Ph.D. Thesis

Inverse Problems in Geosciences:

Modelling the Rock Properties of an Oil Reservoir

Katrine Lange

DTU Compute
Department of Applied Mathematics and Computer Science
Technical University of Denmark
Kongens Lyngby
PHD-2013-296

DTU

Preface

This thesis has been submitted as a partial fulfilment of the requirements for the Danish Ph.D. degree at the Technical University of Denmark (DTU). The work has been carried out during the period from March 1st 2010 to February 28th 2013, jointly in the Section of Scientific Computing at the Department of Applied Mathematics and Computer Science and at the Center for Energy Resources Engineering (CERE) both at DTU. The Ph.D. project has been part of the DTU Compute graduate school ITMAN.

Main supervisor has been Professor Klaus Mosegaard, DTU Compute and CERE. Co-supervisors have been Professor Per Christian Hansen, DTU Compute, and Professor Erling Stenby, DTU Chemistry and CERE. During my stay abroad at Stanford University Associate Professor Jef Caers, Department of Energy Resources Engineering, has also been involved in the study.

The Ph.D. project was sponsored by the Danish Council for Independent Research | Technology and Production Sciences (grant no. 274-09-0332) and DONG Energy. For my research stay at Stanford University I received external funding from *Otto Mønsteds Fond* and *Knud Højgaards Fond*.

Kongens Lyngby, February 28, 2013

Katrine Lange

Summary

Inverse Problems in Geosciences: Modelling the Rock Properties of an Oil Reservoir

Even the most optimistic forecasts predict that Danish oil production will decrease by 80% in the period between 2006 and 2040, and only a strong innovative technological effort can change that. Due to the geological structures of the subsurface in the Danish part of the North Sea, Denmark is currently missing out on approximately 70% of the oil, which is left behind, trapped in unreachable parts of the reservoirs.

An increase in the oil recovery rate can be achieved by better planning and optimisation of oil production. Both require an improved description of the rock properties of the subsurface of the reservoirs. Hence the focus of this work has been on acquiring models of spatial parameters describing rock properties of the subsurface using geostatistical a priori knowledge and available geophysical data. Such models are solutions to often severely under-determined, inverse problems.

The focus of the study has been on the computational aspects of inferring such models. Reservoir modelling is a large-scale problem with great computational complexity. The work should be seen as a first part of a foundation for one day, when the computational resources are available, being able to

handle the large scale problems of the petroleum industry. But for now most of the study is based on simplified and idealised models.

We have proposed a method for efficient and accurate interpolation of rock properties from seismic data. It is based on a recently published paper on interpolation of rock properties that breaks with the dominating influence of spatial coordinates in traditional interpolation methods. The thesis contains work involving a test case study of the method demonstrating how the interpolation in attribute space ensures the geological structures of the computed models and how the method can be further improved by an orthogonal transformation of the attribute space.

We have formulated a closed form expression of an a priori probability density function that quantifies the statistical probability of models describing the rock properties of a reservoir. This can be used to evaluate the probability that a model adhere to prior knowledge by having specific multiple-point statistics, for instance, learned from a training image. Existing methods efficiently sample an a priori probability density function to create a set of acceptable models; but they cannot evaluate the probability of a model.

We have developed and implemented the Frequency Matching method that uses the closed form expression of the a priori probability density function to formulate an inverse problem and compute the maximum a posteriori solution to it. Other methods for computing models that simultaneously fit data observations and honour a priori knowledge are not capable of computing the maximum a posteriori solution. Instead they either sample the posterior probability density function or they sample the a priori probability density function to optimise the likelihood function.

This thesis consists of a summary report and seven research papers submitted, reviewed and/or published in the period 2010 - 2013.

Resumé

Inverse Problemer i Geosciences: Modellering af Bjergartsegenskaber i et Oliereservoir

Selv de mest optimistiske prognoser forudser at den danske olieproduktion vil falde med 80% i perioden fra 2006 til 2040, og kun en kraftig, teknologisk, innovativ indsats kan ændre dette. På grund af de geologiske strukturer i undergrunden af den danske del af Nordsøen går Danmark glip af omkring 70% af olien, som bliver efterladt tilbage, fanget i utilgængelige dele af reservoirerne.

En stigning i olieindvindingsraten kan opnås ved at planlægge og optimere produktionen bedre. Begge dele kræver en bedre beskrivelse af bjergarternes egenskaber i reservoiret. Dette arbejde har derfor haft fokus på at generere modeller, der beskriver rumlige parametre i undergrunden, ved brug af geostatistisk *a priori* viden og tilgængelige geofysiske data. Disse modeller er ofte løsninger til stærkt underbestemte, inverse problemer.

Fokus i dette studium har været på de beregningsmæssige aspekter ved generering af sådanne modller. Modellering af oliereservoirer er et stor-skala problem med høj bereningsmæssig kompleksitet. Dette arbejde skal derfpr ses som en første del af et fundament, der en dag hvor de beregningsmæssige

ressurcer er til stede, vil gøres os i stand til at håndtere oileindustriens stor-skala problemer. Men indtil videre er det meste at studiet baseret på forenklede problemstillinger og idealiserede modeller.

Vi har foreslået en metode, der effektivt og præcist interpolerer bjergartsegenskaber ved brug af seismisk data. Den er basered på en videnskabelig artikel, der udkom for nylig, og som gør op med den dominerende indflydelse af rumlige koordinater i traditionelle interpolationsmetoder. Denne afhandling indeholder et eksempel på metoden, der viser hvordan interpolation i rummet udspændt af seismiske attributter sikrer en meningsfuld geologiske struktur af de beregnede modeller, og der viser, hvordan metoden kan blive forbedret yderligere ved indførsel af en ortogonal transformation af de seismiske attributter.

Vi har formuleret et lukket udtryk for en *a priori* sandsynlighedsfordeling, der kvantificere statistiske egenskaber af modeller af bjergartsegenskaber af et oliereservoir. Dette udtryk kan bruges til at evaluere sandsynligheden for, at en model har den efterspurgte multiple-punkt statistik, som for eksempel kan være givet ved et træningsbillede. Eksisterende metoder kan effektivt sample en *a priori* sandsynlighedsfordeling og dermed generere en mængde af acceptable modeller, men de kan ikke evaluere sandsynligheden af den enkelte model.

Vi har udviklet og implementeret *Frequency Matching*-metoden, der bruger det lukkede udtryk for en *a priori* sandsynlighedsfordeling til at formulere inverse problemer og beregne den model, der har størst *a posteriori* sandsynlighed. Andre metoder, der bruges til at genere modeller, der simultant opfylder observeret data og tilfredsstiller *a priori* forventninger, er ikke i stand til at beregne modellen med størst *a posteriori* sandsynlighed. I stedet sampler de enten *a posteriori* sandsynlighedsfordelingen, eller de sampler *a priori* sandsynlighedsfordelingen for dernæst at maksimere sandsynlighedsfunktionen for datafittet.

Ph.d.-afhandlingen består af en opsamlende rapport samt syv videnskabelige artikler, der er indsendt, bedømt og/eller udgivet i perioden 2010 – 2013.

Contents

List of Figures

Symbols and Abbreviations

The following is a list of symbols and abbreviations used in the thesis. Be aware that some symbols will have multiple meanings. However, for each appearance it will be clear from the context which meaning the symbol refers to.

Generally, upper case bold font letters denote matrices, lower case bold font letters denote vectors, lower case italic font letters denote scalars or scalar functions, upper case italic font letters denote random variables, and calligraphic letters denote sets.

The list is not a complete list. Symbols, or meanings of symbols, which only appear a few times, may have been omitted.

Symbols

$\mathbf{0}$	matrix of all zeros
α	weighting parameter for prior term in the FM method
$\boldsymbol{\alpha}^*$	coefficient vector of an orthogonal transformation, $\boldsymbol{\alpha}^* \in \mathbb{R}^m$
ϵ_i	count of pattern i of the underlying distribution for the image
ϵ_i^{TI}	count of pattern i of the underlying distribution for the TI
$\boldsymbol{\theta}$	vector of parameters of the covariance function C
$\boldsymbol{\theta}^*$	vector of optimal parameters of the covariance function C found by MLE
λ_i	the weight assigned to the ith data observation in kriging interpolation

$\boldsymbol{\lambda}$	vector of kriging weights, $\boldsymbol{\lambda} =\in \mathbb{R}^n$
π_i	count of the ith pattern in a frequency distribution $\boldsymbol{\pi}$
$\boldsymbol{\pi}$	frequency distribution of an image
$\boldsymbol{\pi}^{\mathrm{TI}}$	frequency distribution of a training image
$\overline{\boldsymbol{\pi}}$	frequency distribution of a perturbed image $\overline{\mathcal{Z}}$
$\rho_{\mathbf{m}}(\mathbf{m})$	a priori pdf of the model parameters $\mathbf{m}$
$\sigma_{\mathbf{m}}(\mathbf{m})$	a posteriori pdf of the model parameters $\mathbf{m}$
ω	template function, returns the indices of the neighbouring voxels
C	covariance function of the kriging estimator
c	χ^2 distance function to measure distance between two frequency distributions
$\mathbf{C_d}$	covariance matrix of the noise of the data observations $\mathbf{d}^{\mathrm{obs}}$
$\mathbf{C_m}$	covariance matrix of a Gaussian a priori pdf of the model parameters
$\mathbf{C}_{\widetilde{\mathbf{m}}}$	covariance matrix of the a posteriori pdf for a linear inverse problem with a Gaussian a priori pdf
$\mathbf{d}$	vector of data, $\mathbf{d} \in \mathbb{R}^n$
$\mathbf{d}^{\mathrm{obs}}$	noisy observed data, $\mathbf{d}^{\mathrm{obs}} \in \mathbb{R}^n$
$\mathcal{D}_k$	set of voxels centred in voxel k, $\mathcal{D}_k \subset \mathcal{Z}$
$\mathbf{e}$	column vector of all ones
$f_j(\mathbf{u})$	value of the jth trend function of universal kriging at location $\mathbf{u}$
$\mathbf{F}$	matrix of trend function values used in the universal kriging system, $\mathbf{F} \in \mathbb{R}^{n\times k}$
$\mathbf{f}$	vector of trend function values used in the universal kriging system, $\mathbf{f} \in \mathbb{R}^k$
g	mapping operator from model space to data space, $g : \mathbb{R}^m \to \mathbb{R}^n$
g^{-1}	inverse mapping operator from data space to model space, $g^{-1} : \mathbb{R}^n \to \mathbb{R}^m$
$\mathbf{G}$	coefficient matrix for a linear forward problem, $\mathbf{G} \in \mathbb{R}^{m\times n}$
$\mathbf{K}$	matrix of data-to-data covariances used in kriging
$\mathbf{k}$	vector of data-to-unknown covariances used in kriging

$L(\mathbf{m})$	likelihood function of the model parameters $\mathbf{m}$
m	dimension of the model space, i.e., $\mathbf{m} \in \mathbb{R}^m$
$\widehat{m}^*$	dimension of the transformed and reduced attribute space
$m(\mathbf{u})$	value of the trend model of a kriging estimator at location $\mathbf{u}$
$\mathbf{m}$	vector of model parameters, $\mathbf{m} \in \mathbb{R}^m$
$\mathbf{m}^{\mathrm{FM}}$	the FM model of an inverse problem
$\mathbf{m}^{\mathrm{MAP}}$	model maximising the a posteriori pdf
$\mathbf{m}_0$	mean vector of a Gaussian a priori pdf of the model parameters
$\widetilde{\mathbf{m}}$	mean vector of the a posteriori pdf for a linear inverse problem with a Gaussian a priori pdf
n	dimension of the data space, i.e., $\mathbf{d}^{\mathrm{obs}} \in \mathbb{R}^n$
n	number of voxels in the neighbourhood of an inner voxel
N	number of voxels in an image
$n^{\mathcal{Z}}$	total count of patterns in the frequency distribution of $\mathcal{Z}$
n^{TI}	total count of patterns in the frequency distribution of the TI
$\mathcal{N}_k$	set of neighbouring voxel for the kth voxel of an image, i.e., $\mathcal{N}_k = \omega(k)$
$\mathcal{N}$	subset of data points used for the kriging interpolation at a given location
p_k	pattern value of the pattern centred in voxel k
$\mathbf{p}_i$	the ith PLS component
$\mathbf{u}_i$	position vector in attribute space of the known rock property value z_i, $\mathbf{u}_i \in \mathbb{R}^m$
$\mathbf{u}_0$	position vector in attribute space of the unknown rock property value z_0, $\mathbf{u}_0 \in \mathbb{R}^m$
$\mathbf{U}$	attribute matrix collecting the position vectors row-wise, $\mathbf{U} \in \mathbb{R}^{n \times m}$
$\mathbf{U}_{:,j}$	values of the jth column of the attribute matrix holding the values of the jth seismic attribute
v	number of voxel categories in an image
$\mathbf{v}_i$	the ith principal component

$Z^*(\mathbf{u})$	the kriging estimator at location $\mathbf{u}$
z_i	known rock property value at location $\mathbf{u}_i$ to use for interpolation
$\mathbf{z}$	vector of known rock property values, $\mathbf{z} \in \mathbb{R}^n$
z_i^{est}	estimated value of the rock property at the ith location
$\mathbf{z}^{est}$	vector of estimated values of the rock property
z_0	unknown rock property value to be interpolated at location $\mathbf{u}_0$
$\mathcal{Z}$	image consisting of a set of voxels
$\mathcal{Z}_{\text{in}}$	set of inner voxels of the image $\mathcal{Z}$
z_k	variable describing the value of the kth voxel of an image
$\overline{\mathcal{Z}}$	perturbed image
$\mathbf{z}_k$	vector of ordered variables describing the values of the voxels in the neighbourhood of the kth voxel

Abbreviations

DISTPAT	distance-based pattern modelling algorithm (Honarkhah, 2011)
EOR	enhanced oil recovery
FM	frequency matching (Lange et al, 2012)
GSLIB	geostatistical software library (Deutsch and Journel, 1998)
MAP	maximum a posteriori
MC	Monte Carlo
MCMC	Markov chain Monte Carlo
MLE	maximum likelihood estimation
MPS	multiple-point statistics
PCA	principal component analysis
pdf	probability density function
PLS	partial least squares

SA	simulated annealing
SIMPAT	pattern-based geostatistical sequential simulation algorithm (Arpat, 2005)
SNESIM	single normal equation simulation (Strebelle, 2002)
TI	training image

CHAPTER 1

Introduction

The motivation for the work presented in the thesis is given in Section 1.1. Section 1.2 is an overview of relevant literature on the use of geostatistical prior information for inverse problems. In Section 1.3 we summarise the objectives and key contributions of the study, and in Section 1.4 we outline the remainder of the thesis.

1.1 Motivation: Denmark as an Oil Producing Country

Denmark is, and has been for many years, an oil producing country with currently 19 producing oil and gas fields in the Danish part of the North Sea. The first oil was produced from the Dan field in 1972 and to this day discoveries of new oil fields are made. For example in 2011, oil was found in two exploration drillings.

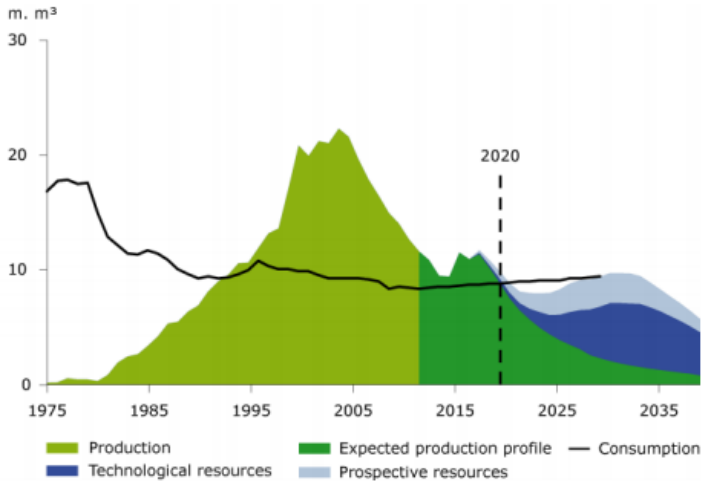


Figure 1.1: Danish oil production – past and future. Source: *Danish Energy Agency: Oil and Gas Production in Denmark 2011*.

Figure 1.1 shows the Danish oil production from 1975 to the present time and the official prognosis from the Danish Energy Agency for the production projections for the next 20 years. The figure illustrates the unfortunate fact that production from the Danish oil fields has peaked. This, as well as increasing consumption, may give rise to concern. The forecasts show that in 2020, Denmark will no longer be self-sufficient and thereafter the expected production is rapidly decreasing.

Nevertheless, what the figure also shows is that we do have prospective and technological resources to keep production near the level of the consumption for some years. This is due to the average oil recovery rate in the Danish North Sea oil fields being estimated at less than 30%. Compared to what has been produced so far, there are still huge amounts of oil in the Danish North Sea, and we therefore have the possibility to exploit the oil fields more fruitfully than what they currently are.

However, the bad news is that the oil that is not currently being produced

is the so-called expensive oil. Some of this oil is considered a technological resource, which means we currently have the technological skills and knowledge to produce it, but it is not economically defensible. It is therefore a very important research task to develop new methods with the objective of increasing the recovery rate by reaching the oil that is currently being left behind, regardless of it being for technological or economical reasons.

1.1.1 Development of an Oil Reservoir

An oil reservoir is a formation of porous rock overlaid by impermeable layers. Water, oil and gas are trapped in the microscopic pores of the rock under high pressure and temperature. The pores are interconnected making the rock permeable, which means a difference in pressure will create a fluid flow.

To develop an oil field, producer wells are drilled. These connect the subsurface reservoirs to surface facilities. This is illustrated in Figure 1.2, which shows a sketch of an offshore oil field with multiple wells connecting the reservoirs to facilities on the seabed and the water surface. From the surface facilities the oil is transported to refineries for processing.

In general, the depletion process consists of two production phases. The first production phase is a passive phase. Here the pressure of the reservoir acts to create a flow from the subsurface to the surface facilities where the oil flows by itself. By the time the flow ceases, because the pressure is no longer high enough to maintain it, only a small quantity of the oil in the reservoir will have been produced. The secondary production phase then takes over and a new set of wells is drilled. These are injector wells and they are not used to produce oil but instead to inject water or gas into the reservoir. This increases the pressure in the reservoir and the water or CO_2 then acts to create a flow by pushing the oil away from the injector wells and, ideally, toward the producer wells. This type of production process is called water flooding or CO_2 flooding. Unfortunately, both of these production processes normally result in recovery rates of only 10% to 50%.

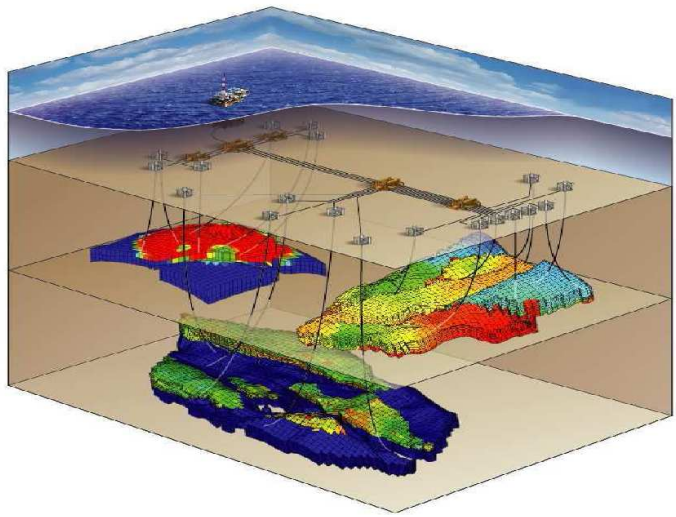


Figure 1.2: An offshore oil field. Subsurface reservoirs are connected by wells to units on the seabed again connected to a surface production unit.

Although not economically defensible at the present time, a third production phase using enhanced oil recovery (EOR) techniques can be introduced. EOR techniques are, among others, chemical flooding by injection of surfactants or polymers, microbial EOR, steam injection and *in situ* combustion. The thesis will not go into further details with EOR but only mention these methods as an example of potential means to use the technological resources we have available but which are not yet economically defensible.

A big research topic is production optimisation of reservoirs. Within this topic work is done on increasing the recovery rate of an oil reservoir without costly investments but simply by optimising the depletion process. Optimising the development of an oil field using water flooding or CO_2 flooding consists of two key problems. First, the location, type and other specifications of the producer and injector wells need to be decided upon. Second, the strategy controlling the injection and production must be determined.

The consequences of both decisions can be investigated by computer programs set up to simulate a reservoir. Designing such programs is a far from trivial task and contains great uncertainties. However, once the properties of the reservoir are estimated, virtual wells can be drilled and mathematical models used to model physical laws can describe the potential fluid flow. This will be used to estimate the outcome of a production strategy.

A crucial aspect for successfully simulating a reservoir is the ability to predict the fluid flow. The fluid flow is vital as it governs the oil production. Hence, the more accurately we can simulate the fluid flow the more precisely we can predict production. Recall that the fluid flow was due to pressure gradients and reservoir fluids being trapped in permeable zones of porous reservoir rock surrounded by impermeable layers. To correctly simulate flow we therefore need to know rock properties such as porosity and permeability of the subsurface of the reservoir. Porosity is the percentage of pore volume or void space, or the volume within the rock that can contain fluids. Permeability is the ability of the rock to transmit fluids.

1.1.2 Describing the Unknown Subsurface from Data

In theory, flow simulations require knowledge of the rock properties of the subsurface in all points in the physical space that the reservoir spans. However, the mathematical methods used to simulate the flow typically discretise the physical space into a grid of blocks, and the rock properties are then assumed constant within each block. For reservoirs of realistic size this amounts to millions of blocks, which means we need to know the rock properties in millions of inaccessible locations in the subsurface. This may seem a nearly impossible task as our only option is to infer these values from the data that is available.

The situation is different from oil field to oil field and what type of data is available will vary. Often we will have one or more of the following three types of data:

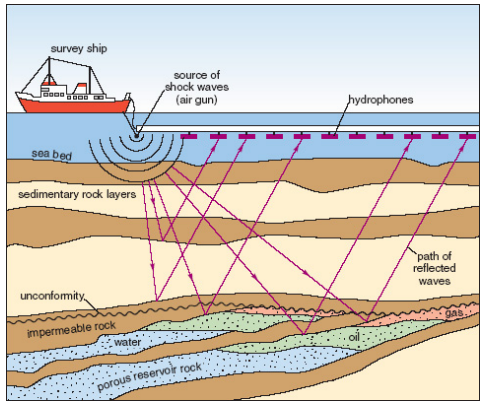


Figure 1.3: Illustration of offshore seismic exploration. A seismic source generates seismic waves that are reflected from different layers in the subsurface and then recorded on hydrophones.

1.1.2.1 Seismic Surveys

Reflection seismology is a method of exploration geophysics that uses the principles of seismology to estimate the properties of the Earth's subsurface from reflected seismic waves. Figure 1.3 illustrates the concept of seismic exploration: a source generates seismic waves which travel through the different layers of the subsurface. Seismic waves change direction when passing the transitions between layers and some will be reflected back to the surface. The arrival times are recorded at different locations using a series of hydrophones. By repeating the process at numerous locations and comparing the recorded arrival times elastic parameters such as density, velocity or impedance of the Earth can be inferred. Knowledge of the elastic parameters can then be used to infer rock properties such as porosity or permeability.

Seismic surveys are a relatively cheap and non-invasive way to gain information about the subsurface of an oil field but, as explained, these do not contain direct measurements of neither porosity nor permeability, and rock properties are only estimated from the elastic parameters.

1.1.2.2 Well Logs

Well logging, also known as borehole logging, is the practice of making a detailed record (a well log) of the geologic formations penetrated by a borehole. The log may be based either on visual inspection of samples brought to the surface (geological logs) or on physical measurements made by instruments lowered into the hole (geophysical logs). Well logging can be done during any phase of a well's history; drilling, completing, producing and abandoning. Well logs can contain observations of the porosity and the permeability of the subsurface penetrated by the well.

Well logs can provide us estimates of the rock properties in a small number of blocks in the grid. However, the uncertainty of the measurements should be taken into account as it translates to uncertainties of the estimates.

Usually, only a few well logs are available as the cost of drilling wells is very high, and well log data is therefore only available in locations with existing wells.

1.1.2.3 Production History

When an oil field is producing there will be production data available. Production data consists of time series of observed values of the parameters related to the production. These are typically the amount of water or CO_2 injected in each of the injectors, the bottom hole pressure in the injectors, amount of oil and water produced in each of the producers, and the water and oil ratio in each producer. The data can be used by history matching, which is a technique searching for models of the rock properties of the subsurface that result in the given production data.

As with seismic surveys, production data does not contain direct observations of the rock properties, which are only inferred indirectly.

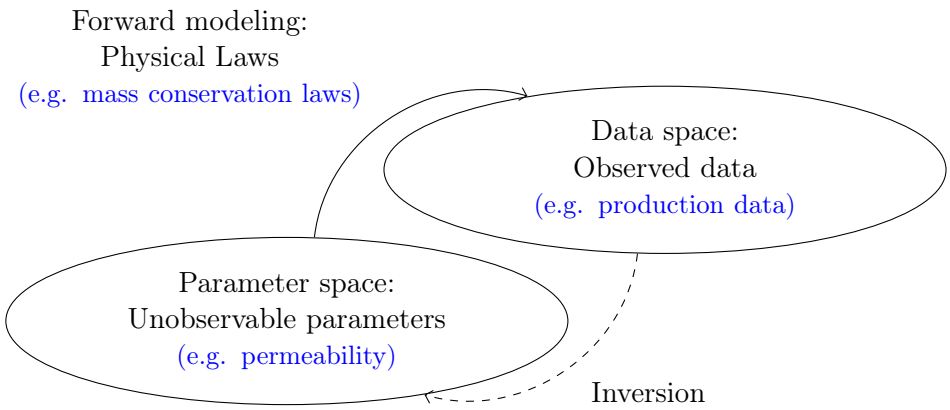


Figure 1.4: The concept of forward modelling versus inversion. Whereas forward modelling can be used to map from parameter space to data space there often exist no inverse mapping to compute the unobservable parameters from the observed data.

1.1.3 The Hidden Remains Uncertain

Describing the rock properties of the subsurface is no straightforward task. Even in the best cases where all three types of data are available, we will only have a few direct observations of porosity and permeability. The rest is a set of data describing a wide range of different parameters that, via physical laws, can be connected to the rock properties.

In fact, determining porosity or permeability from either elastic parameters acquired by seismic surveys or production data is what in mathematics is called an inverse problem, see Figure 1.4. An inverse problem is the kind of problem that arises when trying to compute unobservable parameters based on observable data only knowing the mapping operator from parameter space to data space.

For instance, regarding the problem at hand, we would like to compute the unobservable rock properties of the subsurface given the production data. Physical laws describing fluid flow let us determine the production data for a given reservoir if we know its permeability and porosity; this is called the forward problem or forward modelling. But there exists no physical law describing the opposite mapping going from production data to rock properties; this being the inverse problem. Inverse problems are therefore often much more complicated to solve than their corresponding forward problems.

An inverse problem is often further complicated by the forward mapping not being bijective, which implies that different points in the parameter space can map to the same point in the data space. In our example that is expressed by different permeability fields resulting in the same production data. The solution to the inverse problem of computing the permeability field given the production data is thereby not unique. We can therefore not, with complete certainty, determine the permeability field of a reservoir that gives rise to a specific set of production data.

An inverse problem with a non-unique solution is denoted as an under-determined problem meaning that the data is not sufficient to uniquely determine the parameters. Solving an under-determined inverse problem means determining the set of points in the parameter space that all maps to the same point in data space given by observed data.

Uncertainty of the observed data can make the solution to a problem non-unique when points in parameter space no longer have to map to a single point in data space but to a set of points that all represent the observed data within the accuracy of your observations. The level of uncertainty of the observed data controls the size of the set.

Often in geosciences we have very scarce data and no (economically defensible) possibility of obtaining further data. Solving an inverse problem therefore consists of computing the full set of solutions that all satisfy the data within the required accuracy. The probabilistic approach to solving inverse problems that we have used throughout the study distinguishes be-

tween how likely each of the solutions are by assigning them a probability. For instance, in reservoir development this enables the decision maker to optimise the expected profit or risk of a reservoir strategy by considering the cost and profit for each of the computed scenarios combined with how likely they are.

1.1.4 Increasing Certainty by Ruling out the Impossible

Inverse problems arising in geosciences are often severely under-determined because of scarce data not restraining the solutions sufficiently. Also data can be ambiguous implying that the solutions satisfying the data will be very different. A common approach to reduce the size of the set of solutions is to introduce a priori knowledge. This means besides fitting the data a solution should, in order for us to consider it feasible, adhere to certain expectations that we have. These expectations are assumed independent of the observed data, for instance before we have knowledge of the data; hence the term a priori, or prior, knowledge

For instance, provided with a set of models of permeability fields all resulting in the same production data, a geologist will consider some of the permeability fields to be realistic because they look as he expects a permeability field to look. Others he might deem unrealistic because he knows, being the experienced geologist he is, that the permeability fields they describe are impossible to find in nature.

Figure 1.5 illustrates the subspaces of the parameter space and how the set of feasible solutions are the intersection between solutions that satisfy the data and solutions that fulfil prior expectations.

In geosciences, prior information for a physical parameter can be expectations of a certain spatial structure. Each model parameter is describing the value of a physical property at a specific location in space and perhaps even time, and we might have expectations of how low and high values of the property are distributed. Often we will expect some degree of continuity in

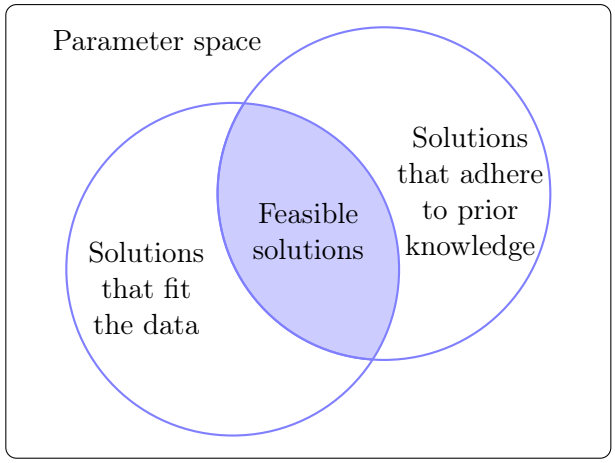


Figure 1.5: Parameter space of an inverse problem. Each solution is represented by a point in the parameter space. The set of solutions that satisfies data is a subset of the full parameter space. So is the set of solutions that satisfies the prior knowledge. The inverse problem consists of computing the intersection of these two subsets, i.e., the set of solutions that both fit the data and adhere to prior knowledge.

the physical property meaning, for example, that at a location surrounded by low values it is more likely to have yet another low value than a high value. We expect to see a correlation between values depending on their location, such that low values and high values are grouped together, respectively.

1.1.5 Geostatistical Prior Information

It is not possible to provide a unique, deterministic description of the model as no such description exists and instead the model is described probabilistically. This is done by considering the model as a random function, which is

purely a conceptual model that we choose because of the lack of deterministic models. The random process used to describe the model is traditionally assumed stationary, meaning the statistical properties of the model parameters are the same in all locations. The simplest random process is defined by its two-point statistics, which is the correlation between values of the parameter in two locations. These locations can be located anywhere in connection to each other; they can be close to or far away from each other in any direction. The two-point statistics is therefore used to describe the expectations previously formulated in words, that values in locations close to each other are highly correlated and values in locations far from each other are not correlated at all.

The two-point statistics of a model is defined by variogram models or covariance models. The latter is the covariance between the value at two locations as a function of the distance and direction between the two locations. Covariance models, and variograms, can have different shapes and types but they usually all have a range parameter. This determines the range of correlations, so values are only correlated if their locations are within the range of each other. How strongly they then correlate depends on the model type at hand. The range can vary depending on the direction.

Figure 1.6 shows an example of a random process defined by its two-point statistics. The longest correlation length is found in a direction 60° below horizontal. The range in this direction is two and a half times longer than the range in the perpendicular direction, which is the direction of shortest range. The direction and the ranges of the correlation causes the realisation to have oblong sloping areas of values of either low (blue) or high (red) values.

The variogram models in the direction of maximum and minimum range can be seen in Figure 1.7. These are of the Gaussian type. The theoretical variogram models defining the process from which the realisation is drawn are shown with solid lines, and the experimental variogram models derived from the realisation in Figure 1.6 are plotted against the theoretical models. The theoretical models defining the process are seen to fit the experimental

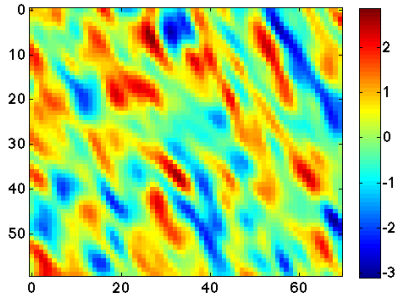
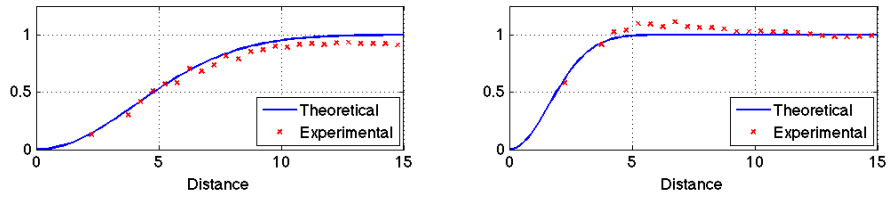


Figure 1.6: Example of a realisation of a random Gaussian process. The covariance model defining the process is anisotropic, and the range in the direction of maximum range is two and a half times the range in the direction of minimum range.



(a) Direction of maximum range **(b)** Direction of minimum range

Figure 1.7: Theoretical variogram models (solid lines) of the random process from which the realisation in Figure 1.6 is drawn. The experimental variogram models have been inferred (markers). The figure shows the models for the direction of maximum and minimum range.

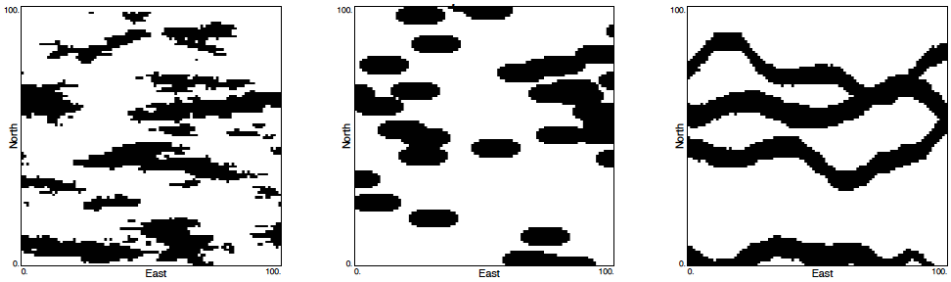


Figure 1.8: Three models showing very different geological structures (Strebelle, 2000). Such binary models are used to model the prior information to describe for instance low permeable (white background) and high permeable (black objects) zones of the subsurface.

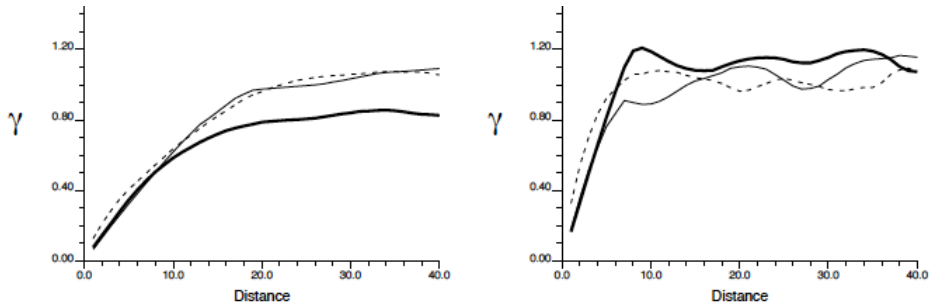


Figure 1.9: Variograms of the three models from Figure 1.8 (left plot: the East–West direction; right plot: the North–South direction). Dashed lines represent the model seen to the left, thin lines represent the model in the middle, and the thick lines represent the model to the right (Strebelle, 2000).

models derived from the realisation. It is also seen that the ratio between the ranges in the two perpendicular directions is two and a half, i.e., the variogram models have the same value at the distance 10 and 4, respectively.

However, two-point statistics has clear limitations. Figure 1.8 shows three examples of very different geological structures — some more geologically

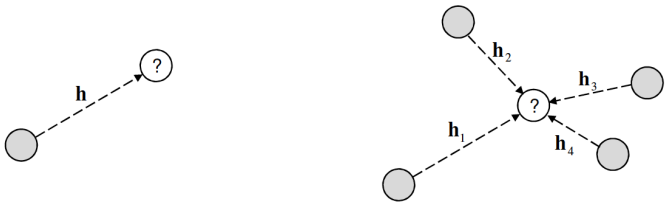


Figure 1.10: The concept of two-point statistics versus the concept of multiple-point statistics (Honarkhah, 2011).

reasonable than others. These models illustrate the core of the problem with two-point statistics. While the human eye has absolutely no difficulties discriminating between the three models, their two-point variograms are surprisingly similar; these can be seen in Figure 1.9. So based on only the two-point variograms it is not possible to distinguish between the three models.

We therefore need something beyond two-point statistics in order to successfully capture the geological structures of the models. Multiple-point statistics does exactly this. It characterises models by looking at constellations of many, or multiple, points.

Figure 1.10 illustrates how multiple-point statistics is a natural extension of two-point statistics. To the left in the figure the unknown value of the white point (marked by a ?) is determined based on its correlation to the value of the known grey point. The distance between the two points are given by the distance vector $\mathbf{h}$. The correlation between the values at the two points is a function of the direction and length of the distance vector.

To the right in Figure 1.10 is illustrated the concept of multiple-point statistics. Here the unknown value at the point marked by ? is determined based on multiple other points surrounding it. The distance to these points are defined by the distance vectors $\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3$ and $\mathbf{h}_4$. This constellation of multiple points allows for non-linear structures to be defined.

Although two-point variograms can be inferred from data, as for example seen in Figures 1.7 and 1.9, it does require a certain amount of data to account for different distances and directions. In reality, when we only have scarce data inferring a mere two-point variogram based on pairs of data can become difficult. Quantifying the multiple-point statistics of the subsurface from a scarce data set is an impossible task. Instead the concept of training images are introduced.

A training image is a conceptual image and can be interpreted as a realisation of an unknown process. It is assumed to describe structures or patterns of a spatial model. Using a priori information in terms of multiple-point statistics learned from a training image translates to having specific expectations as to what structures and patterns in a model are expected and thereby exclude models with unrealistic structures or patterns, as these are deemed impossible to appear in nature.

This study is concerned with using geostatistical a priori information when solving an inverse problem of modelling the subsurface of an oil reservoir. The work has focused on two different problem instances. One is based on two-point statistics and (as the literature review in the next section will reveal) a well-establish interpolation method called kriging. The second instance is incorporating complex prior information in the form of multiple-point statistics. It has involved developing a method called the Frequency Matching method. But before going into more details of the study and to better understand the background motivation for the work we provide a brief overview of the development of geostatistics.

1.2 Literature Review on Geostatistics

Geostatistics goes back to the 1960s where it was first developed based on a need for the methodology to evaluate the recoverable reserves in mining deposits. It is in general concerned with the analysis of data distributed in space and/or time. It has since then been successful in a wide variety

of fields starting from mining but also including petroleum, soil science, oceanography, hydrogeology and environmental sciences.

1.2.1 Two-point Statistics

Traditionally geostatistics was used only as a tool for describing linear spatial patterns and for interpolation of values at unsampled locations, see for example David (1977); Vieira et al (1983); Trangmar et al (1986); Warrick et al (1986) or the books by Journel and Huijbregts (1978) and Isaaks and Srivastava (1989).

One of the most dominant interpolation techniques through time is the kriging technique, which is a set of methods to interpolate the values of a random field. One of the earliest mentions of the interpolation technique now known as kriging was by Krige (1951). Cressie (1990) provides an overview of the development of kriging in its early years.

Kriging is a deterministic approach that generates an interpolated map of a physical parameter. The kriging estimator is per definition the best linear unbiased estimator and has become a powerful tool in many fields not just geostatistics. Multiple variants of kriging were introduced differentiated by their assumptions of the underlying model of the parameter to be interpolated (Journel and Huijbregts, 1978; Goovaerts, 1997). The main disadvantage of kriging techniques is that they tend to provide a spatially smooth model of the parameter, whereas in reality the physical property described is rarely spatially smooth.

This disadvantage is not shared with stochastic simulation, which was first introduced by Matheron (1973) and Journel (1974) and became popular in the literature in the 1980s (Borgman et al, 1984; Alabert, 1987b,a; Mantoglou, 1987; Davis, 1987). Focus in geostatistics then shifted to describing uncertainty by an ensemble of realisations, which are multiple models describing the same physical property. The models are generated by stochastic methods to honour available data and to reproduce certain statistical properties. Geostatistics now included tools for description of spatial patterns,

quantitative modelling of spatial continuity, spatial prediction and uncertainty assessment. See for instance Goovaerts (1997), which by many is considered a must-read in geostatistical literature.

Deutsch and Journel (1998) states the two most significant differences between kriging and stochastic simulation:

1. Kriging provides locally accurate estimations of the variables, where as stochastic simulation is globally oriented and considers the entire set of estimated variables while seeking to produce estimates that correspond to a certain spatial structure.

2. Both kriging and stochastic simulation provide a local uncertainty measure, however, the ensemble of models generated by stochastic simulation also expresses a joint uncertainty measure of events involving multiple locations.

Depending on the problem instance at hand one solution approach might be favourable to the other.

1.2.2 Multiple-point Statistics

Multiple-point statistics (MPS) became increasingly popular as the available computer resources increased dramatically during the last decade and is today a hot topic in geostatistics. Methods involving multiple-points statistics inferred from conceptual training images can generally be divided into three categories: probabilistic-based, optimisation-based, and pattern-based approaches. Here is provided a short list of some of the most noticeable techniques within each category. For a more thorough description of the techniques and respective advantages and disadvantages the reader is referred to the literature.

1.2.2.1 Probabilistic-based Approaches

Multiple-point statistics was first introduced by Guardiano and Srivastava (1993). The concepts of MPS can be viewed as a generalisation of two-point statistics, however, its computational complexity and high CPU demands made it impractical for the computers at that time. It was not until Strebelle (2002) presented the SNESIM algorithm that MPS methods became computationally feasible.

Another probabilistic-based approach is the indicator technique proposed by Ortiz and Deutsch (2004) and later generalised by Ortiz and Emery (2005). Hong et al (2008) expanded it to also include secondary data.

1.2.2.2 Optimisation-based Approaches

Using simulated annealing in a spatial context was first proposed by Deutsch and Lewis (1992). It was used to minimise an objective function expressing the misfit between the a priori assumed multiple-point statistics and the actual multiple-point statistics of a model. It has been used for both categorical (Goovaerts, 1996) and continuous (Fang and Wang, 1997) variables. More recently Peredo and Ortiz (2010) applied a parallellised simulated annealing scheme gaining the expected computational speed-up but unable to reproduce the spatial structures of the training image in the models. The Frequency Matching (FM) method (Lange et al, 2012) discussed in this thesis is also based on simulated annealing.

Caers and Journel (1998) proposed to use neural networks, well-known from the field of machine learning, for multiple-point geostatistical modelling and applied it for unconditional simulation. Later Caers and Ma (2002) expanded the approach to condition the simulations to seismic data.

Besag (1974) made use of Markov random fields to model spatial continuity in the subsurface followed by Tjelmeland (1996); Daly (2005) and Tjelmeland and Eidsvik (2005) who used approaches based on Markov random fields. A subclass of the Markov random fields are the Markov-Mesh

models used by for instance Daly and Knudby (2007) and Parra and Ortiz (2009).

Finally there is the Gibbs sampler originally proposed by Geman and Geman (1984) and later applied to multiple-point simulation of geological phenomena by Lyster and Deutsch (2008).

1.2.2.3 Pattern-based Approaches

The idea behind pattern-based approaches is to minimise the computational cost of probabilistic simulation methods by simulating multiple voxel values simultaneously. Patterns are defined as sub-blocks of an model of identical geometric shape, and similar to the SNESIM algorithm the SIMPAT algorithm (Arpat, 2005; Arpat and Caers, 2007) creates realisations by iteratively simulating sub-blocks of the model. The main disadvantage of the SIMPAT algorithm is the computationally expensive comparison between partially simulated sub-blocks and the patterns of the training image.

Filtersim was proposed as an alternative to SIMPAT (Zhang, 2006; Wu, 2007) and it uses general linear filters to classify the patterns of the training image. This simplifies the comparison of patterns.

Mariethoz and Renard (2010) proposed the direct sampling method which is similar to the SIMPAT algorithm. Direct sampling does not use patterns with predefined shapes but instead it conditions upon the set of, at the time, informed voxels. Mariethoz et al (2010) demonstrated how this implies that the nodes to condition upon can be reached with an increasingly smaller spatial radius as the simulation proceeds.

Recently, Honarkhah (2011) developed the DISTPAT algorithm which uses multi-dimensional scaling to classify the patterns, such that the comparison can be done using only a small set of prototypes representing the entire set of patterns in the training image.

1.3 Objectives and Main Contributions

The primary objective of the work presented in this thesis has evolved around solving the often severely under-determined inverse problem of describing the rock properties of the subsurface of an oil reservoir. We have strived for the overall goal of simultaneously incorporating different kinds of data and statistical a priori information. There exist methods that can handle prior information along with different types of data, but separately. By using such methods we ignore that parameters computed from different data types may, in fact, be physically interrelated. Currently we lack a method that can deal with all available data regardless of its type. The work in this thesis is merely building blocks for a first step in that direction.

The main contributions of the study are:

- The description of a method for efficient prediction of rock properties based on seismic attributes. Kriging interpolation of a rock property is done in a space spanned by the seismic attributes instead of the traditional approach of basing the interpolation on spatial coordinates. An orthogonal transformation of the seismic attributes followed by a reduction of dimensions of the interpolation space reduces the computational complexity of the method without significant loss of information.

- Formulating a closed form expression for an a priori probability density function of a model given multiple-point statistics learned from a training image. No other existing methods have a closed form expression but instead they are based on black box routines that can sample the a priori probability density function but not compute the relative values of the a priori probability density function of sampled models.

- The formulation of the Frequency Matching method which enables us to compute the maximum a posteriori solution to an inverse problem using multiple-point statistics as prior information.

- A general Fortran implementation of the Frequency Matching method that can be used to solve linear inverse problems. The implementation is not restricted to a particular application in geosciences as any training image can be provided and so can the parameters of the linear forward problem.

It should be noted that the objective of the study has been proof of concept rather than solving large-scale, real life problems. Unfortunately, the computational resources of today are not quite there yet. We have focused on the methods themselves and studied their advantages and disadvantages for manageable problems bearing in mind their future potential. Especially the work on multiple-point statistics involves very simplified training images and idealised models. This is a general trend in the literature on multiple-point statistics. We emphasise that this study is only a first step on the onward journey to incorporate complex geostatistical prior information to its fullest when solving authentic industry problems.

1.4 Outline

The work is documented by the thesis consisting of a summary report of five chapters and seven appendices. Chapter 1 gives an introduction to the project. An introduction to probabilistic inverse problem theory is provided in Chapter 2. Chapter 3 is concerned with interpolation in seismic attribute space, which is based on two-point statistics. Chapter 4 presents the Frequency Matching method, which is a recently proposed method based on multiple-point statistics. Finally, Chapter 5 concludes the study.

Appendix A through G holds journal and conference papers documenting the research; the papers are listed chronologically. The material presented in the papers and in the thesis must be expected to overlap to some extent. However, since both contains details that are not present in the other they are considered complementary. The papers are concerned with one of the two topics of two-point statistics and multiple-point statistics.

1.4.1 Papers on Two-point Statistics

These papers are supported by the work described in Chaper 3. The primary paper is:

> *Efficient Prediction of Rock Properties from Seismic Attributes using Orthogonal Transformations* (Appendix G)

The second paper is a conference paper with preliminary work:

> *Kriging in High Dimensional Attribute Space using Principal Component Analysis* (Appendix A)

1.4.2 Papers on Multiple-point Statistics

Chaper 4 is concerned with multiple-point statistics and the Frequency Matching method. The FM method was proposed in the paper:

> *A Frequency Matching Method: Solving Inverse Problems by use of Geologically Realistic Prior Information* (Appendix C)

The very first mentioning of the Frequency Matching method in literature was in a conference paper with preliminary work:

> *A Frequency Matching Method for Generation of a Priori Sample Models from Training Images* (Appendix B)

A recently published report elaborates on the implementation of the Frequency Matching method:

> *An Implementation of the Frequency Matching Method* (Appendix F)

Two conference papers related to applications of the Frequency Matching method are also included:

> *A Novel Approach for Combining Multiple-point Statistics and Production Data in Reservoir Characterization* (Appendix D)
>
> *Improving Multiple-Point-Based a Priori Models for Inverse Problems by Combining Sequential Simulation with the Frequency Matching Method* (Appendix E)

CHAPTER 2

Inverse Problems with A Priori Information

This chapter gives a brief introduction to probabilistic inverse problem theory. We provide a mathematical formulation of the concept of inversion as illustrated in Figure 1.4. Inverse problems arise in a wide variety of fields, where they are often encountered in connection to computation of information about internal or otherwise hidden and inaccessible data. They appear frequently in geosciences where we often wish to describe properties of the subsurface based on measurements made on the surface using knowledge of physical laws; for instance, describing how waves travel in different materials or how fluids flow in porous media.

2.1 Introduction

Consider the mathematical problem of computing a set of n data observations $\mathbf{d} \in \mathbb{R}^n$ based on a model of m parameters $\mathbf{m} \in \mathbb{R}^m$ where the

mapping operator $g : \mathbb{R}^m \to \mathbb{R}^n$ is a known function;

$$g(\mathbf{m}) \quad = \quad \mathbf{d}. \tag{2.1}$$

This is a typical forward problem as it computes the unknown output (the data) given a known input (the model) and a known system (the mapping operator). The corresponding inverse problem is the computation of the unknown model parameters given a set of noisy data observations $\mathbf{d}^{\text{obs}}$. In most cases, the inverse operator $g^{-1} : \mathbb{R}^n \to \mathbb{R}^m$ is non-existing or unknown and the problem is solved only based on knowledge of g.

In some forward problems the mapping operator itself is unknown. The inverse problem then consists of estimating the mapping operator given the observed data for a known set of model parameters. The mapping operator can then be used to predict future data observations given new sets of model parameters.

Examples of common inverse problems in reservoir modelling are seismic tomography and history matching of production data.

2.2 The Probabilistic Approach

Tarantola and Valette (1982b) presents a probabilistic approach for solving inverse problems. The purpose of their approach was to fill an, at that time, scientific void and create a method that can combine information from independent sources (Mosegaard, 2011). Those can be different kinds of data as well as past experience and other considerations a scientist might have.

The approach is based on the introduction of probability density functions. These are assumed to be complete descriptions of the state of information available on the parameters. Tarantola and Valette then found that multiple pieces of independent information can be combined by multiplication to form a new probability density function (pdf).

The solution to an inverse problem is defined by a pdf that to each model in the model space assign a relative probability[1]. This pdf is denoted the a posteriori pdf. The derivation of the solution concludes it to be characterised by the two elements:

$\rho_{\mathbf{m}}(\mathbf{m})$**:** the a priori pdf describing expectations on the model $\mathbf{m}$ considering no new data; only experiences possibly based on previously treated data is considered.

$L(\mathbf{m})$**:** a pdf describing how well a model is explained by data, i.e., how likely a model is. This is referred to as the likelihood[2] function and it arises indirectly from the relation $\mathbf{d} \approx g(\mathbf{m})$.

This pdf can be constructed based on the two elements just mentioned: the a priori pdf of the model parameters and the likelihood function. The a posteriori pdf is given by:

$$\sigma_{\mathbf{m}}(\mathbf{m}) \quad = \quad const\ \rho_{\mathbf{m}}(\mathbf{m})\ L(\mathbf{m}). \tag{2.2}$$

Often the normalization constant is not computed as it is sufficient to know the value of the a posteriori pdf of a model relative to the value of the posteriori pdf of another model.

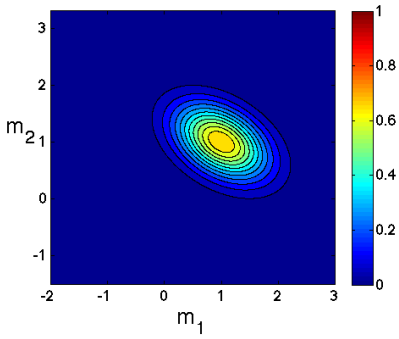
A general expression of the likelihood function is given by Tarantola (2005). Applying the assumption that the forward mapping has no modelling error simplifies this significantly. Furthermore, a popular assumption of Gaussian noise on the observed data implies that the inverse problem from Equation (2.1) gives rise to the following Gaussian likelihood function:

$$L(\mathbf{m}) = const\ \exp\left(-\frac{1}{2}\left(\mathbf{d}^{\text{obs}} - g(\mathbf{m})\right)^T \mathbf{C}_{\mathbf{d}}^{-1}\left(\mathbf{d}^{\text{obs}} - g(\mathbf{m})\right)\right), \tag{2.3}$$

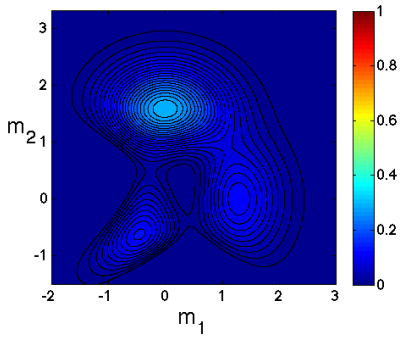
where $\mathbf{C}_{\mathbf{d}}$ is the covariance matrix for the noise on the observed data $\mathbf{d}^{\text{obs}}$.

[1] Not to be confused with the probability of a model, as per definition, all models in a continuous model space have the probability 0.

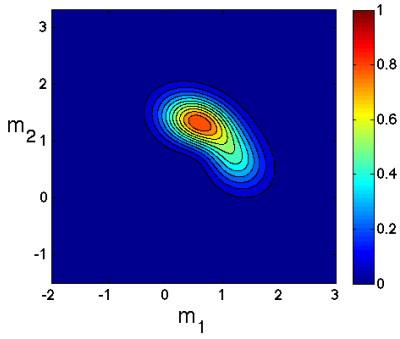
[2] The likelihood function should not be confused with the term "likely" as it was just used. In statistics, the latter is used to denote the value of an arbitrary pdf evaluated of a model in a continuous model space.



(a) A priori pdf $\rho_{\mathbf{m}}(\mathbf{m})$



(b) Likelihood function $L(\mathbf{m})$



(c) The resulting a posteriori pdf $\sigma_{\mathbf{m}}(\mathbf{m})$

Figure 2.1: Probability density functions for a non-linear inverse problem with two model parameters. The a posteriori pdf is computed as the (scaled) product of the a priori pdf and the likelihood function.

2.2.1 Example

Figure 2.1 shows the probability density functions for a two-dimensional non-linear inverse problem. The problem consists of estimating two model parameters m_1 and m_2 and it is synthetic problem with no real-life application, chosen solely for illustration purposes.

An example of an a priori pdf often chosen for its mathematical convenience and simplicity is the Gaussian a priori pdf with mean $\mathbf{m}_0$ and covariance matrix $\mathbf{C}_\mathbf{m}$:

$$\rho_\mathbf{m}(\mathbf{m}) \quad = \quad const \ \exp\left(-\frac{1}{2}\left(\mathbf{m}-\mathbf{m}_0\right)^T \mathbf{C}_\mathbf{m}^{-1}\left(\mathbf{m}-\mathbf{m}_0\right)\right). \qquad (2.4)$$

We assume an a priori pdf of this type with parameters:

$$\mathbf{m}_0 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad \mathbf{C}_\mathbf{m} = \begin{pmatrix} 0.3 & -0.1 \\ -0.1 & 0.2 \end{pmatrix}.$$

This function is shown in Figure 2.1a. Figure 2.1b shows a likelihood function, which, in this case, is just an arbitrary function but as in most real cases it has several local maxima in separate areas of the model space with probability density values significantly greater than zero; the latter is seen by the level curves. Solving this inverse problem without use of prior information would result in three different kinds of solutions, one for each area of the likelihood function with (relative) high probability.

The a posteriori pdf computed using Equation (2.2) is seen in Figure 2.1c. We notice how introducing prior information rules out models of two out of the three possible kinds and leaves us with an a posteriori pdf with only one area of non-zero values. The assumption of prior information expected values of the model parameters as $(1, 1)$ therefore ruled out two of the three kinds of models as being unrealistic. The third kind is a compromise of the a priori expected values and the area with high values of the likelihood function. In this example the remaining type of models is coincident with the area of the likelihood function that has the highest values but that is not always the case.

2.3 Analytical Solution

Under certain circumstances an analytical expression for the a posteriori pdf, $\sigma_{\mathbf{m}}(\mathbf{m})$, can be derived. Typically, in such cases the forward operator is a linear function, i.e., Equation (2.1) can be written as:

$$\mathbf{d} \quad = \quad \mathbf{G}\mathbf{m},$$

where $\mathbf{G}$ is an m by n matrix. For a simple choice of a Gaussian a priori pdf as in Equation (2.4), the a posteriori pdf is then a product of two Gaussian distributions and therefore also Gaussian distributed (Tarantola and Valette, 1982a):

$$\sigma_{\mathbf{m}}(\mathbf{m}) \quad = \quad const \ \exp\left(-\frac{1}{2}\left(\mathbf{m}-\widetilde{\mathbf{m}}\right)^T \mathbf{C}_{\widetilde{\mathbf{m}}}^{-1}\left(\mathbf{m}-\widetilde{\mathbf{m}}\right)\right),$$

with mean and covariance parameters:

$$\begin{aligned}
\widetilde{\mathbf{m}} &= \mathbf{m}_0 + \mathbf{C}_{\mathbf{m}}\mathbf{G}^T\left(\mathbf{G}\mathbf{C}_{\mathbf{m}}\mathbf{G}^T + \mathbf{C}_{\mathbf{d}}\right)^{-1}\left(\mathbf{d}^{\text{obs}} - \mathbf{G}\mathbf{m}_0\right), \\
\mathbf{C}_{\widetilde{\mathbf{m}}} &= \mathbf{C}_{\mathbf{m}} - \mathbf{C}_{\mathbf{m}}\mathbf{G}^T\left(\mathbf{G}\mathbf{C}_{\mathbf{m}}\mathbf{G}^T + \mathbf{C}_{\mathbf{d}}\right)^{-1}\mathbf{G}\mathbf{C}_{\mathbf{m}}.
\end{aligned}$$

These circumstances are unfortunately rarely present, as many of the physical laws modelling the processes of the Earth are highly non-linear. This is also the case for both seismic tomography and history matching. Although, using a simplified wave propagation model some seismic tomography problems can be made linear.

Problems that are weakly non-linear can also be solved analytically by linearising the forward operator $g(\mathbf{m})$ around the a priori expected model $\mathbf{m}_0$ (Tarantola, 2005).

2.4 Sampling-based Solution Methods

In the cases where an analytical expression for either one or both of $\rho_{\mathbf{m}}(\mathbf{m})$ and $L(\mathbf{m})$ is not available and hence no analytical expression for $\sigma_{\mathbf{m}}(\mathbf{m})$ exists one possibility is to use sampling methods. These methods solve inverse

problems by generating samples of models approximating the a posteriori pdf.

Monte Carlo (MC) methods are powerful tools when it comes to solving inverse problems (Mosegaard and Sambridge, 2002). They come in two categories namely sampling methods and optimisation methods; for now we will concern ourselves only with the sampling methods.

The overall idea of a Monte Carlo algorithm is to first generate pseudo-random, uniformly distributed numbers. These numbers are then transformed into a pseudo-random sample from a pdf. If repeated numerous times the samples approximate a (possibly very complex) function. Over time a countless number of MC methods have been described in the literature. Most are considered hybrids and are based on combinations of previously described algorithms.

In the following we will briefly describe five basic approaches to sampling.

2.4.1 The Metropolis Algorithm

One of the earliest and most widely used Monte Carlo method is the Metropolis algorithm. The name Monte Carlo methods was first mentioned in the literature by Metropolis and Ulam (1949). Later, Metropolis et al (1953) published the application of a Markov chain-based Monte Carlo (MCMC) algorithm for sampling of Gibbs-Boltzmann distributions in high-dimensional spaces. This algorithm is now known as the Metropolis algorithm or the Metropolis-Hastings algorithm.

An important feature of the Metropolis algorithm is that it does not require knowledge of the function f to be sampled. Instead it needs only to know the ratio $\frac{f(x_j)}{f(x_i)}$ between values of the function for different samples; the current model x_i and a proposed model x_j. The ratio then determines the acceptance probability of model x_j. This means the algorithm only need to be able to compute the value of a function, which is proportional to f.

Often, pdfs have a normalisation constant that is difficult if not impossible to determine. This is not needed in order to apply the Metropolis algorithm.

2.4.2 Rejection Sampling

von Neumann (1951) proposed the method rejection sampling. Like the Metropolis algorithm, rejection sampling only needs to know a function, $const f$, that is proportional to f, which is the function to sample. Furthermore, rejection sampling needs to know an upper limit f_{max} of the function proportional to f.

A proposed model x_j is accepted with probability $\frac{const\ f(x_j)}{f_{max}}$. This reveals a significant drawback of rejection sampling namely that one can expect a large number of the proposed models to be rejected. This happens of course when a poor upper bound f_{max} is provided but even if that is not the case one can expect many models having a low value of the ratio $\frac{const\ f(x_j)}{f_{max}}$ and hence large risk of rejection.

An adaptive version of rejection sampling was first proposed by Gilks and Wild (1992). It can be used instead of rejection sampling to sample a log-concave density function. The adaptive part of the algorithm adapts the limits of the function to be sampled making them converge to the function itself, which reduces the risk of rejecting subsequent points. Fewer rejection steps implies fewer evaluations of the often computationally expensive log-density function.

2.4.3 The Gibbs Sampler

Another well-known MC method is the Gibbs sampler first described by Geman and Geman (1984). In the Gibbs sampler each iteration consists of multiple sub-steps, where in each one no more than one parameter is perturbed. This means the Gibbs sampler is suited when the conditional

probabilities of each of the parameters conditioned on the remaining parameters are easily accessible.

However, a disadvantage of the Gibbs sampler is that the proposal distribution requires computations of values of the function f to be sampled, and hence the Gibbs sampler is not applicable in cases where f is expensive to evaluate.

2.4.4 The Neighbourhood Algorithm

Sambridge (1999) proposed another MC based algorithm that he referred to as a neighbourhood algorithm. The objective of this algorithm is to sample the region of a parameter space that contains models of acceptable function values. The algorithm is based on an idea of utilising all previous models of which the function has been computed while searching the parameter space. The algorithm made use of Voronoi cells to interpolate the function that is assumed constant within each cell.

2.4.5 Stochastic Simulation

Stochastic simulation is an approach to quantify spatial uncertainty. A general introduction to the paradigm of stochastic simulation and a description of several simulation algorithms is given by Goovaerts (1997). Stochastic simulation can be perceived as the opposite of stochastic estimation. Whereas stochastic estimation seeks to provide a model consisting of local, best estimates, stochastic simulation provides an ensemble of models also referred to as realisations. The realisations sample the a posteriori pdf and at the same time their variation describes the uncertainty of the model parameters.

A realisation is generated by use of a random walk between the model parameters. Each model parameter is simulated conditional to known data

as well as previously simulated model parameters. A realisation is said to be conditional to known data, and a priori statistics of the model parameters is used to determine the conditional probability distributions of which the model parameters are drawn from.

Evaluating the ensemble of models allows for determining the risk of certain events that could be of interest. This feature is the exact reason that many consider simulation favourable to estimation. Literature shows numerous examples of the application of stochastic simulation in geosciences.

The main algorithms of stochastic simulation described by Goovaerts (1997) were made available in the public domain geostatistical software library GSLIB (Deutsch and Journel, 1998).

The first sequential simulation algorithm incorporating multiple-point statistics learned form a training image as prior information was proposed by Guardiano and Srivastava (1993). However, it was not until Strebelle (2002) developed the SNESIM algorithm that simulation conditioning on multiple-point statistics as prior information became computationally feasible. The conditional probabilities used in the simulation were obtained from the training image. Since then a variety of simulation methods incorporating multiple-point statistics as prior information has been proposed, some even simulating multiple model parameters at a time (Arpat, 2005; Wu et al, 2008; Honarkhah, 2011).

2.4.6 Application of a Sampling Method

Figure 2.2 shows the a posteriori pdf from the example in Section 2.2.1. We have used a Metropolis algorithm to generate 1000 samples from the a posteriori pdf. Each sampled model is marked by a dot. The figure illustrates how the intensity of the samples follows the value of the pdf sampled; the higher probability density value of an area the more densely it has been sampled. The size of the dots reflect how many times the model has been sampled.

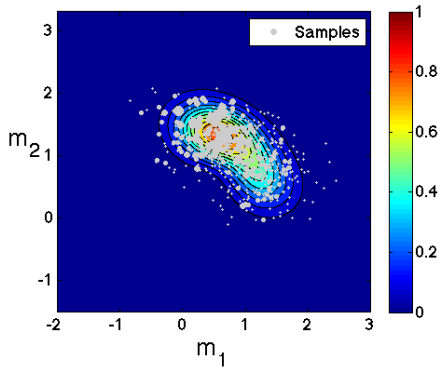


Figure 2.2: The a posteriori probability density function from Figure 2.1c with 1000 sample models marked. Each sampled model is marked by a grey dot and the size of the dot reflects how many times the model was sampled.

One disadvantage of the sampling techniques is that despite the finite number of samples, any model has per definition probability 0 and they cannot evaluate the value of the a posteriori pdf of a given model. Also for large-scale problems computing a sufficiently large set of realisations can be a costly affair. In such cases it might be useful to know only one model that, in some sense, is considered the most representative. This can be done by means of optimisation.

2.5 Optimisation-based Solution Method

Sampling the a posteriori pdf will, as just discussed, result in a set of models where each model is represented according to its a posteriori probability density function value. Some sampling techniques, for example the Gibbs sampler, do not evaluate the a priori probability density value of a model. This is a strength of the sampling methods that they are able to sample the a priori pdf without evaluating it of a given model. However, this also

means they can only sample the a posteriori pdf and cannot compute the value of it for a model.

In some applications it might be useful to know the probability density value of a given model. When a closed form expression of the a priori pdf is available this is possible. However, the a priori pdf can be expensive to evaluate and in some cases we do not have an efficient way to sample it. Sampling the a posteriori pdf will therefore become computationally infeasible. In such cases it might be satisfying to consider only one model to represent the set of solutions to the inverse problem. Often one specific model is considered to be of interest, namely the most likely model, i.e., the model maximising the a posteriori pdf:

$$\mathbf{m}^{\mathrm{MAP}} = \underset{\mathbf{m}}{\operatorname{argmax}} \left\{ \sigma_{\mathbf{m}}(\mathbf{m}) \right\}. \tag{2.5}$$

This model is referred to as the maximum a posteriori (MAP) model and is widely used in regularisation techniques.

In the example from Section 2.2.1 the MAP model is computed as:

$$\mathbf{m}^{\mathrm{MAP}} \simeq \begin{pmatrix} 0.600 & 1.313 \end{pmatrix}^T,$$

as shown in Figure 2.3. The MAP model is plotted against the a posteriori pdf; it is recognised as the global maximiser.

2.6 Summary

In this chapter we have introduced basic probabilistic inverse problem theory. We have discussed different types of solution methods, and when they are applicable to an inverse problem at hand. We have seen how the type of prior information available limits our choice of solution method. In the simplest case of an Gaussian a priori pdf to a linear problem an analytical expression for the a posteriori pdf can be determined.

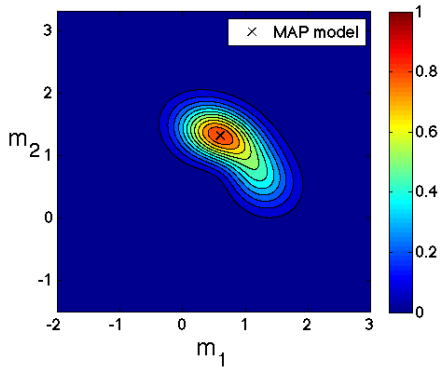


Figure 2.3: A posteriori probability density function with the MAP model marked.

For non-linear problems the choice of solution method depends on whether we have an efficient method to sample the a priori pdf, in which case sampling methods can be applied. One limitation of existing sampling methods is that they cannot compute the probability of a given model, and they therefore cannot compute the maximum a posteriori model.

When a closed form expression of the a priori pdf is available we can optimise the a posteriori pdf directly and thereby compute the MAP model.

CHAPTER 3

Kriging Interpolation in Attribute Space

This chapter is concerned with an application of two-point statistics within the geosiences. The Ph.D. study has included the development of a method for efficient prediction of rock properties from seismic data. This is an interpolation problem solved by use of kriging methods, as mentioned in Chapter 1. That means that interpolation of a rock property is guided by its two-point statistics.

This chapter supports the work described in the paper seen in Appendix G. The conference paper in Appendix A holds preliminary work and results. The chapter provides an introduction to the inverse problem. It then gives an outline of the method followed by a discussion of the key steps of the algorithm. Last, other aspects of the method will be discussed.

3.1 Introduction

Interpolation of, for instance, porosity or permeability between well logs is a well-known problem in seismic exploration. Seismic attributes describing elastic properties of the reservoir are extracted from seismic data. These are then, to different extents, used to guide the interpolation of rock properties.

The interpolation problem is an inverse problem as discussed in the beginning of Chapter 2. It is of the second type of inverse problem, where the mapping operator g is unknown. One approach, opposite to ours, is to attempt to formulate a presumably highly complex, non-linear mapping from the seismic data to the rock property. Such mapping would be based on knowledge from the fields of geophysics, geology etc. It would be a cumbersome, if at all possible, task to formulate such an exact mapping. So instead of approaching the problem from a geological point of view, we take a geostatistical and data analytical point of view. We simply assume that the seismic data characterises the subsurface of the reservoir sufficiently to statistically predict its rock properties.

In the literature, interpolation of rock properties has been done using linear regression (Hampson et al, 2001; Russell et al, 2002; Hansen et al, 2008), spline interpolation, nearest neighbour interpolation, collocated cokriging (Doyen, 1988) and neural networks, (Hampson et al, 2001; Russell et al, 2002; Pramanik et al, 2004; Herrara et al, 2006). Traditionally, interpolation is done in physical space spanned by spatial coordinates. This is based on a assumption that values of a rock property at points located closely together in physical space would be highly correlated, whereas values of a rock property at points located further apart from each other would be less correlated. As valid an assumption this may seem it clearly does not account for sudden changes in rock quality, e.g., across faults.

Seismic attributes have been included, depending on the choice of interpolation method, as data correlating to the rock property. Cokriging, for instance, would in theory be able to fully include the seismic data in the interpolation. However, the problem becomes computationally infeasible

due to inference of all the cross-covariance models needed.

Our approach should be seen as an alternative to the classical interpolation in physical space. It is based on the assumption that the correlation between rock property values at two different locations is proportional to two terms: 1) The physical distance between the two locations, as traditional interpolation. 2) The similarity in the seismic attributes at the two locations. The second term is motivated by the assumption that the seismic data characterises the subsurface geology of the reservoir and thereby its rock properties. The second term gives rise to introduction of seismic attributes as coordinates in a high-dimensional space, and to the definition of a distance in this seismic attribute space. The interpolation is then performed based on the combined distance in physical space and the seismic attribute space.

We will refer to the space spanned by combined spatial coordinates and seismic attributes as the attribute space. Kriging in this space requires the inference of only one covariance model. The attribute space will be high-dimensional, which complicates the inference of a covariance model, with the exact dimensionality depending on the number of seismic attributes available. However, some of the attributes, such as depth and two-way travel time, can be expected to be correlated.

From multivariate data analysis theory we know that a data set consisting of highly correlated variables can be well approximated using a lower-dimensional subspace. We therefore apply a transform to the seismic attributes mapping them into a lower-dimensional attribute space. We refer to this subspace as the transformed and reduced attribute space.

Reducing the dimension of the space in which the interpolation is performed, decreases the complexity of the interpolation approach as it simplifies the inference of a covariance model. Before interpolation, we therefore create a space spanned by transformed attributes. The dimension of this is chosen based on how well the original data is approximated.

Having presented the background and motivation this leads us to the for-

mulation of our method itself.

3.2 Outline of the Method

The interpolation approach can be summarised in six steps. These are presented here and the remainder of the chapter will be used to elaborate on them. The section number in brackets refer to the individual sections discussing each step.

(i) Initialisation: Normalisation of the seismic attributes and normal score transformation of the well log data (Section 3.3.1, 3.3.2 and 3.4.1).

(ii) Transformation of the seismic attributes (Section 3.4.2).

(iii) Reduction of the dimension of the transformed attribute space (Section 3.4.3).

(iv) Inference of a covariance model in the transformed and reduced attribute space by use of maximum likelihood estimation (Section 3.4.4).

(v) Kriging interpolation of the rock property in the transformed and reduced attribute space (Section 3.3.3).

(vi) Inverse normal score transformation of the kriged values back to physical units and evaluation of the results.

3.3 The Key Steps of the Algorithm

The main elements of the method presented in Section 3.2 is the data and the choice of interpolation method. Before discussing the finer details of the method we introduce the form of data used. We then present a quick overview of the theory of kriging interpolation, which is the interpolation technique at the base of our method.

3.3.1 Well Log Data

The interpolation problem consists of interpolating the values of a rock property in the subsurface of a reservoir. A set of n measurements of the rock property is known. These are available from well log data taken from existing wells in the reservoir. Let $z_i \in \mathbb{R}$ for $= 1, \ldots, n$ denote the value of the rock property at the ith location.

We define the vector of known rock property values:

$$\mathbf{z} = \begin{pmatrix} z_1 & z_2 & \cdots & z_n \end{pmatrix}^T \in \mathbb{R}^n,$$

and z_0 as the unknown rock property value to be estimated at a given location in the reservoir.

3.3.2 Seismic Attribute Data

A seismic survey spanning the reservoir is assumed available. From this survey seismic data is extracted. The seismic data covers the entire reservoir that is spanned by a regular grid. Values of the seismic attributes in each of the grid points are generated from the seismic data. Let m denote the number of seismic attributes available. Then $\mathbf{u}_i \in \mathbb{R}^m$ denotes the vector of seismic attributes in the location of the ith well log measurement and we will refer to this vector as the attribute vector of the ith location. Some interpolation of the seismic attribute data might be necessary since well log locations are not necessarily coinciding with grid points.

We define the attribute matrix $\mathbf{U}$ as the matrix of all n attribute vectors:

$$\mathbf{U} = \begin{pmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_n \end{pmatrix}^T \in \mathbb{R}^{n \times m}.$$

The ith row of the attribute matrix holds the transpose of the attribute vector $\mathbf{u}_i$ and we will refer to the jth column of the attribute matrix, i.e., the vector of measurements of the jth seismic attribute for all n locations, as $\mathbf{U}_{:,j}$.

The units of the different seismic attributes are not directly compatible and we therefore assume that the attribute data has been normalised, i.e., it has mean value zero and variance one:

$$\mathrm{E}\{\mathbf{U}_{:,j}\} = 0, \qquad \text{for } j = 1, \ldots, m, \tag{3.1}$$

$$\mathrm{Var}\{\mathbf{U}_{:,j}\} = 1, \qquad \text{for } j = 1, \ldots, m. \tag{3.2}$$

Let $\mathbf{u}_0 \in \mathbb{R}^m$ denote the attribute vector of the grid point, for which the rock property value z_0 should be estimated. The attribute vector $\mathbf{u}_0$ is normalised corresponding to the normalisation of the attribute data in Equations (3.1) and (3.2).

3.3.3 The Kriging Estimator

The description of kriging estimators is based on the description given by Goovaerts (1997). The kriging estimator $Z^*(\mathbf{u})$ is a variant of the basic linear regression estimator defined as:

$$Z^*(\mathbf{u}_0) - m(\mathbf{u}_0) = \sum_{i \in \mathcal{N}} \lambda_i \left(Z(\mathbf{u}_i) - m(\mathbf{u}_i) \right), \tag{3.3}$$

where λ_i is the weight assigned to datum z_i, which is interpreted as a realisation of the random variable $Z(\mathbf{u}_i)$. The interpolation uses a subset, $\mathcal{N}$, of the data points, i.e., $\mathcal{N} \subseteq \{1, \ldots, n\}$ is the set of data points that are in the neighbourhood of $\mathbf{u}_0$ and therefore should be included in the prediction of z_0. Moreover, $m(\mathbf{u})$ is the expected value of the random variable at the location $\mathbf{u}$ and it is denoted the trend function as it describes the general trend in data.

Considering the unknown value z_0 and the data z_i as realisations of random variables $Z(\mathbf{u}_0)$ and $Z(\mathbf{u}_i)$ means the estimation error can be defined as another random variable $Z^*(\mathbf{u}_0) - Z(\mathbf{u}_0)$. The kriging weights λ_i for $i = 1, \ldots, m$ in Equation (3.3) are then defined as the set of weights that minimises the variance of the estimation error:

$$\hat{\sigma}^2 = \mathrm{Var}\{Z^*(\mathbf{u}_0) - Z(\mathbf{u}_0)\}. \tag{3.4}$$

The kriging estimator is ensured to be unbiased as the minimisation of the error variance is done while requiring the mean of the estimation error to be zero. The minimisation of $\hat{\sigma}^2$ from Equation (3.4) is done subject to the constraint:

$$\mathrm{E}\{Z^*(\mathbf{u}_0) - Z(\mathbf{u}_0)\} \quad = \quad 0.$$

This implies that the kriging estimator is, per definition, the best linear unbiased estimator.

While the objective of all kriging methods is to minimise the error variance, their assumptions of trend functions $m(\mathbf{u})$ differ. The three most common types of kriging are simple kriging, ordinary kriging and universal kriging (Goovaerts, 1997).

For each type of kriging, the kriging weights are computed by inserting the trend model into Equation (3.3). This is used to express the error variance from Equation (3.4) in terms of residual covariance values, and the error variance is then minimised. The minimisation, performed using differentiation, includes solving a linear system of equations. For ordinary and universal kriging the non-biased condition calls for definition of a Lagrangian function and Lagrange parameters. For an introduction to constrained optimisation and Lagrangian functions we refer the reader to text books on the subject, for instance the work by Nocedal and Wright (2000).

To formulate the linear system of equations we introduce the following notation. Let C denote the stationary covariance function of the random function $Z(\mathbf{u})$. We then define the data-to-data covariance matrix, $\mathbf{K}$, and the data-to-unknown covariance vector, $\mathbf{k}$, as follows:

$$\mathbf{K} \quad = \quad \begin{pmatrix} C(\mathbf{u}_1 - \mathbf{u}_1) & \cdots & C(\mathbf{u}_1 - \mathbf{u}_n) \\ \vdots & \vdots & \vdots \\ C(\mathbf{u}_1 - \mathbf{u}_n) & \cdots & C(\mathbf{u}_n - \mathbf{u}_n) \end{pmatrix}, \tag{3.5}$$

$$\mathbf{k} \quad = \quad \begin{pmatrix} C(\mathbf{u}_1 - \mathbf{u}_0) \\ \vdots \\ C(\mathbf{u}_n - \mathbf{u}_0) \end{pmatrix}. \tag{3.6}$$

The kriging weights are collected in a vector $\boldsymbol{\lambda}$:

$$\boldsymbol{\lambda} = \begin{pmatrix} \lambda_i \\ \vdots \\ \lambda_n \end{pmatrix}.$$

3.3.3.1 Simple Kriging

As the name implies this is the simplest type of kriging. It is done by assuming a constant and known trend model, $m(\mathbf{u}) = \overline{m}$ for all $\mathbf{u}$. This results in the kriging weights being the solution to the following linear system of equations:

$$\mathbf{K}\boldsymbol{\lambda} = \mathbf{k}. \tag{3.7}$$

The simple kriging system in Equation (3.7) has a unique solution if and only if the data-to-data covariance matrix is non-singular This is the case when:

- no two data are collocated, i.e., $\mathbf{u}_i \neq \mathbf{u}_j$ for all $i \neq j$,
- the covariance function C is permissible.

Basic semivariogram models such as the nugget effect model, the spherical model, the exponential model and the Gaussian model are all permissible (Christakos, 1984). A common choice of covariance model that is guaranteed to be permissible is based on linear combinations of these basic semivariogram models.

Having computed the simple kriging weights, the kriging mean and variance estimate can be computed as:

$$\begin{aligned} Z^*_{SK}(\mathbf{u}_0) &= \sum_{i\in\mathcal{N}} \lambda_i \; Z(\mathbf{u}_i) + \left(1 - \sum_{i\in\mathcal{N}} \lambda_i\right) \overline{m}, \\ \sigma^2_{SK}(\mathbf{u}_0) &= C(0) - \sum_{i\in\mathcal{N}} \lambda_i \; C(\mathbf{u}_i - \mathbf{u}_0). \end{aligned}$$

3.3.3.2 Ordinary Kriging

The assumption from simple kriging of a known and constant trend model is often considered unrealistic. Ordinary kriging relaxes this stationarity assumption and assumes an unknown but constant trend function within a neighbourhood $\mathcal{N}$.

The non-biased condition implies an additional constraint, namely that the ordinary kriging weights sum to 1:

$$\sum_{i\in\mathcal{N}} \lambda_i = 1. \tag{3.8}$$

The ordinary kriging weights are computed as the solution to the following system of equations:

$$\begin{pmatrix} \mathbf{K} & \mathbf{e} \\ \mathbf{e}^T & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{\lambda} \\ \mu \end{pmatrix} = \begin{pmatrix} \mathbf{k} \\ 1 \end{pmatrix}, \tag{3.9}$$

where $\mathbf{e}$ is the column vector of all ones, and μ is the Lagrange parameter for the constraint from Equation (3.8) forcing the kriging estimator to be non-biased. It is seen that the ordinary kriging system in Equation (3.9) resembles the simple kriging system from Equation (3.7).

Having computed the ordinary kriging weights, the kriging mean and variance estimate can be computed as:

$$\begin{aligned} Z^*_{OK}(\mathbf{u}_0) &= \sum_{i \in \mathcal{N}} \lambda_i \; Z(\mathbf{u}_i), \\ \sigma^2_{OK}(\mathbf{u}_0) &= C(0) - \sum_{i \in \mathcal{N}} \lambda_i \; C(\mathbf{u}_i - \mathbf{u}_0) - \mu. \end{aligned}$$

The ordinary kriging mean is computed from Equation (3.3) using the non-biased condition in Equation (3.8).

3.3.3.3 Universal Kriging

Universal kriging proposed by Journel and Huijbregts (1978) is also known as kriging with a trend. The latter name was introduced by Journel and Rossi (1989) and is by some considered the more suitable.

Universal kriging assumes a trend model that is a linear combination of k trend functions. The coefficient for each trend function $f_j(\mathbf{u})$ is unknown and assumed constant within neighbourhoods similar to the unknown trend for ordinary kriging.

Universal kriging imposes the following constraints to ensure the kriging estimator is non-biased:

$$\sum_{i \in \mathcal{N}} \lambda_i f_j(\mathbf{u}_i) = f_j(\mathbf{u}_0) \quad \text{for} \quad j = 1, \ldots, k. \tag{3.10}$$

Universal kriging gives rise to the following linear system of equations:

$$\begin{pmatrix} \mathbf{K} & \mathbf{F} \\ \mathbf{F}^T & \mathbf{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\mu} \end{pmatrix} = \begin{pmatrix} \mathbf{k} \\ \mathbf{f} \end{pmatrix}, \tag{3.11}$$

where:

$$\mathbf{F} = \begin{pmatrix} f_1(\mathbf{u}_1) & \cdots & f_k(\mathbf{u}_1) \\ \vdots & \vdots & \vdots \\ f_1(\mathbf{u}_n) & \cdots & f_k(\mathbf{u}_n) \end{pmatrix},$$

$$\mathbf{f} = \begin{pmatrix} f_1(\mathbf{u}_0) \\ \vdots \\ f_k(\mathbf{u}_0) \end{pmatrix}.$$

Also $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_k)^T$ is a vector of Lagrange parameters for the constraints in Equation (3.10) and $\mathbf{0}$ is the appropriately sized matrix of all zeros. The lower block row of the universal kriging system in Equation 3.11 consists of the constraints from Equation (3.10).

Having computed the universal kriging weights, the kriging mean and variance estimate can be computed as:

$$Z^*_{UK}(\mathbf{u}_0) = \sum_{i \in \mathcal{N}} \lambda_i \; Z(\mathbf{u}_i),$$

$$\sigma^2_{UK}(\mathbf{u}_0) = C(0) - \sum_{i \in \mathcal{N}} \lambda_i \; C(\mathbf{u}_i - \mathbf{u}_0) - \sum_{j=1}^{k} \mu_j f_j(\mathbf{u}_0).$$

The universal kriging mean is computed from Equation (3.3) using the set of constraints from the lower block row of the kriging system in Equation (3.11).

3.3.3.4 Example of Kriging Estimators

To illustrate the differences between the three types of kriging estimators simple kriging, ordinary kriging and universal kriging, we have generated a one-dimensional data set with nine data points. Each of the three types

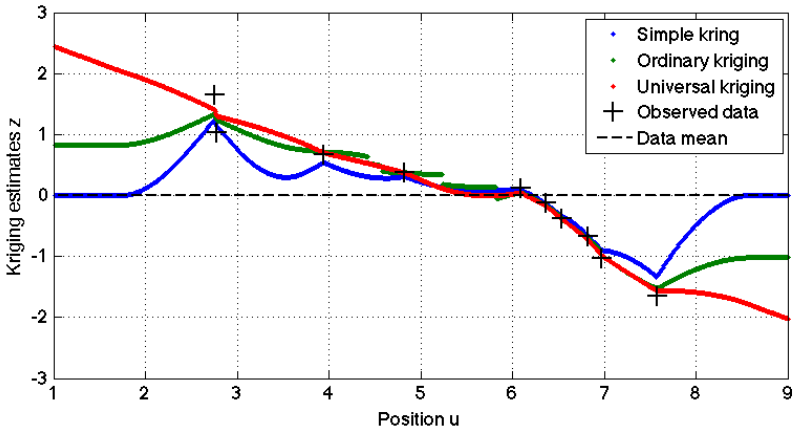


Figure 3.1: Example of the different types of kriging estimators.

of kriging has been used to interpolate the data and the resulting kriging estimates as well as original data are shown in Figure 3.1.

The data is Gaussian distributed with mean value 0 and variance 1. For all three types of kriging we assume a spherical covariance function with a range of 1 and we add a nugget effect of 20% of the sill. The presence of a nugget term in the covariance function becomes clear by the kriging estimates not interpolating the data points exactly.

The choice of trend model becomes most clear when we estimate in locations far from the locations of the data points. This is the case both when we interpolate in areas with low density of data or when we extrapolate beyond the data points. The trend term of the kriging estimator then dominates the kriging estimate causing it to converge to the trend model.

The simple kriging estimate is at all times pulled towards the constant mean of 0. This also means that extrapolating results in constant estimates equal to the trend.

The ordinary kriging estimate assumes a constant trend within each neigh-

bourhood. We have chosen to let a neighbourhood consist of the three nearest data points. This causes the kriging estimate to become discontinuous when the set of the nearest data points changes. It also results in the kriging estimate converging to different values extrapolating left and right of the data points. The trend to the left is higher than the trend to the right, as the first three data points have higher values then the last three data points.

The universal kriging assumes a linear trend. Based on the observed data this appears to be the most suitable of the three trend models. The universal kriging estimator is, when the density of data is high, not that different from the previous estimates. However, when the data points are further apart, the choice of a non-constant trend model is clear, as it, unlike the two previous estimators does not pull the estimates towards a constant. When extrapolating, the universal kriging estimate converges to a straight line. Opposite to ordinary kriging, we have used all data points for estimating in all locations. This means, the coefficients of the linear trend function are constant in the entire domain and it is the same linear trend model the kriging estimate converges to when extrapolating left or right of the data.

3.4 Other Aspects of the Method

After having presented the idea leading to development of the method and its key elements we will now go into details with various aspects of the method. This section elaborates on the steps in the outline of the method from Section 3.2 and discusses the choices made.

3.4.1 Normal Score Transformation

Kriging estimation assumes that the data z_i for $i = 1, \ldots, n$ are samples from a Gaussian distribution. This is rarely the case for rock properties

that, for instance, can be characterised by positive values. This is the case with porosity, where porosity levels by definition are between 0 and 1.

Before performing kriging interpolation the data is therefore transformed such that it represents samples from a Gaussian distribution. This is done by a normal score transformation (Goovaerts, 1997).

Figure 3.2 shows an example of applying a normal score transformation. Figure 3.2a shows the original, clearly non-Gaussian data. The data could, for instance, be describing a rock property of the subsurface that variates around either a low or a high value. This type of distribution is typical for rock properties such as porosity and permeability. These will vary locally in different regions of the subsurface, and the regions will either be characterised by low values because they are not porous / low permeable or characterised by high values as they are highly porous / high permeable.

The cumulative distribution of the data is seen in Figure 3.2c. Also this reveals that the data does not follow a Gaussian distribution. We now apply a normal score transformation to the data. Figure 3.2b shows a histogram of the normal score transformed data and Figure 3.2d shows its cumulative distribution. Both show that the transformed data indeed follows a Gaussian distribution.

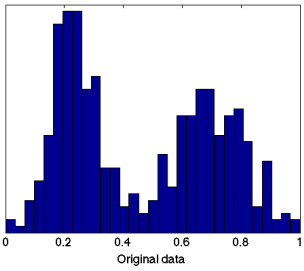
Normal score transformation proceeds in three steps (Goovaerts, 1997):

1. The n data points are ranked from smallest to largest and assigned the ranks $1, 2, \ldots, n$.

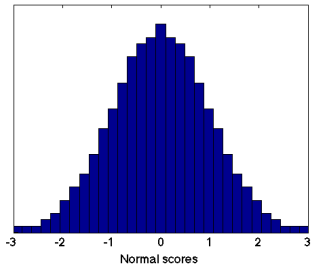
2. The sample cumulative frequency p_k^* of the data point with rank k is computed as:

$$p_k^* = \frac{k - 0.5}{n}.$$

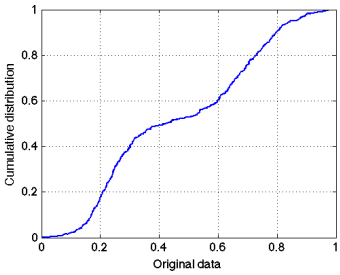
   Alternatively, in case the data is not evenly distributed declustering can be performed by assigning unequal weights to the data points.



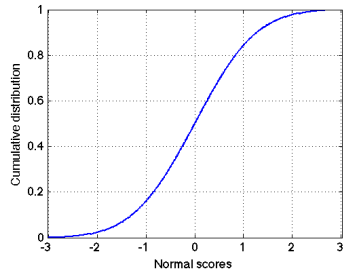
(a) Histogram of original data



(b) Histogram of transformed data



(c) Cumulative distribution of the original data



(d) Cumulative distribution of the normal score transformed data

Figure 3.2: Example of a normal score transformation. The original data and its cumulative distribution is shown to the left. The normal score transformed data and its cumulative distribution is shown to the right. As an example, we see by comparing the cumulative distributions that a data point with original value of 0.6 is transformed into a normal score value of approximately 0.25.

3. The normal score transform of the data point with rank k is matched to the corresponding p_k^*-quantile of the standard Gaussian cumulative density function.

A feature of the normal score transformation is that the probability of being either smaller than the lowest data point or greater than the highest data point is zero. The normal score transformation can also be modified such that these limits are provided by the user and not determined automatically from the data.

As the kriging interpolation is performed of the normal score transformed rock property values, the kriging estimates and standard deviations do not have the same unit as the well log data. The kriging estimates and lower or upper limit of confidence intervals must be back-transformed using the inverse transformation of the normal score transformation before they can be interpreted.

3.4.2 Orthogonal Transformations

A properly chosen orthogonal transformations is a widely used tool to explore the underlying structure in a multidimensional data set. The higher the dimension of the data set, the more complicated it is to analyse the data and detect, for instance, correlation between multiple variables. It is a topic often thoroughly treated in text books on multivariate statistical analysis such as the work by Anderson (1984).

Basically, orthogonal transformations are used to identify a suitable lower-dimensional subspace and approximate the data in this space. An orthogonal transformation projects the data onto a subspace spanned by uncorrelated variables that are linear combinations of the original variables. The new set of variables are usually referred to as components. Using the same number of components as the number of original variables would yield a complete description of the data. Whereas, reducing the number of compo-

nents used to describe the data results in an approximation of the original data.

The goal for any orthogonal transformation is to construct the components wisely. The resulting approximation of data should contain as much information from data as necessary while using as few components as possible.

An orthogonal transformation can also reveal if the data only spans a subspace of the high-dimensional space in which it lives. For two- or three-dimensional data this can be determined simply by plotting the data, but for higher-dimensional data this is no trivial task. However, it is easily determined using an appropriate orthogonal transformation and inspecting the last components. If the data is in fact only spanning a subspace of the high-dimensional space, the last components will contain no information. Also depending on the quality of the data, the last number of components might only describe the noise in data. In that case, an orthogonal transformation of the data will map it into a lower-dimensional subspace by filtering out the noise. This can be done without any significant loss of information.

In connection with kriging interpolation in attribute space we have applied two different orthogonal transformations that we will now discuss.

3.4.2.1 PCA Transformation

Principal component analysis (PCA) as formulated by Hotelling (1933) is a very common technique used in applications in many different fields[1]. It approximates a data set by a new set of variables called principal components. Each principal component is constructed such that it accounts for as much variation in the data as possible. This means, that the first principal component holds the most information of the data, the second component holds the second most information etc.

[1]PCA is closely related to singular value decomposition (SVD) which is a technique well-known in the field of scientific computing (Golub and Reinsch, 1970)

The ith PCA component $\mathbf{v}_i$ is defined as a linear combination of the seismic attributes collected in the data matrix, $\mathbf{U}$:

$$\mathbf{v}_i = \mathbf{U}\boldsymbol{\alpha}^*.$$

The vector of coefficients, $\boldsymbol{\alpha}^*$, is the optimal solution to the following optimisation problem (Hastie et al, 2009):

$$\max_{\boldsymbol{\alpha}} \quad \mathrm{Var}\{\mathbf{U}\boldsymbol{\alpha}\} \tag{3.12}$$

$$\text{w.r.t.} \quad \|\boldsymbol{\alpha}\|_2 = 1, \tag{3.13}$$

$$\mathbf{v}_j^T \mathbf{U}\boldsymbol{\alpha} = 0, \qquad \text{for} \quad j = 1, \ldots, i-1. \tag{3.14}$$

The constraint in Equation (3.13) normalises the coefficient vector to ensure a unique solution. For all but the first component, the set of constraints in Equation (3.14) ensures that the ith component is orthogonal to the previously computed components $\mathbf{v}_1, \ldots, \mathbf{v}_{i-1}$.

Whereas the PCA transformation is well-suited to discover the structures of a high-dimensional data set our objective is to improve the interpolation of the rock property in the high-dimensional seismic attribute space. Applying PCA of the seismic attribute space would be based on a assumption that the variation in the seismic data is key to successfully describe the variation in the rock property. This might, generally, be a valid assumption. On the other hand, we cannot know if the PCA transformation emphasises variation in data that is not useful when describing the rock property.

3.4.2.2 PLS Transformation

The lack of inclusion of the dependant data to the PCA transformation makes us turn to another orthogonal transformation, namely the one known from partial least squares (PLS) regression developed by Wold (1966)[2]. As

[2]PLS is equivalent to the conjugate gradient method used in the field of numerical analysis (Wold et al, 1984)

the following will show, PLS transformation is to some extent similar to PCA transformation. It has also been used in applications in a variety of fields; it was originally used in the social sciences and today it is widely used in chemometrics.

The ith PLS component $\mathbf{p}_i$ is defined as a linear combination of the seismic attributes collected in the data matrix, $\mathbf{U}$:

$$\mathbf{p}_i \quad = \quad \mathbf{U}\boldsymbol{\alpha}^*.$$

Again the vector of coefficients $\boldsymbol{\alpha}^*$ is determined as the optimal solution to an optimisation problem. The optimisation problem is similar to that of Equation (3.12), which determines the coefficient vectors for PCA components. However, as PLS takes the dependant data, $\mathbf{z}$, into account another factor has been inserted in the objective function (Hastie et al, 2009):

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & \mathrm{Var}\left\{\mathbf{U}\boldsymbol{\alpha}\right\} \mathrm{Corr}^2\left\{\mathbf{z}, \mathbf{U}\boldsymbol{\alpha}\right\} \\ \text{w.r.t.} \quad & \|\boldsymbol{\alpha}\|_2 = 1, \\ & \mathbf{p}_j^T \mathbf{U}\boldsymbol{\alpha} = 0, \qquad \text{for} \quad j = 1, \ldots, i-1. \end{aligned} \tag{3.15}$$

The optimisation of objective in Equation (3.15) is done with respect to a set of normalisation and orthogonality constraints similar to the constraints from PCA.

3.4.2.3 Example of Orthogonal Transformations

Figure 3.3 shows a set of $n = 100$ data points located in a two-dimensional space. The value of the data points can be inferred from the colour bar. The data has been generated as a linear combination of the coordinates plus a random Gaussian noise component. Applying linear regression to the data in the two-dimensional space yields an average residual of 0.76. We will now show the effect of reducing the two-dimensional coordinate space to a

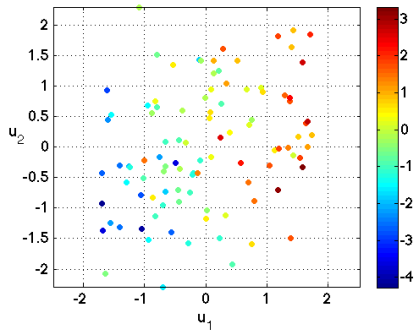


Figure 3.3: The figure shows a set of data points where each data point is assigned a value according to the color bar and a set of coordinates (u_1, u_2). The value of a data point depends on its coordinates as well as random noise of type Gaussian.

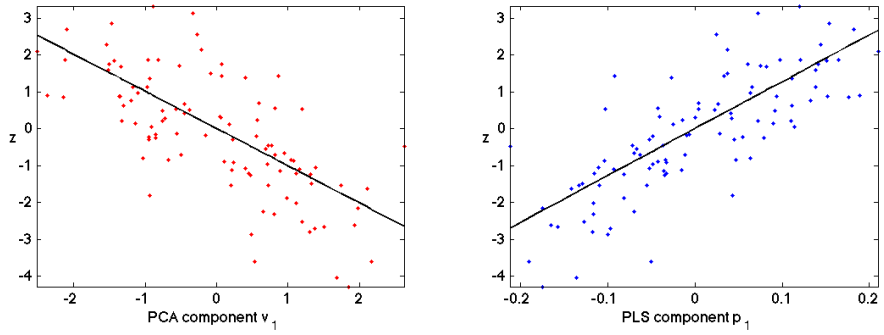
one-dimensional subspace. We will apply both a PCA transformation and a PLS transformation, and discuss the differences in the results.

Figure 3.4 shows linear regression in the one-dimensional transformed coordinate subspace. The coordinate space has been transformed using respectively a PCA transformation (3.4a) and a PLS transformation (3.4b). The one-dimensional space are then spanned by the first component vectors v_1 and p_1, respectively.

To evaluate the performance of the linear regression we compute the correlation between the data values and the estimated data values, $\mathrm{Corr}\left\{\mathbf{z}, \mathbf{z}^{est}\right\}$, and the average absolute residual, $\mathrm{E}\left\{|\mathbf{z} - \mathbf{z}^{est}|\right\}$. These are defined as follows:

$$\mathrm{Corr}\left\{\mathbf{z}, \mathbf{z}^{est}\right\} = \frac{n\sum z_i z_i^{est} - \sum z_i \sum z_i^{est}}{\sqrt{n\sum z_i^2 - \left(\sum z_i\right)^2}\sqrt{n\sum \left(z_i^{est}\right)^2 - \left(\sum z_i^{est}\right)^2}}, \quad (3.16)$$

$$\mathrm{E}\left\{|\mathbf{z} - \mathbf{z}^{est}|\right\} = \frac{1}{n}\sum |z_i - z_i^{est}|. \quad (3.17)$$



(a) PCA transformation **(b)** PLS transformation

Figure 3.4: Linear regression of the data $\mathbf{z}$ performed in the reduced 1D transformed coordinate subspace spanned by the first component $\mathbf{p}_1$ created by either a PCA transformation (left) or a PLS transformation (right). Table 3.1 shows the evaluation of the linear regression for each of the two cases.

	1D space PCA	1D space PLS	full 2D space
$\mathrm{Corr}\left\{\mathbf{z}, \mathbf{z}^{est}\right\}$	-0.70	0.78	0.80
$\mathrm{E}\left\{\lvert\mathbf{z} - \mathbf{z}^{est}\rvert\right\}$	0.91	0.80	0.76

Table 3.1: The performance of linear regression in a low-dimensional (1D) transformed coordinate subspace, transformed using either PCA or PLS, compared to linear regression in the full 2D coordinate space. The table shows the correlation coefficient between the data $\mathbf{z}$ and the estimated values $\mathbf{z}^{est}$ and the average value of the absolute residuals.

The sums in Equations (3.16) and (3.17) run over all $i \in \{1, \ldots, n\}$, as $\mathbf{z}, \mathbf{z}^{est} \in \mathbb{R}^n$. Table 3.1 shows the results for each of the one-dimensional cases as well as for linear regression in the full two-dimensional coordinate space.

The correlation coefficients for the 1D regression are -0.70 and 0.78, re-

spectively. The correlation coefficient for linear regression in the full 2D coordinate space is 0.80. Considering the amount of data it is implausible that the data has been over-fitted and the correlation coefficient from the 2D case can be viewed as an upper limit for what we can achieve at best. This illustrates exactly the main feature of the PLS transformation, namely that it creates components that correlate well with the dependant data.

However, high correlation by itself is not satisfactory unless the linear regression benefits from it. To determine if this is the case we look at the average absolute residual defined in Equation (3.17). Again, the average absolute residuals achieved by linear regression in the full 2D coordinate space acts as a limit, and as expected it is lower than for both the 1D cases. We also notice that the PLS transformation results in an estimation error that is significantly smaller than when using the PCA transformation.

We can conclude that the PLS transformation is significantly better than the PCA transformation, when it comes to creating a lower-dimensional coordinate subspace suitable for performing linear regression. The reduction of dimensions implies a loss of information. It might also be that it filters out noise; that will typically be more evident for high-dimensional data sets. However, the PLS transformation causes a smaller loss of valuable information and it therefore yields more accurate results.

This example illustrates the motivation behind using a PLS transformation of the seismic attribute space rather than the well-known PCA transformation. The latter was used in the original conference paper in Appendix A. Especially when combined with universal kriging with a linear trend, a PLS transformation seems like a very interesting choice. We refer to the paper in Appendix G, where the PLS transformation has been applied to a case study of predicting porosity levels in the South Arne Field.

3.4.3 Reduction of the Transformed Attribute Space

Having decided on the type of orthogonal transformation for transforming the seismic attribute there is still an important question left to answer.

That is, what is the proper number of components to include, i.e., what should the dimension of the reduced coordinate space be. The compromise is between loss of information by leaving out components (which in the best case can be reduced to filtering out noise but in the worst case means loosing out on valuable information), and including too many components, which means increasing the computational complexity without improving the results.

In the illustrative example from Figure 3.4 the choice of dimensionality of the transformed and reduced attribute space was trivial as the original coordinate space was only two-dimensional. This is not the case for actual seismic data.

Both PCA transformation and PLS transformation can provide a measure of how well the data is approximated by the components. For instance, for PCA the components are the eigenvectors of the sample data covariance matrix. The percentage of data explained by a subset of the components is equal to the ratio of the sum of their eigenvalues to the sum of all eigenvalues. Such a measure can provide an indication as to how many components are sufficient to approximate the data, although it does not take estimation abilities into account.

The statistical technique known as cross-validation is widely used to asses how the results of a statistical analysis will generalise to an independent data set. The technique can also be used as a guidance for choosing the appropriate number of components for our interpolation method.

A rough scheme for cross-validation to determine the optimal dimension of the transformed and reduced attribute space based on m seismic attributes is provided by Algorithm 1.

The estimation in the core of the algorthm can be done by linear regression, which is computationally simple and can be done relatively fast even for large number of data sets.

The optimal number of components is typically chosen as the one resulting in the smallest estimation error. However, it can also be chosen as the small-

Algorithm 1: Cross-validation

Input: Dependant data and the m components.
Output: Optimal number of components to include, $\widehat{m}^*$.
Randomly divide the data into K subsets.
for *For each possible number of components* $\widehat{m} \in \{1, 2, \ldots, m\}$ **do**
    for *For each data subset* $k \in \{1, 2, \ldots, K\}$ **do**
        Estimate the data of the kth subset using only the remaining $K - 1$ subsets.
        Compute the estimation error.
    end
    Compute the average estimation error when using $\widehat{m}$ components, i.e. the average of the K estimation errors just computed.
end
Plot the average estimation error as a function of the number of components $\widehat{m}$.
Choose the optimal number of components $\widehat{m}^*$.

est number of components achieving an estimation error of an acceptable level. Which error level is acceptable of course depends on the problem at hand. One can use the lowest achievable estimation error as a target and then accept a certain deviation from this.

A less elegant solution is that applied in the work in Appendix G. Here kriging interpolation is carried out for every number of components and an estimation error based on a blind data set is computed. The behaviour of this estimation error as a function of the number of components is then discussed. This is an elaborate approach that is only possible when a large amount of data is available and whose feasibility depends on the computational complexity of the problem.

3.4.4 Maximum Likelihood Estimation of Covariance Parameters

The optimal parameters of the covariance function appearing in Equations (3.5) and (3.6) are determined using maximum likelihood estimation. We will use the MLE approach of Pardo-Igúzquiza (1997, 1998). In Pardo-Igúzquiza (1998) several techniques for MLE are compared (Samper and Neuman, 1989a,b,c; Kitanidis and Lane, 1985; Diggle et al, 2003).

A spatial variable can typically be divided into two components. One is a low-frequent trend component and the other is a high-frequent residual component. The trend component is usually assumed to be a linear combination of basis functions, such as low-order polynomials, with unknown coefficients. When performing universal kriging in a seismic attribute space with a linear trend model we can write the trend as:

$$m(\mathbf{u}) \quad = \quad \begin{pmatrix} 1 & \mathbf{u}^T \end{pmatrix} \boldsymbol{\beta}.$$

Here $\boldsymbol{\beta} \in \mathbb{R}^{m+1}$ is the vector of unknown coefficients of the basis functions describing the trend $m(\mathbf{u})$. The trend values for the known data is then $\begin{pmatrix} \mathbf{e} & \mathbf{U} \end{pmatrix} \boldsymbol{\beta} = \mathbf{X}\boldsymbol{\beta}$.

The residual component, $\mathbf{z} - \mathbf{X}\boldsymbol{\beta}$, is characterised statistically by its covariance function. The parameters $\boldsymbol{\theta}$ of the covariance function C is inferred using maximum likelihood estimation, i.e., they are defined as the set of parameters $\boldsymbol{\theta}^*$ maximising the likelihood of obtaining the data $\mathbf{z}$ given the trend model and the covariance function.

Assuming the data $\mathbf{z}$ follows a multivariate Gaussian distribution the conditional pdf of the data is defined as:

$$p(\mathbf{z}|\boldsymbol{\beta}, \boldsymbol{\theta}) \quad = \quad (2\pi)^{\frac{2}{n}} |\mathbf{V}|^{-\frac{1}{2}} \exp\left(-\frac{1}{2} \|\mathbf{z} - \mathbf{X}\boldsymbol{\beta}\|^2_{\mathbf{V}} \right). \tag{3.18}$$

Here $|\mathbf{V}|$ is the determinant of the covariance matrix of the of the data with covariance parameters $\boldsymbol{\theta}$. The $\mathbf{V}$-norm of a vector $\mathbf{a}$ is defined from $\|\mathbf{a}\|^2_V = \mathbf{a}^T \mathbf{V}^{-1} \mathbf{a}$.

Factorising the covariance matrix as $\mathbf{V} = \sigma^2\mathbf{Q}$ and using σ^2 as the variance of the residual, Equation (3.18) is equivalent to:

$$p(\mathbf{z}|\boldsymbol{\beta}, \sigma^2, \boldsymbol{\theta}) = (2\pi)^{\frac{2}{n}}\sigma^{-n}|\mathbf{Q}|^{-\frac{1}{2}}\exp\left(-\frac{1}{2\sigma^2}\|\mathbf{z}-\mathbf{X}\boldsymbol{\beta}\|^2_{\mathbf{Q}}\right).$$

This pdf gives rise to the following negative log-likelihood function:

$$L(\boldsymbol{\beta}, \sigma^2, \boldsymbol{\theta}|\mathbf{z}) = \frac{n}{2}\ln 2\pi + n\ln\sigma + \frac{1}{2}\ln|\mathbf{Q}| + \frac{1}{2\sigma^2}\|\mathbf{z}-\mathbf{X}\boldsymbol{\beta}\|^2_{\mathbf{Q}}. \quad (3.19)$$

The negative log-likelihood function is minimised for:

$$\boldsymbol{\beta}^* = \left(\mathbf{X}^T\mathbf{Q}^{-1}\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{Q}^{-1}\mathbf{z}, \quad (3.20)$$

$$\sigma^{2^*} = \frac{1}{n}\|\mathbf{z}-\mathbf{X}\boldsymbol{\beta}\|^2_{\mathbf{Q}}. \quad (3.21)$$

The optimal coefficients $\boldsymbol{\beta}^*$ defined by Equation (3.20) is the generalised least squares estimate of $\boldsymbol{\beta}$. The optimal residual variance σ^{2^*} is obtained by differentiating the negative log-likelihood function in Equation (3.19) with respect to σ^2 and setting the derivative equal to 0.

Inserting Equation (3.20) and (3.21) into the negative log-likelihood function in Equation (3.19) yields a function of only the covariance model parameters $\boldsymbol{\theta}$:

$$L(\boldsymbol{\beta}^*, \sigma^{2^*}, \boldsymbol{\theta}) = \frac{n}{2}\ln 2\pi + \frac{1}{2}\ln\|\mathbf{z}-\mathbf{X}\boldsymbol{\beta}^*\|^2_{\mathbf{Q}} - \frac{n}{2}\ln n + \frac{1}{2}\ln|\mathbf{Q}| + \frac{n}{2}.$$

The optimal covariance parameters $\boldsymbol{\theta}^*$ are then determined as:

$$\begin{aligned}\boldsymbol{\theta}^* &= \operatorname*{argmin}_{\boldsymbol{\theta}} L(\boldsymbol{\beta}^*, \sigma^{2^*}, \boldsymbol{\theta}) \\ &= \operatorname*{argmin}_{\boldsymbol{\theta}} \left\{\ln\|\mathbf{z}-\mathbf{X}\boldsymbol{\beta}^*\|^2_{\mathbf{Q}(\boldsymbol{\theta})} + \ln|\mathbf{Q}(\boldsymbol{\theta})|\right\}. \quad (3.22)\end{aligned}$$

Standard software for optimisation such as the Optimization Toolbox for Matlab can be used to compute the optimal covariance parameters $\boldsymbol{\theta}^*$. See Lophaven et al (2002) for a discussion of the numerical aspects of computing the optimal covariance parameters in Equation (3.22).

3.5 Examples in Papers

The method of kriging in a transformed and reduced attribute space was first presented in an extended abstract in the proceedings from the *14th IAMG Conference, 2010*. This work is listed in Appendix A and it provides an introduction to the method as well as a test case, where data from the South Arne Field is analysed. The data is provided by Hess Corporation.

The main difference between this work and the work presented in the primary paper in Appendix G is the choice of orthogonal transformation of the seismic attribute space. The work in Appendix A uses principal component analysis.

The partial least squares transformation is first introduced in a paper prepared for submission to Geophysics. The paper can be seen in Appendix G. Also this paper presents the test case from the South Arne Field in the Danish part of the North Sea.

3.6 Summary

This chapter presented a method for efficient prediction of rock properties using seismic data. We introduced the interpolation problem, which is well-known in seismic exploration. We discussed how it has been solved previously in the literature by use of different interpolation methods. We then explained the motivation leading to our approach of interpolating in a space spanned by not just spatial coordinates but also seismic attributes.

We provided a clear outline of the method followed by a discussion of the data and interpolation method of which its is based. We discussed how the method is made efficient by use of an orthogonal transformation to approximate the high-dimensional seismic data. The transformation implies that redundant information from correlated seismic attributes are filtered out. This allow for a reduction of dimensionality of the space in which the

interpolation is performed which reduces the computational complexity of the method.

We discussed non-trivial details of the method such as various transformations of both the rock property data and seismic data, how cross-validation can be used to determine the optimal dimension of the subspace used to approximate the seismic data, and how maximum likelihood estimation can be used to determine the optimal parameters of the covariance models of the kriging estimator.

CHAPTER 4

The Frequency Matching Method

A key contribution of this study has been the development of what is now known as the Frequency Matching (FM) method, which is an example of using multiple-point statistics as a priori information when solving inverse problems. The literature review in Chapter 1 revealed the need for a method that can be used to compute the maximum a posteriori (MAP) solution to an inverse problem incorporating multiple-point statistics as prior information, as discussed in Chapter 2.

This chapter supports the work described in the primary paper on the Frequency Matching method, see Appendix C. The chapter provides a brief introduction to the FM method. It defines the notation and terminology necessary to formulate the method. It then presents an outline of the method. And finally, we discuss interesting aspects of the method.

4.1 Introduction

We assume a classical inverse problem as discussed in the beginning of Chapter 2, where the non-linear forward operator, g, is known and we wish to determine the model parameters that map to a set of data observations, $\mathbf{d}^{\text{obs}}$. Applying the Bayesian approach leads to the formulation of the a posteriori pdf of the model parameters in Equation (2.2). Defining a closed form expression for the a priori probability distribution, $\rho_{\mathbf{m}}(\mathbf{m})$, enables us to use optimisation methods to compute the MAP model given by Equation (2.5).

The FM method is applied to an inverse problem associated with a forward problem like the one from Equation (2.1) and with the Gaussian likelihood function from Equation (2.3).

The prior information on the multiple-points statistics of the models is typically available in a training image. The a priori pdf then assesses how likely it is that a given model $\mathbf{m}$, has the same multiple-point statistics as the training image. The FM method assumes an a priori pdf defined as:

$$\rho_{\mathbf{m}}(\mathbf{m}) \quad = \quad const \ \exp\left(-\alpha \ f(\mathbf{m})\right).$$

Here f is a distance function measuring the distance from an image with the model parameters $\mathbf{m}$ to the training image.

The FM model is then defined as the model maximising the a posteriori pdf. Often the more simple but equivalent formulation is used:

$$\mathbf{m}^{\text{FM}} \quad = \quad \underset{\mathbf{m}}{\text{argmin}} \left\{ -\log \sigma_{\mathbf{m}}(\mathbf{m}) \right\}. \tag{4.1}$$

This definition of the prior probability distribution leads to formulation of a combinatorial optimisation problem which will be solved by use of the meta-heuristic Simulated Annealing.

Appendix F holds a paper of the Fortran implementation of the Frequency Matching method. This version is the one that has been used for the test case presented in the paper seen in Appendix C.

4.2 Notation and Terminology

Before formally presenting the FM method we will go though some notation and terminology that is needed in order to formulate the method.

4.2.1 Training Image

The training image is fundamentally central in the frequency matching method. It represents a priori knowledge and thereby it represents the expectations we have of the computed solutions. Journel and Zhang (2006) seeks to give a definition of a training image as a conceptual description of a random process that discloses the prior multiple-point statistics of a model. The training image is assumed to be representative of the random process. Stochastic simulation methods then, for instance, seek to sample other realisations of the same random process.

The training image is representing a priori knowledge and is therefore only reflecting expectations of the model based on previously processed data and other experiences. It is independent of any new data that needs processing and it will therefore not itself be conditioned or capable of conditioning models on such new data.

Regarding the size of the training image, the general consensus is that it should be chosen relative to the size of the structures it describes. Strebelle (2002) recommends choosing a training image that is at least twice as large as the biggest spatial structures it describes. Journel and Zhang (2006) is a bit more vague and only states it should be sufficiently large for ergodicity reasons.

The frequency matching method does not need a training image, only the multiple-points statistics that it describes. In the FM method these are represented by frequency distributions which are to be defined in the Section 4.2.2. In the case that such data is available, the training image itself is not needed.

4.2.2 Basic Image Assumptions

Consider an image $\mathcal{Z} = \{1, 2, \ldots, N\}$ with N voxels (or pixels if the image is only two-dimensional). Each voxel belongs to one out of v categories. We introduce the N variables, $z_1, z_2, \ldots, z_N$, where z_k describes the value of the kth voxel of the image, hence $z_k \in \{0, 1, \ldots, v-1\}$.

It is assumed that the image is a realisation of an unknown, random process satisfying the following three conditions.

1. There exists a subset of the image that we will denote as the neighbourhood of voxel k, $\mathcal{N}_k$. The value of the kth voxel is then conditionally independent of all voxels not in its neighbourhood. Voxel k itself is not contained in the neighbourhood. Let $\mathbf{z}_k$ be a vector of the variables describing the values of the ordered neighbouring voxels in $\mathcal{N}_k$. The conditional independence then implies:

$$f_{\mathcal{Z}}\left(z_k | z_N, \ldots, z_{k+1}, z_{k-1}, \ldots, z_1\right) \quad = \quad f_{\mathcal{Z}}\left(z_k | \mathbf{z}_k\right).$$

2. An image of infinite size has neighbourhoods of identical geometrical shapes. Let (k_x, k_y, k_z) denote the spatial coordinates of voxel k in the image, and similarly let (l_x, l_y, l_z) and (n_x, n_y, n_z) be the coordinates of voxel l and n, respectively. Then the following holds:

$$(n_x, n_y, n_z) \in \mathcal{N}_k \quad \Rightarrow \quad (n_x - k_x + l_x, n_y - k_y + l_y, n_z - k_z + l_z) \in \mathcal{N}_l.$$

3. We assume ergodicity. That means, if the neighbouring voxels of two voxels k and l, have the same values, i.e., the vectors $\mathbf{z}_k$ and $\mathbf{z}_l$ are element-wise identical, then the variables describing the values of voxel k and voxel l follow the same conditional probability distribution:

$$\mathbf{z}_k = \mathbf{z}_l \quad \Rightarrow \quad f_{\mathcal{Z}}\left(z_k | \mathbf{z}_k\right) = f_{\mathcal{Z}}\left(z_l | \mathbf{z}_l\right).$$

4.2.3 Template Function

We define the template function ω as a function that given any voxel k, returns the set of its neighbouring voxels $\mathcal{N}_k$, i.e., $\mathcal{N}_k = \omega(k)$.

4.2.4 Inner Voxels

The second condition concerning images of infinite size is, of course, not applicable in practice. From Section 4.2.1 we know that the image is expected to be significantly larger than the structures it describes and therefore also larger than the neighbourhoods.

Due to the finite size of an image we distinguish between two different kinds of voxels; voxels are either inner voxels or non-inner voxels. The set of inner voxels is defined as follows:

$$\mathcal{Z}_{\text{in}} \quad = \quad \left\{ k \in \mathcal{Z} : |\mathcal{N}_k| = \max_{l \in \mathcal{Z}} |\mathcal{N}_l| \right\}.$$

From this definition follows that inner voxels have the largest neighbourhoods of all voxels in an image. Assuming the size of an image is strictly greater than the size of a neighbourhood, the neighbourhoods of inner voxels are all geometrically identical. Let n be the number of voxels in the neighbourhood of an inner voxel, i.e.:

$$n \quad = \quad \max_{k \in \mathcal{Z}} |\mathcal{N}_k|.$$

4.2.5 Patterns

Each inner voxel is surrounded by identically shaped neighbourhoods. This gives rise to the concept of patterns. The value of an inner voxel and the

values of its neighbouring voxels is interpreted as a pattern, where the inner voxel itself will be denoted the centre of the pattern. An inner voxel is assigned a pattern value, p_k. The pattern value is a unique identifier of the pattern and must be chosen according to the implementation.

Figure 4.1 shows an example of patterns in a two dimensional image consisting of 54 voxels. The template function defines a neighbourhood as follows:

$$\mathcal{N}_k \quad = \quad \{l \in \mathcal{Z} \setminus k : |l_x - k_x| \leq 1, |l_y - k_y| \leq 1\}\,. \tag{4.2}$$

This template function, that marks the grey neighbouring pixels surrounding pixel k in the lower left corner of the figure, yields patterns of nine pixels. Using the template function from Equation (4.2) it means the image has 28 inner pixels. The patterns that they are each the centre of can be seen to the right in the figure.

A straightforward choice of pattern value is a vector of the voxel values themselves, which is a number in the base v numeral system. Another choice is the corresponding number in the base 10 numeral system. Depending on the choice, the pattern of voxel 10 from Figure 4.2 will then have pattern value:

$$p_{10} = \begin{cases} 011001001 & \text{(base } v \text{ numeral system)}, \\ 201 & \text{(decimal numeral system)}. \end{cases}$$

Notice how the pattern value depends not only on the values of the neighbouring pixels in $\mathbf{z}_{10}$ but also on the value of the centre pixel. (For $v = 2$ the base v numeral system is binary numbers.)

A pattern value is uniquely determined by the value of an inner voxel and the values of the n voxels in its neighbourhood. For an image with v categories there are no more than v^{n+1} different pattern values.

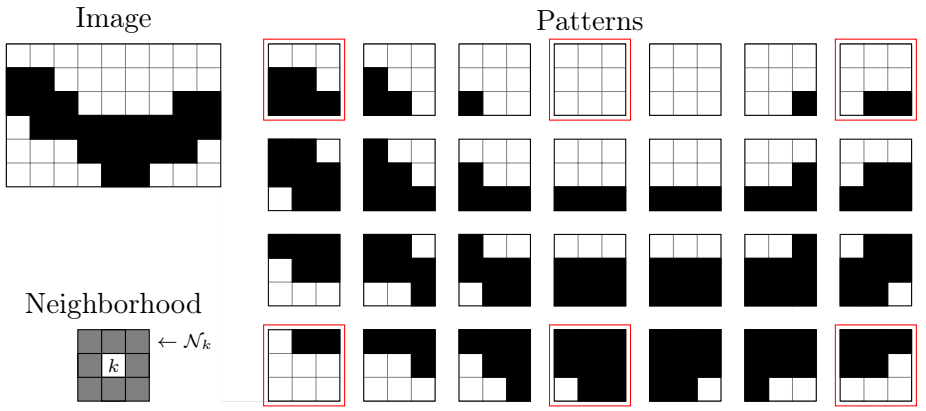


Figure 4.1: Example of patterns found in an image where the template function returns up to 8 closest neighbours of a pixel, resulting in patterns that have geometrical shapes of 3 by 3 subimages. Notice how the patterns are overlapping, i.e., the picture is fully described by only the patterns marked with red.

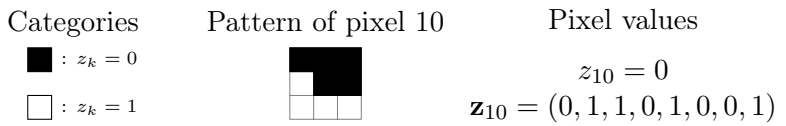


Figure 4.2: Example of possible assignments of pattern values of a pattern found in the image in Figure 4.1. The pixels are numbered column-wise starting in the upper left corner, meaning the 10th pixel is found in the 4th row 2nd column of the image.

4.2.6 Frequency Distributions

The frequency distribution of patterns of an image is the distribution of pattern values. It is generated by scanning through the set of inner voxels, extracting their neighbourhood, and computing their pattern values.

The frequency distribution will typically be extremely sparse. There are two main reasons for this: First, the number of possible patterns v^{n+1} is typically much larger than the number of inner voxels $|\mathcal{Z}_{\text{in}}|$. Therefore, far from all the possible patterns can be seen in an image. Second, images will typically have a certain structure that the patterns capture. This structure is exactly what we wish to reproduce. There will therefore be patterns that do not appear in the image as they are not used in describing its structures. A suitable training image will have many duplicates of patterns, otherwise it is not chosen sufficiently big.

We will use the notation $\boldsymbol{\pi}$ for a frequency distribution of an image:

$$\boldsymbol{\pi} \quad = \quad [\pi_1, \pi_2, \ldots, \pi_{v^{n+1}}] \,,$$

where π_i is the count of appearances of the ith type of pattern in the image.

For the sake of notation we will use $\boldsymbol{\pi}^{\text{TI}}$ when referring to the frequency distribution of a training image. We define the mapping p_ω as the function that takes as input an image and then returns its frequency distribution with respect to the neighbourhood function ω, i.e., $\boldsymbol{\pi} = p_\omega(z_1, z_2, \ldots, z_N)$.

Figure 4.3 shows non-zero entries of the frequency distribution of the image in Figure 4.1. The image has two categories of pixels and the template function in use yields patterns of $n + 1 = 9$ pixels. This means the total number of patterns is 512. The frequency distribution in the figure shows only counts of patterns that are present in the image. Only 25 different patterns are found which is less than 5% of the possible number of patterns. No pattern is found more than twice, which is due to the unrealistically small size of the image.

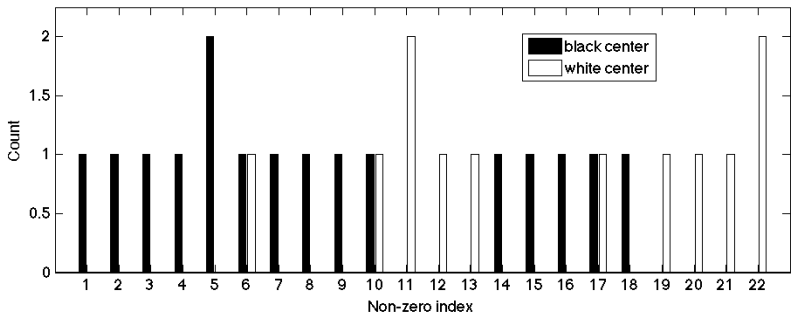


Figure 4.3: Frequency distribution of the image from Figure 4.1. The colour of the bars represents the colour of the centre pixel z_k of each pattern. For instance, the patterns with index 5 have neighbourhood pixel values $\mathbf{z}_k = (1, 1, 0, 1, 0, 1, 0, 0)$. These are the patterns seen in the second row of the two last columns to the right in Figure 4.1.

4.2.7 Distance Measure

The frequency distribution of a training image, $\boldsymbol{\pi}^{\mathrm{TI}}$, is a way of representing the training image by its multiple-point statistics. Using this multiple-point statistics as a priori knowledge when solving inverse problems means we need a way of determining the similarity of images by comparing their frequency distributions.

To define how similar an image is to a training image we introduce a dissimilarity function. This expresses the dissimilarity of the images by computing the distance between their frequency distributions. The Euclidean distance has been used without further explanation by Peredo and Ortiz (2010), who focused on the computational speed-up that could be gained by parallelising the simulated annealing scheme. They fail to reproduce the higher-order statistics of the images in the sense that their computed solution does not show the channel structures seen in the training image. However, the results indicate that using a weighted two-norm is to be preferred over just the

Euclidean distance. So even though none of these choices seem favourable, they show that the choice of dissimilarity function is important.

We have looked for inspiration in the literature, and our choice ended on the distance used by the chi-square test for homogeneity by Sheskin (2004). Given two (or more) samples that each consist of a number of categorical observations, it can be used to test for homogeneity in the distribution of observations.

It is assumed that each of the samples is drawn independently from an unknown underlying population. The test then determines if the data is homogeneous, i.e., if the proportion of observations in each category is consistent in the two samples. We interpret this, letting the image and the training image each be a sample. Each pattern in the images counts as an observation. We then seek a measure of how similar the proportion of observations is in each of the two images. If the images have similar proportions of patterns they will have the same multiple-point statistics, and vice versa.

The statistical test has some requirements that should be satisfied in order for the distance to be χ^2 distributed. However, as we do not wish to perform the statistical test but merely use the distance measure, we are not concerned with these.

As first presented by Lange et al (2012), given the frequency distribution of an image, $\boldsymbol{\pi}$, and of a training image, $\boldsymbol{\pi}^{\mathrm{TI}}$, and by letting

$$\mathcal{I} = \left\{ i \in \left\{1, \ldots, v^{n+1}\right\} : \ \pi_i^{\mathrm{TI}} > 0 \right\} \cup \left\{ i \in \left\{1, \ldots, v^{n+1}\right\} : \ \pi_i > 0 \right\}, \quad (4.3)$$

we can compute what we define as the dissimilarity function value of the image:

$$c(\boldsymbol{\pi}) = \chi^2(\boldsymbol{\pi}, \boldsymbol{\pi}^{\mathrm{TI}}) \quad = \quad \sum_{i \in \mathcal{I}} \frac{(\pi_i^{\mathrm{TI}} - \epsilon_i^{\mathrm{TI}})^2}{\epsilon_i^{\mathrm{TI}}} + \sum_{i \in \mathcal{I}} \frac{(\pi_i - \epsilon_i)^2}{\epsilon_i}. \quad (4.4)$$

Here ϵ_i denotes the count from the underlying distribution of patterns with the ith pattern value for images of the same size as the image. Likewise,

ϵ_i^{TI} denotes the count from the underlying distribution of patterns with the ith pattern value for images of the same size as the training image. These counts are computed as:

$$\epsilon_i = \frac{\pi_i + \pi_i^{\mathrm{TI}}}{n^{\mathcal{Z}} + n^{\mathrm{TI}}}\, n^{\mathcal{Z}}, \tag{4.5}$$

$$\epsilon_i^{\mathrm{TI}} = \frac{\pi_i + \pi_i^{\mathrm{TI}}}{n^{\mathcal{Z}} + n^{\mathrm{TI}}}\, n^{\mathrm{TI}}, \tag{4.6}$$

where $n^{\mathcal{Z}}$ and n^{TI} are the total number of counts of patterns in the frequency distribution of the image and the training image, respectively. We notice when comparing frequency distributions of images of the same size, the counts from the underlying distributions become identical, and are computed as the average of the counts from the two frequency distributions.

4.3 Outline of the Frequency Matching Method

Based on the definition of the FM model from Equation (4.1) we define the Frequency Matching method for solving inverse problems formulated as least squares problems using multiple-point statistics as a priori information as the following optimisation problem:

$$\begin{aligned} \min_{z_1,\ldots,z_N} \quad & \|\mathbf{d}^{\mathrm{obs}} - g(z_1,\ldots,z_N)\|^2_{\mathbf{C_d}} + \alpha\; c(\boldsymbol{\pi}) \\ \text{w.r.t.} \quad & \boldsymbol{\pi} = p_\omega(z_1,\ldots,z_N), \\ & z_k \in \{0,\ldots,v-1\}\,, \qquad \text{for } k = 1,\ldots,N, \end{aligned} \tag{4.7}$$

where $c(\boldsymbol{\pi})$ is the dissimilarity function value of the image defined by Equation (4.4) and α is a weighting parameter. The forward operator g, which traditionally is a mapping from model space to data space, also contains the mapping of the categorical values $z_k \in \{0,\ldots,v-1\}$ for $k = 1,\ldots,N$ of the voxels into the model parameters $\mathbf{m}$ that can take v different discrete values.

The parameter α acts as a regularisation parameter. It is highly dependent on the problem at hand and the user should choose it to balance the weight of the data fit and and the multiple-point statistics of the solution.

Due to the large number of parameters and the highly non-linear behaviour of the prior term in the misfit function from Equation (4.7) we propose to use an heuristic approach for solving the optimisation problem. An obvious choice is to use simulated annealing that is a simple and intuitive yet powerful meta-heuristic for solving combinatorial optimisation problems developed by Kirkpatrick et al (1983). Simulated annealing is a well-known optimisation method that has been used within a wide variety of fields including geosciences (Vestergaard and Mosegaard, 1991; Deutsch and Cockerham, 1994).

Algorithm 2 shows the outline of the FM method for solving inverse problems using an iterative optimisation approach such as simulated annealing.

Algorithm 2: The Frequency Matching Method

Input: Training image, $\mathcal{Z}^{\mathrm{TI}}$, starting image $\mathcal{Z}$
Output: Maximum a posteriori image $\mathcal{Z}^{\mathrm{FM}}$
Compute frequency distribution $\boldsymbol{\pi}^{\mathrm{TI}}$ of training image
Compute frequency distribution $\boldsymbol{\pi}$ of starting image
while *not converged* **do**
    Compute perturbed image $\overline{\mathcal{Z}}$ based on $\mathcal{Z}$
    Compute frequency distribution $\overline{\boldsymbol{\pi}}$ of perturbed image $\overline{\mathcal{Z}}$
    if *accept the perturbed image* **then**
        Set $\mathcal{Z} \leftarrow \overline{\mathcal{Z}}$ and $\boldsymbol{\pi} \leftarrow \overline{\boldsymbol{\pi}}$
    end
end

4.4 Large-Scale and Implementation Aspects

The FM method was developed to model the subsurface of a reservoir. As the reader can surely imagine, this is a large scale problem easily containing millions of variables. It is essential that careful effort is put into the implementation design. Otherwise, we quickly must surrender to out of memory-error messages and unacceptable computation times.

The complexity of the method depends not only on the number of model parameters but also on the number of voxels in a pattern and thereby on n. A naive MATLAB implementation will fail at five by five patterns for a two dimensional test case. This illustrates the need for careful considerations of how to deal especially with the computation of the frequency distributions.

In this section we will discuss precautionary measures that can, and in some cases should, be taken when applying the FM method to a large scale case. Some of the measures are already included in the current version of the Fortran implementation. (The papers in Appendix C and Appendix F discuss some of these. Others are only discussed here.)

4.4.1 Optimal Pattern Size

The size and shape of the patterns depend on the choice of the neighbourhood function ω. Choosing this is no trivial task. The patterns should be big enough to capture the large scale structures of the image but at the same time the complexity of the FM method is dependent on the size of the patterns. The assumptions, discussed in the beginning of this chapter, on which the FM is developed do not set any limitations on the geometrical shape of the neighbourhoods. The most common choice of pattern shape in MPS techniques is, however, the hyper-rectangular pattern of size $(2\Delta_x+1) \times (2\Delta_y+1) \times (2\Delta_z+1)$ voxels resulting from the neighbourhood function:

$$\mathcal{N}_k \quad = \quad \{l \in \mathcal{Z} \setminus k : |l_x - k_x| \leq \Delta_x, |l_y - k_y| \leq \Delta_y, |l_z - k_z| \leq \Delta_z\},$$

where $\Delta_x, \Delta_y, \Delta_z \in \mathbb{N}$. This choice of neighborhood function ensures symmetry in the neighbourhoods, i.e., for any two inner voxels k and l holds $k \in \mathcal{N}_l \Leftrightarrow l \in \mathcal{N}_k$.

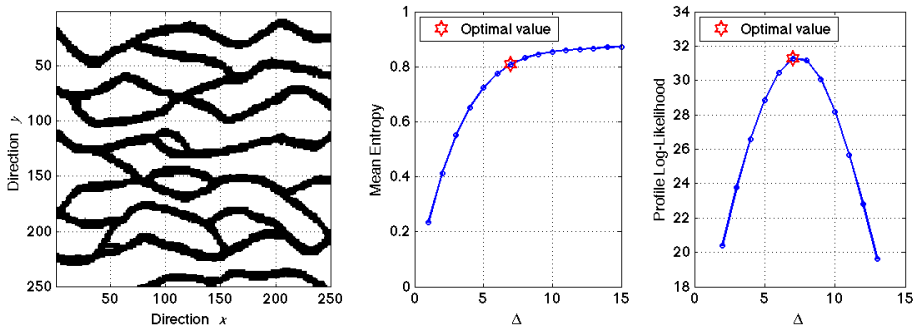
A recent study in stochastic simulations of patterns (Honarkhah, 2011) provides a generic method to determine the optimal size of patterns. In this study the overall idea resembles that of the SNESIM algorithm only instead of simulating one voxel value at a time the method simulates entire patterns. The author has had the goal of developing a method with parameter free learning. As the training image plays the same role in the FM method, we can use the same framework to determine optimal pattern sizes. The approach is based on the work by Mackay (2003) that uses entropy maps to gain an insight into the features of an image.

The approach from Honarkhah (2011) consists of calculating the mean entropy of patterns as a function of increasing pattern sizes. As demonstrated by applying the method to training images with very different structure, the entropy has a specific behaviour as a function of the pattern sizes and this is used to determine the optimal pattern size. Optimal here is defined as the minimal pattern size capturing the stationary features of the image.

The behaviour of the entropy is characterised by two stages. In the first stage is seen a steep increase in entropy. This is while the pattern size has not yet reached its optimum and increasing it further increases also the information captured by the patterns. The second stage appears once the pattern has increased above its optimal size. In this stage the entropy is increasing at a much slower rate.

The point by which the behaviour of the increase changes, i.e., the transition point between the first and the second stage is determined by use of maximum likelihood estimation (MLE) as first introduced by Zhu and Ghodsi (2006). The details are described in Honarkhah (2011) and here we shall only say that the MLE yields the formulation of a profile log-likelihood function of which the optimal pattern size is a maximiser.

Figure 4.4 shows an example of how the optimal pattern size can be deter-



(a) Training image **(b)** Mean Entropy **(c)** Profile Log-Likelihood

Figure 4.4: Example of determining the optimal pattern size for the 250 pixels by 250 pixels training image (TI) shown to the left. Let Δ_x and Δ_y be the size of the patterns in the x and the y direction, respectively. As the image is only two dimensional the pattern size in the z direction is $\Delta_z = 1$. The neighbourhood function is chosen such that patterns become quadratic, i.e. $\Delta_x = \Delta_y = \Delta$, and takes the values $3, 5, \ldots, 29$. The mean entropy of the patterns in the image as a function of Δ is shown in 4.4b. The optimal pattern size is the one that maximises the profile log-likelihood shown in 4.4c. That is seen to happen for $\Delta = 15$, which means the optimal choice is patterns that consist of $15 \times 15 = 225$ pixels.

mined for a given two dimensional training image. The approach assumes quadratic patterns i.e., $\Delta_x = \Delta_y$, and since the training image is only two dimensional $\Delta_z = 1$. Mean entropy as a function of pattern size can be seen in Figure 4.4b and the profile log-likelihood in Figure 4.4c. It is seen that the optimal pattern size is $\Delta_x = \Delta_y = 8$ which yields patterns consisting of $15 \times 15 = 225$ pixels. This seems reasonable as the channels in the training image are approximately 10 pixels wide and the patterns are therefore able to capture the width as well as the direction of the channels.

However, as most training images describe structures with different varia-

tion in different directions it would be interesting to expand this technique to hyper-rectangular patterns.

4.4.2 Simulation of Neighbouring Images

A perturbed image is generated by erasing the values of all voxels in an area, $\mathcal{D}_k$, around voxel k. The voxel value, z_l, for all voxels $l \in \mathcal{D}_k$ is then re-simulated using sequential simulation conditioned upon the values of the remaining voxels $l \notin \mathcal{D}_k$, as well as the already simulated values of voxels in $\mathcal{D}_k$.

In case some voxels should satisfy hard data constraints this should, of course, be taken into consideration. Their values are then not allowed to be erased and re-simulated but should instead be kept and used to condition upon for re-simulation of other voxel values. These voxels are therefore never contained in $\mathcal{D}_k$ for any k.

In order for the method to be computationally feasible the frequency distribution of the new image, $\overline{\mathcal{Z}}$, should never be computed from scratch. Instead the implementation must take advantage of the already known frequency distribution of the image $\mathcal{Z}$.

This approach is similar to the perturbation method for other multiple-point algorithms such as the Sequential Gibbs Sampler (Hansen et al, 2012).

4.4.3 Partial Frequency Distribution

The definition of the dissimilarity function from Equation (4.4) has one significant advantage. As previously discussed, the frequency distributions are expected to be sparse implying a lot of the terms in the dissimilarity function from Equation (4.4) will be zero. Yet the dissimilarity function can be simplified further.

It will be shown that the dissimilarity function value of a frequency distribution, $c(\boldsymbol{\pi})$, given the frequency distribution of a training image, $\boldsymbol{\pi}^{\mathrm{TI}}$, can be computed using only entries of $\boldsymbol{\pi}$ where the corresponding elements of $\boldsymbol{\pi}^{\mathrm{TI}}$ are positive, i.e., those indices i where $\pi_i^{\mathrm{TI}} > 0$. In other words, to compute the dissimilarity function value of an image we need only to know the counts of patterns in the image that also appear in the training image.

Computationally, this is a great advantage as we can disregard the patterns in our solution image that do not appear in the training image and we need not compute nor store the entire frequency distribution of our image.

The expressions of the counts for the underlying distribution, defined by Equation (4.5) and Equation (4.6), are inserted in the expression for the dissimilarity function:

$$\begin{aligned} c(\boldsymbol{\pi}) &= \sum_{i\in\mathcal{I}} \frac{\left(\pi_i^{\mathrm{TI}} - \epsilon_i^{\mathrm{TI}}\right)^2}{\epsilon_i^{\mathrm{TI}}} + \sum_{i\in\mathcal{I}} \frac{\left(\pi_i - \epsilon_i\right)^2}{\epsilon_i} \\ &= \sum_{i\in\mathcal{I}} \frac{\left(\sqrt{\frac{n^{\mathcal{Z}}}{n^{\mathrm{TI}}}}\,\pi_i^{\mathrm{TI}} - \sqrt{\frac{n^{\mathrm{TI}}}{n^{\mathcal{Z}}}}\,\pi_i\right)^2}{\pi_i^{\mathrm{TI}} + \pi_i}. \end{aligned} \tag{4.8}$$

This leads to the introduction of the following partition of the set $\mathcal{I}$ from Equation (4.3):

$$\begin{aligned} \mathcal{I}_1 &= \left\{i \in \mathcal{I} :\ \pi_i^{\mathrm{TI}} > 0\right\}, \\ \mathcal{I}_2 &= \left\{i \in \mathcal{I} :\ \pi_i^{\mathrm{TI}} = 0\right\}. \end{aligned}$$

The dissimilarity function from Equation (4.8) can then be written as:

$$\begin{aligned} c(\boldsymbol{\pi}) &= \sum_{i\in\mathcal{I}_1} \frac{\left(\sqrt{\frac{n^{\mathcal{Z}}}{n^{\mathrm{TI}}}}\,\pi_i^{\mathrm{TI}} - \sqrt{\frac{n^{\mathrm{TI}}}{n^{\mathcal{Z}}}}\,\pi_i\right)^2}{\pi_i^{\mathrm{TI}} + \pi_i} + \frac{n^{\mathrm{TI}}}{n^{\mathcal{Z}}} \sum_{i\in\mathcal{I}_2} \pi_i \\ &= \sum_{i\in\mathcal{I}_1} \frac{\left(\sqrt{\frac{n^{\mathcal{Z}}}{n^{\mathrm{TI}}}}\,\pi_i^{\mathrm{TI}} - \sqrt{\frac{n^{\mathrm{TI}}}{n^{\mathcal{Z}}}}\,\pi_i\right)^2}{\pi_i^{\mathrm{TI}} + \pi_i} + \frac{n^{\mathrm{TI}}}{n^{\mathcal{Z}}} \left(n^{\mathcal{Z}} - \sum_{i\in\mathcal{I}_1} \pi_i\right), \end{aligned} \tag{4.9}$$

using that $\sum_{i \in \mathcal{I}} \pi_i = n^{\mathcal{Z}}$ and that $\pi_i = 0$ for $i \notin \mathcal{I}$.

4.4.4 Non-Inner Voxels

The definition of the frequency distribution of an image assigns to each inner voxel a pattern. Recall that inner voxels have neighbourhoods of size n and voxels that are not inner voxels have smaller neighbourhoods. Due to identical geometrical shape of neighbourhoods, inner voxels are part of n patterns plus the one they are the centre of, and voxels that are not inner voxels are part of less than n patterns.

This means a voxel contributes differently to the frequency distribution depending upon if it is an inner voxel or not. This does not influence which image has the shortest distance to a training image. However, when comparing the distance between several images and a training image, it does influence in an undesirable way which image is closer to the training image. Inner voxels are, so to speak, assigned a higher weight than non-inner voxels and when iteratively minimising the distance to a training image they will be given higher weight. The optimisation therefore prioritises that inner voxels are part of patterns found in the training image; less so, that non-inner voxels are.

Regardless of choice of iterative solution method, the optimisation will focus on inner voxels rather than non-inner voxels. This is highly undesirable as it can cause artefacts on the boundaries of the computed image.

To avoid artefacts caused by unequal weighting of the voxels, we modify the frequency distribution of images that are not training images. Instead of containing one count per inner voxel, all voxels now contribute with one count. Contributions of inner voxels are determined as before, namely based on the pattern of which they are the centre. The contribution of a non-inner voxel is determined by assigning it imaginary neighbouring voxels such that its neighbourhood becomes the same geometrical shape as the neighbourhood of a inner voxel, and the non-inner voxel is then considered the centre of a pattern.

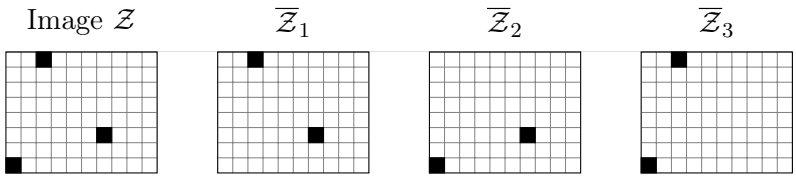


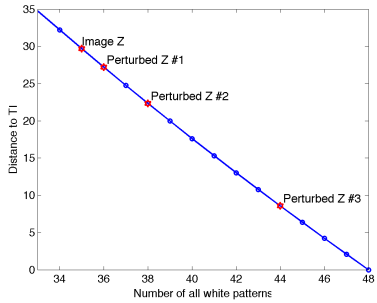
Figure 4.5: The figure shows an image $\mathcal{Z}$, and three perturbed images $\overline{\mathcal{Z}}_1$, $\overline{\mathcal{Z}}_2$ and $\overline{\mathcal{Z}}_3$, which are created by changing one pixel value of the original image $\mathcal{Z}$. Assuming an all white training image each of the perturbed images should have a shorter distance to the training image than the original image.

This raises the question of what values should be assigned to the imaginary neighbouring voxels. However, instead of assigning the imaginary neighbouring voxels an actual value $0, 1, \ldots, v-1$ such that a pattern value can be computed and the corresponding bin in the frequency distribution increased by 1, we look into what values can be assigned to create patterns found in the training image. A non-inner voxel will then contribute to the modified frequency distribution with weights that are proportional to the conditional probability of the value of the imaginary voxels given the known values of the neighbouring voxels.

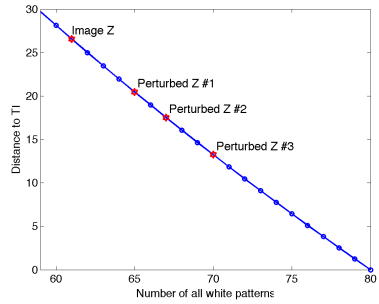
Figure 4.5 and Figure 4.6 illustrate the problem with different weighing of voxels depending on their neighbourhood size and how the problem is partly overcome using the modified definition of the frequency distribution.

Figure 4.5 shows a binary image, $\mathcal{Z}$, and three perturbed images each created by changing the value of one of the black pixels in $\mathcal{Z}$. We now assume an all white training image with $n^{\mathrm{TI}} = 100$ and we choose the neighbourhood function from Equation (4.2). The three perturbed images are each more similar to the training image than the original image, as they all have two instead of three black pixels. More importantly, they are all equally similar to the training image as they have the same number of black pixels.

Let us now have a look at how this similarity is reflected by the dissimilarity function from Equation (4.9). Figure 4.6 shows the distance to the training



(a) Distances computed using the original definition of the frequency distribution



(b) Distances computed using the expanded definition of the frequency distribution

Figure 4.6: Distance from the four different images in Figure 4.5 to an all white training image.

image as a function of the number of white patterns in the image. The total count of patterns in the frequency distributions are 48 and 80, respectively, and the distance 0 is therefore achieved by all white images, i.e., images with respectively 48 and 80 white patterns.

Using the original definition of the frequency distribution, where only inner voxels contribute with their patterns, the dissimilarity value for each of the four images with respect to the training image are as shown in Figure 4.6a. As expected, the dissimilarity is greatest for the image $\mathcal{Z}$. The dissimilarity values of the perturbed images show exactly the point of voxels having different assigned weights. The three black pixels in image $\mathcal{Z}$ have three, five and eight neighbours and they are part of one, three and nine patterns respectively. Changing the value of the pixel with only three neighbours causes the smallest change in the frequency distribution, as it is part of only one pattern. It will therefore cause the smallest improvement in the dissimilarity value. The pixel with five neighbours is part of three patterns and will therefore cause the second smallest improvement in the dissimilarity

value. Finally, the pixel with eight neighbours is part of nine patterns and therefore causes the biggest improvement in dissimilarity.

We notice several things. First, the perturbed image $\overline{\mathcal{Z}}_1$ has almost the same dissimilarity value as the original image. Despite the perturbed image being clearly more similar to the training image than the original, the change in dissimilarity value does not significantly reflect this. Second, the dissimilarity values of the three perturbed images are very different. So although we consider the three perturbed images to be equally similar to the training image, as they each have two black pixels, the dissimilarity function assigns them very different values; the dissimilarity value of $\overline{\mathcal{Z}}_3$ is approximately one third of the dissimilarity value of $\overline{\mathcal{Z}}_1$.

Figure 4.6b shows the dissimilarity values of the images using the modified frequency distribution where all voxels of an image contribute with a total of one count. The ordering of the images by their dissimilarity value is the same as before. This was also expected, as the expanded frequency distribution still assigns different weights to non-inner voxels and to inner voxels. However, this time the three changed pixels are part of four, six, and nine patterns, respectively. The spread of their dissimilarity values is therefore less than before. Now the dissimilarity of $\overline{\mathcal{Z}}_1$ also shows a clear improvement compared to the original image.

Had we modified the frequency distribution even further, such that the imaginary voxels were also contributing with a total count of 1, we would achieve exactly the same dissimilarity value of each of the three perturbed images, as all voxels of an image would then be weighted equally. However, this would complicate the computation of the frequency distribution quite a bit, and results show that the approach we use is sufficient to avoid artefacts.

4.4.5 Skipping Patterns

In the field of multiple-point statistics other methods that involve learning from a training image have dealt with ways of avoiding memory problems

for large test cases. The idea of reading fewer than all patterns in a training image and introducing a spatial lag by skipping patterns was first introduced by Arpat (2005).

The concept consists of only using patterns centred in every kth voxel in a row, column or layer of an image. This will result in the set of inner voxels, that, without skipping is in the order of magnitude as the number of voxels, N, being only of the size approximately $\frac{N}{k^{\text{dim}}}$, where $\text{dim} \in \{2, 3\}$ is the dimension of the image.

For methods that require the construction of a data base of patterns, skipping has shown to be useful, and even necessary, for large test cases. Recent work (Honarkhah, 2011) discusses another approach to assemble a smaller set of patterns that represents the total data base. Instead of simply skipping patterns and thereby choosing patterns based solely on their location, a method based on principle component analysis of the patterns is proposed.

So far, the current implementation of the FM method utilising tree structures has not encountered any memory issues even for large three dimensional test cases with three or four categories of voxel values. Until that happens, skipping patterns is deemed unnecessary.

4.4.6 Clustering of Patterns

The maximum number of different patterns, v^{n+1}, can quickly become astronomical for large test cases with many categories requiring large patterns. The number of different patterns present is, naturally, limited by the number of inner voxels, which is significantly smaller than v^{n+1}.

In fact, the number of patterns present in an image is restricted further as training images are chosen such that they describe a certain structure. This structure is also sought to be described in the solutions. The structure is created by repetition of patterns, and the frequency distributions will reveal this repetition by having multiple counts of the same pattern. This means,

the number of patterns with non-zero frequency is much smaller than v^{n+1} resulting in the frequency distributions becoming extremely sparse. For bigger test cases, with millions of parameters, patterns consisting of hundreds of voxels and multiple categories, this behaviour needs to be investigated further.

The dimension of the images, if they are two- or three-dimensional, is not important to the FM method. The complexity of the method is given by the maximal size of neighbourhoods, n. The increase in n as a result of going from two- to three-dimensional images is therefore more important than the actual increase in physical dimensions. In fact, when it comes to assigning pattern values a neighbourhood is, regardless of its physical dimension, considered one dimensional where the ordering of the voxels is the important aspect.

The number of categories of voxel values, v, also does not influence the running time per iteration. As with the number of neighbours, n, it only influences the number of different possible patterns, v^{n+1}, and thereby influences the sparsity of the frequency distribution of the training image. The higher v is, the sparser is the frequency distribution. It is expected that the sparsity of the frequency distribution affects the level of difficulty of the combinatorial optimisation problem as it affects the size of the model space and thereby the number of iterations needed to converge.

To avoid large, sparse frequency distributions and the complications that follow we let ourselves get inspired by recent work by Honarkhah (2011). Here patterns are organised in clusters such that patterns within the same cluster are similar. Clustering methods are widely used in machine learning and data mining (Fayyad et al, 1996), data compression and vector quantization (Gersho and Gray, 1992), and pattern recognition and pattern classification (Duda and Hart, 1973).

Figure 4.7 shows an example of a cluster. The cluster analysis is done based on the training image in Figure 4.4a, rescaled to 100×100 pixels. For this illustration a pattern is chosen to consist of 7×7 pixels. A cluster is represented by its prototype, which is an average of patterns used when

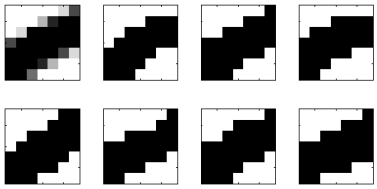


Figure 4.7: Example of a cluster of 7×7 patterns, which all have a channel structure going diagonally through the pattern from the lower left to the upper right corner. The prototype (upper left corner) is the average of the seven patterns used to define the cluster.

performing the clustering. The prototype is shown in the upper left corner of Figure 4.7. As it is an average it has gray-scale values. The remaining seven images are the patterns used to construct the cluster. These are examples of patterns that could belong to the cluster. Notice how they, despite being different if compared pixel by pixel, all show the same channel structure going diagonally through the pattern.

These patterns all represent the same structures in an image, and it therefore seems logically to not distinguish between them when characterising the structures of an image. We do this by combining the counts of their appearances, which means the frequency distribution is clustered. The higher number of voxels in patterns, the more patterns will appear to be describing the same structure despite being different when compared voxel by voxel.

In our case clustering can be applied such that if the number of clusters is chosen appropriately, patterns with different structures will be grouped in the same cluster. And equally important, patterns with different structures will be grouped in different clusters. This could be used to compress the frequency distributions such that each bin, instead of representing a specific pattern, represents a cluster of patterns with similar structures. Each bin is characterised by a prototype.

We will refer to a frequency distribution of clusters of patterns as a clus-

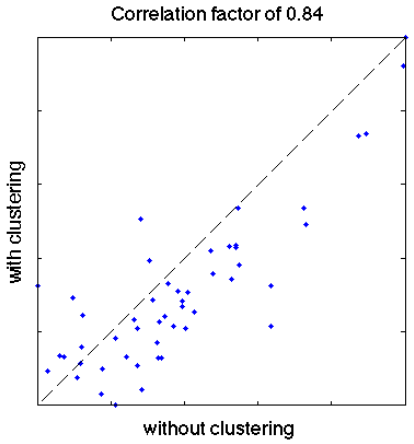


Figure 4.8: Analysis of the similarity values computed between a training image and 50 DISTPAT realisations of the training image. The cross plot shows the correlation between dissimilarity values computed with and without clustering of the frequency distributions.

tered frequency distribution. A generic approach to determining the optimal number of clusters is provided in Honarkhah (2011).

Figure 4.8 shows the effect on the computed similarity values when clustering the frequency distribution using the training image from Figure 4.4a and approximately 150 clusters. We have computed the dissimilarity value of 50 DISTPAT realisations from the training image.

The correlation plot shows the relationship between the dissimilarity of the images using the frequency distributions versus the clustered frequency distributions. The two different measures of distances are strongly correlated with a correlation factor of 0.84. This indicates that the distances between clustered histograms can be used as a mean to avoid the sparse frequency distributions and the complications they impose on large test cases.

The current implementation of the FM method does not include clustered

frequency distributions.

4.4.7 Continuous Voxel Values

A disadvantage of the simulated annealing scheme is the high number of iterations it requires to converge to a solution of acceptable quality. Each iteration consists of three elements: 1) The generation of a perturbed image. 2) The evaluation of the dissimilarity function of the perturbed image. 3) The evaluation of the data misfit of the perturbed image. Depending on the inverse problem at hand, the bottleneck is likely to be the forward computation in the evaluation of the data misfit. The forward computation of a highly non-linear problem, such as history matching of production data, requires flow modelling and is therefore computationally expensive.

In the field of optimisation, a common approach used to solve combinatorial problems or integer programming problems is relaxation. For a generel discussion of combinatorial optimisation problems and possible solution methods including simulated annealing and relaxation, see Wolsey (1998).

For an inverse problem with discrete, and not just categorical voxel values, a relaxation of the voxel values to take on continuous values in the interval $[0, m]$ and a suitable redefinition of the frequency distributions would make the optimisation problem of the FM method continuous. This implies that instead of heuristics like simulated annealing, gradient methods can be applied to solve it. These typically will require a lower number of iterations as knowledge of the gradient allows for a more efficient search through the model space.

The work presented in Appendix D illustrates that this is an idea worth pursuing, especially for large-scale, non-linear problems.

4.5 Examples in Papers

The frequency matching method was first published in 2012, in *Mathematical Geosciences*, found in Appendix C. It introduces the method and discusses certain non-trivial details with focus on implementation and computational feasibility. Furthermore, the paper shows an example of applying the FM method on a seismic tomography problem, which is a linear inverse problem.

The recently published paper in *Mathematical Geosciences* is based on the preliminary work that was presented at the *15th IAMG Conference, 2011*, and an extended abstract can be seen in Appendix B. The work does not include an inverse problem in the sense of a data fitting problem, but it demonstrates how maximising the a priori pdf used in the FM method with respect to the model parameters generates a realisation of the training image.

Another publication describing the FM method is from the proceedings of *74th EAGE Conference, 2012*, and can be seen in Appendix D. Here the FM method is applied when solving the non-linear inverse problem history matching of production data.

Appendix F holds a technical report describing the current implementation of the FM method. It discusses details of the implementation that are interesting for the computational feasibility of the method. This paper also has an example of how to use the FM method. The implementation was used in the published work on the FM method seen in Appendix C.

4.6 Related Research

The frequency matching method plays a role also in related research areas. Appendix E contains an extended abstract, where aspects of the FM method have been used to improve the multiple-point statistics of realisations computed using Sequential Gibbs Sampling (Hansen et al, 2012). This work

formulates a joint a priori pdf that describes the multiple-point statistics learned from a training image. The a priori pdf is partly based on a FM-inspired dissimilarity function and partly based on the SNESIM algorithm (Strebelle, 2002). The paper considers a synthetic crosshole travel time tomography problem. It demonstrates how the reproducibility of the patterns in the training image is improved in the realisations of the a posteriori pdf, when the joint prior is used instead of just the SNESIM prior.

4.7 Summary

This chapter introduced the frequency matching method. It defined the terminology to derive it and defined concepts such as neighbourhoods and patterns. That lead to the definition of frequency distributions and the discussion of how they can be used to represent multiple-point statistics of an image. The idea of measuring similarity between images by the distances between frequency distributions was introduced. This lead to a closed form expression of the a priori pdf, which enabled us to compute the maximum a posteriori model of an inverse problem.

Furthermore, this chapter discussed many details of great importance when applying the frequency matching method. It addressed computational problems that might occur for large cases. Solutions to these problems were suggested although not all are implemented in the current version of the Fortran code.

A deterministic approach to determine the optimal pattern size used by other MPS algorithms was presented and we discussed how it could also be used for the FM method.

We explained our choice of perturbation algorithm for generating new images that can be proposed to the simulated annealing scheme. Here we have chosen to use a variation of the well-established SNESIM algorithm.

We have discussed how to deal with the boundaries of the images given by the set of non-inner voxels. We have explained how this is a trade off be-

tween computational complexity and the risk of artefacts on the boundary. We were satisfied with the current implementation that follows what we believe is a close to optimal trade off.

Literature on MPS simulation methods mentions an approach for thinning out images by systematically skipping patterns. We have dismissed this feature while we do not consider it necessary for the FM method. The implementation therefore does not include the feature of pattern skipping and it is, for now, not high up on our priority list.

We then discussed the possibility of clustering patterns in order to avoid sparse frequency distributions and computational difficulties that might follow for larger cases. We illustrated why we expect this to be favourable, and while this feature is also not in the current implementation it is high on the priority list.

Last, we briefly mentioned a topic of on-going research that we find particularly interesting. Namely, the possibility to relax the inverse problem such that the model parameters become continuous variables. This allows computation of the gradient of the objective function which again allows for other solution methods such as gradient-based iterative methods. It is no straight forward task as it opens up a discussion on how to define the frequency distribution of an image with continuous voxel values.

CHAPTER 5

Conclusions

This chapter will summarise the work done during the study by highlighting the conclusions from the summary report and the papers. The overall topic has been probabilistic inverse problems in the geosciences and more specifically the use of geostatistical methods to infer models of the rock properties of an oil reservoir from observed geophysical data. We have approached the subject from a computational point of view where focus has been on the scientific methods themselves as well as computational aspects of their development and implementation.

The most noteworthy achievements are hereby listed.

Development of a method for efficient prediction of rock properties based on seismic attributes. The method has been applied to a test case from the South Arne Field in the Danish part of the North Sea and the results discussed in the papers seen in Appendices A and G. We demonstrate how seismic attributes such as two-way travel time and acous-

tic impedance can be used to guide interpolation of porosity values and thereby successfully predict the porosity levels in between well logs.

The paper demonstrates how orthogonal transformation techniques, well-known from the field of data analysis, can be used to improve the efficiency of the prediction. The seismic attributes are transformed using principal component analysis (Appendix A) or partial least squares transformation (Appendix G). The choice of transformation allows for the seismic data to be approximated in a lower-dimensional subspace. This reduces the computational complexity of the problem, as covariance models are now inferred in a lower-dimensional interpolation space. For the latter choice of transformation the test case achieves good results when interpolation is performed in a three-dimensional subspace instead of the original eight-dimensional space.

Formulation of a closed form expression for prior knowledge. We have formulated a closed form expression for an a priori probability density function describing the multiple-point statistics of a model of a rock property relative to the multiple-point statistics of a training image. A normalisation constant remains unknown, however, the constant itself is irrelevant as the expression is used to quantify the relative probability of a model compared to other models.

The conference paper in Appendix B shows how models with multiple-point statistics similar to the multiple-point statistics of a training image can be generated as models maximising the expression of this a priori probability density function. The maximisation is performed by a stochastic method which allows for multiple models to be generated. These all share the same multiple-point statistics as the training image.

Development of the FM method. We have developed the frequency matching method which computes the maximum a posteriori model of an inverse problem. The inverse problem consists of a data fit via a possibly

complex non-linear forward mapping and a priori knowledge in the form of multiple-point statistics extracted from a training image. The previously defined expression of an a priori probability density function is applied and the FM model is defined as the model maximising the resulting a posteriori probability density function.

The method is applied to synthetic test problems within crosshole tomography (Appendices C and F) and history matching of production data (Appendix D). Despite the test cases being simplified and the models only idealised images of the rock properties of the subsurface, the test cases serve as proof of concept. They illustrate that the FM method is capable of producing models as solutions to inverse problems and which holds structures similar to the structures of the training images.

General Implementation of the FM method Last, we have made a general Fortran implementation of the Frequency Matching method. It is publicly available via GitHub. It can be used to solve an arbitrary linear inverse problem with multiple point statistics extracted from a training image as a priori information. Appendix F holds a paper documenting the implementation and discusses computational aspects of the FM method and how these are best implemented. The Fortran implementation is the one used in the original paper on the FM method (Appendix C).

Bibliography

Alabert FG (1987a) The practice of fast conditional simulations through the lu decomposition of the covariance matrix. Mathematical Geology 19:369–386, DOI 10.1007/BF00897191, URL http://dx.doi.org/10.1007/BF00897191

Alabert FG (1987b) Stochastic imaging of spatial distributions using hard and soft information. M.sc. thesis, Stanford University

Anderson TW (1984) An introduction to multivariate statistical analysis. John Wiley

Arpat GB (2005) Sequential simulation with patterns. Ph.d. dissertation, Stanford University

Arpat GB, Caers J (2007) Conditional simulation with patterns. Mathematical geology 39(2):177–203

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

Borgman L, Taheri M, Hagan R (1984) Three-dimensional frequency-domain simulations of geological variables. Geostatistics for natural resources characterization, Part 1:517–541

Caers J, Journel AG (1998) Stochastic reservoir simulation using neural networks trained on outcrop data. In: SPE Annual Technical Conference and Exhibition

Caers J, Ma X (2002) Modeling conditional distributions of facies from seismic using neural nets. Mathematical Geology 34(2):143–167

Christakos G (1984) On the problem of permissible covariance and variogram models. Water Resources Research 20(2):251–265, DOI 10.1029/WR020i002p00251, URL http://dx.doi.org/10.1029/WR020i002p00251

Cressie N (1990) The origins of kriging. Mathematical Geology 22:239–252, DOI 10.1007/BF00889887, URL http://dx.doi.org/10.1007/BF00889887

Daly C (2005) Higher order models using entropy, markov random fields and sequential simulation. Geostatistics Banff 2004 pp 215–224

Daly C, Knudby C (2007) Multipoint statistics in reservoir modelling and in computer vision. Petroleum geostatistics p A32

David M (1977) Geostatistical ore reserve estimation. Developments in geomathematics, Elsevier Scientific Pub. Co., URL http://books.google.dk/books?id=nGcZAQAAIAAJ

Davis BM (1987) Uses and abuses of cross-validation in geostatistical. Mathematical Geology 19(3):241–248

Deutsch C, Lewis R (1992) Advances in the practical implementation of indicator geostatistics. In: Proceedings of the 23rd International APCOM Symposium, pp 133–148

Deutsch CV, Cockerham PW (1994) Practical considerations in the application of simulated annealing to stochastic simulation. Mathematical Geology 26(1):67–82

Deutsch CV, Journel AG (1998) GSLIB, Geostatistical Software Library and User's Guide, 2nd edn. Applied Geostatistics, Oxford University Press

Diggle PJ, Ribeiro Jr PJ, Christensen OF (2003) An introduction to model-based geostatistics, Springer, pp 43–86

Doyen PM (1988) Porosity from seismic data - A geostatistical approach. Geophysics 53(10):1263–1275

Duda RO, Hart PE (1973) Pattern Classification and Scene Analysis. Wiley-Interscience publication, John Wiley & Sons, URL http://books.google.dk/books?id=POMGRAAACAAJ

Fang J, Wang P (1997) Random field generation using simulated annealing vs. fractal-based stochastic interpolation. Mathematical geology 29(6):849–858

Fayyad UM, Piatetsky-Shapiro G, Smyth P, Uthurusamy R (eds) (1996) Advances in Knowledge Discovery and Data Mining. AAAI/MIT Press

Geman S, Geman D (1984) Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. IEEE Trans Pattern Anal Machine Intell 6:721–741

Gersho A, Gray AGRM (1992) Vector Quantization and Signal Compression. The Kluwer International Series in Engineering and Computer Science, Springer-Verlag GmbH, URL http://books.google.dk/books?id=DwcDm6xgItUC

Gilks WR, Wild P (1992) Adaptive rejection sampling for gibbs sampling. Journal of the Royal Statistical Society Series C (Applied Statistics) 41(2):pp. 337–348, URL http://www.jstor.org/stable/2347565

Golub G, Reinsch C (1970) Singular value decomposition and least squares solutions. Numerische Mathematik 14:403–420, DOI 10.1007/BF02163027, URL http://dx.doi.org/10.1007/BF02163027

Goovaerts P (1996) Stochastic simulation of categorical variables using a classification algorithm and simulated annealing. Mathematical Geology 28(7):909–921

Goovaerts P (1997) Geostatistics for natural resources evalutaion. Applied Geostatistics Series, Oxford University Press

Guardiano F, Srivastava RM (1993) Multivariate geostatistics: Beyond bivariate moments. Geostat-Troia 1:133–144

Hampson D, Schuelke J, Quirein J (2001) Use of multiattribute transforms to predict log properties from seismic data. Geophysics 66(1):220–236

Hansen TM, Mosegaard K, Pedersen-Tatalovic R, Uldall A, Jacobsen NJ (2008) Attribute guided well log interpolation - applied to low frequency impedance estimation. Geophysics 73(6):R83–R95

Hansen TM, Cordua KS, Mosegaard K (2012) Inverse problems with non-trivial priors - efficient solution through sequential Gibbs sampling. Computational Geosciences 16(3):593–611, DOI 10.1007/s10596-011-9271-1

Hastie T, Tibshirani R, Friedman J (2009) The elements of statistical learning, 2nd edn. Springer Series in Statistics

Herrara VM, Russel B, Flores A (2006) Neural networks in reservoir characterization. The Leading Edge 25(4):402–411

Honarkhah M (2011) Stochastic simulation of patterns using distance-based pattern modeling. Ph.d. dissertation, Stanford University

Hong S, Ortiz J, Deutsch C (2008) Multivariate density estimation as an alternative to probabilistic combination schemes for data integration. In: Geostats 2008-Proceedings of the Eighth International Geostatistics Congress, Gecamin Ltda., Santiago, Chile, vol 1, pp 197–206

Hotelling H (1933) Analysis of a complex of statistical variables into principal components. Journal of Educational Psychology 24:417–441, DOI 10.1037/h0071325

Isaaks EH, Srivastava RM (1989) An Introduction to Applied Geostatistics. Oxford University Press

Journel A (1974) Geostatistics for conditional simulation of ore bodies. Economic Geology 69(5):673–687

Journel A, Rossi M (1989) When do we need a trend model in kriging? Mathematical Geology 21(7):715–739

Journel AG, Huijbregts CJ (1978) Mining Geostatistics. Academic Press

Journel AG, Zhang T (2006) The Necessity of a Multiple-Point Prior Model. Mathematical Geology 38(5):591–610

Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. Science 220(4598):671–680

Kitanidis PK, Lane RW (1985) Maximum likelihood parameter estimation of hydrologic spatial processes by the gauss-newton method. Journal of Hydrology 79(1/2):53–71

Krige DG (1951) A statistical approach to some basic mine valuation problems on the witwatersrand. Journal of Chemical Metall and Mineralogy Sociecty South Africa 52:119–139

Lange K, Frydendall J, Cordua KS, Hansen TM, Melnikova Y, Mosegaard K (2012) A frequency matching method: Solving inverse problems by use of geologically realistic prior information. Mathematical Geosciences pp 1–21, URL http://dx.doi.org/10.1007/s11004-012-9417-2, 10.1007/s11004-012-9417-2

Lophaven S, Nielsen H, Søndergaard J (2002) Aspects of the Matlab toolbox DACE. Informatics and Mathematical Modelling, Technical University of Denmark, DTU

Lyster S, Deutsch CV (2008) Mps simulation in a gibbs sampler algorithm. In: Geostats 2008-Proceedings of the Eighth International Geostatistics Congress, vol 1, pp 79–88

Mackay DJC (2003) Information Theory, Inference & Learning Algorithms, 1st edn. Cambridge University Press

Mantoglou A (1987) Digital simulation of multivariate two-and three-dimensional stochastic processes with a spectral turning bands method. Mathematical Geology 19(2):129–149

Mariethoz G, Renard P (2010) Reconstruction of incomplete data sets or images using direct sampling. Mathematical Geosciences 42(3):245–268

Mariethoz G, Renard P, Straubhaar J (2010) The direct sampling method to perform multiple-point geostatistical simulations. Water Resources Research 46(11):W11,536

Matheron G (1973) The intrinsic random functions and their applications. Advances in applied probability pp 439–468

Metropolis N, Ulam S (1949) The Monte Carlo Method. Journal of the American Statistical Association 44(247):335–341

Metropolis N, Rosenbluth M, Rosenbluth A, Teller A, Teller E (1953) Equation of state calculations by fast computing machines. J Chem Phys 21:1087–1092

Mosegaard K (2011) Quest for consistency, symmetry and simplicity - the legacy of albert tarantola. Geophysics 76(5):W51–W61, copyright 2011 Society of Exploration Geophysicists.

Mosegaard K, Sambridge M (2002) Monte Carlo analysis of inverse problems. Inverse Problems 18(3):29–54

von Neumann J (1951) Various techniques used in connection with random digits. National Bureau of Standards Applied Mathematics Series 12:36–38

Nocedal J, Wright SJ (2000) Numerical Optimization. Springer

Ortiz JM, Deutsch CV (2004) Indicator simulation accounting for multiple-point statistics. Mathematical geology 36(5):545–565

Ortiz JM, Emery X (2005) Integrating multiple-point statistics into sequential simulation algorithms. Geostatistics Banff 2004 pp 969–978

Pardo-Igúzquiza E (1997) Mlreml: a computer program for the inference of spatial covariance parameters by maximum likelihood and restricted maximum likelihood. Computers and Geosciences 23(2):153–162

Pardo-Igúzquiza E (1998) Maximum likelihood estimation of spatial covariance parameters. Mathematical Geology 30(1):95–108

Parra A, Ortiz JM (2009) Conditional multiple-point simulation with a texture synthesis algorithm. In: Proceedings of the 12th IAMG (International Association of Mathematical Geosciences) Conference, Stanford University, CA, USA, vol 5

Peredo O, Ortiz JM (2010) Parallel implementation of simulated annealing to reproduce multiple-point statistics. Comput Geosci 37:1110–1121

Pramanik AG, Singh V, Vig R, Srivastava K, Tiwary DN (2004) Estimation of effective porosity using geostatistics and multiattribute transforms: A case study. Geophysics 69(2):352–372, DOI doi:10.1190/1.1707054

Russell B, Hampson D, Todorov T, Lines L (2002) Combining geostatistics and multi-attribute transforms: a channel sand case study, Blackfoot oilfield (Alberta). Journal of Petroleum Geology 21(1):97–117

Sambridge M (1999) Geophysical inversion with a neighbourhood algorithm-I. Searching a parameter space: Geophysical Journal International 138:479–494

Samper FJ, Neuman S (1989a) Estimation of spatial covariance structures by adjoint state maximum likelihood cross-validation: 1. theory. Water Resources Research 35(3):351–362

Samper FJ, Neuman S (1989b) Estimation of spatial covariance structures by adjoint state maximum likelihood cross-validation: 2. synthetic experiments. Water Resources Research 35(3):363–371

Samper FJ, Neuman S (1989c) Estimation of spatial covariance structures by adjoint state maximum likelihood cross-validation: 3. application to hydrochemical and isotopic data. Water Resources Research 35(3):373–384

Sheskin D (2004) Handbook of Parametric and Nonparametric Statistical Procedures, Chapman & Hal/CRC, pp 493–500

Strebelle S (2000) Sequential simulation drawing structures from training images. Ph.d. dissertation, Stanford University

Strebelle S (2002) Conditional simulation of complex geological structures using multiple-point statistics. Mathematical Geology 34:1–21

Tarantola A (2005) Inverse Problem Theory and Methods for Model Parameter Estimation. SIAM

Tarantola A, Valette B (1982a) Generalized nonlinear inverse problems solved using the least squares criterion. Rev Geophys Space Phys 20(2):219–232

Tarantola A, Valette B (1982b) Inverse problems = quest for information. J Geophys 50:159–170

Tjelmeland H (1996) Stochastic models in reservoir characterization and markov random fields for compact objects. Unpublished doctoral dissertation, Norwegian University of Science and Technology

Tjelmeland H, Eidsvik J (2005) Directional metropolis: Hastings updates for posteriors with nonlinear likelihoods. Geostatistics Banff 2004 pp 95–104

Trangmar B, Yost R, Uehara G (1986) Application of geostatistics to spatial studies of soil properties. Advances in Agronomy, vol 38, Academic Press, pp 45 – 94, DOI 10.1016/S0065-2113(08)60673-2, URL http://www.sciencedirect.com/science/article/pii/S0065211308606732

Vestergaard PD, Mosegaard K (1991) Inversion of post-stack seismic data using simulated annealing. Geophysical Prospecting 39:613–624

Vieira SR, Hatfield JL, Nielsen DR, Biggar JW (1983) Geostatistical theory and application to variability of some agronomical properties. Hilgardia 51:25,933

Warrick AW, Myers DE, Nielsen DR (1986) Geostatistical methods applied to soil science. In: Klute A (ed) Methods of soil analysis. Part 1: Physical and mineralogical methods, no. 9 in Agronomy, ASA, ASSA, Inc., Publisher, Madison, Wisconsin, USA, pp 53–82

Wold H (1966) Estimation of principal components and related models by iterative least squares. Multivariate Analysis pp 391–420

Wold S, Ruhe A, Wold H, Dunn III W (1984) The collinearity problem in linear regression. the partial least squares (pls) approach to generalized inverses. SIAM Journal on Scientific and Statistical Computing 5(3):735–743

Wolsey LA (1998) Integer Programming. Wiley Series in Discrete Mathematics and Optimization, John Wiley & Sons, URL http://books.google.dk/books?id=x7RvQgAACAAJ

Wu J (2007) 4d seismic and multiple-point pattern data integration using geostatistics. PhD thesis, Stanford University

Wu J, Zhang T, Journel AG (2008) Fast filtersim simulation with score-based distance. Mathematical Geosciences 40:773–788, DOI 10.1007/s11004-008-9157-5, URL http://dx.doi.org/10.1007/s11004-008-9157-5

Zhang T (2006) Filter-based training pattern classification for spatial pattern simulation. PhD thesis, Stanford University

Zhu M, Ghodsi A (2006) Automatic dimensionality selection from the scree plot via the use of profile likelihood. Computational Statistics & Data Analysis 51(2):918–930

Appendices

APPENDIX A

Paper I

Kriging in High Dimensional Attribute Space using Principal Component Analysis

Authors:
Katrine Lange, Thomas Mejer Hansen, Juan Luis Fernández Martínez, Jan Frydendall and Klaus Mosegaard

Published in:
Proceedings of the *14th Annual Conference of the International Association for Mathematical Geosciences (IAMG 2010)*
Budapest, Hungary
29 August 2010 - 2 September 2010

Kriging in High Dimensional Attribute Space using Principal Component Analysis

Katrine Lange[1], Thomas Mejer Hansen[1], Juan Luis Fernández Martínez[2], Jan Frydendall[1] and Klaus Mosegaard[1]

[1]Technical University of Denmark, Copenhagen, Denmark. Email: katla@imm.dtu.dk
[2]Universidad de Oviedo, Oviedo, Spain

Abstract

Interpolation between measured well log properties is a well-known problem in seismic exploration. Most often 3D seismic data is available from which a large number of seismic attributes can be extracted quantifying various properties of the seismic data. Such attributes have been used to guide interpolation between well log parameters, using for example neural network, linear regression and cokriging. Full cokriging using seismic attributes as secondary data is though not feasible due to the complexity of inferring a full cross-covariance model. A recently proposed method suggests a kriging alternative to full cokriging, where kriging is performed in the attribute space, relying on a distance measure in the attribute space. One limitation is, however, that all attributes are considered uncorrelated, whereas in reality they are not. We propose to use principal component analysis (PCA) to transform the coordinate system in attribute space to a coordinate system given by the principal components. By construction the principal components are uncorrelated, and we can therefore apply kriging in the principal component space. In addition PCA naturally orders the principal components according to their variance contribution, and can therefore be used to select only the most significant principal components for the kriging, allowing easier inference of the covariance model.

Keywords: seismic inversion, seismic attributes, kriging interpolation.

1. INTRODUCTION TO KRIGING IN ATTRIBUTE SPACE

The objective of our work is to produce accurate and reliable estimations of the porosity levels in the subsurface of a reservoir. To do so we exploit knowledge of seismic attributes, such that the estimated porosity at a certain location relies on the subsurface geology rather than just the spatial location.

We propose to use the kriging technique[1] to interpolate between known values of the porosity level. One of the advantages of kriging is that, unlike traditional interpolation methods such as

linear regression and neural networks, it provides not only an estimate of the value itself but also an uncertainty estimate. The interpolation will be based on a distance measure in the attribute space rather than the more traditional distance measure in the physical XYZ space.

Outline of the method:

1) Normal score transformation of the porosity data to meet the assumption of data being samples from a continuous Gaussian distribution. Normalization of the position coordinates in attribute space for computational reasons.
2) PCA transformation of the attributes to ensure that their cross-correlation is zero.
3) Definition of a distance measure in the principal component space and computation of optimal covariance parameters by use of maximum likelihood estimation[4] given the choice of covariance model type.
4) Formulation of the kriging system given the covariance model, and the system is then solved with respect to the kriging weights.
5) Back transformation of the porosity values estimated in normal score space.

Using a subset of the attributes, and/or a subset of the principal components, the dimension of the space in which the interpolation is conducted can be reduced.

2. CASE STUDY: The South Arne Field

Hess Copenhagen provided us with data from the South Arne Field, which is a chalk reservoir situated in the Danish North Sea. The data set contains known values of 8 seismic attributes of nearly 76.000 points in a regular 2D XY-grid. The attributes associated with each point of the grid are the following: spatial coordinates UTM X, UTM Y and UTM Z, the two way travel times to the top and the base of the reservoir, both have been established from the seismic data, the amplitude and the dip at the top has also been extracted as attributes, and finally the acoustic impedance.

There are additionally 213 well site points where both the values of the seismic attributes and also the values of the porosity levels are known. These points are used to validate our results. To do so they are divided into two sets. The first set contains the data used in the interpolation, and the second set acts as a set of blind wells. Porosity levels are estimated at the location of the blind wells using only the data in the first set and the results are then compared to the known true values. The partition of the data set is done in accordance to the method described in[3]. This yields six different subsets of blind data with respectively {106, 53, 21, 178, 99, 144} points of the total 213 observation points as known data and the remaining data will act as blind data.

Figure 1 shows the known values of the porosity levels in the XY domain of the total data set and known value of each of the six data sets that holds blind data. This implies that the data in the second to the seventh subplot are subsets of the data shown in the leftmost subplot.

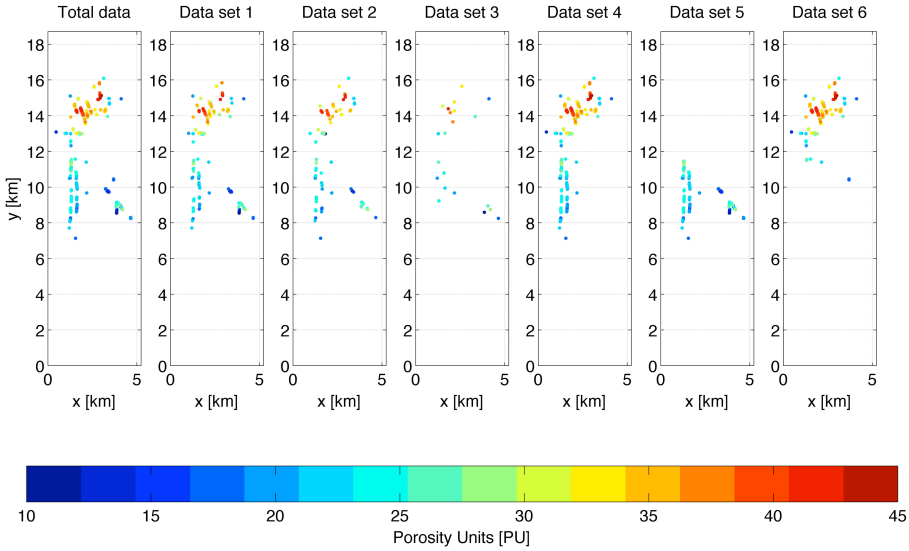


Figure 1: Known values of the porosity levels of the total data set and each of the six data sets containing blind data.

In the following subsection we present and discuss the kriging results. As a measure of performance of our method we will evaluate the root mean squared (RMS) error of the porosity estimates at the blind wells and the uncertainty estimates of the porosity levels estimated for the entire grid for the six different partitions of the known data.

2.1. Parameter choices

Before we evaluate the performance of our method we must discuss the effect of decisions that we have to make during the formulation and solution of the problem. The choices we have are regarding the following aspects:

1) *Partition of the known data to create a blind set.* We have the previously mentioned six different partitions that can also be seen in Figure 1.
2) *Selection of the attributes used for the interpolation.* It is likely that we can reduce the dimension of the transformed attribute space without reducing the quality of our results by only selecting the most relevant of the attributes and simply ignoring the rest.
3) *Number of principal components used.* It is also worthwhile to consider if one can reduce the dimension of the transformed attribute space further by selecting only a subset of the principal components.
4) *Type of covariance model.* We will choose a covariance model for each principal component consisting of two terms. The first term is a nugget effect model and the second term is a model of either of the types spherical, exponential or Gaussian. This term will be of the same type in all directions.

5) *Type of kriging method.* For this extended abstract we will use only kriging with a trend, which in each direction of the principal components is modeled as a polynomial of degree one.

2.2. Attributes vs. principal components

Figure 2 shows the eight different attributes and Figure 3 shows the resulting principal components. It is interesting to see how the principal components, some more than others, resembles single attributes. This can be an inspiration in choosing the trend type of the kriging method.

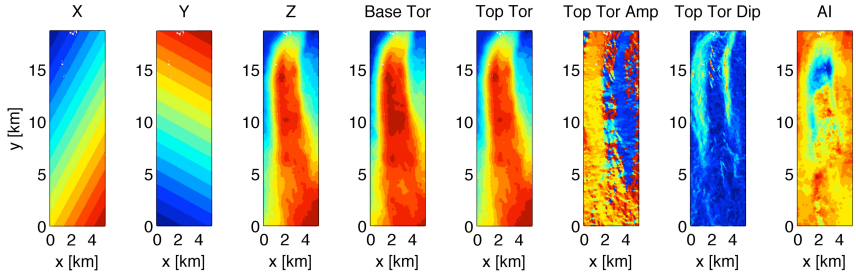


Figure 2: Each of the eight attributes plotted in the physical domain. Notice that the plots have been rotated slightly in order to minimize white space between the subfigures. The attributes do not share a common color scale, but red simply indicate high relative values and blue indicates low relative values.

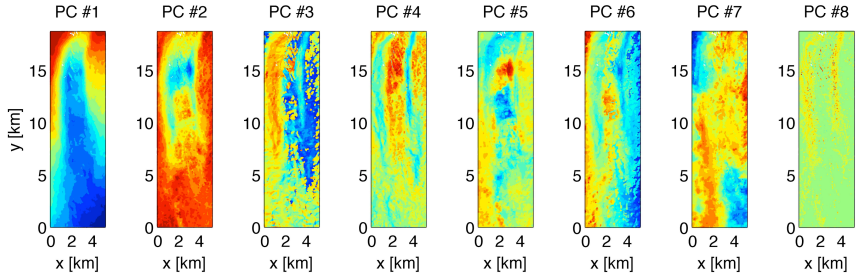


Figure 3: Each of the eight principal components computed based on the attributes shown in Figure 2. As in Figure 2 the principal components do not share a common color scale, but red and blue indicate respectively high and low relative values.

Figure 4 shows the transformation matrix used to transform the attributes of Figure 2 into the principal components seen in Figure 3. To compute the *i*th principal component the attributes has been weighted by the entries of the *i*th column of the transformation matrix. Notice for instance that to compute the third principal component, attribute six has been given a very high positive weight compared to the other attributes. Hence the structure in the third principal component resembles the structure in the top amplitude attribute.

Figure 4: Transformation matrix used to transform the attributes in Figure 2 into the principal components shown

in Figure 3.

2.3. Interpolation using a subset of the attributes

We choose the second term of the covariance model to be a Gaussian model. We now compute the RMS error for six different partitions of the data varying the subset of the attributes used in the interpolation. In all cases the full set of principal components is used. The results can be seen in Table 1. For each of the six data sets containing blind data this table shows the RMS error at the blind wells when using different sets of attributes and the complete set of principal components.

Table 1: RMS error in porosity units of the kriging results when using different subsets of the attributes and all of the principal components.

Data set	X, Y, Z	Impedance	X, Y, Z and impedance	X, Y, Z, amplitude and impedance	All attributes
#1	2.96	4.04	2.91	2.98	3.05
#2	4.22	4.25	3.58	3.82	3.94
#3	5.11	4.80	4.14	4.41	4.51
#4	6.65	4.64	5.52	5.61	5.79
#5	10.2	9.15	9.37	8.91	9.29
#6	13.6	5.82	5.20	5.33	5.90

From the results in Table 1 we see that while the choice of blind data has great influence on the level of the RMS error we experience approximately the same behavior of the RMS error when we change the subset of the attributes used in the interpolation. The second column shows the result when we use only the spatial coordinates while the third column holds the results for when we only use the impedance, which we expect is strongly correlated with the porosity.

For the majority of the data sets the lowest RMS error is achieved by using the spatial coordinates together with the impedance. Adding the amplitude at the top as a fifth attribute seems, in most cases, to increase the error slightly. The same applies for adding the remaining attributes, i.e. for the errors in the last column.
According to the table, using data set #5 results in a relative high RMS error compared to using the other data sets. Figure 1 can explain this. Here we see that the known values of the porosity levels for data set #5 all are relatively low values (all dots are blue or green). This data set is therefore not a representative subset of porosity levels for the entire grid and that complicates the interpolation.

2.4. Reducing the dimension of the principal component space

We now turn to reducing the dimension of the transformed attribute space by reducing the number of principal components included. The principal components are chosen according to their variance contributions.

First we consider the test case from the fourth column of Table 1, i.e. we include four attributes namely the spatial coordinates and the impedance. Table 2 shows the RMS errors when using one to four principal components respectively. Four attributes yields in total four principal components. The table shows a trend, which is that the more principal components we include the lower is the error. Data set #2 is the exception for which three and four principal components yield approximately the same error.

Table 2: RMS error in porosity units of the kriging results when using a subset of the principal components. We have used the attributes X, Y, Z and impedance that result in the last column of Table 2 being a duplicate of the fourth column of Table 1.

Data set	1 out of 4	2 out of 4	3 out of 4	All 4 principal components
#1	4.67	3.56	3.02	2.91
#2	5.58	4.11	3.55	3.58
#3	5.55	4.47	4.28	4.14
#4	5.74	5.46	6.02	5.52
#5	10.3	10.6	9.27	9.37
#6	7.87	6.73	5.86	5.20

In order to derive the general trend we consider all attributes and compute the RMS error when selecting one to all of the eight principal components. These errors can be seen in Figure 5. Notice that data set #5 has been omitted from the figure as its errors are of significantly greater magnitude due to the previously mentioned difficulties regarding the partition of the data.

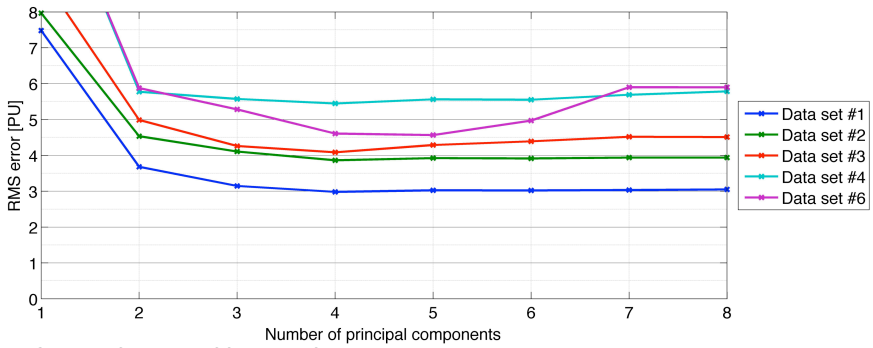


Figure 5: RMS errors for each of five out of the six data sets holding blind data. The interpolation has been done using between one and all principal components. All attributes are used. Data set #5 has been omitted as the errors are of significantly greater magnitude

According to Figure 5 the minimum RMS error is not always achieved by simply selecting all of the principal components. On the contrary, the figure indicates that for data set with a large number of attributes it is some times favorable to only include a subset of the principal components, for instance for data set #3 and data set #6.

To compare the errors seen in Figure 5 to the errors of Table 2, Table 3 shows the errors for each of the six data sets when the number of principal components selected is four. This

means the dimension of the transformed attribute space in both cases is four.

Table 3: RMS errors when using all attributes and four out of the eight principal components. The values in this table can also be seen in Figure 2. These errors are comparable to the last column of Table 2, as the dimension in both cases has been reduced to four.

Data set	4 principal components
#1	2.98
#2	3.86
#3	4.09
#4	5.45
#5	8.45
#6	4.61

Comparing the values of Table 3 to the values in the last column of Table 2 one can conclude that in most of the cases selecting four relevant attributes and using all four principal components yield smaller RMS errors than using all of the eleven attributes but only selecting the four most significant principal components.

2.5. Type of covariance model

Again we turn to the problem of interpolating using the four attributes X, Y, Z and the impedance and all four principal components. Up until now all covariance models has had a Gaussian model as their second term, but Table 4 and Figure 6 show respectively the RMS errors and the porosity estimate varying the type of covariance model.

Table 4: RMS errors when using four attributes – X, Y, Z and the impedance – and all principal components. The second term of the covariance matrix has been modeled as respectively a spherical, an exponential and a Gaussian model. The kriging results can be seen in Figure 6.

Data set	Spherical	Exponential	Gaussian
#1	3.98	3.99	2.91
#2	4.84	5.01	3.58
#3	6.71	6.74	4.14
#4	8.75	7.30	5.52
#5	11.7	12.5	9.37
#6	12.6	8.97	5.20

The table reveals that a Gaussian model yields by far the smallest RMS errors, and the spherical and exponential type of covariance model results in errors of approximately the same magnitude. Which of the latter two models that in this specific test case is the most favorable depends on the data set.

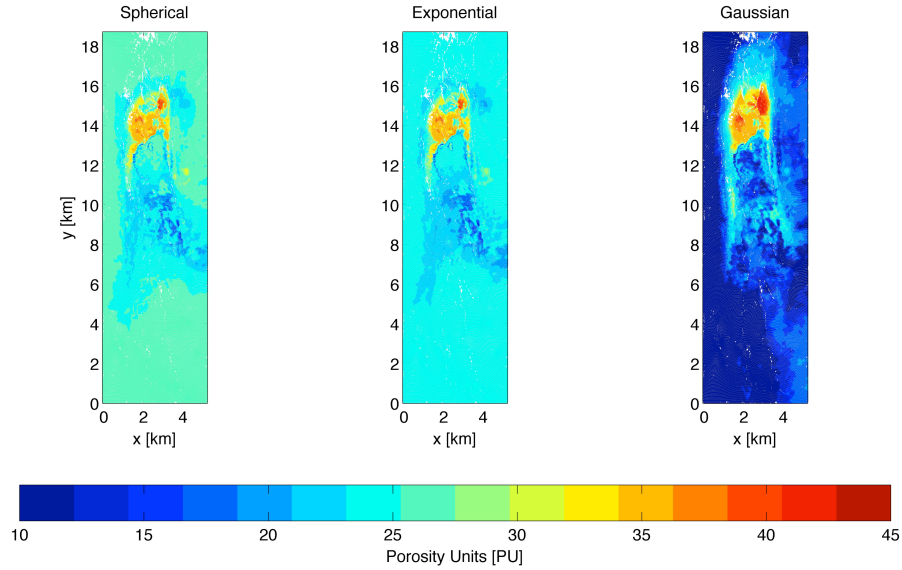


Figure 6: Estimated porosity levels for the same test case but where the second term of the covariance model has been modeled as respectively a spherical, an exponential and a Gaussian covariance model. We have used data set #1. The results have the RMS errors seen in Table 4, second row.

Figure 6 shows the estimated porosity levels for data set #1. It is noticed that the Gaussian type of covariance model allows us better to capture the variety of the subsurface especially for the part of the XY domain in which we have no known data.

2.6. Uncertainty estimates

Figure 7 and Figure 8 shows uncertainty estimates of the kriging results of the test case with data set #1. Again we have used the four attributes X, Y, Z and the impedance and the full set of principal components. The second term of the covariance model is a Gaussian model, which means the kriging results are identical to those that can be seen in Figure 6 to the right. The visual difference is due to a change in the color bar.

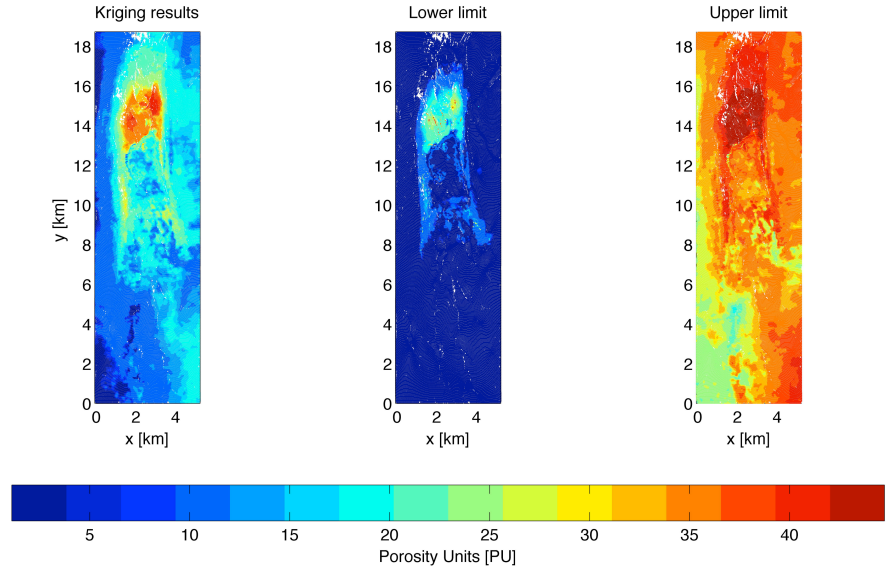


Figure 7: Estimated porosity levels as well as lower and upper limit of the 95% confidence interval for the same test case as shown in the right most plot of Figure 6. Notice the color bar has been altered slightly.

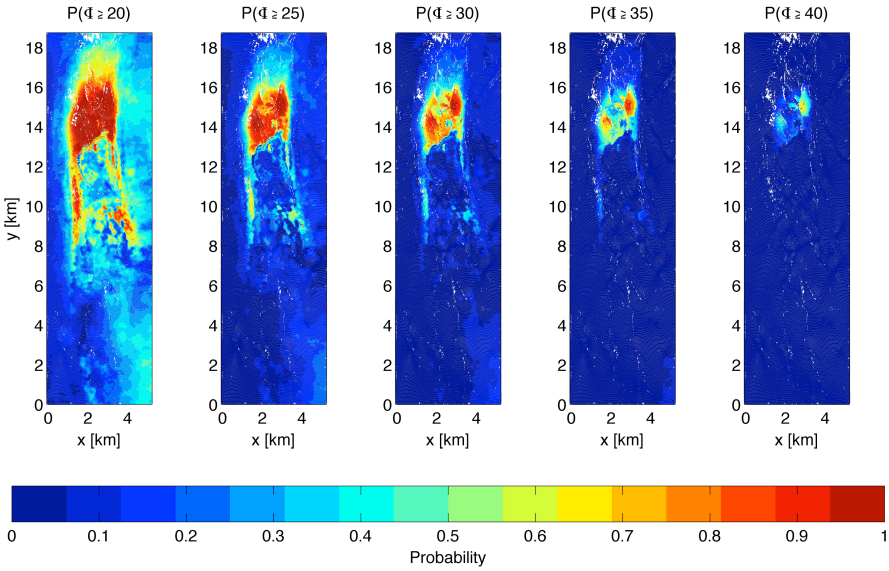


Figure 7: Probability of the estimated porosity levels being higher than certain values. Again we have chosen the test case with four attributes and the complete set of principal components.

2.7. Running times

To give an idea of the performance of the algorithm when it comes to running times we have timed the algorithm solving the test case of data set #1, all attributes and all principal components. Solving the test case we have run MATLAB® R2010a on a MacBook Pro with a 2.66 GHz Intel Core 2 Duo processor and 4 GB of RAM.

The time the algorithm spends has been classified on the following three main tasks:

1) Initialization, scaling and transformation 0.160 seconds
2) Computation of optimal covariance parameters 1.80 seconds
3) Kriging in more than 76,000 grid points 47.0 seconds

It should be noted that the optimization uses a well-qualified initial guess, which of course lowers the number of iterations needed. The kriging itself is in any case the most time consuming tasks, but considering the number of grid points 47 seconds definitely is acceptable.

It should also be noted, that the running time of the kriging task of course also depends on the size of the kriging system. The dimensions of the kriging system translate to the number of known data points. Data set #1 has approximately 100 known data points.

3. ACKNOWLEDGEMENTS

The present work was sponsored by the Danish Council for Independent Research - Technology and Production Sciences (FTP grant no. 274-09-0332) and DONG Energy.

4. REFERENCES

[1] Goovaerts, P. (1997). Geostatistics for Natural Resources Evaluation: New York, Oxford University Press.
[2] Hansen, T. M., Mosegaard, K., Pedersen-Tatalovic, R., Uldall, A., Jacobsen, N. L. (2008). Attribute-guided well-log interpolation applied to low-frequency impedance estimation. Geophysics , 73 (6), 83-95.
[3] Hansen, T. M., Mosegaard, K., Schiøtt, C. (2010). Kriging interpolation in seismic attribute space applied to the South Arne Field, North Sea. Personal communication, accepted for publication in Geophysics.
[4] Pardo-Igúzquiza, E. (1998). Maximum Likelihood Estimation of Spatial Covariance Parameters. Mathematical Geology , 30 (1), 95-108.
[5] Pedersen-Tatalovic, R., Uldall, A., Jacobsen, N. L., Hansen, T. M., Mosegaard, K. (2008, May). Event-based low-frequency impedance modeling using well logs and seismic attributes. The Leading Edge , 592-603.
[6] Shlens, J. (2009, April 22). A Tutorial on Principal Component Analysis. Retrieved Marts 2010, from http://www.snl.salk.edu/~shlens/pca.pdf

APPENDIX B

Paper II

A Frequency Matching Method for Generation of a Priori Sample Models from Training Images

Authors:
Katrine Lange, Jan Frydendall, Thomas Mejer Hansen, Knud Skou Cordua and Klaus Mosegaard

Published in:
Proceedings of the *15th Annual Conference of the International Association for Mathematical Geoscience (IAMG 2011)*
Salzburg, Austria
5-9 September 2011

A Frequency Matching Method for Generation of a Priori Sample Models from Training Images

Katrine LANGE[1,2], Knud Skou CORDUA[1], Jan FRYDENDALL[1], Thomas Mejer HANSEN[1], and Klaus MOSEGAARD[1]

[1] Center of Energy Resources Engineering,
Department of Informatics and Mathematical Modeling,
Technical University of Denmark, Denmark,
[2] katla@imm.dtu.dk

Abstract

This paper presents a Frequency Matching Method (FMM) for generation of a priori sample models based on training images and illustrates its use by an example. In geostatistics, training images are used to represent a priori knowledge or expectations of models, and the FMM can be used to generate new images that share the same multi-point statistics as a given training image.

The FMM proceeds by iteratively updating voxel values of an image until the frequency of patterns in the image matches the frequency of patterns in the training image; making the resulting image statistically indistinguishable from the training image.

1. Background

Consider a training image with N voxels (or pixels if the image is only 2D). Let z_k denote the value of the kth voxel of the image, $k = 1, \dots, N$. Here, we shall assume that the training image is a realization of a random process satisfying:

1) Voxel value z_k depends only on the values of the voxels in a certain neighborhood $\mathcal{N}_k$ around voxel k. Voxel k itself is not contained in $\mathcal{N}_k$. Let $\mathbf{z}_k$ be an ordered vector of the values of the voxels in $\mathcal{N}_k$; we then have:

$$f_Z\left(z_k \middle| z_N, \dots, z_{k+1}, z_{k-1}, \dots, z_1\right) = f_Z(z_k|\mathbf{z}_k).$$

2) For an image of infinite size the geometrical shape of all neighborhoods $\mathcal{N}_k$ are identical. This implies that if voxel k has coordinates (k_1, k_2, k_3), and voxel l has coordinates (l_1, l_2, l_3), then:

$$(n_1, n_2, n_3) \in \mathcal{N}_k \Rightarrow (n_1 - k_1 + l_1, n_2 - k_2 + l_2, n_3 - k_3 + l_3) \in \mathcal{N}_l.$$

3) We assume ergodicity, i.e.:

$$\mathbf{z}_k = \mathbf{z}_l \Rightarrow f_Z(z_k|\mathbf{z}_k) = f_Z(z_l|\mathbf{z}_l).$$

The basis of sequential simulation (e.g. Strebelle, 2002) is to exploit the assumptions above to estimate $f_Z(z_k|\mathbf{z}_k)$, and to use these conditions to generate new realizations of the random process from which the training image is a realization. The FMM does not operate by directly using conditional probabilities but it represents images by their frequency distribution, which is derived using neighborhoods of voxels. The frequency distribution is closely related to conditional probabilities.

2. The Frequency Distribution

Before presenting the FMM we need to define what we denote the frequency distribution. To do so we will reuse the concept of neighborhoods from section 1 as well as the notation. Given an image with the set of voxels $\mathcal{Z} = \{1, \dots, N\}$ and voxel values $z_1, \dots, z_N$ we define the template function Ω as a function that takes as argument a voxel k and returns the set of voxels belonging to the neighborhood of voxel k. The neighborhood is denoted $\mathcal{N}_k$, and we will use the notation $\mathcal{N}_k = \Omega(k)$.

In the FMM the neighborhood of a voxel is indirectly given by the statistical properties of the image itself; however, the shape of a neighborhood satisfying the assumptions from section 1 is unknown. For each training image one must therefore define a template function that seeks to correctly describe the neighborhood.

Let $|\mathcal{N}_k|$ denote the number of voxels in $\mathcal{N}_k$. We define the set of inner voxels, $\mathcal{Z}_{in}$, of the image as:

$$\mathcal{Z}_{in} = \left\{ k \middle| |\mathcal{N}_k| = \max_{l \in \mathcal{Z}} |\mathcal{N}_l| \right\}.$$

Typically, voxels on the boundary or close to the boundary of an image will not be inner voxels. It is the choice of template function that determines whether or not a voxel is an inner voxel.

The frequency distribution of an image is computed by scanning through all inner voxels of the image. For each of these we identify first the neighboring voxels and then the values of those. For voxel $k \in \mathcal{Z}_{in}$, the values of the neighboring voxels are denoted by the vector $\mathbf{z}_k$. The length of this vector equals the number of voxels in the neighborhood $\mathcal{N}_k$, which will be constant for all inner voxels; this follows trivially from the definition of inner voxels. We denote this number n. As each voxel can take on m different values, there exists up to m^n different types of neighborhoods; i.e. m^n different combinations for the values in $\mathbf{z}_k$.

Using the above definition of a neighborhood we now introduce the concept of patterns. The kth pattern $\mathcal{P}_k$ of the image is defined as the union of an inner voxel k and the set of its neighboring voxels. We will denote voxel k the center voxel of the kth pattern regardless of the geometrical shape of $\mathcal{P}_k$. Trivially, it follows that there exist m^{n+1} different types of patterns in the image. The type of a pattern is characterized by the (ordered) values of $\mathbf{z}_k$ and the value of the kth voxel itself. It should be stressed that the subindex k of $\mathcal{P}_k$, as well as of $\mathcal{N}_k$, represents the center voxel and thereby the location of the pattern, and it does not contrain any information on the type of the pattern.

Let p_i, for $i = 1, \dots, m^{n+1}$, count the number of times a pattern of type i appears in the image. These counts are used to represent the frequency distribution of an image. After having scanned through all inner voxels exactly once (the order is irrelevant) the frequency distribution is given by the vector $\mathbf{p}$:

$$\mathbf{p} = [p_1 \quad \cdots \quad p_{m^{n+1}}] = p_\Omega(z_1, \dots, z_N).$$

Here p_Ω is the function that, given an image and a template function Ω, computes the frequency distribution of the image with respect to the template as just described.

We notice that, for a given template, the frequency distribution of an image is uniquely determined. The opposite, however, does not hold. Different images can have the same frequency distribution. This is exactly what we seek to exploit by using the frequency distribution to generate multiple new images, at the same time similar to, and different from, our training image.

3 The Frequency Matching Method

The Frequency Matching Method proceeds by iteratively updating voxel values of an image, until the frequency of patterns in the image matches the frequency patterns in the training image. One of the primary tasks when formulating the method is to define a similarity function for how close the frequency distributions of two images are. Below we shall define the similarity function used in the current implementation, and describe the optimization method we have applied to solve the combinatorial optimization problem arising from this.

3.1 The Similarity Function

The similarity function plays the following two important roles:

I. It allows us to determine if the frequency distribution of an image and the frequency distribution of a training image are identical within the accuracy required and we therefore consider the image a valid realization of the random process from which the training image is a realized.

II. Given two different images, no matter how similar they might be, and a training image, the similarity function should determine which of the two images is most similar to a valid realization of the same process as the training image, or if the two images are equally similar. At the same time it should reflect (in some sense) how close the images are to being a valid realization.

Using an iterative solution method, point I is used to determine if the method has converged to an acceptable solution, whereas point II guides the method through the solution space, helping it to converge.

As we do not know the random process of which the training image is a realization, we have chosen the chi-square measure of 'goodness of fit' between two sets of nominal data as a similarity function for our FMM implementation. This measure determines the distance between to frequency distributions by comparing the proportions of types of pattern in the two.

3.2 Applying the χ^2 Measure in the FMM

The chi-square measure can be applied to our situation using the following interpretations (see Bere and Chimedza, 2007):

samples Each frequency distribution is considered a sample, i.e., we have two independent samples; one for the image itself and one for the training image.

categories The samples are categorized with respect to the m^{n+1} exclusive and exhaustive types of patterns.

observations Each appearance or count of a pattern is an observation. For each sample, the number of observations equals the number of inner voxels in the corresponding image.

Given the frequency distributions of an image, $\mathbf{p}$, and of a training image, $\boldsymbol{\pi}$, we can compute what we denote to be the similarity function value of the image:

$$f(\mathbf{p}) = \chi^2(\mathbf{p},\boldsymbol{\pi}) = \sum_{i=1}^{m^{n+1}} \frac{(p_i - e_i)^2}{e_i} + \sum_{i=1}^{m^{n+1}} \frac{(\pi_i - \varepsilon_i)^2}{\varepsilon_i},$$

where e_i and ε_i denote the expected count of patterns of type i of the image and the training image, respectively. These are computed as:

$$e_i = \frac{(p_i + \pi_i)}{n_p + n_\pi} n_p,$$

$$\epsilon_i = \frac{(p_i + \pi_i)}{n_p + n_\pi} n_\pi,$$

and n_p and n_π are the number of inner voxels in the image and the training image, respectively.

Let χ^2 denote the chi-square value of the image computed from the two frequency distributions $\mathbf{p}$ and $\boldsymbol{\pi}$. f is a function of the frequency distribution $\mathbf{p}$ of the image, and the frequency distribution $\boldsymbol{\pi}$ of the training image. The training image and therefore its frequency distribution will remain unchanged when computing a new image; $\boldsymbol{\pi}$ has therefore been omitted as an argument of the similarity function. Furthermore, the frequency distribution $\mathbf{p}$ of the image is derived given a template function, i.e., the argument $\mathbf{p}$ of f depends on a template as well as on $z_1, \dots, z_N$, which means f is in fact a function of the image and a template function. However, to simplify the text, we have chosen to avoid these dependencies in the notation.

3.3 The Optimization Problem

The function f defined in section 3.2 seems to fulfill the two requirements we had, making the FMM a combinatorial optimization problem. The variables are the voxel values of the image. They can take on m different integer values namely $\{0, \dots, m-1\}$. Binary images, for instance, have $m = 2$. Given a template function Ω, the frequency distributions of the solution image, $\mathbf{p}$, and of a training image, $\boldsymbol{\pi}$, are computed by the frequency function p_Ω. Based on the two frequency distributions the similarity function of the image is computed. By minimizing the similarity function with respect to certain constraints, we can create images sharing the same multiple-point statistics as the training image. The resulting optimization problem can be expressed as follows:

$$\begin{aligned} \min_{z_1,\dots,z_N} \quad & f(\mathbf{p}) \\ \text{w.r.t.} \quad & \mathbf{p} = p_\Omega(z_1, \dots, z_N), \\ & z_k \in \{0, \dots, m-1\}, \quad \text{for } k = 1, \dots, N. \end{aligned}$$

If some of the voxel values are known beforehand, and the voxels are therefore not free variables, the last set of constraints can easily be altered, such that the set of values that the kth voxel can take is only a subset of $\{\mathbf{0}, \dots, \boldsymbol{m-1}\}$.

3. Example

We have now introduced the Frequency Matching Method for generating a priori sample models from training images, and this has led us to a combinatorial optimization problem. Our choice of solution method is, for now, the intuitively simple heuristic Simulated Annealing (SA) (e.g. Kirkpatrick et al., 1983). For future work we would also like to explore other solution methods in the hope of finding one better suited for optimization and sampling problems.

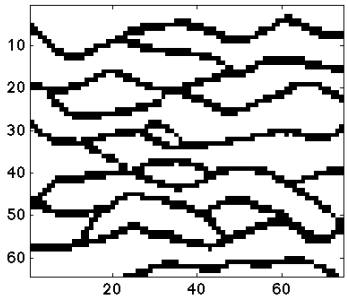


Figure 1: Training image.

The FMM has been implemented in MATLAB. To demonstrate the FMM we will consider a two-dimensional, binary training image with channel structures, see Figure 1.

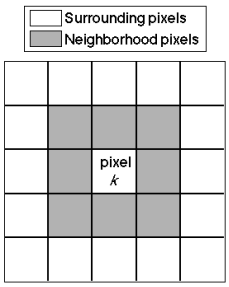


Figure 2: The template

We have defined the template such that the neighborhood of an arbitrary inner pixel k contains exactly the eight nearest pixels, see Figure 2. This relatively small neighborhood size is unlikely to completely satisfy our assumption of a pixel only being conditioned upon the pixels in its neighborhood. However, it will be shown that the method is still able to compute an acceptable solution. Due to the complexity of the method the size of the neighborhood greatly influences the running times, and for using much bigger templates we recommend implementing the method in Fortran, for instance.

We choose the exponential cooling rate for the SA, and the algorithm parameters are chosen manually. Discussing the strategies for choosing these optimally is beyond the scope of this text.

The starting image for SA is chosen to be all white. The SA algorithm searches the solution space consisting of images, and it moves from one image to another by randomly choosing a pixel and changing its value. Figure 3 and Figure 4 show the normalized frequency distributions of the training image and the image computed by the FMM, respectively. By 'normalized' we mean relative to the number of inner pixels in each of the images. Any normalized frequency distribution therefore sums to 1. Here we have truncated the ordinates of Figure 3 and Figure 4, as only one entry is significantly bigger than 0.08. The last entry is approximately 0.42 for both images. This entry is the one representing a white center pixel surrounded by all white neighboring pixels.

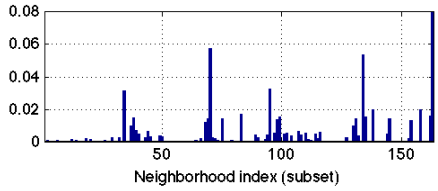


Figure 3: Normalized frequency distribution of the training image.

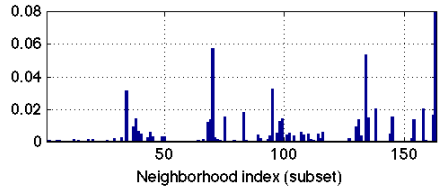


Figure 4: Normalized frequency distribution of the optimal solution image.

Notice that in Figure 3 and Figure 4 indexes corresponding to types of patterns appearing in neither the training image nor the solution image have been omitted

We observe that the FMM in terms of the frequency distributions has managed to match the training image quite well. Summing the bars of Figure 5 reveals that the two images have approximately 96.7% of their patterns in common. This number is likely to be improved by changing the parameters of the SA algorithm.

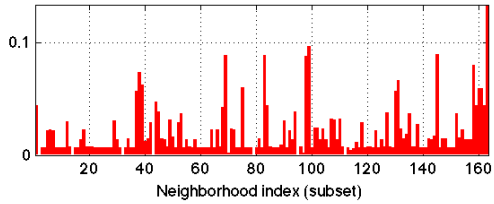


Figure 5: The absolute difference (in percent) between the normalized frequency distributions in Figure 3 and Figure 4.

Keep in mind that matching the frequency distributions only results in a useful image if our assumptions are met; i.e., if we chose a suitable template. Choosing too big a template means very long running times without sufficient gain in accuracy, and choosing too small a template will result in the picture not being similar to the training image. Our choice seems sufficient although not perfect, see Figure 6.

Figure 6 shows the image computed by the FMM. For this test case we have chosen to compute a 60×60 image based on a 64×74 training image but the method can produce images of arbitrary size. We notice that despite the relatively small template size, we have successfully recreated the channel structures. The channels even occasionally form loops, just like the channels of the training image.

One significant difference between the computed image and the training image is that the channels in the computed image are not all horizontally continuous across the image. We expect that this is merely a matter of choice of template and also the number of iterations the algorithm has been allowed to perform.

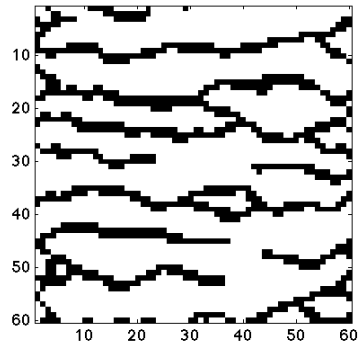


Figure 6: The computed solution image.

Another difference is the boundaries. It seems the method creates some artifacts along the boundaries. The density of channels is much higher on the left and right boundary then in the middle of the image. In the middle it resembles our training image and we therefore could have some issues in the way we treat non-inner pixels.

Notice how matching the frequency distributions indirectly results in the proportion of channels versus background in the computed picture to be in correspondence with the proportion of channels versus background in the training image. As stated, this is merely an example of the performance of the FMM. The method has also been applied to training images with different structures and shown similar results.

4. Conclusions and Future Work

In this paper we have derived the Frequency Matching Method for generation of a priori sample models from training images. We have implemented the method in MATLAB and shown the results of a simplified test case. The test example shows that the method is indeed able to produce an image that shares the same multi-point statistics as the training image.

This paper only scratches the surface of this newly developed method. In order to better understand its potential we would like to:

- Experiment thoroughly with training images with different structures.
- Investigate the convergence rate and performance of the FMM combined with other optimization methods.
- Explain and eventually avoid possible artifacts for non-inner voxels.

References

BERE, A., CHIMEDZA, C. (2007): A Comparative Study of the Accuracy of the Chi-Squared Approximation for the Power-Divergence Statistic and Pearson's Chi-Square Statistic in Sparse Contingency Tables. Journal of Statistical Research, Vol. 41, No. 2, pp. 73-81.

KIRKPATRICK, S., GELATT, C. D., VECCHI, M. P. (1983): Optimization by Simulated Annealing. Science, New Series, Vol. 220, No. 4598, pp. 671-680, May.

STREBELLE, S. (2002): Conditional Simulation of Complex Geological Structures Using Multiple-Point Statistics. Mathematical Geology, Vol. 34, No. 1, January.

APPENDIX C

Paper III

A Frequency Matching Method: Solving Inverse Problems by use of Geologically Realistic Prior Information

Authors:
Katrine Lange, Jan Frydendall, Knud Skou Cordua, Thomas Mejer Hansen, Yulia Melnikova and Klaus Mosegaard

Published in:
Mathematical Geosciences

Math Geosci
DOI 10.1007/s11004-012-9417-2

A Frequency Matching Method: Solving Inverse Problems by Use of Geologically Realistic Prior Information

Katrine Lange · Jan Frydendall · Knud Skou Cordua · Thomas Mejer Hansen · Yulia Melnikova · Klaus Mosegaard

Received: 3 February 2012 / Accepted: 16 July 2012

Abstract The frequency matching method defines a closed form expression for a complex prior that quantifies the higher order statistics of a proposed solution model to an inverse problem. While existing solution methods to inverse problems are capable of sampling the solution space while taking into account arbitrarily complex a priori information defined by sample algorithms, it is not possible to directly compute the maximum a posteriori model, as the prior probability of a solution model cannot be expressed. We demonstrate how the frequency matching method enables us to compute the maximum a posteriori solution model to an inverse problem by using a priori information based on multiple point statistics learned from training images. We demonstrate the applicability of the suggested method on a synthetic tomographic crosshole inverse problem.

Keywords Geostatistics · Multiple point statistics · Training image · Maximum a posteriori solution

1 Introduction

Inverse problems arising in the field of geoscience are typically ill-posed; the available data are scarce and the solution to the inverse problem is therefore not well-determined. In probabilistic inverse problem theory the solution to a problem is given as an a posteriori probability density function that combines states of information provided by observed data and the a priori information (Tarantola 2005). The ambiguities of the solution of the inverse problem due to the lack of restrictions on the solution is then reflected in the a posteriori probability.

K. Lange (✉) · J. Frydendall · K.S. Cordua · T.M. Hansen · Y. Melnikova · K. Mosegaard
Center for Energy Resources Engineering, Department of Informatics and Mathematical Modeling, Technical University of Denmark, Richard Petersens Plads, Building 321, 2800 Kongens Lyngby, Denmark
e-mail: katla@imm.dtu.dk

Published online: 22 September 2012

A priori information used in probabilistic inverse problem theory is often covariance-based a priori models. In these models the spatial correlation between the model parameters is defined by two-point statistics. In reality, two-point-based a priori models are too limited to capture curvilinear features such as channels or cross beddings. It is therefore often insufficient to rely only on the two-point statistics, and thus higher order statistics must also be taken into account in order to correctly produce geologically realistic descriptions of the subsurface. It is assumed that geological information is available in the form of a training image. This image could for instance have been artificially created to describe the expectations for the solution model or it could be information from a previous solution to a comparable inverse problem. The computed models should not be identical to the training image, but rather express a compromise between honoring observed data and comply with the information extracted from the training image. The latter can be achieved by ensuring that the models have the same multiple point statistics as the training image.

Guardiano and Srivastava (1993) proposed a sequential simulation algorithm that was capable of simulating spatial features inferred from a training image. Their approach was computationally infeasible until Strebelle (2002) developed the single normal equation simulation (*snesim*) algorithm. Multiple point statistics in general and the *snesim* algorithm in particular have been widely used for creating models based on training images and for solving inverse problems, see for instance Caers and Zhang (2004), Arpat (2005), Hansen et al. (2008), Peredo and Ortiz (2010), Suzuki and Caers (2008), Jafarpour and Khodabakhshi (2011). A method called the probability perturbation method (PPM) has been proposed by Caers and Hoffman (2006). It allows for gradual deformation of one realization of *snesim* to another realization of *snesim*. Caers and Hoffman propose to use the PPM method to find a solution to an inverse problem that is consistent with both a complex prior model, as defined by a training image, and data observations. PPM is used iteratively to perturb a realization from *snesim* while reducing the data misfit. However, as demonstrated by Hansen et al. (2012), as a result of the probability of the prior model not being evaluated, the model found using PPM is not the maximizer of the posterior density function, but simply the realization of the multiple point based prior with the highest likelihood value. There is no control of how reasonable the computed model is with respect to the prior model. It may be highly unrealistic.

The sequential Gibbs sampling method by Hansen et al. (2012) is used to sample the a posteriori probability density function given, for example a training image based prior. However, as with the PPM it cannot be used for optimization and locating the maximum a posteriori (MAP) model, as the prior probability is not quantified. The focus of our research is the development of the frequency matching (FM) method. The core of this method is the characterization of images by their multiple point statistics. An image is represented by the histogram of the multiple point-based spatial event in the image; this histogram is denoted the frequency distribution of the image. The most significant aspect of this method, compared to existing methods based on multiple point statistics for solving inverse problems, is the fact that it explicitly formulates an a priori probability density distribution, which enables it to efficiently quantify the probability of a realization from the a priori probability.

The classical approach when solving inverse problems by the least squares methods assumes a Gaussian prior distribution with a certain expectation. Solution models

to the inverse problem are penalized depending on their deviation from the expected model. In the FM method, the frequency distribution of the training image acts as the expected model and a solution image is penalized depending on how much its frequency distribution deviates from that of the training image. To perform this comparison we introduce a dissimilarity measure between a training image and a model image as the χ^2 distance between their frequency distributions. Using this dissimilarity measure for quantifying the a priori probability of a model the FM method allows us to directly compute the MAP model, which is not possible using known techniques such as the PPM and sequential Gibbs sampling methods.

Another class of methods are the Markov random fields (MRF) methods (Tjelmeland and Besag 1998). The prior probability density given by Markov methods involves a product of a large number of marginals. A disadvantage is therefore, despite having an expression for the normalization constant, that it can be computationally expensive to compute. Subclasses of the MRF methods such as Markov mesh models (Stien and Kolbjørnsen 2011) and partially ordered Markov models (Cressie and Davidson 1998) avoid the computation of the normalization constant, and this advantage over the MRF methods is shared by the FM method. Moreover, in contrast to methods such as PMM and MRF, the FM method is fully non-parametric, as it does not require probability distributions to be written in a closed form.

This paper is ordered as follows. In Sect. 2 we define how we characterize images by their frequency distributions, we introduce our choice of a priori distribution of the inverse problem and we elaborate on how it can be incorporated into traditional inverse problem theory. Our implementation of the FM method is discussed in Sect. 3. In Sect. 4 we present our test case and the results when solving an inverse problem using frequency matching-based a priori information. Section 5 summarizes our findings and conclusions.

2 Method

In geosciences, inverse problems involve a set of measurements or observations $\mathbf{d}^{\text{obs}}$ used to determine the spatial distribution of physical properties of the subsurface. These properties are typically described by a model with a discrete set of parameters, $\mathbf{m}$. For simplicity, we will assume that the physical property is modeled using a regular grid in space. The model parameters are said to form an image of the physical property.

Consider the general forward problem,

$$\mathbf{d} = g(\mathbf{m}), \tag{1}$$

of computing the observations $\mathbf{d}$ given the perhaps non-linear forward operator g and the model parameters $\mathbf{m}$. The values of the observation parameters are computed straightforwardly by applying the forward operator to the model parameters. The associated inverse problem consists of computing the model parameters $\mathbf{m}$ given the forward operator g and a set of observations $\mathbf{d}^{\text{obs}}$. As the inverse problem is usually severely under-determined, the model $\mathbf{m}$ that satisfies $\mathbf{d}^{\text{obs}} = g(\mathbf{m})$ is not uniquely determined. Furthermore, some of the models satisfying $\mathbf{d}^{\text{obs}} = g(\mathbf{m})$ within the required level of accuracy will be uninteresting for a geoscientist as the nature of the

forward operator g and the measurement noise in $\mathbf{d}^{\text{obs}}$ may yield a physically unrealistic description of the property. The inverse problem therefore consists of not just computing a set of model parameters satisfying Eq. 1, but computing a set of model parameters that gives a realistic description of the physical property while honoring the observed data. The FM method is used to express how geologically reasonable a model is by quantifying its a priori probability using multiple point statistics. Letting the a priori information be available in, for instance, a training image, the FM method solves an inverse problem by computing a model that satisfies not only the relation from Eq. 1 but a model that is also similar to the training image. The latter ensures that the model will be geologically reasonable.

2.1 The Maximum A Posteriori Model

Tarantola and Valette (1982) derived a probabilistic approach to solve inverse problems where the solution to the inverse problem is given by a probability density function, denoted the a posteriori distribution. This approach makes use of a prior distribution and a likelihood function to assign probabilities to all possible models. The a priori probability density function ρ describes the data independent prior knowledge of the model parameters; in the FM method we choose to define it as follows

$$\rho(\mathbf{m}) = \text{const.} \exp\big(-\alpha\ f(\mathbf{m})\big),$$

where α acts as a weighting parameter and f is a dissimilarity function presented in Sect. 2.4. Traditionally, f measures the distance between the model and an a priori model. The idea behind the FM method is the same, except we wish not to compare models directly but to compare the multiple point statistics of models. We therefore choose a traditional prior but replace the distance function such that instead of measuring the distance between models directly, we measure the dissimilarity between them. The dissimilarity is expressed as a distance between their multiple point statistics.

The likelihood function L is a probabilistic measure of how well data associated with a certain model matches the observed data, accounting for the uncertainties of the observed data,

$$L\big(\mathbf{m}, \mathbf{d}^{\text{obs}}\big) = \text{const.} \exp\left(-\frac{1}{2}\big\|\mathbf{d}^{\text{obs}} - g(\mathbf{m})\big\|^2_{\mathbf{C}_\text{d}}\right).$$

Here, $\mathbf{C}_\text{d}$ is the data covariance matrix and the measurement errors are assumed to be independent and Gaussian distributed with mean values 0. The a posteriori distribution is then proportional to the product of the prior distribution and the likelihood

$$\sigma(\mathbf{m}) = \text{const.}\rho(\mathbf{m}) L\big(\mathbf{m}, \mathbf{d}^{\text{obs}}\big).$$

The set of model parameters that maximizes the a posteriori probability density is called the maximum a posteriori (MAP) model

$$\begin{aligned}
\mathbf{m}^{\text{MAP}} &= \arg\max_{\mathbf{m}} \big\{\sigma(\mathbf{m})\big\} \\
&= \arg\min_{\mathbf{m}} \big\{-\log\sigma(\mathbf{m})\big\} \\
&= \arg\min_{\mathbf{m}} \left\{\frac{1}{2}\big\|\mathbf{d}^{\text{obs}} - g(\mathbf{m})\big\|^2_{\mathbf{C}_\text{d}} + \alpha\ f(\mathbf{m})\right\}.
\end{aligned}$$

The dissimilarity function f is a measure of how well the model satisfies the a priori knowledge that is available, for example from a training image. The more similar, in some sense, the image from a set of model parameters $\mathbf{m}$ is to the training image the smaller the function value $f(\mathbf{m})$ is. Equivalently to the more traditional term $\|\mathbf{m} - \mathbf{m}^{\text{prior}}\|^2_{\mathbf{C}_\text{m}}$, stemming from a Gaussian a priori distribution of the model parameters with mean values $\mathbf{m}^{\text{prior}}$ and covariance matrix $\mathbf{C}_\text{m}$, $f(\mathbf{m})$ can be thought of as a distance. It is not a distance between $\mathbf{m}$ and the training image ($f(\mathbf{m})$ may be zero for other images than the training image), but a distance between the multiple point statistics of the image formed by the model parameters and the multiple point statistics of the training image.

2.2 The Multiple Point Statistics of an Image

Consider an image $Z = \{1, 2, \ldots, N\}$ with N voxels (or pixels if the image is only two dimensional) where the voxels can have the m different values $0, 1, \ldots, m-1$. We introduce the N variables, $z_1, z_2, \ldots, z_N$ and let z_k describe the value of the kth voxel of the image. It is assumed that the image is a realization of an unknown, random process satisfying:

1. The value of the kth voxel, z_k, is, given the values of voxels in a certain neighborhood $\mathcal{N}_k$ around voxel k, independent of voxel values not in the neighborhood. Voxel k itself is not contained in $\mathcal{N}_k$. Let $\mathbf{z}_k$ be a vector of the values of the ordered neighboring voxels in $\mathcal{N}_k$; we then have
$$f_Z(z_k|z_N, \ldots, z_{k+1}, z_{k-1}, \ldots, z_1) = f_Z(z_k|\mathbf{z}_k),$$
where f_Z denotes the conditional probability distribution of the voxel z_k given the values of the voxels within the neighborhood.
2. For an image of infinite size the geometrical shape of all neighborhoods $\mathcal{N}_k$ are identical. This implies that if voxel k has coordinates (k_x, k_y, k_z), and voxel l has coordinates (l_x, l_y, l_z), then
$$(n_x, n_y, n_z) \in \mathcal{N}_k \quad \Rightarrow \quad (n_x - k_x + l_x, n_y - k_y + l_y, n_z - k_z + l_z) \in \mathcal{N}_l.$$
3. If we assume ergodicity, that is, when two voxels, voxel k and voxel l, have the same values as their neighboring voxels, then the conditional probability distribution of voxel k and voxel l are identical
$$\mathbf{z}_k = \mathbf{z}_l \quad \Rightarrow \quad f_Z(z_k|\mathbf{z}_k) = f_Z(z_l|\mathbf{z}_l).$$

Knowing the conditionals $f_Z(z_k|\mathbf{z}_k)$ we know the multiple point statistics of the image, just as a variogram would describe the two-point statistics of an image. The basis of sequential simulation as proposed by Guardiano and Srivastava (1993) is to exploit the aforementioned assumptions to estimate the conditional probabilities $f_Z(z_k|\mathbf{z}_k)$ based on the marginals obtained from the training image, and then to use the conditional distributions to generate new realizations of the unknown random process from which the training image is a realization. The FM method, on the other hand, operates by characterizing images by their frequency distributions. As described in the following section, the frequency distribution of voxel values within the given neighborhood of an image is given by its marginal distributions. This means

that comparison of images is done by comparing their marginals. For now, the training image is assumed to be stationary. With the current formulation of the frequency distributions this is the only feasible approach. Discussion of how to avoid the assumption of stationarity exists in literature, see for instance the recent Honarkhah (2011). Some of these approaches mentioned here might also be useful for the FM method, but we will leave this to future research to determine.

2.3 Characterizing Images by their Frequency Distribution

Before presenting the FM method we define what we denote the frequency distribution. Given an image with the set of voxels $Z = \{1, \ldots, N\}$ and voxel values $z_1, \ldots, z_N$ we define the template function Ω as a function that takes as argument a voxel k and returns the set of voxels belonging to the neighborhood $\mathcal{N}_k$ of voxel k. In the FM method, the neighborhood of a voxel is indirectly given by the statistical properties of the image itself; however, the shape of a neighborhood satisfying the assumptions from Sect. 2.2 is unknown. For each training image one must therefore define a template function Ω that seeks to correctly describe the neighborhood. The choice of template function determines if a voxel is considered to be an inner voxel. An inner voxel is a voxel with the maximal neighborhood size, and the set of inner voxels, Z_{in}, of the image is therefore defined as

$$Z_{\text{in}} = \left\{ k \in Z\colon\ |\mathcal{N}_k| = \max_{l \in Z} |\mathcal{N}_l| \right\},$$

where $|\mathcal{N}_k|$ denotes the number of voxels in $\mathcal{N}_k$. Let n denote the number of voxels in the neighborhood of an inner voxel. Typically, voxels on the boundary or close to the boundary of an image will not be inner voxels. To each inner voxel z_k we assign a pattern value p_k; we say the inner voxel is the center voxel of a pattern. This pattern value is a unique identifier of the pattern and may be chosen arbitrarily. The most obvious choice is perhaps a vector value with the discrete variables in the pattern, or a scalar value calculated based on the values of the variables. The choice should be made in consideration of the implementation of the FM method. The pattern value is uniquely determined by the value of the voxel z_k and the values of the voxels in its neighborhood, $\mathbf{z}_k$. As the pattern value is determined by the values of $n + 1$ voxels, which can each have m different values, the maximum number of different patterns is m^{n+1}.

Let π_i, for $i = 1, \ldots, m^{n+1}$, count the number of patterns that have the ith pattern value. The frequency distribution is then defined as $\boldsymbol{\pi}$

$$\boldsymbol{\pi} = [\pi_1, \ldots, \pi_{m^{n+1}}].$$

Let p_Ω denote the mapping from voxel values of an image Z to its frequency distribution $\boldsymbol{\pi}$, that is, $p_\Omega(z_1, \ldots, z_N) = \boldsymbol{\pi}$.

Figure 1 shows an example of an image and the patterns it contains for the template function that defines neighborhoods as follows

$$\mathcal{N}_k = \left\{ l \in Z \setminus \{k\}\colon\ |l_x - k_x| \leq 1, |l_y - k_y| \leq 1 \right\}.$$

Recall from Sect. 2.2 that (l_x, l_y) are the coordinates of voxel l in this two-dimensional example image. We note that for a given template function the frequency

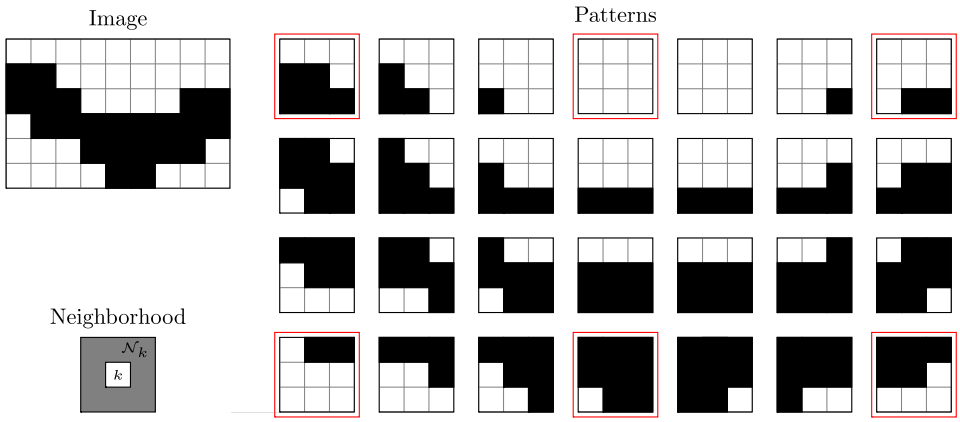


Fig. 1 Example of patterns found in an image. Notice how the image is completely described by the (ordered) patterns in every third row and column; the patterns are marked in red

distribution of an image is uniquely determined. The opposite, however, does not hold. Different images can, excluding symmetries, have the same frequency distribution. This is what the FM method seeks to exploit by using the frequency distribution to generate new images, at the same time similar to, and different from, our training image.

2.4 Computing the Similarity of Two Images

The FM method compares a solution image to a training image by comparing its frequency distribution to the frequency distribution of the training image. How dissimilar the solution image is to the training image is determined by a dissimilarity function, which assigns a distance between their frequency distributions. This distance reflects how likely the solution image is to be a realization of the same unknown process as the training image is a realization of. The bigger the distance, the more dissimilar are the frequency distributions and thereby also the images, and the less likely is the image to be a realization of the same random process as the training image. The dissimilarity function can therefore be used to determine which of two images is most likely to be a realization of the same random process as the training image is a realization of.

The dissimilarity function is not uniquely given but an obvious choice is the χ^2 distance also described in Sheskin (2004). It is used to measure the distance between two frequency distributions by measuring how similar the proportions of patterns in the frequency distributions are. Given two frequency distributions, the χ^2 distance estimates the underlying distribution. It then computes the distance between the two frequency distributions by computing each of their distances to the underlying distribution. Those distances are computed using a weighted Euclidean norm where the weights are the inverse of the counts of the underlying distribution, see Fig. 2. In our research, using the counts of the underlying distribution turns out to be a favorable weighting of small versus big differences instead of using a traditional p-norm as used by Peredo and Ortiz (2010).

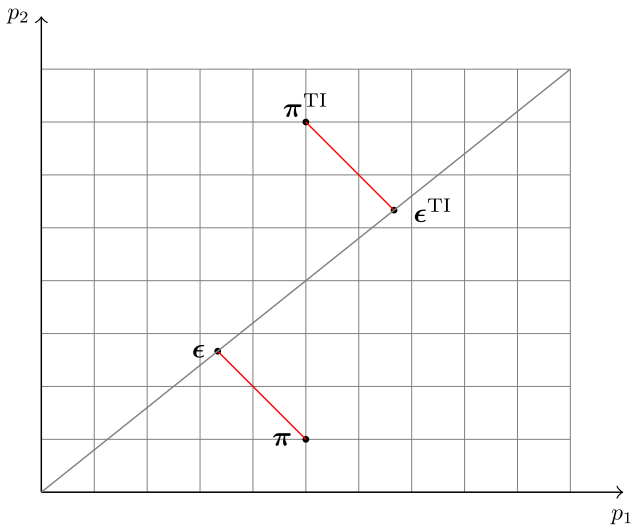


Fig. 2 Illustration of the χ^2 distance between two frequency distributions $\boldsymbol{\pi}$ and $\boldsymbol{\pi}^{\mathrm{TI}}$, each containing the counts of two different pattern values, p_1 and p_2. The difference between the frequency distributions is computed as the sum of the length of the two red line segments. The length of each line segment is computed using a weighted Euclidean norm. The counts of the underlying distribution are found as the orthogonal projection of the frequency distributions onto the a line going through the origin such that $\|\boldsymbol{\pi} - \boldsymbol{\epsilon}\|_2 = \|\boldsymbol{\pi}^{\mathrm{TI}} - \boldsymbol{\epsilon}^{\mathrm{TI}}\|_2$

Hence, given the frequency distributions of an image, $\boldsymbol{\pi}$, and of a training image, $\boldsymbol{\pi}^{\mathrm{TI}}$, and by letting

$$I = \left\{i \in \left\{1, \ldots, m^{n+1}\right\}\colon \pi_i^{\mathrm{TI}} > 0\right\} \cup \left\{i \in \left\{1, \ldots, m^{n+1}\right\}\colon \pi_i > 0\right\}, \tag{2}$$

we compute what we define as the dissimilarity function value of the image

$$c(\boldsymbol{\pi}) = \chi^2\left(\boldsymbol{\pi}, \boldsymbol{\pi}^{\mathrm{TI}}\right) = \sum_{i \in I} \frac{(\pi_i^{\mathrm{TI}} - \epsilon_i^{\mathrm{TI}})^2}{\epsilon_i^{\mathrm{TI}}} + \sum_{i \in I} \frac{(\pi_i - \epsilon_i)^2}{\epsilon_i}, \tag{3}$$

where ϵ_i denotes the counts of the underlying distribution of patterns with the ith pattern value for images of the same size as the image and ϵ_i^{TI} denotes the counts of the underlying distribution of patterns with the ith pattern value for images of the same size as the training image. These counts are computed as

$$\epsilon_i = \frac{\pi_i + \pi_i^{\mathrm{TI}}}{n_Z + n_{\mathrm{TI}}} n_Z, \tag{4}$$

$$\epsilon_i^{\mathrm{TI}} = \frac{\pi_i + \pi_i^{\mathrm{TI}}}{n_Z + n_{\mathrm{TI}}} n_{\mathrm{TI}}, \tag{5}$$

where n_Z and n_{TI} are the total number of counts of patterns in the frequency distributions of the image and the training image, that is, the number of inner voxels in the image and the training image, respectively.

2.5 Solving Inverse Problems

We define the frequency matching method for solving inverse problems formulated as least squares problems using geologically complex a priori information as the fol-

lowing optimization problem

$$\begin{aligned} &\min_{z_1,\ldots,z_N} \left\| \mathbf{d}^{\text{obs}} - g(z_1,\ldots,z_N) \right\|^2_{\mathbf{C}_\text{d}} + \alpha\, c(\boldsymbol{\pi}), \\ &\text{w.r.t. } \boldsymbol{\pi} = p_\Omega(z_1,\ldots,z_N), \\ &z_k \in \{0,\ldots,m-1\} \quad \text{for } k = 1,\ldots,N, \end{aligned} \tag{6}$$

where $c(\boldsymbol{\pi})$ is the dissimilarity function value of the solution image defined by Eq. 3 and α is a weighting parameter. The forward operator g, which traditionally is a mapping from model space to data space, also contains the mapping of the categorical values $z_k \in \{0,\ldots,m-1\}$ for $k = 1,\ldots,N$ of the image into the model parameters $\mathbf{m}$ that can take m different discrete values.

The value of α cannot be theoretically determined. It is expected to depend on the problem at hand; among other factors its resolution, the chosen neighborhood function and the dimension of the data space. It can be thought of as playing the same role for the dissimilarity function as the covariance matrix $\mathbf{C}_\text{d}$ does for the data misfit. So it should in some sense reflect the variance of the dissimilarity function and in that way determine how much trust we put in the dissimilarity value. Variance, or trust, in a training image is difficult to quantify, as the training image is typically given by a geologist to reflect certain expectations to model. Not having a theoretical expression for α therefore allows us to manipulate the α value to loosely quantify the trust we have in the training image. In the case where we have accurate data but only a vague idea of the structures of the subsurface the α can be chosen low, in order to emphasize the trust we have in the data and the uncertainty we have of the structure of the model. In the opposite case, where data are inaccurate but the training image is considered to be a very good description of the subsurface, the α value can be chosen high, to give the dissimilarity function more weight.

Due to the typically high number of model parameters, the combinatorial optimization problem should be solved by use of an iterative solution method; such a method will iterate through the model space and search for the optimal solution. While the choice of solution method is less interesting when formulating the FM method, it is of great importance when applying it. The choice of solution method and the definition of how it iterates through the solution space by perturbing images has a significant impact on the feasibility of the method in terms of its running time. As we are not sampling the solution space we do not need to ensure that the method captures the uncertainty of the model parameters, and the ideal would be a method that converges directly to the maximum a posteriori solution. While continuous optimization problems hold information about the gradient of the objective function that the solution method can use to converge to a stationary solution, this is not the case for our discrete problem. Instead we consider the multiple point statistics of the training image when perturbing a current image and in that way we seek to generate models which better match the multiple point statistics of the training image and thus guide the solution method to the maximum a posteriori model.

2.6 Properties of the Frequency Matching Method

The FM method is a general method and in theory it can be used to simulate any type of structure, as long as a valid training image is available and a feasible template

function is chosen appropriately. If neighborhoods are chosen too small, the method will still be able to match the frequency distributions. However, it will not reproduce the spatial structures simply because these are not correctly described by the chosen multiple point statistics and as a result the computed model will not be realistic. If neighborhoods are chosen too big, CPU cost and memory demand will increase, and as a result the running time per iteration of the chosen solution method will increase. Depending on the choice of iterative solution method, increasing the size n of the neighborhood is likely to also increase the number of iterations needed and thereby increase the convergence time. When the size of neighborhoods is increased, the maximum number of different patterns, m^{n+1}, is also increased. The number of different patterns present is, naturally, limited by the number of inner voxels, which is significantly smaller than m^{n+1}. In fact, the number of patterns present in an image is restricted further as training images are chosen such that they describe a certain structure. This structure is also sought to be described in the solutions. The structure is created by repetition of patterns, and the frequency distributions will reveal this repetition by having multiple counts of the same pattern. This means, the number of patterns with non-zero frequency is greatly smaller than m^{n+1} resulting in the frequency distributions becoming extremely sparse. For bigger test cases, with millions of parameters, patterns consisting of hundreds of voxels and multiple categories, this behavior needs to be investigated further.

The dimension of the images, if they are two or three dimensional, is not important to the FM method. The complexity of the method is given by the maximal size of neighborhoods, n. The increase in n as a result of going from two- to three-dimensional images is therefore more important than the actual increase in physical dimensions. In fact, when it comes to assigning pattern values a neighborhood is, regardless of its physical dimension, considered one dimensional where the ordering of the voxels is the important aspect. Additionally, the number of categories of voxel values m does not influence the running time per iteration. As with the number of neighbors, n, it only influences the number of different possible patterns m^{n+1} and thereby influences the sparsity of the frequency distribution of the training image. The higher m is, the sparser is the frequency distribution. It is expected that the sparsity of the frequency distribution affects the level of difficulty of the combinatorial optimization problem.

Strebelle (2002) recommends choosing a training image that is at least twice as large as the structures it describes; one must assume this advice also applies to the FM method. Like the *snesim* algorithm, the FM method can approximate continuous properties by discretizing them into a small number of categories. One of the advantages of the FM method is that by matching the frequency distributions it indirectly ensures that the proportion of voxels in each of the m categories is consistent between the training image and the solution image. It is therefore not necessary to explicitly account for this ratio. Unlike the *snesim* algorithm, the computed solution images therefore need very little post treatment—in the current implementation the solution receives no post treatment. However, the α parameter does allow for the user to specify how strictly the frequency distributions should be matched. In the case where the data are considered very informative or the training image is considered far from reality, decreasing the α allows for the data to be given more weight and the multiple point statistics will not be as strictly enforced.

Constraints on the model parameters can easily be dealt with by reducing the feasible set $\{0, \ldots, m-1\}$ for those values of k in the constraints of the problem stated in Eq. 6. The constrained voxels remain part of the image Z and when computing the frequency distribution of an image they are not distinguished from non-constrained voxels. However, when perturbing an image all constraints of the inverse problem should at all times be satisfied and conditioned to the hard data. The additional constraints on the model parameters will therefore be honored.

3 Implementation

This section describes the current implementation of the frequency matching method. Algorithm 1 gives a general outline of how to apply the FM method, that is, how to solve the optimization problem from Eq. 6 with an iterative optimization method. In the remainder of the section, the implementation of the different parts of the FM method will be discussed. It should be noted that the implementation of the FM method is not unique; for instance, there are many options for how the solution method iterates through the model space by perturbing models. The different choices should be made depending on the problem at hand and the current implementation might not be favorable for some given problems. The overall structure in Algorithm 1 will be valid regardless of what choices are made on a more detailed level.

Algorithm 1: The Frequency Matching Method

Input: Training image, Z^{TI}, Starting image Z
Output: Maximum a posteriori image Z^{FM}
Compute frequency distribution of training image $\boldsymbol{\pi}^{\mathrm{TI}}$ and pattern list $\mathbf{p}$ (Algorithm 2)
Compute partial frequency distribution of starting image $\boldsymbol{\pi}$ (Algorithm 3)
while *not converged* **do**
  Compute perturbed image $\overline{Z}$ based on Z (Algorithm 4)
  Compute partial frequency distribution of perturbed image $\overline{\boldsymbol{\pi}}$ (Algorithm 5)
  if *accept the perturbed image* **then**
    Set $Z \leftarrow \overline{Z}$ and $\boldsymbol{\pi} \leftarrow \overline{\boldsymbol{\pi}}$
  end
end

The current implementation is based on a Simulated Annealing scheme. Simulated Annealing is a well-known heuristic optimization method first presented by Kirkpatrick et al. (1983) as a solution method for combinatorial optimization problems. The acceptance of perturbed images is done using an exponential cooling rate and the parameters controlling the cooling are tuned to achieve an acceptance ratio of approximately 15 accepted perturbed models for each 100 suggested perturbed models. A perturbed model is generated by erasing the values of the voxels in a part of the image and then re-simulating the voxel values by use of sequential simulation.

3.1 Reformulation of the Dissimilarity Function

The definition of the dissimilarity function from Eq. 3 has one great advantage that we for computational reasons simply cannot afford to overlook. As discussed previously, the frequency distributions are expected to be sparse as the number of patterns present in an image is significantly smaller than m^{n+1}. This means that a lot of the terms in the dissimilarity function from Eq. 3 will be zero, yet the dissimilarity function can be simplified further. It will be shown that the dissimilarity function value of a frequency distribution, $c(\boldsymbol{\pi})$, given the frequency distribution of a training image, $\boldsymbol{\pi}$, can be computed using only entries of $\boldsymbol{\pi}$ where $\boldsymbol{\pi}^{\mathrm{TI}} > 0$. In other words, to compute the dissimilarity function value of an image we need only to know the count of patterns in the image that also appear in the training image. Computationally, this is a great advantage as we can disregard the patterns in our solution image that do not appear in the training image and we need not compute nor store the entire frequency distribution of our solution image, which is shown by inserting the expressions of the counts for the underlying distribution defined by Eqs. 4 and 5

$$\begin{aligned} c(\boldsymbol{\pi}) &= \sum_{i \in I} \frac{(\pi_i^{\mathrm{TI}} - \epsilon_i^{\mathrm{TI}})^2}{\epsilon_i^{\mathrm{TI}}} + \sum_{i \in I} \frac{(\pi_i - \epsilon_i)^2}{\epsilon_i} \\ &= \sum_{i \in I} \frac{(\sqrt{\frac{n_Z}{n_{\mathrm{TI}}}}\, \pi_i^{\mathrm{TI}} - \sqrt{\frac{n_{\mathrm{TI}}}{n_Z}}\pi_i)^2}{\pi_i^{\mathrm{TI}} + \pi_i}. \end{aligned} \tag{7}$$

This leads to the introduction of the following two subsets of I

$$\begin{aligned} I_1 &= \left\{ i \in I : \pi_i^{\mathrm{TI}} > 0 \right\}, \\ I_2 &= \left\{ i \in I : \pi_i^{\mathrm{TI}} = 0 \right\}. \end{aligned}$$

The two subsets form a partition of I as they satisfy $I_1 \cup I_2 = I$ and $I_1 \cap I_2 = \emptyset$. The dissimilarity function Eq. 7 can then be written as

$$\begin{aligned} c(\boldsymbol{\pi}) &= \sum_{i \in I_1} \frac{(\sqrt{\frac{n_Z}{n_{\mathrm{TI}}}}\pi_i^{\mathrm{TI}} - \sqrt{\frac{n_{\mathrm{TI}}}{n_Z}}\pi_i)^2}{\pi_i^{\mathrm{TI}} + \pi_i} + \frac{n_{\mathrm{TI}}}{n_Z} \sum_{i \in I_2} \pi_i \\ &= \sum_{i \in I_1} \frac{(\sqrt{\frac{n_Z}{n_{\mathrm{TI}}}}\pi_i^{\mathrm{TI}} - \sqrt{\frac{n_{\mathrm{TI}}}{n_Z}}\pi_i)^2}{\pi_i^{\mathrm{TI}} + \pi_i} + \frac{n_{\mathrm{TI}}}{n_Z} \left(n_Z - \sum_{i \in I_1} \pi_i \right) \end{aligned} \tag{8}$$

recalling that $\sum_{i \in I} \pi_i = n_Z$ and that $\pi_i = 0$ for $i \notin I$.

A clear advantage of this formulation of the dissimilarity function is that the entire frequency distribution $\boldsymbol{\pi}$ of the image does not need to be known; as previously stated, it only requires the counts π_i of the patterns also found in the training image, which is for $i \in I_1$.

3.2 Computing and Storing the Frequency Distributions

The formulation of the dissimilarity function from Eq. 3 and later Eq. 8 means that it is only necessary to store non-zero entries in a frequency distribution of a training

image $\boldsymbol{\pi}^{\mathrm{TI}}$. Algorithm 2 shows how the frequency distribution of a training image is computed such that zero entries are avoided. The algorithm also returns a list $\mathbf{p}$ with the same number of elements as the frequency distribution and it holds the pattern values corresponding to each entry of $\boldsymbol{\pi}^{\mathrm{TI}}$.

Algorithm 2: Frequency Distribution of a Training Image

Input: Training Image Z^{TI}
Output: Frequency distribution $\boldsymbol{\pi}^{\mathrm{TI}}$, list of pattern values $\mathbf{p}$
Initialization: empty list $\boldsymbol{\pi}^{\mathrm{TI}}$, empty list $\mathbf{p}$
for *each inner voxel, i.e.,* $k \in Z_{\mathrm{in}}^{\mathrm{TI}}$ **do**
  Extract pattern k
  Compute pattern value p_k
  if *the pattern was previously found* **then**
    Add 1 to the corresponding entry of $\boldsymbol{\pi}^{\mathrm{TI}}$
  else
    Add p_k to the list of pattern values $\mathbf{p}$
    Set the corresponding new entry of $\boldsymbol{\pi}^{\mathrm{TI}}$ equal to 1
  end
end

Algorithm 3 computes the partial frequency distribution $\boldsymbol{\pi}$ of an image that is needed to evaluate the dissimilarity function $c(\boldsymbol{\pi}) = \chi^2(\boldsymbol{\pi}, \boldsymbol{\pi}^{\mathrm{TI}})$ from Eq. 8. The partial frequency distribution only stores the frequencies of the patterns also found in the training image.

Algorithm 3: Partial Frequency Distribution of an Image

Input: Image Z, list of pattern values $\mathbf{p}$ from the training image
Output: Partial frequency distribution $\boldsymbol{\pi}$
Initialization: all zero list $\boldsymbol{\pi}$ (same length as $\mathbf{p}$)
for *each inner voxel, i.e.,* $k \in Z_{\mathrm{in}}$ **do**
  Extract pattern k
  Compute pattern value p_k
  if *the pattern is found in the training image* **then**
    Add 1 to the corresponding entry of $\boldsymbol{\pi}$
  end
end

3.3 Perturbation of an Image

The iterative solver moves through the model space by perturbing models and this is the part of the iterative solver that leaves the most choices to be made. An intuitive but naive approach would be to simply change the value of a random voxel. This will result in a perturbed model that is very close to the original model, and it will therefore require a lot of iterations to converge. The current implementation changes the values of a block of voxels in a random place of the image.

Before explaining in detail how the perturbation is done, let $Z^{\text{cond}} \subset Z$ be the set of voxels that we have hard data for, which means their value is known and should be conditioned to. First a voxel k is chosen randomly. Then the value of all voxels in a domain $\mathcal{D}_k \subset (Z \setminus Z^{\text{cond}})$ around voxel k are erased. Last, the values of the voxels in $\mathcal{D}_k$ are simulated using sequential simulation. The size of the domain should be chosen to reflect how different the perturbed image should be from the current image. The bigger the domain, the fewer iterations we will expect the solver will need to iterate through the model space to converge, but the more expensive an iteration will become. Choosing the size of the domain is therefore a trade-off between number of iterations and thereby forward calculations and the cost of computing a perturbed image.

Algorithm 4 shows how an image is perturbed to generate a new image.

Algorithm 4: Perturbation of an Image

Input: Image Z, partial frequency distribution $\boldsymbol{\pi}$ of Z
Output: Perturbed image $\overline{Z}$
Initialization: set $\overline{\boldsymbol{\pi}} = \boldsymbol{\pi}$
Pick random voxel k
for *each voxel l around voxel k, i.e., $l \in \mathcal{D}_k$* **do**
  Erase the value of voxel l, i.e., z_l is unassigned
end
for *each unassigned voxel l around voxel k, i.e., $l \in \mathcal{D}_k$* **do**
  Simulate z_l given all assigned voxels in $\mathcal{N}_l$.
end

3.4 Updating the Frequency Distribution

As a new image is created by changing the value of a minority of the voxels, it would be time consuming to compute the frequency distribution of all voxel values of the new image when the frequency distribution of the old image is known. Recall that n is the maximum number of neighbors a voxel can have; inner voxels have exactly n neighbors. Therefore, in addiction to changing its own pattern value, changing the value of a voxel will affect the pattern value of at most n other voxels. This means that we obtain the frequency distribution of the new image by performing at most $n+1$ subtractions and $n+1$ additions per changed voxel to the entries of the already known frequency distribution.

The total number of subtractions and additions can be lowered further by exploiting the block structure of the set of voxels perturbed. The pattern value of a voxel will be changed when any of its neighboring voxels are perturbed, but the frequency distribution need only be updated twice for each affected voxel. We introduce a set of voxels Z^{aff}, which is the set of voxels who are affected when perturbing image Z into $\overline{Z}$, that is, the set of voxels whose pattern values are changed when perturbing image Z into image $\overline{Z}$

$$Z^{\text{aff}} = \{k \in Z:\ p_k \neq \overline{p}_k\}. \tag{9}$$

How the partial frequency distribution is updated when an image is perturbed is illustrated in Algorithm 5.

Algorithm 5: Update Partial Frequency Distribution of an Image

Input: Image Z, partial frequency distribution $\boldsymbol{\pi}$ of Z, perturbed image $\overline{Z}$, set of affected voxels Z^{aff}, set of pattern values $\mathbf{p}$ from the training image
Output: Partial frequency distribution $\overline{\boldsymbol{\pi}}$ of $\overline{Z}$
Initialization: set $\overline{\boldsymbol{\pi}} = \boldsymbol{\pi}$
for *each affected voxel, i.e.,* $k \in Z^{\text{aff}}$ **do**
  Extract pattern k from both Z and $\overline{Z}$
  Compute both pattern values p_k and $\overline{p}_k$
  if *the pattern* p_k *is present in the training image* **then**
    Subtract 1 from the corresponding entry of $\overline{\boldsymbol{\pi}}$
  end
  if *the pattern* $\overline{p}_k$ *is present in the training image* **then**
    Add 1 to the corresponding entry of $\overline{\boldsymbol{\pi}}$
  end
end

As seen in Algorithm 1, the FM method requires in total two computations of a frequency distribution, one for the training image and one for the initial image. The FM method requires one update of the partial frequency distribution per iteration. As the set of affected voxels Z^{aff} is expected to be much smaller than the total image Z, updating the partial frequency distribution will typically be much faster than recomputing the entire partial frequency distribution even for iterations that involve changing the values of a large set of voxels.

3.5 Multigrids

The multigrid approach from Strebelle (2002) that is based on the concept initially proposed by Gómez-Hernández (1991) and further developed by Tran (1994) can also be applied in the FM method. Coarsening the images allows the capture of large-scale structures with relatively small templates. As in the *snesim* algorithm, the results from a coarse image can be used to condition upon for a higher resolution image.

The multigrid approach is applied by running the FM method from Algorithm 1 multiple times. First, the algorithm is run on the coarsest level. Then the resulting image, with increased resolution, is used as a starting image on the next finer level, and so on. The resolution of an image can be increased by nearest neighbor interpolation.

4 Example: Crosshole Tomography

Seismic borehole tomography involves the measurement of seismic travel times between two or more boreholes in order to determine an image of seismic velocities in the intervening subsurface. Seismic energy is released from sources located in one borehole and recorded at multiple receiver locations in another borehole. In this way a dense tomographic data set that covers the interborehole region is obtained.

Consider a setup with two boreholes. The horizontal distance between them is ΔX and they both have the depth ΔZ. In each borehole a series of receivers and sources

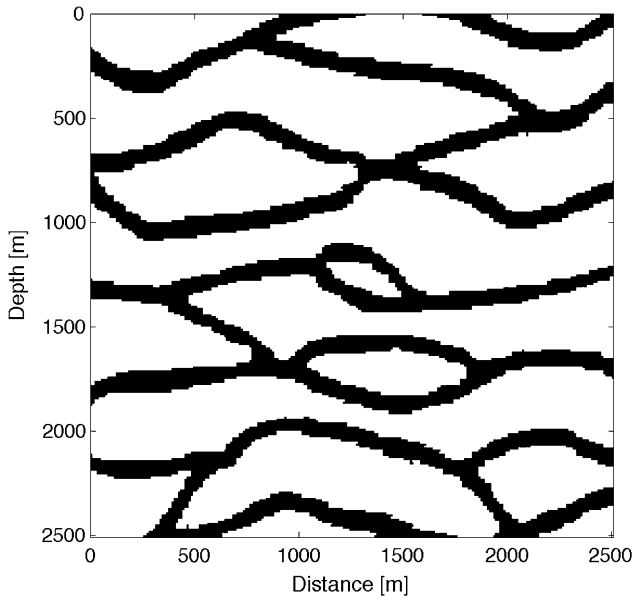


Fig. 3 Training image (resolution: 251 × 251 pixels)

Table 1 Parameter values for the test case

ΔX	500 m
ΔZ	1,200 m
Δx	10 m
Δz	10 m
d_{s}	250 m
d_{r}	100 m
v_{low}	1,600 m/s
v_{high}	2,000 m/s

is placed. The vertical domain between the two boreholes is divided into cells of dimensions Δx by Δz and it is assumed that the seismic velocity is constant within each cell. The model parameters of the problem are the propagation speeds of each cell. The observed data are the first arrival times of the seismic signals. For the series of sources and receivers in each borehole the distances between the sources are d_s and the distances between the receivers are d_r. We assume a linear relation between the data (first arrival times) and the model (propagation speed) from Eq. 1. The sensitivity of seismic signals is simulated as straight rays. However, any linear sensitivity kernel obtained using, for example, curvilinear rays or Fresnel zone-based sensitivity, can be used.

It is assumed that the domain consists of zones with two different propagation speeds, v_{low} and v_{high}. Furthermore a horizontal channel structure of the zones with high propagation speed is assumed. Figure 3 shows the chosen training image with resolution 251 cells by 251 cells where each cell is Δx by Δz. The training image is chosen to express the a priori information about the model parameters. The background (white pixels) represents a low velocity zone and the channel structures (black

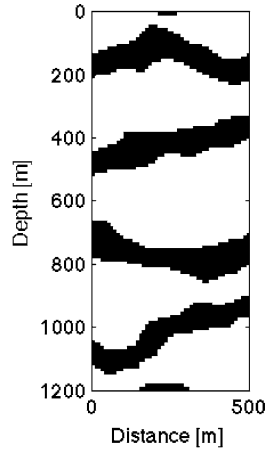


Fig. 4 Reference model (resolution: 50×120 pixels)

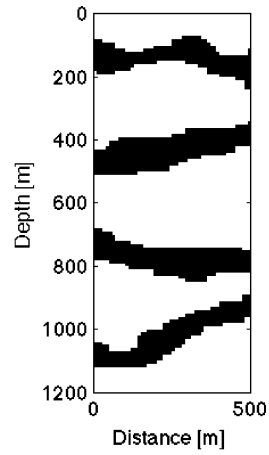


Fig. 5 Computed model for $\alpha = 1.8 \times 10^{-2}$ (resolution: 50×120 pixels)

pixels) are the high velocity zones. The problem is scalable and for the example we have chosen the parameters presented by Table 1.

The template function is chosen, such that the neighborhood of pixel k is the following set of pixels

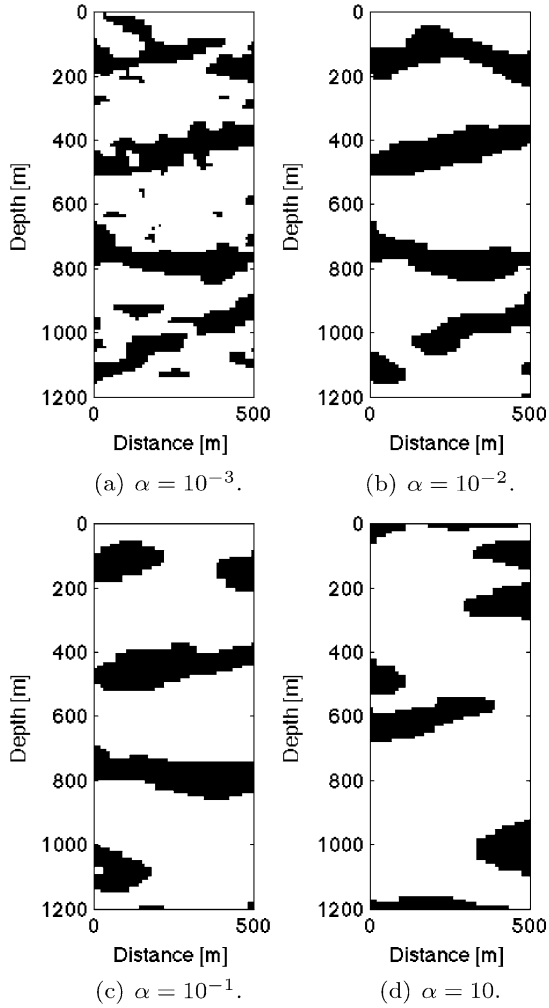
$$\mathcal{N}_k = \big\{ l \in Z \setminus \{k\} \colon |l_x - k_x| \leq 4, |l_z - k_z| \leq 3 \big\}.$$

Recall that pixel l has the coordinates (l_x, l_z); the first coordinate being the horizontal distance from the left borehole and the second coordinate being the depth, both measured in pixels. To compute a perturbed image, the domain used in Algorithm 4 is defined as follows

$$\mathcal{D}_k = \big\{ l \in Z \setminus Z^{\text{cond}} \colon |l_x - k_x| \leq 7, |l_z - k_z| \leq 7 \big\}.$$

The values of all pixels $l \in \mathcal{D}_k$ will be re-simulated using Sequential Simulation conditioned to the remaining pixels $l \notin \mathcal{D}_k$. We are not using any hard data in the example, which means $Z^{\text{cond}} = \emptyset$.

This choice of template function yields $n = 34$ where the geometrical shape of the neighborhood of inner pixels is a 7 pixels by 5 pixels rectangle. This is chosen based



(a) $\alpha = 10^{-3}$. (b) $\alpha = 10^{-2}$.

(c) $\alpha = 10^{-1}$. (d) $\alpha = 10$.

Fig. 6 The computed models for increasing values of α: (**a**) $\alpha = 10^{-3}$, (**b**) $\alpha = 10^{-2}$, (**c**) $\alpha = 10^{-1}$, (**d**) $\alpha = 10$

on the trends in the training image, where the distance of continuity is larger horizontally than vertically. However, it should be noted that this choice of template function is not expected to meet the assumptions of conditional independence of Sect. 2.2. The distance of continuity in the training image appears much larger horizontally than only seven pixels, and vertically the width of the channels is approximately ten pixels. This implies that, despite matched frequency distributions, a computed solution will not necessarily be recognized to have the same visual structures as the training image. The goal is solve the inverse problem which involves fitting the data and therefore, as our example will show, neighborhoods of this size are sufficient. The data-fitting term of the objective function guides the solution method, such that the structures from the training image are correctly reproduced. The low number of neighbors constrains the small-scale variations, which are not well-determined by the travel time data. However, the travel time data successfully determine the large-scale structures. The template function does not need to describe structures of the largest scales of the training image as long as the observed data are of a certain quality.

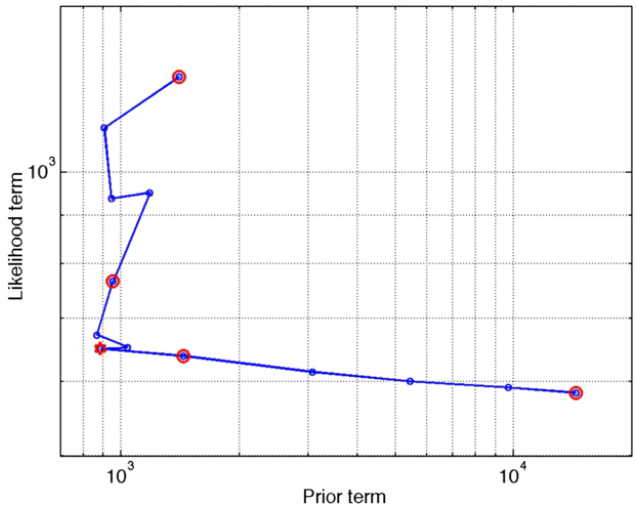


Fig. 7 L-curve used to determine the optimal α value. Models have been computed for 13 logarithmically distributed values of α ranging from 1 (*upper left corner*) to 10^{-3} (*lower right corner*). Each of the 13 models is marked with a blue circle. See the text for further explanation

Figure 4 shows the reference model that describes what is considered to be the true velocity profile between the two boreholes. The image has been generated by the *snesim* algorithm (Strebelle 2002) using the multiple point statistics of the training image. The arrival times $\mathbf{d}$ for the reference model $\mathbf{m}^{\text{ref}}$ are computed by a forward computation, $\mathbf{d} = G\mathbf{m}^{\text{ref}}$. We define the observed arrival times $\mathbf{d}^{\text{obs}}$ as the computed arrival times $\mathbf{d}$ added 5 % Gaussian noise. Figure 5 shows the solution computed using 15,000 iterations for $\alpha = 1.8 \times 10^{-2}$. The solution resembles the reference model to a high degree. The FM method detected the four channels; their location, width and curvature correspond to the reference model. The computations took approximately 33 minutes on a Macbook Pro 2.66 GHz Intel Core 2 Duo with 4 GB RAM.

Before elaborating on how the α value was determined, we present some of the models computed for different values of α. Figure 6 shows the computed models for four logarithmically distributed values of α between 10^{-3} and 10^{1}. It is seen how the model for lowest value of α is geologically unrealistic and does not reproduce the a priori expected structures from the training image as it primarily is a solution to the ill-posed, under-determined, data-fitting problem. As α increases, the channel structures of the training image are recognized in the computed models. However, for too large α values the solutions are dominated by the χ^2 term as the data have been deprioritized, and the solutions are not geologically reasonable either. As discussed, the chosen template is too small to satisfy the conditions from Sect. 2.2, yielding models that do in fact minimize the χ^2 distance, but do not reproduce the structures form the training image. The data misfit is now assigned too little weight to help compensate for the small neighborhoods, and the compromise between minimizing the data misfit and minimizing the dissimilarity that before worked out well is no longer present.

We propose to use the L-curve method (Hansen and O'Leary 1993) to determine an appropriate value of α. Figure 7 shows the value of $\chi^2(\mathbf{m}^{\text{FM}})$ versus the value of $\frac{1}{2}\|g(\mathbf{m}^{\text{FM}}) - \mathbf{d}^{\text{obs}}\|^2_{\mathbf{C}_\text{d}}$ for 13 models. The models have been computed for logarithmically distributed values of α ranging from 1 (upper left corner) to 10^{-3} (lower right corner). Each of the 13 models is marked with a blue circle. The models from Fig. 6

are furthermore marked with a red circle. The model from Fig. 5 is marked with a red star. We recognize the characteristic L-shaped behavior in the figure and the model from Fig. 5 is the model located in the corner of the L-curve. The corresponding value $\alpha = 1.8 \times 10^{-2}$ is therefore considered an appropriate value of α.

5 Conclusions

We have proposed the frequency matching method which enables us to quantify a probability density function that describes the multiple point statistics of an image. In this way, the maximum a posteriori solution to an inverse problem using training image-based complex prior information can be computed. The frequency matching method formulates a closed form expression for the a priori probability of a given model. This is obtained by comparing the multiple point statistics of the model to the multiple point statistics from a training image using a χ^2 dissimilarity distance.

Through a synthetic test case from crosshole tomography, we have demonstrated how the frequency matching method can be used to determine the maximum a posteriori solution. When the a priori distribution is used in inversion, a parameter α is required. We have shown how we are able to recreate the reference model by choosing this weighing parameter appropriately. Future work could focus on determining the theoretically optimal value of α as an alternative to using the L-curve method.

Acknowledgements The present work was sponsored by the Danish Council for Independent Research —Technology and Production Sciences (FTP grant no. 274-09-0332) and DONG Energy.

References

Arpat GB (2005) Sequential simulation with patterns. PhD thesis, Stanford University

Caers J, Hoffman T (2006) The probability perturbation method: a new look at Bayesian inverse modeling. Math Geol 38:81–100

Caers J, Zhang T (2004) Multiple-point geostatistics: a quantitative vehicle for integrating geologic analogs into multiple reservoir models. In: Grammer M, Harris PM, Eberli GP (eds) Integration of outcrop and modern analogs in reservoir modeling, AAPG Memoir 80, AAPG, Tulsa, pp 383–394

Cressie N, Davidson J (1998) Image analysis with partially ordered Markov models. Comput Stat Data Anal 29(1):1–26

Gómez-Hernández JJ (1991) A stochastic approach to the simulation of block conductivity fields conditioned upon data measured at a smaller scale. PhD thesis, Stanford University

Guardiano F, Srivastava RM (1993) Multivariate geostatistics: beyond bivariate moments. In: Geostatics-Troia, vol 1. Kluwer Academic, Dordrecht, pp 133–144

Hansen PC, O'Leary DP (1993) The use of the L-curve in the regularization of discrete ill-posed problems. SIAM J Sci Comput 14:1487–1503

Hansen TM, Cordua KS, Mosegaard K (2008) Using geostatistics to describe complex a priori information for inverse problems. In: Proceedings Geostats 2008. pp 329–338

Hansen TM, Cordua KS, Mosegaard K (2012) Inverse problems with non-trivial priors: efficient solution through sequential Gibbs sampling. Comput Geosci 16:593–611

Honarkhah M (2011) Stochastic simulation of patterns using distance-based pattern modeling. PhD dissertation, Stanford University

Jafarpour B, Khodabakhshi M (2011) A probability conditioning method (PCM) for nonlinear flow data integration into multipoint statistical facies simulation. Math Geosci 43:133–146

Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. Science 220:671–680

Peredo O, Ortiz JM (2010) Parallel implementation of simulated annealing to reproduce multiple-point statistics. Comput Geosci 37:1110–1121

Sheskin D (2004) Handbook of parametric and nonparametric statistical procedures. Chapman & Hal/CRC, London, pp 493–500

Stien M, Kolbjørnsen O (2011) Facies modeling using a Markov mesh model specification. Math Geosci 43:611–624

Strebelle S (2002) Conditional simulation of complex geological structures using multiple-point statistics. Math Geol 34:1–21

Suzuki S, Caers J (2008) A distance-based prior model parameterization for constraining solutions of spatial inverse problems. Math Geosci 40:445–469

Tarantola A (2005) Inverse problem theory and methods for model parameter estimation. Society for Industrial and Applied Mathematics, Philadelphia

Tarantola A, Valette B (1982) Inverse problems = quest for information. J Geophys 50:159–170

Tjelmeland H, Besag J (1998) Markov random fields with higher-order interactions. Scand J Stat 25:415–433

Tran TT (1994) Improving variogram reproduction on dense simulation grids. Comput Geosci 7:1161–1168

APPENDIX D

Paper IV

A Novel Approach for Combining Multiple-point Statistics and Production Data in Reservoir Characterization

Authors:
Yulia Melnikova, Katrine Lange, Jan Frydendall and Klaus Mosegaard

Published in:
Proceedings of *74th EAGE Conference & Exhibition Incorporating SPE EUROPEC 2012*
Copenhagen, Denmark
4-7 June 2012

Introduction

History matching problem arises regularly at the stage of reservoir development and its performance optimization. One wants to match simulation response with production data adjusting reservoir model parameters, e.g. permeability values, location of faults and fractures. Good history-matched model must possess two important properties: be able to match data observations within their uncertainty and be consistent with geological expectations, i.e. prior information. If most of the attention is paid to the minimization of the data misfit, the geological realism of the solution may suffer. Traditionally, prior information is conserved in the form of covariance model that is defined by two-point statistics. This means that spatial variability is accounted for only on a pairwise basis, and as a result, curvilinear long-correlated features become neglected.

With the development of data assimilation techniques, such as the Ensemble Kalman Filter (EnKF), matching the data and estimation of the uncertainty became a feasible task (Aanonsen et al. 2009). However, due to the mentioned traditional approach of incorporating prior information, geologically realistic features, such as channels are hardly ever reproduced. To keep complex geological structures in the solution, multiple-point statistics framework has to be used (Journel and Zhang 2006). Many authors suggest inferring complex spatial information from the so-called *training images* (Strebelle 2002, Caers 2003, Eskandaridalvand 2010). Training images contain expected geological features and can be constructed using geologists' expertise, database of characteristic structures, photographs of the outcrops.

In this study we also use multiple point statistics of training images to provide geological realism of the solution. This paper is the first application of the Frequency Matching (FM) method (Lange et al. 2011) to the solution of the history matching problem. The FM method allows us to accurately quantify the consistency of the model with complex prior, e.g. training image, computing prior probability of the model. Consequently, it enables us to guide the model safely by both prior information and data observations.

Description of the method can be found in the next section. It is followed by a 3D synthetic example and the conclusion.

Methodology

In this study we apply a multiple-point statistics framework defined by the FM method for solving history matching problem. Multiple point statistics gained its popularity in characterization of sedimentary reservoirs that possess channel-like features; see, for example, paper by Hoffman et al. (2006). The Probability perturbation method (PPM) suggested by Caers and Hoffman (2006) aims at finding solution that is consistent both with prior information, i.e. obtained from a training image, and data observations. However, as stated in Hansen et al. (2012), the PPM approach finds the solution that belongs to the space of prior models allowed by training image and only maximizes the data fit. While the FM method is a Bayesian approach and hence maximizes the posteriori model. The FM technique characterizes images by their multiple point statistics. To retrieve multiple point statistics. from an image, a scanning template is applied to it. Further, the scanned information is sorted and forms the frequency distribution of the image. In such a way, the image is uniquely described by the histogram of the multi-point spatial event. For comparing of two images, a dissimilarity measure is introduced, and defined as χ^2 distance between their histograms.

Generally, the FM method can be used for the solution of inverse problems, where one wants to estimate a model $\mathbf{m}$, given some data observations $\mathbf{d}_{\mathbf{obs}}$, with respect to a complex prior information in the form of a training image. Mathematically, this can be formulated as the following optimization problem:

$$\min_{\mathbf{m}} \left\| \mathbf{d}_{\mathbf{obs}} - g(\mathbf{m}) \right\|^2_{C_d} + \alpha \chi^2(\pi,\ \pi^{TI}) \tag{1}$$

where g is non-linear forward operator, C_d is data covariance matrix, π and π^{TI} are the frequency distributions of the test image and training image respectively and α is a weight parameter. The first term in equation (1) minimizes the difference between observations and the forward simulation

response, while the second term minimizes the discrepancy between statistics of the model and of the training image.

We solve the optimization problem (1) with a greedy stochastic annealing algorithm. First, we choose a proper scanning template and construct the histograms of the training image and the starting model. Then we conduct a random walk in model space, suggesting change for the values of voxels (pixels in 2D space). The change is accepted or rejected depending on the optimization method criteria. For the test case, described in the next section we used the "greedy" approach: if the suggested change decreased the value of the objective function (1) the change was accepted. The greedy stochastic annealing algorithm may get stock in local minima. However, it is more computationally efficient and provides sufficiently low values for the both terms in the objective function (1). Since the history matching problem is very much undetermined, we are satisfied with a solution that honours (with desirable accuracy) both data and prior information. The described iterative approach involves one forward simulation per iteration. This is a bottleneck of the method. However, the development of the FM method is an active topic of the research and the strategy for decreasing number of forward simulations is under investigation. For example, increasing amount of flipped blocks may improve the algorithm performance.

Example

In this section we test the Frequency Matching method on the 3D synthetic example. Let us consider a 3D synthetic oil reservoir of 25x25x3 cells. Physical dimension of a cell is 50x50x10 m. Wells configuration is a traditional nine-spot pattern with one water injector in the middle and 8 producers on the sides. We use a streamline simulator for modelling flow response. Initial water saturation is 0, initial pressure 100 bar.

As geological model, we use binary training image of size 60x60x3 with distinct narrow high permeable channels of 500 mD and shale background of 50 mD, as shown in **Figure 1** (all three layers are the same). It should be mentioned, that generally the FM method is suitable for the assessing priors with multiple categorical values. The reference permeability model, shown on **Figure 2**2 together with the well positions, just as the training image, is presented by two discrete values of 500 mD and 50 mD.

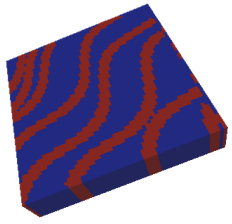


Figure 1 *Training image, 60x60x3 cells*

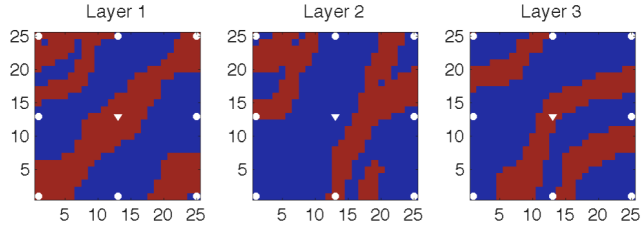


Figure 2 *Reference permeability 25x25x3 cells, layers 1-3*

Production history was generated applying forward simulation to the reference model and adding five percent Gaussian noise. Observations consist of 5 measurements of oil rate for each producer at 600, 1200, 1800, 2400 and 3000 days respectively. Note that we only gauge the cumulative production value of data at each well, and not at each segment of the wells. The corresponding data covariance matrix is a diagonal matrix with the values equal to the added noise variance.

The starting model consists of random combination of channel and shale facies. We follow the algorithm described in the previous section. It is worth saying that for computational efficiency only a 2D scanning template of 3x3 pixels was used. However, this choice is unlikely to have influence on the result as the reference channels have vertical continuity of one pixel. The value of the weight parameter α from equation (1) was chosen empirically to be equal to 1, however, it is clearly a question for the future research.

The solution model was obtained at the moment of approximately of 30000 iterations. Agreement with the prior is assessed by comparing the histograms of the solution and the training image in Figure 3: at first glance the histograms seems the same, however, there are still some discrepancies. With a larger template or with more iteration we may have improved more on the result. Analysis of the solution (**Figure 4**) shows that some of the features, in comparison to the reference model, were successfully reproduced, for example, the diagonal connection in the first layer. Dissimilarities may be explained by the low amount of the data values, which is unavoidable when dealing with the ill-posed inverse problem.

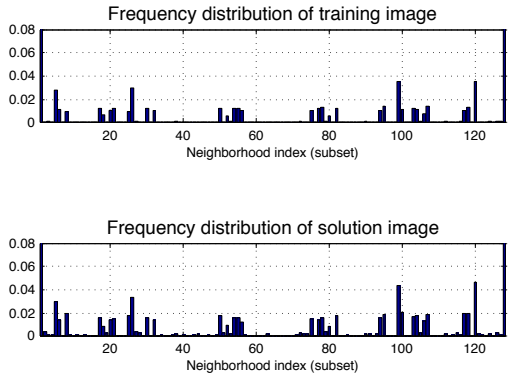


Figure 3 *Histograms of the training image (upper) and the model (lower) at the moment of 30000 iterations.*

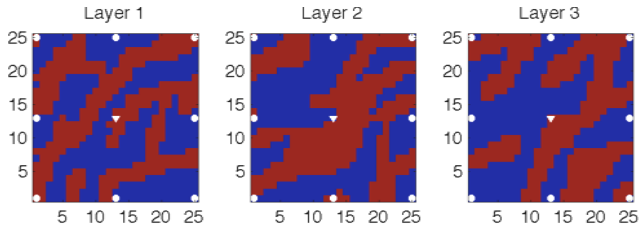


Figure 4 *Permeability model at the moment of 30000 iterations*

As for the quality of the data match, we can infer from **Figure 5** and Figure 6 that most of the wells were matched within their uncertainty. The legend is following: red solid line - data with noise, red dashed line - data without noise, blue line with diamond marker – data from the solution.

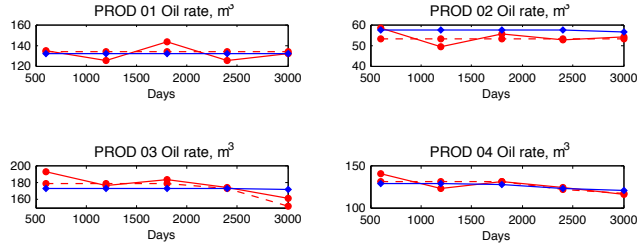


Figure 5 *Oil production rate for wells 1-4.*

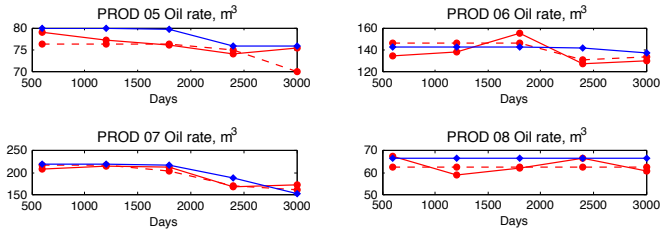


Figure 6 *Oil production rate for wells 4-8.*

Conclusions

We demonstrated a new multiple-points statistics framework for finding geology-consistent solution of history matching problem. We used the Frequency Matching method that allows us to combine prior information based on training image and production data. The 3D synthetic test case showed that the obtained solution was consistent both with prior information and data observations. This demonstrated the potential of the method and suggests that it could be used on more complicated cases. Future work will be related on improving of computational efficiency

References

Aanonsen, S.I., Nævdal, G., Oliver, D.S., Reynolds, A.C., and Vallès, B. [2009] The ensemble Kalman filter in reservoir engineering – a review. *SPE Journal*, **14**(3), 393-412.

Caers, J. [2003] History matching under a training image-based geological model constraint. *SPE Journal*, **8**(3), 218–226.

Caers, J. and Hoffman T. [2006] The probability perturbation method: A new look at bayesian inverse modeling. *Mathematical Geology,* **38**(1), 81–100.

Eskandaridalvand K. and Srinivasan S. [2010] Reservoir modelling of complex geological systems – a multiple-point perspective, *Journal of Canadian Petroleum Technology*, **49**(8), 59-68.

Hansen, T. M., Cordua, K. S., and Mosegaard, K., [2012] Inverse problems with non-trivial priors - Efficient solution through Sequential Gibbs Sampling. *Computational Geosciences*, doi:10.1007/s10596-011-9271-1

Hoffman, T., Caers, J., Wen, X. and Strebelle, S. [2006] A Practical Data-Integration Approach to History Matching: Application to a Deepwater Reservoir. *SPE Journal* **11**(4), 464-479.

Journel, A. and Zhang T. [2006] The necessity of a multiple-point prior model. *Mathematical Geology*, **38**(5), 591-610.

Lange, K., Cordua, K. S., Frydendall, J., Hansen, T. M. and Mosegaard, K. [2011] A frequency matching method for generation of a priori sample models from training images, *IAMG 2011*, Extended Abstract.

Strebelle, S. [2002] Conditional simulation of complex geological structures using multiple-point geostatistics. *Mathematical Geology*, **34** (1), 1–22.

APPENDIX E

Paper V

Improving Multiple-Point-Based a Priori Models for Inverse Problems by Combining Sequential Simulation with the Frequency Matching Method

Authors:
Knud Skou Cordua, Thomas Mejer Hansen, Katrine Lange, Jan Frydendall and Klaus Mosegaard

Published in:
Proceedings of *Society of Exploration Geophysicists 2012 (SEG 2012)*
Las Vegas, Nevada USA
4-9 November 2012

Improving multiple-point-based a priori models for inverse problems by combining Sequential Simulation with the Frequency Matching Method

Knud S. Cordua, Thomas M. Hansen, Katrine Lange, Jan Frydendall, Klaus Mosegaard, Technical University of Denmark, Department of Informatics and Mathematical Modelling.*

Summary

In order to move beyond simplified covariance based a priori models, which are typically used for inverse problems, more complex multiple-point-based a priori models have to be considered. By means of marginal probability distributions 'learned' from a training image, sequential simulation has proven to be an efficient way of obtaining multiple realizations that honor the same multiple-point statistics as the training image. The frequency matching method provides an alternative way of formulating multiple-point-based a priori models. In this strategy the pattern frequency distributions (i.e. marginals) of the training image and a subsurface model are matched in order to obtain a solution with the same multiple-point statistics as the training image. Sequential Gibbs sampling is a simulation strategy that provides an efficient way of applying sequential simulation based algorithms as a priori information in probabilistic inverse problems. Unfortunately, when this strategy is applied with the multiple-point-based simulation algorithm SNESIM the reproducibility of training image patterns is violated. In this study we suggest to combine sequential simulation with the frequency matching method in order to improve the pattern reproducibility while maintaining the efficiency of the sequential Gibbs sampling strategy. We compare realizations of three types of a priori models. Finally, the results are exemplified through crosshole travel time tomography.

Introduction

In geostatistical and probabilistic inverse modeling, a priori models that describe the expectations of the spatial distribution of the geological structures under study are important (Journel and Zhang, 2006). Traditionally, a priori models rely on two-point statistics described through covariance models. However, such a priori models cannot capture realistically geological curvilinear structures such as tortuous channels. In order to overcome this shortcoming, multiple-point statistics has to be introduced (Guardiano and Srivastava, 1993). The Single Normal Equation SIMulation (SNESIM) algorithm is a computationally very efficient way of obtaining realizations from a joint probability density function (pdf) based on multiple-point statistics learned from a training image using sequential simulation (Strebelle, 2002).

The extended Metropolis algorithm is a general sampling algorithm that can be used to sample the solution to nonlinear inverse problems (Mosegaard and Tarantola, 1995). The extended Metropolis algorithm demands an algorithm that is able to produce perturbations between realizations from the a priori model. An efficient way of obtaining this is through sequential Gibbs sampling (Hansen et al., 2012). The extended Metropolis algorithm has previously been used in conjunction with sequential Gibbs sampling for a priori information defined through the SNESIM algorithm to sample the solution of a tomographic full waveform inverse problem (Cordua et al., 2012).

An alternative way of defining the multiple-point-based a priori pdf is the Frequency Matching Method (FMM) (Lange et al., 2011). In this approach the frequency distributions of patterns (i.e. marginal probabilities) counted in a given solution to the subsurface and in the training image are compared. By means of the Chi-square statistics, Lange et al. (2011) quantified the match between frequency distributions. In this way, they were able to jointly optimize for the a priori expectations and a tomographic dataset. Here, we define a FMM-based a priori pdf using the Dirichlet probability distribution. We show the results of sampling this distribution using the Metropolis algorithm.

When sequential Gibbs sampling is applied with the SNESIM algorithm, the reproducibility of the spatial continuity seen in the training image is reduced. This is caused by the conditional simulation technique inhered in SNESIM, which reduces the number of conditional data events when inconsistencies (i.e. singularities) occurs during the simulation. These effects are reduced for full unconditional SNESIM realizations, but are evident for the iterative perturbation strategy performed by the sequential Gibbs sampling. We suggest an a priori pdf that combines the SNESIM and FMM based a priori pdfs in order to overcome these shortcomings. We show that realizations from the combined a priori pdf ensures better reproducibility of spatial structures found in the training image than compared to the individual SNESIM and FMM-based a priori pdfs, respectively.

The importance of the reproducibility when solving inverse problems is demonstrated through a crosshole travel time tomographic inverse problem. The solution to this nonlinear inverse problem is sampled using the extended Metropolis

algorithm with both the SNESIM and the combined SNESIM-FMM-based a priori pdfs, respectively.

Methodology

Consider that the subsurface can be represented by a discrete set of model parameters $\mathbf{m}$. In geophysical inverse problems, information about the unknown model parameters is retrieved based on a set of indirect observations $\mathbf{d}$ (e.g. travel time data), a theoretical forward problem that relates model parameters and the data, and some a priori information on the model parameters. The forward relation between the model parameters and the data can be expressed as (e.g. Tarantola, 2005):

$$\mathbf{d} = g(\mathbf{m}), \tag{1}$$

where g is a linear or nonlinear function that often relies on a physical law. In this study equation 1 is a nonlinear relation that provides a set of travel time data at the receiver positions given a 2D velocity field. The forward relation is based on ray-theory and is calculated using the Eikonal equation (Zelt and Barton, 1998).

In a probabilistic formulation, the solution to the inverse problem is given as an a posteriori probability density over the model parameters (e.g. Tarantola, 2005):

$$\sigma_M(\mathbf{m}) = k\rho_M(\mathbf{m})L(\mathbf{m}), \tag{2}$$

where k is a normalization constant, $\rho_M(\mathbf{m})$ is the a priori pdf, and $L(\mathbf{m})$ is the likelihood function. $\rho_M(\mathbf{m})$ describes the probability that the model satisfies the a priori information. $L(\mathbf{m})$ describes how well the modeled data explains the observed data given a data uncertainty. Hence, the a posteriori probability density describes the combined states of information provided by the data and the a priori information.

The extended Metropolis algorithm

The extended Metropolis algorithm can be used to sample the a posteriori probability density of a general nonlinear inverse problem as formulated in equation 2. This algorithm only requires: 1) A "black box" algorithm that is able to produce perturbations between realizations from the a priori pdf. 2) An algorithm that is able to compute the likelihood for a given set of model parameters. The extended Metropolis algorithm contains the following steps:

1) The exploration step:
An a priori sampler proposes a realization, $\mathbf{m}_{propose}$, from the a priori pdf. $\mathbf{m}_{propose}$ is a perturbation of a current realization, $\mathbf{m}_{current}$.

2) The exploitation step:
The proposed realization is accepted with the probability:

$$P_{accept} = \min\left(1, \frac{L(\mathbf{m}_{propose})}{L(\mathbf{m}_{current})}\right) \tag{3}$$

If the proposed model is accepted, $\mathbf{m}_{propose}$ becomes $\mathbf{m}_{current}$, otherwise $\mathbf{m}_{current}$ counts again.

The above procedure is continued until a desirable number of realizations have been accepted. Together, all the accepted realizations constitute a sample of the a posteriori probability density (Mosegaard and Tarantola, 1995).

Sequential Gibbs sampling

Sequential Gibbs sampling is a computationally efficient way to sample complex a priori models as quantified by most geostatistical simulation algorithms, such as for example the SNESIM algorithm (Hansen et al., 2012). With sequential Gibbs sampling the degree of perturbation between realizations can be controlled. In this way, a priori information quantified by geostatistical simulation algorithms serve as a "black box" algorithm that can be applied with the extended Metropolis algorithm to sample the solution for probabilistic inverse problems.

The flow of sequential Gibbs sampling is:
1) A current unconditional realization of the a priori pdf is provided.
2) A subset of the model parameters in the current realization is randomly chosen.
3) The model parameters within this subset are resimulated using sequential simulation conditional to the remaining model parameters (using e.g. the SNESIM algorithm).
4) Step (2) and (3) of this procedure are repeated in order to obtain multiple realizations of the a priori pdf.

The size of the subset of model parameters to be resimulated is chosen subjectively and controls the explorations nature of the Metropolis algorithm. For large subsets the exploration step becomes large and the probability of accept (in equation 3) decreases. On the other hand, smaller exploration steps leads to a higher accept probability. However, a small exploration step causes successive accepted realizations of the Metropolis algorithm to become statistically more dependent and, hence, more realizations have to be accepted to obtain statistically independent realizations. For more details on this topic see Hansen et al. (2012) and Cordua et al. (2012).

The frequency matching method

Multiple-point sample algorithms rely on sequential simulation, which is based on the fact that the complete joint probability density can be factorized by conditional

probability densities. The conditional probability densities can (according to the product rule) be expressed by means of marginal probability densities. These "marginals" are extracted (or learned) from the training image by simply counting the number of times a certain pattern occurs in image. The number of pixels within the patterns is fixed and determined by a template. The marginal pdf obtained in this way can be viewed as a frequency distribution (i.e. a normalized histogram), which is the same as the content of the search tree, as referred to by Strebelle (2002).

In the frequency matching method (Lange et al., 2011) the multiple-point-based a priori pdf is quantified by measuring the degree of fit between the frequency distribution of the training image and a current realization. In this way it becomes possible to actually quantify the multiple-point a priori pdf, which is not possible using the SNESIM algorithm.

Here, we defined the frequency matching measure using the Dirichlet pdf, which is different from the approach of Lange et al. (2011):

$$\rho_{FMM}(\mathbf{m}) = \frac{N^{cur}!}{H_1^{cur}!,...,H_K^{cur}!}\prod_{k=1}^{K}\left(\frac{H_k^{TI}+H_k^{prior}}{N^{TI}+N^{prior}}\right)^{H_k^{cur}}, \quad (4)$$

where H_k^{cur} is the number of counts in the k'th bin of the (unnormalized) histogram obtained from a current realization $\mathbf{m}$. H_k^{TI} is the number of counts in the k'th bin of the (unnormalized) histogram obtained from training image. $K = c^T$ is the number of possible pattern combinations, which is function of the template size T and the number of categories c. Further, we have that:

$$N^{cur} = \sum_{k=1}^{K} H_k^{cur} \quad (5)$$

$$N^{TI} = \sum_{k=1}^{K} H_k^{TI} \quad (6)$$

$$N^{prior} = \sum_{k=1}^{K} H_k^{prior} \quad (7)$$

where H_k^{prior} is the k'th bin of the a priori (unnormalized) histogram, which represents the a priori expectation of the histogram related to underlying process before the training image histogram is observed. Hence, H_k^{prior} can be used to quantify the degree of expected match between the frequency distributions of a current subsurface image and the training image. For small values of H_k^{prior} the current model is expected to match the training image frequency distribution better than for large values. Note that the Dirichlet distribution only needs to be evaluated for the bins $k \in \{j \mid H_j^{cur} \neq 0\}$. All other bins do not contribute to the probability. Hence, the histograms becomes sparse, which, in particular, saves memory for large template sizes and/or many categories of the model parameter values.

Combining FMM with the SNESIM algorithm

Figure 2 shows realizations from the SNESIM-based priori model using the sequential Gibbs sample strategy. Figure 3 shows realizations from the Dirichlet (i.e. FMM-based) a priori probability distribution. The multiple-point statistics of these a priori models is obtained from the training image seen in figure 1. By comparing figure 2 and 3 with the training image it is obvious that the continuous structures seen in the training image are not very well reproduced.
In order to improve this, we suggest combining the FMM with the SNESIM algorithm such that we obtained an a priori pdf defined as:

$$\rho_M(\mathbf{m}) = \rho_{SNESIM}(\mathbf{m})\rho_{FMM}(\mathbf{m}) \quad (8)$$

This a priori pdf can efficiently be sampled using the extended Metropolis algorithm in conjunction with sequential Gibbs sampling. By substituting $\rho_{FMM}(\mathbf{m})$ with the likelihood function $L(\mathbf{m})$ in equation (2) and (3), realizations from the combined a priori in equation 8 can be obtained. Note that, in this way, the value of $\rho_{SNESIM}(\mathbf{m})$ does not need to be evaluated.

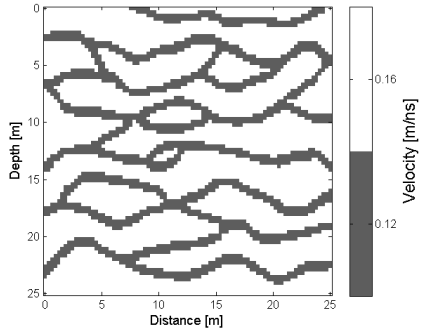


Figure 1. Training image used for obtaining the multiple-point a priori statistics.

Results

Figure 4 shows realizations obtained from the combined a priori model defined in equation 8. In this study we choose the a priori histogram to be a homogenous distribution with $H_k^{prior} = 5, \quad k \in \{j \mid H_j^{cur} \neq 0\}$ and a template size of 3

pixels x 3 pixels. The results demonstrate that the combined FMM-SNESIM-based a priori probability density recovers the structures of the training image better than compared to both the SNESIM and FMM-based a priori pdfs.

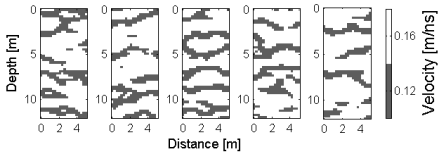


Figure 2. Realizations from the SNESIM a priori model using sequential Gibbs sampling.

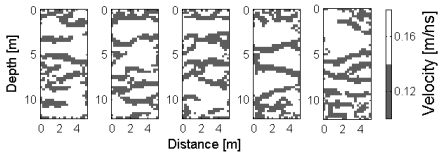


Figure 3. Realizations of the Dirichlet pdf (i.e. FFM-based a priori pdf) using the Metropolis algorithm with a homogenous proposal pdf.

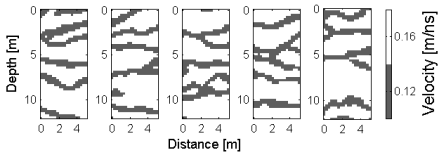


Figure 4. Realizations from the combined SNESIM-FMM-based a priori pdf using the extended Metropolis algorithm in conjunction with sequential Gibbs sampling.

Crosshole travel time tomography

In order to demonstrate how the different a priori models influence the solution to a nonlinear inverse problem, we consider a crosshole ground penetrating radar tomographic inverse problem (see e.g. Cordua et al., 2009). A synthetic reference model, from which a synthetic data set is obtained, is seen in figure 5. This model is a fully unconditional realization of the SNESIM based a priori pdf. A zero mean uncorrelated Gaussian noise component with a standard deviation of 1 ns (~2.7 % of the signal) is added to the data. The likelihood function is a Gaussian pdf that takes into account the statistics of the noise. The result of the inversion is seen in figure 6 and 7. It is clear that the improved FMM-SNESIM-based a priori probability density provides realizations that resemble the reference model better than when using the SNESIM-based a priori pdf. Moreover, the variability between the individual realizations becomes smaller when considering the combined a priori model. This shows that the improved a priori information improves the resolution of the solution.

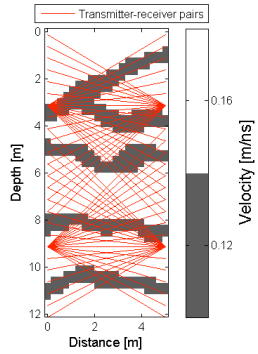


Figure 5. Reference model used for travel time tomography. The red rays give an indication of the data coverage.

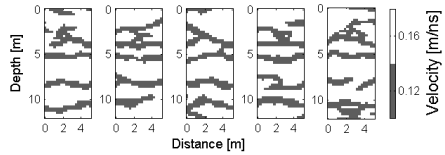


Figure 6. Realizations from the a posteriori pdf with a priori information defined by SNESIM using sequential Gibbs sampling.

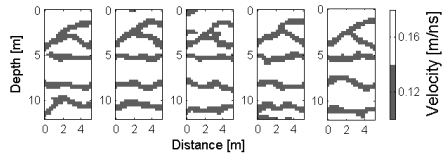


Figure 7. Realizations from the a posteriori pdf based on the combined SNESIM-FMM a priori pdf using sequential Gibbs sampling.

Discussion and Conclusion

We have demonstrated the potential of combining the FMM with the sequential simulation strategy provided by SNESIM. In this way, realizations obtained when using sequential Gibbs sampling reproduces the spatial structures of the training image much better then when only considering SNESIM. At the same time, the suggested strategy ensures that the computationally efficiency of sequential simulation is maintained.

The combined SNESIM-FMM-based a priori model demonstrates to improve the resolution when applied for a tomographic nonlinear inverse problem.

References

Cordua, K. S., T. M. Hansen, and K. Mosegaard, 2012, Monte Carlo full-waveform inversion of crosshole GPR data using multiple-point geostatistical a priori information: Geophysics, **77**, H19 – H31.

Cordua, K. S., L. Nielsen, M. C. Looms, T. M. Hansen, and A. Binley, 2009, Quantifying the influence of static-like errors in least-squares-based inversion and sequential simulation of cross-borehole ground penetrating radar data: Journal of Applied Geophysics, **68**, 71 – 84.

Hansen, T. M., K. S. Cordua, and K. Mosegaard, 2012, Inverse problems with non-trivial priors: Efficient solution through Sequential Gibbs Sampling: Computational Geosciences, DOI: 10.1007/s10596-011-9271-1.

Journel, A. and T. Zhang, 2006, The Necessity of a Multiple-Point Prior Model: Mathematical Geology, **38**, 591 – 610.

Lange, K., J. Frydendall, K. S. Cordua, T. M. Hansen, Y. Melnikova, and K. Mosegaard, 2011, A Frequency Matching Method: Solving Inverse Problems by use of Geologically Realistic Prior Information: IAMG 2011, Salzburg, Austria.

Mosegaard, K., and A. Tarantola, 1995, Monte Carlo sampling of solutions to inverse problems: Journal of geophysical research, **100**, no. B7, 431 – 447.

Strebelle, S., 2002, Conditional simulation of complex geological structures using multiple-point statistics: Mathematical Geology, **34**, 1 – 21.

Tarantola, A., 2005, Inverse problem theory and methods for model parameter estimation: Society of Industrial and Applies Mathematics, Philadelphia, PA., 353pp.

Zelt, C., and P. Barton, 1998, Three-dimensional seismic refraction seismic refraction tomography - a comparison of two methods applied to data from the Faeroe Basin: Journal of Geophysical Research, **103**, no. B4, 7187 – 7210.

Guardiano, F., and R. Srivastava, 1993, Multivariate geostatistics: Beyond bivariate moments, in A. Soares, ed., Geostatistics Tróia '92, v. 1, Kluwer, 133 – 144.

APPENDIX F

Paper VI

An Implementation of the Frequency Matching Method

Authors:
Katrine Lange, Jan Frydendall, Thomas Mejer Hansen, Andrea Zunino and Klaus Mosegaard

Published in:
DTU Compute Technical Report-2013-09

DTU Compute Technical Report-2013-09

An Implementation of the Frequency Matching Method

Katrine Lange*, Jan Frydendall, Thomas Mejer Hansen,
Andrea Zunino, Klaus Mosegaard

DTU Compute, Technical University of Denmark,
Matematiktorvet, Building 303B, 2800 Kongens Lyngby, Denmark
Center of Energy Resources Engineering, Technical University of Denmark,
Søltofts Plads, 2800 Kongens Lyngby, Denmark

Abstract

During the last decade multiple-point statistics has become increasingly popular as a tool for incorporating complex prior information when solving inverse problems in geosciences. A variety of methods have been proposed but often the implementation of these is not straightforward. One of these methods is the recently proposed Frequency Matching method to compute the maximum a posteriori model of an inverse problem where multiple-point statistics, learned from a training image, is used to formulate a closed form expression for an a priori probability density function.

This paper discusses aspects of the implementation of the Frequency Matching method and the techniques adopted to make it computationally feasible also for large-scale inverse problems. The source code is publicly available at GitHub and this paper also provides an example of how to apply the Frequency Matching method to a linear inverse problem.

Keywords: multiple-points statistics, training image, a priori information, maximum a posteriori model

*Corresponding author, email address: katla@dtu.dk

1 Introduction to the Frequency Matching Method

The Frequency Matching (FM) method defines the maximum a posteriori model of an inverse problem using multiple-point statistics learned from a training image (TI) as prior information. Inverse problems having such type of a priori information arise in scientific fields involving modelling of unknown parameters describing spatial properties. They are typical in the geosciences where, for instance, a property of the subsurface of the Earth should be modelled. The available data is often scarce, resulting in severely underdetermined inverse problem. This makes use of a priori information even more beneficiary.

A priori information is often available as expectations to the subsurface showing certain structures, and when modelling spatial properties these structures are provided by so-called training images. Models of parameters describing spatial properties are often referred to as images, letting the colours of the image illustrate the values of the property they are describing.

The Frequency Matching method was introduced by Lange et al. (2012), to which we refer for a detailed motivation for the method and discussion of the general use of multiple-point statistics when solving inverse problems in the geosciences. The present paper describes a Fortran implementation of the FM method with the purpose of making other users able to use the Fortran version of the FM method on their respective problems.

1.1 Probabilistic Inverse Problem Theory

The FM method defines the maximum a posteriori model for an inverse problem, i.e., the model with maximum a posteriori probability, using multiple-points statistics learned from a training image as a priori information. To do so it formulates a closed form expression of the a priori probability density function, which is based on a distance measure, $c(\mathbf{m}, \mathbf{m}^{TI})$, from a model or image $\mathbf{m}$ to a training image $\mathbf{m}^{TI}$. The distance measure expresses how dissimilar the multiple-point statistics of the images are. Models with multiple-point statistics similar to the multiple-point statistics of the training image have short distances to the training image and they are therefore assigned high probabilities. Likewise models with dissimilar multiple-point statistics will have large distances to the training image and they will there-

fore be assigned low probabilities.

The inverse problem is formulated using probabilistic inverse theory (Tarantola, 2005). The data misfit gives rise to the likelihood of a model. Assuming Gaussian measurement noise of the data observations $\mathbf{d}^{\text{obs}}$ with mean zero and covariance matrix $\mathbf{C_d}$, the likelihood function L is defined as:

$$L(\mathbf{m}) \;=\; \text{const}\exp\left(-\frac{1}{2}\|\mathbf{d}^{\text{obs}} - g(\mathbf{m})\|^2_{\mathbf{C_d}}\right),$$

where g is the mapping operator from model space to data space and const is a constant.

Using the distance function c the FM method defines a closed form expression for the a priori probability density function ρ of a model as:

$$\rho(\mathbf{m}) \;=\; \text{const}\exp\left(-\alpha\ c(\mathbf{m}, \mathbf{m}^{TI})\right). \tag{1}$$

According to probabilistic inverse problem theory the posterior probability density function σ is proportional to the product of the prior probability density function and the likelihood function:

$$\sigma(\mathbf{m}) \;=\; \text{const}\ \rho(\mathbf{m})\ L(\mathbf{m}). \tag{2}$$

Specifically using the a priori probability density function from equation (1), the FM method then defines the solution to the inverse problem, $\mathbf{m}^{\text{FM}}$, as the model maximizing the a posteriori probability density function from equation (2)

$$\begin{aligned}
\mathbf{m}^{\text{FM}} &= \underset{\mathbf{m}}{\text{argmax}}\,\{\sigma(\mathbf{m})\} \\
&= \underset{\mathbf{m}}{\text{argmin}}\,\{-\log\sigma(\mathbf{m})\} \\
&= \underset{\mathbf{m}}{\text{argmin}}\left\{\frac{1}{2}\|\mathbf{d}^{\text{obs}} - g(\mathbf{m})\|^2_{\mathbf{C_d}} + \alpha\ c(\mathbf{m}, \mathbf{m}^{TI})\right\}.
\end{aligned} \tag{3}$$

The maximisation and minimisation is done over the set of all valid models $\mathbf{m}$. In this set each model parameter (i.e., each element of the model vector $\mathbf{m}$) belongs to a predefined problem specific set of discrete values taking into account hard data constraints. This means the frequency matching model $\mathbf{m}^{\text{FM}}$ is the solution to a combinatorial optimization problem. The FM method does not dictate how this optimization problem should be solved.

1.2 Formulating A Priori Information

The multiple-point statistics of an image is represented by what the Frequency Matching method defines as the frequency distribution. This is a histogram of the counts of the different patterns found in the image. The patterns, if their size is chosen wisely, are assumed to describe the multiple-point statistics of the image that we seek to reproduce.

Assume the multiple-point statistics extracted from the training image can be expressed as patterns of identical geometric shape consisting of $n+1$ voxels. This implies that a voxel is assumed statistically independent of all voxels except the surrounding n voxels together with which the voxel forms a pattern. The entire training image is scanned for patterns and the count of appearances for each pattern is collected. These counts constitute the frequency distribution. Let $\boldsymbol{\pi}^{\mathrm{TI}}$ and $\boldsymbol{\pi}$ be the frequency distributions of the training image and a model image, respectively.

This section briefly defines the dissimilarity function c used in the closed form expression of the a priori probability density function from equation (1). The current choice of dissimilarity function has roots in the statistical literature regarding the chi-square test for homogeneity (Sheskin, 2004).

Let m be the number of categories of voxels in the image then there exist m^{n+1} different patterns. A majority of these will have the count of 0 as they do not appear in the image. Given the frequency distributions of an image, $\boldsymbol{\pi}$, and of a training image, $\boldsymbol{\pi}^{\mathrm{TI}}$ the dissimilarity function value of the image is defined as follows:

$$c(\mathbf{m}, \mathbf{m}^{TI}) = \chi^2(\boldsymbol{\pi}, \boldsymbol{\pi}^{\mathrm{TI}}) \;=\; \sum_{i \in I} \frac{(\pi_i^{\mathrm{TI}} - \epsilon_i^{\mathrm{TI}})^2}{\epsilon_i^{\mathrm{TI}}} + \sum_{i \in I} \frac{(\pi_i - \epsilon_i)^2}{\epsilon_i}, \qquad (4)$$

where the set I consists of indices of patterns that occur in either the image or the training image. ϵ_i denotes the count of the underlying distribution of patterns with the ith pattern value for images of the same size as the image, and ϵ_i^{TI} denotes the counts of the underlying distribution of patterns with the ith pattern value for images of the same size as the training image. These counts are computed as:

$$\begin{aligned} \epsilon_i &= \frac{\pi_i + \pi_i^{\mathrm{TI}}}{n^{\mathcal{Z}} + n^{\mathrm{TI}}}\, n^{\mathcal{Z}}, \\ \epsilon_i^{\mathrm{TI}} &= \frac{\pi_i + \pi_i^{\mathrm{TI}}}{n^{\mathcal{Z}} + n^{\mathrm{TI}}}\, n^{\mathrm{TI}}, \end{aligned}$$

where $n^{\mathcal{Z}}$ and n^{TI} are the total number of counts of patterns in the frequency distributions of the image and the training image.

2 The Implementation

2.1 Assumptions of the Implementation

The FM method itself has no limitations regarding non-linearity but the current implementation assumes that the inverse problem is linear:

$$\mathbf{Gm} \quad = \quad \mathbf{d}^{\mathrm{obs}}, \tag{5}$$

Here $\mathbf{G}$ is a known system matrix, $\mathbf{d}^{\mathrm{obs}}$ is a set of observed data values and $\mathbf{m}$ is the model parameters to be determined.

The FM assumes that these model parameters can take only a limited number of categorical values. Often the model parameters are binary. This can for instance be the case when modelling the flow of the subsurface. The model parameters are then either 0, which represents zones with high permeability and therefore easy flow, or 1, which represents low-permeable zones. Let $\mathsf{sV} + 1$ be the number of categories voxel values can belong to, i.e., for a binary image $\mathsf{sV} = 1$. Per definition the voxel values of the images are $0, \ldots, \mathsf{sV}$.

The FM model is defined as the minimiser of the negative logarithm of the posterior probability density function as given by (3). The minimization is in the current implementation performed using simulated annealing. We will not go into details about the choice of this optimization method or the method itself, but for more information on simulated annealing see Kirkpatrick et al. (1983). Pseudo code for applying simulated annealing to the FM is available in Lange et al. (2012).

2.2 Overview of the Procedures

Figure 1 shows an overview of the most important interactions among the Fortran procedures in the implementation of the FM method. For now we will provide a short walk-through of the procedures. A description of what each of them do is provided in A.

The implementation is based on the FMM procedure which primary function is to set up and initialize all the inputs for the FM method and the

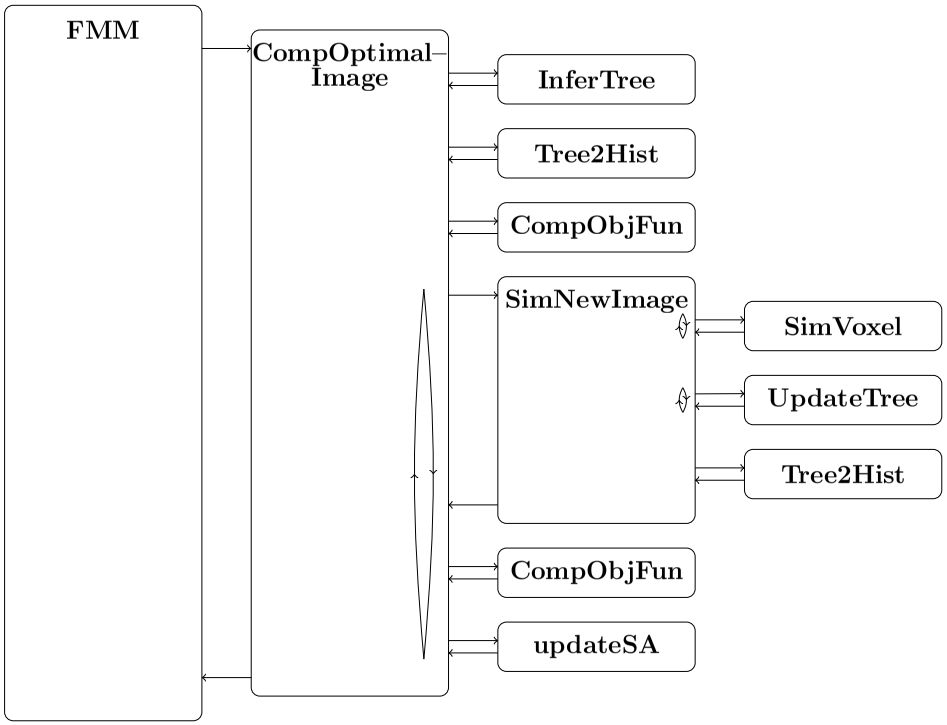


Figure 1: Overall structure of the FM implementation.

simulated annealing scheme. The FMM procedure also has the task of extracting multiple-point statistics of the training image and representing it using the—for that purpose designed—tree structure. (The tree structure will be explained in details in section .) The procedure also generates the frequency distribution. It is the only procedure that uses the training image itself, as it passes on only the tree and its frequency distribution.

The simulated annealing is implemented in the CompOptimalImage procedure. Provided with an initial model, the first thing the CompOptimalImage procedure does is to extract the multiple-point statistics of the initial model (InferTree) and compute its frequency distribution (Tree2Hist). It then computes the objective function value of the initial model (CompObjFun). The CompObjFun procedure calls two other procedures, CompChiDist and CompDataFit, to compute each of the two terms of the objective function.

Vertical arrows in Figure 1 represent a loop, and in the CompOptimalImage procedure it is used to loop through the iterations of the simulated annealing algorithm.

For each iteration a perturbed image is generated by SimNewImage. This is done by erasing the voxel values in a part of the image and then resimulating them using sequential simulation (Guardiano and Srivastava, 1993) with the multiple-point statistics learned from the training image (SimVoxel). Each time a voxel has been re-simulated the tree must be updated to fit the new image (UpdateTree). Here again the vertical arrows represents loops indicating that these two tasks are done voxel by voxel.

Afterwards, the frequency distribution of the perturbed image is computed (again Tree2Hist). The objective function value of the perturbed image is then computed (again CompObjFun) and finally, the perturbed model is possibly accepted and the variables updated accordingly (UpdateSA).

The procedures in the implementation that have been left out of the diagram are auxiliary procedures mostly related to operations on trees. The auxiliary procedures are described in B .

2.3 Compiling the Program

The program is written following the Fortran 2008 standard, hence a compiler complying with this standard is needed to run the software. We have used the open source GFortran-version 4.6-compiler from GNU to compile and run the code. To simplify the compilation and linking process a Makefile to be used with the GNU make program is provided.

It has been tested successfully on Mac (10.6 and 10.8) as well as GNU/Linux.

2.4 The Inverse Problem

The FMM procedure takes among other inputs the parameters from Eq.(5). For that we have defined a structure called inverseproblem. It contains the following parameters:

G: 2D array with the system matrix $\mathbf{G}$.

dobs: 1D array with the vector of data observations $\mathbf{d}^{\text{obs}}$.

invCov: 2D array with the inverse of the covariance matrix, $\mathbf{C}_{\mathbf{d}}^{-1}$.

cat: 1D array with sV+1 elements, one for each category of voxel values. This is used to transform the images with categorical voxel values to models with physical parameter values. The model parameter associated with a voxel with value i will be assigned the value cat(i + 1) in order to compute the data fit.

The parameters specifying the inverse problem are passed between procedures in the variable InvProb. InvProb can be used to specify any arbitrary linear problem. The implementation can therefore be applied to inverse problems also in other scientific fields outside of the geosciences.

2.5 Specifying Neighbourhoods

We distinguish between two types of voxels in an image: inner voxels and non-inner voxels. Inner voxels are those that are sufficiently far from the boundaries to have enough neighbouring voxels to make a pattern. Non-inner voxels are the rest, i.e., those that are close to the boundary and therefore do not have as many neighbouring voxels around them. How far away from the boundary a voxel has to be in order to be an inner voxel depends on how the neighbourhood of voxels are defined.

Several parameters are needed to specify the dimensions of neighbourhoods and these are collected in a special type of structure called NeighborMask. It contains the following parameters:

mat: 3D integer array of the shape of a neighbourhood of an inner voxel. Voxels in positions where the element of mat is 1 are included in the neighbourhood, and voxels where the corresponding element is 0 are not included. The value corresponding to the center voxel does not matter.

nc, mc, pc: integers denoting the coordinates of the center voxel in the mat array. (The origin of a pattern is its bottom left upper corner.)

n, m, p: size of the patterns, i.e., the mat array includes n × m × p voxels.

nodes: 2D integer array with relative coordinates from the center voxel to each of its neighbours. That means, given the image coordinates of a voxel, nodes can be used to compute the image coordinates of all of the neighbours of the voxel. The array has a row for each neighbour in a pattern and three columns that holds the coordinates in each dimension.

Neighbourhoods can be defined in two ways: 1) If the patterns have the simple shape of a hyper-rectangle, the user can simply specify the size of the patterns, m, n and p. These must be odd numbers and the center voxel of the pattern is then assumed to be directly in the middle. This is the most common type of pattern to use. 2) If the patters are not that simple, the user can directly specify the mat array and its center coordinate (nc, mc, pc). The size of the mat array do not have to be odd, and the center can be located anywhere.

The parameter sN is used throughout the procedures as the number of voxels in a pattern excluding the center voxel. This means that nodes has sN rows. In the procedures the structure specifying the neighbourhoods is called Nmask.

Figure 2 shows a tiny example of a training image and a neighbourhood that could be used to describe the multiple-point statistics of it. To define this non-rectangular neighbourhood the user will need to specify the mat array:

$$\mathsf{Nmask\%mat} \;=\; \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix},$$

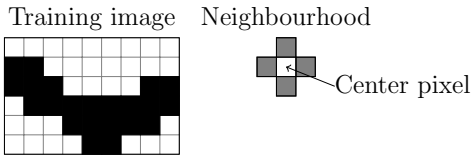


Figure 2: Tiny training image used for illustration purposes. A neighbourhood is chosen to be the (up to) four nearest neighbours as shown on the right of the figure. The pixels within the neighbourhood are numbered according to their row, column and then layer index. How to define this neighbourhood is explained in the text. The resulting tree can be seen in Figure 4.

.

and its center coordinates:

$$\begin{aligned} \mathsf{Nmask\%nc} &= 2, \\ \mathsf{Nmask\%mc} &= 2, \\ \mathsf{Nmask\%pc} &= 1. \end{aligned}$$

Then the FMM procedure derives the size of the patterns:

$$\begin{aligned} \mathsf{Nmask\%n} &= 3, \\ \mathsf{Nmask\%m} &= 3, \\ \mathsf{Nmask\%p} &= 1, \end{aligned}$$

leading to the following relative row, column and layer coordinates of neighbours:

$$\mathsf{Nmask\%nodes} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

Keep in mind, that even for 2D images the third dimension does exist, it will just have the size one.

2.6 The Tree Structure

A tree is a complex structure and most vital for the computational feasibility of the implementation. The purpose of the tree is to store the patterns

of a training image and their counts, and then to use them to describe the multiple-point statistics of the training image.

The tree is implemented as a so-called linked list. This is the same approach that was applied in the SNESIM algorithm (Strebelle, 2002) when creating search trees in order to overcome the problems of generating and storing large data bases of patterns. In our implementation trees are furthermore used to easily generate frequency distributions, which is the actual input to the dissimilarity function c from Equation (4).

We like to think of trees as consisting of a set of nodes and edges as a tree in the mathematical sense of a graph. A tree, T, is based on a root node, and it is often that node we pass along the different procedures. From this root node we can navigate deeper into the tree via links, which in this case are pointers to the next nodes.

A node of a tree is defined as a structure `streenode` containing three variables:

depth: integer defining how deep into the tree this node is located.

repl: real array with $\mathsf{sV}+1$ elements. The array holds the count of patterns in the image that have a certain partial pattern and center value $0, 1, \ldots, \mathsf{sV}$ respectively. The partial pattern is dependent on the depth of the root, and the deeper into the tree a node is located, the more voxels of the patterns are used in the partial pattern.

next: array of pointers with $\mathsf{sV}+1$ elements, these are the links to nodes placed a level deeper into the tree.

Figure 3 shows the structure defined to describe a node. A tree of an image is constructed by first creating the root node and then adding new nodes as the image is being scanned and new patterns found. Assigning the root node depth level 0, the maximum depth of a tree is sN, and at any depth i there can maximum be $(\mathsf{sV}+1)^i$ nodes.

Let T denote the root node of a tree; this node contains information of the center values themselves, not including any neighbouring voxel values. We define the $\mathsf{T\%repl}$ array to hold the unscaled distribution of voxel values of inner voxels in the image, so simply the counts of how many inner voxels in the image have each of the values $0, 1, \ldots, \mathsf{sV}$. $\mathsf{T\%repl(k)}$ holds the count of voxels with value $k-1$.

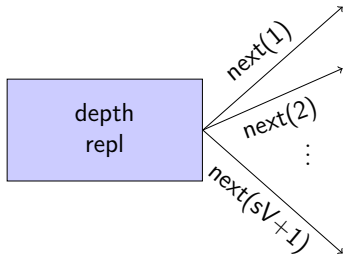


Figure 3: Illustration of a node of a tree as represented by the streenode structure.

The pointers in the array T%next point to sV+1 new nodes of the tree. These nodes are at depth level 1 and they therefore contain information about center voxels taking into consideration the value of their first neighbouring voxel. (The neighboring voxels are ordered according to their voxel index.) As the first neighbouring voxel can have $sV+1$ different values we need the same number of pointers to cover all cases. This means, the pointer T%next(i) points to the node representing all partial patterns, where the first neighbouring voxel has value $i-1$.

The unscaled distribution for all values of center voxels will be stored in the repl array of that node. It can be accessed by T%next(i)%repl. This means the element T%next(i)%repl(k) holds the counts of patterns in the image where the first neighbouring voxel has the value $i-1$ and the center voxel has value $k-1$. In the same manner, the jth pointer of this node, T%next(i)%next(j), points to the node of patterns where the values of the two first neighbouring voxels are $i-1$ and $j-1$, respectively. The node holds the unscaled conditional probability distribution of the value of a center voxel given these specific values of the first two neighbouring voxels. This is repeated until depth level sN, where all sN neighbouring voxels have been included in the partial structure, that each node represents.

To sum up, an arbitrary node T provides us the following information:

T%depth: depth level in the tree where the node is placed.

T%repl(k): count of a specific partial pattern in the image with center value $k-1$. The partial pattern is unknown to this node, but the values of

the first T%depth neighbouring voxels are given by the location of the node in the tree.

T%next(i) pointer to the node with the same partial pattern as the current and where the T%depth+1th voxel has value $i - 1$.

Notice how a node T cannot give us any information about the partial patterns it represents. From a node we can only extract information that are deeper into the tree and not information that belong to a previous level.

The frequency distribution of an image is the (unscaled) distribution of patterns. As the bottom level of the tree contains exactly the counts of patterns with all of the possible combination of center and neighbourhood voxel values, the frequency distribution can simply be constructed by combining all existing repl arrays at depth level sN.

Recall that Figure 2 shows an example of a training image and an example of a neighbourhood. This is a very tiny training image and neighbourhood chosen for illustration purposes only. The resulting tree is seen in Figure 4. The frequency distribution of the image is constructed by extracting all 4th level repl arrays of the tree.

On each node is written the values of its repl array, and on each edge is written the colour of the neighbouring voxel represented by the current depth level. Black voxels are assigned the value 0 and white voxels have the value 1.

To compare an image to a training image a tree describing its multiple-point statistics must be derived and its frequency distribution determined. This allows for the evaluation of c. However, constructing the tree is done in a different manner than for a training image itself. When constructing the tree we take advantage of the fact that the dissimilarity function c depends only on the patterns of the image that also appear in the training image. We therefore generate not the tree containing all patterns found in the image but only those patterns that are also found in the training image. That means, the tree of the image will have the same shape (same nodes and edges) as the tree of the training image. This makes the frequency distribution consisting of all bottom level repl arrays directly comparable to the frequency distribution of the training image, as they, element by element, describe the count of identical patterns in the two images.

Non-inner voxels of an image that is not a training image are treated slightly different than non-inner voxels of a training image. They contribute

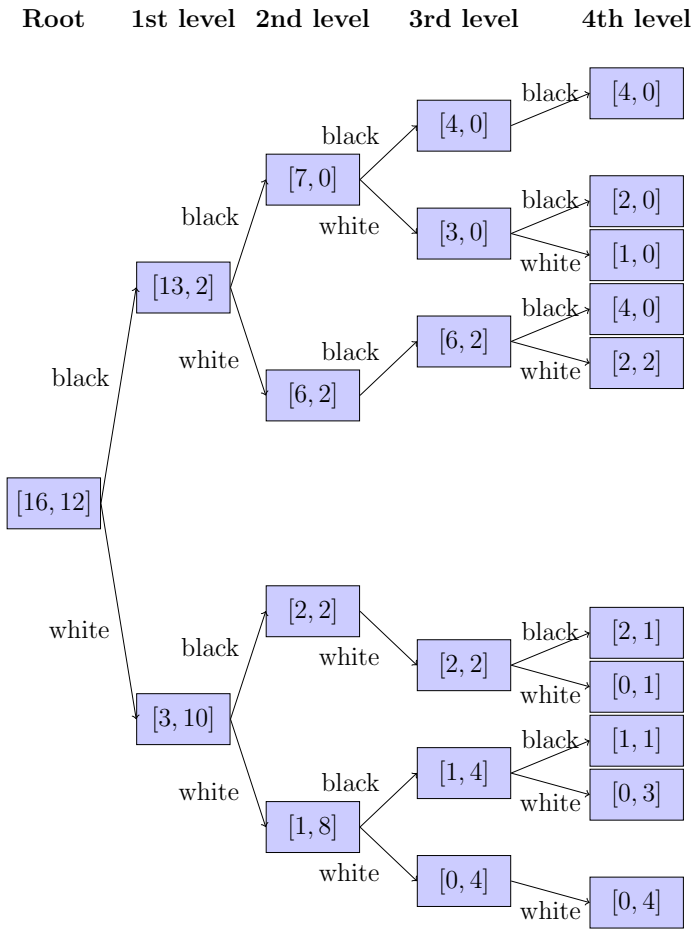


Figure 4: The tree structure of the tiny training image in Figure 2 using a neighbourhood consisting of the four closest voxels. We assign black voxels the value of 0 and white voxels the value of 1. Notice how the count of each pattern is represented in the tree. For instance, the number of black pixels with all black neighbouring pixels is 4. This is seen by starting in the root node, following all edges labelled black until reaching the bottom level, and then accessing the first element of the repl array. Also notice how for every single node, if you sum the repl arrays of the nodes it is pointing to, you get the repl array of the node itself. This means all repl arrays only hold counts of inner pixel.

to the tree like inner voxels, but they are typically represented by multiple patterns. They contribute with a total count of 1, like the inner voxels, and the count for each type of pattern is proportional to their marginal conditional probability density from the tree of the training image.

2.7 Perturbation Domain

The optimization problem defining the FM model from Equation 3 is solved using an iterative solution method that searches through the model space. The search is carried out by visiting new images that are defined as perturbations of current images. They are created by perturbed the voxel values in a certain domain of an image.

A set of parameters is needed to specify the size of the domain of an image that then needs to be erased and re-simulated to create a perturbed image. The domain is assumed to be hyper-rectangular. To define it we have the DomainMask structure that contains the following parameters:

n, m, p: integers, dimensions of the domain to be re-simulated. These must be odd numbers.

nodes: 2D integer array with relative coordinates from the center voxel of the domain to each other voxel in the domain.

mat: 1D integer array with sN elements. This array holds a distance from the center voxel of a neighbourhood to each of its neighbouring voxels. It is used in the re-simulation to determine on which voxels the simulation should be conditioned.

The domain structure is Dblock. Like for the neighbourhood mask the user does not need to specify most of its parameters. In fact, one should only decide on the dimensions of the block, Dblock%n, Dblock%m, Dblock%p, and the FMM procedure then constructs the remaining. For the distance array is used the L_1-norm.

2.8 Optimization Options

Simulated annealing is used as the solution method to the optimization problem defining the $\mathbf{m}^{\mathrm{FM}}$, and to hold the parameters used by the algorithm

we have the type option. It holds the following parameters:
t0: real number, the initial temperature.

tmin: real number, the final temperature.

maxIter: integer, maximum number of iterations allowed to be used per voxel parameter.

runs: integer, number of times to run the simulated annealing algorithm.

multigrid: integer, number of multigrids to use.

condopt: logical, in case of multiple grids used, it determines if the solution on a fine grid should be conditioned on the optimal solution from the coarser grid (condopt = .true.) or not (condopt = .false.).

The simulated annealing uses an exponential cooling rate that is calculated such that the maximum number of iterations allowed is exactly the number of iterations used. The implementation of the FM method has been prepared for multiple grid simulation but these are not yet implemented. This can be done by implementing a loop in the FMM procedure such that CompOptimalImage will be called with different grids. Also in some cases it can be beneficiary to restart the simulated annealing algorithm and although this option has not yet been implemented the code has been prepared. A loop can be inserted in the CompOptimalImage procedure so that the simulated annealing scheme is run multiple times.

3 Example: Crosshole Travel Time Tomography

As an example of how to use the Frequency Matching method we will show how to solve a synthetic crosshole travel time tomography problem similar to the one described in Lange et al. (2012). Crosshole travel time tomography involves the measurement of seismic travel times between two or more boreholes in order to determine an image of seismic velocities in the intervening subsurface. Seismic energy is released from sources located in one borehole and recorded at multiple receiver locations in another borehole.

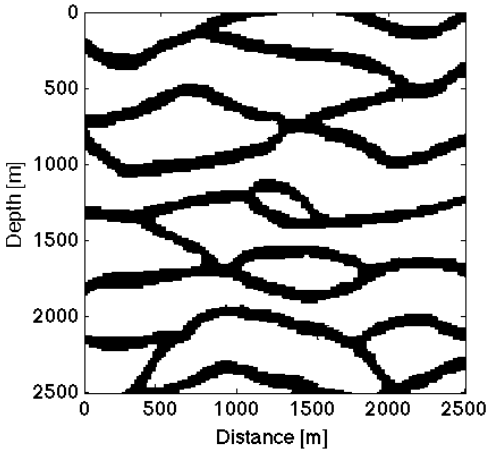
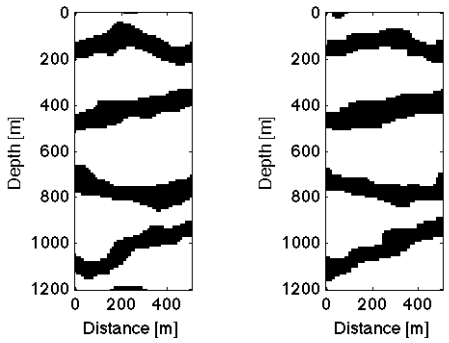


Figure 5: Training image (size: 250 by 250 pixels).

In this way a dense tomographic data set that covers the interborehole region is obtained.

We create a synthetic test case based on a setup with two vertical boreholes. The horizontal distance between them is 500 meters and they each have a depth of 500 meters. The two-dimensional vertical domain between the boreholes is divided into 120 times 50 quadratic cells. The seismic velocity is assumed constant within each cell. The model parameters of the problem are these propagation speeds, meaning the problem has 6000 unknown model parameters. The observed data is the recorded first arrival times from the seismic signals. In each borehole are placed 12 equally distributed sources and 48 equally distributed receivers. We assume a linear relation between the data observations and the model parameters. The sensitivity of a seismic signal is simulated as straight rays.

It is assumed that the inter-borehole region consists of a background with slow propagation speed and a horizontal channel structure of zones of high propagation speed. The speeds are chosen as 1600 meters per second and 2000 meters per second, respectively. The a priori knowledge of the channel structure is assumed described by the training image in Figure 5. We have chosen to let the neighbourhood of a pixel consist of its 36 closest neighbours



(a) Reference model. (b) FM model.

Figure 6: Reference model for the synthetic crosshole travel time tomography example and its computed FM model. The size of the models ares 120 by 50 pixels.

specified by:

$$\mathsf{Nmask\%mat} = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}.$$

A reference model is generated based on the training image using the SNESIM (Strebelle, 2002) algorithm. The first arrival times of the reference model is simulated. These are then added 5% relative Gaussian noise and assumed to be the observed data. Figure 6a shows the reference model.

The forward problem Voxels belonging to the background are assigned to the category 0 and voxels belonging to the zones of high propagation speed are assigned the category 1. The forward problem is linear in the inverse of

the propagation speeds, i.e., the physical values of the voxel values of the two categories are specified as cat $= [1/1600, 1/2000]$.

The observed data and the coefficent matrix from the forward problem is generated using MATLAB. They have then been stored in a text file that can be read into Fortran using the standard read routine and then saved in the parameters dobs and Gmat, respectively. The problem has 1152 data observations and 6000 model parameters so it is severely under-determined.

The % noise added to the reference model is independent and has estimated standard deviation $\hat{\sigma} = 1.9 \cdot 10^{-2}$. This yields the data covariance matrix $\mathbf{C_d} = \hat{\sigma}^2 I$, where I is the identity matrix. The inverse data covariance matrix invCov is then defined as the inverse of this.

The prior term The weighting constant multiplied to the prior term in Equation (3) is chosen as $\alpha = 10^{-1}$, which means alpha $= 10^{-2}$.

Perturbation of images The domain of voxels to be re-simulated when creating a perturbed image is chosen as nD $= 13$, mD $= 13$ and pD $= 1$.

Optimisation parameters The cooling rate is defined by the starting temperature t0 $= 10^2$ and the minimum temperature tmin $= 10^{-5}$. The simulated annealing algorithm is allowed to use iter $= 0.5$ for each of the 6000 pixels in the image.

Computing the optimal model using the allowed 3000 iterations took approximately 16 minutes on a Macbook Pro 2.66 GHz equipped with an Intel Core 2 Duo processor and 4 GB of RAM.

The computed optimal model is shown next to the reference model in Figure 6. It is seen how it correctly locates the channels. The width and curvature of the channels also clearly resembles those of the reference model. We therefore conclude that the choice of weighting constant alpha and the optimisation parameters were suitable for the problem at hand. The example successfully illustrates how the Fortran implementation of the Frequency Matching method can be applied.

4 Bibliography

Guardiano, F., Srivastava, R. M., 1993. Multivariate geostatistics: Beyond bivariate moments. Geostat-Troia 1, 133–144.

Kirkpatrick, S., Gelatt, C. D., Vecchi, M. P., 1983. Optimization by simulated annealing. Science 220 (4598), 671–680.

Lange, K., Frydendall, J., Cordua, K. S., Hansen, T. M., Melnikova, Y., Mosegaard, K., 2012. A frequency matching method: Solving inverse problems by use of geologically realistic prior information. Mathematical Geosciences, 1–2110.1007/s11004-012-9417-2.
URL http://dx.doi.org/10.1007/s11004-012-9417-2

Sheskin, D., 2004. Handbook of Parametric and Nonparametric Statistical Procedures. Chapman & Hal/CRC, pp. 493–500.

Strebelle, S., 2002. Conditional simulation of complex geological structures using multiple-point statistics. Mathematical Geology 34, 1–21.

Tarantola, A., 2005. Inverse Problem Theory and Methods for Model Parameter Estimation. SIAM.

List of Appendices

A Primary Procedures

The following is a list of the primary Fortran procedures in the FM implementation. The purpose of each procedure is briefly explained, and any non-trivial or otherwise interesting details in the implementation are discussed. Also a list of input and/or output variables is provided. The lists hold the variable name, a short description of its use and its type.

To see which procedures call others we refer to Figure 1 and the discussing of it in the text. The procedures are listed in the order they are called which also appears from the figure. Auxiliary procedures are listed in B.

A.1 FMM

This procedure acts as an intermediary between the user specified input parameters and the implemented FM method. It reads the multiple-points statistics from the training image and sets up all the necessary inputs for the simulated annealing scheme based on the user inputs.

The procedure calls the CompOptimalImage procedure to compute the FM model (3). This model along with its frequency distribution, the frequency distribution of the training image and other parameters of special interest are written to files. These can later to loaded into for instance MATLAB to visualize the results.

In case the code is modified to handle multiple grids this would be a suitable procedure in which to loop over the grid levels, and for each level set up the corresponding tree of the training image and do the conversion from coarse to fine grid before calling the CompOptimalImage.

Variable	Description	Type
Z0	Initial image used as starting image for the iterative solution method. This image should satisfy hard data constraints, if any.	3D integer array
Ztrain	Training image.	3D integer array

Zcond	This array has the same dimensions as the image Z0, and it is used to state if any of the voxels should satisfy hard data constraints. If there are no hard data constraints the array should be all false. If some voxels are only allowed to take on a specific value the corresponding element of Zcond should be true.	3D logical array
Nmask	Contains the parameters that define neighbourhoods.	structure
sV	Parameter, the images contain the sV + 1 categories $0, 1, \ldots,$ sV of voxel values.	integer
InvProb	Parameters specifying the linear inverse problem.	structure
alpha	Weighting parameter α^2 of the prior term in the objective function.	real
Dblock	Contains the parameters controlling the perturbation of an image.	structure
options	Parameters for the simulated annealing scheme.	structure
solutions	Structure holding the results from the simulated annealing. These are written to files.	structure
Ps	Holds the values of each of the two terms in the objective function for each iteration of the simulated annealing algorithm.	2D real array

A.2 InferTrainTree

This procedure generates the tree, Ttrain, describing the multiple-point statistics of a training image, Ztrain. Patterns are extracted one by one from the training image and added to the tree.

Variable	Description	Type
Ztrain	Training image.	3D integer array
nodes	Array from the Nmask structure.	2D integer array

sV	Parameter, the images contain the sV + 1 categories $0, 1, \ldots,$ sV of voxel values.	integer
sN	Parameter, number of neighbours for an inner voxel resulting in patterns consisting of sN + 1 voxels.	integer
Ttrain	Tree of patterns extracted from the training image.	tree

A.3 Tree2Hist

This procedure constructs the frequency distribution (or the histogram), H, of an image given its tree, T. The frequency distribution is a two dimensional array with sV+1 rows and a column for each combination of voxel values in a neighbourhood. The ith row has the count of the appearances of the different neighbourhoods with center voxel having the value $i-1$.

This format has the advantage that each column of the frequency distribution is the unscaled conditional probability distribution of the value of center voxel given the values of its neighbouring voxels. This makes the conditional distributions easily assessable; they are used, for instance, for re-simulating voxel values. The disadvantage is that we might store more zero elements than necessary although no more than sV times too many.

Variable	**Description**	**Type**
sV	Parameter, the images contain the sV + 1 categories $0, 1, \ldots,$ sV of voxel values.	integer
sN	Parameter, number of neighbours for an inner voxel resulting in patterns consisting of sN + 1 voxels.	integer
T	Tree of patterns extracted from an image Z.	tree
H	Frequency distribution of the image Z.	2D real array

A.4 CompOptimalImage

This procedure is the most central in the implementation of the FM method as it is the one that solves the inverse problem by use of the simulated annealing algorithm. It takes multiple inputs from the starting image and the multiple-point statistics learned from the training image to the FM parameters specifying the neighbourhoods and the parameters associated with simulating perturbed images. It returns the computed optimal image, $\mathbf{m}^{\mathrm{FM}}$, as well as its frequency distribution. To check the convergence of the simulated annealing algorithm it also returns the objective function values for all iterations.

Variable	Description	Type
Z0	Initial image used as starting image for the iterative solution method. This image should satisfy hard data constraints, if any.	3D integer array
Zcond	This array has the same dimensions as the image Z0, and it is used to state if any of the voxels should satisfy hard data constraints. If there are no hard data constraints the array should be all false. If some voxels are only allowed to take on a specific value the corresponding element of Zcond should be true.	3D logical array
Ttrain	Tree of patterns extracted from the training image.	tree
Htrain	Frequency distribution of the training image.	2D real array
nodes	Array from the Nmask structure.	2D integer array
sV	Parameter, the images contain the $\mathsf{sV}+1$ categories $0, 1, \ldots, \mathsf{sV}$ of voxel values.	integer
sN	Parameter, number of neighbours for an inner voxel resulting in patterns consisting of $\mathsf{sN}+1$ voxels.	integer
InvProb	Parameters specifying the linear inverse problem.	structure
alpha	Weighting parameter α^2 of the prior term in the objective function.	real

Dblock	Contains the parameters controlling the perturbation of an image.	structure
options	Parameters for the simulated annealing scheme.	structure
Zopt	The optimal image, the $\mathbf{m}^{\mathrm{FM}}$, computed by the simulated annealing algorithm.	3D integer array
Hopt	Frequency distribution of the optimal image Zopt.	2D real array
Ps	Holds the values of each of the two terms in the objective function for each iteration of the simulated annealing algorithm.	2D real array

A.5 InferTree

This procedure infers the tree of an image so that its multiple-point statistics can be compared to those of a training image. It takes as input the tree of the training image that the image should be compared to. This is necessary as the tree should only contain patters also found in the training image. Once the image has been scanned and all patterns that should be stored has been added to the tree, the procedure shapeTree is called, to ensure that the tree of the image has the same shape as the tree of the training image, which is needed in order to easily compare their frequency distributions.

Variable	Description	Type
Z	Image.	3D integer array
nodes	Array from the Nmask structure.	2D integer array
Ttrain	Tree of patterns extracted from the training image.	tree
sV	Parameter, the images contain the $\mathsf{sV}+1$ categories $0, 1, \ldots, \mathsf{sV}$ of voxel values.	integer
sN	Parameter, number of neighbours for an inner voxel resulting in patterns consisting of $\mathsf{sN}+1$ voxels.	integer

T	Tree of patterns extracted from the image Z.	tree
Zex	This array is the same size as the image Z and it is used to indicate which voxels of the image are center of patterns found also in the training image. It is used by the procedure getNewIt to propose where an image should be perturbed.	3D logical array

A.6 CompObjFun

This procedure computes the values of each of the terms in the objective function from Equation (3). To be able to track convergence of the simulated annealing algorithm the terms are not added but instead the procedure returns a two-element array P such that:

$$\begin{aligned} \mathsf{P}(1) &= \frac{1}{2}\|\mathbf{d}^{\text{obs}} - \mathbf{G}\mathbf{m}\|^2_{\mathbf{C_d}}, \\ \mathsf{P}(2) &= \alpha^2 c(\mathbf{m}, \mathbf{m}^{TI}). \end{aligned}$$

The dissimilarity function c is evaluated by the CompChiDist procedure and the data misfit is computed by the CompDataFit procedure.

In case the inverse problem is not linear the CompDataFit procedure needs to be replaced by an implementation of the non-linear forward mapping.

Variable	Description	Type
H	Frequency distribution of the image Z.	2D real array
Htrain	Frequency distribution of the training image.	2D real array
N	Number of voxels in the image Z.	integer
Z	Image.	3D integer array
InvProb	Parameters specifying the linear inverse problem.	structure
alpha	Weighting parameter α^2 of the prior term in the objective function.	real
P	Two-element array that holds the values of each of the terms in the objective function.	1D real array

A.7 CompChiDist

This procedure is called by CompObjFun and computes the dissimilarity of an image compared to a training image by computing the distance between their frequency distributions.

Variable	Description	Type
H	Frequency distribution of the image Z.	2D real array
Htrain	Frequency distribution of the training image.	2D real array
N	Number of voxels in the image Z.	integer
X	Value of the dissimilarity function of the two frequency distributions.	real

A.8 CompDataFit

This procedure is called by CompObjFun and computes the data misfit of a model.

Variable	Description	Type
Z	Image.	3D integer array
InvProb	Parameters specifying the linear inverse problem.	structure
L	Value of the data misfit of for the image	real

A.9 SimNewImage

This procedure is used to perturb images. Provided with the current image from the simulated annealing iteration, Ztest, it will return a new, perturbed image, Znew. Znew is generated by erasing the values of a subset of the voxels and then re-simulating them using sequential simulation conditioned on the voxel values of the remaining part of the image.

This procedure takes as input the row, column and layer index of a voxel. The voxel is the center of a domain where voxel values are erased. The values of the voxels in the domain are then re-simulated one by one conditioned on the values of all voxels outside the domain and the already re-simulated values inside the domain. Of course voxels that should satisfy hard data constraints are not allowed to be changed and these are therefore not erased and re-simulated. Their values are kept and instead used to condition the re-simulation on.

The re-simulated values voxels are stored separately. The perturbed image is initially set identical to the original image. Its voxel values are then changed one at a time until all the voxels in the re-simulated domain has been updated. This allows the tree of the original image to be updated to the tree of the new, perturbed image. Tnew is initially set to be an exact copy of Ttest, and it is then updated iteratively for every voxel that has been assigned a new value. This way of iteratively updating the tree is much cheaper than generating the tree of the perturbed image from scratch.

Once the perturbation of the image and the updating of its tree is completed the frequency distribution can be computed from the tree. SimNewImage of course only updates the tree and recomputes the frequency distribution if the perturbed image is in fact different from the original image.

Variable	Description	Type
Ztest	Current image.	3D integer array

Zcond	Array used to specify which voxels are subject to hard data constraints. The values of such voxels are known and the voxels are used to conditioned upon in the simulation of other voxel values. The simulation algorithm is not allowed to erase and re-simulate the values of voxels that are conditioned upon. Instead these values are considered known and should be used when re-simulating other voxel values.	3D logical array
Zex	This array is the same size as the image Ztest and it is used to keep track of if the pattern that a voxel is the center of exists anywhere in the training image or not. It is used by the procedure getNewIt.	3D logical array
Ttest	Tree of patterns in the image Ztest.	tree
Ttrain	Tree of patterns extracted from the training image.	tree
i, j, k	Indices in the image of the voxel that is chosen such that the image is perturbed by erasing and the re-simulating the values of voxels in a domain around it.	integers
nodes	Array from the Nmask structure.	2D integer array
sV	Parameter, the images contain the $\mathsf{sV}+1$ categories $0, 1, \ldots, \mathsf{sV}$ of voxel values.	integer
sN	Parameter, number of neighbours for an inner voxel resulting in patterns consisting of $\mathsf{sN}+1$ voxels.	integer
Dblock	Contains the parameters controlling the perturbation of an image.	structure
Znew	Perturbed image based on the current image Ztest.	3D integer array
Zexnew	Similar to Zex, this array denotes which voxels in the image Znew are centres of patterns also found in the training image.	3D logical array
Tnew	Tree of patterns in the image Znew.	tree
Hnew	Frequency distribution of the perturbed image Znew.	2D real array

newImage	Indicates whether the perturbed image Znew is different from the original image Ztest (newImage = .true.) or not (newImage = .false.).	logical

A.10 SimVoxel

This procedure is needed to simulate the value of a voxel. It extracts from Ttrain the (unscaled) conditional probability distribution of the value of a center voxel given the values of the neighbouring voxels. It returns this conditional probability distribution and the voxel can then be assigned a value drawn from it.

In case the multiple-point statistics of Ttrain does not allow for this occurrence of the values of the neighbourhood voxels, i.e., no patterns of the training image matches the partial pattern, the voxels will be dropped one by one from the conditioning until a conditional probability distribution can be extracted. It is always the voxel furthest away from the center voxel that will be dropped. Dropped voxels are assigned the value -1 and they then appear as unknown.

If all neighbouring voxels are dropped the unconditioned distribution of voxel values in the training image will be used as the unscaled probability distribution of voxels also in the image.

Variable	Description	Type
Zvec	Holds the values of voxels in a neighbourhood. Unknown voxel values are assigned a value of -1.	1D integer array
Ttrain	Tree of patterns extracted from the training image.	tree
sV	Parameter, the images contain the sV + 1 categories $0, 1, \ldots,$ sV of voxel values.	integer
sN	Parameter, number of neighbours for an inner voxel resulting in patterns consisting of sN + 1 voxels.	integer
D	Part of the Dblock structure. Holds the distances from each of the neighbouring voxel to the center voxel.	1D integer array

hc	Holds the unscaled probability distribution of the value of the center voxel conditioned on its neighbouring voxels.	1D real array

A.11 UpdateSA

The point of this procedure is solely to simplify the code. It copies an image and its associated variables into another set of variables. This is for instance used in the simulated annealing scheme when a new image is accepted. Then the procedure is used to copy the new image Znew into the variable of the current image Ztest and to update its associated variables such that they contain the variables Tnew, Hnew etc. The updating of the variables is trivial except for the tree. The updating of the tree is done by the CopyTree procedure.

Variable	Description	Type
sV	Parameter, the images contain the $\mathsf{sV}+1$ categories $0, 1, \ldots, \mathsf{sV}$ of voxel values.	integer
sN	Parameter, number of neighbours for an inner voxel resulting in patterns consisting of $\mathsf{sN}+1$ voxels.	integer
Zin	The image that should be copied.	3D integer array
Zinex	Array to determine which patterns in Zin exist in the training image.	3D logical array
Tin	Tree of the image Zin.	tree
Hin	Frequency distribution of the image Zin.	2D real array
Pin	Array with the value of each term of the objective function of the image Zin.	1D real array
Zout	The image variable into which Zin should be copied.	3D integer array
Zexout	Array to determine which patterns in Zout exist in the training image.	3D logical array
Tout	Variable that holds the tree of the image Zout. This variable should be updated to hold Tin.	tree

Hout	Variable that holds the frequency distribution associated with the image Zout, this variable should be updated to hold Hin.	2D real array
Pout	Array with the values of the terms of the objective function associated with the image Zout, this variable should be updated to hold Pin.	1D real array

B Auxiliary Procedures

The following list describes the auxiliary procedures of the FM implementation. These are not of great importance to understand the code but necessary building blocks.

B.1 getNewIt

The procedure determines which part of an image should be re-simulated in order to compute a perturbed image. It returns the row, column and layer index of the voxel that is the center of the domain, which should be re-simulated.. To reduce the number of iterations needed for the simulated annealing algorithm to converge, we wish to choose a voxel that will result in maximal change to the perturbed image.

Consider two different voxels in the image. Assume that in an area around the first voxel the image looks very similar to the training image. Erasing the voxel values in a domain around in this area and re-simulating them based on the multiple-point statistics of the training image will then likely result in a perturbed image that is very similar to the original image. There is no reason to expect the perturbed image to be a significantly better solution to the inverse problem, than the image was before perturbing it.

Now assume that the original image in an area around the second voxel looks very different from anything seen in the training image. Erasing and re-simulating the voxel values in a domain centred in the second voxel will then create a very different perturbed image. This perturbed image is much more likely to be a better solution to the inverse problem as it will fit the multiple-points statistics of the training image better. In case is not a better data fit, it could potentially belong to a different, unexplored part of the

model space. While the image has areas that are dissimilar to any area of the training image, we would like these areas to have a high relative probability to be re-simulated.

The procedure goes through randomly proposed voxels and picks them with a probability that is proportional to the number of undesirable patterns in their neighbourhood (a pattern is deemed undesirable if it does not exist in the training image).

The procedure suggest a random voxel and then scans the neighbourhood voxels, say it has y neighbouring voxels. It counts how many of these, including the voxel itself, are centrers of undesirable patterns, let us denote that number x. The suggestd voxel is then accepted as a center for the perturbation domain with probability:

$$\text{Prob}(\mathsf{voxel}) \quad = \quad \frac{x}{y+2}$$

The denominator $y + 1$ comes from the number of patterns in the neighbourhood plus the pattern from the voxel itself. The extra +1 is added such that areas with no undesirable patterns, i.e., $x = y$, have a small yet non-zero probability to be chosen. Otherwise the iterative algorithm might be stuck and prevented to converge, as an image can have no undesirable patterns without matching the frequency distribution of the training image and without matching the data fit.

Variable	**Description**	**Type**
Zex	Denotes which voxels in the image are centres of patterns found also in the training image. An element of Zex is true if the pattern centred in the corresponding voxel exists in the training image. And contrary, an element of Zex is false if the corresponding voxel of the image is center of a pattern not found in the training image.	3D logical array
nodes	Array from the Nmask structure.	2D integer array
sN	Parameter, number of neighbours for an inner voxel resulting in patterns consisting of sN + 1 voxels.	integer
i, j, k	Indices of the three dimensions of the image for the voxel that is chosen as the center of the domain to be re-simulated.	integers

B.2 getNeighborhood

Given an image Z and a row, column and layer index of a voxel in the image, this procedure extracts the voxel values of the neighbouring voxels. The extracted voxel values will be flattened into a 1D array.

Variable	Description	Type
Z	Image.	3D integer array
i, j, k	Indices in the image of the center voxel of the neighborhood.	integers
nodes	Array from the Nmask structure.	2D integer array
sN	Parameter, number of neighbours for an inner voxel resulting in patterns consisting of sN + 1 voxels.	integer
Zvec	Holds the values of voxels in a neighbourhood. Unknown voxel values are assigned a value of -1.	1D integer array
innervoxel	Denotes if the voxel with the specified indices was an inner voxel (innervoxel = true) or not (innervoxel = false)	logical

B.3 getCPDF

This is a recursive procedure used by SimVoxel to compute the conditional probability density function of the value of a voxel given the values of its neighbouring voxels.

The procedure searches through the tree (depth first) for patterns that matches the neighbourhood values in Zvec and adds up the counts of patterns depending on the value of their center voxel. cpdf is then a sV+1 element array with the counts of patterns matching the values of the neighbourhood voxels for each of the sV+1 possible value of the center voxel.

Variable	Description	Type
Ttrain	Tree of patterns extracted from the training image.	tree
Zvec	Holds the values of voxels in a neighbourhood. Unknown voxel values are assigned a value of -1.	1D integer array
sV	Parameter, the images contain the $\mathsf{sV}+1$ categories $0, 1, \ldots, \mathsf{sV}$ of voxel values.	integer
sN	Parameter, number of neighbours for an inner voxel resulting in patterns consisting of $\mathsf{sN}+1$ voxels.	integer
cpdf	Current counts of patterns matching the values in Zvec.	1D real array

B.4 ExtendTree

This procedure adds another node to a tree. It takes as input a tree node where the pointers in the next array are not yet associated. The procedure initialises them by allocating their repl arrays and setting the counts to 0. It also sets their depth values to be one deeper than the current T%depth, and it nullifies their next arrays.

Variable	Description	Type
sV	Parameter, the images contain the $\mathsf{sV}+1$ categories $0, 1, \ldots, \mathsf{sV}$ of voxel values.	integer
T	Tree to be extended	tree

B.5 ShapeTree

The procedure shapes a tree T such that it has the same shape as the Ttrain, based on which it was constructed. By construction the tree cannot be

any bigger than Ttrain, as it holds only patterns that are also found in the training image. However, it can be smaller, as some patterns may appear in the training image but not in the image from which T is constructed. The nodes representing such patterns need to be added with the appearance count of zero.

Ensuring the trees have the same shape simplifies future operations such as updating of trees and comparison of frequency distributions.

Variable	Description	Type
T	Tree of patterns extracted from an image Z.	tree
Ttrain	Tree of patterns extracted from the training image.	tree
sV	Parameter, the images contain the sV + 1 categories $0, 1, \ldots,$ sV of voxel values.	integer
sN	Parameter, number of neighbours for an inner voxel resulting in patterns consisting of sN + 1 voxels.	integer

B.6 CopyTree

As trees are complex structures we cannot just copy the content of one tree, Told, into another tree, Tnew, in the way we usually do with arrays. Simply saying Tnew = Told is not defined and therefore has no meaning. The procedure is therefore needed whenever we need to make a copy of a tree. It is used, for instance, by the UpdateSA procedure to copy the tree of the perturbed image Tnew into the variable holding the tree for the current image Ttest.

The procedure initialises a new tree from scratch and then copies the content from the old tree into it node by node, without overwriting Told.

Variable	Description	Type
Told	Tree of patterns extracted from an image.	3D integer array

sV	Parameter, the images contain the sV + 1 categories $0, 1, \ldots,$ sV of voxel values.	integer
sN	Parameter, number of neighbours for an inner voxel resulting in patterns consisting of sN + 1 voxels.	integer
Tnew	Tree, which is a copy of Told.	3D integer array

B.7 DeallocateTree

Recursive procedure that deallocates a tree. This is done by deallocating the repl array and the next array associated with each node for all nodes, one node at a time.

Variable	Description	Type
sV	Parameter, the images contain the sV + 1 categories $0, 1, \ldots,$ sV of voxel values.	integer
T	Tree node to be deallocated.	tree

B.8 UpdateTrainTree

This recursive procedure is used to add a pattern to the tree of the training image. For each pattern, InferTrainTree calls the procedure that then recursively calls itself while going deeper and deeper into the tree. This causes the count of the pattern to be added all the way to the bottom of the tree. If a type of pattern has not already been added to the tree the procedure ExtendTree is used to extend the tree with extra nodes before the pattern can be added.

Variable	Description	Type
Ttrain	Current tree node	tree

Zvec	Holds the values of the voxels in a neighbourhood.	1D integer array
cv	Voxel value of the center voxel of the pattern.	integer
sV	Parameter, the images contain the $\mathsf{sV}+1$ categories $0, 1, \ldots, \mathsf{sV}$ of voxel values.	integer
sN	Parameter, number of neighbours for an inner voxel resulting in patterns consisting of $\mathsf{sN}+1$ voxels.	integer

B.9 wrapUpdateTree

This procedure is called by InferTree to add a pattern to the tree when the center of the pattern is an inner voxel. It works as a wrapper for the recursive UpdateTree.

Variable	Description	Type
T	Tree of patterns so far extracted from the image Z.	tree
Ttrain	Tree of patterns extracted from the training image.	tree
Zvec	Holds the values of voxels in a neighbourhood.	1D integer array
cv	Voxel value of the center voxel of the pattern.	integer
sV	Parameter, the images contain the $\mathsf{sV}+1$ categories $0, 1, \ldots, \mathsf{sV}$ of voxel values.	integer
sN	Parameter, number of neighbours for an inner voxel resulting in patterns consisting of $\mathsf{sN}+1$ voxels.	integer
exist	Indicate whether the pattern is found in the training image (exist = true) or not (exist =false).	logical

B.10 UpdateTree

Recursive procedure that adds the contribution from a pattern which center voxel is an inner voxel. It goes through the pattern voxel by voxel, and adds it contribution node by node as deep into the tree as allowed. Recall that the shape of the tree must not be changed as it shall remain the same as the shape of the tree of the training image. Therefore patterns from the image that are not found in the training image will not contribute to any counts after a certain level of depth as the nodes representing them do not exist. This also means they do not appear in the frequency distribution.

The inputs of the procedure is the same as of wrapUpdateTree.

B.11 wrapUpdateTreeBoundary

Like wrapUpdateTree this procedure is called by InferTree and it works as a wrapper for UpdateTreeBounday. The procedure is used to add a pattern which center is not an inner voxel.

Variable	Description	Type
T	Tree of patterns so far extracted from the image Z.	tree
Ttrain	Tree of patterns extracted from the training image.	tree
cpdfold	Marginal conditional probability distribution of the value of a center voxels conditioned on the values of the neighbouring voxels in Zvec.	1D real array
Zvec	Holds the values of the neighbouring voxels in the pattern. Imaginary voxels have been assigned the value -1.	1D integer array
cv	Voxel value of the center voxel of the pattern.	integer
sV	Parameter, the images contain the $\mathsf{sV}+1$ categories $0, 1, \ldots, \mathsf{sV}$ of voxel values.	integer
sN	Parameter, number of neighbours for an inner voxel resulting in patterns consisting of $\mathsf{sN}+1$ voxels.	integer

exist	Indicate whether the pattern is found in the training image (exist = true) or not (exist =false).	logical

B.12 UpdateTreeBoundary

Recursive procedure that adds the contribution from a pattern which center voxel is not an inner voxel. The contribution of the pattern is set to be proportional to the marginal conditional probability of the center value given the voxel values of the neighbouring voxels. This distinction between inner voxels (handled by UpdateTree) and non-inner voxels (handled by UpdateTreeBoundary) is necessary as they contribute differently to the tree.

UpdateTreeBoundary goes through the tree. For each level it assigns the contribution computed based on the counts of patterns in the tree of the training image. It uses the procedure getCPDF to compute the marginal conditional distributions. Like UpdateTree it never changes the shape of the tree as it only add contributions from patterns that can also be found in the training image.

The inputs of the procedure is the same as of wrapUpdateTreeBoundary.

B.13 GrowTree

This procedure is called by SimNewImage. It is used to iteratively update the tree of an image when the latter is being perturbed. For each changed voxel value up to $\mathsf{sN}+1$ patterns may have changed and the tree needs to be updated for each of these changes.

GrowTree is called each time a voxel value has been changed, and it uses of the procedures CenterCount, AddCount, SubtractCount and wrapUpdateTreeBoundary to update the tree. Out of the possibly $\mathsf{sN}+1$ changed patterns, updating the tree with respect to the pattern of the changed voxel is relatively simple. However, it is done differently depending on whether the voxel is an inner voxel or not, as this makes it contribute differently to the tree.

The changed voxel might be a neighbour of up to sN voxels, and it therefore might be a part of equally many other patterns. By changing the voxel value these patterns have changed too. The updating of these patterns is a bit tricky and depends on whether or not the changed voxel, as well as

the voxels of which it is a neighbour, are inner voxels or not. Patterns not centred in inner voxels are handled by the same procedure as when the tree was first constructed, namely wrapUpdateTreeBounday.

When pattern change it might happen that they go from not being a part of the tree to being part of the tree. The way the perturbation of images is done this will often be the case. It might also happen the opposite, namely that patterns used to be in the tree but are not any more. The number of counts in the frequency distribution of the image is for the same reason varying in the different iterations of the simulated annealing algorithm.

Variable	**Description**	**Type**
Z	The image after the voxel value has been changed.	3D integer array
zold	Old value of the changed voxel.	integer
Zex	Array used by getNewIt, should be updated according to the new patterns created by changing the voxel values.	3D logical array
i,j,k	Indices of the changed voxel.	integers
nodes	Array from the Nmask structure.	2D integer array
sV	Parameter, the images contain the $\mathsf{sV}+1$ categories $0, 1, \ldots, \mathsf{sV}$ of voxel values.	integer
sN	Parameter, number of neighbours for an inner voxel resulting in patterns consisting of $\mathsf{sN}+1$ voxels.	integer
Ttrain	Tree of patterns extracted from the training image.	tree
T	Tree of the image before the voxel was changed. The procedures updates it according to the new image Z.	tree

B.14 CenterCount

This is a recursive procedure called by GrowTree. The procedure is used to update the tree with respect to the pattern centred in the changed voxel when it is an inner voxel. Due to the structure of the tree, the tree is updated by following the (unchanged) voxel values of the neighbourhood voxels and updating the counts of the repl arrays of the corresponding nodes. Say the value of the voxel was changed from i to j then the repl arrays are updated by subtracting 1 count from the i+1th element and adding one count to the j+1th element.

This accounts for the updating for 1 out of the sN+1 patterns possibly affected as explained in the description of GrowTree.

Variable	Description	Type
T	Tree node to be updated	tree
sN	Parameter, number of neighbours for an inner voxel resulting in patterns consisting of sN + 1 voxels.	integer
znew	New value of the changed voxel	integer
zold	Old value of the changed voxel	integer
Zvec	Holds the values of the voxels that are in the neighbourhood of the changed voxel.	1D integer array
exist	Indicate whether the pattern is found in the training image (exist = true) or not (exist =false).	logical

B.15 AddCount

This is a recursive procedure called by GrowTree. The procedure is used to add a count representing new patterns appearing when the image is perturbed. It is used when the voxel, which value was changed, was an inner voxel. This procedure performs the updating of the patterns for those of the sN neighbouring voxels, that are inner voxels.

It loops through those inner voxels, determines which patterns they are now centres of, and adds the count in the tree. Only one value has changed, namely the one belonging to the changed voxel, and the remaining $\mathsf{sN}-1$ values are unchanged. Therefore, if the changed voxel is the ith neighbour in the pattern, then the $i-$!1th first values of the pattern are unchanged and the tree should only be altered from level i and deeper.

Variable	Description	Type
T	Tree node to be updated	tree
sN	Parameter, number of neighbours for an inner voxel resulting in patterns consisting of $\mathsf{sN}+1$ voxels.	integer
level	Depth level of the tree where the changed voxel will first have effect.	integer
Zvec	Holds the values of the neighbouring voxels in the changed pattern.	1D integer array
zcen	Value of the neighbouring voxel that is center in the changed pattern	integer
exist	Indicate whether the pattern is found in the training image (exist = true) or not (exist =false).	logical

B.16 SubtractCount

Like AddCount this is a recursive procedure called by GrowTree. Also this procedure is used to update the tree when the image is perturbed. It is used to subtract the count of the old pattern. It starts from the depth of change in voxel values and goes all the way to the bottom of the tree.

Variable	Description	Type
T	Tree node to be updated	tree

sN	Parameter, number of neighbours for an inner voxel resulting in patterns consisting of sN + 1 voxels.	integer
level	Depth level of the tree where the changed voxel will first have effect.	integer
Zvec	Holds the values of the neighbouring voxels in the changed pattern.	1D integer array
zcen	Value of the neighbouring voxel that is center in the changed pattern.	integer

APPENDIX G

Paper VII

Efficient Prediction of Rock Properties from Seismic Attributes using Orthogonal Transformations

Authors:
Katrine Lange, Thomas Mejer Hansen, Knud Skou Cordua, Yulia Melnikova and Klaus Mosegaard

Published in:
Prepared for submission to Geophysics

Efficient Prediction of Rock Properties from Seismic Attributes using Orthogonal Transformations

Katrine Lange[a,c,*], Thomas Mejer Hansen[b,c], Jan Frydendall[a,c], Klaus Mosegaard[b,c], Christian Rau Schiøtt[d]

[a]*Department of Applied Mathematics and Computer Science, Technical University of Denmark, Matematiktorvet, Building 303B, 2800 Kongens Lyngby, Denmark*
[b]*Department of Mathematical and Computational Gesocience, Technical University of Denmark, Matematiktorvet, Building 305, 2800 Kongens Lyngby, Denmark*
[c]*Center of Energy Resources Engineering, Technical University of Denmark, Søltofts Plads, 2800 Kongens Lyngby, Denmark*
[d]*Hess Corp., 1501 McKinney St., 77010 Houston, USA*

Abstract

Interpolation of rock properties between measured well log data is a well-known problem in seismic exploration. Three-dimensional seismic data is available; from which researchers can extract a large number of seismic attributes and then use them to quantify various characteristics of the data. These extractable attributes have been used to guide interpolation of rock properties between well logs, using neural networks, linear regression and collocated cokriging.

Recently, an alternative method was proposed. Based on kriging interpolation performed in a space spanned by the seismic attributes, the alternative method relies on a distance measure in attribute space in addition to the spatial distance in physical space. Furthermore an orthogonal transformation of the seismic attributes reduces the dimension of the attribute space and thereby reduces the complexity of the problem without losing significant accuracy. We chose a transformation known from Partial Least Squares Regression, because it considers the rock property data and seeks to create transformed variables that have increasing correlation to this data.

We applied the method to data from the South Arne field in the Danish North Sea. We present predictions of porosity using a number of seismic

*Corresponding author
Email address: katla@dtu.dk (Katrine Lange)

attributes combined with information from well logs.

Keywords: geostatistics, kriging, porosity estimation, partial least squares

1. Introduction

Comprehending rock properties, such as porosity and permeability, is important when modelling physical flow in reservoirs, when planning new well locations and managing existing injectors and producers. Unfortunately, rock property measurements are only available in well logs, and are therefore sparse and spatially scattered.

The property distribution in the subsurface must be determined indirectly by performing interpolation of logged properties. Interpolation has been done using linear regression (Hampson et al., 2001; Russell et al., 2002; Hansen et al., 2008b), spline interpolation, nearest neighbor interpolation, collocated cokriging (Doyen, 1988) and neural networks, (Hampson et al., 2001; Russell et al., 2002; Pramanik et al., 2004; Herrara et al., 2006). Our work will be based on kriging interpolation as kriging techniques have the advantages of determining an estimate of the interpolated value and providing an uncertainty estimate.

Traditionally, interpolation has been done in the space spanned by the spatial coordinates. This has been based on an expectation of the points located closely together in physical space would be highly correlated, and points located further apart from each other would be less correlated. The expectation did not account for sudden changes in rock quality (e.g. across faults). The basic assumption behind this paper is that rock property variations will manifest in seismic attribute data (Hansen et al., 2010). Rock property interpolation is therefore not limited to physical space, but rather a high dimensional space spanned by seismic attributes and spatial coordinates.

Seismic attributes assumed to describe the rock properties can include, but are not limited to, depth, two-way travel times, velocity and density. Hansen et al. (2010) performed kriging interpolation in a high-dimensional space spanned by seismic attributes and spatial coordinates. However, it is unclear how a good covariance model can be constructed in a meaningful way in a high dimensional space where the coordinates may not be independent.

Based on Hansen et al. (2010), this paper discusses the possibility of interpolating in a transformed space that is created by an orthogonal transforma-

tion of the coordinates given by seismic attributes and spatial coordinates. In the multivariate data analysis field, orthogonal transformations are known to have two advantages (Anderson, 1984): 1) When utilizing multivariate data, there is often a redundancy effect caused by correlation between variables. For example, depth and two-way travel time,are two seismic attributes expected to be strongly correlated. Redundancy can by removed by projecting the attributes to an orthogonal space, which may be of a lower dimension. 2) An advantage of constructing a lower dimensional data space occurs when the data is assumed also to contain substantial noise and variation that is not correlated to the rock property. In theory, if the transformation is performed correctly, the noise can be filtered out by the projection of data.

Principal component analysis (PCA) as defined by Hotelling (1933) is a widely used method for orthogonal transformation of multivariate data. PCA creates new, uncorrelated, multivariate variables with decreasing variances that are linear combinations of the original data. Other transformations include minimum/maximum autocorrelation factor transformation (Switzer and Green, 1984) and the maximum noise fraction transformation (Green et al., 1988).

Yet another orthogonal transformation is used in partial least squares (PLS) regression, originating from Wold (1966). This transformation is particularly useful when one or more dependant variables are predicted from a large set of highly collinear factors. Similar to PCA, PLS creates a set of orthogonal components with decreasing variance. However, as PLS is usually used for linear regression, it also maximizes the correlation between the dependant variable and component, making PLS well-suited for general estimation.

2. Methodology

In this study we interpolated porosity using related values of seismic attributes. We measured the porosity level in well logs at n distinct points in physical space given by the spatial coordinates. From each of the n points a number of seismic attributes, such as two-way travel time and acoustic impedance, were extracted.

Let z_i for $i = 1, \ldots, n$ denote the value of porosity at the ith location (this notation is chosen consistently with kriging theory and should not be confused with the usual depth notation). To each value z_i we associate a vector $\mathbf{u}_i \in \mathbb{R}^m$ of attribute values. The first three entries of this vector

are typically the spatial coordinates of the ith location, and the remaining entries hold the values of the seismic attributes at this location. We refer to $\mathbf{u}_i$ as the attribute vector of z_i. For simplicity, we refer to all the m elements of $\mathbf{u}_i$, including the spatial coordinates, as attributes. We define the vector of known porosity values as $\mathbf{z} = \big(z_1, \ldots, z_n\big)^T \in \mathbb{R}^n$.

Let $z_0 \in \mathbb{Z}_+$ denote an unknown value of porosity with the known attribute vector $\mathbf{u}_0 \in \mathbb{R}^m$. Typically, seismic attributes are available in a regular grid spanning the reservoir, and the goal is to estimate the porosity level, z_0, at any of these locations $\mathbf{u}_0$.

2.1. Outline of the Method

Applying the method to predict porosity levels can be summarized in the following six steps:

(i) Initialization: Normalization of the attribute data and normal score transformation of the porosity values.

(ii) PLS transformation of the seismic attributes.

(iii) Reducting the interpolation space dimension by selecting a subset of the transformed attributes, i.e., remove redundant information.

(iv) Inference of a covariance model in the transformed and reduced attribute space by use of maximum likelihood estimation.

(v) Kriging interpolation in the transformed and reduced attribute space.

(vi) Inverse normal score transformation of the kriged porosity values back to porosity units and evaluation of the results.

We will elaborate on these steps and additional aspects of the method.

2.2. Normalization of the Attribute Data

Define the attribute matrix $\mathbf{U}$ as the matrix of all n position vectors, i.e., $\mathbf{U} = \big(\mathbf{u}_1, \ldots, \mathbf{u}_n\big)^T \in \mathbb{R}^{n \times m}$. Let $\mathbf{U}_{:,j}$ refer to the jth column of $\mathbf{U}$, i.e., the vector of values of the jth attribute for all n positions.

As the PLS transformation is not invariant to scaling of the data and the seismic attribute units are not directly comparable, we assume the attribute data has been normalized, i.e.:

$$\begin{aligned} \mathrm{E}\{\mathbf{U}_{:,j}\} &= 0 \quad \text{for } j = 1, \ldots, m, \\ \mathrm{V}\{\mathbf{U}_{:,j}\} &= 1 \quad \text{for } j = 1, \ldots, m. \end{aligned}$$

2.3. Normal Score Transformation of the Porosity Data

Kriging interpolation techniques assume the dependant variable is normal distributed. As this is rare for geophysical parameters we apply a normal score transformation (Deutsch and Journel, 1998) of the rock property values.

2.4. Orthogonal Transformation of the Attributes

The PLS components are computed directly as linear combinations of the seismic attributes. While maximizing a combined correlation and covariance criterion the PLS components are defined to satisfy a number of orthogonality and normalization constraints. The advantage of using PLS over PCA is that PLS creates components that have a high variance (i.e., contains as much as possible of the information of the original variables) and creates components that correlate to the dependent variable.

The PLS components $\mathbf{p}_i$ for $i = 1, \ldots, m$ are computed as linear combinations of the seismic attributes $\mathbf{U}_{:,1}, \ldots, \mathbf{U}_{:,m}$. Let $\mathbf{a}_i \in \mathbb{R}^m$ be the vector of coefficients for the ith component, i.e.,:

$$\mathbf{p}_i \;=\; \mathbf{U}\mathbf{a}_i.$$

The ith coefficient vector $\mathbf{a}_i$ is defined as the optimal solution $\boldsymbol{\alpha}^*$ to the optimization problem (Hastie et al., 2009):

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & \mathrm{Var}\left\{\mathbf{U}\boldsymbol{\alpha}\right\} \mathrm{Corr}^2\left\{\mathbf{z}, \mathbf{U}\boldsymbol{\alpha}\right\}, \\ \text{w.r.t.} \quad & \|\boldsymbol{\alpha}\| = 1, \\ & \mathbf{p}_j^T \mathbf{U}\boldsymbol{\alpha} = 0, \qquad \text{for} \quad j = 1, \ldots, i-1. \end{aligned} \tag{1}$$

The coefficient vectors are computed iteratively by use of the SIMPLS algorithm (de Jong, 1993). PCA components are defined in a similar fashion, however the objective consists of maximizing only the component variance.

Let $\mathbf{P} = \left(\mathbf{p}_1, \ldots, \mathbf{p}_m\right) \in \mathbb{R}^{n \times m}$ denote the matrix of PLS components of the attribute data in $\mathbf{U}$ with respect to the dependent data in $\mathbf{z}$.

2.5. Selecting only a Subset of the Attributes

Use only a subset of the components for the interpolation to reduce the interpolation space dimension, i.e., the dimension of the space spanned by the PLS components $\mathbf{p}_i$ for $i = 1, \ldots, m$. To do so we define a new and reduced component matrix $\widehat{\mathbf{P}} = \left(\mathbf{p}_1, \ldots, \mathbf{p}_{\widehat{m}}\right) \in \mathbb{R}^{n \times \widehat{m}}$, with $\widehat{m} < m$. This matrix is achieved by disregarding the last $m - \widehat{m}$ PLS components.

2.6. The Kriging Estimator

The kriging estimator $Z^*(\mathbf{u}_0)$ is a variant of the basic linear regression estimator (Goovaerts, 1997) defined by:

$$Z^*(\mathbf{u}_0) - m(\mathbf{u}_0) \;=\; \sum_{i=1}^{n} \lambda_i \left(Z(\mathbf{u}_i) - m(\mathbf{u}_i) \right), \tag{2}$$

where λ_i is the weight assigned to z_i, which is interpreted as a realization of the random variable $Z(\mathbf{u}_i)$. The expected value of the random variable at the jth position is $m(\mathbf{u}_j)$, which can also be interpreted as a trend in data.

The random function Z is usually decomposed into the trend component, $m(\mathbf{u})$, and the residual component from Eq. (2) . The residual component is modelled as a second order stationary random function with zero mean. Stationarity assumption implies the residual component covariance function $C(\mathbf{h})$ is identical to the kriging estimator covariance function. The distance vector between two positions $\mathbf{u}_i$ and $\mathbf{u}_j$ is $\mathbf{h} \in \mathbb{R}^m$.

In this section we kept the traditional notation by letting $\mathbf{u}_j$ denote a position vector. However, when the kriging interpolation is performed in the transformed and reduced attribute space, it is the transformed attribute vectors $\mathbf{p}_0$ and $\mathbf{p}_i$ that are used in equation (2). We apply universal kriging in the transformed and reduced attribute space, and use first order polynomials to model the trend making it equivalent to linear regression of the PLS components.

2.7. Inference of a Covariance Model

The covariance model consists of a Gaussian term for each of the components $\mathbf{p}_1, \ldots, \mathbf{p}_{\widehat{m}}$ and a nugget term. The model therefore has $\widehat{m} + 1$ parameters collected in the vector $\boldsymbol{\theta}$. The first $\widehat{m}$ parameters are the ranges for each of the Gaussian terms, the last parameter determines the nugget effect.

$$C(\mathbf{h}) \;=\; \theta_{\widehat{m}+1} \left(1 - nugget(\mathbf{h})\right) + (1 - \theta_{\widehat{m}+1}) \sum_{i=1}^{\widehat{m}} \exp\left(-\frac{3h_i^2}{\theta_i^2} \right). \tag{3}$$

The parameters should satisfy $\theta_i > 0$ for $i = 1, \ldots, \widehat{m}$ and $0 < \theta_{\widehat{m}+1} < 1$. The covariance function from Equation (3) is an anisotropic covariance model where the direction of maximum continuity is assumed to coincide with one of the PLS components.

The optimal parameters of the covariance model are determined using maximum likelihood (ML) estimation. We use the ML approach of Pardo-Igúzquiza (1997, 1998). In Pardo-Igúzquiza (1998) several ML techniques are compared (Samper and Neuman, 1989a,b,c; Kitanidis and Lane, 1985; Diggle et al., 2003). We use in the test case presented in the next section the fmincon function from the MATLAB optimization toolbox to solve the optimization problem arising from the ML estimation of the parameters in $\boldsymbol{\theta}$.

3. Case Study: The South Arne Field

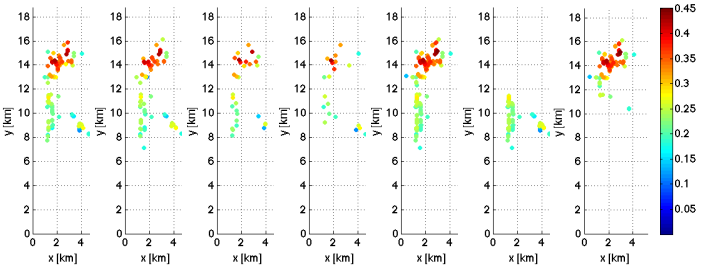
Using data from the Souch Arne Field - a chalk reservoir in the Danish part of the North Sea (Mackertich and Goulding, 1999) - we will demonstrate the method to efficiently predict a rock property from seismic attributes. Hess provided the data previously studied by Hansen et al. (2010).

3.1. Introduction to the Case Study

The oil-bearing chalk of the South Arne Field contains porosities in the range of approximately 0.20 to 0.45 based on well log data. The data contain 203 porosity measurements from well logs that will be used to interpolate (or extrapolate) the porosity values to approximately $76,000$ points in a regular grid covering the entire South Arne Field. The 203 measurements are divided into two subsets to evaluate the method. The first set is the known data, used for the interpolation. The second set is the blind data. The blind data is not used in the interpolation; instead the porosity level at these locations is estimated. The estimated porosity values are then compared to the blind data values to evaluate the performance of the method.

3.2. Porosity Data

Figure 1 shows the known data for seven porosity data divisions. Two different techniques were used to generate divisions. The first technique was used when only every ith data point was considered known. This was the case for the first four divisions, where every second (1a), third (1b), 7th (1c) and 10th (1d) data point was considered known. The second technique was used when only considering data points whose spatial coordinates were above or below a line in the UTM X and UTM Y space. For the fifth data set, all data points with UTM X less than 578.5 km were considered known (1e),



(a) (b) (c) (d) (e) (f) (g)

Figure 1: Known data points for the different blind data sets. Notice the displayed (x, y) coordinates are achieved by a linear transformation of the UTM X and UTM Y coordinates.

and for the two last data sets, all points respectively below (1f) and above (1g) the line where UTM Y equals 6214 km were considered known.

Because of the data point numbering system the first technique resulted in a reduced data set that still spans the physical space and to some extent the seismic attribute space and the original data. The known data set becomes increasingly sparse as i increases. The second approach implies the physical space is not spanned as well as before and we will have to extrapolate beyond the line of division.

3.3. The Seismic Attributes

The interpolation is based on eight seismic attributes, which are all available in the approximately $76,000$ grid points. The attributes are: the depth to the reservoir measured in meters, the two-way travel time to the top and the base of the reservoir, the amplitude and the dip at the top of the reservoir, and the acoustic impedance derived from seismic waveforms.

Figure 2 shows six of the attributes; The spatial coordinates UTM X and UTM Y were omitted because the attributes are plotted against the rotated spatial coordinates. Although the color maps in the figure are not identical, red indicates a relatively high value and blue indicates a relatively low value, for all attributes.

Figure 3 shows the PLS components created by a PLS transformation of the seismic attributes. The PLS components are linear combinations of the seismic attributes constructed to simultaneously maximize their internal

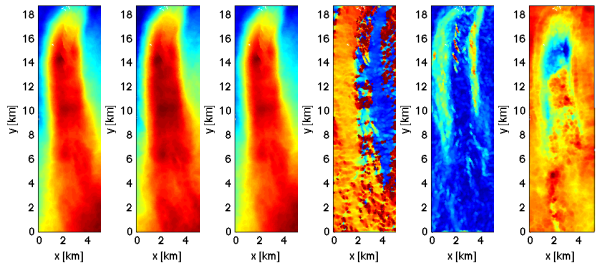


Figure 2: Six of the eight seismic attributes of the South Arne Field. Red indicates a relatively high value and blue indicates a relatively low value. The attributes are UTM Z (2a), two-way travel time to the reservoir base (2b), two-way traveltime to the reservoir top (2c), amplitude at the top of the reservoir (2d), dip at the top of the reservoir (2e) and the acoustic impedance (2f).

variance and maximize their correlation to porosity. Based on a previous study (Hansen et al., 2008a) of chalk reservoirs in the North Sea we expect a strong correlation between acoustic impedance and porosity. This expectation is met by the first PLS component (3a) clearly resembling the acoustic impedance (2f). The second PLS component resembles the three very similar seismic attributes (2a)-(2c), which relate to the depth to the reservoir.

3.4. Assessing Kriging Interpolation via its Estimation Error

Consider the effect the PLS transformation of the seismic attributes has on the kriging results and the consequences of only including a subset of the PLS components. As the available data has been divided into two subsets with known and blind data, we will evaluate the performance of the method in terms of the root mean square (RMS) error of the estimated porosity values in the locations from the blind data set.

Figure 4 shows normalized RMS errors for each of the seven partitions of data as a function of the number of PLS components included. The RMS errors have been normalized in comparison to the RMS error achieved by using kriging in the high-dimensional attribute space, i.e., without applying the PLS transformation. This means for all seven divisions, kriging in the seismic attribute space yields relative RMS errors of one as illustrated by the

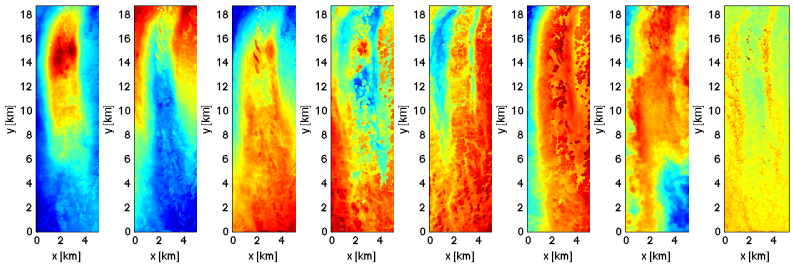


Figure 3: The eight PLS components ordered from left to right of the seismic attributes that except from UTM X and UTM Y can be seen in Figure 2. The PLS components are pictured using individual color scales, red denotes a relatively high value and blue denotes a relatively low value.

dashed line in the figure. The relative error of kriging in the PLS transformed attribute space is represented by the bars. For each of the seven blind data sets, we have kriged using between one and all of the eight PLS components. The figure shows in most of the seven cases the error can be improved by kriging in the PLS transformed attribute space.

Before commenting further on Figure 4, recall how the blind data sets were generated after different criteria. The tendency of the error in Figure 4 is therefore different for blind data set numbers five through seven where we see, in a few cases, including a high number of PLS components causes a high relative error. When this occurs the last PLS components do not contribute to the porosity estimation; instead they can be considered to contain noise.

Test cases that used all eight PLS components are the easiest comparable test cases when discussing the effect of the PLS transformation. In those cases, the method used exactly the same information, and the kriging was performed in an eight-dimensional space. The problems therefore have the same computational complexity. The different results obtained are because of rotation of the axes spanning the space caused by the PLS transformation, making the PLS components perpendicular to one and another. It is the same eight-dimensional space only the directions of the axes spanning it are different. However, as the axes are different, the covariance models inferred in the two cases are different. Recall that the PLS components were

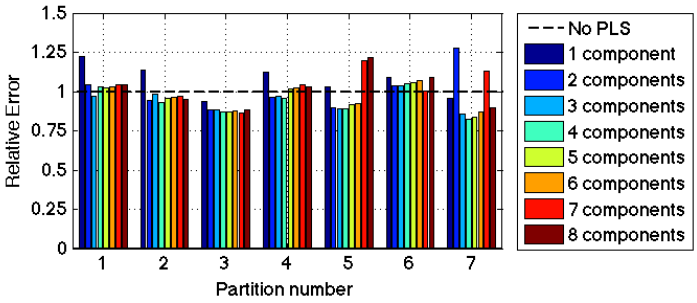


Figure 4: Relative root mean square (RMS) error of the kriging interpolation as a function of the number of PLS components included for each of the seven partitions of data seen in Figure 1. The error is computed relative to the RMS error for interpolation without use of PLS transformation, i.e., without PLS transformation, each data set has a relative RMS error of 1.

constructed partly to maximize their variance. The assumption that the direction of maximum continuity is along one of the axes could therefore be more appropriate in the transformed attribute space rather than the original attribute space.

Figure 4 shows for the blind data sets numbers 1, 2 and 4 the error is approximately unchanged when the PLS transformation is applied and all PLS components are used, i.e., when the information level is identical. For blind data set numbers 3 and 7, the error improved because of the PLS transformation, whereas the opposite true for the blind data sets numbers 5 and 6. Therefore it is only possible to conclude decisively the choice of orientation of the covariance model has significant effect.

More interesting conclusions can be made when we consider the relative error for interpolation in the PLS transformed attribute space as a function of its dimension (i.e., the number of PLS components used). The minimum error is not achieved by including all eight PLS components in any of the seven partitions. The computational complexity of the kriging technique is determined partly by the dimension of the space in which the interpolation is done, allowing us to solve a computationally less complex problem and still achieve results of similar or superior quality.

For all blind data sets including only three PLS components results in an

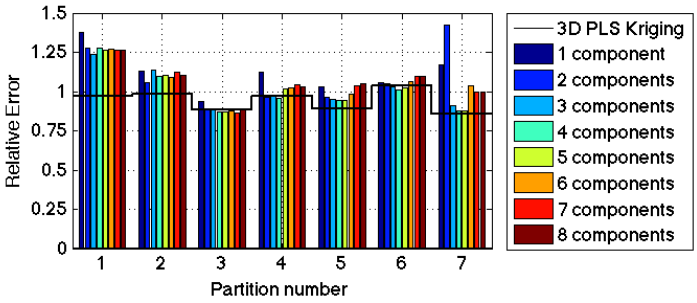
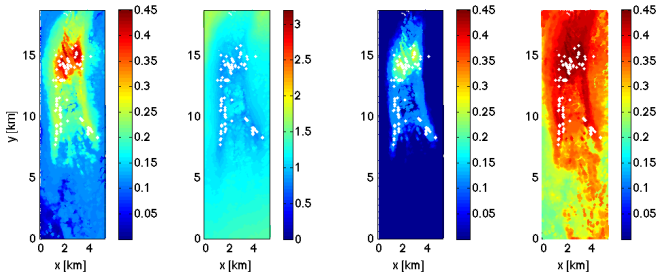


Figure 5: Relative root mean square (RMS) error of linear regression as a function of the number of PLS components included for each of the seven partitions of data seen in Figure 1. The error is computed relative to the RMS error for kriging interpolation without use of PLS transformation to enable comparison with Figure 4. The solid black line represents the error of kriging interpolation using 3 PLS components.

error that is similar or superior to the error when using all components. This is promising since, traditionally, kriging is performed in spatial coordinates, meaning the interpolation space is three-dimensional. In a tree-dimensional space it is possible to infer a full, geometrically anisotropic 3D covariance model. When considering only three PLS components, it is possible to infer a full 3D covariance model instead of the simpler one from Equation 3.

3.5. Comparison to Linear Regression

Figure 5 shows the RMS errors of linear regression interpolation in the PLS transformed attribute space as a function of the number of PLS components included for each of the seven data sets from Figure 1. The RMS errors have been normalized to be comparable to those from kriging interpolation (Figure 4). Using kriging interpolation we determined including three PLS components was the optimal choice; the solid black line in Figure 5 represents this RMS error for each partition. As expected, regardless of the number of PLS components used for the linear regression, linear regression results in similar or larger errors than kriging with a linear trend.



(a) Kriging Mean (b) Uncertainty (c) Lower limit (d) Upper limit

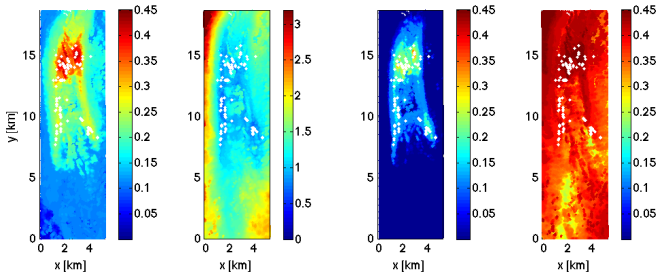
Figure 6: Results from kriging interpolation performed in the transformed seismic attribute space using 3 PLS components. White dots mark locations with known porosity levels. We see the expected value represented by the kriging mean (back transformed to porosity) and the standard deviation of the normal score transformed values. A lower and upper limit for an approximate 95% confidence interval is given by the value of the kriging mean plus/minus twice the standard deviation back transformed to porosity levels.

3.6. Assessing Kriging Interpolation Via its Uncertainty

Although we have shown kriging in the transformed and reduced attribute space has a tendency of increasing the ability to predict blind data values, there are other ways of assessing the performance of the kriging interpolation. One way is to inspect the uncertainty of the kriging estimates (i.e., how certain the interpolation is). This is reflected in the standard deviation.

Figure 5 shows the most significant effect of using kriging interpolation instead of linear regression for the first data set. Therefore we will evaluate the effects of using kriging interpolation in a three-dimensional transformed attribute space for this data set. Figure 6 shows the kriging mean (6a) and the standard deviation of the kriging estimate (6b) when interpolating in the transformed attribute using three PLS components. As expected the kriging standard deviation is lower in the area around the data and higher in the areas without data (as far as that is possible judging by spatial coordinates).

Using the standard deviation of the kriging estimates we have calculated 95% confidence intervals. The lower and upper limit of these estimates can be seen in Figure 6c and Figure 6d, respectively. A high kriging standard deviation causes the limits of the confidence intervals to equal the end points



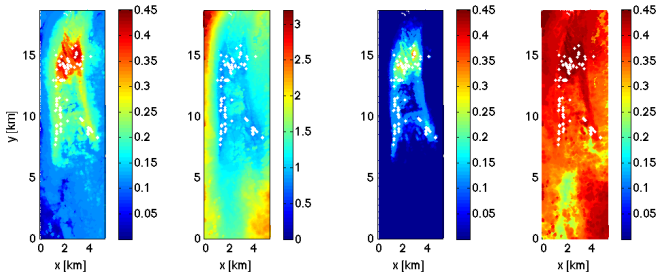
(a) Kriging Mean (b) Uncertainty (c) Lower limit (d) Upper limit

Figure 7: Kriging interpolation performed in the transformed seismic attribute space using all of the 8 PLS components. Again white dots represent locations with known porosity levels and the figure shows the expected value represented by the kriging mean; the standard deviation of the kriging mean expresses the uncertainty. A lower and upper limit for a 95% confidence interval is given by the kriging mean plus/minus twice the standard deviation.

of the interval of valid porosity values imposed by the normal score transformation. When this happens, the kriging estimate is a poor estimator of the porosity level in the reservoir.

For comparison, we will present the results from kriging in the full PLS-transformed attribute space (Figure 7) and from kriging in the original space spanned by the seismic attributes themselves (Figure 8). From Figure 4 we recall kriging in the transformed seismic attribute space using all eight PLS components yields a relative RMS error similar to when using only three PLS components. And the relative RMS error is significantly higher when the PLS transformation is not used (relative error of one).

Based on RMS error of the predictions of the three sets of results presented in Figures 6-8, the first two sets of results are equally good and the third is significantly worse than these two. However, when assessing the accuracy of the interpolation method based on the first set of results (Figure 6) and using the standard deviation of the kriging estimator, we see this set of results is clearly less uncertain than the other two. In this particular case kriging in the transformed and reduced attribute space does not only increase the prediction ability, but it also reduces the uncertainty of the kriging estimate.



(a) Kriging Mean (b) Uncertainty (c) Lower limit (d) Upper limit

Figure 8: Kriging interpolation performed in the seismic attribute space without applying a transformation of the seismic attributes. Also in this figure white dots represent locations with known porosity levels, and it shows the expected value represented by the kriging mean and the standard deviation of the kriging mean expresses the uncertainty. A lower and upper limit for a 95% confidence interval is given by the kriging mean plus/minus twice the standard deviation.

4. Conclusion

In this report we: 1) Introduced the approach for kriging interpolation of well log data in a space spanned by orthogonal components created as linear transformations of the seismic attributes. 2) Stated the motivation for our choice of transformation, namely the PLS transformation otherwise known from partial least squares or PLS regression. 3) Demonstrated how the approach is applied to a test case with data from the South Arne Field in the Danish part of the North Sea.

Through those observations we demonstrated how the prediction ability of the interpolation method remains unchanged or improves when we reduce the dimension of the space in which the interpolation is performed. The reduction of dimensions results in a computationally simpler problem. Finally we discussed the effect of the PLS transformation on the accuracy of the interpolation results expressed by the standard deviation of the kriging estimate.

Acknowledgements

The Danish Council for Independent Research | Technology and Production Sciences (FTP grant no. 274-09-0332) and DONG Energy sponsored the present work. We would like to thank Hess Denmark for permission to publish the results.

Anderson, T. W., 1984. An introduction to multivariate statistical analysis. John Wiley.

de Jong, S., 1993. Simpls: An alternative approach to partial least squares regression. Chemometrics and Intelligent Laboratory Systems 18 (3), 251 – 263.

Deutsch, C. V., Journel, A. G., 1998. GSLIB, Geostatistical Software Library and User's Guide, 2nd Edition. Applied Geostatistics. Oxford University Press.

Diggle, P. J., Ribeiro Jr., P. J., Christensen, O. F., 2003. An introduction to model-based geostatistics. Springer, pp. 43–86.

Doyen, P. M., 1988. Porosity from seismic data - A geostatistical approach. Geophysics 53 (10), 1263–1275.

Goovaerts, P., 1997. Geostatistics for natural resources evalutaion. Applied Geostatistics Series. Oxford University Press.

Green, A. A., Berman, M., Switzer, P., Craig, M. D., jan 1988. A transformation for ordering multispectral data in terms of image quality with implications for noise removal. Geoscience and Remote Sensing, IEEE Transactions on 26 (1), 65 –74.

Hampson, D., Schuelke, J., Quirein, J., 2001. Use of multiattribute transforms to predict log properties from seismic data. Geophysics 66 (1), 220–236.

Hansen, T. M., Mosegaard, K., Cordua, K. S., 2008a. Using geostatistics to describe complex a priori information for inverse problems. In: Ortiz, J. M., Emery, X. (Eds.), VIII International Geostatistics Congress. Vol. 1. pp. 329–338.

Hansen, T. M., Mosegaard, K., Pedersen-Tatalovic, R., Uldall, A., Jacobsen, N. J., 2008b. Attribute guided well log interpolation - applied to low frequency impedance estimation. Geophysics 73 (6), R83–R95.

Hansen, T. M., Mosegaard, K., Schiøtt, C. R., 2010. Kriging interpolation in seismic attribute space applied to the south arne field, north sea. Geophysics 75 (6), P31–P41.
URL http://link.aip.org/link/?GPY/75/P31/1

Hastie, T., Tibshirani, R., Friedman, J., 2009. The elements of statistical learning, 2nd Edition. Springer Series in Statistics.

Herrara, V. M., Russel, B., Flores, A., 2006. Neural networks in reservoir characterization. The Leading Edge 25 (4), 402–411.

Hotelling, H., 1933. Analysis of a complex of statistical variables into principal components. Journal of Educational Psychology 24, 417–441.

Kitanidis, P. K., Lane, R. W., 1985. Maximum likelihood parameter estimation of hydrologic spatial processes by the gauss-newton method. Journal of Hydrology 79 (1/2), 53–71.

Mackertich, D. S., Goulding, D. R. G., 1999. Exploration and appraisal of the South Arne Field, Danish North Sea. In: Fleet, A., Boldy, S. (Eds.), Petroleum geology of Northwest Europe, Proceedings of the 5th conference. Vol. 5. pp. 959–974.

Pardo-Igúzquiza, E., 1997. Mlreml: a computer program for the inference of spatial covariance parameters by maximum likelihood and restricted maximum likelihood. Computers and Geosciences 23 (2), 153–162.

Pardo-Igúzquiza, E., 1998. Maximum likelihood estimation of spatial covariance parameters. Mathematical Geology 30 (1), 95–108.

Pramanik, A. G., Singh, V., Vig, R., Srivastava, K., Tiwary, D. N., 2004. Estimation of effective porosity using geostatistics and multiattribute transforms: A case study. Geophysics 69 (2), 352–372.

Russell, B., Hampson, D., Todorov, T., Lines, L., 2002. Combining geostatistics and multi-attribute transforms: a channel sand case study, Blackfoot oilfield (Alberta). Journal of Petroleum Geology 21 (1), 97–117.

Samper, F. J., Neuman, S., 1989a. Estimation of spatial covariance structures by adjoint state maximum likelihood cross-validation: 1. theory. Water Resources Research 35 (3), 351–362.

Samper, F. J., Neuman, S., 1989b. Estimation of spatial covariance structures by adjoint state maximum likelihood cross-validation: 2. synthetic experiments. Water Resources Research 35 (3), 363–371.

Samper, F. J., Neuman, S., 1989c. Estimation of spatial covariance structures by adjoint state maximum likelihood cross-validation: 3. application to hydrochemical and isotopic data. Water Resources Research 35 (3), 373–384.

Switzer, P., Green, A. A., 1984. Min/max autocorrelation factors for multivariate spatial imagery. Tech. rep., Stanford University.

Wold, H., 1966. Estimation of principal components and related models by iterative least squares. Multivariate Analysis, 391–420.