Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Engineering
Department of Geoscience and Petroleum

Jalal Alali

Optimizing the Rate Balance Between Multiple CO2 Injectors for the Aurora Storage Site

Master's thesis in Petroleum Engineering
Supervisor: Carl Fredrik Berg
Co-supervisor: Alv-Arne Grimstad
July 2023

NTNU
Norwegian University of
Science and Technology

Jalal Alali

Optimizing the Rate Balance Between Multiple CO2 Injectors for the Aurora Storage Site

Master's thesis in Petroleum Engineering
Supervisor: Carl Fredrik Berg
Co-supervisor: Alv-Arne Grimstad
July 2023

Norwegian University of Science and Technology
Faculty of Engineering
Department of Geoscience and Petroleum

NTNU
Norwegian University of
Science and Technology

TPG4920 - Petroleum Engineering, Master's Thesis

Optimizing the Rate Balance Between Multiple CO_2 Injectors for the Aurora Storage Site

Jalal Alali

July 2023

To my family

whose unwavering support and encouragement have been the driving force behind my successful completion of this master's thesis.

Thank you.

Abstract

The global challenge of reducing greenhouse gas emissions has necessitated the development of effective carbon capture and storage (CCS) strategies. This thesis focuses on the optimization of CO_2 injection operation on a resized model of the *Aurora* storage site, aiming to enhance storage efficiency while minimizing leakage risks. Utilizing the `Flow` simulator, we investigate the balance between multiple CO_2 injectors and employ the `CO2STORE` module with the black-oil model to evaluate their performance. The results highlight the significance of effectively managing well control parameters and strategically positioning wells in order to optimize a CCS project whereas minimizing associated risks. By utilizing a `Python` coding methodology, a scenario cluster organizer script is developed to aid in building a comprehensive scenario evaluation. The results showcase the potential for maximizing storage efficiency while mitigating CO_2 leakage risks. The research outcomes provide valuable insights into the optimization of CO_2 injectors and contribute to advancing CCS technologies.

In Chapter 1, an incentive description is provided to clarify the imperative nature of CCS initiatives. Chapter 2 provides a concise overview of the historical context and benefits associated with CCS, alongside technical explanations pertaining to the injection of CO_2 into storage units and the mechanics involved in its retention. In Chapter 3, the materials employed in this study are explained, which serve to contribute to the exploration of the several issues addressed in the thesis. Chapter 4 provides a comprehensive overview of the methodologies employed and the underlying theoretical foundations supporting them. Chapter 5 presents the outcomes of many scenarios, accompanied by simple discussions for the reader's consideration in each section. Chapter 6 looks into an analysis of the limitations and dependability of the models and situations offered, as well as potential avenues for their improvement. Chapter 7 concludes the report by providing a comprehensive analysis of the entire project, including its underlying concept, the obtained outcomes, and possible spots for improvement in the future.

Contents

Abstract . . . . . **v**
Contents . . . . . **vii**
Figures . . . . . **xi**
Tables . . . . . **xv**
Code Listings . . . . . **xvii**
Nomenclature . . . . . **xix**
1 Introduction **1**
1.1 Motivation 1
1.2 Exploitation License 3
1.2.1 Field Exploitation License (*EL001*) 3
1.2.2 Well Exploitation Licenses 3
1.3 Hydrocarbon Emissions 5
1.4 Tax Effect On Tackling Hydrocarbon Emissions 6
1.5 Research Objectives 7
1.6 Research Outline 7
2 Background on Geological CO_2 Storage **9**
2.1 Injection Process 10
2.2 CO_2 Plume Migration 11
2.3 Trapping Mechanisms 11
2.4 CO_2 Properties 12
2.5 Storage Safety Parameters 14
2.6 Reservoir Geometry and Stratigraphy 15
2.7 Leakage Potential 18
2.8 Reservoir Simulation 19
2.8.1 The Aim of Reservoir Simulation 20
2.8.2 Classification of Models 20
2.8.3 Static Model 21
2.8.4 Dynamic Model 21
2.9 Simulation Tools 23
2.9.1 Geo-modeling Tools 23
2.9.2 Reservoir Simulators 23
2.9.3 Asset Simulators 23
2.10 Grids 24
2.10.1 Grids Classification 25

2.10.2 Regular Grids . . . 25
2.10.3 Cornerpoint Grids (Needs to be organized) . . . 26
2.11 Reservoir Simulators . . . 27
2.11.1 Thermal and Iso-thermal Simulators . . . 27
2.11.2 Compositional Simulators . . . 27
2.11.3 Black-Oil Simulator . . . 28
2.12 Reservoir Model . . . 29
2.12.1 Geo-Model . . . 30
2.12.2 PVT Model . . . 30
2.12.3 Grid Model . . . 30
2.12.4 Faults . . . 30
2.12.5 Uncertainty . . . 30
2.12.6 History Matching . . . 31
2.13 Building a Reservoir Model . . . 31
2.14 Effect Of The Penalty Cost On CO_2 Leakage Deterrence . . . 32
3 Material . . . 33
3.1 Software . . . 33
3.1.1 Simulator . . . 33
3.1.2 Visualization Interface . . . 34
3.2 Geological Model . . . 35
3.2.1 Sector Models . . . 37
3.2.2 Full Field Model . . . 38
3.2.3 PVT and Relative Permeability Data . . . 39
3.2.4 Permeability . . . 40
3.2.5 Grids . . . 41
3.2.6 Faults . . . 42
3.3 PVT Data . . . 43
3.3.1 Water PVT Data . . . 43
3.4 Reservoir Model Setup and Modifications . . . 44
3.4.1 Configuration . . . 44
3.4.2 Keywords Adjustment . . . 46
3.5 Database . . . 48
3.6 Coding . . . 48
3.7 Boundary Conditions . . . 49
4 Methodology . . . 53
4.1 Proposal for a Project Idea . . . 53
4.2 Penalty Area . . . 55
4.3 Injection and Monitoring Period . . . 55
4.4 Base Case Scenario . . . 56
4.5 Rate Balance Scenario . . . 57
4.6 Leakage Application Scenario . . . 59
4.7 Python Programming . . . 61
4.8 Calculations . . . 63
4.8.1 Dissolved Gas Mass Calculation . . . 63

4.8.2 Trapped Gas Mass Calculation 64
5 Results 65
5.1 Base Case Scenario 65
5.1.1 Plots 65
5.2 Rate Balance Scenario 75
5.2.1 Rate Balance Plot: 75
5.2.2 Selected Rate Balance Case 78
5.3 Penalty Leakage Scenario 87
5.3.1 Plots 88
6 Discussion 101
6.1 Data Analysis 101
6.1.1 CO_2 Storage Capacity 101
6.1.2 Pressure Distribution and Reservoir Management 102
6.2 Modeling Review 102
6.3 Scripts Review 102
6.4 NPV Coding Review 102
6.5 Parameters Analysis 103
7 Conclusion 107
7.1 Summary Of Findings 107
7.1.1 Importance of Reservoir Characterization 107
7.1.2 Optimization of CO_2 Injection Strategies 108
7.1.3 Estimation of Leakage To the Penalty Area 108
7.1.4 Integration of Technical and Economic Factors 108
7.2 Uncertainties 108
7.3 Future Work and Recommendations 109
7.4 Conclusion 109
Bibliography 111
A Code Scripts 115
A.1 OPM-Flow Base Case Input Deck 115
A.2 CO_2 PVT Interpolation Python Script 126
A.3 Water PVT Interpolation Python Script 127
A.4 NPV Python Script 128
B CO_2 PVT Data 147
B.1 Span-Wagner Thermodynamic Properties of CO_2 in The Standard Condition 147
B.2 Span-Wagner Thermodynamic Properties of CO_2 in T = 98 °C . . . 147
B.2.1 Table 147
B.2.2 Plot 148
C Equinor Well 31/5-7 Data 151
C.1 Well Drilling and Logging Data 151

Figures

1.1 CCS comprises the capture, transport and storage of CO_2 emissions 2
1.2 *EL001* License fact map 4
1.3 Location Stratigraphic Map of the Aurora exploitation license (*EL001*) 5

2.1 Overview of Geological Storage Options 10
2.2 CO_2 migration process and various trapping mechanisms 12
2.3 CO_2 pressure-temperature phase diagram 13
2.4 The density and viscosity of CO_2 at various temperatures and pressures in the liquid and supercritical phases. 14
2.5 Density of CO_2 Vs. Depth 14
2.6 Contribution of different trapping mechanisms. 15
2.7 Stratigraphic chart of *Triassic* to *Quaternary* deposits in the *Horda* platform. 17
2.8 *Dunlin* group stratigraphy 18
2.9 Flow chart of typical reservoir simulation workflow 20
2.10 Logical Cartesian grids 25
2.11 Cornerpoint grids scheme 26
2.12 Logical vs. cornerpoint grids 26
2.13 Reservoir modeling general intent flow diagram 31

3.1 Overview of the *Aurora* field geological model 37
3.2 Overview of the *Aurora* field geological model in `ResInsight`. . . . 38
3.3 *Aurora* full field model porosity distribution 39
3.4 *Aurora* full field model permeability distribution in x and y directions 40
3.5 Permeability representation in the *Eigestad* model of the *Aurora* field 41
3.6 Faults overview of the *Aurora* field. 42
3.7 Overview of the *Aurora* field sector model in `ResInsight` 45
3.8 `MULTPV` correction effect on the right boundary pore volume distribution. 50
3.9 `MULTPV` correction effect on the reservoir's top layer pore volume distribution. 51

4.1 Scenario cases database example. 61
4.2 Overview of the penalty area in `ResInsight` 62

5.1 Total injected mass of CO_2 (FGIT) in the base case model. 66
5.2 Maximum gas saturation (S_g) profile in overall the reservoir and the penalty area in the base case model. 67
5.3 Top side view of the gas saturation (S_g) in the reservoir in the base case scenario 68
5.4 West side view of the gas saturation (S_g) in the reservoir in the base case scenario 69
5.5 East side view of the gas saturation (S_g) in the reservoir in the base case scenario 69
5.6 Maximum gas dissolution rate in water (R_s) ratio profile in the base case model 70
5.7 West side view of gas solution ratio (R_s) in the reservoir in the base case scenario 70
5.8 East side view of gas solution ratio (R_s) in the reservoir in the base case scenario 71
5.9 CO_2 mass captured in the reservoir by different mechanisms in the base case model 71
5.10 CO_2 mass leaked to the penalty zone 72
5.11 Data accuracy test by comparing the data gathered by Python by different calculations 73
5.12 Pressure distribution in the reservoir top layer (below cap rock) in the base case scenario 74
5.13 Maximum pressure enforced to the cap rock by the reservoir top layer in the base case model. 75
5.14 Rate balance plot color spectrum definition 76
5.15 Well injection rate balance 2D plot for the BHP limit of 350 bar . . . 76
5.16 Well injection rate balance 2D plot desired trend 77
5.17 Selected case of the well injection rate balance scenario 77
5.18 Total injected mass of CO_2 (FGIT) in the selected rate balance model. 78
5.19 West side view of the gas saturation (S_g) in the reservoir in the selected rate balance scenario 79
5.20 West side view of the gas saturation (S_g) in the reservoir in the selected rate balance scenario 80
5.21 East side view of the gas saturation (S_g) in the reservoir in the selected rate balance scenario 81
5.22 Maximum gas saturation (S_g) profile in overall the reservoir and the penalty area in the selected rate balance model. 81
5.23 Maximum gas dissolution rate in water (R_s) profile in the selected rate balance model. 82
5.24 West side view of gas solution ratio (R_s) in the reservoir in the selected rate balance case scenario 83
5.25 East side view of gas solution ratio (R_s) in the reservoir in the selected rate balance case scenario 84

5.26 CO_2 mass captured in the reservoir by different mechanisms in the selected rate balance model. 84
5.27 CO_2 mass leaked to the penalty zone in the selected rate balance model. 85
5.28 Data accuracy test by comparing the data gathered by `Python` by different calculations 85
5.29 Pressure distribution in the reservoir top layer (below cap rock) in the selected rate balance case scenario 86
5.30 Maximum pressure enforced to the cap rock by the reservoir top layer in the leakage case model. 87
5.31 Total injected mass of CO_2 (`FGIT`) in the penalty leakage case model. 88
5.32 Top side view of the gas saturation (S_g) in the reservoir in the penalty leakage scenario 89
5.33 West side view of the gas saturation (S_g) in the reservoir in the penalty leakage scenario 90
5.34 East side view of the gas saturation (S_g) in the reservoir in the penalty leakage scenario 91
5.35 Maximum gas saturation (S_g) profile in overall the reservoir and the penalty area in the leakage case model. 91
5.36 Maximum gas dissolution rate in water (R_s) profile in the leakage case model. 92
5.37 West side view of gas solution ratio (R_s) in the reservoir in the penalty leakage scenario 93
5.38 East side view of gas solution ratio (R_s) in the reservoir in the penalty leakage scenario 93
5.39 CO_2 mass captured in the reservoir by different mechanisms in the leakage case model 94
5.40 CO_2 mass leaked to the penalty zone in the penalty leakage case model 95
5.41 Top side view of the gas saturation (S_g) in the penalty area in the penalty leakage scenario 97
5.42 Data accuracy test by comparing the data gathered by *Python* by different calculations 98
5.43 Maximum pressure enforced to the cap rock by the reservoir top layer in the penalty leakage case model. 98
5.44 Pressure distribution in the reservoir top layer (below cap rock) in the penalty leakage scenario 99

6.1 Well injection rate balance 2D plot for the BHP limit of 400 bar . . . 103

B.1 CO_2 properties in 98 °C condition 149

Tables

2.1 Reservoir Characteristics 16

4.1 Relocation scenario naming system. 61

B.1 CO_2 Properties in standard condition [15] 147
B.2 CO_2 Properties in 98 °C condition[15] 147

Code Listings

A.1 CO_2 PVT Data Interpolation Script 126
A.2 Water PVT Data Interpolation Script 127
A.3 NPV Code Script 128

Nomenclature

BHP	Bottom Hole Pressure
CCS	Carbon Capture and Storage
CO2STORE	The keyword that activates the CO_2 storage model for the run
DISGAS	The keyword activated the dissolved gas in the live oil model
E100	ECLIPSE100 Black-Oil Simulator
E300	ECLIPSE300 Compositional Simulator
EOR	Enhanced Oil Recovery
EoS	Equation of State
FieldOPT	Field Development Optimization Framework
Flow	OPM Flow
Gt	Giga Tonnes
IMPES	Implicit Pressure, Explicit Saturation
IPCC	United Nation Panel on Climate Change
km	Kilometer
Mt	Million Tonnes
m	Meter
NCS	Norwegian Continental Shelf
NPD	Norwegian Petroleum Directory
NPV	Net Present Value
PVT	Pressure, Volume, Temperature
THP	Tubing Head Pressure

VE	Vertical Equilibrium
CO_2	Carbon dioxide

Chapter 1

Introduction

The simulation models used in the current master's thesis study are based on a model developed by *Eigestad* [1]. In addition, it involves an alteration of the model in accordance with the `Flow` two-phase black-oil model. The adjustment is done by modifying the *'.DATA'* file using the `DISGAS` and `CO2STORE` modules, while considering the effect of temperature change. The primary goals of this study are to ascertain the most effective CO_2 injection rates through the utilization of rate balance theory, examine the distribution of pressure, assess the potential for leakage, and conduct various sensitivity analyses to evaluate the uncertainties in the final model. This section provides an introduction to the motivation, subject area, research objectives, and project's structure.

1.1 Motivation

Capturing, transporting, and storing CO_2 emissions from industrial output and fossil fuel combustion is essential for reducing global greenhouse gas emissions. CCS will be required to reduce global greenhouse gas emissions in order to meet climate targets at the lowest cost possible [2]. With the *Paris Climate Agreement* of 2015, all nations agreed to contribute to an immediate CO_2 reduction plan, recognizing this reality. According to the International Energy Agency (IEA), billions of tonnes of CO_2 must be stored annually in order to meet the *United Nations'* climate goals. Globally, CCUS facilities absorb nearly 45 Mt of CO_2 at present, but by 2030, this must increase to over 220 Mt. Approximately 35 commercial facilities that use CCUS for industrial processes, fuel transformation, and electricity generation can annually capture nearly 45 Mt of CO_2. Although CCUS adoption has historically lagged behind projections, momentum has recently increased, and there currently have 300 projects across the CCUS value chain in various stages of development.

Greenhouse gases (GHG) are chemicals that absorb heat, similar to greenhouse glass, and therefore contribute to climate change. As opposed to permitting heat to escape into space, the gases absorb solar energy and keep the heat close to the Earth's surface. This phenomenon of heat trapping is known as the greenhouse

effect. The most pervasive and detrimental greenhouse gases are carbon dioxide, methane, nitrous oxide, and a number of industrial gases [3]. Due to the combustion of fossil fuels for manufacturing processes and the generation of electricity, the greenhouse effect is increasing in the atmosphere and releasing a significant quantity of CO_2. Every sector of the global economy that utilizes fossil fuels, including manufacturing, agriculture, transportation, and power generation, contributes to greenhouse gas emissions. All the above-mentioned industries should transition away from using fossil fuels as a source of energy and toward pure, renewable energy sources in order to mitigate the most severe effects of climate change.

To attain the *Paris Agreement*'s goal of keeping the global mean temperature in 2100 below 2 degrees Celsius above pre-industrial levels, substantial deployment of carbon capture and storage (CCS) will be necessary [4]. When fossil fuels are used to generate electricity and for other industrial applications, carbon dioxide (CO_2) emissions are produced. Carbon capture and storage (CCS) technology can capture up to 90% of these emissions, prevent them from entering the atmosphere, and permanently store them in permeable, void reservoirs.

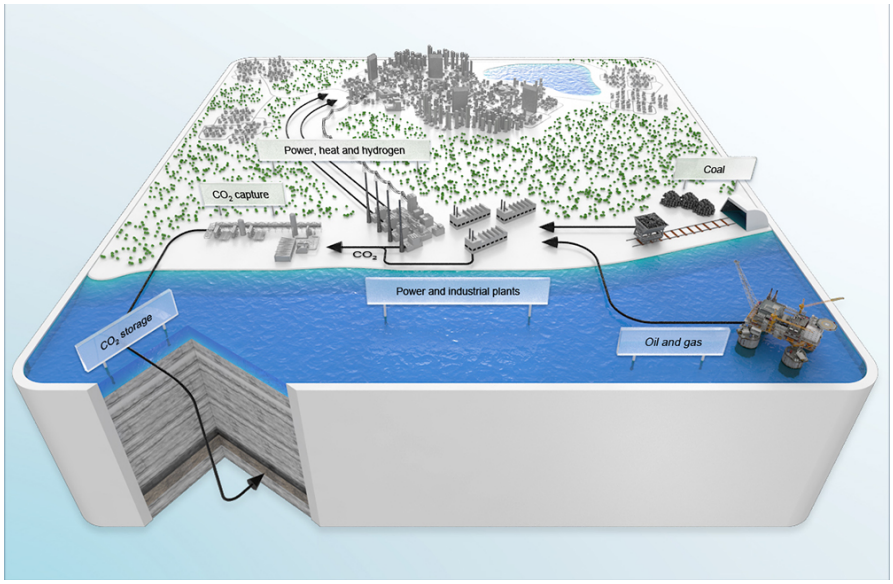


Figure 1.1: CCS comprises the capture, transport and storage of CO_2 emissions [3].

The *Sleipner* and *Snøvit* projects are two notable CCS projects that have been successfully implemented on the NCS for a period exceeding two decades. Norway possesses favorable circumstances that enable it to make significant contributions towards the advancement of CCS technology. The country possesses a strong technological community in the domain of carbon capture, transport, and storage. The Norwegian government has implemented a CCS initiative known as '*Longship*' or '*Langskip*' in Norwegian, drawing upon its rich and valued historical background.

The corporation known as the "*Northern Lights JV DA*" is a legally recognized and established *General Partnership with Shared Liability (DA)*. It is jointly owned by *Equinor*, *Shell*, and *Total Energies*, and operates as a component of the Longship project [3].

The comprehensive project encompasses the establishment of a versatile and accessible framework for the transportation of CO_2 from industrial capture sources located in the *Oslo Fjord* region, specifically from cement and waste-to-energy facilities. The liquid CO_2 will be shipped from these industrial capture sites to an onshore terminal situated on the western coast of Norway. Subsequently, the CO_2 will be transported via pipeline to a geologic reservoir for permanent storage.

1.2 Exploitation License

1.2.1 Field Exploitation License (*EL001*)

The *Aurora* storage site has been selected as the chosen location for this storage project. The Norwegian government has put out a proposal to use the *Lower Jurassic Dunlin Group* as a storage complex at the *Aurora* storage site for CO_2 storage purposes [5]. The storage units consist of the *Johansen* and *Cook* formations, which are sandstones comprising saline aquifers. The *Cook* formation is positioned just below the clay-rich *Drake* formation, which is well recognized as a regional seal for the storage units [6]. In January 2019, the Norwegian government granted an *Aurora* exploitation license to the *Northern Lights* project. The license is located within 30 km to the west of the CCS potential storage of *Smeaheia*, 15 km to the east of the active *Troll* west and east oil and gas fields, and directly to the south of these producing fields [7]. The eastern *Tusse* fault zone and the western *Svartalv* fault zone are both associated with basement structures and include the fault block that exhibits an eastward inclination, within which the *Aurora* exploitation license is located. The storage complex is intersected by several smaller faults at a lesser scale, which do not affect the basement [6][8].

There are at least four prospective storage sites with the following CO_2 capacities in the *Horda* platform study area, according to previous simulations of CO_2 storage and capacity estimates (sorted from lower to higher storage capacity) [4]:

1. ***Alpha*** structure, in the `Sognefjord` formation.
2. ***Aurora*** structure, in the *Johansen* formation.
3. ***Gamma*** structure, in the `Sognefjord` formation.
4. ***Troll*** field, `Sognefjord` formation.

1.2.2 Well Exploitation Licenses

EOS 31/5-7

The *Northern Lights* project drilled the *Well 31/5-7* confirmation well (`Eos`) between the *Troll* and *Brage* fields on the *Horda* platform in the *North Sea* from December

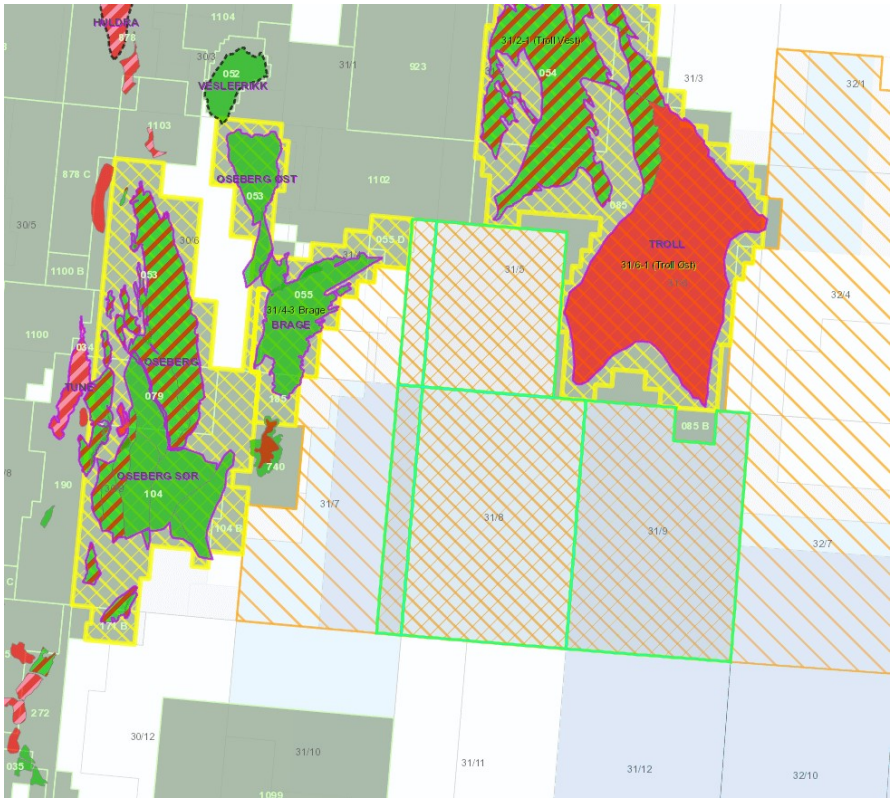


Figure 1.2: `EL001` license fact map [7].

2019 to March 2020 to verify the existence of acceptable storage and seal units under the *Aurora* exploitation license. The well is the first carbon capture and storage (CCS) well on the *Norwegian Continental Shelf (NCS)*. The overall objective of the well was to determine if *Early Jurassic* reservoir layers are suitable for carbon dioxide (CO_2) storage [9]. It is located about 10 km south of the line between the *Troll* License and the *Aurora* exploitation license (see figure 1.3).

EOS 31/8-1

On the *Breiflabb* prospect in the *Stord Basin*, south of the *Troll* field in the *North Sea*, *Well 31/8-1* was drilled. The primary objective was to demonstrate the presence of commercial quantities of hydrocarbons in the *Upper Jurassic Sognefjord* formation, which is also the primary reservoir in the *Troll* field [10]. Lower to *Middle Jurassic Johansen* formation, *Brent* group, *Krossfjord* formation, and *Fensfjord* formation are the deeper secondary targets for the well (see figure 1.3).

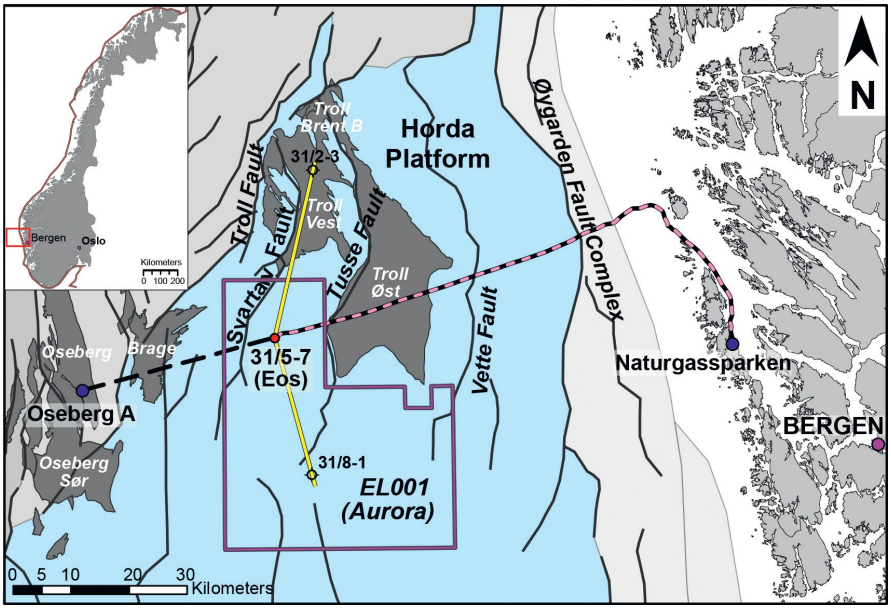


Figure 1.3: Exploitation Licence *EL001* and confirmation *well 31/5-7* (*Eos*) offshore the western coast of Norway are shown on a location map in relation to the major structural components. Route of the control umbilical and cable, in black stippled line. CO_2 pipeline path, pink stippled line. The *Horda* platform is shown in blue [8].

1.3 Hydrocarbon Emissions

Tackling emissions from hydrocarbon production is a necessity because hydrocarbon production will last for a prolonged time. The exploitation of hydrocarbon resources has severe environmental impacts, and emissions from hydrocarbon extraction activities may constitute a substantial portion of domestic emissions.

As a popular hydrocarbon production method, the waterflooding operation is energy-intensive and accounts for significant CO_2 emissions Our methodology is waterflooding optimization in reservoir simulation models, specifically optimizing well-controls. Unlike traditional studies, our optimization objective comprises two components: the profitability of hydrocarbon production and an additional tax proportional to CO_2 emissions. The associated CO_2 emissions are estimated using a scheme developed upon an integrated model of reservoir, surface network, and topside facility.

There is a non-linear relationship between reduced production and emissions. For increasing tax levels there are diminishing returns on lower emissions, reflecting reduced opportunities for emission reduction by changes in the drainage strategy. Some increments in the tax rate will therefore have negligible impacts on the optimal drainage strategy, and hence an adverse effect on the profitability with negligible emission reduction.

We are in the middle of an energy transition away from fossil fuels. In 2022,

oil accounted for 31.6% of the global energy supply, more than any other energy source. Natural gas contributed 22.8% of the global energy blend that year, making it the third-largest contributor. From 1990 to 2018, the proportions of oil and natural gas in the total energy supply have not changed significantly, and are likely to remain high in the near future. Despite a minor decline in 2020 due to the pandemic, the stated policies scenario defined by the *International Energy Agency* predicts that the global energy demand will continue to rise, surpassing 2019's energy demand by approximately 10% by 2030. The upward trend of the total energy demand suggests that substantial quantities of oil and natural gas will be produced in the years to come [11].

1.4 Tax Effect On Tackling Hydrocarbon Emissions

To support this emission reduction objective, it is crucial to develop and implement economically viable low-emission technologies for hydrocarbon extraction processes. Emissions have typically received little consideration in the optimization of hydrocarbon production, with the exception of a cost-based indirect penalty for emissions associated with leakage potential. Due to the rising CO_2 levy on emissions, the cost of emissions must now be considered in the optimization of hydrocarbon production. In this study, we will demonstrate how varying quantities of the CO_2 tax will result in a variety of optimal drainage strategies for regions with minimal infrastructure modification potential.

In order to depict production and injection systems more accurately, some optimization studies utilize integrated models that incorporate both surface and subsurface sections. The majority of CO_2 injection optimization studies seek to optimize capacity or injection rate (especially in subsurface reservoirs). Due to the absence of CO_2 emission estimates in these conventional optimization studies, the tax on CO_2 emissions will not be incorporated into the objective function.

Before going about evaluating the impact of a CO_2 tax on emissions from CO_2 injection processes, we need methodologies for calculating the CO_2 emissions associated with a particular injection strategy for CO_2 in the subsurface. Calculating the CO_2 emissions resulting from the CO_2 injection procedure can also be performed using empirical models. Various parameters, such as field injectivity, reservoir, leakage quantity, penalty cost, CO_2 tax rate, time, and other variables, can be utilized to construct such an empirical model.

However, The author noticed that there is a correlation between the CO_2 tax rate and emission density, with a higher CO_2 tax rate resulting in lower emission intensity. Instead, the free software `Flow` can be used to calculate the CO_2 emissions resulting from CO_2 injection into the prohibited zone.

This research presents a strategy for enhancing the CO_2 injection procedure that incorporates the emission cost into the optimization objective. This work optimizes the well control parameters, such as well rates and bottom-hole pressures, of a reservoir simulation model.

1.5 Research Objectives

The primary objective of this research is to examine innovative approaches for enhancing the efficiency of CO_2 injection operations in alignment with the desired well control parameters. The well control parameters may exhibit variability across different wells and fields. Hence, it is necessary to have a cohesive `Python` script and a customizable input deck in order to generate scenario clusters and a database in accordance with the field growth plan.

The utilization of project primitives has considerable importance in facilitating the adaptation and resolution of situations. As such, the project's secondary objective is to implement modifications to all processes in a manner that is easily adaptable and can be applied to other projects and datasets.

The project involved the implementation of injection rate optimization based on the BHP limit and the pressure exerted by the formation of rock fractures. Nevertheless, it is worth noting that the `Python` script (see appendix A.4) allows for convenient modification of all the well control methods. The *Aurora* storage site is used as an example due to the availability and accessibility of open-source genuine data, which aligns with the project's objectives.

1.6 Research Outline

This report contains 7 chapters. Motivation, the *Aurora* site, and research objectives are introduced in Chapter 1. In Chapter 2's executive summary, the essential components of CO_2 sequestration in saline aquifers are outlined. In Chapter 3, the project's materials and tools, as well as the reservoir simulation's results and processes, are described. Chapter 3 provides an overview of the model and details the *Aurora* base case model, rate balance model, and leakage simulator model's characteristics.

In Chapter 4, the methodology's performance, findings, and specifics are discussed. The results are presented in detail in Chapter 5. The primary focus of Chapter 6 is on discussions of the sources of concern and the uncertainties. This topic's conclusion and suggestions for future research are presented in Chapter 7. The A.1, A.4, and C appendixes contain the input deck, code scripts, and *Equinor Well 31/5-7* data, respectively.

Chapter 2

Background on Geological CO_2 Storage

Since roughly 85% of the energy used in industrial operations comes from the combustion of fossil fuels, carbon dioxide emissions that are frequently produced as a result of this process are the main cause of climate change. Following the extraction of coal from mines and the production of oil and gas from hydrocarbon reserves, these resources are transported to power plants where they are used to provide electricity and heat homes. The carbon dioxide must be separated from other gases, such as hydrogen, and then captured at the carbon capture facilities built near industrial and power plants. This will prevent as much carbon from escaping into the atmosphere as possible.

In order to stop greenhouse gases from escaping into the atmosphere, *Marchetti* [12] first proposed the concept of capturing carbon and storing it underground in 1977. Once the CO_2 has been separated from it using chemical solvents, the captured carbon is compressed to create a liquid that can be transferred to a suitable geological storage location and injected into subterranean formations [2]. Selecting the right storage location can make sure there are no major leaks for thousands of years. Power plants, refineries, steelworks, fertilizer plants, and cement plants are among the many types of plants that can use CCS.

When the Norwegian government enacted a carbon tax on CO_2 emissions from burning natural gas offshore in 1991, *Equinor* started conducting CO_2 storage tests [13]. The first CO_2 project was run by *Equinor* in the *Sleipner* field in Norway in 1996. It started to capture 1 Mt/year from the natural gas output in *Sleipner Vest* and store it in an aquifer known as *Utsira* formation that is more than 800 meters below the seabed. So far, there has not been a severe operational issue with the CO_2 storage at *Sleipner Vest*. Since the initial years of injection, the Utsira formation has been observed in order to research CO_2 circulation inside the reservoir.

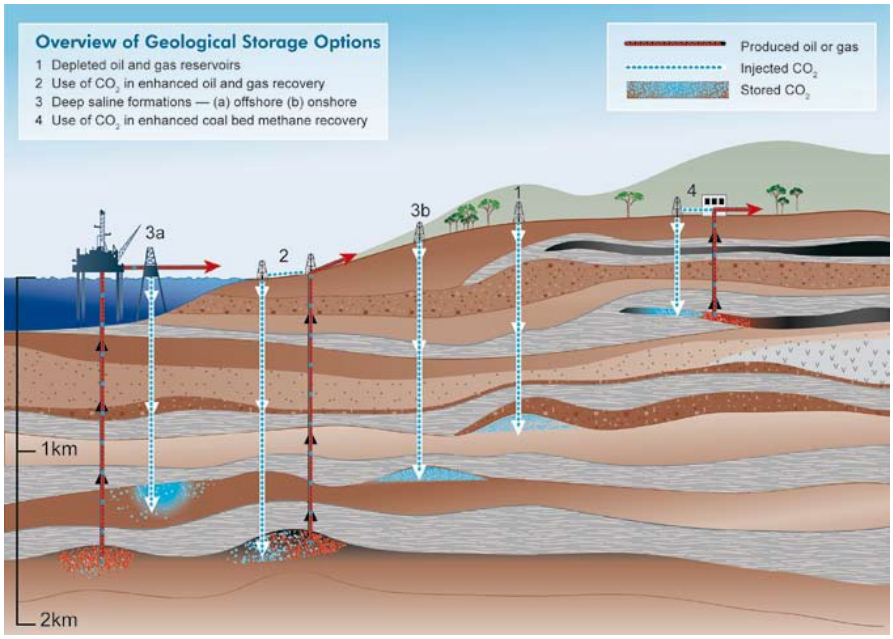


Figure 2.1: Overview of geological storage options (Courtesy of *CO2CRC*)[2].

2.1 Injection Process

When CO_2 is injected into a formation, the pressure gradient generated by the high injection builds internal viscous flow, which is the main driving force. This flow pushes away the in-situ brine and fills some of the pores near the injection well during the drainage process. Despite the fact that two distinct phases share a small proportion of pores, only a little amount of CO_2 will begin to dissolve into the brine, while only a smaller amount of brine will evaporate into the CO_2. This will lead to a limited exchange of mass between the two phases and Dissolution simply does not matter all that much at this point. This suggests that rather than having a uniform piston-like CO_2 front distribution, some front portions will flow significantly faster and form fingers inside the brine phase. Due to the spatial variation in permeability of the rock formation, the heterogeneity of the rock formation can also lead to fingering and channeling, which can channelize the injected CO_2 to permeable channels.

The injected CO_2 has a tendency to migrate vertically and cover the top of the formation until the sealing unit prevents vertical movement because of the considerable density differences between the phase fluids. The direction of the flow will also be significantly influenced by gravitational force. When CO_2 reaches the top of the formation, it must first overcome the capillary entry pressure of the sealing unit before flowing vertically again. The geometry of the plume is influenced by a number of factors, including the density differential between the CO_2 and brine, flow rate, and reservoir heterogeneity. A high fluid flow rate will raise viscous force close to the wellbore while a significant density differential will result in in-

creased vertical migration and lateral distribution on top of the formation, distant from the well. Since some water will always remain in the pores or be absorbed by the grain surfaces, injecting CO_2 won't be able to entirely remove the in-situ brine from the grain.

Two primary mechanisms play a significant part in supplying the extra space needed to store the CO_2 in the formation, with less concern about the limit exceeding pressure elevation, which can lead to shattering the sealing unit. The slow migration of the brine through the caprock, into adjacent formations, or across boundaries is the first mechanism. The second process is compressibility. As the pressure inside the reservoir rises as a result of the ongoing CO_2 injection, more fluid is allowed to fill the reservoir. This is due to the CO_2 and brine's starting to get slightly heavier.

2.2 CO_2 Plume Migration

Following the end of CO_2 injection period and ensuing shutdown of the well, the pressure accumulated in the formation progressively diminishes. The upper portion of CO_2 undergoes expansion in a radial conduct, forming a slender plume beneath the caprock. As the fluid phases within the porous media approach equilibrium, the influence of viscous forces diminishes slightly away from the well, giving way to advection. This transition is facilitated by the combined effects of gravitational and capillary forces.

The movement of the CO_2 plume will be influenced by the in-situ brine as it ascends in the imbibition cycle, driven by gravitational attraction. This process is the opposite of drainage. The imbibition mechanism's constraints, specifically the irreducible water saturation, result in the retention of a portion of CO_2 within the pores due to capillary forces. This phenomena is commonly referred to as residual trapping.

As the plume undergoes lateral movement, the interface between CO_2 and brine will experience increased extension. Consequently, the brine will initiate the dissolution of CO_2, a phenomenon referred to as dissolution trapping. This process continues until the brine becomes enriched with CO_2 rendering it denser than the undersaturated brine. Subsequently, the CO_2-rich brine descends towards the lower regions of the formation through a process known as convective-mixing dissolution.

2.3 Trapping Mechanisms

Several mechanisms could regulate the flow rate of the injected fluid once CO_2 was injected into the subsurface and prevent CO_2 leakage. Some of these mechanisms involved gradually immobilizing the CO_2, while others involved wrapping it in tight layers to prevent leakage into the sea and eventually the atmosphere. From

a conceptual standpoint, the containment of CO_2 at the geological storage site is divided into two categories: physical trapping and chemical trapping.

Physical Trapping

Physical trapping includes 3 mechanisms of trapping:

- Stratigraphic and structural trapping
- Hydrodynamic trapping
- Residual trapping

Geochemical Trapping

On the other side, geochemical trapping is also sub-categorized into two different types:

- CO_2 dissolution
- Mineral trapping (mineralization)

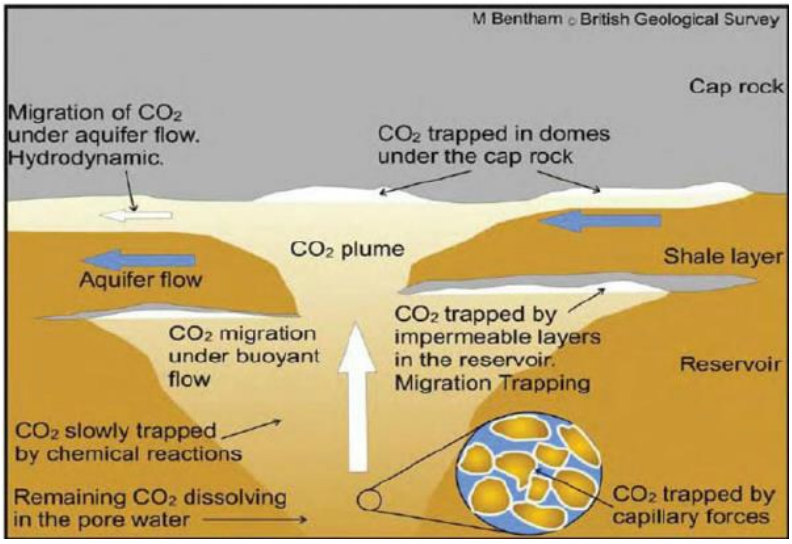


Figure 2.2: CO_2 migration process and various trapping mechanisms in the reservoir [14].

2.4 CO_2 Properties

CO_2 is a gas that is thermodynamically stable, reasonably compressible, and slightly denser than air with a density of 1.87 kg/m^3 at atmospheric pressure. CO_2 first forms a liquid phase in deep saline aquifer reservoirs after reaching a critical point at T = 30.98 °C and P = 73.8 bara, where pressure and temperature are both high. Around this time, CO_2 also forms a super-critical phase, with a density that can

range from 150 to over 800 kg/m^3. It has been mentioned even before CO_2 injection in a dense phase, such as liquid or supercritical, is more effective than CO_2 in its gas phase since it requires considerably less space, allowing the operators to inject more CO_2 into the formation.

CO_2 density and FVF is correlated according to *Span-Wagner*[15] CO_2 PVT data which is interpolated for every single cell in their specific reservoir condition (P, T). CO_2 density in the standard condition is considered to be equal to 1.871849 kg/m^3 [15] (see appendix B.1). In Span-Wagner[15] thermodynamic properties of carbon dioxide (CO_2) are presented according to pressure and temperature, in a manner that every table has a constant pressure in MPa, and all the thermodynamic properties are presented according to a range of temperatures in K (Kelvin). Since the measuring units in this project are Metric units, the pressure and temperature units are converted into psi and °C and presented the data in a range that is valid referring to the reservoir condition in *Johansen* formation. All the collected data are presented in appendix B.2 and collected to a Python script to do the interpolation for each unique cell in different time steps and PVT conditions (see appendix A.2).

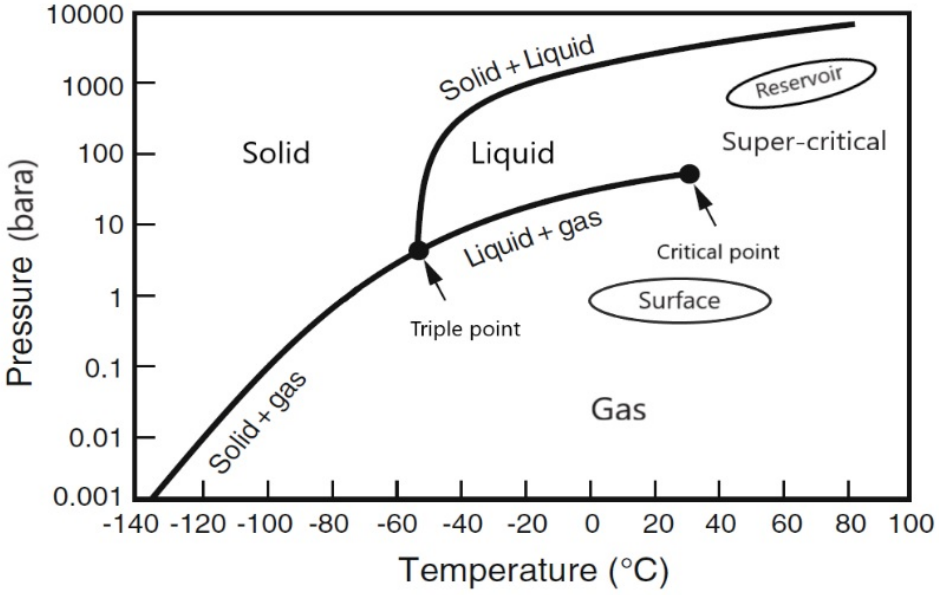
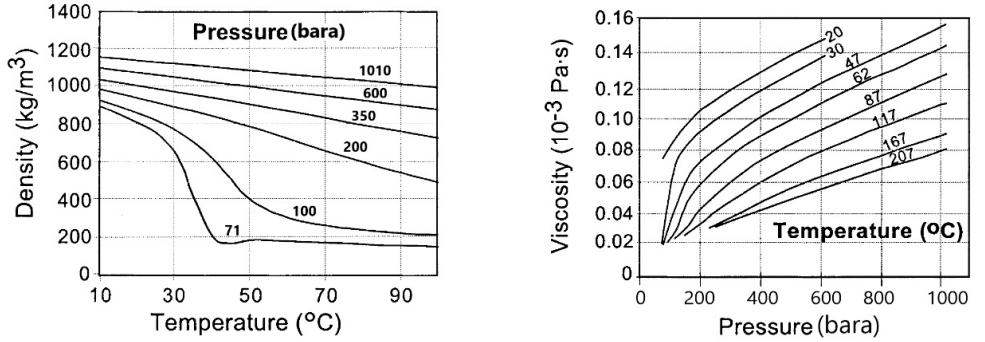


Figure 2.3: CO_2 pressure-temperature phase diagram [16].

CO_2 is in the gaseous phase at standard conditions, which means it has a low density and viscosity and is very compressible. Depth is essential for increasing site safety as there is a higher likelihood of sealing units such as shales, faults, and salt units the injection location at depths of 800 m or more.



(a) Density, Pressure vs. Temperature.

(b) Viscosity, Pressure vs. Temperature.

Figure 2.4: The density and viscosity of CO_2 at various temperatures and pressures in the liquid and supercritical phases. (a) Liquid and super-critical CO_2 density-pressure and temperature relationship (b) Liquid and supercritical CO_2 viscosity-pressure and temperature relationship [16].

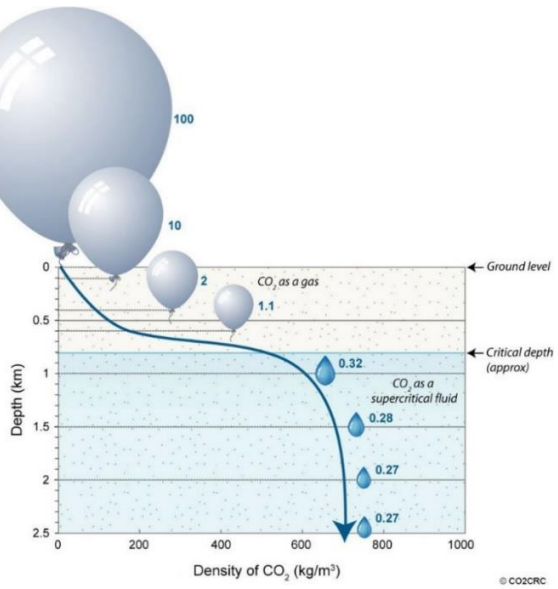


Figure 2.5: Density of CO_2 Vs. Depth (*CO2RC*)[2].

2.5 Storage Safety Parameters

The CO_2 is kept in the ground by a number of trapping mechanisms that begin operating as soon as the injection procedure begins. Over time, the storage becomes more secure if there are no leaks during injection or after injection. Despite all the debates and disputes about the specific contributions of each trapping mechanism at different time scales, a general principle is accepted by all parties involved in this topic (figure 2.6). In the first ten years, structural and stratigraphical trapping will be the key contributors because of an upper sealing unit above the injection

formation. While residual trapping is projected to gradually contribute after a decade to a century, dissolution trapping is often less effective during the first century, with significant contributions beyond that. Mineralization will gradually begin to emerge after a century with little influence, but after many centuries, it will significantly contribute. All physical and chemical trapping methods must be utilized to ensure secure storage.

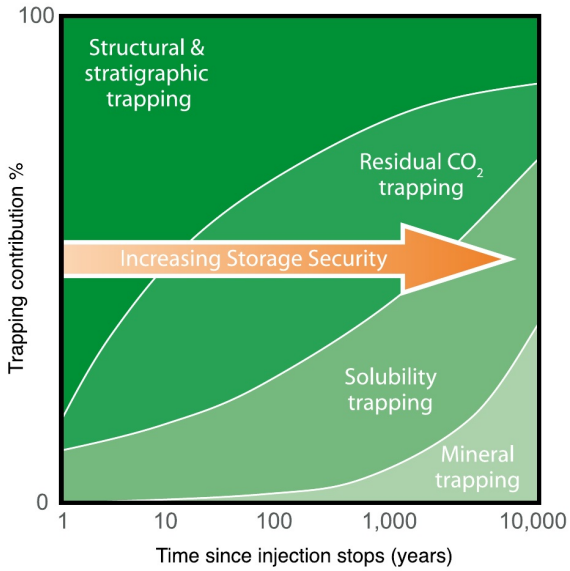


Figure 2.6: Projected contribution of different trapping mechanisms over 10,000 years [2].

2.6 Reservoir Geometry and Stratigraphy

There will be an explanation of the wells used in the model's construction, the geology of the reservoir and sealing units, the petrophysical characteristics, the fluid characteristics, and changes made to the default model to produce a new base case model. The benchmark model for this study was amended *Eigestad*'s model [17] for the *Johansen* formation based on these investigations, data gathering, new mineralogical samples, and re-interpolation of wells.

The *Dunlin* group of geological layers, which includes the *Early Jurassic Johansen* and *Cook* formations. The *Johansen* formation contains numerous siltstone and mudstone interlayers with low porosity values that are connected to flooding occurrences [18]. Within the *Johansen* formation, frequent calcite-cemented sandstones (carbonate layers) have been identified, most of which are less than one meter thick. The *Well 31/5-7* confirmation well verified the *Johansen* formation's 116 m thickness with more than 2000 m depth. At numerous different stratigraphic stages, sandstone tongues interfinger with the *Drake* mudstones to dominate the *Early Jurassic Cook* formation [19]. The Horda platform's *Cook* form-

ation, which is composed of clear sandstones with thin hetero-lithic intervals below them and stratigraphically overlies the *Johansen* formation, is divided from the *Johansen* formation in some places and from the shaly *Amundsen* formation in others [4]. The *Johansen* and *Cook* formations are separated by the *Amundsen* formation, which is made up of lateral marine silts and mudstones that were deposited on a shallow maritime shelf.

The lower *Drake* formation shale thickness is 75 m, and an extended leak-off test confirmed the sealing potential for future CO_2 injection. The underlying injection and storage units, *Johansen*, and *Cook* formations consist of high-quality sandstones with a total thickness of 173 meters. Formation pressure data acquired on wireline show that the rocks over and under the *Drake* shale cap rock are not in communication and the pressure gradients indicate water in all *Viking* and *Brent Group* sands.

Variable	**Value**
Reservoir	*Johansen*, Cook Fms
Reservoir Thickness (m)	173
Slope	2°N-S
Porosity (%)	7.3 - 31.4
Horizontal Permeability (mD)	0.1 - 500
Vertical to Horizontal Permeability Ratio	0.1
Well1 (31/5-7) Location (i, j)	(109,92)
Well1 (31/5-7) Depth (m)	2700
Well2 (31/8-1) Location (i, j)	(106,106)
Well2 (31/8-1) Depth (m)	2700
Perforation Status	Partially Perforated
Injection Rate $\left(\frac{Mt}{Year}\right)$	Various
Injection Period (Years)	30
Pore Volume $\left(Gm^3\right)$	75
Grid Type	Cornerpoint
Grid Dimension	(149 X 189 X 16)
No. of Active Grids	17,782
No. of Faults	13
Fluid in Place Region	1
Initial Pressure	260 bar at 2600 m
Temperature	97 C
Water Density at SC $\left(\frac{KG}{m^3}\right)$	1000
CO_2 Density at SC $\left(\frac{KG}{m^3}\right)$	1.871849
Salinity $\left(10^{-3}\frac{Kg/M}{Kg}\right)$	1.1
Residual Gas Saturation	0.254
Irreducible Water Saturation	0.15
Simulation Start Date	January 2025
Simulation End Date	January 2500
Simulator	`OPM-Flow`

Table 2.1: Reservoir Characteristics

Dunlin Group

The group mostly consists of dark to black argillaceous sea deposits, but on the basin's margins, marine sandstones are well-developed at a number of stratigraphic levels and can penetrate deeply. Gamma-ray log breaks delineate the lower boundary with the *Statfjord* group and the top boundary with the *Brent* group. There are 5 formations in the *Dunlin* group and the ages of this group, which go by the names of the *Amundsen* (base), *Johansen*, *Burton*, *Cook*, and *Drake* formations, vary from the *Hettangian* to the *Bajocian* [20].

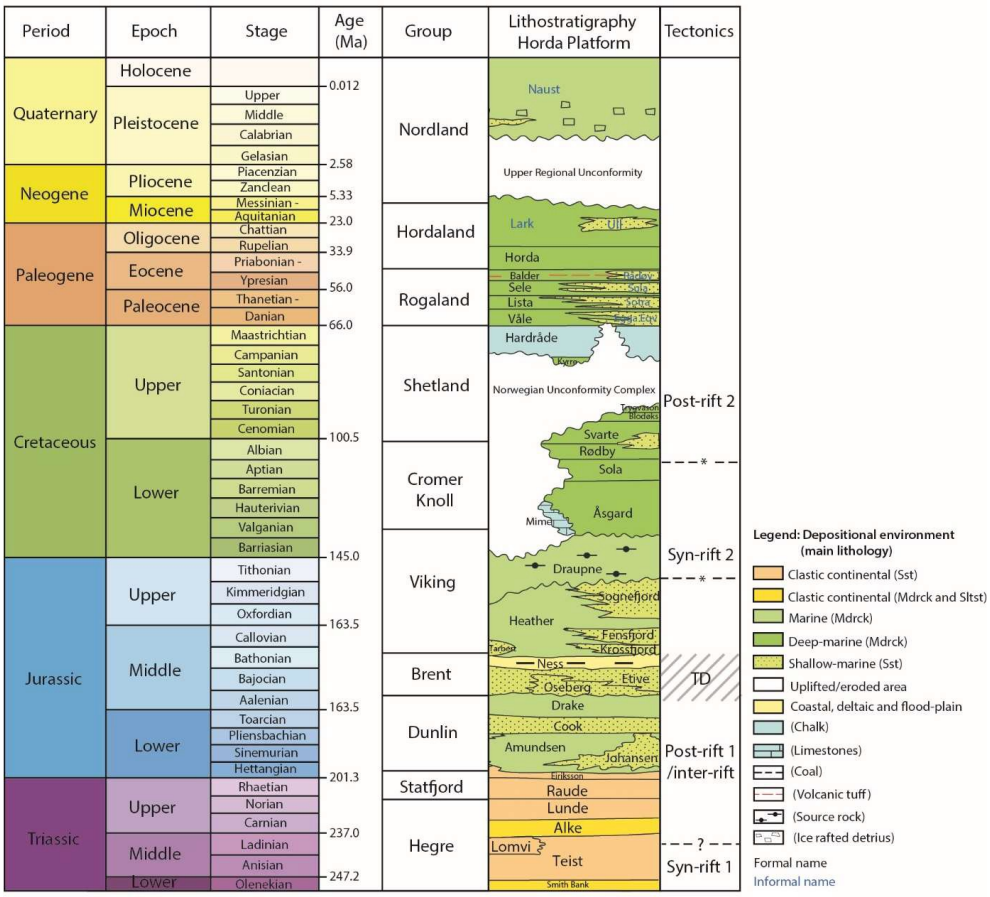


Figure 2.7: Stratigraphic chart of *Triassic* to *Quaternary* deposits in the `Horda` platform [2].

Johansen Formation

The formation was most likely deposited on a shallow sea shelf with strong energy. The formation is made up of a series of sandstones with thin, calcite-cemented streaks running through them. The *Johansen* formation divides the *Amundsen* formation area. Medium to fine-grained, micaceous, well-sorted sandstone makes

up the lower section before descending into light grey silty micaceous claystone. Medium-grained, friable sandstones with well-sorted quartz grains that are angular to sub-rounded make up the majority of the formation's main portion. Medium to fine-grained, micaceous, silty, argillaceous, and fairly sorted sandstones make up the highest portion. The formation is only found in the region running north from the *Måløy* fault blocks to the eastern portion of the *Horda* platform.

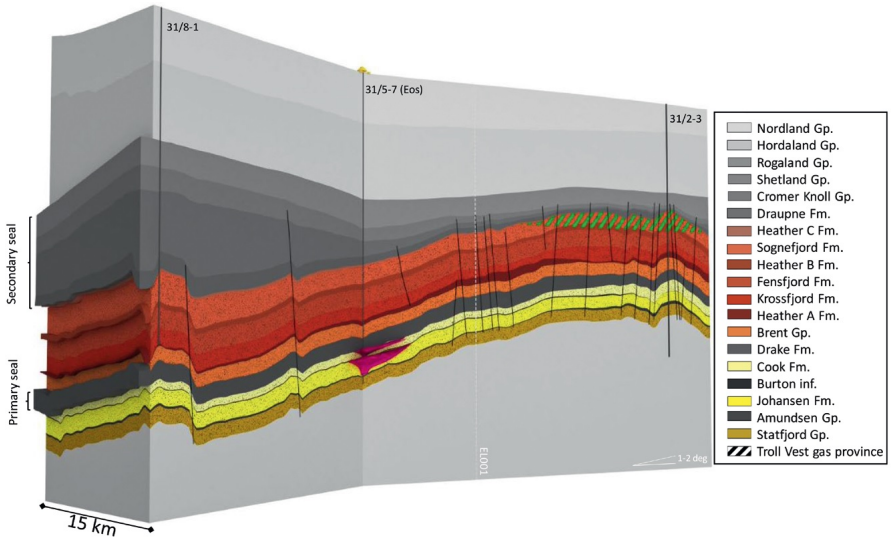


Figure 2.8: A rendering of the storage complex showing the primary (*Drake* formation) and secondary (*Draupne* formation, *Cromer Knoll* group, Shetland group, and *Rogaland* group) seals in grey and the storage units (*Cook* and *Johansen* formations) in yellow. Through a side-track of the *Well 31/5-7* and *Well 31/8-1* CO_2 will be pumped into the *Johansen* formation. The CO_2 is expected to move laterally inside both formations, below the *Drake* formation primary seal and the informal *Burton* formation barrier, respectively, as it migrates upward from *Johansen* formation to *Cook* formation [8].

2.7 Leakage Potential

Leakage Into Active (Productive) Fields

As it has been mentioned earlier in section 1.2 More specifically, the license is situated about 30 km west of the *Smeaheia* CCS possibilities, 15 km east of the producing *Troll* West and East oil and gas fields, and just south of the producing *Troll* West and East oil and gas fields[7] as it can bee observed in figure 1.2.

The *Svartalv* Fault Zone to the west both involves basements and surrounds the eastward slanted fault block where the *Aurora* exploitation license (section 1.2) is located. The *Svartalv* fault zone lies on the boundary of *Aurora* structure and *Troll* which is one of the biggest current gas producers in *NCS*. Any leakage from

Aurora reservoirs toward *Troll* gas field would interfere with the gas production operation and also there would be a possibility of gas leakage into *Aurora* aquifer.

Leakage Into Surface or Adjacent Reservoirs

Leakage into top layers and adjacent reservoir(s) is possible through faults and fractures which are already presented in the structure or would be established due to the injection operation. Thermal fractures are inevitable and the fractures due to pressure should be observed and monitored by reservoir simulation in beforehand and by geological monitoring methods afterward. Reservoir safety and sealing system is the most important aspect of the CCS discussions topic among the societies in order to accept this new valuable method of decarbonization. Therefore, there is special attention to this issue in this project and the credit of *Northern lights* project would be highly maintained by ensuring storage safety as a big-scale lead CCS project. As mentioned above, the leakage could happen through below mechanisms:

- Leakage through cap rock
- Leakage through existing faults
- Leakage through pre-made or new fissures and fractures

2.8 Reservoir Simulation

The solution of issues and queries relating to CO2 storage in the deep subsurface requires the use of mathematical models and numerical simulators. They are essential for elucidating the safety, viability, and financial hardship. Theoretical models will probably also be necessary for the development of a legal framework that enables the wide applicability of carbon capture and storage (CCS) technology.

Reservoir simulation is a method for simulating the behavior of subsurface reservoirs by numerically resolving fluid flow equations, such as those involving the conservation of mass, momentum, and energy, and equations of state (EoS) that describe how fluid phase behavior varies with temperature and pressure (PVT).

Reservoir modeling's goal is to provide quantitative answers to questions regarding reservoir development and management by integrating as much of our collected qualitative and quantitative data about the reservoir as possible. A reservoir model is the final outcome of reservoir modeling. These virtual models are actually referred to as model realizations. They can be represented as static geomodel realizations or dynamic simulation-model realizations, the latter of which we will refer to as simply reservoir models in the context of reservoir simulation.

2.8.1 The Aim of Reservoir Simulation

Since the use of reservoir simulation is linked to capital expenditure and operating costs, it is proportional to the size and complexity of the reservoir. While big offshore hydrocarbon reservoirs will have a professional staff of reservoir engineers mainly focusing on a single field, small onshore reservoirs may not use reservoir simulation at all. After a field is discovered, geologists and reservoir engineers start working on it to see whether it can be developed economically. The next step is to assess the cost, risk, and production potential of various development strategies. Prior to conducting any reservoir simulation study, the study's objectives must be well specified.

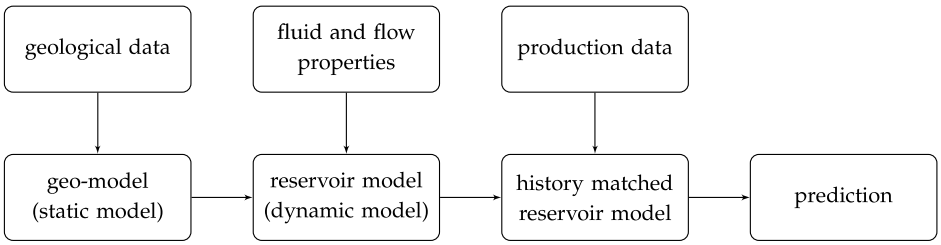


Figure 2.9: Flow chart of typical reservoir simulation workflow [21].

from the reservoir modeling study, and they serve as a decision-making tool for reservoir management.

2.8.2 Classification of Models

There are two methods to classify reservoir models, first by their scope, or the underlying properties they cover, and the second by the issues they are intended to address [21]. The proposed objective could be:

- Field development
- Asset optimization
- Long-term production optimization
- Selection of in-fill well targets
- Well planning and Geo-steering
- Tertiary recovery
- Data interpretation

The fluids in the reservoir will start to move in the opposite direction as the pressure drops when a pressure gradient is established. For a given pressure gradient, more fluid will be transferred with a higher permeability (and lower viscosity). This indicates a higher rate of fluid production or even that injecting fluid into the reservoir is easier. Thus, high permeability is generally always preferred. Pollutant storage is one exception, as limited permeability prevents the spread of the contaminant and provides safer storage.

Fluids are injected into the subsurface reservoir during the CO_2 sequestration process and CO_2 is usually at supercritical conditions and has a lower density than the displaced brine at the pressure and temperature found in the subsurface. Due to gravity, this density differential causes upward migration. The upward migration needs to be avoided, for example by injecting CO_2 under an impermeable barrier, as carbon dioxide should be kept for the foreseeable future. Hence, suitable reservoirs for the storage of carbon dioxide often match petroleum reservoirs.

Even though the process for the migration of hydrocarbons into reservoirs from source rocks is similar to what occurs with carbon dioxide (CO_2) storage, the distance between the reservoir and the source rock is typically much greater than the distance between the CO_2 injection well and the trap for the CO_2 that is migrating upward. So, the basin scale, which is much larger than the reservoirs from which hydrocarbons are produced, is filled by the migration of hydrocarbons.

In comparison to the time frame for the resource's production, the buildup of a resource also takes place over a long period of time. On the other hand, in order to ensure safe storage for CO_2 sequestration, models on different timescales far longer than the period of CO_2 injection are needed, as well as simulations of the period following the injection period.

Fluids will begin to flow into the well if the reservoir pressure is greater than the hydrostatic pressure of the fluids inside the well. In this case, a scenario database is designed according to the need for well placement redesign and having leakage into the penalty area to observe the *Python* code function and report in order to define a reasonable penalty value settlement and debug the procedure.

2.8.3 Static Model

The geological characteristics that do not change during production are included in the reservoir static model. The notion that a static model is a constant need not be totally accurate; for instance, the pore volume may be allowed to change as a result of compaction. A framework model in the static model describes faults and horizons. an object model that illustrates how geological objects are distributed geographically. a spatial pattern of properties is represented by a property model. An initial fluid interfaces model will be included in the static model and the fault seal and compartment communication models must be consistent with this contact model.

The static model must also include a model for capillary pressure and a, possibly simplified, fluid model in order to be able to compute initioriginal in-place hydrocarbon volumes. However, the dynamic model is typically considered along with the fluid model as well as models for capillary pressure and relative permeability.

2.8.4 Dynamic Model

dynamic models have the extra elements required to simulate fluid flow in operation. The fluid model in the dynamic model describes the characteristics and

spatial distribution of the reservoir fluids (PVT regions). It is also necessary to develop a model for the spatial distribution of rock types, relative permeability, and capillary pressure functions (saturation-function regions, keyword `SATNUM` in `Flow`).

To properly anticipate production from a reservoir with multiple wells, the production strategy must be represented in a robust manner that the reservoir simulator can understand. The well model explains the coupling between the reservoir and the wells as well as the flow in the wells. As our knowledge of what lies beneath the subsurface is never sufficient, a single realization cannot provide the range of uncertainty for results required to make wise decisions regarding reservoir management. An ensemble of statistically representative realizations is required to stretch out the uncertainty, and as a result, the actual reservoir model consists of a set of parameterized recipes for developing model realizations.

While multiple models are required to cover various concerns, each model should be built on a similar set of ideas and information that is kept in a knowledge database. Although this database has the potential to be a full knowledge management system, most businesses actually use a combination of databases (usually for measured data) and less formal storage formats. Building trust in the results and outputs of numerical simulators and mathematical models is therefore vital. It is crucial to comprehend how sensitive various model concepts and numerical methods are to a specific set of concerns. A practical way might be to look at the range of model predictions for a number of benchmark cases where specific descriptions of model domains, boundary conditions, etc. are provided since it is not possible to completely test models for CO_2 storage through well-controlled assessments.

The major objective of reservoir simulation is to direct the management of the subsurface reservoir through future forecasting and prediction, preventing resource waste. There are many subsurface optimization problems involving forecasting whether one is attempting to maximize net present value during the extraction of oil from a hydrocarbon reservoir, optimize freshwater production from an aquifer, or detect the optimum well placement for CO_2 sequestration into a deep subsurface aquifer, reservoir simulation could provide a prescriptive input into the optimization.

The part of the porous material where a resource accumulates is typically referred to as a reservoir. For example, under a dome structure, where hydrocarbon accumulates during its gravity-driven upward movement, or in a subsurface structure, where precipitation is captured during its gravity-driven downward movement. In fact, we are contaminating or storing unwanted material when we keep an eye on potentially dangerous fluids in soil or plan carbon dioxide sequestration. The name "reservoir" will still be used, but it will now refer to the area of porous material that is essential to the method, such as the area of the subsurface that will be used for storage.

2.9 Simulation Tools

2.9.1 Geo-modeling Tools

Static model developers, and geo-modeling software like `PETREL` and `RMS` can include modules for everything from seismic interpretation to creating reservoir simulation model grids with the right attributes.

2.9.2 Reservoir Simulators

The dynamic behavior of a reservoir to proposed drainage techniques and interventions is assessed using reservoir simulators, such as `OPM-Flow`, `ECLIPSE`, `Intersect`, `Tempest MORE`, and `tNavigator`.

2.9.3 Asset Simulators

Whereas asset simulators including `Pipe-It` are used to optimize the systems of multiple reservoirs connected to a single top-side infrastructure and component.

Field Development

The number of wells and their placement, subsea vs. platform, and the capacities of subsea and top-side equipment like separators are the primary issues that need to be resolved in field development. In most cases, a full-field reservoir simulation model is the primary tool for solving those issues.

Asset Optimization

The objective of asset optimization in a reservoir modeling scenario is to optimize top-side infrastructure and equipment that couple a number of fields, both in terms of which equipment is to be installed and their capacities, as well as in terms of how the production of each field should be prioritized.

Long-term Production Optimization

The fluid and pressure distribution near the well bores is mostly affected by short-term production optimization methods like rapidly changing production rates to reduce gas production. This often occurs on a scale that is far smaller in both space and time than what reservoir simulation models can capture. However, long-term production optimization sets the parameters within which short-term optimization operates by setting objectives and limitations on the operation of individual wells and groups of wells.

Selection of In-fill Well Targets

Diagrams of the remaining in-place resources and reservoir simulation are used to identify targets for in-fill wells, which are drilled to substitute or add to existing wells with deteriorating production.

Well Planning and Geo-steering

Using *Logging While Drilling* (LWD) data to control the well path while drilling is known as geo-steering. For this technology to be applied successfully, it is essential to have comprehensive, continuously adaptable near-well models with major factors connected to the logging tools, such as resistivity.

Tertiary Recovery

Tertiary recovery techniques frequently have an impact on reservoir flow in ways that are difficult for conventional reservoir scale simulation models to adequately capture. In these situations, it is necessary to use models with a fine-grid representation of the geology and often, specific reservoir simulators in order to assess the potential effects of employing the tertiary recovery method.

Data Interpretation

Reservoir models that cover the volumes influenced by a good test are used both for planning the test and for well-test interpretation.

2.10 Grids

Grid generation is the initial step for numerical modeling. A grid (or a mesh) is a division of a geometric domain into (simple) subobjects and is a common technique for numerical methods in many sciences. In our case, the geometric domain is the reservoir. The sub-objects are defined as cells. grid cells for control volumes, as conservation laws can be applied to these discrete volumes which are covering the geometric domain. Another term in common use is grid blocks.

A grid might refer to a discretization of the space consisting of both the time and spatial domain. However, in reservoir simulation, a grid usually refers to the spatial domain only. In reservoir simulation, we thus use grids to refer to the tessellation of the geometric domain consisting of the subsurface reservoir. The simulation progress through time in steps, and the time step lengths are determined based on the dynamic state of the simulation and not set in advance on a time grid.

The main geological features of a reservoir are vertical changes in lithology (rock types), which gives rise to horizontal surfaces, and faults, which give rise to vertical (or inclined) surfaces. These two main geological features guide the

division of the reservoir into grid cells. There is usually a small part of a larger connected porous medium that is of primary interest for our simulations. For hydrocarbon reservoirs, we are mostly interested in the part containing hydrocarbons, but this part is interconnected with a much larger porous medium, including the aquifer below where the hydrocarbons have migrated upwards into the trap which is our current reservoir. While the hydrocarbon-filled porous medium is of primary interest, part of the aquifer is commonly also included in the grid for better handling of the boundary conditions.

As the flow in the aquifer is single phase and the driving mechanism is the fluid expansion due to pressure depletion, the flow process in the aquifer is simple compared to the reservoir. The major part of the aquifer can therefore be represented by boundary conditions without significantly deterring the accuracy.

2.10.1 Grids Classification

There are many types of grids used in reservoir simulation. They are typically distinguished by the geometry of the individual grid cells and how these grid cells are interconnected.

There are several grid types such as:

- Regular grid
- Cornerpoint grid
- 2D Voronoi grid
- Unstructured grid
- Local grid refinements

2.10.2 Regular Grids

The most basic grids are regular grids with cuboid grid cells. We distinguish between a normal grid, as seen in figure 2.10, and a logical grid, which is essentially just a technique to number cells and corners (see figure 2.12a).

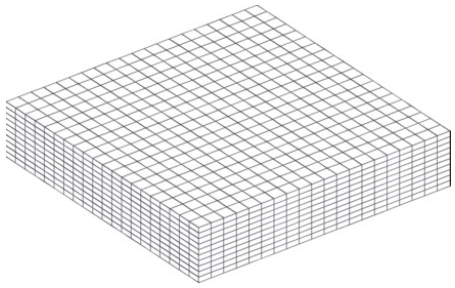
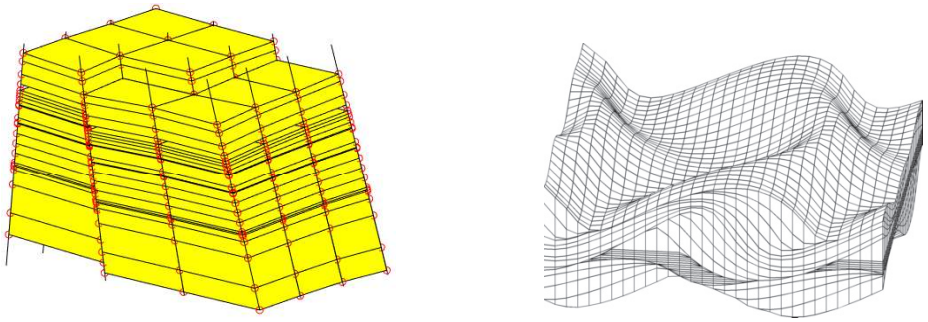


Figure 2.10: Logical Cartesian grids [22].

2.10.3 Cornerpoint Grids (Needs to be organized)

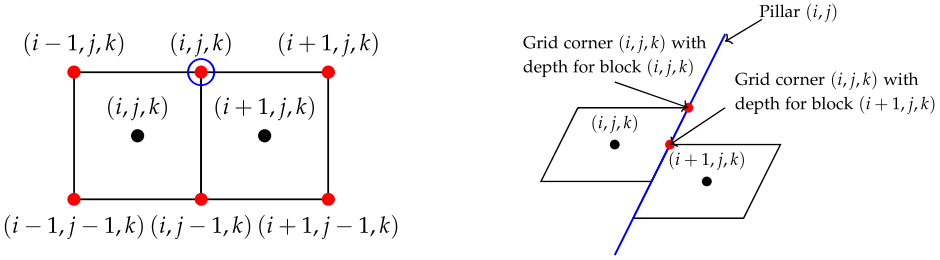
The cornerpoint (or pillar) grid, a traditional grid design that is supported by all reservoir simulators, is the most popular grid type. At many companies, corner point grids are the only type of grid that is used in official models. Figure 2.11 is a reference figure that shows the type of grid. It is noteworthy to mention that `Flow` only works with corner-point grids.



(a) Cornerpoint grids schematic [21]. **(b)** Cornerpoint Cartesian grids [22].

Figure 2.11: Cornerpoint grids scheme.

Regular i, j, and k ordering (labeling) is used for grid cells. All grid cells have eight corners, however, because the corners can be folded vertically, the number of real corners in a grid cell can range from four to eight. Cells with four corners have 0 volume and are completely collapsed. On pillars are defined all of the corners. The pillars (i-1, j-1), (i-1, j), (i, j-1), and (i, j) support the corners of the grid cell (i, j, k). Normal pillars are not vertical but rather normal to the layers since this improves K-orthogonality. The height of the grid corners along the pillars could differ for different grid cells in the corner point grid, as shown in figure 2.12b, and logical neighbors do not necessarily need to have a shared face.



(a) Regular grids. **(b)** Cornerpoint grids

Figure 2.12: Logical vs. cornerpoint grids [21].

As shown in figure 2.12, logical neighbors in the corner point grid are not guaranteed to have the same face. Moreover, the height of the grid corners along

the pillars may vary depending on the grid cell. In the standard (i, j, k) grid, eight grid cells share each logical corner. To indicate faults and non-reservoir gaps, several depths might be utilized for the same logical corner.

showing non-reservoir gaps and faults. It would seem simple to define pillars along the fault planes to illustrate 2D fault patterns. This approximation may have a significant impact on compartment volumes. In order for well paths near faults to fill the proper compartments, the model may also need to be modified. Without a second vertical zigzag representation, it is impossible to express complex fault patterns in 3D. The calculation of across-fault communication can in this case best be described as complex until the faults are totally sealed. Cornerpoint grids cannot be modified to match well paths. Because of this, it is typically hard to obtain precise reservoir-well couplings using this grid pattern.

2.11 Reservoir Simulators

2.11.1 Thermal and Iso-thermal Simulators

The variation between thermal and isothermal simulators is the first important difference. Except for the region near the wellbore, where injected fluids might be much warmer than the reservoir, temperature fluctuations in reservoirs are frequently so negligible that they have minimal effect on fluid and rock properties. In this circumstance, we just need to solve for the conservation of mass because we might assume the temperature as constant. It should be mentioned that in this situation, changes in near-well characteristics brought on by the injection of fluids at temperatures much higher than the reservoir should also be taken into consideration. In a thermal simulation like this one, we must take into account both the conservation of mass and the conservation of energy.

2.11.2 Compositional Simulators

The second distinction is between simulators using table-based thermodynamic descriptions, usually called the black oil model, and compositional simulators, which are based on an explicit equation of state (EoS). The fluid components in the pore space may be spread throughout various phases in mutual equilibrium depending on pressure, temperature, and total composition.

A liquid phase, primarily made of water, and a gas phase, with a composition comparable to that of atmospheric air, are both present in the pore space of the vadose zone. The pore space in the phreatic zone (below the water table) is completely saturated with a single-liquid phase. The composition and quantity of dissolved particles, salts, and gases have only a small impact on the density and viscosity of water-rich phases, such as brine and groundwater. We will need to calculate the fluid characteristics from the composition if the composition significantly affects the liquid properties, as is the case with salinization in freshwater aquifers and dissolved CO_2 in brine during CO_2 sequestration.

The formation water, or brine phase, is typically considered as being immiscible with the hydrocarbon-rich phases and the composition is presumed constant when modeling petroleum reservoirs. As a result, the process of creating water is thought of as a phase with a single pseudo component termed water. On the other hand, the oil and gas phase can exchange components, and thermodynamic equilibrium governs the saturation and composition of the two phases. This phase equilibrium is generated in a compositional simulator utilizing an explicit equation of state and many pseudo-components that are closely comparable to the actual hydrocarbon constituents.

Compositional Two-phase Flow Models

This is relevant for both basic water-air systems in hydrology and water-gas systems, such as gas reservoirs with aquifer support or generated with support from water injection. Also, it applies to oil-water systems, such as those with water-injected oil reservoirs.

Actual subsurface systems have many different components and a variety of different molecular species. As a function of pressure, temperature, and composition, these systems often reveal a rich phase behavior. A comprehensive understanding of this behavior is vital to the development and operation of top-side equipment, such as separators, in order to maximize liquid production and prevent wax and asphaltene precipitation. Even with the most unique of recovery procedures, the subsurface reservoir's complete range of pressure and composition is still far from being explored. Despite potentially significant temperature variations around injectors, reservoir temperature may typically be assumed to remain constant.

2.11.3 Black-Oil Simulator

Flash calculations are used to determine the equilibrium state for a particular pressure, temperature, and total composition. In black oil simulators, this equilibrium is derived from tables. Evidently, using a table lookup is considerably more effective than using a computer to solve a set of equilibrium equations. A black oil simulator will require significantly less computing power than a compositional simulator because it also operates with just two pseudo-components. Conventionally, oil and gas are used to refer to the two artificial components in the black oil concept. Although the oil pseudo component represents the non-volatile fraction of the hydrocarbon mixture, which is in the liquid (oil) phase in standard conditions, the gas component represents the volatile fraction of the hydrocarbon mixture, which is in the gas phase. In general, each component may be present in both reservoir gas and oil. This symmetric formulation is sometimes referred to as the "wet-gas model," and a distinction is drawn between it and the "dry-gas model," in which there is no oil component in the gas phase.

The black oil model is used to describe how the thermodynamic equilibrium between the different hydrocarbon phases is expressed. The model can, however,

be used to simulate practically any system with two hypothetical components due to being table-based. The black oil model can be used, for instance, to simulate the injection of CO_2 into an aquifer. In this instance, we have the two gas phases (usually a supercritical mixture of CO_2 with H_2O and extra gas components in injected gas) and brine, as well as the two pseudo components injected gas (carbon dioxide with additional gas components) and water (with dissolved CO_2). Even though the labeling can be unclear, as the brine phase is typically denser than the CO_2-rich phase, it is conventional in this context to identify the brine with the oil phase in the black-oil model and the water component with the oil pseudo component.

Black-oil Two-phase Flow Models

The black oil model is a two-part model that describes the density and phase characteristics of a fluid system. It provides a solid description of the oil reservoirs generated by pressure depletion in its original form, which is based on the direct tabulation of measured volume ratios in a relatively basic pressure depletion experiment. The black oil model is by far the most prominent phase behavior description used in petroleum reservoir simulation, and the model has been expanded to include different scenarios [23].

2.12 Reservoir Model

A reservoir simulation model realization, also known as a reservoir model, is the acronym for the description of the subsurface that is presented as reservoir simulation model realization. All critical details regarding the reservoir must be included in the reservoir model, including a description of the reservoir geometry (grid model), petrophysical characteristics, and fluid characteristics.

The input data for the simulator must also include information on wells, well controls, and production limits in order to mimic various production scenarios. A simulation model constitutes all of the simulator input. Reservoir simulation can be trans-disciplinary and difficult because it involves a variety of skills. It can also be lucrative since it provides a clear picture of the processes involved in reaching reservoir management decisions.

In oil reservoirs, in addition to brine, the pores also hold hydrocarbons (oil and gas). Since impermeable material prevents the hydrocarbons from moving upward, they accumulate. In a reservoir used for CO_2 sequestration in an aquifer, CO_2 is present as a cap deep beneath an impermeable layer, but there will also be a substantial amount of CO_2 dissolved in the water. Sedimentary rocks like sandstone and carbonates are often the porous material for deeper aquifers, oil reservoirs, and CO_2 storage sites. As has been mentioned earlier, there is an impermeable bed on top of both CO_2 storage sites and hydrocarbon reservoirs that prevent the upward migration of the CO_2 or hydrocarbons. On the other hand, im-

permeable beds capture freshwater resources in aquifers by preventing the downward flow of deposited water.

2.12.1 Geo-Model

The gathering of geological information, such as seismic data, well log data, and core analysis data, is the preliminary stage in every reservoir simulation study. This geological data must be integrated into the geomodel along with analog data (the static model). The distribution of porosity and permeability within reservoir compartments, together with information on reservoir compartments, are all included in the geomodel.

2.12.2 PVT Model

The second step is the gathering of additional data for the dynamic model, including data for the saturation functions from *special core analysis (SCAL)* and fluid sample analysis to create a PVT model (relative permeability and capillary pressure). Based on the distribution of various geological bodies, saturation functions will be linked to various reservoir regions.

2.12.3 Grid Model

The suitable grid size for reservoir modeling must also be decided. A suitable balance between computational efficiency and numerical and geological accuracy must be met by the grid size.

2.12.4 Faults

Reservoirs may be constrained on both sides by faults. Rock forms discontinuities as a result of the harsh movement of the Earth's crust, which causes the rock to move in opposing directions on either side of the fault. Surveying can detect faults because they are abrupt shifts in generally continuous beds. Since some faults are essentially impermeable, they serve as the reservoir's boundary. The degree of permeability of faults fluctuates. They can then utilize this method's fluid trap to act for upward mobility..

2.12.5 Uncertainty

Reservoir simulations are almost never certain, especially early on and throughout pre-production. The distribution of faults, relative permeability shape, reservoir heterogeneity, and other uncertain parameters all add to the total uncertainty of the reservoir simulation results.

2.12.6 History Matching

An ensemble of model realizations, each of which gives a unique forecast, best captures the uncertainty in a prediction. Although this is a major endeavor and commodity, simpler procedures using a single base-case realization plus sensitivity techniques are commonly used. Our best approximation for the subsurface behavior based on all available data is the history-matched reservoir model, which is the process of adjusting reservoir models to be compatible with production data.

2.13 Building a Reservoir Model

The fundamental objective of developing a reservoir model is to preserve subsurface data in a way that allows for quantitative predictions to be made about the reservoir and its behavior. The model depends on information that is generally recognized about the subsurface, including seismic, well logs, core material, observable fluid contacts, etc., all of which have associated uncertainties. Managing decisions, such as development plans, well placement goals, recovery optimization, etc., are based on the model prediction (see figure 2.13).

The model must include geological and other subsurface information in a format that may be used to develop reservoir simulation models. These realizations are specific input data for reservoir simulators that, when used in the simulator, match the behavior of a potential reservoir.

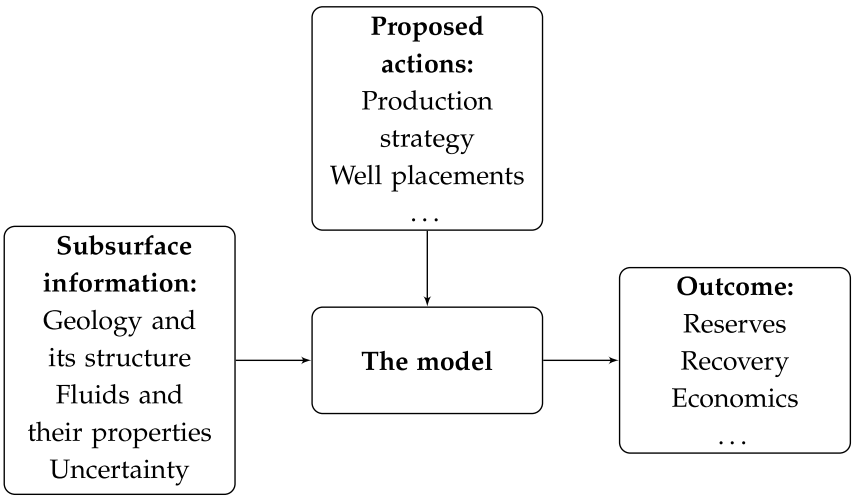


Figure 2.13: Reservoir modeling general intent flow diagram [21].

2.14 Effect Of The Penalty Cost On CO_2 Leakage Deterrence

Within the optimization loop, an algorithm is employed to choose appropriate well control parameters before allowing the simulator to execute the reservoir model. The utilization of computer resources throughout the optimization cycle is primarily attributed to the simulation process. The objective function of the optimization problem is subsequently computed based on the simulation findings.

In the context of well-control optimization and leakage avoidance, it is common to develop an objective function that focuses on either oil production or NPV. In order to assess the influence of emission costs, we will utilize a `Python` script for calculating the non-profit value as outlined in appendix A.4. The provided code script is designed to compute the penalty cost incurred by firms as a result of CO_2 leakage within restricted zones. In conjunction with the traditional NPV method, we will incorporate an additional component to accommodate the expenses associated with carbon dioxide (CO_2) emissions. The goal function employed throughout the optimization process is the enhanced NPV formulation.

The variables utilized in this study to manage the well are the targeted bottomhole pressure (BHP) for the producers and the targeted injection rate of CO_2 for the injectors. The estimation of leakage cost in the extended NPV code pertains to the financial gain achieved by injection operators through the prevention of leakage into the restricted region under the jurisdiction of the authorities. In contrast to the conventional optimization of reservoir simulation models, the present approach necessitates the use of a subsurface characterization model. Although the integrated model is simplified, it is sufficient for establishing a connection between an injection technique and the leakage of CO_2.

Chapter 3

Material

The monitoring of injected CO_2 is mandated by regulations to ensure safe storage practices and serves as a crucial tool for improving injection operations and confirming storage capacities. In this thesis, The captured CO_2 will be pumped into a deep saline aquifer situated at a depth of approximately 2700 meters. The design of the optimal monitoring program necessitated consideration of the injectivity, cost, and the evaluation of leakage risk at the chosen injection site. The primary objective of this program is to provide evidence of conformance, wherein the CO_2 plume acts in accordance with predictions, and to ensure containment by preventing any leakage from the storage facility. The proposed component encompasses the use of downhole behavior simulation techniques to observe the well and its surrounding vicinity, alongside the utilization of different simulators and code scripts.

This chapter will examine the materials utilized to model the storage reaction in the context of the CCS project, offering a comprehensive analysis of their technical and authoritative aspects. The investigation and description encompass the geological model, reservoir simulators, computer languages, and visualizers in their entirety. The project aims to enhance the simulation by incorporating extra details obtained from various sources. These resources are utilized to effectively achieve the objectives of the project.

3.1 Software

3.1.1 Simulator

To develop the current model of *Aurora*, a collection of commercial and open-source software is available. Highly regarded programs such as `E100`, `E300`, `Pflotran`, `tNavigator` and `Flow` which is developed by the `OPM` project. Since the primary objective of the research is the availability of data for future development and public accessibility is of extreme importance, the `OPM` project's software is chosen due to its high efficiency, open-source licenses, and support for the other commercial software formats.

OPM [24] facilitates collaborative software development, maintains and distributes open-source software and open data sets, and strives to ensure that these are available under a free license for years to come.

OPM-Flow

The OPM initiative started in 2009 as a collaboration between *Equinor* and several research institutions including *SINTEF*, and the focus of the initiative has been to develop open-source software for simulation and visualization of flow and transport in porous media [21]. OPM-Flow is an entirely implicit black-oil simulator that can execute industry-standard simulation models. Additionally, Flow can read and parse ECLIPSE input decks, as well as output restore and summary files in ECLIPSE format. The most essential engineering and mathematical features of this simulator are the two-point flux approximation in space with upstream mobility weighting and adaptive step-size controls, which are employed by the Flow simulator.

Operating Systems and License

OPM-Flow is licensed by *GPL 3+* [25] and the main operating system for OPM-Flow is Linux, and there are repository packages for the most common Linux distributions (Ubuntu and CentOS). OPM-Flow can also be run using a *Docker* container, which is possible in Microsoft Windows. It can also be compiled from sources on both Linux and MacOS. The input data files for OPM-Flow are similar to input files used for ECLIPSE100, and many simulation models can be run with both these reservoir simulation software. However, there are keywords that are not compatible between these two simulators.

3.1.2 Visualization Interface

ResInsight

For visualization of reservoir models and simulation results, we will use ResInsight, also part of OPM. ResInsight is an open-source visualization software that is part of the OPM project and was developed by *Ceetron Solutions* and *Equinor* in partnership. This software can visualize output from the Flow simulator.

ResInsight is a cross-platform, open-source 3D visualization, curve plotting, and post-processing utility for ECLIPSE reservoir models and simulations. With specialized visualizations of properties, defects, and wells, the user interface is tailored for the efficient interpretation of reservoir simulation data and it facilitates the management of numerous realizations and the calculation of statistics. The software is licensed under *GPL 3+* [26] and an important feature in ResInsight is Octave and Python integration. The Python integration is a newer add-on for using Python to post-process simulation results [27].

3.2 Geological Model

Eigestad Model

The *Johansen* formation is an ideal candidate for CO_2 storage due to the pressure regimes that exist between 2200 and 3100 meters below sea level. The formation has lateral extensions of up to 100 kilometers north-south and 60 kilometers east-west and a thickness of approximately 100 meters. Taking into account residual saline saturation (approximately 20%), the *Johansen* formation has a theoretical storage capacity of greater than 1 Gt CO_2 with average porosities of approximately 25%.

It is decided to apply the last updated, upscaled and licensed model that is made accessible by *Open Data Commons (ODbL v1.0)* [28]. One of the latest updated open-licensed geological models for Johansen formation in the *Aurora* zone is the *Eigestad* model updated in 2022 [29]. This dataset represents a model of the *Johansen* formation developed as part of the *MatMoRA* strategic research initiative (2007-2011), which was funded in part by *Norsk Hydro*, *Equinor*, and *Shell* [29].

The effect of various boundary conditions and sensitivity is considered with regard to vertical grid refinement, permeability data, transmissibility data, and the effect of residual gas saturations, as these have a significant impact on the distribution of the CO_2 plume. The geological investigation of the *Johansen* formation is conducted using seismic and well data (appendix C).

The model is based on a visualization of the extant seismic and well data, which includes high-quality 3D seismic data sets in the *Troll* field region and a 2D seismic grid of fair quality from the 1990s south of the field. Log data from 12 exploration wells in the *Troll* field and a few wells from neighboring fields that penetrated the *Johansen* formation or analogs are utilized. The objective of the model was to facilitate a rapid assessment of the viability of utilizing the *Johansen* formation for CO_2 sequestration. Consequently, the model simplifies the geological stratification, and in particular, the *Dunlin* group, which is predominately shale but can be subdivided into distinct formations, some of which are locally sandstone, can be subdivided into distinct formations [1].

The *Johansen* Formation has been subdivided into 3 zones that have been distributed throughout the region. The interpretation of the fissure is based on seismic data. In the southern part, the fault movements and intensity are significantly lower than in the northern part. The most significant fault in the study area is the main north-south trending fault that causes a small high in the center of the cross-section and proceeds to the north, dividing the western portion of the *Troll* field into two segments.

Layers of shale and sandstone separate the *Johansen* formation from the formations above and function primarily as horizontal seals. In particular, the *Dunlin* group shales lie immediately above the *Johansen* formation and function as the formation's cap rock. The *Dunlin* group is typically very thick, reaching hundreds of meters in certain locations, but it may disappear in some of the formation's

eastern regions. *Dunlin* group is rich in clay and sediment and as a result, the model's flaws primarily have a hermetic effect.

From a CO_2 storage perspective, the combination of dense layers of shale and sealing faults is preferable due to the high number of static traps that can be created. If shale dissipates or if faults are not completely sealed, there may be potential CO_2 leakage pathways. Currently, the geological model does not account for these issues in any detail.

Estimates of leakage can only be derived from more comprehensive geological modeling and numerical flow simulations. The geophysical modeling of *Johansen* and neighboring formations suggests that *Johansen* possesses favorable sandstone properties. Geologists' interpretations of geological layer horizons form the basis of the geological model for the *Johansen* formation and the sedimentary sequences above and below where there have been geological modeling within the groups. Seismic and well correlations have been utilized to model petrophysical data, including porosity and permeability while there was no model of geological facies available. From the definition of geological category horizons, a geological grid is constructed. Specifically, the primary order of geological zones from top to bottom is as follows:

1. Top *Sognefjord*
2. *Sognefjord* shale
3. *Fensfjord* formation
4. *Krossfjord* formation
5. *Krossfjord-Brent* group
6. *Brent* group
7. *Dunlin* group
8. *Johansen* formation (thickness varies between 80 m and 120 m)
9. *Amundsen* shale
10. *Statfjord* formation

Some of these are divided into sub-zones, bringing the total number of geological zones in the model to 16 layers. Specifically, the *Johansen* formation is composed of 3 subzones. The complete model's lateral extent is approximately 75 km by 100 km. This sector consists of the *Amundsen*, *Johansen*, and *Dunlin* geological zones.

The *Johansen* formation is located 60 kilometers off the coast of Norway's *Mongstad* region, in the deeper portion of the *Sognefjord* delta. The *Troll* field is one of the greatest gas fields in the *North Sea* and is situated roughly 500 meters above the northwestern parts of the *Johansen* formation. The uppermost sediment of the *Sognefjord* delta is the *Sognefjord* formation, which is more than 500 meters above the *Johansen* formation. For the immense *Troll* oil and gas field, the *Sognefjord* formation is the main hydrocarbon reservoir.

The porosity and permeability values of the model are derived from exploration well log and core data. The most reliable information for use in this simulation model are extracted from *Equinor* well drilling data in appendix C. During

an extended evaluation, log data, geological data, and geothermal data are extracted from well logs and laboratory test results to obtain the most accurate data possible. The model calculates the porosities of each *Johansen* Formation stratum using a porosity-depth trend for each of the zones. Since all *Troll* wells are shallow, additional well data from the *Fram* field to the north of *Troll* were used to acquire reliable depth trends.

3.2.1 Sector Models

The *Johansen* formation comprises three distinct layers. The sector models correspond to the geological domain's southeastern regions and they are separated by a 100×100 lateral grid with three vertical resolutions of the *Johansen* formation:

- Sector5: 5 layers in *Johansen*, $100 \times 100 \times 11$
- Sector10: 10 layers in *Johansen*, $100 \times 100 \times 16$
- Sector15: 15 layers in *Johansen*, $100 \times 100 \times 21$

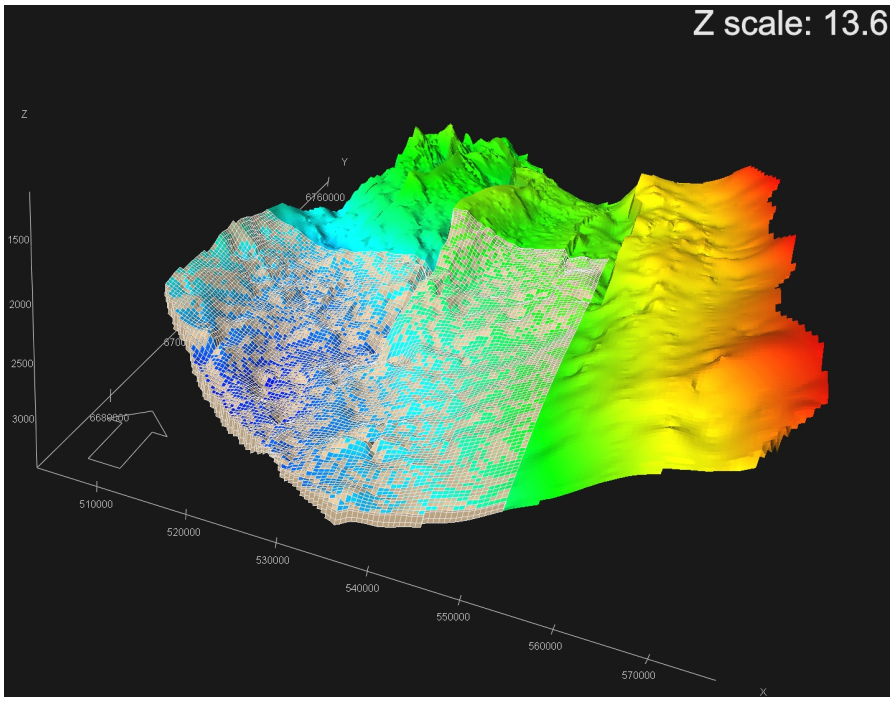


Figure 3.1: Overview of the *Aurora* field geological model [29].

In all sector models, the shale (*Dunlin*) above *Johansen* is depicted by five grid layers. Similarly, all sector models contain one shale stratum beneath the *Johansen* formation. The permeability description within the sector model, in which the *Johansen* formation is depicted by five layers of grid cells (see figure 3.1). The five layers above *Johansen* correspond to the grid representation of the *Dunlin* shale. Notice the primary flaw that divides *Johansen* into two distinct sections.

The geometry files are provided as input files for `ECLIPSE`.

3.2.2 Full Field Model

In addition to the 3 sector models, there is also a full-field model provided in figure 3.2.

- FullField: 149 × 189 × 16

It is noteworthy to mention that the sector models correspond to the identical portion of the full-field model.

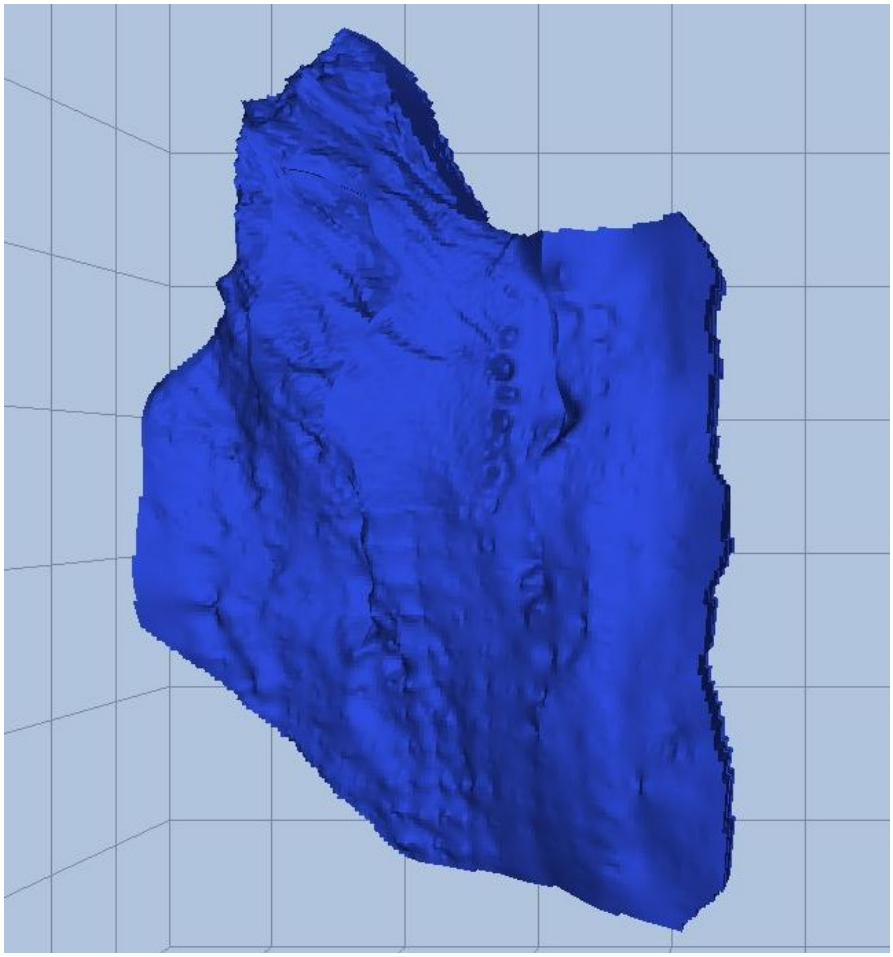


Figure 3.2: Overview of the *Aurora* field geological model in `ResInsight`.

Petrophysical Data

Each grid cell, including inactive cells, is assigned a non-negative value between 0 and 1 that represents the grid's porosity (figure 3.3). The x-direction permeability of the grid is represented by a positive value for each cell, including inactive

cells. In this data set, the y-direction permeability is equivalent to the x-direction permeability, whereas the z-direction permeability is one-tenth of the x-direction permeability (see figure 3.4).

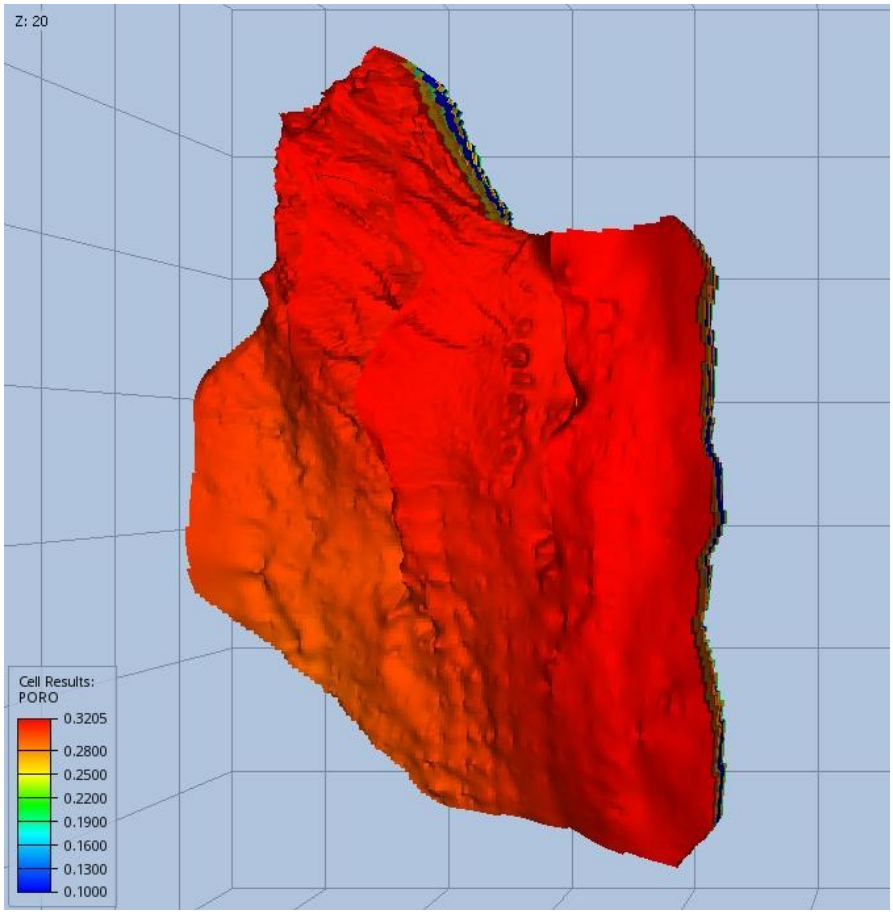


Figure 3.3: *Aurora* full field model porosity distribution.

3.2.3 PVT and Relative Permeability Data

PVT Data

Based on the properties of water and CO_2 at a constant temperature of 98 °C, PVT data is provided. The fluid system is comprised of two immiscible phases: water and carbon dioxide. The density and viscosity values for both phases are tabulated as a function of pressure. The isothermal properties for 98 °C are given for pressures between 230 and 400 bars.

There are three columns in the data files:

1. Pressure
2. Density

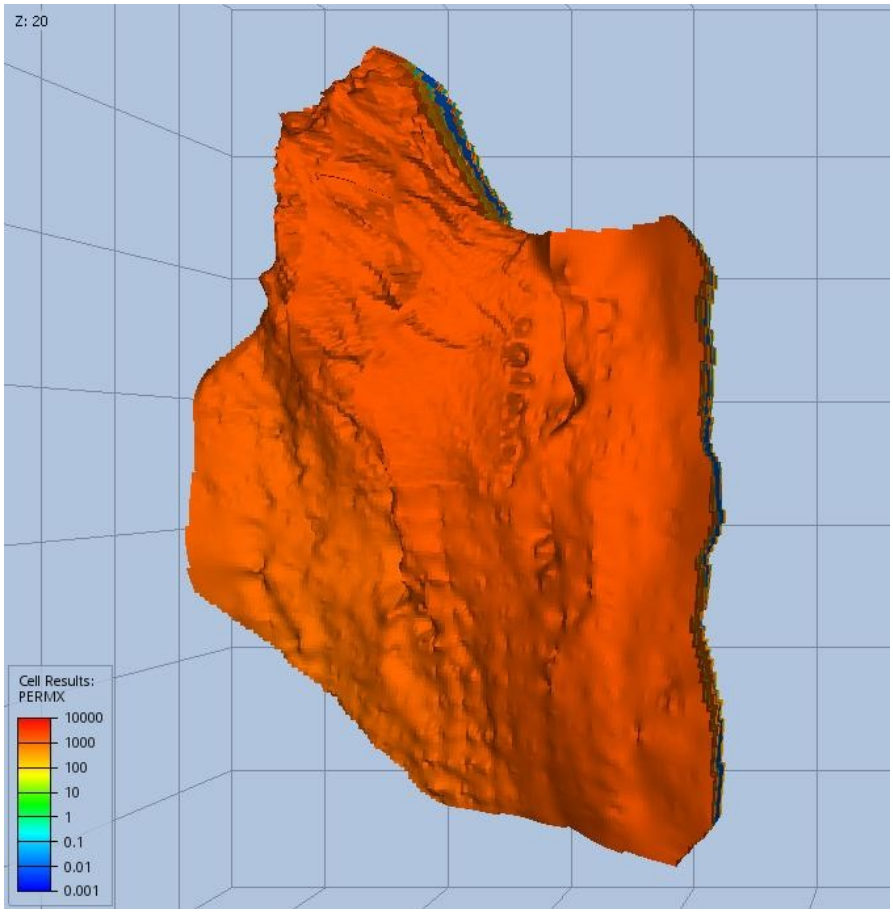


Figure 3.4: *Aurora* full field model permeability distribution in x and y directions.

3. Viscosity

Relative Permeability Data

Three sets of relative permeability profiles for the two-phase water and CO_2 system are provided. The relative permeability values are tabulated as a function of the corresponding phase saturation. There are two columns in the data files: phase saturation and phase relative permeability. A dataset employs a two-phase flow to determine the relative permeability of the water phase and CO_2 phase and all the data was compiled into a single simulation code file before being conducted.

3.2.4 Permeability

The geometry of the formation, as well as its porosities, permeabilities, and fluid data, are available on the website [29] along with the information described below in figure 3.5). For the permeability of the *Dunlin* and *Amundsen* shales, constant

values are used; 0.01 mD and 0.1 mD (1D = 9.8691013m2), respectively. The *Johansen* formation's permeability ranges between 64 and 1660 mD. The data upon permeability are derived from well interpretation and correlations of depth. Figure 2.1 depicts, from west to east, the vertical grid representation (with permeability values) at a cross-section within an extracted sector model.

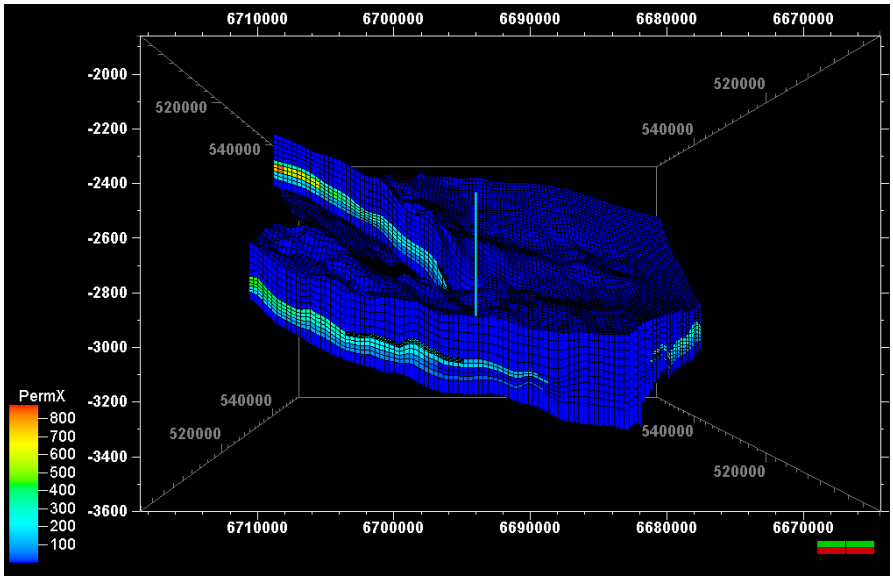


Figure 3.5: Permeability representation in the *Eigestad* model of the *Aurora* field[29].

3.2.5 Grids

The grid resolution is an important numerical factor to consider. We contend that a relatively high vertical resolution is required to spot the CO_2 plume that accompanies the formation's sealed roof as a result of gravity override. Without grid refinement 3, upstream weighting results in numerical diffusion, which may manifest as leakage into the formations above.

The simulation grids are provided in the form of cornerpoint grids, which is an extension of the format for logical *Cartesian* grids as already mentioned in Chapter 2. In general, due to the vertical faulting of geological zones, the outer edges of grid cells do not conform from one grid block to the next. Using a $149 \times 189 \times 16$ grid dimension, the full-field model is discretized. This grid defines every zone and the entire lateral domain. The cornerpoint format implies the distribution of grid cell corners along vertical pillars, see [30], for details. There are eight corners on each grid cell, but these may not be distinct due to pinch-outs. Since grids are permitted to contain vertical faults, each grid block contains all eight corners. This produces an abundance of data. This grid's typical cell size is $500m \times 500m$ horizontally and 16 to 24 meters vertically. It is important to mention that the *Johansen* formation comprises three distinct layers.

3.2.6 Faults

The *Aurora* storage site is situated on *Troll West*'s fault, an eastward inclined fault block bordered by the *Tusse* and *Svarta* fault zones to the east and west, respectively. In parallel with their fault strike, these fault zones have substantial fault throws. Along with these thick-skinned fault zones, the *Aurora* model's storage complex is intersected by a number of intra-block faults with throws lower than the principal sealing unit (the *Drake* formation) [8]. The eastern and north-eastern dipping intra-block faults are more likely to impede fluid movement than the western and south-western dipping faults, according to the most recent across-fault evaluation on the *Aurora* storage site.

Since the geological model indicates that faults seal when the fault throw is substantial, we do not account for vertical flow in faults towards shallower regions. In [31], the flow of CO_2 through vertical faults was investigated by incorporating fault transmissibility multipliers into their model and investigating injection points further north, closer to *Troll*. When choosing an injection site, it is important to consider injectivity, avoiding CO_2 from flowing into higher sections of the formation, and preventing CO_2 from moving into regions near faults with uncertain sealing properties (Chapter 1). The location of the *Troll* field will also influence the injection site selection.

Except for `FAULT3`, `FAULT9`, and `FAULT11N`, which have `MULTFLT` values of 0.1, all fault transmissibility values are set to `MULTFLT=1`. `FAULT4` is the fault that is closest to the *Well 31/5-7* well and covers the largest area in the model (figure 3.6), making it the most significant fault in this study. Sensitivity research was conducted on `FAULT4` to investigate its effect acting as a sealing fault or conduit, which will be detailed in the sensitivity analysis section, as the injection operation has not yet begun and doubts about how faults operate when facing the fluid flow are still considerable.

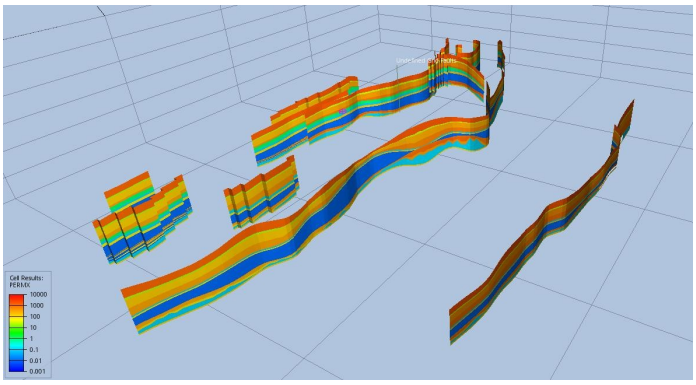


Figure 3.6: Faults overview of the *Aurora* field.

3.3 PVT Data

The CO_2-brine PVT model calculates PVT parameters such as density, viscosity, and enthalpy internally as functions of pressure, temperature, and composition by employing analytic correlations and models from the literature rather than interpolating from tabulated values. The `Flow` simulator converts these data to the standard black-oil equivalent PVT tables. The dataset presented here provides geometry, geology, and petrophysical properties for a realistic storage site. We consider a two-phase immiscible flow with CO_2 being the nonwetting phase, and the resident brine being wetting. The phases are isothermal, and compressible, and have densities and viscosities that vary with pressure.

CO_2 PVT Data

Isothermal PVT data for CO_2 at 98 °C are generated from Span-Wagner[15] model values which are proved by experience to be the most realistic data available for CO_2 PVT properties in the reservoir and the standard conditions (appendix B.2). An interpolation of the data is used in order to find the exact volume or mass of CO_2 on the surface for each cell according to its specific pressure value. To achieve this aim a `Python` code was developed to do the interpolation for each as shown in appendix A.2. The CO_2 density in the standard condition is assumed to be 1.87185 Kg/m^3 referring to the calculated value by [15].

3.3.1 Water PVT Data

Eigestad PVT Table

To do evaluations according to the brine PVT data, the density and viscosity values in different pressure conditions in isothermal condition of 98 °C is provided by the model's PVT tables. Similarly, the density of brine in standard conditions is provided by the model and is assumed to be qual to 1 Kg/m^3. Utilizing a similar method, an interpolation `Python` script of brine data is used in order to find the exact volume or mass of formation water on the surface for each cell according to its specific pressure value (see appendix A.3).

Conversion Factor

The measurement system should be configured in the simulator so that values inputted in the input deck are calculated using the correct system and units. By selecting the *Metric* system in the simulations depicted, a conversion factor (equation 3.1) is computed to convert the mass of CO_2 or ($\frac{Mt}{Year}$) to the volume of CO_2 or ($\frac{Sm^3}{Day}$). The conversion factor formula is as follows:

$$R\left(\frac{Sm^3}{Day}\right) = R\left(\frac{Mt}{Year}\right) \times \frac{10^6}{1}\left(\frac{t}{Mt}\right) \times \frac{10^3}{1}\left(\frac{Kg}{t}\right) \times \frac{1}{1.872}\left(\frac{Sm^3}{Kg}\right) \times \frac{1}{365}\left(\frac{Year}{Day}\right) \tag{3.1}$$

$$R\left(\frac{Sm^3}{Day}\right) = 1465094,13 \times R\left(\frac{Mt}{Year}\right) \tag{3.2}$$

3.4 Reservoir Model Setup and Modifications

3.4.1 Configuration

Grids Refinement

Considering that it is the closest zone to the *Troll* gas field and that the injected CO_2 shouldn't interfere with the *Troll* field production operation, a specific area has been designated as the primary area to be monitored for leakage (see figure 3.7). The `ACTNUM` keyword is used for eliminating ineffective zones in an effort to reduce computational costs. The penalty zone has been designated on the west side of `Fault4`, and the zone's boundary.

Well Placement

The well placements are used as the real *well 31/5-7* and *well 31/8-1* locations as had been drilled. To achieve the precise location, the intersection point given by *NPD* for each well [9][10] are investigated and adapted by the grid cells intersection points to find the exact cell index (i, j, k) for the above-mentioned wells. The locations in the model are:

- Well 31/5-7 cell index: [109, 92]
- Well 31/8-1 cell index: [106, 106]

and the wells' connections (perforations) are set on layers 9, 11, and 13 separately, where the *Johansen* formation layers exist.

Grid and Boundary condition

The `Petrel SE` software platform uses cornerpoint grid formats, and there are 450,576 total grid cells $[149 \times 189 \times 16]$, while there are 17,782 total active cells after setting `ACTNUM` keyword. It is critical to enhance the simulation speed as many different instances need to be simulated. Computation time is one of the restrictions of the black-oil and compositional models, especially when working with multiphase systems and millions of grid cells. As previously stated, the prior work on the *Aurora* model mostly concentrated on the southern portion of the model in order to put the well at an appropriate distance from the *Troll* field and inject in the greatest depth possible. In this project, a new southern and eastern

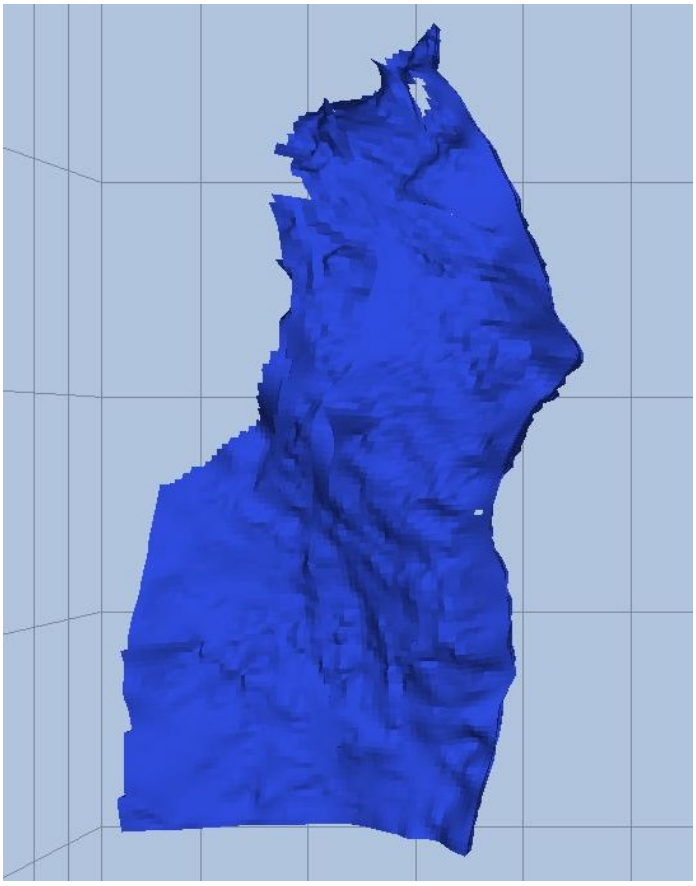


Figure 3.7: Overview of the *Aurora* field sector model in `ResInsight`.

boundary was set and cells from the lower center region of the model toward the southern and eastern half were deactivated mostly for two reasons:

1. The confirmation well *31/5-7* (*Eos*) and *31/8-1*, which are almost in the center of the model and will later be sidetracked and finished for injection operations, act as the basis for the base case model[8].
2. Gravitational force will predominately cause the injected CO_2 to migrate upwards due to the N-S structural dip of the reservoir and the low density of CO_2, which means the grid cells beyond the injection well will no longer contribute to the injection plan. Nearly half of the remaining active cells will be deactivated in the J-direction after defining the new southern boundary, enabling 17,782 cells for pressure and saturation calculations. This means that the simulation time will be significantly reduced.

Thermal Condition

The assumption of isothermal reservoir conditions is made based on slight fluctuations in temperature. The reservoir temperature was set to 98 °C and afterward

applied to the input deck using the RTEMP keyword.

Pore Volume

The total pore volume of the *Johansen* and *Cook* formations is estimated to be around 75 Gm^3, according to the Norwegian CO_2 storage Atlas[5]. The overall pore volume is roughly 280 Gm^3, assuming a sizable pore volume in association with the *Johansen* and Cook formations. In order to ensure that the reservoir has an open/semi-closed boundary and that the pressure dissipates out of the reservoir, the communicating pore volume multiplier of 5.95 was assigned using MULTPV in the eastern boundary. The total pore volume used in the base case model was derived from [4]. Only the western part of the *Tusse* fault's *Johansen* and *Cook* formations was taken into account when determining the total pore volume of $7.5E10\ Sm^3$.

Since accurate estimation can only be made after several years of injection by monitoring and modeling the injection pressure, there is a substantial degree of uncertainty over the *Aurora* site's actually linked pore volume [4]. Both the northern and southern boundaries were altered to achieve the total connected pore volume of $7.5E10\ Sm^3$.

Relative Permeability Curve

Another important modification was changing the saturation end-point values for the base case model. In the model's scaled PVT table used to measure the saturation end-point values where S_{gr}=0.254 and S_{wirr}=0.15 are reported as the residual gas saturation and irreducible water saturation, respectively. The water and gas relative permeability values would be calculated by the simulator using CO2STORE keyword, and the saturation end-point values were changed to *Johansen*'s formation values.

3.4.2 Keywords Adjustment

DISGAS

The term should only be used if the model has both the oil and gas phases since it shows that there is dissolved gas in living oil. Oil-water and oil-water-gas input decks that contain the oil and gas phases may utilize the keyword. The keyword will also activate data input file checks to make sure that the input deck contains definitions for all necessary oil and gas phase input parameters. When calculating and reporting, Flow takes the constant R_s into consideration [24]. To calculate R_s and the dissolved gas volume and mass of CO_2, DISGAS keyword has been put in the input deck (appendix A.1).

CO2STORE

The `CO2STORE` keyword enables the CO_2 storage model during the run, which uses the simulator's CO_2-brine PVT model to account for both carbon dioxide and water phase solubility. Despite being a compositional keyword in the commercial simulator, `Flow`'s black-oil model has adopted it.

The CO_2-brine PVT model uses analytic correlations and models from the literature rather than interpolating from tabulated values to calculate PVT parameters including density, viscosity, and enthalpy internally as functions of pressure, temperature, and composition. The simulator internally transforms these data to the conventional black oil equivalent PVT tables. As a result, while this model is activated, `Flow` does not require the usual PVT keywords like `DENSITY`, `PVTO`, `PVDG`, etc., and the simulator will ignore them if they are entered [24]. The CO_2-brine PVT properties, which must be entered in the `PROPS` section, are determined by salinity. In this model, the `SALINITY` is set to 0.0011 $\left(\frac{Kg-M}{Kg}\right)$.

DRSDTCON

Based on the assumption of vertical equilibrium (VE), DRSDTCON defines a dimensionless parameter (χ) that regulates the convective dissolution of carbon dioxide (CO_2) into in-situ brine inside a grid cell [32]. The `DRSDT` keyword in the `SCHEDULE` section is unnecessary since the keyword internally instructs the simulator to calculate the solution gas-oil ratio R_s, which is typically defined by the `DRSDT1` option on the `DRSDT` keyword[24]. In this case, `DRSDT` is set to 0.04.

The maximum rate at which the solution gas-oil ratio R_s can be raised in a grid cell per time is the dissolving rate (as described by the `DRSDT` keyword) in the black oil model. Therefore, in order to transform equation 3.3 into a black oil formulation, we must first substitute the maximum concentration at the solubility limit with R_{ssat}, which represents the same quantity in black oil. where R_{ssat} is defined as:

$$R_s = \left(\frac{\chi_o^g}{1-\chi_o^g}\right)\left(\frac{\rho_{o,ref}}{\rho_{g,ref}}\right) \tag{3.3}$$

SGOF

The `SGFN` keyword specifies the gas relative permeability and the oil-gas capillary pressure data versus gas saturation tables when gas is present in the input deck and it is used only if the gas is in the run. In this model, only `SGOF` is used in the `PROPS` section to provide the relative permeability and saturation correlation data for the simulator.

3.5 Database

Since models are assessed from several reservoir engineering perspectives, particularly for formations where the injection/production operation has not yet begun, it is usually preferable to look into and learn from prior works while conducting simulation studies of a particular model. As a project is evaluated over time, additional seismic surveys and wells aid in modifying the models with more precise data, producing more accurate findings. Inaccurate or incomplete data on measurements of porosity and permeability, PVT and saturation tables, the condition of faults acting as sealing units or conduits, estimation of pore volume, and many other crucial factors can produce misleading, implausible simulation results that can result in the failure of a project on an economic and environmental level.

The *Johansen* formation has been the subject of some of the most significant investigations on the part of geology and reservoir engineering organizations. Each study's emphasis and key findings have been grouped and briefly discussed. One of the first teams working on the western portion of the *Johansen* formation model to assess the reservoir as a prospective storage site for future uses was "*P. S. Bergmo, E. Lindeberg, F. Riis and W. T. Johansen*" [31] using `E100` black-oil simulator, with a single vertical CO_2 injection well configured to inject 3 Mt of CO_2 annually for 110 years [1]. conducted dynamic simulation using `E100`, restricting the fluid flow simulation to two-phase immiscible flow while ignoring the impact of CO_2 solubility in the native brine system. In different research by *Sundal* [33], inspired by Eigestad [1], new 3D seismic data that Gassnova acquired in [6] was added and combined with prior surveys covering the *Troll* field to add further scientific knowledge basis for further examination of the *Johansen* formation as a prospective CO_2 storage site.

The 3D seismic surveys '*GN1001*', '*NPD-TW-08-4DTROLL*', and '*NH0701*' were combined via processing from field data into a new seismic volume '*GN10M1*' in order to create a consistent seismic database for seismic interpretation, inversion, and analysis. One of the recent 3D seismic survey, '*GN1001*', was gathered by *Gassnova* in 2010[6]. By measuring the amount of time in milliseconds, it takes a seismic wave to travel down to an interface and reflect back up to the receiver, scientists can use seismic data to build a representation of the subsurface that includes the reservoir, sealing units, faults, and other features. Utilizing sonic log data and giving minimal uncertainty in well-correlated regions, the velocity model '*hiQbeR*' was used to construct time-depth conversion and formation thickness maps [33].

3.6 Coding

We will use the `Python` programming language, including the *ecl* module, for scripting. `Python` is an open-source programming language that prioritizes readability at its center. It is designed to be extensible and offers a vast selection of libraries and programs. Numerous scientific computing libraries, such as *NumPy*,

SciPy, *Matplotlib*, *math*, *OS*, and *Numpy.Interp*, have made it very popular.

ecl Package

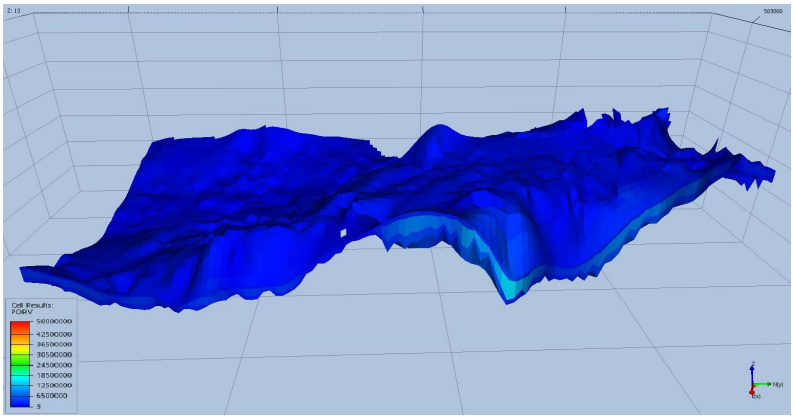
The Python module *ecl* is a reader and writer for `Flow` and other reservoir simulators' output files. that can read or write files compatible with the market standard formats used by the `ECLIPSE` simulator and defined by it. The `restart`, `init`, `rft`, `summary`, and `grid` file types are covered. Both formatted and unformatted files, as well as unified and non-unified files, are supported. This package is a powerful instrument for both pre-processing and post-processing `Flow` simulation results, such as in sensitivity analyses and optimization cycles.

ecl package has 4 main sub-libraries including *EclFile*, *EclKw*, *EclSum*, and *EclGrid*. Each library has an extended range of modules that can be utilized mostly for a desired model's post-processing data analysis. In order to have comprehensive guidance about the packages, *help* module of `Python` could be used for each specific library. There is also the possibility to investigate through the *ecl* library root folder for the help file of each library with is placed with the format of *'.py'*. *ecl* package could be downloaded through its web page with a similar name in *GitHub* [34].

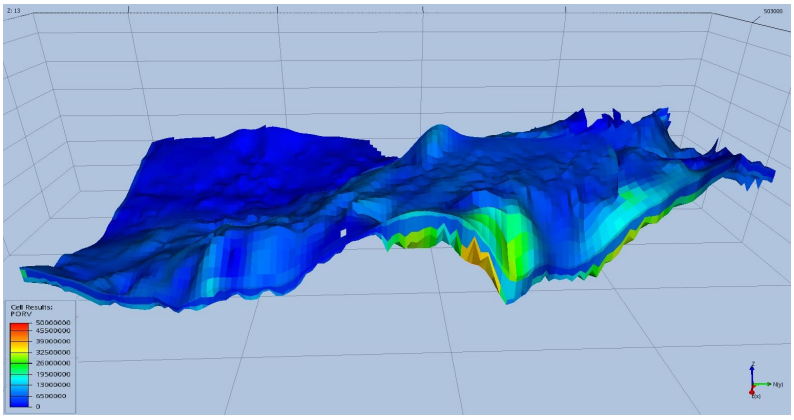
3.7 Boundary Conditions

Typically, abrupt changes in material properties must be compensated for by interface or boundary conditions. However, the finite volume methods are partially based on an integral formulation, and it is assumed that the permeability field is piecewise constant. The two-point flux approximation is derived assuming a piecewise linear pressure drop within the grid cells and performs effectively so long as abrupt permeability changes are confined to cell boundaries. Thus, it is preferable that zone boundaries, faults, and geological object boundaries coincide with cell faces. Consequently, these geological features should define a grid generation framework.

Since numerical issues are frequently associated with the flow through small grid cells, extending the zones and modifying the properties of the zone and its neighboring grid cells had been done in this project. The `MULTPV` keyword is used at the boundaries to expand the zone properties by increasing the total pore volume to 7.5E10 Sm^3 and to attempt to reduce the boundary effect (figure 3.8 and figure 3.9).

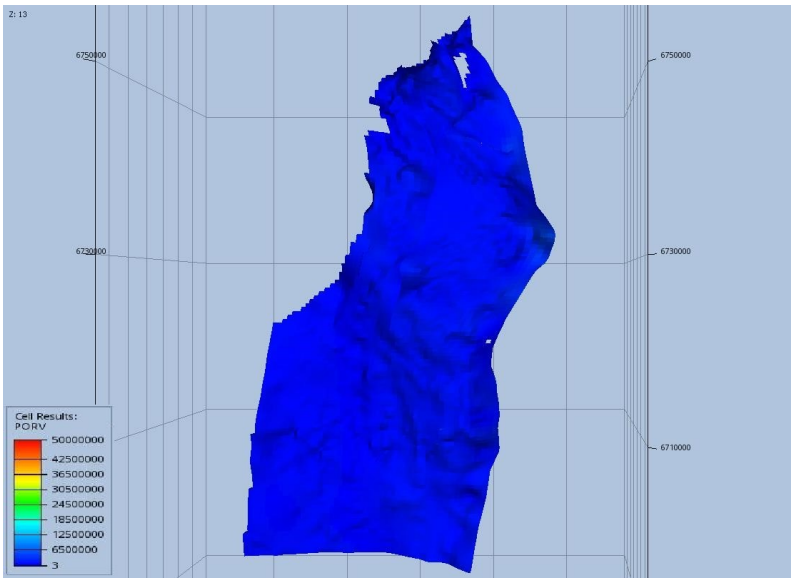


(a) Before `MULTPV` correction.

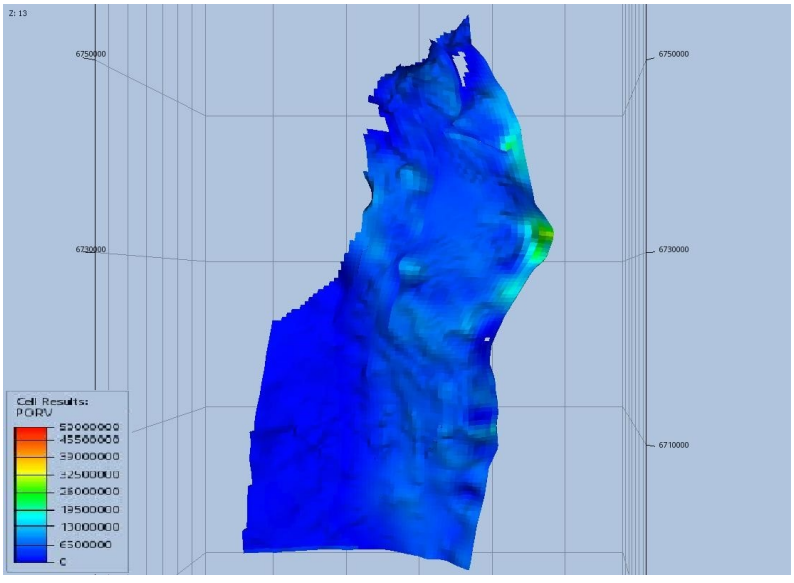


(b) After `MULTPV` correction.

Figure 3.8: `MULTPV` correction effect on the right boundary pore volume distribution.



(a) Before `MULTPV` correction.



(b) After `MULTPV` correction.

Figure 3.9: `MULTPV` correction effect on the reservoir's top layer pore volume distribution.

Chapter 4

Methodology

Reservoir simulation is an ongoing procedure that involves the integration of reservoir data into a simulation model during the production or injection phase. To ensure the utmost precision in estimating, the simulation model will be consistently revised alongside forthcoming projections, utilizing the existing reservoir data. Reservoir simulators are specifically engineered to replicate the intricate dynamics of fluid flow through a permeable base. A multitude of numerical simulation software options are currently accessible. The reservoir simulator utilized in this study is the `Flow` program, which is part of the `OPM` initiative.

Undertaking projects aimed at capturing CO_2 necessitates a substantial financial investment and a thorough assessment of the hazards linked to different sequestration strategies.

4.1 Proposal for a Project Idea

The primary objective of this project is to develop routines for CO_2 injection optimization. Various data and plot modules have been integrated within the simulators and visualization interfaces. However, it should be noted that these modules consist of pre-determined algorithms that may not adequately cater to the specific data and graphs in terms of their structure and range, hence limiting their usefulness for conducting a thorough analysis.

When used alongside a well-structured input deck, the *ecl* package application is an effective method for generating algorithms that can execute a range of scenarios on a given geological model. The first step in this endeavour is to create a `Flow` input deck that can be utilized and adjusted smoothly by the `Python` script. The programming language `Python` has the capability to make use of the pre-existing adaptable *'.DATA'* file in order to carry out the provided simulation scenarios and generate a comprehensive report together with graphical representations. By employing pre-established algorithms, this approach effectively reduces time consumption, enhances precision, and boosts operational efficiency.

In this project, the aim is to:

1. Establish an adjustable input deck (*'.DATA'* file).
2. Write a `Python` script to create a scenario cluster, where we store results in separate folders for every single scenario.
3. Establish a scenario naming system to apply for the folders and *'.DATA'* file names.
4. Report the results and warnings in a *'.txt'* file.
5. Plot the target data versus time in a comprehensive *'.pdf'* file for each scenario in a separate folder which could be recognized by its name.
6. Create a wells injection rate balance plot for further analysis.

While this workflow is general, this project focuses on a specific set of data, namely the Aurora example case. The factors that follow are the most important variables that will be investigated in each case:

Plots and warnings:

- Total injected mass of CO_2 (FGIT)
- Mass of injected CO_2 from each well (GIT)
- Unwanted shutdowns during the injection operation (GIR evaluation)
- Pressure build-up during the injection operation (BHP evaluation)
- Pore volume in the reservoir (PORV and MULTPV)
- Pore volume in the penalty area (PORV)
- Water and CO_2 saturation plots versus time (S_{wat})
- Dissolved gas in water ratio (R_s)
- Total mass of CO_2 in the reservoir as free gas
- Total mass of CO_2 in the reservoir as capillary trapped gas
- Total mass of CO_2 in the reservoir as dissolved gas
- Mass of CO_2 leaked to the penalty area
- Perform potential penalty cost for the CO_2 leaked to the penalty area
- Perform calculation to estimate the pressure towards the cap rock (the reservoir top layer) due to CO_2 accumulation below the cap rock
- Warnings for reaching potential fracture pressure in the sealing layer
- Rate balance plot

This `Python` script should be written in a way that makes it applicable to a variety of simulation projects and geo-models through the use of less general modifications.

It is necessary to conduct sensitivity analyses to determine how uncertain parameters will influence pressure changes, how much CO_2 will dissolve in the brine phase, and how the injected CO_2 will be distributed in the reservoir over time. In this undertaking, the provided *Aurora* model was enhanced and modified to create a new base case model. The `CO2STORE` option was then utilized to convert the *Aurora* model from a traditional black-oil model into a model that applies an equation of state and solubility models to calculate the amount of dissolved CO_2 in the brine phase over time.

Through a sensitivity analysis, many reservoir parameter uncertainties were

also considered which are mentioned in the previous section 4. Using the open-source simulator `Flow`, we investigate a number of the factors that we believe are essential for obtaining precise estimations of a formation's capacity. Possibly one of the most crucial factors is determining the formation's appropriate boundary conditions. By simulating post-injection migration, the potential effect of residual trapping and relative permeability models is then investigated. Though dissolution and structural capture mechanisms, the simulations are typically significant over longer timescales, and both are considered in this work.

4.2 Penalty Area

Previous chapters 1 mention the possibility of leakage into the *Troll* gas field through the north-western boundary of the *Aurora* field. Consequently, one of the primary focuses of the sensitivity studies should be the leakage toward the *Troll* field. Since this issue is of crucial significance, a penalty area is defined to analyze the data and results in accordance with it, and it serves as the benchmark for invasion into the penalty area. This penalty area serves as a safety factor for preventing CO_2 leakage into the *Troll* field, and there is the capacity to set a penalty cost system for each specific mass of CO_2 discharged into this area by means of any mechanism. This could serve as an effective safeguard to ensure the security of both reservoirs.

The eastern boundary of the penalty area is assigned to `FAULT4` and the northern and western boundaries are extended to the model boundaries nearby the *Troll* gas field. `FAULT4` has been marked as the southernmost boundary (see figure 3.6). Since there is a level discrepancy between the layers as a result of the `FAULT4` discontinuity, the leakage process to this location would most likely occur in the long-term post-injection period. The minimal and maximum grid indexes of the penalty area in the current model are as follows:

- Minimum: [115, 20, 9]
- Maximum: [144, 80, 13]

4.3 Injection and Monitoring Period

It has been attempted to align the scenario cases as closely as feasible with the *Northern Light* project. Since the project's injection rate is set to 30 years [35], the injection period applied to the current thesis studies accounts for 30 years of carbon dioxide injection.

Due to the long-term nature of gas dissolution in formation water, convective mixing dissolution effect, and certain geological trapping mechanisms, such as mineral trapping, post-injection monitoring was set to continue until 2500, or for nearly 445 years after injection operations cease. To be more detailed, the injection period is thirty years, beginning on January 1, 2025 and ending on January

1, 2055 and the observation would continue until January 1st, 2500, after which the well would be decommissioned.

The establishment of a pressure limit serves the purpose of maintaining the BHP within a predetermined range, hence ensuring the smooth operation of injection and a consistent injection rate. This approach prevents the need to reduce the injection rate in order to rectify BHP deviations. In this scenario, the `Python` can be used to extract the desired data, provided that the model meets our requirements. If the BHP exceeds the specified limit, `Python` will generate a warning, and the color of the injection rates point in the rate balance plot will be modified. Additionally, the case data will be accessible in the scenario cluster for further analysis.

4.4 Base Case Scenario

Due to the presence of multiple trapping mechanisms that act on different time scales, evaluating the storage capacity of deep saline aquifers is highly challenging. Geological uncertainty and the absence of geological characterization contribute to the situation's complexity. In conclusion, conservative estimates of the quantity of CO_2 that breaks the boundaries of an aquifer within a given time period must be specified, along with the effects of CO_2 leakage. On the other hand, *North Sea* aquifers may present numerous obstacles, such as complex geometries, fault zones, and fracture zones which may serve as leakage conduits. To fully comprehend the primary effects that must be accounted for in capacity and risk analysis, extensive simulations utilizing validated simulation tools are required.

Since the primary objective of the *Northern Lights* project is to start injecting 2 Mt per year for each well, the wells have already been drilled, and the actual locations of the wells are known, it has been decided to evaluate the primary objective of the *Northern Lights* project as the base case scenario using the most recent geo-data. Monitoring the injection operation with the keywords `CO2STORE` and `DISGAS` in order to measure the dissolved gas over an extended period of post-injection monitoring, which is accounted for until the beginning of the year 2500. The `THERMAL` keyword is used to instruct the simulator to account for the thermal changes of the reservoir after injecting CO_2 under the supercritical phase condition and an unexpected temperature conduit. To set the PVT flash calculations of the `Flow` simulator in the correct direction, the initial temperature of the reservoir must be set with the `RTEMP` keyword when `CO2STORE` is used. `RTEMP` keyword is set to 98 °C (see appendix A.1).

By extracting the pressure vector results in every time step using `Python` after running the simulation case, a specific investigation of the pressure values in the reservoir top layer (i.e., the layer below the cap rock) is conducted, and the maximum pressure in all time steps is accounted for to identify the occurrence of cap rock fracture. If the pressure in any cell or time step exceeds the pre-defined cap rock fracture pressure, a warning is displayed in the *'simulation report'* file noting that the pressure has exceeded the cap rock fracture pressure limit. Also, the

reservoir formation fracture pressure would undergo a pressure sensitivity analysis. In this case, the fundamental difference is that the reservoir pressure is taken into account when calculating the formation fracture sensitivity. In the *'simulation report'* file, a comparable warning would appear immediately after the cap rock fracture pressure status report. The cap rock fracture pressure is set to 400 bar in the base case.

The modules mentioned in the preceding Chapter 4 are investigated in every simulation run, regardless of the type of scenario discussed in this master's thesis.

Well Control Measures

In addition to the injection rate, the bottom hole pressure (BHP) limit is of the utmost significance in the topic of well control. Since the well injection rate is predefined for the base case scenario of 2 Mt per year, the pressure limit is the primary governing module for the injection operation and must be observed throughout. In the event of pressure build-up through *well 31/5-7*, *well 31/8-1*, or in the overall area of the field (known as field average pressure of the reservoir (FPR)), a preventive plan (ERP) would be automatically activated, and the primary ERP plan would be to decrease the injection rate until there is no pressure build-up in that module. In accordance with the geomechanical study data in appendix B.2, the pressure limit in the base case is set to 350 bar.

Well Placement

Since the most realistic evaluation possible is the target of this project, the well placement in the base case scenario is set to be the real locations of the wells which are mentioned in the previous chapter in section 3.4.1.

Leakage To The Penalty Area

As previously stated, CO_2 plume leakage to the penalty area is a crucial aspect of this endeavor. In this case, the CO_2 flow through the penalty area would be monitored inquisitively, and the results would be presented as curves of gas saturation and gas volume in the penalty zone, as well as numerical values in the simulation report file, for a precise review of the condition during the injection and post-injection period.

4.5 Rate Balance Scenario

As indicated in the previous section, the *Northern Lights* project intends to begin with an injection rate of 2 Mt per year. In addition, the outline of the project specifies that the injection rate for each is to be increased to 5 Mt per year if capacity is available [35]. In this scenario, we anticipate investigating the reservoir behavior and data for all rates, ranging from the minimum rate of 0.5 Mt per year

per well to the maximum rate of 5 Mt per year per well. This means that each well will be evaluated at 10 distinct injection rates ranging from 0.5 to 5 Mt of CO_2 per year. Given that each of the wells would be evaluated at 10 various rates, and that this evaluation would include both wells *31/5-7* and *31/8-1*, 100 cases would be examined to define this scenario.

Establishing 100 distinct simulation runs, creating a unique *'.DATA'* file for each case, running it, organizing the data, creating separate folders, and contributing them to a comprehensive study of the injection rate changes and their effects is a time-consuming undertaking with a high risk of human error. Digitalization and programming are highly-discussed topics that could aid us in defining this scenario in order to have the fewest possible mistakes and errors, the lowest possible cost of human resources, and highly organized results, such as defining a separate *'.DATA'* file for each case, creating a separate folder for each well control measure, organizing the results, and finally providing a comprehensive evaluation of all the cases and the best injection point.

Programming

Due to the reasons discussed above, it was decided to develop a `Python` program that performs all the intended stages and provides a comprehensive evaluation of the simulation process. Reviewing the numerous *ecl* package modules, such as `EclFile`, `EclKw`, `EclGrid`, and `EclSum`, is a time-consuming coding procedure. The initial step is to construct a flexible input deck that will be used to feed `Python` code. This code should contain a loop according to the desired well control measure and limit definition, take only the name of the original adjustable *'.DATA'* file that is created based on the actual field data, and generate the entire simulation process and results. To achieve this objective, adjustable string values were inserted into the input deck in order to generate new cases.

Database

In the first step, the code generates a scenario folder with the name *'SCENARIO CLUSTER'*, followed by a folder with the name *well 31/5-7* that denotes the injection rate. In each folder that is allocated to a specific injection rate for *well 31/5-7*, a folder for a specific injection rate for *well 31/8-1* is created. Therefore, all *'.DATA'* files and simulation results will be separated and organized for future research. The second step is to generate a separate *'.DATA'* file for each scenario case, followed by the third step of conducting the simulation using the `'nohup'` function, which ensures that all simulations run on a separate cluster without interruption. The final phase is extracting the data predefined in the code and desired for the project's objective.

On the actual well locations, a number of sensitivity studies were conducted to ascertain the upper limit of injection rates based on well control characteristics that were assumed to be realistic. A `Python` code was developed to implement the

desired measures in the actual location wells to determine the maximal injection rate that could be sustained for 30 years without leaking into the neighborhood.

Well Control Measures

Since the well injection rate is variable and predefined for the rate balance case scenario of 0.5-5 Mt per year, the pressure limit is the primary governing module for the injection operation and must be observed throughout. In the event of pressure build-up through *well 31/5-7*, *well 31/8-1*, or in the overall area of the field (FPR), a preventive plan (ERP) would be automatically activated, and the primary ERP plan would be to decrease the injection rate to keep the well pressure below the defined limit.

Python is instructed to generate a rate balance 2D plot after all simulation runs of the scenario have been completed and all results have been collected and organized based on the injection rate of *well 31/5-7* and *well 31/8-1* and the pressure build-up status of the wells or the overall reservoir. In this way, a color definition was established for every possible circumstance that could occur during simulations, making the evaluation of the rate balance much quicker and more intuitive. After displaying the rate balance plot, the next chapter provides a color definition to facilitate comprehension of the plot. This method facilitates the selection of a set of options and reduces the time required for the authorities to make a crucial decision regarding such a large project.

Well Placement

Since the most realistic evaluation possible is the target of this project, the well placement in the rate balance scenario is set to be the real locations of the wells which are mentioned in the previous section 3.4.1 in a similar way to the base case scenario.

4.6 Leakage Application Scenario

The last possible specification for this project is aligned with the possibility of leakage to the *Troll* gas field, which is of high importance and plays a crucial part in the final decision regarding well placement and CO_2 injection rate, and the injection period for the *Northern Lights* project. As mentioned in the previous Chapter 4, a penalty zone has been established in order to mark the boundaries of a potentially hazardous area due to the possibility of leakage. The objective of organizing a scenario to cover this crucial topic is to investigate the possibility of the current real locations and study additional locations to determine how various well placement locations may affect the leakage process. Even with the uncertainty surrounding the relocation of the wells, focusing on a specific region might help in the decision-making process and provide a broader perspective of the set strategies for making the best decision possible.

Well Placement and Relocation Logic

Additionally, higher rates have been tested in the green locations to determine whether the new location offers more advantages than the current location. According to the table 4.1, these cases are distinguished by *'R'* keyword. The primary objective is to define wells on the four principal sides of the actual well locations and to determine how the repositioning of each side affects the plume extension and the status of leakage through faults and fractures. Table 4.1 has been created as a naming system to organize this scenario case process. Certain identifiers are defined so that any special case can be identified by its name. And greater numbers indicate that the well's location was further along in that direction.

Database And Scenario Naming System

On the same path, an *'Excel'* file collecting a database including the results and details of all the tested cases is created. The information includes the case name, well location, pressure build-up in the wells, and `FPR` status, as well as a brief summary of the case's results. As an illustration, a brief line of the database file was imported in appendix A.4. In addition to the location of the well, the injection rate is also a significant factor, and various injection rates could be used to evaluate the cases. There are also cases that responded positively to the tests and were colored in green, indicating that the new location could prove suitable for the setting up of a well.

Well Control Measures

As is obvious, the purpose of this case is to investigate the worst-case scenarios that could occur and how to address and prevent them. Since then, the *NPV* coding has been conducted in the main `Python` code, which assigns a specific value for each ton of CO_2 that leaks into the penalty zone and calculates the total quantity of CO_2 that leaked into this area. This scenario has been examined due to the government has the ability to establish a specific penalty in the project definition in order to compel and encourage companies to take preventive measures more seriously and precisely select a well location option and well control measures.

Penalty Cost Settlement

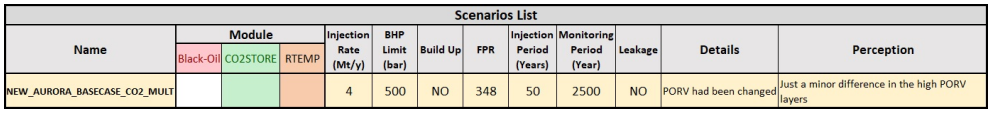
In a comparable manner, a specific case was investigated due to the possibility of a change in the injection period, reservoir capacity, and the potential to inject additional CO_2 for subsurface storage. Since then, in verdant cases where the results were positive, two injection periods of 30 and 50 years have been examined to assess the potential of different well locations.

In order to have a comprehensive view of the results and data, a Scenario nomenclature system was established and the data were compiled in an individual *'Excel'* file. Different sensitivity study scenarios have been studied to figure out if

there would be any leakage into the penalty area. Different injection rates were tested to determine the target zone's sensitivity to rate change and well pressure behavior.

Table 4.1: Relocation scenario naming system.

Element	**Module**
NEW_AURORA	Black-oil
BASECASE	Base Case
RW	Relocate wells to the West
RE	Relocate wells to the East
RS	Relocate wells to the South
RN	Relocate wells to the North
CO2	*CO2STORE* simulation module
P	Different pressure limit



Scenarios List												
Name	Module			Injection Rate (Mt/y)	BHP Limit (bar)	Build Up	FPR	Injection Period (Years)	Monitoring Period (Year)	Leakage	Details	Perception
	Black-Oil	CO2STORE	RTEMP									
NEW_AURORA_BASECASE_CO2_MULT				4	500	NO	348	50	2500	NO	PORV had been changed	Just a minor difference in the high PORV layers

Figure 4.1: Scenario cases database example.

4.7 Python Programming

`Python` is currently the most popular programming language, so utilizing it to organize and redefine simulations in order to provide a simple example of machine learning in order to review the entire project's ideas and scenarios based on the simulations' calculated data seems to be an excellent idea. By encoding the concepts, well control measures, candidate well locations, geological data, and desired actions in a programming language, time and effort could be saved while precision could be significantly enhanced. The recommended modules for automating the simulation procedure are `Python`, `FieldOPT`, and the desired simulators and visualizing interfaces.

Implemented Modules

The sole components of this module are `Python` and some supplied libraries for sorting output and viewing source code. The libraries are characterized as the *ecl* library for extracting simulation data and the *OS* library for modifying the variable string in *'.DATA'* files and instructing the operating system to execute simulations.

Penalty Area Definition

The top-right and the left-bottom limit indexes were below vectors for the penalty zone:

- Right limit: [115, 20, 9]
- Left limit: [144, 80, 13]

The penalty zone is specified by converting the extracted data from the simulation files to 3D vectors using the `EclGrid create 3d` module and then extracting the desired data by specifying the penalty zone's indexes (see figure 4.2).

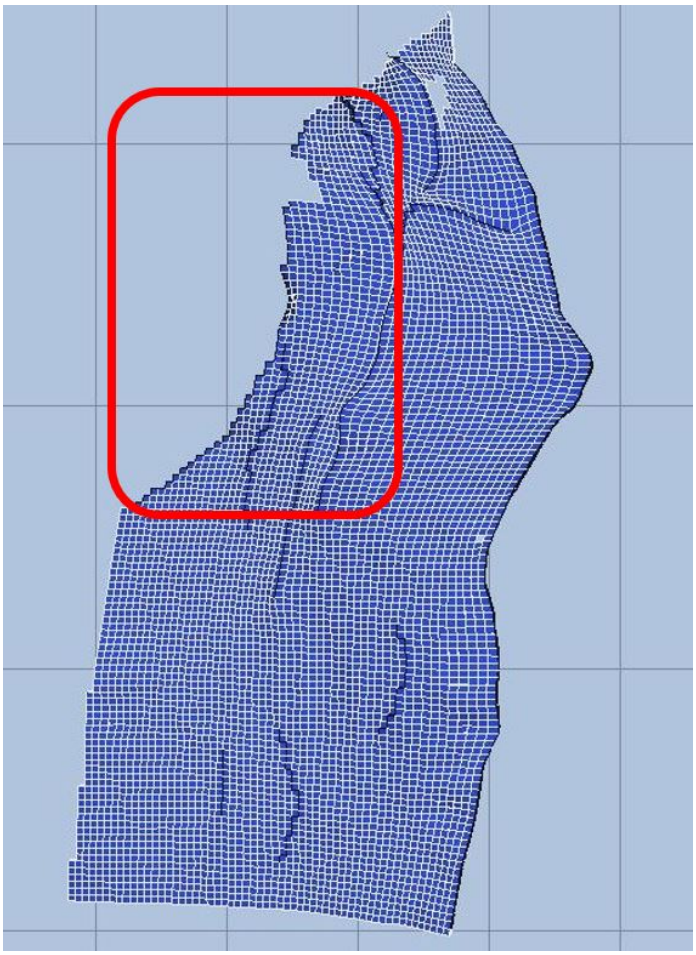


Figure 4.2: Overview of the penalty area in `ResInsight`.

Penalty Cost

Since the cost of the penalty should be preventative, it should be high to incentivize businesses (and optimization algorithms) to prevent such leakage. On the

other hand, the project risk assessment level should not be so high that companies lose motivation and interest in CCS projects as a newly emerging technology. Therefore, the penalty cost is presumed to be 25 USD or 275 NOK (according to the currency exchange rate on the date of publication) per ton of CO_2 released into the penalty area.

Penalty Settlement

The final penalty amount would be proportional to the quantity of CO_2 discharged and the unit-specific penalty cost. Multiplication of these two factors would reveal the operator company's final penalty, and additional curves might help in anticipating the leakage process and formulating preventative measures.

Data Matching

For each of the aforementioned scenarios, a compatibility test is conducted to ensure that the data provided by the simulator and the data calculated by the defined `Python` code match in terms of precision. It is essential to know how precise the data is and how we can rely on the calculated data.

4.8 Calculations

4.8.1 Dissolved Gas Mass Calculation

The dissolved gas ratio (R_s) of formation water volume at the standard condition is used to calculate the amount of gas dissolved in the water. Calculating the water volume under standard conditions by interpolating the formation volume factor (FVF) of bring into `Python` code. Multiplying the dissolved gas volume in the standard condition by the CO_2 density value in the standard condition is the final step. According to the [23] definition of R_s, the solution rate is the ratio of the surface gas from reservoir oil to the stock-tank oil from reservoir oil (see equation 4.1). Therefore, R_s calculations should be driven using the standard condition units, and the solution rate provided by the simulator in the `METRIC` unit system is $(\frac{Sm^3}{Sm^3})$. Consequently, the water volume on the surface should be calculated using the FVF interpolated from the PVT table data utilizing the written `Python` code (appendix A.3), and the R_s is subsequently used to extract the quantity of CO_2 dissolved in the aquifer. In the final step, the volume of dissolved CO_2 is converted to the mass unit using the *Span-Wagner* provided density of CO_2 in standard conditions [15].

$$R_{\mathrm{s}} = \frac{V_{\bar{g}\mathrm{o}}}{V_{\bar{o}\mathrm{o}}} \tag{4.1}$$

4.8.2 Trapped Gas Mass Calculation

The definition of irreducible gas saturation (S_{girr}) is the quantity of gas saturation in a porous medium at which gas begins to flow in the porous medium. According to data from the PVT tables of the *Eigestad* model [1] and as stated in the preceding chapter, the irreducible gas saturation is equal to 0.254. Then, in the calculations of the `Python` script, the volume of the porous medium with a gas saturation of less than 0.254 is accounted for as capillary trapped gas, whereas the volume which is equal to or greater than this volume is accounted for as free (mobile) gas in the reservoir.

Consequently, the volume of free and capillary trapped gas would be calculated in each time step, and the volume would be converted to the standard condition utilizing the CO_2 FVF interpolation (equation 4.2) script which the data in appendix B.2 is provided by *Span-Wagner* [15]. Finally, this volume is converted to CO_2 mass using the CO_2 density under standard conditions, and the resulting curves for each case are plotted.

$$B_{\mathrm{o}} = \frac{V_{\mathrm{o}}}{V_{\bar{o}\mathrm{o}}} \tag{4.2}$$

Chapter 5

Results

In this chapter, we present the results obtained from the simulations conducted for the alternative CO_2 injection scenarios at the *Aurora* field. The simulations were performed using the `Flow` simulator and the optimized scenario cluster organizer `Python` script. The results are analyzed and compared to assess the performance and effectiveness of each scenario in terms of CO_2 storage capacity, plume migration, pressure distribution, and economic viability.

According to the cores recovered from the reservoir unit, the *Johansen* formation has a high residual trapping capacity, and because of the geological heterogeneities, the CO_2 plume is anticipated to form effectively along the migratory path.

5.1 Base Case Scenario

5.1.1 Plots

Total Injected CO_2 (FGIT):

In every simulation run and scenario case, the total injected gas mass (FGIT) and the gas injection mass (GIT) for both wells are the first and most essential data plotted. The initial step in analyzing each scenario is to determine the GIT of each well and the FGIT of the scenario case. In figure 5.1, it can be seen that the total gas injected (GIT) of both wells is 60 Mt over 30 years, while the GIR of each well is 2 Mt per year. Due to both wells' equal GIR, the curves are overlapping, and only the curve of *Well 31/8-1* could be seen. Since there was no injection rate decline in this instance, the total FGIT is 120 Mt and the total GIR is 4 Mt per year.

Gas Saturation (S_g):

During the injection operation, the maximum gas saturation calculated rises abruptly from 0 to above 0.82 by the end of the injection period. Because the selected reservoir is an aquifer and not a depleted reservoir, the gas saturation begins at zero. It can be observed that the maximum gas saturation decreases significantly

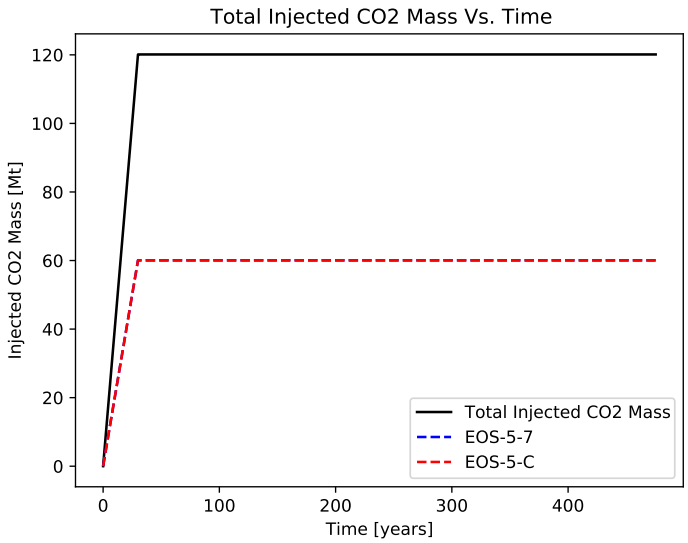


Figure 5.1: Total injected mass of CO_2 (FGIT) in the base case model.

after the end of the injection period (figure 5.2). That is primarily due to the CO_2 dispersion distribution during the initial phase of the monitoring period. As a result of the injection procedure stopping, there is no pressure source; therefore, the plume is distributed throughout the reservoir to stabilize the pressure.

The second stage of the decline in gas saturation is caused by the activation of the solution mechanisms. In addition, the gas saturation diagram reveals that the penalty zone has zero gas saturation, indicating that there is no CO_2 plume leakage into the penalty zone (see figures 5.3, 5.4, and 5.5).

Gas Dissolution Rate (R_s):

By activating the `DISGAS` module in the simulation method, the simulator provides the rate of gas dissolution in the saline aquifer (R_s). An R_s value greater than 0 indicates the presence of dissolved gas in the liquid phase. As the mentioned reservoir is an aquifer, no dissolved gas is anticipated in the aquifer before the onset of injection. Observing the aforementioned R_s curve reveals that the maximum solution ratio is 0 at the beginning of the CO_2 injection operation, while the R_s rate increases during the injection operation (figure 5.6).

During the post-injection monitoring period, the R_s rate does not diminish and remains stable at 26 $\frac{Sm^3}{Sm^3}$. The pressure equilibrium and convective mixing phenomena continue until the end of the monitoring period, indicating that the dissolution will continue for an extended period of time, and it is possible that the volume of the dissolved gas will surpass the volume of the free gas and trapped gas in the reservoir at a specific time after the injection operation (figures 5.7 5.8).

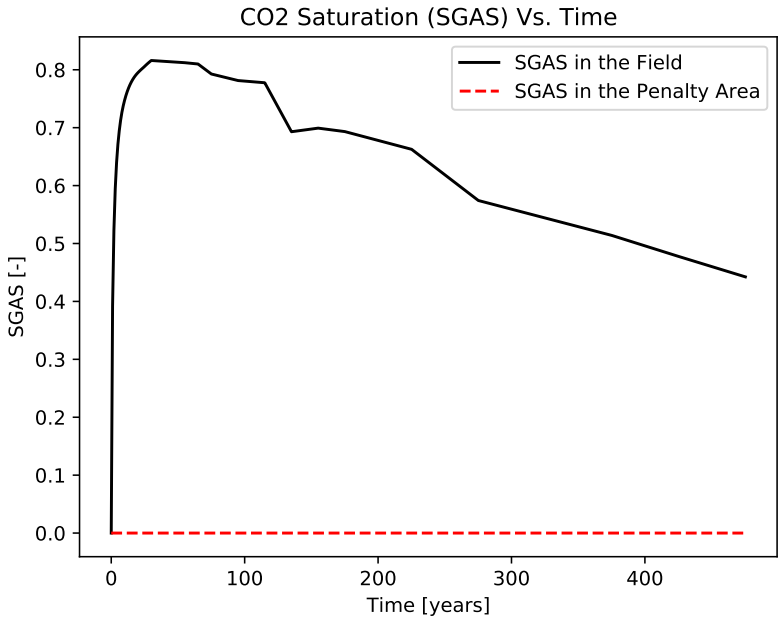


Figure 5.2: Maximum gas saturation (S_g) profile in overall the reservoir and the penalty area in the base case model.

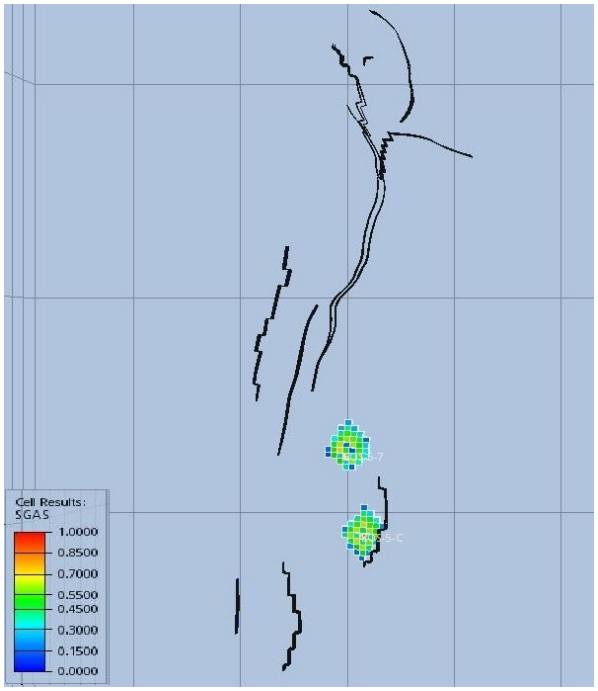
Captured CO_2 Mass By Various Mechanisms:

An extensive collection of plots is provided to compare the amount of CO_2 captured by various mechanisms in the reservoir (figure 5.9). According to the simulator and *Python* data extraction modules, three primary mechanisms could be explored. These captured gas conditions include:

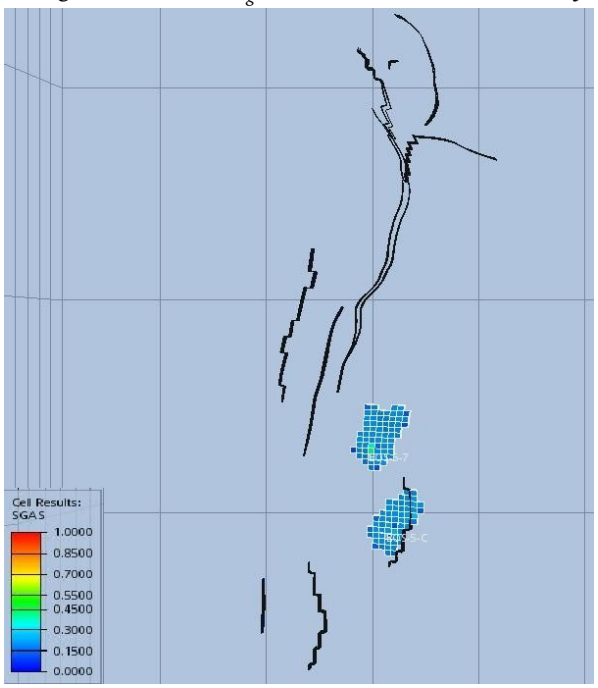
1. Free (mobile) gas
2. Structural trapped (irreducible) gas
3. Dissolved gas in the formation water

Free Gas And Trapped Gas:

Free gas refers to the quantity of gas that is mobile and could flow through the porous medium. In this model, the irreducible gas saturation is measured to be approximately 0.254. Consequently, gas grid cells with a higher gas saturation than 0.254 or 25.4% are regarded as unrestricted gas. By separating the gas saturations lower and higher than S_{girr} in the *Python* script, the amount less than S_{girr} gives the immobile gas (trapped gas), while the amount greater than S_{girr} is accounted as free gas.

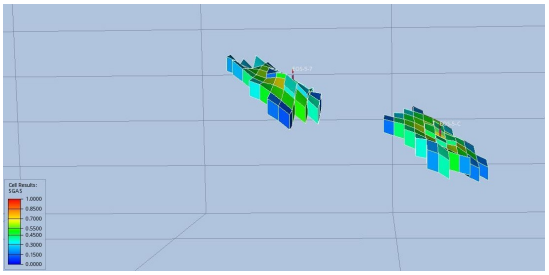


(a) Top side view of gas saturation (S_g) in the reservoir after the injection operation.

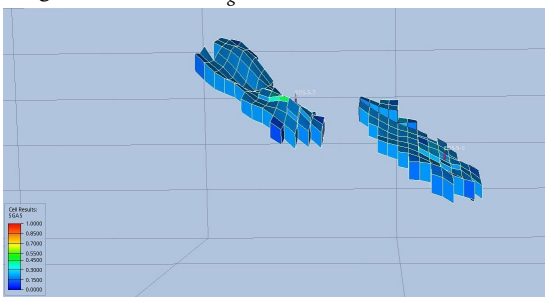


(b) Top side view of gas saturation (S_g) in the reservoir after the monitoring period.

Figure 5.3: Top side view of the gas saturation (S_g) in the reservoir in the base case scenario.

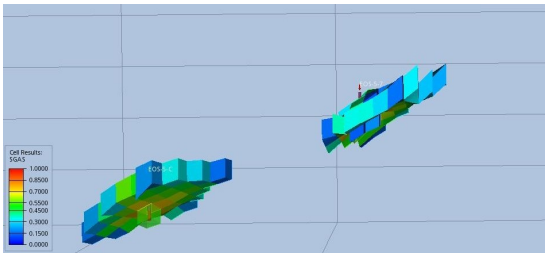


(a) West side view of gas saturation (S_g) in the reservoir after the injection operation.

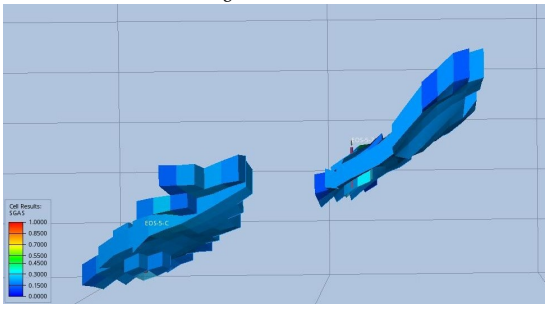


(b) West side view of gas saturation (S_g) in the reservoir after the monitoring period.

Figure 5.4: West side view of the gas saturation (S_g) in the reservoir in the base case scenario.



(a) East side view of gas saturation (S_g) in the reservoir after the injection operation.



(b) East side view of gas saturation (S_g) in the reservoir after the monitoring period.

Figure 5.5: East side view of the gas saturation (S_g) in the reservoir in the base case scenario.

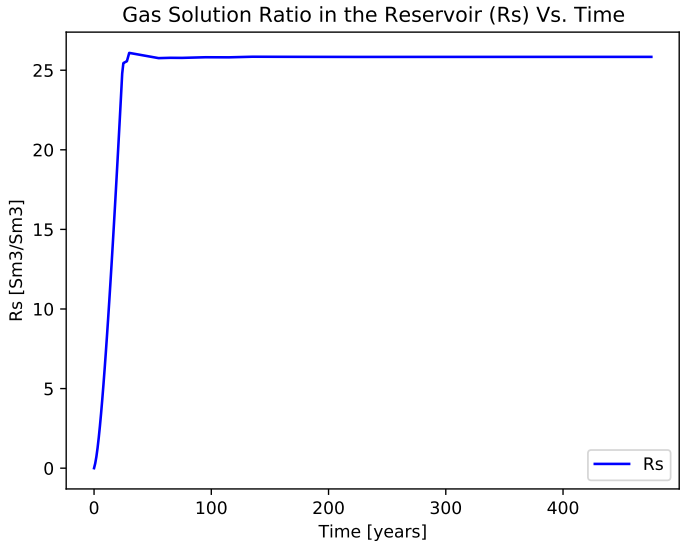
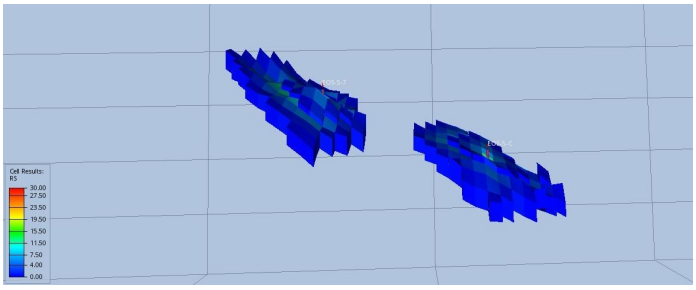
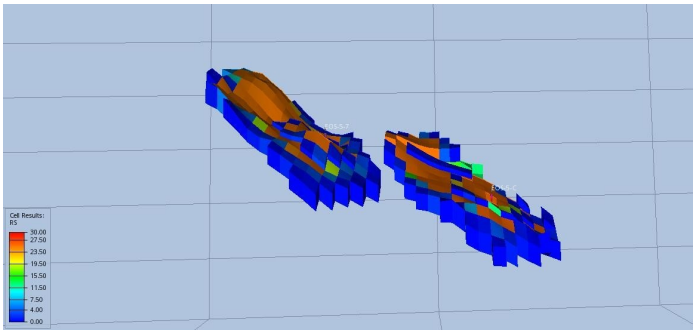


Figure 5.6: Maximum gas dissolution rate in water (R_s) ratio profile in the base case model.



(a) West side view of gas solution ratio (R_s) in the reservoir after the injection operation.

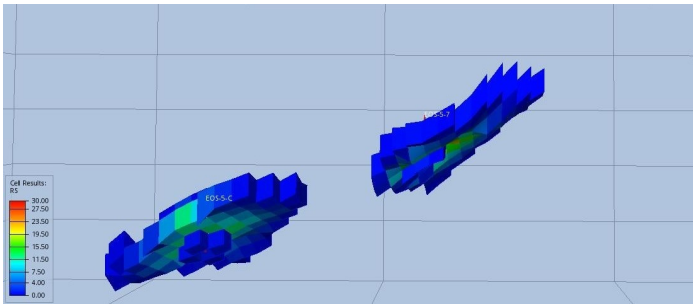


(b) West side view of gas solution ratio (R_s) in the reservoir after the monitoring period.

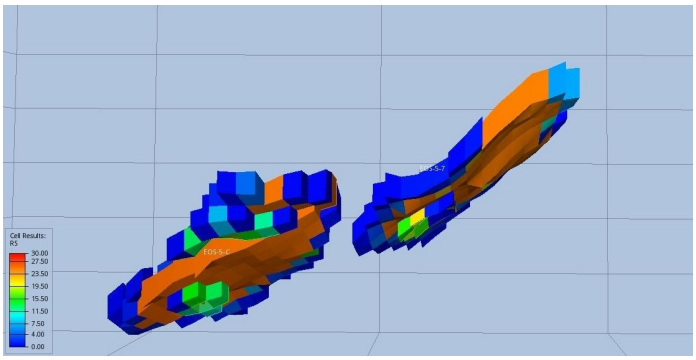
Figure 5.7: West side view of gas solution ratio (R_s) in the reservoir in the base case scenario.

Dissolved Gas:

An R_s value greater than 0 indicates the presence of dissolved gas in the liquid phase. Using a *Python* script, the brine volume of grid cells with R_s greater than 0



(a) East side view of gas solution ratio (R_s) in the reservoir after the injection operation.



(b) East side view of gas solution ratio (R_s) in the reservoir after the monitoring period.

Figure 5.8: East side view of gas solution ratio (R_s) in the reservoir in the base case scenario.

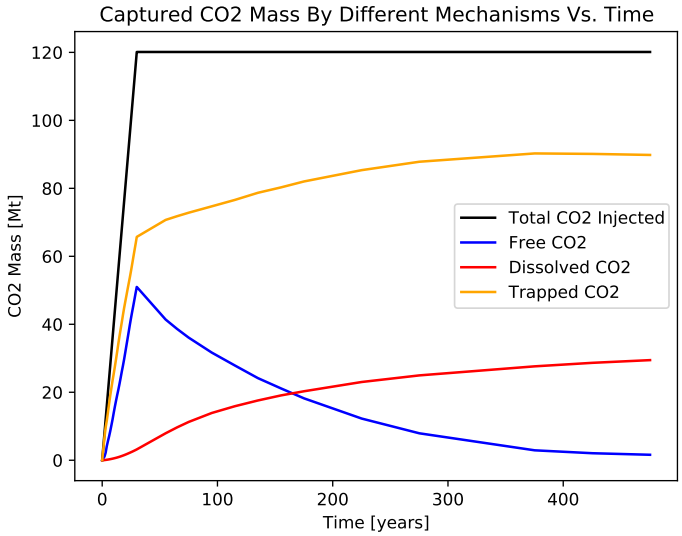


Figure 5.9: CO_2 mass captured in the reservoir by different mechanisms in the base case model.

is extracted; these values are then multiplied by the brine FVF, and the mass of gas

is determined using the R_s provided for the grid cell. Consequently, the quantity of dissolved gas in the reservoir is computed using the method described in Chapter 4.

CO2 Mass In The Reservoir And In The Penalty Zone:

For each proposed scenario, the CO_2 mass injected into the reservoir and the CO_2 mass released into the penalty zone are investigated. In the base case scenario, no CO_2 plume leakage into the prohibited zone is observed in figure 5.10.

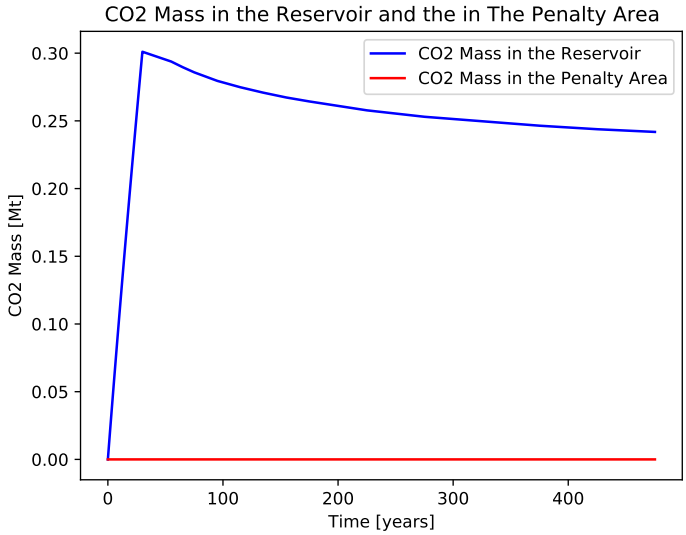


Figure 5.10: CO_2 mass leaked to the penalty zone in the leakage case model.

Data Compatibility Test:

As shown in figure 5.11, the amount of mass calculated by the `Python` code, which additionally separates the mass according to various trapping mechanisms, is compatible with the output provided by the simulator using the FGIT module. This demonstrates that there are no major errors in the calculations in the Python code.This plot serves as a function or compatibility test for the implemented method of coding and area restriction. There are minor differences between the two curves plotted in figure 5.11, which probably can be attributed to slight differences in FVF.

Maximum Pressure Enforced To The Cap Rock:

Figure 5.13 plots the maximum pressure towards the cap rock. In order to locate the point of maximum pressure, we must investigate the pressure contribution image generated by the simulator in the base case model (figure 5.13). Observing the top figure 5.12a reveals that the reservoir's cap rock is subjected to the most

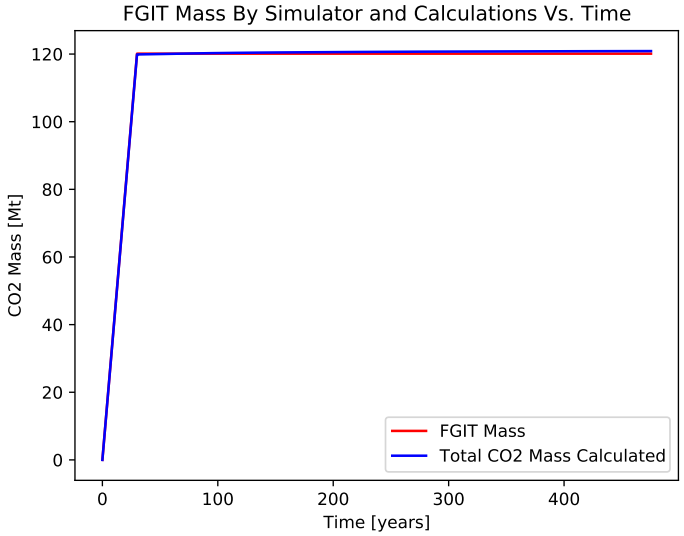


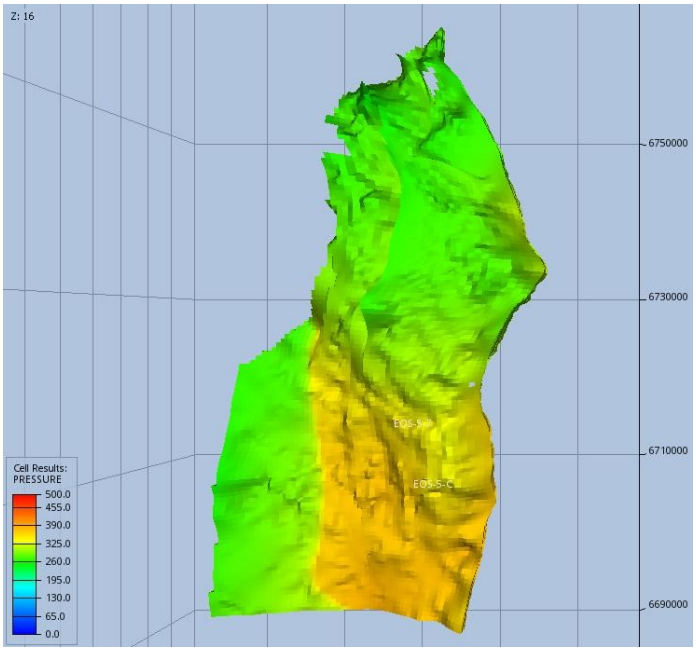
Figure 5.11: Data accuracy test by comparing the data gathered by `Python` by different calculations and the data received by the simulator itself in the base case model. The red curve is the simulator data and the blue curve is the data contributed by `Python`.

pressure in the reservoir's top layer near *Well 31/8-1* and predominately in the reservoir's southern region. Looking at figure 5.13 reveals that the pressure decreases abruptly after the injection operation ceases. By analyzing the graph in figure 5.13, it is evident that the cap rock will be subjected to a maximum pressure of 390 bar during the injection operation's final year.

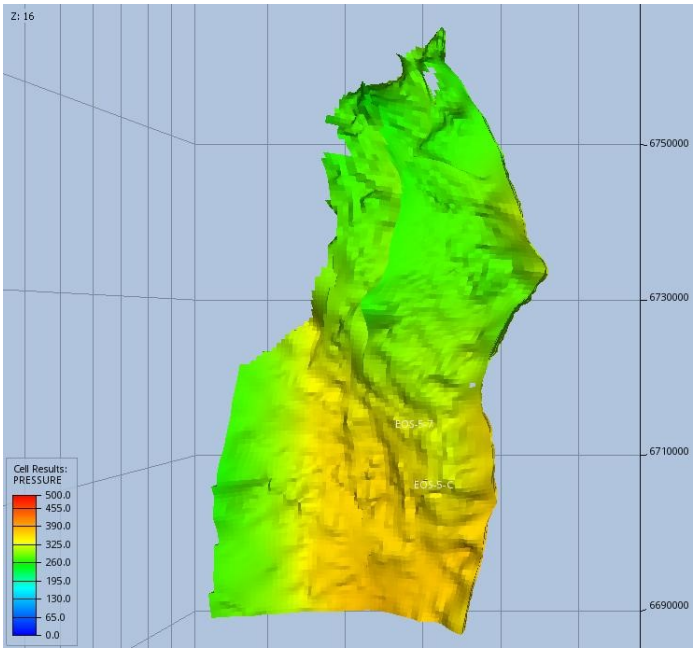
At the conclusion of the injection period, when the majority of the injected CO_2 has accumulated below the cap rock and gas dissolution has not played a significant role in the entrapment mechanism, the cap rock is subjected to the greatest pressure. The injected gas migrates upward due to its reduced density, increasing the pressure in the reservoir's highest layer. It is assumed that the maximum pressure exerted on the cap rock is close to 400 bar, which is greater than the 350 bar cap rock fracture pressure assumed in this study. As a result, this well placement scenario is hazardous in terms of the cap rock integrity.

After an equal period of injection operation, the reservoir's upper layer pressure would be equilibrated and stabilized at 378 bar. As can be seen at the conclusion of the monitoring period, the highest pressure is exerted on the cap rock along the southern boundaries (figure 5.12b).

As a result of the activation of various capturing mechanisms, such as structural trapping and convective mixing dissolution, the pressure in the top layer drops quickly after the injection operation has been stopped. After nearly the same amount of time as the injection period, the pressure decline swings to a horizontal line, and the pressure is stabilized at approximately 377 bar.



(a) Pressure distribution in the reservoir top layer after the injection operation.



(b) Pressure distribution in the reservoir top layer at the end of the monitoring period.

Figure 5.12: Pressure distribution in the reservoir top layer (below cap rock) in the base case scenario.

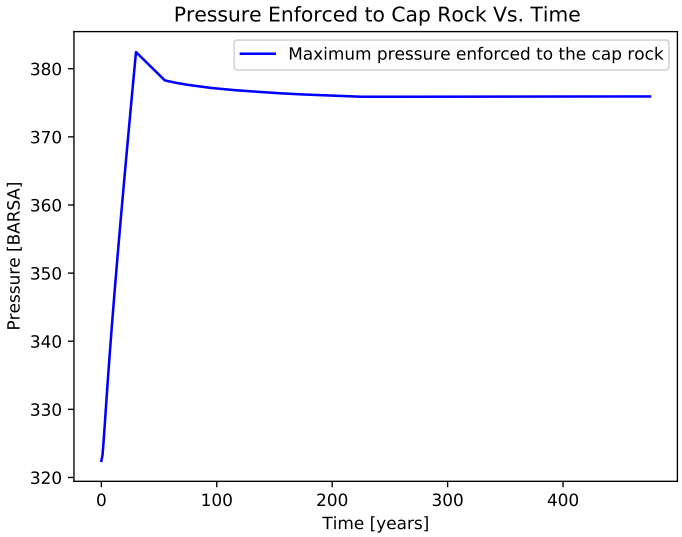


Figure 5.13: Maximum pressure enforced to the cap rock by the reservoir top layer in the base case model.

5.2 Rate Balance Scenario

After running the rate balance case which are created by the developed Python script, a plot is created to analyze the simulations. A colouring system is invented to separate the condition of each specific rate case. The colouring system could work according to each well control measure, but since the chosen well control measure in this project is set on the bottom hole pressure (BHP) limit, the colouring system is established according to the BHP situation of the wellbores, in addition to fracturing limit on the cap-rock pressure.

Since there are 2 wellbores and the cap-rock pressure situation, then there would be 6 different combinations of pressure limits that can occur depending on the injection operation. These cases are defined by a colour spectrum starting from the blue colour which means the most desired case to the red colour which means the worst case scenario. A full colour spectrum definition is shown in figure 5.14.

5.2.1 Rate Balance Plot:

The injection rate for each well was varied from 0.5 to 5 Mt/year with step size of 0.5 Mt/year, resulting in 10 cases for each well. This results in 100 simulation cases in total in the created scenario cluster. Figure 5.15 is the output of NPV code script which is run for the current well locations and the pressure limit of 350 bar for both well injection pressure and cap-rock fracture pressure (see appendix A.4).

As can be seen different colors are contributing to the rate balance plot in figure 5.15, the cases are from blue which means there is no pressure build-up in any of the wells, or in the overall field by the injected CO_2.

Color	Status
	Well and cap-rock pressures within set limits (best case)
	Well *31/5-7* reaching maximum injection pressure
	Well *31/8-1* reaching maximum injection pressure
	Cap-rock experiencing fracture pressure
	Well *31/5-7* and well *31/8-1* reaching maximum injection pressure
	Well *31/5-7* and cap-rock reaching max pressure
	Well *31/8-1* and cap-rock reaching max pressure
	Well *31/5-7*, well *31/8-1* and cap-rock reaching max pressure (worst case)

Figure 5.14: Rate balance plot color spectrum definition.

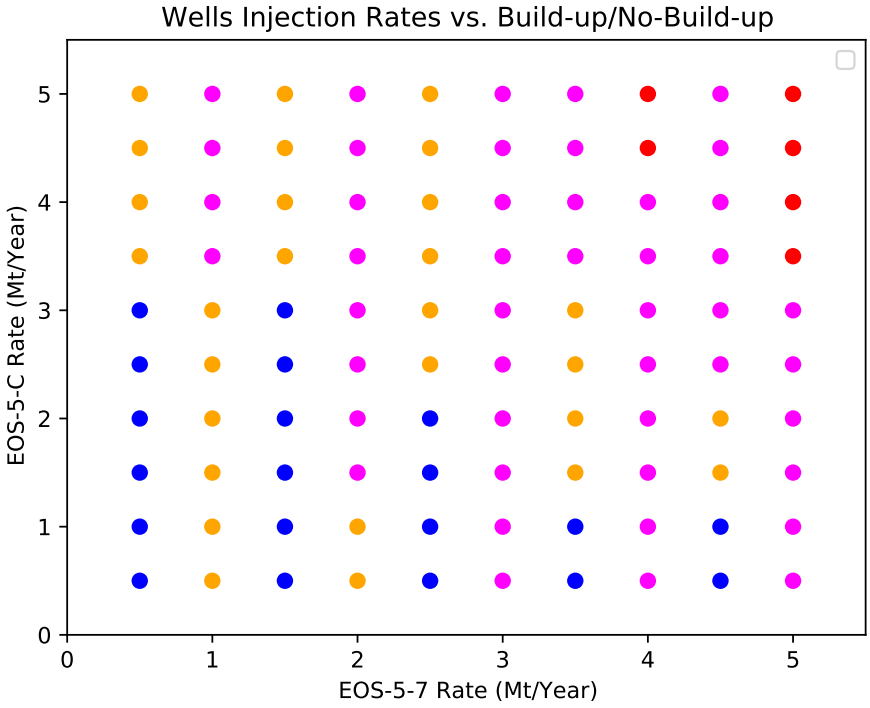


Figure 5.15: Well injection rate balance 2D plot for the BHP limit of 350 bar.

Desired Area of Rate Balance Plot:

To have a better analysis and selection procedure, a desired area should be defined to select the best case. As the aim of the project is the highest performance and profitability, higher rates of injection are preferred. This means that the trend should be to the end of both axes each one belongs to one of the wells injection rates(figure 5.16).

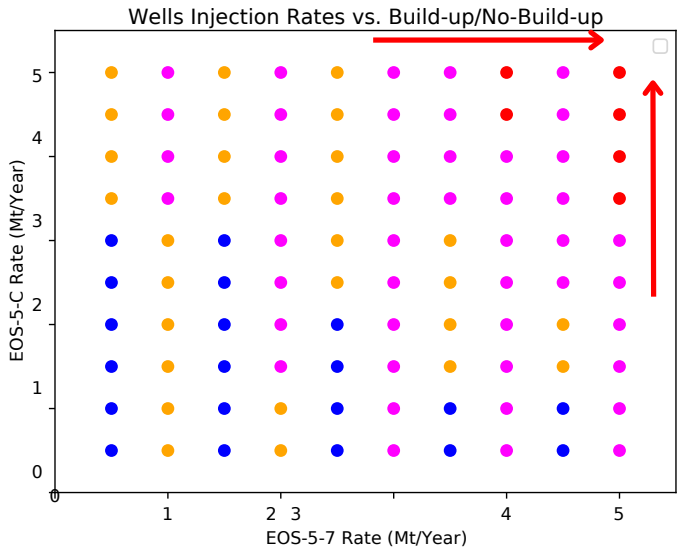


Figure 5.16: The desired trend in the well injection rate balance 2D plot case selection.

The Selected Case:

According to the specific condition of this case, we can observe that there is only one blue case in the green area located by a red circle in the green area in figure 5.17. This case would be chosen as the best case to be further studied and the data would be extracted from the scenario cluster which is already created by the *Python* script. The selected rates are 4.5 Mt/year for *Well 31/5-7* and 1 Mt/year for *Well 31/8-1*.

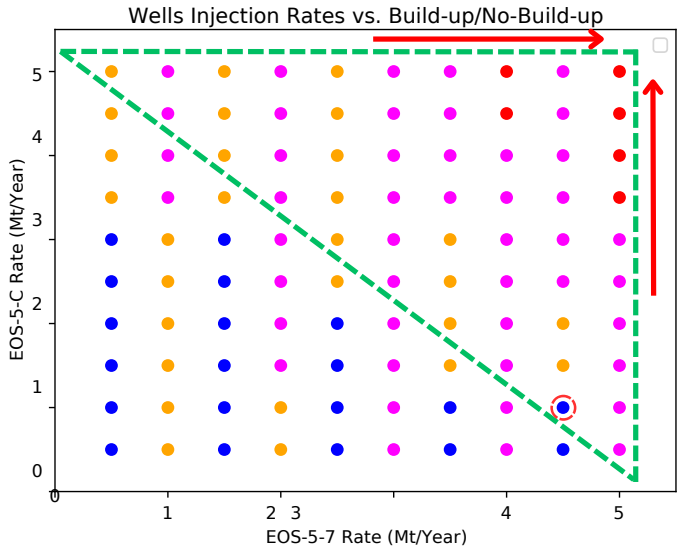


Figure 5.17: The selected case of the well injection rate balance scenario.

Similar Rates Analysis:

If there were two cases with similar injection rates, the next step would be checking the other well control measures such as cap rock fracture pressure, formation fracture pressure, leakage statement, and CO_2 plume shape.

5.2.2 Selected Rate Balance Case

As already mentioned, the case of 4.5 Mt/year for *Well 31/5-7* and 1 Mt/year for *Well 31/8-1* is selected as the optimum rate balanced case for the current state of the *Northern Lights* project to be further studied. The simulation report and the plots of the simulation are all existed in the scenario cluster which is already created by `Python`. A comprehensive evaluation of the case is then provided.

Total Injected CO_2 (FGIT)

As illustrated in figure 5.18, the total amount of CO_2 injected over the duration of the injection operation is 165 Mt. While the GIT for *Well 31/5-7* is equal to 4.5 Mt/Year during the entire injection operation, the GIT for *Well 31/8-1* is equal to 1 Mt/Year for the same time period. This results in a total injection rate of 5.5 Mt/Year, which is a substantial amount of CO_2. No pressure limit break has occurred, as the graph reveals neither a shutdown nor a rate decrease in the injection rates.

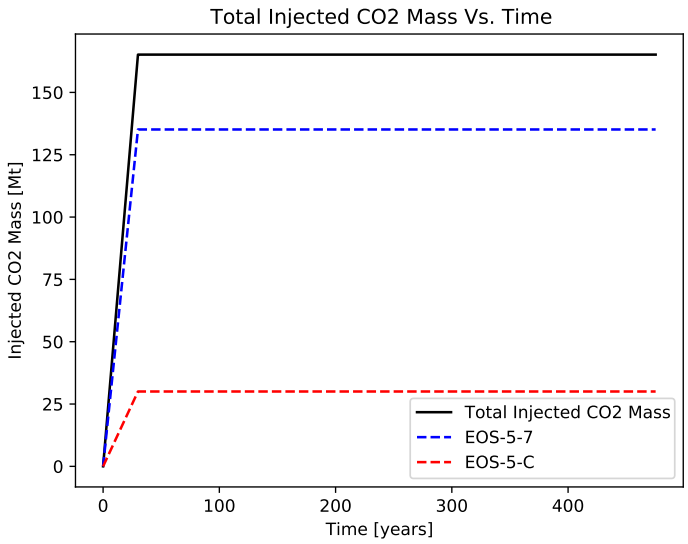
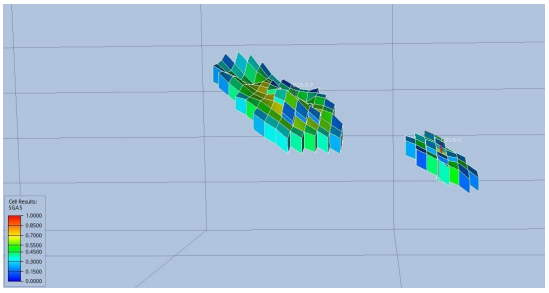


Figure 5.18: Total injected mass of CO_2 (FGIT) in the selected rate balance case model.

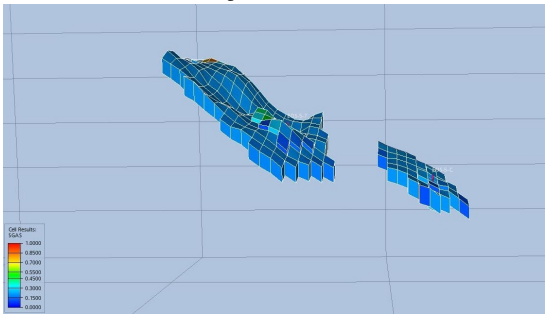
Gas Saturation (S_g):

Figure 5.22 demonstrates that the CO_2 plume does not cross *Fault4* and that there is no leakage into the prohibited zone. At the end of the injection period, the majority of the CO_2 concentration is in the zone close to the wellbore and in the reservoir's top layer. As previously mentioned, the first concentration of CO_2 would be below the cap rock and in the reservoir's uppermost stratum.

In the second stage, the pressure begins to equalize throughout the entire reservoir. At the end of the monitoring period, the CO_2 plume is observed to have expanded to a larger area of the reservoir's upper layers, with the majority of the expansion occurring on the north side of the reservoir, where the depth is shallower (see figure 5.20). Since the gas is lighter than the liquid brine, the concentration would be primarily in the reservoir's higher zones (figures 5.19 and 5.21).

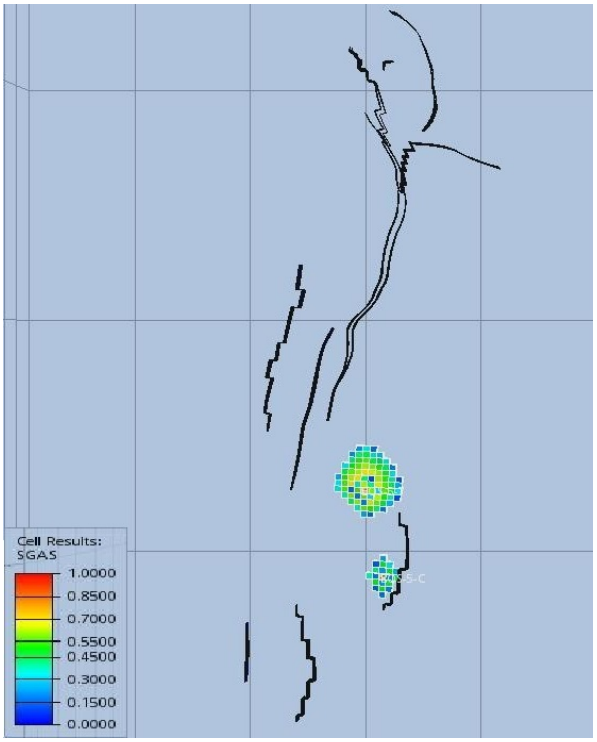


(a) West side view of gas saturation (S_g) in the reservoir after the injection operation.

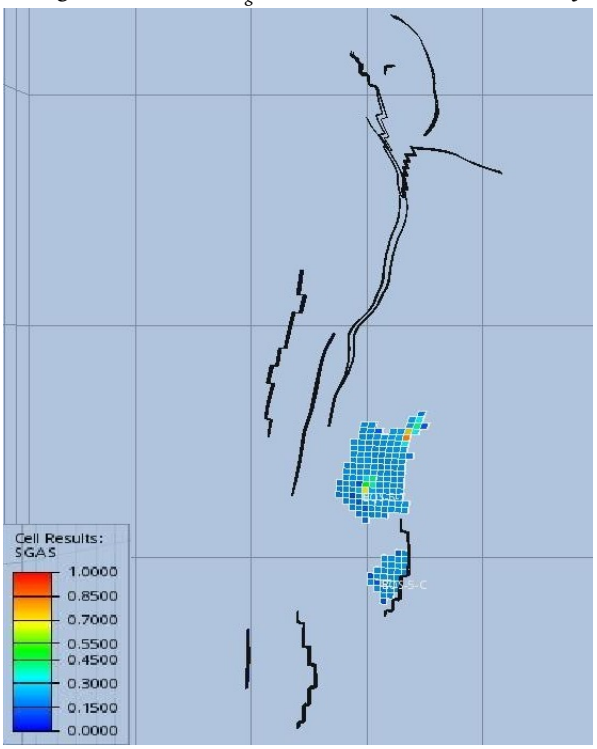


(b) West side view of gas saturation (S_g) in the reservoir after the monitoring period.

Figure 5.19: West side view of the gas saturation (S_g) in the reservoir in the selected rate balance scenario.

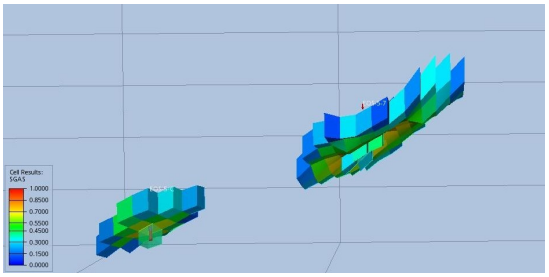


(a) Top side view of gas saturation (S_g) in the reservoir after the injection operation.

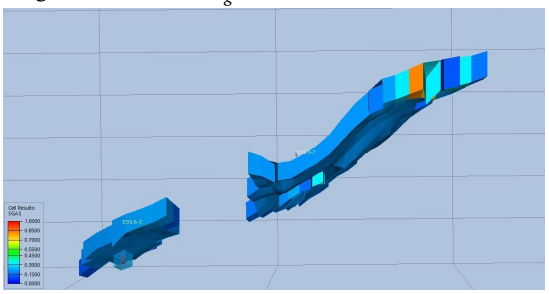


(b) Top side view of gas saturation (S_g) in the reservoir after the monitoring period.

Figure 5.20: West side view of the gas saturation (S_g) in the reservoir in the selected rate balance scenario.



(a) East side view of gas saturation (S_g) in the reservoir after the injection operation.



(b) East side view of gas saturation (S_g) in the reservoir after the monitoring period.

Figure 5.21: East side view of the gas saturation (S_g) in the reservoir in the selected rate balance scenario.

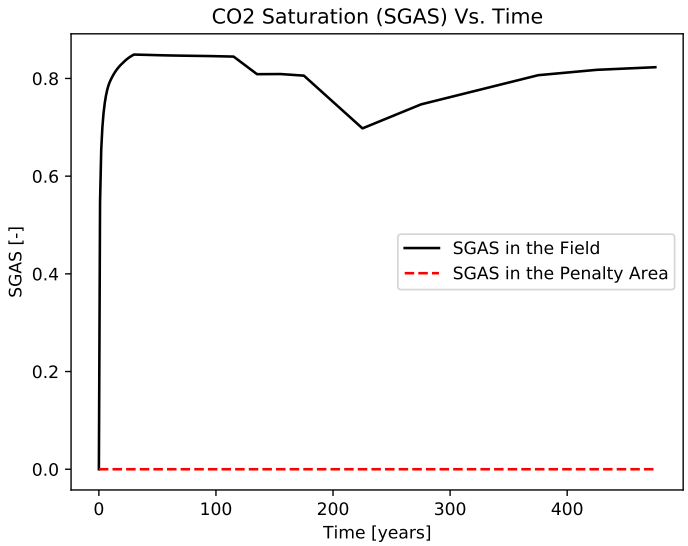


Figure 5.22: Maximum gas saturation (S_gas) profile in overall the reservoir and the penalty area in the selected rate balance case model.

Gas Dissolution Rate (R_s):

The maximum gas dissolution ratio, in this case, is the most stabilized among the other investigated cases. At the start of the injection operation, R_s is equal to 0

and it rises sharply simultaneously with the injection start. By stopping the injection operation, maximum R_s would be increased and affected by the accumulated pressure in the reservoir, and after reaching the value of around 27 $\frac{Sm^3}{Sm^3}$ it stabilizes till the end of the monitoring period. No decrease in the rate of gas dissolution reveals that the dissolution process continues for a very long time period after the monitoring times end (figure 5.23).

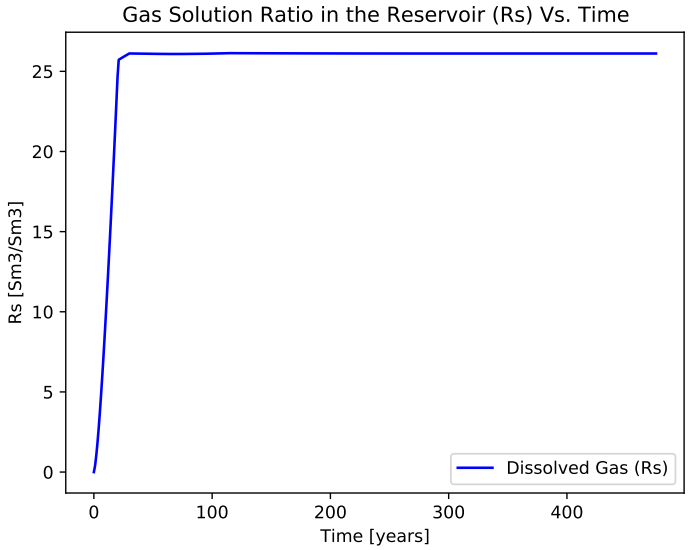


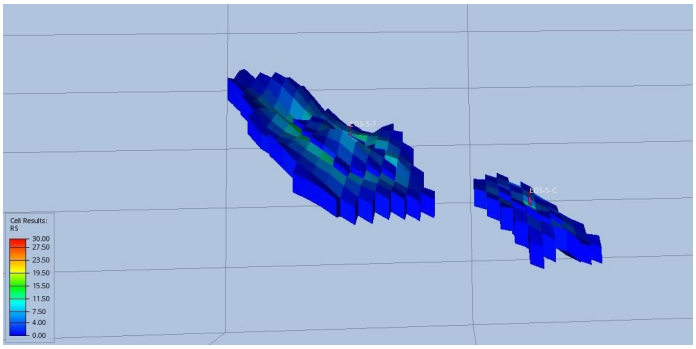
Figure 5.23: Maximum gas dissolution rate in water (R_s) profile in the selected rate balance case model.

The region that is closest to the well contributes the most to the gas dissolution ratio in the aquifer. The majority of dissolution can be observed around *well 31/8-1* and on the north and east sides of the wells toward the eastern boundary where `FAULT4` is located (figures 5.24 5.25).

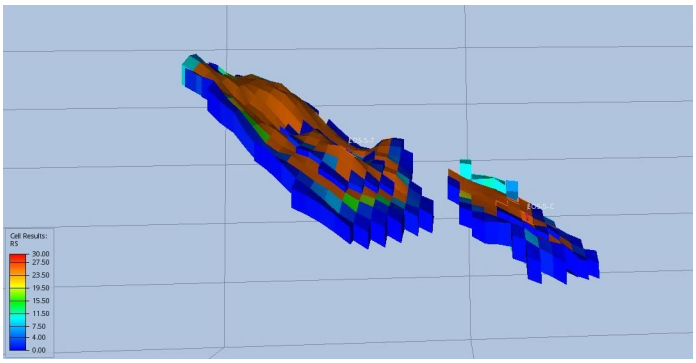
Captured CO_2 Mass By Various Mechanisms:

Free Gas And Trapped Gas:

At the end of the monitoring period in 2500, more than half of the captured CO_2 in the reservoir will have been captured by the structural trapping mechanism, contributing nearly 110 Mt to the total captured CO_2 mass in the reservoir (figures 5.26). At the conclusion of the injection period, the amount of free gas in the reservoir surges to nearly 70 Mt. However, as time passes, this free gas transforms into various varieties of captured gas, such as structurally trapped gas and dissolved gas in the brine.



(a) West side view of gas solution ratio (R_s) in the reservoir after the injection operation.



(b) West side view of gas solution ratio (R_s) in the reservoir after the monitoring period.

Figure 5.24: West side view of gas solution ratio (R_s) in the reservoir in the selected rate balance case scenario.

Dissolved Gas:

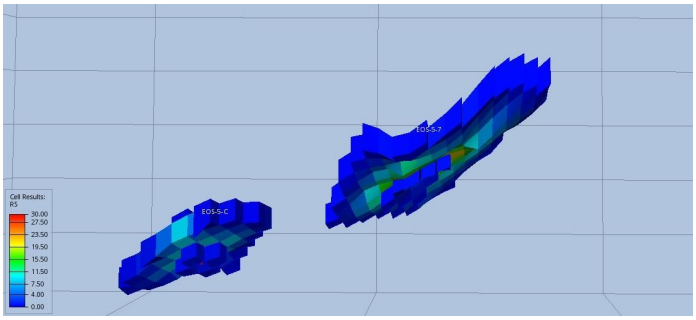
Part of the remaining portion of the gas dissolves in the formation of water and becomes dissolved gas. Due to the time-consuming nature of the dissolution process, the amount of dissolved gas is low during the injection operation. With time, the quantity of dissolved gas in the reservoir would reach 40 Mt of CO_2 mass, which is greater than one-third of the total CO_2 mass injected.

CO2 Mass In The Penalty Zone:

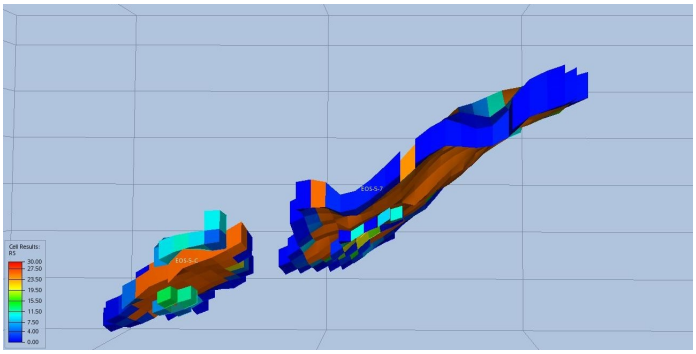
As is evident from the diagram, there is no leakage in the penalty zone, making this rate-balanced case suitable from a leakage-importance standpoint (see figure 5.27).

Data Compatibility Test:

The data compatibility test graph in figure 5.28 demonstrates the accuracy of the data calculated by `Python`.



(a) East side view of gas solution ratio (R_s) in the reservoir after the injection operation.



(b) East side view of gas solution ratio (R_s) in the reservoir after the monitoring period.

Figure 5.25: East side view of gas solution ratio (R_s) in the reservoir in the selected rate balance case scenario.

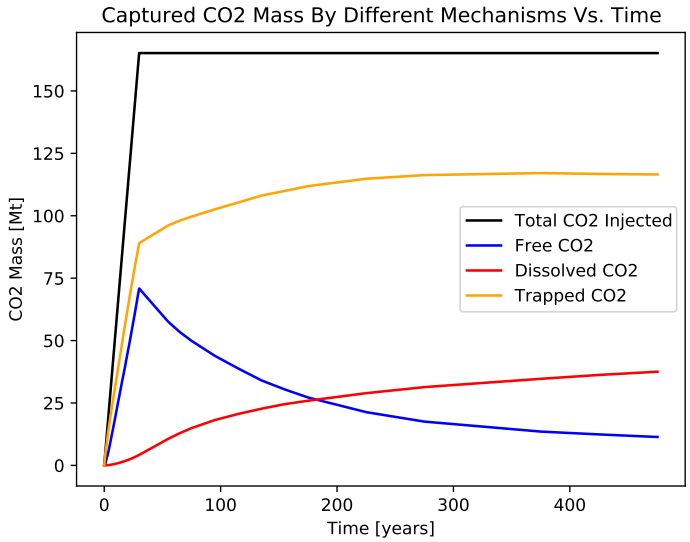


Figure 5.26: CO_2 mass captured in the reservoir by different mechanisms in the selected rate balance case model.

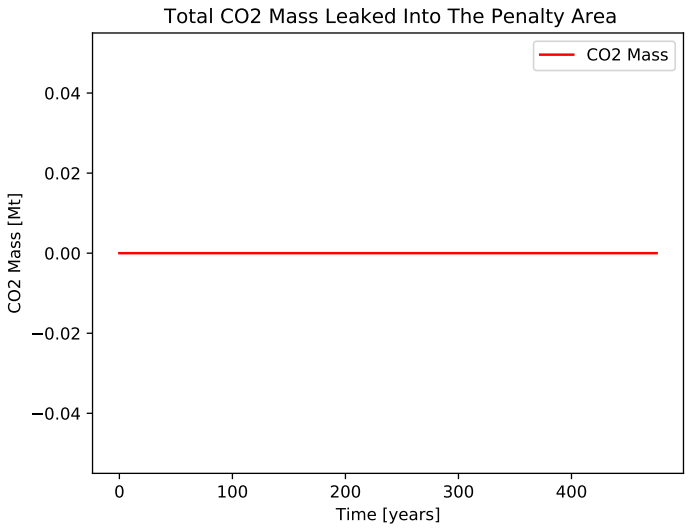


Figure 5.27: CO_2 mass leaked to the penalty zone in the selected rate balance case model.

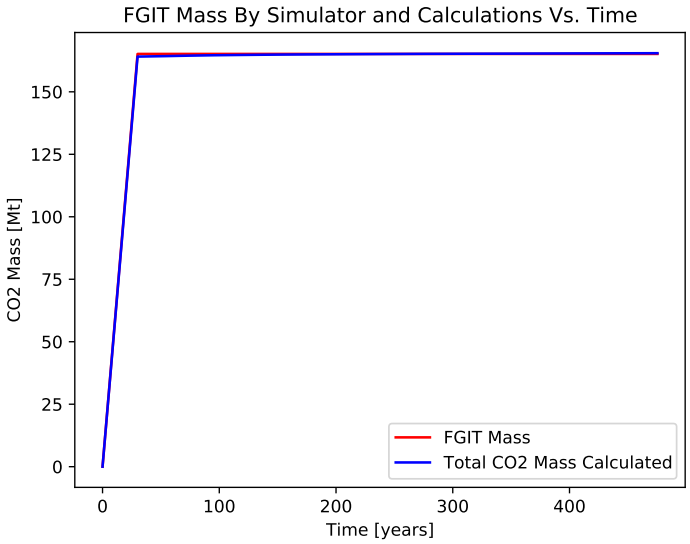
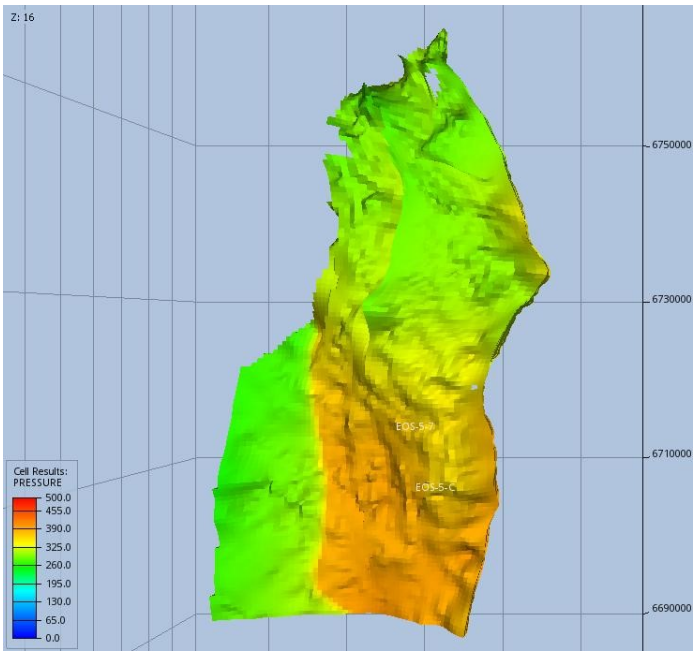


Figure 5.28: Data accuracy test by comparing the data gathered by `Python` by different calculations and the data received by the simulator itself in the selected rate balance case model. The red curve is the simulator data and the blue curve is the data contributed by `Python`.

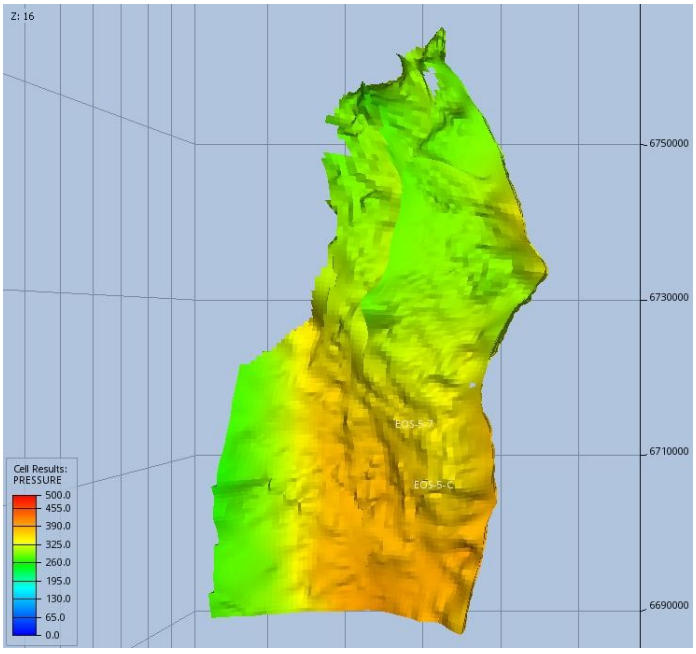
Maximum Pressure Enforced To The Cap Rock:

The pressure profile graph in the reservoir's uppermost layer in figure 5.30 reveals that the greatest pressure below the cap rock barely reached 400 bar, which is the cap rock fracture pressure, and then decreased gradually after the injection operation was halted in 2055. The pressure eventually stabilizes at 397 bar until

the end of the monitoring period.



(a) Pressure distribution in the reservoir top layer after the injection operation.



(b) Pressure distribution in the reservoir top layer at the end of the monitoring period.

Figure 5.29: Pressure distribution in the reservoir top layer (below cap rock) in the selected rate balance case scenario.

Figure 5.29 depicts the pressure distribution along the southern boundary of the reservoir using `ResInsight`. Notably, it is evident that the pressure average after the monitoring period is considerably lower than at the end of the injection period, as long-term mechanisms tend to begin only after a considerable amount of time and contribute to reducing the amount of free gas in the reservoir, particularly in the top layer.

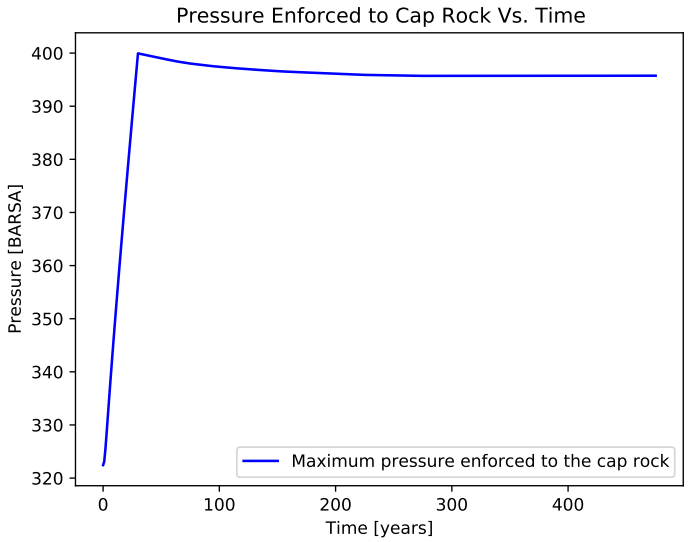


Figure 5.30: Maximum pressure enforced to the cap rock by the reservoir top layer in the selected rate balance case model.

5.3 Penalty Leakage Scenario

Well Placement

As described previously in Chapter 4, numerous well locations and well placements were tested for penalty leakage to determine the status of leakage into the prohibited zone. Investigations reveal that the western locations closest to the faults, particularly `FAULT4`, are the most susceptible to discharge. Eventually, case *'RW6'*, which indicates that the location was shifted six times west of the actual well locations, contributed to leakage in the penalty zone.

These are the *'RW6'* well location indexes:

- *Well A*: [109, 92, 9] to [109, 92, 13]
- *Well B*: [106, 106, 9] to [106, 106, 13]

Injection Rate

For this scenario case, the injection rates selected are 3.5 Mt/Year for *Well A* and 4.5 Mt/Year for *Well B*. This indicates that the total injection rate for the scenario

case is 8 Mt/year.

5.3.1 Plots

Total Injected CO_2 (FGIT):

According to the *FGIT* profile curve, the total amount of CO_2 injected is 240 Mt, with *Well A* contributing 105 Mt and *Well B* contributing 135 Mt. There is no shutdown or decrease in rates observed in figure 5.31.

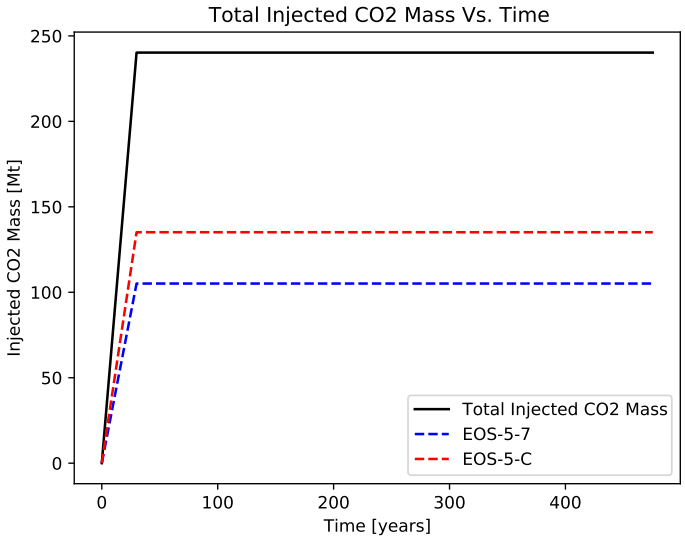


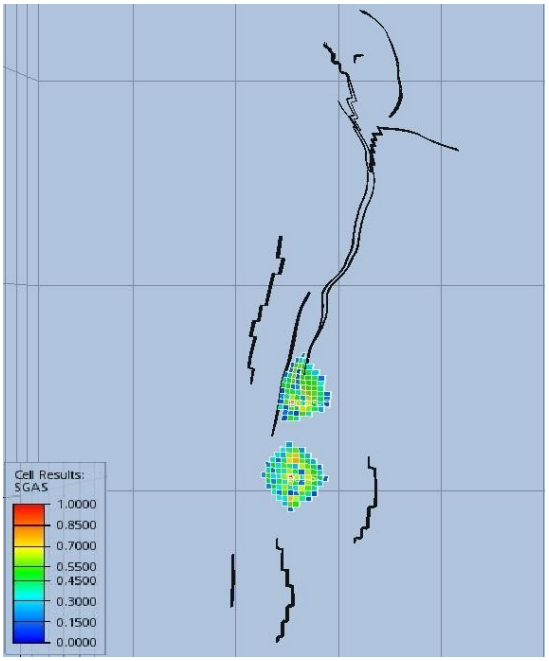
Figure 5.31: Total injected mass of CO_2 (FGIT) in the penalty leakage case model.

Gas Saturation (S_g):

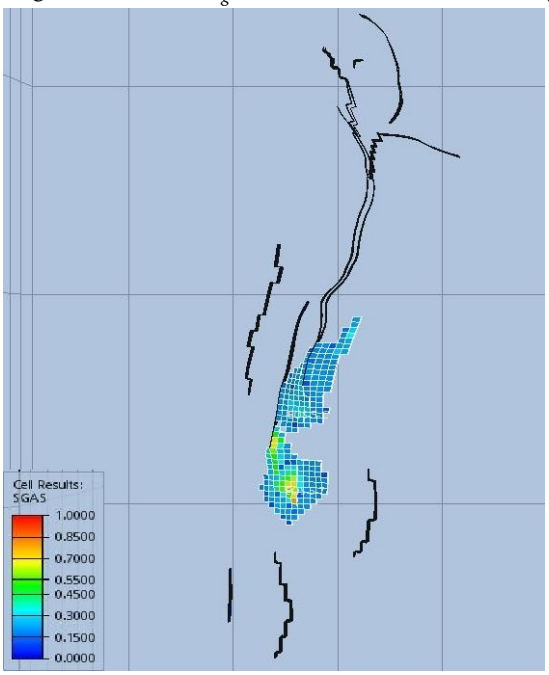
According to the maximum gas saturation diagram, there would be a significant amount of leakage into the penalty zone. At the end of the injection operation, the maximum S_g would be increased to 0.6 while the total S_g in the reservoir is close to 0.87. This means that a significant quantity of CO_2 would leak into the penalty area, as depicted in figure 5.40.

The gas saturation behavior is regular, but the first stage of free gas saturation decrease would be sluggish due to a large volume of trapped gas, while the second stage relating to the dissolution process occurs more rapidly (figure 5.35).

`OPM` data indicate that the rate of leakage is significantly higher after injection operations cease than at the conclusion of the monitoring period. This occurs as a result of the extension and pressure equilibrium of the CO_2 plume following the injection period (figures 5.33 and 5.34). Examining the gas saturation distribution in the reservoir would reveal that the plume tends to expand to the north of the reservoir in order to equalize pressure (see figure 5.32).

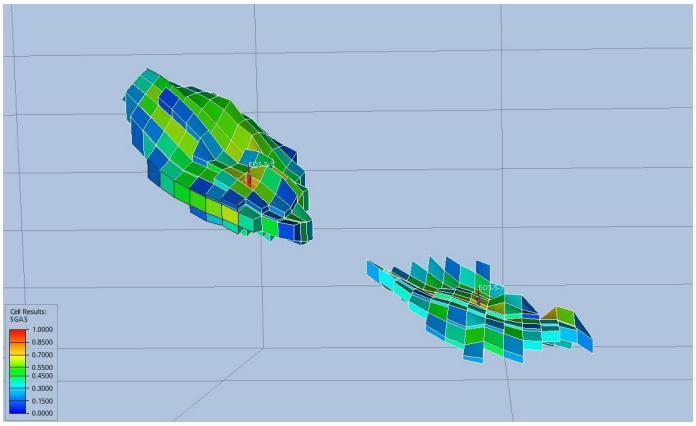


(a) Top side view of gas saturation (S_g) in the reservoir after the injection operation.

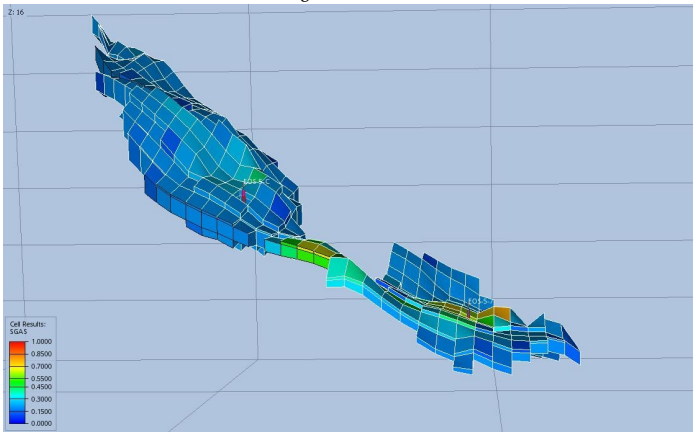


(b) Top side view of gas saturation (S_g) in the reservoir after the monitoring period.

Figure 5.32: Top side view of the gas saturation (S_g) in the reservoir in the penalty leakage scenario.

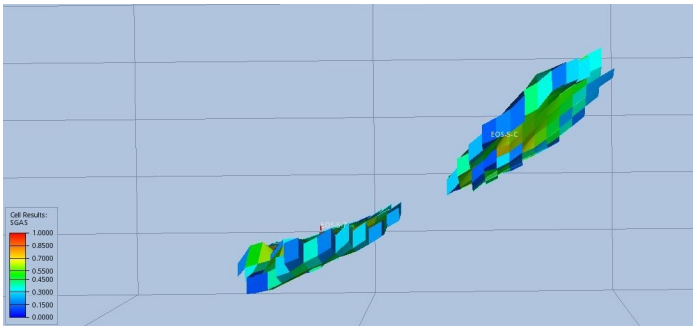


(a) West side view of gas saturation (S_g) in the reservoir after the injection operation.

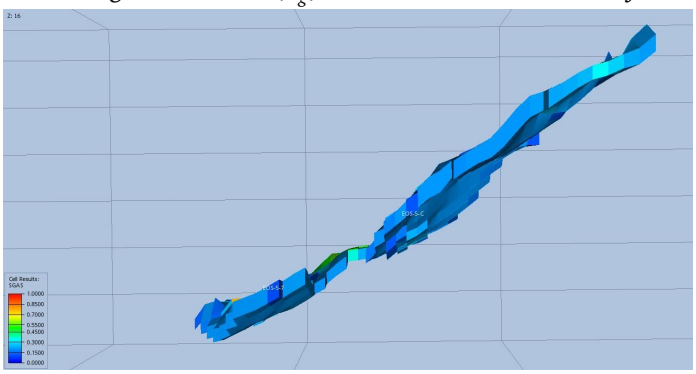


(b) West side view of gas saturation (S_g) in the reservoir after the monitoring period.

Figure 5.33: West side view of the gas saturation (S_g) in the reservoir in the penalty leakage scenario.



(a) East side view of gas saturation (S_g) in the reservoir after the injection operation.



(b) East side view of gas saturation (S_g) in the reservoir after the monitoring period.

Figure 5.34: East side view of the gas saturation (S_g) in the reservoir in the penalty leakage scenario.

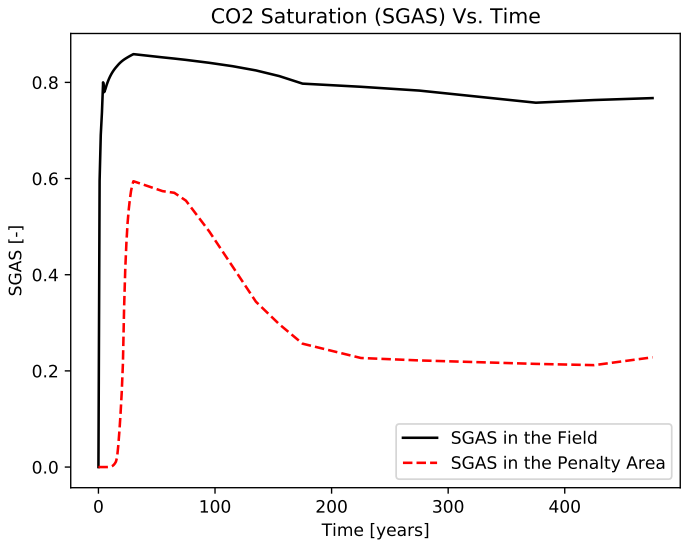


Figure 5.35: Maximum gas saturation (S_g) profile in overall the reservoir and the penalty area in the leakage case model.

Gas Dissolution Rate (R_s):

Figure 5.36 illustrates a sharp increase in the maximum gas dissolution ratio at the beginning of the injection operation to 28 $\frac{Sm^3}{Sm^3}$, followed by a minor decrease post the injection operation. The maximum gas dissolution ratio eventually stabilizes at 27 $\frac{Sm^3}{Sm^3}$ until the end of the monitoring period.

The images of the gas dissolution factor 5.38 reveal that dissolution is primarily concentrated near wellbores. Notably, the northern side of the CO_2 plume contains a high concentration of dissolved gas as well. This occurs because the formation water and the injected CO_2 are at the same level, making it simpler and quicker for the gas to dissolve into the aquifer (figure 5.37).

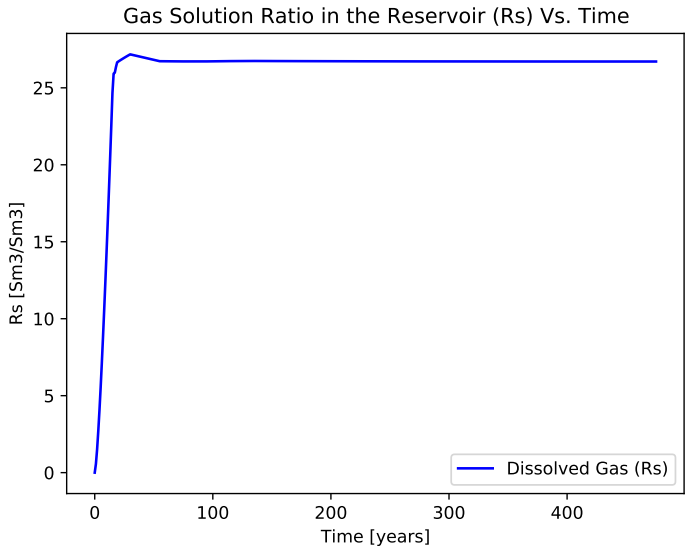
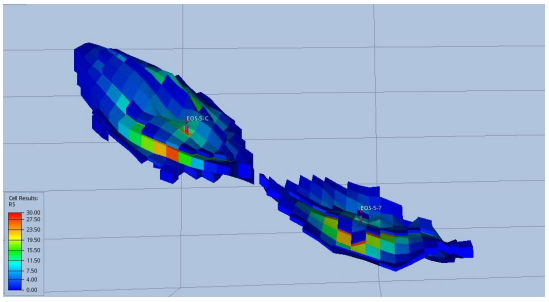
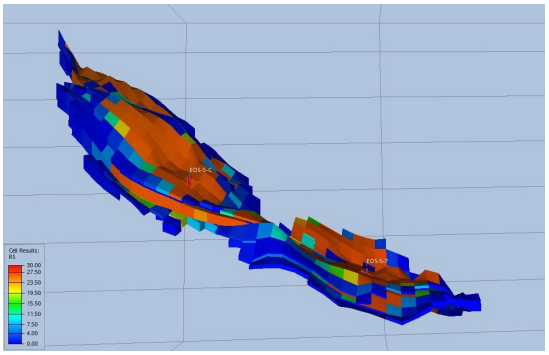


Figure 5.36: Maximum gas dissolution rate in water (R_s) profile in the leakage case model.

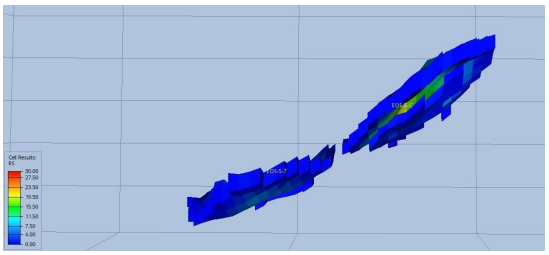


(a) West side view of gas solution ratio (R_s) in the reservoir after the injection operation.

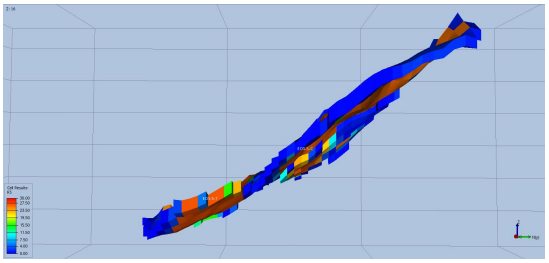


(b) West side view of gas solution ratio (R_s) in the reservoir after the monitoring period.

Figure 5.37: West side view of gas solution ratio (R_s) in the reservoir in the penalty leakage scenario.



(a) East side view of gas solution ratio (R_s) in the reservoir after the injection operation.



(b) East side view of gas solution ratio (R_s) in the reservoir after the monitoring period.

Figure 5.38: East side view of gas solution ratio (R_s) in the reservoir in the penalty leakage scenario.

Captured CO_2 Mass By Various Mechanisms:

In a manner comparable to other scenario cases, the structural trapping mechanism dominates the CO_2 capture in the reservoir. It contributes nearly 160 Mt of the 240 Mt of CO_2 captured at the end of the monitoring period or 65% of the total CO_2 captured. The second main mechanism is the dissolution of 50 Mt of injected carbon dioxide (see figure 5.39).

Ultimately, the free gas that comprises the majority of the reservoir at the start of the injection operation would only account for 20 Mt or less than 10% of the total quantity. This guarantees the safety of the CCS because it reduces the mobility of the CO_2 plume and the likelihood of leakage to the surface.

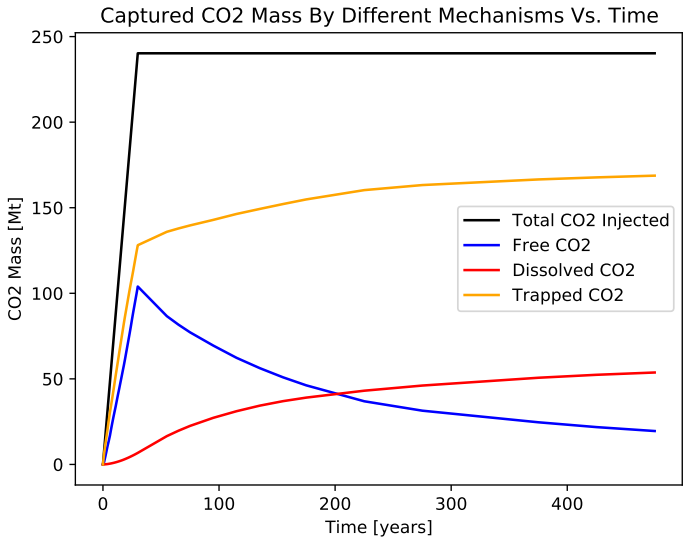


Figure 5.39: CO_2 mass captured in the reservoir by different mechanisms in the leakage case model.

CO_2 Mass In The Penalty Zone:

The most relevant plot, in this case, is the CO_2 mass leaked into the penalty area, see figure 5.40. The graph depicts an abrupt increase in the amount of CO_2 leaking into the penalty zone during injection, indicating that the leakage has been occurring since the beginning of the injection period.

Leakage would occur at the same rate as during the injection period until 20 years after the injection period. This would attain the maximum amount of CO_2 mass leaked into the penalty area for the duration of the project, which is 6.8 Mt. Then, the CO_2 plume mass in the penalty region decreases until the year 2250, as a result of the CO_2 plume pressure equalization process that occurs over the course of these years.

By activating the dissolution process, the quantity of gas in the penalty would increase marginally, indicating that the penalty area is a suitable location for the

CO_2 plume to settle as a dissolved gas in the aquifer.

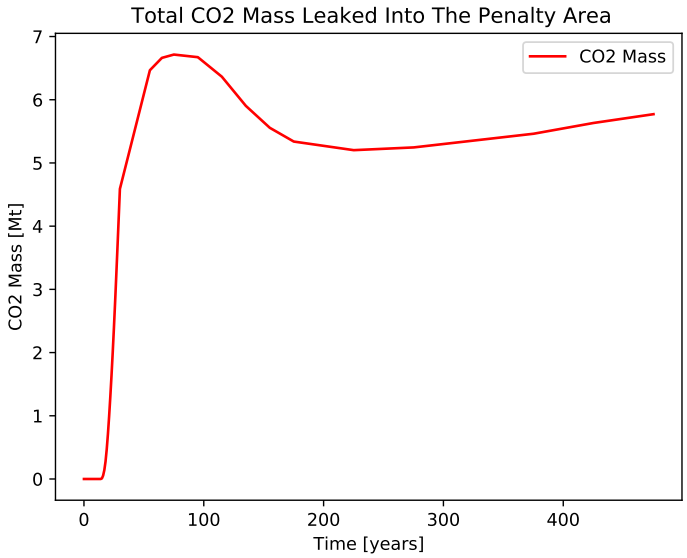


Figure 5.40: CO_2 mass leaked to the penalty zone in the penalty leakage case model.

The distribution of the CO_2 plume in the penalty zone indicates that the northern portion of the penalty zone is a permeable zone, which facilitates the migration of the plume and may play a significant role in the expansion of the CO_2 plume in the prohibited zone. Notably, the maximum mass of CO_2 leaked into the penalty area would be the basis for calculating the cost of the penalty for the operator, as it contributed to varying dangers in different instances (figure 5.41).

Data Compatibility Test:

The compatibility test plot in figure 5.42 reveals an accurate data separation. This plot is of the utmost importance in this instance, as many more calculations are performed in this section to determine the quantity of gas in the penalty zone. Numerous 3D data separations and calculations were performed.

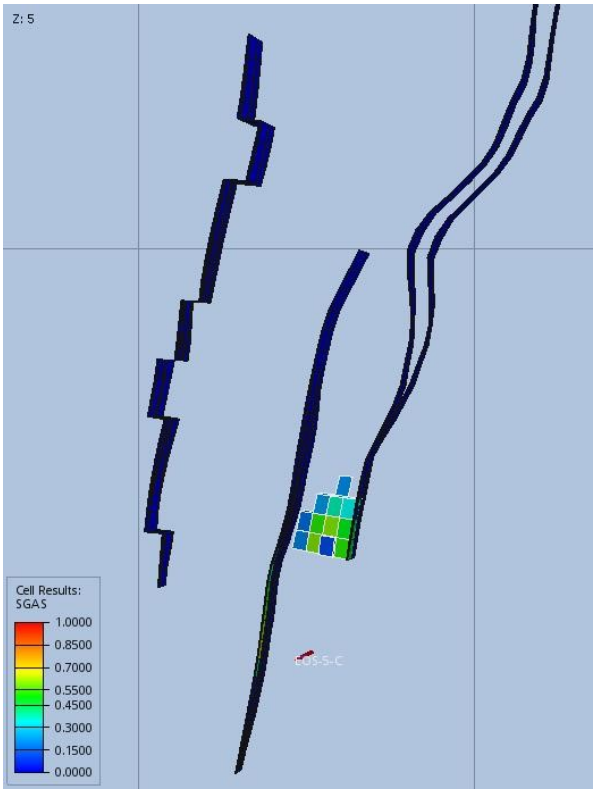
Maximum Pressure Enforced To The Cap Rock:

At the conclusion of the injection process, the maximum pressure exerted on the cap rock, as indicated by the plot ref, exceeds 440 bar. This is significantly greater than the cap rock fracture pressure we have determined (see figure 5.43). Even if there was no leakage into the penalty zone, it indicates that the injection rate is too high for this well location.

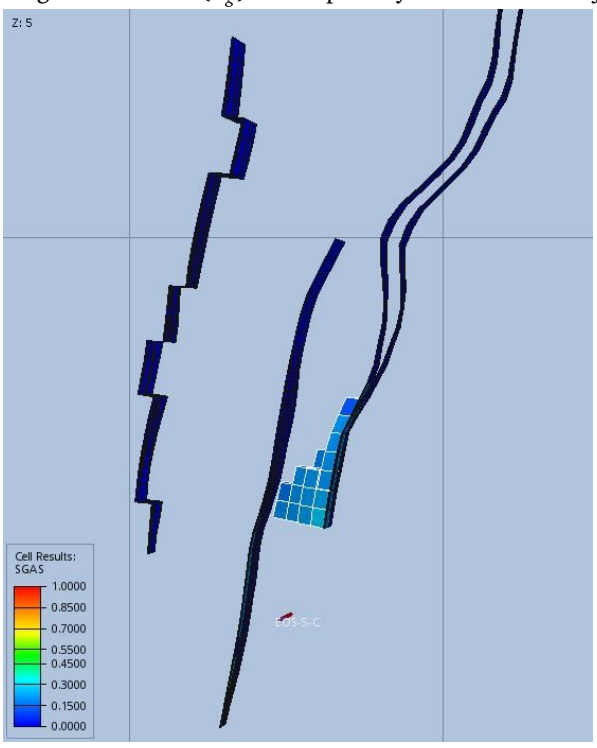
The pressure distribution scheme provided by the simulator indicates that the majority of the pressure in the reservoir's top layer is concentrated around the

locations of the wells and in the penalty area immediately following the cessation of injection operations (figure 5.44a).

By ceasing injection operations, the average pressure decreases dramatically over a period of more than 30 years, while the average pressure in the top layer decreases slightly to nearly 430 bar and stabilizes until the end of the monitoring period. The southern boundary of the reservoir would be where the pressure would be concentrated the most (figure 5.44).



(a) Top side view of gas saturation (S_g) in the penalty area after the injection operation.



(b) Top side view of gas saturation (S_g) in the penalty area after the monitoring period.

Figure 5.41: Top side view of the gas saturation (S_g) in the penalty area in the penalty leakage scenario.

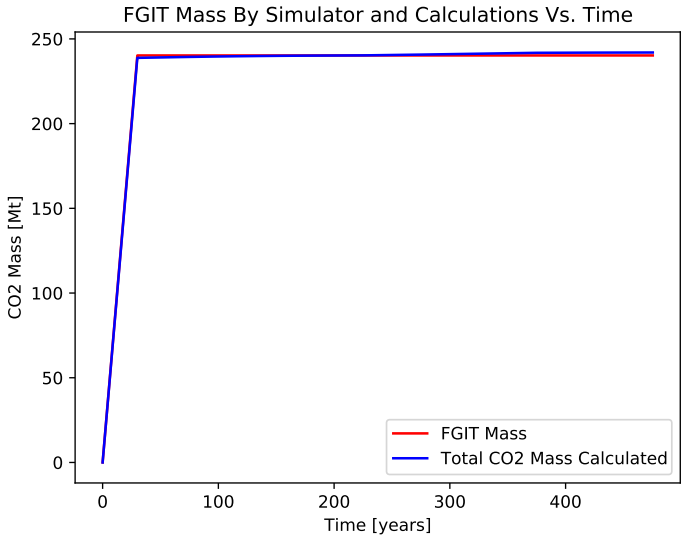


Figure 5.42: Data accuracy test by comparing the data gathered by *Python* by different calculations and the data received by the simulator itself in the leakage case model. The red curve is the simulator data and the blue curve is the data contributed by *Python*.

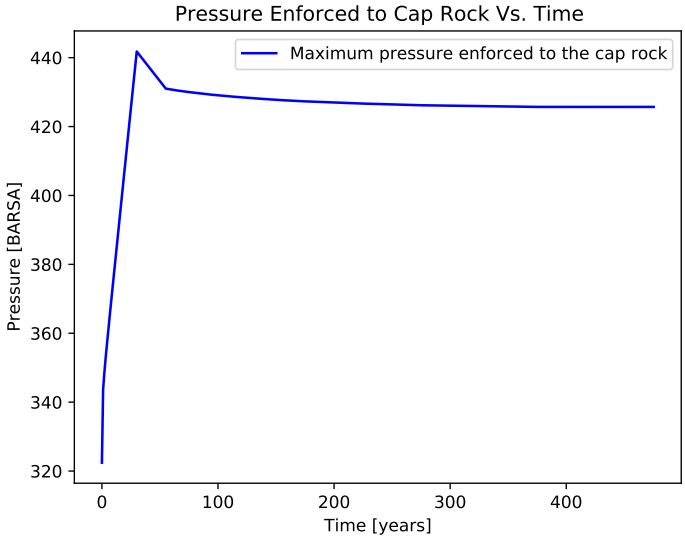
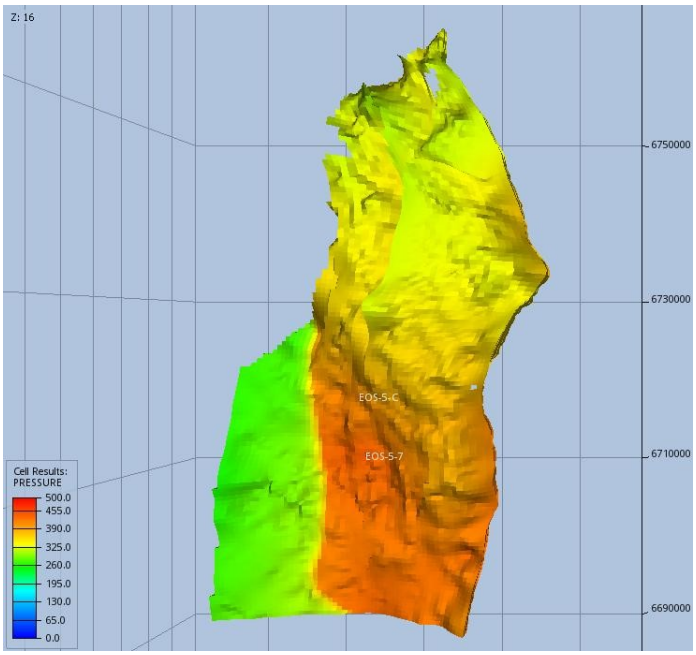
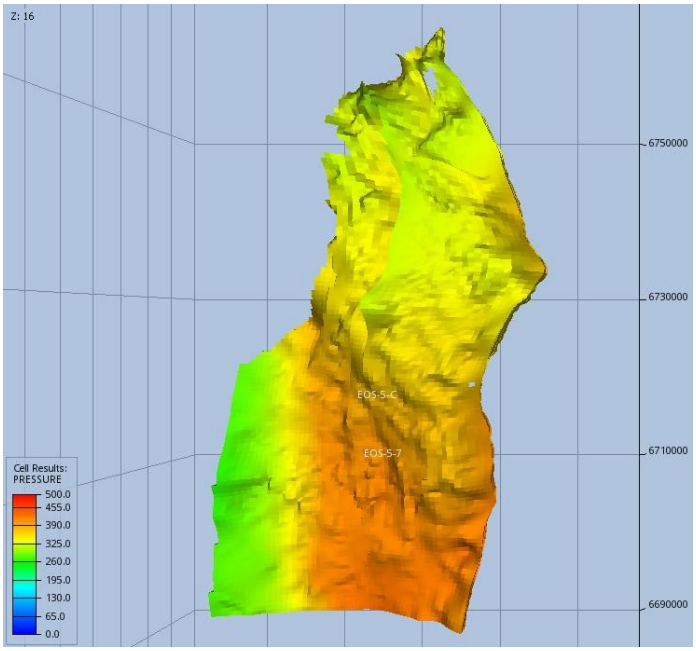


Figure 5.43: Maximum pressure enforced to the cap rock by the reservoir top layer in the penalty leakage case model.



(a) Pressure distribution in the reservoir top layer after the injection operation.



(b) Pressure distribution in the reservoir top layer at the end of the monitoring period.

Figure 5.44: Pressure distribution in the reservoir top layer (below cap rock) in the penalty leakage scenario.

Chapter 6

Discussion

In this chapter, we discuss the findings obtained from the simulations and analysis of the alternative CO_2 injection scenarios at the *Aurora* storage site. In Chapter 5, an analysis of each measurable factor and module is presented, accompanied by relevant data and charts within the corresponding section and scenario. This chapter focuses on the identification and subsequent discussion of the most significant concerns.

6.1 Data Analysis

The utilization of data collected from simulation output files for analytical purposes provides significant advantages in terms of observing the functionality of the simulation and monitoring a particular domain. The reports generated by `Python` are highly valuable and can be utilized in the development of advanced software to provide warnings during the reservoir modeling process. This capability has the potential to significantly improve and optimize the overall operation.

The role of plots in decision-making and the perception of reports is undeniably significant, as it enhances both the speed and accuracy of decision-making processes. Furthermore, plots provide a full view of several potential scenarios. The identification of uncertainties and weaknesses in simulators may also be detected through compatibility tests and data measurement using various functions.

6.1.1 CO_2 Storage Capacity

The analysis of CO_2 storage capacity highlights the scenarios that achieve the maximum storage potential. We discuss the impact of different factors such as well placement, injection rates, and reservoir characteristics on storage capacity. Additionally, we examine the trade-offs between storage capacity and other performance indicators, such as pressure distribution and plume migration. The discussion section provides insights into the strategies for optimizing CO_2 storage capacity while maintaining reservoir integrity.

6.1.2 Pressure Distribution and Reservoir Management

We analyzed the impact of different well placement strategies, injection rates, and reservoir heterogeneities on pressure distribution. Furthermore, we discussed the implications of pressure management for reservoir management and the mitigation of potential risks, such as induced seismicity or fracturing. The discussion contributes to the understanding of pressure control strategies in CO_2 storage operations.

6.2 Modeling Review

`Python` scripting can assist software and reservoir engineers in modeling software development. The model exhibits a notable degree of data reliability. By augmenting the resolution, a higher level of detail on the plume's structure and the flow in all orientations can be obtained. Working on optimizing the parameters and uncertainties which would be followed in the section 6.5 would be the procedure that helps to enhance the simulation output to a high extent.

6.3 Scripts Review

`Python` is a highly suitable tool for processing data obtained from simulators due to its flexibility in performing calculations, presenting data in preferred formats, and facilitating modification and selection of data elements. In this study, open-source software has been employed, facilitating the establishment of a robust connection between the software.

The previously mentioned advantages encompass the essential requirements for reservoir engineers to conduct data analysis through simulations and history matching, thereby obtaining the most accurate and realistic outcomes. The significance of ensuring the dependability of the output data and visualization has great value in the decision-making process, as previously discussed in the simulation workflow (see section 2.8.1). Numerous `Python`-based scripts are created by academic institutions and prominent companies for the purpose of optimizing parameters related to well control, well placement, and various other factors.

6.4 NPV Coding Review

It is possible to restructure and improve the presented code scripts in order to save computational costs. The computational cost is a significant consideration when conducting running instances in complex projects. Reducing the duration of calculations would afford additional time for decision-making and concurrently diminish the likelihood of computational errors.

6.5 Parameters Analysis

Different BHP Limit

Well control measures are of utmost significance in this project. Specifically, in another rate balancing scenario, the BHP limit is augmented to 400 bar, surpassing the current limit by 50 bar (figure 6.1). Consequently, the injection rates of the candidates (represented by blue points) escalate from 20 points to 60 cases out of a total of 100. The chosen scenario for the 400 bar BHP limit involves a *Well 31/5-7* with a 4.5 Mt/Year injection rate and a *Well 31/8-1* with a 5 Mt/Year injection rate. This combination yields a total injection rate of 9.5 Mt/Year, resulting in the injection of 285 Mt of CO_2 into the reservoir over a period of 30 years.

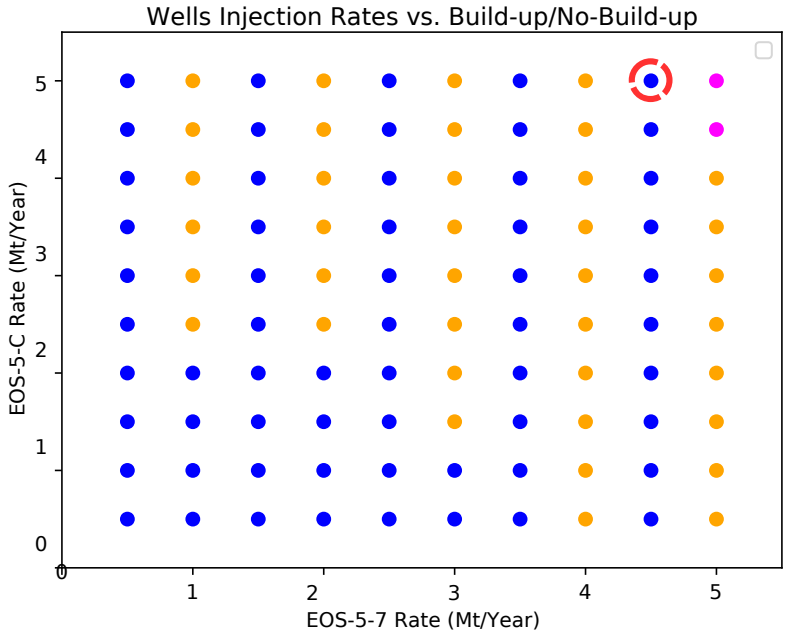


Figure 6.1: Well injection rate balance 2D plot for the BHP limit of 400 bar.

This illustrates the importance and delicacy of the standards established for proper management by firms, operators, and regulatory bodies.

Amount of Leaked CO_2

The present code exclusively assesses the quantities of free and capillary trapped gas in the penalty zone, prioritizing their measurement over that of dissolved gas due to its perceived significance. Mobile gas, also known as free gas, exhibits a notable tendency to migrate towards regions of lower pressure or upward due to buoyancy. The possibility for trapped gas to become mobile is also taken into consideration in relation to certain fracture processes or tectonic events.

Penalty Cost

The determination of penalty cost should be assigned to qualified professionals who possess expertise in assessing the potential environmental impact and operational consequences associated with a leaking incident. It is crucial to ensure a precise and accurate evaluation of the extent of harm that may be inflicted upon the environment and industrial processes.

Data Reliability

The clarity on the validity of the data output from simulators is contingent upon conducting data matching and performing precise studies on uncertainties for future research. This study, being the foremost endeavor in the field of CCS, focuses on the utilization of an inactive CO_2 storage facility. However, there exist certain errors in the model that necessitate rectification and validation to ensure more accurate simulations in subsequent studies. The computations performed by simulators for the model may have numerous possible flaws caused by the misrepresentation of the data input for the structured model.

Human Error

Due to the complexity of calculations and the extensive usage of coding, the potential for human error and calculation inaccuracies exists, hence impacting the outcomes in accordance with the nature of the error.

Penalty Area Definition

The selection of the penalty region in this thesis is based on the reservoir's current heterogeneity and the researcher's experience. The absence of a definitive source or measurement about the optimal distance between the penalty area and the danger zone, as well as the appropriate radius for the banned area, is evident.

The potential overestimation of the penalty area has the potential to significantly impair the accuracy and reliability of computations and cost assessments, hence posing substantial ramifications for the relevant authorities. Hence, it is imperative that specialists determine or recommend an appropriate distance from the reservoir boundary to serve as the foundation for future calculations.

MULTPV Multiplier

The incorporation of the boundary effect in simulation scenarios and the inclusion of the `MULTPV` parameter in the input deck to capture the correlation of pore volume distribution at reservoir boundaries introduces a potential source of uncertainty. This uncertainty arises from potential inaccuracies in the estimation of pore volume distribution and errors in the multiplier applied during the simulation.

Density

At present, the `Flow` software lacks a built-in feature that allows users to visualize the computed density of CO_2. This limitation hinders the ability to assess the impact and precision of the density during simulation.

Salinity

The salinity of the formation water is an essential variable in this paradigm, but its precise nature remains uncertain. The role of salinity is significant in the determination of gas solubility in water and the measurement of dissolved carbon dioxide mass. In the present computations, the salinity is assigned a value of $0.0011\ \frac{Kg-M}{Kg}$. Nevertheless, it is necessary to conduct a sensitivity analysis in order to ascertain the impact of the salinity parameter on the dissolved CO_2 concentration and the structure of the plume.

Rock Compressibility

Based on the findings presented in the study by [36], it is evident that rock compressibility and pore volume significantly influence the process of pressure development and stabilization. Conversely, the relative permeability curve, permeability ratio, and fault transmissibility exert substantial influence on the distribution of CO_2 plumes across the reservoir.

The rock's compressibility has been determined to be $1.2 \times 10^{-6}\ \frac{1}{BASRSA}$, a number that is considered reliable according to laboratory test findings [1]. Nevertheless, the aforementioned component remains a source of uncertainty in the simulations, and its influence on the elements described above, specifically the computation of pore volume and subsequently all the parameters assessed by those aforementioned factors, is of considerable importance.

Grid Resolution Augmentation Through PETREL

Based on the trade-off between computational expense and simulation accuracy, it is not universally advisable to employ high resolution in grid structures. Enhancing the precision of the grids, particularly on the z-axis, would contribute to the refinement of the calculation's accuracy. Hence, the impact of enhancing the grid resolution in the z direction and subdividing the reservoir layers on the current model becomes a topic for future research.

Chapter 7

Conclusion

In this concluding chapter, we summarize the key findings and contributions of this study on optimizing the balance between multiple CO_2 injectors at the *Aurora* storage site. We revisit the research objectives and provide a concise overview of the results obtained from the simulations, analysis, and discussions conducted throughout this thesis.

7.1 Summary Of Findings

The present effort has the potential to be classified as a primitive utilization of the `Python` scripts that aim to generate an integrated and all-encompassing report pertaining to a field of exploration or injection. This report would provide a rapid overview of the most favorable conditions and locations for optimal development of the field. The definition of well control methods can be determined based on the priorities and objectives outlined in the field development plan and budget. By evaluating the most cost-effective solutions and minimizing the time spent on deliberation, significant time and energy can be saved.

7.1.1 Importance of Reservoir Characterization

Understanding the potential ramifications and economic benefits of a project, as well as the environmental and production effects, is of paramount significance in the context of novel and developing technologies such as *Carbon Capture and Storage (CCS)* initiatives. Hence, the successful completion of this project, which aims to identify and propose optimal well control measures for addressing leakage issues, holds substantial importance for enterprises operating within the energy sector. Nevertheless, there are certain constraints pertaining to the acquisition of accurate data and the attainment of crucial metrics and reservoir features. In certain instances, parameters derived from prior simulations or assumptions are employed to advance the simulation endeavor. The aforementioned parameters represent sources of uncertainty that possess a significant influence on the outcome.

7.1.2 Optimization of CO_2 Injection Strategies

The results of this study demonstrate the effectiveness of employing alternative CO_2 injection scenarios for optimizing storage capacity, pressure distribution, CO_2 plume migration, and economic viability at the *Aurora* storage site for instance. The simulations revealed that well placement, injection rates, and reservoir characteristics play crucial roles in achieving optimal performance. By strategically balancing the CO_2 injectors, it is possible to maximize storage capacity while maintaining pressure control and minimizing the risk of leakage.

The primary objective of this project is to achieve balance in the rate of CO_2 injection within the proposed initiative known as *Northern Lights* and it is accomplished through the utilization of diverse rate balance plots, the establishment of a comprehensive and well-structured database, and the provision of an extensive array of reports and graphs. The *'Simulation Report.txt'* and *'Plots.pdf'* files associated with each of the examined cases constitute a valuable collection of data that should be carefully examined following the selection process of candidate injection rate scenarios. The classification of distinct situations based on the intended well control parameter state, specifically the injection rate, is achieved through the implementation of an organized naming system and the construction of a file known as the *Scenario Cluster*.

7.1.3 Estimation of Leakage To the Penalty Area

The thesis study successfully investigates the worry regarding probable leakage from the producing *Troll* gas field in close proximity to the *Aurora* storage site. The model correctly extracts the indices, defines a penalty zone, and calculates the amount of CO_2 released in the prohibited region, despite ongoing discussions regarding the trustworthiness of the data. In addition, an examination is conducted to assess the financial implications for both the operator companies and the authorities. This is accomplished by doing an evaluation of the potential costs associated with penalties.

7.1.4 Integration of Technical and Economic Factors

The economic viability of CO_2 storage projects is a crucial consideration in decision-making. The *Net Present Value (NPV)* analysis conducted allows us to assess the financial performance of the alternative scenarios, taking into account storage costs, and revenue streams. The integration of technical and economic factors helps identify economically favorable scenarios that align with sustainability goals and policy objectives.

7.2 Uncertainties

It is important to acknowledge the uncertainties and limitations associated with this study. The simulations and analysis conducted herein are based on specific

assumptions, input parameters, and modeling approaches. Uncertainties related to reservoir characterization, fluid behavior, and data availability influence the accuracy of the results. Future research should focus on reducing uncertainties through improved data collection, advanced modeling techniques, and enhanced monitoring strategies.

The *Northern Lights* project faces possible risks in terms of securing the target market and finance necessary to sustain its viability and adhere to the project timeline, as elaborated upon in the prior chapters. Utilize the aggregate injection capability facilitated by the four buildings of the *Northern Lights* project to effectively secure this object. In order to enhance the effectiveness of injection and production processes, it is imperative to provide access to innovative scenarios, a diverse range of well locations, and more sophisticated programming and simulation tools.

Ensuring the historical accuracy of simulations and models is of paramount significance for future endeavors. Numerous uncertainties have the potential to be mitigated, both in a general sense and within the context of the present field. This parameter has the potential to exert a substantial influence on the anticipation of reservoir behavior and the increase of injection/production to a heightened degree. Similarly, additional geological modeling can be conducted to assess the efficacy of fault sealing, taking into consideration the dispersion of fault planes caused by fault throw and clay composition.

7.3 Future Work and Recommendations

FieldOPT

The utilization of the open-source optimization framework, known as the Petroleum Field Development Optimization Framework (FieldOpt), developed at NTNU, could be employed for the purpose of optimization. `FieldOPT` is a comprehensive framework designed to facilitate the expeditious development and evaluation of optimization techniques.

Openpyxl Library

The report is presented in the simplified format of `'.txt'` for convenience, although more advanced and comprehensive `Python` reporting libraries like `'openpyxl'` could be employed to gain a comprehensive understanding of the simulation data at each stage, particularly during the initial stages where critical decisions need to be made.

7.4 Conclusion

In conclusion, this thesis has demonstrated the effectiveness of optimizing the balance between multiple CO_2 injectors at the *Aurora* storage site through simu-

lations, analysis, and discussions. The alternative scenarios presented in this study provide insights into strategies for maximizing storage capacity, managing pressure distribution, controlling CO_2 plume migration, and ensuring economic viability. The findings contribute to the ongoing efforts to develop sustainable and efficient CO_2 storage practices.

The optimization of CO_2 injection strategies is a complex and evolving field, and further research and collaboration are essential to address uncertainties, refine methodologies, and improve the understanding of CO_2 storage behavior. It is our hope that this thesis serves as a foundation for future studies and provides valuable insights for stakeholders involved in CO_2 storage projects. By implementing optimized CO_2 injection strategies, we can contribute to mitigating greenhouse gas emissions, addressing climate change concerns, and advancing the development of environmentally sustainable energy systems.

Bibliography

[1] G. T. Eigestad, H. K. Dahle, B. Hellevang, F. Riis, W. T. Johansen and E. Øian, 'Geological modeling and simulation of CO2 injection in the Johansen formation,' en, *Computational Geosciences*, vol. 13, no. 4, p. 435, Aug. 2009, ISSN: 1573-1499. DOI: 10.1007/s10596-009-9153-y. [Online]. Available: https://doi.org/10.1007/s10596-009-9153-y (visited on 26/01/2023).

[2] B. Metz, O. Davidson, H. C. De Coninck, M. Loos and L. Meyer, *IPCC special report on carbon dioxide capture and storage*. Cambridge: Cambridge University Press, 2005.

[3] *Northern Lights*, en. [Online]. Available: https://www.equinor.com/energy/northern-lights (visited on 10/01/2023).

[4] A. E. Lothe, P. E. Bergmo and A.-A. Grimstad, 'Storage resources for future European CCS deployment; A roadmap for a Horda CO2 storage hub, offshore Norway,' 2019.

[5] *CO2 atlas for the Norwegian Continental Shelf*, en. [Online]. Available: https://www.npd.no/en/facts/publications/co2-atlases/co2-atlas-for-the-norwegian-continental-shelf/ (visited on 05/07/2023).

[6] M. J. Rahman, M. Fawad, J. Chan Choi and N. H. Mondol, 'Effect of overburden spatial variability on field-scale geomechanical modeling of potential CO2 storage site Smeaheia, offshore Norway,' en, *Journal of Natural Gas Science and Engineering*, vol. 99, p. 104 453, Mar. 2022, ISSN: 1875-5100. DOI: 10.1016/j.jngse.2022.104453. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1875510022000440 (visited on 10/01/2023).

[7] *BAA: EL001 - Factpages - NPD*. [Online]. Available: https://factpages.npd.no/en/bsns_arr_area/pageview/c02licences/34751726 (visited on 26/12/2022).

[8] A.-K. Furre, R. Meneguolo, L. Pinturier and K. Bakke, 'Planning deep subsurface CO2 storage monitoring for the Norwegian full-scale CCS project,' en, *First Break*, vol. 38, no. 10, pp. 55–60, Oct. 2020, ISSN: 0263-5046, 1365-2397. DOI: 10.3997/1365-2397.fb2020074. [Online]. Available:

https://www.earthdoc.org/content/journals/10.3997/1365-2397.fb2020074 (visited on 25/12/2022).

[9] *Wellbore: 31/5-7 - Factpages - NPD*. [Online]. Available: https://factpages.npd.no/en/wellbore/PageView/Exploration/All/8951 (visited on 04/07/2023).

[10] *Wellbore: 31/8-1 - Factpages - NPD*. [Online]. Available: https://factpages.npd.no/en/wellbore/PageView/Exploration/All/6604 (visited on 04/07/2023).

[11] *IEA – International Energy Agency*, en-GB. [Online]. Available: https://www.iea.org (visited on 27/07/2023).

[12] C. Marchetti, 'On geoengineering and the CO2 problem,' en, *Climatic Change*, vol. 1, no. 1, pp. 59–68, Mar. 1977, ISSN: 1573-1480. DOI: 10.1007/BF00162777. [Online]. Available: https://doi.org/10.1007/BF00162777 (visited on 17/01/2023).

[13] E. Heiskanen, 'Case 24: Snohvit CO2 capture and storage project,' *Helsinki: European Commission Within the Sixth Framework*, 2006.

[14] C. Gough and S. Shackley, *: An Integrated Assessment*. London: Routledge, Dec. 2016, ISBN: 9781315260891. DOI: 10.4324/9781315260891.

[15] R. Span and W. Wagner, 'A New Equation of State for Carbon Dioxide Covering the Fluid Region from the Triple-Point Temperature to 1100 K at Pressures up to 800 MPa,' *Journal of Physical and Chemical Reference Data*, vol. 25, no. 6, pp. 1509–1596, Nov. 1996, ISSN: 0047-2689. DOI: 10.1063/1.555991. [Online]. Available: https://doi.org/10.1063/1.555991 (visited on 29/06/2023).

[16] H. Naeimi and R. Shaabani, 'Ultrasound promoted facile one pot synthesis of triazole derivatives catalyzed by functionalized graphene oxide Cu(I) complex under mild conditions,' en, *Ultrasonics Sonochemistry*, vol. 34, pp. 246–254, Jan. 2017, ISSN: 1350-4177. DOI: 10.1016/j.ultsonch.2016.05.043. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1350417716301869 (visited on 18/01/2023).

[17] *A benchmark study on problems related to CO2 storage in geologic formations | SpringerLink*. [Online]. Available: https://link.springer.com/article/10.1007/s10596-009-9146-x (visited on 27/07/2023).

[18] A. Sundal, J. P. Nystuen, K.-L. Rørvik, H. Dypvik and P. Aagaard, 'The Lower Jurassic Johansen Formation, northern North Sea–depositional model and reservoir characterization for CO2 storage,' *Marine and Petroleum Geology*, vol. 77, pp. 1376–1401, 2016.

[19] *Stratigraphy: JOHANSEN FM - Factpages - NPD*. [Online]. Available: https://factpages.npd.no/en/strat/PageView/Litho/Formations/73 (visited on 26/12/2022).

[20] *Stratigraphy: DUNLIN GP - Factpages - NPD*. [Online]. Available: https://factpages.npd.no/en/strat/PageView/Litho/Groups/29 (visited on 26/12/2022).

[21] Carl Fredrik Berg, *Tutorial for using the FieldOpt software*, Mar. 2020. [Online]. Available: https://www.youtube.com/watch?v=Iks1vGNIXE4 (visited on 26/01/2023).

[22] I. G. A. G. Angga, M. Bellout, B. S. Kristoffersen, P. E. S. Bergmo, P. A. Slotte and C. F. Berg, 'Effect of CO2 tax on energy use in oil production: Waterflooding optimization under different emission costs,' en, *SN Applied Sciences*, vol. 4, no. 11, p. 313, Oct. 2022, ISSN: 2523-3971. DOI: 10.1007/s42452-022-05197-4. [Online]. Available: https://doi.org/10.1007/s42452-022-05197-4 (visited on 26/01/2023).

[23] C. H. Whitson and M. R. Brulé, *Phase Behavior*, en. Society of Petroleum Engineers, 2000, ISBN: 978-1-55563-087-4 978-1-61399-927-1. DOI: 10.2118/9781555630874. [Online]. Available: https://onepetro.org/books/book/27/Phase-Behavior (visited on 29/06/2023).

[24] *OPM Flow manual | OPM*, en-US. [Online]. Available: https://opm-project.org/?page_id=955 (visited on 05/07/2023).

[25] *The GNU General Public License v3.0 - GNU Project - Free Software Foundation*. [Online]. Available: https://www.gnu.org/licenses/gpl-3.0.html (visited on 04/07/2023).

[26] *ResInsight*, original-date: 2012-05-15T13:08:52Z, Jul. 2023. [Online]. Available: https://github.com/OPM/ResInsight/blob/56888388998dfcb3aa20aba78a029b6110eb6eea/COPYING (visited on 04/07/2023).

[27] *About :: ResInsight*. [Online]. Available: https://resinsight.org/about/ (visited on 04/07/2023).

[28] *Open Data Commons Open Database License (ODbL) v1.0 — Open Data Commons: Legal tools for open data*. [Online]. Available: https://opendatacommons.org/licenses/odbl/1-0/ (visited on 04/07/2023).

[29] *The Johansen Dataset - CO2DataShare*, en. [Online]. Available: https://co2datashare.org/dataset/the-johansen-dataset (visited on 04/07/2023).

[30] D. K. Ponting, 'Corner Point Geometry in Reservoir Simulation,' en, in *ECMOR I - 1st European Conference on the Mathematics of Oil Recovery*, Cambridge, UK, European Association of Geoscientists & Engineers, 1989, ISBN: 978-94-6282-134-7. DOI: 10.3997/2214-4609.201411305. [Online]. Available: https://www.earthdoc.org/content/papers/10.3997/2214-4609.201411305 (visited on 01/07/2023).

[31] P. S. Bergmo, E. Lindeberg, F. Riis and W. T. Johansen, 'Exploring geological storage sites for CO2 from Norwegian gas power plants: Johansen formation,' en, *Energy Procedia*, Greenhouse Gas Control Technologies 9, vol. 1, no. 1, pp. 2945–2952, Feb. 2009, ISSN: 1876-6102. DOI: 10.1016/j.egypro.2009.02.070. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1876610209007139 (visited on 11/01/2023).

[32] T. H. Sandve, S. E. Gasda, A. Rasmussen and A. B. Rustad, 'Convective Dissolution in Field Scale Co2 Storage Simulations Using the OPM Flow Simulator,' in *TCCS–11. CO2 Capture, Transport and Storage. Trondheim 22nd–23rd June 2021 Short Papers from the 11th International Trondheim CCS Conference*, SINTEF Academic Press, 2021.

[33] A. Sundal, R. Miri, T. Ravn and P. Aagaard, 'Modelling CO2 migration in aquifers; considering 3D seismic property data and the effect of site-typical depositional heterogeneities,' en, *International Journal of Greenhouse Gas Control*, vol. 39, pp. 349–365, Aug. 2015, ISSN: 1750-5836. DOI: 10.1016/j.ijggc.2015.05.021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S175058361500184X (visited on 11/01/2023).

[34] *Ecl*, original-date: 2017-03-16T08:39:13Z, Jul. 2023. [Online]. Available: https://github.com/equinor/ecl (visited on 05/07/2023).

[35] *Northern Lights*, en-US. [Online]. Available: https://norlights.com/ (visited on 05/01/2023).

[36] S. V. Marashi, 'Northern Lights Project: Aurora Model Investigation with Sensitivity Studies and Using Different Simulation Methods,' eng, M.S. thesis, NTNU, 2022. [Online]. Available: https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/2992379 (visited on 14/12/2022).

[37] *Equinor Open Data*. [Online]. Available: https://data.equinor.com/dataset/NorthernLights (visited on 01/08/2023).

Appendix A

Code Scripts

A.1 OPM-Flow Base Case Input Deck

```
-- ========================
--
-- RUNSPEC SECTION
--
-- ========================
--
RUNSPEC
--

TITLE
 3DCO2 Injection Johansen Formation, Grid from Eigestad,
 Modified by Jalal Alali
-- Grid 500x500m faults sealing

DIMENS
--Grid Dimensions
--NX NY NZ
--------------
149 189 16 /

-- NX: the number of grid blocks in the X direction
-- NY: the number of grid blocks in the Y direction
-- NZ: the number of grid blocks in the Z direction

-- Active Phases Present
OIL
GAS
DISGAS
CO2STORE
```

```
METRIC

-- Unit Convention
-- BIGMODEL
-- DIFFUSE
-- Enables Molecular Diffusion
-- PARALLEL
-- 2 DISTRIBUTED /
-- MEMORY
-- 30000 /

TABDIMS
-- Table Of Dimensions
--
--NTSFUN NTPVT NSSFUN NRPVT NTENDP
------------------------------
1 1 40 1* 2 /

-- NTSFUN: Max. no. of relative permeability tables entered.
-- NTPVT : Max. no. of PVT tables entered (in the PROPS section).
-- NSSFUN: Max. no. of the maximum number of saturation entries in the
   relative permeability tables, ie.,
-- NRPVT : Max. no. of Rs and Rv entries in the PVT tables.
-- NTENDP: Max. no. of saturation edn point depth tables.
/

WELLDIMS
-- Well Dimension Data
--MXWELS MXCONS MXGRPS MXGRPW
--------------------------
6 16 5 10 /

-- MXWELS: Max. no. of wells in the models.
-- MXCONS: Max. no. of connections per well (i.e., no. of perforations).
-- MXGRPS: Max. no. of groups in the model.
-- MXGRPW: Max. no. of wells in any group.

REGDIMS
--
--NTFIP NMFIPR
-------------
9 3 /

-- NTFIP : Max. no. of regions in the FIPNUM region array.
```

```
-- NMFIPR: the total maximum number of fluid-in-place regions.


FAULTDIM
2000 /

START
-- Specifies a Start Date
--DAY MONTH YEAR
--------------
1 JAN 2025 /

--NSTACK
-- Stack Size For Linear Solver
--250 /

UNIFOUT
-- SWITCH ON THE UNIFIED OUTPUT FILES OPTION

UNIFIN
-- Restart From A Unified Restart File

EQLDIMS
/

-- ==========================
--
GRID
--
-- ==========================


INCLUDE
'../../../INCLUDE/FULLFIELD_IMAXJMAX.GRDECL' /


EQUALS
--
--Array Const. ---------- -------- BOX -------- ----------
--I1 I2 J1 J2 K1 K2
-------------------------------------
'ACTNUM'0 1 92 1 189 1 16 /
    'ACTNUM'0 93 149 139 176 1 16 /
'ACTNUM'0 93 149 1 138 1 8 /
```

```
'ACTNUM'0 93 149 1 138 14 16 /
/

--
-- ACTIVATE WRITING OUT OF A FLUX FILE
--
-- DUMPFLUX

-- FAULTS

INCLUDE
'../../../INCLUDE/FULLFIELD_FAULTS_IMAXJMAX.INC' /
/

-- PERM (Included in the GRDECL file)

-- INCLUDE
-- 'INCLUDE/FULLFIELD_PERMEABILITY.INC' /
-- /

COPY
PERMX PERMY /
PERMX PERMZ /
/


--MULTPV
EQUALS
--
--Array Const. ---------- -------- BOX -------- ----------
--I1 I2 J1 J2 K1 K2
-------------------------------------
'MULTPV'5.95 93 149 1 138 9 9 /
'MULTPV'5.95 93 149 1 138 13 13 /
'MULTPV'5.95 93 149 138 138 9 13 /
/


--
-- Minimum pore volume for ACTIVE cells
--
-- MINPV
-- 3000 /
--
```

```
INIT

-- ========================
--
EDIT
--
-- ========================

-- PORO (Included in the GRDECL File)

-- 'INCLUDE/FULLFIELD_POROSITY.INC' /


-- ========================
--
PROPS
--
-- ========================

-- Swirr=0.15
-- Sgr= 0.254

-- ========================
-- Saturation Dependent Data
-- ========================

--ROCK COMPRESSIBILITY
--
--REFERENCE PRESASURE IS TAKEN FROM THE HCPV WEIGHTED
    RESERVOIR PRESSURE
--

ROCK
--REF PRES   CF
----------   --------
1.0000     1.2E-06 /
/

--
-- OIL RELATIVE PERMEABILITY DATA
--
SGOF
-- Drainage
```

```
--SGAS KRG KROG PCGO
--------PSIA
------------------------------
0.00000 0.00000 0.90000 0.00000
0.20000 0.00000 0.85000 0.00000
0.25400 0.01260 0.71672 0.00000
0.30800 0.02860 0.59480 0.00000
0.36300 0.04820 0.48420 0.00000
0.41700 0.07120 0.38500 0.00000
0.47100 0.09760 0.29710 0.00000
0.52500 0.12760 0.22060 0.00000
0.57900 0.16100 0.15550 0.00000
0.63300 0.19780 0.10170 0.00000
0.68800 0.23820 0.05920 0.00000
0.74200 0.28200 0.02810 0.00000
0.79600 0.32920 0.00840 0.00000
0.85000 0.38000 0.00200 0.00000
0.90000 0.42840 0.00000 0.00000 /
/


--
--SET SALINITY FOR ALL CELLS (OPM-FLOW) KEYWORD
--

SALINITY
1.1 /

-- 1E-3 * Kg-M/Kg


RTEMP
--Reservoir
--Temperature
-------------
98 /
/

-- ============================

-- ========================
--
REGIONS
--
```

```
-- ========================

--EQUALS
--
--Array Const. ---------- -------- BOX -------- ----------
--I1 I2 J1 J2 K1 K2
-------------------------------------
--'FIPNUM'1 1 149 1 139 1 16 /
--'FIPNUM'2 1 149 1 139 1 16 /
--/
--

-- ========================
--
SOLUTION
--
-- ========================

EQUIL
-- Equilibration Data Specification
-- Datum  Pi@Datum  WOC     Pc@WOC  GOC    Pc@GOC  Rs  Rv  Accuracy
-- -----  --------  ------  ------  -----  ------  --  --  --------
   2600   260.0     5050.0  0.0     100.0  0.0     1   0 /
/

RPTRST
BASIC=2 /

-- BASIC  : Defines the frequency at which the restart data for
   restarting a run and the additional requested data is written
   to the RESTART file.
-- BASIC=2: The restart files are written at every report
   time step until this switch is reset and all the restarts
   are kept.
-- DENO   : Oil phase in-situ densities.
-- PORV   : Pore Volume.


RSVD
-- Variation Of Solution GOR With Depth
--Depth Rs
--MSCF/STB
---------------
800 0.00000
```

```
4150 0.00000 /


-- ========================
--
SUMMARY
--
-- ========================

--FIELD AVERAGES
FPR
--FRS
FGIR
--FGSAT
FGIT
FOPR
FOPT
FGIPL
FGIPG
FGIP
--FGVIS
--FGDEN
FOIP
FOIPR
FRPV
--FOPV
--FGPV
RGIP
/
RGIPL
/
RGIPG
/
-- WELL
WBHP
/
WGIR
/
WGIT
/
WGOR
/
WOPR
/
```

```
WOPT
/

-- FPR: Field Oil Phase Pressure
-- FRS : Field Gas-Oil Ratio
-- FGIR: Field Gas Injection Rate
-- FGSAT: Field Gas Saturation
-- FGIT: Field Gas Injection Total
-- FOPR: Field Oil Production Rate
-- FOPT: Field Oil Production Total
-- FGIPL: Field Gas In-Place Liquid Phase)
-- FGIPG: Field Gas In-Place (Gas Phase)
-- FGIP: Field Gas In-Place
-- FGVIS: Field Gas Viscosity
-- FGDEN: Gas Reservoir Density
-- FOIP: Field Oil In-Place
-- FOIPR: Field Oil In-Place (Reservoir Conditions)
-- FRPV: Field Pore Volume (Reservoir Conditions)
-- FOPV: Field Pore Volume (Oil)
-- FGPV: Field Pore Volume (Gas)
-- RGIP: Region Gas In-Place
-- RGIPL: Region Gas In-Place (Liquid Phase)
-- RGIPG: Region Gas In-Place (Gas Phase)
-- WBHP: Well Bottom-Hole Pressure Target/Limit
-- WGIR: Well Gas Injection Rate
-- WGIT: Well Gas Injection Total
-- WGOR: Well Gas-Oil Ratio
-- WOPR: Well Oil Production Rate
-- WOPT: Well Oil Production Total


-- ========================
--
SCHEDULE
--
-- ========================

--
DRSDTCON
--CO2 CONV
--DISSOLN
----------
0.04 /
```

```
-- RPTSCHED

-- PRESSURE SGAS SWAT PORV DENG DENW VGAS VWAT XMF AQPH SSOLID /

-- PRESSURE: Pressure
-- SGAS    : Gas Saturation
-- SWAT    : Water (Oil) Saturation
-- PORV    : Pore Volume
-- DENG    : Gas Density
-- DENW    : Water (Oil) Density
-- VGAS    : Gas Volume
-- VWAT    : Water (Oil) Volume
-- XMF     : Fluid Injection Rate
-- AQPH    :
-- SSOLID  :


WELSPECS
--Well Well I-loc J-loc BHP Phase
--Name Group Depth Fluid
-----------------------------
EOS-5-7 G1 109 92 1* GAS /
EOS-5-C G1 106 106 1* GAS /
/

-- Drainage Radius
---------------
-- 0.2 /




-- =========================
--
COMPDAT
--
-- =========================
--Well -------  Location  -------Open Sat. Trans. Well Pen.
--Name II JJ K1 K2 Shut Table Factor Dia. Dir.
-----------------------------------------
EOS-5-7 109 92 9 13 OPEN 0 1* 0.2 3* Z /
EOS-5-C 106 106 9 13 OPEN 0 1* 0.2 3* Z /
/
```

```
--
--
---------------------
--


WCONPROD
--
--Well Production Well Controls
--
--Well Open CNTL BHP
--Name Shut Mode Pres
------------------
EOS-5-7 OPEN BHP  500 /
EOS-5-C OPEN BHP  500 /
/

-- <total_injection> Mt/y for 30 years (1 jan 25)
-- <rate1m> Mt/y for "EOS-5-7"
-- <rate2m> Mt/y for "EOS-5-C"
-- Unit Concersion: Mt/Year * 1465094,13

WCONINJE
--
--Well Injection Controls
--
--Well Fluid Open CNTL Surf RESV BHP
--Name Type Shut Mode Rate Pres
--------------------------------------
EOS-5-7 GAS OPEN RATE <rate1>  1* 500  /
EOS-5-C GAS OPEN RATE <rate2>  1* 500  /
/

-- Time steps until 2055
TSTEP
--Yearly
--------
30*365
/

WELOPEN
EOS-5-7 SHUT /
```

```
EOS-5-C SHUT /
/

DATES
1 JAN 2080 /
1 JAN 2090 /
1 JAN 2100 /
1 JAN 2120 /
1 JAN 2140 /
1 JAN 2160 /
1 JAN 2180 /
1 JAN 2200 /
1 JAN 2250 /
1 JAN 2300 /
1 JAN 2400 /
1 JAN 2450 /
1 JAN 2500 /
/
END
```

A.2 CO_2 PVT Interpolation Python Script

Code listing A.1: CO_2 PVT Data Interpolation Script

```
"""
@author: Jalal Alali
"""

import numpy as np
import matplotlib.pyplot as plt

p_spann=np.array([150,160,170,180,190,200,210,220,230,240,250,260,270,280,290,300,
310,320,330,340,350,360,370,380,390,400,410,420,430,440,450,460,470,480,490,
500,510,520,530,540,550,560,570,580,590,600])

den=np.array([339.2598818,371.5183818,403.317185,434.0160957,463.1216168,
490.3323815,515.5301064,538.7369052,560.0634069,579.6650226,597.7122246,
614.3729516,629.8032612,644.1431781,657.5157308,670.0278043,681.7718488,
692.8278309,703.265095,713.1439957,722.5172736,731.431196,739.9265001,
748.0391761,755.8011209,763.2406882,770.3831547,777.2511161,783.864826,
790.2424864,796.4004982,802.3536764,808.1154374,813.6979602,819.1123271,
824.368646,829.4761569,834.4433252,839.2779231,843.9871012,848.5774518,
853.0550638,857.4255722,861.694201,865.8658017,869.9448878])

fvf=np.array([0.00551745,0.005038376,0.004641135,0.004312857,0.004041809,
0.003817511,0.003630921,0.003474515,0.00334221,0.003229191,0.00313169,
0.003046764,0.002972118,0.002905952,0.002846851,0.002793689,0.002745566,
0.002701753,0.002661655,0.002624785,0.002590733,0.00255916,0.002529777,
0.002502341,0.002476643,0.002452502,0.002429764,0.002408294,0.002387975,
0.002368702,0.002350387,0.002332948,0.002316314,0.002300423,0.002285217,
```

```
0.002270646,0.002256664,0.002243231,0.002230309,0.002217865,0.002205867,
0.002194289,0.002183104,0.00217229,0.002161824,0.002151687])

#CO2 Density at STD (1.01325 bar, 15  C )
den_co2_sc=1.87184934007942 #kg/m3

# Define the point where we want to interpolate
p_target=350 #bar

# Perform linear interpolation
fvf_interp = np.interp(p_target, p_spann, fvf)
# Print the interpolated values
print("FVF at "+str(p_target)+" bar= ", fvf_interp)


# Perform linear interpolation
den_interp = np.interp(p_target, p_spann, den)
# Print the interpolated values
print("CO2 Density at Standard Condition (SC)= ", den_co2_sc, "kg/m3")
print("CO2 Density at "+str(p_target)+" bar= ", den_interp, "kg/m3")
```

A.3 Water PVT Interpolation Python Script

Code listing A.2: Water PVT Data Interpolation Script

```
"""
@author: Jalal Alali
"""

import numpy as np
import matplotlib.pyplot as plt

den_water_sc=1E1 #kg/m3

p_water=np.array([230,235,240,245,250,255,260,265,270,275,280,285,290,295,300,
305,310,315,320,325,330,335,340,345,350,355,360,365,370,375,380,385,390,395,
400])

den_water=np.array([972.85,973.06,973.28,973.5,973.71,973.93,974.15,974.36,
974.58,974.79,975.01,975.22,975.43,975.65,975.86,976.08,976.29,976.5,976.71,
976.93,977.14,977.35,977.56,977.77,977.99,978.2,978.41,978.62,978.83,979.04,
979.25,979.46,979.67,979.87,980.08])

fvf_water=np.array([1.0279077,1.0276859,1.0274536,1.0272214,1.0269998,
1.0267678,1.0265360,1.0263147,1.0260830,1.0258620,1.0256305,1.0254097,
1.0251889,1.0249577,1.0247372,1.0245062,1.0242858,1.0240655,1.0238454,
1.0236148,1.0233948,1.0231749,1.0229551,1.0227354,1.0225053,1.0222858,
1.0220664,1.0218471,1.0216279,1.0214087,1.0211897,1.0209707,1.0207519,
1.0205435,1.0203249])

# Water Density at STD (1.01325 bar, 15  C )
den_water_sc=1000.00 #kg/m3
```

A.4 NPV Python Script

Code listing A.3: NPV Code Script

```
"""
@author: Jalal Alali
"""

import numpy as np
import matplotlib.pyplot as plt
from matplotlib.backends.backend_pdf import PdfPages
import ecl
from ecl.eclfile import EclFile
from ecl.summary import EclSum
from ecl.eclfile import EclKW, EclFileView
from ecl.grid import EclGrid
import os
import sys
import pandas as pd
from CO2_interpolation import den_co2_sc,p_spann,fvf,den
from Water_interpolation import den_water_sc,p_water,den_water,fvf_water

#simulation=input("Simulation File Name:")
# Source Files  Naming
file_name='NEW_AURORA_BASECASE_CO2_MULT'
original_file=file_name+'.DATA'
well1='EOS-5-7'
well2='EOS-5-C'

#Create a Figure Counter
n=0

#Rate Unit Convert
r_unit_convert=1465094.13  #Sm3/Day
scf_to_sm3=0.028317

#Well Control Variables
BHP_limit=350 #bar
S_girr=0.254

#Openning Data File For Further Precesses
with open(original_file, 'r') as basecase_file:
    file_contents = basecase_file.read()

#Create a Scenario Cluster to Save the Files
os.makedirs('SCENARIO_CLUSTER', exist_ok=True)
os.chdir('SCENARIO_CLUSTER')

#3D Plotting
#Plotting
x_values_list=[]
y_values_list=[]
condition_list=[]

#Initial Amount (Initial Guess)
#Well EOS-5-7
#Initial Rate
r1_MT=0.5 #Mt/Year
for i in range(10):
```

```
    r1=r1_MT
    r1_SM3=r1*r_unit_convert
    #Creating First Directory
    r1_folder=well1+'-RATE_'+str(r1)+'_Mt'
    r1_folder=r1_folder.replace('.','P')
    os.makedirs(r1_folder, exist_ok=True)
    #Openning BaseCase DATA File
    os.chdir(r1_folder)
    #Well EOS-5-C
    #Initial Rate
    r2_MT=0.5 #Mt/Year
    for j in range(10):
        r2=r2_MT
        r2_SM3=r2*r_unit_convert
        total_injection=r1+r2
        # All the actions should be take place here
        #Creating Second Directory
        r2_folder=well1+'-RATE_'+str(r1)+'_Mt'+'_AND_'+well2+'-RATE_'+str(r2)+'_Mt'
        r2_folder=r2_folder.replace('.','P')
        os.makedirs(r2_folder, exist_ok=True)
        os.chdir(r2_folder)
        #Copy Original Data To a New-Established File
        new_file_name=file_name+'_R1-'+str(r1)+'_R2-'+str(r2)
        new_file_name=new_file_name.replace('.','P')
        with open(new_file_name+'.DATA', 'w') as new_file:
            new_file.write(file_contents)
        with open(new_file_name+'.DATA', 'r') as new_file:
            new_file=new_file.read()
        # Replace the target string
        new_file = new_file.replace('<rate1>', str(r1_SM3))
        new_file = new_file.replace('<rate2>', str(r2_SM3))
        new_file = new_file.replace('<total_injection>', str(total_injection))
        new_file = new_file.replace('<rate1m>', str(r1))
        new_file = new_file.replace('<rate2m>', str(r2))
        # Write the file out in the correct directory
        with open(new_file_name+'.DATA', 'w') as new_file_2:
            new_file_2.write(new_file)
        os.system('nohup␣flow␣'+new_file_name+'.DATA')

        #sys.stdout=open('Simulation Report.txt','a')
        report_file = open("Simulation␣Report.txt", "w")
        # Redirect the Standard Output to the Report File
        sys.stdout = report_file

        # Create a PDF file
        pdf_file = "Plots.pdf"
        # Create a PdfPages object to save the plots to the PDF file
        pdf_pages = PdfPages(pdf_file)

        #simulation=input("Simulation File Name:")
        # Source Files
        hRestartFile=EclFile(new_file_name+'.UNRST')
        UNS=EclFile(new_file_name+'.UNSMRY')
        Smspec=EclSum(new_file_name+'.SMSPEC')
        Sum=EclSum(new_file_name+'.DATA')
        Grid=EclGrid(new_file_name+'.EGRID')
        Init=EclFile(new_file_name+'.INIT')
        iTimeSteps=hRestartFile.num_report_steps()

        # Density Data
```

```
print("CO2␣Density␣at␣SC=␣", den_co2_sc, "Kg/Sm3")
print("Brine␣Density␣at␣SC=␣", den_water_sc, "Kg/Sm3")

#Time Steps
print("Time␣Steps=␣", iTimeSteps, "Total␣Time␣Steps=␣", np.sum(iTimeSteps))
#Time Function
fTime_list=[]
for iTimeStep in range (0,iTimeSteps):
    fTime=hRestartFile.iget_restart_sim_days(iTimeStep)/365
    fTime_list.append(fTime)
fTime=np.array(fTime_list)
print("fTime␣Emelements=␣", len(fTime))
fTime2=Sum.numpy_vector('TIME')
fTime2=fTime2/365
#Pressure Unit
pressure_unit=Sum.unit('FPR')

#Listing the Accesible Data
keys_file=hRestartFile.keys()
# print("Keys from EclFile= ", keys_file)
keys_sum=Sum.keys(pattern=None)
# print("Keys from EclSum= ", keys_sum)
keys_init=Init.keys()
# print("Keys from INIT= ", keys_init)
# index_list=Sum.report_index_list()
# print("Index List= ", index_list)
num_rate=hRestartFile.num_named_kw('Rate')
print("Rate␣Keyword␣and␣Their␣Related␣Indexes=␣", num_rate)
num_porv=hRestartFile.num_named_kw('PORV')
print("PORV␣Keyword␣and␣Their␣Related␣Indexes=␣", num_porv)
wells=Sum.wells(pattern=None)
print("Wells=␣", wells)

#Injection and Monitoring Time Steps
injection=30
injection_period=hRestartFile.iget_restart_sim_days(injection)
print("Injection␣Period=␣", injection_period, "days␣or␣",
injection_period/365, "years")
#injection=input("Injection Period: ") can be used instead
to give the injection period to the file at the beginning
monitoring=43
monitoring_period=hRestartFile.iget_restart_sim_days(iTimeSteps-1)
print("Monitoring␣Period␣", monitoring_period, "days␣or␣",
monitoring_period/365, "years")

#Time Steps Calling
time_range=Sum.time_range(start=None, end=None, interval="1Y",
num_timestep=None, extend_end=True)
# print("Time Range= ", time_range)
print("Time␣Range␣Elements=␣", len(time_range))

#report_steps=hRestartFile.report_steps()
#print("Report Steps:", report_steps)
#report_dates=hRestartFile.report_dates()
#print("Report Dates:", report_dates)

#Data Formatted in the File
#data_formatted=hRestartFile.fprintf_data('NEW_AURORA_CO2_RW6.UNRST',
fmt=None)
#print("Data Formatted in the File= ", data_formatted)
```

```
#File Information
file_info=Grid.load_from_file(new_file_name+'.EGRID')
print("File Info=", file_info)
Active=Grid.get_num_active()
print("Number of Active Cells= ", Active)
#Define NX,NY,NZ
NX=Grid.get_nx()
print("NX= ", NX)
NY=Grid.get_ny()
print("NY= ", NY)
NZ=Grid.get_nz()
print("NZ= ", NZ)

#Index Import
index=Grid.export_index(active_only=True)
# print("Indexes Panada Frame= ", index)
# i=index['i']
# print("Pandas i Column= ", i)

indexf=np.array(index)
# print("Indexes Array= ", indexf)
print("Length of indexes: ", len(indexf))
print("Active ACTNUM (Cells) Indexes:")
#INDEX I
#i=[115, 144]
i=indexf[:,0]
print("i=", i)
i_min=np.min(i)
i_max=np.max(i)
# print('i_min= ', i_min)
# print('i_max= ', i_max)
#INDEX J
j=indexf[:,1]
print("j=", j)
j_min=np.min(j)
j_max=np.max(j)
# print('j_min= ', j_min)
# print('j_max= ', j_max)
#INDEX K
k=indexf[:,2]
print("k=", k)
k_min=np.min(k)
k_max=np.max(k)
# print('k_min= ', k_min)
# print('k_max= ', k_max)

#Defining The Penalty Area
#i=[115, 144]
i_penalty_min=115
i_penalty_max=144
print('i_min_penalty= ', i_penalty_min)
print('i_max_penalty= ', i_penalty_max)
#j=[20, 80]
j_penalty_min=20
j_penalty_max=80
print('j_min_penalty= ', j_penalty_min)
print('j_max_penalty= ', j_penalty_max)
#j=[9, 13]
k_penalty_min=9
```

```
k_penalty_max=13
print('k_min_penalty=␣', k_penalty_min)
print('k_max_penalty=␣', k_penalty_max)
print("Penalty␣Area␣Lower␣Limit=␣", [i_penalty_min,j_penalty_min,
k_penalty_min])
print("Penalty␣Area␣Top␣Limit=␣", [i_penalty_max,j_penalty_max
,k_penalty_max])
# penalty_area_cells=(indexf[:,0]>=(115))&(indexf[:,0]<=144)&
(indexf[:,1]>=20)&(indexf[:,1]<=80)
penalty_area_cells=(i>=(i_penalty_min-1))&(i<=(i_penalty_max-1))&
(j>=(j_penalty_min-1))&(j<=(j_penalty_max-1))&((k>=(k_penalty_min-1))&
(k<=(k_penalty_max-1)))
#ijk_penalty=((indexf[:,0])&(indexf[:,1])&(indexf[:,2]))
#penalty_area=indexf[penalty_area_indexes, indexf]
#PP=np.create_3d(penalty_area)
#print("Penalty Area ijk= ", PP)
cell_penalty=indexf[penalty_area_cells,3]
print("Cell␣Numbers␣of␣the␣Penalty␣Area=␣", cell_penalty)
min_cell_penalty=np.min(cell_penalty)
max_cell_penalty=np.max(cell_penalty)
print("Number␣of␣active␣Cells␣in␣the␣Penalty␣Area=␣", len(cell_penalty))

#FPR Calling
FPR=Sum.numpy_vector('FPR')
# print("FPR= ", FPR)
Max_FPR=np.max(FPR)
print("Max␣FPR:␣", Max_FPR, pressure_unit)

#WBHP: EOS-5-7 Calling
BHP1_total=Sum.numpy_vector('WBHP:'+well1)
BHP1=[x for x in BHP1_total if x!=0]
# print("WBHP:EOS-5-7=", BHP1, pressure_unit)
print("WBHP:"+well1+"␣Elements=␣", len(BHP1_total))
BHP1_min=np.min(BHP1_total)
BHP1_max=np.max(BHP1_total)
# print("Minimum WBHP:EOS-5-7= ", BHP1_min, pressure_unit)
print("Maximum␣WBHP:"+well1+"=␣", BHP1_max, pressure_unit)

#WBHP: EOS-5-C Calling
BHP2_total=Sum.numpy_vector('WBHP:'+well2)
BHP2=[x for x in BHP2_total if x!=0]
# print("WBHP:EOS-5-C=", BHP2, pressure_unit)
print("WBHP:"+well2+"␣Elements=␣", len(BHP2_total))
BHP2_min=np.min(BHP2_total)
BHP2_max=np.max(BHP2_total)
# print("Minimum WBHP:EOS-5-C= ", BHP2_min, pressure_unit)
print("Maximum␣WBHP:"+well2+"=␣", BHP2_max, pressure_unit)

#Well Control Loop
print("BHP␣Limit=", BHP_limit, pressure_unit)
if BHP1_max>=BHP_limit:
    condition1=0.2
    print("Pressure␣Build-Up␣Happened␣in␣Well␣"+well1)
else:
    condition1=0
    print("No␣Pressure␣Build-up␣Happened␣in␣Well␣"+well1)

if BHP2_max>=BHP_limit:
    condition2=0.3
    print("Pressure␣Build-Up␣Happened␣in␣Well␣"+well2)
```

```
        else:
            condition2=0
            print("No␣Pressure␣Build-up␣Happened␣in␣Well␣"+well2)
        if Max_FPR>=BHP_limit:
            condition3=0.4
            print("Pressure␣Build-Up␣Due␣to␣FPR␣Happened␣During␣the
␣␣␣␣␣␣␣␣␣␣␣␣Injection␣Operation")
        else:
            condition3=0
            print("No␣Pressure␣Build-Up␣Due␣to␣FPR␣Happened␣During␣the
␣␣␣␣␣␣␣␣␣␣␣␣Injection␣Operation")
        condition=condition1+condition2+condition3
        condition_list.append(condition)
        #FGIR Calling
        FGIR=Sum.numpy_vector('FGIR')
        FGIR_unit=Sum.unit('FGIR')
        # print("FGIR: ", FGIR)
        # print("FGIR Elements= ", len(FGIR_total))
        FGIR_min=np.min(FGIR)
        FGIR_max=np.max(FGIR)
        print("Minimum␣FGIR=␣", FGIR_min, FGIR_unit)
        print("Maximum␣FGIR=␣", FGIR_max, FGIR_unit)

        if FGIR_min==FGIR_max:
            print("No␣Rate␣Decline␣Happened␣During␣the␣Injection␣Operation")
        else:
            print("Rate␣Decline␣Happened␣During␣the␣Injection␣Operation")

        #WGIR: EOS-5-7 Calling
        GIR1=Sum.numpy_vector('WGIR:'+well1)
        rate_unit=Sum.unit("FGIR")
        # print("WGIR of EOS-5-7= ", rate1)
        # print("WGIR:"+well1+" Elements= ", len(rate1_total))
        GIR1_min=np.min(GIR1)
        GIR1_max=np.max(GIR1)
        print("Minimum␣WGIR:"+well1+"␣=␣", GIR1_min, rate_unit)
        print("Maximum␣WGIR:"+well1+"␣=␣", GIR1_max, rate_unit)

        if GIR1_min==GIR1_max:
            print("No␣Rate␣Decline␣Happened␣in␣Well␣"+well1)
        else:
            print("Rate␣Decline␣Happened␣in␣Well␣"+well1)

        #WGIR: EOS-5-C Calling
        GIR2=Sum.numpy_vector('WGIR:'+well2)
        GIR2_min=np.min(GIR2)
        GIR2_max=np.max(GIR2)
        print("Minimum␣WGIR:"+well2+"=␣", GIR2_min, rate_unit)
        print("Maximum␣WGIR:"+well2+"=␣", GIR2_max, rate_unit)

        if GIR2_min==GIR2_max:
            print("No␣Rate␣Decline␣Happened␣in␣Well␣"+well2)
        else:
            print("Rate␣Decline␣Happened␣in␣Well␣"+well2)

        #Calling PORV From the ".INIT" File
        PORV=Init.iget_named_kw('PORV',0)
        PORV3D=Grid.create_3d(PORV)
        # PORV3D_nonzero=[x for x in PORV3D if x!=0]
        PORV3D_penalty=PORV3D[(i_penalty_min-1):(i_penalty_max-1),
```

```
(j_penalty_min-1):(j_penalty_max-1),
(k_penalty_min-1):(k_penalty_max-1)]
PORV_unit='Sm3'
print("Minimum␣PV␣in␣the␣Penalty␣Area=␣", np.min(PORV3D_penalty),
PORV_unit)
print("Maximum␣PV␣in␣the␣Penalty␣Area=␣", np.max(PORV3D_penalty),
PORV_unit)
print("Total␣Number␣of␣PORV␣Elements␣in␣the␣Penalty␣Area:␣",
len(PORV3D_penalty))
print("Total␣Pore␣Volume␣of␣the␣Penalty␣Area=␣", np.sum(PORV3D_penalty),
PORV_unit)

#FGIT Calling (Total Injected Gas Calculation)
FGIT=Sum.numpy_vector('FGIT')
FGIT_mass=FGIT*den_co2_sc
FGIT_max=np.max(FGIT)
FGIT_mass_max=np.max(FGIT_mass)
FGIT_unit=Sum.unit('FGIT')
print("Total␣Injected␣Gas␣(FGIT)=␣", FGIT_max, FGIT_unit,"␣=␣",
FGIT_mass_max/1E9, "Mt")
#print("FGIT Vector= ", FGIT)


#WGIT:EOS-5-7 Calling
GIT1=Sum.numpy_vector('WGIT:'+well1)
GIT1_mass=GIT1*den_co2_sc
GIT1_max=np.max(GIT1)
GIT1_mass_max=np.max(GIT1_mass)
print("Total␣Injected␣Gas␣Through␣Well␣"+well1+"=␣", GIT1_max, FGIT_unit,
"␣=␣", GIT1_mass_max/1E9, "Mt")

#WGIT:EOS-5-C Calling
GIT2=Sum.numpy_vector('WGIT:'+well2)
GIT2_mass=GIT2*den_co2_sc
GIT2_max=np.max(GIT2)
GIT2_mass_max=np.max(GIT2_mass)
print("Total␣Injected␣Gas␣Through␣Well␣"+well2+"=␣", GIT2_max, FGIT_unit,
"␣=␣", GIT2_mass_max/1E9, "Mt")

#GIT & FGIT Volume Plotting
FGIT_transposed=FGIT.T
GIT1_transposed=GIT1.T
GIT2_transposed=GIT2.T
fig1=plt.figure(n+1)
plt.plot(fTime2,FGIT,linestyle='-', label='Total␣Injected␣CO2␣(FGIT)',
color='black')
plt.plot(fTime2,GIT1,linestyle='--', label=well1,color='blue')
plt.plot(fTime2,GIT2,linestyle='--', label=well2,color='red')
plt.title('Total␣Injected␣CO2␣Volume␣Vs.␣Time')
plt.xlabel('Time␣[years]')
plt.ylabel('Injected␣CO2␣Volume␣[Sm3]')
plt.legend()
plt.show()
plt.savefig('FGIT␣Profile.pdf')
pdf_pages.savefig(fig1)

#GIT & FGIT Mass Plotting
fig2=plt.figure(n+2)
plt.plot(fTime2,FGIT_mass/1E9,linestyle='-',
label='Total␣Injected␣CO2␣Mass',color='black')
```

```
        plt.plot(fTime2,GIT1_mass/1E9,linestyle='--', label=well1,color='blue')
        plt.plot(fTime2,GIT2_mass/1E9,linestyle='--', label=well2,color='red')
        plt.title('Total␣Injected␣CO2␣Mass␣Vs.␣Time')
        plt.xlabel('Time␣[years]')
        plt.ylabel('Injected␣CO2␣Mass␣[Mt]')
        plt.legend()
        plt.show()
        plt.savefig('FGIT␣Mass␣Profile.pdf')
        pdf_pages.savefig(fig2)

        # Penalty for 1 Sm3 of Injected CO2 Into the Penalty Area
        CO2_penalty_price=20 #Dollar($/Ton)
        Dollar_to_NOK=10.05
        CO2_penalty_price_NOK=CO2_penalty_price*Dollar_to_NOK
        print("Penalty␣Cost␣for␣each␣ton␣of␣Leaked␣CO2␣to␣the␣Penalty␣Area=␣",
        CO2_penalty_price, "Dollars␣($)␣=␣", CO2_penalty_price_NOK, "NOK")

        #Penalty Area Free Gas (CO2) for the Injection Time Step
        SGASi_list=[]
        SGASi_penalty_max_list=[]
        GVi_penalty_list=[]
        GVi_penalty_mass_list=[]
        for iTimeStep in range(0,injection+1):
            SGAS=hRestartFile.iget_named_kw('SGAS',iTimeStep)
            SGAS3D=Grid.create_3d(SGAS)
            SGAS3D_max=np.max(SGAS3D)
            SGASi_list.append(SGAS3D_max)
            SGAS3D_penalty=SGAS3D[(i_penalty_min-1):(i_penalty_max-1),
            (j_penalty_min-1):(j_penalty_max-1),(k_penalty_min-1):(k_penalty_max-1)]
            SGASi_penalty_max=np.max(SGAS3D_penalty)
            SGASi_penalty_max_list.append(SGASi_penalty_max)
            P=hRestartFile.iget_named_kw('PRESSURE',iTimeStep)
            P3D=Grid.create_3d(P)
            #FVF Interpolation
            fvf_interp_3D = np.interp(P3D, p_spann, fvf)
            fvf_interp_penalty_3D=fvf_interp_3D[(i_penalty_min-1):(i_penalty_max-1),
            (j_penalty_min-1):(j_penalty_max-1),(k_penalty_min-1):(k_penalty_max-1)]
            GVi_penalty=SGAS3D_penalty*PORV3D_penalty/fvf_interp_penalty_3D
            GVi_penalty_mass=GVi_penalty*den_co2_sc
            GVi_penalty_sum=np.sum(GVi_penalty)
            GVi_penalty_mass_sum=np.sum(GVi_penalty_mass)
            GVi_penalty_list.append(GVi_penalty_sum)
            GVi_penalty_mass_list.append(GVi_penalty_mass_sum)
            #A grapg of SGAS vs. Time is needed
        GVi_penalty_injection=GVi_penalty_list[-1]
        GVi_penalty_mass_injection=GVi_penalty_mass_list[-1]
        # print("GVi Penalty List= ", GVi_penalty_list)
        # print("SGAS in the Penalty Area after the Injection Period= ", SGASi_list)
        print("Accumulated␣Gas␣Volume␣in␣the␣Penalty␣Area␣After
␣␣␣␣␣␣␣␣the␣Injection␣Period=␣",
        GVi_penalty_injection, FGIT_unit, "␣=␣",
        GVi_penalty_mass_injection/1E9, "Mt")
        GVi_penalty_max=np.max(GVi_penalty_list)
        GVi_penalty_mass_max=np.max(GVi_penalty_mass_list)
        print("Maximum␣Accumulated␣Gas␣Volume␣in␣the␣Penalty␣Area␣During␣the
␣␣␣␣␣␣␣␣Injection␣Period=␣", GVi_penalty_max, FGIT_unit, "␣=␣",
        GVi_penalty_mass_max/1E9, "Mt")

        #Penalty Area Free Gas (CO2) Loop for the Monitoring Time Step
        SGASm_list=[]
```

```
        SGASm_penalty_max_list=[]
        GVm_penalty_list=[]
        GVm_penalty_mass_list=[]
        for iTimeStep in range(injection+1,monitoring+1):
            SGAS=hRestartFile.iget_named_kw('SGAS',iTimeStep)
            SGAS3D=Grid.create_3d(SGAS)
            SGAS3D_max=np.max(SGAS3D)
            SGASm_list.append(SGAS3D_max)
            SGAS3D_penalty=SGAS3D[(i_penalty_min-1):(i_penalty_max-1),
            (j_penalty_min-1):(j_penalty_max-1),
            (k_penalty_min-1):(k_penalty_max-1)]
            SGASm_penalty_max=np.max(SGAS3D_penalty)
            SGASm_penalty_max_list.append(SGASm_penalty_max)
            P=hRestartFile.iget_named_kw('PRESSURE',iTimeStep)
            P3D=Grid.create_3d(P)
            #FVF Interpolation
            fvf_interp_3D = np.interp(P3D, p_spann, fvf)
            fvf_interp_penalty_3D=fvf_interp_3D[(i_penalty_min-1):(i_penalty_max-1),
            (j_penalty_min-1):(j_penalty_max-1),(k_penalty_min-1):(k_penalty_max-1)]
            GVm_penalty=SGAS3D_penalty*PORV3D_penalty/fvf_interp_penalty_3D
            GVm_penalty_sum=np.sum(GVm_penalty)
            GVm_penalty_mass=GVm_penalty*den_co2_sc
            GVm_penalty_mass_sum=np.sum(GVm_penalty_mass)
            GVm_penalty_list.append(GVm_penalty_sum)
            GVm_penalty_mass_list.append(GVm_penalty_mass_sum)
            #A grapgh of SGAS vs. Time is needed
        GVm_penalty_monitoring=GVm_penalty_list[-1]
        GVm_penalty_mass_monitoring=GVm_penalty_mass_list[-1]
        # print("GVm Penalty List= ", GVm_penalty_list)
        # print("SGAS in the Penalty Area after the Monitoring Period= ",
        SGASm_list)
        print("Accumulated␣Gas␣Volume␣in␣the␣Penalty␣Area␣After␣the␣End␣of
␣␣␣␣␣␣␣␣the␣Monitoring␣Period=␣", GVm_penalty_monitoring, FGIT_unit, "␣=␣",
        GVm_penalty_mass_monitoring/1E9, "Mt")

        GVm_penalty_max=np.max(GVm_penalty_list)
        GVm_penalty_mass_max=np.max(GVm_penalty_mass_list)
        print("Maximum␣Accumulated␣Gas␣Volume␣in␣the␣Penalty␣Area␣During
␣␣␣␣␣␣␣␣the␣Monitoring␣Period=␣", GVm_penalty_max, FGIT_unit, "␣=␣",
        GVm_penalty_mass_max/1E9, "Mt")

        #Contribute a Total GV List
        GV_penalty_list=[]
        GV_penalty_mass_list=[]
        GV_penalty_list.append(GVi_penalty_list+GVm_penalty_list)
        GV_penalty_mass_list.append(GVi_penalty_mass_list+GVm_penalty_mass_list)
        GV_penalty=np.array(GV_penalty_list)
        GV_penalty_mass=np.array(GV_penalty_mass_list)
        GV_penalty_max=np.max(GV_penalty)
        GV_penalty_mass_max=np.max(GV_penalty_mass)
        GV_penalty_transposed=GV_penalty.T
        GV_penalty_mass_transposed=GV_penalty_mass.T
        # print("GV Penalty List= ",GV_penalty_list)
        # print("GV Penalty List Arrays= ", GV_penalty_transposed)
        # print("GV Penalty Elements= ", len(GV_penalty_transposed))

        #Maximum Gas Volume Status Loop
        if GVi_penalty_max>=GVm_penalty_max:
            GV_max=GVi_penalty_max
            GV_mass_max=GVi_penalty_mass_max
```

```
        else:
            GV_max=GVm_penalty_max
            GV_mass_max=GVm_penalty_mass_max
        print("Maximum␣Accumulated␣Gas␣Mass␣in␣the␣Penalty␣Area␣During
␣␣␣␣␣␣␣␣the␣Hole␣Project␣Period=␣", GV_mass_max/1E3, 'tons')

        #Penalty Amount Settlement
        Penalty_cost=(GV_mass_max/1E3)*CO2_penalty_price
        Penalty_cost_NOK=Penalty_cost*Dollar_to_NOK
        print("Penalty␣Cost␣That␣the␣Contractor␣Company␣Should␣Pay␣to␣Compensate=␣",
        Penalty_cost/1e+6, "Million␣Dollars␣(m$)","=␣", Penalty_cost_NOK/1e+6,
        "Million␣NOK␣(mNOK)")

        if (Penalty_cost/1e+6)>1:
            print("The␣Project␣has␣no␣economic␣justification!")
        else:
            print("The␣Project␣is␣economically␣justified!")

        #Contribute a Total SGAS List
        SGAS_list=[]
        SGAS_list.append(SGASi_list+SGASm_list)
        SGAS=np.array(SGAS_list)
        SGAS_max=np.max(SGAS)
        SGAS_transposed=SGAS.T
        print("Maximum␣SGAS␣in␣the␣Reservoir=␣", SGAS_max)
        SGAS_penalty_list=[]
        SGAS_penalty_list.append(SGASi_penalty_max_list+SGASm_penalty_max_list)
        SGAS_penalty=np.array(SGAS_penalty_list)
        SGAS_penalty_max=np.max(SGAS_penalty)
        SGAS_penalty_transposed=SGAS_penalty.T
        print("Maximum␣SGAS␣in␣the␣Penalty␣Area=␣", SGAS_penalty_max)

        #SGAS Plotting
        fig3=plt.figure(n+3)
        plt.plot(fTime,SGAS_transposed,linestyle='-',label='SGAS␣in␣the␣Field',
        color='black')
        plt.plot(fTime,SGAS_penalty_transposed,linestyle='--',label='SGAS␣in
␣␣␣␣␣␣␣␣the␣Penalty␣Area',color='red')
        plt.title('CO2␣Saturation␣(SGAS)␣Vs.␣Time')
        plt.xlabel('Time␣[years]')
        plt.ylabel('SGAS␣[-]')
        plt.legend()
        plt.show()
        plt.savefig('SGAS␣Profile.pdf')
        pdf_pages.savefig(fig3)

        #Rs Calling and Dissolved Gas Calculating
        Rs_max_list=[]
        DISGAS_list=[]
        DISGAS_penalty_list=[]
        SWAT_list=[]
        SWAT_penalty_list=[]
        GV_list=[]
        GV_mass_list=[]
        GV_free_list=[]
        GV_trapped_list=[]
        DISGAS_mass_list=[]
        DISGAS_penalty_mass_list=[]
        GV_free_mass_list=[]
        GV_trapped_mass_list=[]
```

```
GV_free_penalty_mass_sum_list=[]
for iTimeStep in range(0,iTimeSteps):
    Rs=hRestartFile.iget_named_kw('RS', iTimeStep)
    Rs3D=Grid.create_3d(Rs)
    Rs3D_penalty=Rs3D[(i_penalty_min-1):(i_penalty_max-1),
    (j_penalty_min-1):(j_penalty_max-1),(k_penalty_min-1):(k_penalty_max-1)]
    Rs3D_max=np.max(Rs3D)
    Rs_max_list.append(Rs3D_max)
    P=hRestartFile.iget_named_kw('PRESSURE',iTimeStep)
    P3D=Grid.create_3d(P)
    #FVF Interpolation
    fvf_interp_3D = np.interp(P3D, p_spann, fvf)
    fvf_interp_penalty_3D=fvf_interp_3D[(i_penalty_min-1):(i_penalty_max-1),
    (j_penalty_min-1):(j_penalty_max-1),(k_penalty_min-1):(k_penalty_max-1)]
    fvf_water_interp_3D = np.interp(P3D, p_water, fvf_water)
    fvf_water_penalty_interp_3D=fvf_water_interp_3D
    [(i_penalty_min-1):(i_penalty_max-1),(j_penalty_min-1):(j_penalty_max-1)
    ,(k_penalty_min-1):(k_penalty_max-1)]
    SGAS=hRestartFile.iget_named_kw('SGAS',iTimeStep)
    SGAS3D=Grid.create_3d(SGAS)
    SGAS3D_array=np.array(SGAS3D)
    GV=SGAS3D*PORV3D
    GV_mass=GV*den_co2_sc
    GV_sum=np.sum(GV)
    GV_mass_sum=np.sum(GV_mass)
    GV_list.append(GV_sum)
    GV_mass_list.append(GV_mass_sum)
    SGAS3D_free=np.where(SGAS3D_array > S_girr, SGAS3D_array-S_girr, 0)
    SGAS3D_free_penalty=SGAS3D_free[(i_penalty_min-1):(i_penalty_max-1),
    (j_penalty_min-1):(j_penalty_max-1),(k_penalty_min-1):(k_penalty_max-1)]
    GV_free=SGAS3D_free*PORV3D/fvf_interp_3D
    GV_free_mass=GV_free*den_co2_sc
    GV_free_sum=np.sum(GV_free)
    GV_free_mass_sum=np.sum(GV_free_mass)
    GV_free_list.append(GV_free_sum)
    GV_free_mass_list.append(GV_free_mass_sum)
    GV_free_penalty=SGAS3D_free_penalty*PORV3D_penalty/fvf_interp_penalty_3D
    GV_free_penalty_mass=GV_free_penalty*den_co2_sc
    GV_free_penalty_mass_sum=np.sum(GV_free_penalty_mass)
    GV_free_penalty_mass_sum_list.append(GV_free_penalty_mass_sum)
    SGAS3D_trapped=np.where(SGAS3D_array > S_girr, S_girr, SGAS3D_array)
    GV_trapped=SGAS3D_trapped*PORV3D/fvf_interp_3D
    GV_trapped_mass=GV_trapped*den_co2_sc
    GV_trapped_sum=np.sum(GV_trapped)
    GV_trapped_mass_sum=np.sum(GV_trapped_mass)
    GV_trapped_list.append(GV_trapped_sum)
    GV_trapped_mass_list.append(GV_trapped_mass_sum)
    SWAT3D=1-SGAS3D
    SWAT3D_min=np.min(SWAT3D)
    SWAT_list.append(SWAT3D_min)
    SWAT3D_penalty=SWAT3D[(i_penalty_min-1):(i_penalty_max-1),
    (j_penalty_min-1):(j_penalty_max-1),(k_penalty_min-1):(k_penalty_max-1)]
    SWAT3D_penalty_min=np.min(SWAT3D_penalty)
    SWAT_penalty_list.append(SWAT3D_penalty_min)
    V_water_penalty_rc=SWAT3D_penalty*PORV3D_penalty
    V_water_penalty_sc=V_water_penalty_rc/fvf_water_penalty_interp_3D
    DISGAS_penalty=Rs3D_penalty*V_water_penalty_sc
    DISGAS_penalty_mass=DISGAS_penalty*den_co2_sc
    DISGAS_penalty_sum=np.sum(DISGAS_penalty)
    DISGAS_penalty_mass_sum=np.sum(DISGAS_penalty_mass)
```

```
            DISGAS_penalty_list.append(DISGAS_penalty_sum)
            DISGAS_penalty_mass_list.append(DISGAS_penalty_mass_sum)
            V_water_rc=SWAT3D*PORV3D
            V_water_sc=V_water_rc/fvf_water_interp_3D
            DISGAS=Rs3D*V_water_sc
            DISGAS_mass=DISGAS*den_co2_sc
            DISGAS_sum=np.sum(DISGAS)
            DISGAS_mass_sum=np.sum(DISGAS_mass)
            DISGAS_list.append(DISGAS_sum)
            DISGAS_mass_list.append(DISGAS_mass_sum)
        # print("Rs Max List= ", Rs_max_list)
        Rs=np.array(Rs_max_list)
        Rs_max=np.max(Rs)
        Rs_transposed=Rs.T
        print("Maximum␣Rs=␣", Rs_max)

        #Dissolved Gas (CO2) of the Field and the Penalty Area
        DISGAS_penalty=np.array(DISGAS_penalty_list)
        DISGAS_penalty_mass=np.array(DISGAS_penalty_mass_list)
        DISGAS_penalty_max=np.max(DISGAS_penalty)
        DISGAS_penalty_mass_max=np.max(DISGAS_penalty_mass)
        DISGAS_penalty_transposed=DISGAS_penalty.T
        DISGAS_penalty_mass_transposed=DISGAS_penalty_mass.T
        # print("Penalty Area Dissolved Gas= ", DISGAS_penalty_list)

        # Field Dissolved Gas Amount
        DISGAS=np.array(DISGAS_list)
        DISGAS_mass=np.array(DISGAS_mass_list)
        DISGAS_max=np.max(DISGAS)
        DISGAS_mass_max=np.max(DISGAS_mass)
        DISGAS_transposed=DISGAS.T
        DISGAS_mass_transposed=DISGAS_mass.T
        # print("Dissolved Gas in the Reservoir= ", DISGAS_list)
        print("Dissolved␣Gas␣in␣the␣Penalty␣Area␣Till␣the␣End␣of
␣␣␣␣␣␣␣␣the␣Monitoring␣Period=␣", DISGAS_penalty[-1], "Sm3", "␣=␣",
        DISGAS_penalty_mass[-1]/1E9, "Mt")
        print("Dissolved␣Gas␣in␣the␣Reservoir␣Till␣the␣End␣of
␣␣␣␣␣␣␣␣the␣Monitoring␣Period=␣", DISGAS[-1], "Sm3", "␣=␣", DISGAS_mass[-1]/1E9,
        "Mt")

        #Free Gas in the Penalty Area
        GV_free_penalty_mass=np.array(GV_free_penalty_mass_sum_list)
        # GV_free_penalty_mass_transposed=GV_free_penalty_mass.T
        print("Mobile␣CO2␣Mass␣in␣The␣Penalty␣Area=␣",
        np.max(GV_free_penalty_mass)/1E9, "Mt")

        #Total Leaked Gas
        GV_penalty_area_mass=GV_penalty_mass+DISGAS_penalty_mass
        GV_penalty_area_mass_transposed=GV_penalty_area_mass.T

        #CO2 Volume Leakage into the Penalty Area Plotting
        fig4=plt.figure(n+4)
        plt.plot(fTime,GV_penalty_mass_transposed/1E9,label='CO2␣Mass',color='red')
        plt.title('Total␣CO2␣Mass␣Leaked␣Into␣The␣Penalty␣Area')
        plt.xlabel('Time␣[years]')
        plt.ylabel('CO2␣Mass␣[Mt]')
        plt.legend()
        plt.show()
        plt.savefig('CO2␣Mass␣in␣the␣Penalty␣Area.pdf')
        pdf_pages.savefig(fig4)
```

```
        #CO2 Mass Leakage into the Penalty Area Plotting
        fig5=plt.figure(n+5)
        plt.plot(fTime,GV_penalty_mass_transposed/1E9,label='CO2␣Mass␣in
␣␣␣␣␣␣␣␣the␣Penalty␣Area',color='red')
        plt.title('Total␣CO2␣Mass␣Leaked␣Into␣The␣Penalty␣Area')
        plt.xlabel('Time␣[years]')
        plt.ylabel('CO2␣Mass␣[Mt]')
        plt.legend()
        plt.show()
        plt.savefig('CO2␣Mass␣in␣the␣Penalty␣Area.pdf')
        pdf_pages.savefig(fig5)

        #Rs Plotting
        fig6=plt.figure(n+6)
        plt.plot(fTime,Rs_transposed,linestyle='-', label='Rs',color='blue')
        plt.title('Gas␣Solution␣Ratio␣in␣the␣Reservoir␣(Rs)␣Vs.␣Time')
        plt.xlabel('Time␣[years]')
        plt.ylabel('Rs␣[Sm3/Sm3]')
        plt.legend()
        plt.show()
        plt.savefig('Rs␣Profile.pdf')
        pdf_pages.savefig(fig6)

        #Dissolved CO2 Volume Plotting
        fig7=plt.figure(n+7)
        plt.plot(fTime,DISGAS_transposed,linestyle='-',label='Dissolved␣CO2
␣␣␣␣␣␣␣␣Volume␣in␣The␣Reservoir',color='black')
        plt.plot(fTime,DISGAS_penalty_transposed,linestyle='--',
        label='Dissolved␣CO2␣Volume␣in␣the␣Penalty␣Area',color='red')
        plt.title('Dissolved␣CO2␣Volume␣Vs.␣Time')
        plt.xlabel('Time␣[years]')
        plt.ylabel('Dissolved␣CO2␣Volume␣[Sm3]')
        plt.legend()
        plt.show()
        plt.savefig('DISGAS␣Volume␣Profile.pdf')
        pdf_pages.savefig(fig7)

        #Dissolved CO2 Mass Plotting
        fig8=plt.figure(n+8)
        plt.plot(fTime,DISGAS_mass_transposed/1E9,linestyle='-',
        label='Dissolved␣CO2␣Mass␣in␣The␣Reservoir',color='black')
        plt.plot(fTime,DISGAS_penalty_mass_transposed/1E9,linestyle='--',
        label='Dissolved␣CO2␣Mass␣in␣the␣Penalty␣Area',color='red')
        plt.title('Dissolved␣CO2␣Mass␣Vs.␣Time')
        plt.xlabel('Time␣[years]')
        plt.ylabel('Dissolved␣CO2␣Mass␣[Mt]')
        plt.legend()
        plt.show()
        plt.savefig('DISGAS␣Mass␣Profile.pdf')
        pdf_pages.savefig(fig8)

        #CO2 Leakage Mass into the Penalty Area & in the Reservoir Plotting
        GV_mass=np.array(GV_mass_list)
        GV_mass_max=np.max(GV_mass)
        GV_mass_transposed=GV_mass.T
        print("Maximum␣GV␣in␣the␣Reservoir=␣", GV_max, "Sm3", "␣=␣",
        GV_mass_max/1E9, "Mt")
        fig9=plt.figure(n+9)
        plt.plot(fTime, GV_mass_transposed/1E9,linestyle='-',
```

```
        label='CO2 Mass in The Reservoir',color='blue')
        plt.plot(fTime, GV_penalty_mass_transposed/1E9,
        label='CO2 Mass in the Penalty Area',color='red')
        plt.title('CO2 Mass in The Reservoir and the in The Penalty Area')
        plt.xlabel('Time [years]')
        plt.ylabel('CO2 Mass [Mt]')
        plt.legend()
        plt.show()
        plt.savefig('CO2 Mass in the Penalty Area and in the Reservoir.pdf')
        pdf_pages.savefig(fig9)

        #Volume of Free CO2 Captured in the Reservoir (Free CO2 Volume Plotting)
        # print("Free SGAS List= ", SGAS3D_free)
        # print("Free Gas Volume= ", GV_free)
        GV_free=np.array(GV_free_list)
        GV_free_max=np.max(GV_free)
        GV_free_transposed=GV_free.T
        fig10=plt.figure(n+10)
        plt.plot(fTime,GV_free_transposed,linestyle='-', label='Free CO2 Volume',
        color='blue')
        plt.title('Free CO2 Volume in the Reservoir Vs. Time')
        plt.xlabel('Time [years]')
        plt.ylabel('CO2 Volume [Sm3]')
        plt.legend()
        plt.show()
        plt.savefig('Free CO2 Volume Profile.pdf')
        pdf_pages.savefig(fig10)

        #Mass of Free CO2 Captured in the Reservoir (Free CO2 Mass Plotting)
        # print("Free SGAS List= ", SGAS3D_free)
        # print("Free Gas Volume= ", GV_free)
        GV_free_mass=np.array(GV_free_mass_list)
        GV_free_mass_max=np.max(GV_free_mass)
        GV_free_mass_transposed=GV_free_mass.T
        fig11=plt.figure(n+11)
        plt.plot(fTime,GV_free_mass_transposed/1E9,linestyle='-',
        label='Free CO2 Mass',color='blue')
        plt.title('Free CO2 Mass in the Reservoir Vs. Time')
        plt.xlabel('Time [years]')
        plt.ylabel('CO2 Mass [Mt]')
        plt.legend()
        plt.show()
        plt.savefig('Free CO2 Mass Profile.pdf')
        pdf_pages.savefig(fig11)
        print("Free CO2 After the Monitoring Period=", GV_free[-1], "Sm3", " = ",
        GV_free_mass[-1]/1E9, "Mt")

        #Volume of the CO2 Captured in the Reservoir By the Trapping Mechanism
        Plotting (Trapped CO2 Volume Plotting)
        # print("Trapped SGAS List= ", SGAS3D_trapped)
        # print("Trapped Gas Volume= ", GV_trapped)
        GV_trapped=np.array(GV_trapped_list)
        GV_trapped_max=np.max(GV_trapped)
        GV_trapped_transposed=GV_trapped.T
        fig12=plt.figure(n+12)
        plt.plot(fTime,GV_trapped_transposed,linestyle='-',
        label='Trapped CO2 Volume',color='black')
        plt.title('Trapped CO2 Volume in the Reservoir Vs. Time')
        plt.xlabel('Time [years]')
        plt.ylabel('CO2 Volume [Sm3]')
```

```
plt.legend()
plt.show()
plt.savefig('Trapped␣CO2␣Volume␣Profile.pdf')
pdf_pages.savefig(fig12)

#Mass of the CO2 Captured in the Reservoir By the Trapping Mechanism
Plotting (Trapped CO2 Mass Plotting)
# print("Trapped SGAS List= ", SGAS3D_trapped)
# print("Trapped Gas Volume= ", GV_trapped)
GV_trapped_mass=np.array(GV_trapped_mass_list)
GV_trapped_mass_max=np.max(GV_trapped_mass)
GV_trapped_mass_transposed=GV_trapped_mass.T
fig13=plt.figure(n+13)
plt.plot(fTime,GV_trapped_mass_transposed/1E9,linestyle='-',
label='Trapped␣CO2␣Mass',color='black')
plt.title('Trapped␣CO2␣Mass␣in␣the␣Reservoir␣Vs.␣Time')
plt.xlabel('Time␣[years]')
plt.ylabel('CO2␣Mass␣[Mt]')
plt.legend()
plt.show()
plt.savefig('Trapped␣CO2␣Mass␣Profile.pdf')
pdf_pages.savefig(fig13)
print("Trapped␣CO2␣After␣the␣Monitoring␣Period=", GV_trapped[-1], "Sm3",
"␣=␣", GV_trapped_mass[-1]/1E9, "Mt")

#A comprehensive Plot Comparing Volume of CO2 Captured in the Reservoir
By Different Mechanisms
fig14=plt.figure(n+14)
plt.plot(fTime2,FGIT,linestyle='-', label='Total␣CO2␣Injected',
color='black')
plt.plot(fTime,GV_free_transposed,linestyle='-', label='Free␣CO2',
color='blue')
plt.plot(fTime,DISGAS_transposed,linestyle='-',label='Dissolved␣CO2',
color='red')
plt.plot(fTime,GV_trapped_transposed,linestyle='-', label='Trapped␣CO2',
color='orange')
plt.title('Captured␣CO2␣Volume␣By␣Different␣Mechanisms␣Vs.␣Time')
plt.xlabel('Time␣[years]')
plt.ylabel('CO2␣Volume␣[Sm3]')
plt.legend()
plt.show()
plt.savefig('Captured␣CO2␣Volume␣by␣Different␣Mechanisms.pdf')
pdf_pages.savefig(fig14)

#A comprehensive Plot Comparing Mass of CO2 Captured in the Reservoir
By Different Mechanisms
fig15=plt.figure(n+15)
plt.plot(fTime2,FGIT_mass/1E9,linestyle='-', label='Total␣CO2␣Injected',
color='black')
plt.plot(fTime,GV_free_mass_transposed/1E9,linestyle='-',
label='Free␣CO2',
color='blue')
plt.plot(fTime,DISGAS_mass_transposed/1E9,linestyle='-',
label='Dissolved␣CO2',
color='red')
plt.plot(fTime,GV_trapped_mass_transposed/1E9,linestyle='-',
label='Trapped␣CO2',
color='orange')
plt.title('Captured␣CO2␣Mass␣By␣Different␣Mechanisms␣Vs.␣Time')
plt.xlabel('Time␣[years]')
```

```
        plt.ylabel('CO2 Mass [Mt]')
        plt.legend()
        plt.show()
        plt.savefig('Captured CO2 Mass by Different Mechanisms.pdf')
        pdf_pages.savefig(fig15)

        #Perform a Compatibilty Test by the Total Injected Volume
        CO2_sum=DISGAS_transposed+GV_free_transposed+GV_trapped_transposed
        fig16=plt.figure(n+16)
        plt.plot(fTime2,FGIT,linestyle='-', label='FGIT',color='red')
        plt.plot(fTime,CO2_sum,linestyle='-', label='Total CO2 Volume Calculated',
        color='blue')
        plt.title('FGIT Volume By Simulator and Calculations Vs. Time')
        plt.xlabel('Time [years]')
        plt.ylabel('CO2 Volume [Sm3]')
        plt.legend()
        plt.show()
        plt.savefig('Calculated Injected CO2 Volume Compatibility Test.pdf')
        pdf_pages.savefig(fig16)

        #Perform a Compatibilty Test by the Total Injected Mass
        CO2_mass_sum=DISGAS_mass_transposed+GV_free_mass_transposed+
        GV_trapped_mass_transposed
        fig17=plt.figure(n+17)
        plt.plot(fTime2,FGIT_mass/1E9,linestyle='-', label='FGIT Mass',color='red')
        plt.plot(fTime,CO2_mass_sum/1E9,linestyle='-',
        label='Total CO2 Mass Calculated',color='blue')
        plt.title('FGIT Mass By Simulator and Calculations Vs. Time')
        plt.xlabel('Time [years]')
        plt.ylabel('CO2 Mass [Mt]')
        plt.legend()
        plt.show()
        plt.savefig('Calculated Injected CO2 Mass Compatibility Test.pdf')
        pdf_pages.savefig(fig17)

        #Contribute a Total SWAT List
        SWAT_penalty=np.array(SWAT_penalty_list)
        SWAT_penalty_min=np.min(SWAT_penalty)
        SWAT_penalty_transposed=SWAT_penalty.T
        SWAT=np.array(SWAT_list)
        SWAT_min=np.min(SWAT)
        SWAT_transposed=SWAT.T
        print("Minimum SWAT in the Penalty Area= ", SWAT_penalty_min)
        print("Minimum SWAT in the Reservoir= ", SWAT_min)

        #SWAT Plotting
        fig18=plt.figure(n+18)
        plt.plot(fTime,SWAT_transposed,linestyle='-',label='SWAT in the Reservoir',
        color='black')
        plt.plot(fTime,SWAT_penalty_transposed,linestyle='--',
        label='SWAT in the Penalty Area',color='blue')
        plt.title('Water Saturation Profile Vs. Time')
        plt.xlabel('Time [years]')
        plt.ylabel('SWAT [-]')
        plt.legend()
        plt.show()
        plt.savefig('SWAT Profile.pdf')
        pdf_pages.savefig(fig18)

        # Maximum Pressure Below the Cap Rock Till the End of the Injection Period
```

```
        P1_list=[]
        P1_max_list=[]
        for iTimeStep in range(0,injection+1):
            P1=hRestartFile.iget_named_kw('PRESSURE',iTimeStep)
            P13D=Grid.create_3d(P1)
            P13D_top_layer=P13D[:,:,(k_penalty_min-1):(k_penalty_min)]
            P13D_top_layer_max=np.max(P13D_top_layer)
            P1_list.append(P13D_top_layer)
            P1_max_list.append(P13D_top_layer_max)
        Pi=np.max(P1_list)
        print('Time Step=', injection, 'Maximum Pressure Below the Cap Rock Till
␣␣␣␣␣␣␣␣the End of the', "Injection", "Period= ", Pi, pressure_unit)
        # Maximum Pressure Below the Cap Rock at the End of the Monitoring Period
        P2_list=[]
        P2_max_list=[]
        for iTimeStep in range(injection+1,monitoring+1):
            P2=hRestartFile.iget_named_kw('PRESSURE',iTimeStep)
            P23D=Grid.create_3d(P2)
            P23D_top_layer=P23D[:,:,(k_penalty_min-1):(k_penalty_min)]
            P23D_top_layer_max=np.max(P23D_top_layer)
            P2_list.append(P23D_top_layer)
            P2_max_list.append(P23D_top_layer_max)
        Pm=np.max(P2_list)
        print('Time Step=', monitoring, 'Maximum Pressure Below the Cap Rock Till
␣␣␣␣␣␣␣␣the End of the', "Monitoring", "Period= ", Pm, pressure_unit)

        #Pressure Status Loop (Pc)
        if Pi>=Pm:
            Pc=Pi
            print("The highest pressure applies to the cap rock during the",
            "injection", "operation")
        else:
            Pc=Pm
            print("The highest pressure applies to the cap rock after the",
            "monitoring", "period")

        #Maximum Pressure Definition
        #Choose an array between the two values
        #Maximum Pressure Toward Cap Rock
        Pc_list=[P1_max_list+P2_max_list]
        Pc_top_layer=np.array(Pc_list)
        Pc_top_layer_transposed=Pc_top_layer.T
        # print("Pc List= ", Pc_top_layer_transposed)
        # print("Pc List Element= ", len(Pc_top_layer_transposed))
        print("Maximum Pressure enforced to the Cap Rock= ", Pc, pressure_unit)

        #Cep Rock Fracture Pressure
        CRFP=350
        print("Cap Rock Fracture Pressure= ", CRFP, pressure_unit)

        #Cap Rock Pressure Status Loop
        print("Cap Rock Pressure Status:")
        if Pc>=CRFP:
            print("Maximum Pressure enforced to the Cap Rock is",
            "higher than the cap rock fracture pressure")
        else:
            print("Maximum Pressure enforced to the Cap Rock is",
            "less than the cap rock fracture pressure")

        #Cap Rock Pressure Profile Plotting
```

```
        fig19=plt.figure(n+19)
        plt.plot(fTime,Pc_top_layer_transposed,label='Maximum pressure
␣␣␣␣␣␣␣␣enforced to the cap rock',color='blue')
        plt.title('Pressure Enforced to Cap Rock Vs. Time')
        plt.xlabel('Time [years]')
        plt.ylabel('Pressure [BARSA]')
        plt.legend()
        plt.show()
        plt.savefig('Cap Rock Pressure Profile.pdf')
        pdf_pages.savefig(fig19)


        #Reservoir Formation Fracture Pressure
        RFFP=500
        print("Reservoir Formation Fracture Pressure= ", RFFP, pressure_unit)

        #FPR Calling
        FPR=Sum.numpy_vector('FPR')
        FPR_max=np.max(FPR)
        print("Maximum FPR= ", np.max(FPR), pressure_unit)
        BHP1=Sum.numpy_vector('WBHP:'+well1)
        BHP1_max=np.max(BHP1)
        print("Maximum BHP of "+well1+"=", BHP1_max, pressure_unit)
        BHP2=Sum.numpy_vector('WBHP:'+well2)
        BHP2_max=np.max(BHP2)
        print("Maximum BHP of "+well2+"=", BHP2_max, pressure_unit)
        # FGIR=Smspec.numpy_vector('FGIR')
        # print("FGIR: ", FGIR)

        #Formation Pressure Status Loop
        print("Reservoir Formation Pressure Status:")
        if BHP1_max>=RFFP:
            print("Injection pressure in well "+well1+" is
␣␣␣␣␣␣␣␣␣␣␣␣higher than the formation fracture presure")
        else:
            print("Injection pressure in well "+well1+" is
␣␣␣␣␣␣␣␣␣␣␣␣less than the formation fracture presure")

        if BHP2_max>=RFFP:
            print("Injection pressure in well "+well2+" is
␣␣␣␣␣␣␣␣␣␣␣␣higher than the formation fracture presure")
        else:
            print("Injection pressure in well "+well2+" is
␣␣␣␣␣␣␣␣␣␣␣␣less than the formation fracture presure")

        if FPR_max>=RFFP:
            print("Field Average Pressure is higher than
␣␣␣␣␣␣␣␣␣␣␣␣the formation fracture presure")
        else:
            print("Field Average Pressure is less than
␣␣␣␣␣␣␣␣␣␣␣␣the formation fracture presure")

        # Close the Report File
        report_file.close()

        #3Dplotting
        x_values_list.append(r1_MT)
        y_values_list.append(r2_MT)
        condition_list.append(condition)
```

```
        # Close the Pdf Pages object
        n=n+20
        pdf_pages.close()
        r2_MT=r2_MT+(0.5)
        os.chdir("..")
    r1_MT=r1_MT+(0.5)
    os.chdir("..")
#3D Plotting
x_values=np.array(x_values_list)
y_values=np.array(y_values_list)
condition=np.array(condition_list)
fig=plt.figure(n+1)
color_mapping = {
    0.0: 'blue',
    0.2: 'yellow',
    0.3: 'orange',
    0.4: 'pink',
    0.5: 'magenta',
    0.6: 'purple',
    0.7: 'maroon',
    0.9: 'red'}

# Create a new array by replacing values with corresponding colors
new_condition = np.array([color_mapping[val] if val in color_mapping
else val for val in condition])
# print("Colors= ", new_condition)

# Iterate over the points and plot them
for k in range(len(x_values)):
    x = x_values[k]
    y = y_values[k]
    color = new_condition[k]

    # Plot the point
    plt.scatter(x, y, s=40, c=color)

    # Annotate the point with x,y values
    # plt.annotate(f'({x},{y})', (x, y), textcoords="offset points",
    xytext=(5, 5), ha='center',fontsize=7)

#  Set axis range limits
plt.xlim(x_values[0]-0.5, x_values[-1]+0.5)  # X-axis range: 2 to 5
plt.ylim(y_values[0]-0.5, y_values[-1]+0.5)  # Y-axis range: 2 to 5
# Set labels for the axes
plt.title('Wells␣Injection␣Rates␣vs.␣Build-up/No-Build-up')
plt.xlabel('EOS-5-7␣Rate␣(Mt/Year)')
plt.ylabel('EOS-5-C␣Rate␣(Mt/Year)')
plt.legend()
# Show the plot
plt.show()
plt.savefig('Rates␣Balance␣2D␣Plot.pdf')
os.chdir("..")
basecase_file.close()
```

Appendix B

CO_2 PVT Data

B.1 Span-Wagner Thermodynamic Properties of CO_2 in The Standard Condition

Table B.1: CO_2 Properties in standard condition [15]

Pressure (bar)	**Temperature** $(^\circ C)$	**Density** $\left(\frac{Kg}{m^3}\right)$
1.01325	15	1.871849

B.2 Span-Wagner Thermodynamic Properties of CO_2 in T = 98 °C

B.2.1 Table

Table B.2: CO_2 Properties in 98 °C condition[15]

Pressure (bar)	**Density** $\left(\frac{Kg}{m^3}\right)$	**FVF** $\left(\frac{rm^3}{Sm^3}\right)$
150	339.2598818	0.00551745
160	371.5183818	0.005038376
170	403.317185	0.004641135
180	434.0160957	0.004312857
190	463.1216168	0.004041809
200	490.3323815	0.003817511
210	515.5301064	0.003630921
220	538.7369052	0.003474515
230	560.0634069	0.00334221

Pressure (bar)	**Density** $\left(\frac{Kg}{m^3}\right)$	**FVF** $\left(\frac{rm^3}{Sm^3}\right)$
240	579.6650226	0.003229191
250	597.7122246	0.00313169
260	614.3729516	0.003046764
270	629.8032612	0.002972118
280	644.1431781	0.002905952
290	657.5157308	0.002846851
300	670.0278043	0.002793689
310	681.7718488	0.002745566
320	692.8278309	0.002701753
330	703.265095	0.002661655
340	713.1439957	0.002624785
350	722.5172736	0.002590733
360	731.431196	0.00255916
370	739.9265001	0.002529777
380	748.0391761	0.002502341
390	755.8011209	0.002476643
400	763.2406882	0.002452502
410	770.3831547	0.002429764
420	777.2511161	0.002408294
430	783.864826	0.002387975
440	790.2424864	0.002368702
450	796.4004982	0.002350387
460	802.3536764	0.002332948
470	808.1154374	0.002316314
480	813.6979602	0.002300423
490	819.1123271	0.002285217
500	824.368646	0.002270646
510	829.4761569	0.002256664
520	834.4433252	0.002243231
530	839.2779231	0.002230309
540	843.9871012	0.002217865
550	848.5774518	0.002205867
560	853.0550638	0.002194289
570	857.4255722	0.002183104
580	861.694201	0.00217229
590	865.8658017	0.002161824
600	869.9448878	0.002151687

B.2.2 Plot

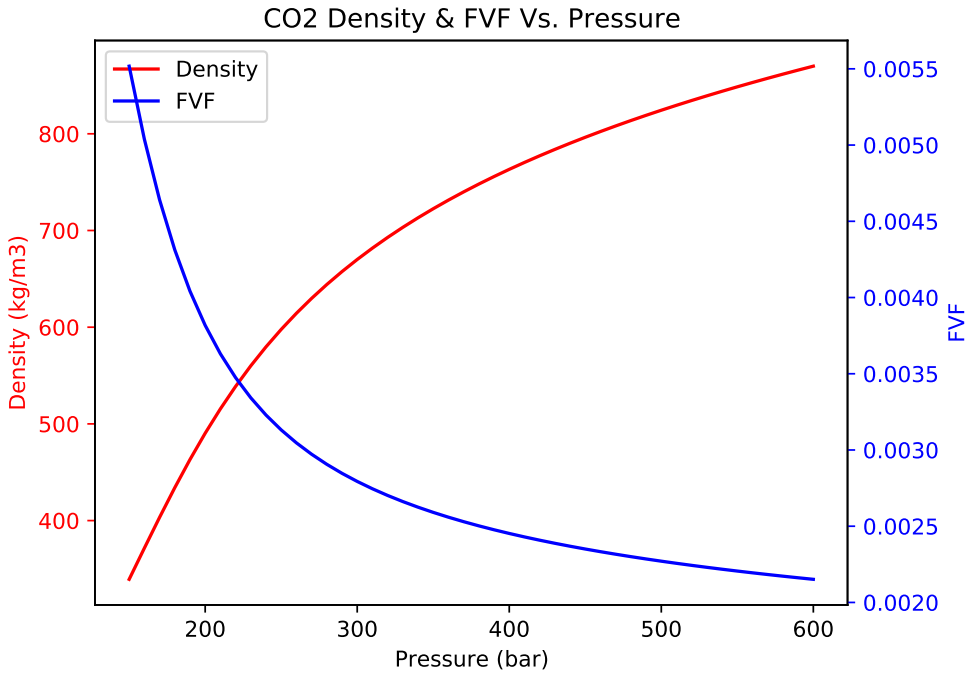


Figure B.1: CO_2 properties in 98 °C condition [15]

Appendix C

Equinor Well 31/5-7 Data

C.1 Well Drilling and Logging Data

The dataset for *Well 31/5-7* (*Eos*) is available to the public in the *'Open Data'* database (see the link [37].)

INTEGRATED BIOSTRATIGRAPHY OF THE EQUINOR ENERGY AS EXPLORATION WELL NO 31/5-7 "EOS", OFFSHORE NORWAY

Prepared by

Martin Pearce (coordinator), David Bailey, Paul Dodsworth,

Peter Morris and Stephen Packer

On behalf of the client

Equinor Energy AS

PROJECT NUMBER: 2020–09

APRIL, 2020

1. INTRODUCTION AND PROJECT SCOPE

The biostratigraphic analysis of the Equinor Energy AS exploration well 'Eos' N31/5-7 (60° 34' 35.15'' N, 3° 26' 36.15'' E) west of the Troll West field in the northern Stord Basin within the Horda Platform in the Norwegian North Sea (Figure 1), has been requested by the Equinor Energy AS. The study interval is the Hordaland Group to the Statfjord Group, as a part of an ongoing stratigraphic evaluation of CO_2 reservoir storage potential on the Horda Platform (Northern Lights Project), and includes the analysis of three cores from the Dunlin Group.

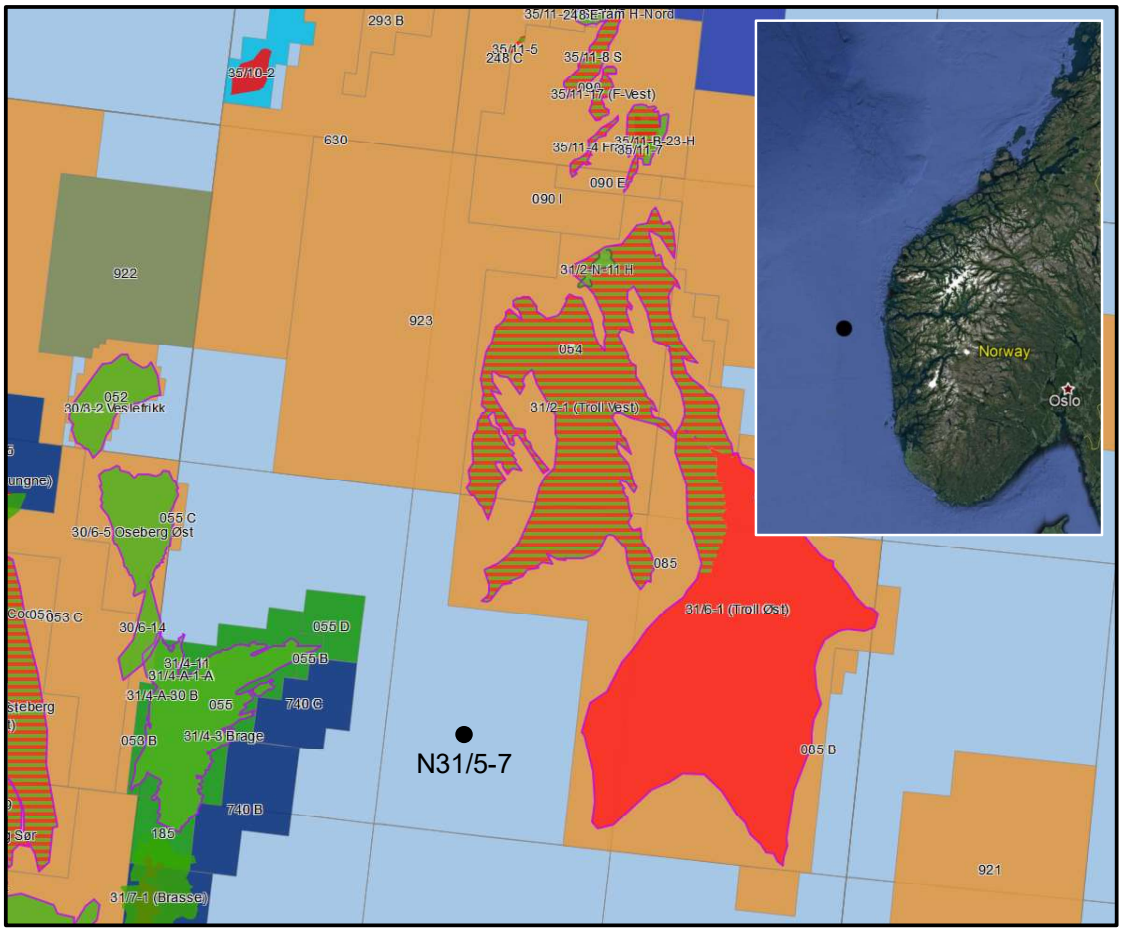


Figure 1: Location map of the N31/1-1 well. Source of main map © kartverket/norgeskart.no, source of inset map: Google Earth, Data SIO, NOAA, U.S. Navy, NGA, GEBCO).

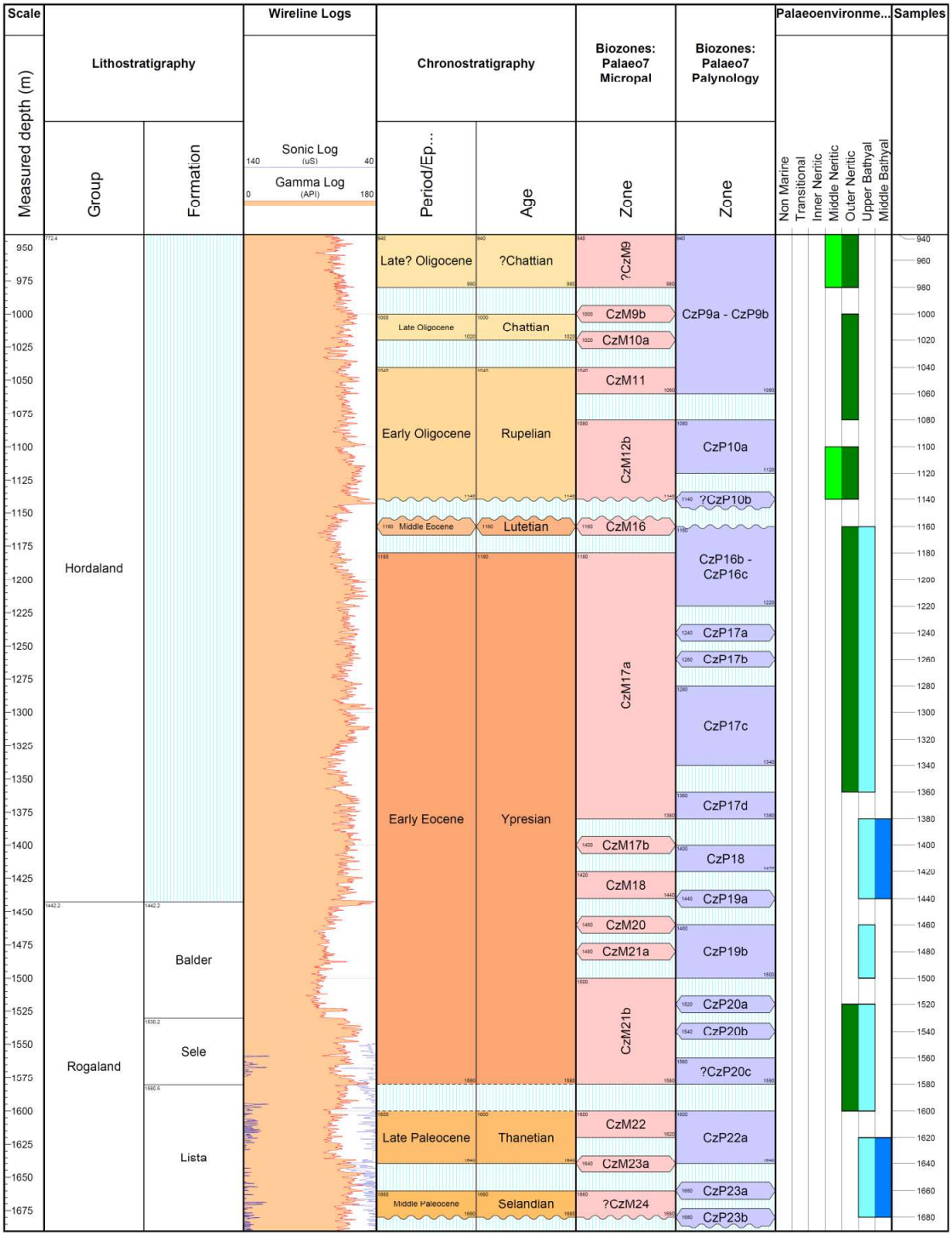


Figure 3. Biochronostratigraphic and palaeoenvironmental summary of the Hordaland to Rogaland groups of the NO 31/5-7 (Eos) well.

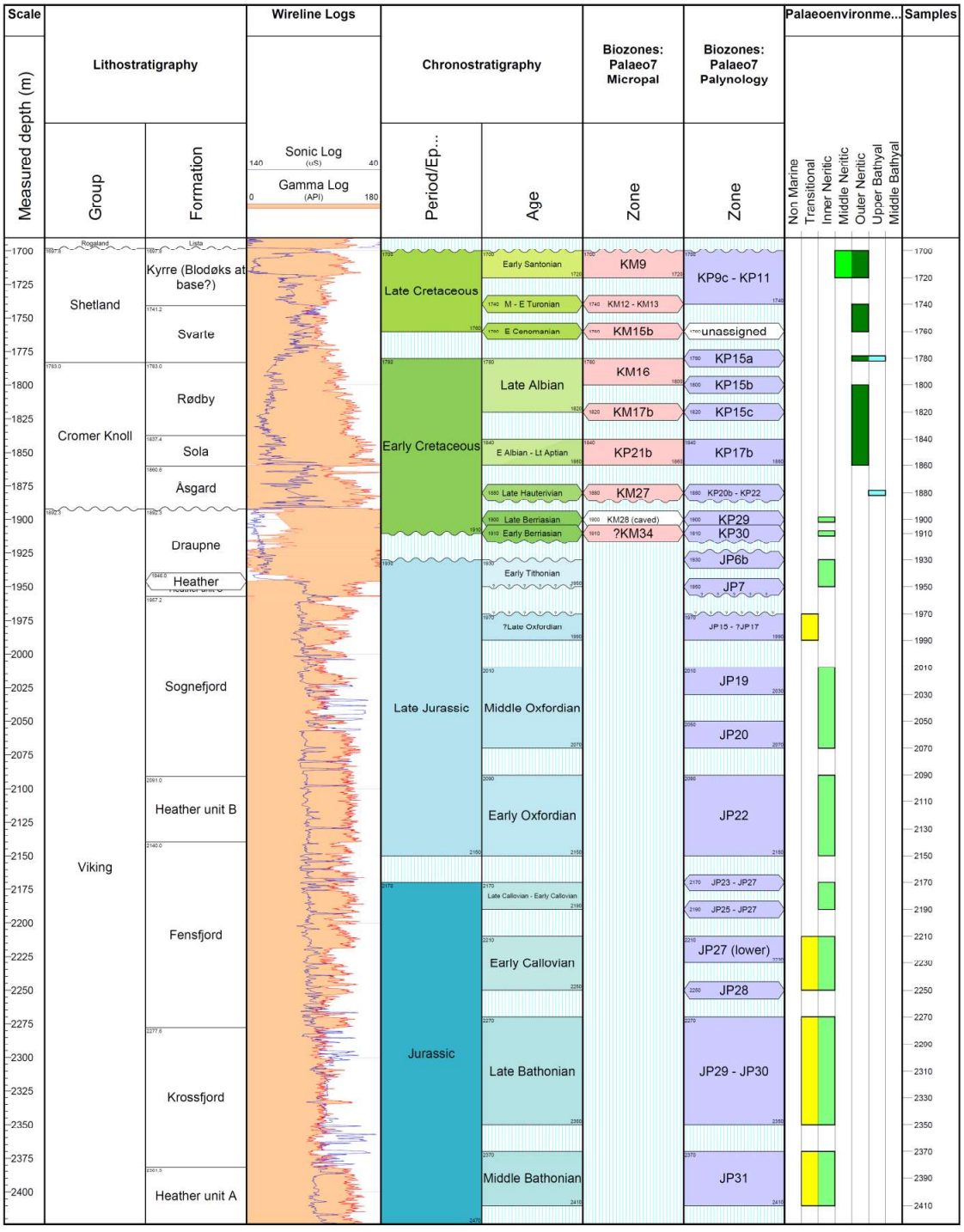


Figure 4. Biochronostratigraphic and palaeoenvironmental summary of the Shetland to Viking groups of the NO 31/5-7 (Eos) well.

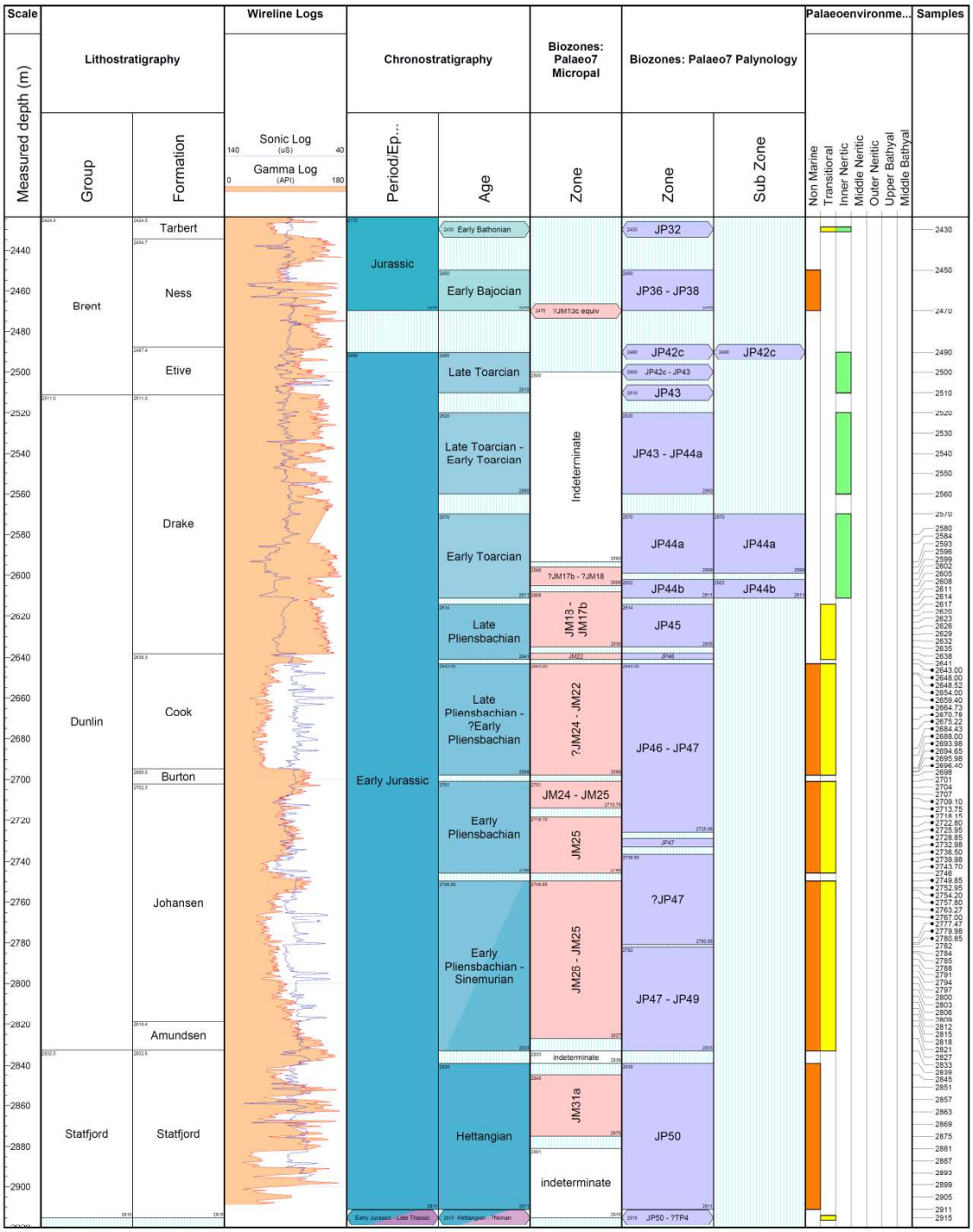


Figure 5. Biochronostratigraphic and palaeoenvironmental summary of the Brent to Statfjord groups of the NO 31/5-7 (Eos) well.

INTERVAL TOP DEFINED ON THE FOLLOWING EVENT(S):

The LO of *Halosphaeropsis liassica* OCC at 2210 m.

MICROPALAEONTOLOGY

Indeterminate Zone: Cuttings samples analysed from the interval continued to be dominated by carbonised and non-carbonised plant cuticle with no diagnostic megaspores / microfauna recovered. Notably no marine agglutinated foraminifera typical of the Drake Formation were present indicating the prevalence of non-marine, fluvio-deltaic derived sediments.

PALYNOLOGY

JP43 – 44a zones: Frequent specimens of the acritarch *Halosphaeropsis liassica* appear down hole at 2210 m, slowly increasing in numbers with inceasing depth. The quantities of *H. liassica* are insufficient to confidently indicate an Early Toarcian age and a ranging JP43 – 44a zonal assignment is appropriate. Apart from *Nannoceratopsis gracilis/senx*, dinocysts are sparse, though more common and diverse than the overlying interval. Additional dinocysts include *Valvaeodinium* spp. OCC/CMN, *Scriniocassis priscus* PRES at 2530 m, *Mendicodinium spinosum* and *Nannoceratopsis symmetrica*. The microfloras are dominated by miospores and debris of pteridophytic and bryophytic plants. There is a slight increase in numbers of *Cerebropollenites thiergartii* towards the base of the interval.

PALAEOENVIRONMENT

Marine, inner neritic.

LITHOSTRATIGRAPHY

Drake Formation of the Dunlin Group.

3.28. EARLY TOARCIAN

INTERVAL: 2570 m – 2611 m (10 samples from 41 m).

PALAEO7 MICROZONE(S): ?JM17b – 18 , JM17b – 18.

PALAEO7 PALYNOZONE(S): JP44a, JP44b.

INTERVAL TOP DEFINED ON THE FOLLOWING EVENT(S):

The LCO of *Halosphaeropsis liassica* at 2570 m.

SUPPORTING EVENT(S):

1. The LAO of *Halosphaeropsis liassica* at 2580 m;
2. The LO's of *Haplophragmoides* spp. and ?*Trochammina* spp. at 2596 m;
3. The LO of *Luehndea spinosa* at 2602 m;
4. The LCO of *Luehndea spinosa* at 2605 m
5. LO *Haplophragmoides* aff. *barrowensis* at 2608 m;
6. LO *Trochammina* aff. *topagorukensis* and LCO *Trochammina* cf. *canningensis* at 2611 m;
7. The FAO of *Halosphaeropsis liassica* 2611 m.

MICROPALAEONTOLOGY

?JM17b – 18 zones: At 2596 m the appearance of a poorly preserved agglutinant assemblage with LO's of *Haplophragmoides* spp. and ?*Trochammina* spp. suggest penetration of marine Toarcian sediments of the JM17b **–** 18 zones. Due to the poor preservation however, cavings/comtamination from higher in the section cannot be ruled out. Of note also is the LSAO of Algal cyst? sp. 2 occurring in the fines: this occurrence whilst not age diagnostic may be of local correlative significance.

JM17b – 18 zones: Cuttings from 2599 m **–** 2605 m continued to yield abundant/superabundant ?algal cysts in fine residues. At 2608 m the top occurrence of consistent agglutinant microfaunas was recorded with increasing down-hole diversity evident from this depth down to 2611 m and below. Species occurring including *Haplophragmoides* aff. *barrowensis*, *Trochammina* aff. *topagorukensis* and *Trochammina* cf. *canningensis* (CMN) mark penetration of the JM17b **–** 18 zones.

PALYNOLOGY

JP44a Zone: At 2570 m, the LCO of *Halosphaeropsis liassica* confirms the presence of the Early Toarcian JP44a Zone. This acritarch is associated with the Falciferum Global Anoxic Event and is abundant between 2580 m and 2611 m, commonly observed in large clusters (each cluster counted as a single specimen) and associated with superabundant amorphous organic material (AOM). Dinocysts are sparse and dominated by *N. gracilis*/*senex*, though also of note is the LO of "*Morgenrothia*" sp. 4 at 2599 m. Other marine taxa present include several types of acanthomorph acritarchs, *Tasmanites* and *Cymatiosphaera*, which are all observed quite frequently. The microfloras are all dominated by miospores and debris of pteridophytic and bryophytic plants, including significant numbers of *Classopollis torosus*, *Chasmatosporites*, "*Sangarelladinium asperum*", *Leiotriletes*, *Dictyophyllidites*, *Triplanosporites* and *Osmundacidites*.

JP44b Zone: The LO of *Luehndea spinosa* at 2602 m is good evidence for penetration of Early Toarcian sediments equivalent to JP44b. The LCO of *Luehndea spinosa*is seen in the underlying

sample at 2605 m. The assemblages include abundant *Halosphaeropsis liassica*, though they are increasingly likely to be caved towards the base of the Toarcian.

PALAEOENVIRONMENT

Marine, inner neritic.

LITHOSTRATIGRAPHY

Drake Formation of the Dunlin Group.

3.29. LATE PLIENSBACHIAN

INTERVAL: 2614 m – 2641 m (10 samples from 27 m).

PALAEO7 MICROZONE(S): JM17b – 18 (?caved); JM22.

PALAEO7 PALYNOZONE(S): JP45, JP46.

INTERVAL TOP DEFINED ON THE FOLLOWING EVENT(S):

The LO of *Mendicodinium reticulatum* at 2614 m.

SUPPORTING EVENT(S):

1. The LO of *Nannoceratopsis symmetrica* OCC, LO of the *Mancodinium semitabulatum* ACME and the LO of *Cerebropollenites thiergartii* OCC at 2614 m;
2. The LO of *Kraueselisporites reissingeri* at 2617 m;
3. The FO of the *Mancodinium semitabulatum* ACME at 2620 m.
4. The FCO of *Luehndea spinosa* at 2635 m;
5. The LO of *Haplophragmoides lincolnensis* at 2638 m;
6. The LCO of *Cerebropollenites thiergartii* at 2638 m.

MICROPALAEONTOLOGY

JM17b – 18 zones (?caved): Cuttings from 2614 m – 2635 m continued to yield common agglutinant microfaunas including *Trochammina* aff. *topagorukensis* and *T*. cf. *canningensis*, *T*. cf. *lathetica* and ? *Verneuilinoides* sp. indicative JM17b – 18 zones, as above. At 2632 m a marked increase in agglutinated microfaunal diversity and abundance was recorded with LSAO of *Haplophragmoides* aff. *barrowensis* and LO's of LO *H*. cf. *kingakensis*, *Trochammina* cf. *gryci* and *T*. cf. *sablei*. This diverse assemblage with notable absence of calcareous benthics is typical of younger

Early Toarcian, Drake Formation and is therefore, possibly caved given the palynological evidence for penetration of the JP45 Zone.

JM22 Zone: At 2638 m the LO of the marker species *Haplophragmoides lincolnensis* was recorded consistent with the Late Pliensbachian assignment. Agglutinant dominant microfaunas including *Haplophragmoides* aff. *barrowensis* range down to 2641 m defining the base of the JM22 zone.

PALYNOLOGY

JP45 Zone: At 2614 m, the LO of *Mendicodinium reticulatum* indicates a Late Pliensbachian age, equivalent to JP45, with supporting evidence provided by the LO of *Nannoceratopsis symmetrica* OCC, LO of the *Mancodinium semitabulatum* ACME and the LO of *Cerebropollenites thiergartii* OCC at 2614 m. In the underlying sample at 2617 m, the LO of LO of *Kraueselisporites reissingeri* further substantiates the age and zonal assignments. All samples are heavily contaminated with Early Toarcian caving. Forms which are obviously caved, such as *Halosphaeropsis liassica* are excluded from the count, but many of the recorded taxa are also likely to be caved, including many/all of the dinocysts, which are present down to the deepest cuttings sample above Core 2, in which dinocysts are virtually absent (see 3.30. below). Despite the abundant caving, the FO of the *Mancodinium semitabulatum* ACME and the FCO of *Luehndea spinosa* probably offer some correlative potential.

JP46 Zone: The LCO of *Cerebropollenites thiergartii* at 2638 m indicates penetration of the JP46 Zone. Samples are heavily contaminated with caved Early Toarcian taxa, which where possible are excluded from the count.

PALAEOENVIRONMENT

Transitional?

LITHOSTRATIGRAPHY

Lower beds of the Drake Formation and uppermost Cook Formation of the Dunlin Group.

3.30. LATE – EARLY PLIENSBACHIAN

INTERVAL: 2643.00 m – 2698 m (15 samples from 55 m).

PALAEO⁷ MICROZONE(S): JM22 – ?24.

PALAEO⁷ PALYNOZONE(S): JP46 – 47 (*pars*).

INTERVAL AGE DEFINED ON STRATIGRAPHIC RELATIONSHIPS

SUPPORTING EVENT(S):

1. The LAO of *Botryococcus* 2643.00 m;
2. The LO of *Kuqaia quadrata* at 2648.52 m.

MICROPALAEONTOLOGY

JM22 – ?24 zones: Core and cuttings samples analysed from the interval were dominated by carbonised and non-carbonised plant cuticle with the sporadic occurrence of the mesofossil *Kuqaia quadrata* and megaspores **–** *Trileites* sp. *Kuqaia quadrata* has its LO in the Late Pliensbachian becoming consistent regionally in the Early Pliensbachian (see below; Morris *et al*., 2009). Also of note is the single occurrence of *Trileites candoris* at 2670.78 m which is consistent with the age assignment.

PALYNOLOGY

JP46 – 47 zones (*pars*): The interval is assigned a ranging Early-Late Pliensbachian, JP46 **–** JP47 age, based on stratigraphic relationships between definitive Early and Late Pliensbachian intervals. There is a sudden and significant change in the microfloras in the uppermost core samples, which may be a coincidence, or may reflect masking of *in situ* assemblages by caving in the cuttings samples above Core 2. The core samples from throughout the Cook Formation are of virtually non-marine aspect. A minor marine influence is indicated by sparse, though consistent occurrences of acanthomorph acritarchs, *Tasmanites* and *Cymatiosphaera*, together with an isolated and questionable record of the dinocyst *Valvaeodinium flos* at 2675.22 m.

All assemblages are dominated, almost exclusively by terrestrial taxa, with large numbers of *Cerebropollenites thiergartii*, "*Sangarelladinium asperum*", *Concentrisporites*, *Osmundacidites*, *Leiotriletes* and *Triplanosporites*.

There are no significant changes in the composition of the microfloras associated with the upper boundary of the Burton Formation at 2695.0 m (log). Cuttings samples between cores 2 and 3, in the lower Burton and uppermost Johansen formations are heavily contaminated with Early Toarcian caving.

PALAEOENVIRONMENT

Transitional, intermitently non-marine.

LITHOSTRATIGRAPHY

Cook Formation (*pars*), Burton Formation (*pars*).

5. SAMPLE LIST

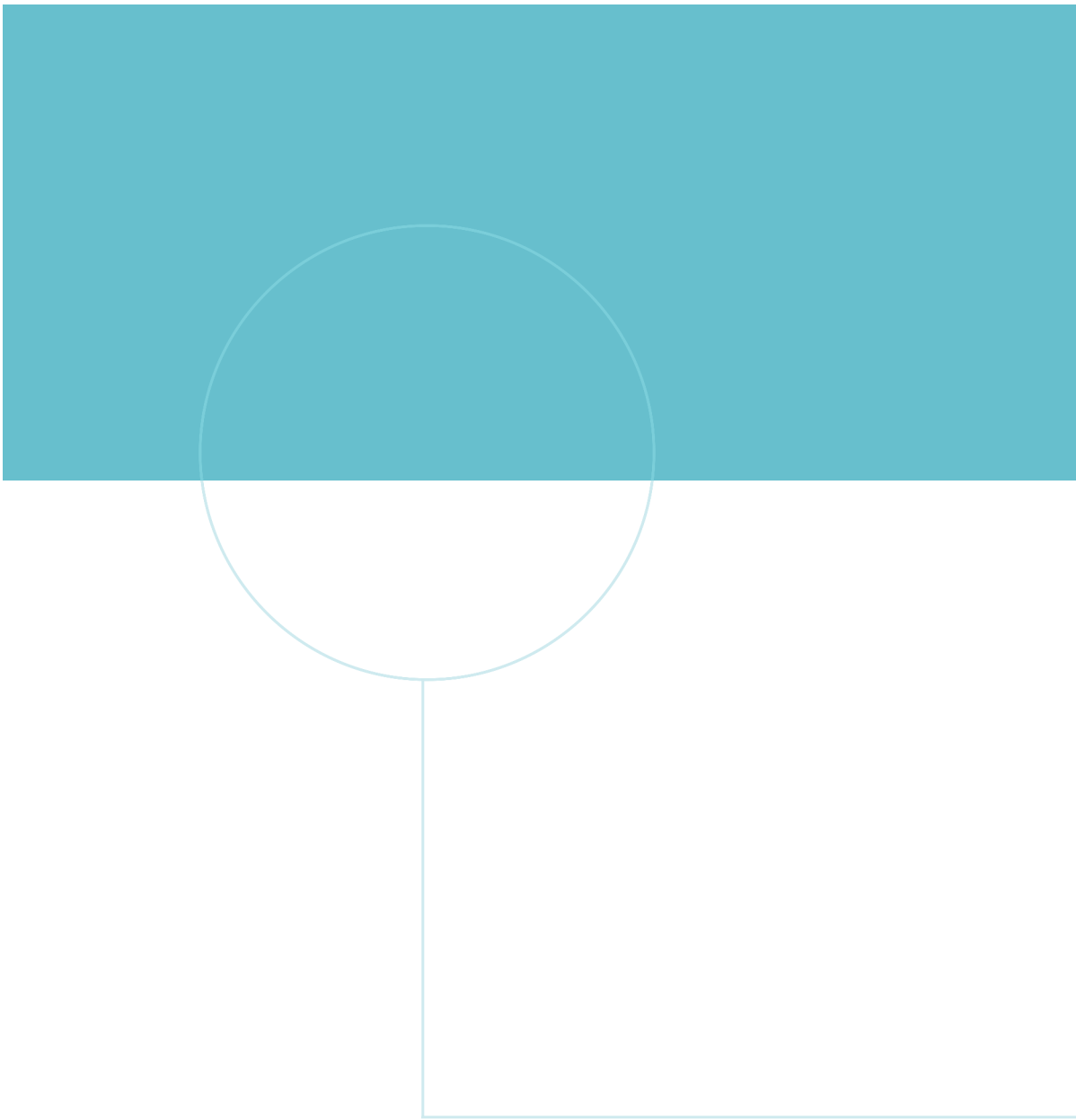
Ditch cuttings depths are listed as integers, core sample depths are listed to two decimal places, with depths in meters. 175 samples in total.

N31/5-7 (Eos)

940	M,P	1640	M,P	2330	P	2641	M,P	2767.00	M,P
960	M,P	1660	M,P	2350	P	2643.00	M,P	2777.47	M,P
980	M,P	1680	M,P	2370	P	2648.00	M,P	2779.98	M,P
1000	M,P	1700	M,P	2390	P	2648.52	M,P	2780.85	M,P
1020	M,P	1720	M,P	2410	P	2654.00	M,P	2782	M,P
1040	M,P	1740	M,P	2430	P	2659.40	M,P	2784	M,P
1060	M,P	1760	M,P	2450	P	2664.73	M,P	2785	M,P
1080	M,P	1780	M,P	2470	M,P	2670.78	M,P	2788	M,P
1100	M,P	1800	M,P	2490	P	2675.22	M,P	2791	M,P
1120	M,P	1820	M,P	2500	M,P	2684.43	M,P	2794	M,P
1140	M,P	1840	M,P	2510	M,P	2688.00	M,P	2797	M,P
1160	M,P	1860	M,P	2520	M,P	2693.98	M,P	2800	M,P
1180	M,P	1880	M,P	2530	M,P	2694.65	M,P	2803	M,P
1200	M,P	1900	M,P	2540	M,P	2695.98	M,P	2806	M,P
1220	M,P	1910	M,P	2550	M,P	2696.40	M,P	2809	M,P
1240	M,P	1930	P	2560	M,P	2698	M,P	2812	M,P
1260	M,P	1950	P	2570	M,P	2701	M,P	2815	M,P
1280	M,P	1970	P	2580	M,P	2704	M,P	2818	M,P
1300	M,P	1990	P	2584	M,P	2707	M,P	2821	M,P
1320	M,P	2010	P	2593	M,P	2709.10	M,P	2827	M,P
1340	M,P	2030	P	2596	M,P	2713.75	M,P	2833	M,P
1360	M,P	2050	P	2599	M,P	2718.15	M,P	2839	M,P
1380	M,P	2070	P	2602	M,P	2722.80	M,P	2845	M,P
1400	M,P	2090	P	2605	M,P	2725.95	M,P	2851	M,P
1420	M,P	2110	P	2608	M,P	2728.85	M,P	2857	M,P
1440	M,P	2130	P	2611	M,P	2732.98	M,P	2863	M,P
1460	M,P	2150	P	2614	M,P	2736.50	M,P	2869	M,P
1480	M,P	2170	P	2617	M,P	2739.98	M,P	2875	M,P
1500	M,P	2190	P	2620	M,P	2743.70	M,P	2881	M,P
1520	M,P	2210	P	2623	M,P	2746	M,P	2887	M,P
1540	M,P	2230	P	2626	M,P	2749.85	M,P	2893	M,P
1560	M,P	2250	P	2629	M,P	2752.95	M,P	2899	M,P
1580	M,P	2270	P	2632	M,P	2754.20	M,P	2905	M,P
1600	M,P	2290	P	2635	M,P	2757.80	M,P	2911	M,P
1620	M,P	2310	P	2638	M,P	2763.27	M,P	2915	M,P

M= micropalaeontology, P = palynology.

Finish



NTNU

Norwegian University of
Science and Technology