IDEA League

Master of Science in Applied Geophysics

Research Thesis

Coupling unsupervised segmentation in wells with automatic implicit modeling in a Bayesian framework

Tobias Giesgen

August 9, 2018

Coupling unsupervised segmentation in wells with automatic implicit modeling in a Bayesian framework

Master of Science Thesis

for the degree of Master of Science in Applied Geophysics at
Delft University of Technology
ETH Zürich
RWTH Aachen University
by

Tobias Giesgen

August 9, 2018

Department of Geoscience & Engineering · Delft University of Technology
Department of Earth Sciences · ETH Zürich
Faculty of Georesources and Material Engineering · RWTH Aachen University

Delft University of Technology

Printed in The Netherlands, Switzerland, Germany

IDEA LEAGUE
JOINT MASTER'S IN APPLIED GEOPHYSICS

Delft University of Technology, The Netherlands
ETH Zürich, Switzerland
RWTH Aachen, Germany

Dated: *August 9, 2018*

Committee Members:

Florian Wellmann

Cédric Schmelzbach

Supervisor(s):

Prof. Florian Wellmann

Miguel de la Varga

Dr. Hui Wang

Eidesstattliche Versicherung

Giesgen, Tobias | 318054

Name, Vorname | Matrikelnummer (freiwillige Angabe)

Ich versichere hiermit an Eides Statt, dass ich die vorliegende ~~Arbeit/Bachelorarbeit/~~ Masterarbeit* mit dem Titel

Coupling unsupervised segmentation in wells with automatic implicit modeling in a Bayesian framework

selbständig und ohne unzulässige fremde Hilfe erbracht habe. Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel benutzt. Für den Fall, dass die Arbeit zusätzlich auf einem Datenträger eingereicht wird, erkläre ich, dass die schriftliche und die elektronische Form vollständig übereinstimmen. Die Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Aachen, _____

Ort, Datum | Unterschrift

*Nichtzutreffendes bitte streichen

Belehrung:

§ 156 StGB: Falsche Versicherung an Eides Statt

Wer vor einer zur Abnahme einer Versicherung an Eides Statt zuständigen Behörde eine solche Versicherung falsch abgibt oder unter Berufung auf eine solche Versicherung falsch aussagt, wird mit Freiheitsstrafe bis zu drei Jahren oder mit Geldstrafe bestraft.

§ 161 StGB: Fahrlässiger Falscheid; fahrlässige falsche Versicherung an Eides Statt

(1) Wenn eine der in den §§ 154 bis 156 bezeichneten Handlungen aus Fahrlässigkeit begangen worden ist, so tritt Freiheitsstrafe bis zu einem Jahr oder Geldstrafe ein.

(2) Straflosigkeit tritt ein, wenn der Täter die falsche Angabe rechtzeitig berichtigt. Die Vorschriften des § 158 Abs. 2 und 3 gelten entsprechend.

Die vorstehende Belehrung habe ich zur Kenntnis genommen:

Aachen, _____

Ort, Datum | Unterschrift

Abstract

The automatic interpretation of well logs has been the focus of research, especially in oil and gas industry, for more than 50 years and, aside from that, benefits from the fast developments of machine learning algorithms during the recent decades. Moreover, Bayesian inference is increasingly utilized to model geological data, enabling the consideration of all available information and a quantification of uncertainties. In order to combine unsupervised segmentation of well data with 3D geological modeling, a fully automated approach to directly create three-dimensional structural models from raw well data is intended and, further, tested on synthetic data with different standard deviations.

For this purpose, unsupervised segmentation, which considers the statistical nature as well as the spatial correlation of the data, is combined with a zonation method that extracts interface information from clustered data by maximizing probabilities within continuous zones. This data is then screened to automatically obtain geological information and, furthermore, is inserted into the structural modeling algorithm, which is based on implicit potential-field interpolation while at the same time honoring the geological spatial continuity.

It is shown that unsupervised segmentation is capable of segmenting raw well logs and that the zonation appropriately determines boundaries between stratigraphic units. Model reconstruction demonstrates that the fully automated process is proficient at recovering several common subsurface structures. Moreover, the implementation of a three-dimensional model in the segmentation process, filling the empty space between boreholes, reduces uncertainties in the geological modeling routine. The combination of unsupervised segmentation and 3D geological modeling, resulting in a fully automated process, taking all available information into consideration, is found to be a suitable method in order to build structural geological modeling directly from raw well logs.

Acknowledgements

I would like to extend my sincere thanks to all the people contributed to this project. In particular I would like to express my deepest appreciation to Professor Florian Wellmann and Miguel de la Varga, the supervisors of this project, for the motivating discussions, practical suggestions and helpful advices in the process of this work. Their always very positive nature made working on complex topics much easier. I also wish to thank Dr. Hui Wang for inspiring discussions.
Moreover, I am deeply grateful to all the dedicated officials of the Joint Masters in Applied Geophysics. Their continuous commitment in the background of this M.Sc. programme offered by three of Europe's leading technical universities ensures a smooth and straightforward procedure during the last two years. I would like to emphasize in particular Kathrin Heinzmann, the mentor of the Geological Institute at RWTH Aachen University, who is always particularly committed to the benefits of the students.
Furthermore, I would like to offer my special thanks to all lecturers of the Applied Geophysics programme who offer the highest possible education and to all fellow students who accompanied me for the last two years.
Eventually, I would like to show my greatest appreciation to my family, who always supported me in the last 27 years. It is redundant to mention that I would not be standing at this point without you. I owe my deepest gratitude to my parents, Birgit and Ingolf, who made it possible for me to follow my interests and to receive the best possible education. I am very glad to call them my parents.

RWTH Aachen University
August 9, 2018

Tobias Giesgen

Table of Contents

List of Figures

List of Tables

Acronyms

MCMC Markov Chain Monte Carlo

HMRF Hidden Markov Random Field

MRF Markov Random Field

FGM Finite Gaussian Mixture

MAP Maximum a posteriori estimation

BIC Bayesian information criterion

EM expectation - maximization

GHMRF Gaussian Hidden Markov Random Field

WTMM wavelet transform modulus maxima

LPD log prior density

LMD log mixture density

SVN support-vector network

Chapter 1

Introduction

Information about physical properties of layers in the subsurface and especially structural geologic 3D models based on or supported by geophysical measurements in boreholes are essential for many industries (e.g. geothermics, oil & gas, reservoir engineering). The interpretation of these well log data is a complex process. It has engaged researchers as well as the industry equally for nearly a century. Since the Schlumberger brothers and Henri Doll have performed the first measurements in boreholes to characterize layers in the subsurface (1927), in terms of their mineral composition, texture and physical properties (Serra, 1983; Hilchie, 1990), the interpretation of well log data transformed from a knowledge- and men-intensive to a computational-intensive discipline (Hall, 2016; Shi et al., 2017). Recently, it benefits greatly from developments in data science.
One-dimensional data provided by the recordings in boreholes are often heterogeneous and/or noisy. Their interpretation via 3D modeling involves (i) a segmentation or the so called facies-classification of each data point, (ii) a zonation of each borehole and (iii) the actual geological modeling. Although, the definition and description of these processes varies in literature, in this work they are referred to as follows:

(i) Segmentation or facies-classification: assigning a label $l = 1, 2, 3, ..., L$ to each data point of the raw well logs

(ii) Zonation: division of data, raw or segmented, into as homogeneous as possible continuous intervals or zones also referred to as layers in a geological context

(iii) Geological modeling: creating a three-dimensional structural geologic model from zone-representative data

Considering well logs only from a few spatially distributed boreholes makes the interpretation straight-forward and can easily be performed by geo-scientists manually. Although the human brain is a good pattern recognizer, it is unable to take all the available information into consideration if the complexity and volume of the data are increased (Testerman et al., 1962; Hoyle, 1986; Hall, 2016). Thus, nowadays, efficient and powerful algorithms are widely

used to extract meaning from large well log data sets obtained from enormous projects.
The first automated statistical interpretation of univariate well log data goes back to Beghtol (1958), who introduced a zonation approach using variance maximization between layers in a borehole. This ansatz is later expanded considering additionally the variance minimum within zones (Testerman et al., 1962). Automatic search for changepoints in the data via window shifting and mean differences, which aims to find zone boundaries was introduced by Webster and Wong (1969). Afterwards it was extended based on the data's derivative to multivariate data by Webster (1973). The maximum-likelihood method by Hawkins (1976) uses a global optimizer to maximize the likelihood function of homogeneity within a layer by dynamic processing. It were Hawkins and Ten Krooden (1979) who stated that in case of huge data volume, global optimizer need to be approximated. Another zonation method based on changepoints was introduced by Lanning and Johnson (1983), who used low pass filtering via Walsh functions to discretize significant changes and determine zone boundaries. A comprehensive overview about the beginnings of automatic interpretation of well logs based on statistical criteria is given in Hoyle (1986).
The first approach using "Artificial intelligence"-techniques for (i) contact recognition and (ii) interval identification is provided by Wu and Nyland (1987). In 1989 Moghaddamjoo released a generalized segmentation system combining the prior knowledge (number of layers/zones) with a user-defined segmentation criterion (e.g. variance or means). To overcome the problem of revealing the multiscale behaviour of well logs, Vermeer and Alkemade (1992) constructed multiscale representations of the borehole measurements, where high frequencies unveil small scale behaviour and low frequency representations disclose global changes. Their approach was widely used in combination with changepoint search via extrema of the first derivative. Further methods were based on multivariate cluster analysis Gill et al. (1993) or wavelet transform of well data extracting extrema as layer boundaries (Hui et al., 2000).
The ability to segment well data by utilizing neural networks was stated by Rogers et al. (1992), who introduced such a network of nodes and their connections being able to learn from examples. This approach can be combined with the multiscale representation of well logs (Ouadfeul et al., 2011). Based on that, Ouadfeul and Aliouane (2012) combined self-organizing map neural network models and the multilayer perceptron for classification of borehole recordings. Another segmentation by Saucier and Muller (2002) extracts the log-generating function and refines it by several basis functions. Based on the analysis of the spectrum of time series by Dahlhaus et al. (1997), stationary intervals in well log data can be identified (Ligges et al., 2002). Combing several of the aforementioned segmentation and zonation techniques, Velis (2005) published a work identifying changepoints by probability density function analysis considering not only mean and variance, but also skewness and kurtosis in a window, that is moving along the signal. Ouadfeul (2006) introduced segmentation based on the sensitivity of wavelet transform modulus maxima (WTMM). Moreover, Ofuyah et al. (2014) adopted short time Fourier transformation converting the data from time to frequency domain for spectral analysis and zonation in spectral domain.
More recent approaches are using Markov Chain Monte Carlo (MCMC) methods in a Bayesian network to quantify uncertainties related to rock type recognition (Xu et al., 2016) or Hilbert-Huang transformation to measure the degree of heterogeneity within layers (Gaci, 2017), etc. Realizing that segmentation is an essential and widely discussed point of automatic well log interpretation, this work is based on recent developments in the field of unsupervised segmentation of n-dimensional soft data sets (Wang et al., 2017). The stochastic modeling approach is based on Hidden Markov Random Field (HMRF), which represents the "hidden link" in

spatially sparse geophysical data sets and can be considered as the heterogeneity of the subsurface. Furthermore, Finite Gaussian Mixture (FGM) models are utilized to characterize statistical parameters and Gibbs sampling enables the quantification of uncertainties in a Bayesian framework (Wang et al., 2017).
Additionally, De la Varga and Wellmann (2016) introduced a structural geologic modeling approach combining prior information with geologically motivated likelihood functions in a Bayesian framework. Their one-step forward modeling method based on implicit potential-field interpolation using cokriging (after Lajaunie et al., 1997; Calcagno et al., 2008) enables a direct recalculation of the model, when changing input parameters. Recently, de la Varga et al. (2018) presented GemPy, a fully open-source geomodeling package for the programming language Python, which uses these techniques to construct complex full 3D structural geologic models (de la Varga et al., 2018).
The main objective of this work is the coupling of the modeling approaches developed by Wang et al. (2017) and de la Varga et al. (2018) in one framework to benefit the interpretation of one-dimensional well log data in multiple fields of applications. Therefore, the 3D structural geologic modeling method is implemented into unsupervised segmentation of well-logs using Hidden Markov Random Field (HMRF) in one Bayesian network to create a 3D geological model directly from the raw well log data. The multi-dimensional segmentation approach (Wang et al., 2017) is therefore applied to a single well first and then to several wells independently in a second step. The spatial correlations between the resulting segmented boreholes are investigated and used to create a 3D structural geologic model (de la Varga et al., 2018). In the final step, the geological geometry of the geological model is considered during the segmentation process. This is achieved by performing the segmentation of all boreholes simultaneously, followed by a zonation of each borehole separately, based on different criteria. Using these zoned data as input for the structural modeling creates a 3D model containing additional information about the spatial correlation of the data. By extracting plains with constant depth and comparing each data point of the boreholes with its neighbouring ones (expanding neighborhood system of FGM model), an uncertainty about the layer affiliation can be extracted, assuming general geological continuity.
The main hypothesis of this work is that considering all available information resulting from different modeling approaches in one Bayesian framework will reduce uncertainties in the segmentation and, thus, interpretation of data provided by one-dimensional borehole measurements. The approach enables the consideration of several wells and keeps logical continuation in terms of geology over the entire model space. It has the potential to be the starting point of creating 3D structural geologic models from raw 1D well log data automatically and, furthermore, considering spatial 3D information in the segmentation and, thus, the interpretation of 1D well log data. The principal part of this thesis is divided into four sections. The first deals with the theoretical background of Bayesian networks, while the second section describes the methods utilized in this work. Moreover, it expounds the way of combining the modeling approaches. Third section shows the results of the segmentation and modeling processes, while the fourth and final section discusses the method of combination, the resulting segmentations and the geological models. It also reports on further advancements and potential fields of further research.

Chapter 2

Theory

In this chapter the theory that forms the basis of this work is presented. In Section 2-1 the basic concept of Bayesian statistics and inference in Bayesian networks is described. This includes the determination of the posterior from the prior, the likelihood and the evidence. How the posterior distribution can be obtained for complex problems, where a simple calculation is not directly possible is outlined in section 2-2. The Maximum a posteriori estimation (MAP) and the MCMC method are also introduced in the same.

2-1 Inference in Bayesian networks

The beginning of Bayesian statistics reaches back to more than 250 years to Thomas Bayes († 1761) and his interpretation of statistics gained a lot of attention in the last decades (Jaynes, 1986; Bolstad and Curran, 2016). Its basics are discussed in literature comprehensively (see Jaynes (1986); Congdon (2007); Berger (2013); Davidson-Pilon et al. (2015); Bolstad and Curran (2016) and Martin (2016)). But, probability does not always equal probability. A frequentist interprets probability as the long-run frequency of an event, which results from an often repeatable experiment. He always starts at zero without considering prior information. The Bayesian interpretation assumes probability as a quantity that measures the uncertainty level of an event or statement, taking prior knowledge into account, which can also result from a large number of repeatable experiments. The prior is then updated by new information about that event (Davidson-Pilon et al., 2015; Martin, 2016). This is an intuitive approach, since most situations in our everyday life are unpredictable. We deal with this uncertainty by plausible reasoning and base our decisions on the occurrence or non-occurrence of other events (Bolstad and Curran, 2016).
An example is making the decision of taking an umbrella with us in the morning or not. Even taking the weather forecast, the weather in the morning and the weather from yesterday into account, we can never be certain whether it will rain or not and, thus, if we will need an umbrella. But we can reduce the uncertainty about that event or statement by updating our belief (prior) by considering all possible information or data (Martin, 2016). The concept of

such a model that describes all events and their relationships is called Bayesian network. The following section depicts how to make decisions (reduce uncertainty) within such a network.

2-1-1 Fundamentals

Probabilities $p(\theta)$ are numbers between 0 and 1 including both extremes, which describe how likely an event or statement is. The probabilities of several different possible events are collected in a probability distribution (Martin, 2016). In a Bayesian network, inferences are made on an uncertain parameter set $\theta = (\theta_1, ..., \theta_d)$ of dimension d, which includes fixed and random effects, hierarchical parameters, unobserved indicator variables, and missing data (Gelman and Rubin, 1996).
The prior knowledge about the parameters are expressed through the prior distribution $p(\theta)$. The likelihood of parameters is called likelihood $p(y|\theta)$ and the probability of observing the data averaged over all values which can be taken by the parameter is named the evidence or marginal likelihood $p(y)$ (MacKay, 2003; Congdon, 2007; Martin, 2016). The probability distribution containing all knowledge about the parameters is the posterior distribution $p(\theta|y)$. It is also the result of the Bayesian analysis (Martin, 2016) and the next subsection states how it is obtained.

2-1-2 Bayes' theorem

The relationship between prior and posterior distribution is formulated in the well-known Bayes' theorem (Jaynes, 1986; MacKay, 2003; Congdon, 2007; Berger, 2013; Davidson-Pilon et al., 2015; Martin, 2016; Bolstad and Curran, 2016):

$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{p(y)}. \tag{2-1}$$

Assuming that the evidence or marginal likelihood $p(y)$ is constant and simply a normalization factor, Bayes' theorem can be written as a proportionality (MacKay, 2003; Congdon, 2007; Martin, 2016):

$$p(\theta|y) \propto p(y|\theta)p(\theta). \tag{2-2}$$

Equation 2-1 exhibits that the posterior distribution $p(\theta|y)$ is a balance of the prior distribution and the likelihood, or in other words the updated prior considering the given data y. It is worth to mention, that the likelihood $p(y|\theta)$ is not a probability distribution and depends on both, the data y and the parameters θ. In case of sequential available data, the posterior of one analysis can be the prior of a following, which is one of the advantages of Bayesian statistics. Also keeping in mind, that the posterior is not a single value, but a probability distribution of the parameters θ (MacKay, 2003; Martin, 2016). Due to the complexity of almost all practical cases, the calculation of the posterior is not as simple as equation 2-1 suggests and need to be approximated, which is explained in the next section.

2-2 Sampling the posterior

As mentioned above, the calculation of the posterior distribution $p(\theta|y)$ is possible for simple optimized Bayesian inference problems only, thus must be obtained via numerical methods in more complex cases, like structural geologic modeling. Depending on the purpose of the posterior model, different approaches can be used to estimate it: The MAP method, in case one is interested in the "best" model (e.g. to visualize the result) and the MCMC method resulting in "all" possible models and their corresponding uncertainties (De la Varga and Wellmann, 2016). The concepts of both approaches are presented in the following subsections.

2-2-1 MAP - Maximum a posteriori estimation

The Maximum a posteriori estimation (MAP) method uses the proportionality of the uncertainty space to likelihood $p(\theta|y)$ and prior $p(\theta)$ (see equation 2-2) and obtains only the global maximum of the uncertainty space instead of considering the whole space (Gauvain and Lee, 1994):

$$\hat{\theta}_{MAP}(y) = arg\,\underset{\theta}{max}\,[\,p(\theta|y)\,] = arg\,\underset{\theta}{max}\,[\,p(y|\theta)p(\theta)\,] \tag{2-3}$$

De la Varga and Wellmann (2016) point out that due to the normalization factor's neglect, the method is computationally cheap compared to other posterior sampler, but results in one possible solution only. This solution does not necessarily match best the likelihood functions and the prior, depending on the posterior's shape.

2-2-2 MCMC - Markov Chain Monte Carlo method

The Markov Chain Monte Carlo (MCMC) method is an iterative algorithm to sample probability distributions, whose calculation is infeasible (Liang et al., 2011) or in other words, it approximates the posterior distribution of an unknown parameter. Gilks (2005) underlined its ability to evaluate the posterior distribution of complex models in Bayesian frameworks. It is based on the 2-step algorithm introduced by Metropolis et al. (1953), starting at any point x_0 and (1) propose a random perturbation of the current state x_t generated from a symmetric proposal distribution and (2) decide about acceptance or rejection of the new state x_{t+1}, based on comparison of how likely the states x_t and x_{t+1} are to describe the data, given the prior distribution (Liang et al., 2011).
Hastings (1970) improved the algorithm (to the so-called Metropolis-Hastings algorithm) to enable the usage of asymmetric proposal distributions for the generation of the new state, whose location is still random but in the "neighbourhood" of the current state. That on the other hand increases the propability of the algorithm getting trapped in local extrema. During the last decades, intense researches were made in the field of MCMC algorithms to overcome the local-trap problem and improve the algorithm in term of computational efficiency and posterior approximation (Liang et al., 2011).

2-3 Uncertainty quantification

Three-dimensional structural geologic modeling never recovers the true geological image of the subsurface and, thus, is always tied to uncertainties. Especially in a probabilistic programming framework are the quantification, analysis and visualization of these uncertainties essential. This enables in a logical next step to interpret and reduce the uncertainties to improve the modeling results (Wellmann et al., 2010; Wellmann and Regenauer-Lieb, 2012; Wellmann, 2013; de la Varga et al., 2018).
To measure the variation of the results and, thus, the variation in the posterior distribution, a common approach in uncertainty quantification is the concept of information entropy, which is also utilized in this work. Therefore, the probability $P_\ell(i)$ of assigning a class, layer or label $\ell \in L = \{1, 2, 3, ..., K\}$ to a voxel $i \in V = \{1, 2, 3, ..., N\}$, where K is the number of labels and N the number of voxels, is calculated with (Cover and Thomas, 2012; Wellmann and Regenauer-Lieb, 2012; Wellmann, 2013):

$$P_\ell(X_i) = \frac{1}{n} \sum_{k=1}^{n} \vec{I}_\ell(x_i^k) \tag{2-4}$$

where n is the number of realizations and $\vec{I}_\ell(x_i^k)$ is an indicator function for the certain label ℓ, which is either one for $x_i = \ell$ or zero for $x_i = \ell$. This probability could be visualized for each label ℓ individually, but in this work the concept of information entropy based on Shannon (1948) is utilized, which enables uncertainty evaluation via a single parameter (Cover and Thomas, 2012):

$$H(X_i) = -\sum_{\ell \in L} P_\ell(X_i)\, log(P_\ell(X_i)) \tag{2-5}$$

Here, $H(X_i)$ is the information entropy, which in other words represents the number of results and their relative probability at each voxel. This concept is common in research to visualize and analyse uncertainties. It can be visualized over the whole model and low values indicates sufficient information and a reliable modeling result (Cover and Thomas, 2012; Wellmann and Regenauer-Lieb, 2012; Wellmann, 2013).

Chapter 3

Methods

As described in the introduction, the interpretation of one-dimensional well log data through 3D models includes a segmentation, zonation and structural modeling of the raw data. The following chapter describes the methods used and applied to the data for executing these tasks. Preliminary to the segmentation, the well logs need to be standardized and the number of segments is estimated. This is ensured by a preprocessing described in section 3-1. Section 3-2 presents BaySeg, an unsupervised segmentation approach for n-dimensional data (introduced by Wang et al., 2017),which is used for the segmentation. Subsequently, section 3-3 outlines two methods for the zonation of the segmented well log data, one minimizing the variance within zones and another considering the extrema of the segmentation uncertainty. In the next section 3-4 the concepts and methods of GemPy (introduced by de la Varga et al., 2018) are presented, which is used for the 3D structural geologic modeling. Finally, the combination approaches of all these methods and, thus, how to create 3D geological models directly and automatically from one-dimensional well logs is explained in the last section of this chapter.

3-1 Preprocessing

The raw data preprocessing is suggested to ensure equally weighted well logs from different measurement devices during the segmentation process. It is also applied to estimate the number of segments or classes if it is unknown prior to the analysis.

3-1-1 Standardization

Well log data from different measurement techniques (e.g. gamma ray, electric induction or resistivity, photoelectric effect) can differ in absolute values by factors of up to 10 or even 10^2 (Ellis and Singer, 2007). Thus, the raw well data x_i are standardized before being segmented to ensure an equal weighting of each method. This preprocessing step is not imperative, but improves the performance of the segmentation process in terms of robustness. Therefore, the standardized score or so-called Z-score of each value is calculated by (Yamane, 1973):

$$Z = \frac{x - \mu}{\sigma} \quad \text{with} \quad Z \in [-3, 3]. \tag{3-1}$$

In equation 3-1, Z is the Z-score of a single data point x, μ represents the mean of x_i and σ its standard deviation. Assuming normally distributed original data x_i, the Z-score ranges from -3 to 3 and measures how many standard deviation a value x differs from the mean.

3-1-2 Bayesian information criterion

An important step and major problem of unsupervised data segmentation is the determination of the number of segments/clusters, if this information is unknown beforehand, which is often the case for geophysical data provided by borehole measurements. A reasonable choice of the segment number ensures low variations within segments and minimizes the similarities between them (Guo et al., 2002). Beside using geoscientist's expertise, there are several approaches to determine the number based on the data. One of them is the Bayesian information criterion (BIC), which is utilized in this work and it is given by (Findley, 1991):

$$BIC = 2\ln(\hat{L}) - k\ln(n) \quad \text{with} \quad \hat{L} = p(x|\hat{\theta}, M) \tag{3-2}$$

where $\hat{L}$ is the maximized likelihood of the model M given the observed data x and the parameters $\hat{\theta}$ that maximizes the likelihood function. Furthermore, k represents the number of unknown parameters and n of data points in the observed data x. To perform the segment number analysis, the BIC is calculated for all values up to a reasonable user-defined bound, where the global minimum of the BIC defines the number of segments. Due to the fact that the upper boundary of the BIC affects its result, the analysis is made several times with increasing upper boundaries and the most common outcome defines the final number of segments. How the BIC performs for normally distributed synthetic well data is investigated in section 4-1-1. Nevertheless, the crucial factor of an effective segment number analysis is the data itself and its diversity.

3-2 BaySeg - Stochastic geological modeling via unsupervised segmentation

In computational geoscience, stochastic modeling of multiple geophysical data is a widely-used method to extract the subsurface's heterogeneity and provide insights into its structure (Wang et al., 2017). In this subject Wang et al. (2017) introduced *BaySeg*, a segmentation approach of n-dimensional geophysical data, which utilizes Hidden Markov Random Field models and Finite Gaussian Mixture models to "learn" the underlying spatial correlation between the spatial data. Additionally, an MCMC algorithm is applied to explore the posterior distribution via Gibbs sampling.
The FGM model is a common segmentation method, but it suffers with noisy and hardly-separable data due to its assumption of spatially independent data points in feature space. This is often the case for geophysical data, for example well logs from borehole measurements, which exhibit a high noise level and, thus, classification with FGM is not sufficient. To take the spatial correlation into consideration, HMRF models are developed, which make use of a neighborhood system allocating an additional likelihood to data points belonging to the same class as their neighboring ones (Wang et al., 2017). The following section outlines the basic concepts, the FGM model, the HMRF method as well as the segmentation process.

Stochastic modeling methods and uncertainty quantification are important tools for gaining insight into the geological variability of subsurface structures.

3-2-1 Fundamental concepts

Discretization

The three-dimensional physical space in the concept of Wang et al. (2017) is represented by a feature space, which is then investigated via unsupervised segmentation to discover its intrinsic statistical structure. This feature space is spanned by data points, of which each represents a voxel of the discretized "real" space with corresponding voxel features. Here, the voxel size depends on the resolution of the geophysical data, i.e. the density of measurement locations in a plain for 2D data or in a volume for 3D data.

Finite Gaussian Mixture model

To understand and apply the concept of Finite Gaussian Mixture (FGM) models lets assume that $p(\vec{y})$ is a random field in d-dimensional feature space $\mathbb{R}^d$, where the vector $\vec{y} = (y_1, y_2, y_3, ..., y_N)$ represents the properties to realize this particular field. Furthermore, $L = \{1, 2, 3, ..., K\}$ is a set of labels and all voxels j are assigned to one of these labels ℓ, then the conditional probability of the local properties y_j being observed at the point x_j, which belongs to label ℓ is given by (Wang et al., 2017):

$$p(y_j|x_j = \ell) = f(y_j; \theta_\ell) \quad \text{with} \quad \ell \in L. \tag{3-3}$$

Here, $f(y_j;\theta_\ell)$ is the distribution function of the data points in feature space, while θ_ℓ represents a set of distribution parameters corresponding to label ℓ. Assuming a voxel-location independent probability $P(x_j = \ell) = \alpha_\ell$ of a voxel j falling into a certain label ℓ, the marginal distribution $p(y_j)$ can be calculated by

$$p(y_j) = \sum_{\ell \in L} P(x_j = \ell)\ p(y_j|x_j = \ell) = \sum_{\ell \in L} \alpha_\ell\ f(y_j;\theta_\ell) \overset{*}{=} \underbrace{\sum_{\ell \in L} \alpha_\ell\ f(y_j;(\mu_\ell,\Sigma_\ell))}_{FGM\ model}. \tag{3-4}$$

The last step in equation 3-4 (marked by $*$) is due to the distribution function $f(y_j;\theta_\ell)$ being a multivariate Gaussian distribution with $\theta_\ell = (\mu_\ell, \Sigma_\ell)$, where μ_ℓ is a vector containing the mean of every feature and Σ_ℓ represents the covariance structure of all the features belonging to class ℓ (Wang et al., 2017). For a definition of the covariance structure see Celeux and Govaert (1995). While an FGM model describes data only statistically in feature space without taking their physical location or position to each other into consideration, Markov Random Field (MRF) models and in particular, HMRF models are able to do both, correlate observed data in physical space and evaluate them in feature space. These concepts are described in the following sections.

Graph structure and neighborhood system

For the representation of the discretized three-dimensional physical space's topology, Wang et al. (2017) adopt a graph modeling approach. The graph $G = (V, E)$ is characterized by a set of vertices $V\{i|i = 1, 2, 3, ..., N\}$ and a set of edges $E\{i, j\}$ satisfying $i, j \in V$ and $i \neq j$. Each vertex stands for a corresponding voxel and its label ℓ_i, while the edges donate the relation between each voxel and all other voxels given the labels of its neighbors. Each edge $\{i, j\}$ holds one orientation vector pair $(\Psi_{i,j}, \Psi_{j,i})$, which indicates the opposed orientations of the edge-connected vertices $(V_{ij} \longleftrightarrow V_{ji})$ (Wang et al., 2017).
If two voxels are connected about one of their eight nodes in the three-dimensional grid, their corresponding vertices are neighbors and linked with an edge in the graph G. All neighbors of a voxel i build a local neighborhood system $\{\partial_i|i \in V\}$:

$$\partial_i = \{j \mid \{i, j\} \in E,\ j \in V\}. \tag{3-5}$$

An MRF model is constructed on the graph G (Wang et al., 2017) and its concept is presented in the next section.

3-2-2 Markov Random Field and Gibbs distribution

A Markov Random Field (MRF) is a statistical model that describes undirected relations in a system of random variables. One can image a field consisting of cells that contain random variables that interact with each other in a limited space. A simple example is displayed in figure 3-1 (Kindermann and Snell, 1980). In a graphical representation of the Markov Random Field (MRF) like the graph form described in the last section, the set of all

random fields $\Omega = \{\vec{x} = \{x_i\}, i \in V, x_i \in L\}$ contains all possible segmentation configurations $\vec{x} = \{x_i\}, i \in V, x_i \in L$, wherein x_i indicates the segmentation result at vertex i. Any random field $\vec{x} = \{x_i\}, i \in V, x_i \in L$ is an MRF with respect to the neighborhood system iff

$$P(\vec{x}) > 0 \;\; for\ all\ \vec{x} \in \Omega\,, \tag{3-6}$$

where $P(\vec{x})$ represents the probability of the random field $\vec{x}$ and its local characteristics are given by (Wang et al., 2017):

$$P(x_s|x_r, r \neq s) = P(x_s|x_r, r \in \partial_s)\,. \tag{3-7}$$

The Hammersley-Clifford theorem introduced by Hammersley and Clifford (1971) and restated in chapter 3 of Besag (1974) establishes MRF-Gibbs equivalence, which provides the Gibbs random field equivalent $\pi(\vec{x})$ to the Markov Random Field (MRF). This includes an explicit formulation for the probability $P(\vec{x})$ in equation 3-7 depending on the energy function $U(\vec{x})$. The Gibbs distribution in relation to the neighborhood system $\{\partial_i | i \in V\}$ reads (Wang et al., 2017):

$$\pi(\vec{x}) = \frac{1}{Z}\ \exp(-U(\vec{x})) \quad \text{with } Z = \sum_{\vec{x}\in\Omega} \exp(-U(\vec{x}))\,. \tag{3-8}$$

In equation 3-8, $\pi(\vec{x})$ is the probability of segmentation result $\vec{x}$ and Z is referred to as the partition function. The energy function $U(\vec{x})$ takes the form of:

$$U(\vec{x}) = \sum_{c\in C} V_c(\vec{x}) \tag{3-9}$$

where C is a set of c, while c denotes a so called clique (subset of all vertices V within which all vertices are neighbors) and $V_c(\vec{x})$ its corresponding potential function depending on the labels of the neighborhood system (Wang et al., 2017).
Due to the complexity of the partition function's Z computation, an expression of the conditional probability $P(x_j|\vec{x}_{\partial_j})$ provided by the aforementioned Hammersley-Clifford theorem (Besag, 1986) is adopted:

$$P(x_j|\vec{x}_{\partial_j}) = \frac{P(x_j, \vec{x}_{\partial_j})}{\sum_{x'_j\in L} P(x'_j, \vec{x}_{\partial_j})} = \frac{\exp[-U(x_j, \vec{x}_{\partial_j})]}{\sum_{x'_j\in L} \exp[-U(x'_j, \vec{x}_{\partial_j})]} \tag{3-10}$$

The formulation of $P(x_j|\vec{x}_{\partial_j})$ in equation 3-10 is the crucial element of the MRF simulation via Gibbs sampler, but the local energy function $U(x_j, \vec{x}_{\partial_j})$, including the potential function $V_{i,j}(x_i, x_j)$ within a neighborhood system, still remains undefined. Wang et al. (2017) overcomes that by utilizing the Potts model (Koller and Friedman, 2009):

$$U(x_j, \vec{x}_{\partial_j}) = \underbrace{V_j(x_j)}_{\substack{\text{allows for preferred} \\ \text{label by voxel j}}} + \underbrace{\sum_{i\in\partial_j} V_{i,j}(x_i, x_j)}_{\substack{\text{allows for neigh-} \\ \text{boring labels}}} \tag{3-11}$$

with the isotropic potential function containing the granularity coefficient β:

$$V_{i,j}(x_i, x_j) = \begin{cases} 0 & if\ x_i = x_j \\ \beta & if\ x_i \neq x_j \end{cases} \tag{3-12}$$

Equation 3-12 shows that two neighbored voxels with the same label have zero potential, while different-labelled voxels have the potential β, which leads to a higher local energy $U(x_j, \vec{x}_{\partial_j})$ and, thus, a lower probability to assign this certain label to the voxel in question. This is an intuitive approach in geological modeling since a data point in the subsurface is more likely equal to his neighboring ones due to geological continuity. Additionally, Wang et al. (2017) introduced an anisotropic potential function $V_{i,j}^{aniso}(x_i, x_j)$, which takes the orientation and distances between voxel in a local neighborhood system into consideration. It is worth to say that the choice of the granularity coefficient β influences the resulting MRF model significantly. Its effect on the segmentation of one-dimensional well data is investigated in section 4-1-2.

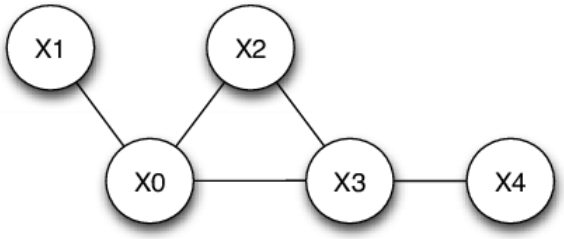

Figure 3-1: A simple example of a Markov Random Field (MRF) model with five random variables x_i connected omnidirectionally representing their relationship (from Liu et al., 2013)

3-2-3 Hidden Markov Random Field model

A Hidden Markov Random Field (HMRF) model is a statistical process with an underlying MRF and it is characterized by the following: (i) the label configuration $\vec{x}$ of the MRF is unobservable or "hidden" and (ii) the data's origin is assumed to be one certain label configuration $\vec{x}$ and a so-called emission probability function $f(y_i; \theta_{x_i})$. Furthermore, (iii) the assumption of pairwise independence is made (Chatzis and Tsechpenakis, 2010; Zhang et al., 2001):

$$P(\vec{y}|\vec{x}) = \prod_{j=1}^{s} P(y_i|x_i) \tag{3-13}$$

where s are all voxel locations and $P(\vec{y}|\vec{x})$ is the emitted field. Additionally, Wang et al. (2017) define the pairwise probability $P(x_j y_j|\vec{x}_{\partial_j})$ of a pair $(x_j, \vec{y}_j)$ with label x_j at voxel location j, given the data $\vec{y}_j$ and the neighborhood system $\vec{x}_{\partial_j}$:

$$P(x_j y_j|\vec{x}_{\partial_j}) = P(x_j|\vec{x}_{\partial_j}) P(y_j|x_j). \tag{3-14}$$

As mentioned before, the overall label configuration in the HMRF model is hidden, but the marginal probability of one data feature y_j reads:

$$P(y_j|\vec{x}_{\partial_j}) = \sum_{\ell \in L} P(\ell|\vec{x}_{\partial_j})\, f(y_j|\theta_j)\,, \tag{3-15}$$

where ℓ is a label of all labels L and $\theta_j = (\mu_\ell, \Sigma_\ell)$ are the parameters defining the multivariate Gaussian distribution as introduced in equation 3-4.

Equation 3-8, 3-9, 3-13 and 3-15 define the Gaussian Hidden Markov Random Field (GHMRF) model. The probability of the data $\vec{y}$ within this model is given by (Wang et al., 2017):

$$P(\vec{y}|\vec{\mu}, \vec{\Sigma}, \beta) = \sum_{\vec{x}} P(\vec{y}|\vec{x}, \vec{\mu}, \vec{\Sigma})\, \pi(\vec{x}|\beta) \tag{3-16}$$

where $\pi(\vec{x}|\beta)$ is the Gibbs probability, which is intractable and needs to be approximated with:

$$\tilde{\pi}(\vec{x}|\beta) = \prod_{j \in V} P(x_j|\tilde{\vec{x}}_{\partial_j}, \beta)\,. \tag{3-17}$$

Equation 3-17 is derived from the mean field-like approximation principle, which estimates many small components of complex stochastic models via a single averaged process, given a suitable choice of segmentation $\tilde{\vec{x}}$ and its local neighborhood system $\tilde{\vec{x}}_{\partial_j}$ (Celeux et al., 2003). Hence, the GHMRF model's probability can be rewritten as:

$$P(\vec{y}|\vec{\mu}, \vec{\Sigma}, \beta) \approx \prod_{j \in V} \sum_{\vec{x}} P(\vec{y}|\vec{x}, \theta_{x_j})\, P(x_j|\tilde{\vec{x}}_{\partial_j}, \beta) = \prod_{j \in V} P(y_j|\tilde{\vec{x}}_{\partial_j}, \vec{\mu}, \vec{\Sigma}, \beta) \tag{3-18}$$

The starting point of the segmentation as well as the numerical implementation of 3-18 resulting in the actual segmentation process is outlined in the next section.

3-2-4 Starting point: The initial configuration and the priors

The final results of the unsupervised segmentation approach is theoretically independent of the initial configuration $\vec{x}_0$ and the distribution parameters $\theta_0 = (\mu_0, \Sigma_0)$. Nevertheless, using a good estimation of the result as initial setting can reduce the burn-in period of the sampling significantly. Therefore, Wang et al. (2017) investigated the data sets $\vec{y}$ in an initial preprocessing step with an expectation - maximization (EM) algorithm coupled with a FGM model to obtain a rough estimation of $\vec{x}$ and θ. This is computationally efficient and robust as long as the EM algorithm converges. Furthermore, prior distributions of the parameters θ and β are initialized utilizing multivariate normal distributions. The Gibbs sampling in the actual segmentation process (illuminated in the next section) ensures a global search for the energy minimum in the parameter space even though the EM algorithm gets trapped in a local minimum (Wang et al., 2017).

3-2-5 Segmentation and parameter estimation

To perform the actual segmentation process, Wang et al. (2017) utilized a Bayesian method as described in section 2-1 and similar to equation 2-2 they stated the following relation between the posterior, prior and likelihood function L:

$$p(\vec{x}, \phi|\vec{y}) \;\propto\; p(\vec{x}, \phi)\; L(\vec{y}|\vec{x}, \phi)\,. \tag{3-19}$$

where $\vec{y}$ are the observed data, $\vec{x}$ the segmentation result (label configuration), which is unknown and to be determined just as the distribution parameter $\phi = (\mu, \Sigma, \beta)$. Given equation 3-19, the unknowns can be iteratively sampled in two steps using a Markov Chain Monte Carlo (MCMC) method (see 2-2-2) via two posterior distributions, $p(\vec{x}|\vec{y}, \phi)$ and $p(\phi|\vec{y}, \vec{x})$, respectively (Wang et al., 2017).

Step 1: Sample configurations $\vec{x}$ from posterior $p(\vec{x}|\vec{y}, \phi)$

The posterior $p(\vec{x}|\vec{y}, \phi)$ can be interpreted as the probability distribution of segmentation result $\vec{x}$ given the data $\vec{y}$ and the distribution parameters ϕ; and furthermore, it is a Gibbs distribution, whose calculation depends on the evaluation of the energy function $U'(\vec{x}) = U(\vec{x}) + U(\vec{y}|\vec{x}, \phi)$ and its local equivalent $U'_j(x_j, \vec{x}_{\partial_j})$ given by (Wang et al., 2017):

$$U'_j(x_j, \vec{x}_{\partial_j}) = \underbrace{U(x_j, \vec{x}_{\partial_j})}_{MRF\ energy} \;+\; \underbrace{U(y_j|x_j, \theta_{x_j})}_{likelihood\ energy} \tag{3-20}$$

The MRF energy is nothing but the number of unequally-labelled neighbors (for 1D well data either 0, 1 or 2) multiplied by the granularity coefficient β and assigned to each label ℓ at each data point (see equation 3-12). The likelihood energy on the right hand side of equation 3-20 can be determined using the local Bayes' Theorem for the conditional distribution:

$$\begin{aligned} p(x_j|y_j, \vec{x}_{\partial_j}, \theta_{x_j}) \propto \exp\Big(-U(x_j, \vec{x}_{\partial_j}) - U(y_j|x_j, \theta_{\partial_j})\Big) = \\ = \exp\Big(-U(x_j, \vec{x}_{\partial_j}) - \frac{1}{2}(x_j - \mu_{x_j})^T\, \Sigma_{x_j}^{-1}(x_j - \mu_{x_j}) + \frac{1}{2}\, log|\Sigma_{x_j}|\Big) \end{aligned} \tag{3-21}$$

Now, that the total energy function can be evaluated, equation 3-10 is applied to determine the probability of each label ℓ being assigned to each data point (voxel j). The sampling process is performed using a chromatic parallel Gibbs sampler, which utilizes a graph structure splitting all voxel in equally coloured subsets enabling a parallel sampling. In the case of one-dimensional well data, this results in parallel sampling of two voxels. The MRF and total energy as well as the probability distribution is then recalculated for the randomly drawn new candidates and compared to the previous probability distribution of these voxels to decide about acceptance or rejection. Afterwards, the next two voxels are drawn and compared to be rejected or not until the updated segmentation result $\vec{x}_{t+1}$ is found (Wang et al., 2017).

Step 2: Sample distribution parameters $\phi = \vec{\mu}, \vec{\Sigma}, \beta$ from posterior $p(\phi|\vec{y}, \vec{x})$

Simply put, this sampling step proposes new parameters ϕ, compares them to the previous ones in term of their likelihood and evaluates the acceptance of the new parameters. Therefore, the previously described Gibbs sampler is utilized again. For this purpose, Bayes' Theorem (see equation 2-2) is established for each parameter separately:

$$p(\vec{\mu}|\vec{y}, \vec{x}, \vec{\Sigma}, \beta) \propto p(\vec{\mu})\ L(\vec{y}|\vec{x}, \vec{\mu}, \vec{\Sigma}, \beta), \tag{3-22}$$

$$p(\vec{\Sigma}|\vec{y}, \vec{x}, \vec{\mu}, \beta) \propto p(\vec{\Sigma})\ L(\vec{y}|\vec{x}, \vec{\mu}, \vec{\Sigma}, \beta), \tag{3-23}$$

$$p(\beta|\vec{y}, \vec{x}, \vec{\mu}, \vec{\Sigma}) \propto p(\beta)\ L(\vec{y}|\vec{x}, \vec{\mu}, \vec{\Sigma}, \beta). \tag{3-24}$$

In the above equations, the likelihood function $L(\vec{y}|\vec{x}, \vec{\mu}, \vec{\Sigma}, \beta) = \prod_{j \in V} P(y_j|\tilde{\vec{x}}_{\partial_j}, \vec{\mu}, \vec{\Sigma}, \beta)$ is calculated using equation 3-18, in which the segmentation result $\vec{x}_{t+1}$ is utilized as mean field-like approximation $\tilde{\vec{x}}$ (Wang et al., 2017).
The procedure starts with calculating the component coefficient for each voxel, which is its probability distribution based on the neighborhood system only. Afterwards, a new β_{prop} and μ_{prop} are proposed, which are randomly disturbed versions of their predecessors, but user-defined length apart from them, referred in this work to as $jump - parameters$. For the sampling of Σ, it is represented by a label-wise eigenvalue decomposition:

$$\Sigma_\ell = \lambda_\ell\, D_\ell\, A_\ell\, D_\ell^T\ ; \tag{3-25}$$

where Σ_ℓ is the covariance matrix of label ℓ, λ_ℓ its volume and A_ℓ its shape, while D_ℓ is its orientation (Wang et al., 2017). The proposal of σ_{prop} is then nothing but an update of all these covariance matrix characteristics for each layer or label ℓ, using user-defined distance of the new shape and volume as well as a predefined angle for the rotation matrix. Now, that the new parameters ϕ_{prop} are drawn, they are to be compared to their present versions ϕ_{prev}.
Therefore, the log prior density (LPD), which allows for the prior probability distribution and the log mixture density (LMD), which takes the probability distribution of the segmentation based on the parameters ϕ and the neighborhood system into consideration, is calculated for both, the previous and the proposed parameters. Combining the LPD and LMD results in the target log likelihood for the proposed and previous parameter $log(p_t)_{prev/prop}$, which are evaluated using the ratio r:

$$r = \exp\left(log(p_t)_{prop} - log(p_t)_{prev}\right) \tag{3-26}$$

If r is greater than one (in other words, $log(p_t)_{prop}$ is greater than $log(p_t)_{prev}$) or r is greater than a random uniform distributed variable, the new samples of ϕ_{prop} are accepted, otherwise the next iteration is run with the previous parameters. It is to be noted that μ is updated first, then Σ and eventually β and each parameter update utilizes all previously updated parameters.

Once the updated parameters are obtained, they are stored and the procedure restarts with step 1. The above outlined algorithm runs user-defined times and yields in a probability distribution for each voxel j, which describes how likely each label ℓ is assigned to that voxel (fourth column in table 3-1). Moreover, a vector is extracted containing the most likely label for each data point (see MAP method in section 2-2-1), which can be used for further analysis or visualization (third column in table 3-1). Furthermore, once the final label configuration is obtained, uncertainties can be quantified and visualized using the concept of information entropy by Wellmann and Regenauer-Lieb (2012) as explained in section 2-3. The next section describes which zonation concepts are applied to the segmentation results to find continuous and maximal homogeneous layers.

Table 3-1: A simple example of the outcome of the segmentation algorithm $BaySeg$ applied to 15 data points from two different boreholes with the assumption of three labels (first & second column: input data; third column: most likely label; fourth column: probabilities of each label)

Borehole	**Data #**	**Label**	**Probabilites**		
			0	1	2
1	0	2	[0.10	0.10	0.80]
1	1	2	[0.06	0.02	0.92]
1	2	2	[0.00	0.11	0.89]
1	3	2	[0.00	0.00	1.00]
1	4	0	[0.77	0.02	0.21]
1	5	0	[0.89	0.01	0.10]
1	6	1	[0.09	0.91	0.00]
1	7	1	[0.11	0.86	0.03]
2	0	2	[0.11	0.06	0.83]
2	1	2	[0.01	0.01	0.98]
2	2	0	[0.65	0.06	0.29]
2	3	0	[0.73	0.09	0.18]
2	4	1	[0.08	0.85	0.07]
2	5	1	[0.13	0.84	0.03]
2	6	1	[0.03	0.97	0.00]

A parameter test and an accuracy estimation of the segmentation algorithm are performed in section 4-1-2 and 4-2-1. Furthermore, the results are compared to an support-vector network (SVN) segmentation approach suggested by Hall (2016). The basic principles of this method are outlined in appendix A-1.

3-3 Zonation of well data

The purpose of well log zonation is identifying homogeneous zones within a borehole and moreover, to characterize these with the aim to detect a correlation between data from different wells (Beghtol, 1958). As long as the amount of data is clearly arranged, the zonation is straight-forward and can be conducted manually by well data experts, but with increasing volume and complexity of the data the human brain becomes less able to handle all available information and, thus, statistical approaches are utilized to perform the zonation computationally (Testerman et al., 1962). Automatic statistical methods furthermore ensure an objective investigation of the data and the reproducibility of the outcome. Modern algorithms are capable of assigning a rock type automatically to each data point, but their purpose is a manual interpretation and correlation of several boreholes. Since the zonation applied in this work aims automatic 3D modelling afterwards, zonation approaches are utilized, which create continuous zones.
Zonation can be applied on the raw well log data itself or as in this work on the segmented data (*BaySeg* outcome), meaning that each data point is already assigned to a label or layer and the zonation aims to find the most likely way to split the data into a fixed number of zones. The number of zones equals the number of labels utilized for the segmentation, which is predefined or estimated via BIC (see section 3-1-2). In the following section, different approaches are explained to perform the zonation. It is worth to say that a statistical correlation between wells is never a guarantee for continuous geological layers (zones) in the subsurface (Testerman et al., 1962).

3-3-1 Minimizing the variance within zones

The first zonation approach applied in this work is based on the minimization of variances within zones introduced by Beghtol (1958) and generalised by Testerman et al. (1962). This method was initiated to be applied on a single set of raw data, but due to multiple data sets in this work the label of each data point resulting from the segmentation algorithm (third column in table 3-1) is treated as a characteristic value, whose variation can be statistically analysed, although it has no physical meaning. The variance var_ℓ of a zone ℓ has the following form (Agarwal, 2006):

$$var_\ell(\vec{x}) \;=\; \frac{1}{n_\ell}\sum_{i=1}^{n_\ell}(x_i - \bar{\vec{x}})^2\,, \tag{3-27}$$

where n_ℓ is the total number of data x_i within a zone or layer $\ell \in \{1, 2, 3, ..., L\}$ and $\bar{\vec{x}}$ its mean value. Due to the minimal effect on the result and the implementation in the python package *numpy*, the Bessel's correction is neglected. While the algorithm introduced by Testerman et al. (1962) first divides the data into two zones and then these zones into several zones until the number of user-defined layers is achieved, the algorithm applied in this work, does not fix any zone boundary, but considers all possible combinations. It is a function directly dependent on the number if layers L, where the total variance of all zone combinations is calculated. Therefore, the variances of all zones within one zone configuration are summed up and the minimum of all total variances var_t is determined for each borehole separately:

$$b_{config} \;=\; min\,(var_t = \sum_{\ell=1}^{L} var_\ell). \tag{3-28}$$

where b_{config} is is the zonation result with the lowest summed variances in all layer and zones, respectively. One weaknesses of this approach is its exponential increasing computation time with growing data volume and additionally, in case of very noisy data it divides the boreholes close to top and bottom. This results in a few very small zones with low variances and a huge zone in the middle with an extremely high variance. Being aware of the algorithm's weak points, it is still applied first in this work, as a solid starting point for the zonation and a comparison to further approaches. Moreover, this straightforward method is used to make the whole process of automatically 3D modeling from one-dimensional well log data running. Besides the aforementioned flaws, this approach does neither consider nor conserve uncertainties provided by the segmentation using *BaySeg*, thus a second approach described in the next section is utilized. The results and a short comparison of the zonation approaches are presented in section 4-2-2.

3-3-2 Probability maximization within zones

With the aim to take the probability distribution resulting from *BaySeg* into consideration, the second zonation approach applied in this work is based on the optimization of probability sums. Therefore, the label probabilities (fourth column in table 3-1) are separated borehole-wise and cumulatively summed up from top (surface) to borehole bottom. To compensate for the number of occurrences in the segmentation process of each layer these probabilities are normalized. The cumulative normalized probability $P_{cn}(x_j|\vec{x}_{\partial_j}) \in [0,1]$ is then given by:

$$P_{cn_m}(x_j|\vec{x}_{\partial_j}) \;=\; \frac{\overbrace{\sum_{i=0}^{m} P_m(x_j|\vec{x}_{\partial_j})}^{cumulative\ probability\ P_c(x_j|\vec{x}_{\partial_j})} \quad - P_{c_0}(x_j|\vec{x}_{\partial_j})}{P_{c_M}(x_j|\vec{x}_{\partial_j}) - P_{c_0}(x_j|\vec{x}_{\partial_j})} \tag{3-29}$$

where $m \in (1,2,3,...,M)$ is the location inside the borehole (data vector) and M the last location. $P_{c_0}(x_j|\vec{x}_{\partial_j})$ is the cumulative probability of the first borehole location (minimum), while $P_{c_M}(x_j|\vec{x}_{\partial_j})$ the one of the last location (maximum). The most likely position of the lower boundary b_ℓ of a layer ℓ is then the location where $P_{cn_\ell}(x_j|\vec{x}_{\partial_j})$ of layer $\ell \in (1,2,3,...,L)$ is maximal, while the cumulative normalized probabilities $P_{cn_k}(x_j|\vec{x}_{\partial_j})$ of all other layers $k \in (1,2,3,...,L)$ with $k \;\neq\; \ell$ is minimal:

$$b_\ell \;=\; max\,[(L-1)\,P_{cn_\ell}(x_j|\vec{x}_{\partial_j}) - \sum_{i=0}^{L-1} P_{cn_k,i}(x_j|\vec{x}_{\partial_j})]. \tag{3-30}$$

The determination of the lower boundary is then performed for all layers ℓ to find the best layer configuration and zonation, respectively. The zonation results of this method are outlined in section 4-2-2, where it is also compared to the previously described approach. The major

limitation of this zonation is the constraint to single occurrence of each layer. Although, the existence of two layers with similar properties is common in the subsurface, this simplification is kept for this work because it is one of the requirements of the geological modeling with *GemPy*, whose concepts and methods are presented in the next section. *GemPy* utilizes the estimated lower boundaries as interface data points. Further investigations on zonation and how to extract interface location data from the borehole segmentation are outlined in the discussion of this work.

3-4 GemPy - Geologic modeling as an inference problem

Reliable structural geological modeling based on local samples ("hard data") is a key aspect of several geoscientific questions and applications (e.g. geofluid movement or raw material investigations). While most powerful methods are highly cost intensive and obscure, de la Varga et al. (2018) introduced GemPy, a fully open-source geomodeling method implemented in the programming language Python. GemPy's accessibility of the source-code reveals the inner processes of the modeling and enables an extension of the code itself and/or coupling with other libraries and packages. It is based on an implicit potential-field approach making use of a CoKriging interpolation and combines prior information with likelihood functions (containing geological knowledge) in a Bayesian inference framework (De la Varga and Wellmann, 2016; de la Varga et al., 2018). GemPy's numerical implementation is based on Theano and utilizes several more efficient packages (e.g. NumPy, PyMC3, pandas). Furthermore, *GemPy* allows an uncertainty quantification and visualization as described in section 2-3 compressed in a single parameter using the concept of information entropy after Wellmann and Regenauer-Lieb (2012). In the following section the theory of structural geological modeling as an inference problem, the underlying potential-field approach and the numerical implementation in Python are described.

3-4-1 Bayes' theorem in the context of geological modeling

De la Varga and Wellmann (2016) consider geological modeling as a Bayesian inference problem. Bayes' theorem and the basics of Bayesian statistics are introduced in section 2-1, whose components need to be specified in the context of geological modeling:

Mathematical forward model M: The link between parameters and observed data is described by a mathematical model, which is a direct function of the input parameters in case of structural geologic models (De la Varga and Wellmann, 2016; Wellmann et al., 2017):

$$M = f(\vec{x}, \phi_i, k_j, \alpha_k, \beta_\ell), \tag{3-31}$$

where ϕ_i represents the mathematical forward model, based on interpolation functions and depending on the position $\vec{x}$. Primary information or data such as surface contact point and orientation information are considered as k_j and additional parameters of the interpolation function as α_k. Moreover, the topological description is referred to as β_ℓ. Considering the geological model as a direct function of the input parameters ensures complete automation of the modelling step and therefore, the model can be computed instantaneously when changing input parameters (Wellmann et al., 2017).

Model parameter θ: These are either deterministic (exact value) or stochastic (probability distribution) parameters define the mathematical model ϕ_i. In the context of geological modeling these can be $\vec{x}, k_j, \alpha_k$ or β_ℓ (see equation 3-31).

Observed data y: These are any kind of further information, which can be linked or compared to the geological model. These auxiliary data can be derived from measurements (e.g.

seismic data) or from geological expertise (e.g. geometrical constrains) (De la Varga and Wellmann, 2016; Wellmann et al., 2017).

Likelihood functions $p(y|\theta, M)$: These are the likelihood of the model parameter given the data, or in other words, they describe the relation between the parameter and the data (Martin, 2016). Except the fact that they depend on the data y and not on the parameters θ, they are mathematically equivalent to probability density functions (Patil et al., 2010).

3-4-2 Inference process

Gelman et al. (2013) presented a work flow for solving complex Bayesian inference problems with large uncertainty spaces, which is not as simple as Bayes' theorem may suggest. This sequence is adapted by De la Varga and Wellmann (2016) for the propose of structural geologic modeling:

1. Probability model setup (prior): All model parameters θ define a multidimensional environment, a kind of joint probability space. A 2D example of such an environment defined by two random parameter is displayed in figure 3-2.

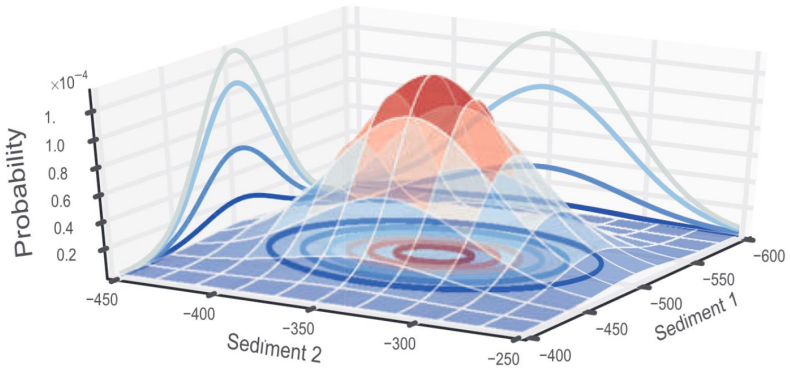

Figure 3-2: Two-dimensional joint probability space created by two random parameters θ resulting from probability density function describing the location of Sediment 1 (distribution on left backplane) and Sediment 2 (right backplane) (from De la Varga and Wellmann, 2016)

2. Consideration of observed data y: Next, all conditional probabilities need to be set to apply equation 2-1 and calculate the posterior distribution given the likelihood of the parameters in light of the data $p(y|\theta, M)$. Therefore, the parameters of the probability model must be related to the likelihood functions of the data y by deterministic functions of the model M introduced in equation 3-31. De la Varga and Wellmann (2016) alluded to the fact, that there is not necessarily a relation between all parameters and all data, while any combination is possible.

3. Posterior sampling: Due to the complexity of almost all practical cases, the calculation posterior is not as simple as Bayes' theorem (equation 2-1) suggests. There exist a

variety of numerical sampling methods to obtain the posterior, of which two are used in this work and explained previously in section 2-2.

4. Postprocessing posterior analysis: For the evaluation of the posterior model, De la Varga and Wellmann (2016) included two approaches: (1) the analysis of the parameters θ via Gaussian kernel density estimation and (2) a information entropy measurement on all obtained models within the posterior to illustrate uncertainties (Shannon, 1948; Wellmann, 2013).

3-4-3 The GemPy basis - potential field method

The crucial process of the 3D model generation in *GemPy* as described in equation 3-31 is a potential field approach developed by Lajaunie et al. (1997). Its concept is built on a global interpolation function $Z(x_0)$ with $x_0(x, y, z) \in \mathbb{R}^3$. The continuous space (x, y, z) is characterized by a scalar field, while the scalar field value is dimensionless and has no physical or chronological meaning. Isovalue surfaces of the field represent synchronously deposited sediments of one layer and the scalar field's gradient is directed parallel to the change in physical properties of the subsurface, thus, perpendicular to the isovalue surfaces. Figure 3-3 shows an example of such a global interpolated scalar field.
Interpolating the whole space instead of each surface of interest (here: geological layer) comes with two advantages: On the one hand, the entire interpolated scalar field, including parts between layers of interest can be evaluated and used for further analysis or the interpolation. On the other hand, the approach ensures geological continuity, meaning that two layers can never cross (de la Varga et al., 2018). To obtain a geological 3D model from the given sparse data, a suitable choice of the interpolation method resulting in the continuous scalar field is essential, which is explained in the next section.

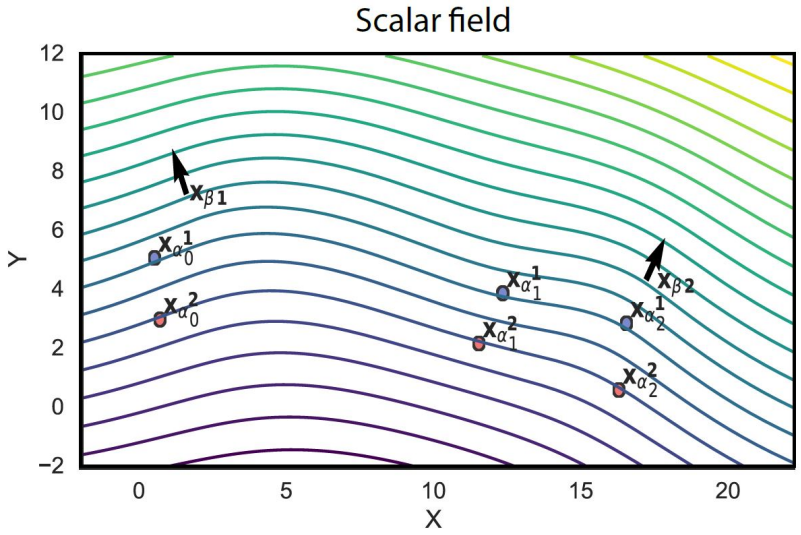

Figure 3-3: Scalar field interpolated from six interface data points of two different layers (layer 1 = blue dots: $x_{\alpha i}^1$ and layer 2 = red dots: $x_{\alpha i}^2$), which are connected by isovalue surfaces and two orientation information $x_{\beta i}$ representing the gradient of the potential field (black arrows). (from de la Varga et al., 2018)

3-4-4 Structural geological forward modeling

CoKriging - from data to scalar field

The forward modeling, described in equation 3-31 is the central component of structural geologic modeling. In *GemPy*, the application of a suitable interpolation method ϕ_i to obtain the global scalar field is of particular importance because it enables a direct and automatic model update when changing sensitive input parameter (De la Varga and Wellmann, 2016). Therefore, de la Varga et al. (2018) achieve the interpolation function $Z(x_0)$ via Universal CoKriging (after Chilès and Delfiner, 2009). Kriging is closely related to regression analysis and results in a unbiased linear predictor (random function) for the scalar value by minimizing the covariance function. The advantage of CoKriging in geological applications is the consideration of data from multiple locations and the preface "Universal" comes from the usage of polynomial drift functions, which take the linear behaviour of layer thickness into account.
In the following, the procedure of Universal CoKriging to obtain the global interpolation function $Z(x_0)$ for the scalar field from the data is expounded. Keep in mind, that the scalar field value does not has any physical meaning, but simply represents the layer affiliation. Therefore, the exact value is irrelevant, but constant within each layer and instead of the value itself, the difference of each point relative to a reference point is considered. According to de la Varga et al. (2018) this yields:

$$Z(x^k_{\alpha i}) - Z(x^k_{\alpha 0}) \;=\; 0 \tag{3-32}$$

where k describes the layer affiliation. The Kriging input data can either be (i) layer interface points (x_α in figure 3-3), which describe the isovalue interfaces or (ii) orientation data (x_β in figure 3-3) representing the gradients of the scalar field. Latter ones are perpendicular to the isovalue surfaces and mathematically normal vectors to the dip plane. Each of the input data is interpolated with its own random function, Z_α and $\frac{\partial Z}{\partial u}$, respectively. Their relationship is given by (de la Varga et al., 2018):

$$\frac{\partial Z}{\partial u}(x) \;=\; \lim_{\rho \to 0} \frac{Z(x+\rho u) - Z(x)}{\rho}, \tag{3-33}$$

which is used to relate the scalar field and its gradient in the cross-covariance matrices. Given the information above, the potential field estimator depends on a term allowing for the potential field difference and one representing the orientation information and is evaluated via (De la Varga and Wellmann, 2016):

$$Z(\vec{x}) - Z(\vec{x}_0) = \underbrace{\sum_{M}^{\alpha=1} \mu_\alpha (Z(\vec{x}_\alpha) - Z(\vec{x}'_\alpha))}_{\substack{\text{potential difference} \\ \text{(interface data)}}} + \underbrace{\sum_{N}^{\beta=1} \nu_\beta \frac{\partial Z}{\partial u_\beta}(\vec{x}_\beta)}_{\substack{\text{potential gradient} \\ \text{(orientation data)}}}, \tag{3-34}$$

where M and N are the total number of interface points and orientation data, respectively. μ_α and ν_β represent weighting factors. As mentioned earlier, Kriging aims to minimize the

covariance function resulting in the best unbiased interpolator. Therefore, de la Varga et al. (2018) propose a Universal CoKriging system in the following form:

$$\begin{bmatrix} C_{\frac{\partial Z}{\partial u},\frac{\partial Z}{\partial v}} & C_{\frac{\partial Z}{\partial u},Z} & U_{\frac{\partial Z}{\partial u}} \\ C_{Z,\frac{\partial Z}{\partial u}} & C_{Z,Z} & U_Z \\ U'_{\frac{\partial Z}{\partial u}} & U'_Z & 0 \end{bmatrix} \underbrace{\begin{bmatrix} \lambda_{\frac{\partial Z}{\partial u},\frac{\partial Z}{\partial v}} & \lambda_{\frac{\partial Z}{\partial u},Z} \\ \lambda_{Z,\frac{\partial Z}{\partial u}} & \lambda_{Z,Z} \\ \mu_{\partial u} & \mu_u \end{bmatrix}}_{weights\ vector} = \begin{bmatrix} c_{\frac{\partial Z}{\partial u},\frac{\partial Z}{\partial v}} & c_{\frac{\partial Z}{\partial u},Z} \\ c_{Z,\frac{\partial Z}{\partial u}} & c_{Z,Z} \\ f_{10} & f_{20} \end{bmatrix}. \tag{3-35}$$

where $C_{\frac{\partial Z}{\partial u}}$ in the left hand side vector is the gradient covariance-matrix, $C_{Z,Z}$ the covariance-matrix and $C_{Z,\frac{\partial Z}{\partial u}}$ the cross-covariance function. The drift functions and their gradients are represented by U_Z and $U_{\frac{\partial Z}{\partial u}}$, respectively, which are indicated in the vector on the right hand side by f_{10} and f_{20}. Furthermore, $c_{\frac{\partial Z}{\partial u},\frac{\partial Z}{\partial v}}$ is the covariance function's gradient; $c_{\frac{\partial Z}{\partial u},Z}$ and $c_{Z,\frac{\partial Z}{\partial u}}$ are the cross-covariance functions and $c_{Z,Z}$ the general covariance function. Finally, the unknown vectors are encapsulated in the weight vectors by weights λ and constants of the drift functions μ. Due to its dependence on the gradient at least one scalar field gradient (geological orientation data) is required to solve the system of equations.
Equation 3-35 can be solved for both, the scalar value field Z and its gradient $\frac{\partial Z}{\partial u}$. While Z is used for the segmentation described in the next section, the gradient of the scalar field $\frac{\partial Z}{\partial u}$ can be utilized for further analysis (de la Varga et al., 2018). It is worth to say, that *GemPy* is able to model unconformities and fault systems by combining their scalar fields, but due to missing fault and unconformity information provided by one-dimensional well data these features are not used in this work (for more details see chapter 2.2.2 in de la Varga et al. (2018)).

Segmentation - from scalar field to 3D geological model

The previous section captured how to obtain the interpolation function $Z(x)$, which allows the calculation of the potential or scalar field value at any point in space (x, y, z). Now, *GemPy* utilizes two different approaches to segment the space and create a structural geologic 3D model. The simplest way is discretizing the space and calculating the scalar field vector for each point of the mesh grid. Then, by comparison points with equal or similar values can be assigned to a layer of interest (de la Varga et al., 2018). A two-dimensional section of a generic example is shown in figure 3-4.
A further segmentation method in *GemPy* based on layer's isosurface location utilizes the marching cube algorithm after Lorensen and Cline (1987), where the space is discretized in 3D voxels. By interpolating the values at corners of the voxels, in which the value of the isosurface of interest appears, the intersections between voxel edges and isovalue surfaces can be obtained. Finally, these intersections are used as vertices to build simplices, which can be used in the 3D visualization.

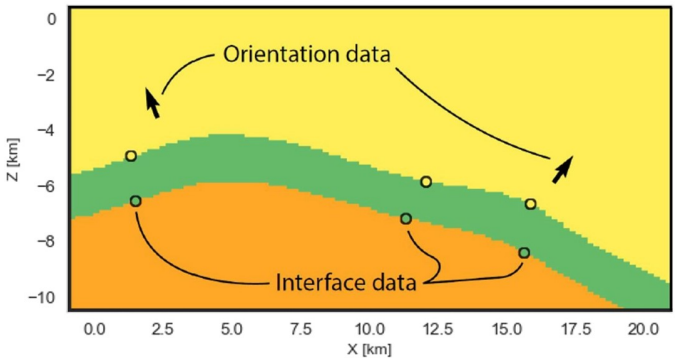

Figure 3-4: Generic example of a single stratigraphic group extracted from a structural geological 3D model containing 3 layers (yellow, green, orange) created with *GemPy*; interface data are represented by dots, while orientation data are represented by black arrows (modified after de la Varga et al., 2018)

3-4-5 Numerical implementation into probabilistic programming framework

The numerical implementation of *GemPy* is based on symbolic automatic differentiation using the Python package *Theano* and its code structure or design aims a maximum re-usability and modularity. Therefore, the code is divided in independent functions and modules, containing a single logical step or calculation and a whole concept (like *datamangement*) including several functions, respectively, which avoids duplications, simplifies modifications and increases readability.
The automatic differentiation requires symbolic coding handled here by *Theano*, which is an efficient and fast solver of algebraic equations and particularly its derivatives. It creates an acyclic graph, similar to the MRF model outlined in section 3-2-2, where notches represent parameters and their connection (edges) describe the relation between the parameters (mathematical operations). Each method applied to the data is related to a specific part of the graph, which is finally analysed and evaluated to perform tasks like optimization and derivative computation (more details about *Theano* can be found in Bergstra et al., 2010). Note that *Theano* not only takes over and fastens calculations and optimizations by code compiling into more quickly programming languages like C, but particularly performs the interpolation of the scalar field (geological modeling).
Additionally, the data management in *GemPy*, which needs to be done before the graph construction, makes use of the Python package *pandas* for data manipulation and analysis (for more details see McKinney (2011)). This enables an efficient data storage and preparation before the symbolic graph construction and, thus, the data import in *Theano*, which safes computation time in the actual modeling part. The results of the geological modeling are displayed and discussed in section 4-2-3. The next section describes the methods used to combine the concepts outlined in the previous sections.

3-5 Concatenation and merger of BaySeg and GemPy

The importance of a reliable segmentation and zonation of well log data to interpret and extract meaning from them is expounded comprehensively in the Introduction. The main objective of this work is the coupling of the concepts of *GemPy* and *BaySeg* to create 3D structural models directly from one-dimensional raw well data and furthermore, to implement that geological model into the segmentation process. This aims for an uncertainty reduction and a faster convergence of the segmentation algorithm. Based on that, the next section describes the data management used to transform the output data of *BaySeg* into a suitable input file for *GemPy*. Moreover, it is outlined how the 3D geological model is used to benefit the segmentation process by expanding the neighborhood system of the FGM model.

3-5-1 Coupling of segmentation, zonation and geological modeling

The crucial task of concatenating the segmentation results of *BaySeg* and the structural modeling of *GemPy* is neither the application of the zonation approaches nor the creation of the *GemPy* input file, but the automatic extraction of information about the input data, which are necessary for the geological modeling. When utilizing *GemPy* only, this essential information is user-defined and inserted to the modeling process manually. Nevertheless, a few parameters need to be set before starting the fully automatic segmentation, zonation and modeling process:

- Number of layers: The number of layers, zones or formations (these words are equivalent) is either determined by the BIC (explained in section 3-1-2) or known and user-defined.
- Considered borehole: The titles of the boreholes, which should be considered in the segmentation, need to be specified beforehand.
- Considered measurement: The measurement types to be included in the segmentation process need to be named in advance.
- Resolution of *GemPy* model: The voxel length in x-, y- and z-direction is set by a single parameter called *Gempy*-resolution.
- Plotting type: The plotting routine allows the visualization of (i) a two-dimensional section, either in x- or in y-direction or (ii) the whole 3D model. In both cases the input data are displayed in the model.

The first step to achieve the automatic input parameter extraction, is by splitting the output of *BaySeg* (table 3-1) either according the labels or probability distributions borehole-wise and applying one of the zonation algorithms, which are explained in section 3-3 to identify boundaries separating layers based on an optimization criterion.
Subsequently, the modeling input file is created as a *pandas*-dataframe by considering the data points right above the determined boundaries as lowest points of the corresponding zone, which contain information about X, Y, Z and *borehole*. The spatial and borehole affiliation information are extracted from the raw data directly and additionally these data points are

Table 3-2: Example of a GemPy input dataframe to reconstruct a horizontal layer model resulting from the synthetic raw data and the applied zonation approach; contains the coordinates of each data point (column 1, 2, 3) as well as its borehole and layer affiliation (column 4 & 5)

X	Y	Z	borehole	formation
100	50	-50	BH1	Layer 1
150	150	-50	BH2	Layer 1
200	100	-50	BH3	Layer 1
100	50	-125	BH1	Layer 2
150	150	-125	BH2	Layer 2
200	100	-125	BH3	Layer 2
100	50	-175	BH1	Layer 3
⋮	⋮	⋮	⋮	⋮

assigned to the layer or zone, which is either most common inside itself or whose probability is maximum. An example of a *GemPy* input dataframe is given in table 3-2.

The dataframe is then stored as a csv-file, which is automatically loaded into *GemPy* via it's input data import routine. This enables the import of interface as well as orientation data, but as long as expensive borehole imaging data are unavailable, one-dimensional well logs do not provide orientation information. Nevertheless, as described in section 3-4-4, one orientation data (gradient of scalar field) per layer is necessary to solve the CoKriging system of equations. Therefore, the dips are calculated from three interface data points of each layer by spanning a plane between them and calculating the centre as well as the normal of the plain. The choice of these data points is essential for the success of the 3D modeling because the resulting layer will be perpendicular to the orientation vector and also cross it. Therefore, not only the dip but also the location of the dipping data needs to be as accurate as possible. Assuming that geological heterogeneity is lower within a smaller area, the orientation data is created from the three interface data points, which are closest to each other. An alternative approach that puts the orientation data far away from the area of interest, is outlined in the discussion of this work.

As a next step, the sequential order of the geological formations or in other words, the age-related layer order is set automatically, wherefore the layer affiliation data is sorted by depth and duplications are removed. Once the input data is prepared and the sequential pile is defined, the input data for the interpolation is generated. That includes (i) setting the interpolation parameters for the CoKriging, (ii) rescaling the input data extent according to the minimal and maximal extend of the data and the pre-defined resolution, and (iii) numbering the layer from young to old along with creating a default basement formation.

These steps also compile the *Theano* graph function. This is computationally expensive, but needs to be done only once because the interpolation data can be updated only in the following iterations, instead of recreating it (De la Varga, 2018). After the aforementioned preparation tasks, the geological 3D model is computed, which results in a lithological block model containing information about the layer or formation at each voxel and its gradient. Eventually, this is displayed as specified previously in 2D or 3D. The results of the automatic coupling of segmentation, zonation and geological modeling are displayed and analysed in

section 4-2, while the next section outlines an implementation approach of *GemPy* into the segmentation.

3-5-2 Implementation of 3D geological model into the segmentation

The main innovation of Wang et al.'s geological segmentation approach is its consideration of the neighborhood system via the calculation of the MRF energy. But the neighborhood system of one-dimensional well data is limited to a voxel above and below the considered one. With the aim to extend this system, the three-dimensional structural model created by *GemPy* is utilized to fill the empty space between the boreholes. This principle in two dimensions is visualized in figure 3-5. The borehole is located at $X = 90m$ and its data points are represented by stars. Around the borehole a XZ-section of the 3D geological model is displayed, whose data points are symbolized by black dots.

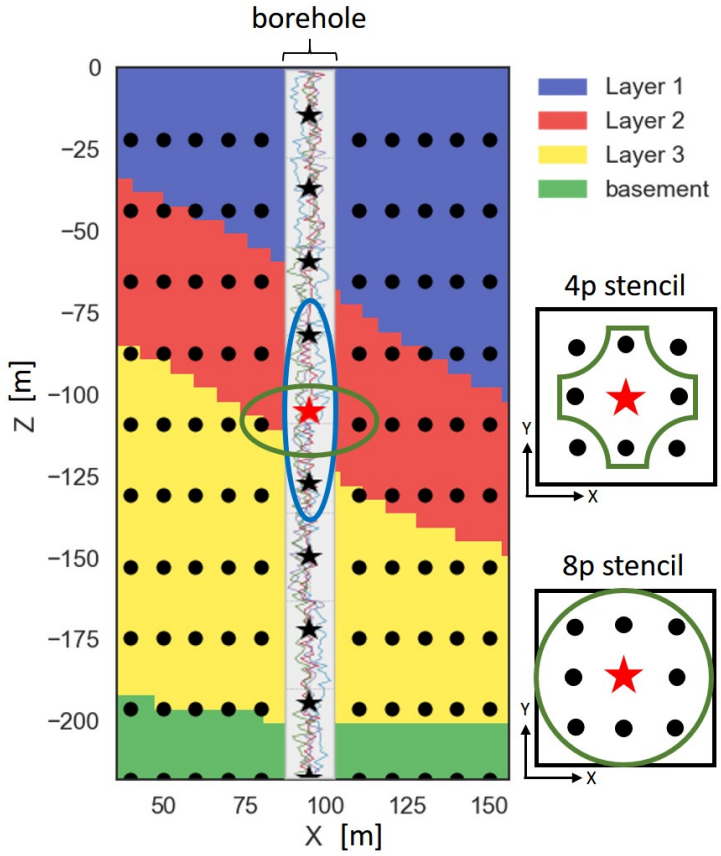

Figure 3-5: Visualization of the neighborhood system extension; where the stars represent data points in the borehole and the black dots data points of the 3D geological model. The red star denotes the considered point, whose neighborhood system is regarded. The blue circle marks the neighborhood system considered by $BaySeg$, while the green circle represents its extension. The X-Y-sections at the bottom right visualize the 4 and 8 point stencil neighborhood systems.

Considering the neighborhood system of the red star, *BaySeg* includes the data points right above and below in the segmentation process (marked by blue circle). The approach developed in this work additionally considers four or eight neighbored points resulting from the *GemPy* model. In 2D these are only two points, which are tagged by the green circle in figure 3-5. The XY-section below the legend in figure 3-5 visualize the four and eight point stencil neighborhood system, respectively, where the red star represents again the considered borehole data point. It is worth to mention, that the resolution of the well logs (measurement points per distance) does not automatically equal the resolution of the geological 3D model and, thus, the well data voxels do not overlap with the points of the *GemPy* grid. Therefore, the closest point $\vec{x}_c$ of the *GemPy* model to each borehole data $\vec{x}_{BH_i}$ is considered, which is determined by calculating their minimum distance:

$$\vec{x}_{c_i} = min[\vec{x}_{GemPy} - \vec{x}_{BH_i}] \quad \text{with } i \in [1, 2, 3, ..., n], \tag{3-36}$$

where $\vec{x}_{GemPy}$ are all coordinates in the *GemPy* grid and n is the number of borehole data points. Numerically, the expansion of the neighborhood system is implemented by extracting XY-plains of the lithological block model for all depth levels and the MRF energy at each point depending on their direct neighbors (4 or 8 stencil) is calculated (see equation 3-12). This energy is referred to as *GemPy* energy, which is finally added to the total energy in the segmentation process (after equation 3-20):

$$U'_j(x_j, \vec{x}_{\partial_j}) = \underbrace{U(x_j, \vec{x}_{\partial_j})}_{MRF\ energy} + \underbrace{U(y_j|x_j, \theta_{x_j})}_{likelihood\ energy} + \underbrace{U(x_j, \vec{x}_{\partial_j}^{GemPy})}_{GemPy\ energy} \tag{3-37}$$

This principle might seems to be simple, but by implementing a completely new two-dimensional neighborhood system, the granularity coefficient becomes two-dimensional as well and needs to be sampled and updated in each iteration. This is achieved by introducing an additional β_{gp} and expending the parameter estimation by Bayes' Theorem for β_{gp} (see equations 3-22, 3-23 and 3-24):

$$p(\beta_{gp}|\vec{y}, \vec{x}, \vec{\mu}, \vec{\Sigma}) \propto p(\beta_{gp})\ L(\vec{y}|\vec{x}, \vec{\mu}, \vec{\Sigma}, \beta_{gp}). \tag{3-38}$$

Numerically, this implies the creation of an additionally prior distribution for β_{gp} in the initial configuration of *BaySeg*. Furthermore, the update of the segmentation itself (step 1 in section 3-2-5) remains independent of the *GemPy* neighborhood system, due to the computational costs. Recalculating the 3D model during the sampling in case of ten boreholes with 190 data points each would require the creation of 950 *GemPy* models in each iteration. The results of this method and an analysis if it reduces uncertainties or causes a steeper convergence in the segmentation process is detailed in section 4-3.

3-6 Creation of synthetic well log data

An important factor, when testing the performance of segmentation algorithms, is the creation of synthetic data, where the model to recover is known as a reference and the data deviation can be controlled. There exist several distributions with different advantages and disadvantages, but in this work the synthetic data are created using normally distributed data. This enables the control of the standard deviation σ or variance σ^2 defining the scattering of the data points from the mean. The normal or Gaussian distribution is defined by (Bulmer, 1979):

$$f(x) = \frac{1}{\sqrt{2\pi\sigma}}\, exp\left(-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}\right), \tag{3-39}$$

where x is the data and μ its mean. In case, one also wants to control the covariance matrix, a multivariate normal distribution can be used to create the synthetic data.
To ensure a simple comparison of the segmentation result and the desired model, the synthetic data are created from *GemPy*-models directly. Therefore, the borehole length, the layer notation, and the number of layers are extracted automatically from the input data (lithological model). Afterwards, the basement values are removed and randomly located boreholes are extracted from the model, assigned with their coordinates as well as the corresponding layers at each depth point. Finally, Gaussian distributed random variables are created with a user-defined standard deviation σ as well as number of features and boreholes. Thereby, the number of data centres equals the number of different layers and the data is then assigned to each data point corresponding to its labeling. Several examples are displayed in figure 3-6, where synthetic well logs are created directly from a *GemPy* model with 4 horizontal layers. All of them vary in standard deviation increasing from $\sigma = 3$ in a) to $\sigma = 7.5$ in e). A detailed inspection and the performance of *BaySeg*'s segmentation of the data displayed in figure 3-6 is presented in section 4-1. In the next chapter, the results of all the methods outlined in this chapter are described and discussed. Moreover, the optimal parameter setting for the modeling approaches is investigated.

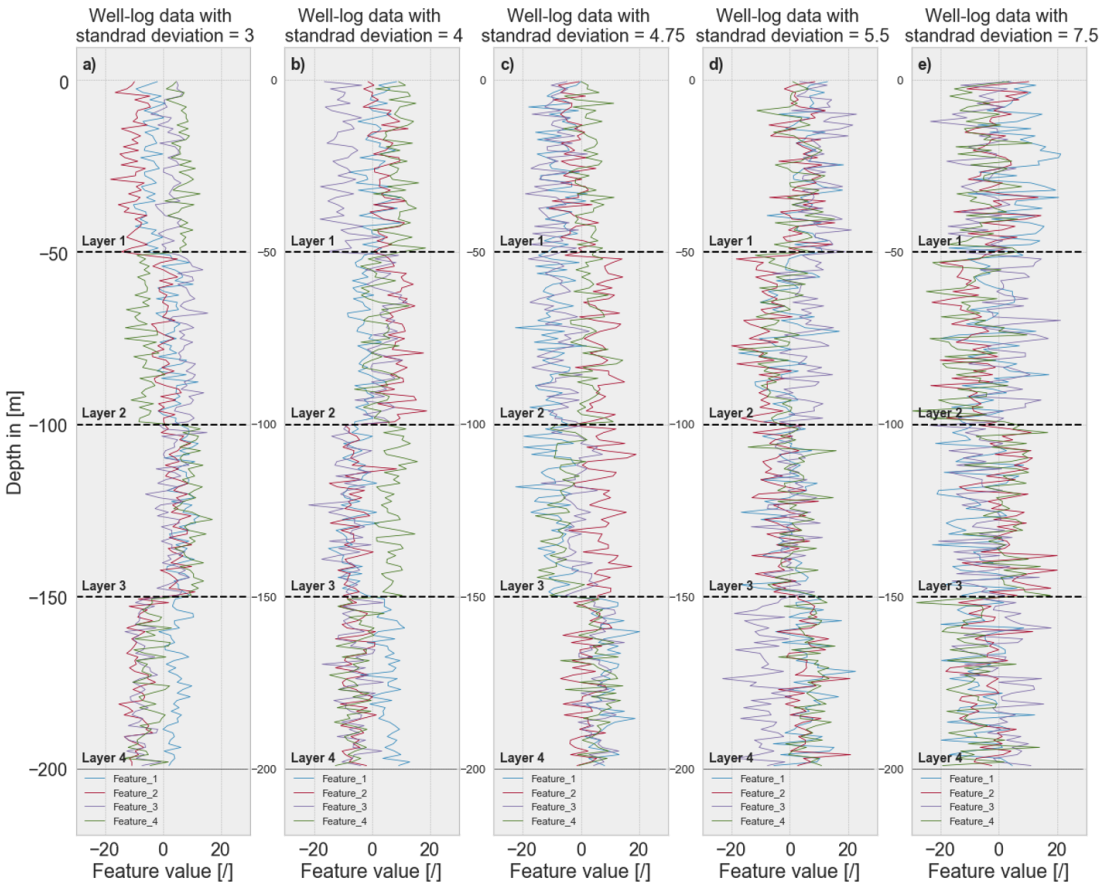

Figure 3-6: Synthetic well log data for increasing standard deviations from left ($a : \sigma = 3$) to right ($e : \sigma = 7.5$), created directly from a four-layer horizontal $GemPy$ model and including four features (measurements) at each data variation level; layer boundaries are represented by dashed black lines.

Chapter 4

Results

This chapter visualizes and analyses the results of the methods utilized in this work. This includes statistical performance investigations of different parts of the automatic geological 3D modeling algorithm to determine appropriate parameter settings; and furthermore, the comparison of outcome models with their origin to expose limitations of this approach. In section 4-1 the segmentation is applied to different distributed single wells to check its performance and optimize the segmentation parameters. During the next section 4-2, the zonation approaches are tested on several segmented wells. The outcome is used to create 3D geological models automatically. The effect on the uncertainties within the process after implementing the structural modeling into Wang et al.'s method is investigated in section 4-3.

4-1 Segmentation of single well logs - Parameter testing

Originally, Wang et al. (2017) developed their work to model geophysical data in n-dimensions. In this work their method is applied to one-dimensional well data, which is tested in this section. Therefore, the performance of the BIC on synthetic well data with varying standard deviation is investigated first. Then, the segmentation process is applied to the same data with different granularity coefficients. Moreover, the influence of the $jump - parameters$ is examined, which control the proposal of new candidates in the MCMC sampling.

4-1-1 Bayesian information criterion performance

The Bayesian information criterion (BIC) can be used to investigate the statistical nature of the data to find the optimal number of clusters or labels before the actual clustering process. This number is a necessary input parameter and influences the results of the segmentation process significantly. Table 4-1 displays the estimated label number for synthetic well logs with different standard deviations, using the BIC as described in section 3-1-2. The data are created as outlined in section 3-6 and plotted in figure 3-6. The results demonstrate that up

to a standard deviation $\sigma = 3$ the BIC is able to reconstruct the original number of layers (4 layers in this case).

Table 4-1: BIC estimation of the number of layers on synthetic well data with 4 different layers and varying standard deviation σ, which is given in the first line, while the second line displays the estimated number of layers.

σ	2	3	4	4.75	5.5	7.5
number of layers	4	4	3	3	1	1

However, for standard deviations $\sigma \geq 4$ the BIC's finding is incorrect. This result is of significance for the following analysis in this work because well logs can vary strongly in their statistical nature due to the huge amouns of influencing factors. Therefore, when working with real data sets, where the number of labels is unknown beforehand, a manual double checking is advised to ensure data correctness. This information can be extracted by either an investigation by experts of a tiny part of the data itself or from other measurements, e.g. seismic or geo-electric.

4-1-2 Segmentation performance on a single well

In this section, the performance of Wang et al.'s segmentation approach is tested on synthetic well data with different standard deviations σ. This aims to investigate the effect of the granularity coefficient β on the segmentation result and to find appropriate parameter settings. Moreover, the $jump - parameters$ are tested, which control the drawing of new samples in the MCMC algorithm for μ, Σ and β (see section 3-2-5). As a reminder, each data set has its own distribution parameter setting ($\Phi = \beta, \mu, \Sigma$) that needs to be estimated. In case the "true" model is unknown, the optimal parameter setting is impossible to determine, but the results of this section serve as a clue for the segmentation of one-dimensional well data. The performance of the segmentation process is quantified by the normalized number of misclassified voxels δ_{last} in the last iteration and the mean of all δ after a user-defined number of iterations $\overline{\delta}$ (equals the percentage of incorrect segmented voxels):

$$\delta = \frac{number\ of\ missed\ voxels}{total\ number\ of\ voxels}. \tag{4-1}$$

As visible in figure 3-6, the separation between the 4 layers is straight forward while looking at it just until the standard deviation of $\sigma = 3$. As for $\sigma =\geq 4.75$, the data is widely distributed and a visual separation becomes much more difficult. This separation is nearly impossible for $\sigma = 7.5$. It is also visible that the different features add different amounts of information to each label boundary. For example, in figure 3-6 b), with a standard deviation $\sigma = 4$, the separation of layer 3 and 4 is impossible by just considering feature 2 (red) and 3 (purple). Nevertheless, taking all four features into consideration ensures a visual and statistical difference at the layer boundary. This also holds true for real well data because the measurement techniques (e.g. resistivity, gamma-ray etc.) are sensitive to different physical properties in the subsurface (e.g. water content, density, mineral composition etc.).
Table 4-2 lists the determined δ_{last} and $\overline{\delta}$ for five different synthetic well data with a constant

granularity coefficient β after 1000 iterations, while table 4-3 recites these parameters with varying β (mostly increasing). This means that for the latter case the β_{init} is user-defined and β is updated in each iteration or, in other words, that the neighborhood system is taken into consideration to a greater extend. The data proves that in general the segmentation process labels more voxels correctly with varying β, which is obvious for one-dimensional well data from a 4-layer horizontal model. Furthermore, it indicates that δ_{last} as well as $\overline{\delta}$ are smaller for high granularity coefficients and for a model where only 3% of the voxels differ from at least one of their neighbors.

Table 4-2: Segmentation error in the last iteration δ_{last} and its mean $\overline{\delta}$ over all iterations for synthetic well logs with different standard deviations σ and different β values over 1000 iterations (other parameters: $beta\_jump\_length = 0.02$, $mu\_jump\_length = 0.0005$, $cov\_volume\_jump\_length = 0.00005$, $theta\_jump\_length = 0.0000005$).

β \ σ		**3**	**4**	**4.75**	**5.5**	**7.5**
0.2	$\overline{\delta}$	0.070	0.063	0.153	0.318	0.484
	δ_{last}	0	0.068	0.179	0.305	0.468
0.4	$\overline{\delta}$	0.001	0.055	0.133	0.215	0.486
	δ_{last}	0	0.074	0.142	0.205	0.468
0.6	$\overline{\delta}$	0	0.050	0.131	0.310	0.373
	δ_{last}	0	0.047	0.126	0.314	0.353
0.8	$\overline{\delta}$	0	0.044	0.112	0.314	0.391
	δ_{last}	0	0.047	0.111	0.295	0.411
1	$\overline{\delta}$	0	0.037	0.103	0.184	0.456
	δ_{last}	0	0.047	0.095	0.211	0.437

Furthermore, the difference between $\overline{\delta}$ and δ is an indicator of the segmentation algorithm's convergence towards the desired model during all iterations. For fixed β the differentiation of data points depends mostly on their statistical nature and, thus, the error σ does not converge. Alternatively, when the granularity coefficient is updated in each iteration the spatial correlation of the data is more strongly taken into consideration and the segmentation results converge. This behaviour is visualized exemplary for $\sigma = 4.75$ and $\beta_{init} = 0.6$ in figure B-1. For constant granularity coefficients the difference is very small (e.g. $\beta = 0.4$; $\sigma = 5.5$) and for some scattered cases the mean is even lower than the δ of the last iteration (e.g. $\beta = 0.4$; $\sigma = 4.75$). This suggests a small or even no improvement of the segmentation results. When updating β in each iteration, this behaviour is also obtained occasionally, but for other parameter settings the difference and, thus, the improvement is large (e.g. $\beta_{init} = 0.4$; $\sigma = 4.75$). This indicator is found useless for data with small standard deviations ($\sigma = 3, 4$) because the correct labelling is just found in the first few iterations. The data also clarifies that there is no "best" parameter setting for the data. Due to the above findings, an updating granularity coefficient β is applied for all segmentations in this work with different initial granularity coefficients $\beta_{init} \in [0.6, 1]$. An investigation of the granularity coefficient's convergence for this data is unusable as the original model of the synthetic well logs consists

of large blocks and, thus, several β will result in the same model. This is different for data with many small intervals, which are to be distinguished.

Table 4-3: Segmentation error in the last iteration δ_{last} and its mean $\overline{\delta}$ over all iterations for synthetic well logs with different standard deviations and updated β over 1000 iterations with different initial granularity coefficients β_{init} (other parameters: $beta\_jump\_length = 0.02$, $mu\_jump\_length = 0.0005$, $cov\_volume\_jump\_length = 0.00005$, $theta\_jump\_length = 0.0000005$).

β_{init} \ σ		**3**	**4**	**4.75**	**5.5**	**7.5**
0.2	$\overline{\delta}$	0	0.010	0.047	0.216	0.335
	δ_{last}	0	0.005	0.053	0.211	0.211
0.4	$\overline{\delta}$	0	0.011	0.044	0.231	0.478
	δ_{last}	0	0.016	0.005	0.232	0.468
0.6	$\overline{\delta}$	0	0.009	0.015	0.231	0.297
	δ_{last}	0	0.005	0.005	0.221	0.216
0.8	$\overline{\delta}$	0	0.010	0.043	0.073	0.260
	δ_{last}	0	0.016	0.021	0.074	0.247
1	$\overline{\delta}$	0	0.008	0.037	0.310	0.262
	δ_{last}	0	0.005	0.021	0.221	0.205

To finetune the parameter setting different $\beta\ jump\ length$ values with one of the afore estimated β-settings are tested (updating β with $\beta_{init} = 0.6$). Figure 4-1 displays the normalized number of missed voxels δ_{last} in the last iteration and the mean of all δ after 100 iterations $\overline{\delta}$. The results of the first run (blue & red graphs) suggest that the $\beta\ jump\ length$ does not influence the segmentation of one-dimensional well data significantly and, thus, δ_{last} as well as $\overline{\delta}$ are varying only slightly. Therefore, a second and third test is run, which substantiate the hypothesis partly.

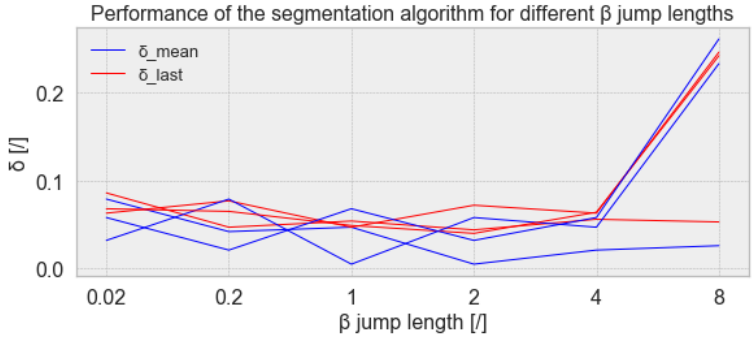

Figure 4-1: Segmentation error in the last iteration δ_{last} (red) and its mean $\overline{\delta}$ (blue) for synthetic well logs with different $\beta\ jump\ lengths$ and updated β over 100 iterations (other parameters: $\beta_{init} = 0.6$, $mu\_jump\_length = 0.0005$, $cov\_volume\_jump\_length = 0.00005$, $theta\_jump\_length = 0.0000005$).

It can be concluded that the $\beta\, jumplength$ negatively affects the results only if β is excessively high, while it is has minimal influence for $\beta\, jump\, lenght \in [0.02, 4]$. Similar observations are made for the other $jump\, parameters$, which control the drawing of new μ, β and Σ samples in the MCMC algorithm.
Therefore, the default $jump parameter$ setting is kept, which yields a final parameter setting for the $BaySeg$ segmentation as listed in table 4-4. An important point to consider is that the estimated parameter setting is optimal only for the utilized synthetic data and the "hidden" pattern behind it or, in other words, the shape of the posterior distribution. It might completely or partly different for real data, where the "reality" is unknown and, thus, the optimal parameter setting is hard to determine. Nevertheless, this setting is utilized for the zonation approach performance test because it is applied on the same synthetic data. This is explained in the next section. A comparison to the support-vector network (SVN) approach by Hall (2016), as described in appendix A-1 for a single well is impossible because it requires one interpreted well to train the algorithm. Thus this comparison is also drawn in the next section.

Table 4-4: Final parameter setting determined by several parameters tests.

Final parameter setting	
update β	True
β_{init}	$\in$ [0.6, 1]
$\beta\, jump\, length$	0.02
$\mu\, jump\, length$	0.0005
$\Sigma\, volume\, jump\, length$	0.00005
$\theta\, jump\, length$	0.0000005

4-2 Segmentation of several wells and geological model creation

In this section the segmentation algorithm $BaySeg$ is first applied to several well logs from different boreholes to analyse its accuracy depending on the data volume and the number of boreholes, respectively. Moreover, the zonation approaches are tested for synthetic data with differing standard deviations. Eventually, the ability of model reconstruction of the whole process of automatic structural geological modeling from raw data is evaluated trying to reproduce 3D models with different features (e.g. tilted layers, faults, folds).

4-2-1 Segmentation of well logs from different boreholes

In general, the segmentation of a single well or data from several wells utilizing $BaySeg$ is identically because the data is not split borehole-wise, but rather segmented globally. Nevertheless, the amount of data points influences the unsupervised segmentation outcome. To investigate this effect, the method is tested on synthetic data with standard deviation $\sigma \in \{4, 5.5\}$, while the number of boreholes # differs. A first observation is made while applying the BIC on the data. In fact, the estimation of the number of labels or segments does not improve (remains constant) with increasing number of data points (190 data points

per borehole, thus 4750 data points maximum).
Figure 4-2 exhibits δ_{last} and $\overline{\delta}$; thus, the last and the mean segmentation error rate for different numbers of boreholes $\# \in \{1, 4, 10, 25\}$. The graphs indicate that an increasing number of boreholes/data points yields to a more accurate segmentation process. Nevertheless, the enhancement between ten and 25 boreholes is minimal. This finding becomes important when applying the whole process of automatic 3D geological modeling from raw well logs to a data set with a huge amount of boreholes, while the geology is expected to differ only slightly between two boreholes.

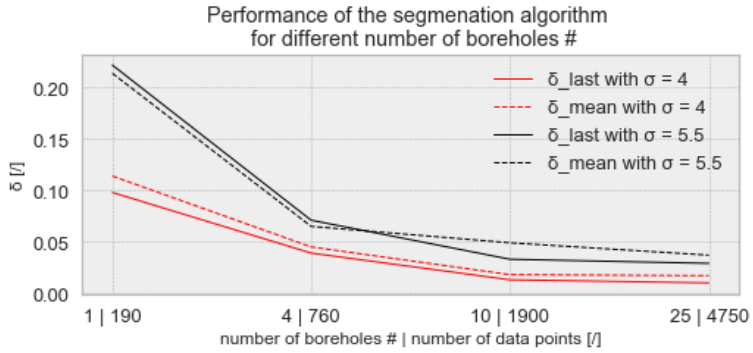

Figure 4-2: Segmentation error in the last iteration δ_{last} (continuous lines) and its mean $\overline{\delta}$ (dashed lines) for synthetic well logs with standard deviation $\sigma = 4$ (red), 5.5 (black) and different number of boreholes $\#$ and updated β over 500 iterations (other parameters: $\beta_{init} = 0.6$, $beta\_jump\_length = 0.02$, $mu\_jump\_length = 0.0005$, $cov\_volume\_jump\_length = 0.00005$, $theta\_jump\_length = 0.0000005$).

Eventually, the segmentation of several boreholes enables a splitting in training data and data to predict and, thus, a comparison to the support-vector network (SVN) approach, described in appendix A-1. Therefore, each synthetic dataset of the 4-layer horizontal model with different standard deviations is segmented with varying C and $gamma$ values to estimate the optimal parameter setting. The results are listed in appendix A-2.
Table A-1 displays the segmentation error δ_{SVN} for C (smoothness of decision surface) varying from 0.1 to 1000000. The results demonstrate that the accuracy of the supervised segmentation increases with increasing C and that data with high standard deviations require higher C values to obtain the best results. To keep the computational costs under control, C is set to 10000 for the following comparison. In table A-2, the accuracy depending on $gamma$ (inverse radius of the influence of a single training data point) is investigated. It is observable that $gamma = 10$ yields the most precise labelling of all voxels for data with $\sigma = 4,\ 4.75,\ 5.5,\ 7.5$. Comparing these results with the unsupervised segmentation leads to the information given in table 4-3, revealing that the accuracy is similar for slightly varying data ($sigma = 3,\ 4$), while the SVN based method never performs a perfect segmentation. However, it is more precise for data with $\sigma \geq 4.75$. Considering that the method suggested by Hall (2016) is supervised and requires a training from interpreted well logs, this approach does not fit in the fully automated process developed in this work.

4-2-2 Zonation of the segmented data

This section investigates the performance of the zonation methods on synthetic data with varying standard deviation σ. The principles of these zonation approaches are outlined in section 3-3. As a reminder, the first method is based on the minimization of the variance within zones, referred to as zonation 1, which is applied on the labelling. Zonation 2 processes the label probabilities directly by maximizing the ones according to layer l, while minimizing the probability of all other labels $k \neq l$ within a zone.
The reconstructed boundaries of both zonations are visualized in figure 4-3, the true boundaries being represented by black lines, the ones resulting from zonation 1 by red lines and the green lines representing the outcome of zonation 2.

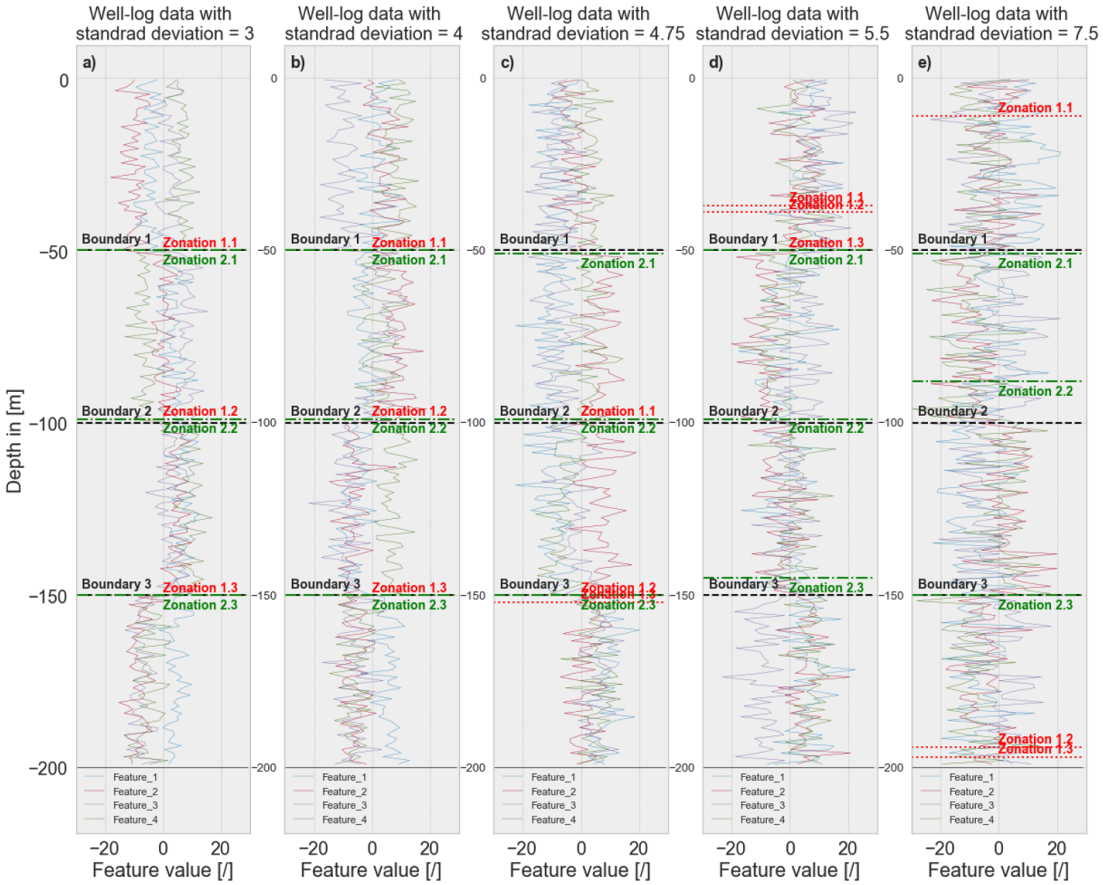

Figure 4-3: Reconstructed boundaries of zonation approaches 1 (minimization of variation within zones) represented by red lines and 2 (maximization of probabilities within zones) represented by green lines; the true boundaries, which are to be reconstructed, are represented by black lines. The standard deviation of the synthetic well data increases from left (a: $sigma = 3$) to right (e: $sigma = 7.5$).

It is observable that zonation 1 perfectly re-establishes the zone boundaries for low standard deviation ($\sigma = 3, 4$). But with decreasing data separability the approach fails. This is caused by the transformation of the segmentation results from four large intervals with homogeneous labelling to a mixed up segmentation with plenty small intervals. This can

either lead to very small zones (figure 4-3d) or to very large zones, in which the deviations cancel out (figure 4-3e: boundaries 1.2 & 1.3).
Examining the boundaries determined by zonation 2 reveals a more accurate boundary estimation. In detail, it is discernible that the method misses some boundaries by one data point (e.g. boundary 2.2 with $\sigma = 3, 4, 4.75, 5.5$), which is caused by the uncertainties introduced at transitions from one zone to another. But, apart from boundary 2.2 in figure 4-3e, it reconstructs all boundaries at an acceptable position for the ensuing 3D modeling.
Besides their zonation performance, zonation 2 is computationally much faster ($1.39\ ms \pm 17.4\ \mu s$ per loop), compared to the minimum-variance approach ($3\ min \pm 4.43\ s$ per loop). This is due to the fact that it uses the probabilities directly as an optimization criterion instead of calculating an extra quantity (variance) to determine boundaries. It has been shown that zonation 2 is more capable of reconstructing the true boundaries. Thus, it is utilized for the automatic structural model creation investigated in the next section.

4-2-3 Automatic 3D geological modeling from raw well logs

In the following the algorithm's capability to reproduce initial models m_{rc} with different features is examined. Instead of investigating the results of the stepwise applied segmentation, zonation and modelling first, the results of the fully automated process are considered directly because their outcomes are equal. Therefore, several structural 3D models are utilized as an origin to create synthetic well logs as described in section 3-6. Moreover, this data is inserted into the algorithm and the reconstructed model, resulting from the automatic segmentation ($BaySeg$), zonation and geological modeling ($GemPy$) is compared to its origin. It is important to keep in mind that the segmentation as well as the structural modeling are global processes applied to all data simultaneously, while the zonation is applied to each borehole individually.
The first column in table 4-5 displays a Y-Z-section of the original models, while the other columns contain the same section of the reconstructed models for increasing standard deviation from left to right. Within each line the dimensions in Y- and Z-direction are constant and, thus, the axis labels are neglected. Additionally, the error δ of each reconstructed model compared to its origin is given, which results from the model differences normalized by the number of data points. One may observe that several data points lie outside their corresponding layer, which is a visualization issue, since each figure shows a two-dimensional section only, while the interface points are projected from 3D onto the Z-Y-sphere. This issue is illustrated in figure B-3, where the left-hand side shows a X-Z section. The blue interface data at Z = -1100 and X = 2800 seems to lie outside the corresponding layer. However, the 3D model on the right-hand side of the figure demonstrate that these points are indeed part of the blue layer. Moreover, the bottom layer (basement) in each model can be ignored because $GemPy$ models the bottom of each interface and tacks the basement automatically below the oldest one.
In table 4-5 line a) the results are displayed for a horizontal 4-layer model. They demonstrate that for standard deviations up to $\sigma = 4.75$ the simple subsurface model can be reconstructed almost perfectly. This finding is underlined by small errors ($< 3\%$), which are due to minimal dipping of layers. This is visible in the Z-Y-section for $\sigma = 3$, where the layers dip slightly to the right. Furthermore, it is observable that scattered boundary points are mis-

placed throughout the zonation, which coincides with the finding in section 4-2-2, where the zonation approach is examined in detail. A massive misplacement through zonation can be observed in the last model ($\sigma = 7.5$) of line a), where the locations of the red and yellow interface points are distinctly wrong, which yields an overestimated extension in Z-direction of the red layer and a compression of the layers below.

Table 4-5: Reconstructed models m_{rc} utilizing the full automatic coupling of segmentation ($BaySeg$), zonation and modeling ($GemPy$) for synthetic data after 500 iterations. Each row displays the model, which is to be reconstructed, at the left-hand side and the reconstructed models for increasing standard deviation σ from the left to the right. The segmentation error δ above each reconstructed section lists the percentage of misclassified voxels.

m_{rc} \ σ		**3**	**4**	**4.75**	**5.5**	**7.5**
a)	δ	0.022	0.010	0.003	0.349	0.507
b)	δ	0.065	0.289	0.041	0.043	0.105
c)	δ	0.076	0.052	0.048	0.219	0.583
d)	δ	0.188	0.129	0.591	0.321	0.407
e)	δ	0.105	0.164	0.094	0.089	0.821

An interesting outcome is discovered for the modeling with a standard deviation of 5.5. The layers form an anticline, although the single data points of each layer are located at around the right depth level. This example clarifies the importance of the point selection for the orientation vector creation. As is visible here, even slight mislocation of the yellow points leads to a dipping into the image plane and ruins the entire model. A closer look at the orientation data creation and the corresponding points selection is given in the discussion section.
A model with three layers dipping from left to right and the corresponding reconstructions is displayed in line b). The sections indicate that the algorithm recovers the origin properly for all standard deviations, even for $\sigma = 7.5$. But, when considering the error $\delta = 0.289$ of the second model ($\sigma = 4$) and its strongly tilted orientation data, it becomes clear that only a part of the model, including the displayed section, matches the original model, while the other parts mismatch. Table 4-5 line c) demonstrates the results for reconstructing a 4-layer fold model, where the interfaces form an anticline. Once again, the models for low standard deviations depict the initial one. The model for $\sigma = 5.5$ is another proof for the importance of a suitable point selection for the orientation data preparation. The last model in line c) exemplifies a total failure of the reconstruction in the right part of the section, while the rest differs only slightly from the origin.
One-dimensional well logs do not provide direct information on faults (perhaps indicate existence of faults) and, thus, cannot be insert to the modeling algorithm, although *GemPy* is capable of processing this information. Therefore, subsurface patterns including fault networks or even a single fault is expected not to be reconstructible through well data only. This is underlined by the results exhibited in line d) of table 4-5. None of the models reconstruct the fault and, thus, the differences δ are large for slightly varying data. As expected, the fault is modeled as a continuous interface for all standard deviations. A special case that needs to be discussed is the reconstruction for $\sigma = 4.75$. While the algorithm recovers all other original sections with this standard deviation, this model is totally misconstrued and indicates that the algorithm got either trapped in a local minimum or did not reach the global minimum within the 500 iterations. This results from the "randomness" of the segmentation approach (*BaySeg*), which is introduced through the initial model on one hand and the MCMC algorithm with random sampling on the other hand. A discussion on that and the associated reproducibility of the models is provided in the discussion part of this work.
Assuming a correct zonation, one could claim that an infinite number of boreholes could reconstruct the fault because each data point would be known. Although this is not economical in practise, figure B-2 confirms this hypothesis. It shows the reconstructed model for 50 randomly placed boreholes and indeed one could interpret a fault, especially considering the red layer's jump at X = 650.
A common situation in the subsurface is erosion of a layer, which is then overlaid by a younger sediment. Such an unconformity model is reproduced in line e) of table 4-5, where the red layer is eroded and overlaid by the blue one. Considering the models for $\sigma \leq 5.5$, the original model is reconstructed with a maximum error of $\delta = 16.4\,\%$ and the unconformity is recognizable in all sections. As seen before, small deviations due to minimal zonation errors and the resulting dip calculation can be observed in the lower right of the second reconstructed model. The model estimation from the most varying data failed completely. At the first glance, one could conclude that the incorrect zonation of the red layer causes the surprising modeling result. But, on closer inspection, the yellow orientation data pointing horizontally in the image plain, destroys the model.

It has been demonstrated that except for fault structures all common features in the subsurface can be reconstructed for standard deviations up to 4.75. Considering only borehole data, more complex models with several features like a Graben or dome structure are not recoverable. Possible combination with other geophysical measurements is discussed in the last chapter of this work.

4-3 Uncertainty reduction in segmentation process

As mentioned in the introduction, the main hypothesis of this work is that considering the geological model created with *GemPy* in the segmentation of one-dimensional well logs will reduce uncertainties in the segmentation itself. This approach is implemented by introducing a neighborhood system based on the geological 3D model. As described in section 3-5-2, this additional neighborhood system is not considered in the model $\vec{x}$ sampling and, thus, a significant improvement of the segmentation itself is not expected. Figure 4-4 displays the segmentation error δ at each iteration resulting from the 4-layer horizontal model in four different runs. The red plots represent the error considering the *GemPy* model, while the blue plots display the segmentation course without including the geological 3D model. It is observable that the graphs' behaviour is similar for all runs and the segmentation results remain unimproved due to the consideration of the additional neighborhood system. This is underlined by the number of misclassified voxels, displayed in the table on the right hand side of figure 4-4, where the first and second run are taking *GemPy* into consideration and the latter runs do not.

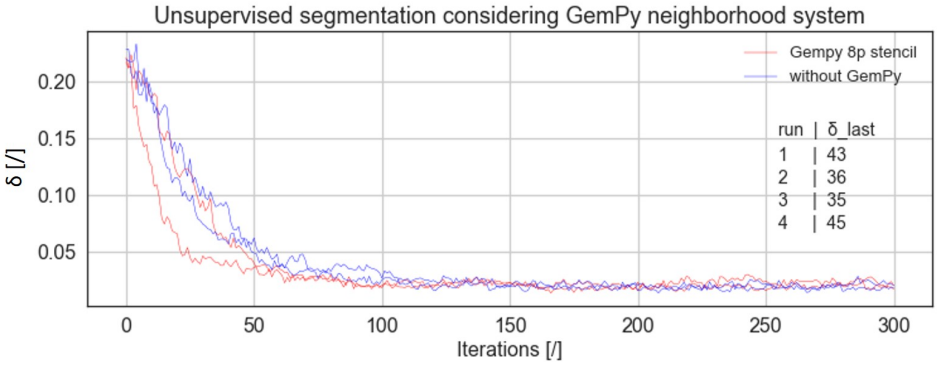

Figure 4-4: Segmentation error δ over all iterations in the unsupervised segmentation with *BaySeg* applied on synthetic well data from a 4-layer horizontal model with a standard deviation of 5.5; modeled twice including the *GemPy* neighborhood system (red) and twice excluding it (blue); the table lists the total number of misclassified voxels in the last iteration.

It has to be kept in mind that the results presented above are based on the most likely result only, without taking a closer look at the uncertainties. To further investigate the voxels' probability of being assigned to a specific label or layer, an uncertainty quantification based on the concept of information entropy is performed (see section 2-3). As a reminder, low information entropy values indicate low uncertainty areas, while values around one represent high uncertainties in layer assigning. Figure 4-5 visualizes the development of the information

entropy during the segmentation of the 4-layer horizontal model (line a in table 4-5).
Row A displays the information entropy taking the 3D model into consideration, while row B displays the same excluding the additional neighborhood system. The information entropy is shown at the start of the segmentation as well as at iteration 50, 100, 150, and 200. Considering the image after 200 iterations (right-hand side) in row B, large information entropies can be observed at layer boundaries because the one-dimensional neighborhood system considers the data points above and below each voxel, which are different at the boundaries. Taking the geological model from *GemPy* into account (right-hand side of row A), leads not only to reduced uncertainties at layer transitions, but in the entire model. Especially the transition at Z = -150 is to be assigned more clearly. This is caused by the additional consideration of the horizontal neighborhood system of the *GemPy* model.

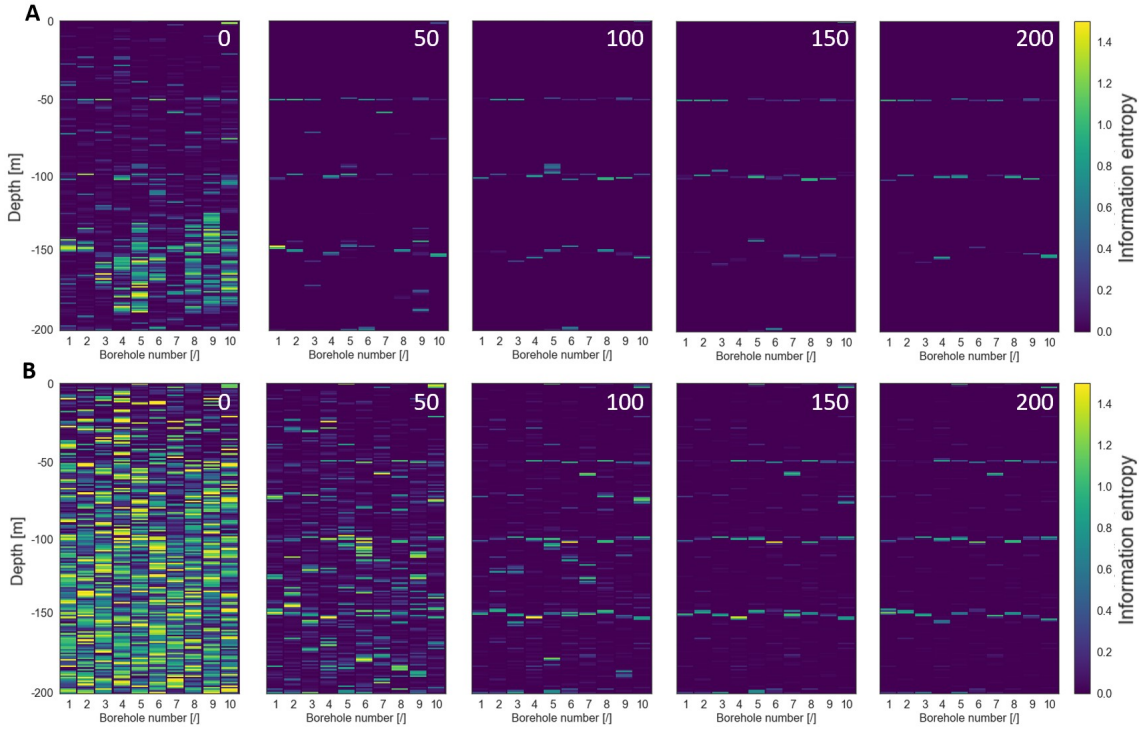

Figure 4-5: Development of information entropy in the segmentation of synthetic well data with standard deviation $\sigma = 5.5$ from a 4-layer horizontal model displayed for ten boreholes; A: including the *GemPy* neighborhood system and B: excluding it for iterations 0, 50, 100, 150 & 200 from left to right.

As a reminder the zonation approach utilized in the automated process is based on the uncertainties resulting from the segmentation. That means reducing the uncertainties will also enhance the zonation and, thus, the whole 3D modeling in each iteration. To prove this hypothesis, the development of the information entropy in figure 4-5 is investigated. While the entropies differ only slightly for the last 100 iterations, the uncertainty reduction becomes evident at the starting point. The usual segmentation (part B) reveals high uncertainties over the entire model at iteration 0; however, the first picture in part A shows strongly reduced uncertainties and gives already an idea of the upper two boundaries. The above findings have shown that introducing an additional neighborhood system, which considers the horizontally

neighbored labels at each well data location, reduces uncertainties. This in turn enhances the whole process of automatic geological 3D modeling from raw data provided by borehole measurements.

Chapter 5

Discussion

The purpose of this work was to apply recent developments by Wang et al. (2017) in unsupervised segmentation of n-dimensional geophysical measurements, taking the spatial correlation of the data into consideration, on one-dimensional well logs and, furthermore, to couple this clustering with structural geologic 3D modeling (de la Varga et al., 2018) in one Bayesian framework. Moreover, the main hypothesis was that taking information from both modeling approaches into consideration, reduces uncertainties in the segmentation. The automatic interpretation of raw well data has been a subject of research for more than 50 years and is of great importance not only in hydrocarbon exploration, but also in other fields, like reservoir engineering and geothermics.

The results of this work revealed that Wang et al.'s approach is capable of segmenting raw well logs in an unsupervised manner, although algorithms trained on the data itself are slightly more accurate. Based on the clustering results, it was shown that the zonation, which maximizes probabilities within zones, determines layer boundaries in an appropriate way, but is also limited in terms of borehole correlation. The constraint that comes along with the zonation to force the stratigraphic pile to be constant over all boreholes might seem strong, but is a requirement of the geological modeling with *GemPy*. Eventually, after coupling segmentation, zonation and 3D modeling in a fully automated process, the modeling reconstructions demonstrated that the algorithm is capable of recovering models including tilted layers, folds and nonconformities even if the data are comparatively noisy. Nevertheless, the approach has limited capability in recovering complex subsurface structures due to the information provided from borehole measurements and the limitations of *GemPy*. Finally, it was demonstrated that an additional neighborhood system, which takes the 3D structural model into consideration decreases uncertainties in the segmentation process.

Realizing that the whole process of automatic 3D structural modeling from raw well logs consists of several complex steps, which are coupled through intensive data management and certain assumptions, further work is required to make this approach applicable to real datasets and enhance its performance and accuracy. Some promising approaches and ideas are discussed in the following chapter.

5-1 Randomness and reproducibility

As stated in section 3-2-4, the segmentation result is theoretically independent of the initial configuration $\vec{x_0}$ and $\theta_0 = (\mu_0,\ \Sigma_0)$ and influences only the number of iterations to sample the posterior distribution and find the global optimum. This statement is underlined by the findings visualized in figure 5-1, where the segmentation is repeated three times for synthetic well data with different standard variations and the segmentation error δ is plotted over 500 iterations. It is observable that for standard deviations $\sigma = 3,\ 4.75$ (blue and red plots) the course of the graphs differs, while the segmentation results after 500 iterations are very much the same. The differences of the green plots ($\sigma = 7.5$) are larger, but it can be expected that the upper two plots converge to the lower one, if the segmentation would have run for more iterations.

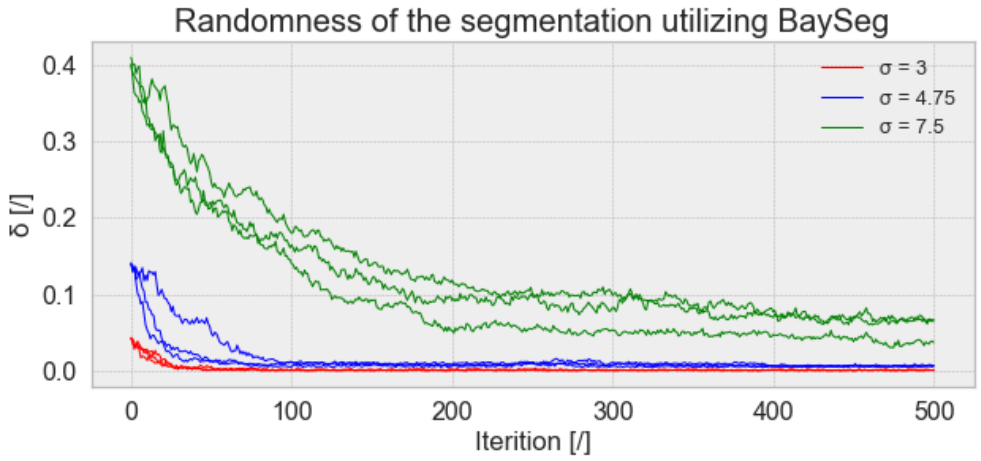

Figure 5-1: Segmentation error δ of synthetic well data over 500 iterations repeated three times with different standard deviations σ (red: 3; blue: 4.75; green: 7.5).

Moreover, the zonation based on maximizing likelihoods within zones can be seen as a filter for tiny variations in the segmentation, meaning that small anomalies are eliminated and, thus, neglected in the 3D modeling. Therefore, several similar segmentation outcomes can result in the same geological model. It has to be kept in mind that MCMC algorithms can be trapped in local optimums and, thus, lead to a strongly incorrect layering and ensuing modeling as seen in table 4-5 d) column four ($\sigma = 4.75$). Nevertheless, it has been demonstrated that the results of the segmentation and, hence, the geological models are reproducible.

5-2 Orientation data: potential and peril

As seen in section 4-2-3, the point selection for the creation of orientation data is essential if gradient information from expensive borehole imaging measurements are unavailable. The modeling results in table 4-5 (e.g. line a) with $\sigma = 5.5$) revealed that even small errors in the interface point determination (zonation) can influence the outcome strongly if these points are considered in the orientation data preparation. Furthermore, it has been demonstrated that assuming less geological changes in a smaller area, the consideration of the three closest

boreholes to each other decreases the likelihood of model destruction by orientation data. One possible solution to overcome the influence of incorrect potential field gradients on the modeling process is the usage of synthetic orientation data numerically located far away from the model part of interest. This concept was tested on the model mentioned above, which results from synthetic well data of the 4-layer horizontal model with a standard deviation of $\sigma = 5.5$. The outcome is visualized in figure 5-2. Part a) displays the geological model including the orientation data (black arrows), which are calculated from the three nearest boreholes. In part b) the gradient data are manually set to $X, Y = -1000$ with a dip of 45° (the dip is chosen to demonstrate the independence on the angle). Thus, the model area of interest depends mainly on the interface data points, which increases the accuracy of the recovered model significantly.

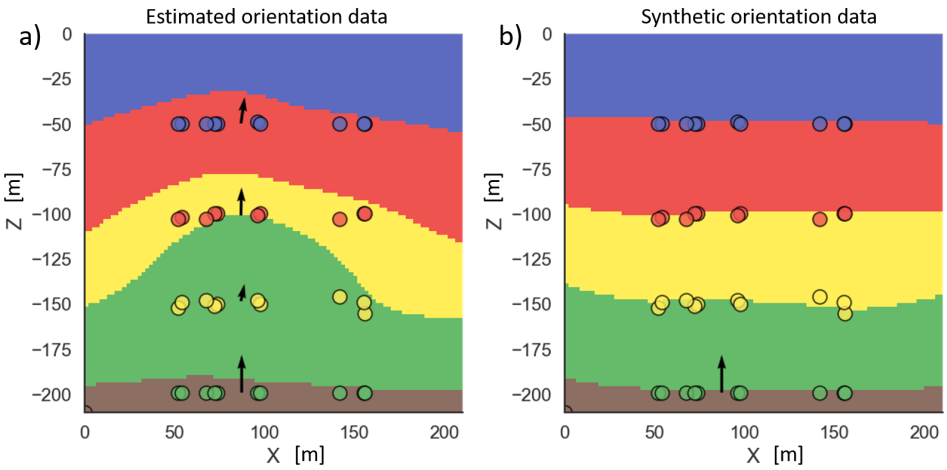

Figure 5-2: Reconstructed models from synthetic well data ($\sigma = 5.5$) created from the original model [first model in line a) at table 4-5] with (a) orientation data calculated from three nearest interface points and (b) synthetic orientation data far away from the area of interest (X, Y = -1000; dip = 45; Z = average depths of all interface data); orientation data are represented by black arrows, while the coloured dots represent interface data points.

It remains to implement this concept into the automatic 3D modeling from raw well logs and to determine a criterion, determining when to move the orientation data outside the model and when to keep the calculated potential field gradients, respectively. Besides the risk of errors, that comes with the orientation data creation from several interface points, the gradients of the potential field reveal high potential to couple well logs with other geophysical measurements. One promising principle is discussed in the next section.

Gradients as coupling

Geophysical measurements from boreholes are often considered as hard data in geological modeling methods because they are taken directly from the subsurface structures. However, seismic data are often inserted as soft data because it is obtained from qualitative observations (Stright et al., 2009). Assuming that dips of migrated seismic data are good approximations

of the real layer dipping, *GemPy* enables a simple coupling of interface points from well logs and dipping data extracted from seismic sections or models.
Dip estimator algorithms, investigating seismic models and automatically extracting dip information, are widely used in practise. For example, Marfurt (2006) developed a robust method based on volumetric estimates of reflector curvature and angular unconformities in overlapping windows to analyse dip and azimuth information. Showing that his algorithm can analyse dips across faults, unconformities, and other discontinuities, this information can be inserted in *GemPy* as hard data, while the required interface points are extracted from well logs, as shown in this work.
Furthermore, Hale (2013) proposed methods to compute fault images including strikes and dips, extract fault surfaces and estimate fault throws from seismic images, but these methods are limited in handling intersecting faults. Nevertheless, additionally utilizing this information in the structural geologic modeling with *GemPy* has the potential to overcome the fault imaging problem seen in table 4-5 d). However, further research is required to implement the above mentioned algorithms in the fully automated process developed in this work.

5-3 Implementation of stratigraphic well correlation

The major limitation of this work is the occurrence of two layers with similar properties in one borehole. This does not imply that tiny beds inside a huge section lead to a misconstruction of the model, but, for example, two sandstone layers with similar fluid content at different depth locations are not mappable. These two sandstones would be equally labelled in the segmentation with *BaySeg*, but need to be separated to get inserted in *GemPy*. If the stratigraphic pile would be identical in each borehole, meaning that the number of occurrences of each layer is constant over all boreholes, a simple zonation approach could overcome this problem. One example is numerical bed-discrimination, which analyses inflection points along well logs or methods utilizing a moving window along the data and detecting changes in mean value (Hawkins and Ten Krooden, 1979). Nonetheless, the likelihood of a constant stratigraphic pile is low and converges to zero with increasing structural complexity and an increasing number of considered boreholes.
Instead of utilizing a zonation approach to identify homogeneous zones, a stratigraphic correlation between all boreholes can be applied on the segmented data to determine the stratigraphic column over the area of interest. This approach requires a smoothing after the segmentation to ensure that outliers are precluded and, thus, not interpreted as individual layers. Recent developments in the field of stratigraphic well correlation allow the creation of stochastic models, while considering all possible realizations and their probabilities (Edwards et al., 2018).
The principle of Edwards et al.'s method is visualized in figure 5-3, where the numbers in A, B and C represent four boreholes, while D displays all possible stratigraphic realizations. In part A, the boreholes 1 to 3 are the only ones considered and, therefore, the stratigraphic model 1 in D is the only possible realization. Image B and C demonstrate possible results after the correlation with borehole 4. The middle layer in well 4 has the same properties as the green layer in borehole 1 and, thus, can be interpreted as the same stratigraphy (B) or an extra one (C), which yields the three realizations of the stratigraphic column shown in image part D.

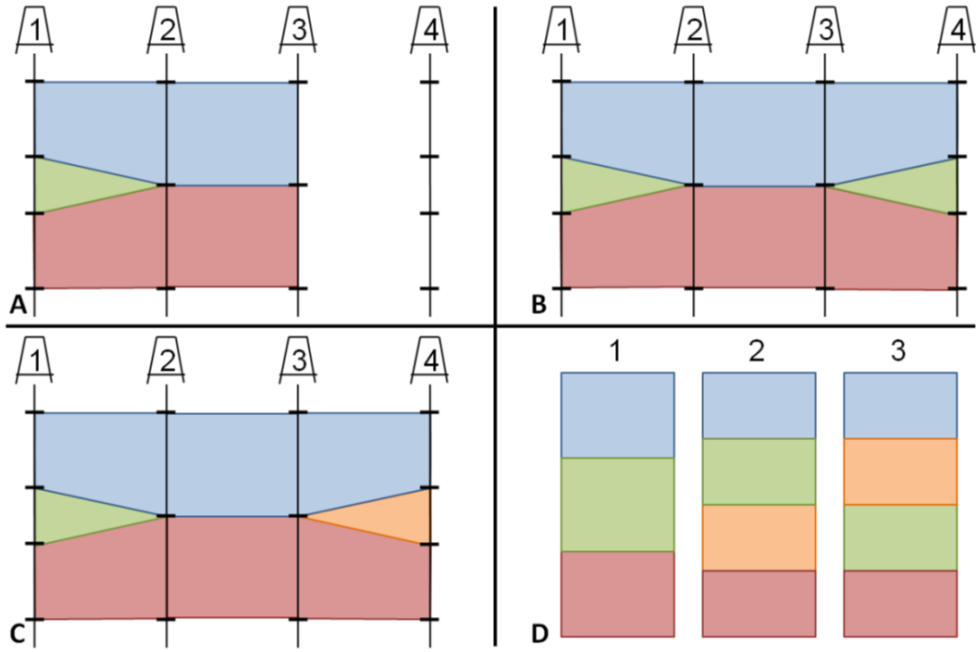

Figure 5-3: Potential stratigraphic columns while correlating wells; A: Correlation of well 1,2 & 3; B and C: Scenarios after correlating well 4; D: Possible stratigraphic columns after correlation all four wells; (from Edwards et al., 2018)

The method of creating a global stratigraphic column during the correlation of well logs from Edwards et al. (2018) is not jet well-engineered and requires further work to be applicable on real data. But so far, it represents one option to replace zonation in the automated process developed in this work and, thus, to overcome the limitation of similar layers occurring at different depth levels.

5-4 Numerical analysis of the automated process

From a numerical point of view, the automatic 3D modeling process from raw well logs, which considers the additional neighborhood system provided by *GemPy* in each segmentation iteration, is embryonic. The computational costs are much higher compared to the pure segmentation utilizing *BaySeg*. However, the interpolation of the potential field utilized in *GemPy* is simply a function, whose computation time depends mainly on the number of evaluated points and, thus, the resolution of the geological model. This is illustrated in table 5-1, where the running time is compared to the model resolution and the resulting number of points, which are to be evaluated. The computation time increases with an increase in locations at which the potential field is calculated, while the evaluated points per second remain around 5000-6000.

Table 5-1: Computational time of the fully automated 3D modelling process depended on the resolution of the *GemPy* model and the resulting number of points, which are to be evaluated. Furthermore, the evaluated points per seconds.

Computation time vs evaluated points

Resolution	Points to evaluate	Computation time	Evaluated points
25	15625	2.97/it	5260/s
50	125000	20.99/it	5952/s
100	1000000	159.21/it	6281/s

The above findings reveal that decreasing the number of evaluated points enhances the running time significantly. This can be achieved by evaluating only those points, which are considered in the additional neighborhood system. For the synthetic data utilized in this work (10 boreholes with 190 data points each), it yields 7600 points considering the 4-point stencil and 15200 points taking the 8-point stencil into account. Both scenarios are solvable in less than 3 s.
Moreover, this resolves the issue of discrepancy between the well log resolution and the grid of the *GemPy* model because it enables to evaluate the potential field at the exact well data locations. Furthermore, implementing a parameter d, that defines the distance between the well logs and the points considered in the *GemPy* neighborhood system, allows a weighting of the two neighborhood systems by their physical distance. This ensures more weight on closer data and vice versa.

Due to the complexity of the whole process of automatic 3D geological modeling from one-dimensional raw data observed in boreholes, several other things could be discussed here for further research; for example, an alternative approach for the BIC to enhance the determination of the number of layers or a weighting factor for the different borehole measurements depending on the modeling target. Nevertheless, the process developed in this work, which couples unsupervised segmentation in wells with automatic implicit modeling in a Bayesian framework, might form the basis for automated geological interpretation of well logs for exploration and research purposes.

Bibliography

Agarwal, B. L. (2006). *Basic statistics*. New Age International.

Beghtol, L. A. (1958). A statistical approach to the zonation of a petroleum reservoir. Master's thesis, Missouri University of Science and Technology.

Berger, J. O. (2013). *Statistical decision theory and Bayesian analysis*. Springer Science & Business Media.

Bergstra, J., Breuleux, O., Bastien, F., Lamblin, P., Pascanu, R., Desjardins, G., Turian, J., Warde-Farley, D., and Bengio, Y. (2010). Theano: A cpu and gpu math compiler in python. In *Proc. 9th Python in Science Conf*, volume 1.

Besag, J. (1974). Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 192–236.

Besag, J. (1986). On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 259–302.

Bolstad, W. M. and Curran, J. M. (2016). *Introduction to Bayesian statistics*. John Wiley & Sons.

Bulmer, M. G. (1979). *Principles of statistics*. Courier Corporation.

Calcagno, P., Chilès, J.-P., Courrioux, G., and Guillen, A. (2008). Geological modelling from field data and geological knowledge: Part i. modelling method coupling 3d potential-field interpolation and geological rules. *Physics of the Earth and Planetary Interiors*, 171(1-4):147–157.

Celeux, G., Forbes, F., and Peyrard, N. (2003). Em procedures using mean field-like approximations for markov model-based image segmentation. *Pattern recognition*, 36(1):131–144.

Celeux, G. and Govaert, G. (1995). Gaussian parsimonious clustering models. *Pattern recognition*, 28(5):781–793.

Chatzis, S. P. and Tsechpenakis, G. (2010). The infinite hidden markov random field model. *IEEE Transactions on Neural Networks*, 21(6):1004–1014.

Chilès, J. and Delfiner, P. (2009). *Geostatistics: Modeling spatial uncertainty*. John Wiley & Sons, New York.

Congdon, P. (2007). *Bayesian statistical modelling*, volume 704. John Wiley & Sons.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3):273–297.

Cover, T. M. and Thomas, J. A. (2012). *Elements of information theory*. John Wiley & Sons.

Dahlhaus, R. et al. (1997). Fitting time series models to nonstationary processes. *The annals of Statistics*, 25(1):1–37.

Davidson-Pilon, C. et al. (2015). *Probabilistic Programming & Bayesian Methods for Hackers*. Addison-Wesley Data & Analytics Series.

De la Varga, M. (2018). Gempy - software for 3d structural geologic implicit modeling in python.

de la Varga, M., Schaaf, A., and Wellmann, F. (2018). Gempy 1.0: open-source stochastic geological modeling and inversion. *Interpretation*.

De la Varga, M. and Wellmann, J. F. (2016). Structural geologic modeling as an inference problem: A bayesian perspective. *Interpretation*, 4(3):SM1–SM16.

Edwards, J., Lallier, F., Caumon, G., and Carpentier, C. (2018). Uncertainty management in stratigraphic well correlation and stratigraphic architectures: A training-based method. *Computers & Geosciences*, 111:1–17.

Ellis, D. V. and Singer, J. M. (2007). *Well logging for earth scientists*, volume 692. Springer.

Findley, D. F. (1991). Counterexamples to parsimony and bic. *Annals of the Institute of Statistical Mathematics*, 43(3):505–514.

Gaci, S. (2017). 4. a lithological segmentation technique from well logs using the hilbert-huang transform. *Oil and Gas Exploration: Methods and Application*, 72:61.

Gauvain, J.-L. and Lee, C.-H. (1994). Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains. *IEEE transactions on speech and audio processing*, 2(2):291–298.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2013). *Bayesian data analysis*. CRC press.

Gelman, A. and Rubin, D. B. (1996). Markov chain monte carlo methods in biostatistics. *Statistical methods in medical research*, 5(4):339–355.

Gilks, W. R. (2005). Markov chain monte carlo. *Encyclopedia of Biostatistics*.

Gill, D., Shomrony, A., and Fligelman, H. (1993). Numerical zonation of log suites and logfacies recognition by multivariate clustering. *AAPG Bulletin*, 77(10):1781–1791.

Guo, P., Chen, C. P., and Lyu, M. R. (2002). Cluster number selection for a small set of samples using the bayesian ying-yang model. *IEEE Transactions on neural networks*, 13(3):757–763.

Hale, D. (2013). Methods to compute fault images, extract fault surfaces, and estimate fault throws from 3d seismic images. *Geophysics*, 78(2):O33–O43.

Hall, B. (2016). Facies classification using machine learning. *The Leading Edge*, 35(10):906–909.

Hammersley, J. M. and Clifford, P. (1971). Markov fields on finite graphs and lattices.

Hastings, W. K. (1970). Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109.

Hawkins, D. and Ten Krooden, J. (1979). A review of several methods of segmentation. In *Geomathematical and Petrophysical Studies in Sedimentology*, pages 117–126. Elsevier.

Hawkins, D. M. (1976). Point estimation of the parameters of piecewise regression models. *Applied Statistics*, pages 51–57.

Hilchie, D. W. (1990). Wireline: A history of the well logging and perforating business in the oil fields. *Boulder, Colorado: Privately Published*.

Hoyle, I. (1986). Computer techniques for the zoning and correlation of well-logs. *Geophysical prospecting*, 34(5):648–664.

Hui, Y., Kun-peng, L., XUE-GONG, Z., and YAN-DA, L. (2000). Wavelet transform properties of well log and their application in automatic segmentation [j]. *Chinese Journal of Geophysics*, 4:017.

Jaynes, E. T. (1986). Bayesian methods: General background. *St. John's Collage and Cavendish Laboratory*.

Kindermann, R. and Snell, J. L. (1980). *Markov random fields and their applications*, volume 1. American Mathematical Society.

Koller, D. and Friedman, N. (2009). *Probabilistic graphical models: principles and techniques*. MIT press.

Lajaunie, C., Courrioux, G., and Manuel, L. (1997). Foliation fields and 3d cartography in geology: principles of a method based on potential interpolation. *Mathematical Geology*, 29(4):571–584.

Lanning, E. N. and Johnson, D. M. (1983). Automated identification of rock boundaries: An application of the walsh transform to geophysical well-log analysis. *Geophysics*, 48(2):197–205.

Liang, F., Liu, C., and Carroll, R. (2011). *Advanced Markov chain Monte Carlo methods: learning from past samples*, volume 714. John Wiley & Sons.

Ligges, U., Weihs, C., and Hasse-Becker, P. (2002). Detection of locally stationary segments in time series. In *Compstat*, pages 285–290. Springer.

Liu, K., An, T., Cai, H., Naviner, L., Naviner, J.-F., and Petit, H. (2013). A general cost-effective design structure for probabilistic-based noise-tolerant logic functions in nanometer cmos technology. In *EUROCON, 2013 IEEE*, pages 1829–1836. IEEE.

Lorensen, W. E. and Cline, H. E. (1987). Marching cubes: A high resolution 3d surface construction algorithm. In *ACM siggraph computer graphics*, volume 21-4, pages 163–169. ACM.

MacKay, D. J. (2003). *Information theory, inference and learning algorithms*. Cambridge university press.

Marfurt, K. J. (2006). Robust estimates of 3d reflector dip and azimuth. *Geophysics*, 71(4):P29–P40.

Martin, O. (2016). *Bayesian Analysis with Python*. Packt Publishing Ltd.

McKinney, W. (2011). pandas: a foundational python library for data analysis and statistics. *Python for High Performance and Scientific Computing*, pages 1–9.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092.

Moghaddamjoo, A. (1989). Constraint optimum well-log signal segmentation. *IEEE Transactions on Geoscience and Remote Sensing*, 27(5):633–641.

Moghaddamjoo, A. (1991). Estimation of the number of segments in a well-log signal via the information theoretic approach. *IEEE Transactions on Geoscience and Remote Sensing*, 29(1):177–178.

Neal, R. M. (2012). How to view an mcmc simulation as a permutation, with applications to parallel simulation and improved importance sampling. *arXiv preprint arXiv:1205.0070*.

Ofuyah, W., Olatunbosun, A., Idoko, R., and Funmi, O. (2014). Well log segmentation in spectral domain. *Journal of Energy Technologies and Policy*, 4(9).

Ouadfeul, S. (2006). Automatic lithofacies segmentation using the wavelet transform modulus maxima lines wtmm combined with the detrneded fluctuation analysis dfa.

Ouadfeul, S., Zaourar, N., Boudella, A., and Hamoudi, M. (2011). Modeling and classification of lithofacies using the continuous wavelet transform and neural network: a case study from berkine basin (algeria). *Bulletin du service géologique National-Algerie*, 22(1).

Ouadfeul, S.-A. and Aliouane, L. (2012). Lithofacies classification using the multilayer perceptron and the self-organizing neural networks. In *International Conference on Neural Information Processing*, pages 737–744. Springer.

Patil, A., Huard, D., and Fonnesbeck, C. J. (2010). Pymc: Bayesian stochastic modelling in python. *Journal of statistical software*, 35(4):1.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.

Rogers, S. J., Fang, J., Karr, C., and Stanley, D. (1992). Determination of lithology from well logs using a neural network (1). *AAPG bulletin*, 76(5):731–739.

Saucier, A. and Muller, J. (2002). Using principal component analysis to enhance the generalized multifractal analysis approach to textural segmentation: Theory and application to microresistivity well logs. *Physica A: Statistical Mechanics and its Applications*, 309(3-4):419–444.

Serra, O. (1983). Fundamentals of well-log interpretation. *Developments in Petroleum Science*.

Shannon, C. E. (1948). Communication theory of secrecy systems. *Bell Labs Technical Journal*, 28(4):656–715.

Shi, Y., Wu, X., Fomel, S., et al. (2017). Finding an optimal well-log correlation sequence using coherence-weighted graphs. In *2017 SEG International Exposition and Annual Meeting*. Society of Exploration Geophysicists.

Stright, L., Bernhardt, A., Boucher, A., Mukerji, T., and Derksen, R. (2009). Revisiting the use of seismic attributes as soft data for subseismic facies prediction: Proportions versus probabilities. *The Leading Edge*, 28(12):1460–1468.

Testerman, J. et al. (1962). A statistical reservoir-zonation technique. *Journal of Petroleum Technology*, 14(08):889–893.

Velis, D. R. (2005). Segmentation of well-log data. In *9th International Congress of the Brazilian Geophysical Society*.

Vermeer, P. and Alkemade, J. (1992). Multiscale segmentation of well logs. *Mathematical Geology*, 24(1):27–43.

Wang, H., Wellmann, J. F., Li, Z., Wang, X., and Liang, R. Y. (2017). A segmentation approach for stochastic geological modeling using hidden markov random fields. *Mathematical Geosciences*, 49(2):145–177.

Webster, R. (1973). Automatic soil-boundary location from transect data. *Journal of the International Association for Mathematical Geology*, 5(1):27–37.

Webster, R. and Wong, I. (1969). A numerical procedure for testing soil boundaries interpreted from air photographs. *Photogrammetria*, 24(2):59–72.

Wellmann, J. F. (2013). Information theory for correlation analysis and estimation of uncertainty reduction in maps and models. *Entropy*, 15(4):1464–1485.

Wellmann, J. F., de la Varga, M., Murdie, R. E., Gessner, K., and Jessell, M. (2017). Uncertainty estimation for a geological model of the sandstone greenstone belt, western australia–insights from integrated geological and geophysical inversion in a bayesian inference framework. *Geological Society, London, Special Publications*, 453.

Wellmann, J. F., Horowitz, F. G., Schill, E., and Regenauer-Lieb, K. (2010). Towards incorporating uncertainty of structural data in 3d geological inversion. *Tectonophysics*, 490(3-4):141–151.

Wellmann, J. F. and Regenauer-Lieb, K. (2012). Uncertainties have a meaning: Information entropy as a quality measure for 3-d geological models. *Tectonophysics*, 526:207–216.

Wu, X. and Nyland, E. (1987). Automated stratigraphic interpretation of well-log data. *Geophysics*, 52(12):1665–1676.

Xu, C., Yang, Q., and Torres-Verdín, C. (2016). Bayesian rock classification and petrophysical uncertainty characterization with fast well-log forward modeling in thin-bed reservoirs. *Interpretation*, 4(2):SF19–SF29.

Yamane, T. (1973). *Statistics: An introductory analysis*. Harper & Row New York, NY.

Zhang, Y., Brady, M., and Smith, S. (2001). Segmentation of brain mr images through a hidden markov random field model and the expectation-maximization algorithm. *IEEE transactions on medical imaging*, 20(1):45–57.

Appendix A

Supervised segmentation using Scikit-learn

A-1 Basic principle

Scikit-learn is a Python package that provides state-of-the-art machine learning algorithms implemented in the computational language Python and was introduced by Pedregosa et al. (2011). It aims to bring machine learning tools to non-specialists. In 2016, Hall stated that geoscientists could benefit from Scikit-learn greatly, since the volume of data sets in geoscience become larger and larger. Furthermore, Hall (2016) demonstrated that one of Scikit-learn's supervised clustering algorithms is capable to segment well data sufficiently. Its methodology is explained in the following and used as a comparison for the unsupervised segmentation results achieved in this work.

The method is based on an SVN or so called support-vector machine, which maps the input data non-linearly to high-dimensional feature space and separates it by a linear decision surface. The construction of the decision surface is implemented by means of a training data set, whose labelling is known (Cortes and Vapnik, 1995). Therefore, while applying the approach to well logs from several boreholes, one borehole with known segmentation is split from the data and utilized to train the algorithm and later on, to evaluate the algorithm's accuracy. As described in the preprocessing section, the data are also standardized to ensure Gaussian distribution. To train the algorithm or in other words to select a model, different parameters, which affect the accuracy of the segmentation need to be set. Therefore, when comparing the segmentation approach utilized in this work with the one suggested by Hall (2016), suitable values for the C and *gamma* parameters are estimated. *Gamma* controls the influence radius of a single training example, where low values correspond to a large radius and vice versa, while the smoothness of the decision surface is steered by the parameter C. Its value handles the trade-off between misclassification of training data points and simplicity of the decision surface, where a low C value forces a high smoothness. A more detailed description of the theory can be found in Cortes and Vapnik (1995).

Once a suitable model is selected, the classifier is trained using the separated borehole with

known labels. Eventually, the algorithm is able to predict the segmentation for all well logs (Hall, 2016). The next section lists the segmentation results for synthetic well data with varying standard deviation σ. In chapter 4, the results of the SVN method are compared to the segmentation utilized and refined in this work.

A-2 Segmentation results

Table A-1: Error of segmentation δ_{SVN} for testing different C values for synthetic wells data with varying standard deviation, while $gamma = 1$. Segmentation utilizing the SVN approach suggested by Hall (2016).

C \ σ		**3**	**4**	**4.75**	**5.5**	**7.5**
0.1	δ_{SVN} :	0.038	0.075	0.128	0.196	0.302
1	δ_{SVN} :	0.033	0.067	0.112	0.181	0.269
10	δ_{SVN} :	0.024	0.048	0.087	0.150	0.207
100	δ_{SVN} :	0.016	0.021	0.055	0.090	0.113
1000	δ_{SVN} :	0.007	0.011	0.031	0.043	0.057
10000	δ_{SVN} :	0.007	0.007	0.015	0.022	0.030
100000	δ_{SVN} :	0.007	0.007	0.011	0.018	0.023
1000000	δ_{SVN} :	0.007	0.007	0.011	0.018	0.022

Table A-2: Error of segmentation δ_{SVN} for testing different $gamma$ values for synthetic wells data with varying standard deviation, while $C = 10000$. Segmentation utilizing the SVN approach suggested by Hall (2016).

$gamma$ \ σ		**3**	**4**	**4.75**	**5.5**	**7.5**
0.01	δ_{SVN} :	0.036	0.068	0.127	0.204	0.307
0.1	δ_{SVN} :	0.027	0.047	0.095	0.175	0.255
1	δ_{SVN} :	0.007	0.007	0.015	0.022	0.030
10	δ_{SVN} :	0.004	0.009	0.012	0.019	0.023
100	δ_{SVN} :	0.025	0.035	0.033	0.037	0.036
1000	δ_{SVN} :	0.037	0.039	0.039	0.039	0.039

Appendix B

Figures, tables and code availability

B-1 Additional figures and tables

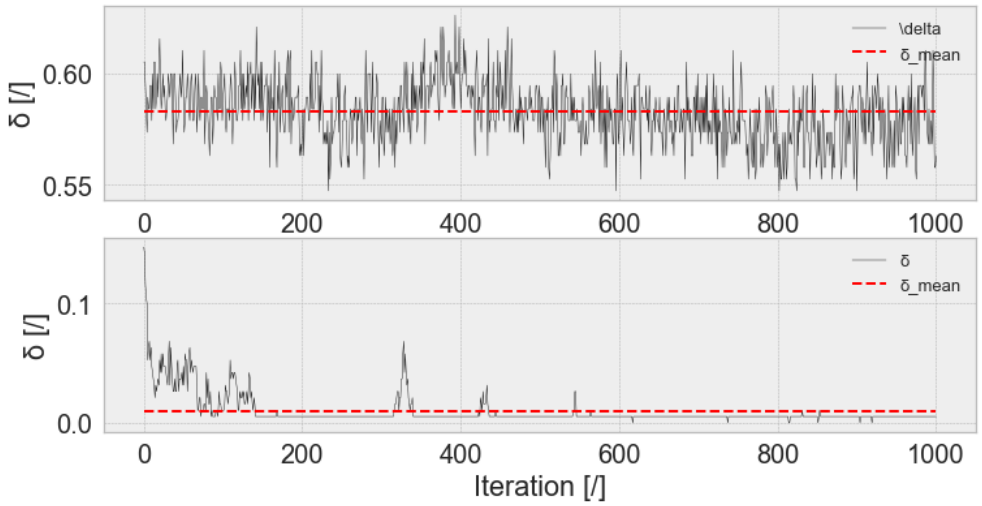

Figure B-1: Segmentation error δ for synthetic well data with a standard variation $\sigma = 4$ over 1000 iterations and its mean (dashed red line) for a fixed β during the segmentation in the upper image and a updated β in the lower one.

Table B-1: Segmentation error in the last iteration δ_{last} and its mean $\overline{\delta}$ for synthetic well logs from different number of boreholes $\#$ and standard deviations $\sigma = 4, 5.5$ with updated β over 100 iterations (other parameters: $beta\_jump\_length = 0.02$, $mu\_jump\_length = 0.0005$, $cov\_volume\_jump\_length = 0.00005$, $theta\_jump\_length = 0.0000005$)

σ \ $\#$		**1**	**4**	**10**	**25**
4	$\overline{\delta}$	0.114	0.045	0.019	0.017
	δ_{last}	0.098	0.039	0.014	0.010
5.5	$\overline{\delta}$	0.214	0.065	0.049	0.037
	δ_{last}	0.222	0.071	0.033	0.029

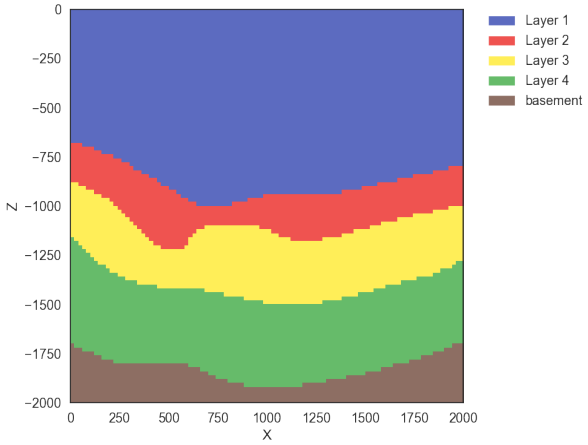

Figure B-2: Reconstruction of the fault model in table 4-5 d) with 50 randomly placed boreholes utilizing the fully automated 3D modeling process based on raw well logs.

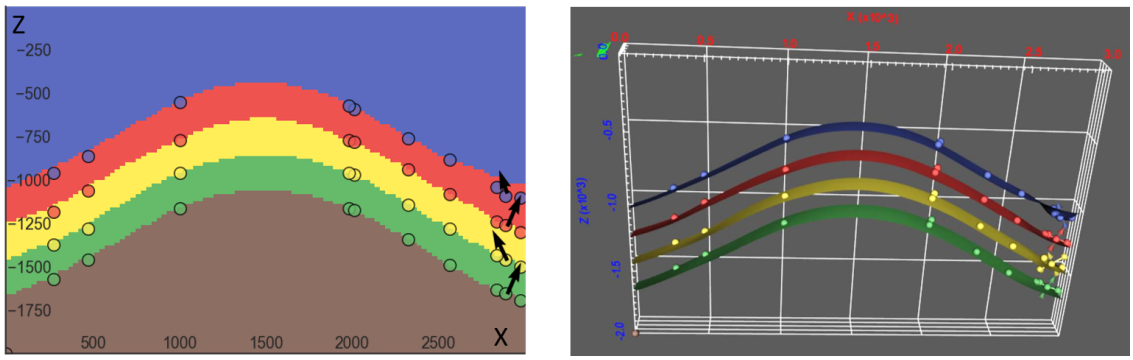

Figure B-3: Structural geologic model of a 3-layer anticline structure resulting from synthetic well data with standard deviation 4.75. Left-hand side: X-Z section; Right-hand side: 3D model.

B-2 Code availability

All numerical tools utilized in this work are open-source and implemented in the computational language Python. The structural geologic modeling tool *GemPy* introduced by de la Varga et al. (2018) can be downloaded from https://github.com/cgre-aachen/gempy. The unsupervised segmentation approach *BaySeg* by Wang et al. (2017) is available at https://github.com/cgre-aachen/bayseg. Moreover, the data analysis package *pandas* developed by McKinney (2011), utilized mainly for the data management in this work, can be found at https://github.com/pandas-dev/pandas. The machine learning library Scikit-learn introduced by Pedregosa et al. (2011) and used for the supervised segmentation as a comparison is ready for more geophysical application at https://github.com/scikit-learn/scikit-learn. Eventually, the developments of this work, including code, data and notebooks are accessible at https://github.com/cgre-aachen/MSc_theses/tree/master/Well_analysis_BaySeg_GemPy_coupled.