NTNU
Norwegian University of
Science and Technology

The Traveling Circus

Automated generation of parametric wellbore trajectories minimizing wellbore lengths for different subsea field layouts

Eirin Lillevik
Ingvild Evensen Standal

Petroleum Geoscience and Engineering
Submission date: June 2018
Supervisor: Tor Berge Gjersvik, IGP
Co-supervisor: Sigbjørn Sangesland, IGP

Norwegian University of Science and Technology
Department of Geoscience and Petroleum

Preface

This study is the result of our Master Thesis completing our Master's degree in Petroleum Engineering at the Norwegian University of Science and Technology (NTNU), with specialization in drilling. The thesis was written at the Department of Geoscience and Petroleum during the spring semester of 2018. The problem description and issues discussed were described and given by Professor Tor Berge Gjersvik.

Different subsea field architectures and their effect on drilling length are studied. The subject is highly relevant in present time, mainly to decrease the total costs of subsea field developments.

This Master Thesis is written to an audience familiar with the petroleum industry. Further, knowledge within the fields of subsea field development and drilling is beneficial, as no background theory is presented. In addition, the audience should be familiar with the project report, *Optimized Wellbore Trajectories*. This report was written during the fall semester of 2017 by Eirin Lillevik and Ingvild Evensen Standal. It was a starting point of the tool that is further developed in this Master Thesis.

Trondheim, 2018-06-11

Eirin Lillevik   Ingvild E. Standal

Eirin Lillevik   Ingvild Evensen Standal

Acknowledgment

We would like to thank our supervisor Professor Tor Berge Gjersvik for providing us with this topic. His guidance and suggestions along the way are highly appreciated.

We would also like to thank our co-supervisor Professor Sigbjørn Sangesland for taking his time to discuss the subject.

E.L.

I.E.S.

Summary

It is known from the media that Equinor is trying a new generation of unmanned wellhead platforms on the Oseberg field. According to them, this will cut costs dramatically. The reason is that dry well components are used. These are much cheaper than wet ones. They call the solution "Subsea on a Stick" (SoS) (Lorentzen, 2015). Gathering all wells on one platform means that all wells have to be drilled from the same position. This increases the average well path length (WPL) dramatically (Lillevik and Standal, 2017). How does this solution affect the drilling costs?

Lillevik and Standal (2017) initiated the way towards an automatic field development tool. The tool compares different subsea field layouts and identifies the optimal one. The optimal solution is the one that minimizes the sum of the drilling costs and the subsea hardware and installation costs. The program developed by Lillevik and Standal (2017) is further developed in this thesis. As a large part of the subsea field development costs are drilling related costs, this thesis focuses on comparing the drilling length in different field developments. This is done by studying five particular layouts: only satellite wells, 2-slots templates, 4-slots templates, 6-slots templates, and SoS. When templates are used, the optimal grouping of completion intervals is computed using the Traveling Circus Method (TCM). TCM identifies which completion intervals that should be reached from the same template to minimize the average WPL. In every field layout, the shortest drillable wellbore trajectories are constructed using trigonometric relations.

The comparison shows that the average and total drilling length are highly sensitive to the field development concept. Satellite wells yield the shortest average WPL and the lowest drilling cost. SoS, on the other hand, yields a significant increase in WPL and cost. Thus, the program developed in this thesis favors a field layout with satellite wells, without taking the costs of subsea hardware and installation into account. Subsea hardware and installation costs remain to be included to complete the automated tool and identify the optimal field layout. The extension of this work will include a development of a subsea EPCI (Engineering, Procurement, Construction, and Installation) program.

Sammendrag

Fra media er det kjent at Equinor prøver ut en ny generasjon plattform på Osebergfeltet. Den nye plattformen er en ubemannet brønnhodeplattform. Ifølge dem vil dette kutte kostnadene drastisk fordi brønnkomponentene er mye billigere når de kan stå tørt. Den nye løsningen kalles "Subsea on a Stick" (SoS) (Lorentzen, 2015). Når brønnene samles på én plattform må alle brønnene bores fra samme startpunkt. Dette øker den gjennomsnittlige borelengden betraktelig (Lillevik og Standal, 2017). Hvordan påvirker denne løsningen borekostnadene sammenlignet med en tradisjonell undervannsutbygginging?

Lillevik og Standal (2017) startet utviklingen av et automatisk feltutviklingsverktøy. Verktøyet skal sammenligne ulike havbunnsarkitekturer og identifisere den optimale. Den optimale løsningen er den som minimerer summen av borekostnader og kostnadene knyttet til undervannsutstyr og installasjon. Denne oppgaven fortsetter arbeidet med programmet som ble utviklet av Lillevik og Standal (2017). Siden en stor del av feltutviklingskostnadene er borekostnader, fokuserer denne oppgaven på å sammenligne borelengden i ulike feltarkitekturer. Fem ulike arkitekturer studeres: satelittbrønner, 2-slots brønnrammer, 4-slots brønnrammer, 6-slots brønnrammer og SoS. Når en rammearktitektur blir studert, brukes Traveling Circus Method (TCM) til å gruppere kompletteringsintervallene. TCM identifiserer hvilke kompletteringsintervall som skal bores fra samme brønnramme for å minimere den gjennomsnittlige brønnlengden. I alle feltarkitekturene blir de kortest mulige brønnbanene konstruert ved hjelp av trigonometriske beregninger.

Sammenligningen viser at gjennomsnittlig og total borelengde er følsomme for valg av feltutviklingskonsept. Satelittbrønner har kortest brønnlengde og minimerer borekostnadene. SoS derimot, fører til en betraktelig økning i brønnlengde og borekostnader. Kostnader knyttet til undervannsutstyr og installering må inkluderes for å kunne identifisere det optimale feltutviklingskonseptet for ulike felt. Videre arbeid inkluderer utvikling av et subsea EPCI (Engineering, Procurement, Construction and Installation) program.

Table of Contents

List of Figures

List of Tables

Abbreviations

BUA	Build-up angle
BUR	Build-up rate
DC	Drill center
DOR	Drop-off rate
DRILLEX	Drilling expenditure
EPCI	Engineering, Procurement, Construction and Installation
FBS	Flow base structure
FD1	`fielddata1.mat`
FD2	`fielddata2.mat`
HOST	Hinge-over subsea template
ITS	Integrated template structure
KOP	Kickoff point
MD	Measured depth
NCS	Norwegian continental shelf
ROC	Radius of curvature
ROP	Rate of penetration
ROT	Radius of turn
SoS	Subsea on a Stick
TCM	Traveling Circus Method
TR	Turn rate
TSM	Traveling Salesman Method
TSP	Traveling Salesman Problem
TVD	True vertical depth
UTA	Umbilical termination assembly
WPL	Well path length
XT	Christmas tree

Chapter 1

Introduction

1.1 Background

4-slots templates have become a standard subsea solution at the Norwegian continental shelf (NCS). Tying four and four wells into templates increases the average well path length (WPL). Increased drilling lengths increase the well construction costs accordingly. By distributing the wells and placing the wellheads into satellites/cluster layouts, the average WPL will decrease. On the other hand, the distribution causes more complex piping and more subsea structures are required. Additionally, there are costs related to frequent replacement of the drilling rig (Lillevik and Standal, 2017, Chapter 1.1).

Lillevik and Standal (2017) mention that Equinor is trying a new generation of platforms, Subsea on a Stick (SoS). This platform gathers all wellheads at one unmanned platform, thus all wells have to be drilled from the same position. This will increase the average WPL significantly. On the other hand, dry well components are used. Since these are cheaper than wet ones, the new platform will cut costs dramatically according to Equinor (Lorentzen, 2015).

A large part of the subsea field development costs are drilling related costs. Consequently, there is a need to compare the drilling length in different subsea field layouts. In addition, there are different expenditures related to hardware and installation in each field layout. A tool that identifies the optimal field layout, minimizing the total field development cost, is necessary (Lillevik

& Standal, 2017, Chapter 1.1).

Lillevik and Standal (2017) initiated the way towards an automatic tool that finds the optimal subsea field layout. The costs related to drilling were investigated. The work was initiated by developing a tool that studies 12 given completion intervals. Four different field architectures were compared and the wellbore trajectories in each layout were optimized. Finally, the average WPL in each of the following field layouts were obtained (Lillevik and Standal, 2017, Chapter 1.2) :

- Satellite wells.
- One drill center (SoS).
- Two drill centers (two 6-slots templates).
- Three drill centers (three 4-slots templates).

As only 12 completion intervals were compared, there is a need to continue the work related to the drilling costs in the tool. Most fields on the NCS have more than 12 wells and the tool should consequently handle a random number of completion intervals. Increasing the number of completion intervals will increase the differences in average WPLs. Furthermore, a new field layout is included in the study, as the use of 2-slots templates are increasing.

In the field layouts with templates, the completion intervals that should be reached from the same template must be identified. The method developed by Lillevik and Standal (2017) identifies every possible combination of 4 or 6 completions out of 12. When increasing the number of completions, or decreasing the number of slots per template, the number of possible combinations increases. The number of combinations raises concern and a satisfying method that solves this problem of combinatorics must be developed.

Another limitation mentioned by Lillevik and Standal (2017) is that the program only handles horizontal completion intervals. This restriction should be removed because most wells today have an inclined completion interval. In addition, Lillevik and Standal (2017) constructed the wells with one input build-up rate (BUR). The user should be able to decide the build-up rate in both build sections and the drop-off rate when required.

One of the most critical concerns about the method used by Lillevik and Standal (2017), is that there is no maximum limit for the turn rate (TR). The TR increases unlimited when required (Lillevik and Standal, 2017, p. 19). Excessive TRs cause trouble during drilling and completion. In some cases, the TRs may also be adjusted to unrealistic rates. Therefore, a maximum TR must be defined.

1.2 Objectives

Lillevik and Standal (2017) initiated the way towards an automatic tool that identifies the optimal field layout, by minimizing the average WPL. The optimized wellbore trajectories in each field layout were constructed, and the average WPLs were compared. The aim of this Master Thesis is to complete the drilling aspect of this tool. To complete it, a layout with 2-slots templates must be introduced, a random number of completion intervals must be handled, and the other improvements mentioned above must be implemented. The purpose of this Master Thesis is to compare the average WPLs of a random number of completion intervals in five different field layouts. The results will later be significant in the decision of choosing the optimal field development.

To complete the drilling part of this field development tool, the objectives are:

1. Make the program applicable for a random number of completion intervals.

2. Find a method that efficiently eliminates poor template combinations and identifies favorable template combinations.

3. Remove the limitation of only horizontal completion intervals.

4. Construct wells that allow for different build-up rates when two build sections are required.

5. Construct wells that allow for a drop section when required.

6. Introduce a maximum turn rate.

7. Create a layout with 2-slots templates that can be compared with the other field layouts.

All procedures and calculations are set up to minimize the WPL, since longer wells increase the costs (Lillevik and Standal, 2017, Chapter 1.2). All calculations are implemented in MATLAB. The tool has the following input parameters:

- Completion interval start coordinates (X_{cs}, Y_{cs}, Z_{cs}).
- Completion interval end coordinates (X_{ce}, Y_{ce}, Z_{ce}).
- Build-up rate for the first build section (BUA).
- Build-up rate for the second build section (BUA2).
- Drop-off rate for the drop section (DOR).
- Vertical depth of kick-off point (KOPz).
- Preferred turn rate (TR).
- Maximum turn rate (TR_max).

The completion interval coordinates must be UTM-coordinates. The depth coordinates have to be defined positive, where zero is sea floor level.

1.3 Limitations

The main limitation of this Master Thesis is the running time in MATLAB. When analyzing which completion intervals that should be reached from the same template, different combinations of completion intervals are studied. As the number of completion intervals increases, the number of combinations increases at the same time. More importantly, as the number of slots per template decreases, the number of combinations increases significantly. As the number of combinations increases, the running time increases simultaneously. Depending on the computer that is used, running the program for 30 completion intervals in a field layout with 2-slots templates takes several hours. To sum up, computing the optimal combination of completion intervals is time demanding, particularly in the case of a 2-slots template architecture. The process of computing all combinations of interest is extensive and time-consuming.

An essential limitation is the selection of satellite wells. Since the program is based on the codes developed by Lillevik and Standal (2017), the calculations are performed on groups of 12 completion intervals at a time. Thus, the number of required satellite wells in the layouts with templates depends on the remainder after division by 12. For example, if the user studies the drilling length of 35 completion intervals, 11 satellite wells are required in the template architectures. This affects the results since the differences in average WPL between template architectures and SoS will increase when the number of satellite wells increases.

Another limitation is the formulas for calculating the drill center (DC) placements. Lillevik and Standal (2017) based the formulas on the assumption that the completion intervals are located at the same depth. The DC placements are calculated from the arithmetic mean of the XY coordinates, neglecting the Z coordinates (Lillevik and Standal, 2017, Chapter 2.1). When the completion intervals that are drilled from the same DC have different depths, it will impact the DC location, as the goal is to minimize the average WPL (Lillevik and Standal, 2017, Chapter 1.3). This limitation has not been prioritized as wells that produce to the same template produce from the same reservoir. Thus, their depths are approximately the same.

1.4 Approach

The approach is to first identify a method that efficiently computes an optimal combination of completion intervals. It must handle a random number of completion intervals above 11, and group these into different template architectures. This is done on a trial and error basis. At first, the method developed by Lillevik and Standal (2017) is tested. Then, a self-developed method called the Grid Method is investigated. Further, a method based on the Traveling Salesman Problem (TSP) is studied. Finally, a method based on the TSP and the approach developed by Lillevik and Standal (2017) is used, called the Traveling Circus Method (TCM). The optimal combination of completion intervals is the combination that yields the shortest total distance between the completion interval start coordinates and the associated DCs. This combination generates the shortest wellbore trajectories. Additionally, the required TRs are considered when identifying the optimal combination.

When the TCM is developed, trigonometric relations are identified to enable construction of wells with non-horizontal completion intervals. The new trigonometric relations also allow for two different build-up rates by introducing new input parameters.

Further, a maximum TR criteria is added to the program to make sure that the wells will be drillable. Some combinations may yield a DC location that lies too close to the completion intervals. Such combinations require a TR higher than the maximum allowed TR. In the case of one common DC, the DC is moved away from the completions that are too close. In the case of several templates, alternative combinations of completion intervals are identified. First, the second best combination is tested, then the third best combination is tested, and so on, until it finds a combination that has wells with TRs within the TR criteria.

At last, a 2-slots template architecture is introduced and the optimal combination is computed using the TCM.

1.5 Structure of the Report

The first chapter introduces the study with problem definition and approach. The second chapter describes different solutions that were considered to achieve the first two objectives. The third chapter shows how the program developed by Lillevik and Standal (2017) was improved to achieve the remaining objectives. The fourth chapter presents the results. The results are discussed in the fifth chapter. A conclusion is made in Chapter 6, including recommendations for further work.

Chapter 2

Methods

The purpose of this thesis is to develop a program that compares the average well path length (WPL) in five different field layouts. The wellbore trajectories in each layout are optimized. The following field layouts are considered:

- Only satellite wells
- Subsea On a Stick (SoS)
- 2-slots templates
- 4-slots templates
- 6-slots templates

Lillevik and Standal (2017) considered 12 completion intervals. In the case of template architectures, every possible combination of 4 or 6 completion intervals out of 12 were studied. The new program will handle a random number of completion intervals above 11 and new combinations must be set up. Each combination represents a template arrangement, as the wells that will be drilled from the same template are set up systematically. The main problem is to efficiently identify the combination of completion intervals that minimizes the average WPL. This chapter explains various methods that are tested. The methodology of each method and the problems that arise are explained and illustrated.

2.1 Matrix Method

This section explains the combination method developed by Lillevik and Standal (2017), called the Matrix Method. Their program was made for only 12 completion intervals. These were investigated and every possible combination of completion intervals were tested. Since the new program will handle a random number of completion intervals above 11, the Matrix Method results in problems. The aim of this section is to illustrate the problems that arise.

Lillevik and Standal (2017) used the Matrix Method to find the optimal grouping of completion intervals. A field layout with two 6-slots templates and a field layout with three 4-slots templates were studied. The Matrix Method was applied to distribute the 12 completion intervals optimally between the templates, with respect to minimizing the average WPL. To find the optimal combination, every possible combination was identified. When investigating a random number of completion intervals using the Matrix Method, two main problems arise:

- The maximum variable size allowed by MATLAB is exceeded.
- A new code must be tailor-made for each random number of completion intervals.

To illustrate the problems, the number of possible combinations are studied. Using the binomial coefficient in Equation 2.1, the number of possible combinations is calculated. The equation yields the number of combinations of n elements taken k at a time. Table 2.1 lists some of the combinations of interest.

$$\text{binomial coefficient} = \binom{n}{k} = \frac{n!}{k!(n-k)!} \tag{2.1}$$

Table 2.1

Possible combinations of completion intervals for three different field layouts.

Completion Intervals	Field layout	Possible Combinations
12	6 slots templates	924
12	4 slots templates	34650
24	6 slots templates	3.247×10^{15}
24	2 slots templates	1.515×10^{20}
96	6 slots templates	1.901×10^{104}
100	4 slots templates	2.916×10^{123}
100	2 slots templates	8.289×10^{142}

Table 2.1 shows that if the number of completion intervals is increased from 12, the number of combinations increases drastically. The size of these variables is not feasible in MATLAB. This is shown by the following example. The example shows how the combinations are set up by Lillevik and Standal (2017) and how they would be set up if the same method is applied in the new program.

Every combination is set up in matrices. The following matrices illustrate the case of 12 completion intervals in a subsea field with three 4-slots templates. The completions are numbered from 1-12. R_1 represents the possible combinations of completions in the first template. The remaining completions are represented in a new matrix called R_rest.

$$\text{R\_1},\ [495 \times 4] = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix}$$

$$\text{R\_rest},\ [495 \times 8] = \begin{pmatrix} 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\ 1 & 3 & 4 & 2 & 9 & 10 & 11 & 12 \\ 1 & 2 & 7 & 8 & 3 & 4 & 5 & 6 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix}$$

For each combination (row) in R_1, there are 70 different combinations in the second template. R_2 represents the possible combinations of completions in the second template. The next matrix, R_3, represents the possible combinations of completions in the third template. R_2 and R_3 have the same dimensions since each combination in R_2 has one combination of the remaining completions. The rows in R_2 and R_3 have the same color to show that they belong together in a combination, combined with the corresponding row in R_1. See Lillevik and Standal (2017) for the details of the matrix generation and their structures.

$$\text{R\_2, } [70 \times 1980] = \begin{pmatrix} 5 & 6 & 7 & 8 & 1 & 2 & 4 & 3 & \dots \\ 6 & 7 & 8 & 9 & 2 & 3 & 4 & 9 & \dots \\ 7 & 8 & 9 & 10 & 3 & 4 & 9 & 10 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

$$\text{R\_3, } [70 \times 1980] = \begin{pmatrix} 9 & 10 & 11 & 12 & 9 & 10 & 11 & 12 & \dots \\ 5 & 10 & 11 & 12 & 1 & 10 & 11 & 12 & \dots \\ 5 & 6 & 11 & 12 & 1 & 2 & 11 & 12 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

The generation of the matrices above is implemented in a function called `get_three_dc`, see Appendix B.5. It is tailor-made for 12 completions and 4-slots templates. If the same approach is used in the new program, new codes have to be implemented. For each random number of completion intervals, three codes are required. One for 2-slots templates, one for 4-slots templates, and one for 6-slots templates. Each code has to be tailor-made to every unique problem. The following example illustrates the case of 100 completion intervals in a subsea field with 25 4-slots templates. Compared to the previous example, this example shows how the dimensions of the matrices change.

$$\text{R\_1, } [3921225 \times 4] = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 5 \\ 3 & 4 & 5 & 6 \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix}$$

$$\text{R\_rest, } [3921225 \times 96] = \begin{pmatrix} 5 & 6 & \dots & \dots & \dots & 98 & 99 & 100 \\ 6 & 7 & \dots & \dots & \dots & 99 & 100 & 1 \\ 7 & 8 & \dots & \dots & \dots & 100 & 1 & 2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix}$$

$$\text{R\_2, } [3321960 \times 15684900] = \begin{pmatrix} 5 & 6 & 7 & 8 & 6 & 7 & 8 & 9 & \dots \\ 6 & 7 & 8 & 9 & 7 & 8 & 9 & 10 & \dots \\ 7 & 8 & 9 & 10 & 8 & 9 & 10 & 11 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

MATLAB fails to generate the combinations in template 2 (R_2) because of its size. This is just the start of the combination generation, and no further calculations are possible. The conclusion is to use a new method that MATLAB can handle.

2.2 Grid Method

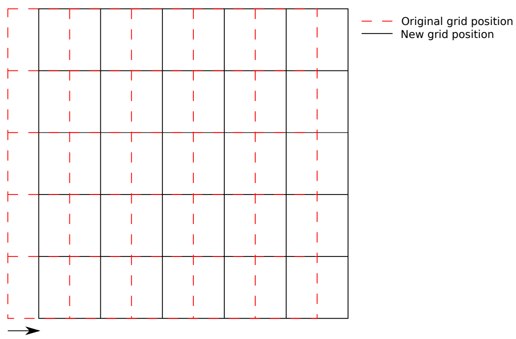
In the Matrix Method, every combination is identified, thus the completion intervals that confine the subsea field will be in groups together. For example, the three most southern placed wells will be in a group with the most northern placed well. The combinations of such groups are unnecessary. They will cause long wells, because one group of wells share one common drill center (DC) and they are drilled from the same template. In the example above, the most northern well will extend from one side of the field to another (this can also cause collision problems and long wells are expensive to drill). The Grid Method is assumed to exclude such groups, thus reduce the number of combinations.

2.2.1 Concept

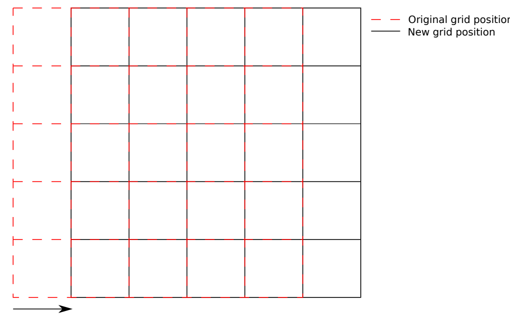
The concept of the Grid Method is to group completions by placing a grid over the subsea field. The completions that are in the same grid quadrant are associated with the same template. Each group's DC is calculated and the wells are constructed. Then the average WPL is calculated. The calculations are performed as they are in the program developed by Lillevik and Standal (2017).

The grid is moved over the subsea field, to find a combination of completion intervals that minimizes the average WPL. When the grid moves, new groups are formed. When the grid has moved enough times to cover the entire field, the combination that yields the lowest average WPL is identified. The grid should also be rotated from different axes to obtain more combinations.

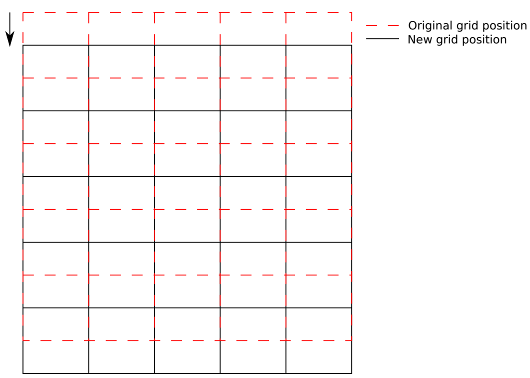
Figure 2.1 illustrates the stepwise movement of the grid. First, the grid moves a certain distance to the right. It moves to the right until it repeats itself. Then the grid moves a certain distance down. The grid is then moved to the right again, until it repeats itself. The grid has covered the entire field when it has moved enough times to repeat itself in the vertical direction.



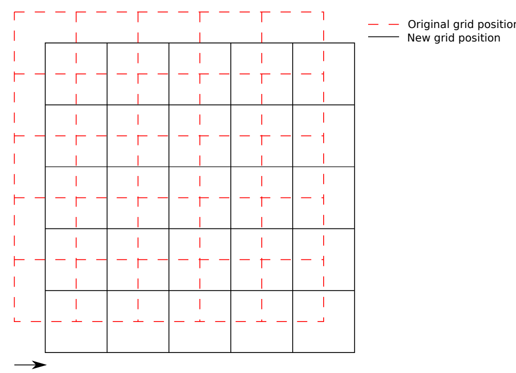
(a) Grid moving to the right.



(b) Grid repeating itself.



(c) Grid moving down.



(d) Grid moving to the right.

Figure 2.1. Stepwise movement of grid in the Grid Method. It moves to the right, one step at a time. When it repeats itself, it moves down one step. Then, it moves to the right one step at a time, and so on.

2.2.2 Creating the Grid

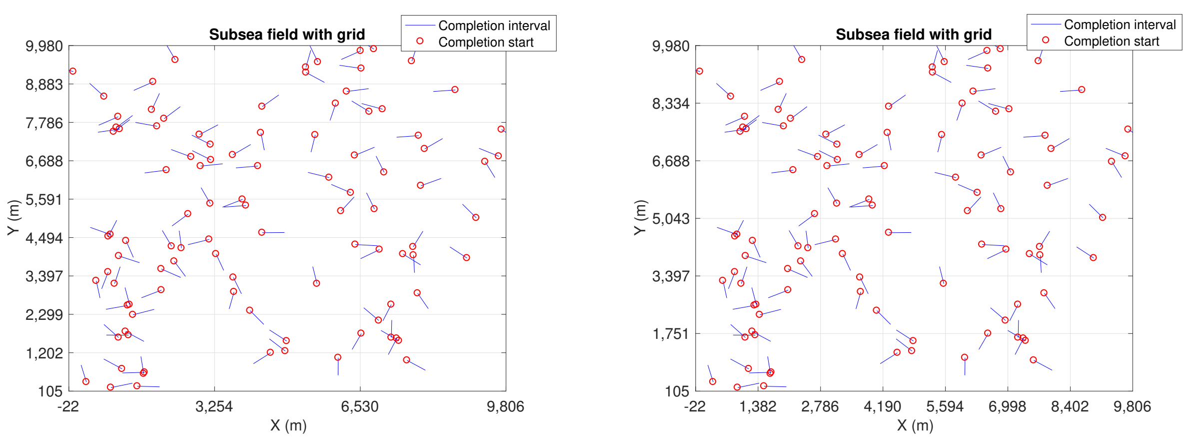
The grid boundaries are created from the minimum and maximum values of the completion start coordinates. First, the minimum and maximum X- and Y-values are obtained from the `min` and `max` functions in MATLAB. The minimum values are rounded to the nearest integers towards minus infinity by the `floor` function, then 100 is subtracted from these values. The maximum values are rounded to the nearest integer towards infinity by the `ceil` function, then 100 is added to these values. Subtracting and adding 100 are necessary to later identify which grid quadrant each completion lies in. If the completion coordinates lie on the edge of the grid, the `find` function used to locate the completion intervals does not work.

The grid is created with a quadrant size of $m \times n$. Equation 2.2 and 2.3 show how the size of m and n is calculated. The number of quadrants on the X- and Y-axes are decided by the parameters k and l, respectively.

$$n = \frac{x_{max} - x_{min}}{k} \tag{2.2}$$

$$m = \frac{y_{max} - y_{min}}{l} \tag{2.3}$$

Several grids are made and compared to each other to find the optimal grid. The parameter k varies between 2 and 10, while l varies between 1 and 10, thus 72 $\big((10-2) \times (10-1)\big)$ different grids are compared. In the case of a subsea field with only 4-slots templates, the optimal grid is the one that has the most groups of four completions. If two grids have the same number of groups of four, the most square grid is the most optimal. This is because of the assumption that the more rectangular the grid is, the higher will the turn rate (TR) be, because the completion starts are more likely to lie on a straight line. In addition, the completions are more gathered inside a quadratic grid. See Figure 2.2 for a comparison of two types of grids. 100 completion intervals were randomly made and both grids made eight groups of four completions.



(a) Grouping with a rectangular grid.

(b) Grouping with a quadratic grid.

Figure 2.2. Grouping of completions with different shape of grid quadrants. The number of completions within one grid quadrant changes depending on the shape of the grid quadrant.

2.2.3 Challenges

The number of completions within one grid quadrant is random. If the Grid Method is used to find wells that will be drilled from a 4-slots template, the completions inside the grid must be distributed, so that all groups have four completions. The main challenge is to find a distribution strategy that is efficient and easy to implement in MATLAB. An example of a strategy is presented below.

As mentioned in Chapter 2.2.2, the number of completions within one grid quadrant are counted to find the optimal grid. This number is placed in a matrix, N_c. This is a matrix of size $l \times k$, because each element in the matrix corresponds to one quadrant in the grid. The number in the uppermost left corner corresponds to the number of completions within the grid quadrant in the uppermost left corner, and so on. The matrix below is the resulting matrix of the quadratic grid in Figure 2.2b.

$$\mathrm{N\_c} = \begin{pmatrix} 2 & 2 & 0 & 3 & 4 & 1 & 1 \\ 4 & 4 & 4 & 3 & 3 & 3 & 2 \\ 0 & 2 & 4 & 1 & 4 & 2 & 2 \\ 5 & 4 & 2 & 1 & 2 & 3 & 1 \\ 5 & 2 & 3 & 1 & 1 & 2 & 0 \\ 5 & 3 & 0 & 3 & 2 & 4 & 0 \end{pmatrix}$$

The next step is to distribute the completions to make the matrix elements equal to 0 or 4. Figure 2.3 shows an example of how this can be done. The ninth element (third row and second column) in N_c is equal to 2. By considering the surrounding elements, completions are distributed to make the ninth element equal to 4. The example starts at the element above the ninth element and moves clockwise. If the element is equal to 0 or 4, there are no completions to distribute. There are completions to distribute from the 16th element (fourth row and third column). These are added to the ninth element and subtracted from the 16th element.

$$
\begin{array}{ccc} 4 & 4 & 4 \\ 0 & 2 & 4 \\ 5 & 4 & 2 \end{array}
\qquad
\begin{array}{ccc} 4 & 4 & 4 \\ 0 & 2 & 4 \\ 5 & 4 & 2 \end{array}
\qquad
\begin{array}{ccc} 4 & 4 & 4 \\ 0 & 4 & 4 \\ 5 & 4 & 0 \end{array}
$$

Figure 2.3. Creating a group of four completions in the Grid Method. This method considers surrounding elements in a clockwise direction. If there are no elements to distribute in the first surrounding quadrant, the elements in the following surrounding quadrant will be considered.

In the example, one element is considered at a time. The matrices below show an example of the order of consideration for the lower right corner of N_c. It starts in the lowermost right corner. When this element is equal to 0 or 4, the next sequence of elements is considered.

$$
\begin{pmatrix} 1 & 4 & 2 & 2 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 0 \\ 3 & 2 & 4 & 0 \end{pmatrix}
\qquad
\begin{pmatrix} 1 & 4 & 2 & 2 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 0 \\ 3 & 2 & 4 & 0 \end{pmatrix}
\qquad
\begin{pmatrix} 1 & 4 & 2 & 2 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 0 \\ 3 & 2 & 4 & 0 \end{pmatrix}
$$

Figure 2.4 shows the example strategy step-by-step. When a new sequence is considered, the lowermost element is investigated first. Then the following elements are considered in a clockwise manner.

1	4	2	2
1	2	3	1
1	1	2	0
3	2	4	0

(a) Start in the lower right corner.

1	4	2	2
1	2	3	1
1	1	2	0
3	2	4	0

(b) Step out to next sequence.

1	4	2	2
1	2	3	1
1	1	2	0
3	2	4	0

(c) Move clockwise in the sequence.

1	4	2	2
1	2	1	1
1	1	4	0
3	2	4	0

(d) Move clockwise in the sequence.

1	4	2	2
1	2	1	1
1	1	4	0
3	2	4	0

(e) Step out to new sequence.

1	4	2	2
1	2	1	1
1	0	4	0
2	4	4	0

(f) Move clockwise in the sequence.

1	4	2	2
1	2	1	1
1	0	4	0
2	4	4	0

(g) Move clockwise in the sequence.

1	4	0	2
1	4	1	1
1	0	4	0
2	4	4	0

(h) Move clockwise in the sequence.

Figure 2.4. Creating groups of four in the Grid Method step-by-step. The current element is marked in green. This method starts in the lower right corner and considers surrounding elements in a clockwise direction.

The outcome of this example strategy depends on factors such as:

- Which element that is considered fist.
- Which element to start with in a new sequence.
- Clockwise or counterclockwise sequential order.
- Which surrounding element that is considered first.
- Clockwise or counterclockwise consideration of surrounding elements.

This shows that the example strategy does not necessarily yield the optimal combination of completion groups. Thus, this strategy has to be tested several times with different starting points, different sequential orders etc. This will be inefficient, and it is complicated to implement.

Furthermore, one of the reasons for using the Grid Method was to eliminate groups of confining completion intervals. Figure 2.5 shows that this will happen in the fifth sequence. Although this method excludes many of the combinations that are unnecessary, it does not eliminate them completely. The conclusion is to discard the Grid Method.

2	2	0	2	4	0	0
4	4	4	4	4	4	2
0	2	4	0	4 [2]	0	0
5	4	4	4	4	4	0
5	0	0	0	0	4	0
5	3	0	4	4	4	0

Figure 2.5. The Grid Method does not eliminate groups of confining completions.

2.3 Traveling Salesman Method

This section describes the Traveling Salesman Method (TSM) as an alternative to the methods above. The Traveling Salesman Problem (TSP) is introduced and followed by how it is applied.

2.3.1 Traveling Salesman Problem

The TSP is described by (Vanderbei, 2001, p. 375):

> "Consider a salesman who needs to visit each of n cities, which we shall enumerate as $0, 1, ..., n-1$. His goal is to start from his home city, 0, and make a tour visiting each of the remaining cities once and only once and then returning to his home. We assume that the “distance” between each pair of cities is known (distance does not necessarily have to be distance - it could be travel time or, even better, the cost of travel) and that the salesman wants to make the tour that minimizes the total distance. This problem is called the *traveling salesman problem.*"

Figure 2.6 illustrates a feasible tour in 12 cities. This figure is adapted from Vanderbei (2001).

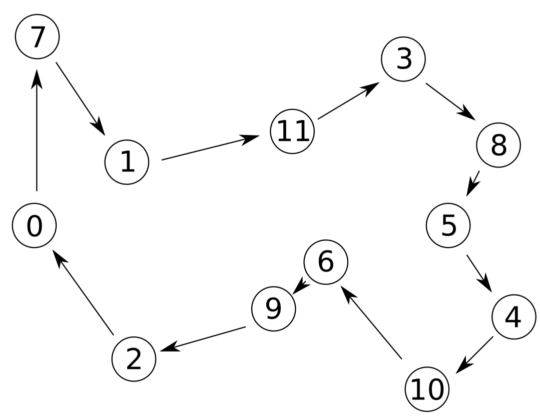


Figure 2.6. Example of a TSP with 12 cities. The salesman starts from his home city, 0, and returns back home after visiting the other cities only once (Vanderbei, 2001, p. 375).

2.3.2 Traveling Salesman Method

The TSM is based on the TSP. Instead of finding the shortest route between cities, the algorithm finds the shortest route between several target points. The target points are the start of the completion intervals. Since the program will handle more than 11 completion intervals, the number of target points changes respectively. MathWorks (2014) has developed a code that solves the TSP. The algorithm uses binary integer programming to solve classic TSP. In their example there are 200 stops (target points), but the parameter, nStops, can easily be changed. This code is used in this Master Thesis, and the number of stops are changed to the number of completion intervals, see Appendix B.2. Using e.g. 12 completion intervals as input, the TSP algorithm calculates and plots the target points and the resulting route, see Figure 2.7.

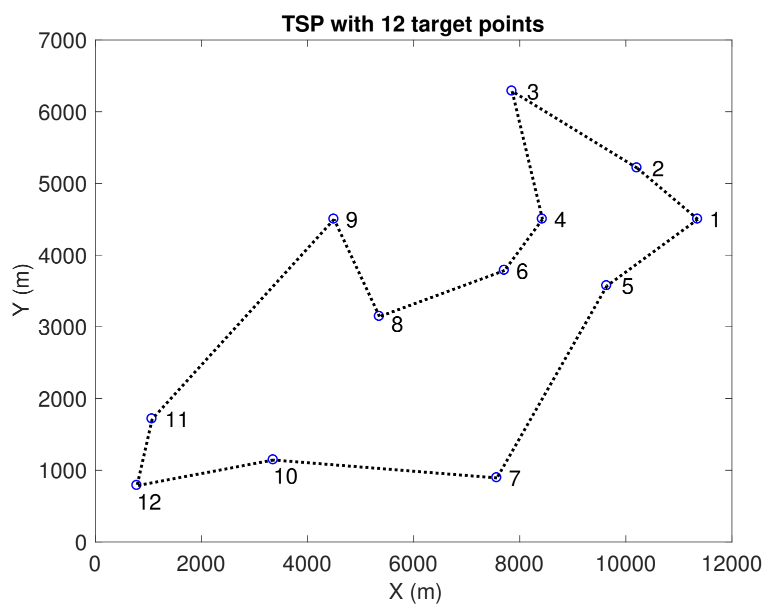


Figure 2.7. Travelling Salesman Problem (TSP) with 12 target points. The first target point is the starting point.

The output is also given as a matrix, called $order$. This matrix lists the target points that are connected in order, as illustrated in the matrix below.

$$order,\ [12 \times 1] = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 6 \\ 8 \\ 9 \\ 11 \\ 12 \\ 10 \\ 7 \\ 5 \end{pmatrix}$$

The matrix above is then used to divide the completions into templates. For a field development with 4-slots templates, the 12 completions are grouped four by four. The resulting groups are illustrated in the following matrices, C_1, C_2, C_3 and C_4. The groups change each time the $order$ matrix is shifted. The $order$ matrix is shifted without rearranging the connected points. The green, olive and yellow color represent template 1, 2 and 3 respectively. The flow chart in Figure 2.8 explains the methodology used in the TSM.

$$C_1, [12\times 1] = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 6 \\ 8 \\ 9 \\ 11 \\ 12 \\ 10 \\ 7 \\ 5 \end{pmatrix} \quad C_2, [12\times 1] = \begin{pmatrix} 5 \\ 1 \\ 2 \\ 3 \\ 4 \\ 6 \\ 8 \\ 9 \\ 11 \\ 12 \\ 10 \\ 7 \end{pmatrix} \quad C_3, [12\times 1] = \begin{pmatrix} 7 \\ 5 \\ 1 \\ 2 \\ 3 \\ 4 \\ 6 \\ 8 \\ 9 \\ 11 \\ 12 \\ 10 \end{pmatrix} \quad C_4, [12\times 1] = \begin{pmatrix} 10 \\ 7 \\ 5 \\ 1 \\ 2 \\ 3 \\ 4 \\ 6 \\ 8 \\ 9 \\ 11 \\ 12 \end{pmatrix}$$

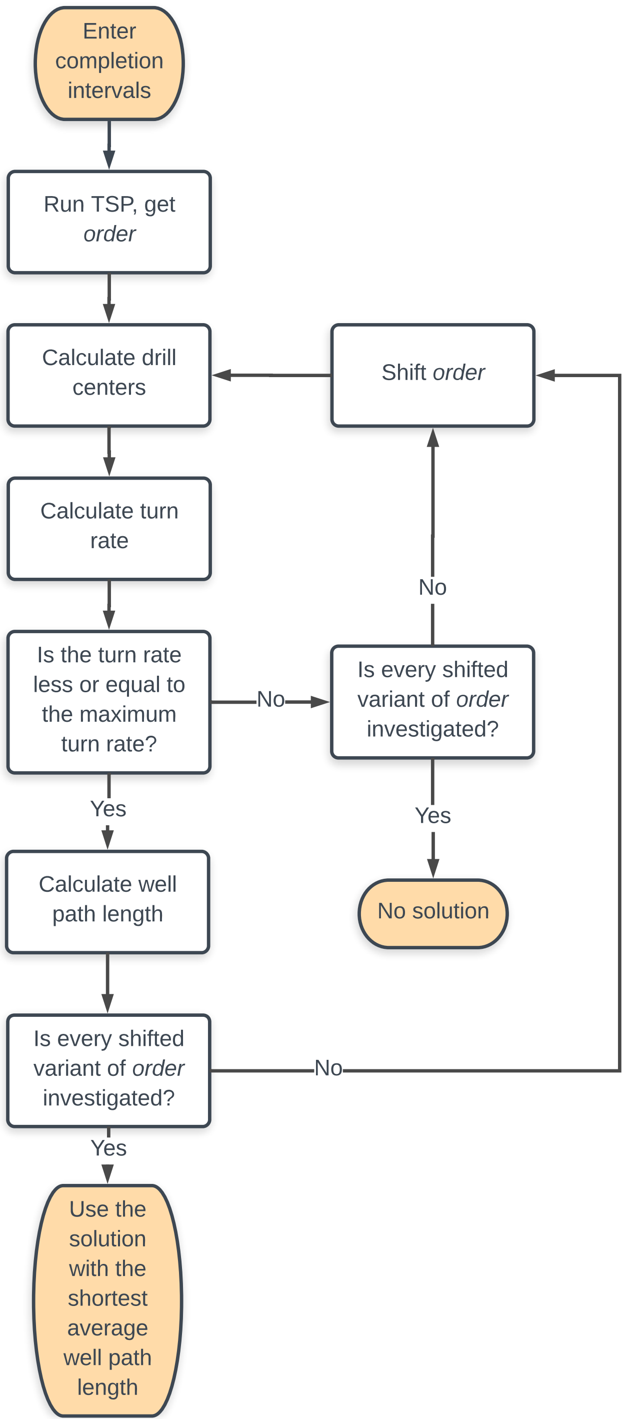


Figure 2.8. Methodology in the Traveling Salesman Method (TSM). The *order* matrix is obtained from the Traveling Salesman Problem (TSP). If the required turn rate(s) (TR) is higher than the maximum turn rate (TR_max), the current variant of order is discarded.

2.3.3 Matrix Method vs. Traveling Salesman Method

The TSM is compared to the Matrix Method to evaluate if the TSM is a competitive method.

The DCs and well paths are calculated as in the program developed by Lillevik and Standal (2017). The complete TSM code is found in Appendix B.2. Table 2.2 compares the Matrix Method and the TSM using 12 completion intervals. The results from the Matrix Method are exactly the same as those presented by Lillevik and Standal (2017), since the same 12 completion intervals and subsea layout are studied.

Table 2.2

The Matrix Method versus the TSM using 12 completion intervals.

Matrix Method			Traveling Salesman Method		
Average Well Path Length (m)	Total Well Path Length (km)	Turn Rate (°/30m)	Average Well Path Length (m)	Total Well Path Length (km)	Turn rate (°/30m)
4303.70	51.64	6	5032.10	57.15	5

The comparison shows that the TSM yields 5.51 km in excessive drilling length. This is a significant increase, and the method can not be considered satisfying. Consequently, the method is not further tested for a higher number of completion intervals. The dividing of completion intervals at different templates is obviously not competitive to the Matrix Method. The conclusion is to discard the TSM.

2.4 Traveling Circus Method

This section describes the final method, called the Traveling Circus Method (TCM). It combines the TSM and an improved Matrix Method. The improvements are explained in detail in Chapter 3. The TSM is used to divide the completion intervals in groups of 12. Table 2.1 shows that the number of combinations resulting from 12 completion intervals is feasible in MATLAB. Thus, the improved Matrix Method is applied to one group of 12 completion intervals at a time.

Since the number of completion intervals is a random number above 11, the chances are high that some satellite wells are required. The number of satellite wells required depends on the remainder after division by 12. The built-in function `rem` finds the remainder after division. E.g, if 40 completion intervals are entered by the user, 4 wells will be satellite wells. When the number of satellite wells is established, the next concern is to determine which of the completion intervals that will be constructed as satellites wells. The function `get_satellites` determines the satellite wells by calculating the distances between every completion interval start in the XY-plane. The completion intervals with the longest distance to the neighbouring completion intervals are selected to be satellite wells, see Appendix B.6.

When the satellite wells are identified, the remaining completion intervals are divided into groups of 12. First, the TSP algorithm is run and the $order$ matrix is obtained, see Chapter 2.3. The dimensions of $order$ are $[12n \times 1]$, where $n \in \mathbb{N}$, since the satellite wells are eliminated. The following matrix shows an example of the arrangement of 24 completions. The first 12 rows, colored olive, are in the first group, the next 12 rows, colored yellow, are in the second group.

$$order,\ [24 \times 1] = \begin{pmatrix} 1 \\ 11 \\ 12 \\ 21 \\ 5 \\ 9 \\ 13 \\ 24 \\ 19 \\ 4 \\ 22 \\ 3 \\ 2 \\ 6 \\ 7 \\ 8 \\ 10 \\ 14 \\ 15 \\ 16 \\ 17 \\ 18 \\ 20 \\ 23 \end{pmatrix}$$

When the first arrangement is established, the improved Matrix Method is applied to one group of 12 completion intervals at a time. Every combination of the 12 completion intervals at different templates are set up and studied. The number of combinations depends on the number of slots in the templates of interest, as explained in Chapter 2.1. The combination that yields the shortest average WPL and satisfying TRs is saved. In the example above, with 24 completion intervals, two solutions are saved, since there are two groups of 12 completion intervals in the $order$ matrix. Finally, the average WPL of all 24 completion intervals in that arrangement is calculated.

Regardless of the number of completion intervals, the *order* matrix is shifted 11 times. Each time, new groups of 12 are set up, the best combination in each group is saved, and the average well path length of that *order* arrangement is calculated.

Figure 2.9 shows an illustrative example of three variants of *order*, and the resulting average WPLs. If none of the nine other variants of *order* yield an average WPL below 3000 meters, the second version of *order* is the final solution in this 4-slots field architecture.

order arrangment		
TSP (1)	2	3
1	23	20
11	1	23
12	11	1
21	12	11
5	21	12
9	5	21
13	9	5
24	13	9
19	24	13
4	19	24
22	4	19
3	22	4
2	3	22
6	2	3
7	6	2
8	7	6
10	8	7
14	10	8
15	14	10
16	15	14
17	16	15
18	17	16
20	18	17
23	20	18

Group 1
Group 2
Template 1
Template 2
Template 3
Template 4
Template 5
Template 6

	Best combination calculated in Matrix Method		
	TSP (1)	2	3
	1	23	20
	11	1	23
	12	11	1
	21	12	11
	5	21	12
	9	5	21
	13	9	5
	24	13	9
	19	24	13
	4	19	24
	22	4	19
	3	22	4
	2	3	22
	6	2	3
	7	6	2
	8	7	6
	10	8	7
	14	10	8
	15	14	10
	16	15	14
	17	16	15
	18	17	16
	20	18	17
	23	20	18
Average WPL (m)	4000	3000	5000

Figure 2.9. 4-slots template architecture with three variants of *order*. The resulting optimal combination of each variant and the associated average well path length (WPL) are listed.

Figure 2.10 shows the same three variants of $order$, in a 6-slots template architecture. If none of the nine other variants of $order$ yield an average well path length below 2700 meters, the second version of $order$ is the final solution. The flowchart in Figure 2.11 summarizes the methodology in the TCM.

order arrangment		
TSP (1)	2	3
1	23	20
11	1	23
12	11	1
21	12	11
5	21	12
9	5	21
13	9	5
24	13	9
19	24	13
4	19	24
22	4	19
3	22	4
2	3	22
6	2	3
7	6	2
8	7	6
10	8	7
14	10	8
15	14	10
16	15	14
17	16	15
18	17	16
20	18	17
23	20	18

Group 1
Group 2
Template 1
Template 2
Template 3
Template 4
Template 5
Template 6
Template 7
Template 8
Template 9
Template 10
Template 11
Template 12

	Best combination calculated in Matrix Method		
	TSP (1)	2	3
	1	23	20
	11	1	23
	12	11	1
	21	12	11
	5	21	12
	9	5	21
	13	9	5
	24	13	9
	19	24	13
	4	19	24
	22	4	19
	3	22	4
	2	3	22
	6	2	3
	7	6	2
	8	7	6
	10	8	7
	14	10	8
	15	14	10
	16	15	14
	17	16	15
	18	17	16
	20	18	17
	23	20	18
Average WPL (m)	3000	2700	3900

Figure 2.10. 2-slots templates architecture with three variants of *order*. The resulting optimal combination of each variant and the associated average well path length (WPL) are listed.

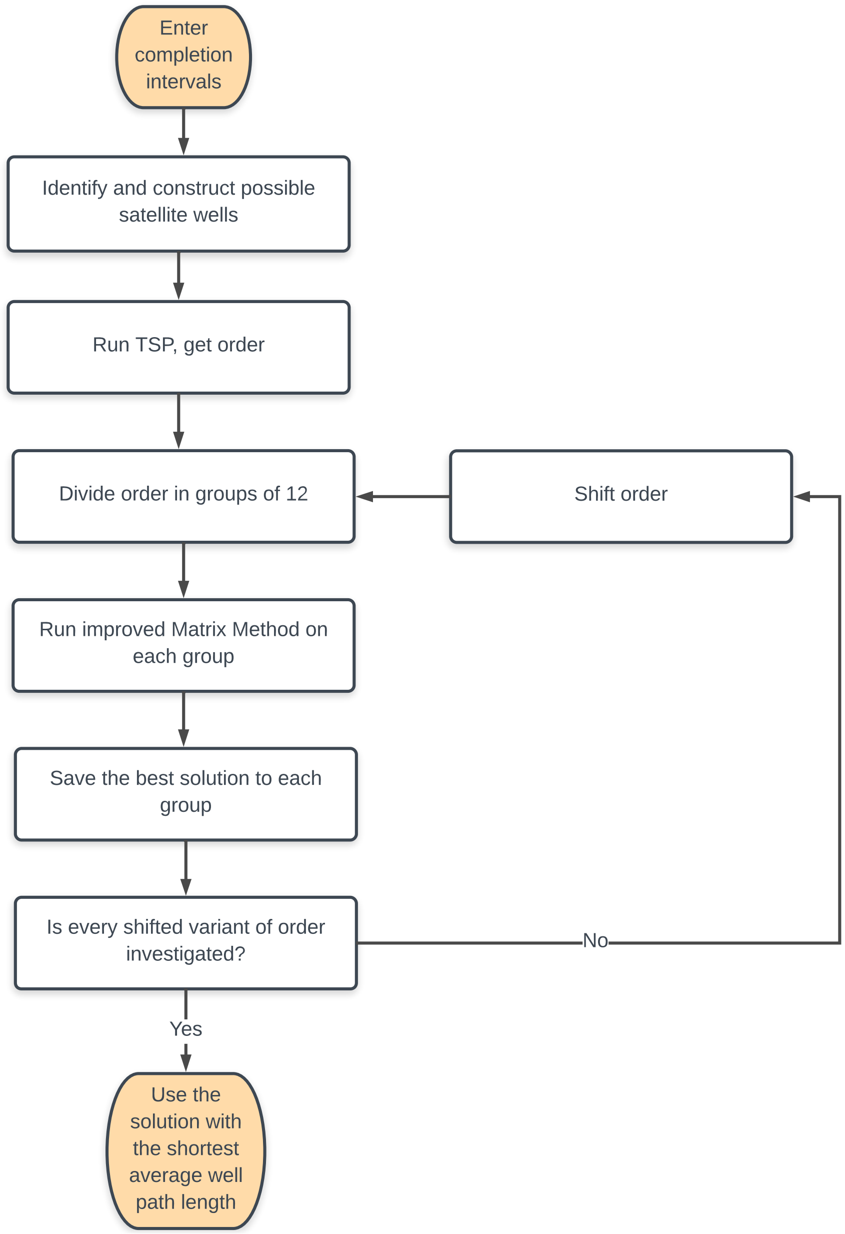


Figure 2.11. Combination methodology in the Traveling Circus Method (TCM). The calculations are performed on 12 and 12 completion intervals at a time. This process is repeated for every variant of $order$.

2.4.1 Matrix Method vs. Traveling Circus Method

The TCM is compared to the Matrix Method to show why TCM is a better method to use.

Table 2.3 compares the Matrix Method and the TCM. Two different sets of 12 completion intervals and one set of 24 completion intervals are tested. Lillevik and Standal (2017) used the same TR for all wells. In the TCM, the optimal TR of each well is calculated. The maximum TR requirement used is 6 °/30m. This is explained in detail in Chapter 3.3. The Matrix Method therefore refers to the method developed by Lillevik and Standal (2017), and the Travelling Circus Method uses an improved Matrix Method.

Table 2.3

Matrix Method versus the TCM using two different sets of 12 completion intervals and one set of 24 completion intervals.

	Matrix Method		Traveling Circus Method	
Completion Intervals (-)	Average Well Path Length (m)	Turn Rate (°/30m)	Average Well Path Length (m)	Average Turn Rate (°/30m)
12	4078.60	10.00	4454.40	4.25
12*	4303.70	6.00	4421.40	3.50
24	N/A	N/A	4288.80	4.25

*A new (not the original) set of 12 completion intervals.

The first set of 12 completions yields an unsatisfying turn rate in the Matrix Method. Because the turn rate calculations in the TCM are improved and contain restrictions, the average well path length consequently increases. The second set of 12 completions yields satisfying results in both methods, and gives an improved basis of comparison. Using the average well path length values, the excessive drilling length is 1.4124 km. The trend is that the turn rate decreases and the average well path length increases in the Traveling Circus Method (TCM). The increase in well path length is noteworthy, but equally important is the turn rate decrease. In contrast to the Matrix Method, the Traveling Circus Method (TCM) handles random numbers of completion intervals, as indicated in Table 2.3. The results from this evaluation show that the Traveling Circus Method (TCM) is the preferred method to use.

Chapter 3

Improvements

This chapter explains measures that are done to improve the calculations in the program that was developed by Lillevik and Standal (2017). First, the codes are improved to construct wells with non-horizontal completion intervals, as many wells today have an inclined completion interval. Further, the user is given more freedom as the construction of the wells now allows for different build-up rates (BUR).

The most important improvement is a restriction in allowed turn rate (TR). This is necessary as the wells that are constructed, shall also be drillable. Too high TRs cause excessive torque and drag.

The last improvement is introducing 2-slots templates as these are becoming more popular in the industry today.

3.1 Non-Horizontal Completions

The program made by Lillevik and Standal (2017) considers horizontal completion intervals. There is a need to perform calculations that are applicable for non-horizontal completion intervals as well. Upwards completion intervals, where the build-up angle (BUA) is above 90° have not been considered.

3.1.1 Satellite Wells

Lillevik and Standal (2017) presented the formulas needed to construct non-horizontal satellite wells. The formulas were presented in case there would be a need to construct non-horizontal satellite wells in the future (Lillevik and Standal, 2017, Chapter 2.2). Thus, these formulas are used in the MATLAB code `get_sat_WPL`, see Appendix B.6.

3.1.2 Wells from Common Drill Center

Two types of non-horizontal wells are considered, J-type and S-type. A J-well is characterized by two build sections, while a S-well is characterized by one build and one drop section. Note that the wells are constructed in the RZ-plane. Lillevik and Standal (2017) defines R as the measured horizontal displacement and Z as the true vertical depth (TVD) of the well. Since the construction of the wells is based on the calculations presented by Lillevik and Standal (2017), the J-wells were first constructed with one BUR, and the S-wells were constructed with a drop-off rate (DOR) equal to the BUR.

J-wells

Lillevik and Standal (2017) constructed the wells by placing the circle center of the second build circle straight above the completion start coordinate, perpendicular to the completion interval. In the case of non-horizontal completions, the equations for the center coordinates of the second build circle, R_{cc2} and Z_{cc2}, are adjusted.

The completion interval starts at the coordinates dR_{tot} and dZ_{tot}. The inclination of the completion interval is equal to the total BUA. Both build sections have the same BUR, thus they have the same radius of curvature (ROC). This leads to Equation 3.1 and 3.2. They are valid for both horizontal completions (BUA = 90°) and non-horizontal completions (BUA < 90°).

$$R_{cc2} = dR_{tot} + \text{ROC} \times \sin\left(90° - \text{BUA}\right) \tag{3.1}$$

$$Z_{cc2} = dZ_{tot} - \text{ROC} \times \cos\left(90° - \text{BUA}\right) \tag{3.2}$$

See Lillevik and Standal (2017) for the derivation of the parameters in Equation 3.1 and 3.2. Figure 3.1 shows a typical J-well indicating the parameters above.

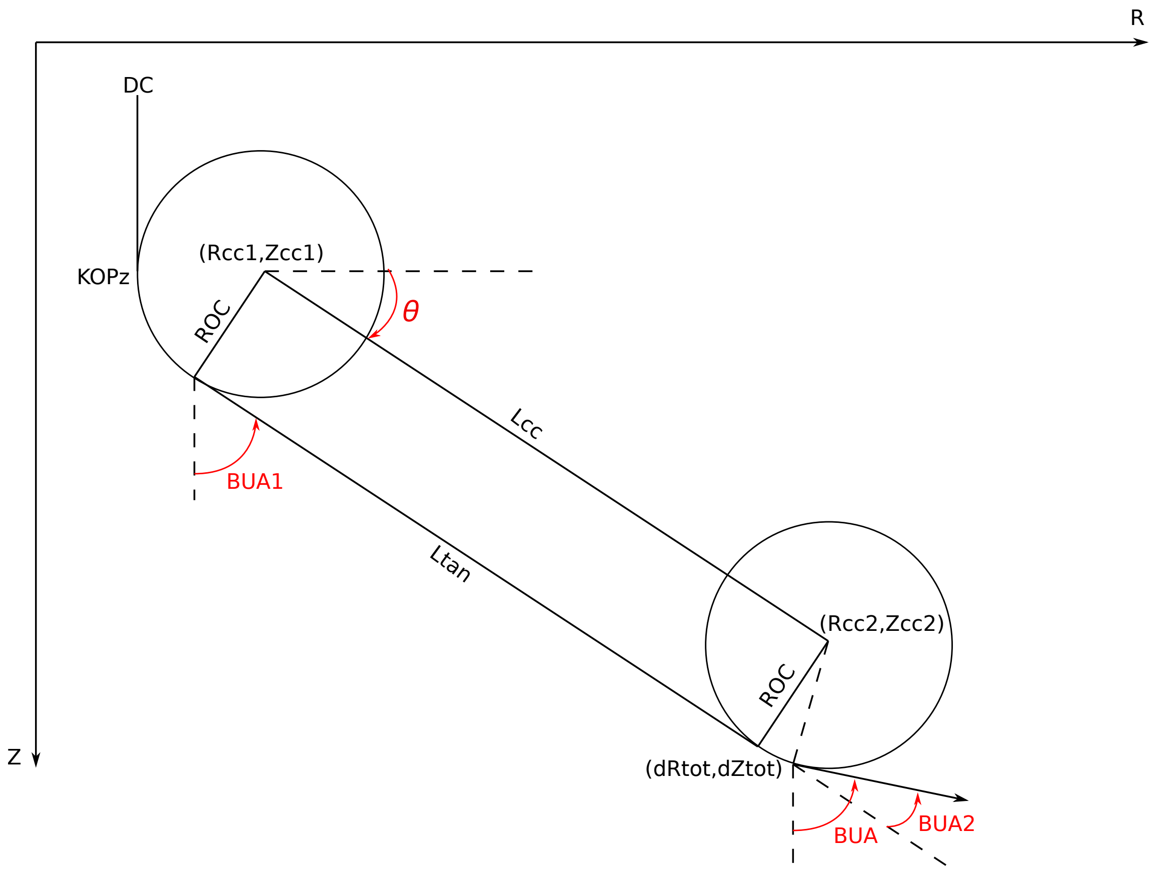


Figure 3.1. Construction of a J-well in the RZ-plane. Both build sections have one common BUR, thus the build circles have the same radius of curvature (ROC).

A new approach for constructing the wells is made. This makes the implementation of the construction of all wells easier. Some of the parameters calculated by Lillevik and Standal (2017) are now calculated using the equations below. The equations are valid for J-type wells with horizontal completions and J-wells with non-horizontal completions.

The length of the tangent, L_{tan}, is calculated by Equation 3.3. The length of the tangent is equal to the length between the two build circle centers, L_{cc}, because they form a rectangle together. They are parallel to each other, and the tangent intersects both circles perpendicular to the radii. The circle center coordinates are R_{cc1} and Z_{cc1} for the first circle of build.

$$L_{tan} = \sqrt{\left(Z_{cc2} - Z_{cc1}\right)^2 + \left(R_{cc2} - R_{cc1}\right)^2} \tag{3.3}$$

The inclination of the tangent is equal to the first build-up angle, BUA1. BUA1 is calculated using the dip angle, θ, of the line between the two circle centers. The dip angle is calculated using the trigonometric relations in Equation 3.4.

$$\theta = \arctan\left(\frac{Z_{cc2} - Z_{cc1}}{R_{cc2} - R_{cc1}}\right) \tag{3.4}$$

BUA1 is then calculated by Equation 3.5. The difference between BUA and BUA1 is equal to the second build-up angle, BUA2, expressed in Equation 3.6.

$$\mathrm{BUA1} = 90° - \theta \tag{3.5}$$

$$\mathrm{BUA2} = \mathrm{BUA} - \mathrm{BUA1} \tag{3.6}$$

Lillevik and Standal (2017) used the total arc length of both build sections to construct the wells. The implementation of S-type wells requires two separate arc lengths: one arc length for the build section, arc, and one arc length for the drop section, arc2. To make the vector calculations in MATLAB less complicated, the total arc length of J-type wells is also split in two. The expressions are shown in Equation 3.7 and 3.8. The BUR is an input parameter, and as mentioned, both circles have the same BUR.

$$\mathrm{arc} = \frac{\mathrm{BUA1}}{\mathrm{BUR}} \tag{3.7}$$

$$\mathrm{arc2} = \frac{\mathrm{BUA2}}{\mathrm{BUR}} \tag{3.8}$$

The well path length, WPL, of each well is calculated using Equation 3.9. The depth of the first kick off point, KOPz, is an input parameter. Lillevik and Standal (2017) derived the equation for the length of the completion interval, L_c.

$$\mathrm{WPL} = \mathrm{KOPz} + \mathrm{arc} + L_{tan} + \mathrm{arc2} + L_c \tag{3.9}$$

S-wells

Some of the non-horizontal completion intervals form S-shaped wells. The criteria for S-type wells is that BUA < 90°, and BUA1 > BUA. Since a S-well has one build section and one drop section, the well follows two different arcs, see Figure 3.2.

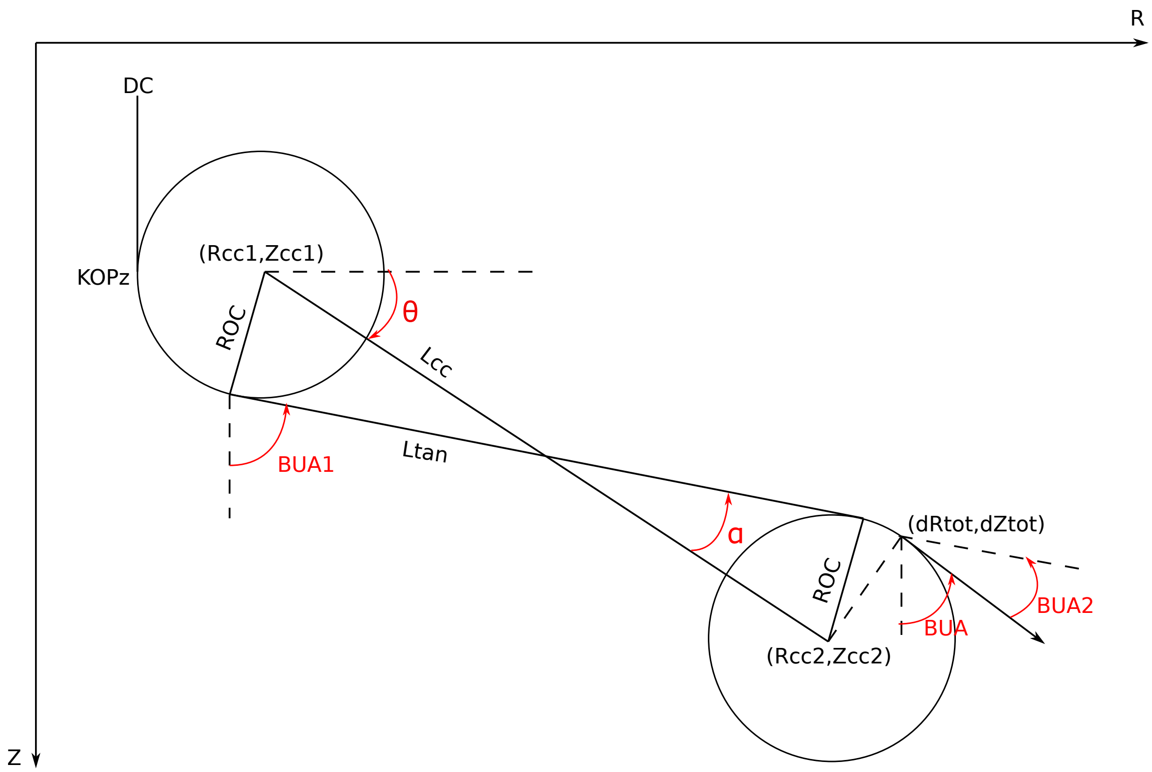


Figure 3.2. Construction of a S-well in the RZ-plane. The BUR is equal to the DOR, thus the build and drop circles have the same radius of curvature (ROC).

The circle of drop is located below the completion interval. The center coordinates of the drop circle, R_{cc2} and Z_{cc2}, are calculated using Equation 3.10 and 3.11. Since the DOR is equal to the BUR, the build and drop circles have the same ROC.

$$R_{cc2} = dR_{tot} - \text{ROC} \times \sin\left(90° - \text{BUA}\right) \tag{3.10}$$

$$Z_{cc2} = dZ_{tot} + \text{ROC} \times \cos\left(90° - \text{BUA}\right) \tag{3.11}$$

To find the length of the tangent, the length between the build circle center and the drop circle center, L_{cc}, is calculated. This is done using the Pythagorean theorem, see Equation 3.12. The coordinates of the build circle center are R_{cc1} and Z_{cc1}.

$$L_{cc} = \sqrt{\left(Z_{cc2} - Z_{cc1}\right)^2 + \left(R_{cc2} - R_{cc1}\right)^2} \tag{3.12}$$

The line between the circle centers intersects the tangent halfway, since the tangent intersects both circles perpendicular to the radii. The angle between the tangent and the crossing line is calculated using Equation 3.13.

$$\alpha = \arcsin\left(\frac{\mathrm{ROC}}{0.5 \times L_{cc}}\right) \tag{3.13}$$

The length of the tangent, L_{tan}, is then calculated using trigonometric relations, see Equation 3.14.

$$L_{tan} = 2 \times \frac{\mathrm{ROC}}{\tan(\alpha)} \tag{3.14}$$

BUA1 and BUA2 are then calculated using Equation 3.15 and 3.16. The drop angle is named BUA2 to keep the calculations in MATLAB systematic. The dip angle, θ, is calculated from Equation 3.4.

$$\mathrm{BUA1} = 90° - \theta + \alpha \tag{3.15}$$

$$\mathrm{BUA2} = \mathrm{BUA1} - \mathrm{BUA} \tag{3.16}$$

The arc length of the build circle, arc, and the arc length of the drop circle, arc2, are calculated using Equation 3.7 and 3.8. The WPL is calculated using Equation 3.9.

3.2 Build and Drop Rates

Two new input parameters are introduced. These are the second build-up rate, BUR2, and the DOR. The parameters are used for J- and S-wells, respectively. The calculations are made general to make them valid when BUR ≥ BUR2 ∩ DOR and BUR < BUR2 ∩ DOR.

J-wells

With different BURs, the circles of build have different radii. In this case, the tangent and the line between the circle centers do not form a rectangle. Thus, the length of the tangent is therefore not equal to the distance between the two circle centers, see Figure 3.3.

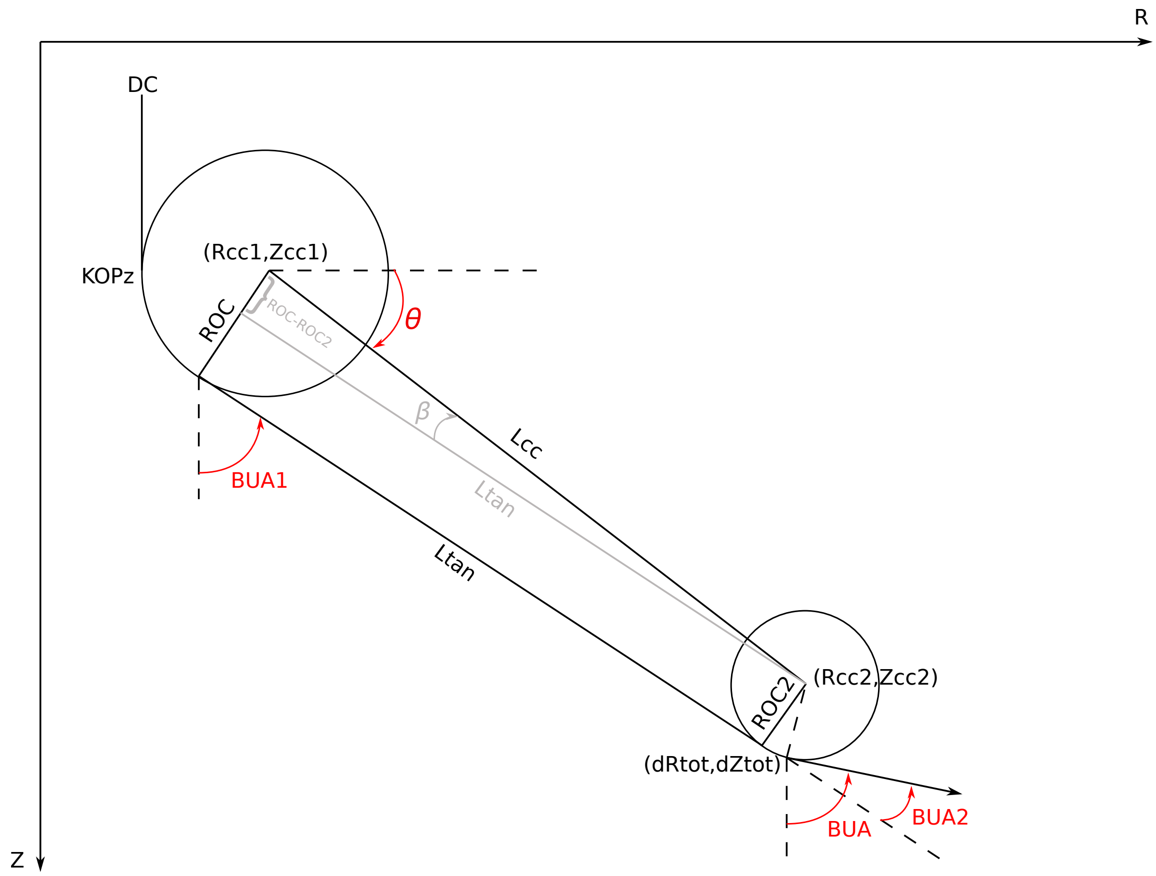


Figure 3.3. Construction of a J-well in the RZ-plane. The build sections have different build-up rates, thus the build circles have different radius of curvature, ROC and ROC2.

The radii of curvature, ROC and ROC2, are calculated using BUR and BUR2, respectively. The relations are shown in Equation 3.17 and 3.18.

$$\mathrm{ROC} = \frac{360°}{2\pi \times \mathrm{BUR}} \tag{3.17}$$

$$\mathrm{ROC2} = \frac{360°}{2\pi \times \mathrm{BUR2}} \tag{3.18}$$

The distance between the circle centers, L_{cc}, is calculated using Equation 3.12. The length of the tangent section, L_{tan}, is calculated using the Pythagorean theorem in Equation 3.19. This equation verifies that the length of the tangent is equal to L_{cc} when ROC = ROC2.

$$L_{tan} = \sqrt{\left(L_{cc}^2 - \left(ROC - ROC2\right)^2\right)} \tag{3.19}$$

BUA1 is calculated using the angle between the tangent and the line between the circle centers, β. This angle is calculated using the trigonometric relation in equation 3.20. The equation verifies that β is equal to 0 when ROC = ROC2. BUA1 is calculated using Equation 3.21.

$$\beta = \arctan\left(\frac{\mathrm{ROC} - \mathrm{ROC2}}{L_{tan}}\right) \tag{3.20}$$

$$\mathrm{BUA1} = 90 - \theta + \beta \tag{3.21}$$

The arc length of the first circle of build, arc, and the arc length of the second circle of build, arc2, are calculated using Equation 3.22 and 3.23.

$$\mathrm{arc} = \frac{\mathrm{BUA1}}{\mathrm{BUR}} \tag{3.22}$$

$$\mathrm{arc2} = \frac{\mathrm{BUA2}}{\mathrm{BUR2}} \tag{3.23}$$

The remaining equations to calculate the WPL are the same as for the J-wells in Chapter 3.1.2.

S-wells

Since the DOR is introduced as a new input parameter, the ROC in the drop circle is calculated by Equation 3.24.

$$\mathrm{ROC2} = \frac{360°}{2\pi \times \mathrm{DOR}} \tag{3.24}$$

When the DOR does not equal the BUR, the two associated circles will have different radii. This changes the intersection point between the tangent and the line between the circle centers, see Figure 3.4.

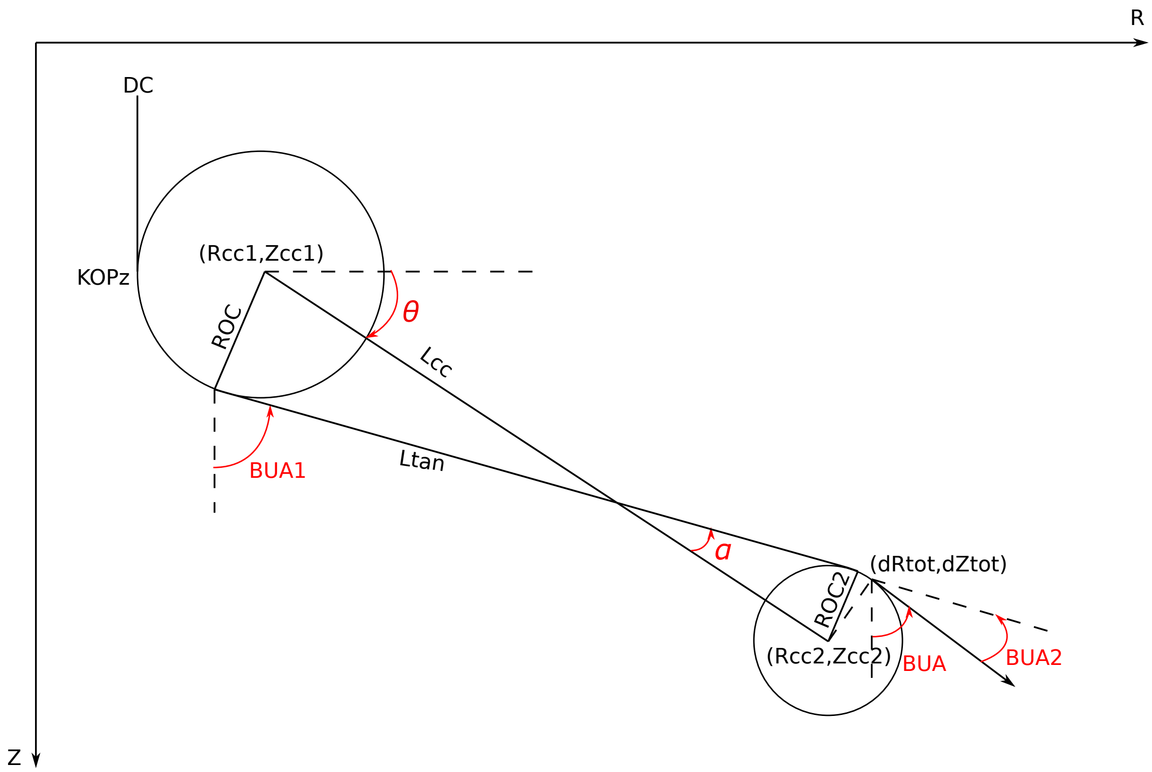


Figure 3.4. Construction of a S-well in the RZ-plane. The BUR is not equal to the DOR, thus the build and drop circles have different radii of curvature, ROC and ROC2.

The point of intersection can be described as a ratio, p, between ROC2 and the sum of the two radii, see Equation 3.25.

$$p = \frac{\mathrm{ROC2}}{\mathrm{ROC} + \mathrm{ROC2}} \tag{3.25}$$

The angle between the tangent and the intersecting line, α, is calculated by multiplying L_{cc} by the factor p, see Equation 3.26.

$$\alpha = \arcsin\left(\frac{\mathrm{ROC}}{p \times L_{cc}}\right) \tag{3.26}$$

The length of the tangent, L_{tan}, is then calculated by dividing the expression in Equation 3.14 by the factor p, see Equation 3.27.

$$L_{tan} = \frac{1}{p} \times \frac{\mathrm{ROC}}{\tan(\alpha)} \tag{3.27}$$

The remaining equations to calculate the WPL are the same as for the S-wells in Chapter 3.1.2.

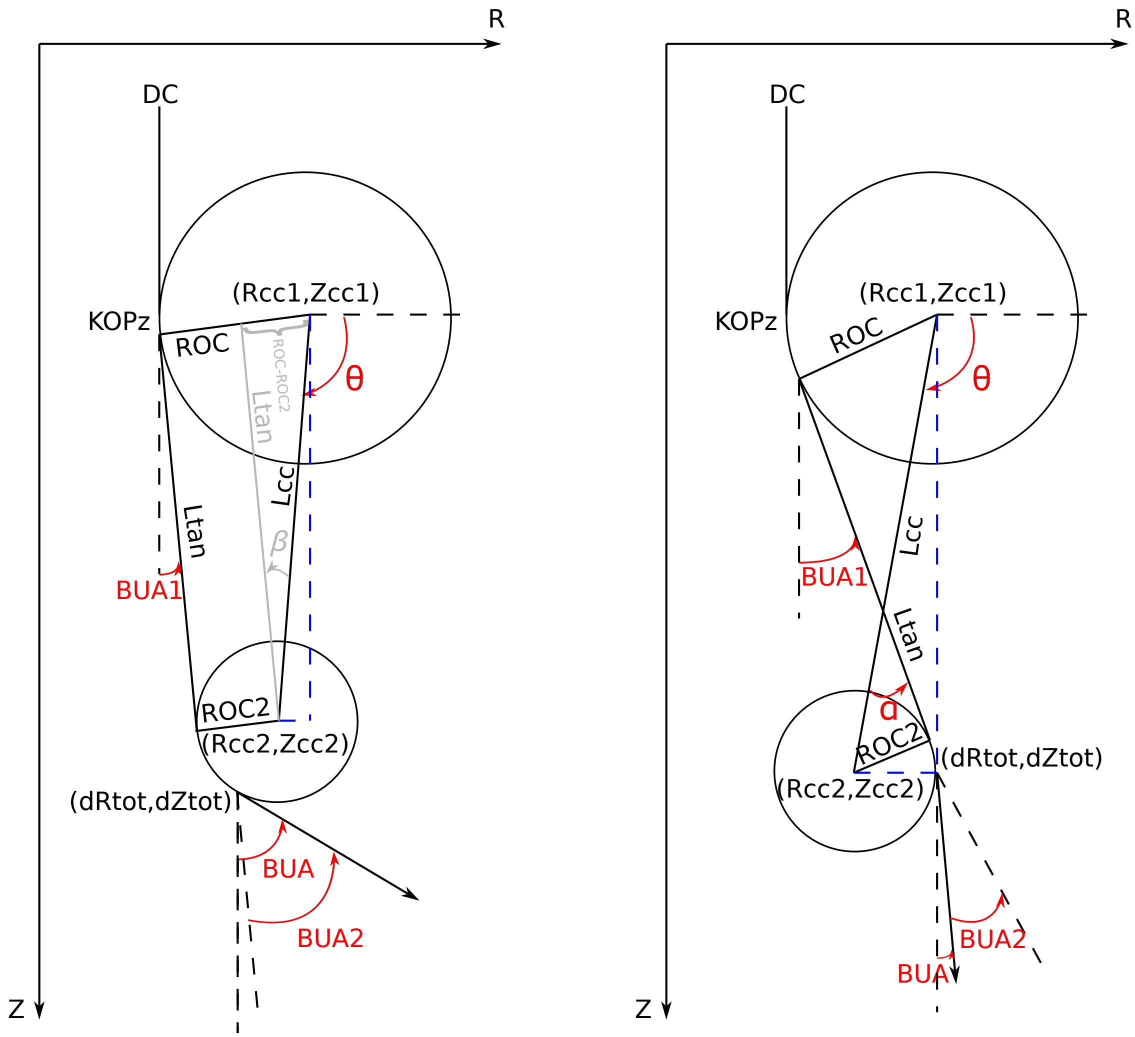
Short J- and S-wells

For short wells, the R-coordinate of the second circle center, R_{cc2}, is lower than the R-coordinate of the first circle center, R_{cc1}. Figure 3.5 shows an example of a short J-well and a short S-well.

The dip angle, θ, of the line between the two circle centers become larger than 90°, thus the trigonometric relations in Equation 3.4 changes. For both types of wells it is seen from trigonometric relations in Figure 3.5 that the new expression for the dip angle is equal to the expression in Equation 3.28. The expression is also applicable when ROC=ROC2.

$$\theta = 90° + \arctan\left(\frac{R_{cc1} - R_{cc2}}{Z_{cc2} - Z_{cc1}}\right) \qquad \text{for} \quad R_{cc1} > R_{cc2} \tag{3.28}$$

The remaining equations required to calculate the WPL are the same as for the J- and S-wells in Chapter 3.1.2 and Chapter 3.2.



(a) Construction of a short J-well.

(b) Construction of a short S-well.

Figure 3.5. Construction of short J- and S-wells in the RZ-plane. The BURs of the J-well are not equal, and the DOR of the S-well is not equal to the BUR, thus the build and drop circles have different radii of curvature, ROC and ROC2. The blue lines are drawn as help lines. They are used to find the dip angle, θ.

3.3 Turn Rate

The TR is the change in azimuth angle per 30 meters drilled. The azimuth angle is a measure of the angle between a reference axis (North) and the well path in the horizontal plane. The TR is a critical parameter because of the resulting drag and torque forces. If the drag and torque forces are too high, the drilling or completion operation may cease. In order to construct wells with low technical risk, the TR is minimized (Lillevik and Standal, 2017, Chapter 3.3).

Lillevik and Standal (2017) constructed each well with the same TR. The well that required the highest TR, set the standard for all the other wells. This is a poor solution when the objective is to the minimize the TRs.

3.3.1 Turn Rate in the Traveling Circus Method

The TR to each individual well path is minimized in the Traveling Circus Method (TCM) . First, the user enters a preferred TR, and the resulting radius of turn (ROT) is calculated by Equation 3.29.

$$\mathrm{ROT} = \frac{360°}{2\pi \times \mathrm{TR}} \tag{3.29}$$

The well path follows a circle of turn with radius equal to the radius of turn (ROT) in the XY-plane, see Figure 3.6. This figure illustrates the projection of the well onto the XY-plane. The drill center (DC), the completion interval and the ROT are indicated.

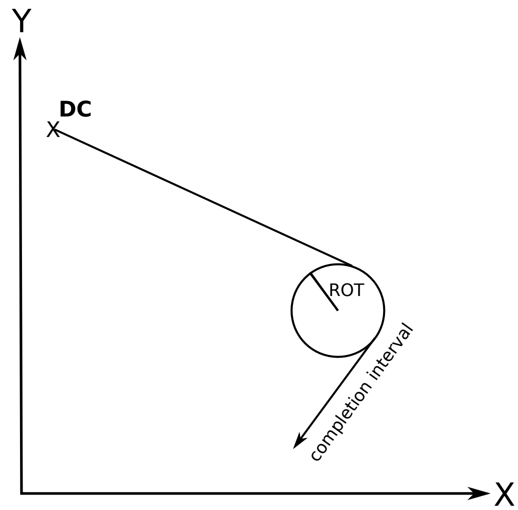


Figure 3.6. The projection of a well onto the XY-plane. The well turns around a circle that has a radius equal to the radius of turn (ROT).

As explained in Chapter 2.4, the $order$ matrix is divided in groups of 12 and shifted 11 times. When calculating the optimal combination of completion intervals for each variant of $order$, the required TR of each completion is calculated. This means that although the TR is an input parameter, it will be adjusted when required.

An adjustment is required if the distance in the XY-plane between the DC and the completion start is too short for the well to follow the arc. This is done by use of the criteria in Equation 3.30. X_{DC} and Y_{DC} are the DC coordinates, and X_{cs} and Y_{cs} are the completion start coordinates. When this criteria is fulfilled, the TR is increased by 1°/30m, and the criteria is checked again until it is not fulfilled (Lillevik and Standal, 2017, Chapter 2.2).

$$\sqrt{(X_{DC} - X_{cs})^2 + (Y_{DC} - Y_{cs})^2} < 2 \times \text{ROT} \tag{3.30}$$

To minimize the TR, a new input parameter is introduced. This is the maximum allowed turn rate (TR_max). If one of the wells in the combination with the shortest average WPL causes a TR above the maximum, this combination is discarded and the combination that yields the second shortest WPL is tested. This procedure is repeated until none of the TRs exceed TR_max. The flowchart in Figure 3.7 explains the methodology. This procedure refers to the 4th, 5th, and 6th box in Figure 2.11.

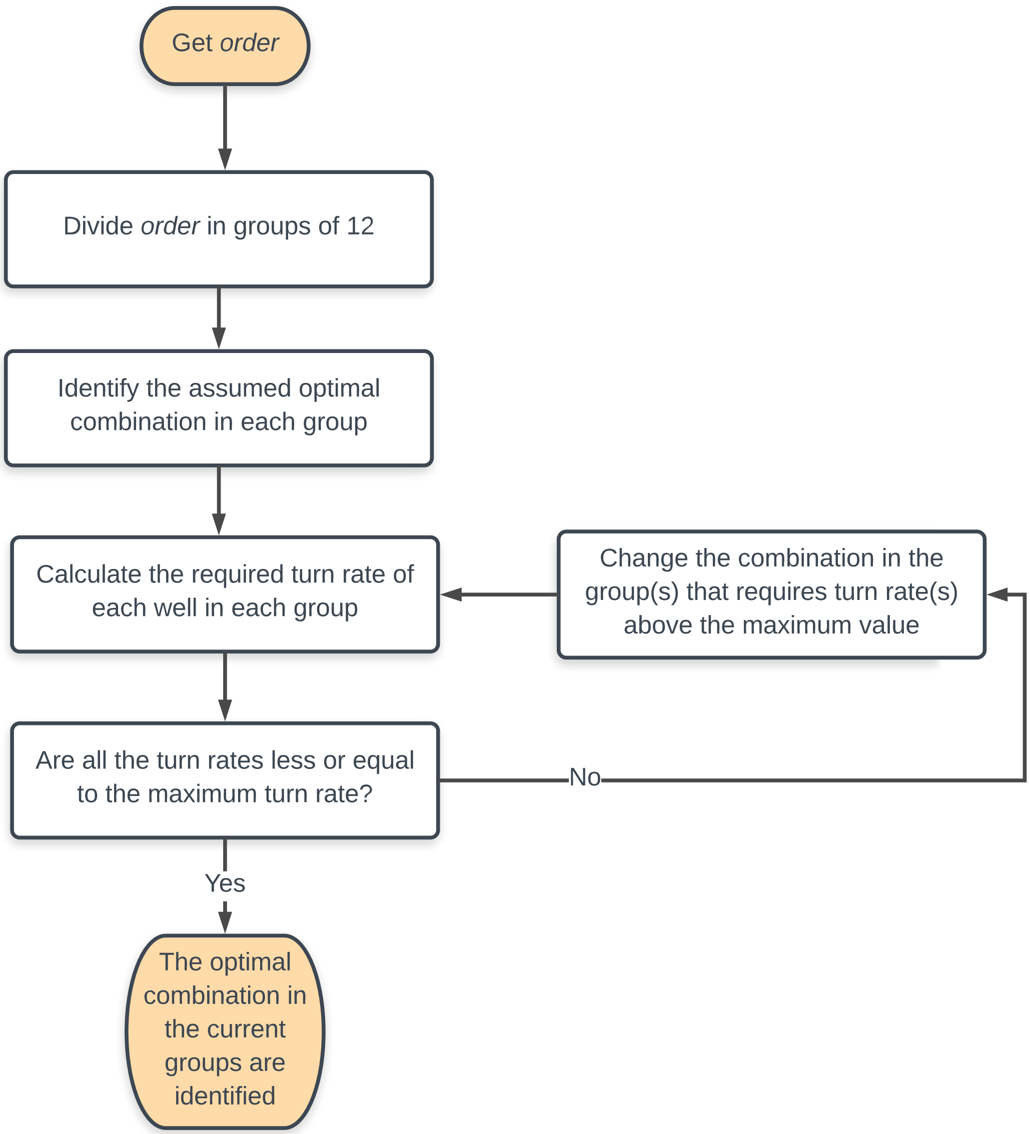


Figure 3.7. Methodology in the turn rate (TR) calculation of template layouts. If the turn rate is increased above the maximum turn rate (TR_max), a new combination of completion intervals is tested.

3.3.2 One Drill Center

A DC is where the drilling operation commences. In the case of one DC, all completion intervals are reached from the same starting point. The DC coordinates of n completion intervals is calculated by taking the average of the completion start coordinates, X_{cs} and Y_{cs} (Lillevik and Standal, 2017, Chapter 2.1).

$$X_{DC} = \frac{1}{n}\sum_{i=1}^{n} X_{cs} \tag{3.31}$$

$$Y_{DC} = \frac{1}{n}\sum_{i=1}^{n} Y_{cs} \tag{3.32}$$

As mentioned, preferred TR and TR_max are input parameters. The TR to each well is adjusted as in the case of several DCs, by increasing the preferred TR by 1°/30m if the criteria in Equation 3.30 is fulfilled. When the TR increases above TR_max, the methodology change. Since all completions are drilled from the same DC, there are no other alternative combinations to choose from.

When one of the TRs exceeds TR_max, the current TR is set to be TR_max and the associated radius of turn (ROT_max) is calculated. Then, the direction (dir) and length (dist) of the straight line between the DC and the current completion interval start are calculated, see Equation 3.33 and 3.34. The calculation of dir is performed using the built-in function `atan2d`. Lillevik and Standal (2017) derived the expression for `atan2d`.

$$\text{dir} = (X_{DC} - X_{cs}, Y_{DC} - Y_{cs}) \tag{3.33}$$

$$\text{dist} = \sqrt{(X_{DC} - X_{cs})^2 + (Y_{DC} - Y_{cs})^2} \tag{3.34}$$

The DC is moved as in Equation 3.35 and 3.36. Figure 3.8 illustrates the DC before (black) and after (red) relocation. The parameters used in the calculation are indicated. After relocation, the TR calculations returns to the beginning.

$$X_{DC} = X_{DC} + (2 \times \mathrm{ROT\_max} - \mathrm{dist}) \times \sin(\mathrm{dir}) \tag{3.35}$$

$$Y_{DC} = Y_{DC} + (2 \times \mathrm{ROT\_max} - \mathrm{dist}) \times \cos(\mathrm{dir}) \tag{3.36}$$

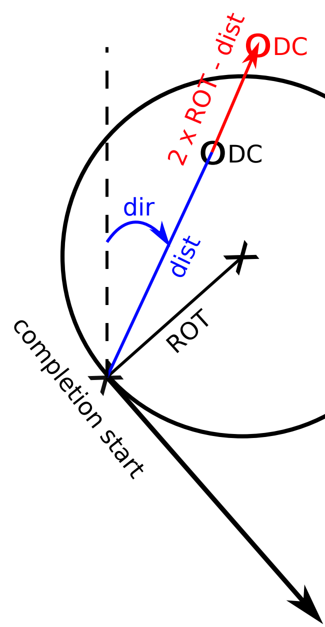


Figure 3.8. Relocation of the drill center (DC). If the DC is too close to the completion start, the drill center is relocated a distance (2 × ROT) away from the completion.

This method minimizes the TR in each well since the TR requirement is studied for each completion interval, one by one. The movement of the DC is always equal to $2 \cdot \mathrm{ROT\_max} - \mathrm{dist}$, to guarantee that the new DC location is placed a distance of $2 \times \mathrm{ROT}$ away from the current completion start. Consequently, when all calculations are repeated for the new DC, the completion interval that required movement the last time will meet all restrictions. If none of the completion intervals meet the movement criteria after relocation, the final DC location and TRs are obtained. The flowchart in Figure 3.9 summarizes the methodology.

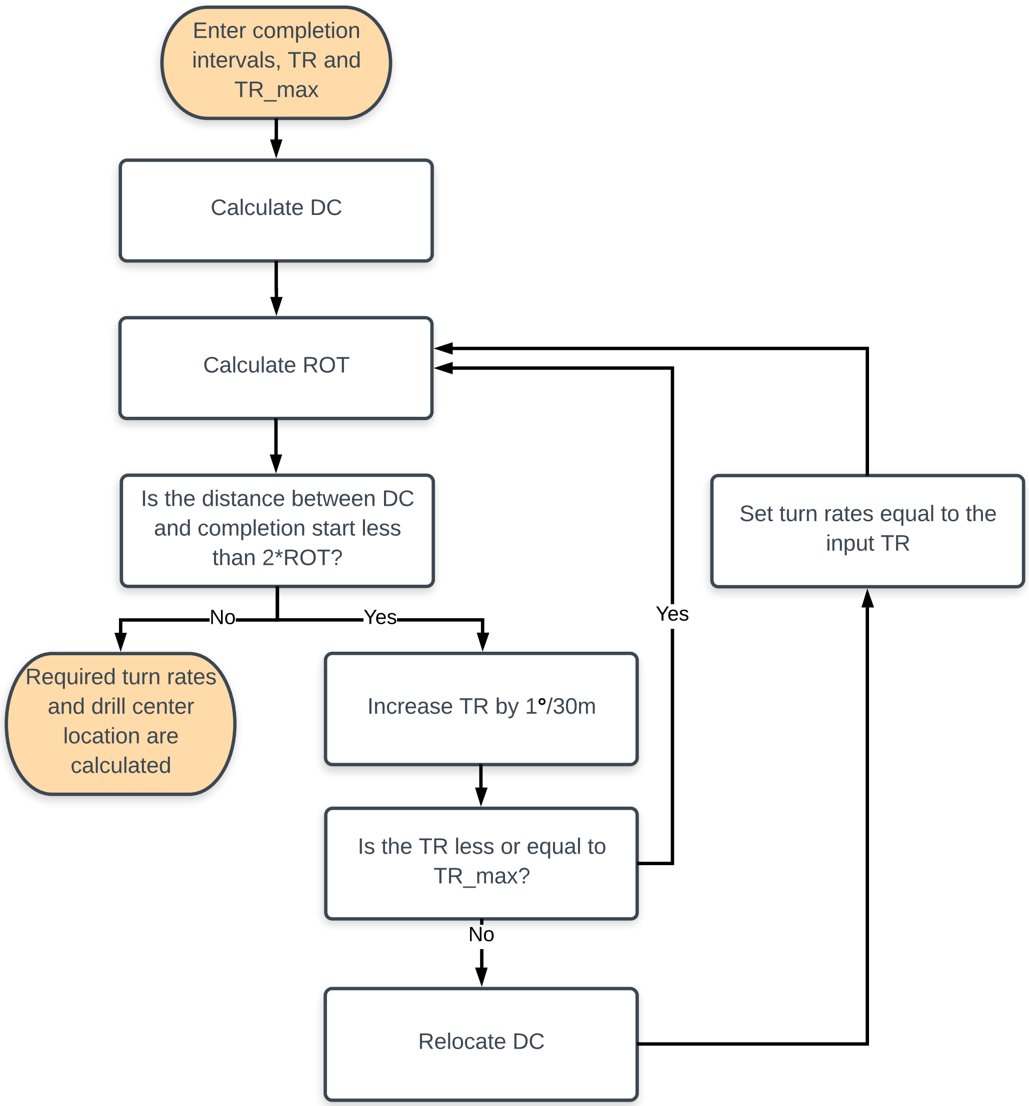


Figure 3.9. Methodology in the turn rate (TR) calculation in a Subsea on a Stick (SoS) architecture. If the TR is increased above the maximum turn rate (TR_max), the drill center (DC) is relocated.

3.4 2-slots Templates

Lillevik and Standal (2017) considered only 4-slots and 6-slots templates. The use of 2-slots templates is increasing on the Norwegian continental shelf (NCS). To improve the program, 2-slots templates are included. The combinations are set up with the methodology explained by Lillevik and Standal (2017) and in Chapter 2.1. Table 3.1 lists the dimensions of the matrices required when setting up all combinations in a 2-slots field development with 12 wells. The last matrix, Pos, contains all combinations.

Table 3.1

Matrix dimensions in 2-slots combination generation with 12 completion intervals.

Matrix name	Rows	Columns	Comment
R_1	66	2	Combinations in the 1st template
R_rest	66	10	Remaining completions
R_2	45	132	Combinations in the 2nd template
R_rest	45	528	Remaining completions
R_3	1260	132	Combinations in the 3rd template
R_rest	1260	396	Remaining completions
R_4	18900	132	Combinations in the 4th template
R_rest	18900	264	Remaining completions
R_5	113400	132	Combinations in the 5th template
R_6	113400	132	Combinations in the 6th template
Pos	113400	792	All combinations combined

The number of elements in Pos is 89812800 (rows × columns). Since every combination contains 12 elements, 7484400 ($\frac{89812800}{12}$) combinations exist for a 2-slots field architecture with 12 wells. This number is verified using Equation 2.1, see Equation 3.37.

$$\frac{12!}{2!(12-2)!} \times \frac{10!}{2!(10-2)!} \times \frac{8!}{2!(8-2)!} \times \frac{6!}{2!(6-2)!} \times \frac{4!}{2!(4-2)!} \times \frac{2!}{2!(2-2)!} = 7484400 \tag{3.37}$$

The function `get_six_dc` generates all combinations and calculates the optimal solution, see Appendix B.5. The flowchart in Figure 3.10 illustrates how the matrices subsequently are set up in `get_six_dc`.

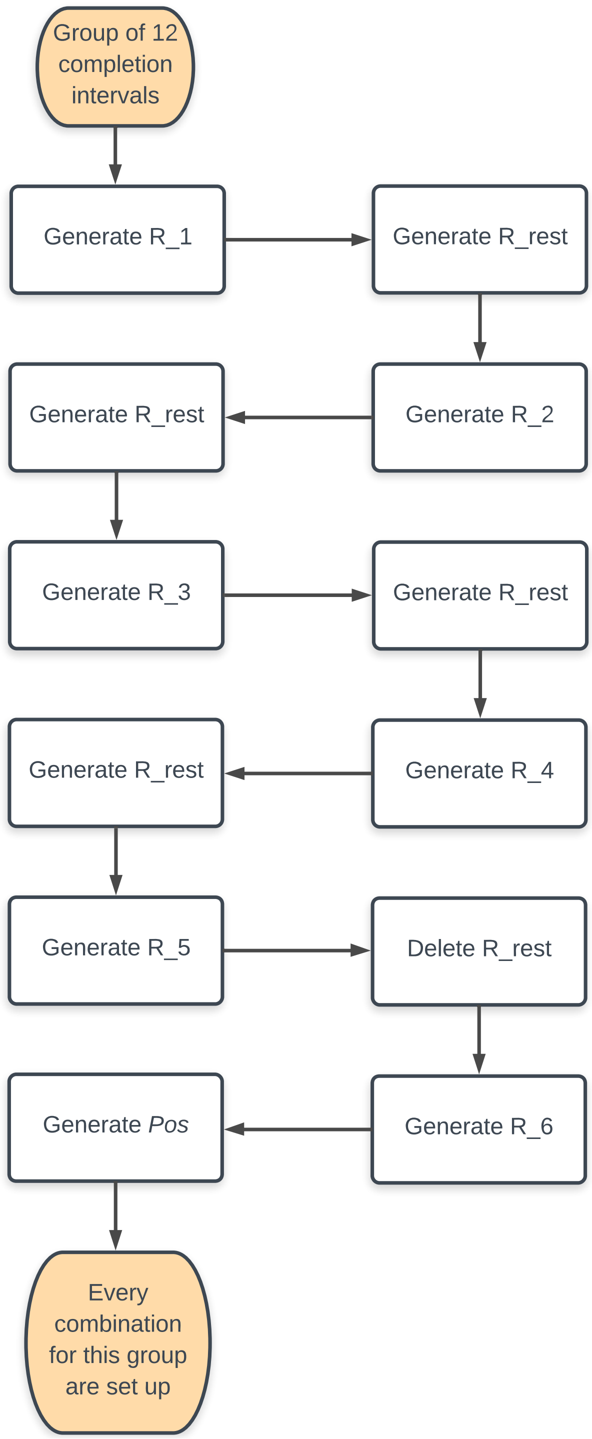


Figure 3.10. Generation of combinations in 2-slots template layout. Each R_n ($n \in 1,2,3,4,5,6$) matrix contains the possible combinations of the different templates. R_rest contains the remaining completions to be distributed among the remaining templates. The resulting Pos matrix contains all possible combinations of the 12 completions.

Chapter 4

Results

The results are presented in this chapter. As mentioned in Chapter 1.2, some parameters are set by the user: the kickoff point (KOPz), the first build-up rate (BUR1), the second build-up rate (BUR2), the drop-off rate (DOR), the turn rate (TR) and the maximum allowable turn rate (TR_max). Table 4.1 displays the input parameters that are used.

Table 4.1

Input parameters used in the calculations, their values and their units.

Input parameter	Abbreviation	Value	Unit
Kickoff point	KOPz	500.00	m
First build-up rate	BUR	3.00	°/30m
Second build-up rate	BUR2	3.00	°/30m
Drop-off rate	DOR	3.00	°/30m
Turn rate	TR	3.00	°/30m
Maximum turn rate	TR_max	6.00	°/30m

4.1 Combinations and Placement of Drill Centers

Two data sets are given, `fielddata1.mat` (FD1) and `fielddata2.mat` (FD2). The coordinates of the completion intervals are listed in Table A.1 and A.2 in Appendix A.1 and A.2. Both data sets are based on real field data, but they are adjusted to make them unrecognizable. See Appendix A for a description of which adjustments that are made.

As mentioned in Chapter 1.3 and Chapter 2.4, all calculations are performed on groups of 12 completion intervals at a time and the number of satellite wells required depends on the remainder after division by 12. To study the effect of required satellite wells, the 24 first completions of FD1 and FD2 are also studied.

4.1.1 `fielddata1.mat`

The resulting well paths from the completion intervals in FD1 are projected onto the XY-plane in the figures below. Each group of wells is colored to distinguish between the completion groups. The associated drill center(s) (DC) are placed were the wells meet. The results of the satellite wells, 2-slots template, 4-slots template, 6-slots template, and Subsea on a Stick (SoS) field layouts are shown in Figure 4.1, 4.2, 4.3, 4.4 and 4.5, respectively. The coordinates of each DC and the associated completion interval coordinates of the satellite wells, 2-slots template, 4-slots template, 6-slots template, and SoS field layouts are listed in Table A.3, A.4, A.5, A.6 and A.7, respectively. 3D plots of each field layout are shown in Figure A.1, A.2, A.3, A.4 and A.5, respectively. These tables and figures are found in Appendix A.3.1. Note that the wells are plotted for evenly spaced data points (every 75 meters). Thus, some wells may look crooked.

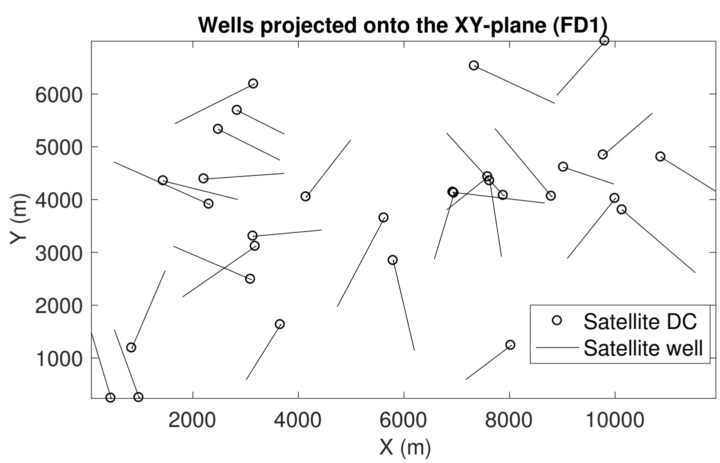


Figure 4.1. Projection of the well paths of the satellite wells in FD1. The drill centers (DCs) are marked with circles.

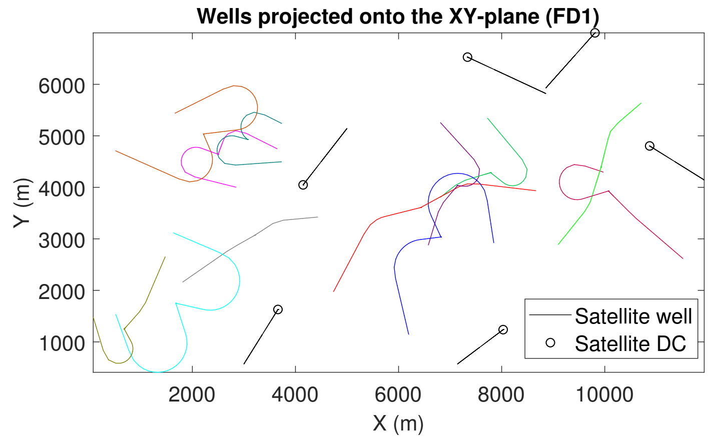


Figure 4.2. The projection of the resulting well paths from FD1 using 2-slots templates. Each group of wells have their own color to show that they belong together and are drilled from the same template. The satellite wells are colored black.

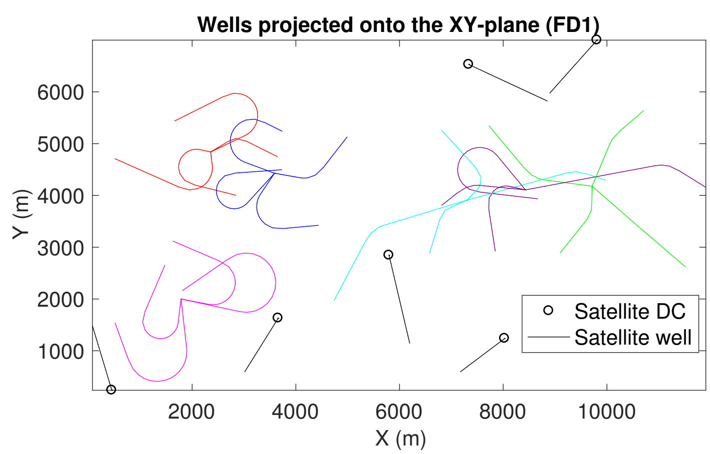


Figure 4.3. The projection of the resulting well paths from FD1 using 4-slots templates. Each group of wells have their own color to show that they belong together and are drilled from the same template. The satellite wells are colored black.

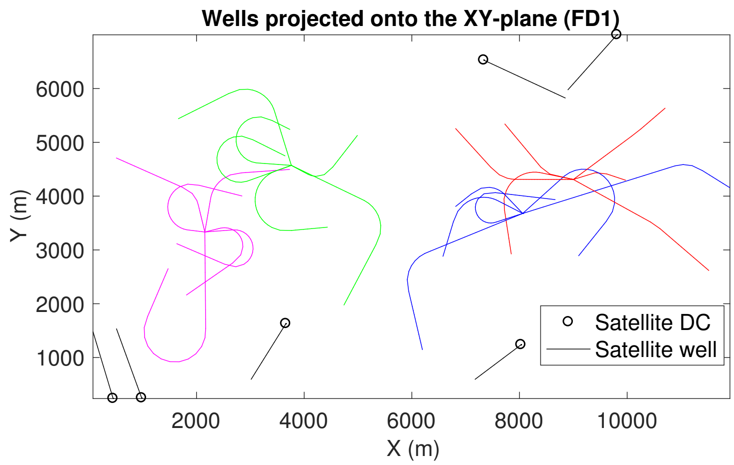


Figure 4.4. The projection of the resulting well paths from FD1 using 6-slots templates. Each group of wells have their own color to show that they belong together and are drilled from the same template. The satellite wells are colored black.

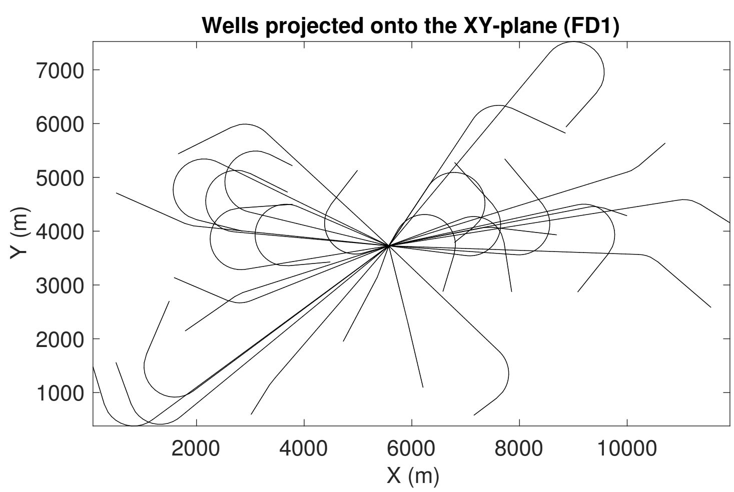


Figure 4.5. The projection of the resulting well paths from FD1 in a SoS layout.

4.1.2 `fielddata2.mat`

The resulting well paths from the completion intervals in FD2 are projected onto the XY-plane in the figures below. Each group of wells is colored to distinguish between the completion groups. The associated DC(s) are placed were the wells meet.

The results of the satellite wells, 2-slots template, 4-slots template, 6-slots template, and SoS field layouts are shown in Figure 4.6, 4.7, 4.8, 4.9 and 4.10, respectively.

The coordinates of each DC and the associated completion interval coordinates of the satellite wells, 2-slots template, 4-slots template, 6-slots template, and SoS field layouts are listed in Table A.8, A.9, A.10, A.11 and A.12, respectively. 3D plots of each field layout are shown in Figure A.6, A.7, A.8, A.9 and A.10, respectively. These tables and figures are found in Appendix A.3.2. Note that the wells are plotted for evenly spaced data points (every 75 meters). Thus, some wells may look crooked.

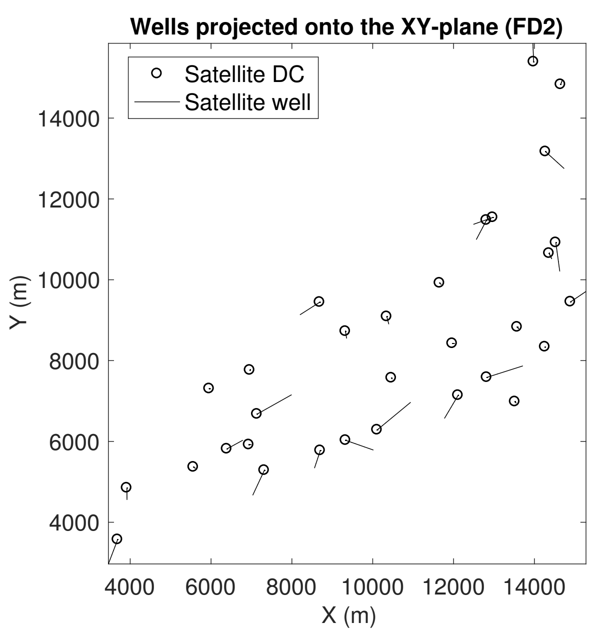


Figure 4.6. Projection of the well paths of the satellite wells in FD2. The drill centers (DCs) are marked with circles.

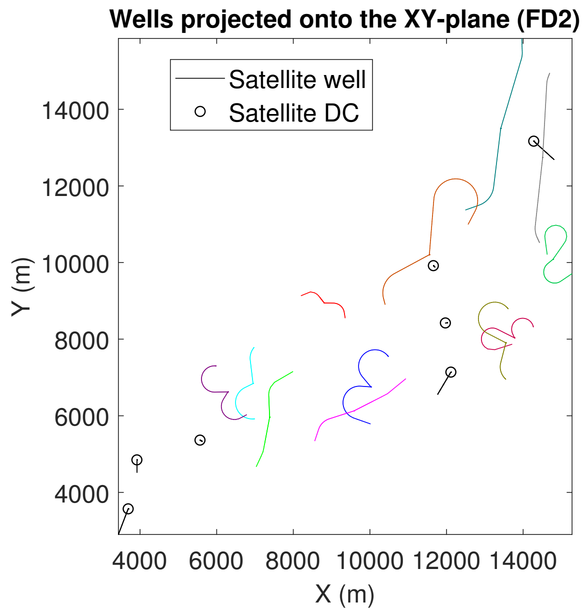


Figure 4.7. The projection of the resulting well paths from FD2 using 2-slots templates. Each group of wells have their own color to show that they belong together and are drilled from the same template. The satellite wells are colored black.

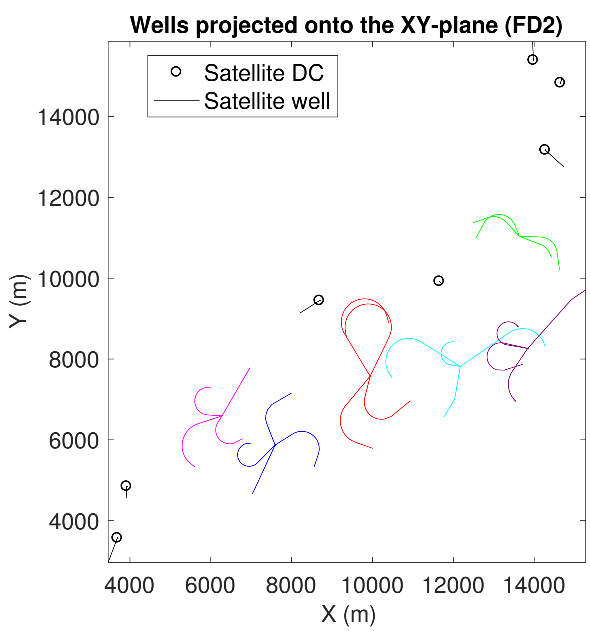


Figure 4.8. The projection of the resulting well paths from FD2 using 4-slots templates. Each group of wells have their own color to show that they belong together and are drilled from the same template. The satellite wells are colored black.

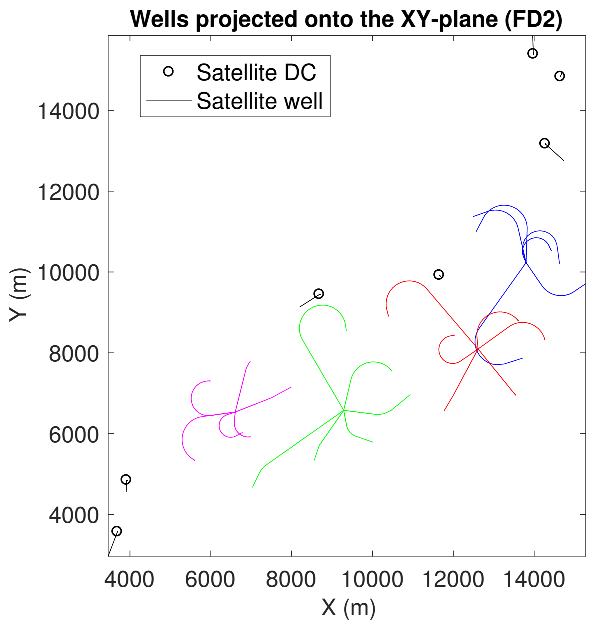


Figure 4.9. The projection of the resulting well paths from FD2 using 6-slots templates. Each group of wells have their own color to show that they belong together and are drilled from the same template. The satellite wells are colored black.

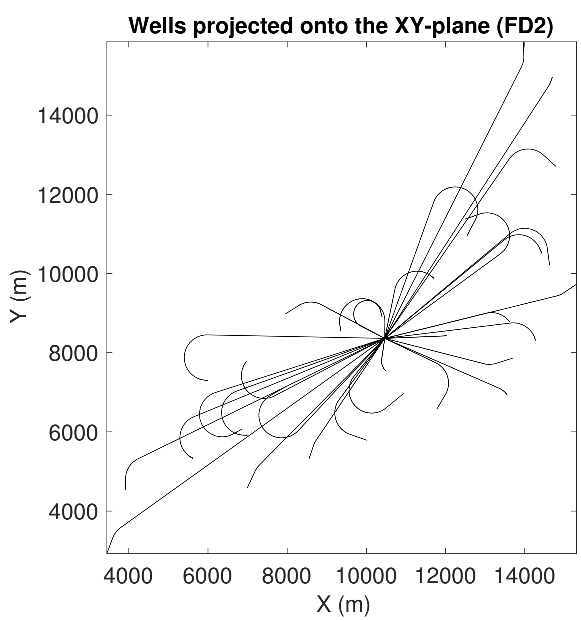


Figure 4.10. The projection of the resulting well paths from FD2 in a SoS layout.

4.1.3 Completion 1-24 from `fielddata1.mat`

The resulting well paths from the first 24 completion intervals in FD1 are projected onto the XY-plane in the figures below. Each group of wells is colored to distinguish between the completion groups. The associated DC(s) are placed were the wells meet.

The results of the satellite wells, 2-slots template, 4-slots template, 6-slots template, and SoS field layouts are shown in Figure 4.11, 4.12, 4.13, 4.14 and 4.15, respectively.

The coordinates of each DC and the associated completion interval coordinates of the satellite wells, 2-slots template, 4-slots template, 6-slots template, and SoS field layouts are listed in Table A.13, A.14, A.15, A.16 and A.17, respectively. 3D plots of each field layout are shown in Figure A.11, A.12, A.13, A.14 and A.15, respectively. These tables and figures are found in Appendix A.3.3. Note that the wells are plotted for evenly spaced data points (every 75 meters). Thus, some wells may look crooked.

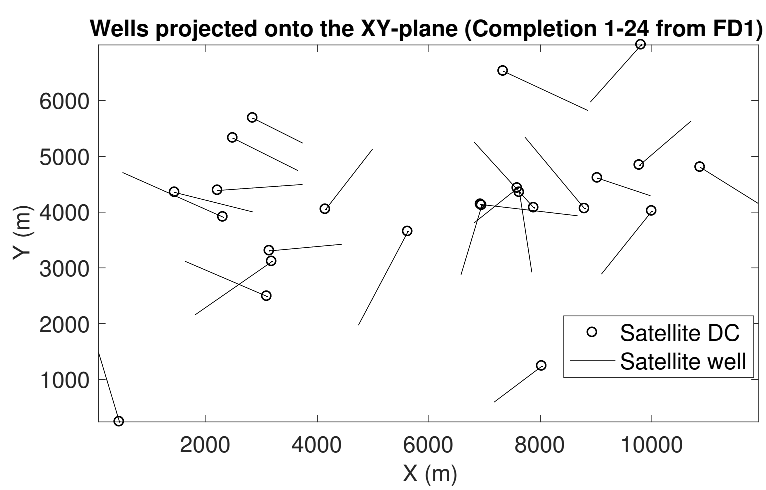


Figure 4.11. Projection of the resulting well paths of the first 24 completion intervals in FD1 in a satellite field layout. The drill centers (DCs) are marked with circles.

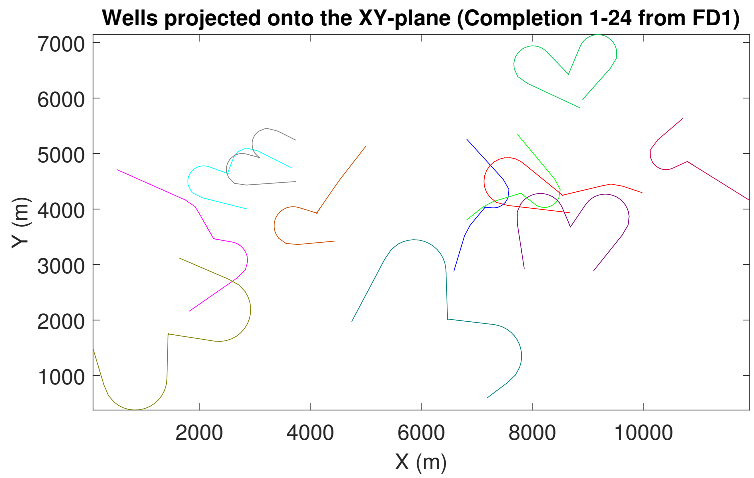


Figure 4.12. The projection of the resulting well paths from the first 24 completion intervals in FD1 using 2-slots templates. Each group of wells have their own color to show that they belong together and are drilled from the same template.

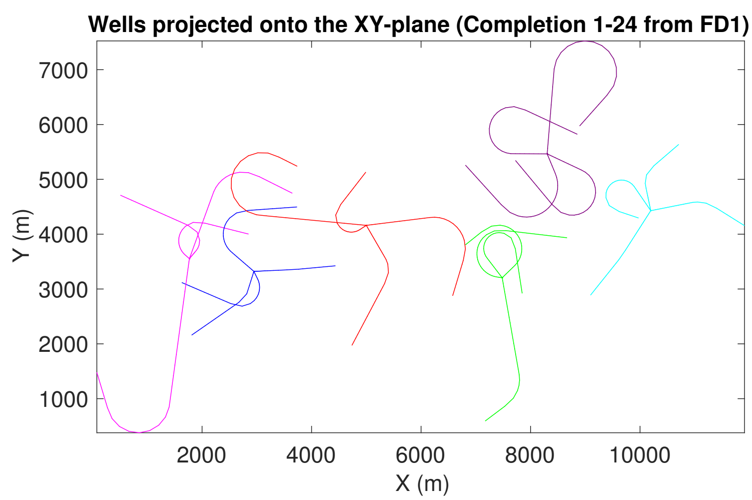


Figure 4.13. The projection of the resulting well paths from the first 24 completion intervals in FD1 using 4-slots templates. Each group of wells have their own color to show that they belong together and are drilled from the same template.

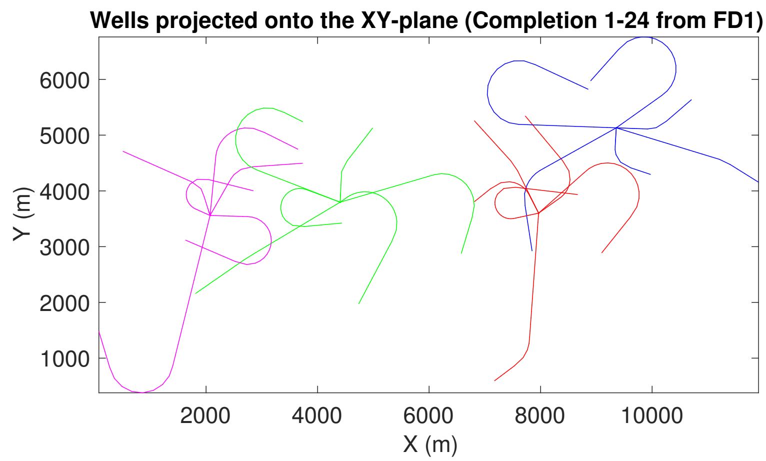


Figure 4.14. The projection of the resulting well paths from the first 24 completion intervals in FD2 using 6-slots templates. Each group of wells have their own color to show that they belong together and are drilled from the same template.

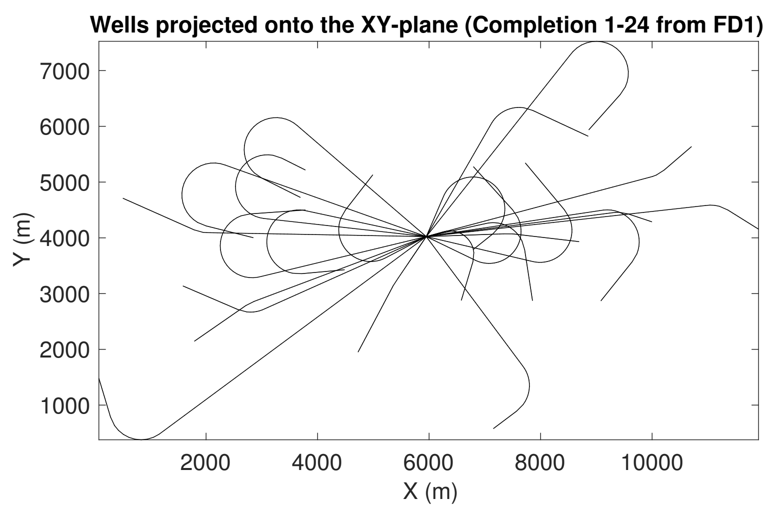


Figure 4.15. The projection of the resulting well paths from the first 24 completion intervals in FD1 in a SoS layout.

4.1.4 Completion 1-24 from `fielddata2.mat`

The resulting well paths from the first 24 completion intervals in FD2 are projected onto the XY-plane in the figures below. Each group of wells is colored to distinguish between the completion groups. The associated DC(s) are placed were the wells meet.

The results of the satellite wells, 2-slots template, 4-slots template, 6-slots template, and SoS field layouts are shown in Figure 4.16, 4.17, 4.18, 4.19 and 4.20, respectively.

The coordinates of each DC and the associated completion interval coordinates of the satellite wells, 2-slots template, 4-slots template, 6-slots template, and SoS field layouts are listed in Table A.18, A.19, A.20, A.21 and A.22, respectively. 3D plots of each field layout are shown in Figure A.16, A.17, A.18, A.19 and A.20, respectively. These tables and figures are found in Appendix A.3.4. Note that the wells are plotted for evenly spaced data points (every 75 meters). Thus, some wells may look crooked.

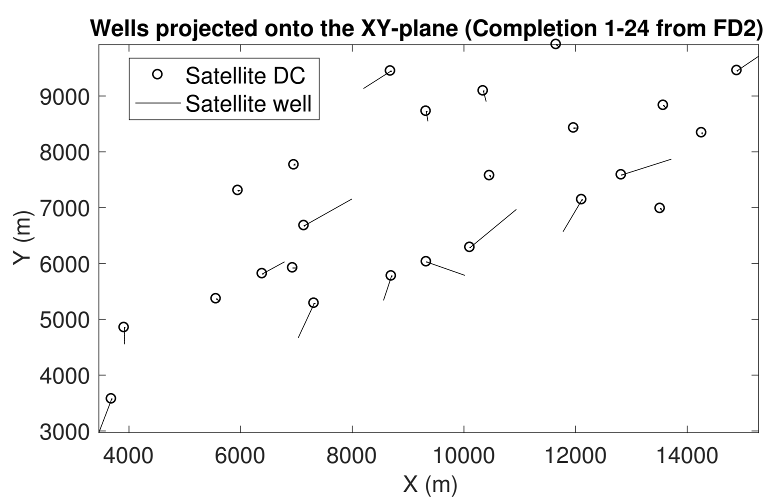


Figure 4.16. Projection of the resulting well paths of the first 24 completion intervals in FD2 in a satellite field layout. The drill centers (DCs) are marked with circles.

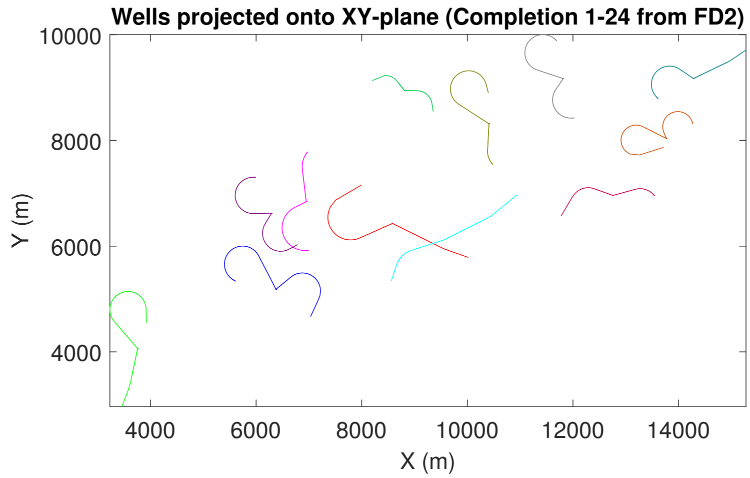


Figure 4.17. The projection of the resulting well paths from the first 24 completion intervals in FD2 using 2-slots templates. Each group of wells have their own color to show that they belong together and are drilled from the same template.

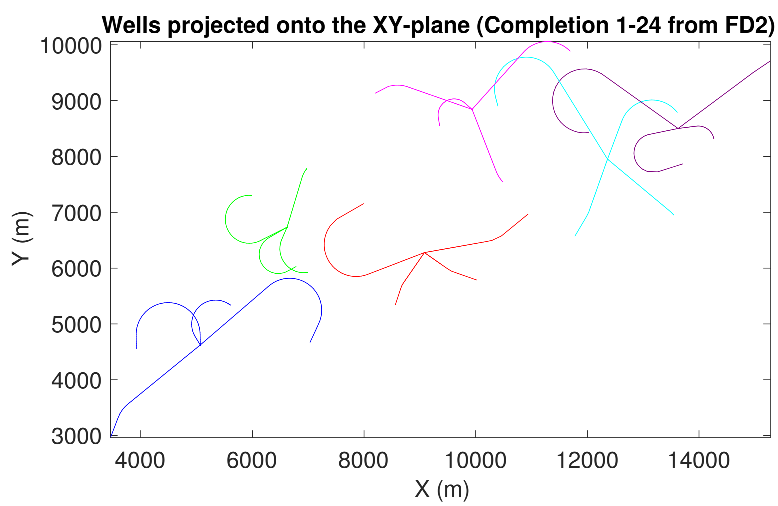


Figure 4.18. The projection of the resulting well paths from the first 24 completion intervals in FD2 using 4-slots templates. Each group of wells have their own color to show that they belong together and are drilled from the same template.

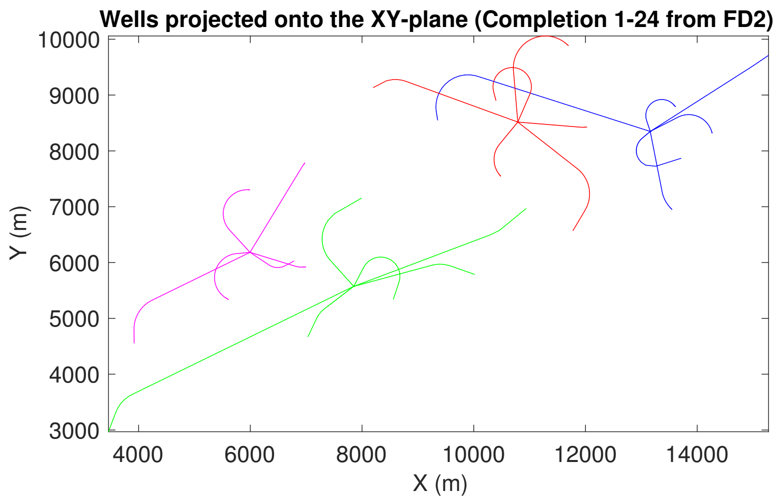


Figure 4.19. The projection of the resulting well paths from the first 24 completion intervals in FD2 using 6-slots templates. Each group of wells have their own color to show that they belong together and are drilled from the same template.

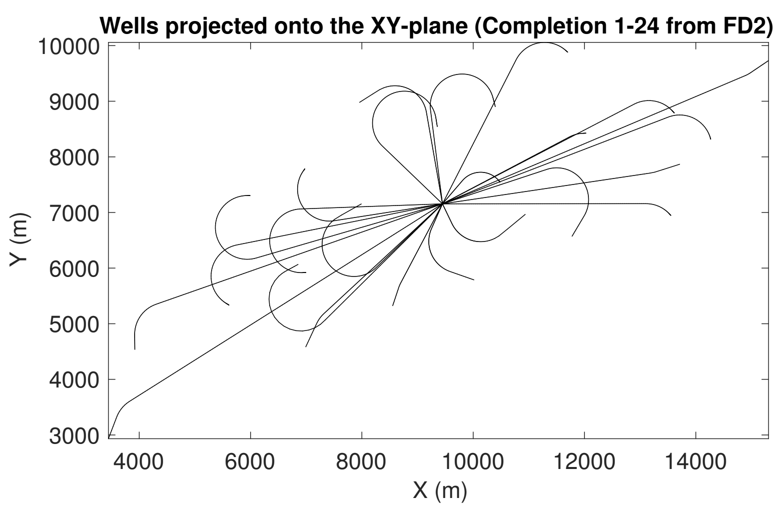


Figure 4.20. The projection of the resulting well paths from the first 24 completions in FD2 in a SoS layout.

4.2 Wellbore Trajectory Calculations

This section presents the average well path lengths (WPL) calculated from the methods in Chapter 2.4 and Chapter 3. The average and total WPL is calculated for the original field datas FD1 and FD2. To study the effect of required satellite wells, the average and total WPL is also calculated for the 24 first completion intervals in both field datas. At last, to investigate the effect of the target depth of the wells, the total and average WPL is calculated for FD1 with different depths.

Table 4.2 and Table 4.3 show the resulting WPLs from each field layout for the original field datas FD1 and FD2, respectively.

Table 4.2

Resulting well path lengths (WPL) from FD1. 30 completion intervals were used as input, which resulted in 6 satellite wells in the field architectures with templates.

	Satellite wells	2-slots template	4-slots template	6-slots template	Subsea on a Stick
Average WPL (m)	3744,84	3916,92	4103,51	4195,02	6058,83
Total WPL (km)	112,35	117,51	123,11	125,85	181,77

Table 4.3

Resulting well path lengths (WPL) from FD2. 31 completion intervals were used as input, which resulted in 7 satellite wells in the field architectures with templates.

	Satellite wells	2-slots template	4-slots template	6-slots template	Subsea on a Stick
Average WPL (m)	4772,66	5031,87	5079,67	5166,39	7185,42
Total WPL (km)	147,95	155,99	157,47	160,16	222,75

Table 4.4 and Table 4.5 show the resulting WPL from each field layout for the first 24 completion intervals in FD1 and FD2, respectively.

Table 4.4

Resulting well path lengths (WPL) from FD1. 24 completion intervals were used as input, thus there were no satellite wells in the field architectures with templates.

	Satellite wells	2-slots template	4-slots template	6-slots template	Subsea on a Stick
Average WPL (m)	3712,56	4016,33	4270,72	4413,32	5895,90
Total WPL (km)	89,10	96,39	102,50	105,92	141,50

Table 4.5

Resulting well path lengths from FD2. 24 completion intervals were used as input, thus there were no satellite wells in the field architectures with integrated templates.

	Satellite wells	2-slots template	4-slots template	6-slots template	Subsea on a Stick
Average WPL (m)	4781,23	4984.81	5226,75	5282,45	6643,21
Total WPL (km)	114,75	119,64	125,44	126,78	159,44

Table 4.6 and Table 4.7 show the average and total WPLs of FD1 with three different target depths. The original depth is 2500 meters. Since the grouping of completions and the resulting position of the DCs is not affected by the target depth, the coordinates of each DC and the associated completion interval coordinates of the satellite wells-, 2-slots template, 4-slots template, 6-slots template and SoS field layouts are listed in Table A.3, A.4, A.5, A.6 and A.7, respectively. These tables are found in Appendix A.3.1.

Table 4.6

Resulting average well path lengths with changing depths in FD1. 30 completion intervals were used as input, which resulted in 6 satellite wells in the field architectures templates.

Depth (m)	Average well path length (m)				
	Satellite wells	2-slots template	4-slots template	6-slots template	Subsea on a Stick
1500	2744,84	3072,34	3358,60	3494,78	5739,96
2500	3744,84	3916,92	4103,51	4195,02	6058,83
3500	4744,84	4857,92	4990,46	5057,55	6591,59

Table 4.7

Resulting total well path lengths with changing depths in FD1. 30 completion intervals were used as input, which resulted in 6 satellite wells in the field architectures with templates.

Depth (m)	Total well path length (km)				
	Satellite wells	2-slots template	4-slots template	6-slots template	Subsea on a Stick
1500	82,35	92,17	100,76	104,84	172,20
2500	112,35	117,51	123,11	125,85	181,77
3500	142,35	145,74	149,71	151,73	197,75

Chapter 5

Discussion

This chapter discusses the results in Chapter 4. Simplified cost calculations are performed to show how much the drilling costs can be reduced by subsea developments, compared to a Subsea on a Stick (SoS) field. Sensitivities such as true vertical depth (TVD), field distribution, and required satellite wells are also discussed.

5.1 Costs

The program calculates the average well path length (WPL) based on the total WPL of each field architecture. The total WPL is included in the comparison of costs to highlight the economical differences between the field architectures. To perform an economical evaluation, the following assumptions are made:

- Average rate of penetration (ROP) is 90 meters per day
- Drilling expenditure (DRILLEX) is 5,000,000 NOK per day

DRILLEX is a rough estimate of costs associated with daily rig fee, well construction, well completion, and well services. The assumptions are based on numbers presented by Stanko (2017) and Brechan et al. (2016).

5.1.1 Total Well Path Length

Table 4.2 and 4.3 list the total WPL of each field architecture in `fielddata1.mat` (FD1) and `fielddata2.mat` (FD2). These results are plotted in Figure 5.1. The diagram shows a significant reduction in total WPL when choosing a subsea development over a SoS layout. The most significant decrease in drilling length is obtained when choosing a satellite development compared to a SoS solution. Comparing the template architectures, the total WPL reduction stagnates. As the number of templates increase, and the number of slots per template decrease, the drilling length advantage is insignificant.

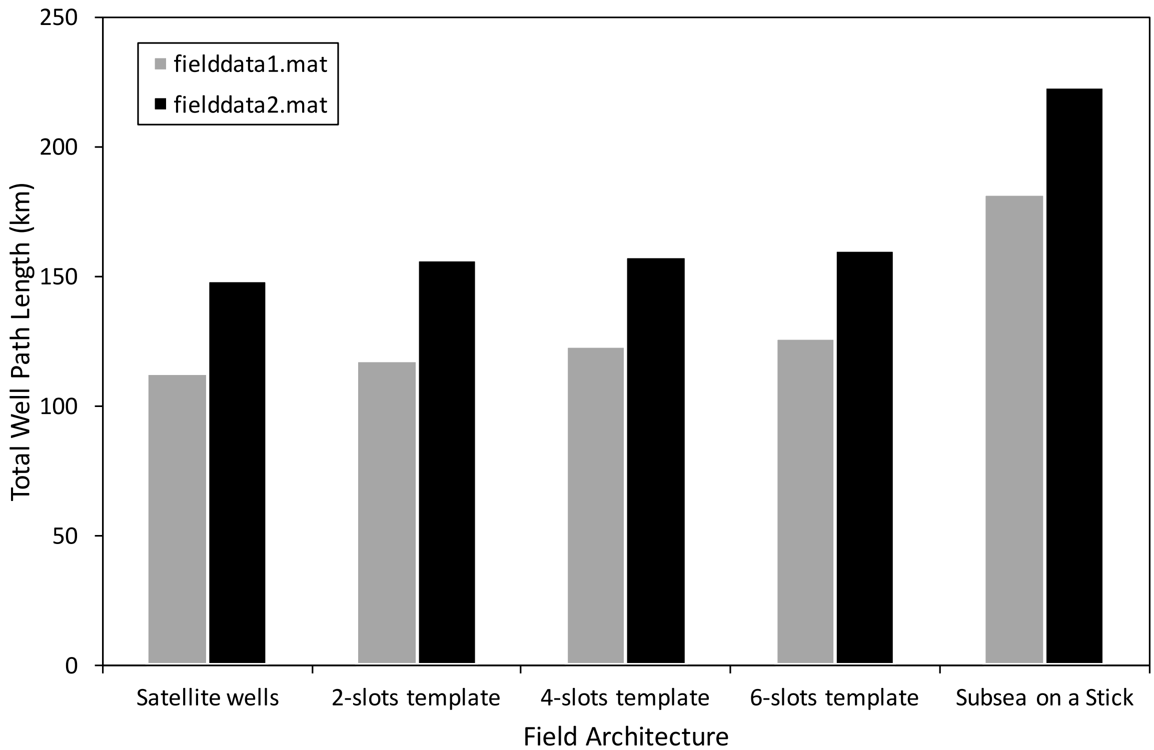


Figure 5.1. The total well path length for five different field layouts using `fielddata1.mat` (FD1) and `fielddata2.mat` (FD2).

Subsea On a Stick versus Template Architectures

Table 4.2 and Table 4.3 are used to calculate the exact differences in total WPL. Using FD1, the reductions are 55.91 km, 58.66 km, and 64.26 km when comparing a SoS layout to a layout with 6-slots, 4-slots, and 2-slots templates, respectively. Using Equation 5.1, 5.2, and 5.3, these differences equal 3.11×10^9 NOK, 3.26×10^9 NOK, and 3.57×10^9 NOK in drilling costs.

$$\frac{55910\text{m} \times 5,000,000\frac{\text{NOK}}{\text{day}}}{90\frac{\text{m}}{\text{day}}} = 3.11 \text{ BNOK} \tag{5.1}$$

$$\frac{58660\text{m} \times 5,000,000\frac{\text{NOK}}{\text{day}}}{90\frac{\text{m}}{\text{day}}} = 3.26 \text{ BNOK} \tag{5.2}$$

$$\frac{64260\text{m} \times 5,000,000\frac{\text{NOK}}{\text{day}}}{90\frac{\text{m}}{\text{day}}} = 3.57 \text{ BNOK} \tag{5.3}$$

Using FD2, the drilling lengths are reduced by 62.59 km, 65.28 km, and 66.76 km when choosing a template layout with 6-slots, 4-slots, and 2-slots over a SoS layout, respectively. Equation 5.4, 5.5, and 5.6 estimate these reductions to equal 3.48×10^9 NOK, 3.63×10^9 NOK, and 3.71×10^9 NOK cut in drilling costs.

$$\frac{62590\text{m} \times 5,000,000\frac{\text{NOK}}{\text{day}}}{90\frac{\text{m}}{\text{day}}} = 3.48 \text{ BNOK} \tag{5.4}$$

$$\frac{65280\text{m} \times 5,000,000\frac{\text{NOK}}{\text{day}}}{90\frac{\text{m}}{\text{day}}} = 3.63 \text{ BNOK} \tag{5.5}$$

$$\frac{66760\text{m} \times 5,000,000\frac{\text{NOK}}{\text{day}}}{90\frac{\text{m}}{\text{day}}} = 3.71 \text{ BNOK} \tag{5.6}$$

To summarize, the total WPL and drilling costs decrease as the number of drill centers (DC) increases. Regardless of the number of slots, template structures significantly reduce the drilling length compared to SoS. If the remaining field development costs are kept below the cost reductions presented above, a subsea development with templates is favourable compared to SoS.

Subsea on a Stick versus Satellite Architecture

The satellite wells are constructed to have the shortest possible WPL. Consequently, the economical gains with respect to drilling cost are highest when choosing a satellite field development. Studying FD1 and FD2 in Table 4.2 and 4.3, the drilling cutbacks are peaking at 69.42 km and 74.80 km, respectively. These savings are obtained when comparing the most opposing field architectures: SoS vs satellite wells. Equation 5.7 and 5.8 convert the reduced kilometers to NOK for FD1 and FD2, respectively.

$$\frac{69420\text{m} \times 5,000,000\frac{\text{NOK}}{\text{day}}}{90\frac{\text{m}}{\text{day}}} = 3.86 \text{ BNOK} \tag{5.7}$$

$$\frac{74800\text{m} \times 5,000,000\frac{\text{NOK}}{\text{day}}}{90\frac{\text{m}}{\text{day}}} = 4.16 \text{ BNOK} \tag{5.8}$$

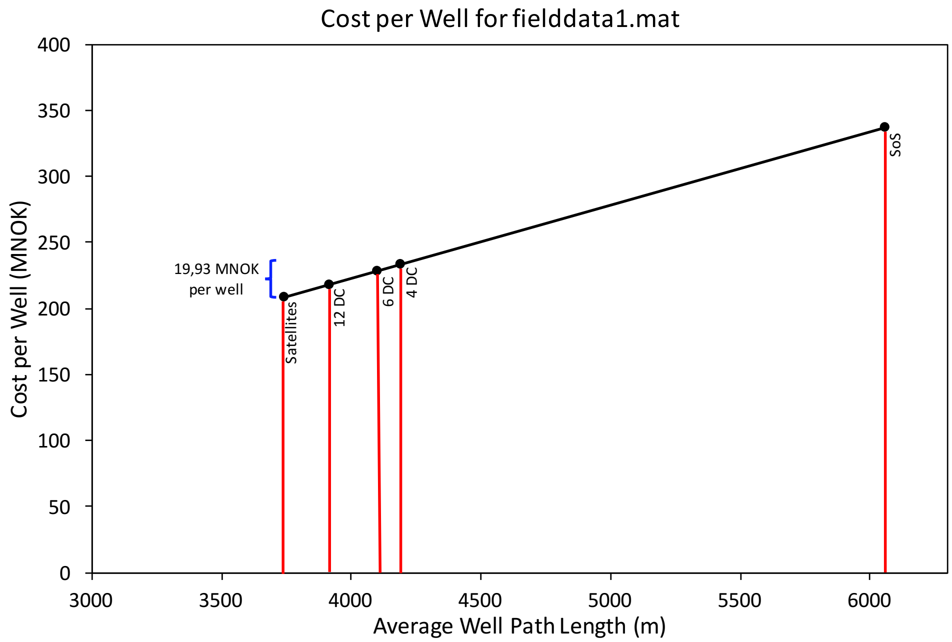
3.86×10^9 NOK and 4.16×10^9 NOK are significant amounts of capital. If the remaining field development costs are kept below these numbers, satellite wells are favourable compared to SoS.

5.1.2 Average Well Path Length

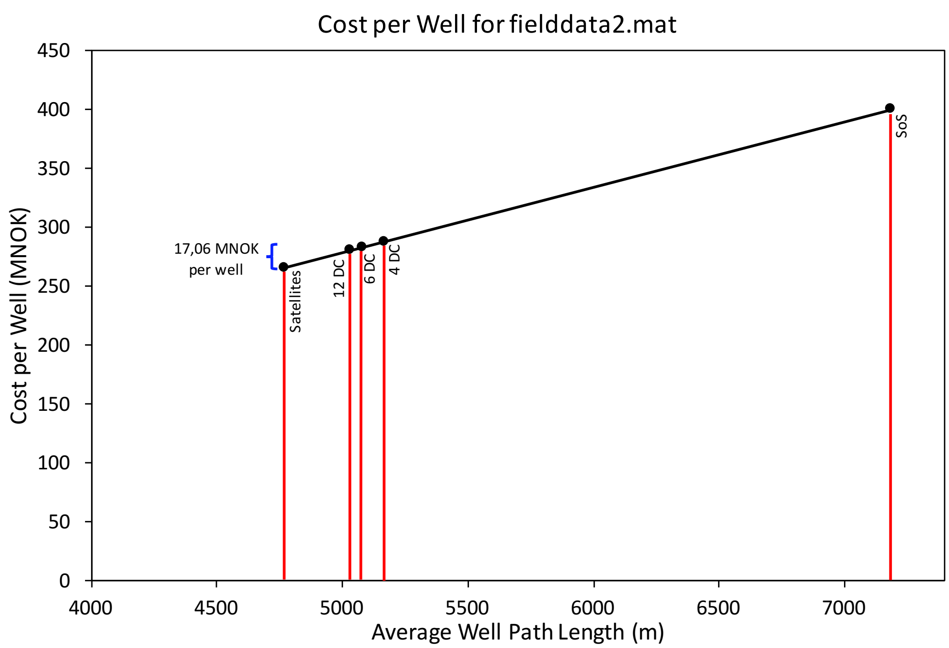
Average WPL is listed in Table 4.2 and 4.3 for FD1 and FD2, respectively. Using the average WPL, the cost per well is calculated by Equation 5.9.

$$\text{Cost per well (MNOK)} = \frac{\text{Average WPL} \times 5,000,000\frac{\text{NOK}}{\text{day}} \cdot 10^{-6}}{90\frac{\text{m}}{\text{day}}} \tag{5.9}$$

Cost per well versus average WPL is plotted for FD1 and FD2 in Figure 5.2. The different field architectures are marked, and the template layouts are denoted by the number of DCs. There are 12 DCs, 6 DCs and 4 DCs in the case of 2-slots, 4-slots and 6-slots template architectures, respectively.



(a) FD1.



(b) FD2.

Figure 5.2. Cost per well versus average well path length for FD1 and FD2. The red lines represent the different field architectures. There are 12 DCs, 6 DCs and 4 DCs in the case of 2-slots, 4-slots and 6-slots template architectures, respectively. The blue bracket represents the decrease in drilling cost per well when choosing a satellite development over a layout with 4-slots templates.

Table 5.1 highlights the cost per well for three different field layouts; satellite wells, 4-slots templates and SoS. The comparison below is based on these numbers.

Table 5.1

Cost per well for three different field layouts. The calculations are performed for FD1 and FD2.

	Cost per well (MNOK)	
Field layout	`fielddata1.mat`	`fielddata2.mat`
Satellite wells	208.05	265.15
4-slots template	227.97	282.20
Subsea on a Stick	336.60	399.19

Choosing a 4-slots template layout instead of a SoS solution yields a reduction of 32% and 29% in drilling cost per well, for FD1 and FD2, respectively. Choosing a satellite layout over a SoS solution yields a corresponding reduction of 38% and 34%, respectively. Furthermore, developing FD1 and FD2 with satellite wells instead of 4-slots templates yields an average cutback of 9% and 6% per well, respectively. The most competitive field layout is identified when the remaining field development costs are added to these estimates.

5.2 Sensitivites

5.2.1 True Vertical Depth

The results from Chapter 5.2 are affected by the TVD. Therefore, the effect of different target depths is studied. In FD1, all completion intervals are horizontal and originally located at 2500 meters depth. The TVD is manipulated; all depths are first reduced by 1000 meters and then increased by 1000 meters from the original depth. Table 4.6 and Table 4.7 in Chapter 4 lists the results.

In the reference case of FD1 (TVD=2500 meters), the difference in total WPL is 55.91 km when comparing the 6-slots templates layout with the SoS solution. When the TVD is 1500 meters, the same difference in total WPL is 67.36 km. When the TVD is 3500 meters, the same difference is 46.02 km. Hence, the differences in total WPL between subsea development and SoS increases with shallower wells.

The effect is also evident when studying the average WPL. In the reference case of FD1, the average WPL decreases with 1863.81 meters when developing the field with 6-slots templates instead of SoS. Considering the increase and decrease in TVD, the resulting cutbacks per well are 1534.04 meters and 2245,18 meters. These results demonstrate that there is a significant correlation between TVD and drilling expenses saved when choosing a subsea layout instead of SoS. As the TVD increases, the drilling length reduction decreases.

5.2.2 Field Distribution

FD1 and FD2 are plotted in Figure 5.3. The figure is plotted in the XY-coordinate system that all calculations are based upon. Figure 5.3 shows that FD2 is more distributed than FD1.

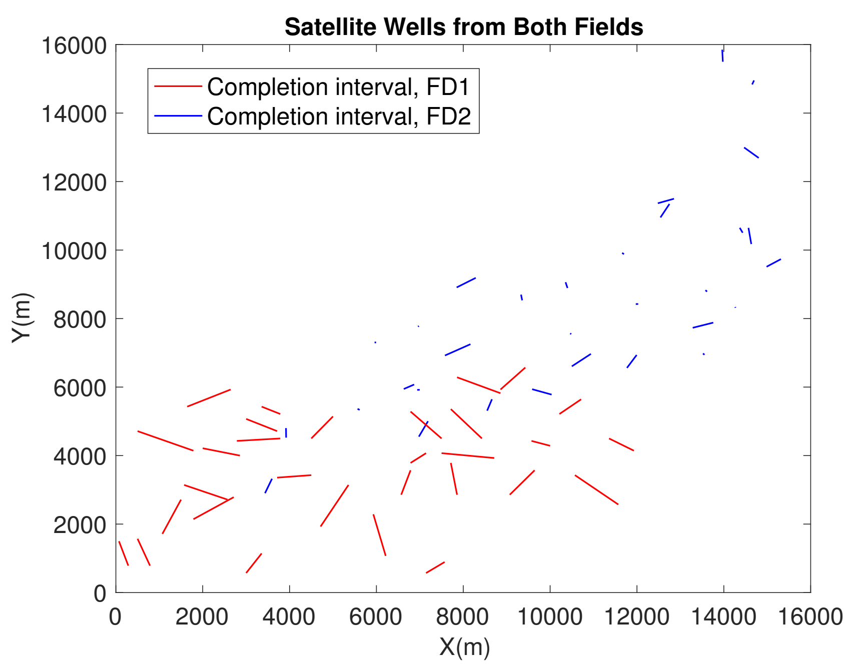


Figure 5.3. `fielddata1.mat` (FD1), colored red, and `fielddata2.mat` (FD2), colored blue, plotted in the XY-plane to show their different distribution.

Studying Table 4.2 and Table 4.3 in Chapter 4, the results show that field distribution affect the WPL considerably. As FD1 is more gathered, the economical advantage of several DCs decreases.

For example, choosing a 4-slots template over a SoS solution saves a total WPL of 58.66 km in FD1. In the more distributed field, FD2, the same decision saves a total WPL of 65.28 km. Using equation 5.10, this difference is equivalent to 367.70 MNOK.

$$\frac{(65,28-58,66)\mathrm{m}\times 5,000,000\frac{\mathrm{NOK}}{\mathrm{day}}\cdot 10^{-6}}{90\frac{\mathrm{m}}{\mathrm{day}}}=367.70\mathrm{MNOK} \tag{5.10}$$

Considering Figure 5.3 and the results in Table 4.2 and Table 4.3, it can be seen that as the input completions are scattered, the benefit of subsea development increases. However, regardless of the distribution, both fields strongly imply that subsea development is favourable with respect to minimizing drilling length.

5.2.3 Required Satellite Wells

As mentioned in Chapter 1.3 and Chapter 2.4, the subsea layouts with templates will have some satellite wells if the remainder after division by 12 is not equal to 0. FD1 consists of 30 completion intervals, and consequently 6 satellite wells are required in the template layouts. FD2 consists of 31 completion intervals, and consequently 7 satellite wells are required. The effect of required satellite wells on average WPL is studied. Calculations are performed on the first 24 completion intervals in FD1 and FD2.

Comparing the layouts with templates in Table 4.2 and Table 4.4, the trend is that the template layouts with satellite wells have shorter average WPL. This is expected as the required satellite wells have the shortest possible WPL. Consequently, the average WPL of all the wells is reduced. Using FD1, 6 satellite wells minimize the average WPL in the developments with templates. The same overall trend is observed when comparing Table 4.3 and Table 4.5. Using FD2, seven satellite wells minimize the average WPL in two out of three template layouts.

Figure 5.4 shows how the average WPL decreases when developing a field with 4-slots templates instead of a SoS solution. From FD1, the average decrease is 1955.32 meters per well with satellite wells and 1625.18 meters without. These results indicate that as the number of required satellite wells increases, the subsea template developments get an overestimated advantage.

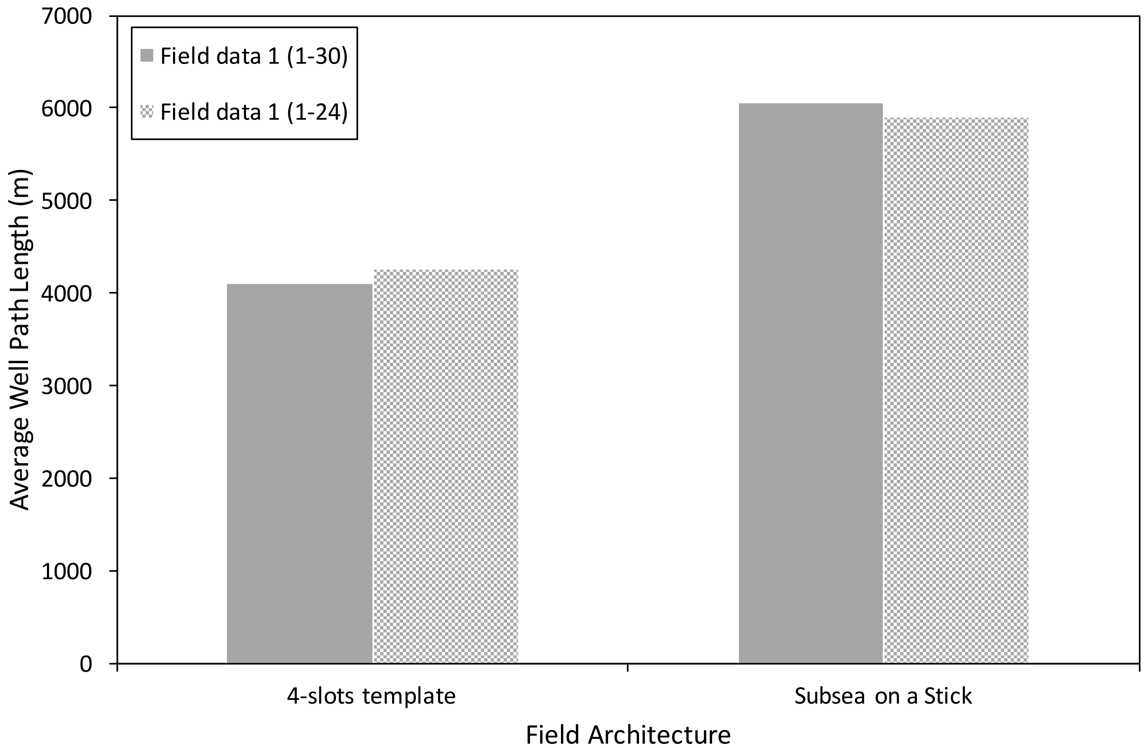


Figure 5.4. Average WPL vs. two field architectures, using `fielddata1.mat` (FD1). The grey columns represent the original field with 30 completion intervals, and the patterned columns represent the field with only 24 completion intervals (no satellite wells).

Chapter 6

Conclusion

6.1 Conclusion

The aim of this thesis is to complete the drilling part of a tool that can identify the optimal subsea field development for fixed completion intervals. Instead of minimizing the cost of individual contracts (drilling contractor, subsea hardware and subsea installation), the final tool will minimize the sum of these costs. This thesis completes the drilling aspect of this tool by comparing the drilling lengths in different field layouts. The wellbore trajectories in each layout are optimized to be as short as practically possible.

Developing `fielddata1.mat` (FD1) and `fielddata2.mat` (FD2) with 4-slots templates reduces the the total well path length (WPL) with 58.66 kilometers (3.26 billions NOK) and 65.28 kilometers (3.63 billions NOK), respectively, compared to a Subsea on a Stick (SoS) layout. Developing the fields with satellite wells, equals a cutback in total WPL of 69.42 km (3.86 billions NOK) and 74.80 km (4.16 billions NOK) for FD1 and FD2, respectively. To conclude, the drilling contractor costs can be significantly reduced by choosing a subsea solution. However, cutbacks in drilling length have penalties. A subsea development bring along tie-backs from satellite wells to manifold, wet well components, trawl protections, and subsea intervention among other components. The costs of these components may be higher than the costs saved in drilling. Consequently, the program and results provided in this thesis can not identify the optimal field layout yet. The remaining field development costs have to be included and considered. Nevertheless, the WPL and wellbore trajectories provided by this program minimize the drilling contractor costs in each field architecture.

The extension of this work would include a development of a subsea EPCI (Engineering, Procurement, Construction and Installation) program. This program has to be compatible with the drilling program developed in this thesis. The future subsea EPCI program combined with this drilling program, will be the first version of an automatic tool that can identify the optimal subsea field layout for any set of fixed completion intervals.

6.2 Recommendations for Further Work

This section discusses work that remains to be done. The first section describes the limitation with the parameters used to calculate drilling costs. The second section gives a short description of the main subsea hardware and installation costs that should be included in an analysis of the overall field development costs.

6.2.1 Drilling Costs

The calculations performed on drilling costs are based on a constant relation between drilling depth and daily rate of penetration (ROP). The ROP is based on an average and is equal to 90 meters per day if drilling takes place at 100 meters measured depth (MD) or 1000 meters MD. This is not the case in practice. The daily ROP decreases when the depth of drilling increases, due to tripping time among other factors. Thus, a more precise equation should be used to calculate the drilling costs.

6.2.2 Subsea Hardware and Installation

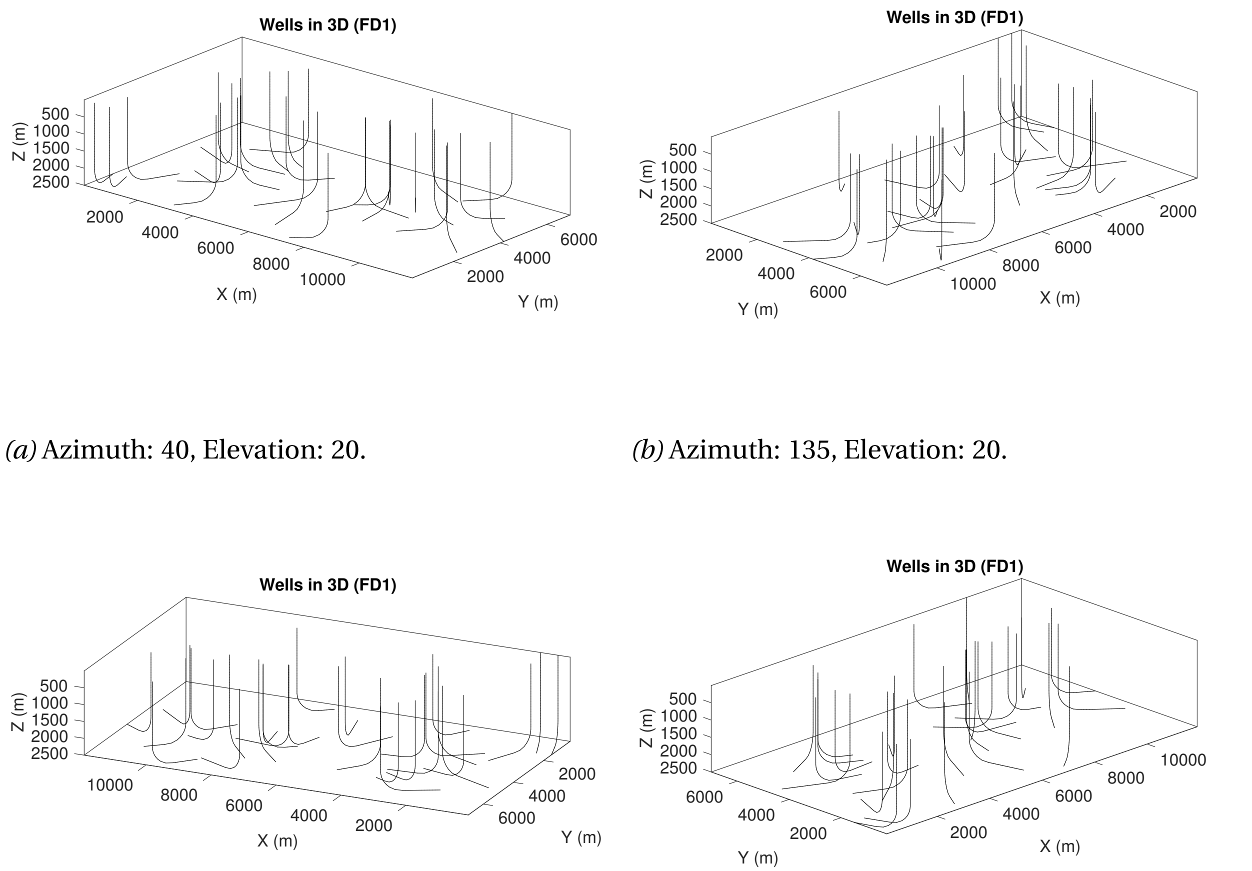
As mentioned in Chapter 6.1, a program that calculates the subsea hardware and installation costs for field layouts must be developed to complete the automatic tool that calculates the overall field development costs. There are a lot of factors that must be considered, and the user should be able to influence the field architecture to some extent. For example which foundation the templates shall have: pile, suction anchor or mud mats. Then there are different template systems: hinge-over subsea template (HOST), flow base structure (FBS), Cap-X or integrated template structure (ITS). There are also different types of pipeline, flowline and jumper spool solutions. Each solution has their own price tag.

In addition, there are different installation methods for the different template systems, pipelines and flowlines. For example, pipelines can be installed in several ways: reel based, S- and J-lay, in addition to three different towed-based installation methods. For the template systems, their size decide whether they can be installed through a moonpool on a drilling rig or by an installation vessel. Thus, there are many factors to consider when creating the subsea EPCI program.

The example below is made to show how the different architectures affect the total field development costs. The example is made for FD1 in two different layouts: satellite wells and 4-slots ITS.

Satellite Wells Layout

`WellboreTrajectorySatellites` is run for the completion intervals in FD1, using the codes in Appendix B.3 and B.6. Figure 6.1 shows the resulting satellite wells in 3D. The wells are shown from different angles to show the distribution of the satellite wells. From Figure 6.1 it is seen that the rig has to be moved between most of the drilling operations. On the other hand, the wells are short and have little deviation, thus they are easier to drill.



(a) Azimuth: 40, Elevation: 20.

(b) Azimuth: 135, Elevation: 20.

(c) Azimuth: -155, Elevation: 20.

(d) Azimuth: -45, Elevation: 20.

Figure 6.1. 3D plot of wells from FD1 in a field layout with satellite wells.

The program developed in this thesis favors satellite wells as these are cheapest to drill, see Chapter 5.2. It does not take the disadvantage of anchor handling between each drilling operation into concern. This takes time, thus the rental expenses of the drilling rig increases.

Figure 6.2 shows a simplified sketch of the subsea field layout. This figure was drawn on top of Figure 4.1, such that the placement of the satellite well trawl protections are correct. The field layout is sketched with 8-slots cluster manifolds. The manifolds are placed randomly. Manifolds with less or more slots can also be used.

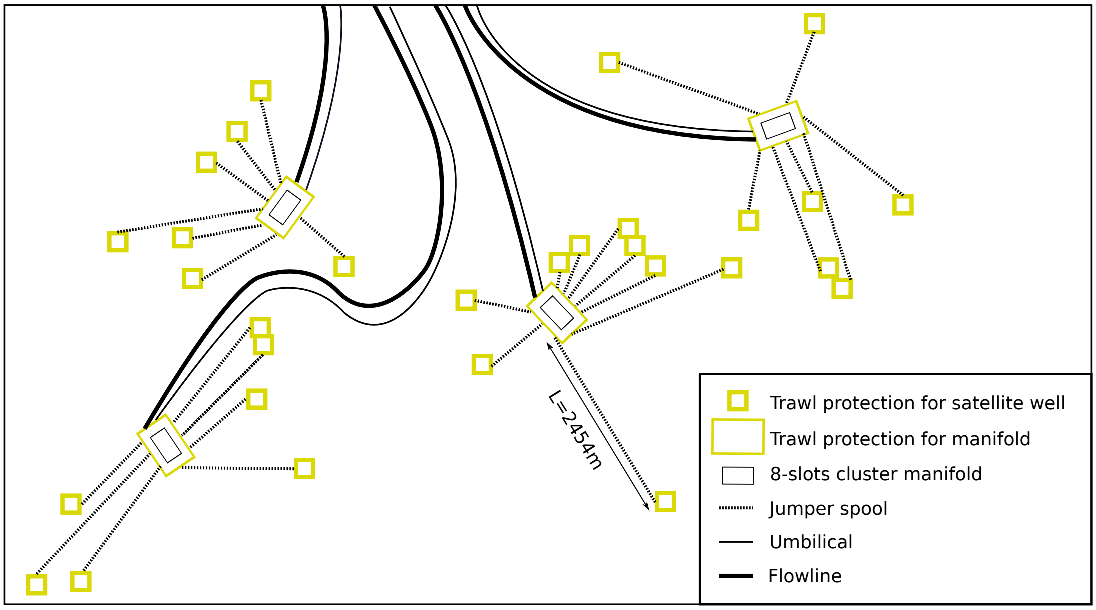


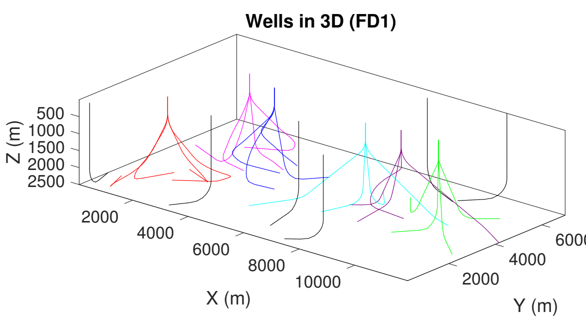
Figure 6.2. Simplified sketch of a subsea field layout with satellite wells from FD1. Details such as process facilities and umbilical termination assembly (UTA) are not included in the sketch.

Note the jumper spools that connects the satellite wells to the manifolds. There are 30 jumper spools, one for each well, and some of them are several kilometers long. Thus, the fabrication cost of these are high. In addition, the installation of the jumper spools require many lifting operations as they are installed using cranes.

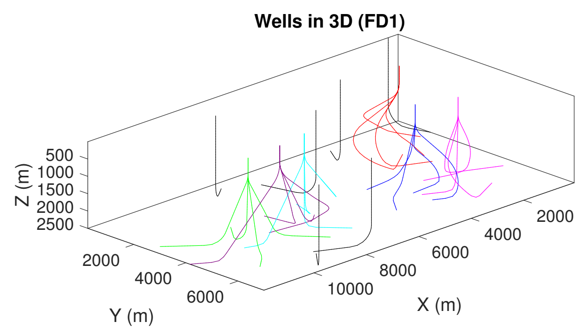
There are 34 trawl protection structures in Figure 6.2. These drive fabrication costs up, in addition to the installation costs as many lifting operations are required. On the other hand, it may not be necessary to use installation vessels, as the manifold and christmas trees (XT) may go through the moonpool on a drilling rig, depending on their size.

4-Slots Template Layout

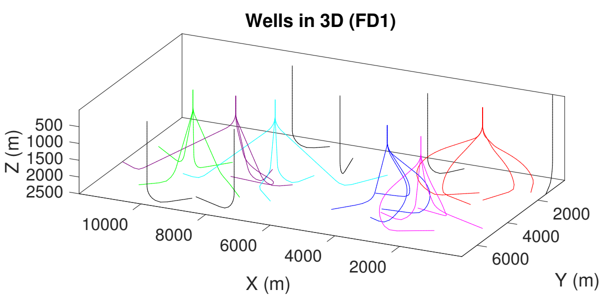
`WellboreTrajectoryTemplates` is run for the completion intervals in FD1 (where Nwt=4), using the codes in Appendix B.5 and B.6. Figure 6.3 shows the resulting wells in 3D. It is seen that the rig can stay in the same position during drilling from one ITS. On the other hand, the wells are long and deviated, thus they can cause problems during drilling.



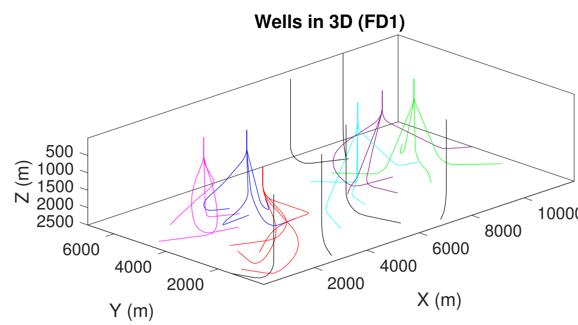
(a) Azimuth: 40, Elevation: 20.



(b) Azimuth: 135, Elevation: 20.



(c) Azimuth: -155, Elevation: 20.



(d) Azimuth: -45, Elevation: 20.

Figure 6.3. 3D plot of wells from FD1 in a field layout with 4-slots integrated template structures (ITS). The black wells are satellite wells. Each color represent one template and its associated wells.

Figure 6.4 shows a simplified sketch of the subsea field layout. This figure was drawn on top of Figure 4.3, such that the placement of the 4-slots templates and satellite well trawl protections are correct. The field layout is sketched with 4-slots ITS and 2-slots cluster manifolds. The cluster manifolds are placed randomly.

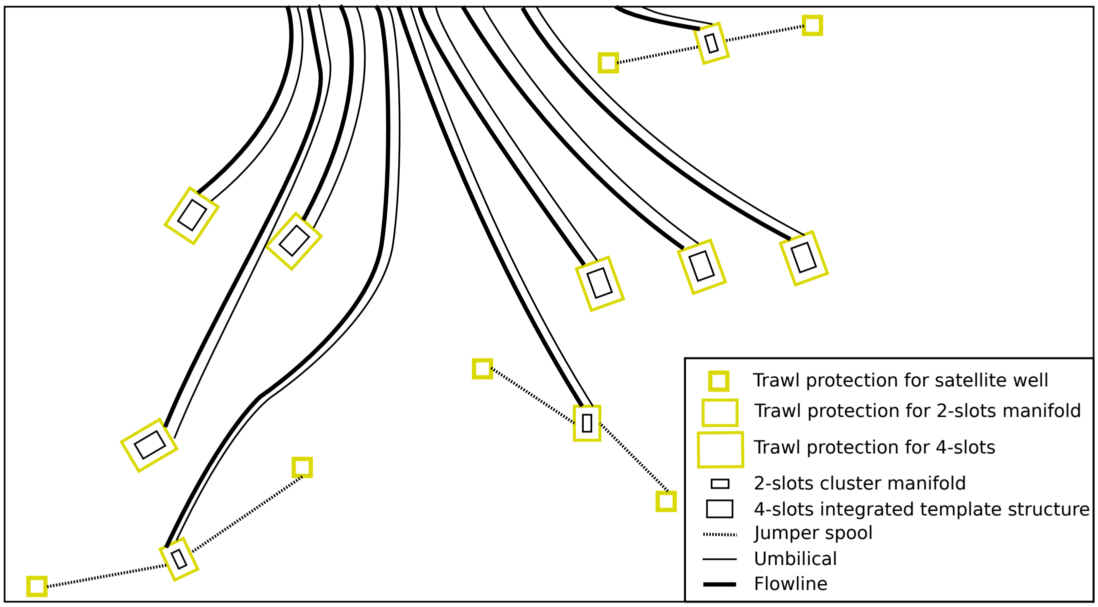


Figure 6.4. Simplified sketch of a field layout with wells from FD1 using six 4-slots templates and 6 satellite wells. Details such as process facilities and umbilical termination assembly (UTA) are not included in the sketch.

Note that the wells go directly into the 4-slots ITS. One of the advantages of the field layout with ITS is that jumper spool costs are reduced. In addition, there are only 15 trawl protection structures in Figure 6.4. Thus, a lot of fabrication costs are saved, and there are less lifting operations. On the other hand, due to the size of an ITS, an installation vessel may be required to install it.

Figure 6.4 shows that there are eight flowlines and eight umbilicals. Thus, these fabrication and installation costs are higher in the case of the 4-slots template architecture compared to the satellite wells. However, these differences in costs depend upon how many cluster manifolds there are in the satellite layout.

Appendix A

Completion Interval Coordinates

The completion interval coordinates used in this Master Thesis are based on real field data. The coordinate systems and depths are changed to make the field data unrecognizable. In addition, each completion interval is simplified by a linear relationship between the completion start coordinates and the completion end coordinates.

Some of the completion interval coordinates were negative. All coordinates are made positive as this program is developed for a UTM coordinate system. In addition, the depth axis was defined negative in the field data. The program is developed for a positive depth axis, thus the absolute value of the Z-coordinates are used.

The last adjustment made is flipping the upwards completion intervals to make them downwards. The formulas for construction of the wells are made only for wells with inclination equal to or less than 90°.

A.1 `fielddata1.mat`

Table A.1 lists the coordinates of the completion intervals in the first data set. All completion intervals are horizontal and located at a depth of 2500 meters. There are 30 completion intervals in this data set.

Table A.1

Completion interval coordinates from the first field data file used in the program.

	Completion start coordinates			Completion end coordinates		
No. (-)	X (m)	Y (m)	Z (m)	X (m)	Y (m)	Z (m)
1	286	786	2500	71	1500	2500
2	2714	2786	2500	1786	2143	2500
3	2571	2714	2500	1571	3143	2500
4	1786	4143	2500	500	4714	2500
5	2000	4214	2500	2857	4000	2500
6	2786	4429	2500	3786	4500	2500
7	3000	5071	2500	3714	4714	2500
8	3357	5429	2500	3786	5214	2500
9	3714	3357	2500	4500	3429	2500
10	6786	3571	2500	6571	2857	2500
11	7143	4071	2500	6786	3786	2500
12	7500	4500	2500	6786	5286	2500
13	7500	4071	2500	8714	3929	2500
14	9429	6571	2500	8857	5929	2500
15	9571	4429	2500	10000	4286	2500
16	11357	4500	2500	11929	4143	2500
17	10214	5214	2500	10714	5643	2500
18	7857	6286	2500	8857	5821	2500
19	8429	4500	2500	7714	5357	2500
20	9643	3571	2500	9071	2857	2500
21	7714	3786	2500	7857	2857	2500
22	7571	893	2500	7143	571	2500
23	5357	3143	2500	4714	1929	2500
24	4500	4500	2500	5000	5143	2500
25	3357	1143	2500	3000	571	2500
26	1071	1714	2500	1500	2714	2500
27	786	786	2500	500	1571	2500
28	2643	5929	2500	1643	5429	2500
29	5929	2286	2500	6214	1071	2500
30	10571	3429	2500	11571	2571	2500

A.2 `fielddata2.mat`

Table A.2 lists the coordinates of the completion intervals in the second data set. All completion intervals are inclined and located at different depths. There are 31 completion intervals in this data set.

Table A.2

Completion interval coordinates from the second field data file used in the program.

	Completion start coordinates			Completion end coordinates		
No. (-)	X (m)	Y (m)	Z (m)	X (m)	Y (m)	Z (m)
1	5565	5364	3850	5614	5335	4852
2	7578	6922	4368	8166	7252	4451
3	6939	5918	4179	6997	5920	4801
4	6630	5944	2066	6869	6075	2235
5	7188	5004	4311	6978	4553	4567
6	10500	6606	4284	10939	6969	4352
7	9586	5938	4143	10034	5781	4438
8	8661	5646	4117	8550	5311	4536
9	3595	3324	4409	3434	2902	4698
10	3918	4803	4219	3921	4524	4870
11	6960	7764	4361	6976	7785	4962
12	5957	7304	4234	5993	7303	4965
13	14262	8338	3992	14268	8316	4512
14	13577	8829	3527	13614	8788	4614
15	14984	9513	4086	15315	9737	4640
16	11993	6938	4114	11769	6560	4434
17	11972	8424	4009	12029	8426	4667
18	13520	6981	3800	13554	6945	4422
19	10463	7570	3939	10484	7545	4737
20	13283	7731	4355	13757	7882	4434
21	8287	9189	4858	7848	8910	4949
22	9330	8699	4299	9357	8534	4848
23	11658	9918	4265	11700	9881	4855
24	10356	9062	4278	10400	8892	4825
25	14367	10650	3584	14434	10507	4795
26	12852	11499	4474	12480	11367	4967
27	14568	10649	1901	14632	10176	2195
28	13977	15499	4540	13963	15852	5030
29	14467	12996	4712	14803	12691	5011
30	12748	11346	4424	12541	10953	4929
31	14652	14836	4163	14696	14954	5054

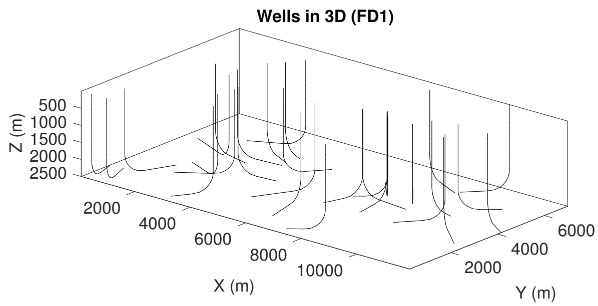
A.3 Groups of Completions and Drill Centers

A.3.1 `fielddata1.mat`

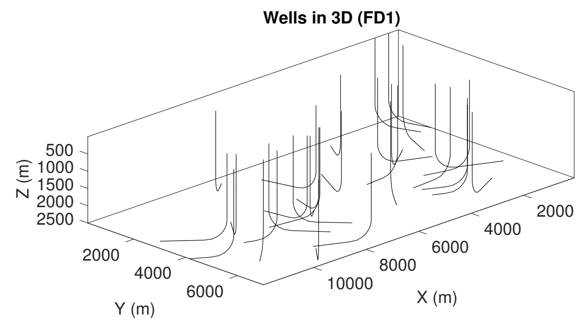
Table A.3

Satellite wells from FD1 and their resulting drill centers.

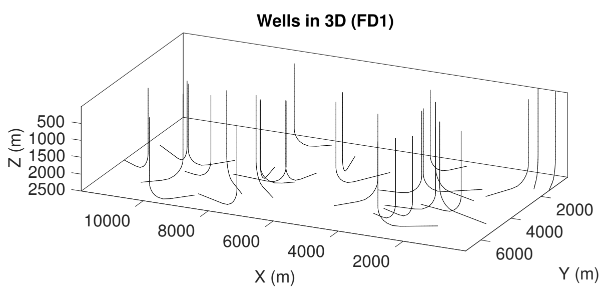
Completion Interval	Drill center coordinates	
(-)	X (m)	Y (m)
1	451	237
2	3185	3112
3	3098	2488
4	2310	3910
5	1444	4353
6	2214	4388
7	2488	5327
8	2845	5686
9	3143	3305
10	6951	4120
11	7591	4428
12	7885	4076
13	6931	4138
14	9810	6999
15	9027	4610
16	10871	4803
17	9779	4841
18	7337	6528
19	8796	4060
20	10001	4018
21	7627	4352
22	8029	1237
23	5625	3649
24	4148	4048
25	3660	1629
26	845	1187
27	982	248
28	3155	6185
29	5798	2844
30	10136	3802



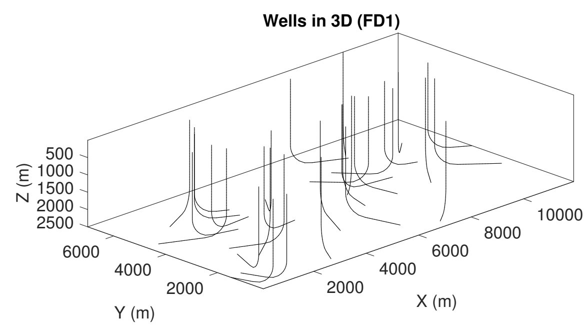
(a) Azimuth: 40, Elevation: 20.



(b) Azimuth: 135, Elevation: 20.



(c) Azimuth: -155, Elevation: 20.



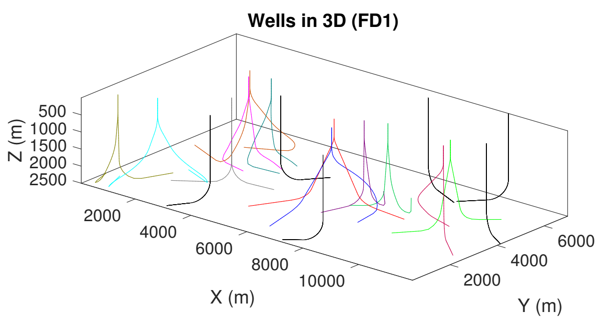
(d) Azimuth: -45, Elevation: 20.

Figure A.1. 3D plot of wells from FD1 in the field layout with satellite wells.

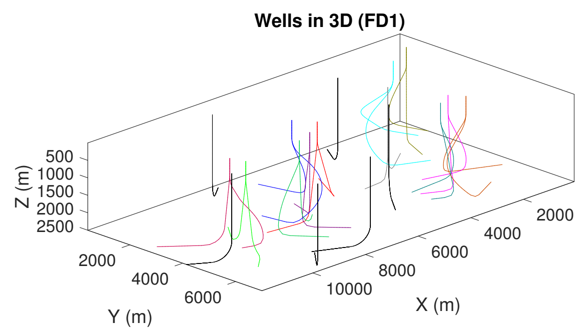
Table A.4

Groups of completions from FD1 in 2-slots templates and their resulting drill centers. The six lowermost completions are satellite wells and their associated drill centers.

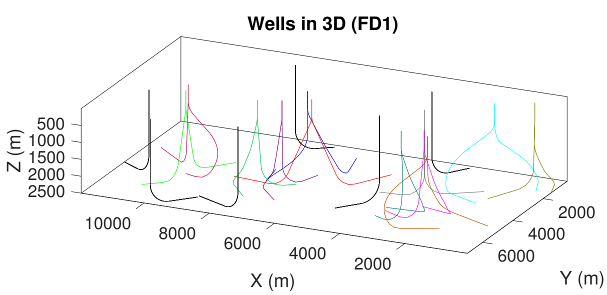
Completion Interval	Drill center coordinates	
(-)	X (m)	Y (m)
9 2	3214	3072
4 28	2215	5036
1 26	679	1250
27 3	1679	1750
6 8	3072	4929
5 7	2500	4643
12 10	7143	4036
23 13	6429	3607
29 21	6822	3036
20 17	9929	4393
19 11	7786	4286
30 15	10071	3929
22	8029	1237
25	3660	1629
18	7337	6528
14	9810	6999
24	4148	4048
16	10871	4803



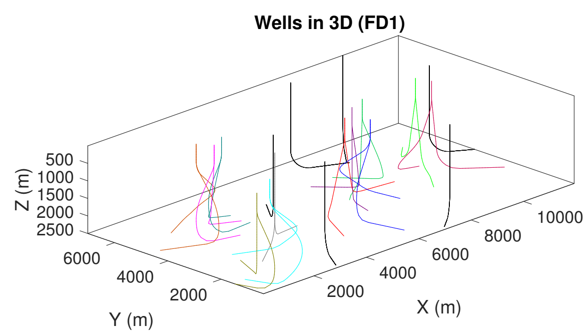
(a) Azimuth: 40, Elevation: 20.



(b) Azimuth: 135, Elevation: 20.



(c) Azimuth: -155, Elevation: 20.



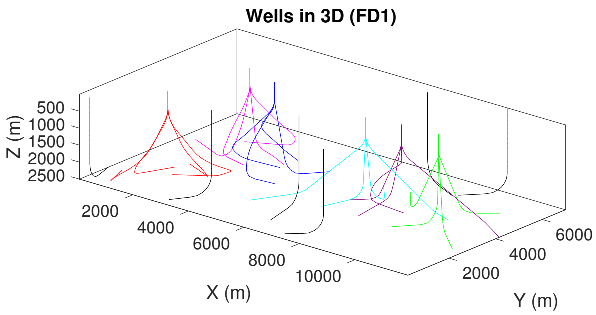
(d) Azimuth: -45, Elevation: 20.

Figure A.2. 3D plot of wells from FD1 in the field layout with 2-slots templates. Each group of wells have their own color to show that they belong together and are drilled from the same template. The satellite wells are colored black.

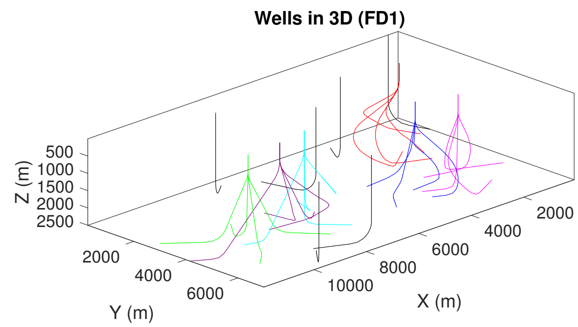
Table A.5

Groups of completions from FD1 in 4-slots templates and their resulting drill centers. The six lowermost completions are satellite wells and their associated drill centers.

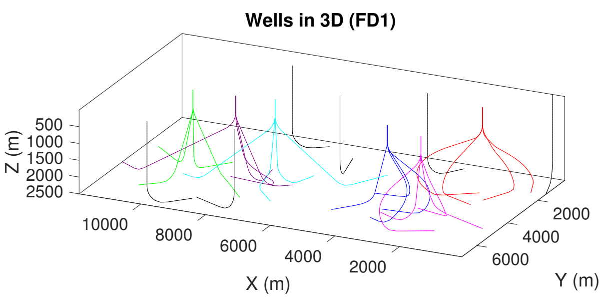
Completion Interval	Drill center coordinates	
(-)	X (m)	Y (m)
9 6 8 24	3589	4429
26 2 27 3	1786	2000
5 4 7 28	2357	4839
12 15 10 23	7304	3911
16 21 11 13	8429	4107
30 20 17 19	9714	4179
22	8029	1237
25	3660	1629
14	9810	6999
18	7337	6528
29	5798	2844
1	451	237



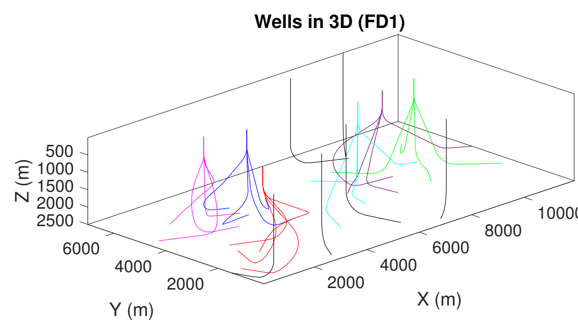
(a) Azimuth: 40, Elevation: 20.



(b) Azimuth: 135, Elevation: 20.



(c) Azimuth: -155, Elevation: 20.



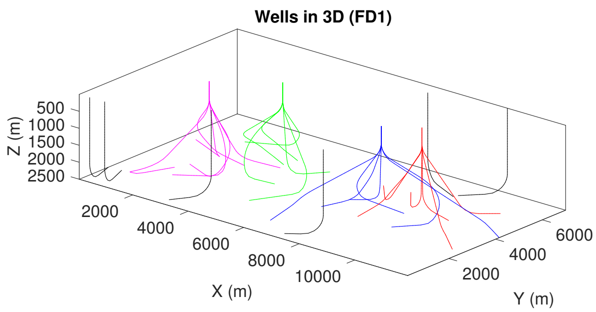
(d) Azimuth: -45, Elevation: 20.

Figure A.3. 3D plot of wells from FD1 in the field layout with 4-slots templates. Each group of wells have their own color to show that they belong together and are drilled from the same template. The satellite wells are colored black.

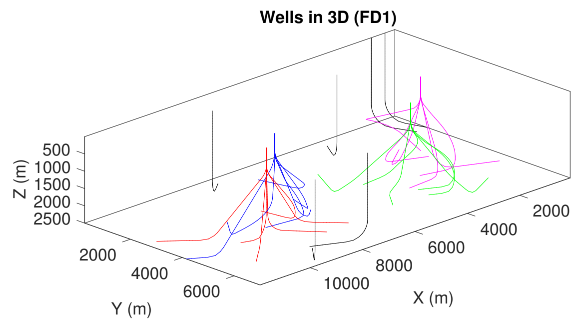
Table A.6

Groups of completions from FD1 in 6-slots templates and their resulting drill centers. The six lowermost completions are satellite wells and their associated drill centers.

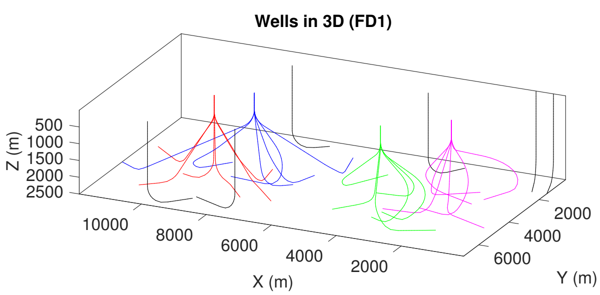
Completion Interval	Drill center coordinates	
(-)	X (m)	Y (m)
23		
9		
7	3762	4572
28		
8		
24		
2		
4		
26	2155	3333
6		
5		
3		
12		
19		
15	9000	4310
17		
30		
21		
16		
29		
13	8060	3678
11		
20		
10		
22	8029	1237
1	451	237
25	3660	1629
14	9810	6999
18	7337	6528
27	982	248



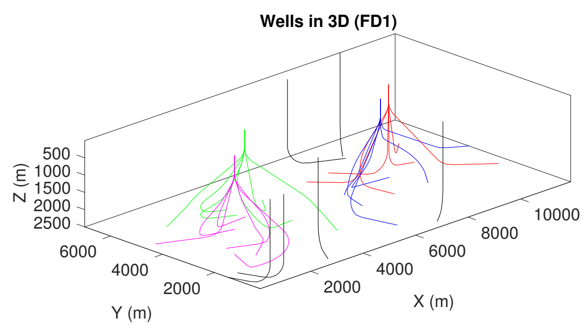
(a) Azimuth: 40, Elevation: 20.



(b) Azimuth: 135, Elevation: 20.



(c) Azimuth: -155, Elevation: 20.



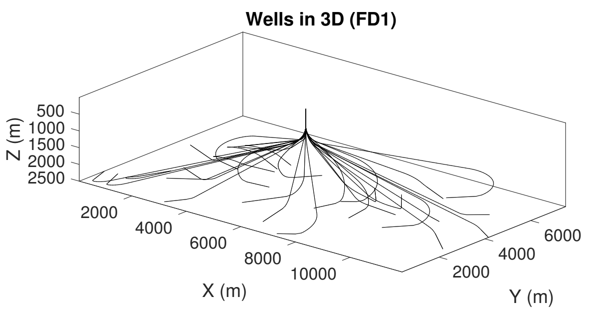
(d) Azimuth: -45, Elevation: 20.

Figure A.4. 3D plot of wells from FD1 in the field layout with 6-slots templates. Each group of wells have their own color to show that they belong together and are drilled from the same template. The satellite wells are colored black.

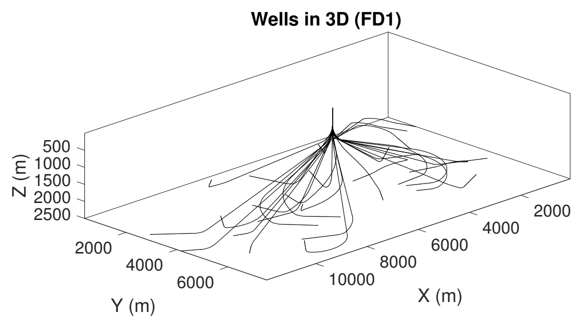
Table A.7

The completion intervals from FD1 and the coordinates of their common drill center.

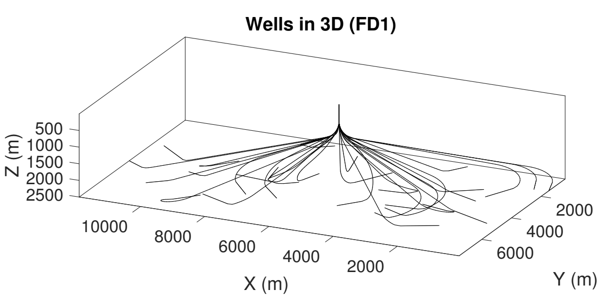
Completion Interval	Drill center coordinates	
(-)	X (m)	Y (m)
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15	5571	3727
16		
17		
18		
19		
20		
21		
22		
23		
24		
25		
26		
27		
28		
29		
30		



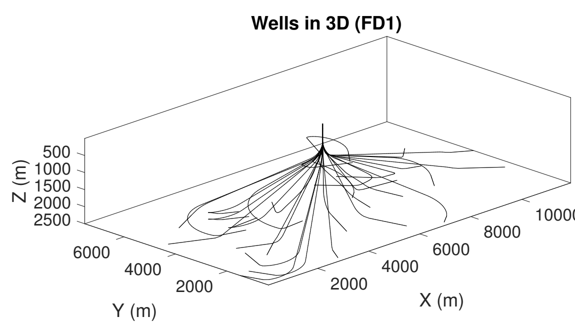
(a) Azimuth: 40, Elevation: 20.



(b) Azimuth: 135, Elevation: 20.



(c) Azimuth: -155, Elevation: 20.



(d) Azimuth: -45, Elevation: 20.

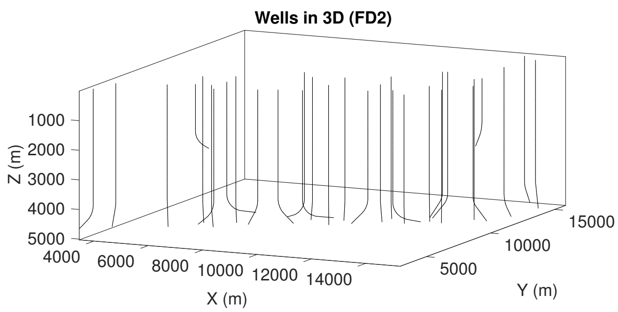
Figure A.5. 3D plot of wells from FD1 in the field layout with Subsea On a Stick.

A.3.2 `fielddata2.mat`

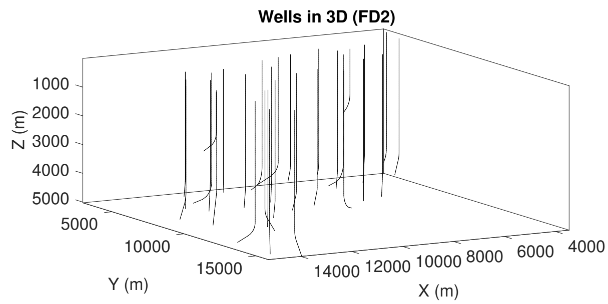
Table A.8

Satellite wells from FD2 and their resulting drill centers.

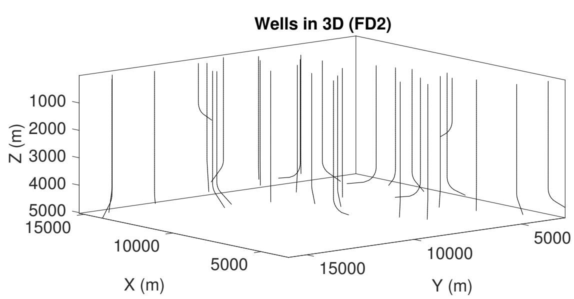
Completion Interval	Drill center coordinates	
(-)	X (m)	Y (m)
1	5564	5365
2	7139	6675
3	6936	5918
4	6392	5814
5	7319	5284
6	10111	6285
7	9331	6027
8	8703	5773
9	3690	3571
10	3917	4849
11	6960	7763
12	5957	7304
13	14262	8338
14	13576	8829
15	14895	9452
16	12114	7140
17	11970	8424
18	13519	6982
19	10463	7570
20	12823	7585
21	8688	9444
22	9326	8724
23	11656	9919
24	10349	9088
25	14365	10654
26	12971	11541
27	14531	10919
28	13981	15391
29	14276	13169
30	12814	11473
31	14650	14831



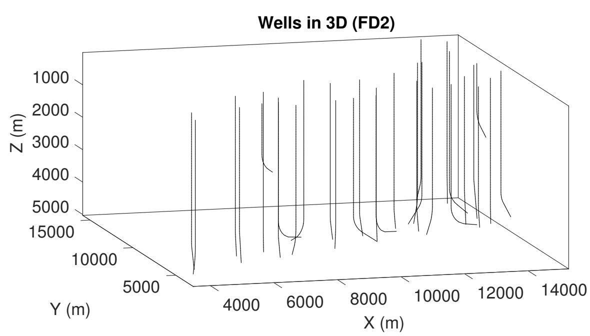
(a) Azimuth: 25, Elevation: 10.



(b) Azimuth: 150, Elevation: 10.



(c) Azimuth: -130, Elevation: 10.



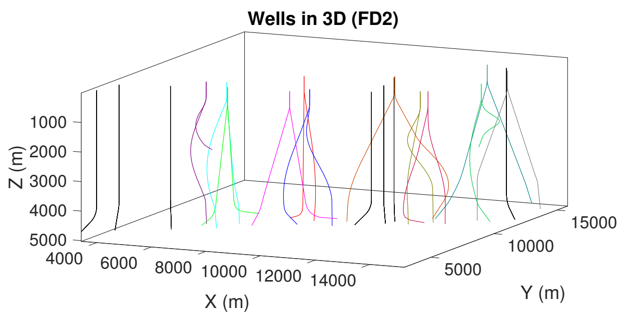
(d) Azimuth: -15, Elevation: 10.

Figure A.6. 3D plot of wells from FD2 in the field layout with satellite wells.

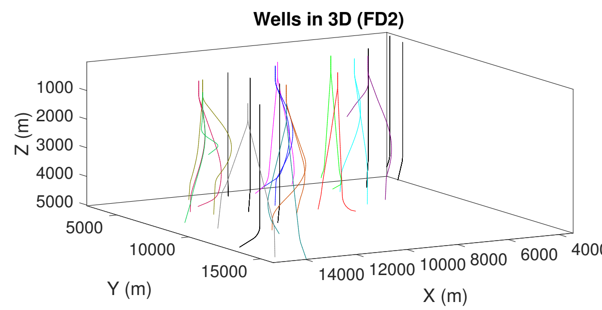
Table A.9

Groups of completions from FD2 in 2-slots templates and their resulting drill centers. The seven lowermost completions are satellite wells and their associated drill centers.

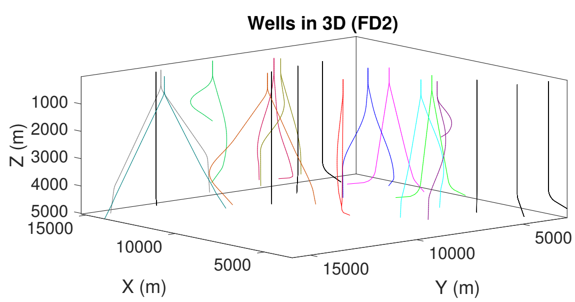
Completion Interval	Drill center coordinates	
(-)	X (m)	Y (m)
22 21	8809	8944
3 11	6949	6841
5 2	7383	5963
6 8	9580	6126
7 19	10025	6754
12 4	6294	6624
18 14	13548	7905
26 28	13415	13499
24 30	11552	10204
13 20	13773	8035
27 15	14776	10081
31 25	14509	12743
29	14276	13169
23	11656	9919
9	3690	3571
10	3917	4849
16	12144	7140
17	11970	8424
1	5564	5365



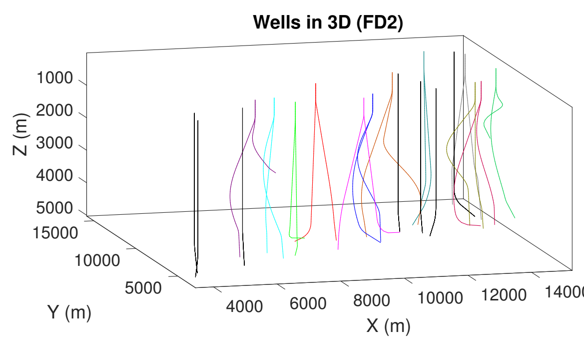
(a) Azimuth: 25, Elevation: 10.



(b) Azimuth: 150, Elevation: 10.



(c) Azimuth: -130, Elevation: 10.



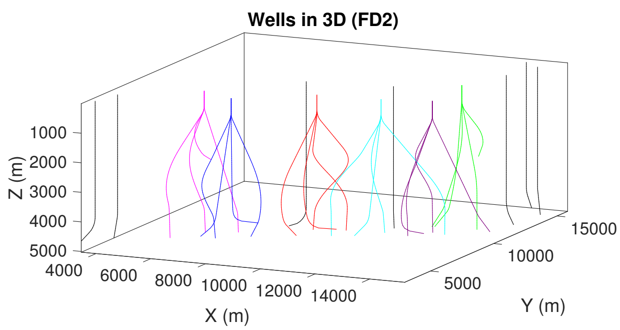
(d) Azimuth: -15, Elevation: 10.

Figure A.7. 3D plot of wells from FD2 in the field layout with 2-slots templates. Each group of wells have their own color to show that they belong together and are drilled from the same template. The satellite wells are colored black.

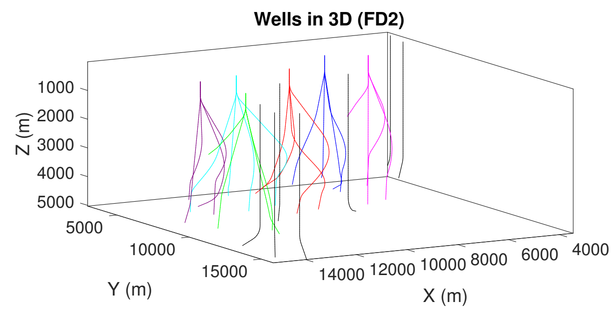
Table A.10

Groups of completions from FD2 in 4-slots templates and their resulting drill centers. The seven lowermost completions are satellite wells and their associated drill centers.

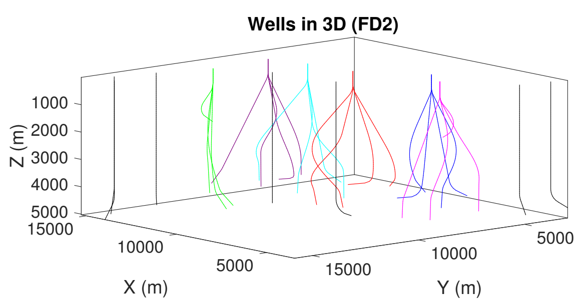
Completion Interval	Drill center coordinates	
(-)	X (m)	Y (m)
24, 22, 7, 6	9943	7576
8, 3, 5, 2	7591	5872
1, 12, 4, 11	6278	6594
19, 16, 13, 17	12173	7817
15, 18, 14, 20	13841	8263
26, 25, 27, 30	13634	11036
9	3690	3571
28	13981	15391
31	14650	14831
29	14276	13169
10	3917	4849
21	8688	9444
23	11656	9919



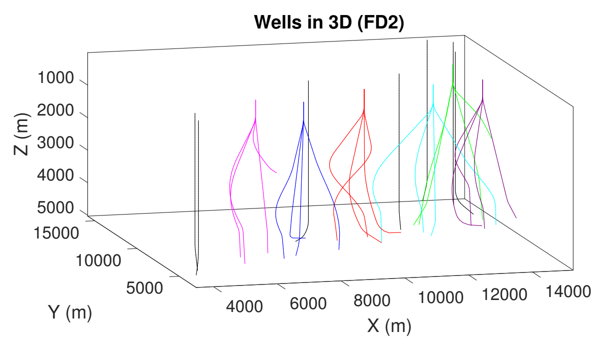
(a) Azimuth: 25, Elevation: 10.



(b) Azimuth: 150, Elevation: 10.



(c) Azimuth: -130, Elevation: 10.



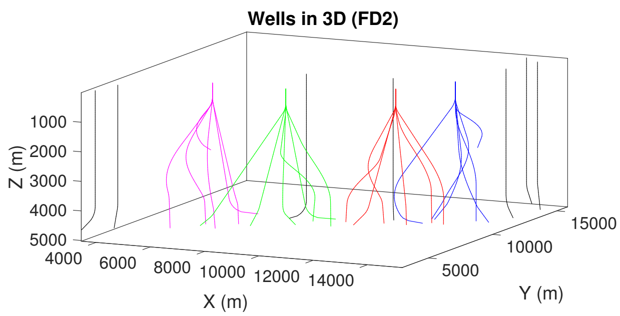
(d) Azimuth: -15, Elevation: 10.

Figure A.8. 3D plot of wells from FD2 in the field layout with 4-slots templates. Each group of wells have their own color to show that they belong together and are drilled from the same template. The satellite wells are colored black.

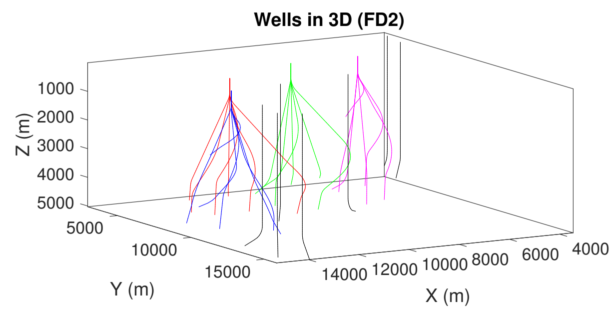
Table A.11

Groups of completions from FD2 in 6-slots templates and their resulting drill centers. The seven lowermost completions are satellite wells and their associated drill centers.

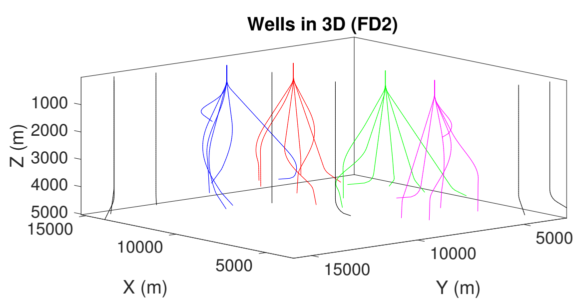
Completion Interval	Drill center coordinates	
(-)	X (m)	Y (m)
19, 22, 5, 8, 7, 6	9288	6577
11, 3, 1, 4, 2, 12	6605	6536
16, 18, 14, 13, 17, 24	12613	8095
20, 26, 15, 30, 27, 25	13800	10231
28	13981	15391
9	3690	3571
31	14650	14831
10	3917	4849
29	14276	13169
21	8688	9444
23	11656	9919



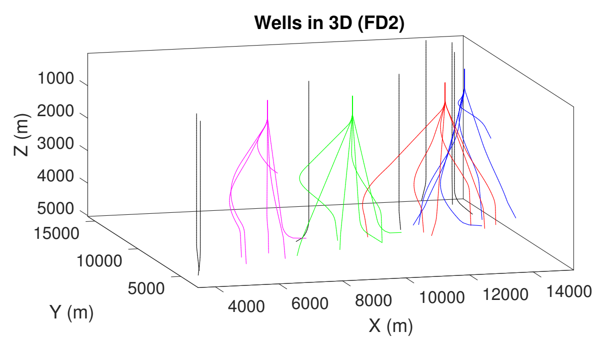
(a) Azimuth: 25, Elevation: 10.



(b) Azimuth: 150, Elevation: 10.



(c) Azimuth: -130, Elevation: 10.



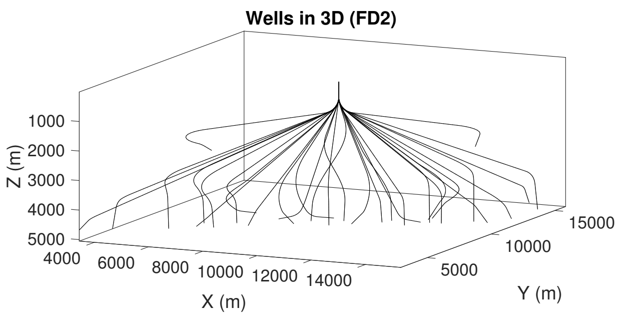
(d) Azimuth: -15, Elevation: 10.

Figure A.9. 3D plot of wells from FD2 in the field layout with 6-slots templates. Each group of wells have their own color to show that they belong together and are drilled from the same template. The satellite wells are colored black.

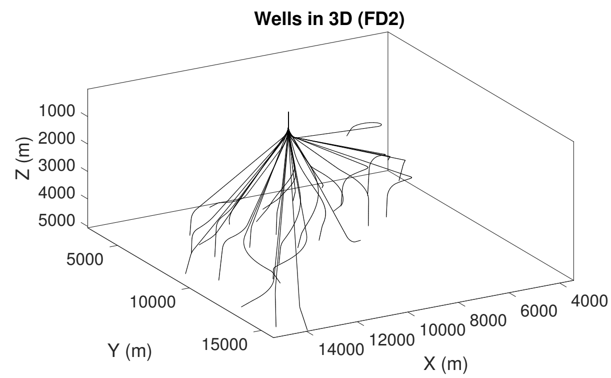
Table A.12

The completion intervals from FD2 and the coordinates of their common drill center.

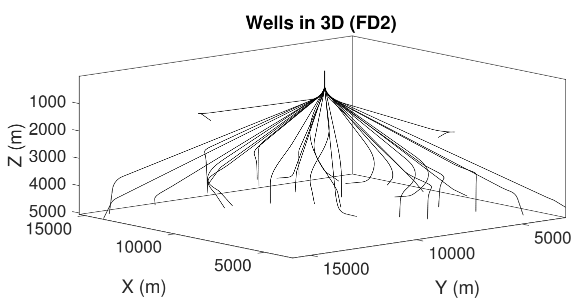
Completion Interval	Drill center coordinates	
(-)	X (m)	Y (m)
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
16	10464	8361
17		
18		
19		
20		
21		
22		
23		
24		
25		
26		
27		
28		
29		
30		
31		



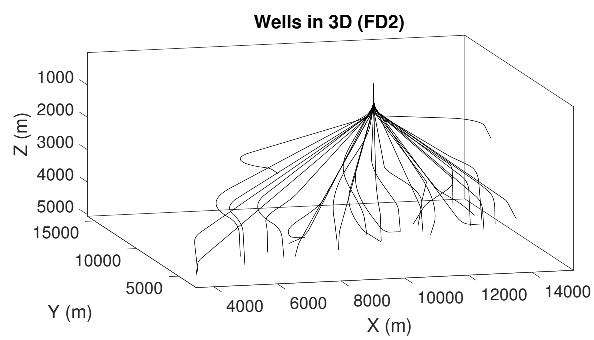
(a) Azimuth: 25, Elevation: 10.



(b) Azimuth: 150, Elevation: 10.



(c) Azimuth: -130, Elevation: 10.



(d) Azimuth: -15, Elevation: 10.

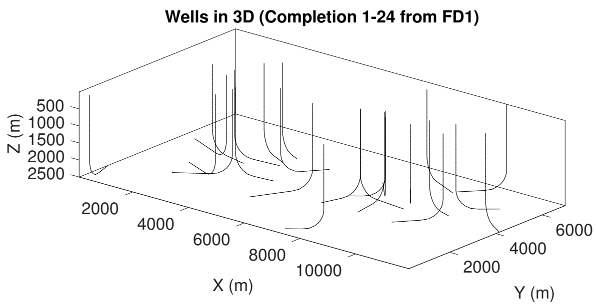
Figure A.10. 3D plot of wells from FD2 in the field layout with Subsea On a Stick.

A.3.3 Completion 1-24 from `fielddata1.mat`

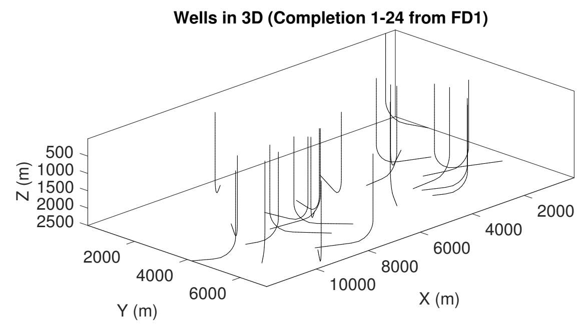
Table A.13

Satellite wells from the first 24 completions in FD1 and their resulting drill centers.

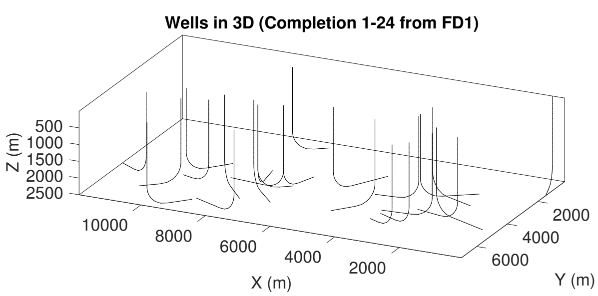
Completion Interval	Drill center coordinates	
(-)	X (m)	Y (m)
1	451	237
2	3185	3112
3	3098	2488
4	2310	3910
5	1444	4353
6	2214	4388
7	2488	5327
8	2845	5686
9	3143	3305
10	6951	4120
11	7591	4428
12	7885	4076
13	6931	4138
14	9810	6999
15	9027	4610
16	10871	4803
17	9779	4841
18	7337	6528
19	8796	4060
20	10001	4018
21	7627	4352
22	8029	1237
23	5625	3649
24	4148	4048



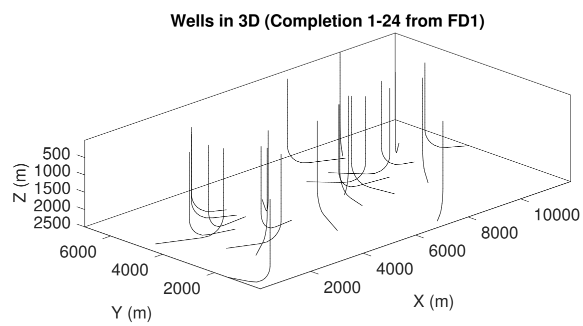
(a) Azimuth: 40, Elevation: 20.



(b) Azimuth: 135, Elevation: 20.



(c) Azimuth: -155, Elevation: 20.



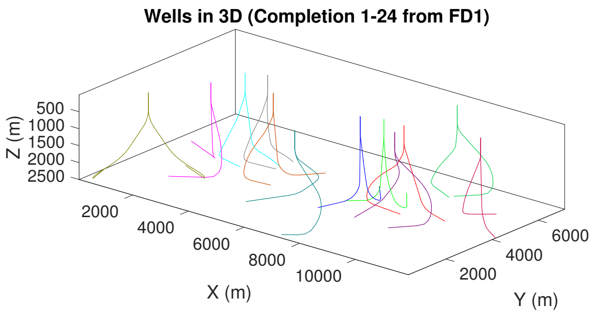
(d) Azimuth: -45, Elevation: 20.

Figure A.11. 3D plot of the first 24 wells from FD1 in the field layout with satellite wells.

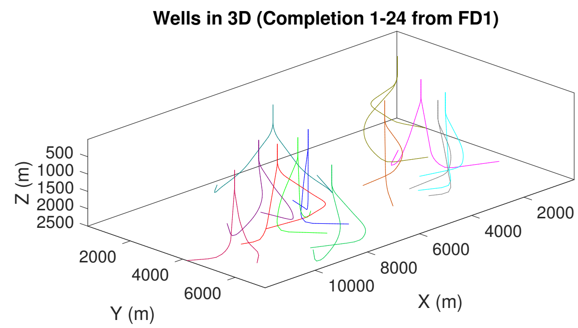
Table A.14

Groups of completions from the first 24 completions in FD1 in 2-slots templates and their resulting drill centers.

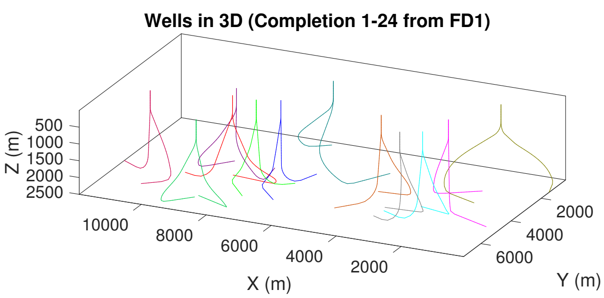
Completion Interval	Drill center coordinates	
(-)	X (m)	Y (m)
8 6	3072	4929
22 23	6464	2018
24 9	4107	3929
5 7	2500	4643
4 2	2250	3465
3 1	1429	1750
20 21	8679	3679
17 16	10786	4857
10 12	7143	4036
13 15	8536	4250
14 18	8643	6429
19 11	7786	4286



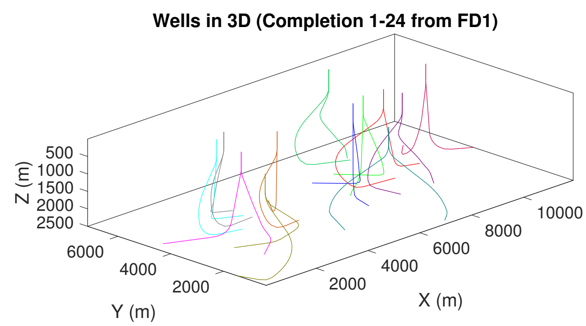
(a) Azimuth: 40, Elevation: 20.



(b) Azimuth: 135, Elevation: 20.



(c) Azimuth: -155, Elevation: 20.



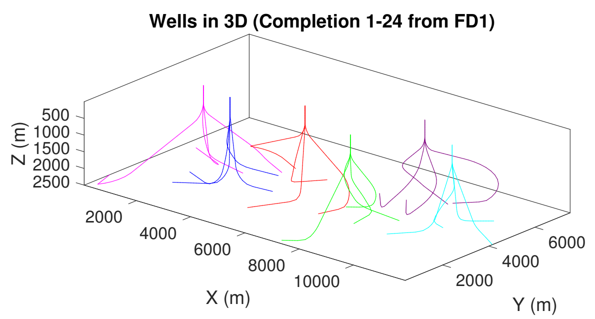
(d) Azimuth: -45, Elevation: 20.

Figure A.12. 3D plot of the first 24 wells from FD1 in the field layout with 2-slots templates. Each group of wells have their own color to show that they belong together and are drilled from the same template. The satellite wells are colored black.

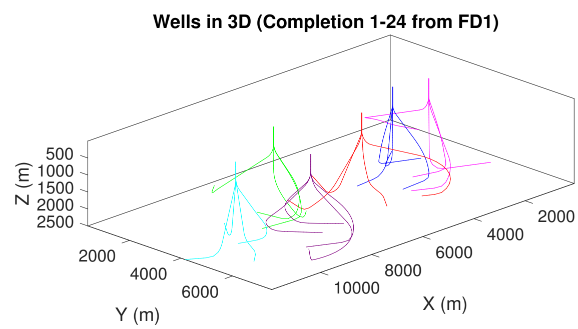
Table A.15

Groups of completions from the first 24 completions in FD1 in 4-slots templates and their resulting drill centers.

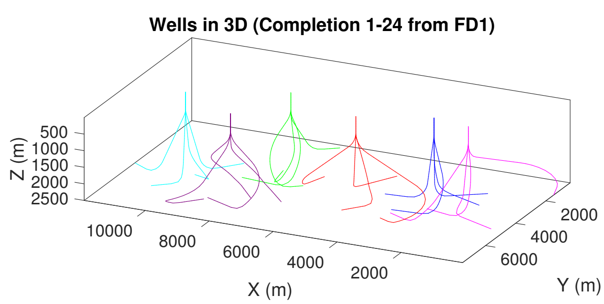
Completion Interval	Drill center coordinates	
(-)	X (m)	Y (m)
10 24 8 23	5000	4161
1 7 5 4	1768	3554
6 3 9 2	2946	3322
22 13 21 11	7482	3205
16 20 17 15	10196	4429
18 19 14 12	8304	5464



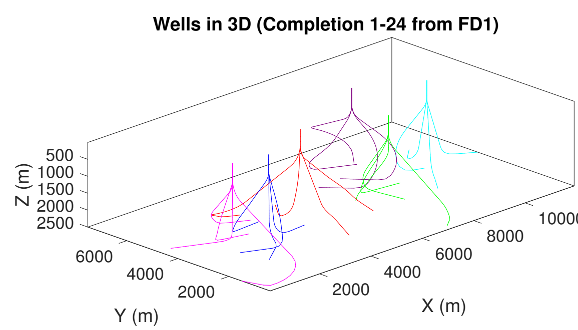
(a) Azimuth: 40, Elevation: 20.



(b) Azimuth: 135, Elevation: 20.



(c) Azimuth: -155, Elevation: 20.



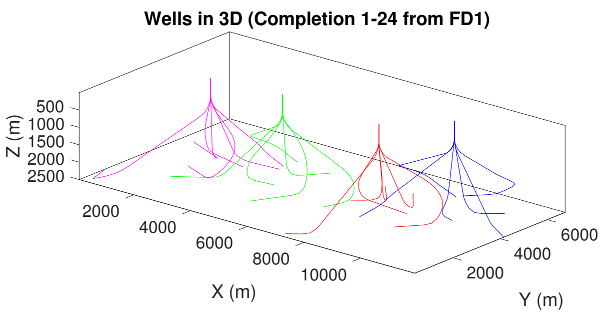
(d) Azimuth: -45, Elevation: 20.

Figure A.13. 3D plot of the first 24 wells from FD1 in the field layout with 4-slots templates. Each group of wells have their own color to show that they belong together and are drilled from the same template. The satellite wells are colored black.

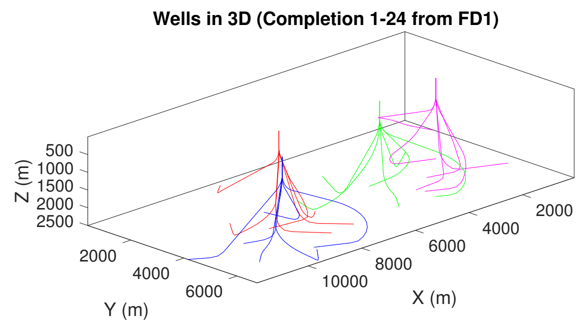
Table A.16

Groups of completions from the first 24 completions in FD1 in 6-slots templates and their resulting drill centers.

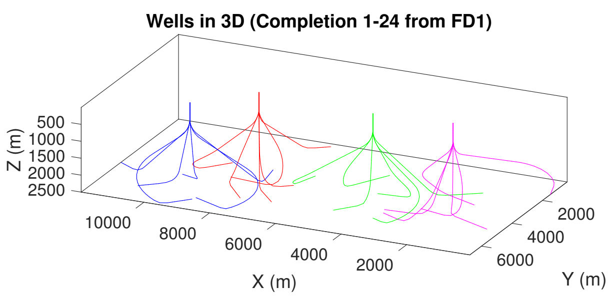
Completion Interval	Drill center coordinates	
(-)	X (m)	Y (m)
10		
24		
8	4405	3798
2		
9		
23		
5		
6		
7	2072	3560
1		
4		
3		
22		
20		
19	7964	3601
12		
13		
11		
17		
14		
15	9357	5131
16		
18		
21		



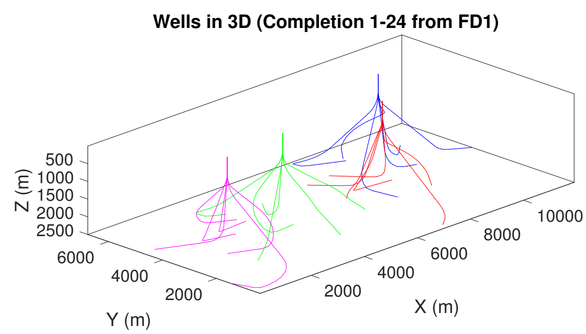
(a) Azimuth: 40, Elevation: 20.



(b) Azimuth: 135, Elevation: 20.



(c) Azimuth: -155, Elevation: 20.



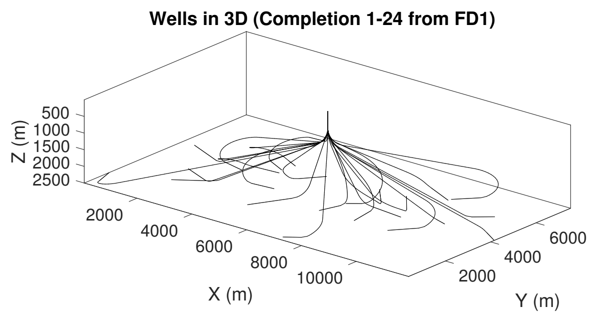
(d) Azimuth: -45, Elevation: 20.

Figure A.14. 3D plot of the first 24 wells from FD1 in the field layout with 6-slots templates. Each group of wells have their own color to show that they belong together and are drilled from the same template. The satellite wells are colored black.

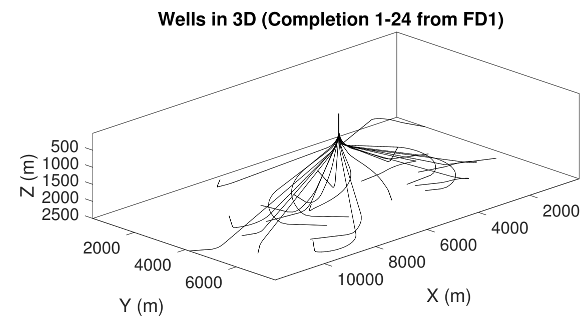
Table A.17

The first 24 completion intervals from FD1 and the coordinates of their common drill center.

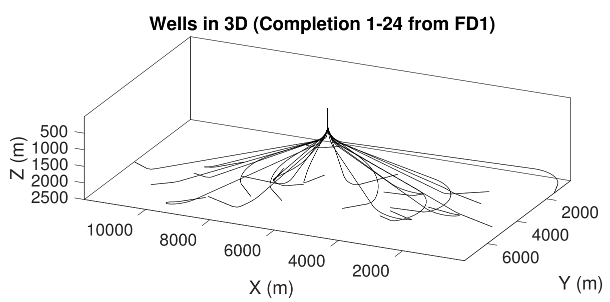
Completion Interval	Drill center coordinates	
(-)	X (m)	Y (m)
1	5949	4022
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
16		
17		
18		
19		
20		
21		
22		
23		
24		



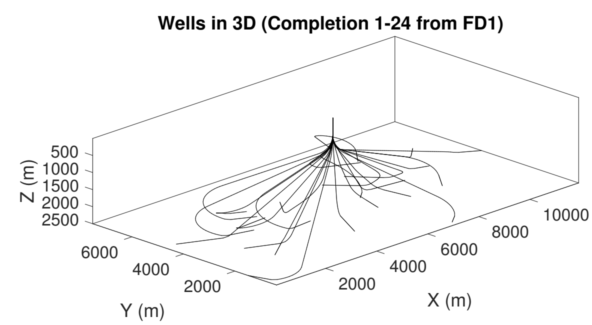
(a) Azimuth: 40, Elevation: 20.



(b) Azimuth: 135, Elevation: 20.



(c) Azimuth: -155, Elevation: 20.



(d) Azimuth: -45, Elevation: 20.

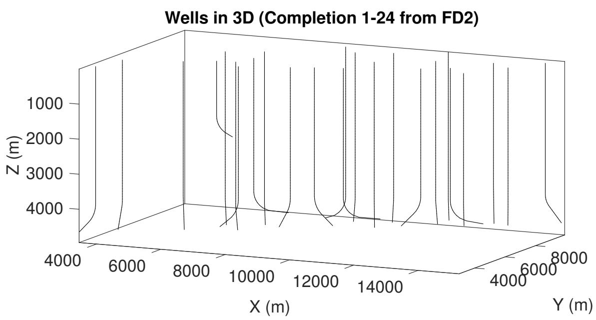
Figure A.15. 3D plot of the first 24 wells from FD1 in the field layout with Subsea On a Stick.

A.3.4 Completion 1-24 from `fielddata2.mat`

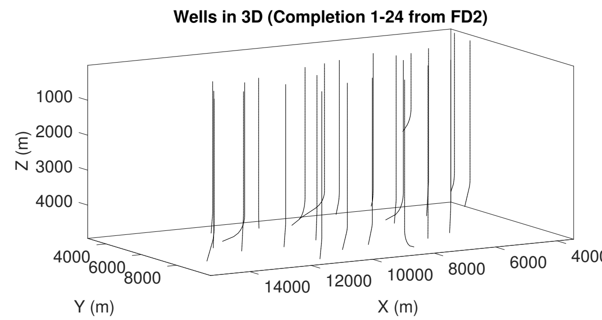
Table A.18

Satellite wells from the first 24 completions in FD2 and their resulting drill centers.

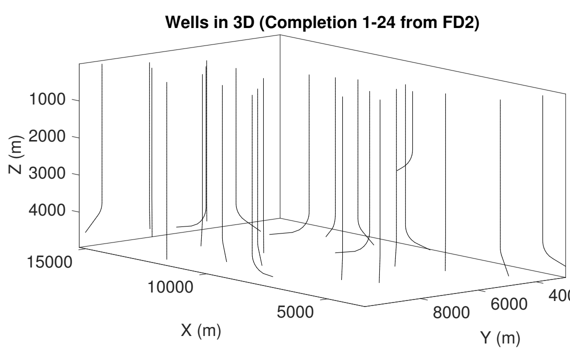
Completion Interval	Drill center coordinates	
(-)	X (m)	Y (m)
1	5564	5365
2	7139	6675
3	6936	5918
4	6392	5814
5	7319	5284
6	10111	6285
7	9331	6027
8	8703	5773
9	3690	3571
10	3917	4849
11	6960	7763
12	5957	7304
13	14262	8338
14	13576	8829
15	14895	9452
16	12144	7140
17	11970	8424
18	13519	6982
19	10463	7570
20	12823	7585
21	8688	9444
22	9326	8724
23	11656	9919
24	10349	9088



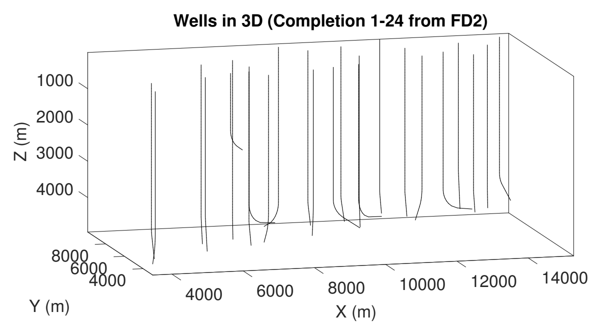
(a) Azimuth: 25, Elevation: 10.



(b) Azimuth: 150, Elevation: 10.



(c) Azimuth: -130, Elevation: 10.



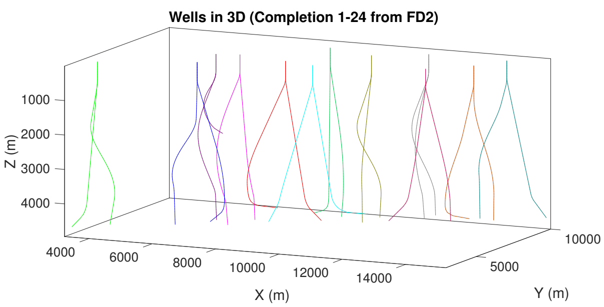
(d) Azimuth: -15, Elevation: 10.

Figure A.16. 3D plot of the first 24 wells from FD2 in the field layout with satellite wells.

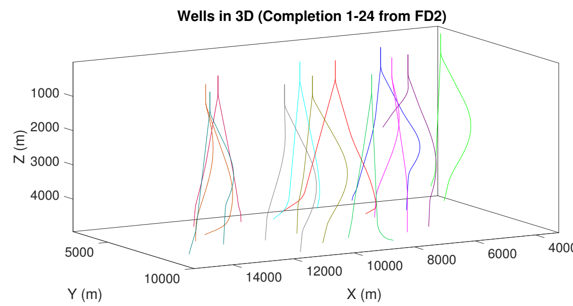
Table A.19

Groups of completions from the first 24 completions in FD2 in 2-slots templates and their resulting drill centers.

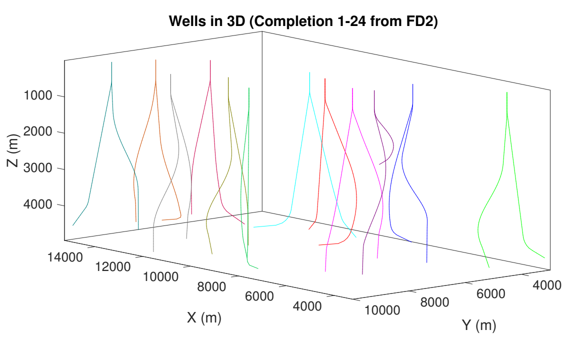
Completion Interval	Drill center coordinates	
(-)	X (m)	Y (m)
2 7	8582	6430
10 9	3757	4063
11 3	6949	6841
5 1	6376	5184
8 6	9580	6126
12 4	6294	6624
19 24	10410	8316
14 15	14280	9171
23 17	11815	9171
21 22	8809	8944
18 16	12757	6959
20 13	13773	8035



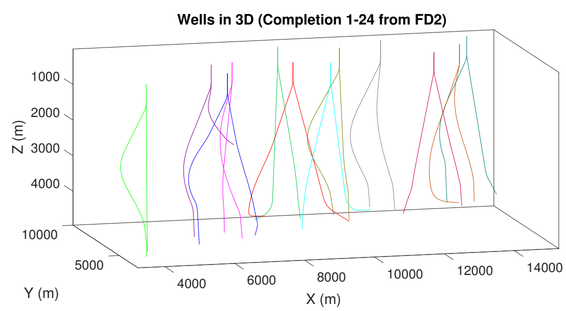
(a) Azimuth: 25, Elevation: 10.



(b) Azimuth: 150, Elevation: 10.



(c) Azimuth: -130, Elevation: 10.



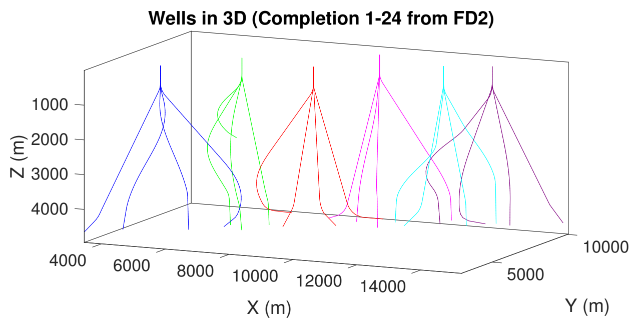
(d) Azimuth: -15, Elevation: 10.

Figure A.17. 3D plot of the first 24 wells from FD2 in the field layout with 2-slots templates. Each group of wells have their own color to show that they belong together and are drilled from the same template. The satellite wells are colored black.

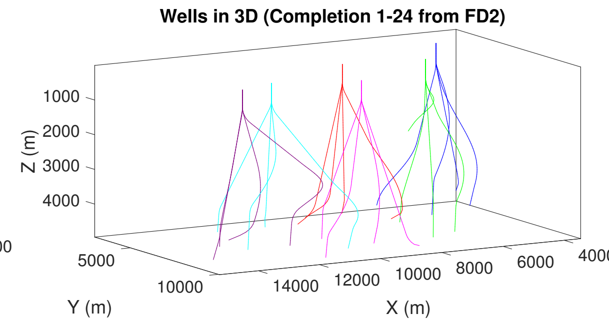
Table A.20

Groups of completions from the first 24 completions in FD2 in 4-slots templates and their resulting drill centers.

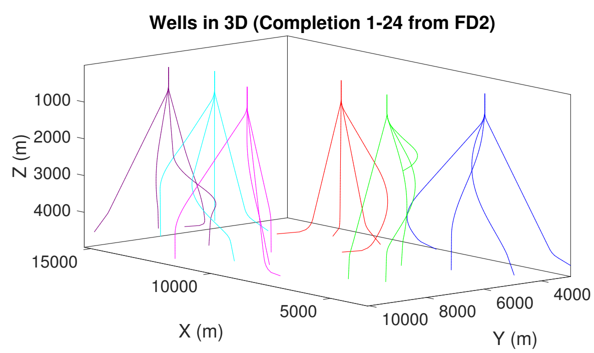
Completion Interval	Drill center coordinates	
(-)	X (m)	Y (m)
2 8 7 6	9081	6278
3 12 4 11	6621	6732
10 1 9 5	5067	4624
19 23 22 21	9935	8844
18 14 16 24	12361	7952
17 20 13 15	13625	8501



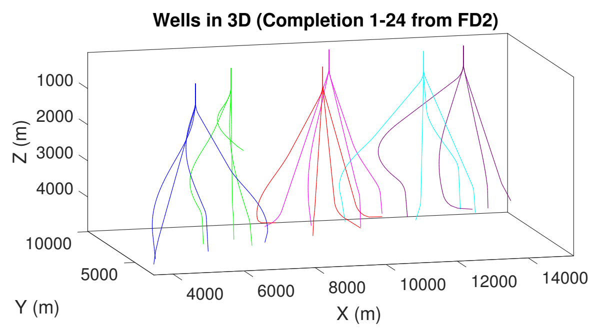
(a) Azimuth: 25, Elevation: 10.



(b) Azimuth: 150, Elevation: 10.



(c) Azimuth: -130, Elevation: 10.



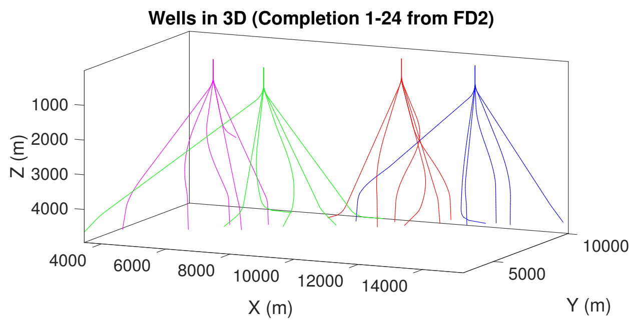
(d) Azimuth: -15, Elevation: 10.

Figure A.18. 3D plot of the first 24 wells from FD2 in the field layout with 4-slots templates. Each group of wells have their own color to show that they belong together and are drilled from the same template. The satellite wells are colored black.

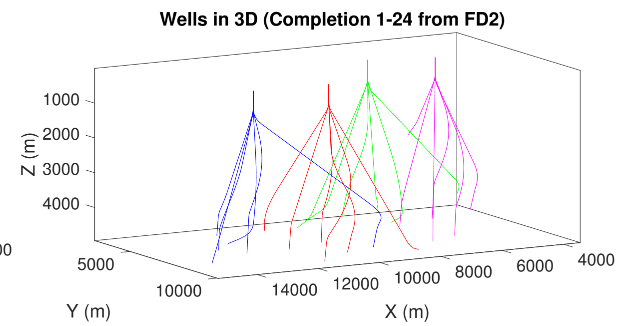
Table A.21

Groups of completions from the first 24 completions in FD2 in 6-slots templates and their resulting drill centers.

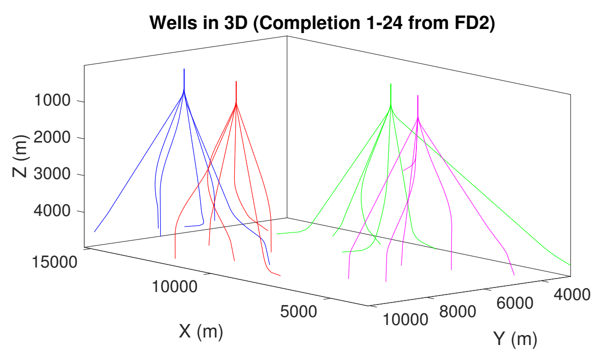
Completion Interval	Drill center coordinates	
(-)	X (m)	Y (m)
2 9 5 8 7 6	7851	5573
10 12 11 1 4 3	5995	6183
19 16 17 23 24 21	10788	8517
18 14 13 22 20 15	13159	8348



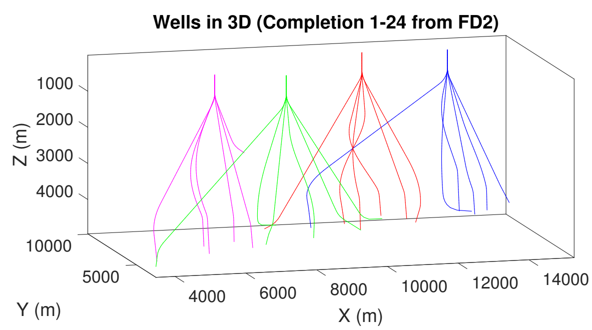
(a) Azimuth: 25, Elevation: 10.



(b) Azimuth: 150, Elevation: 10.



(c) Azimuth: -130, Elevation: 10.



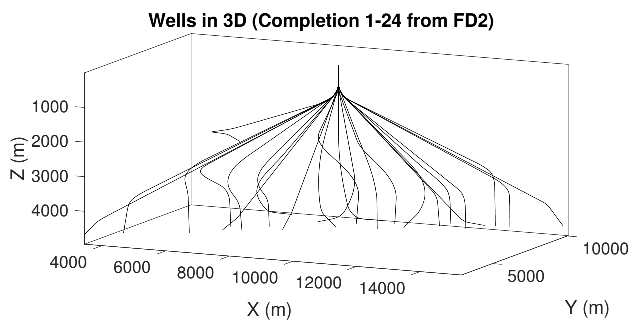
(d) Azimuth: -15, Elevation: 10.

Figure A.19. 3D plot of the first 24 wells from FD2 in the field layout with 6-slots templates. Each group of wells have their own color to show that they belong together and are drilled from the same template. The satellite wells are colored black.

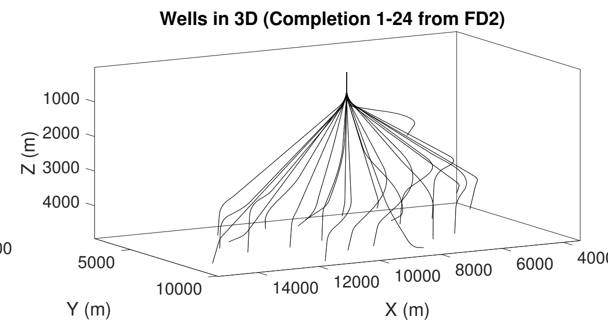
Table A.22

The first 24 completion intervals from FD2 and the coordinates of their common drill center.

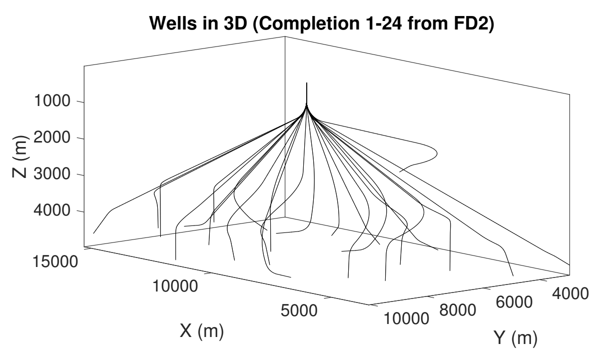
Completion Interval	Drill center coordinates	
(-)	X (m)	Y (m)
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12	9449	7155
13		
14		
15		
16		
17		
18		
19		
20		
21		
22		
23		
24		



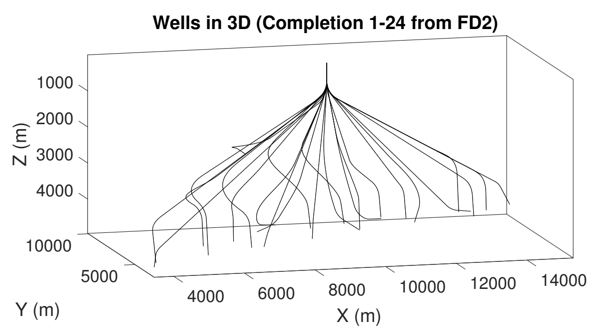
(a) Azimuth: 25, Elevation: 10.



(b) Azimuth: 150, Elevation: 10.



(c) Azimuth: -130, Elevation: 10.



(d) Azimuth: -15, Elevation: 10.

Figure A.20. 3D plot of the first 24 wells from FD2 in the field layout with Subsea On a Stick.

Appendix B

MATLAB

B.1 Grid Method

```
%load completion intervals
C=load('fielddata1.mat');
C=cell2mat(struct2cell(C));

%number of wells per template
N_wt=4;

%minimum and maximum values of grid
x_min=floor(min(C(:,1)))-100;
y_min=floor(min(C(:,2)))-100;
x_max=ceil(max(C(:,1)))+100;
y_max=ceil(max(C(:,2)))+100;

%create grid with quadrant size mxn
g4=0;
f=10;
g=0;
for k=2:10
    for l=1:10
      n=(x_max-x_min)/k;
      m=(y_max-y_min)/l;
      x_range=x_min:n:x_max;
      y_range=y_min:m:y_max;

      %count number of completions inside each quadrant
      Nc=zeros(l,k);
      for i=1:l
          range_y=[y_max-m*i  y_max-m*(i-1)];
          j=1;
          while j<=k
              range_x=[x_min+n*(j-1)  x_min+n*j];
              Nc(i,j)=sum((C(:,1)-range_x(1)>=0) & (C(:,1)-range_x(2)<=0) & ...
                      (C(:,2)-range_y(1)>=0) & (C(:,2)-range_y(2)<=0));
              j=j+1;
          end
      end

```

```
    %find the optimal grid
    d=sum(sum(Nc==N_wt));
        if d>g4 || d==g4 && abs(k-l)<abs(f-g)
            g4=d;
            f=k;
            g=l;
            x_range_opt=x_range;
            y_range_opt=y_range;
            Nc_opt=Nc;
        end
      end
end

%locate the quadrant min and max values for each completion
M_loc=zeros(size(C,1),4);
for i=1:size(C,1)
    x_grid=find(x_range_opt<=C(i,1),1,'last');
    y_grid=find(y_range_opt<=C(i,2),1,'last');
    M_loc(i,1:2)=[x_range_opt(x_grid)  x_range_opt(x_grid+1)];
    M_loc(i,3:4)=[y_range_opt(y_grid)  y_range_opt(y_grid+1)];
end

%plot completion intervals with grid on
for i=1:size(C,1)
    plot([C(i,1) C(i,4)], [C(i,2) C(i,5)],'b')
    hold on
    scatter(C(i,1), C(i,2),'r')
    %scatter(C(i,4), C(i,5),'b')
    hold on
    grid on
    title('Subsea field with grid')
    xlabel('X (m)')
    ylabel('Y (m)')
    xlim([x_range_opt(1) x_range_opt(size(x_range_opt,2))])
    xticks(x_range_opt)
    xtickformat('%,.0f')
    ylim([y_range_opt(1) y_range_opt(size(y_range_opt,2))])
    yticks(y_range_opt)
    ytickformat('%,.0f')

    set(findall(gcf,'-property','FontSize'),'FontSize',15)
    legend({'Completion interval','Completion start'},'FontSize',13)

end
```

B.2 Traveling Salesman Method

```
function subTours = detectSubtours(x,idxs)
% Returns a cell array of subtours. The first subtour is the first row of x, etc.

%   Copyright 2014 The MathWorks, Inc.

x = round(x); % correct for not-exactly integers
r = find(x); % indices of the trips that exist in the solution
substuff = idxs(r,:); % the collection of node pairs in the solution
unvisited = ones(length(r),1); % keep track of places not yet visited
curr = 1; % subtour we are evaluating
startour = find(unvisited,1); % first unvisited trip
    while ~isempty(startour)
```

```
        home = substuff(startour,1); % starting point of subtour
        nextpt = substuff(startour,2); % next point of tour
        visited = nextpt; unvisited(startour) = 0; % update unvisited points
        while nextpt ~= home
            % Find the other trips that starts at nextpt
            [srow,scol] = find(substuff == nextpt);
            % Find just the new trip
            trow = srow(srow ~= startour);
            scol = 3-scol(trow == srow); % turn 1 into 2 and 2 into 1
            startour = trow; % the new place on the subtour
            nextpt = substuff(startour,scol); % the point not where we came from
            visited = [visited,nextpt]; % update nodes on the subtour
            unvisited(startour) = 0; % update unvisited
        end
        subTours{curr} = visited; % store in cell array
        curr = curr + 1; % next subtour
        startour = find(unvisited,1); % first unvisited trip
    end
end
```

```
function [Lon, Lat] = TSP(C)

%   Copyright 2014-2016 The MathWorks, Inc.

figure;

nStops = size(C,1); % you can use any number, but the problem size scales as N^2
stopsLon = C(:,1); % allocate x-coordinates of nStops
stopsLat = C(:,2); % allocate y-coordinates

plot(stopsLon,stopsLat,'ob')
hold on

%calculates distances between points
idxs = nchoosek(1:nStops,2);

dist = hypot(stopsLat(idxs(:,1)) - stopsLat(idxs(:,2)), ...
             stopsLon(idxs(:,1)) - stopsLon(idxs(:,2)));
lendist = length(dist);

%equality constraints
Aeq = spones(1:length(idxs)); % Adds up the number of trips
beq = nStops;

Aeq = [Aeq;spalloc(nStops,length(idxs),nStops*(nStops-1))]; % allocate a sparse matrix
for ii = 1:nStops
    whichIdxs = (idxs == ii); % find the trips that include stop ii
    whichIdxs = sparse(sum(whichIdxs,2)); % include trips where ii is at either end
    Aeq(ii+1,:) = whichIdxs'; % include in the constraint matrix
end
beq = [beq; 2*ones(nStops,1)];

%binary bounds
intcon = 1:lendist;
lb = zeros(lendist,1);
ub = ones(lendist,1);

%optimize using intlinprog
opts = optimoptions('intlinprog','Display','off','Heuristics','round-diving',...
    'IPPreprocess','none');
[x_tsp,costopt,exitflag,output] = intlinprog(dist,intcon,[],[],Aeq,beq,lb,ub,opts);
```

```

%visualize the solution
segments = find(x_tsp); % Get indices of lines on optimal path
lh = zeros(nStops,1); % Use to store handles to lines on plot
[Lon, Lat, lh] = updateSalesmanPlot(lh,x_tsp,idxs,stopsLon,stopsLat);
title('Solution with Subtours');

%subtour constraints
tours = detectSubtours(x_tsp,idxs);
numtours = length(tours); % number of subtours
fprintf('# of subtours: %d\n',numtours);

A = spalloc(0,lendist,0); % Allocate a sparse linear inequality constraint matrix
b = [];
while numtours > 1 % repeat until there is just one subtour
    % Add the subtour constraints
    b = [b;zeros(numtours,1)]; % allocate b
    A = [A;spalloc(numtours,lendist,nStops)]; % a guess at how many nonzeros to allocate
    for ii = 1:numtours
        rowIdx = size(A,1)+1; % Counter for indexing
        subTourIdx = tours{ii}; % Extract the current subtour
%         The next lines find all of the variables associated with the
%         particular subtour, then add an inequality constraint to prohibit
%         that subtour and all subtours that use those stops.
        variations = nchoosek(1:length(subTourIdx),2);
        for jj = 1:length(variations)
            whichVar = (sum(idxs==subTourIdx(variations(jj,1)),2)) & ...
                       (sum(idxs==subTourIdx(variations(jj,2)),2));
            A(rowIdx,whichVar) = 1;
        end
        b(rowIdx) = length(subTourIdx)-1; % One less trip than subtour stops
    end

    % Try to optimize again
    [x_tsp,costopt,exitflag,output] = intlinprog(dist,intcon,A,b,Aeq,beq,lb,ub,opts);

    % Visualize result
    [Lon, Lat, lh] = updateSalesmanPlot(lh,x_tsp,idxs,stopsLon,stopsLat);

    % How many subtours this time?
    tours = detectSubtours(x_tsp,idxs);
    numtours = length(tours); % number of subtours
    fprintf('# of subtours: %d\n',numtours);
end

title('Solution with Subtours Eliminated');
hold off

end
```

```
function [Lon, Lat, lh] = updateSalesmanPlot(lh,xopt,idxs,stopsLat,stopsLon)
% Plotting function for tsp_intlinprog example

%   Copyright 2014-2016 The MathWorks, Inc.

if ( lh ~= zeros(size(lh)) ) % First time through lh is all zeros
    set(lh,'Visible','off'); % Remove previous lines from plot
end

segments = find(round(xopt)); % Indices to trips in solution

```

```
% Loop through the trips then draw them
Lat = zeros(3*length(segments),1);
Lon = zeros(3*length(segments),1);
for ii = 1:length(segments)
    start = idxs(segments(ii),1);
    stop = idxs(segments(ii),2);

    % Separate data points with NaN's to plot separate line segments
    Lat(3*ii-2:3*ii) = [stopsLat(start); stopsLat(stop); NaN];
    Lon(3*ii-2:3*ii) = [stopsLon(start); stopsLon(stop); NaN];
end

lh = plot(Lat,Lon,'k:','LineWidth',2);

set(lh,'Visible','on'); drawnow; % Add new lines to plot
```

B.3 Wellbore Trajectory for Satellites

```
%calculates the wellbore trajectories for a satellite field
global BUR

%get completions
C_S = get_completions;

%input parameter
BUR = 3/30; %build-up rate for the build section (deg/m)

%average and total wellpath length of wells
[WPL_S,KOP_S,ROC_S,BUA_S,dR_S,azi_cS] = get_sat_WPL(C_S);

WPL_avg_tot = WPL_S/size(C_S,1);
WPL_tot = WPL_S;

%plot wells in 3D
plot_satellite_3D(C_S,KOP_S,ROC_S,BUA_S,dR_S,azi_cS);
```

B.4 Wellbore Trajectory for Subsea on a Stick

```
%calculates the wellbore trajectory for a "subsea on a stick" field
global BUR BUR2 DOR KOPz TR_max

%get completions
C = get_completions;

%input parameters
BUR = 3/30; %build-up rate for first build section (deg/m)
BUR2 = 3/30; %build-up rate for second build section (deg/m)
DOR = 3/30; %drop-off rate (deg/m)
KOPz = 500; %depth of kick-off point (m)
TR = 3/30; %turn rate (deg/m)
TR_max = 6/30; %maximum turn rate (deg/m)

%average and total wellpath length of wells
[WPL,ROC,ROC2,BUA,BUA1,Ltan,R_cc2,Z_cc2,dRtot,clock,R_st,KOP,azi_t,X_cc1,Y_cc1,TR,ROT,azi_c] = get_SoS_WPL(C,TR);

WPL_avg_tot = sum(WPL)/size(C,1);
```

```
WPL_tot = sum(WPL);

%plot wells
plot_common_dc_3D(C,ROC,ROC2,Ltan,dRtot,R_cc2,Z_cc2,BUA,BUA1,clock,R_st,KOP,azi_t,X_cc1,Y_cc1,TR,ROT,azi_c);
```

```
function [WPL,ROC,ROC2,BUA,BUA1,Ltan,R_cc2,Z_cc2,dRtot,clock,R_st,KOP,azi_t,X_cc1,Y_cc1,TR,ROT,azi_c] = get_SoS_WPL(C,TR)
%calculates average wellpath length of template wells
global KOPz Nwt N_dc row_C fill

%compute optimized drillcenter
DC=get_one_dc(C);

%to use common functions, make C divisible by 12
row_C=size(C,1);
if rem(row_C,12)~=0
    %fill C with zeros to make C divisible by 12
    rest=rem(row_C,12);
    fill=12-rest;
    C(row_C+1:row_C+fill,:)=0;
end
Nwt=size(C,1); %number of wells in C (added due to practical reasons) (-)
N_dc=1; %number of drill centers (-)

%find turn point in the XY-plane
[X2,Y2,TR,clock,X_cc1,Y_cc1,azi_c,DC,ROT]=get_turn_one_dc(C,DC,TR);

%kickoff point
KOP=[DC KOPz];

%completion interval length and build-up angle
[L_c,BUA]=get_BUA(C);

%calculate well path lengths
[WPL,dRtot,BUA,BUA1,Ltan,R_cc2,Z_cc2,ROC,ROC2,azi_t,R_st]=get_temp_WPL(C,L_c,BUA,TR,KOP,X2,Y2);

%remove all calculations made for the excess rows
if rem(row_C,12)~=0
    WPL(row_C+1:row_C+fill)=[];
    ROC(row_C+1:row_C+fill)=[];
    ROC2(row_C+1:row_C+fill)=[];
    BUA(row_C+1:row_C+fill)=[];
    BUA1(row_C+1:row_C+fill)=[];
    Ltan(row_C+1:row_C+fill)=[];
    R_cc2(row_C+1:row_C+fill)=[];
    Z_cc2(row_C+1:row_C+fill)=[];
    dRtot(row_C+1:row_C+fill)=[];
    azi_c(row_C+1:row_C+fill)=[];
    azi_t(row_C+1:row_C+fill)=[];
    R_st(row_C+1:row_C+fill)=[];
    clock(row_C+1:row_C+fill)=[];
    X_cc1(row_C+1:row_C+fill)=[];
    Y_cc1(row_C+1:row_C+fill)=[];
    ROT(row_C+1:row_C+fill)=[];
end

end
```

```
function [DC]=get_one_dc(C)
%calulates the optimized coordinates of one common drill center

%compute average X and Y coordinates
```

```
X_avg=sum(C(:,1))/size(C,1);
Y_avg=sum(C(:,2))/size(C,1);

%gather the coordinates in a common drill center vector
DC=[X_avg Y_avg 0];

end
```

```
function [X2,Y2,TR,clock,X_cc1,Y_cc1,azi_c,DC,ROT]=get_turn_one_dc(C,DC,TR)
%calculates the turn point in the XY-plane
global TR_max

%radius of turn
TR=ones(1,size(C,1))*TR;
ROT=360./(2*pi*TR);

%check if distance between WH and completion start is less than 2*ROT
k=1;
while k <= size(C,1)
    if hypot(DC(1)-C(k,1),DC(2)-C(k,2)) < 2*ROT(k)
        %increase the turn rate
        TR(k)=TR(k)+1/30;
        ROT(k)=360/(2*pi*TR(k));
        if TR(k)>TR_max
            TR(k)=TR(k)-1/30;
            ROT(k)=360/(2*pi*TR(k));
            dir=atan2d(DC(1)-C(k,1),DC(2)-C(k,2));
            dist=hypot(C(k,1)-DC(1),C(k,2)-DC(2));
            DC=[DC(1)+(2*ROT(k)-dist)*sind(dir) DC(2)+(2*ROT(k)-dist)*cosd(dir) 0];
            TR=ones(1,size(C,1))*3/30;
            ROT=360./(2*pi*TR);
            k=1;
        end
    else
        k=k+1;
    end
end

%place coordinates in X_opt and Y_opt
X_opt(1,:)=C(:,1);
X_opt(2,:)=C(:,4);
Y_opt(1,:)=C(:,2);
Y_opt(2,:)=C(:,5);

%find turn point in the XY-plane
[X2,Y2,clock,X_cc1,Y_cc1,azi_c]=get_turn(DC,X_opt,Y_opt,ROT);

end
```

B.5 Wellbore Trajectory for Templates

```
%calculates the wellbore trajectories from 2-slots templates
global BUR BUR2 DOR KOPz TR_max Nwt N_dc

%get completions
C = get_completions;

%input parameters
```

```
BUR = 3/30; %build-up rate for first build section (deg/m)
BUR2 = 3/30; %build-up rate for second build section (deg/m)
DOR = 3/30; %drop-off rate (deg/m)
KOPz = 500; %depth of kick-off point (m)
TR = 3/30; %turn rate (deg/m)
TR_max = 6/30; %maximum turn rate (deg/m)
Nwt = 6; %number of wells per template (-)

%number of drill centers
N_dc = 12/Nwt;

%find satellite wells
[C,C_S] = get_satellites(C);

%get order of completion intervals
order = get_order(C);

%average wellpath length of template and satellite wells
[C_T,WPL_T,dRtotT,BUA_T,BUA1_T,LtanT,R_cc2T,Z_cc2T,ROC_T,ROC2_T,TR_T,clock,R_st,KOP_T,azi_t,X_cc1,Y_cc1,ROT,azi_cT] = get_avg_temp_WPL(C,order,TR);

if sum(sum(C_S))>0
    [WPL_S,KOP_S,ROC_S,BUA_S,dR_S,azi_cS] = get_sat_WPL(C_S);
    WPL_avg_tot = (WPL_T + WPL_S)/(size(C_T,1) + size(C_S,1));
    WPL_tot = WPL_T + WPL_S;
else
    WPL_avg_tot = WPL_T/size(C_T,1);
    WPL_tot = WPL_T;
end

%plot wells in 3D
plot_common_dc_3D(C_T,ROC_T,ROC2_T,LtanT,dRtotT,R_cc2T,Z_cc2T,BUA_T,BUA1_T,clock,R_st,KOP_T,azi_t,X_cc1,Y_cc1,TR_T,ROT,azi_cT)

if sum(sum(C_S))>0
    plot_satellite_3D(C_S,KOP_S,ROC_S,BUA_S,dR_S,azi_cS);
end
```

```
function [C_T,WPL_T,dRtotT,BUA_T,BUA1_T,LtanT,R_cc2T,Z_cc2T,ROC_T,ROC2_T,TR_T,clock_T,R_stT,KOP_T,azi_tT,X_cc1T,Y_cc1T,ROT_T,azi_cT] =
        get_avg_temp_WPL(C,order,TR)
%calculates average wellpath length of 2-slots template wells
global KOPz Ncg N_dc

%parameters
row_C=size(C,1); %number of completions
Ncg=row_C/12; %number of groups with 12 wells
last=order(row_C,1:2); %the last completion coordinates in order

%creating empty matrices
DC=zeros(Ncg*N_dc,3);
KOP=zeros(Ncg*N_dc,3);
KOP(:,3)=KOPz;
ROT=zeros(Ncg,12);
TR_n=zeros(Ncg,12);
azi_t=zeros(Ncg,12);
azi_c=zeros(Ncg,12);
clock=zeros(Ncg,12);
L_c=zeros(Ncg,12);
BUA=zeros(Ncg,12);
BUA1=zeros(Ncg,12);
WPL_avg=zeros(1,12);
WPL=zeros(Ncg,12);
dRtot=zeros(Ncg,12);
```

```
Ltan=zeros(Ncg,12);
ROC=zeros(Ncg,12);
ROC2=zeros(Ncg,12);
R_cc2=zeros(Ncg,12);
Z_cc2=zeros(Ncg,12);
X_cc1=zeros(Ncg,12);
Y_cc1=zeros(Ncg,12);
R_st=zeros(Ncg,12);
for ii=1:12
    %rearrange order until all combinations are tested
    if ii>1
        order(2:row_C,1:2)=order(1:row_C-1,1:2);
        order(1,1:2)=last;
        last=order(row_C,1:2);
    end

    %create new C with the arrangement from order
    C_new=zeros(size(C));
    for i=1:row_C
        j=1;
        while j<=size(C,1)
            if order(i,1:2)==C(j,1:2)
                C_new(i,:)=C(j,:);
            end
        j=j+1;
        end
    end

%calculate properties for all groups of 12 completions
k=1;
while k<=Ncg
    %compute optimized drillcenters and corresponding groups
    if N_dc == 6
        [DC(6*k-5:6*k,:),X_opt,Y_opt,Z_opt,TR_n(k,:),ROT(k,:)]=get_six_dc(C_new(12*k-11:12*k,:),TR);
    elseif N_dc == 3
        [DC(3*k-2:3*k,:),X_opt,Y_opt,Z_opt,TR_n(k,:),ROT(k,:)]=get_three_dc(C_new(12*k-11:12*k,:),TR);
    elseif N_dc == 2
        [DC(2*k-1:2*k,:),X_opt,Y_opt,Z_opt,TR_n(k,:),ROT(k,:)]=get_two_dc(C_new(12*k-11:12*k,:),TR);
    end

    %rearrange the completion intervals
    C(12*k-11:12*k,1)=X_opt(1,:);    C(12*k-11:12*k,2)=Y_opt(1,:);    C(12*k-11:12*k,3)=Z_opt(1,:);
    C(12*k-11:12*k,4)=X_opt(2,:);    C(12*k-11:12*k,5)=Y_opt(2,:);    C(12*k-11:12*k,6)=Z_opt(2,:);

    %find turn point in the XY-plane
    [X2,Y2,clock(k,:),X_cc1(k,:),Y_cc1(k,:),azi_c(k,:)]=get_turn(DC(N_dc*k-(N_dc-1):N_dc*k,:),X_opt,Y_opt,ROT(k,:));

    %coordinates of the 1st kickoff points
    KOP(N_dc*k-(N_dc-1):N_dc*k,1:2)=DC(N_dc*k-(N_dc-1):N_dc*k,1:2);

    %completion interval length and build-up angle
    [L_c(k,:),BUA(k,:)]=get_BUA(C(12*k-11:12*k,:));

    %calculate well path lengths
    [WPL(k,:),dRtot(k,:),BUA(k,:),BUA1(k,:),Ltan(k,:),R_cc2(k,:),Z_cc2(k,:),ROC(k,:),ROC2(k,:),azi_t(k,:),R_st(k,:)]=get_temp_WPL(C(12*k-11:12*k,:),
        L_c(k,:),BUA(k,:),TR_n(k,:),KOP(N_dc*k-(N_dc-1):N_dc*k,:),X2,Y2);

    k=k+1;
end

%average well path length of each arrangement
WPL_avg(ii)=sum(sum(WPL,2))/(12*Ncg);
```

```

%save the solution with lowest WPL
if ii==1
    WPL_avgT=WPL_avg(ii);
    C_T=C;
    BUA_T=BUA;
    BUA1_T=BUA1;
    LtanT=Ltan;
    ROC_T=ROC;
    ROC2_T=ROC2;
    R_cc2T=R_cc2;
    Z_cc2T=Z_cc2;
    dRtotT=dRtot;
    TR_T=TR_n;
    clock_T=clock;
    R_stT=R_st;
    KOP_T=KOP;
    azi_tT=azi_t;
    azi_cT=azi_c;
    X_cc1T=X_cc1;
    Y_cc1T=Y_cc1;
    ROT_T=ROT;
    WPL_T=sum(sum(WPL));
elseif ii>1 && WPL_avg(ii)<WPL_avgT
    WPL_avgT=WPL_avg(ii);
    C_T=C;
    BUA_T=BUA;
    BUA1_T=BUA1;
    LtanT=Ltan;
    ROC_T=ROC;
    ROC2_T=ROC2;
    R_cc2T=R_cc2;
    Z_cc2T=Z_cc2;
    dRtotT=dRtot;
    TR_T=TR_n;
    clock_T=clock;
    R_stT=R_st;
    KOP_T=KOP;
    azi_tT=azi_t;
    azi_cT=azi_c;
    X_cc1T=X_cc1;
    Y_cc1T=Y_cc1;
    ROT_T=ROT;
    WPL_T=sum(sum(WPL));
end
end

end
```

```
function [DC,X_opt,Y_opt,Z_opt,TR,ROT]=get_six_dc(C,TR)
%calulates the optimized coordinates of six drill centers
global TR_max Nwt N_dc

%number of completion intervals
N=size(C,1);
A=(1:N);

%all possible combinations of template 1
R1=nchoosek(A,Nwt);

%remaining combinations in R_rest
```

```
R_rest=zeros(size(R1,1),N-Nwt);
for i=1:N-Nwt
    for j=1:size(R1,1)
        Rtemp=R1(j,:);
        R=zeros(1,N-Nwt);
        k=1;
        while k<=N-Nwt
            V=randi(N);
            if sum(Rtemp==V)==0 && sum(R==V)==0
                R(k)=V;
                k=k+1;
            end
        end
        R_rest(j,:)=R;
    end
end

%all possible combinations of template 2
R2=zeros(45,size(R1,1)*size(R1,2));
for i=1:size(R_rest,1)
    R2(:,2*i-1:2*i)=nchoosek(R_rest(i,:),Nwt);
end

%remaining combinations in R_rest
R_rest=zeros(size(R2,1),(N-2*Nwt)*size(R1,1));
for i=1:size(R1,1)
    Rtemp=R1(i,:);
    for j=1:size(R2,1)
        Rtemp2=R2(j,2*i-1:2*i);
        R=zeros(1,N-2*Nwt);
        k=1;
        while k<=N-2*Nwt
            V=randi(N);
            if sum(Rtemp==V)==0 && sum(R==V)==0 && sum(Rtemp2==V)==0
                R(k)=V;
                k=k+1;
            end
        end
        R_rest(j,8*i-7:8*i)=R;
    end
end

%all possible combinations of template 3
R3=zeros(28*size(R2,1),size(R2,2));
for i=1:size(R_rest,2)/8
    for j=1:size(R_rest,1)
        R3(28*j-27:28*j,2*i-1:2*i)=nchoosek(R_rest(j,8*i-7:8*i),Nwt);
    end
end

%remaining combinations in R_rest
R_rest=zeros(size(R3,1),(N-3*Nwt)*size(R1,1));
l=1;
for i=1:size(R1,1)
    Rtemp=R1(i,:);
    for j=1:size(R2,1)
        Rtemp2=R2(j,2*i-1:2*i);
        while l<=size(R3,1)
            Rtemp3=R3(l,2*i-1:2*i);
            R=zeros(1,N-3*Nwt);
            k=1;
            while k<=N-3*Nwt
```

```
                V=randi(N);
                if sum(Rtemp==V)==0 && sum(R==V)==0 && sum(Rtemp2==V)==0 ...
                        && sum(Rtemp3==V)==0
                    R(k)=V;
                    k=k+1;
                end
            end
            R_rest(l,6*i-5:6*i)=R;
            l=l+1;
            if rem(l-1,28)==0
                break
            end
        end
    end
    l=1;
end

%all possible combinations of template 4
R4=zeros(15*size(R3,1),size(R3,2));
for i=1:size(R_rest,2)/6
    for j=1:size(R_rest,1)
        R4(15*j-14:15*j,2*i-1:2*i)=nchoosek(R_rest(j,6*i-5:6*i),Nwt);
    end
end

%remaining combinations in R_rest
R_rest=zeros(size(R4,1),(N-4*Nwt)*size(R1,1));
l=1;
m=1;
for i=1:size(R1,1)
    Rtemp=R1(i,:);
    for j=1:size(R2,1)
        Rtemp2=R2(j,2*i-1:2*i);
        while l<=size(R3,1)
            Rtemp3=R3(l,2*i-1:2*i);
            while m<=size(R4,1)
                Rtemp4=R4(m,2*i-1:2*i);
                R=zeros(1,N-4*Nwt);
                k=1;
                while k<=N-4*Nwt
                    V=randi(N);
                    if sum(Rtemp==V)==0 && sum(R==V)==0 && sum(Rtemp2==V)==0 ...
                            && sum(Rtemp3==V)==0 && sum(Rtemp4==V)==0
                        R(k)=V;
                        k=k+1;
                    end
                end
                R_rest(m,4*i-3:4*i)=R;
                m=m+1;
                if rem(m-1,15)==0
                    break
                end
            end
            l=l+1;
            if rem(l-1,28)==0
                break
            end
        end
    end
    m=1;
    l=1;
end
```

```
%all possible combinations of template 5
R5=zeros(6*size(R4,1),size(R4,2));
for i=1:size(R_rest,2)/4
    for j=1:size(R_rest,1)
        R5(6*j-5:6*j,2*i-1:2*i)=nchoosek(R_rest(j,4*i-3:4*i),Nwt);
    end
end

%delete R_rest
R_rest=[];

%all possible cominations of template 6
R6=zeros(size(R5));
l=1;
m=1;
n=1;
for i=1:size(R1,1)
    Rtemp=R1(i,:);
    for j=1:size(R2,1)
        Rtemp2=R2(j,2*i-1:2*i);
        while l<=size(R3,1)
            Rtemp3=R3(l,2*i-1:2*i);
            while m<=size(R4,1)
                Rtemp4=R4(m,2*i-1:2*i);
                while n<=size(R5,1)
                    Rtemp5=R5(n,2*i-1:2*i);
                    R=zeros(1,N-5*Nwt);
                    k=1;
                    while k<=N-5*Nwt
                        V=randi(N);
                        if sum(Rtemp==V)==0 && sum(R==V)==0 && sum(Rtemp2==V)==0 ...
                                && sum(Rtemp3==V)==0 && sum(Rtemp4==V)==0 && sum(Rtemp5==V)==0
                            R(k)=V;
                            k=k+1;
                        end
                    end
                    R6(n,2*i-1:2*i)=R;
                    n=n+1;
                    if rem(n-1,6)==0
                        break
                    end
                end
                m=m+1;
                if rem(m-1,15)==0
                        break
                end
            end
            l=l+1;
            if rem(l-1,28)==0
                break
            end
        end
    end
    n=1;
    m=1;
    l=1;
end


%combine all 6 templates
M=size(R1,1)*size(R1,2)*N_dc;
```

```
Pos=zeros(size(R6,1),M);
for i=1:size(R1,1)
    Pos(:,12*i-1:12*i)=R6(:,2*i-1:2*i);
    Pos(:,12*i-3:12*i-2)=R5(:,2*i-1:2*i);
    Pos(:,12*i-11:12*i-10)=ones(113400,2).*R1(i,:);
    for j=1:size(R2,1)
        Pos(2520*j-2519:2520*j,12*i-9:12*i-8)=ones(2520,2).*R2(j,2*i-1:2*i);
    end
    for k=1:size(R3,1)
        Pos(90*k-89:90*k,12*i-7:12*i-6)=ones(90,2).*R3(k,2*i-1:2*i);
    end
    for l=1:size(R4,1)
        Pos(6*l-5:6*l,12*i-5:12*i-4)=ones(6,2).*R4(l,2*i-1:2*i);
    end
end

%coordinates corresponding to values in Pos
X=reshape(C(Pos',1),792,113400)';
Y=reshape(C(Pos',2),792,113400)';
Z=reshape(C(Pos',3),792,113400)';

%drill center to every template
X_row=zeros(size(R6,1),M/Nwt);
Y_row=zeros(size(R6,1),M/Nwt);
X_dc=zeros(size(R6,1),M);
Y_dc=zeros(size(R6,1),M);
for j=1:size(R6,1)
    X_row(j,:)=sum(reshape(X(j,:),Nwt,M/Nwt))/Nwt;
    Y_row(j,:)=sum(reshape(Y(j,:),Nwt,M/Nwt))/Nwt;
    for i=1:M/Nwt
        X_dc(j,2*i-1:2*i)=X_row(j,i);
        Y_dc(j,2*i-1:2*i)=Y_row(j,i);
    end
end

%distance from drill center to every completion coordinate
dist_dc=hypot(X-X_dc,Y-Y_dc);

%average distances for each group
dist_avg=zeros(size(R6,1),size(R1,1));
for j=1:size(R6,1)
    dist_avg(j,:)=sum(reshape(dist_dc(j,:),12,66))/12;
end

%the combination that yields the shortest well path lengths
[min_avg_col,J]=min(dist_avg,[],1);
[min_avg_dist,I]=min(min_avg_col);
row_min=J(I);

X_opt(1,:)=X(row_min,I*12-11:I*12);
Y_opt(1,:)=Y(row_min,I*12-11:I*12);
Z_opt(1,:)=Z(row_min,I*12-11:I*12);

X_opt(2,:)=C(Pos(row_min,I*12-11:I*12),4);
Y_opt(2,:)=C(Pos(row_min,I*12-11:I*12),5);
Z_opt(2,:)=C(Pos(row_min,I*12-11:I*12),6);

%corresponding DC coordinates
DC(:,1:2)=[(sum(reshape(X_opt(1,:),Nwt,N_dc))/Nwt)' ...
    (sum(reshape(Y_opt(1,:),Nwt,N_dc))/Nwt)'];
DC(:,3)=0;
```

```
%radius of turn
TR=ones(1,12)*TR;
ROT=360./(2*pi*TR);

%check if distance between WH and completion start is less than 2*ROT
l=1;
k=1;
while l<=6 && k<=size(C,1)
    if hypot(DC(l,1)-X_opt(1,k),DC(l,2)-Y_opt(1,k)) < 2*ROT(k)
        %increase the turn rate
        TR(k)=TR(k)+1/30;
        ROT(k)=360/(2*pi*TR(k));
        if TR(k)>TR_max
            %set minimum value equal to maximum value
            [max_avg_col,K]=max(dist_avg,[],1);
            [max_avg_dist,J]=max(max_avg_col);
            row_max=K(J);
            dist_avg(row_min,I)=dist_avg(row_max,J);

            %the combination that yields the shortest well path lengths
            [min_avg_col,J]=min(dist_avg,[],1);
            [min_avg_dist,I]=min(min_avg_col);
            row_min=J(I);

            X_opt(1,:)=X(row_min,I*12-11:I*12);
            Y_opt(1,:)=Y(row_min,I*12-11:I*12);
            Z_opt(1,:)=Z(row_min,I*12-11:I*12);
            X_opt(2,:)=C(Pos(row_min,I*12-11:I*12),4);
            Y_opt(2,:)=C(Pos(row_min,I*12-11:I*12),5);
            Z_opt(2,:)=C(Pos(row_min,I*12-11:I*12),6);

            %corresponding DC coordinates
            DC(:,1:2)=[(sum(reshape(X_opt(1,:),Nwt,N_dc))/Nwt)' ...
                (sum(reshape(Y_opt(1,:),Nwt,N_dc))/Nwt)'];

            %set turn rate and radius of turn back to inital values
            TR=ones(1,12)*3/30;
            ROT=360./(2*pi*TR);
            k=1;
            l=1;
        end
    else
        k=k+1;
        if rem(k-1,2)==0
            l=l+1;
        end
    end
end

end
```

```
function [DC,X_opt,Y_opt,Z_opt,TR,ROT]=get_three_dc(C,TR)
%calulates the optimized coordinates of three drill centers
global TR_max Nwt N_dc

%number of completion intervals
N=size(C,1);
A=(1:N);

%all possible combinations of template 1
R1=nchoosek(A,Nwt);
```

```
%remaining combinations in R_rest
R_rest=zeros(size(R1,1),N-Nwt);
for i=1:N-Nwt
    for j=1:size(R1,1)
        Rtemp=R1(j,:);
        R=zeros(1,N-Nwt);
        k=1;
        while k<=N-Nwt
            V=randi(N);
            if sum(Rtemp==V)==0 && sum(R==V)==0
                R(k)=V;
                k=k+1;
            end
        end
        R_rest(j,:)=R;
    end
end

%all possible combinations of template 2
R2=zeros(70,size(R1,1)*size(R1,2));
for i=1:size(R_rest,1)
    R2(:,4*i-3:4*i)=nchoosek(R_rest(i,:),Nwt);
end

%deleting R_rest
R_rest=[];

%the remaining combinations of template 3
R3=zeros(70,size(R1,1)*size(R1,2));
 for i=1:size(R1,1)
     Rtemp=R1(i,:);
     for j=1:size(R3,1)
        Rtemp2=R2(j,4*i-3:4*i);
        R=zeros(1,Nwt);
        k=1;
        while k<=Nwt
            V=randi(N);
            if sum(Rtemp==V)==0 && sum(R==V)==0 && sum(Rtemp2==V)==0
                R(k)=V;
                k=k+1;
            end
        end
        R3(j,4*i-3:4*i)=R;
     end
 end

%combine all 3 templates
M=size(R1,1)*size(R1,2)*N_dc;
Pos=zeros(size(R3,1),M);
for i=1:size(R1,1)
    Pos(:,12*i-3:12*i)=R3(:,4*i-3:4*i);
    Pos(:,12*i-7:12*i-4)=R2(:,4*i-3:4*i);
    for j=1:size(R3,1)
        Pos(j,12*i-11:12*i-8)=R1(i,:);
    end
end

%coordinates corresponding to values in Pos
X=reshape(C(Pos',1),5940,70)';
Y=reshape(C(Pos',2),5940,70)';
Z=reshape(C(Pos',3),5940,70)';
```

```
%drill center to every template
X_row=zeros(size(R3,1),M/Nwt);
Y_row=zeros(size(R3,1),M/Nwt);
X_dc=zeros(size(R3,1),M);
Y_dc=zeros(size(R3,1),M);
for j=1:size(R3,1)
    X_row(j,:)=sum(reshape(X(j,:),Nwt,M/Nwt))/Nwt;
    Y_row(j,:)=sum(reshape(Y(j,:),Nwt,M/Nwt))/Nwt;
    for i=1:M/Nwt
        X_dc(j,4*i-3:4*i)=X_row(j,i);
        Y_dc(j,4*i-3:4*i)=Y_row(j,i);
    end
end

%distance from drill center to every completion coordinate
dist_dc=hypot(X-X_dc,Y-Y_dc);

%average distances for each group
dist_avg=zeros(size(R3,1),size(R1,1));
for j=1:size(R3,1)
    dist_avg(j,:)=sum(reshape(dist_dc(j,:),12,495))/12;
end

%the combination that yields the shortest well path lengths
[min_avg_col,J]=min(dist_avg,[],1);
[min_avg_dist,I]=min(min_avg_col);
row_min=J(I);

X_opt(1,:)=X(row_min,I*12-11:I*12);
Y_opt(1,:)=Y(row_min,I*12-11:I*12);
Z_opt(1,:)=Z(row_min,I*12-11:I*12);

X_opt(2,:)=C(Pos(row_min,I*12-11:I*12),4);
Y_opt(2,:)=C(Pos(row_min,I*12-11:I*12),5);
Z_opt(2,:)=C(Pos(row_min,I*12-11:I*12),6);

%corresponding DC coordinates
DC(:,1:2)=[(sum(reshape(X_opt(1,:),Nwt,N_dc))/Nwt)' ...
    (sum(reshape(Y_opt(1,:),Nwt,N_dc))/Nwt)'];
DC(:,3)=0;

%radius of turn
TR=ones(1,12)*TR;
ROT=360./(2*pi*TR);

%check if distance between WH and completion start is less than 2*ROT
l=1;
k=1;
while l<=3 && k<=size(C,1)
    if hypot(DC(l,1)-X_opt(1,k),DC(l,2)-Y_opt(1,k)) < 2*ROT(k)
        %increase the turn rate
        TR(k)=TR(k)+1/30;
        ROT(k)=360/(2*pi*TR(k));
        if TR(k)>TR_max
            %set minimum value equal to maximum value
            [max_avg_col,K]=max(dist_avg,[],1);
            [max_avg_dist,J]=max(max_avg_col);
            row_max=K(J);
            dist_avg(row_min,I)=dist_avg(row_max,J);

            %the combination that yields the shortest well path lengths
```

```
            [min_avg_col, J]=min(dist_avg, [], 1);
            [min_avg_dist, I]=min(min_avg_col);
            row_min=J(I);

            X_opt(1,:)=X(row_min, I*12-11:I*12);
            Y_opt(1,:)=Y(row_min, I*12-11:I*12);
            Z_opt(1,:)=Z(row_min, I*12-11:I*12);
            X_opt(2,:)=C(Pos(row_min, I*12-11:I*12),4);
            Y_opt(2,:)=C(Pos(row_min, I*12-11:I*12),5);
            Z_opt(2,:)=C(Pos(row_min, I*12-11:I*12),6);

            %corresponding DC coordinates
            DC(:,1:2)=[(sum(reshape(X_opt(1,:),Nwt,N_dc))/Nwt)' ...
                (sum(reshape(Y_opt(1,:),Nwt,N_dc))/Nwt)'];

            %set turn rate and radius of turn back to inital values
            TR=ones(1,12)*3/30;
            ROT=360./(2*pi*TR);
            k=1;
            l=1;
        end
    else
        k=k+1;
        if rem(k-1,4)==0
            l=l+1;
        end
    end
end

end
```

```
function [DC,X_opt,Y_opt,Z_opt,TR,ROT]=get_two_dc(C,TR)
%calulates the optimized coordinates of two drill centers
global TR_max Nwt

%number of completion intervals
N=size(C,1);
A=(1:N);

%all possible combinations of template 1
R1=nchoosek(A,Nwt);

%the corresponding combinations of template 2
R2=zeros(size(R1));
for i=1:size(R1,1)
    Rtemp=R1(i,:);
    R=zeros(1,Nwt);
    j=1;
    while j <= size(R1,2)
        V=randi(N);
        if sum(Rtemp==V)==0 && sum(R==V)==0
            R(j)=V;
            j=j+1;
        end
    end
    R2(i,:)=R;
end

%coordinates corresponding to values in R1 and R2
X1=reshape(C(R1',1),Nwt,924);
Y1=reshape(C(R1',2),Nwt,924);
```

```
X2=reshape(C(R2',1),Nwt,924);
Y2=reshape(C(R2',2),Nwt,924);

%drillcenters
X_dc1=sum(X1)/Nwt;
Y_dc1=sum(Y1)/Nwt;
X_dc2=sum(X2)/Nwt;
Y_dc2=sum(Y2)/Nwt;

%distance from drill center to every completion coordinate
dist=hypot(X1-X_dc1,Y1-Y_dc1);
dist_2=hypot(X2-X_dc2,Y2-Y_dc2);

%average distances for each group
dist_avg=sum(dist)/Nwt;
dist_avg_2=sum(dist_2)/Nwt;

%the combination that yields the shortest well path lengths
dist_tot=(dist_avg+dist_avg_2)/2;
[g,col]=min(dist_tot);

X_opt=[C(R1(col,:),1)' C(R2(col,:),1)'];
Y_opt=[C(R1(col,:),2)' C(R2(col,:),2)'];
Z_opt=[C(R1(col,:),3)' C(R2(col,:),3)'];

X_opt(2,:)=[C(R1(col,:),4)' C(R2(col,:),4)'];
Y_opt(2,:)=[C(R1(col,:),5)' C(R2(col,:),5)'];
Z_opt(2,:)=[C(R1(col,:),6)' C(R2(col,:),6)'];

%corresponding DC coordinates
DC1=[X_dc1(col) Y_dc1(col) 0];
DC2=[X_dc2(col) Y_dc2(col) 0];
DC=[DC1;DC2];

%radius of turn
TR=ones(1,12)*TR;
ROT=360./(2*pi*TR);

%check if distance between WH and completion start is less than 2*ROT
l=1;
k=1;
while l<=2 && k<=size(C,1)
    if hypot(DC(l,1)-X_opt(1,k),DC(l,2)-Y_opt(1,k)) < 2*ROT(k)
        %increase the turn rate
        TR(k)=TR(k)+1/30;
        ROT(k)=360/(2*pi*TR(k));
        if TR(k)>TR_max
            %set minimum value equal to maximum value
            [max_avg_dist,K]=max(dist_tot);
            [min_avg_dist,J]=min(dist_tot);
            dist_tot(J)=max_avg_dist;

            %the combination that yields the shortest well path lengths
            [min_avg_dist,col]=min(dist_tot);

            X_opt=[C(R1(col,:),1)' C(R2(col,:),1)'];
            Y_opt=[C(R1(col,:),2)' C(R2(col,:),2)'];
            Z_opt=[C(R1(col,:),3)' C(R2(col,:),3)'];

            X_opt(2,:)=[C(R1(col,:),4)' C(R2(col,:),4)'];
            Y_opt(2,:)=[C(R1(col,:),5)' C(R2(col,:),5)'];
            Z_opt(2,:)=[C(R1(col,:),6)' C(R2(col,:),6)'];
```

```
            %corresponding DC coordinates
            DC1=[X_dc1(col) Y_dc1(col) 0];
            DC2=[X_dc2(col) Y_dc2(col) 0];
            DC=[DC1;DC2];

            %set turn rate and radius of turn back to inital values
            TR=ones(1,12)*3/30;
            ROT=360./(2*pi*TR);
            k=1;
            l=1;
        end
    else
        k=k+1;
        if rem(k-1,6)==0
            l=l+1;
        end
    end
end

end
```

B.6 Common Functions

```
function C = get_completions

%%field data 1 (horizontal completion intervals)
wellc=load('fielddata1.mat');
C=cell2mat(struct2cell(wellc));

%%field data 2 (non-horizontal completion intervals)
load('fielddata2.mat')
for i=1:31
    %get start and end completion interval coordinates
    C(i,1:3)=wellc{1,i}(1,:);
    C(i,4:6)=wellc{1,i}(end,:);
    if C(i,3)<C(i,6)
        %remove upwards completions
        m=C(i,3);
        C(i,3)=C(i,6);
        C(i,6)=m;
    end
end
% make all coordinates positive
C(:,1:2)=abs(min(min(C)))+C(:,1:2)+100;
C(:,4:5)=abs(min(min(C)))+C(:,4:5)+100;
C(:,3)=abs(C(:,3));
C(:,6)=abs(C(:,6));

%manipulate vertical wells to make them compatible for the calculations
for i=1:size(C,1)
    if C(i,1)==C(i,4)
        C(i,4)=C(i,4)+1;
    elseif C(i,2)==C(i,5)
        C(i,5)=C(i,5)+1;
    end
end

end
```

```
function [L_c,BUA]=get_BUA(C)
%calculates the length and inclination of each completion interval

%lengt and height of completion interval
L_c=(sqrt((C(:,4)-C(:,1)).^2+(C(:,5)-C(:,2)).^2+(C(:,6)-C(:,3)).^2))';
dZ_c=C(:,6)'-C(:,3)';

%inclination of completion interval (build-up angle)
BUA=acosd(dZ_c./L_c);

end
```

```
function order = get_order(C)
%places the completions from traveling salesman in order

%find connecting completions from traveling salesman
row_C = size(C,1);
[Lon, Lat] = TSP(C);

%place the first two completions in order
order = zeros(row_C,2);
order(1:2,1) = Lat(1:2);
order(1:2,2) = Lon(1:2);

for i = 2:row_C-1
    %every third row in Lat and Lon is equal to zero
    j = 4;
    while j < row_C*3
        if Lat(j)==order(i,1) && Lon(j)==order(i,2)
            order(i+1,1:2) = [Lat(j+1) Lon(j+1)];
            %set the matching numbers equal to 0 to avoid repetition
            Lat(j) = 0;
            Lon(j) = 0;
            %if the numbers placed in order were 0, pick the numbers above
            if isnan(Lat(j+1))==1
                order(i+1,1:2) = [Lat(j-1) Lon(j-1)];
                %set the chosen numbers equal to 0 to avoid repetition
                Lat(j-1) = 0;
                Lon(j-1) = 0;
            end
        %set the chosen numbers equal to 0 to avoid repetition
        Lat(j+1) = 0;
        Lon(j+1) = 0;
        end
    j = j+1;
    end
end

end
```

```
function [WPL_S,KOP,ROC,BUA,dR,azi_c] = get_sat_WPL(C)
%calculates the average well path length of all satellite wells
global BUR

%completion interval length and build-up angle
[L_c,BUA]=get_BUA(C);

%arc length and radius of curvature
arc=BUA/BUR;
```

```
ROC=(360*arc)./(2*pi*BUA);

%azimuth and completion start coordinates
azi_c=atan2d((C(:,4)'-C(:,1)'),(C(:,5)'-C(:,2)'));
dR=ROC+ROC.*sind(BUA-90);
dZ=ROC.*cosd(BUA-90);
for i=1:size(C,1)
    if BUA(i)<90
        dR(i)=ROC(i)-ROC(i)*cosd(BUA(i));
        dZ(i)=ROC(i)*sind(BUA(i));
    end
end

%coordinates of kick off point
KOP(:,1)=C(:,1)-dR'.*sind(azi_c');
KOP(:,2)=C(:,2)-dR'.*cosd(azi_c');
KOP(:,3)=C(:,3)-dZ';

%total well path length of all satellite wells
WPL=L_c+arc+KOP(:,3)';
WPL_S=sum(WPL);

end
```

```
function [C,C_sat] = get_satellites(C)
%finds coordinates of the satellite wells
global Nwt

%number of satellite wells
Nsat=0;
if rem(size(C,1),12)~=0
    Nsat=rem(size(C,1),12);
end
C_sat=zeros(Nsat,size(C,2));

%calculate distance between each well
dist=zeros(size(C,1),size(C,1));
for i=1:size(C,1)
    for j=1:size(C,1)
        dist(j,i)=hypot(C(i,1)-C(j,1),C(i,2)-C(j,2));
    end
end

%get satellite wells
F=sort(dist);
if Nsat>0
    k=1;
    while k<=Nsat
        %the well with the longest distance to surrounding wells is found
        [max_dist,sat]=max(sum(F(1:Nwt,:)),[],2);
        C_sat(k,:)=C(sat,:);
        %remove the coordinates of the identified satellite well
        F(:,sat)=[];
        C(sat,:)=[];
        k=k+1;
    end
end


end
```

```
function [X2,Y2,clock,X_cc1,Y_cc1,azi_c]=get_turn(DC,X_opt,Y_opt,ROT)
```

```
%calculates the turn point in the XY-plane
global Nwt N_dc

%parameter
col_X=size(X_opt,2);

%azimuth of completion interval
azi_c=atan2d((X_opt(2,:)-X_opt(1,:)),(Y_opt(2,:)-Y_opt(1,:)));

%completion start coordinates in the new coordinate system (a,b) with originin in the drill centers
a1=reshape(reshape(X_opt(1,:),Nwt,N_dc)-DC(:,1)',1,col_X);
b1=reshape(reshape(Y_opt(1,:),Nwt,N_dc)-DC(:,2)',1,col_X);

%rotation angle needed to place the coordinates in a rotated coordinate system
rotate=180-azi_c;

%completion start coordinates in the rotated coordinate systems (xi,yi) with origin in the drill centers
xi1=a1.*cosd(rotate)+b1.*sind(rotate);
yi1=-a1.*sind(rotate)+b1.*cosd(rotate);

%circle of turn center coordinates in (xi,yi)
x_cci=xi1+ROT;
y_cci=yi1;
for i=1:col_X
    if xi1(i)>0
        x_cci(i)=xi1(i)-ROT(i);
    end
end

%direction of turn (clock<0: clockwise)
clock=x_cci-xi1;

%distances from drill centers to the circles of turn
H=sqrt((x_cci).^2+(y_cci).^2);
B=sqrt(H.^2-ROT.^2);
tetha=(asind(ROT./H));

%coordinates of turn point in (xi,yi)
betha=zeros(1,col_X);
dxi=zeros(1,col_X);
dyi=zeros(1,col_X);
for i=1:col_X
if x_cci(i)>0 && y_cci(i)>0
    if x_cci(i)>xi1(i)
        betha(i)=tetha(i)+atand(x_cci(i)/y_cci(i));
        dxi(i)=B(i)*sind(betha(i));
        dyi(i)=B(i)*cosd(betha(i));
    elseif x_cci(i)<xi1(i)
        betha(i)=tetha(i)+atand(y_cci(i)/x_cci(i));
        dxi(i)=B(i)*cosd(betha(i));
        dyi(i)=B(i)*sind(betha(i));
    end
elseif x_cci(i)>0 && y_cci(i)<0
    if x_cci(i)>xi1(i)
        betha(i)=tetha(i)-atand(y_cci(i)/x_cci(i));
        dxi(i)=B(i)*cosd(betha(i));
        dyi(i)=-B(i)*sind(betha(i));
    elseif x_cci(i)<xi1(i)
        betha(i)=tetha(i)-atand(x_cci(i)/y_cci(i));
        dxi(i)=B(i)*sind(betha(i));
        dyi(i)=-B(i)*cosd(betha(i));
    end
```

```
elseif x_cci(i)<0 && y_cci(i)<0
    if x_cci(i)>xi1(i)
        betha(i)=tetha(i)+atand(x_cci(i)/y_cci(i));
        dxi(i)=-B(i)*sind(betha(i));
        dyi(i)=-B(i)*cosd(betha(i));
    elseif x_cci(i)<xi1(i)
        betha(i)=tetha(i)+atand(y_cci(i)/x_cci(i));
        dxi(i)=-B(i)*cosd(betha(i));
        dyi(i)=-B(i)*sind(betha(i));
    end
elseif x_cci(i)<0 && y_cci(i)>0
    if x_cci(i)>xi1(i)
        betha(i)=tetha(i)-atand(y_cci(i)/x_cci(i));
        dxi(i)=-B(i)*cosd(betha(i));
        dyi(i)=B(i)*sind(betha(i));
    elseif x_cci(i)<xi1(i)
        betha(i)=tetha(i)-atand(x_cci(i)/y_cci(i));
        dxi(i)=-B(i)*sind(betha(i));
        dyi(i)=B(i)*cosd(betha(i));
    end
end
end

%coordinates of turn point in (a,b)
a2=dxi.*cosd(rotate)-dyi.*sind(rotate);
b2=dxi.*sind(rotate)+dyi.*cosd(rotate);
x_cc=x_cci.*cosd(rotate)-y_cci.*sind(rotate);
y_cc=x_cci.*sind(rotate)+y_cci.*cosd(rotate);

%coordinates of turn point in (X,Y)
X2=reshape(DC(:,1)'+reshape(a2,Nwt,N_dc),1,col_X);
Y2=reshape(DC(:,2)'+reshape(b2,Nwt,N_dc),1,col_X);
X_cc1=reshape(DC(:,1)'+reshape(x_cc,Nwt,N_dc),1,col_X);
Y_cc1=reshape(DC(:,2)'+reshape(y_cc,Nwt,N_dc),1,col_X);

end
```

```
function [WPL,dRtot,BUA,BUA1,Ltan,R_cc2,Z_cc2,ROC,ROC2,azi_t,R_st]=get_temp_WPL(C,L_c,BUA,TR,KOP,X2,Y2)
%calculates the well path lengths
global BUR BUR2 DOR KOPz Nwt N_dc

%azimuth of the tangent section in the XY-plane
dtanX=reshape(X2,Nwt,N_dc)-KOP(:,1)';
dtanY=reshape(Y2,Nwt,N_dc)-KOP(:,2)';
if N_dc==1
    azi_t=atan2d(dtanX,dtanY)';
else
    azi_t=reshape(atan2d(dtanX,dtanY),1,Nwt*N_dc);
end

%arc length in the XY-plane
vec_t=[reshape(dtanX,Nwt*N_dc,1) reshape(dtanY,Nwt*N_dc,1) zeros(size(C,1),1)];
vec_c=[C(:,4)-C(:,1) C(:,5)-C(:,2) zeros(size(C,1),1)];
alpha_azi=zeros(1,size(C,1));
a_t=(vec_t(:,2)./vec_t(:,1))';
a_c=(vec_c(:,2)./vec_c(:,1))';
x=(C(:,2)'-Y2+a_t.*X2-a_c.*C(:,1)')./(a_t-a_c);
N=(x-C(:,1)')./(C(:,4)-C(:,1))';
for i=1:size(C,1)
    %calculate angle between two vectors
    alpha_azi(i)=atan2d(norm(cross(vec_t(i,:),vec_c(i,:))),dot(vec_t(i,:),vec_c(i,:)));
```

```
    if N(i)>0
        alpha_azi(i)=360-alpha_azi(i);
    end
end
arc_azi=alpha_azi./TR;

%RZ coordinates of start of completion interval
if N_dc==1
    R_st=hypot(dtanX,dtanY)';
    dRtot=R_st+arc_azi;
else
    R_st=reshape(hypot(dtanX,dtanY),1,Nwt*N_dc);
    dRtot=R_st+arc_azi;
end
dZtot=C(:,3)';

%radius of curvature in the RZ-plane
ROC=ones(1,Nwt*N_dc)*360/(2*pi*BUR);
ROC2=ones(1,Nwt*N_dc)*360/(2*pi*BUR2);

%coordinates of the circle of build centers
R_cc1=ROC;
Z_cc1=KOPz;
R_cc2=dRtot+ROC2.*sind(90-BUA);
Z_cc2=dZtot-ROC2.*cosd(90-BUA);

%distance between circle centers and length of tangent section
Lcc=hypot(Z_cc2-Z_cc1,R_cc2-R_cc1);
Ltan=sqrt(Lcc.^2-(ROC-ROC2).^2);

%build-up angles and arc lengths in the RZ-plane
theta=atand((Z_cc2-Z_cc1)./(R_cc2-R_cc1));
BUA1=90-(theta-atand((ROC-ROC2)./Ltan));
for i=1:size(C,1)
    if R_cc1(i)>R_cc2(i)
        %short J-wells
        theta(i)=90+atand((R_cc1(i)-R_cc2(i))/(Z_cc2(i)-Z_cc1));
        BUA1(i)=90-(theta-atand((ROC-ROC2)./Ltan));
    end
end
BUA2=BUA-BUA1;
arc=BUA1/BUR;
arc2=BUA2/BUR2;

%S-wells
for i=1:size(C,1)
    if BUA(i)<90 && BUA2(i)<0
        ROC2(i)=360/(2*pi*DOR);
        p=ROC2(i)/(ROC(i)+ROC2(i));
        R_cc2(i)=dRtot(i)-ROC2(i)*sind(90-BUA(i));
        Z_cc2(i)=dZtot(i)+ROC2(i)*cosd(90-BUA(i));
        Lcc(i)=hypot(Z_cc2(i)-Z_cc1,R_cc2(i)-R_cc1(i));
        alpha=asind(ROC2(i)/(Lcc(i)*p));
        Ltan(i)=1/p*ROC2(i)/tand(alpha);
        theta(i)=atand((Z_cc2(i)-Z_cc1)/(R_cc2(i)-R_cc1(i)));
        BUA1(i)=90-theta(i)+alpha;
        if R_cc1(i)>R_cc2(i)
            %short S-wells
            theta(i)=90+atand((R_cc1(i)-R_cc2(i))/(Z_cc2(i)-Z_cc1));
            BUA1(i)=90-theta(i)+alpha;
        end
        BUA2(i)=BUA1(i)-BUA(i);
```

```
        arc(i)=BUA1(i)/BUR;
        arc2(i)=BUA2(i)/DOR;
    end
end

%total wellpath length of each well
WPL=L_c+arc+Ltan+KOPz+arc2;

end
```

```
function plot_common_dc(C,ROC,ROC2,L_c,Ltan,dRtot,R_cc2,Z_cc2,BUA,BUA1)
%plots all template wells in 2D
global KOPz N_dc Ncg

%reshape matrices [Ncg,12] to vectors [1,12*Ncg]
if N_dc>1
    ROC=reshape(ROC',1,(Ncg*12));
    ROC2=reshape(ROC2',1,(Ncg*12),1);
    L_c=reshape(L_c',1,(Ncg*12),1);
    Ltan=reshape(Ltan',1,(Ncg*12),1);
    dRtot=reshape(dRtot',1,(Ncg*12),1);
    R_cc2=reshape(R_cc2',1,(Ncg*12),1);
    Z_cc2=reshape(Z_cc2',1,(Ncg*12),1);
    BUA=reshape(BUA',1,(Ncg*12),1);
    BUA1=reshape(BUA1',1,(Ncg*12),1);
end

%amount of columns needed in the R and Z matrices
K=C(:,6)';
for i=1:size(C,1)
    if C(i,3)-C(i,6)==0
        K(i)=C(i,3)+L_c(i);
    end
end

%creating the R and Z matrices
R=zeros(size(C,1),ceil(max(K)));
Z=zeros(size(C,1),ceil(max(K)));

%R and Z coordinates of 1st and 2nd build sections and completion coordinates
R_cc1=ROC;
Z_cc1=KOPz;
Rc1e=ROC-ROC.*cosd(BUA1);
Zc1e=ROC.*sind(BUA1)+KOPz;
Rc2s=Ltan.*sind(BUA1)+Rc1e;
Zc2s=Ltan.*cosd(BUA1)+Zc1e;
m1=(Rc2s-Rc1e)./(Zc2s-Zc1e);
b1=Rc1e-m1.*Zc1e;
R_ce=dRtot+(sqrt((C(:,4)-C(:,1)).^2+(C(:,5)-C(:,2)).^2))';
Rc2e=dRtot;
Zc2e=C(:,3)';
m2=(R_ce-Rc2e)./(C(:,6)'-Zc2e);
b2=Rc2e-m2.*Zc2e;

%filling the Z matrix with numbers from 1 to depth of completion end
for i=1:size(C,1)
    for j=1:C(i,6)
    A=1:C(i,6);
    Z(i,j)=A(j);
    end
end
```

```
%filling the R matrix with the corresponding coordinates
N=zeros(1,size(C,1));
for j=1:size(C,1)
for i=1:C(j,6)
    %coordinates above the 1st kickoff point
    if Z(j,i)<=KOPz
        R(j,i)=0;
    %coordinates of the 1st build section
    elseif Z(j,i)>KOPz && Z(j,i)<=Zc1e(j)
        R(j,i)=-sqrt(ROC(j)^2-(Z(j,i)-Z_cc1)^2)+R_cc1(j);
    %coordinates of the tangent section
    elseif Z(j,i)>Zc1e(j) && Z(j,i)<=Zc2s(j)
        R(j,i)=m1(j)*Z(j,i)+b1(j);
    %coordinates of the 2nd build section
    elseif Z(j,i)>Zc2s(j) && Z(j,i)<C(j,3)
        R(j,i)=-sqrt(ROC2(j)^2-(Z(j,i)-Z_cc2(j))^2)+R_cc2(j);
        %S_wells
        if BUA(j)<90 && BUA1(j)>BUA(j)
            R(j,i)=sqrt(ROC2(j)^2-(Z(j,i)-Z_cc2(j))^2)+R_cc2(j);
        end
    %coordinates of the completion interval
    else
        R(j,i)=m2(j)*Z(j,i)+b2(j);
        N(j)=floor(C(j,6));
    end

    %coordinates of a horizontal completion interval
    if C(j,3)-C(j,6)==0
        B=dRtot(j):(dRtot(j)+L_c(j));
        R(j,C(j,3)+length(B))=dRtot(j)+L_c(j);
        N(j)=floor(C(j,3))+length(B);
                for k=1:length(B)
                    R(j,C(j,3)+k-1)=B(k);
                    Z(j,C(j,3)+k)=Z(j,C(j,3)+k-1);
                end
     end
end
end

%plotting all wells as a two-dimensional figure
for i=1:size(C,1)
    figure()
    plot(R(i,1:N(i)),flipud(Z(i,1:N(i))));
    set(gca,'XAxisLocation','top','YAxisLocation','left','ydir','reverse')
    axis equal
    title('Well path in the RZ-plane')
    xlabel('R (m)')
    ylabel('Z (m)')
    xlim([-50 (R(i,N(i))+50)])
    ylim([0 C(i,6)+50])
end

end
```

```
function plot_satellite(C,KOP,L_c,ROC,BUA,dR)
%plots the satellite wells in 2D

%amount of columns needed in the R and Z matrices
K=C(:,6)';
for i=1:size(C,1)
```

```
    if C(i,3)-C(i,6)==0
        K(i)=C(i,3)+L_c(i);
    end
end

%creating the R and Z matrices
R=zeros(size(C,1),ceil(max(K)));
Z=zeros(size(C,1),ceil(max(K)));

%R and Z coordinates of completion start (end of build) and completion end
R_cc1=ROC;
Z_cc1=KOP(:,3)';
Rc1e=ROC-ROC.*cosd(BUA);
Zc1e=ROC.*sind(BUA)+KOP(:,3)';
R_ce=dR+(sqrt((C(:,4)-C(:,1)).^2+(C(:,5)-C(:,2)).^2))';
m1=(R_ce-Rc1e)./(C(:,6)-C(:,3))';
b1=Rc1e-m1.*Zc1e;

%filling the Z matrix with numbers from 1 to depth of completion end
for i=1:size(C,1)
    for j=1:C(i,6)
        A=1:C(i,6);
        Z(i,j)=A(j);
    end
end

%filling the R matrix with the corresponding coordinates
N=zeros(1,size(C,1));
for j=1:size(C,1)
    for i=1:C(j,6)
        %coordinates above KOP
        if Z(j,i)<=KOP(j,3)
            R(j,i)=0;
        %coordinates of the build section
        elseif Z(j,i)>KOP(j,3) && Z(j,i)<=Zc1e(j)
            R(j,i)=-sqrt(ROC(j)^2-(Z(j,i)-Z_cc1(j))^2)+R_cc1(j);
        %coordinates of the completion interval
        else
            R(j,i)=m1(j)*Z(j,i)+b1(j);
            N(j)=floor(C(j,6));
        end

        %coordinates of a horizontal completion interval
        if C(j,3)-C(j,6)==0
            B=dR(j):(dR(j)+L_c(j));
            R(j,C(j,3)+length(B))=dR(j)+L_c(j);
            N(j)=floor(C(j,3))+length(B);
                for k=1:length(B)
                    R(j,C(j,3)+k-1)=B(k);
                    Z(j,C(j,3)+k)=Z(j,C(j,3)+k-1);
                end
        end
    end
end

%plotting all wells as a two-dimensional figure
for i=1:size(C,1)
    figure()
    plot(R(i,1:N(i)),flipud(Z(i,1:N(i))));
    set(gca,'XAxisLocation','top','YAxisLocation','left','ydir','reverse')
    axis equal
    title('Well path in the RZ-plane')
```

```
    xlabel('R (m)')
    ylabel('Z (m)')
    hold on
    xlim([-50 (R(i,N(i))+50)])
    ylim([0 C(i,6)+50])
end

end
```

Bibliography

Brechan, B. A., Corina, A. N., Gjersvik, T. B., Sangesland, S., and Skalle, P. (2016). *Drilling, Completion, Intervention and PA –design and operations [Unpublished].* Norwegian University of Science and Technology, Trondheim, Norway, 2nd edition.

Lillevik, E. and Standal, I. E. (2017). *Optimized Wellbore Trajectories.* Norwegian University of Science and Technology, Trondheim, Norway, 2nd edition.

Lorentzen, M. (2015). Kværner mot utlandet i kampen om ny plattformkontrakt. Retrieved from `https://e24.no/energi/equinor/statoil-sparer-hundrevis-av-millioner-paa-ny\-plattformloesning-naa-konkurrer-kvaerner-med-utlendinger-for-aa-vinne\-kontrakten/23582550`.

MathWorks, I. (2014). Traveling Salesman Problem:Solved-Based. [MATLAB]. Retrieved from `https://se.mathworks.com/help/optim/ug/travelling-salesman-problem.html`.

Stanko, M. E. W. (2017). Exercise set s01. Problem 3: Early NPV calculations for the Goliat field. [PDF file]. Retrieved from `http://folk.ntnu.no/stanko/Courses/TPG4230/2017/Exercises/Exercise_01/Exercise%2001_v1.pdf`.

Vanderbei, R. J. (2001). *Linear Programming: Foundations and Extensions.* Princeton University, Princeton, NJ, 2nd edition.