The ML models exhibit an increase in error when transitioning from train to test scenarios. However, even for the two new test scenarios, the RMSE remains at an acceptable level. The DTR model had the largest difference in RMSE between the train and test sets, which clearly indicates overfitting. The RFR and the chain models yield a better balance in terms of RMSE between the train and test data, suggesting their improved performance and ability to generalize.

Fig. 5.7 shows the numerically simulated BTCs of two test scenarios and the outputs of three ML methods. For one of the test scenarios, all three ML methods achieved similar and reliable results compared to the simulation results. For the other test scenario, the DTR

Table 5.1 RMSE values of the three designed ML models within the train and test sets.

	DTR	RFR	Chain model
Train set	1.1×10^{-4}	1.0×10^{-4}	1.2×10^{-4}
Test set	1.5×10^{-1}	4.0×10^{-2}	5.2×10^{-2}

method was less accurate than the other methods, likely due to overfitting. The RFR and chain models had similar levels of accuracy.

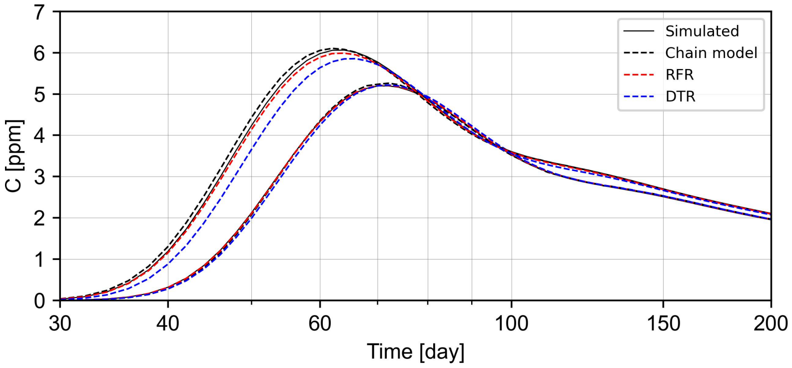

Fig. 5.7 Two different test cases were investigated to understand the accuracy of ML models. The chain model and RFR have a high accuracy in both cases.

To further evaluate the trained models, an additional set of 2’000 different structural scenarios is generated and imported into ML models. In this step, only the connecting fault is perturbed, and the coordinates of its four corners are inputted into the three ML models. Fig. 5.8 provides a visualization of the BTCs generated by the three ML models. These 6’000 BTCs presented in Figure 8 are calculated in the scale of milliseconds using DTR, RFR, and chain models. Two extreme cases from the training data are highlighted with blue colour and dots to illustrate the boundaries of expectations. The RFR method perfectly follows the trend, generating 2’000 almost unique and parallel BTCs (Fig. 5.8-a), which suggests that it may be underfitting the training data. The underfit models have a high bias due to oversimplifications and ignoring the underlying patterns in the training data. This problem can directly originate from the insufficient input data used to train the RFR model. The bagging procedure of RFR splits 50 input data sets into parallel bags making it difficult for each tree to be a balanced predictor. On the other hand, DTR has generated far fewer unique BTCs as shown in Fig. 5.8-b. The covered area with BTC curves in Fig. 5.8-a and b differs dramatically. DTR mainly repeats what it has observed in the training step. As Fig. 5.9 shows, only a few new BTCs are generated and the majority of 2’000 BTCs overlap the 50 BTCs used in the training step.

The chain model consistently generated more reliable BTCs compared to RFR and DTR (Fig. 5.8-c). However, in some cases, the chain model generated BTCs with irregular patterns, such as concentration values fluctuating around the peak. Despite these local discrepancies, the chain model is still the most reliable ML model for predicting BTCs.

Another notable point is that three data-driven ML methods are unable to be used for extrapolation. Even the frequency of generated BTCs decreases close to the extreme point for three subplots shown in Figure 8. This issue is the worst with the DTR method while the chain model has generated more BTCs in the adjacency of the extreme cases.

5.3.2 Complex case

For the complex case, 100 BTCs are simulated in the numerical solver and used to train and test the three ML models. The number of scenarios has increased compared to the simple case (with 50 simulations) due to the complexity of the model. In the complex case, a 24-hour tracer injection on day five of the simulation is assumed and monitored for two months in the production well. The simulation time is decreased due to the shorter/faster connection between the injection and production wells. Fig. 5.10 shows a 2D section of the 100 unique pathways that connect injection and production wells. Two pathways are plotted with red and blue colors and are used to test the validity of the ML methods because they have not been used in the training process. Test 1 scenario visually demonstrates how the two connecting fractures can have different depths and dipping angles.

Fig. 5.11 shows the numerically simulated BTCs for 100 scenarios of the complex case. Similar to the simple case, the peak concentration time and magnitude of the BTCs vary due to the change in the geometrical properties of the fracture network. The color coding of the train and test scenarios (1 and 2) remains consistent with Figure 10.

Two test scenarios of the ML methods are shown in Fig. 5.12. The RMSE values confirm the higher accuracy of the chain method. The cumulative RMSE for both scenarios is 0.05 ppm for the chain model, 0.18 ppm for DTR, and 0.16 ppm for RFR. Notably, all three machine learning models were employed with the same hyperparameters for both the simple and complex cases.

5.4 Discussion

Detailing the observed errors is crucial for future work aimed at improving the interpretability of the ML methods' performance. The (negligible) discrepancy likely stems from the distribution of test scenarios and size of input data (50 and 100 scenarios). This finding

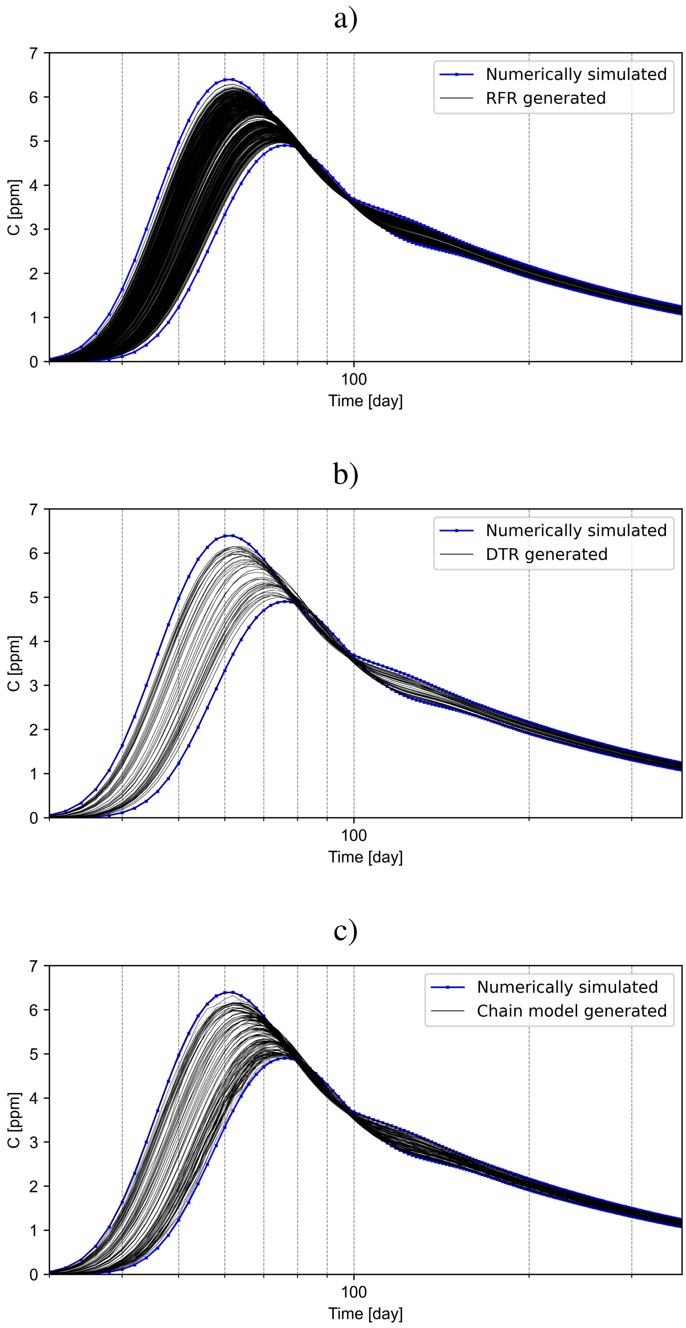

Fig. 5.8 Two thousand generated BTCs using RFR (a), DTR (b), and chain model (c). Two extreme cases coming from the simulation are highlighted as blue curves with dots.

underscores the sensitivity of data-driven models to input data distribution. As extrapolation is a known challenge for such models, selecting a test sample near the boundary in this

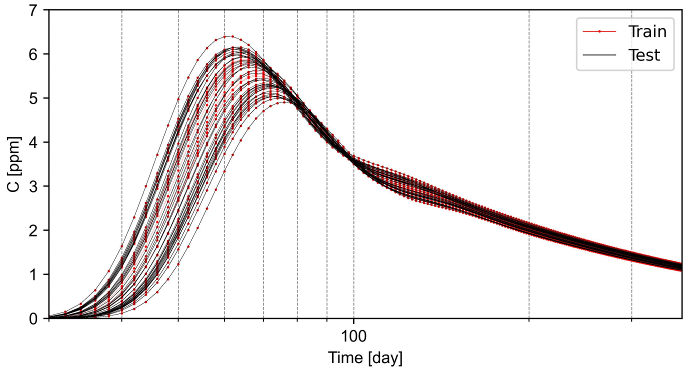

Fig. 5.9 Most of the 2’000 BTCs generated by DTR (named Test and shown as a solid black line) exactly match the input data used for training the model.

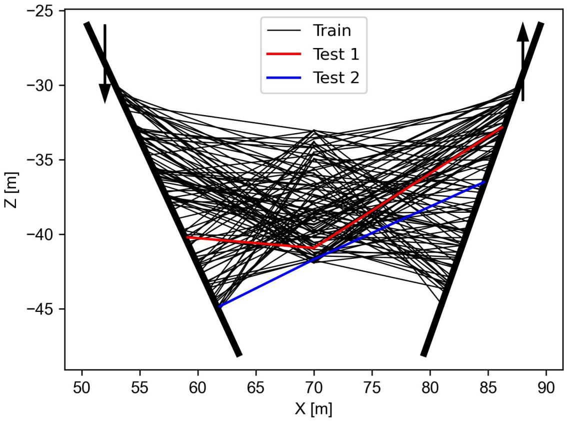

Fig. 5.10 A 2D cross-section from the middle of the complex model. Thin black lines represent the trace of the two uncertain fractures that connect certain fractures shown via two thick black lines. The red and blue traces represent the geometry of the uncertain fractures in two tests. Arrows show the location of the injection and production wells.

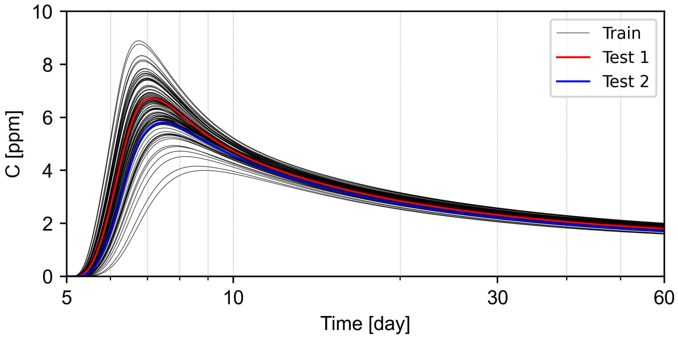

Fig. 5.11 Thin black curves represent 98 BTCs simulated using the finite element solver. Two test scenarios are also named as Test 1 and Test 2. To see the geological model of the test cases refer to Fig. 5.10

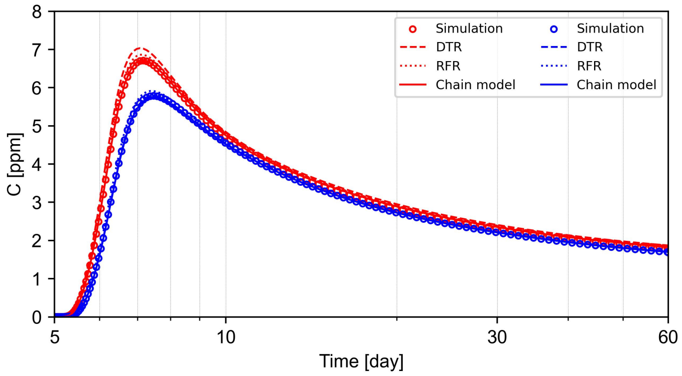

Fig. 5.12 Simulation and ML-generated results for Test 1 are plotted as red circles and lines. Results related to Test 2 are plotted as blue circles and lines.

study exemplifies this limitation. A uniform high-density sampling strategy may prove more effective than the Gaussian distribution.

Even with large datasets, data-driven ML methods can still deviate from the underlying physics. Degen et al. (2023) [206] proposed promising physics-based ML methods using order reduction techniques e.g., non-intrusive reduced basis, to build the solution based on basis functions that preserve the structure of the physics. In this study, we employed a sequence of concentration values as the target variable, allowing the ML models to learn the temporal relationships. Three tested ML methods have been able to capture the trend for two different cases. The current limitation is that the concentration prediction is restricted to a single point within the model. However, our strategy can be extended to develop ML models that predict the target variable at various points over time.

Meanwhile, the ML methods were significantly faster than the numerical solver, with up to six orders of magnitude reduction in computational time. This time reduction has been achieved only after building the ML surrogate models. In most of the feasibility studies, comprehensive sensitivity analyses constructed by several simulations are carried out. To numerically solve the problem of the simple case, 12 cores on a high-performance computing cluster should run for 4 hours. To avoid running hundreds of simulations for sensitivity analysis purposes, some meaningful ratios of the simulations can be run and used to train the ML model to further predict the remaining scenarios. This way, the ML surrogate can pay off the train and test time. The whole time for constructing (offline) and applying (online) the ML models remains in the scale of seconds. This substantial reduction makes uncertainty analysis feasible using fast and reliable ML models, without relying on time-consuming simulations that typically span multiple days. This concept can also be suited for including structural uncertainties in more complicated EGS settings with several intersecting fractures.

5.5 Conclusion and outlook

This study presents a novel approach for using ML methods that enables quantifying the impact of structural uncertainty on BTCs in two EGS reservoirs. The approach was the first test to expand the range of structural reservoir models using ML techniques, based on an original set of a limited number of the numerical scenarios. This meets the specific requirement of uncertainty quantification, which is to provide a broad range of scenarios.

Different ML approaches are trained using the available numerical simulations to predict the BTCs based on the geometries of the perturbed elements. One ML approach used DTR and RFR algorithms to predict the entire BTC at once. Another ML approach employed a chain of GBR models to predict each time step of the BTCs while considering the correlation between consecutive time steps. The DTR model suffered from overfitting, while the RFR and chain models were more reliable, achieving an acceptable accuracy with a balanced accumulated RMSE in train and test scenarios. In the simple case, the RMSE for the DTR model jumped from 0.00011 to 0.15 between train and test scenarios, while for the RFR and chain models, it reached from 0.0001 to 0.04 and from 0.00012 to 0.052, respectively.

The trained ML models are further applied to generate BTCs for 2'000 unique structural scenarios in the model with a simple geometry. The chain model was more accurate than the RFR and DTR models. The RFR method produced 2'000 BTCs that closely follow the trend observed in the training set indicating the underfitting issue, whereas DTR can only replicate the BTCs from the training set. The chain model captures both the general trend and small-scale patterns of the data. However, the accuracy and reliability in all three methods decreases for test cases that are close to the boundaries of the input test data. A uniform sampling for selecting the input data can help the ML methods to have a wide and homogeneous distribution in the test data.

The presented approach can be adopted for a broader number of forward calculation schemes. This opens up new possibilities for more complex fractured rock settings. Rather than coordinates of one/two fractures, a more complex structural network from a real-world EGS case can be used as the input features for the ML methods.

While only structural models were varied herein to assess their impact on the BTCs, future applications could encompass modifications to specific petrophysical properties of the reservoir, further expanding the possibilities of stochasticity. Conversely, integrating more data into the model, such as BTC's or hydraulic testing data obtained from specific EGS well configurations (e.g. Schill et al. (2017) [233]), can reduce structural uncertainties. This allows for the rapid elimination of non-viable models using ML-driven routines.

Harnessing the computational efficiency of ML, this innovative approach can be transformed into a surrogate model, effectively representing the core of an inverse, backward calculation scheme for parameter identification. This transformation has the potential to replace conventional analytical solutions, which are currently the primary method for estimating parameters from tracer campaigns. The ML-based surrogate model offers several advantages, including significantly faster calculation speeds and the ability to capture the non-uniqueness inherent in mathematical solutions. In this framework, BTC data serve as the primary input, while the parameters of the complex EGS reservoir represent the target variables.

Acknowledgment

Ali Dashti is receiving the financial support from The German Academic Exchange Service (Deutscher Akademischer Austauschdienst: DAAD) to do his PhD in Germany as the Research Grants-Doctoral programmes in Germany 2019/20. This organization is appreciated for giving the opportunity to researchers. This study is part of the subtopic “Geoenergy” in the program “MTET—Materials and Technologies for the Energy Transition” of the Helmholtz Association.

Code and data availability

Required data and developed ML methods for the simple and complex cases are documented and available in a Zenodo repository (https://zenodo.org/records/10810243).

Competing interests

The authors declare that they have no conflict of interest.

Chapter 6

Thermo-hydro-mechanical impacts of the operation style (storage triplet vs geothermal doublet) and well placement in the faulted Malm reservoir

---

This chapter is going to be submitted as Thermo-hydro-mechanical impacts of the operation style (storage triplet vs geothermal doublet) and well placement in the faulted Malm reservoir

Abstract

The existing heat in the Earth can be considered as an unlimited source of clean energy. Geothermal applications have targeted this resource via typical reservoir exploitation schemes or recently developed high-temperature aquifer thermal energy storage (HT-ATES) technology that balances the temporal mismatch of the energy demand and supply. Herein we compared the thermo-hydro-mechanical (THM) impacts of two different operation styles: a geothermal doublet and a storage triplet in the faulted Malm reservoir located in the north of Munich (south of Germany). Both operation styles have negligible impact of c. ±0.03 MPa on the pressure field of the Malm reservoir. The continuous injection and production operation of the geothermal doublet perturbs the stress field more significantly. Accordingly, the risk of fault reactivation in the geothermal doublet is twice due to the bigger cold plume induced around the injection well. The induced displacement in the adjacency of the wells is also affected by the operation style with less significant values in the storage triplet (±16 mm versus ±22 mm for the displacement in the y direction). In the storage triplet, as the less risky operation, the distance of the wells (especially the cold one) to the fault surface plays a significant role in the possibility of fault reactivation. A sensitivity analysis on the well location revealed that in the case of locating the cold well in distances less than 145 m, the slip tendency on the fault plane can exceed the critical values in less than 10 years of storage

operation. The exact time of the reactivation also depends on the distance between the fault surface and the cold well.

6.1 Introduction

With the rising energy demands and the trend of greenhouse gases emission, efficient and stable clean energy generation is key to a sustainable future. Geothermal systems present a significant resource for abundant clean energy. Both the classic reservoir production schemes in hydrothermal systems [234] and storage applications [235] are proven to be practical for CO_2-free power/heat generation. The South German Molasse Basin stands out as a premier region for geothermal power generation utilizing hydrothermal resources [236]. This prominence is attributed to the highly productive Upper Jurassic Malm aquifer [237]. Deep geothermal applications in this region target the Upper Jurassic carbonates. These units, reaching thicknesses of up to 600 meters, comprise limestones and dolomites. Notably, these carbonate sediments exhibit facies variations and have undergone multiphase karstification [238]. The lithospheric bending caused by the Alpine Orogeny further influenced the reservoir's characteristics, promoting extensive normal faulting [238, 239]. As stated by Konrad et al. (2021) [103], the aforementioned factors exert a significant influence on the spatial distribution of hydraulic properties within the Malm reservoir.

A geothermal project within the Molasse Basin experienced a premature thermal breakthrough, jeopardizing the long-term sustainability of heat extraction [240]. Premature thermal breakthrough in reservoir exploitation is primarily caused by challenges associated with cold water plume migration. The continuous 24-7 injection of cold water creates a progressively expanding cold front, potentially reaching the production well and compromising system efficiency [241]. While sensitivity analysis can evaluate how individual parameters, like reservoir geometry, flow rate, well spacing, and permeability, influence overall performance, predicting the thermal breakthrough remains a complex challenge [242–244].

Integrating an aquifer thermal energy storage (ATES) concept into existing geothermal doublets within the Molasse Basin presents a promising optimization strategy. This approach could enhance system performance and mitigate risk. Although fractures substantially accelerate thermal breakthrough in conventional reservoir exploitation [245], their impact on ATES schemes is lower [40]. By storing surplus heat generated during summer months within the Malm reservoir, the high-temperature (HT)-ATES concept can bridge the temporal mismatch between district heating demand and geothermal production temperature [246]. Compared to the typical geothermal reservoir exploitation, storage schemes offer less in-situ pressure, temperature, and stress perturbations. HT-ATES operates with minimal environmental impact due to the absence of chemical additives and/or high-energy processes. Additionally, HT-ATES yields long-term reliability, with properly maintained systems delivering stable heating for decades [146, 247]. These advantages position HT-ATES as a compelling solution for sustainable district heating and cooling in municipalities like Munich (Germany).

A crucial challenge in HT-ATES [147] engineering is the dynamic response of the underground media. Thermal changes induced by the operations alter the mechanical properties of the rock, while the hydraulic behavior also undergoes a complex interplay with these human-induced (thermal and mechanical) modifications [248]. Therefore, a comprehensive understanding of the coupled thermo-hydro-mechanical (THM) processes is essential [249, 250]. Quantifying the coupled THM processes in geothermal applications remains a significant scientific hurdle. The inherent complexity arises from the strong inter-dependencies between thermal transport, fluid flow, and deformation of the rock matrix. Analytical solutions for THM problems are often scarce or nonexistent, and replicating the complex interactions (within the reservoir) through experimentation proves equally challenging [251]. Compounding these challenges are the prohibitive costs and extensive time requirements associated with experimental investigations [252]. These limitations (time, safety, technical constraints, cost, etc.) render numerical modeling a more practical approach for exploring THM coupling effects. However, developing robust THM models for subsurface simulations also remains highly challenging due to the complexities of multiphysics interactions and the presence of discontinuities e.g., fractures.

Despite significant advancements in numerical simulations for underground applications, previous studies on geothermal reservoirs have largely overlooked coupled THM calculations. Our study employs a suite of fully coupled THM simulations to evaluate the influence of two different geothermal applications on the mechanical behaviour of a deep faulted reservoir. The fault surface is assumed to be realistic with changing dipping angles and strikes. This study for the first time addresses complex (THM) processes in a structurally realistic model of the Malm reservoir. Assuming a homogeneous reservoir, we compare the mechanical response of a geothermal doublet with the HT-ATES triplet concept to identify a less hazardous approach. Subsequently, a sensitivity analysis of the possible well locations is conducted to quantify the potential risk of fault reactivation for the storage triplet.

6.2 Methodology

This study uses the open-source code finite element simulator PorousFlow module [84] within the MOOSE framework [78]. The PorousFlow module, with a strict quality control procedure, enables fully coupled THM simulations. The relevant equations for the THM simulations of this study are described in detail below.

6.2.1 Governing equations

Fluid flow

To solve the hydraulic field in a THM simulation, PorousFlow uses mass conservation which is described by the continuity equation:

$$\frac{\partial m}{\partial t} + m\nabla \cdot v_s + \nabla \cdot F - q = 0, \tag{6.1}$$

where m is the fluid mass, v_s is the velocity of the solid skeleton, F represents the flux, and q is the source term. The coupling to the solid mechanics in PorousFlow is done via the $m\nabla \cdot v_s$ part of the equation. The changes in porosity and permeability of the solid skeleton (through stress or strain changes) also facilitate the coupling. PorousFlow can handle multi-phase and multi-component flows which is elaborated by Wilkins et al. (2021) [84]. For this study, a fully saturated single-phase and single-component fluid flow is simulated. The flux term F is composed of the advective flux and diffusive-and-dispersive flux:

$$F = F_{Advection} + F_{Diffusion+Dispersion}. \tag{6.2}$$

Due to the single-phase nature of our problem, the dispersive and diffusive fluxed are ignored and the advective flux of the fluid is solved by Darcy's law:

$$F_{Advection} = \rho v = -\rho \frac{k}{\mu}(\nabla P + \rho g). \tag{6.3}$$

where v is the Darcy velocity, P is the pressure of each specific phase (one fully saturating phase in our case), k is the permeability tensor, μ is the fluid dynamic viscosity, ρ represents fluid density, and g is the gravitational acceleration vector.

Heat flow

PorousFlow assumes a local thermodynamic equilibrium for the liquids and solid phases. The energy conservation for heat is described by the continuity equation

$$\frac{\partial e}{\partial t} + e\nabla \cdot v_s + \nabla \cdot F^T - q^T = 0. \tag{6.4}$$

where e the heat energy, F^T represents the heat fluid flux, and q^T is a heat source. Rock grain density, the specific heat capacity of the rock, and the internal energy of the fluid phase are used to calculate the heat energy (e). Again the $m\nabla \cdot v_s$ term of the equation, and changes in porosity and permeability of the solid skeleton allow for the coupling to solid mechanics. Refer to Wilkins et al. (2021) [84] to see the details of plastic deformation in the heat flow.

The heat flux is a sum of heat conduction and convection with the fluid:

$$F^T = -\lambda \nabla T + F_{Advection}, \tag{6.5}$$

where λ is the thermal conductivity tensor, which in PorousFlow is a function of the thermal conductivities of the fluid and rock phases.

Solid mechanics

PorousFlow leverages the available functionalities from other MOOSE module in order to handle solid mechanics and the coupling. The evolution of porosity with pressure, strain, and temperature is fundamental to the coupling between flows and solid mechanics. The effective stress tensor is calculated as [84]

$$\sigma_{ij}^{eff} = \sigma_{ij}^{tot} + \alpha \delta_{ij} P, \tag{6.6}$$

where σ^{eff} represents the effective stress tensor, σ^{tot} is the total stress tensor, α is the Biot coefficient, and δ is the Kronecker delta. Then, the elastic constitutive law [85] links the strain of the porous solid to the effective stress tensor:

$$\sigma_{ij}^{eff} = E_{ijkl}(\varepsilon_{kl}^{elastic} - \delta_{kl} \alpha_T T), \tag{6.7}$$

where E_{ijkl} is the elasticity tensor of the solid skeleton (drained version) which is inverse of the drained compliance tensor, $\varepsilon^{elastic}$ is the strain tensor, α_T is the volumetric drained thermal expansion coefficient, and T is the temperature. The elastic strain tensor $\varepsilon^{elastic}$ is composed of the displacement tensor (u)

$$\varepsilon_{kl}^{elastic} = \frac{(\nabla_k u_l + \nabla_l u_k)}{2}, \tag{6.8}$$

All the 9 stress components and unit normal vectors to each face of the mesh are calculated as auxiliary variables in PorousFlow. Based on Cauchy's law, the traction vector (t) acting on a surface can be calculated based on the stress tensor (n) and unit normal vector (n) [253, 254]

$$t_i = \sigma_{ji} \times n_j, \tag{6.9}$$

which in a matrix form is

$$\begin{bmatrix} t_1^{(n)} \\ t_2^{(n)} \\ t_3^{(n)} \end{bmatrix} = \begin{bmatrix} \sigma_{11} & \sigma_{21} & \sigma_{31} \\ \sigma_{12} & \sigma_{22} & \sigma_{32} \\ \sigma_{13} & \sigma_{23} & \sigma_{33} \end{bmatrix} \times \begin{bmatrix} n_1 \\ n_2 \\ n_3 \end{bmatrix}. \tag{6.10}$$

Then, the normal stress (σ_N) can be calculated as the dot product of the traction tensor and unit normal vector

$$\sigma_N = t^{(n)} \cdot n, \tag{6.11}$$

which is a matrix form will be:

$$\sigma_N = \begin{bmatrix} \sigma_{11} & \sigma_{21} & \sigma_{31} \\ \sigma_{12} & \sigma_{22} & \sigma_{32} \\ \sigma_{13} & \sigma_{32} & \sigma_{33} \end{bmatrix} \times \begin{bmatrix} n_1 \\ n_2 \\ n_3 \end{bmatrix} \cdot \begin{bmatrix} n_1 \\ n_2 \\ n_3 \end{bmatrix}. \tag{6.12}$$

The following equation also gives the magnitude of the shear stress (τ) acting on each surface

$$\tau = \sqrt{(t^{(n)})^2 - \sigma_N^2}. \tag{6.13}$$

In this study, the ratio between the shear and normal stress is used to define the normalized slip tendency (ST) value as

$$ST = \frac{\tau}{\sigma_N \mu + C}, \tag{6.14}$$

where C is the cohesion value and is considered to be 0 in this study. The ST values can be normalized by dividing them with the friction coefficient (μ). Values equal to or greater than the cut-off (1.0) indicate exceeding the friction (slip) criterion.

6.2.2 Model description

A 3-D model of a potential geothermal site in the Molasse basin represents the core of the numerical study. The top of the Malm reservoir is assumed to be at a depth of 1470 m. The lateral extension of the model is big enough (18 km × 16 km) to avoid possible boundary effects. Vertically, the model extends over 4 km with three main layers assumed: the cap rock, 6 reservoir units, and the basement (Fig. 6.1-a). The topmost reservoir unit is assumed to be 30 m thick while the other 5 units have a fixed 100 m thickness. The selected thickness values of the confining cap rock and basement also assure the independence of the results from the top and bottom boundaries. For the geothermal doublet application, two vertical wells are located in the model with a 1250 m distance. Two wells inject and produce fluid within the reservoir unit i.e. 530 m in the middle of the model. To implement the injection and production of the heat and mass in the model, the Dirac Kernel in PorousFlow requires acting on nodes. Therefore, equally distanced points are generated in the mesh to represent each well. For this doublet setting one well is assumed to be used for continuous fluid production, i.e. “production well”, and another well, “injection well”, injects cold fluid with a temperature of 50 °C. The injection and production flow rate is assumed to be 150 kg s^{-1}. The simulation time is set to be 10 years.

For the HT-ATES application, a third well is added to the model. In this triplet setting there is one “hot well”, one “cold well”, and one “buffer well”. The hot well is an alias for the production well of the geothermal doublet but with a difference in the operational mode. While the production well only produces reservoir fluid, the hot well is a bidirectional one injecting hot fluid in the 6-month period of summer and producing fluid in the following 6-month cycle of winter. The unidirectional cold well is also comparable to the injection well while it operates only in 6-month injection cycles rather than a continuous mode. During the hot fluid injection in the hot well, the unidirectional buffer well in the model is considered for the producing the reservoir fluid. This production from the buffer well avoids temporal mass unbalancing. During the production phase from the hot well, the cold well in the model is under injection operation with a 50 °C fluid. Fig. 6.1 shows the location of the mentioned wells in the triplet. The injection and production operations are implemented as Dirac kernel and the MOOSE control system is used to dynamically update it. In this application also the flow rate is assumed to be 150 kg s^{-1}. The simulation time is also set to be 10 years. The whole simulation time is split into 20 injection and production cycles. MOOSE allows for gradually increasing the time steps in each cycle of the simulation. For the first 6 months period, the time step size increases from 1 hour to 10 days. For the second period (months 7 to 12) also the time step size is set to gradually increase from 1 hour to 10 days. This approach allows for a better numerical convergence at the time that the operation mode of the wells changes.

Regarding the initial condition (IC) and boundary condition (BC) both applications follow the same strategy. For the pressure and temperature, depth-dependent functions are defined to set their ICs:

$$T(z) = T_{\text{surface}} + Z \times GT, \tag{6.15}$$

where T is the temperature, z denotes the depth and GT the geothermal gradient. For the pressure, the IC is calculated based on the following function (assuming a hydrostatic equilibrium):

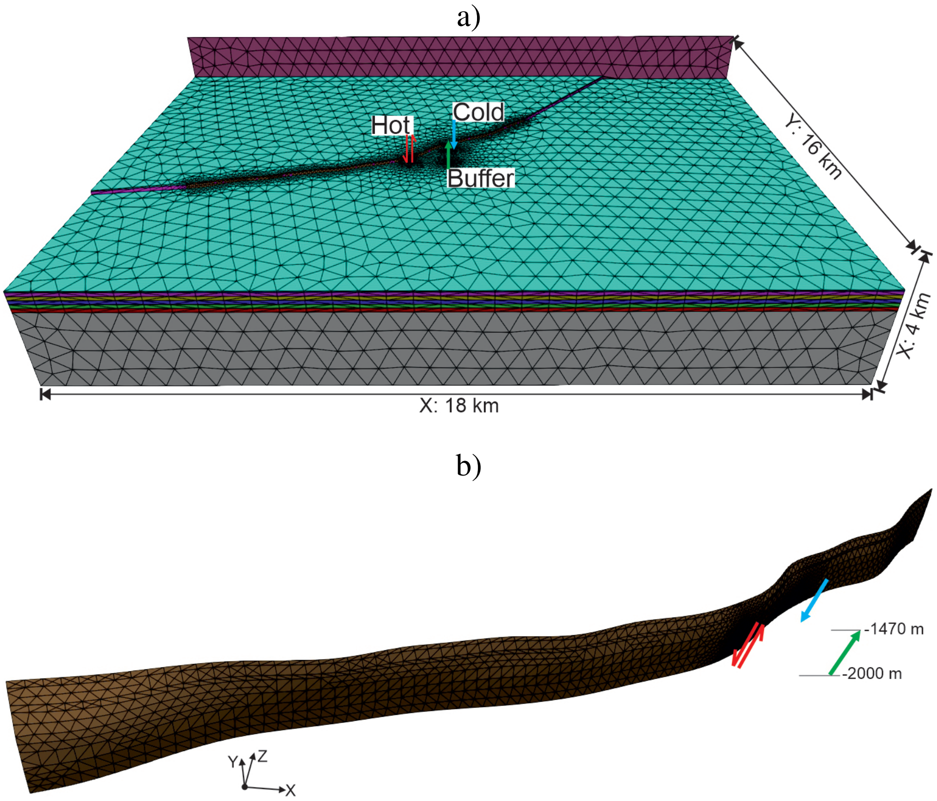

Fig. 6.1 a) The 3-D mesh generated for the simulations in the Molasse basin. The cap rock is not shown to have a better view of the designed triplet. The location of the hot well is highlighted by red arrows in the middle of the model. The green downward arrow shows the location of the buffer well. The blue upward arrow also denotes the location of the cold well. b) A closer view into the implemented normal fault with changing dipping angle and orientation.

$$P(z) = (Z - WT) \times \rho \times g, \tag{6.16}$$

where P is the pressure, WT represents the depth of the water table (c. 10 m for the Molasse basin), ρ is the density of the fluid and g denotes the gravitational acceleration.

For the other variables of the model, the displacement values in three directions, the IC is set to be 0.

The same depth-dependent temperature and pressure functions are applied as BC for both variables. The pressure BC is set on the four side faces of the model while the temperature BC is applied on all the six faces. With this approach, MOOSE calculates the depth-dependent function in the surfaces of the model. The displacement and stress BCs are set based on the stress regime in the Molasse basin which is fully elaborated by Ziegler and Heidbach (2020)

[255]. A strike-slip regime with 0° N [255, 256] as the direction of the maximum horizontal stress (S_H) is chosen in this study. Therefore, the maximum amount of stress acts on the y direction while in the z and x directions, the medium (S_v) and minimum (S_h) stress are set, respectively. For a rock density of 2300 kg m^{-3}, the S_v rate is set to 22600 Pa m^{-1}. The (S_h) value in this study comes from the formation integrity leak-off tests and is determined to be 19600 Pa m^{-1}. In the case of (S_H), Zoback (2010) [253] proposed the calculation of it as a function of the pore pressure (p_p), S_h and the friction coefficient of fault (μ)

$$\frac{(S_H - p_p)}{(S_h - p_p)} = [(\mu^2 + 1)^{1/2} + \mu]^2. \tag{6.17}$$

In this study, μ is assumed to be 0.6 which is a typical value for the faults in the underground that lack any measurement [253]. A 34600 Pa m^{-1} rate has been calculated for the S_H in our study.

The displacement BC is set in a way to fix one face and put the related amount of the stress in the opposite face. For example, in the y direction, the back side of the model is forced to have 0 displacement (in the y direction) while on the front side, the stress value is acting on the surface. The stress values in each direction are calculated based on the depth-dependent functions:

$$\sigma(z) = Z \times \rho_g \times g \times coef \tag{6.18}$$

where ρ_g is the density of the rock and $coef$ denotes the coefficient that defines the stress regime. In our case, the maximum coefficient value is set to be 1.54 (as the ratio of S_H and S_h) and acts on the y direction. For the z and x directions, it is set to be 1.0 and 0.77, respectively.

A normal fault surface with a 100-m offset is embedded in the middle of the model. The dipping angle and strike of the fault surface changes. The shape of the fault surface can be subjected to uncertainty. However, the change in the dipping angle and strike of the fault can already prove the impact of uncertainty. In previous chapters of this study, the faults were considered as simple plane surfaces with one fixed dip, strike, and depth. Therefore, it was necessary to change their dipping angle or depth to quantify the impact of uncertainty. To generate the complex fault surface in the mesh (Fig. 6.1-b), the new surface fitting functionality of Gmsh [38, 39] is used in this study. The fault geometry and offset are calculated in the geological model. Then, a grid of points representing the fault surface has been imported into Gmsh to reconstruct and mesh it. The unstructured mesh of the model is composed of tetrahedral elements. In the mesh, the element size gradually increases from 2 m around the wells to 600 m at the boundaries of the model. The fault surface is also refined due to its importance in the THM simulations.

A "reference case" is developed to investigate the general coupled THM behavior of the geothermal doublet and storage triplet. The parameterization scheme which follows the studies of Egert et al. (2022) [257], Bohnsack et al. (2021) [258], and Potten (2020) [259] is presented in Table 6.1. After making a comparison between the performance of the two applications, a more detailed sensitivity analysis has been performed on the more successful, i.e. less risky one.

Table 6.1 Parameters selected to run THM simulations for the reference case.

Parameters		**Values**
Simulation time [year]		10
Friction coefficient (μ) [-]		0.6
Orthogonal distance to the fault [m]	Hot well	118
	Cold well	195
Injection temperature [°C]	Hot well	140
	Cold well	50
Flow rate [l s^{-1}]	Injection	150
	Production	150
Permeability [m^2]	Reservoir	3.9×10^{-12}
	Cap rock and basement	9.8×10^{-17}
Porosity [-]	Reservoir	0.023
	Cap rock and basement	0.01
Specific heat capacity [JKg^{-1} K^{-1}]	Cap rock	2800
	Reservoir	2700
	Basement	2500
Thermal conductivity [Wm^{-1} K^{-1}]	Cap rock	2.1
	Reservoir	2.5
	Basement	3.2
Young's modulus [GPa]	Cap rock	25
	Reservoir	53
	Basement	63
Poisson's ratio [-]	Cap rock	0.36
	Reservoir	0.29
	Basement	0.29

6.3 Results and discussion

6.3.1 Storage triplet versus geothermal doublet

As Fig. 6.2-a shows, a thermal plume with an approximate radius of 45 m is formed around the hot well after 10 years of hot fluid injection and production. In the case of the cold well, the radius of the cold plume is far bigger (c. 110 m) due to the lack of any production from this well. Therefore, the injected fluid can move farther into the reservoir. Compared to the storage triplet, the cold front in the geothermal doublet propagates more strongly due to the continuous cold water injection (Fig. 6.2-b). The plotted trace in Fig. 6.2-c can give a better view of the propagation of the temperature signal. As this plot shows, in the storage triplet the performance of the hot well is improving because the reservoir is heating up. While the injection temperature at the hot well is 140 °C, at the end of the first production cycle (1.0 a), the maximum withdrawn temperature is 118 °C while after 10 years it increased to 132 °C. The geothermal doublet has not shown any breakthrough due to the high thickness of the reservoir and the distance between the injection and production well. Therefore, a continuous reservoir ambient temperature production is guaranteed over a 10-year snapshot of the reservoir lifetime. It should be noted that the orthogonal distance of the fault and the cold well in the reference case is 195 m that is still significantly higher than the hot/cold plume radius.

To evaluate the hydraulic impact of the injection and production operations on the highly transmissive Malm reservoir, the pressure change after ten years of operation for the storage triplet is illustrated in Fig. 6.2-d. As the figure shows, the pressure build up in the Malm reservoir is negligible. For both applications, the pressure changes are very small and close to each other, hence the three-dimensional distribution of the storage triplet is presented. A negligible change of ±0.02 MPa in the ambient pressure values of nearly 19.5 MPa indicates the insignificance of the hydraulics of the model within the reservoir. As shown in Fig. 6.2-d, the pressure perturbation caused by the cold well is slightly higher than the hot well. In the hot well the pressure build-up at the end of the injection cycle well decreases dramatically due to the following production cycle while in the cold well, the pressure build-up can rest at the end of each 6-month injection phase.

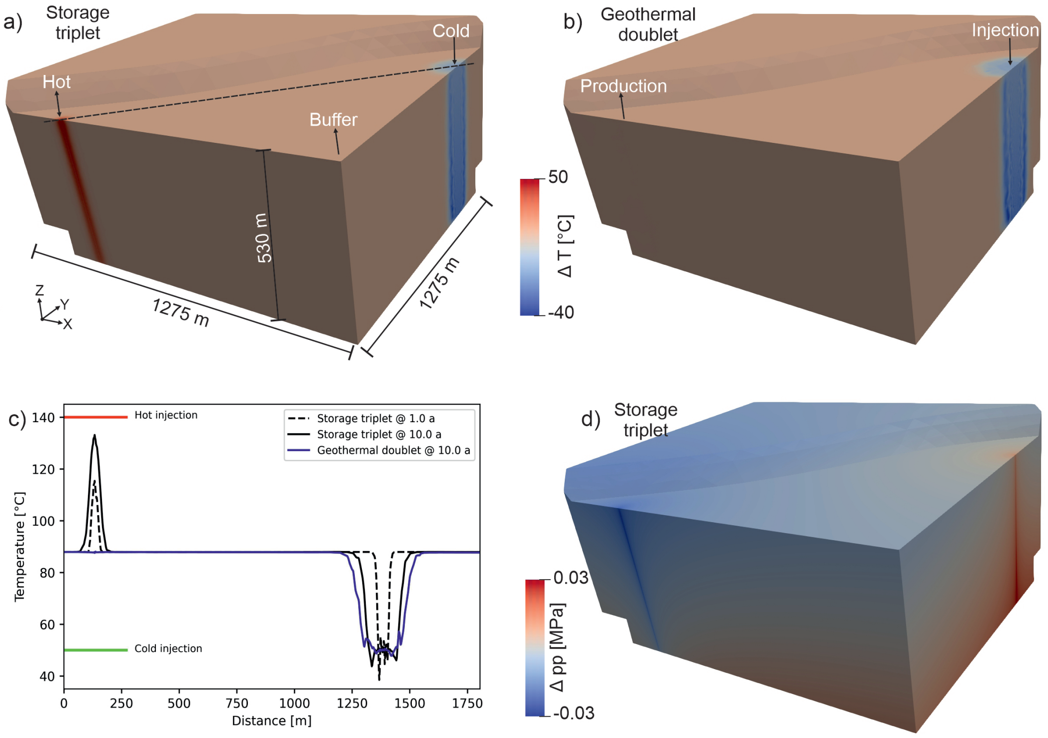

Fig. 6.2 Temperature changes after 10 years of operation in the storage triplet (a) and geothermal doublet (b). The locations of the hot, cold, and buffer wells are highlighted via arrows and their names in subplot a. Dimension and orientation of the next three-dimensional visualizations remain the same as what is presented in subplot a. The dashed line in subplot a is used to plot the temperature values versus distance in subplot c. The location and name of the production and injection well are also highlighted in subplot b. Subplots a and b share the same color bar. d) Pressure changes after 10 years of the operation in the storage triplet. In all the plots the coloring scale increases linearly from the minimum value to the maximum.

The mechanical behaviour of the two applications varies significantly mainly caused by the cold front propagation. The effective stress change for the direction of the maximum horizontal stress is visualized as Fig. 6.3-a and b. The changes in the stress are caused by the 10-year operation of the storage triplet and geothermal doublet. In the storage triplet, the affected area around the cold well is again bigger than the hot well. A maximum 20 MPa decrease in the stress value is caused due to the injection of the cold fluid. By injecting cold fluid, the porous matrix contracts (thermoelasticity), and the pore pressure (poroelasticity) also increases which both result in the decrease of the effective stress around the cold well. A maximum of 11 MPa increase of the effective stress in the y direction is observed around the hot well after the last production cycle. Around the hot borehole, the matrix contracts which dominates the overall behaviour of the reservoir. For the geothermal doublet, the affected area around the injection well is bigger than the cold well of the storage triplet and the absolute value of the stress changes also increases to 22 MPa. The lack of any changes around the production well of the geothermal doublet also indicates the dominance of the thermoelasticity over the poroelasticity in the Malm reservoir.

Displacement values are calculated for both applications as an effect of the stress changes. Similar to the stress, the displacement is also calculated in three directions. Fig 6.3-c and d contain three-dimensional visualizations that illustrate the displacement value in the y direction. The production well of the geothermal doublet again shows stronger perturbations due to the continuous operation (±22 mm versus ±16 mm). Some minor changes can also be detected around the hot well of the storage triplet which due to the scale of the distribution are hardly observable.

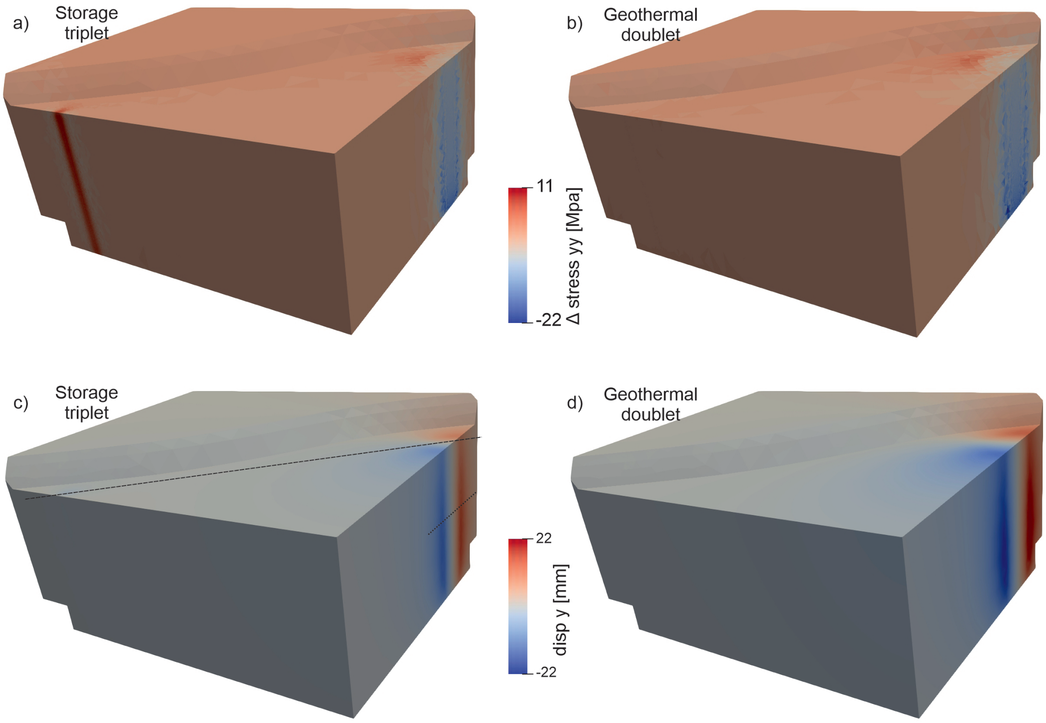

Fig. 6.3 Changes of stress in y direction after 10 years of the operation in the storage triplet (a) and geothermal doublet (b). The operation-induced displacement in the storage triplet (c) and geothermal doublet (d). The dashed and dotted traces in subplot c are later used for line plots. The dimension and orientation of the visualization see Fig. 6.2-a.

The slip tendency variations also have been another factor applied here to evaluate the mechanical behaviour of two applications and assess the possibility of fault reactivation. The first notable point is that the fault surface with its complex geometry (with changing strike and dip) shows a wide range of slip tendency values from 0 to 0.76. Fig. 6.4-a shows the slip tendency distribution on the fault plane at the initial condition. The maximum values are calculated to be in the areas where the fault aligns with the maximum horizontal stress (y), i.e. the north direction. A closer view into the slip tendency change after 10 years of operation in the storage triplet and geothermal doublet is presented in Fig. 6.4-b and c. Even though the distance of the fault to the cold well is 195 m, the variation in the initial value of slip tendency in the geothermal doublet is calculated to be twice the storage triplet. For both cases, the changes were nearly one order of magnitude lower than the maximum slip tendency value, i.e. 0.76. Again, the cold well is affecting the fault surface in a much stronger way than the hot well.

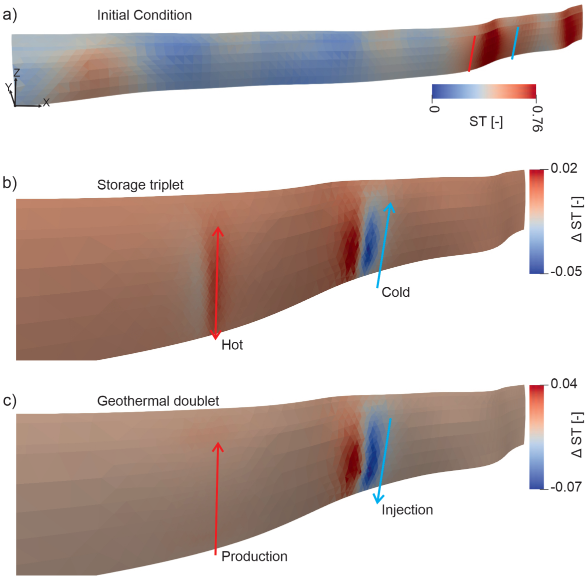

Fig. 6.4 a) Slip tendency (ST) distribution on the fault surface at the initial condition. ST changes in a zoomed part of the fault after 10 years of operation in the storage triplet (b) and geothermal doublet (c). Wells are annotated accordingly.

Well monitoring

To evaluate the magnitude of the stress perturbation at the hot well, Fig. 6.5-a shows the profile of the three principal stress components at the hot well. The first note is that all three components follow the increasing trend with depth and the stress value in the y direction is the maximum horizontal component. After half a year of hot fluid injection in this well, the effective stress magnitude increased in spit of the pore pressure increase. The reason for this contradiction is related to the expansion of the host rock due to the temperature increase. As mentioned, the thermoelasticity dominates the mechanical response of the reservoir. After one year, the stress increase relaxes due to the withdrawal of the heat. At the end of the first hot fluid injection cycle, the reservoir experienced the maximum change in the stress. The effective stress variation with time and depths for the cold well is illustrated in Fig. 6.5-b.

In this well the (dominating) thermoelastic and (negligible) poroelastic stress variations are in the same direction. While the injection phase increases the pore pressure which in turn decreases the effective stress, the contraction caused by the cold fluid injection also results in the decrease of the effective stress. At the buffer well no changes in the initial stress values are observed due to the lack of any temperature change, i.e. the predominant thermoelasticity does not act there.

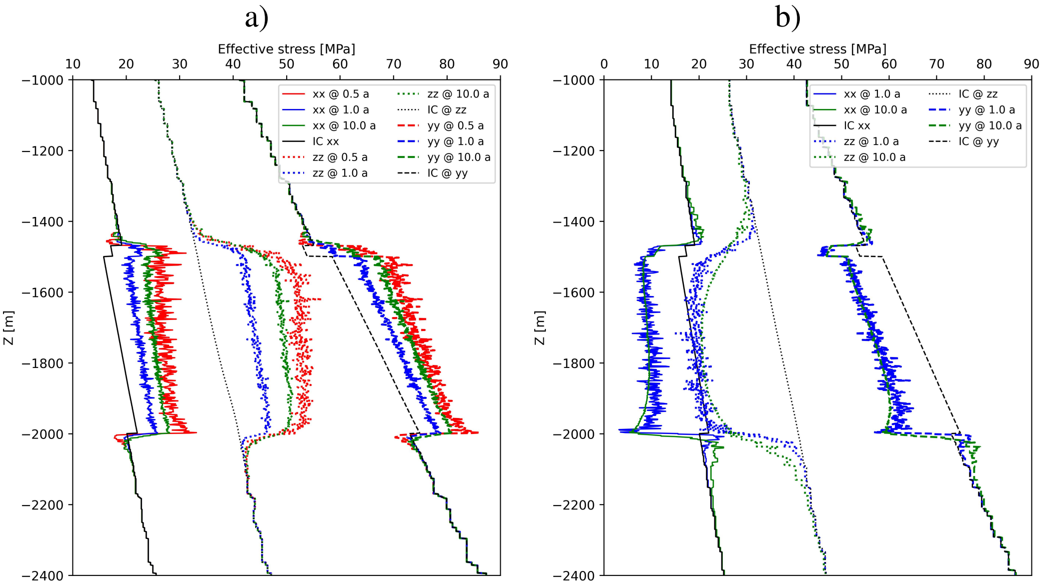

Fig. 6.5 a) Temporal behaviour of the three stress components around the hot (a) and cold (b) wells. The Malm reservoir starts from the depth of -1470 m to -2000 m.

This stress variation can directly result in displacement within the model. Fig. 6.6 shows the amount of the uplift and subsidence, i.e. displacement in the z direction, at the end of the 10 years of the storage operation. A dashed trace in Fig. 6.3-c shows the location of the data. The displacement data in the z direction show the reservoir expansion, uplift, around the hot well whereas subsidence is observed in the adjacency of the cold well. Both the absolute values and also propagation of the z displacement around the cold well are higher than the hot well. In the bidirectional hot well, as a bidirectional one, the change in the operation styles results in stress relaxations in each half-year cycle. For example, the uplift value reduces from 8.5 mm to 1 mm during the first production cycle from the hot well.

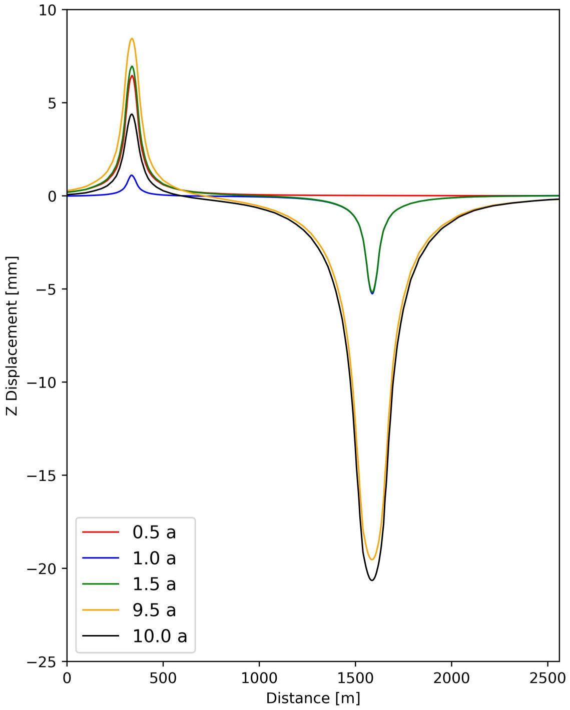

Fig. 6.6 Z displacement evolution over different periods of the triplet operation. The plotted displacement data are extracted from the traces visualized as a dashed line in Fig. 6.3-c. The location of the hot and cold wells can be distinguished via the maximum uplift and subsidence.

Fig. 6.7 shows the temporal evolution of the y displacement around the cold well. The first line plots the y displacement values after 3 hours (0.125 days) of injecting cold fluid in the well. Both sides of the well are stretching alongside of the y direction up to 5 mm. The whole stress that creates such movement in the 30 m distance from the well is generated due to the cold water injection. The 5 mm displacement may seem insignificant but the cyclic behaviour of the system can result in accumulation of the displacement in the next steps. As 6.3-c shows that accumulated y displacement at the end of the operation is c. 16 mm. Not only in the y direction, the well path is under stress in the z and x directions which results in more movement.

Stress changes and induced deformation necessitate a real-time monitoring plan in the adjacency of the wells. For example, distributed acoustic sensing (DAS) is a well-known method for measuring the rate of displacement [260]. In the cold well the reduction in the three stress components can result in fracturing as proven by the movement of the Mohr circle toward the failure envelope [253]. Such fracturing methods are typically applied to increase the contact area of the injected fluid and the hot host rock in geothermal applications like enhanced geothermal systems [126]. However, a storage concept in a faulted reservoir

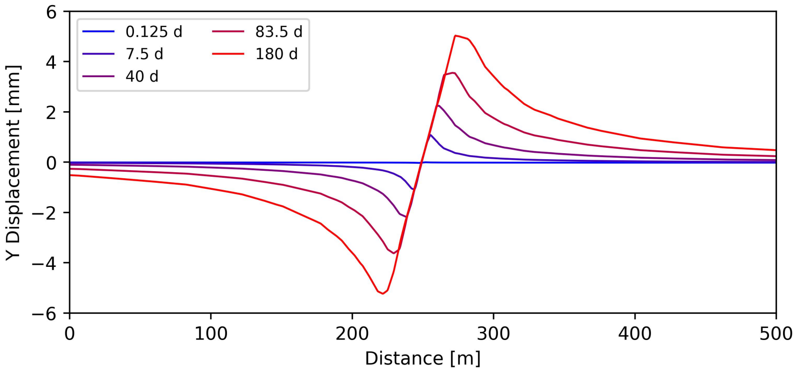

Fig. 6.7 Temporal evolution of the y displacement during the first injection cycle around the cold well. The time, mentioned in days, covers the first 6-month injection cycle of the cold water in the cold well.

close to municipalities should avoid fracturing risks. Either the decreased injection pressure or increased injectivity of the cold well indicates the generation of the fractures. Such local matrix failure can result in acoustic emissions that can be recorded by a seismic monitoring network. Decades of practical experience in fracturing for more efficient hydrocarbon extraction have provided valuable insights into the deployment of such networks, e.g. Van Der Baan et al. (2013) [261].

As the thermoelastic stress is the dominant factor in deforming the reservoir, decreasing the temperature difference between the ambient and injection fluid can also mitigate the fracturing risk at the vicinity of the cold well. Historically also all the major seismic events in the Munich area have happened in the vicinity of the injection wells [262, 82]. Therefore, the vicinity of the cold well is the best location for the monitoring well to observe the cold front propagation and possible seismic hazards. To fully understand the potential risks, the simulations should account for the possibility of the fracturing (when, where, and to which extent) and plastic deformation of the matrix. Modelling the wells as three-dimensional volumes, rather than 0-D points, and coupling them into the reservoir can allow for a better well stability assessment. In the hot well, the long-term exposure to high temperatures decreases cement strength, forms microcracks caused by cement cracking, and can even affect the casing. Prestressing the casing before implementing it in the hot well can counteract the thermal expansion.

6.3.2 Well location sensitivity

Slip tendency has been chosen in this study as a metric for possible fault reactivation. In the previous section the sensitivity of this parameter on the operation style, fault dipping angle, and orientation is discussed. Here, some different well locations for the storage concept are tried to evaluate the risk of fault reactivation. The reference case with a 195 m orthogonal distance between the cold well and the fault plane promises insignificant changes in the initial slip tendency of the fault. However, different distances are included in this study. While the parametrization scheme of the different cases remains the same (Table 6.1), the well location changes.

As already declared, the fault surface is highly complex that results in different distances between the fault surface and a direct line representing the well. Fig. 6.8-a shows two distances are highlighted to emphasize the complexity of the fault surface. While $d1$ denotes the orthogonal distance from the top of the cold well to the fault surface, $d1'$ connects the middle of the cold well to the fault surface. For the case with the closest distance represented by Fig. 6.8-a, $d1$, and $d1'$ are nearly 100 m and 45 m, respectively. For the second case, the cold well location is moved farther to make a 110 m distance ($d2$) to the fault surface. The case with a 110-m distance has made bigger changes compared to the first case with 100 m due to the smaller affected area of the fault surface. It should be noted that the maximum initial slip tendency in the adjacency of the cold well is around 0.76 which needs a +0.25 increase to surpass the threshold and activate the fault surface. The time of passing the threshold is also another difference for the first two cases. In the case with a 100-m distance, it happens in the second injection cycle (c. 1.9 a) while in the other case, it is later (fourth injection cycle, i.e. 4.6 a). The third case with a distance of 130 m is visualized as Fig. 6.8-c. For this well placement, the maximum value of the slip tendency decreased and also the threshold has been passed in the 8th injection cycle. Adding a slight deviation toward the east will keep the distance of 130 m between that fault surface and the cold well which in turn guarantees the operation without a fault reactivation. Two safe cases are presented in Fig. 6.8-d and e with distances of 145 m and 155 m between the cold well and the fault surface. As a direct observation, the closer the cold well to the fault plane is, the higher and faster perturbation of slip tendency is observed. However, the fault surface can be reactivated only if the cold plume approaches it that can be avoided either by locating the cold well far from it or adding deviation to the well trajectory.

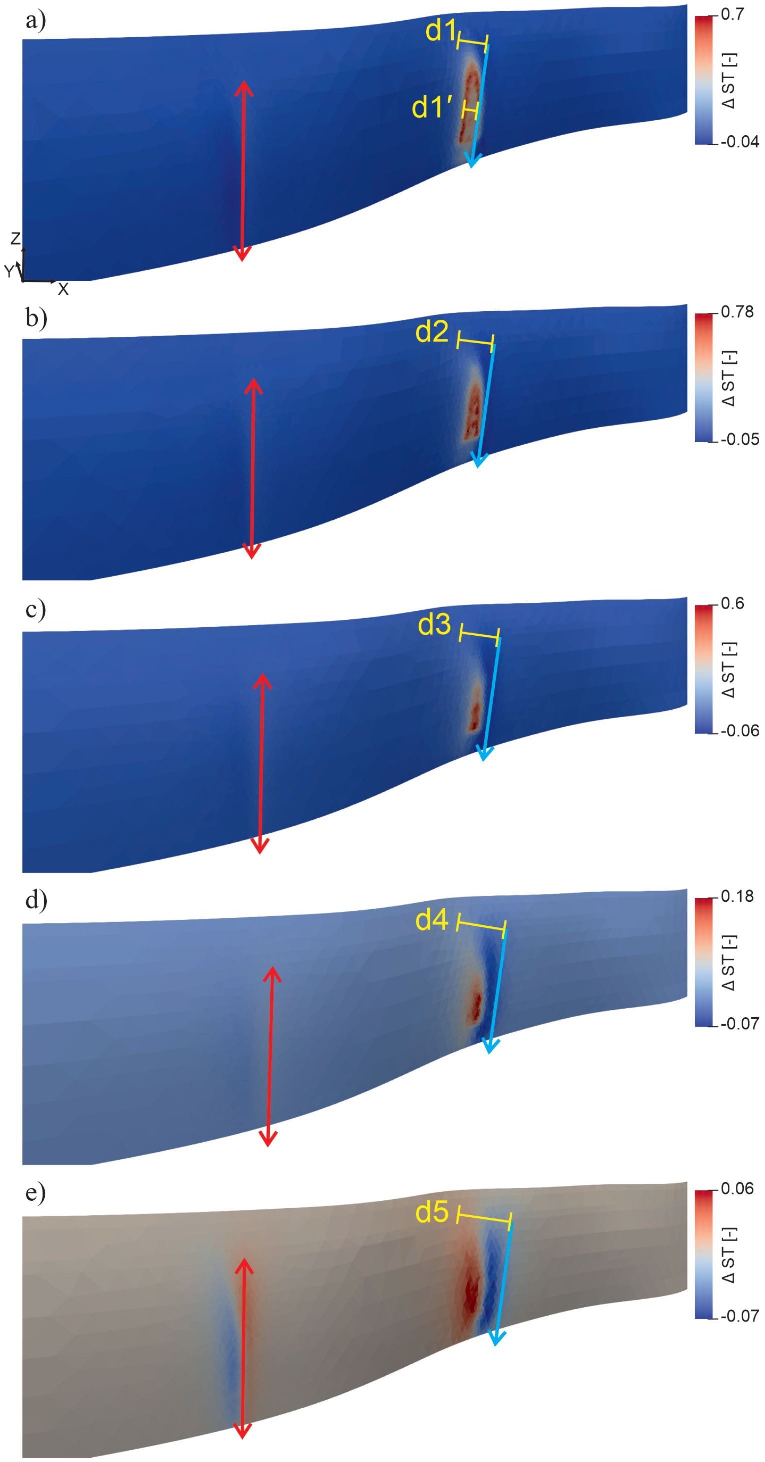

Fig. 6.8 Slip tendency perturbation of the fault surface after 10 years of operation in the bidirectional hot (red arrow) and unidirectional cold well (blue arrow). a) The distance between the top of the well to the fault surface ($d1$) is equal to 100 m while $d1'$ is 45 m. b) The cold well is 110 m away from the fault surface, i.e. $d2$ = 110 m, c) The distance $d3$ is 130 m. d) The distance $d4$ is 145 m. e) The distance $d5$ is 155 m.

6.4 Conclusion and outlook

In the scope of the presented site-specific work, a comparison has been made between the THM simulation results of a geothermal doublet and an HT-ATES triplet. Both concepts guaranteed very negligible pressure build-ups within the Malm reservoir. The storage triplet is confirmed to have an improving thermal performance and the geothermal doublet also has not shown any sign of a thermal breakthrough. In both models, it is observed that the thermoelasticity is the dominant factor in the dynamic responses of the reservoir.

Based on the numerical simulations, it is revealed that even in a 10-year time frame, the storage concept yields lower risks. In fact, the cyclic behaviour of the storage allows for the relaxation of the induced operational stresses. At the top of the hot well in the storage triplet, the uplift value decreased from 8.5 mm to 1 mm from the end of the first injection cycle (0.5 a) to the end of the first production cycle (1.0 a) which implicates the temporal relaxation, i.e. less perturbation.

The highly complex embedded fault surface in the model shows a wide variety of normalized slip tendency values ranging from 0 to 0.76 at the initial condition. Even with a distance of 195 m between the cold well and the fault surface, the continuous cold water injection in the geothermal doublet is perturbing the slip tendency two times higher than the cold well of the storage triplet.

In the storage triplet also the distance between the fault and the cold well plays a major role in the fault reactivation potential. Distances higher than 145 m promise a safe operation in terms of the fault reactivation.

As the next step, the fault surface can be subjected to uncertainty. This study revealed the significance of the fault geometry and moved the location of the wells to assess the importance of the distance between the fault surface and the wells. However, rather than a deterministic model, a range of possible geological models can be generated to have a wider range of expectations. This study also considers the Malm reservoir as a homogeneous and isotropic medium which in reality can not be the case.

Acknowledgment

Ali Dashti is receiving the financial support from The German Academic Exchange Service (Deutscher Akademischer Austauschdienst: DAAD) to do his PhD in Germany as the Research Grants-Doctoral programmes in Germany 2019/20. This organization is appreciated for giving the opportunity to researchers. This study is part of the subtopic “Geoenergy” in the program “MTET—Materials and Technologies for the Energy Transition” of the Helmholtz Association.

Competing interests

The authors declare that they have no conflict of interest.

Chapter 7

Conclusions and outlook

7.1 Conclusions

Geothermal energy as a renewable source utilizes the Earth's natural heat to be a game-changer in our fight against climate change. By significantly reducing our reliance on fossil fuels, it paves the way for a more sustainable energy future. Central to the underground geothermal systems are structural models and their inherent uncertainties. The efficiency and reliability of geothermal systems tightly hinge on the accuracy of the numerical simulations performed on these structural models. This thesis amalgamates findings from four diverse studies, each contributing a unique perspective on the intricate relationship between geological structures, subsurface processes, and the application of ML methods for uncertainty quantification. The studies revolve around the impact of the structural uncertainties on the tracer flow for an EGS setting and the thermo-hydraulic performance of an HT-ATES application. Despite the importance of the structural uncertainties, their incorporation into studies have been always postponed due to the complexity of the meshing. A new Python library is developed within this study to facilitate the generation of meshes for complex geological models. Several automated Python scripts are also developed allowing for generating as much as required meshes using Gmsh.

Developement of GeoMeshPy: The initial study introduces GeoMeshPy as a powerful tool that bridges GemPy and Gmsh, allowing for automated mesh generation in the context of tracer injection in fractured systems. Vertices of modelling tools, e.g. Gmepy are processed in GeoMeshPy and later fed into the Python API of the Gmsh. GeoMeshPy can regularize the vertices of as many exiting layers in the model. To make vertices applicable for later surface generation in Gmsh, it is necessary to remove redundant points close to the existing faults. This way, GeoMeshPy splits the continuous output of Gempy. Then, it is possible to fit a surface to each patch separately. Manual mesh generation for different geological realizations is a highly time-demanding and error-prone process. This tool can transfer the structural uncertainty of GemPy into a numerical mesh. In parallel, Gmsh, has been updated with the new functionality to automatically emulate structures created in geomodelling tools.

Tracer flow: By meticulously varying the orientation of connecting faults, the study demonstrates the significant impact of structural uncertainty on BTCs. The dependence of the flow regime on the fracture network is an already documented fact but the first study of this thesis provides a robust framework for quantifying this relation. Structural models with several uncertain faults can be generated and meshed automatically. The length of production wells and subsequent flow circulation, influenced by geological uncertainties, emerges as a critical factor affecting tracer peak concentrations and arrival times. This study lays the groundwork for acknowledging and integrating structural uncertainties in numerical simulations as small variations to the dipping angle of one fault (among the existing three ones) resulted in differences of up to 26 and 30 percent for peak arrival time and magnitude of the tracer signal.

Heat storage: Building upon GeoMeshPy's insights, the second study explores uncertainty quantification in HT-ATES through an innovative automated workflow in Python and Gmsh. The focus shifts to decision-making, with geological uncertainty serving as a key input in various phases, from exploration to development. A developed model for the Geneva basin confirmed the independence of the temperature from the geometry of the thick reservoir. The Malm layer with 100 m thickness can possibly be detected through 2D seismic slices. Therefore, surveys for finding the exact geometry of the top and bottom surfaces with higher accuracies seem unnecessary for such cases. This study highlights the necessity of running computationally affordable simulations prior to any exploration campaign. The porous sand layers existing within Oligocene Meletta beds beneath the KIT campus are promising storage space. For DeepStor adding one more level of complexity (a sealing sub-seismic fault) to interpreted data expresses the performance risks such as possible significant heat losses and/or pressure increase. With the proposed material properties, the presented evaluation on DeepStor proved that only in cases where a sealing fault is closer than the radius of the heat plume, the thermal performance of the system can be affected. The effect on the thermal recovery of the borehole is hardly observable but the overall dimension of the heat plume can change due to such faults in the vicinity. Numerically calculated pressure values at the borehole location can decipher the faults even in 118 m distances assuming the fixed and certain petrophysical properties. The relation between pressure changes and the location of the introduced sealing fault can be used to derive a case-specific forecasting scheme for detecting possible locations of the barriers in the DeepStor model.

ML-Driven Structural Uncertainty Assessment: Transitioning seamlessly, the third study pioneers the application of ML techniques to assess the impact of structural uncertainty on BTCs in two EGS settings. By employing Decision Tree Regression (DTR), Random Forest Regression (RFR), and a chain of Gradient Boosting Regression (GBR) models, this research showcases the adaptability of ML in predicting BTCs for diverse structural scenarios. The study goes beyond isolated structural variations, hinting at the potential of ML-driven approaches in incorporating variations in petrophysical properties and leveraging additional data sources. ML emerges not only as a predictor of subsurface processes but also as a tool with transformative potential in replacing conventional analytical solutions for parameter identification. The trained ML models are further applied to generate BTCs for 2’000 different geological realizations. The chain model was more accurate than the RFR and DTR models. The RFR method produced 2’000 BTCs that closely follow the trend observed in the training set indicating the underfitting issue, whereas DTR can only replicate the BTCs from the training set. It has been possible to generate thousands of new BTCs as the ML methods were significantly faster than the numerical solver, with up to six orders of magnitude reduction in computational time. The chain model captures both the general trend and small-scale patterns of the data.

Fault reactivation analysis: Last but not least, this chapter of the thesis addresses the possibility of induced seismicity due to the geothermal activities close to Munich. Thanks to the new development in Gmsh, a highly complex fault surface has been embedded into the Malm aquifer. The coupled THM simulations revealed that even the initial distribution of the slip tendency on the fault surface varies dramatically due to the change in the dipping angle and orientation of the highly complex fault surface. This study highlighted the importance of fault geometry by evaluating the influence of the distance between the fault surface and the wells. It also demonstrated that small zones of the fault exhibit elevated slip tendencies in the natural state. Therefore, simplification of the fault surfaces can debase the results of the THM studies. The numerical simulations demonstrated that the seasonal operation of the HT-ATES triplet results in a significantly lower environmental impact compared to the conventional geothermal doublet. This reduction is primarily due to the injection well being operated only during the winter, thereby experiencing only half the thermal stress induced by cooling. Both concepts lead to very negligible pressure build-ups within the Malm reservoir. In both applications, thermoelasticity was identified as the dominant factor influencing the dynamic THM responses of the reservoir. In the triplet model, well location significantly affects the possible reactivation risk of the fault surface.

7.2 Outlook

Collectively, these studies underscore the necessity of adopting a holistic approach to understanding and quantifying structural uncertainties in subsurface systems. I developed GeoMeshPy to facilitate the incorporation of structural uncertainties in the numerical simulations but at the moment it can only cover some specific types of models generated in GemPy. It also only generates readable inputs for Gmsh as the mesh generator while other meshing codes are also available. At the first step, a more robust connection between GemPy and Gmsh can provide us with a wider range of possible uncertainties. The best solution is that the code will be developed in a way to process and adjust results of other modelling tools, e.g. Petrel, Leapfrog, etc., as readable inputs for other meshing tools like Cubit and so on. This way, not only the uncertainty of a fracture but also every feature of the more complex structural model (like the Soultz-sous-Forêts EGS reservoir) can be perturbed and included in the numerical simulations. In the case of achieving this target, a wider range of users can benefit from it.

The source of the uncertainty in most examples of this study is supposed to be generic and random. However, the developed methodology is able to handle any kind of uncertainty for a more complex and realistic geological model. It only requires real cases and the exact knowledge on how to introduce underlying uncertainties coming from any possible source, e.g. 3-D seismic data, borehole measurements, etc.

Data-driven ML methods have been deployed in this study as a potential substitute for time-consuming mesh generation and numerical simulation. However, addressing the errors is crucial for future work aimed at improving the interpretability of the ML methods. First, the data-driven methods require big data sets as the input. Second, they may contradict the physics. Promising physics-based ML methods using order reduction techniques e.g., non-intrusive reduced basis, can be developed to build the solution based on basis functions that preserve the structure of the physics and are less sensitive to the size and distribution of the input data.

References

[1] H. H. Cho, V. Strezov, and T. J. Evans, “A review on global warming potential, challenges and opportunities of renewable hydrogen production technologies,” *Sustainable Materials and Technologies*, vol. 35, p. e00567, 2023.

[2] A. Dashti and M. Gholami Korzani, “Study of geothermal energy potential as a green source of energy with a look at energy consumption in iran,” *Geothermal Energy*, vol. 9, no. 1, p. 28, 2021.

[3] C. McGlade and P. Ekins, “The geographical distribution of fossil fuels unused when limiting global warming to 2 c,” *Nature*, vol. 517, no. 7533, pp. 187–190, 2015.

[4] M. Siddik, M. Islam, A. Zaman, and M. Hasan, “Current status and correlation of fossil fuels consumption and greenhouse gas emissions,” *Int. J. Energy Environ. Econ*, vol. 28, pp. 103–119, 2021.

[5] Z. Liu, T. Sun, Y. Yu, P. Ke, Z. Deng, C. Lu, D. Huo, and X. Ding, “Real-time carbon emission accounting technology toward carbon neutrality,” *Engineering*, 2022.

[6] A. Akaev and O. Davydova, “Climate and energy: Energy transition scenarios and global temperature changes based on current technologies and trends,” in *Reconsidering the Limits to Growth: A Report to the Russian Association of the Club of Rome*, pp. 53–70, Springer, 2023.

[7] “Iea data and statistics.” Accessed on 2023-10-11.

[8] “Ipcc special report on global warming of 1.5°c.” Accessed on 2023-10-11.

[9] S. Bouckaert, A. F. Pales, C. McGlade, U. Remme, B. Wanner, L. Varro, D. D’Ambrosio, and T. Spencer, “Net zero by 2050: A roadmap for the global energy sector,” 2021.

[10] NASA, “Nasa announces summer 2023 hottest on record,” 2023. Accessed on 2023-10-11.

[11] A. Burke, “An architecture for a net zero world: Global climate governance beyond the epoch of failure,” *Global Policy*, vol. 13, pp. 24–37, 2022.

[12] D. I. Armstrong McKay, A. Staal, J. F. Abrams, R. Winkelmann, B. Sakschewski, S. Loriani, I. Fetzer, S. E. Cornell, J. Rockström, and T. M. Lenton, “Exceeding 1.5 c global warming could trigger multiple climate tipping points,” *Science*, vol. 377, no. 6611, p. eabn7950, 2022.

[13] H. Lee, K. Calvin, D. Dasgupta, G. Krinner, A. Mukherji, P. Thorne, C. Trisos, J. Romero, P. Aldunce, K. Barrett, *et al.*, “Ar6 synthesis report: Climate change 2023,” *Summary for Policymakers*, 2023.

[14] M. Meinshausen, J. Lewis, C. McGlade, J. Gütschow, Z. Nicholls, R. Burdon, L. Cozzi, and B. Hackmann, “Realization of paris agreement pledges may limit warming just below 2 c,” *Nature*, vol. 604, no. 7905, pp. 304–309, 2022.

[15] S. Mandal, M. S. Islam, M. H. A. Biswas, and S. Akter, “A mathematical model applied to investigate the potential impact of global warming on marine ecosystems,” *Applied Mathematical Modelling*, vol. 101, pp. 19–37, 2022.

[16] P. A. Stott, N. Christidis, F. E. Otto, Y. Sun, J.-P. Vanderlinden, G. J. van Oldenborgh, R. Vautard, H. von Storch, P. Walton, P. Yiou, *et al.*, “Attribution of extreme weather and climate-related events,” *Wiley Interdisciplinary Reviews: Climate Change*, vol. 7, no. 1, pp. 23–41, 2016.

[17] E. Cruz, S. J. D’Alessio, and L. Stolzenberg, “The effect of maximum daily temperature on outdoor violence,” *Crime & Delinquency*, vol. 69, no. 6-7, pp. 1161–1182, 2023.

[18] X. Yu, N. S. Sandhu, Z. Yang, and M. Zheng, “Suitability of energy sources for automotive application–a review,” *Applied Energy*, vol. 271, p. 115169, 2020.

[19] A. Kalair, N. Abas, M. S. Saleem, A. R. Kalair, and N. Khan, “Role of energy storage systems in energy transition from fossil fuels to renewables,” *Energy Storage*, vol. 3, no. 1, p. e135, 2021.

[20] D.-A. Ciupageanu, L. Barelli, and G. Lazaroiu, “Real-time stochastic power management strategies in hybrid renewable energy systems: A review of key applications and perspectives,” *Electric Power Systems Research*, vol. 187, p. 106497, 2020.

[21] W. Huihui, M. Alharthi, I. Ozturk, A. Sharif, I. Hanif, and X. Dong, “A strategy for the promotion of renewable energy for cleaner production in g7 economies: By means of economic and institutional progress,” *Journal of Cleaner Production*, vol. 434, p. 140323, 2024.

[22] M. Z. Jacobson, M. A. Delucchi, Z. A. Bauer, S. C. Goodman, W. E. Chapman, M. A. Cameron, C. Bozonnat, L. Chobadi, H. A. Clonts, P. Enevoldsen, *et al.*, “100% clean and renewable wind, water, and sunlight all-sector energy roadmaps for 139 countries of the world,” *Joule*, vol. 1, no. 1, pp. 108–121, 2017.

[23] A. I. Osman, L. Chen, M. Yang, G. Msigwa, M. Farghali, S. Fawzy, D. W. Rooney, and P.-S. Yap, “Cost, environmental impact, and resilience of renewable energy under a changing climate: a review,” *Environmental Chemistry Letters*, vol. 21, no. 2, pp. 741–764, 2023.

[24] Y. Yang, Q. Cui, and H. Wang, “Assessing the socioeconomic and environmental impacts of china’s power sector changes in 2010–2020,” *Applied Energy*, vol. 364, p. 123159, 2024.

[25] M. Bortoluzzi, C. C. de Souza, and M. Furlan, “Bibliometric analysis of renewable energy types using key performance indicators and multicriteria decision models,” *Renewable and Sustainable Energy Reviews*, vol. 143, p. 110958, 2021.

[26] D. A. Konneh, H. O. R. Howlader, R. Shigenobu, T. Senjyu, S. Chakraborty, and N. Krishna, “A multi-criteria decision maker for grid-connected hybrid renewable energy systems selection using multi-objective particle swarm optimization,” *Sustainability*, vol. 11, no. 4, p. 1188, 2019.

[27] A. Rahman, O. Farrok, and M. M. Haque, “Environmental impact of renewable energy source based electrical power plants: Solar, wind, hydroelectric, biomass, geothermal, tidal, ocean, and osmotic,” *Renewable and Sustainable Energy Reviews*, vol. 161, p. 112279, 2022.

[28] M. H. Dickson and M. Fanelli, “Geothermal energy: utilization and technology,” 2013.

[29] J. W. Lund and T. L. Boyd, “Direct utilization of geothermal energy 2015 worldwide review,” *Geothermics*, vol. 60, pp. 66–93, 2016.

[30] T. B. Johansson, A. P. Patwardhan, N. Nakićenović, and L. Gomez-Echeverri, *Global energy assessment: toward a sustainable future*. Cambridge University Press, 2012.

[31] E. Efficiency, “Renewable energy technology characterizations,” *Topical Report TR-109496, Energy Efficiency and Renewable Energy (EERE), US Department of Energy< http://www. nrel. gov/docs/gen/fy98/24496. pdf*, 1997.

[32] C. R. Chamorro, M. E. Mondéjar, R. Ramos, J. J. Segovia, M. C. Martín, and M. A. Villamañán, “World geothermal power production status: Energy, environmental and economic study of high enthalpy technologies,” *Energy*, vol. 42, no. 1, pp. 10–18, 2012.

[33] J.-L. Mallet and A.-L. Tertois, “Solid earth modeling and geometric uncertainties,” in *SPE Annual Technical Conference and Exhibition?*, pp. SPE–134978, SPE, 2010.

[34] J. F. Wellmann and K. Regenauer-Lieb, “Uncertainties have a meaning: Information entropy as a quality measure for 3-d geological models,” *Tectonophysics*, vol. 526-529, pp. 207–216, 2012.

[35] J. C. Refsgaard, S. Christensen, T. O. Sonnenborg, D. Seifert, A. L. Højberg, and L. Troldborg, “Review of strategies for handling geological uncertainty in groundwater flow and transport modeling,” *Advances in Water Resources*, vol. 36, pp. 36–50, 2012.

[36] C. E. Bond, C. Philo, and Z. K. Shipton, “When there isn’t a right answer: Interpretation and reasoning, key skills for twenty-first century geoscience,” *International Journal of Science Education*, vol. 33, no. 5, pp. 629–652, 2011.

[37] M. de La Varga, A. Schaaf, and F. Wellmann, “Gempy 1.0: open-source stochastic geological modeling and inversion,” *Geoscientific Model Development*, vol. 12, no. 1, pp. 1–32, 2019.

[38] C. Geuzaine and J.-F. Remacle, “Gmsh: A 3-d finite element mesh generator with built-in pre- and post-processing facilities,” *International Journal for Numerical Methods in Engineering*, vol. 79, no. 11, pp. 1309–1331, 2009.

[39] A. Dashti, M. G. Korzani, C. Geuzaine, R. Egert, and T. Kohl, “Impact of structural uncertainty on tracer test design in faulted geothermal reservoirs,” *Geothermics*, vol. 107, p. 102607, 2023.

[40] A. Dashti, J. C. Grimmer, C. Geuzaine, F. Bauer, and T. Kohl, “Developing meshing workflows in gmsh v4. 11 for the geologic uncertainty assessment of high-temperature aquifer thermal energy storage,” *Geoscientific Model Development*, vol. 17, no. 8, pp. 3467–3485, 2024.

[41] A. Dashti, T. Stadelmann, and T. Kohl, “Machine learning for robust structural uncertainty quantification in fractured reservoirs,” *Geothermics*, vol. 120, p. 103012, 2024.

[42] K. A. Frings, C. von Hagke, F. Wellmann, E. Heim, and M. de la Varga, “Constraining the 3-d geometry of fold-thrust belts using section balancing vs. 3-d interpolative structural and probabilistic modeling,”

[43] C. Lajaunie, G. Courrioux, and L. Manuel, “Foliation fields and 3d cartography in geology: principles of a method based on potential interpolation,” *Mathematical geology*, vol. 29, pp. 571–584, 1997.

[44] L. Souche, G. Iskenova, F. Lepage, and D. Desmarest, “Construction of structurally and stratigraphically consistent structural models using the volume-based modelling technology: Applications to an australian dataset,” in *International petroleum technology conference*, pp. IPTC–18216, IPTC, 2014.

[45] D. Dhont, P. Luxey, and J. Chorowicz, “3-d modeling of geologic maps from surface data,” *AAPG bulletin*, vol. 89, no. 11, pp. 1465–1474, 2005.

[46] G. Caumon, P. Collon-Drouaillet, C. Le Carlier de Veslud, S. Viseur, and J. Sausse, “Surface-based 3d modeling of geological structures,” *Mathematical geosciences*, vol. 41, pp. 927–945, 2009.

[47] S. D. G. Campbell, J. E. Merritt, B. E. O Dochartaigh, M. Mansour, A. G. Hughes, F. M. Fordyce, D. C. Entwisle, A. A. Monaghan, and S. C. Loughlin, “3d geological models and their hydrogeological applications: supporting urban development: a case study in glasgow-clyde, uk,” *Zeitschrift der Deutschen Gesellschaft fur Geowissenschaften*, vol. 161, no. 2, pp. 251–262, 2010.

[48] M. W. Jessell, L. Ailleres, and E. A. De Kemp, “Towards an integrated inversion of geoscientific data: What price of geology?,” *Tectonophysics*, vol. 490, no. 3-4, pp. 294–306, 2010.

[49] P. Ringrose and M. Bentley, *Reservoir model design*. Springer, 2016.

[50] D. Harris, “The role of geology in reservoir simulation studies,” *Journal of Petroleum Technology*, vol. 27, no. 05, pp. 625–632, 1975.

[51] F. Wellmann and G. Caumon, “3-d structural geological models: Concepts, methods, and uncertainties,” vol. 59 of *Advances in Geophysics*, pp. 1–121, Elsevier, 2018.

[52] J. F. Wellmann, F. G. Horowitz, E. Schill, and K. Regenauer-Lieb, “Towards incorporating uncertainty of structural data in 3d geological inversion,” *Tectonophysics*, vol. 490, no. 3-4, pp. 141–151, 2010.

[53] M. Lindsay, L. Ailleres, M. Jessell, E. de Kemp, and P. Betts, “Integrating geological uncertainty into combined geological and potential field inversions,” in *Proceedings of the GeoMod 2010 Conference, Lisbon, Portugal*, pp. 27–29, 2010.

[54] J. F. Wellmann, “Information theory for correlation analysis and estimation of uncertainty reduction in maps and models,” *Entropy*, vol. 15, no. 4, pp. 1464–1485, 2013.

[55] P. Thore, A. Shtuka, M. Lecour, T. Ait-Ettajer, and R. Cognot, “Structural uncertainties: Determination, management, and applications,” *Geophysics*, vol. 67, no. 3, pp. 840–852, 2002.

[56] N. Cherpeau, G. Caumon, and B. Lévy, “Stochastic simulations of fault networks in 3d structural modeling,” *Comptes Rendus. Géoscience*, vol. 342, no. 9, pp. 687–694, 2010.

[57] C. E. Shannon, “A mathematical theory of communication,” *The Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948.

[58] L. Holden, P. Mostad, B. F. Nielsen, J. Gjerde, C. Townsend, and S. Ottesen, “Stochastic structural modeling,” *Mathematical geology*, vol. 35, pp. 899–914, 2003.

[59] M. Perrin, B. Zhu, J.-F. Rainaud, and S. Schneider, “Knowledge-driven applications for geological modeling,” *Journal of Petroleum Science and Engineering*, vol. 47, no. 1-2, pp. 89–104, 2005.

[60] N. S.-N. Lam, “Spatial interpolation methods: a review,” *The American Cartographer*, vol. 10, no. 2, pp. 129–150, 1983.

[61] T. Mueller, N. Pusuluri, K. Mathias, P. Cornelius, R. Barnhisel, and S. Shearer, “Map quality for ordinary kriging and inverse distance weighted interpolation,” *Soil Science Society of America Journal*, vol. 68, no. 6, pp. 2042–2047, 2004.

[62] M. A. Oliver and R. Webster, “Kriging: a method of interpolation for geographical information systems,” *International Journal of Geographical Information System*, vol. 4, no. 3, pp. 313–332, 1990.

[63] M. Wojciech, “Kriging method optimization for the process of dtm creation based on huge data sets obtained from mbess,” *Geosciences*, vol. 8, no. 12, p. 433, 2018.

[64] Y. Wang, O. V. Akeju, and T. Zhao, “Interpolation of spatially varying but sparsely measured geo-data: A comparative study,” *Engineering Geology*, vol. 231, pp. 200–217, 2017.

[65] G. Bohling, "Introduction to geostatistics and variogram analysis," *Kansas geological survey*, vol. 1, no. 10, pp. 1–20, 2005.

[66] M. A. Oliver, R. Webster, *et al.*, "Basic steps in geostatistics: the variogram and kriging," tech. rep., Springer, 2015.

[67] E. Gringarten and C. V. Deutsch, "Teacher's aide variogram interpretation and modeling," *Mathematical Geology*, vol. 33, pp. 507–534, 2001.

[68] D. J. Gorsich and M. G. Genton, "Variogram model selection via nonparametric derivative estimation," *Mathematical geology*, vol. 32, pp. 249–270, 2000.

[69] J.-P. Chiles and P. Delfiner, *Geostatistics: modeling spatial uncertainty*, vol. 713. John Wiley & Sons, 2012.

[70] M. J. Pyrcz and C. V. Deutsch, *Geostatistical reservoir modeling*. Oxford University Press, USA, 2014.

[71] T. Frank, A.-L. Tertois, and J.-L. Mallet, "3d-reconstruction of complex geological interfaces from irregularly distributed and noisy point data," *Computers & Geosciences*, vol. 33, no. 7, pp. 932–943, 2007.

[72] K. J. Liew, A. Ramli, and A. Abd Majid, "B-spline surface fitting on scattered points," *Applied Mathematics & Information Sciences*, vol. 10, no. 1, p. 273, 2016.

[73] F. Esmaeili, A. Amiri-Simkooei, V. Nafisi, and A. Alizadeh Naeini, "Application of b-spline method in surface fitting problem," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 42, pp. 343–348, 2019.

[74] F. S. Cohen, W. Ibrahim, and C. Pintavirooj, "Ordering and parameterizing scattered 3d data for b-spline surface approximation," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 6, pp. 642–648, 2000.

[75] S. Lee, G. Wolberg, and S. Y. Shin, "Scattered data interpolation with multilevel b-splines," *IEEE transactions on visualization and computer graphics*, vol. 3, no. 3, pp. 228–244, 1997.

[76] C. de Boor, T. Lyche, and L. L. Schumaker, "On calculating with b-splines ii. integration," *Numerische Methoden der Approximationstheorie/Numerical Methods of Approximation Theory: Vortragsauszüge der Tagung über numerische Methoden der Approximationstheorie vom 25. bis 31. Mai 1975 im Mathematischen Forschungsinstitut Oberwolfach (Schwarzwald)*, pp. 123–146, 1976.

[77] M. Gholami Korzani, S. Held, and T. Kohl, "Numerical based filtering concept for feasibility evaluation and reservoir performance enhancement of hydrothermal doublet systems," *Journal of Petroleum Science and Engineering*, vol. 190, p. 106803, 2020.

[78] A. D. Lindsay, D. R. Gaston, C. J. Permann, J. M. Miller, D. Andrš, A. E. Slaughter, F. Kong, J. Hansel, R. W. Carlsen, C. Icenhour, L. Harbour, G. L. Giudicelli, R. H. Stogner, P. German, J. Badger, S. Biswas, L. Chapuis, C. Green, J. Hales, T. Hu, W. Jiang, Y. S. Jung, C. Matthews, Y. Miao, A. Novak, J. W. Peterson, Z. M. Prince,

A. Rovinelli, S. Schunert, D. Schwen, B. W. Spencer, S. Veeraraghavan, A. Recuero, D. Yushu, Y. Wang, A. Wilkins, and C. Wong, “2.0 - moose: Enabling massively parallel multiphysics simulation,” *SoftwareX*, vol. 20, p. 101202, 2022.

[79] S. Balay, K. Buschelman, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith, and H. Zhang, “Petsc,” *See http://www. mcs. anl. gov/petsc*, 2001.

[80] B. S. Kirk, J. W. Peterson, R. H. Stogner, and G. F. Carey, “libmesh: a c++ library for parallel adaptive mesh refinement/coarsening simulations,” *Engineering with Computers*, vol. 22, pp. 237–254, 2006.

[81] G. Blöcher, S. Regenspurg, S. Kranz, M. Lipus, L. Pei, B. Norden, T. Reinsch, J. Henninges, R. Siemon, D. Orenczuk, *et al.*, “Best practices for characterization of high temperature-aquifer thermal energy storage (ht-ates) potential using well tests in berlin (germany) as an example,” *Geothermics*, vol. 116, p. 102830, 2024.

[82] R. Seithel, E. Gaucher, B. Mueller, U. Steiner, and T. Kohl, “Probability of fault reactivation in the bavarian molasse basin,” *Geothermics*, vol. 82, pp. 81–90, 2019.

[83] T. Kraft, J. Wassermann, N. Deichmann, and S. Stange, “The 2008 earthquakes in the bavarian molasse basin-possible relation to deep geothermics?,” *EGU General Assembly*, 2009.

[84] A. Wilkins, C. P. Green, and J. Ennis-King, “An open-source multiphysics simulation code for coupled problems in porous media,” *Computers & Geosciences*, vol. 154, p. 104820, 2021.

[85] D. F. McTigue, “Thermoelastic response of fluid-saturated porous rock,” *Journal of Geophysical Research: Solid Earth*, vol. 91, no. B9, pp. 9533–9542, 1986.

[86] D. Malik, Z. Abidin, and B. Pramono, “Tritium tracer injection test at wayang windu geothermal field, west java, indonesia,” *Geothermics*, vol. 83, p. 101718, 2020.

[87] B. Sanjuan, J.-L. Pinault, P. Rose, A. Gérard, M. Brach, G. Braibant, C. Crouzet, J.-C. Foucher, A. Gautier, and S. Touzelet, “Tracer testing of the geothermal heat exchanger at soultz-sous-forêts (france) between 2000 and 2005,” *Geothermics*, vol. 35, no. 5-6, pp. 622–653, 2006.

[88] L. Li, H. Jiang, K. Wu, J. Li, and Z. Chen, “An analysis of tracer flowback profiles to reduce uncertainty in fracture-network geometries,” *Journal of Petroleum Science and Engineering*, vol. 173, pp. 246–257, 2019.

[89] J. Lu, P. J. Cook, S. A. Hosseini, C. Yang, K. D. Romanak, T. Zhang, B. M. Freifeld, R. C. Smyth, H. Zeng, and S. D. Hovorka, “Complex fluid flow revealed by monitoring co 2 injection in a fluvial formation,” *Journal of Geophysical Research: Solid Earth*, vol. 117, no. B3, 2012.

[90] R. S. Middleton, J. W. Carey, R. P. Currier, J. D. Hyman, Q. Kang, S. Karra, J. Jiménez-Martínez, M. L. Porter, and H. S. Viswanathan, “Shale gas and non-aqueous fracturing fluids: Opportunities and challenges for supercritical co2,” *Applied Energy*, vol. 147, pp. 500–509, 2015.

[91] A. Flohr, J. M. Matter, R. H. James, K. Saw, R. Brown, J. Gros, S. Flude, C. Day, K. Peel, D. Connelly, C. R. Pearce, J. A. Strong, A. Lichtschlag, D. J. Hillegonds, C. J. Ballentine, and R. L. Tyne, "Utility of natural and artificial geochemical tracers for leakage monitoring and quantification during an offshore controlled co2 release experiment," *International Journal of Greenhouse Gas Control*, vol. 111, p. 103421, 2021.

[92] C. Jenkins, A. Chadwick, and S. D. Hovorka, "The state of the art in monitoring and verification—ten years on," *International Journal of Greenhouse Gas Control*, vol. 40, pp. 312–349, 2015.

[93] T. Hadgu, S. Karra, E. Kalinina, N. Makedonska, J. D. Hyman, K. Klise, H. S. Viswanathan, and Y. Wang, "A comparative study of discrete fracture network and equivalent continuum models for simulating flow and transport in the far field of a hypothetical nuclear waste repository in crystalline host rock," *Journal of Hydrology*, vol. 553, pp. 59–70, 2017.

[94] E. Klein, S. M. L. Hardie, W. Kickmaier, and I. G. McKinley, "Testing repository safety assessment models for deep geological disposal using legacy contaminated sites," *The Science of the total environment*, vol. 776, p. 145949, 2021.

[95] C. Cao, Z. Xu, J. Chai, and Y. Li, "Radial fluid flow regime in a single fracture under high hydraulic pressure during shear process," *Journal of Hydrology*, vol. 579, p. 124142, 2019.

[96] D. L. Siler and J. D. Pepin, "3-d geologic controls of hydrothermal fluid flow at brady geothermal field, nevada, usa," *Geothermics*, vol. 94, p. 102112, 2021.

[97] Q. Zhang, S. Luo, H. Ma, X. Wang, and J. Qian, "Simulation on the water flow affected by the shape and density of roughness elements in a single rough fracture," *Journal of Hydrology*, vol. 573, pp. 456–468, 2019.

[98] Y. W. Tsang, C. F. Tsang, F. V. Hale, and B. Dverstorp, "Tracer transport in a stochastic continuum model of fractured media," *Water Resources Research*, vol. 32, no. 10, pp. 3077–3092, 1996.

[99] S. F. Cox, "Chapter 2: The dynamics of permeability enhancement and fluid flow in overpressured, fracture-controlled hydrothermal systems," in *APPLIED STRUCTURAL GEOLOGY OF ORE-FORMING HYDROTHERMAL SYSTEMS* (J. V. Rowland and D. A. Rhys, eds.), pp. 25–82, Society of Economic Geologists, 2020.

[100] R. Egert, F. Nitschke, M. Gholami Korzani, and T. Kohl, "Stochastic 3d navier–stokes flow in self–affine fracture geometries controlled by anisotropy and channeling," *Geophysical Research Letters*, vol. 48, no. 9, 2021.

[101] J. P. Fairley and J. J. Hinds, "Rapid transport pathways for geothermal fluids in an active great basin fault zone," *Geology*, vol. 32, no. 9, p. 825, 2004.

[102] T. Phillips, N. Kampman, K. Bisdom, 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*, vol. 211, p. 103390, 2020.

[103] F. Konrad, A. Savvatis, D. Degen, F. Wellmann, F. Einsiedl, and K. Zosseder, “Productivity enhancement of geothermal wells through fault zones: Efficient numerical evaluation of a parameter space for the upper jurassic aquifer of the north alpine foreland basin,” *Geothermics*, vol. 95, p. 102119, 2021.

[104] T. S. Faleide, A. Braathen, I. Lecomte, M. J. Mulrooney, I. Midtkandal, A. J. Bugge, and S. Planke, “Impacts of seismic resolution on fault interpretation: Insights from seismic modelling,” *Tectonophysics*, vol. 816, p. 229008, 2021.

[105] E. A. H. Michie, M. J. Mulrooney, and A. Braathen, *Fault Interpretation Uncertainties using Seismic Data, and the Effects on Fault Seal Analysis: A Case Study from the Horda Platform, with Implications for CO<sub>2</sub> storage*. 2021.

[106] P. Røe, F. Georgsen, and P. Abrahamsen, “An uncertainty model for fault shape and location,” *Mathematical Geosciences*, vol. 46, no. 8, pp. 957–969, 2014.

[107] Y. An, J. Guo, Q. Ye, C. Childs, J. Walsh, and R. Dong, “Deep convolutional neural network for automatic fault recognition from 3d seismic datasets,” *Computers & Geosciences*, vol. 153, p. 104776, 2021.

[108] A. Cunha, A. Pochet, H. Lopes, and M. Gattass, “Seismic fault detection in real data using transfer learning from a convolutional neural network pre-trained with synthetic seismic data,” *Computers & Geosciences*, vol. 135, p. 104344, 2020.

[109] Z. Wang, H. Di, M. A. Shafiq, Y. Alaudah, and G. AlRegib, “Successful leveraging of image processing and machine learning in seismic structural interpretation: A review,” *The Leading Edge*, vol. 37, no. 6, pp. 451–461, 2018.

[110] A. Genter, C. Castaing, C. Dezayes, H. Tenzer, H. Traineau, and T. Villemin, “Comparative analysis of direct (core) and indirect (borehole imaging tools) collection of fracture data in the hot dry rock soultz reservoir (france),” *Journal of Geophysical Research: Solid Earth*, vol. 102, no. B7, pp. 15419–15431, 1997.

[111] M. J. Pyrcz and C. D. White, “Uncertainty in reservoir modeling,” *Interpretation*, vol. 3, no. 2, pp. SQ7–SQ19, 2015.

[112] V. Cao, M. Schaffer, R. Taherdangkoo, and T. Licha, “Solute reactive tracers for hydrogeological applications: A short review and future prospects,” *Water*, vol. 12, no. 3, p. 653, 2020.

[113] R. N. Horne and F. Rodriguez, “Dispersion in tracer flow in fractured geothermal systems,” *Geophysical Research Letters*, vol. 10, no. 4, pp. 289–292, 1983.

[114] J. Caers, *Modeling Uncertainty in the Earth Sciences*. Chichester, UK: John Wiley & Sons, Ltd, 2011.

[115] F. A. Stamm, M. de La Varga, and F. Wellmann, “Actors, actions, and uncertainties: optimizing decision-making based on 3-d structural geological models,” *Solid Earth*, vol. 10, no. 6, pp. 2015–2043, 2019.

[116] Z. Yin, S. Strebelle, and J. Caers, “Automated monte carlo-based quantification and updating of geological uncertainty with borehole data (autobel v1.0),” *Geoscientific Model Development*, vol. 13, no. 2, pp. 651–672, 2020.

[117] S. Erol, P. Bayer, T. Akın, and S. Akın, “Advanced workflow for multi-well tracer test analysis in a geothermal reservoir,” *Geothermics*, vol. 101, p. 102375, 2022.

[118] X. Li, Z. Wen, H. Zhan, F. Wu, and Q. Zhu, “Laboratory observations for two-dimensional solute transport in an aquifer-aquitard system,” *Environmental science and pollution research international*, vol. 28, no. 29, pp. 38664–38678, 2021.

[119] N. Toride, F. J. Leij, and M. T. van Genuchten, “A comprehensive set of analytical solutions for nonequilibrium solute transport with first-order decay and zero-order production,” *Water Resources Research*, vol. 29, no. 7, pp. 2167–2182, 1993.

[120] M. T. van Genuchten, J. Å imunek, F. J. Leij, N. Toride, and M. Å ejna, “Stanmod: Model use, calibration, and validation,” *Transactions of the ASABE*, vol. 55, no. 4, pp. 1355–1368, 2012.

[121] A. Gérard, A. Genter, T. Kohl, P. Lutz, P. Rose, and F. Rummel, “The deep egs (enhanced geothermal system) project at soultz-sous-forêts (alsace, france),” *Geothermics*, vol. 35, no. 5-6, pp. 473—483, 2006.

[122] C. Dezayes, A. Genter, and B. Valley, “Structure of the low permeable naturally fractured geothermal reservoir at soultz,” *Comptes Rendus Geoscience*, vol. 342, no. 7-8, pp. 517–530, 2010.

[123] R. Egert, M. G. Korzani, S. Held, and T. Kohl, “Implications on large-scale flow of the fractured egs reservoir soultz inferred from hydraulic data and tracer experiments,” *Geothermics*, vol. 84, p. 101749, 2020.

[124] S. Held, A. Genter, T. Kohl, T. Kölbel, J. Sausse, and M. Schoenball, “Economic evaluation of geothermal reservoir performance through modeling the complexity of the operating egs in soultz-sous-forêts,” *Geothermics*, vol. 51, pp. 270–280, 2014.

[125] G. R. Hooijkaas, A. Genter, and C. Dezayes, “Deep-seated geology of the granite intrusions at the soultz egs site based on data from 5km-deep boreholes,” *Geothermics*, vol. 35, no. 5-6, pp. 484–506, 2006.

[126] T. Kohl and T. Mégel, “Predictive modeling of reservoir response to hydraulic stimulations at the european egs site soultz-sous-forêts,” *International Journal of Rock Mechanics and Mining Sciences*, vol. 44, no. 8, pp. 1118–1131, 2007.

[127] S. Karmakar, J. Ghergut, and M. Sauter, “Early-flowback tracer signals for fracture characterization in an egs developed in deep crystalline and sedimentary formations: a parametric study,” *Geothermics*, vol. 63, pp. 242–252, 2016.

[128] J. Sausse, C. Dezayes, L. Dorbath, A. Genter, and J. Place, “3d model of fracture zones at soultz-sous-forêts based on geological data, image logs, induced microseismicity and vertical seismic profiles,” *Comptes Rendus Geoscience*, vol. 342, no. 7-8, pp. 531–545, 2010.

[129] J. Schmittbuhl, A. Steyer, L. Jouniaux, and R. Toussaint, “Fracture morphology and viscous transport,” *International Journal of Rock Mechanics and Mining Sciences*, vol. 45, no. 3, pp. 422–430, 2008.

[130] A. Schaaf, M. de La Varga, F. Wellmann, and C. E. Bond, *Constraining stochastic 3-D structural geological models with topology information using Approximate Bayesian Computation using GemPy 2.1.* 2020.

[131] K. Ho-Le, “Finite element mesh generation methods: a review and classification,” *Computer-Aided Design*, vol. 20, no. 1, pp. 27–38, 1988.

[132] M. Okereke and S. Keates, eds., *Finite Element Applications*. Springer Tracts in Mechanical Engineering, Cham: Springer International Publishing, 2018.

[133] D. Gaston, C. Newman, G. Hansen, and D. Lebrun-Grandié, “Moose: A parallel computational framework for coupled systems of nonlinear equations,” *Nuclear Engineering and Design*, vol. 239, no. 10, pp. 1768–1778, 2009.

[134] A. N. Brooks and T. J. Hughes, “Streamline upwind/petrov-galerkin formulations for convection dominated flows with particular emphasis on the incompressible navier-stokes equations,” *Computer Methods in Applied Mechanics and Engineering*, vol. 32, no. 1-3, pp. 199–259, 1982.

[135] A. Masud and T. J. Hughes, “A stabilized mixed finite element method for darcy flow,” *Computer Methods in Applied Mechanics and Engineering*, vol. 191, no. 39-40, pp. 4341–4370, 2002.

[136] J. Kestin, H. E. Khalifa, and R. J. Correia, “Tables of the dynamic and kinematic viscosity of aqueous kcl solutions in the temperature range 25–150 °c and the pressure range 0.1–35 mpa,” *Journal of Physical and Chemical Reference Data*, vol. 10, no. 1, pp. 57–70, 1981.

[137] Q. Zhou, H.-H. Liu, F. J. Molz, Y. Zhang, and G. S. Bodvarsson, “Field-scale effective matrix diffusion coefficient for fractured rock: results from literature survey,” *Journal of contaminant hydrology*, vol. 93, no. 1-4, pp. 161–187, 2007.

[138] M. W. Becker and A. M. Shapiro, “Tracer transport in fractured crystalline rock: Evidence of nondiffusive breakthrough tailing,” *Water Resources Research*, vol. 36, no. 7, pp. 1677–1686, 2000.

[139] M. W. Becker and A. M. Shapiro, “Interpreting tracer breakthrough tailing from different forced-gradient tracer experiment configurations in fractured bedrock,” *Water Resources Research*, vol. 39, no. 1, 2003.

[140] J. Bodin, “Mfit 1.0.0: Multi-flow inversion of tracer breakthrough curves in fractured and karst aquifers,” *Geoscientific Model Development*, vol. 13, no. 6, pp. 2905–2924, 2020.

[141] I. T. Tenebe, A. S. Ogbiye, D. O. Omole, and P. C. Emenike, “Estimation of longitudinal dispersion co-efficient: A review,” *Cogent Engineering*, vol. 3, no. 1, p. 1216244, 2016.

[142] P. Maloszewski and A. Zuber, “Mathematical modeling of tracer behavior in short-term experiments in fissured rocks,” *Water Resources Research*, vol. 26, no. 7, pp. 1517–1528, 1990.

[143] M. Hauns, P.-Y. Jeannin, and O. Atteia, “Dispersion, retardation and scale effect in tracer breakthrough curves in karst conduits,” *Journal of Hydrology*, vol. 241, no. 3-4, pp. 177–193, 2001.

[144] M. Huysmans and A. Dassargues, “Review of the use of péclet numbers to determine the relative importance of advection and diffusion in low permeability environments,” *Hydrogeology Journal*, vol. 13, no. 5-6, pp. 895–904, 2005.

[145] Z. Zhao, L. Jing, and I. Neretnieks, “Evaluation of hydrodynamic dispersion parameters in fractured rocks,” *Journal of Rock Mechanics and Geotechnical Engineering*, vol. 2, no. 3, pp. 243–254, 2010.

[146] P. Fleuchaus, B. Godschalk, I. Stober, and P. Blum, “Worldwide application of aquifer thermal energy storage – a review,” *Renewable and Sustainable Energy Reviews*, vol. 94, pp. 861–876, 2018.

[147] M. Wesselink, W. Liu, J. Koornneef, and M. van den Broek, “Conceptual market potential framework of high temperature aquifer thermal energy storage - a case study in the netherlands,” *Energy*, vol. 147, pp. 477–489, 2018.

[148] A. Réveillère, V. Hamm, H. Lesueur, E. Cordier, and P. Goblet, “Geothermal contribution to the energy mix of a heating network when using aquifer thermal energy storage: Modeling and application to the paris basin,” *Geothermics*, vol. 47, pp. 69–79, 2013.

[149] P. Fleuchaus, S. Schüppler, B. Godschalk, G. Bakema, and P. Blum, “Performance analysis of aquifer thermal energy storage (ates),” *Renewable Energy*, vol. 146, pp. 1536–1548, 2020.

[150] Q. Wang, W. Shi, H. Zhan, and X. Xiao, “New model of single-well push-pull thermal test in a fracture-matrix system,” *Journal of hydrology*, vol. 585, p. 124807, 2020.

[151] T. Schmidt, T. Pauschinger, P. A. Sørensen, A. Snijders, R. Djebbar, R. Boulter, and J. Thornton, “Design aspects for large-scale pit and aquifer thermal energy storage for district heating and cooling,” *Energy Procedia*, vol. 149, pp. 585–594, 2018.

[152] H. Mahon, D. O’Connor, D. Friedrich, and B. Hughes, “A review of thermal energy storage technologies for seasonal loops,” *Energy*, vol. 239, p. 122207, 2022.

[153] M. Bloemendal, T. Olsthoorn, and F. Boons, “How to achieve optimal and sustainable use of the subsurface for aquifer thermal energy storage,” *Energy Policy*, vol. 66, pp. 104–114, 2014.

[154] L. Gao, J. Zhao, Q. An, X. Liu, and Y. Du, “Thermal performance of medium-to-high-temperature aquifer thermal energy storage systems,” *Applied Thermal Engineering*, vol. 146, pp. 898–909, 2019.

[155] D. Dinkelman and F. van Bergen, eds., *Evaluation of the country-wide potential for High-Temperature Aquifer Thermal Energy Storage (HT-ATES) in the Netherlands*, 2022.

[156] L. Pasquinelli, M. Felder, M. L. Gulbrandsen, T. M. Hansen, J.-S. Jeon, N. Molenaar, K. Mosegaard, and I. L. Fabricius, "The feasibility of high-temperature aquifer thermal energy storage in denmark: the gassum formation in the stenlille structure," *Bulletin of the Geological Society of Denmark*, vol. 68, pp. 133–154, 2020.

[157] D. T. Birdsell and M. O. Saar, "Modeling ground surface deformation at the swiss heatstore underground thermal energy storage sites," p. 22046, ETH Zurich, Institute of Geophysics, 2020.

[158] H. Böhm and J. Lindorfer, "Techno-economic assessment of seasonal heat storage in district heating with thermochemical materials," *Energy*, vol. 179, pp. 1246–1264, 2019.

[159] K. Stricker, J. C. Grimmer, R. Egert, J. Bremer, M. G. Korzani, E. Schill, and T. Kohl, "The potential of depleted oil reservoirs for high-temperature storage systems," *Energies*, vol. 13, no. 24, p. 6510, 2020.

[160] S. Green, J. McLennan, P. Panja, K. Kitz, R. Allis, and J. Moore, "Geothermal battery energy storage," *Renewable Energy*, vol. 164, pp. 777–790, 2021.

[161] J. Mindel and T. Driesner, "Heatstore: Preliminary design of a high temperature aquifer thermal energy storage (ht-ates) system in geneva based on th simulations."

[162] P. Fleuchaus, S. Schüppler, M. Bloemendal, L. Guglielmetti, O. Opel, and P. Blum, "Risk analysis of high-temperature aquifer thermal energy storage (ht-ates)," *Renewable and Sustainable Energy Reviews*, vol. 133, p. 110153, 2020.

[163] Z. Li, M. Dong, S. Li, and S. Huang, "Co2 sequestration in depleted oil and gas reservoirs—caprock characterization and storage capacity," *Energy Conversion and Management*, vol. 47, no. 11-12, pp. 1372–1382, 2006.

[164] N. S. Muhammed, M. B. Haq, D. A. Al Shehri, A. Al-Ahmed, M. M. Rahman, E. Zaman, and S. Iglauer, "Hydrogen storage in depleted gas reservoirs: A comprehensive review," *Fuel*, vol. 337, p. 127032, 2023.

[165] A. Dashti, M. Gholami Korzani, C. Geuzaine, R. Egert, and T. Kohl, "Impact of structural uncertainty on tracer test design in faulted geothermal reservoirs," *Geothermics*, vol. 107, p. 102607, 2023.

[166] M. Collignon, Ø. S. Klemetsdal, O. Møyner, M. Alcanié, A. P. Rinaldi, H. Nilsen, and M. Lupi, "Evaluating thermal losses and storage capacity in high-temperature aquifer thermal energy storage (ht-ates) systems with well operating limits: insights from a study-case in the greater geneva basin, switzerland," *Geothermics*, vol. 85, p. 101773, 2020.

[167] J. Kuhlemann and O. Kempf, "Post-eocene evolution of the north alpine foreland basin and its response to alpine tectonics," *Sedimentary Geology*, vol. 152, no. 1-2, pp. 45–78, 2002.

[168] C. Chelle-Michou, D. Do Couto, A. Moscariello, P. Renard, and E. Rusillon, "Geothermal state of the deep western alpine molasse basin, france-switzerland," *Geothermics*, vol. 67, pp. 48–65, 2017.

[169] L. Rybach, *Geothermal potential of the Swiss Molasse basin*. 1992.

[170] L. Guglielmetti, M. Heidinger, F. Eichinger, and A. Moscariello, "Hydrochemical characterization of groundwaters' fluid flow through the upper mesozoic carbonate geothermal reservoirs in the geneva basin: An evolution more than 15,000 years long," *Energies*, vol. 15, no. 10, p. 3497, 2022.

[171] R. Feng, D. Grana, and N. Balling, "Uncertainty quantification in fault detection using convolutional neural networks," *GEOPHYSICS*, vol. 86, no. 3, pp. M41–M48, 2021.

[172] C. E. Bond, "Uncertainty in structural interpretation: Lessons to be learnt," *Journal of Structural Geology*, vol. 74, pp. 185–200, 2015.

[173] F. Wellmann, J. von Harten, J. Niederau, A. Jüstel, and N. Koltzer, "The role of probabilistic geomodelling in geothermal resource estimation," in *Proceedings Stanford Geothermal Workshop*, 2024.

[174] E. Lüschen, M. Dussel, R. Thomas, and R. Schulz, "3d seismic survey for geothermal exploration at unterhaching, munich, germany," *First Break*, vol. 29, no. 1, 2011.

[175] J. C. Grimmer, J. R. R. Ritter, G. H. Eisbacher, and W. Fielitz, "The late variscan control on the location and asymmetry of the upper rhine graben," *International Journal of Earth Sciences*, vol. 106, no. 3, pp. 827–853, 2017.

[176] P. Dèzes, S. M. Schmid, and P. A. Ziegler, "Evolution of the european cenozoic rift system: interaction of the alpine and pyrenean orogens with their foreland lithosphere," *Tectonophysics*, vol. 389, no. 1-2, pp. 1–33, 2004.

[177] M. E. Schumacher, "Upper rhine graben: Role of preexisting structures during rift evolution," *Tectonics*, vol. 21, no. 1, pp. 6–1–6–17, 2002.

[178] T. Agemar, R. Schellschmidt, and R. Schulz, "Subsurface temperature distribution in germany," *Geothermics*, vol. 44, pp. 65–77, 2012.

[179] P. Baillieux, E. Schill, J.-B. Edel, and G. Mauri, "Localization of temperature anomalies in the upper rhine graben: insights from geophysics and neotectonic activity," *International Geology Review*, vol. 55, no. 14, pp. 1744–1762, 2013.

[180] D. Pribnow and R. Schellschmidt, "Thermal tracking of upper crustal fluid flow in the rhine graben," *Geophysical Research Letters*, vol. 27, no. 13, pp. 1957–1960, 2000.

[181] C. Reinhold, M. Schwarz, and M. Perner, "The northern upper rhine graben : re-dawn of a mature petroleum province?," 2016.

[182] E. Wirth, "Die erdöllagerstätten badens," *Abh. Geol. Landesamt Baden-Württemberg: Freiburg, Germany*, vol. 4, pp. 63–80, 1962.

[183] J. Böcker, R. Littke, and A. Forster, “An overview on source rocks and the petroleum system of the central upper rhine graben,” *International Journal of Earth Sciences*, vol. 106, no. 2, pp. 707–742, 2017.

[184] S. Glubokovskikh, E. Saygin, S. Shapiro, B. Gurevich, R. Isaenkov, D. Lumley, R. Nakata, J. Drew, and R. Pevzner, “A small co 2 leakage may induce seismicity on a sub–seismic fault in a good–porosity clastic saline aquifer,” *Geophysical Research Letters*, vol. 49, no. 12, 2022.

[185] L. Gong, B. Liu, X. Fu, H. Jabbari, S. Gao, W. Yue, H. Yuan, R. Fu, and Z. Wang, “Quantitative prediction of sub-seismic faults and their impact on waterflood performance: Bozhong 34 oilfield case study,” *Journal of Petroleum Science and Engineering*, vol. 172, pp. 60–69, 2019.

[186] A. Rotevatn and H. Fossen, “Simulating the effect of subseismic fault tails and process zones in a siliciclastic reservoir analogue: Implications for aquifer support and trap definition,” *Marine and Petroleum Geology*, vol. 28, no. 9, pp. 1648–1662, 2011.

[187] R. Harris, K. Bracken, B. Miller, S. Angelovich, and T. O’Toole, “Subseismic fault identification using the fault likelihood attribute: Application to geosteering in the dj basin,” in *Proceedings of the 7th Unconventional Resources Technology Conference*, (Tulsa, OK, USA), American Association of Petroleum Geologists, 2019.

[188] E. Damsleth, V. Sangolt, and G. Aamodt, “Sub-seismic faults can seriously affect fluid flow in the njord field off western norway - a stochastic fault modeling case study,” in *All Days*, SPE, 1998.

[189] M. Cacace and A. B. Jacquey, “Flexible parallel implicit modelling of coupled thermal–hydraulic–mechanical processes in fractured rocks,” *Solid Earth*, vol. 8, no. 5, pp. 921–941, 2017.

[190] M. Cacace and G. Blöcher, “Meshit—a software for three dimensional volumetric meshing of complex faulted reservoirs,” *Environmental Earth Sciences*, vol. 74, pp. 5191–5209, 2015.

[191] P. Bauer, A. Thorpe, and G. Brunet, “The quiet revolution of numerical weather prediction,” *Nature*, vol. 525, no. 7567, pp. 47–55, 2015.

[192] S. N. Doost, D. Ghista, B. Su, L. Zhong, and Y. S. Morsi, “Heart blood flow simulation: a perspective review,” *Biomedical engineering online*, vol. 15, no. 1, p. 101, 2016.

[193] A. Benoit, L. Lefèvre, A.-C. Orgerie, and I. Raïs, “Reducing the energy consumption of large-scale computing systems through combined shutdown policies with multiple constraints,” *The International Journal of High Performance Computing Applications*, vol. 32, no. 1, pp. 176–188, 2018.

[194] E. Borgonovo and E. Plischke, “Sensitivity analysis: A review of recent advances,” *European Journal of Operational Research*, vol. 248, no. 3, pp. 869–887, 2016.

[195] A. Abbaszadeh Shahri, C. Shan, and S. Larsson, “A novel approach to uncertainty quantification in groundwater table modeling by automated predictive deep learning,” *Natural Resources Research*, vol. 31, no. 3, pp. 1351–1373, 2022.

[196] C. Soize, *Uncertainty Quantification*, vol. 47. Cham: Springer International Publishing, 2017.

[197] S. L. Brunton and J. N. Kutz, *Data-driven science and engineering: Machine learning, dynamical systems, and control.* Cambridge, United Kingdom, New York, NY: Cambridge University Press, 2022.

[198] T. Stadelmann, V. Tolkachev, B. Sick, J. Stampfli, and O. Dürr, "Beyond imagenet: Deep learning in industrial practice," in *Applied Data Science* (M. Braschler, T. Stadelmann, and K. Stockinger, eds.), pp. 205–232, Cham: Springer International Publishing, 2019.

[199] E. R. Okoroafor, C. M. Smith, K. I. Ochie, C. J. Nwosu, H. Gudmundsdottir, and M. Aljubran, "Machine learning in subsurface geothermal energy: Two decades in review," *Geothermics*, vol. 102, p. 102401, 2022.

[200] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová, "Machine learning and the physical sciences," *Reviews of Modern Physics*, vol. 91, no. 4, 2019.

[201] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019.

[202] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, "Physics-informed machine learning," *Nature Reviews Physics*, vol. 3, no. 6, pp. 422–440, 2021.

[203] E. Kharazmi, Z. Zhang, and G. E. Karniadakis, "Variational physics-informed neural networks for solving partial differential equations."

[204] E. Knapp, M. Battaglia, T. Stadelmann, S. Jenatsch, and B. Ruhstaller, "Xgboost trained on synthetic data to extract material parameters of organic semiconductors," in *2021 8th Swiss Conference on Data Science (SDS)*, pp. 46–51, IEEE, 2021.

[205] J. Yu, L. Lu, X. Meng, and G. E. Karniadakis, "Gradient-enhanced physics-informed neural networks for forward and inverse pde problems," *Computer Methods in Applied Mechanics and Engineering*, vol. 393, p. 114823, 2022.

[206] D. Degen, D. Caviedes Voullième, S. Buiter, H.-J. Hendriks Franssen, H. Vereecken, A. González-Nicolás, and F. Wellmann, *Perspectives of Physics-Based Machine Learning for Geoscientific Applications Governed by Partial Differential Equations.* 2023.

[207] P. Meakin and A. M. Tartakovsky, "Modeling and simulation of pore-scale multiphase fluid flow and reactive transport in fractured and porous media," *Reviews of Geophysics*, vol. 47, no. 3, 2009.

[208] D. Zhou, A. Tatomir, A. Niemi, C.-F. Tsang, and M. Sauter, "Study on the influence of randomly distributed fracture aperture in a fracture network on heat production from an enhanced geothermal system (egs)," *Energy*, vol. 250, p. 123781, 2022.

[209] H. Gudmundsdottir and R. Horne, "Prediction modeling for geothermal reservoirs using deep learning," *PROCEEDINGS, 45th Workshop on Geothermal Reservoir Engineering*, 2020.

[210] A. S. Weigend and N. A. Gershenfeld, eds., *Time series prediction: Forecasting the future and understanding the past : proceedings of the NATO Advanced Research Workshop on Comparative Time Series Analysis, held in Santa Fe, New Mexico, May 14-17, 1992*, vol. XV of *A Proceedings volume, Santa Fe Institute studies in the sciences of complexity*. New York, NY: Routledge, Taylor & Francis Group, 1994.

[211] A. Alakeely and R. N. Horne, "Simulating the behavior of reservoirs with convolutional and recurrent neural networks," *SPE Reservoir Evaluation & Engineering*, vol. 23, no. 03, pp. 0992–1005, 2020.

[212] S. B. Kotsiantis, "Decision trees: a recent overview," *Artificial Intelligence Review*, vol. 39, no. 4, pp. 261–283, 2013.

[213] Y. Liu, Y. Wang, and J. Zhang, "New machine learning algorithm: Random forest," in *Information Computing and Applications* (D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, B. Liu, M. Ma, and J. Chang, eds.), vol. 7473 of *Lecture Notes in Computer Science*, pp. 246–252, Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.

[214] M. XU, P. WATANACHATURAPORN, P. VARSHNEY, and M. ARORA, "Decision tree regression for soft classification of remote sensing data," *Remote Sensing of Environment*, vol. 97, no. 3, pp. 322–336, 2005.

[215] W.-Y. Loh, "Classification and regression trees," *Wiley interdisciplinary reviews: data mining and knowledge discovery*, vol. 1, no. 1, pp. 14–23, 2011.

[216] L. Breiman, "Random forests," *Machine learning*, vol. 45, pp. 5–32, 2001.

[217] J. H. Friedman, "Stochastic gradient boosting," *Computational Statistics & Data Analysis*, vol. 38, no. 4, pp. 367–378, 2002.

[218] M. Gupta, "A comparative study on supervised machine learning algorithm," *International Journal for Research in Applied Science and Engineering Technology*, vol. 10, no. 1, pp. 1023–1028, 2022.

[219] P. Bühlmann and B. Yu, "Boosting with the l2 loss," *Journal of the American Statistical Association*, vol. 98, no. 462, pp. 324–339, 2003.

[220] G. Fanelli, M. Dantone, J. Gall, A. Fossati, and L. van Gool, "Random forests for real time 3d face analysis," *International Journal of Computer Vision*, vol. 101, no. 3, pp. 437–458, 2013.

[221] C. Shu and D. H. Burn, "Artificial neural network ensembles and their application in pooled flood frequency analysis," *Water Resources Research*, vol. 40, no. 9, 2004.

[222] L. Breiman, "Arcing classifier (with discussion and a rejoinder by the author)," *The Annals of Statistics*, vol. 26, no. 3, 1998.

[223] R. E. Schapire, "The boosting approach to machine learning: An overview," in *Nonlinear Estimation and Classification* (P. Bickel, P. Diggle, S. Fienberg, K. Krickeberg, I. Olkin, N. Wermuth, S. Zeger, D. D. Denison, M. H. Hansen, C. C. Holmes, B. Mallick, and B. Yu, eds.), vol. 171 of *Lecture Notes in Statistics*, pp. 149–171, New York, NY: Springer New York, 2003.

[224] J. He, K. Li, X. Wang, N. Gao, X. Mao, and L. Jia, "A machine learning methodology for predicting geothermal heat flow in the bohai bay basin, china," *Natural Resources Research*, vol. 31, no. 1, pp. 237–260, 2022.

[225] E. Alpaydin, *Introduction to machine learning*. Adaptative Computation and Machine Learning, Cambridge and London: The MIT Press, fourth edition ed., 2020.

[226] F. Hutter, L. Kotthoff, and J. Vanschoren, *Automated Machine Learning*. Cham: Springer International Publishing, 2019.

[227] S. Raschka and V. Mirjalili, *Python machine learning: Machine learning and deep learning with python, scikit-learn, and tensorflow 2*. Birmingham: Packt Publishing, Limited, third edition ed., 2019.

[228] T. M. Oshiro, P. S. Perez, and J. A. Baranauskas, "How many trees in a random forest?," in *Machine Learning and Data Mining in Pattern Recognition* (D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, and P. Perner, eds.), vol. 7376 of *Lecture Notes in Computer Science*, pp. 154–168, Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.

[229] P. Probst, M. N. Wright, and A.-L. Boulesteix, "Hyperparameters and tuning strategies for random forest," *WIREs Data Mining and Knowledge Discovery*, vol. 9, no. 3, 2019.

[230] G. Zhang, M. Y. Hu, B. Eddy Patuwo, and D. C. Indro, "Artificial neural networks in bankruptcy prediction: General framework and cross-validation analysis," *European Journal of Operational Research*, vol. 116, no. 1, pp. 16–32, 1999.

[231] S. An, W. Liu, and S. Venkatesh, "Fast cross-validation algorithms for least squares support vector machine and kernel ridge regression," *Pattern Recognition*, vol. 40, no. 8, pp. 2154–2162, 2007.

[232] G. Erdogan Erten, M. Yavuz, and C. V. Deutsch, "Combination of machine learning and kriging for spatial estimation of geological attributes," *Natural Resources Research*, vol. 31, no. 1, pp. 191–213, 2022.

[233] E. Schill, A. Genter, N. Cuenot, and T. Kohl, "Hydraulic performance history at the soultz egs reservoirs from stimulation and long-term circulation tests," *Geothermics*, vol. 70, pp. 110–124, 2017.

[234] I. S. Moeck, "Catalog of geothermal play types based on geologic controls," *Renewable and Sustainable Energy Reviews*, vol. 37, pp. 867–882, 2014.

[235] K. S. Lee, "A review on concepts, applications, and models of aquifer thermal energy storage systems," *Energies*, vol. 3, no. 6, pp. 1320–1334, 2010.

[236] C. Cröniger, R. Tretter, P. Eichenseer, B. Kleinertz, C. Timpe, V. Bürger, and J. Cludius, "Approach to climate neutral heat supply in munich 2035," *Eur Geotherm Congr*, vol. 2022, pp. 1–29, 2022.

[237] S. H. Wadas, J. F. Krumbholz, V. Shipilin, M. Krumbholz, D. C. Tanner, and H. Buness, "Advanced seismic characterization of a geothermal carbonate reservoir–insight into the structure and diagenesis of a reservoir in the german molasse basin," *Solid Earth*, vol. 14, no. 8, pp. 871–908, 2023.

[238] M. Wolfgramm, J. Bartels, F. Hoffmann, G. Kittl, G. Lenz, P. Seibt, R. Schulz, R. Thomas, and H. Unger, "Unterhaching geothermal well doublet: structural and hydrodynamic reservoir characteristic; bavaria (germany)," in *Proceedings European Geothermal Congress*, vol. 30, p. 2007, Citeseer, 2007.

[239] E. Mraz, *Reservoir characterization to improve exploration concepts of the Upper Jurassic in the Southern Bavarian Molasse Basin.* PhD thesis, Technische Universität München, 2019.

[240] M. Fadel, J. Reinecker, D. Bruss, and I. Moeck, "Causes of a premature thermal breakthrough of a hydrothermal project in germany," *Geothermics*, vol. 105, p. 102523, 2022.

[241] T. Li, S. Shiozawa, and M. W. McClure, "Thermal breakthrough calculations to optimize design of a multiple-stage enhanced geothermal system," *Geothermics*, vol. 64, pp. 455–465, 2016.

[242] Y. Kong, Z. Pang, H. Shao, and O. Kolditz, "Optimization of well-doublet placement in geothermal reservoirs using numerical simulation and economic analysis," *Environmental Earth Sciences*, vol. 76, pp. 1–7, 2017.

[243] M. D. Aliyu and H.-P. Chen, "Sensitivity analysis of deep geothermal reservoir: Effect of reservoir parameters on production temperature," *Energy*, vol. 129, pp. 101–113, 2017.

[244] H. Shi, G. Wang, and C. Lu, "Numerical investigation on delaying thermal breakthrough by regulating reinjection fluid path in multi-aquifer geothermal system," *Applied Thermal Engineering*, vol. 221, p. 119692, 2023.

[245] K. J. Bakhsh, M. Nakagawa, M. Arshad, and L. Dunnington, "Modeling thermal breakthrough in sedimentary geothermal system, using comsol multiphysics," in *Proceedings, 41st Workshop on Geothermal Reservoir Engineering, Stanford, California*, pp. 22–24, 2016.

[246] J. Dickinson, N. Buik, M. Matthews, and A. Snijders, "Aquifer thermal energy storage: theoretical and operational analysis," *Geotechnique*, vol. 59, no. 3, pp. 249–260, 2009.

[247] H. Mahon, D. O'Connor, D. Friedrich, and B. Hughes, "A review of thermal energy storage technologies for seasonal loops," *Energy*, vol. 239, p. 122207, 2022.

[248] X. Lei, S. He, A. Abed, X. Chen, Z. Yang, and Y. Wu, "A generalized interpolation material point method for modelling coupled thermo-hydro-mechanical problems," *Computer Methods in Applied Mechanics and Engineering*, vol. 386, p. 114080, 2021.

[249] A. Ghassemi, “A review of some rock mechanics issues in geothermal reservoir development,” *Geotechnical and Geological Engineering*, vol. 30, pp. 647–664, 2012.

[250] F. Parisio, V. Vilarrasa, W. Wang, O. Kolditz, and T. Nagel, “The risks of long-term re-injection in supercritical geothermal systems,” *Nature communications*, vol. 10, no. 1, p. 4391, 2019.

[251] X. Cui and L. N. Y. Wong, “A 3d fully thermo–hydro–mechanical coupling model for saturated poroelastic medium,” *Computer Methods in Applied Mechanics and Engineering*, vol. 394, p. 114939, 2022.

[252] S. Pandey, V. Vishal, and A. Chaudhuri, “Geothermal reservoir modeling in a coupled thermo-hydro-mechanical-chemical approach: A review,” *Earth-Science Reviews*, vol. 185, pp. 1157–1169, 2018.

[253] M. D. Zoback, *Reservoir geomechanics*. Cambridge university press, 2010.

[254] G. V. Middleton and P. R. Wilcock, *Mechanics in the earth and environmental sciences*. Cambridge University Press, 1994.

[255] M. O. Ziegler and O. Heidbach, “The 3d stress state from geomechanical–numerical modelling and its uncertainties: a case study in the bavarian molasse basin,” *Geothermal Energy*, vol. 8, no. 1, p. 11, 2020.

[256] J. Reinecker, M. Tingay, B. Müller, and O. Heidbach, “Present-day stress orientation in the molasse basin,” *Tectonophysics*, vol. 482, no. 1-4, pp. 129–138, 2010.

[257] R. Egert, E. Gaucher, A. Savvatis, P. Goblirsch, and T. Kohl, “Numerical determination of long-term alterations of thm characteristics of a malm geothermal reservoir during continuous exploitation,” in *Proceedings of the European Geothermal Congress*, 2022.

[258] D. Bohnsack, M. Potten, S. Freitag, F. Einsiedl, and K. Zosseder, “Stress sensitivity of porosity and permeability under varying hydrostatic stress conditions for different carbonate rock types of the geothermal malm reservoir in southern germany,” *Geothermal Energy*, vol. 9, no. 1, p. 15, 2021.

[259] M. Potten, *Geomechanical characterization of sedimentary and crystalline geothermal reservoirs*. PhD thesis, Technische Universität München, 2020.

[260] N. J. Lindsey, H. Rademacher, and J. B. Ajo-Franklin, “On the broadband instrument response of fiber-optic das arrays,” *Journal of Geophysical Research: Solid Earth*, vol. 125, no. 2, p. e2019JB018145, 2020.

[261] M. Van Der Baan, D. Eaton, and M. Dusseault, “Microseismic monitoring developments in hydraulic fracture stimulation,” in *Isrm international conference for effective and sustainable hydraulic fracturing*, pp. ISRM–ICHF, ISRM, 2013.

[262] T. Megies and J. Wassermann, “Microseismicity observed at a non-pressure-stimulated geothermal power plant,” *Geothermics*, vol. 52, pp. 36–49, 2014.

Appendix A

EGC PROCEEDING

Impact of structural uncertainty on tracer test design in faulted geothermal reservoirs

Abstract

Geological models act as a framework for numerical simulations of any underground property. In essence, the sparsity of data and lack of knowledge raise doubts about the reliability of the model. Uncertainty in the data representing subsurface faults is included here to monitor its effect on solute transport. A probabilistic open-source geomodeller (Gempy) is used to construct a generic faulted model in which the location, orientation, and dip of only one fault are perturbed. The developed Python library (named GeoMeshPy) of this study focuses on the discretization of probabilistic geological structures. Hereby we present a synthetic show-case for the capacity of this library to quantify the impact of structural uncertainty. To make the perturbation free of bias, random numbers are used to modulate one of the existing faults. Instead of relying on one single deterministic model, 50 geological realizations are generated to act as frameworks for later numerical simulations. GMSH open-source code is used to discretize each geological model. The new functionality of GMSH also allows for creating surfaces with arbitrary shapes. A tracer test in a doublet system (one injection and one production well) penetrating a geothermal reservoir is simulated in this study. Results of 50 simulations revealed the influential role of the geological uncertainty on both the magnitude and arrival time of the tracer in production well. Even small angular variations up to ±15° for one of the faults would yield differences of up to 26 and 30 percent for peak arrival time and magnitude, respectively.

INTRODUCTION

An understanding of the underground flow regime is a pivotal necessity for any long-term subsurface application. Flow characterization through tracer tests is vital in a wide variety of reservoir engineering. Applications are in the fields of geothermal [86], conventional hydrocarbons [88], shale gas [90], CO_2 storage [91], nuclear or toxic waste disposals [94] and so on. For safety issues, these tests are of high importance since they provide robust underground knowledge on the field scale. Since the fluid flow in fractures typically dominates the total regime [95, 96], the characterization of these zones is crucial for designing exploitation concepts. Wellmann and Regenauer-Lieb 2012 [34] highlighted that lack of data, measurement errors, external disturbances, and human factors lead to ubiquitous uncertainty in geological models. Today, it is stressed that uncertainty estimations should be included in the engineering of underground application [115, 116, 114, 52]. In this work, geological uncertainty is included in designing the field tracer experiments and its effect on flow and transport properties is quantified. Rather than one single deterministic model, 50 realizations are included to cover a range of structural possibilities in a hydraulic solute (HS) modelling framework. These possibly varying geological scenarios should be considered to design a reliable and optimum tracer test. Each model is discretized based on its geological features (layers and faults) before being fed into numerical solver.

METHODOLOGY

From a numerical modelling perspective, usually a deterministic geological realization is constructed and parameters are again adjusted in a way to fit the field data to results. It is highly time-consuming to generate several structural models and later mesh each one. Therefore, the gap between geological models and numerical solvers hampers including structural uncertainty in numerical simulation. Here, we present an automated way to construct these meshes, providing still a large complexity of geological information. Gempy [37], as the modelling tool, is connected to the mesh generator, GMSH [38], to create meshes based on the changing structural model. The developed package, named as GeoMeshPy (available in https://github.com/Ali1990dashti/GeoMeshPy), is able to handle a variety of existing functionalities in Gempy. Available examples in this github repository can visualize the functionality of the developed package. Therefore, 50 different structural realizations with their unique geometries are meshed and fed into the simulator. An open source finite element (FE) application called TIGER (Thermo-Hydro-Chemical simulator for Geoscience Research) [77] is used to simulate a tracer test. Each model is discretized based on its geological features e.g. layers and faults before being fed into numerical solver. BTCs are simulated for several perturbed geological realizations with fixed small scale parameters e.g. permeability, diffusion and dispersion coefficients, and so on. As the next step, generated BTCs are assessed analytically to quantify the effect of the structural uncertainty on extracted parameters. Then, extracted parameters are compared to the input data of simulations regarding the structural model. Completing this loop can help to quantify the effect of structural uncertainty on the flow characters of underground. A schematic view on 50 different geological realizations is visualized in Fig. A.1. Table A.1 also represents parameters used in this study. Designed model is a simplified show-case of Soultz-sous-Forêts Enhanced Geothermal Systems (EGS). The following bullet points summarize the general scheme of this study:

- Generating 50 structural realizations
- Converting outputs of geomodelling tool into useable inputs for the mesh generator
- Meshing each realization based on its geometry
- Simulating tracer test for each realization with fixed operational input parameters

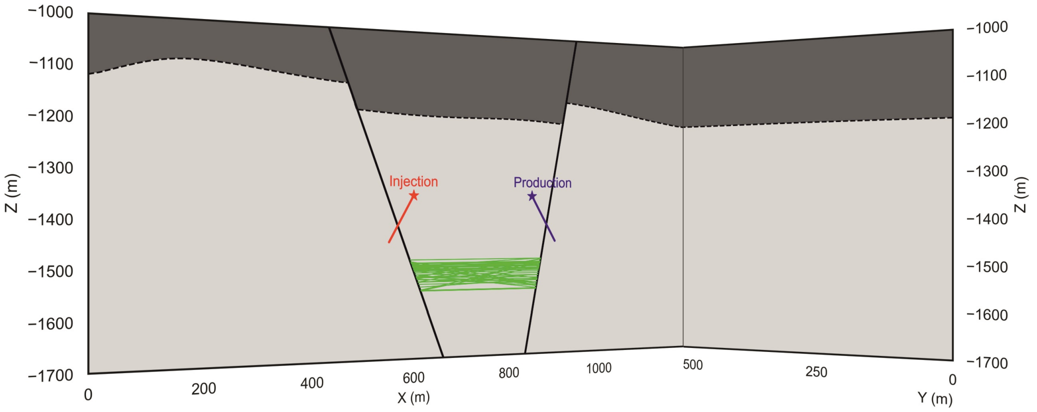

Fig. A.1 A 3D view on the synthetic model and designed doublet system. Injection fault is cut by the injection well and the production fault is cut by the production well. Each green trace is making a unique geological realization.

GEOMESHPY

In this study, Gempy, an open-source code developed in Python ecosystem, is used for probabilistic geological modelling [37]. There are three main issues regarding the outputs of Gempy which GeoMeshPy solves. Firstly, some redundant point exist among vertices. In fact, if we remove these redundancies, the remaining vertices will make a regular grid. As the next step, the relation between faults and layers will be included in GeoMeshPy to split the vertices of layers correctly. As Fig. A.2 shows, the contact of two layers is divided into three patches due to the existence of two active faults in the model. Comparing Fig. A.1 and A.2 can reveal the reason behind splitting the surfaces into three different patches. Fig. A.3-a and b also show another synthetic model to highlight the capabilities of the GeoMeshPy. This model includes five different layers with their unique geometries.

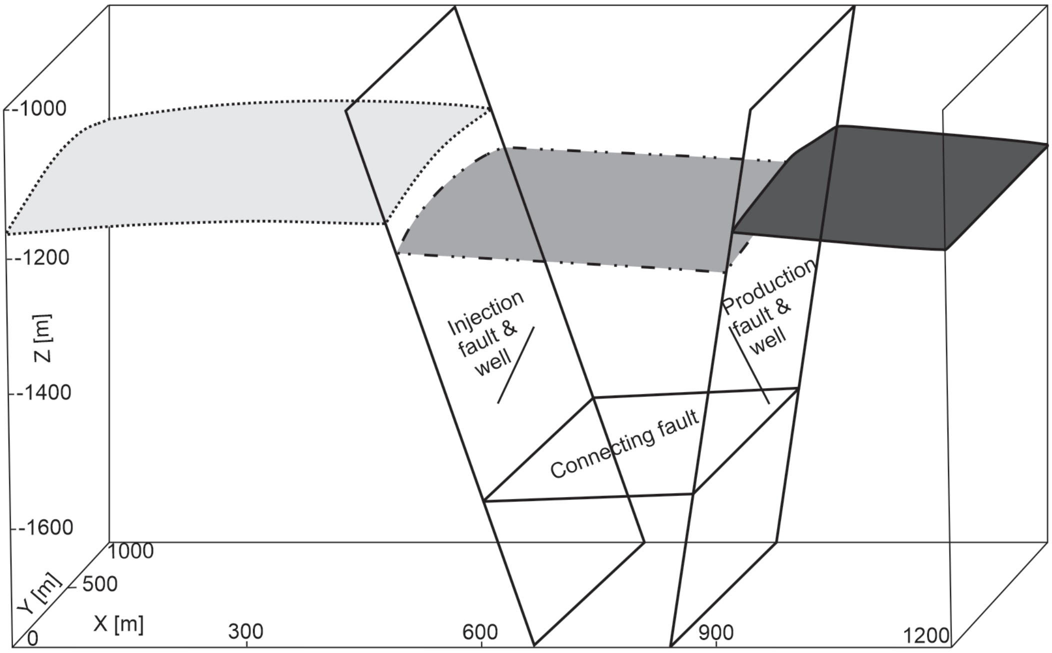

Fig. A.2 All used elements (one contact divided into three patches, three fault surfaces, and two wells) for meshing. Different parts of the unified surface created in Gempy are shown as three patches with different shades of grey and dotted, dash-dotted, and continuous borderlines.

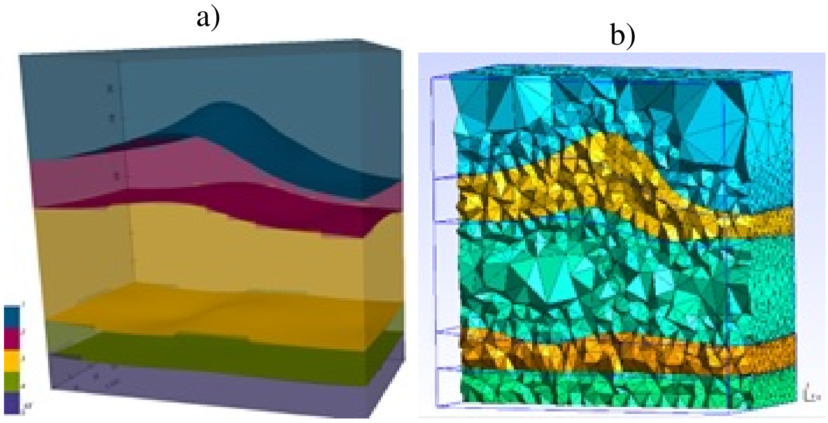

Fig. A.3 a) Another example representing the strength of GeoMeshPy in automating the mesh generation process from Gempy to GMSH. a- a view on the output generated in Gempy. b- a crossed view into the generated mesh in GMSH. Surfaces resulted from Gempy are regenerated in GMSH.

RESULTS AND DISCUSSION

Quantifying the effect of geological uncertainty is overarching goal of the paper. To do so, 50 different geological scenarios are designed to simulate a tracer test. In fact, 50 different structural model with the same parameters (Table A.1) are simulated. Then, resulted BTC and pressure field each scenario is analysed to highlight the effect of geological difference. Fig. A.4 shows a box plot of peak concentration values on the basis of the arrival times. Injection of the tracer has been done one day continuously (from the 8^{th} to the 9^{th} day) and the response is monitored for a one year period in production well. As the plot shows, the absolute peak value and time of the observed peak is changing from scenario to scenario. The earliest peaks (the highest peak magnitudes) are recorded 54 days after stopping the tracer injection (9^{th} day), while the latest peaks are observed after 68 days. This 26% relative change in the time can clearly affect real tracer field experiments. In the case of peak magnitude, the relative change is about 30% that can also affect the expectations from such an important field test.

Table A.1 Parameters used to design a generic tracer test. Data origin is marked a to show it is considered in this study, b, c and d are coming from Egert et al. (2020) [123], Kestin et al. (1981) [136] and Zhou et al. (2007) [137], respectively.

Parameters		**Values**
Permeability [m^2]	Matrix (reservoir)	1.0×10^{-16} [123]
Porosity [-]	Faults	1 [136]
	Matrix (reservoir)	0.05 [136]
Aperture [m]	Injection Fault	0.2 [123]
	Production Fault	0.15 [123]
	Connecting Fault	0.5 [123]
Transmissivity [$m^2\ s^{-1}$] at 25 °C	Injection Fault	1.2×10^{-6}
	Production Fault	3.3×10^{-7}
	Connecting Fault	5.6×10^{-7}

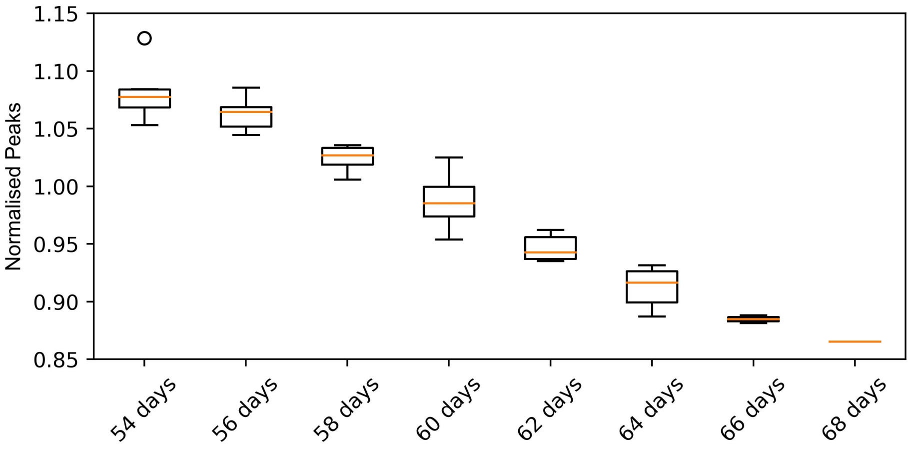

Fig. A.4 A box plot showing the distribution of peak concentrations based on the arrival times.

Considering the generated results from the FE solver (TIGER), structural geology has an influential role in flow parameters in geothermal reservoirs. A schematic 3D view of the designed model (Fig. A.5) demonstrates some of the probably controlling geometrical elements. As mentioned before, the model is certain about the position of two big faults because of both seismic data and their traces in the drilled boreholes. Therefore, the location of two yellow stars representing wells and big faults connection (Fig. A.5) is fixed for all 50 models. However, the third connecting fault is uncertain which fluctuates the probable pathways for tracer transport. Two red stars in Fig. A.5 represent where the injection and connecting faults (connecting$_{p_{\text{inj}}}$) and also production and connecting faults (connecting$_{p_{\text{pro}}}$) meet each other. Different pathways stem from changes in the location of these two points. Therefore, different distances will be resulted from injection fault and well connection with injection and connecting faults (connecting$_{p_{\text{inj}}}$) and so on. These uncertain distances are annotated on the figure as L$_{Inj}$, L$_{Con}$ and LPro. In the generic model, wells are located exactly in the middle of the model (y=500). The mentioned lines (L$_{Inj}$, L$_{Con}$ and L$_{Pro}$) are also passing through the middle of faults surfaces (y=500 m for all three lines).

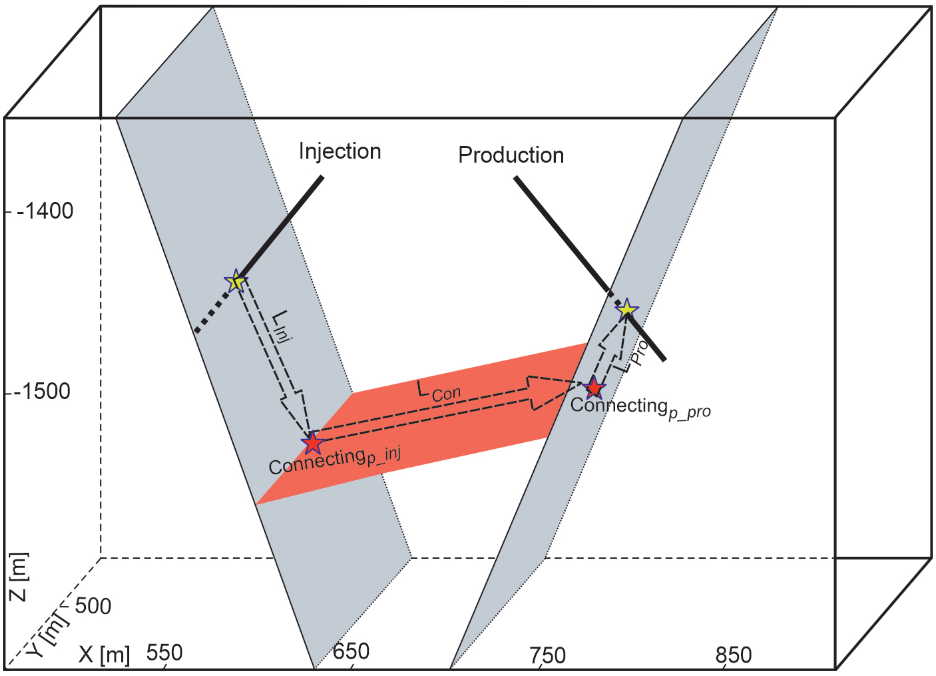

Fig. A.5 A 3D view on the model. The most important geometrical parameters are annotated. Length of injection (L$_{Inj}$) represents the distance from the point of injection well and fault connections (yellow star) to injection and connecting fault connection points (red star). Length of production (L$_{Pro}$) represents the distance from the point of production well and fault connection (yellow star) to production and connecting fault connection points (red star). Length of connection (L$_{Con}$) also represents the length of three faults connection points. Two important and uncertain connection points between faults are also shown by red stars.

The direction of the dip of the connecting fault is also expected to have effects on the results. Therefore, 50 different connecting faults are clustered based on their dipping direction. Fig. A.5 represents a connecting fault that is dipping upward from the injection well toward the production well. In this figure, depth of the connecting$_{p_{\mathrm{inj}}}$ is much higher than connecting$_{p_{\mathrm{pro}}}$. Downward dipping is when the connecting fault cuts the production fault deeper compared to the injection fault. In the other words, connecting$_{p_{\mathrm{inj}}}$ will be shallower than connecting$_{p_{\mathrm{pro}}}$. The situations in which the depth difference of two points is less than 5 m, the connecting fault is considered to be horizontal. Fig. A.6 is presented to visualize the effect of uncertain connection points between connecting and production faults i.e. connecting$_{p_{\mathrm{pro}}}$ (refer to Fig. A.5 to find the red star representing it). Data in this figure are clustered into three groups. The first straightforward conclusion is that the less the length of the pathway, the higher the peak. Secondly, the correlation coefficient converges to a reasonable value for realizations in which the connecting fault is horizontal, while it decreases in the other two clusters. Thirdly, the upward dip realizations have resulted in higher peak concentrations compared to downward dipping realizations. The main fact is that upward dipping connecting faults makes much less values in $_{Pro}$. The reason is that with fewer lengths in the production well the flow should move less distance against gravity i.e. upward in the production well.

CONCLUSION

In this study, stochastic analyses are carried out on a universally applied field experiment. Geological models act as a framework for the next numerical simulations. Meanwhile, experts usually have a basic idea or knowledge about underground structural situations and then analyze available data on the basis of their understanding. Therefore, geology has a twofold application: makes the framework for numerical modelling and is used to interpret available data. An erroneous underground knowledge endangers all the following data

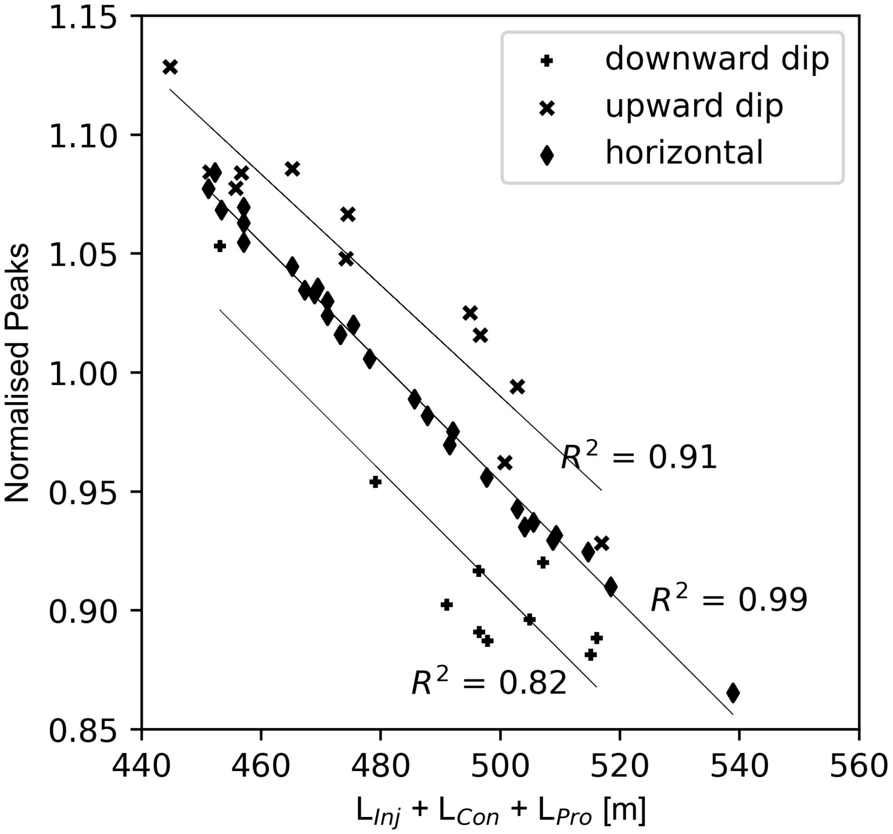

Fig. A.6 Normalized peak concentration values plotted versus the length of the pathway. Based on the dipping angle pf the connecting fault, scenarios are split into three categories: downward, upward and horizontal.

interpretations. This study assessed the effect of a small perturbation in one of the structural models on the outcomes of a tracer test. The tracer test in a model with two big certain (?) faults and one small uncertain fault revealed the effect of the basic knowledge. Routinely one deterministic structural model is used as the framework for parametric sensitivity analysis or arbitrary material sets are introduced on an extremely refined mesh. But a rightful structural uncertainty analysis necessitates a robust link between structural modelling and mesh generation. Otherwise, manually creating meshes for different structural realizations hinders considering the structural uncertainty as it is highly time-consuming. In this study, a new fast and efficient tool named GeoMeshPy is developed to automate the generation of customized well-refined meshes from a range of geological realizations. The mesh generator, GMSH, has been equipped with the new functionality to emulate structures created in geomodelling tools like GemPy automatically. A notably wide range of distinct BTCs are generated thanks to structural uncertainty rather than parametric sensitivity or perturbation of material sets. Accordingly, for the same operational parameters, different responses are generated due to variations of the structural model. This proves that the routine procedure of conducting a tracer test experiment should be adapted to ensure efficient field tests.

Appendix B

GeoMeshPy Example

This appendix presents one of the examples of the GeoMeshPy available in my GitHub repository (https://github.com/Ali1990dashti/GeoMeshPy/blob/main/Examples/No_Fault/Four_Layers.ipynb)

```
# This block of code initializes the first model in Gempy
import gempy as gp
import numpy as np
geo_model = gp.create_model('4_layers')
gp.init_data(geo_model, [0., 100., 0., 50., 0., 100.], [50, 25, 50],
            path_i = 'no_f_interfaces_4.csv',
            path_o = 'no_f_orientations_4.csv');
gp.map_series_to_surfaces(geo_model,
                         {"Strati_series":('unit_1', 'unit_2', 'unit_3', 'unit_4')})
gp.set_interpolator(geo_model,
                         compile_theano=True,
                         theano_optimizer='fast_compile',
                         verbose=[])
sol = gp.compute_model(geo_model)

# this block is exporting required parameters from Gmpy to be used in GeoMeshPy
extent = geo_model.grid.regular_grid.extent
resolution = geo_model.grid.regular_grid.resolution.reshape(-1,1)
df=geo_model.series.df
if len (np.unique (sol.fault_block))>1:
    no_of_faults=df.groupby(by='BottomRelation').count().iloc[1,0]
```

```
else:
    no_of_faults=0
surfaces=geo_model.surface_points.df['surface']
if no_of_faults==0:
    surfaces_layer=[i for i in surfaces.unique()]
else:
    surfaces_layer=[i for i in surfaces.unique()[no_of_faults:]]
    fault_name=[i for i in surfaces.unique()[:no_of_faults]]
surfaces_layer.append('Basement')
grid=geo_model.grid.values
z_resolution = abs (grid[0,-1] - grid[1,-1])
# visualizazion
gp.plot_3d(geo_model, plotter_type='background')
```

Next block of code makes a probabilistic model. In fact, the depths of one the layers (unit_1) are considered to be uncertain. Again, please refer to GemPy documentations for more details.

```
import copy
import math
import random
from numpy import savetxt
ver=[]
fault_ind=[]
repre_pts=[]
n_iter = 10
for i in range(n_iter):
#      Initialization of the Gempy model
    df_int_X      = copy.copy(geo_model.surface_points.df['X'])
    df_int_Y      = copy.copy(geo_model.surface_points.df['Y'])
    df_int_Z      = copy.copy(geo_model.surface_points.df['Z'])
    df_or_X       = copy.copy(geo_model.orientations.df['X'])
    df_or_Y       = copy.copy(geo_model.orientations.df['Y'])
    df_or_Z       = copy.copy(geo_model.orientations.df['Z'])
    df_or_dip     = copy.copy(geo_model.orientations.df['dip'])
    df_or_azimuth = copy.copy(geo_model.orientations.df['azimuth'])
    surfindexes = list(geo_model.surface_points.df.index)
    orindexes = list(geo_model.orientations.df.index)
    geo_model.modify_surface_points(surfindexes, X=df_int_X, Y=df_int_Y,
   Z=df_int_Z)
```

```
    geo_model.modify_orientations(orindexes, X=df_or_X, Y=df_or_Y, Z=df_or_Z,dip = df_or_dip, azimuth = df_or_azimuth)
#     Layers surface point and orientations
    layer_surfpoints = geo_model.surface_points.df.surface.isin([’unit_1’])
    indexes_La_sp = geo_model.surface_points.df[layer_surfpoints].index
    layer_orient = geo_model.orientations.df.surface.isin([’unit_1’])
    index_La_o = geo_model.orientations.df[layer_orient].index
#     Randomization_Method
    std1=1
    rand1 = np.random.uniform(-std1, std1, size=1)
#     Randomized_input
    new_Z_La = geo_model.surface_points.df[’Z’].values[layer_surfpoints] + rand1
    #     Modifier
    geo_model.modify_surface_points(indexes_La_sp, Z=new_Z_La)

#     this block updates the model
    geo_model.update_to_interpolator()
    sol=gp.compute_model(geo_model)

    # Export Block
    ver.append(geo_model.solutions.vertices)
    geo_model.solutions.lith_block = np.round(geo_model.solutions.lith_block)
    fault_ind.append (sol.fault_block.T[0:sol.grid.values.shape[0]]) # of course if you have no fault, it will be an empty list
```

Next cell reshapes exported data from GemPy and also exports them as .csv files to be usable for next times, without running the previous cells. If you do not have GemPy installed, you can use the exported .csv files in the output folder of this example.

```
lays_fault_name=geo_model.surface_points.df.loc[:, ’surface’].unique()
all_vers=[list(column) for column in zip(*ver)]
no_layers = len(all_vers)
df=geo_model.series.df
if len (np.unique (sol.fault_block))>1:
    no_of_faults=df.groupby(by=’BottomRelation’).count().iloc[1,0]
else:
    no_of_faults=0
```

```
len_rough = np.array([[len(x) for x in all_vers[y]] for y in range(len(all_vers))])
savetxt(’len_rough.csv’, len_rough, delimiter=’,’)
rough_vers = [l.tolist() for l in [np.concatenate(d) for d in all_vers]]
for ind, vertices in enumerate(rough_vers):
    savetxt(f’rough_vertices_{ind}.csv’, np.array(vertices), delimiter=’,’)
savetxt(’extent.csv’, extent, delimiter=’,’)
savetxt(’z_resolution.csv’, np.array([z_resolution]), delimiter=’,’)
```

The next block of code imports exported data from GemPy and some features of the model need also be introduced by the user.

```
import numpy as np
import glob
from numpy import genfromtxt
import copy
import math
import random
from numpy import savetxt
extent = genfromtxt(’extent.csv’, delimiter=’,’) # comes from Gempy export cell
z_resolution = genfromtxt(’z_resolution.csv’, delimiter=’,’) # comes from Gempy export cell or can be easily defines
# by the user. this value is an important one because it should be based on the expected mesh size around your surface
len_rough = genfromtxt(’len_rough.csv’, delimiter=’,’) # comes from Gempy export cell
all_roughs = []
files_ve = glob.glob("rough_vertices_*.csv")
files_ver = sorted(files_ve, key=lambda name: int(name[15:-4]))
for name_ver in files_ver:
    data = genfromtxt(name_ver, delimiter=’,’)
    all_roughs.append(data)
all_roughs = [[i] for i in all_roughs]
all_vers = []
for vert, length in zip (all_roughs, len_rough):
    [a] = vert
    new_arrs = np.split(a, np.cumsum(length).astype(’int’).tolist())[:-1]
    all_vers.append(new_arrs)
no_of_faults = 0 # can be also exported from Gempy export cell
n_iter = 10 # can be also exported from Gempy export cell
```

```
name_of_layers = np.array(['unit_1', 'unit_2', 'unit_3', 'unit_4', 'basement']) # can be also exported from Gempy export cell
resolution = np.array([[50], [25], [50]]) # can be also exported from Gempy export cell
```

After installing GeoMeshPy using the geomeshconv package existing in Pypi, you can use the module. For installation you can simply try *pip install geomeshconv*. You can find it in this link https://pypi.org/project/geomeshconv/

```
from GeoMeshPy import vmod # the class vmod allows you for doing all the required calculations
```

Inputs of the class

The next cell allows the vertice_modifier of GeoMeshPy to convert raw Gempy output into readable points for GMSH. Details of needed inputs for the class are addressed in the following:

n_iterations: defines how many geological realizations are going to be used,

no_of_faults: how many faults does exist in the model, if no fault is there keep it as 0,

all_vers: a list of arrays representing all the features, i.e. faults or layers of the model. In case of having faults, mention them first and after them put formations,

formations: an array of formations' names including the last formation (usually called basement in GemPy terminology),

mesh_resolution: for this factor it is needed to have a prediction on the mesh size in the adjacency of layers,

fault_relation: it defines whether there is a passive fault in the model or not. Refer to Examples to see it in more details. Also see https://www.gempy.org/tutorials for more details,

extent: defines the extent of the model in x, y and z directions, resolution: resolution of the model in all direction. If the extent in X direction goes from 0 to 100 m, a 20 resolution in x direction says that in every five meter there should be a vertice.

```
fr = np.array([[]])
mesh_rosultion = 5
n_iterations = 10
no_of_faults = 0
model = vmod.vertice_modifier(10, 0, all_vers, name_of_layers,
    mesh_rosultion, fr, extent, resolution)
new_result_list = model.contact_generator()[0]
length_layers = model.contact_generator()[1]
repre_pts = model.contact_generator()[2]
```

Outputs of the class

In case of having no Fault in the model (like the example of this notebook), only three parameters are important for us to be fed into GMSH:

- Points representing the layers' contacts (called as new_result_list in previous cell)
- Length of each layers' contacts (called as length_layers in previous cell). 3D visualization can help to see what we mean by this length.
- Some parts of the model that we are sure for example Formation A is there. This list is called repre_pts as an abbreviation for representative points. One of the advantages of GeoMeshPy is this list. Because by perturbing the model position of formations changes but GeoMeshPy can help to attribute Formations to their positions based on the results of GemPy. This list can help us in GMSH to call each volume of the model by its geological name which can be used later as physical properties in numerical solvers.

Outputs of the GMSH

Next block of code makes 10 different meshes based on the created geological model. For sure GMSH installation is required. Please have a look at the following links

https://gmsh.info/doc/texinfo/gmsh.html

https://gitlab.onelab.info/gmsh/gmsh/-/tree/master/tutorials/python

to get yourself familiarized with GMSH. The following cell is quite intense because several operations are included:

1. Creating a box with the same site as geological model
2. Importing all the vertices of the layers as points
3. Splitting vertices based on the length of each layers (as shown in 3D visualization cell).
4. Passing a surface through each layers or patches (refer to in faulted examples)
5. Importing wells
6. Cutting the created box based on the generated surfaces and embedding all surfaces, lines and points to their related higher dimensional entities.
7. Mapping each created volume to its related formation. Representative point are doing this correlation. At some point of the model we can find out with formation exist there and then we can call that volume with name of that point, i.e. the formations name.
8. A myriad of refinement approaches are available in Gmsh and some are also applied in the next cell to refine mesh close to well and contact of layers.
9. As much as iterations, mesh files will be generated at the end.

```python
import gmsh
import itertools
from collections import defaultdict
from itertools import chain
gmsh.initialize()
if no_of_faults == 0: # LIFE IS EASIER WITHOUT ANY FAULT :-)
    def cleanup_and_mesh(entities_to_preserve):
        # remove all embedded constraints, i.e. the entities that are not on the
        # boundary of others
        entities = gmsh.model.getEntities()
        for e in entities:
            emb = gmsh.model.mesh.getEmbedded(e[0], e[1])
            gmsh.model.mesh.removeEmbedded([e])
            for p in entities_to_preserve:
                if p in emb:
                    gmsh.model.mesh.embed(p[0], [p[1]], e[0], e[1])
        # remove all surfaces, curves and points that are not connected to any
        # higher-dimensional entities
        gmsh.model.removeEntities(gmsh.model.getEntities(2), True)
        cc = gmsh.model.getEntities(1)
        for c in curves_to_preserve:
            cc.remove(c)
        gmsh.model.removeEntities(cc, True)
        gmsh.model.removeEntities(gmsh.model.getEntities(0))

        # get all surfaces that are not of type "Plane", i.e. all surfaces except the
        # box
        surfaces = [s[1] for s in gmsh.model.getEntities(2) if gmsh.model.getType(s[0], s[1])
                    != 'Plane']
        layers_contacts = np.array(surfaces).reshape(-1,1).tolist()
        surface_after = gmsh.model.getEntities(2)
        points = copy.deepcopy(surface_new_tag)
        check_values = [row[-1] for row in surface_after]
        extracted = []
        for sublist in points:
            second_vals = [sec for fir, sec in sublist]
```

```
        if all(val in check_values for val in second_vals):
            extracted.append(second_vals)

    surrounding_box = [x for x in extracted if x not in layers_contacts]

    # create a distance + threshold mesh size field
    gmsh.model.mesh.field.add("Distance", 1) # refiement close to the well
    gmsh.model.mesh.field.setNumbers(1, "CurvesList", np.array(curves_to_preserve)[:,1].tolist())
    gmsh.model.mesh.field.setNumber(1, "Sampling", 100)
    gmsh.model.mesh.field.add("Threshold", 2)
    gmsh.model.mesh.field.setNumber(2, "InField", 1)
    gmsh.model.mesh.field.setNumber(2, "SizeMin", 2)
    gmsh.model.mesh.field.setNumber(2, "SizeMax", 20)
    gmsh.model.mesh.field.setNumber(2, "DistMin", 6)
    gmsh.model.mesh.field.setNumber(2, "DistMax", 30)

    gmsh.model.mesh.field.add("Distance", 3) # refiement close to contacts of layers
    gmsh.model.mesh.field.setNumbers(3, "SurfacesList", surfaces)
    gmsh.model.mesh.field.setNumber(3, "Sampling", 100)
    gmsh.model.mesh.field.add("Threshold", 4)
    gmsh.model.mesh.field.setNumber(4, "InField", 3)
    gmsh.model.mesh.field.setNumber(4, "SizeMin", 3)
    gmsh.model.mesh.field.setNumber(4, "SizeMax", 20)
    gmsh.model.mesh.field.setNumber(4, "DistMin", 10)
    gmsh.model.mesh.field.setNumber(4, "DistMax", 30)

    gmsh.model.mesh.field.add("Min", 5)
    gmsh.model.mesh.field.setNumbers(5, "FieldsList", [2, 4])
    gmsh.model.mesh.field.setAsBackgroundMesh(5)
    gmsh.option.setNumber("Mesh.MeshSizeMax", 30)
    # don't extend mesh sizes from boundaries and use new 3D algo
    gmsh.option.setNumber("Mesh.MeshSizeExtendFromBoundary", 0)
    gmsh.option.setNumber("Mesh.Algorithm3D", 10)
    gmsh.model.mesh.generate(3)
    # using representative points to create physical volumes
    rep = [list(x) for _,x in itertools.groupby(rep_pnt,lambda x:x[3])]
    vol_num = np.arange(1, 1+len(rep))
```

```
    for ind, surfaces in enumerate (rep):
        tags = []
        for sects in surfaces:
            eleTag = gmsh.model.mesh.getElementByCoordinates(float (sects[0]), float (sects[1]), float (sects[2]))[0]
            eleType, eleNodes, entDim, entTag = gmsh.model.mesh.getElement(eleTag)
            tags.append(entTag)
        gmsh.model.addPhysicalGroup(3, tags, vol_num[ind])
        gmsh.model.setPhysicalName(3, vol_num[ind], sects[-1])

    # adding wells as physical lines
    for lines, well_name in zip (sp_well, wl_names):
        l1 = gmsh.model.addPhysicalGroup(1, lines.tolist())
        gmsh.model.setPhysicalName(1, l1, well_name)

    # adding surrounding surfaces as physical surfaces to be useable as boundary conditions
    around_box = [’in’, ’out’, ’front’, ’back’, ’bottom’, ’top’]
    for tag_nu, name in zip (surrounding_box, around_box):
        ps1 = gmsh.model.addPhysicalGroup(2, tag_nu)
        gmsh.model.setPhysicalName(2, ps1, name)
    gmsh.write("Four_layers_" + str(kk) + ".msh")
    gmsh.fltk.run()
gmsh.initialize()
degree = 3
numPointsOnCurves = 10
numIter = 10
anisotropic = False
tol2d = 0.00001
tol3d = .1
tolAng = 1
tolCurv = 1
maxDegree = 3
maxSegments = 100
sets = zip(new_result_list, repre_pts, wells_cord, well_points)
for kk, (vertices, rep_pnt, well_cord, well_p) in enumerate(sets):
    ar = np.array(vertices)
    l_tags = []
    gmsh.model.occ.addBox(min(ar[:,0]),min(ar[:,1]),extent[-2],
```

```
max(ar[:,0])-min(ar[:,0]),max(ar[:,1])-min(ar[:,1]),extent[-1])
    spl_num = np.cumsum(length_layers[kk]).tolist()[:-1] # each layer is
separated
    spl_num = [int (i) for i in spl_num]
    sep_ar = np.split(ar,spl_num)
    for ind, point_clouds in enumerate (sep_ar):

point_clouds=point_clouds[np.lexsort((point_clouds[:,1],point_clouds[:,0]))]
        i_l = point_clouds.tolist()
        for [x, y, z] in i_l:
            gmsh.model.occ.addPoint(x, y, z)
        corn_layers= point_clouds [(point_clouds[:,0] ==
np.min(point_clouds[:,0])) |
                        (point_clouds[:,0] == np.max(point_clouds[:,0])) |
                        (point_clouds[:,1] == np.min(point_clouds[:,1])) |
                        (point_clouds[:,1] == np.max(point_clouds[:,1]))]
        corn_layers =
corn_layers[np.lexsort((corn_layers[:,1],corn_layers[:,0]))]
        h = corn_layers
        pnt = h[:,0:-1].tolist()
        arround_pts = model.rotational_sort(pnt,
(np.mean(np.array(pnt)[:,0]),np.mean(np.array(pnt)[:,1])),True)
        arround_pts = np.array(arround_pts)
        tags = np.where((point_clouds[:,:-1] ==
np.array(arround_pts)[:,None]).all(-1))[1]+1
        l_tags.append(len(tags))
        start_point = int (8+np.sum(length_layers[kk][0:ind]))
        start_line = int (13+np.sum(l_tags[0:ind]))
        for i in range (len(tags)): # this for loop creates the exterior
lines of each cloud
            if i!=len(tags)-1:
                gmsh.model.occ.addSpline([tags[i] +
start_point,tags[i+1] + start_point])
            else:
                gmsh.model.occ.addSpline([tags[i] + start_point,tags[0]
+ start_point])
        gmsh.model.occ.addCurveLoop([i for i in range (start_line,
start_line + len(tags))], start_line)
        gmsh.model.occ.addSurfaceFilling(start_line, start_line,
```

```
                                           [m for m in range(start_point+1, start_point+np.max(tags))
                                            if m not in tags + start_point],
                                           degree,
                                           numPointsOnCurves,
                                           numIter,
                                           anisotropic,
                                           tol2d,
                                           tol3d,
                                           tolAng,
                                           tolCurv,
                                           maxDegree,
                                           maxSegments) # create surface by connecting exterior lines
                                           # and inclding interior ones

    gmsh.model.occ.synchronize()

    in_surf = gmsh.model.occ.getEntities(2)

    # Importing Wells:

    tag_well = np.arange(10000, 10000+len(np.array(well_cord)))
    well_p = np.array(well_p[0]).astype('int')
    tag_well_po = np.split (tag_well, np.cumsum(well_p)[:-1])
    well_po = np.split (np.array(well_cord), np.cumsum(well_p)[:-1])
    for cord, tag_nu in zip (well_po, tag_well_po):
        for [x, y, z], num in zip (cord,tag_nu):
            gmsh.model.occ.addPoint(x, y, z, tag=num)
    well_l = well_p-1
    tag_w_l = np.arange(5000, 5000+np.sum(well_l))
    wl = [i.tolist() for i in tag_w_l]
    tag_well_l = np.split (tag_w_l, np.cumsum(well_l)[:-1])

    for po, tag_num in zip (tag_well_po, tag_well_l):
        for i in range (len(po)-1):
            gmsh.model.occ.addLine (po[i], po[int(i+1)], tag = tag_num[i])
```

```
        in_wells = [(1, i) for i in tag_w_l]
        out_all = gmsh.model.occ.fragment(in_surf+in_wells,
gmsh.model.occ.getEntities(3))#[1]
        out = out_all[1]
        surface_new_tag = out[0:len(in_surf)]
        c = out[len(in_surf):len(in_surf+in_wells)]
        curves_to_preserve = [item for sublist in c for item in sublist]
        gmsh.model.occ.synchronize()
        points_to_preserve = gmsh.model.getBoundary(curves_to_preserve,
combined=False)
        line_sp = np.array([])
        for i in range (len(points_to_preserve)-1):
            if i%2 != 0:
                if points_to_preserve[i][1] != points_to_preserve[i+1][1]:
                    brk=int ((i+1)/2)
                    line_sp=np.append(line_sp, brk)
        sp_well =
np.split(np.array(curves_to_preserve)[:,1],line_sp.astype('int'))
        cleanup_and_mesh(curves_to_preserve + points_to_preserve)
        gmsh.clear()
    gmsh.finalize()
```