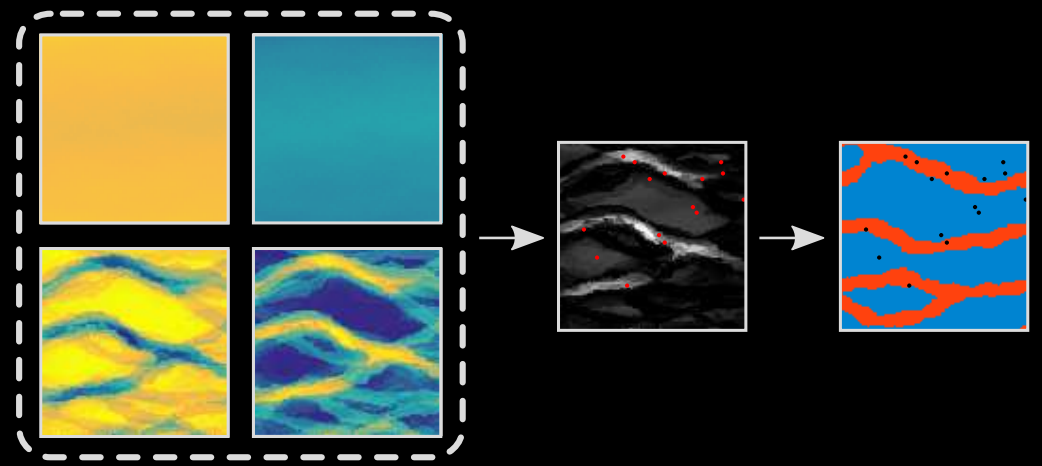
UNIVERSITÉ DE
NEUCHÂTEL

C. Jäggli PoPEx – An adaptive conditional sampler for solving inverse problems in hydrogeology

XII



PoPEx – An adaptive conditional sampler
for solving inverse problems in hydrogeology

Christoph Jäggli

https://doi.org/10.35662/unine-thesis-2753

PoPEx - An adaptive conditional sampler for solving inverse problems in hydrogeology

Thesis

presented at the faculty of science
for the degree of Doctor of Science by

Christoph Jäggli

defended on December 7th 2018
and accepted on the recommendation of

Dr. Julien Straubhaar	co-director
Prof. Philippe Renard	co-director
Prof. Guillaume Caumon	reporter
Prof. Klaus Mosegaard	reporter

Center for Hydrogeology and Geothermics (CHYN)
University of Neuchâtel, Switzerland

UNIVERSITÉ DE NEUCHÂTEL

Faculté des Sciences
Secrétariat-décanat de Faculté
Rue Emile-Argand 11
2000 Neuchâtel – Suisse
Tél : + 41 (0)32 718 21 00
E-mail : secretariat.sciences@unine.ch

IMPRIMATUR POUR THESE DE DOCTORAT

La Faculté des sciences de l'Université de Neuchâtel autorise l'impression de la présente thèse soutenue par

Monsieur Christoph JÄGGLI

Titre:

"PoPEx – An adaptive conditional sampler for solving inverse problems in hydrogeology"

sur le rapport des membres du jury composé comme suit:

- Dr Julien Straubhaar, co-directeur de thèse, UniNE
- Prof. ass. Philippe Renard, co-directeur de thèse, UniNE
- Prof. Klaus Mosegaard, University of Copenhagen, Niels Bohr Institute, Danemark
- Prof. Guillaume Caumon, Université de Lorraine, Ecole Nationale Supérieure de Géologie, Nancy, France

Neuchâtel, le 21.12.2018

Le Doyen, Prof. P. Felber

"In our endeavor to understand reality we are somewhat like a man trying to understand the mechanism of a closed watch."
— Albert Einstein

To Barbara...

Acknowledgements

En tout premier lieu je souhaite remercier mes deux co-directeur de thèse Julien (responsable du projet) et Philippe. Leurs capacité de comprendre des relations complexes dans toute sorte de différentes sujets est impressionnant. En plus de cela, ils sont une source inépuisable de nouveaux idées. Je vous remercie, Julien et Philippe, de m'avoir donné la possibilité de réaliser cette thèse dans le meilleur environnement possible, tant social que techniquement. C'étais un grand privilège et incroyablement enrichissante de vous avoir rencontré et d'avoir eu la chance de travailler avec vous.

Secondly, I want to thank Guillaume Caumon and Klaus Mosegaard for accepting to be part of my committee and for their helpful review of this thesis.

Merci beaucoup à tout le personnel qui m'a aidé avec toutes les choses administratives, notamment Corinne, Carine et Thomas.

Many thanks also to the other people of the CHYN that represented a solid support for me: Guillaume, Martin, Fabio, Axa, Przemyslaw, Dan, Claire, James and Laurent.

Ce travail n'aurai pas eu lieu sans le concours de mes bons amis d'études et de la colocation à Lausanne. Adrien, Alessandro, David, Arnaud et Loïc je vous remercie infiniment d'avoir souffert, célébré, pleuré et rit de nombreux moments avec moi. La valeur de vous avoir rencontré et de votre amitié est inestimable. Merci aussi d'avoir adapté votre niveau de français à la mienne, ça permettait de me faire croire que j'apprenais rapidement.

Während der vergangenen vier Jahren haben Freundschaften für mich einen noch höheren Stellenwert erreicht. Die meisten meiner engsten Freunde habe ich vor vielen Jahren im Kindergarten, der Sekundarschule, der Berufsschule oder im Fussballclub kennengelernt. Pascal, Laura, Martin, Marc, Matthias, Gelu, Alexandre und Simon, Eure Freundschaft war und ist mir unglaublich wichtig, berührend, wunderbar und einfach unbezahlbar. Vielen Dank, dass Ihr mit mir tanzt, lacht, weint, feiert und auch mal leidet (... Ihr wisst schon was ich meine...).

Emanuel und Annemarie, Matthias und Angela, Andreas, Caro, Noelia, Lorine und Roan, Martin und Lauriane, Reto, Sarah, Alina und Ylvi, Sonja und Ronny, Claudia, Thomi und Manuel und David, ich habe das überwältigende Glück, Euch meine Familie nennen zu dürfen. Während nunmehr 33 Jahren

habe ich stets auf Eure bedingungslose Unterstützung zählen dürfen; sei das moralisch, finanziell oder in jeglicher anderen Form. Ich bin tief berührt und dankbar für all die schönen Erinnerungen, Erlebnisse und Momente die ich mit Euch teilen durfte und darf.
Auch von ausserhalb der Familie habe ich grosse Unterstützung erhalten. Beatrice und Roy, was Ihr für mich getan habt, ist alles Andere als selbstverständlich und einfach unbeschreiblich, vielen herzlichen Dank.
Danke auch dem zweiten Teil meiner Familie: Grosi Ida, Käthi, Kathrin, Céline, Dänu, Lemmy, Aurin, Johnny, Eric und Sylvie. Ihr habt mich vom ersten Tag an bei Euch aufgenommen und mir keine Sekunde das Gefühl gegeben nicht willkommen zu sein. Dafür möchte ich mich von tiefstem Herzen bei Euch bedanken.
Als letztes geht mein grösstmöglicher Dank an die wichtigste Person in meinem Leben; an Barbara. Du bist der Grund, warum ich mit einem Lächeln einschlafe und wieder erwache. Was Du in den letzten Jahren geleistet hast ist unfassbar. Deine selbstlose und unermesslich grosszügige Art, mich in Allem zu unterstützen, bewegt mich zutiefst. Ich danke Dir für unglaublich viel Arbeit, Geduld, Aufmunterung, Motivation, Ablenkung, Ausgleich, Freude, Halt, Zuwendung, Ruhe, Enthusiasmus, Antrieb, Hingabe und Liebe die Du mir schenkst. Danke, dass Du bei mir bist.

Neuchâtel, 7. December 2018 *Christoph Jäggli*

Abstract

In the field of geophysics and more precisely of groundwater hydrology, many scientific works rely on accurate estimations of aquifer properties. Solving inverse problems in a complex, geologically realistic, and possibly discrete model space is very challenging. Optimization or smoother techniques often heavily depend on continuous manifolds with linear relations between the model parameters and the physical observations. Monte Carlo methods, on the other hand, frequently require unaffordably large computational efforts. To overcome this dilemma, we propose a sampling method called **Posterior Population Expansion (PoPEx)**. This algorithm combines advanced machine learning techniques with an adaptive importance sampling strategy and yields a highly efficient, parallelizable, and ergodic Monte Carlo scheme. It can be used for solving a broad range of inverse problems, even beyond the field of geostatistics. Its parallel implementation scales perfectly in the sense of Amdal's law. This means that the required computational time is inversely proportional to the number of parallel chains. The asymptotic convergence of the method is demonstrated analytically and empirically on three synthetic test problems. The examples include complex prior information and use state of the art geostatistical modeling tools. They are trained to produce spatial heterogeneity maps with 10 000 to 20 000 discrete model parameters that describe up to 4 different geological facies (categories). However, the method is not restricted to discrete model values and can handle any other type of uncertainty such as initial conditions, boundary conditions, and sources/sinks.

Keywords: adaptive importance sampling, machine learning, uncertainty quantification, bayesian inversion, monte carlo, multiple-point statistics, parallelization

Résumé

Dans le domaine de la géophysique et plus précisément de l'hydrologie des eaux souterraines, de nombreux travaux scientifiques s'appuient sur des estimations précises des propriétés des aquifères. Résoudre des problèmes inverses dans un espace complexe, réaliste et discret est très difficile. Les techniques d'optimisation ou de lissage dépendent souvent fortement de variétés continues avec des relations linéaires entre les paramètres du modèle et les observations physiques. Les méthodes de Monte Carlo, en revanche, nécessitent fréquemment des efforts de calcul inestimables. Pour surmonter ce dilemme, nous proposons une méthode d'échantillonnage appelée **Posterior Population Expansion (PoPEx)**. Cet algorithme combine des techniques avancées d'apprentissage automatique et d'une stratégie d'échantillonnage préférentiel adaptatif et donne un schéma de Monte Carlo extrêmement efficace, parallélisable et ergodique. Il peut être utilisé pour résoudre une vaste diversité de problèmes inverses, même au dehors du domaine de la géostatistique. Sa mise en œuvre parallélisée évolue parfaitement au sens de la loi d'Amdal. Cela signifie que le temps de calcul requis est inversement proportionnel au nombre de chaînes parallèles. La convergence asymptotique de la méthode est démontrée analytiquement et empiriquement sur trois problèmes synthetiques. Les exemples incluent de l'information prior complexe et utilisent des outils de modélisation géostatistiques de pointe. Ils sont entraîné pour produire des cartes d'hétérogénéité spatiale avec 10 000 à 20 000 paramètres discrets décrivant jusqu'à 4 faciès géologiques différents. Cependant, la méthode n'est pas limitée aux valeurs de modèle discrètes et peut gérer tout autre type d'incertitude, telle que des conditions initiales, des conditions limites et des sources / puits.

Mots-clefs: échantillonnage préférentiel adaptatif, apprentissage automatique, quantification d'incertitude, inversion bayésienne, monte carlo, statistique multi-point, parallélisme

Contents

Chapter 1

Introduction

1.1 Motivation

The success or failure of many scientific and engineering problems depends on accurate estimations of aquifer properties. Unfortunately, when studying subsurface complexities that involve groundwater flow and mass transport, most of the crucial parameters are not easily accessible and difficult to measure. *Inverse modeling* aims to obtain information about structures and processes (e.g. hydraulic conductivities, recharge rates, boundary conditions) from indirect measurements (e.g. hydraulic heads, records of tracer concentrations, water temperature, analysis of field samples). Indeed, this is often the only approach for an extensive characterization of hidden earth structures and for estimating physical properties of the buried rocks. In groundwater hydrology, the aim is generally to infer the position of highly permeable or impermeable materials and estimate their porosity and conductivity values from punctual measurements of state variables. For these reasons, inverse methods are of utmost importance in quantitative hydrogeological studies [de Marsily et al., 2000, Carrera et al., 2005, Zhou et al., 2014] as well as environmental modeling [Moles et al., 2003, Wainwright and Mulligan, 2005] of physical systems.

A *physical system* can be understood in a vast number of different ways. Even when restricted to the field of geophysics, it can embrace a glacier snout embedded in a mountain area, an extensive karst network, the interaction of subsurface and surface processes in a river bed, etc.

Numerical Modeling

Many earth scientists reach the point where they need to generate numerical models of their study area. They assume that at a given point in geological time, there is a single 'truth' that involves all physical properties and laws that may explain physical, chemical, and biological processes and structures. Char-

acterizing spatially or temporally distributed natural variables may include quantitative descriptions of

- rainfall time series
- deposits of resources (oil, gas, water, ore)
- petrophysical subsurface properties (permeability, porosity, specific storage)
- spatial features (faults, lenses, karst conduits), and
- dynamic effects (stress, strain).

Although many physical interactions may be understood quite well, it is not possible to access and understand all the details and provide a complete image of the system. In other words, no existing numerical facility is able to continuously handle such characteristics in all detail. The only thing we can hope for, is to engineer numerical models that mimic all the significant features with sufficient accuracy. Suitability of the models is then measured by the ability of explaining and predicting observable processes. But we absolutely need to keep in mind that models have their limits and only represent an approximation of reality.

Numerically, they can be considered as a simplistic set of parameters, each of which representing well-behaved (and often spatially limited) mathematical functions. The parameters must be understood as representative variables or even as one point on a coordinate that is used to describe the geological system. From this perspective, the nature of numerical parameters can be distinguished as follows

- *Continuous variables*: The parameters can continuously change in a given range and choose from an uncountable number of permitted values. It is not rare that they have a natural ordering and that one can perform common mathematical operations on them (sum, multiplication, apply logarithm, compute mean and standard deviation, etc.).
- *Discrete variables*: These parameters are categorical and may only take a countable (often even finite) number of values. Typically, they are seen as indicators that represent material types rather than physical values. The categories are often called *facies* or *facies values* and cannot go into standard mathematical operations.

It is important to note that the differences between the above parameter setups is significant. Even if we might argue that in many cases, it can be switched between the two states by:

- *truncation* (from continuous to discrete values)
- *step functions* (from discrete to continuous values)

but the mathematical rules *are not the same*. Moreover, while from continuous variables it is usually possible to derive smooth (and differentiable) behaviors, they often fail to represent infinitely sharp contrasts. Categorical variables on the other hand may provide well distinguished parameters but are often unable to compare setups by average values or distance measures between realizations.

Geostatistical (conditional) simulation tools are powerful algorithms that produce realistic pictures of continuous and discrete variables. They do so by honoring inter-parametric variability and fall in the scope of Monte Carlo (MC) methods. From this perspective, they can be combined with the results of complex procedures and evaluate the impact of model uncertainty. Assessing parameter variability from a number of different models however, does ask the question of 'how many simulations are sufficient'. The answer highly depends on the complexity of the system and the considered processes. In very simple setups, few realizations can answer most questions very precisely. Conversely, when studying large and nonstationary fields, one realization only provides a single answer among myriads of others.

Data Observations

In most geostatistical studies, a vast number of observable data is involved. In this sense, the term 'data' incorporates outcomes of experiments that consider:

- the state of the system (hydraulic head values, boundary and initial conditions, core extractions, physical conditions);
- the behavior of the system (seismic data, oil and/or water production rates, tectonic movements);
- chemical and biological processes and principles.

Although there might be a vast amount of data, their information content is often sparse relative to the complexity of the geological models. Furthermore, the single data points can be highly correlated or tragically defective, what decreases the carried information even further. On the other hand, acquiring additional and high quality data is usually very time and cost intensive. Think of an oil production company that is interested in deploying new oil fields. Extracting test samples from a candidate reservoir on a very remote place can require huge amounts of specialized infrastructure and personnel.

Using raw and unprocessed data points is rarely advisable. The data set is often transformed and engineered to reduce data-related blunders and enhance the significant information. Unfortunately there are no trusted measures that assess the 'best' way of preprocessing data in general. This makes the efficient usage of the data even more challenging.

Uncertainty Quantification

The first and most important goal of many geostatistical studies is to make the numerical model consistent with the available data. But unfortunately, it is in general not possible to establish the unique 'truth' in terms of one model. This is due to the sparsity of the data as well as the many sources of errors and uncertainties that are involved in such surveys. It is therefore necessary to establish suitable techniques that are able to *quantify the uncertainties*. This is where a stochastic nature of the models can be very useful. From a sufficient large number of realizations, we are able to materialize parameter uncertainty and, more importantly, inherent information credibility. The derived quantities may depend on the models in a complex manner. Computing such measures of interest for a large number of models, results in a statistical distribution or a state of information that can be exploited in a decision making process.

The inverse study of a physical system is mainly divided into the following five steps:

(i) *parameterization*: choosing a finite set of model parameters that fully describes the system;

(ii) *observations*: obtaining information from the system;

(iii) *forward problem*: using physical theories and processes to develop a link between the parameterization and the observations;

(iv) *inverse problem*: inferring knowledge about the values of the underlying model parameters;

(v) *prediction*: measuring posterior probabilities of events.

The forward and the inverse problems often stand opposite one another without being separable. While (in deterministic setups) the forward operator is uniquely determined, its inverse counterpart is commonly considered to be ill-posed. As an example, consider a foggy day in the late November along the beautiful lake of Neuchâtel. Two persons, A and B, find themselves at a distance where it is possible to communicate without being able to see each other. Knowing the distance d between the two persons and the travel time of sound, we can uniquely predict the arrival time of any sound signal starting at A and reaching B (forward problem; e.g. A shouts the name of B). But there are many different position/starting time combinations that give exact the same arrival instant at B. Therefore, the inverse problem (inferring the position of A when knowing the arrival time of the sound signal) has multiple solutions. Even if B roughly knows the direction from where the signal comes from, the number of solutions is still infinite. Furthermore, it is possible that the two persons A and B started their journey together and then suddenly lost each other. This means that in addition to the direction, person B might also have a vague idea of the distance that could possibly lie in between. This increases the information content that B is holding and can ease the finding of A.

When solving an inverse problem, it is therefore beneficial to make explicit any available a priori information and carefully embed all data uncertainties. For these reasons, a very general and simple theory about inverse problems is obtained when using a probabilistic point of view, where the information is represented by measure functions on manifolds.

1.2 Dissertation Preview and Scope

Despite its huge significance and despite more than 50 years of research, current methods are still unable to answer many types of questions efficiently. One of the remaining challenges, for instance, is to solve probabilistic inverse problems that involve discrete structures [Omre and Tjelmeland, 1996, Oliver et al., 1997]. A considerable number of popular inverse algorithms were and are still either special cases or variants of the least squares problem [McLaughlin and Townley, 1996, Zimmerman et al., 1998, Carrera et al., 2005, Zhou et al., 2014].

A general idea, commonly known as the Ensemble-Kalmann filter (EnKF), has widely been used in multiple disciplines of geostatistics [Burgers et al., 1998, Evensen, 2003, Chen and Zhang, 2006, Evensen, 2006, Gu and Oliver, 2007, Bailey and Baù, 2010, Li et al., 2012]. EnKF performs optimally for inverting multi-Gaussian distributed fields with a linear relationship between model parameters and observations. In such cases, two-point statistics (i.e., covariances) are the only statistics that is necessary and sufficient to fully characterize the uncertainties. Because of their efficient usage of computational resources, EnKF (or related variants of) became very popular. Unfortunately, because of their simplistic representation of uncertainties and correlations, in most setups it is highly advised against their usage (e.g. Tarantola [2005] p. 68). Similar reasonings also hold for many iterative smoothing techniques [Zimmerman et al., 1998].

Optimization algorithms on the other hand, minimize the deviation of the predicted states and observation data. Beside of suffering from the initial guess, their main weakness is resumed by Kitanidis [2013] as *"the degree of data reproduction is a poor indicator of the accuracy of estimates"*. Moreover, they also highly depend on continuous model parameters and simplistic uncertainty quantification techniques.

Even more importantly, it is known that channels, lenses, karst conduits, or faults cannot be represented by standard multi-Gaussian fields [Gómez-Hernández and Wen, 1998, Journel and Zhang, 2006, Linde et al., 2015]. Inverse methods that heavily rely on continuity assumptions or simple statistical distributions (typically multi-Gaussian) are not capable to manage such structures. Formally speaking, the widely used hypothesis of the marginal uncertainties (modeling, observation, prior) to be Gaussian may not be feasible. Gaussian density functions over a manifold implicitly assume the space to be linear (continuous). In many applications however, the considered space is discrete or even finite, what makes this assumption unrealistic. Nevertheless, an accurate

identification and representation of discrete geological features is indispensable because they heavily control fluid flow in the underground [Feyen and Caers, 2006, Cherpeau et al., 2012, Caumon, 2018]. Using a wrong and smoothed representation of such features is known to bias significantly the groundwater forecasts and the corresponding uncertainty analysis [Gómez-Hernández and Wen, 1998, Kerrou et al., 2008].

Several attempts have tried to preserve model heterogeneities by improved ensemble techniques [Li et al., 2017], optimization methods [Ravalec-Dupin and Nœtinger, 2002] or a mixture of such schemes [Laloy et al., 2018]. However, up to date, Monte Carlo sampling methods are the only inverse algorithms that are able to produce fully consistent realizations with an accurate uncertainty analysis in highly complex model spaces [Oliver et al., 1997, Mosegaard, 1998, Robert and Casella, 2004, Fu and Gómez-Hernández, 2008, Alcolea and Renard, 2010, Mariethoz et al., 2010a, Hansen et al., 2012, Laloy et al., 2016, Rubinstein and Kroese, 2016]. For such techniques, it is very common to implicitly build the prior knowledge into the sampling procedure. In other words, no closed-form expression for the prior probability measure is needed. A geostatistical modeling algorithm (in form of a black box tool) that samples from a (unknown) prior distribution is sufficient [Mosegaard and Sambridge, 2002, Hansen et al., 2008]. In the past decades, the community has developed a large number of algorithms and methods that are able to model increasingly complex geological features; many of which being based on random function theory. Among these, conditional simulations [Chilès and Delfiner, 2009, Caers, 2011, Pyrcz and Deutsch, 2014] or in particular Multiple-point Statistics (MPS) [Guardiano and Srivastava, 1993, Strebelle, 2002, Arpat and Caers, 2007, Honarkhah and Caers, 2010, Straubhaar et al., 2011, Straubhaar, 2011, Straubhaar et al., 2013, Mariethoz and Caers, 2014] forms a very powerful family of methods. Several such strategies have been combined with Monte Carlo sampling techniques for solving inverse problems [Mariethoz et al., 2010a, Hansen et al., 2012, Linde et al., 2015]. However, one of the most important prerequisites when using Monte Carlo methods is their efficiency [Sambridge and Mosegaard, 2002, Romary, 2010]. Unfortunately, for most of the existing approaches, the computational time can be unsuitably large [Fu and Gómez-Hernández, 2008, Romary, 2010, Linde et al., 2015]. One possibility to solve this dilemma is to design efficient algorithms with powerful parallel behavior and run the sampling scheme on powerful computer facilities.

Parallel Computing

A vast number of algorithms are suitable for running on single CPU machines in which instructions are executed sequentially, one at a time. In these days, high performance supercomputers become more and more available. Their capacity to encompass parallel algorithms that run on multiple processors and simultaneously perform an important number of operations, opens the door for

many real world applications. Especially in the field of Monte Carlo methods, highly parallelized sampling algorithms are a huge promise. An important class of concurrency programming is *dynamic multiprocessing*, which allows users to specify the degree of parallelism without worrying about communication protocols. Some computational algorithms are **embarrassingly parallel** (also called **perfectly parallel**) what means that little or no effort is required to divide them into components that can be executed simultaneously. When considering parallelized workloads, this property is very important. It is closely related to the speedup factor that measures the ratio between the time it takes one processor to complete the job, versus the time it takes n_{par} concurrent processors to execute the same job. Amdahl's Law [Herlihy and Shavit, 2008] limits the maximum speedup of a complex job by the amount of work that must be executed sequentially. If p and $1-p$ denote the respective fraction of sequential and parallelizable workload, the law quantifies the overall computation time as

$$t(n_{\text{par}}) \propto p + \frac{1-p}{n_{\text{par}}}. \tag{1.1}$$

This means that when passing from n_1 to $n_2 > n_1$ concurrent processes, the expected speedup factor S is

$$S = \frac{t(n_1)}{t(n_2)} = \frac{p + \frac{1-p}{n_1}}{p + \frac{1-p}{n_2}}.$$

If the fraction of sequential workload is small $\left(\text{i.e. } p << \frac{1-p}{n_i}\right)$, the speedup is close to $\frac{n_2}{n_1}$ and the parallelization is embarrassing (or perfect). As an example, this means that if the computational power is doubled (i.e. $n_2 = 2n_1$), the same job is completed twice as fast. In the context of concurrent programming, there are two distinctions in scalability: strongly and weakly. The first embeds time scaling for a fixed *overall problem size* while the second considers the speedup factor for a fixed *processor workload*.

Postrior Population Expansion (PoPEx)

In this work we introduce, discuss, and test the **Posterior Population Expansion (PoPEx)** algorithm. The main idea of PoPEx is to expand iteratively an existing set of geological models by using randomly sampled pattern information. In each iteration, the existing set of samples is used to learn, in a statistical sense, the relation between model parameters and state variables. From this point of view, the PoPEx scheme is an *adaptive importance sampler (AIS)* [Naylor and Smith, 1988, Oh and Berger, 1992] that benefits from the previously generated realizations. The method can be interpreted as a combination of a Monte Carlo and an unsupervised machine learning scheme [Russell and Norvig, 2010, Murphy, 2012] that aims to learn an optimal proposal density. Our algorithm can handle all kind of geological structures (channels, lobes,

braided systems, fractures, etc.) and is suitable for a wide range of problems even beyond the field of geostatistics. It is only required to use a forward problem solver, a likelihood measure and a conditional simulation tool that randomly draws models according to the prior information. On top of that, we show how the algorithm can be parallelized and scales embarrassingly well (in the strong sense). This means that the computational time is inversely proportional to the number of parallel chains that are used. Therefore, the only limitation towards the usage of PoPEx for real world applications, is the number of available CPU's, or more precisely, the number of forward problem evaluations that can be run in parallel.

So far, we published two papers that are directly linked to the work presented here. In Jäggli et al. [2017] the original ideas of the (serial) PoPEx sampling and learning scheme was introduced. The method proved to be very efficient but produced slightly biased predictions. In Jäggli et al. [2018] we revisited the PoPEx algorithm and improved its usability, accuracy, and computational time. This second publication provided two very important features. First, it introduced a strategy for computing unbiased predictions. The bias happened because the generation of a new realization is influenced by all the previous models in the chain. This sampling strategy favors some realizations over others. When computing predictions however, these correlations must be taken into account. Secondly, it suggested that the new algorithm is embarrassingly parallelizable. However, the present work still provides original material. The most important part of which is given in section 4.3 where the PoPEx scheme is supported by a complete theoretical description and concludes in lemma 4.3.3 that established the asymptotic convergence of the method.

A PoPEx software framework is implemented in Python, what is due not least because of the excellent work of Ramalho [2015]. Object-oriented design patters [Gamma et al., 1994, Martin, 2008] make for a dynamic and extensible structure. The efficiency of the implementation is mainly due to the numerical python module called *NumPy* [Oliphant, 2015, Robert Johansson, 2015] and the parallelization module called *multiprocessing* [Heimes et al., 2008].

The main goal of this dissertation is to provide a complete and self-contained description of the PoPEx sampling scheme by simultaneously enriching the discussions with examples and illustrations. It is organized as follows. Chapter 2 reviews the formal probabilistic inverse problem as suggested by Mosegaard and Tarantola [2002]. It also provides some simple examples that help to visualize the underlying concepts. In chapter 3 we briefly discuss conditional sampling tools that are widely used in the geostatistical framework. However, it is not aimed to provide a full overview of existing algorithms but rather discuss a powerful MPS tool that is used within the PoPEx sampling scheme. Chapter 4 is dedicated to the theoretical description of PoPEx, proofs its asymptotic convergence, and provides insights in the software implementation and performance. Finally, in chapter 5, we provide three applications of PoPEx that are fully designed to test and evaluate the proposed sampling scheme. At the end of

this work, in chapter 6, we summarize the results, discuss some important features, and provide a brief outlook for future works. Appendix A provides some clarifications on volumetric density functions while the appendix B summarizes the most important symbols and abbreviations that are used throughout the manuscript.

Chapter 2

Probabilistic Inverse Problem

This chapter is dedicated to the mathematical definition of a probabilistic inverse problem. Most of the underlying theory has been introduced and discussed elsewhere. However, in contrast to a theoretical textbook, that are often kept very general, we will focus the discussion to the field of geostatistics and subsurface modeling and provide examples and applications to that field. This also means that the theoretical presentation in this section does not consider the inverse problem in its most general form, but still provides enough details to get full insights in the applicability of the present work.

In his lecture, Hesterberg [2003] requests a mathematical description of a physical problem to be **well-posed** in the sense that

- there is at least one solution (existence);
- there is at most one solution (uniqueness);
- the output depends continuously on the input (stability).

A (linear or non-linear) mapping $\mathbf{g} : \mathcal{M} \to \mathcal{D}$ between two normed spaces $\mathcal{M}$ and $\mathcal{D}$ is well-posed if it fulfills the above requirements [Kirsch, 1996]. Roughly speaking, $\mathbf{g}$ is assumed to be a stable function in the common sense. In the inverse problem framework, $\mathcal{D}$ is often understood to embed all possible outcomes of some measurements, while $\mathcal{M}$ contains full description of the underlying system. It is assumed that physical theories allow to pick any point $\mathbf{m} \in \mathcal{M}$ and predict the related outcome $\mathbf{d} \in \mathcal{D}$ such that $\mathbf{d} = \mathbf{g}(\mathbf{m})$. Roughly speaking, for any model $\mathbf{m}$ the operator $\mathbf{g}$ predicts the set of measurement outcomes in the case where $\mathbf{m}$ represents reality. Many inverse problems aim to characterize the (possibly ill-posed) reverse operator $\mathbf{g}^{-1} : \mathcal{D} \to \mathcal{M}$ and then apply it to a given set of observations $\mathbf{d}^{\text{obs}}$. This approach however, can easily be misinterpreted because of the numerous types of uncertainties that are involved. Furthermore,

the definition of well-posedness requires the spaces $\mathcal{M}$ and $\mathcal{D}$ to be normed, what can be very restricting for many applications. Mosegaard and Tarantola [2002] avoid these difficulties by obtaining a general and elegant theory from a probabilistic point of view. Central to their description is the concept of *states of information* in terms of probability densities defined over manifolds. Their theory defines the solution of an inverse problem to be a conjunction of two states of information that results in a (up to a constant) unique measure function over the joint manifold $\mathcal{M} \times \mathcal{D}$. It is pointed out that, in particular cases, this can be linked to the well known Bayesian approach. The latter strategy however is not free from difficulties and may suffer inconsistencies due to the usage of conditional probabilities. When working with probability densities it is often useful to consider them relative to a *volume measure* of the inherent space. Resulting measure functions are called *volumetric* and are independent of the underlying parameterization (cf. appendix A). In linear vector spaces with Cartesian coordinate system, the volume measure is constant and can be ignored. In a smooth manifold with arbitrary parameterization, volumetric densities can be considered as the ratio of a 'common' density function divided by the Jacobian of the coordinate system with respect to Cartesian parameters. Because of their convenient behavior under a change of variable and especially when considering conditional probabilities we will almost exclusively consider volumetric measures.

In this manuscript, any definition of a term is written in **boldface**. It is common to write multi-dimensional quantities in boldface as well. Some quantities are often multi-dimensional but in some simple examples they only contain one single variable. For consistency, we keep the bold notation even for those single-dimensional cases. The theory discussed in the following section heavily relies on notions from the field of *probability theory* and *manifolds* so that overall, the text shares some common points with the extensive books of Tarantola [2005], Durrett [2010], and Tu [2010].

2.1 From the Model to the Data Space

The way of describing a physical system from a set of parameters is usually not unique. On the other hand, different sets of observations provide other types of information. From a practical point of view it is also necessary to consider the accessibility of some measurements. Other types of data require different efforts and may cause time and/or financial issues. On top of that, physical theories for some configurations might be available and easy to use, while others are extremely complicated and hardly accessible. For all those reasons, it can be be very complicated to derive a suitable inverse problem that can be studied with sufficient accuracy. In the following sections we will discuss the underlying concepts in more detail.

2.1.1 Model Space

We assume that the physical system can be parameterized by a finite number of parameters. Two different parameterizations are called equivalent if they are related by a one-to-one correspondence. Any particular set of parameter values will henceforth be called a **model** or equivalently a **realization** and denoted by $\mathbf{m} = \{m_1, \ldots, m_n\}$. In this regard, a model can cover a vast number of physical and conceptual quantities that consider, for instance, spatial heterogeneities, initial conditions, boundary conditions, and sources/sinks. Let us introduce three simple examples based on which we may illustrate the technical terms of this chapter and ease the reading of the text. More advanced examples of inverse problems will be considered in chapter 5.

Example 2.1.1 (Simple linear regression)**.** *Let's assume that we are given r independent value couples $(x_1, y_1), \ldots, (x_r, y_r)$. The usual setup for simple linear regression is to assume that y_i is a linear function of x_i with random error noise:*

$$y_i = m_1 + m_2 x_i + \epsilon_i, \qquad \text{for} \quad i = 1, \ldots, r.$$

The Gaussian misfits $\epsilon_1, \ldots, \epsilon_r$ are assumed to be independent and identically distributed (i.i.d) variables with expectation 0 and variance σ^2, while the model $\mathbf{m}$ is defined to describe the linear relation and reads $\mathbf{m} = \{m_1, m_2\}$.

Example 2.1.2 (Darcy's law)**.** *Darcy's law plays a key role not only in hydrogeology for estimating the flow rate of subsurface water in aquifers, but also, more generally, to study fluid movement in porous media (geothermics, petroleum reservoirs, etc.). It states that the flow rate (Q in (m^3/s)) through a porous medium is proportional to the difference in hydraulic heads (Δh in (m)) and inversely proportional to the traversed distance (L in (m)):*

$$Q = -kA\frac{\Delta h}{L}$$

where k the hydraulic conductivity in (m/s) and A the cross sectional area in (m^2). A common exercise in undergraduate studies is to get in touch with Darcy's law by setting up an experiment with known A and L, measure Q and Δh, and invert the conductivity k. In this case the set of model parameters is chosen to be $\mathbf{m} = \{k\}$.

Example 2.1.3 (Tunnel construction)**.** *One issue when drilling a tunnel in a mountain area is to set up a time schedule. The time consumption heavily depends on the material that lies in and around the considered region. It is known that solid materials like gneiss and granite are not very problematic, while layers of unconsolidated material can heavily delay the entire process. If the bedrock is established for a given segment, available techniques together with specialist knowledge may allow to estimate the corresponding breakthrough time. In this regard, a (very simplified) model for estimating a time schedule*

could consider the tunnel region as a 1-dimensional line that is split into n (small) segments. A parameterization is then obtained by collecting n material indicators into $\mathbf{m} = \{m_1, \ldots, m_n\}$.

The collection of all possible models is called **model space** and denoted by $\mathcal{M}$. This space must not be understood as a linear space but as an abstract collection of points, a **manifold** [Tu, 2010], where each point represents a valid set of parameters for the system. Models $\mathbf{m} = \{m_1, \ldots, m_n\}$ define a set of representative dimensions, what is equivalent to parameterize the space $\mathcal{M}$. In the hydrogeological framework, a common approach is to subdivide an aquifer into a finite number of volume elements and characterize the hydraulic conductivity in each cell. In this case, the underlying model $\mathbf{m}$ includes one parameter m_i per element. This single parameter defines (constant) physical properties (permeability, porosity, etc) in the corresponding small subdomain. Note that a coordinate system is not unique. 'Permeability', for example, can be replaced by 'resistivity', 'speed' with 'slowness' or 'frequency' with 'period'.

It is important to note that there is no restriction about the nature of the model variables m_i. They may take continuous and/or discrete values that can be real and/or symbolic. There is no need to define a function for measuring the 'closeness' of parameter values. This is convenient, because in many cases, the common operations of a linear space, i.e. summation and multiplication by a scalar, can not be defined in $\mathcal{M}$. Therefore, we do not assume to know a distance measure for $\mathcal{M}$. Although a model $\mathbf{m} = \{m_1, \ldots, m_n\}$ is not a vector (i.e. not a member of a linear space) we will abuse the terminology and call the variables m_i the *components* of $\mathbf{m}$. We will see that the following theory only requires to define probability densities over $\mathcal{M}$. They will be defined to be *independent of the parameterization* such that the solution of an inverse problem is the same in any coordinate system.

2.1.2 Data Space

The data space encompasses every effort for measuring a set of observable variables. For doing so, it is often necessary to perform physical experiments that use (imperfect) observational devices. Let's say there are r observed values that can be gathered within a set $\mathbf{d} = \{d_1, \ldots, d_r\}$, then $\mathbf{d}$ is called **data set** or simply **data**. Collecting data is far from trivial because not only the observations must be carried out as precisely as possible, but also because the information content of each measurement might be unknown in advance. This means that when we must decide between different types of observable variables, it is natural to aim for the one that carries the most information about the model parameters. But before solving an inverse problem such backwards relations are very hard to be estimated. The (abstract) set of all possible observations is called **data space** and denoted by $\mathcal{D}$. Again, there is no reason to assume that this space is linear or accommodates any kind of distance measure. The outcome of the physical experiments is considered to be one point in the data

manifold. For such points, the notation $\mathbf{d}$ is used whenever the data set is considered to be synthetic; if we refer to an actual outcome of some observations we write $\mathbf{d}^{\text{obs}} = \left\{d_1^{\text{obs}}, \ldots, d_r^{\text{obs}}\right\}$. The nature of these values can differ widely and may depend on the overall framework. When studying subsurface properties, they often represent measurements of state variables such as, for instance, hydraulic head values, production rates, or contaminant concentrations.

Example 2.1.4 (continued from 2.1.1)**.** *Let's recall that we are considering a set of r independent couples of metric values $(x_1, y_1), \ldots, (x_r, y_r)$. The independent variables x_i are assumed to be known precisely while the dependent ones, y_i, are distributed around $m_1 + m_2 x_i$ with variance σ^2 and form the set of observations such that $\mathbf{d}^{\text{obs}} = \left\{y_1, \ldots, y_r\right\}$.*

Example 2.1.5 (continued from 2.1.2)**.** *A simple experimental setup that allows to study Darcy's law is the following: a horizontal cylindric container of length L and inner radius r is filled with a porous media. Let's assume that it is possible to pass a constant flow Q alongside the length of the cylinder and measure the hydraulic head at the inlet (h_{in}) and the outlet (h_{out}). Both quantities must be measured as accurately as possible. A straightforward way of defining $\mathcal{D}$ is to consider a two dimensional space that hosts the measurements of Q and Δh. Darcy's law however can also be formulated as*

$$\frac{Q}{\Delta h} = -\frac{kA}{L}.$$

This equation expresses a linear relation between the ratio $Q/\Delta h$ and k. For this reason, we choose to define the data set as $\mathbf{d} = \left\{Q/\Delta h\right\}$ with physical unit equal to (m^2/s). Geostatisticians may compare this quantity to hydrogeological transmissivity values that often represent a vertical integration of material conductivities and measure the amount of water that can be transmitted horizontally.

In practice it is often possible to directly observe parameters that could also be included in $\mathbf{m}$. Boreholes, for example, may provide cores from which petrophysical values can be deduced with high precision (i.e. with negligible uncertainty). If the model parameters are designed to describe the same quantities, such observations might be used for the construction of the model space $\mathcal{M}$.

Example 2.1.6 (continued from 2.1.3)**.** *Let us recall that the problem has been parameterized by a collection of n material indicators such that each model reads $\mathbf{m} = \{m_1, \ldots, m_n\}$. When planing the construction of a tunnel, they usually obtain information about the material properties by extracting a set of soil samples (say p) along the planned tunnel line. What is special in this case is that this kind of 'data' (observations of 'model parameters' with high accuracy) would not go into the data space $\mathcal{D}$, but reduce the dimension of $\mathcal{M}$. This means that the corresponding coordinates in $\mathcal{M}$ have no degree of freedom*

and can be removed. It follows that each model only contains $n - p$ parameter values and reads $\mathbf{m} = \{m_1, \ldots, m_{n-p}\}$ (see also section 3.2.2).

In reality, a tunnel construction problem is very complicated and possibly considers large amounts of data including observations from seismic and groundwater flow experiments. For this reason, we will not go into more detail. This example should only serve to distinguish the two types of observations that can be obtained: high precision measurements that are used in the construction of $\mathcal{M}$ and uncertain observations that are used in the inverse procedure.

In some situations, the above separation between model parameters $\mathbf{m}$ and data set $\mathbf{d}$ can require some reasoning or might not even be desirable. In these cases, it is possible to consider one single manifold that combines all problem parameters. However, in the geostatistical framework $\mathcal{M}$ and $\mathcal{D}$ can usually be well separated. For any other situation, we refer the interested reader to Tarantola [2005].

2.1.3 Forward Operator

From physical theories and/or techniques it is possible to establish a link between the model and the data space. This means that for a well defined physical system it is possible to predict the outcome of observations. From a mathematical point of view, this *forward problem* is captured by $\mathbf{g} : \mathcal{M} \to \mathcal{D}$, called **forward operator** and denoted by $\mathbf{g} = \{g_1, \ldots, g_r\}$. If $\mathbf{m}$ fully defines a physical setup, $\mathbf{g}$ predicts the outcomes of measurements $\mathbf{d}$ such that

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

or equivalently by $d_i = g_i(m_1, \ldots, m_n)$ for $i = 1, \ldots, r$.

We mentioned earlier that the continuity in the definition of well-posedness requires the spaces $\mathcal{M}$ and $\mathcal{D}$ to be normed. Although this is often not the case, many theories do require this assumption for obtaining stability of the mathematical problem. In the probabilistic formulation of Mosegaard and Tarantola [2002] however, the forward operator simply provides an other state of information (cf. 2.2) and avoids any notion of continuity.

Example 2.1.7 (continued from 2.1.4)**.** *A model $\mathbf{m} = \{m_1, m_2\}$ can be linked with a data set $\mathbf{d} = \mathbf{g}(\mathbf{m})$ by simply computing the linear prediction for every independent variable x_i such that*

$$g_i(\mathbf{m}) = m_1 + m_2 x_i, \qquad \text{for} \quad i = 1, \ldots, r.$$

This forward operation can be written as a linear relation $\mathbf{g}(\mathbf{m}) = G\mathbf{m}$ with

$$G = \begin{pmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_r \end{pmatrix} \quad \text{and} \quad \mathbf{m} = \begin{pmatrix} m_1 \\ m_2 \end{pmatrix}.$$

The easiest setup to solve inverse problems is when $\mathbf{g}$ *is invertible. In this case the inverse of* $\mathbf{g}$ *can simply be applied to a set of observations* $\mathbf{d}^{\text{obs}}$ *in order to obtain the underlying model* $\mathbf{m} = \mathbf{g}^{-1}(\mathbf{d}^{\text{obs}})$*. But for* $r \neq 2$*, the above matrix* G *is not invertible. In traditional linear regression, this issue is solved by projecting the data on a smaller subspace* $\mathcal{D}'$*, such that the adapted forward operator* $\mathbf{g}' : \mathcal{M} \to \mathcal{D}'$ *is invertible. The modified forward problem reads*

$$\left(G^{\mathrm{T}}G\right)\mathbf{m} = G^{\mathrm{T}}\mathbf{y},$$

with $G^{\mathrm{T}}G$ *being invertible if* G *is full rank. The underlying model* $\mathbf{m}$ *can then be backtracked by*

$$\mathbf{m} = \left(G^{\mathrm{T}}G\right)^{-1} G^{\mathrm{T}}\mathbf{y}.$$

Although this is a very simple solution, it is hard to asses its reliability. Approaching the same problem from a probabilistic inversion viewpoint, can make its solution much more flexible and meaningful (cf. example 2.2.13).

Example 2.1.8 (continued from 2.1.5)**.** *In example 2.1.2, observation points were considered at the 'inlet' and the 'outlet' of the cylinder. This allows us to assume that* $Q > 0$ *and* $\Delta h = h_{\text{out}} - h_{\text{in}} < 0$*. We recall that the model parameters are one-dimensional and describe the conductivity* k *of the porous media. The forward operator is derived from the linear relation between* $Q/\Delta h$ *and* k *such that*

$$\mathbf{g}(\mathbf{m}) = \Big\{g(k)\Big\} = \Big\{ -\frac{kA}{L}\Big\}.$$

This operator defines a one-to-one map between $\mathcal{M}$ *and* $\mathcal{D}$*.*

Groundwater modelers often use discrete solvers of partial differential equations within the forward operation. In most setups, there are many reasons (approximated theories, choice of parameterization, discretized operator, etc.) to not assume that the predictions are free from uncertainties. Furthermore, extracting a set of observations $\mathbf{d}^{\text{obs}}$ is usually done with imperfect measuring devices. For these reasons, in general, it cannot be assumed that there exists $\mathbf{m} \in \mathcal{M}$ with $\mathbf{d}^{\text{obs}} = \mathbf{g}(\mathbf{m})$. For considering all uncertainties in one inverse problem, we need the concept of *states of information*.

2.2 States of Information

This section describes the mathematical constructs for defining a probabilistic inverse problem. It is again aimed to provide enough background for fully understanding the present work, without having the ambition to present a complete discussion of the topics. Information is represented by a measure function defined over a manifold. Usually, in probability theory such functions are normalized such that the total measure is equal to 1. In this work however, we will only compare relative measure values for submanifolds that live without the need of normalization assumptions.

2.2.1 Volumetric Probability Density

Roughly speaking, a *volume* defined on a manifold must be considered as a non-negative quantity that is independent of any parameterization. For a manifold $\mathcal{X}$ with a parameterization $\mathbf{x} = \{x_1, \ldots, x_n\}$ there might exists a **volumetric density** $u = u(\mathbf{x})$. Thus, the volume of any subset $A \subset \mathcal{X}$ can be quantified as

$$V(A) = \int_{\mathbf{x}^{-1}(A)} u(\mathbf{x}) d\mathbf{x},$$

where $d\mathbf{x} = dx_1 \cdot \ldots \cdot dx_n$ and $\mathbf{x}^{-1}(A)$ is the parameterization subset that spans A through $\mathbf{x}$. The notation $d\mathbf{x}$ is taken from the field of measure theory and must be understood as reference to the measure function that is used in the definition of the integral. From this point of view, it also makes sense to use arithmetics (summation, multiplication, comparison, etc.) on such measures. Let $\{y_1, \ldots, y_n\}$ be a second parameterization with volume density $v = v(\mathbf{y})$ such that

$$V(A) = \int_{\mathbf{y}^{-1}(A)} v(\mathbf{y}) d\mathbf{y} = \int_{\mathbf{x}^{-1}(A)} u(\mathbf{x}) d\mathbf{x}$$

for any $A \subset \mathcal{X}$. Considering the expression $V(A) = \int_A dV$, we obtain $dV = u(\mathbf{x})d\mathbf{x} = v(\mathbf{y})d\mathbf{y}$. Note that if the volume density is known for one parameterization, it can be transformed into any other parameterization by the Jacobian rule applied to the transition map $\mathbf{x} = \mathbf{x}(\mathbf{y})$ [Tu, 2010]. Using elementary properties of the integral, this simply means that

$$v(\mathbf{y}) = u(\mathbf{x}) \left\| \frac{\partial \mathbf{x}}{\partial \mathbf{y}} \right\|, \tag{2.2}$$

where $\frac{\partial \mathbf{x}}{\partial \mathbf{y}}$ determines the Jacobian of the transformation between $\mathbf{x}$ and $\mathbf{y}$.

Example 2.2.1 (Volume in $\mathbb{R}^2$)**.** *Let us consider a compact subset $C \subset \mathbb{R}^2$ with volume equal to $V(C)$. Using the Cartesian coordinates $\mathbf{x} = \{x, y\}$, we have that the volume element is given by $dV = dxdy$. For the polar coordinate system $\mathbf{y} = \{r, \theta\}$ however, the volume element is $dV = rdrd\theta$. The two volume densities $u(\mathbf{x}) = 1$ and $v(\mathbf{y}) = r$ depend on the underlying parameterization, while dV does not. The total volume of C is given by*

$$V(C) = \int_{\mathbf{x}^{-1}(C)} dxdy = \int_{\mathbf{y}^{-1}(C)} rdrd\theta.$$

As $\mathbf{x}$ is a Cartesian parameterization of $\mathbb{R}$, it is clear that $\mathbf{x}^{-1}(C) = C$ for any $C \subset \mathbb{R}$.

Example 2.2.2 (Volume in a countable manifold)**.** *Let us consider a countable manifold $\mathcal{X}$. It is possible to define a one-to-one relationship between $\mathcal{X}$ and the natural numbers $\mathbb{N}$. Any point in $\mathcal{X}$ can thus be represented by the corresponding integer in $\mathbb{N}$. This link is independent from any parameterization and therefore*

can be used to define a volume over $\mathcal{X}$, by using a function $f : \mathbb{N} \to [0, +\infty)$. A subset $A \subset \mathcal{X}$ is then linked to an index set $I \subset \mathbb{N}$ and obtains a volume measure $V(A) = \sum_{i \in I} f(i)$. Without loss of generality, we may eliminate indices with zero measure. Note that in the case of finite manifolds, the same strategy can be used by simply replacing $\mathbb{N}$ by $\{1, \ldots, N\}$.

Example 2.2.3 (Jeffrey's parameter)**.** *Tarantola [2005] extensively studied and discussed Jeffrey's parameter in his textbook. In the geostatistical framework, such parameterizations are widely used and therefore it is worth to mention them briefly here. Jeffrey's parameters are strictly positive variables that can easily be transformed in their inverse, e.g. 'period' and 'frequency', 'resistivity' and 'conductivity', 'velocity' and 'slowness', etc. Let us consider a 1-dimensional manifold that can be parameterized by the strictly positive conductivity $\mathbf{x} = \{k\}$ or equivalently by the resistivity $\mathbf{y} = \{r\}$. The only volume densities that respect the symmetry of this problem are $u(\mathbf{x}) \propto 1/k$ and $v(\mathbf{y}) \propto 1/r$. By the Jacobian rule in equation (2.2) we have that*

$$v(r)r = u\left(\frac{1}{r}\right)\frac{1}{r}.$$

But the symmetry of the problem (i.e. the ratio $r = \frac{1}{k}$) requires the volume measure in $\mathbf{y}$ at r to be inversly proportional to the volume measure in $\mathbf{x}$ at $k = \frac{1}{r}$, or equivalently $u(k) = \frac{1}{v(r)}$. Thus, together with the Jacobian rule we obtain

$$\big(v(r)\big)^2 = \frac{1}{r^2}$$

so that the only non-negative volume measures are the ones given above.

Let us consider a third parameterization $\mathbf{z} = \big\{\log(k/k_0)\big\}$ with $k_0 > 0$. From the Jacobian rule it can be deduced that the volume density corresponding to $\mathbf{z}$ is constant, i.e. $w(\mathbf{z}) = c$ and $dV = cd\mathbf{z}$. This is a so called Cartesian coordinate system and can be compared to the real line $\mathbb{R}$ with the standard Lebesgue measure.

In what follows, we will always consider integral expressions to be *volumetric*. In particular, whenever we study a manifold $\mathcal{X}$ with parameterization $\mathbf{x} = \{x_1, \ldots, x_n\}$ and an integral expression

$$\int_{\mathcal{X}} f(\mathbf{x}) d\mathbf{x},$$

we will henceforth understand $d\mathbf{x}$ to represent the unitless *volume* measure (according to dV) in the coordinate system of $\mathbf{x}$ (cf. appendix A). This is very convenient because the integral expression then becomes independent of the parameterization. When considering separable product spaces, such as $\mathcal{M} \times \mathcal{D}$, the unitless volume element reads $d\mathbf{m}d\mathbf{d}$. However, it must be kept in mind that the measures $d\mathbf{m}d\mathbf{d}$ do depend on the parameterization. For a volume

measure $d\mathbf{x}$, the above function f is called volumetric, because it expresses function values with respect to a volume measure. In this sense, any measure function can be expressed as a volumetric measure.

Let $\mathbb{P}$ be a probability over the manifold Ω. This means that $\mathbb{P}$ assigns probability values to events $A \subset \Omega$ such that $\mathbb{P}(\emptyset) = 0$, $\mathbb{P}(\Omega) = 1$, and for a countable number of disjoint sets $A_1, A_2, \cdots \subset \Omega$:

$$\mathbb{P}(\cup_i A_i) = \sum_i \mathbb{P}(A_i).$$

But the concept of probabilities becomes most useful after introducing **random variables** $X : \Omega \to \mathcal{X}$. Most of the commonly used probability theory is based on the notion of 'measurability' with respect to a σ-algebra (e.g. Durrett [2010]). Therefore, this will be assumed to be true for any considered function between two manifolds. It is important to note that if $X_1, \ldots, X_N$ are random variables and f is a measurable function, then $f(X_1, \ldots, X_N)$ is also a random variable. Any random variable $X : \Omega \to \mathcal{X}$ introduces a probability measure, called its **distribution**, by

$$\varphi(A) = \mathbb{P}(X \in A) \quad \text{for} \quad A \in \mathcal{X},$$

where $\{X \in A\}$ is commonly used for $X^{-1}(A) := \{\omega : X(\omega) \in A\}$. The **volumetric probability density** of X is a function ν such that

$$\varphi(A) = \int_{\{X \in A\}} d\mathbb{P} = \int_A \nu(\mathbf{x}) d\mathbf{x}.$$

The measure ν is called the *Radon-Nikodym derivative* of $d\varphi$ with respect to $d\mathbf{x}$ and denoted by $\nu = \frac{d\varphi}{d\mathbf{x}}$. Intuitively, this expression is very helpful in understanding ν as measure of change in probability ($d\varphi$) with respect to a change in volume ($d\mathbf{x}$), i.e. as a volumetric probability density. It is important to keep in mind that under a coordinate change from $\mathbf{x}$ to $\mathbf{y}$, the volume element changes from $d\mathbf{x}$ to $d\mathbf{y}$ while ν stays untouched (cf. example 2.2.5).

Example 2.2.4 (continued from 2.2.2)**.** *The distribution of a random variable X, mapping Ω into a countable manifold $\mathcal{X}$, can be denoted by φ such that $\varphi(i) = \mathbb{P}(X = i)$ for any $i \in \mathbb{N}$. The corresponding probability density is therefore given by the function ν such that $\nu(i) = \frac{\varphi(i)}{V(i)}$. We have seen that this expression does not depend on the coordinate system. The probability for $A \subset \mathcal{X}$ is then given by $\mathbb{P}(X \in A) = \sum_{i \in I} \nu(i)$, where $I \subset \mathbb{N}$ is the index set that corresponds to A.*

Example 2.2.5 (continued from 2.2.1)**.** *Let us consider the two dimensional Gaussian density function of $\mathcal{N}(0, \sigma^2 I_2)$ and express $\mathbb{P}(X \in A)$ for a given $A \subset \mathbb{R}^2$. In the case of Cartesian coordinates ($\mathbf{x} = \{x, y\}$) we have*

$$\mathbb{P}(X \in A) = \frac{1}{2\pi\sigma^2} \int_{\mathbf{x}^{-1}(A)} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) dxdy,$$

while for a polar parameterization ($\mathbf{y} = \{r, \theta\}$) it reads

$$\mathbb{P}(X \in A) = \frac{1}{2\pi\sigma^2} \int_{\mathbf{y}^{-1}(A)} \exp\left(-\frac{r^2}{2\sigma^2}\right) r dr d\theta.$$

It can be seen that with $r^2 = x^2 + y^2$, the volumetric densities

$$\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \quad \textit{and} \quad \exp\left(-\frac{r^2}{2\sigma^2}\right)$$

in the two coordinate systems are the same while only the volume measure changed from $d\mathbf{x} = dxdy$ to $d\mathbf{y} = rdrd\theta$.

Example 2.2.6 (continued from 2.2.3)**.** *In the hydrogeological framework it is very common to consider log-normal distributions of conductivity values. This means that in the coordinate system $\mathbf{z} = \{z\}$ with $z = \log(k/k_0)$, the probability density follows a Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$ and $\mathbb{P}(X \in A)$ is given by*

$$\mathbb{P}(X \in A) = \frac{1}{\sqrt{2\pi}\sigma} \int_{\mathbf{z}^{-1}(A)} \exp\left(-\frac{(z-\mu)^2}{2\sigma^2}\right) dz.$$

Transforming this expression into the coordinate system $\mathbf{x} = \{k\}$ gives

$$\mathbb{P}(X \in A) = \frac{1}{\sqrt{2\pi}\sigma} \int_{\mathbf{x}^{-1}(A)} \exp\left(-\frac{(\log(k/k_0)-\mu)^2}{2\sigma^2}\right) \frac{1}{k} dk.$$

Again, the volumetric density was not modified but only the volume measure changed from $d\mathbf{z} = dz$ to $d\mathbf{x} = \frac{1}{k}dk$.

In general, when considering a mapping X into an abstract parameter space, we will start by defining a volume element $d\mathbf{x}$ and express any probability measure as volumetric densities with respect to $d\mathbf{x}$. For some parameter manifolds, defining a volume measure however, can be very difficult. An additional family of volumetric elements on parameter spaces can be obtained by Riemannian manifolds and is discussed extensively by Lee [2019]. Once a volume measure has been defined, it is possible to consider probability densities and integral equations to be volumetric. This insures an inverse problem theory that is independent of any parameterization.

2.2.2 Conjunction of States of Information

Any density function can be understood as to represent information on a manifold. If for one manifold there are multiple states of information available, they can be combined into a new state by a 'conjunction'. It is thanks to this concept that the considered theory lives without the definition of conditional probabilities in the (problematic) traditional sense (cf. appendix A). Intuitively, a conjunction can be understood as to connect two states of information similar

to a logical 'and' relation. The advantage of this approach is again, that the nature behind any state of information is not important; the only requirement is that the knowledge can be transformed into a density measure function.

Example 2.2.7 (Contaminant source)**.** *Let us consider a groundwater production well that produces potable water. Periodic quality tests assure that in case of a contamination, the production can be stopped immediately. Once this happens we are interested in locating the source of pollution. Some (rough) knowledge about groundwater flow and subsurface contaminant transport may provide one probability density $p(x, y, z)$ that provides measure values for possible source locations. A second density $q(x, y, z)$ captures knowledge about suspicious sites such as waste deposits, industrial areas, construction zones, etc. The question is now how these two states of information should be combined to obtain a resulting probability density.*

It is clear that the conjunction of two volumetric densities must be commutative and again produce a volumetric density. Furthermore, the conjunction must be *absolutely continuous* with respect to any of the densities. This means that if one of the two densities is zero, then the conjunction must vanish as well. The combination of two volumetric densities p and q is denoted by $p \wedge q$ and defined to be proportional to their product such that

$$p \wedge q(\mathbf{x}) = p(\mathbf{x})q(\mathbf{x}). \tag{2.3}$$

It is clear that the above definition is commutative (i.e. $p \wedge q = q \wedge p$) and absolutely continuous with respect to p and q. What is left to be verified is that $p \wedge q$ also defines a volumetric density. But this is clear as both densities, p and q are invariant under a change of variable and likewise $p \wedge q$ is.

Example 2.2.8 (Conjunction of Gaussian densities)**.** *An example for the conjunction of two states of information can be obtained by combining two Gaussian distributions $p \sim \mathcal{N}(-1, 1)$ and $q \sim \mathcal{N}\left(1, \sigma_q^2\right)$ for different values of σ_q. Figure 2.1 shows the conjunctions $p \wedge q$ for $\sigma_q = 2$ (left), $\sigma_q = 8$ (center), and $\sigma_q = \frac{1}{2}$ (right). For a large value of the standard deviation σ_q, the density q is almost constant in the considered region (often called 'uninformative') and therefore, $p \wedge q$ is close to p. On the other hand, if σ_q is small, q does not allow values far away from 1. In this case, both, q and $p \wedge q$, approximate a Dirac delta density that is centered at 1. These results can also be obtained analytically. The conjunction of two Gaussian distributions is again a Gaussian distribution*

$$p \wedge q \sim \mathcal{N}(\mu_{p\wedge q}, \sigma^2_{p\wedge q})$$

with

$$\mu_{p\wedge q} = \frac{1 - \sigma_q^2}{1 + \sigma_q^2}$$

$$\sigma^2_{p\wedge q} = \frac{\sigma_q^2}{\sigma_q^2 + 1}.$$

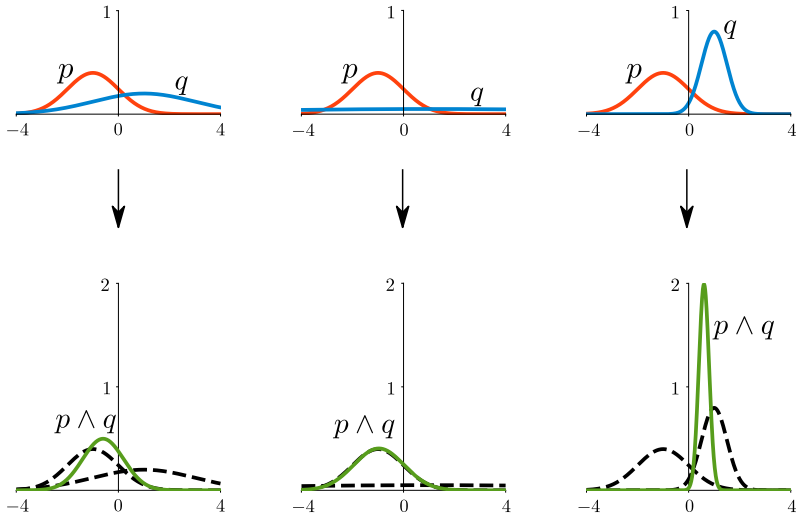


Figure 2.1: Conjunction of two Gaussian densities; a fixed $p \sim \mathcal{N}(-1, 1)$ is combined with $q \sim \mathcal{N}\left(1, 2^2\right)$ (left), $q \sim \mathcal{N}\left(1, 8^2\right)$ (center), and $q \sim \mathcal{N}\left(1, 2^{-2}\right)$ (right).

It can be concluded that $p \wedge q$ converges to the Gaussian distribution of $\mathcal{N}(-1, 1)$ as $\sigma_q \to +\infty$, and that it approximates a Delta dirac distribution centered at 1 as $\sigma_q \to 0$.

Conditional Probability Densities

Conditional probabilities can be understood as a particular case of a conjunction of two states of information. For this we consider an event A with probability $\mathbb{P}(A) > 0$ and volumetric density q_A such as

$$q_A(\mathbf{x}) = \begin{cases} \frac{1}{k} & \text{if } \mathbf{x} \in A \\ 0 & \text{otherwise.} \end{cases}$$

The rescaling constant $k = \mathbb{P}(A)$ normalizes the density q_A but will be most useful when linking the conjunction to conditional probability measures. Let p be the volumetric probability density for $\mathbb{P}$, i.e.

$$\mathbb{P}(B) = \int_B p d\mathbf{x},$$

for all events $B \subset \Omega$. The conjunction of p and q_A is obtained as in equation 2.3 such that $p \wedge q_A(\mathbf{x}) = p(\mathbf{x}) q_A(\mathbf{x})$. Let Q be the probability distribution defined from the volumetric density $p \wedge q_A$ so that the measure of any event

$B \subset \Omega$ reads

$$Q(B) = \int_B p \wedge q_A(\mathbf{x}) d\mathbf{x} = \frac{1}{k} \int_{A \cap B} p(\mathbf{x}) d\mathbf{x} = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(A)}.$$

By definition, this expression matches the conditional probability of B knowing A. It means that $p \wedge q_A$ can be interpreted as the conditional (volumetric) probability density when A is know. For simplicity we will denote this density as $p(\cdot \,|\, A)$. It follows that conditional probability can be obtained from a conjunction of states of information. However, the equation (2.4) can easily be extended to submanifolds $A \subset \mathcal{M}$ with $\mathbb{P}(A) = 0$. It is only needed to define a volume measure on A (denoted by $d\mathbf{x}_A$) that is coherent with the one in $\mathcal{M}$ (cf. appendix A for more details), set $k = \int_A p(\mathbf{x}) d\mathbf{x}_A$ and define

$$Q(B) = \frac{\int_{A \cap B} p(\mathbf{x}) d\mathbf{x}_A}{\int_A p(\mathbf{x}) d\mathbf{x}_A}. \tag{2.4}$$

This can then be interpreted as the conditional distribution of B knowing A when the submanifold probability vanishes, i.e. $\mathbb{P}(A) = 0$. Notice that the concept of volumetric densities is very important for above equation so that the conditional probability Q depends on the volume measure introduced on A.

2.2.3 Solution of an Inverse Problem

This section presents the solution of an inverse problem in terms of the so called *posterior density* function. This equation is crucial because the rest of the manuscript will be based on it.

Measurement Uncertainties

The act of physical measurements is always subject to errors. For this reason, it is not sufficient to describe the outcome of an experiment by a set of values but by a 'state of information'. In this sense, the observations $\mathbf{d}^{\text{obs}} = \{d_1^{\text{obs}}, \ldots, d_r^{\text{obs}}\}$ must produce a density function $\nu(\mathbf{d})$ that is defined on $\mathcal{D}$. A common and often used idea to do so is to use minimal precision of the measuring devices.

Example 2.2.9 (Gaussian measurement uncertainties)**.** *One way to construct a density ν is to assume Gaussian error distributions on a linear data space $\mathcal{D}$. Measuring a set of observable variables can be seen as an instrument that obtains a signal $\mathbf{d}^{\text{in}}$ and presents an output $\mathbf{d}^{\text{out}}$ to the operator. All apparatuses have some minimal precision that often are assumed to be Gaussian. From a statistical point of view this means that*

$$\mathbf{d}^{\text{out}} = \mathbf{d}^{\text{in}} + \epsilon$$

where $\epsilon \sim \mathcal{N}(\mathbf{0}, \Sigma^{\text{obs}})$. *By symmetry of the problem, the density* ν *is then obtained from a Gaussian distribution with mean* $\mathbf{d}^{\text{obs}}$ *and variance* Σ *such that*

$$\nu(\mathbf{d}) \propto \exp\left(-\frac{1}{2}(\mathbf{d}-\mathbf{d}^{\text{obs}})^{\mathrm{T}}(\Sigma^{\text{obs}})^{-1}(\mathbf{d}-\mathbf{d}^{\text{obs}})\right).$$

It is not rare that these errors are independent such that Σ^{obs} *is a diagonal matrix.*

Example 2.2.10 (Perfect instruments)**.** *Assuming perfect measuring devices is usually not feasible. However, there might be situations where the observational uncertainties are negligible with respect to other sources of error. In this case,* ν *can be defined by a Dirac delta function* δ *centered at* $\mathbf{d}^{\text{obs}}$ *such that*

$$\nu(\mathbf{d}) = \delta(\mathbf{d}-\mathbf{d}^{\text{obs}}).$$

Likelihood Uncertainties

The forward operator $\mathbf{g}$ in equation 2.1 predicts the outcome of experiments in a given system. From a naïve point of view, this suggest that the error-free values of observable variables can be predicted for any parameter set. In reality however, this forward procedure undergoes many sources of uncertainties that must be studied carefully. In this regard, the exact relation $\mathbf{d} = \mathbf{g}(\mathbf{m})$ is replaced with a probabilistic expression between the model and the data space. Formally, this is characterized by a volumetric density Θ defined over the joint space $\mathcal{M} \times \mathcal{D}$. We want to slightly simplify this relation by making two assumptions:

(i) the volume measure in $\mathcal{M} \times \mathcal{D}$ is separable and reads $d\mathbf{m}d\mathbf{d}$

(ii) the marginal density $\int_{\mathcal{D}} \Theta(\mathbf{m}, \mathbf{d}) d\mathbf{d}$ is equal to the volume measure $d\mathbf{m}$.

While the first assumption is usually true, the second is not very restrictive (and implicitly assumed in all Bayesian formulations of the inverse problem). More details about the above assumptions can be obtained from Mosegaard and Tarantola [2002].

Under those conditions, the forward operation can be replaced by a volumetric density $\theta(\mathbf{d}\,|\,\mathbf{m})$ over the data space $\mathcal{D}$. The notation suggests to interpret this function as a conditional density, which is less problematic when working with volumetric probabilities. But we rather suggest to view $\theta(\mathbf{d}\,|\,\mathbf{m})$ as probability density over $\mathcal{D}$ that takes $\mathbf{m}$ as an input.

Example 2.2.11 (Gaussian modeling uncertainties)**.** *Let* $\mathcal{M}$ *be an arbitrary model manifold and* $\mathcal{D}$ *a linear space. Gaussian modeling uncertainties on the relation* $\mathbf{d} = \mathbf{g}(\mathbf{m})$ *can be allowed by putting*

$$\theta(\mathbf{d}\,|\,\mathbf{m}) \propto \exp\left(-\frac{1}{2}\big(\mathbf{d}-\mathbf{g}(\mathbf{m})\big)^{\mathrm{T}}(\Sigma^{\text{mod}})^{-1}\big(\mathbf{d}-\mathbf{g}(\mathbf{m})\big)\right),$$

where Σ^{mod} *captures the error covariances.*

Example 2.2.12 (Perfect modeling)**.** *If the modeling uncertainties are neglected with respect to other sources of error,* $\theta(\mathbf{d}\,|\,\mathbf{m})$ *can again be defined from a Dirac delta function such that*

$$\theta(\mathbf{d}\,|\,\mathbf{m}) = \delta(\mathbf{d} - \mathbf{g}(\mathbf{m})).$$

It is often difficult to properly honor modeling uncertainties. In complex cases, it can be possible that an extensive description of $\Theta(\mathbf{m}, \mathbf{d})$ is needed. For simpler setups however, approximating modeling errors by Gaussian densities can be sufficient (cf. example 2.2.11).

Measurement errors and modeling uncertainties are two states of information that are defined on the data space $\mathcal{D}$. For a given model $\mathbf{m}$, the conjunction $(\nu \wedge \theta)(\mathbf{d})$ measures the probability of predicting *and* observing the values in $\mathbf{d}$. Integrating this quantity over the data space is interpreted as a **likelihood** measure for $\mathbf{m}$ such that

$$L(\mathbf{m}) = \int_{\mathcal{D}} \nu(\mathbf{d})\theta(\mathbf{d}\,|\,\mathbf{m})d\mathbf{d}. \tag{2.5}$$

This quantity evaluates the quality of a model $\mathbf{m}$ for explaining $\mathbf{d}^{\text{obs}}$. In the case of example 2.2.12 where $\mathcal{D}$ is a linear vector space and the modeling uncertainties are negligible, we have

$$L(\mathbf{m}) = \nu\big(\mathbf{g}(\mathbf{m})\big). \tag{2.6}$$

When using the equation in example 2.2.12, one must be cautious. If the data space is not linear, the difference $\mathbf{d} - \mathbf{g}(\mathbf{m})$ might not make sense and the Dirac delta function must be redefined properly.

Prior Information

In the literature, *prior information* is understood as 'any information that is obtained independently of the results of measurements'. In the field of geostatistics, there is a high risk that this definition can be misunderstood. The reason for that is illustrated in example 2.1.6. Sometimes it is possible to gather data that can be used in the construction of the model space $\mathcal{M}$. Any prior information defined on the models can therefore not be independent of such observations (because they were used in the definition of $\mathcal{M}$). For this reason we will still use the above definition of prior information but implicitly exclude measurements that had an immediate impact on the definition of the model space.

The **prior information** on model parameters is represented by a density $\rho(\mathbf{m})$. If there is no information available, we could use the volume measure in $\mathcal{M}$ as a prior density function. A lot of scientific texts use prior information densities ρ that follow Gaussian distributions. Beside from being a very smooth representation of information, the problem with this approach is the (inherent)

assumption that $\mathcal{M}$ is a linear vector space. But as it was pointed out earlier, continuous parameters m_i are undesirable for many problem setups. In chapter 3 we will discuss geostatistical modeling tools that are able to simulate highly complex prior information that does not require such assumptions.

Definition of the Inverse Solution

The **posterior information** (i.e. the solution of an inverse problem) in the model space is a conjunction of the prior and the likelihood measure such as

$$\sigma(\mathbf{m}) = cL(\mathbf{m})\rho(\mathbf{m}). \tag{2.7}$$

This equation gives *the* (unique) solution of an inverse problem. Unfortunately, an analytical expression of the posterior density is rarely available and approximation techniques are needed. We can see that the notation of the posterior density function is the same as the often used notation of standard deviation; both being denoted by σ and both are used in this text. However, we believe that throughout the following work, the context will be sufficiently clear such that no confusion about the meaning of σ will occur.

Example 2.2.13 (continued from 2.1.7)**.** *For performing linear regression, we may assumed that $\mathcal{M} = \mathbb{R}^2$ with uniform prior information and that the forward operation is exact. In this case, the likelihood measure $L(\mathbf{m}) = \nu\big(\mathbf{g}(\mathbf{m})\big)$ is proportional to the posterior density function and reads*

$$L(\mathbf{m}) \propto \exp\left(-\frac{1}{2}\sum_{i=1}^{r}\frac{(g_i(\mathbf{m}) - d_i^{\mathrm{obs}})^2}{\sigma^2}\right).$$

Usually, people are interested in estimating the model that maximizes the likelihood function $\mathbf{m}^{\mathrm{ML}} = \{m_1^{\mathrm{ML}}, m_2^{\mathrm{ML}}\}$. It can be shown that a non-biased estimator $\hat{\mathbf{m}}^{\mathrm{ML}} = \{\hat{m}_1^{\mathrm{ML}}, \hat{m}_2^{\mathrm{ML}}\}$ is obtained by

$$\begin{aligned}\hat{m}_1^{\mathrm{ML}} &= \overline{y} - \hat{m}_2^{\mathrm{ML}}\overline{x}\\ \hat{m}_2^{\mathrm{ML}} &= \frac{\sum_{i=1}^{r}(x_i - \overline{x})(y_i - \overline{y})}{\sum_{i=1}^{r}(x_i - \overline{x})^2}\end{aligned}$$

where $\overline{x} = \frac{1}{r}\sum_{i=1}^{r} x_i$ and $\overline{y} = \frac{1}{r}\sum_{i=1}^{r} y_i$. The uncertainty of the estimator is approximated by the matrix $\hat{\Sigma}$ such that $\hat{\Sigma} = \sigma^2\left(G^{\mathrm{T}}G\right)^{-1}$ (with G defined in example 2.1.7), i.e.

$$\hat{\Sigma} = \begin{pmatrix} a & b \\ b & c \end{pmatrix}$$

where

$$a = D\sum_{i=1}^{r} x_i^2, \qquad b = -D\sum_{i=1}^{r} x_i, \qquad c = rD,$$

and $D = \sigma^2 / \left[r \sum_{i=1}^{r} x_i^2 - \left(\sum_{i=1}^{r} x_i \right)^2 \right]$.

In the case where the prior density ρ *is uniform, the maximum likelihood point* $\mathbf{m}^{\mathrm{ML}}$ *is equal to the point* $\mathbf{m}^{\mathrm{MAP}}$ *that maximizes the posterior probability density. This means that* $\hat{\mathbf{m}}^{\mathrm{ML}}$ *is an approximation of the 'most probable' model parameters in* $\mathcal{M}$*. However, the probabilistic inversion is much more flexible than that. It is possible, for instance, to involve prior knowledge about* $\mathcal{M}$ *as well as more complex likelihood functions.*

Example 2.2.14 (continued from 2.1.8)**.** *From example 2.1.8 and with* $\mathbf{d} = \{d\} = \{Q/\Delta h\}$*, the inverse of the forward operator is*

$$\mathbf{g}^{-1}(\mathbf{d}) = \left\{ g^{-1}(d) \right\} = \left\{ -\frac{QL}{\Delta h A} \right\}.$$

A common technique used in similar setups is called 'propagation of uncertainties'. This method propagates measurement errors from the data into the model space. Assume that one instrument observed Q^{obs} *with a precision* δQ*, while a second apparatus measured* h_{in} *and* h_{out} *each with error* δh*. Uncertainty propagation starts by computing an error estimate for* $\mathbf{d}^{\mathrm{obs}} = \left\{ \frac{Q^{\mathrm{obs}}}{\Delta h^{\mathrm{obs}}} \right\}$ *and then uses the inverse operator* $\mathbf{g}^{-1}$ *to generate an uncertainty estimate for* $\mathbf{m} = \{k\}$*. From our point of view, these steps can be interpreted as follows. The permeability* k *can take any positive value so that* $\mathcal{M} = (0, +\infty)$*. There is no additional information about* k *available and we put the prior density* ρ *to be constant on* $\mathcal{M}$*. The data parameter* $Q/\Delta h$ *can be any real value so that* $\mathcal{D} = \mathbb{R}$*. The forward operator is exact and the likelihood function becomes* $L(\mathbf{m}) = \nu\big(\mathbf{g}(\mathbf{m})\big)$*. Together with* $\mu = g^{-1}(d^{\mathrm{obs}}) = \frac{Q^{\mathrm{obs}} L}{\Delta h^{\mathrm{obs}} A}$ *it is deduced that the posterior density follows a Gaussian distribution such that*

$$\sigma(\mathbf{m}) \sim \mathcal{N}\left(\mu,\ \mu^2 \left[\left(\frac{\delta Q}{Q^{\mathrm{obs}}} \right)^2 + 2 \left(\frac{\delta h}{\Delta h^{\mathrm{obs}}} \right)^2 \right] \right),$$

what simply derives from the measurement density ν *with propagated errors.*

Generally, solving an inverse problem is not only about the characterization of the posterior distribution in equation (2.7) but more importantly its usage for computing quantities of interest such as mean values, maximum a-posteriori values (MAP), or posterior probabilities of events. The latter is a basic problem in measure theory and can be migrated to inverse problems whenever we are interested in the probability that a model $\mathbf{m}$ belongs to a given region $A \subset \mathcal{M}$ of the model manifold. In section 4.1 a family of techniques, called Monte Carlo sampling methods, are introduced briefly. They can be used to approximate very complex (e.g. multi-modal) probability densities in high-dimensional spaces.

Chapter 3

Conditional Sampler

Implementing a reasonable model space $\mathcal{M}$ and equipping it with a suitable prior distribution ρ can be very challenging. An outstanding amount of time and effort has been put in the design and description of advanced geostatistical modeling tools [Chilès and Delfiner, 2009, Caers, 2011, Pyrcz and Deutsch, 2014]. Many algorithms do not provide enclosed expressions of the prior density. The prior information is only used in an implicit form for generating (pseudo-) stochastic realizations. These simulations can be interpreted as samples from a random vector with a distribution that follows the prior probability density. In this work, we will not present new modeling algorithms or techniques, but recall the main ideas of geostatistical modeling and establish some examples that can be used in the inverse methods in chapter 4. The task that is addressed by geostatistical modeling can be illustrated with the following problem.

Loch Ness is a large freshwater lake in the highlands of Scotland southwest of Inverness. The elongated lake extends for approximately 37 (km) and is nested between steeply rising mountains to either side. It is most famous for its legendary inhabitant called Nessie, which existence is controversial and was not proven undoubtedly yet. What is interesting is that even high-tech sonar surveys have failed to come up with conclusive evidence for its non-existence. The biggest challenge when studying the interior of Loch Ness is due to its incredible depth. Best estimates say that at some places the lake is over 250 meters deep and contains complex structures of silt and mud at its bottom. Loch Ness counts as the largest lake by volume in the British Isles. Estimating the volume in a lake however is not trivial and requires accurate and meaningful models of its bottom surface. Without any additional information, the model space $\mathcal{M}$ would therefore consist of all two-dimensional surfaces that represent the bottom of the lake. This is a huge number of possibilities and not very suitable for practical purposes. From 'a geological point of view' many of them are not very realistic. Even more importantly, there might exist some expert knowledge (tectonic evolution in the past, landscape in the close environment,

etc.) that can distinguish realistic bottoms from unrealistic ones. At first hand, a *conditional sampler* is a stochastic process that is able to generate plausible realizations for such a problem. But they have a very important feature; the simulations can be conditioned. In the case of Loch Ness, sonar experiments from the surface of the lake may provide a two-dimensional grid of bathymetry values. Conditional simulation tools then provide geologically realistic models that honor all the depth measurements. Let the black curve in the left figure of 3.1 represent a one-dimensional section of the (unknown) lake bottom. The red dots indicate the locations where the depth measurements

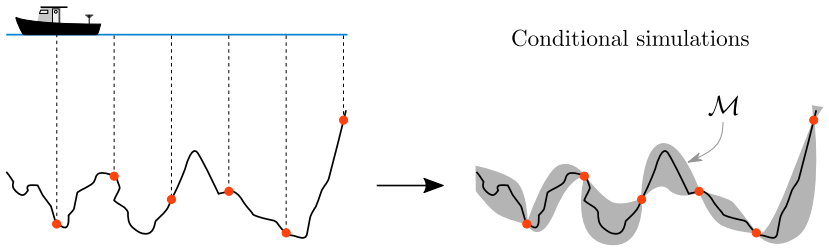


Figure 3.1: Conditional simulations are able to provide a set of geologically realistic models that respect some constraints. The union of all possible realization represents the prior distribution ρ defined on the model space $\mathcal{M}$.

take place and therefore represent one source of information about the shape of the base. A conditional simulation tool is an algorithm that can be fed with the set of observations and provides random scenarios that model the lake bottom and coincide with all measurements. From a statistical point of view, we may want to characterize all possible lake bottoms that coincide with the observations. In this regard, the gray region in the right figure can be understood as an overlapped drawing of all possible bottom curves that honor these measurements and represents the model space $\mathcal{M}$. However, there is no reason to expect that any realization is a precise replica of reality. Among an immense number of others, they are all just possible version of it. In fact, many realizations quite heavily differ from the real surface, but they still share similar characteristics. In this sense, $\mathcal{M}$ is understood as the union of all possible outcomes of the simulation tool, while ρ is related to its stochastic nature. The conditioning values (that must be 'interpolated' by any simulation) are commonly called *hard conditioning data* or simply *hard data (HD)*. It is clear that such a set can significantly reduce the complexity of $\mathcal{M}$. Up to this point there are still some open questions; the most important one might ask what a 'realistic model' is and how it can be obtained from a very sparse set of sonar experiments. For answering this, we first need to explain the mathematical formalism behind conditional sampling.

3.1 Mathematical Concepts

In most textbooks, conditional sampling tools are characterized by a random function Z. As in this work, conditional sampling is used to produce members of $\mathcal{M}$, we will denote the corresponding random function as a (multi-dimensional) random vector $\mathbf{M} : \Omega \to \mathcal{M}$ where each realization $\mathbf{m} = \mathbf{M}(\omega)$ represents one point in the model space. The prior density ρ is related to the stochastic nature of $\mathbf{M}$, what formally means that for any subset $A \subset \mathcal{M}$, the simulator produces models within A with probability

$$\mathbb{P}(\mathbf{M} \in A) = \int_A \rho(\mathbf{m}) d\mathbf{m}. \tag{3.1}$$

The number of coordinates in $\mathcal{M}$ is assumed to be finite, hence a simulation $\mathbf{m}$ can be written as $\mathbf{m} = \{m_1, \ldots, m_n\}$. As mentioned in the example above, it is possible to consistently impose some of the model parameters. This means that they are known in advance and the conditional tool only simulates the remaining ones. Let $I \subset \{1, \ldots, n\}$ be a set of indices and $\mathbf{v} = \{v_1, \ldots, v_{|I|}\}$ a set of values. The conditional simulation can be understood as to randomly select one model in the subset $\{\mathbf{m} \in \mathcal{M} \,:\, \mathbf{m}_I = \mathbf{v}\}$. The subscript I in $\mathbf{m}_I$ indicates a parameter reduction of $\mathbf{m}$ to the set of indices in I. The hard conditioning data is represented by index-value pairs from I and $\mathbf{v}$ and will henceforth be denoted by $\mathbf{hd} = (I, \mathbf{v})$.

One of the most important properties for conditional algorithms is their natural way of handling hard conditioning data. They restrict the random variable $\mathbf{M}$ such that the conditioned simulations follow the conditional density $\rho(\cdot\,|\mathbf{hd})$ (given by equation (2.4)). This might seem trivial but is not. The equation (2.4) combines two density functions on the (theoretically fully known) model space $\mathcal{M}$. But there is no way for the conditional simulator to know the entire model space nor the prior distribution ρ. It can happen that some algorithms handle conditional sampling by an approximation of $\rho(\cdot\,|\mathbf{hd})$. Usually, these inaccuracies are small so that the inherent uncertainties can be neglected (but it might require to limit the number of conditioning data in $\mathbf{hd}$). In conclusion, a hard conditioning data set $\mathbf{hd} = (I, \mathbf{v})$ can be included in a conditional sampler such that it generates models $\mathbf{m} \in \mathcal{M}$ with $\mathbf{m}_I = \mathbf{v}$ and according to the conditional prior density $\rho(\cdot\,|\mathbf{hd})$. A realization $\mathbf{m}$ that is obtained from an *empty* conditioning set $\mathbf{hd}$ is called *unconditioned*.

As mentioned in the introduction, there are two basic model parameter classes that can be distinguished: continuous and discrete. Complex model spaces might consider many different parameters with mixed classes. In the geostatistical framework, it happens that a model simultaneously considers boundary conditions, initial conditions, spatial heterogeneities, recharge time series, etc. Some of them may be continuous while the others are discrete. In this case, multiple simulation tools can be combined and produce realizations $\mathbf{m} = \{\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_t\}$, where each subset $\mathbf{m}_i = \{m_{i\,1}, \ldots, m_{i\,n_i}\}$ is obtained from a different simulator and represents an other set of physical variables. The

total number of parameters (i.e. the sum of all n_i) is equal to n. Each part in such a model separation is called a **model type**. An example that involves two different model types is discussed in section 5.2. For multiple model types, $\mathbf{M} : \Omega \to \mathcal{M}$ can be decomposed into a set of random vectors such that $\mathbf{M} = \{\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_t\}$. It is clear that the model types may depend on each other, so that we should not consider $\mathcal{M}$ to be a Cartesian product of model type specific subspaces (cf. section 5.2). Each parameter set is supposed to be either fully continuous or fully discrete. Standard statistical methods, as the computation of histograms, require a classification of all possible parameter values into a set of categories. This can be done for either continuous or discrete values. We will assume that for a given model type, the same categories apply for all parameter values. They can be defined as follows: for $i = 1, \ldots, t$ let $\mathcal{F}_i = \{f_{i\,1}, \ldots, f_{i\,s_i}\}$ be a partition of the image of $\mathbf{M}_i$, denoted by $\mathbf{M}_i(\Omega)$. This means that all possible image values of the random vector $\mathbf{M}_i$ are distinguished into s_i categories. The subsets $f_{i\,j}$ can be countable (discrete image space) or dense (continuous image space) and are all distinct. If the model type i has a finite number of possible values, then $f_{i\,j}$ usually defines a singleton and is called **facies value** or simply **facies**.

Example 3.1.1 (Discrete)**.** *Let $\mathbf{M}_i$ be a random variable associated to a discrete model type with image $\mathbf{M}_i(\Omega) = \{0, 1, 2\}$. A partition $\mathcal{F}_i = \{f_{i\,1}, f_{i\,2}, f_{i\,3}\}$ of the image set is obtained by*

$$f_{i\,1} = \{0\}, \quad f_{i\,2} = \{1\}, \quad \textit{and} \quad f_{i\,3} = \{2\}.$$

It is clear that there are multiple options for partitioning the set $\{0, 1, 2\}$. A second (trivial) possibility could be to define $\mathcal{F}_i = \{f_{i\,1}\}$ with $f_{i\,1} = \{0, 1, 2\}$. However, in the first case, the subsets $f_{i\,j}$ are singletons and can be called facies values.

Example 3.1.2 (Continuous)**.** *Let $\mathbf{M}_i(\Omega)$ be a univariate random variable with image $M_i(\Omega) = \mathbb{R}$. A partition of $\mathbb{R}$ into s_i intervals can be obtained from $s_i - 1$ real values $c_1 < c_2 < \cdots < c_{s_i-1}$ such that*

$$\mathbb{R} = (-\infty, c_1] \times (c_1, c_2] \times \cdots \times (c_{s_i-1}, +\infty).$$

Categorization intervals in $\mathcal{F}_i = \{f_{i\,1}, \ldots, f_{i\,s_i}\}$ then read

$$f_{i\,1} = (-\infty, c_1], \quad \ldots, \quad f_{i\,s_i-1} = (c_{s_i-2}, c_{s_i-1}], \quad f_{i\,s_i} = (c_{s_i-1}, +\infty).$$

But again, there is an infinite number of possible partitions. In figure 3.2 a Gaussian distribution has been categorized by consecutive (left) and centralized (right) subsets.

In the above setup, realizations $\mathbf{m}$ are obtained from multiple conditional simulators, thus it is possible to condition the model types individually. Any hard conditioning data set $\mathbf{hd}$ is divided into $\mathbf{hd} = \{\mathbf{hd}_1, \ldots, \mathbf{hd}_t\}$ where

$$\mathbf{hd}_i = (I_i, \mathbf{v}_i), \quad \text{for } i = 1, \ldots, t.$$

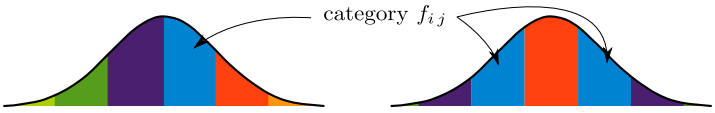


Figure 3.2: Consecutive (left) and centralized (right) categorization of a Gaussian distribution.

However, for simplifying the reading of the remaining text, let us return to notations $\mathbf{M} : \Omega \to \mathcal{M}$, $\mathbf{m} = \{m_1, \ldots, m_n\}$, and $\mathbf{hd} = (I, \mathbf{v})$ without specifying the model type separation explicitly. In fact, the following concepts are considered to be model type specific and (in case of multiple types) can be established individually.

From a partition $\mathcal{F} = \{f_1, \ldots, f_s\}$, it is possible to collect s indicator functions $\mathbf{1}_{f_1}, \ldots, \mathbf{1}_{f_s} : \mathcal{M} \to \{0,1\}^n$ such that

$$\Big(\mathbf{1}_{f_i}(\mathbf{m})\Big)_j = \begin{cases} 1 & \text{if } m_j \in f_i \\ 0 & \text{otherwise.} \end{cases}, \qquad \text{for } j = 1, \ldots, n. \tag{3.2}$$

As $\mathbf{m}$ is a random realization of $\mathbf{M}$, the above functions can be interpreted as random vectors $\mathbf{1}_{f_i} \circ \mathbf{M}$ or equivalently $\mathbf{1}_{f_i}(\mathbf{M})$. For simplicity, the notation of $\mathbf{M}$ is omitted so that they read $\mathbf{1}_{f_i} : \Omega \to \{0,1\}^n$. These indicator functions are very important throughout the present work. They can be used for computing categorical histograms (continuous variables) or facies probability maps (discrete variables) with respect to the prior density ρ:

$$q_i = \int_{\mathcal{M}} \mathbf{1}_{f_i}(\mathbf{m}) \rho(\mathbf{m}) d\mathbf{m}, \quad i = 1, \ldots, s. \tag{3.3}$$

As the random vectors $\mathbf{1}_{f_i}$ are defined on the n-dimensional space $\{0,1\}^n$, the quantities q_i can be considered as n-dimensional vectors with values in $[0,1]^n$. We prefer however to understand them as maps $q_i : \{1, \ldots, n\} \to [0,1]$ such that $q_i(j)$ is the prior probability of the parameter m_j to take a value within the category f_i. The set $Q = \{q_1, \ldots, q_s\}$ is called the **categorical prior distribution** with $Q(j) = \{q_1(j), \ldots, q_s(j)\}$ being the categorical histogram of the model parameter m_j. In practice, it is often not possible to compute the quantities in equation (3.3) precisely. We assumed that the model simulator produces realizations according to the prior density ρ. This means that if there is a large set of independent models $\{\mathbf{m}^1, \ldots, \mathbf{m}^N\}$, the law of large numbers (LLN) (c.f. Durrett [2010]) suggests to approximate q_i from a weighted sum such as

$$q_i \approx \frac{1}{N} \sum_{j=1}^{N} \mathbf{1}_{f_i}(\mathbf{m}^j). \tag{3.4}$$

By increasing the number of samples, the above approximation can reach any desired accuracy such that it is not needed to derive the analytical expression of equation (3.3).

3.2 Examples of Conditional Sampling

The goal of this section is not to give a complete overview of existing conditional sampling strategies. It only aims to illustrate the theoretical concepts of the previous section and explain conditional simulators that can be used for the applications in chapter 5.

3.2.1 Discrete Selection Sampler

A discrete selection sampler is a machine that randomly selects one element from a set of s-items. Let's label the elements as integers and denote them by $\{1, \dots, s\}$.

Model Space

The model space $\mathcal{M} = \big\{1, \dots, s\big\}$ is a one-dimensional finite manifold. Any model $\mathbf{m} = \big\{m\big\}$ is generated from a scalar random variable $\mathbf{M} = \{M\}$ such that $M(\omega) \in \{1, \dots, s\}$. A natural choice for the volume measure on a finite space is to associate constant volume density of $\frac{1}{s}$ to each member in $\mathcal{M}$.

A model can be conditioned by simply enforcing $m = i$ for an integer value $i \in \{1, \dots, s\}$. In this case, the hard conditioning data $\mathbf{hd} = (I, \mathbf{v})$ is defined by $I = \{1\}$ (for the single coordinate in $\mathbf{m}$) and $\mathbf{v} = \{i\}$. But a permanent conditioning of such a model fully imposes any information and reduces $\mathcal{M}$ to be a one-element set. However, we will still see in chapter 5 that the discrete conditional sampler can be used in combination with other simulators.

Prior Probability

The prior density is defined from a discrete set of probability values $\{p_1, \dots, p_s\}$ where $\mathbb{P}(M = i) = p_i$, so that its volumetric counterpart ρ reads

$$\rho(\mathbf{m}) = s p_i.$$

This is clearly a normalized density, because the total measure is

$$\int_{\mathcal{M}} \rho(\mathbf{m}) d\mathbf{m} = \sum_{i=1}^{s} s p_i \frac{1}{s} = 1.$$

However, there is no need for the volume density to be normalized over the model manifold. Equivalently, the volume measure could be equal to 1 for any model in $\mathcal{M}$, so that the volumetric probability density would be rescaled such that $\rho(\mathbf{m}) = p_i$ for all $i = 1, \dots, s$.

Categorical Prior Distribution

Categories of m can be obtained by defining facies values

$$\mathcal{F} = \{f_1, \ldots, f_s\}$$

with

$$f_i = \{i\}, \qquad \text{for } i = 1, \ldots, s.$$

In this simple case, the analytical expressions of the categorical prior distributions in $Q = \{q_1, \ldots, q_s\}$ are available. As there is only one single coordinate in $\mathcal{M}$ they read

$$\begin{aligned} q_i &= \int_{\mathcal{M}} \mathbf{1}_{f_i}(\mathbf{m})\rho(\mathbf{m})d\mathbf{m} \\ &= \sum_{j=1}^{s} \mathbf{1}(j = i)p_j = p_i, \end{aligned}$$

so that $Q = \{p_1, \ldots, p_s\}$ represent the original (prior) probabilities for selecting members in $\mathcal{M}$.

3.2.2 Multiple-point Statistics

In the past decades, a huge effort has been put into the development of Multiple-point Statistics (MPS) algorithms. The idea of MPS is to produce geostatistical models by using analog information from a training image (TI). An extensive guide for many existing algorithms can be obtained from Mariethoz and Caers [2014]. Broadly speaking, they can be classified into *pattern-based* and *pixel-based* methods. The basic idea for the second type was first illustrated by Guardiano and Srivastava [1993] and has been expanded by a large number of contributors. Even within the field of pixel-based MPS, there are several different algorithms, each of which having its own characteristics, advantages, and disadvantages. In this example we will focus on a very powerful algorithm called *Direct Sampling (DS)*, introduced by Mariethoz et al. [2010b] and further developed by Straubhaar [2011].

Pixel-based methods require a spatial subdivision of the modeling domain into a finite number of $n \in \mathbb{N}$ elements (**pixels** or equivalently **nodes**). The union of all pixels is called the **simulation grid**. In practice such a grid often represents a discretization of a one, two, or three dimensional area. Similarly to the example 2.1.3, let the n pixels be represented by indices $1, \ldots, n$. DS generates random realizations of spatial heterogeneities by reproducing multiple-point statistics from the TI as follows. Let S_n be the set of all permutations on the n pixels. The algorithm starts by choosing a simulation path $\varsigma \in S_n$, before sequentially assigning simulation values to each pixel in the order

$$\varsigma(1) \to \varsigma(2) \to \cdots \to \varsigma(n).$$

To simulate a value for a node $\varsigma(j)$, a pattern of some *informed* neighbors is extracted and used (in terms of conditional probabilities) to obtain a simulation value from the training image. Therefore, for a fixed random path ς, the simulation at $\varsigma(j)$ only depends on the previously informed nodes $\varsigma(1), \ldots, \varsigma(j-1)$. For pixel-based simulation, the distinction into 'continuous' and 'discrete' realizations must be qualified. 'Continuous' simulations are often understood such that the nodes can possibly obtain any simulation value from a dense range. But all values in an MPS realization are directly extracted from the TI, which in turn is also discretized into a finite number of pixels. This means that the set of different simulation values is finite so that the terminology of 'continuous' simulations is not fully precise. However, they can still be understood as (step-wise) approximations of a continuous field.

Model Space

Any DS realization can be transformed into a model $\mathbf{m} = \{m_1, \ldots, m_n\}$ by putting the simulation value from pixel j into the parameter m_j. Following this line, the model space $\mathcal{M}$ then contains all possible outcomes, while the random vector $\mathbf{M}$ embeds the stochastic behavior of the DS algorithm. As there is a finite number of different values in the training image, there is a finite number of different realizations in $\mathcal{M}$. A suitable volume measure can be obtained by putting it equal to 1 for any point in the model space.

It was mentioned in example 2.1.6 that a conditioning $\mathbf{hd} = (I, \mathbf{v})$ can reduce the number of parameters in $\mathbf{m}$. This means that whenever the algorithm obtains p conditioning index-value pairs, DS simply runs through the remaining $n - p$ nodes by considering the hard conditioned pixels as 'informed'. For this reason, the models $\mathbf{m}$ can be considered as to contain only $n - p$ parameters. However, the hard conditioning is taken into account whenever a pixel in the neighborhood of the imposed values is simulated. Roughly speaking, it might be helpful to consider $\mathbf{hd}$ as a set of 'boundary conditions' that is included within the DS tool.

Prior Probability

Capturing the analytic expression of the volumetric probability density for MPS realizations is usually not possible. We only know that for continuous and discrete simulations, the model space is discrete so that ρ can be described from a countable set of probability values $\{\rho_1, \ldots, \rho_{|\mathcal{M}|}\}$. In this notation, $|\mathcal{M}|$ denotes the number of different realizations in $\mathcal{M}$. This density is usually not uniform because some spatial structures may appear more (resp. less) frequently in the training image, what makes the simulation of some models more (resp. less) probable.

Categorical Prior Distribution

Partitions $\mathcal{F} = \{f_1, \ldots, f_s\}$ are obtained depending on the nature of the simulation. If there are continuous values allowed, f_i is a dense interval, while in the discrete case it usually represents a facies value. As there is no analytical expression for ρ, the quantities in Q usually need to be computed empirically. Let us consider an example based on the famous training image of Strebelle [2002] shown in figure 3.3. The image is discretized into 250×250 pixels with

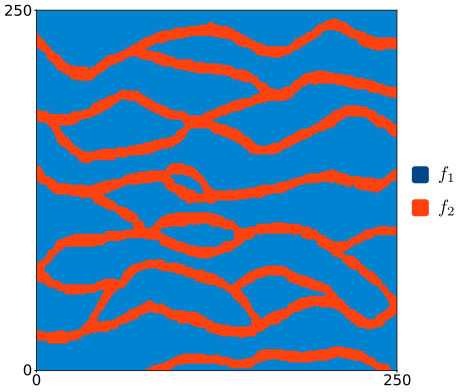


Figure 3.3: Training image

two different facies values (blue for f_1 and red for f_2). In the geostatistical framework, the two facies types are often associated to two different values of some petrophysical properties (e.g. permeability, specific storage, porosity, etc.). If, for example, we want to use this training image for modeling spatial groundwater permeability maps, we choose a set of two different permeability values $\{k_1, k_2\}$ and define the one-to-one correspondence

$$k_i \leftrightarrow f_i, \quad i = 1, 2.$$

Thanks to this bijection between physical parameters and facies values, the petrophysical permeabilities are uniquely defined by the facies map of an MPS realization and vice-versa. In this sense, a model can be understood as a spatial image defined on the simulation grid.

For this example we define a two-dimensional grid of 50×50 nodes, with totally $n = 2500$ pixels. If the DS algorithm is trained to produce a large number of unconditioned simulations, the categorical prior distribution $Q = \{q_1, q_2\}$ can be approximated as in equation (3.4). The three top figures in 3.4 show the empirical distribution computed from 10^4 unconditioned simulations. If the parameters in the MPS algorithm are not selected too strictly, the unconditioned prior probabilities match the facies proportions of the training image. This can be helpful whenever the empirical computation of Q wants to be avoided. In

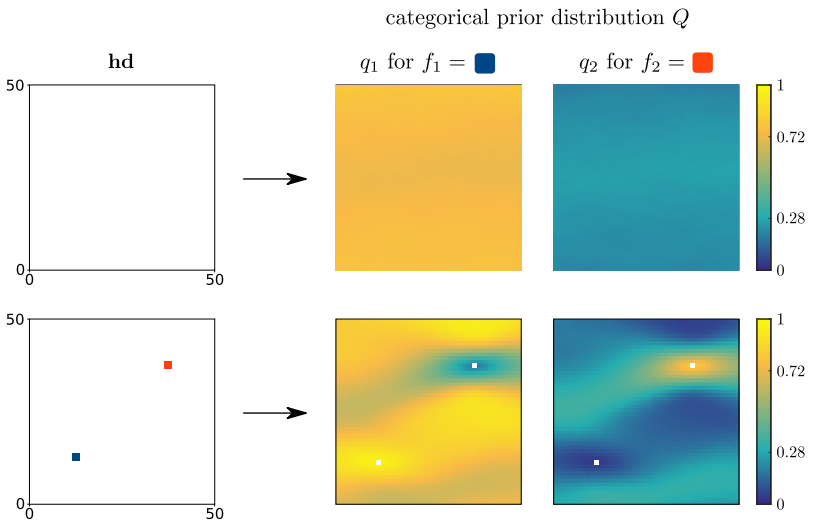


Figure 3.4: Categorical prior density maps for unconditioned (top) conditioned (bottom) simulations.

the above case, the facies probabilities in the training image are 0.72 for f_1 and 0.28 for f_2. These values coincide with the unconditioned category probability maps in figure 3.4. However, extracting facies proportions from the training image is usually not possible for conditioned simulations. A hard conditioning for the above problem can be obtained from an index set $I \subset \{1, \dots, n\}$ and values $\mathbf{v}$ such that v_i is either 'red' or 'blue'. Let us condition the simulations on two positions; at $(12, 12)$ with 'blue' and at $(38, 38)$ with 'red'. If we consider a x-through-y numbering of the pixels, the **hd** is defined by $I = \{526, 1888\}$ and $\mathbf{v} = \{\text{'blue'}, \text{'red'}\}$. The bottom figures in 3.4 show the hard conditioning data and the recomputed categorical densities from 10^4 conditioned realizations. It can be seen that the probability maps are no longer constant. The conditioning has a strong influence on the close neighborhood of the imposed values. For this reason, we suggested to understand the hard data as boundary conditions that impact the simulation process.

Let us recall that in this second case, the model space is reduced by 2 dimensions. This means that the coordinates $\{526, 1888\}$ are removed from the model space and any realization $\mathbf{m}$ only contains 2498 parameters. However, in the description of the underlying physical system, the two conditioned pixel values are still necessary. Therefore, the model space $\mathcal{M}$ only contains 'uncertain' model parameters.

Chapter 4

PoPEx Algorithm

Solving an inverse problem is the task of characterizing the posterior measure function in equation (2.7). There are many different methods available that can be used for approximately represent a measure function over a manifold. *Monte Carlo (MC)* is a terminology for a class of random sampling methods that are often employed in cases where it is difficult or impossible to use other algorithms. They can be trained to explore highly complex manifolds and measure functions by taking a large number of coupled degrees of freedom into account. Unfortunately a major constraint is often that their usage requires an unsuitably large computational effort. For this reason it is of paramount importance to design efficient and parallelizable sampling schemes.

4.1 Monte Carlo Sampling

A primitive strategy of representing the posterior measure function is to approximate the 'most probable' model (e.g. maximum a posteriori (MAP) or maximum likelihood (ML)) and estimate the dispersion around that point (e.g. by a linear covariance matrix). This approach is suitable if the marginal uncertainties (modelization, observation, prior) are close to be Gaussian. In many applications however, the considered manifolds are discrete or even finite, what makes this assumption unrealistic. Furthermore, the posterior density might be multi-modal such that an optimization algorithm only finds one local maximum among a large number of equally important scenarios.

Example 4.1.1 (Multi-modal posterior distribution)**.** *Let the posterior measure function be composed of n well separated univariate Gaussian distributions on $\mathcal{M} = \mathbb{R}$ such that $\mathbf{m} = \{m\}$ and*

$$\sigma(\mathbf{m}) \propto \sum_{i=1}^{n} \exp\left(-\frac{1}{2}(m - \mu_i)^2\right).$$

Let's assume that 'well separated' is understood such that the distance between a pair μ_i and μ_j is at least 6, i.e. 6 times the standard deviation of 1. Depending on the initial value, a simplistic optimization algorithm would converge to one of the mean values and estimate the uncertainties from the local behavior of σ around $\mathbf{m}^{\mathrm{MAP}} = \{\mu_i\}$. In the above case, the posterior density is locally Gaussian. A simplistic approximation of the 99.5% confidence interval arround $\mathbf{m}^{\mathrm{MAP}}$ would result in an interval $[\mu_i - 3, \mu_i + 3]$ (cf. top figure of 4.1). For

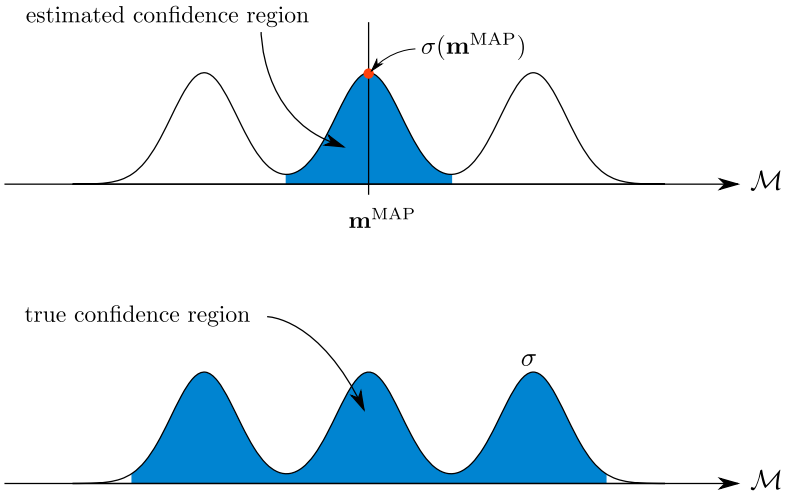


Figure 4.1: Illustration of a serious underestimation of the confidence region.

different values of $n \in \mathbb{N}$, the fraction of the estimated and the true confidence interval behaves like the function $\frac{1}{n}$ (cf. bottom figure of 4.1). But for $n > 1$ this result is unsatisfactory. Therefore, it is quite easy to find examples where simplistic estimation techniques fail to have any interesting meaning.

Monte Carlo methods collect a large number of random models that are able to provide better approximations and can be used to answer all sorts of interesting questions. The algorithms are usually trained such that regions with same measure values have equal chance to get explored. This is especially useful when working with multi-modal probability densities. Similar regions are sampled with the same frequency and therefore, it investigates all the important areas in the model space.

Example 4.1.2 (continued from 4.1.1)**.** *The posterior measure function in example 4.1.1 is defined from n similar Gaussian density functions. For $i = 1, \ldots, n$ let A_i be defined as $A_i = \big(\mu_i - 3, \mu_i + 3\big)$. The Gaussian density functions are well separated so that these intervals are all distinct. Whenever a sampling scheme is trained to produce models according to the density σ, it*

explores each region A_i with frequency

$$\mathbb{P}(A_i) \approx \frac{0.995}{n}.$$

The probability of not generating any model within A_i decreases exponentially with the number of samples. It follows that from a sufficiently large collection (say more that $100n$ realizations) it is possible to compute an accurate approximation of σ. A visual illustration of a Monte Carlo sampling according to σ is shown in figure 4.2. The red bars represent models that have been gen-

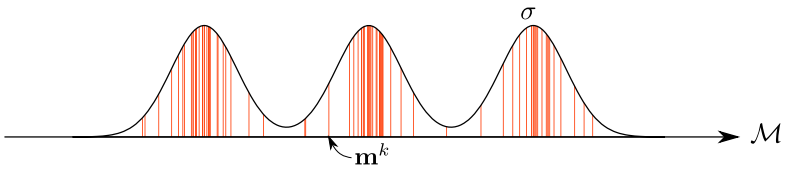


Figure 4.2: Illustration of a Monte Carlo sampling according to a multi-modal density.

erated during the random process. It can be seen that each region is explored with approximately the same frequency. In this case, excessive sampling of low-probability regions is avoided. For the development of efficient Monte Carlo schemes, this criterion is vital.

When using MC methods for solving an inverse problem, we do not only want to compute an approximation of σ. What is usually more important, is to evaluate the posterior measure of some events, or equivalently, the computation of integrals

$$\mu = \int_{\mathcal{M}} f(\mathbf{m})\sigma(\mathbf{m})d\mathbf{m}, \tag{4.1}$$

where $f : \mathcal{M} \to \mathbb{R}^d$ is any measurable function of interest. This is where Monte Carlo methods can become very useful. From a set of N independent samples $\mathbf{m}^1, \ldots, \mathbf{m}^N \sim \sigma$, the value of μ can be approximated by

$$\hat{\mu} = \frac{1}{N}\sum_{k=1}^{N} f(\mathbf{m}^k). \tag{4.2}$$

The quality increases with N so that any desired level of accuracy can be reached. But producing a large number of samples in a high dimensional model space can result in serious efficiency issues.

Importance Sampling

Another method for studying integral expressions as in (4.1) is called *importance sampling (IS)* [Hesterberg, 2003, Liu, 2008, Rubinstein and Kroese, 2016].

This family of algorithms can be used whenever sampling from another distribution than σ is more suitable. Key to this approach is the so called *sampling* or *proposal distribution* ϕ. Notice first that if $|f(\mathbf{m})\sigma(\mathbf{m})|$ is absolutely continuous with respect to $\phi(\mathbf{m})$, i.e.

$$f(\mathbf{m})\sigma(\mathbf{m}) \neq 0 \quad \Longrightarrow \quad \phi(\mathbf{m}) > 0, \tag{4.3}$$

then it is possible to rewrite the integral in equation (4.1) as

$$\mu = \int_{\mathcal{E}} \frac{f(\mathbf{m})\sigma(\mathbf{m})}{\phi(\mathbf{m})}\phi(\mathbf{m})d\mathbf{m},$$

where $\mathcal{E} = \{\mathbf{m} \in \mathcal{M} : \phi(\mathbf{m}) > 0\}$. Similarly to standard MC methods, an IS algorithm samples N independent realizations $\mathbf{m}^1, \ldots, \mathbf{m}^N \sim \phi$ and approximates μ by

$$\hat{\mu} = \frac{1}{N}\sum_{k=1}^{N} f(\mathbf{m}^k)\frac{\sigma(\mathbf{m}^k)}{\phi(\mathbf{m}^k)}. \tag{4.4}$$

It can be seen that for all $k = 1, \ldots, N$, the quantities $f(\mathbf{m}^k)$ are weighted by the ratios $\frac{\sigma(\mathbf{m}^k)}{\phi(\mathbf{m}^k)}$. Roughly speaking, this ratio compares the importance of $\mathbf{m}^k$ (measured by $\sigma(\mathbf{m}^k)$) with the probability of being sampled (measured by $\phi(\mathbf{m}^k)$). Figure 4.3 illustrates a continuous setup where a model $\mathbf{m}^k$ is such that $\sigma(\mathbf{m}^k) > \phi(\mathbf{m}^k)$. An algorithm that employs a random sampling according

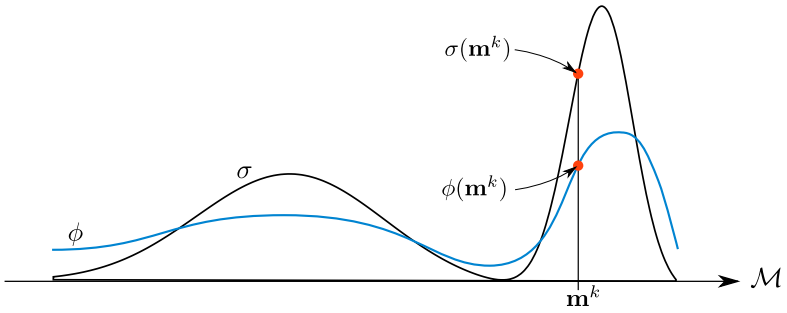


Figure 4.3: Importance sampling: examine a density σ by sampling from ϕ.

to ϕ, does not visit the neighborhood of $\mathbf{m}^k$ sufficiently often. Therefore, the impact of $f(\mathbf{m}^k)$ in $\hat{\mu}$ must be increased. It happens that the right amount of enforcement is equal to $\frac{\sigma(\mathbf{m}^k)}{\phi(\mathbf{m}^k)}$.

Example 4.1.3 (Prior sampling)**.** *In chapter 3 we discussed conditional sampling procedures that generate random realization according to the prior distribution ρ. This means that the proposal density ϕ is equal to the prior density*

and the ratio becomes

$$\frac{\sigma(\mathbf{m}^k)}{\rho(\mathbf{m}^k)} \;=\; \frac{c\rho(\mathbf{m}^k)L(\mathbf{m}^k)}{\rho(\mathbf{m}^k)} \;=\; cL(\mathbf{m}^k).$$

Therefore, for N models $\mathbf{m}^1, \ldots, \mathbf{m}^N \sim \rho$ the estimator $\hat{\mu}$ reads

$$\hat{\mu} = \frac{c}{N} \sum_{k=1}^{N} f(\mathbf{m}^k)L(\mathbf{m}^k).$$

Although sampling according to the prior distribution performs an extensive exploration of the entire model space, it is usually very inefficient. In a complex space $\mathcal{M}$, the subset of realizations with high posterior measure has often very small volume. A plain importance sampling according to ρ may take a large amount of realizations to have even one point inside the important region. Similar problems arise in the field of rare event simulations. For these reasons, a more efficient proposal distribution must be chosen.

Owen [2013] provides a theorem for computing the variance of the estimate $\hat{\mu}$ such that

$$\begin{aligned} \mathrm{Var}(\hat{\mu}) &= \frac{1}{N}\left(\int_{\mathcal{E}} \frac{\big(f(\mathbf{m})\sigma(\mathbf{m})\big)^2}{\phi(\mathbf{m})} d\mathbf{m} - \mu^2\right) \\ &= \frac{1}{N}\int_{\mathcal{E}} \frac{\big(f(\mathbf{m})\sigma(\mathbf{m}) - \mu\phi(\mathbf{m})\big)^2}{\phi(\mathbf{m})} d\mathbf{m}. \end{aligned} \tag{4.5}$$

This is a very useful result for selecting good sampling distributions. Especially in the second equation we can see that successful proposals are the ones that make $f(\mathbf{m})\sigma(\mathbf{m}) - \mu\phi(\mathbf{m})$ close to zero or equivalently that are nearly proportional to $f(\mathbf{m})\sigma(\mathbf{m})$. It can be shown [Owen, 2013] that an optimal proposal density is given by

$$\phi^{\mathrm{opt}}(\mathbf{m}) = \frac{|f(\mathbf{m})\sigma(\mathbf{m})|}{\mu}.$$

This expression is not very practical however. For its construction it is needed to know the exact value of μ; but characterizing μ is precisely the original task of the importance sampling procedure. Nevertheless ϕ^{opt} illustrates how importance sampling can succeed or fail. From the second integral in equation (4.5) we can see that in the regions where ϕ is small, any lack of proportionality in the numerator is greatly amplified. It is advantageous for ϕ to have spikes wherever $|f(\mathbf{m})\sigma(\mathbf{m})|$ does. Or likewise, if we want ϕ to work for many different functions f, it can be beneficial to design it as close as possible to the posterior density function σ.

Another requirement on ϕ is coming from the ratio quantities $\frac{\sigma(\mathbf{m})}{\phi(\mathbf{m})}$. Let's consider a normalized posterior measure σ and a random vector $\mathbf{M}$ with density

ϕ. It is clear that the expectation of $\frac{\sigma(\mathbf{M})}{\phi(\mathbf{M})}$ is equal to 1 because

$$\int_{\mathcal{E}} \frac{\sigma(\mathbf{m})}{\phi(\mathbf{m})}\phi(\mathbf{m})d\mathbf{m} \;=\; 1.$$

A common requirement on this ratio is to have finite second moment, or equivalently

$$\int_{\mathcal{E}} \left(\frac{\sigma(\mathbf{m})}{\phi(\mathbf{m})}\right)^2 \phi(\mathbf{m})d\mathbf{m} \;=\; \int_{\mathcal{E}} \frac{\sigma(\mathbf{m})}{\phi(\mathbf{m})}\sigma(\mathbf{m})d\mathbf{m} \;<\; +\infty. \tag{4.6}$$

This is where the popular expression "ϕ must have heavier tails that σ" comes from. It refers to the requirement that in a non-compact model manifold $\mathcal{M}$, the set

$$A := \{\mathbf{m} \in \mathcal{E} \,:\, \sigma(\mathbf{m}) > \phi(\mathbf{m})\}$$

must be compact and does not allow the ratio $\frac{\sigma(\mathbf{m})}{\phi(\mathbf{m})}$ to go to infinity. These two requirements are sufficient for (4.6) to hold because for $\frac{\sigma(\mathbf{m})}{\phi(\mathbf{m})}$ being bounded by a constant $C \geq 1$, σ-almost everywhere, it follows that

$$\begin{aligned}\int_{\mathcal{E}} \frac{\sigma(\mathbf{m})}{\phi(\mathbf{m})}\sigma(\mathbf{m})d\mathbf{m} \;&=\; \int_A \frac{\sigma(\mathbf{m})}{\phi(\mathbf{m})}\sigma(\mathbf{m})d\mathbf{m} + \int_{A^c} \frac{\sigma(\mathbf{m})}{\phi(\mathbf{m})}\sigma(\mathbf{m})d\mathbf{m} \\ &\leq\; \int_A C\sigma(\mathbf{m})d\mathbf{m} + \int_{A^c} \sigma(\mathbf{m})d\mathbf{m} \;\leq\; C\end{aligned}$$

where A^c is the complement of A in $\mathcal{E}$. Note that even if ϕ has heavier tails than σ, the constant C and therefore the variance of the ratio $\frac{\sigma(\mathbf{M})}{\phi(\mathbf{M})}$ can still be very large. This happens when there are regions within A that have high posterior measure but are unlikely to be sampled, i.e. when $\frac{\sigma(\mathbf{m})}{\phi(\mathbf{m})}$ is large on a σ-important subset. Importance sampling is therefore often successful when it is easy to sample from ϕ which itself is closely inspired by the posterior measure σ.

Sometimes it is only possible to compute unnormalized measure values for σ and ϕ, that is $\sigma_u = c_1\sigma$ and $\phi_u = c_2\phi$. In this case, the estimator $\hat{\mu}$ in equation (4.4) can be replaced by the so called *self-normalized importance sampling estimate* $\hat{\mu}_{\text{sn}}$ that reads

$$\hat{\mu}_{\text{sn}} \;=\; \frac{\sum_{k=1}^N f(\mathbf{m}^k)w_k}{\sum_{k=1}^N w_k}, \tag{4.7}$$

where $w_k = \frac{\sigma_u(\mathbf{m}^k)}{\phi_u(\mathbf{m}^k)}$ for any $k = 1, \ldots, N$. The results for the self-normalized estimator are similar to the ones for $\hat{\mu}$ (cf. Owen [2013]). This is very intuitive, because the weights $w_1, \ldots, w_N$ can be interpreted as random variables that are independent and identically distributed. Their mathematical expectation is equal to $\frac{c_1}{c_2}$ and therefore the sum $\frac{1}{N}\sum_{k=1}^N w_k$ converges to $\frac{c_1}{c_2}$. Furthermore,

$\frac{1}{N}\sum_{k=1}^{N} f(\mathbf{m}^k)w_k$ converges to $\frac{c_1}{c_2}\mu$ and Slutsky's theorem (e.g. Panaretos [2016]) suffices to conclude that

$$\hat{\mu}_{\text{sn}} \;=\; \frac{\frac{1}{N}\sum_{k=1}^{N} f(\mathbf{m}^k)w_k}{\frac{1}{N}\sum_{k=1}^{N} w_k} \;\longrightarrow\; \mu.$$

In the framework of inverse problems, the self-normalized estimator is very useful, because the normalization constant c in equation (2.7) is often unknown.

Adaptive Importance Sampling

As the construction of suitable sampling densities ϕ can be a huge challenge, Naylor and Smith [1988] and Oh and Berger [1992] proposed to start with an initial guess and iteratively improve it. They called their technique *adaptive importance sampling (AIS)*. Up to date, there are many advanced AIS developments that are hard to explain because they are still mostly intuitive and highly related to their respective field of application.

The general idea of AIS is to iteratively change the sampling distribution ϕ. This has the potential to generate a sequence of proposal distributions $\phi_1, \phi_2, \ldots,$ that converges to a (closely) optimal proposal density. Integral expressions are approximated as in equation 4.4 and 4.7 but with weights w_k that take the different proposal measures into account, i.e.

$$w_k = \frac{\sigma(\mathbf{m}^k)}{\phi_k(\mathbf{m}^k)}, \quad \text{for} \;\; k = 1, \ldots, N.$$

When using an AIS scheme together with self-normalized estimation, it is usually important that the different sampling densities ϕ_k have the same normalization constant, i.e. c_2. We will see that in the PoPEx algorithm this requirement naturally disappears (cf. equation (4.18)).

4.2 PoPEx - Posterior Population Expansion

We introduced the general ideas of the PoPEx sampling scheme in Jäggli et al. [2017] and further refined them in Jäggli et al. [2018]. The present section provides a complete discussion of the PoPEx algorithm in its most recent state. Empirical results are supported by an original theoretical description of the sampling scheme.

PoPEx is an adaptive importance scheme that uses conditional sampling to generate a large number of models $\mathbf{m}^1, \ldots, \mathbf{m}^N$ that represent the posterior probability density in equation 2.7, i.e. $\sigma(\mathbf{m}) = cL(\mathbf{m})\rho(\mathbf{m})$. The sampling procedure requests to compute $L(\mathbf{m}^k)$ for every model $\mathbf{m}^k$ what can be very intensive in terms of computational costs. For this reason, it is important to make the sampling as efficient as possible. Each simulation of a new model $\mathbf{m}^{k+1}$ is therefore guided by *all* the previous samples $\mathbf{m}^1, \ldots, \mathbf{m}^k$ and exploits

information of *all* previous steps. For doing so, a learning scheme ensures that the sampling of $\mathbf{m}^{k+1}$ is strongly guided by 'good' models with high posterior values. Key for the learning procedure are two information maps (denoted by P^k and $D(P^k||Q)$, see below) that are updated iteratively and aim to quantify the meaning of 'good' models. Furthermore, these maps are used to transfer information from $\mathbf{m}^1, \ldots, \mathbf{m}^k$ to $\mathbf{m}^{k+1}$ by a synthetic set of conditions (denoted by $\mathbf{hd}^k$, see below) imposed on the new model. This is a very brief introduction that only gives a rough overview of the basic concepts of PoPEx. Nevertheless, it can be beneficial to throw a first glance on the visualization of the algorithm in section 5.1.

Set of models $\mathcal{M}^k$

It was mentioned in chapter 3 that the model space $\mathcal{M}$ collects all possible realizations of the conditional simulation tool. Each model $\mathbf{m} \in \mathcal{M}$ has n parameters and is denoted by $\mathbf{m} = \{m_1, \ldots, m_n\}$. Sampling a model space for solving an inverse problem means to iteratively produce a finite number of N realizations

$$\mathbf{m}^1 \to \mathbf{m}^2 \to \cdots \to \mathbf{m}^N,$$

that characterize (in some way) the posterior distribution. After each iteration $k = 1, \ldots, N$, the models can be assembled within the collection

$$\mathcal{M}^k = \left\{\mathbf{m}^1, \ldots, \mathbf{m}^k\right\}.$$

This is a rather small subset of the complete model space $\mathcal{M}$. It will be used for approximating quantities of interest (learning scheme in PoPEx, integral predictions, maximum probability models, etc.).

Probability maps Q and P^k

The categorical prior distribution Q has been introduced in equation (3.3) as a set that collects all the prior probabilities for the model type specific categories. It was mentioned that Q can be approximated from a large number of independent simulations. In the corresponding equation (3.4), every term $\mathbf{1}_{f_i}(\mathbf{m}^j)$ is weighted equally by $\frac{1}{N}$. Similarly, we want to produce a second collection $P^k = \left\{p_1^k, \ldots, p_s^k\right\}$ where the weights are more inspired by the posterior or alternatively the likelihood measure. The idea is to generate a categorical distribution by favoring 'good' models over 'bad' ones. From a set of normalized weights

$$\widetilde{\Sigma}^k = \left\{\tilde{\sigma}_1^k, \ldots, \tilde{\sigma}_k^k\right\}$$

let p_i^k be defined as

$$p_i^k = \sum_{j=1}^{k} \mathbf{1}_{f_i}(\mathbf{m}^j)\tilde{\sigma}_j^k, \qquad \text{for } i = 1, \ldots, s. \tag{4.8}$$

The superscript k in the notation p_i^k indicates the number of realizations that have been used in its computation. For this definition and any inherited quantity, the set of weights $\left\{\tilde{\sigma}_1^k, \ldots, \tilde{\sigma}_k^k\right\}$ pays a crucial role. In fact they decide what a 'good' or a 'bad' model is. First of all, it is important to notice that P^k will have a significant impact on the sampling density ϕ_k (used for generating the next model $\mathbf{m}^{k+1}$) which itself should be closely inspired by the posterior density (cf. section 4.1). 'Being inspired by' means that it is not necessary for the weights $\tilde{\sigma}_j^k$ to be proportional to the posterior measure values $\sigma(\mathbf{m}^j)$. It only signifies that there must be a strong link in between (also see the discussion in chapter 6 for more details). An obvious choice is to define the weights in $\widetilde{\Sigma}^k$ to be proportional to the likelihood values, i.e.

$$\tilde{\sigma}_j^k = \frac{L(\mathbf{m}^j)}{\sum_{r=1}^k L(\mathbf{m}^r)}, \qquad j = 1, \ldots, k.$$

In fact, this definition will be used for the rest of the present work. However, it is not necessary to do so and one can think of many other appropriate choices. Some alternative ideas are discussed in chapter 6. Nevertheless, selecting the weights $\tilde{\sigma}_j^k$ proportional to the likelihood values $L(\mathbf{m}^k)$ is often suitable. First, we notice that in many setups the prior density is fairly flat. In this case, the likelihood measure and therefore the weights $\tilde{\sigma}_j^k$ are almost proportional to the posterior density function. Secondly, if the models $\mathbf{m}^k$ are sampled according to the prior density ρ (cf. example 4.1.3) then the maps p_i^k are unbiased estimators of the posterior category probabilities (cf. importance sampling and equation (4.4)). The tilde notation in $\widetilde{\Sigma}^k$ indicates that a normalization has taken place; a convention that will be used throughout this work. There are two different kinds of normalization that will be used. Above, the total weight was computed by summing all likelihood values from the previous iterations. This act must be renewed whenever a new model $\mathbf{m}^{k+1}$ is sampled. Secondly, we will consider model type specific maps as for example $D(P^k||Q)$. The normalization, denoted by $\widetilde{D}(P^k||Q)$, is then performed through all model indices. The resulting maps can be interpreted as a discrete probability density defined individually for each model type.

Let's consider some more details about the weights in P^k. It is important to perceive the consequences of weighting the summands by the values $\tilde{\sigma}_j^k$. If $\mathbf{m}^{j_0}$ is a model with a large likelihood value (with respect to the other ones), this means that some patterns in $\mathbf{m}^{j_0}$ may be very important. Therefore, the probability maps in equation (4.8) are formed by weighting 'good' category patterns more heavily than 'bad' ones. Consequently, the distribution P^k may be able to provide information that can be used to generate 'good' models. But at this point it is unclear where this information can be found and how it could be used. The answer to this question lies in the relation between Q and P^k; let's see how.

Kullback-Leibler divergence $D(P^k||Q)$

Kullback and Leibler [1951] introduced a measure called *Kullback-Leibler divergence (KLD)* that compares two probability distributions. It computes how a candidate probability diverges from an expected one. This is precisely what is needed to measure the information content of P^k with respect to Q. In other words, the Kullback-Leibler divergence can be used to identify index values, where the category probabilities in P^k are 'extreme' with respect to Q. This divergence is computed individually for each model type and is given by

$$D(P^k||Q) = \sum_{i=1}^{s} p_i^k \log\left(\frac{p_i^k}{q_i}\right). \tag{4.9}$$

Whenever $q_i(j) > 0$ for all $i = 1, \dots, s$ and all $j = 1, \dots, n$, this equation is well defined. But let's assume that there is $i \in \{1, \dots, s\}$ and an index j with $q_i(j) = 0$. This means that it is impossible for the conditional tool to produce a model $\mathbf{m}$ where the value m_j falls into the i-th category. From equation (4.8) it follows that $p_i^k(j)$ must vanish as well. In short, $q_i(j) = 0$ implies $p_i^k(j) = 0$, and the corresponding terms in equation (4.9) can be ignored. A brief comment on the relation between p_i^k and q_i may help to better understand the meaning of $D(P^k||Q)$. Both maps use the same indicator functions, but are weighted differently. $D(P^k||Q)$ provides a information map, that indicates how surprising the category patterns become, when they are weighted by the likelihood values. As mentioned earlier, it is possible to normalize the Kullback-Leibler divergence map individually for each model type. The rescaled map is denoted by $\widetilde{D}(P^k||Q)$ and can be interpreted as a discrete probability density that allows to localize model indices that are most informative with respect to Q.

Hard conditioning data $\mathbf{hd}^k$

In chapter 3 we considered hard conditioning data that is used for building the model space $\mathcal{M}$. It was mentioned that whenever such a set of values is available, it can greatly help to reduce the complexity of $\mathcal{M}$. In this section however, we will consider sets of *synthetic hard conditioning data* that do not modify tho model space. Instead, they are learned during the PoPEx algorithm and aim to accelerate the sampling procedure. These synthetic sets are randomly updated in each iteration k and denoted by $\mathbf{hd}^k = (I, \mathbf{v})$. Recall that in this notation, I is a set of model indices and $\mathbf{v}$ embeds the corresponding conditioning values. It is helpful to understand $\mathbf{hd}^k$ to be model type specific so that it can be considered individually. In every iteration k, it is deduced from the previous realizations $\mathbf{m}^1, \dots, \mathbf{m}^k$ and used for generating $\mathbf{m}^{k+1}$. A reliable set of hard conditioning data may enhance the chance to generate a new model $\mathbf{m}^{k+1}$ with high likelihood value $L(\mathbf{m}^{k+1})$.

Considering the previous explanations, it seems natural to sample an index set $I \subset \{1, \dots, n\}$ of hard conditioning locations (*where* conditioning should ap-

ply) from the normalized Kullback-Leibler information $\widetilde{D}(P^k||K)$. This means that indices where P^k is importantly different from Q are extracted more frequently. For every selected $i \in I$, the algorithm then samples a model index $j \in \{1, \ldots, k\}$ according to $\widetilde{\Sigma}^k$, extracts the conditioning value (*which* value should be imposed) from m_i^j, and puts it into $\mathbf{v}$. Drawing model indices according to $\widetilde{\Sigma}^k$ preferentially selects 'good' models with high likelihood values. These two steps produce a synthetic set of hard conditioning data $\mathbf{hd}^k = (I, \mathbf{v})$. The number of conditioning points is defined by n_k, the cardinality of I, and changes randomly in each iteration. In the PoPEx algorithm n_k is restricted by an upper bound $n_{\max}$. This means that in the beginning of the k-th iteration, the method decides on the length of the synthetic set by drawing a random number n_k from the uniform distribution over the set $\{0, 1, \ldots, n_{\max}\}$. It is therefore possible to occasionally generate unconditioned realizations (i.e. when $n_k = 0$). The number $n_{\max}$ must also be understood to be model type specific and can be chosen individually. In summary, this means that for each model type the index set I is generated by having a fixed number $n_{\max}$ at hand, randomly draw n_k from $\{0, 1, \ldots, n_{\max}\}$ and finally sample n_k indices according to $\widetilde{D}(P^k||Q)$. Then, for each model index $i \in I$, the weights in $\widetilde{\Sigma}^k$ are used to selected a conditioning value from the previous models $\mathbf{m}^1, \ldots, \mathbf{m}^k$ and put it in $\mathbf{v}$.

In opposite to the hard conditioning data in chapter 3, the synthetic set $\mathbf{hd}^k$ does not modify the model space $\mathcal{M}$. Instead, it is deduced randomly at each iteration and represents the PoPEx learning scheme that accelerates the sampling procedure.

4.2.1 Serial PoPEx Sampling

The above concepts are combined into the PoPEx sampling algorithm. For each model type, PoPEx requires an integer $n_{\max}$ and a categorical prior distribution Q. The total number of models N (after which the algorithm stops) is also predefined and usually results from a rough estimate of the total computational time. In this case, the **serial PoPEx** algorithm is defined as in algorithm 1. Let us look in more detail at the most important steps. First of all, notice that the sampling procedure does not use the normalized posterior information σ. The algorithm only computes likelihood values for $\widetilde{\Sigma}^k$ which itself is re-normalized in each iteration. Therefore, the constant c in equation (2.7) can be ignored. The three inputs of this algorithm are $n_{\max}$ (the maximum length of $\mathbf{hd}^k$), N (the total number of samples), and Q (the categorical prior distribution). [4] The first step in every iteration $k \geq 0$ is to sample the length of synthetic hard conditioning data that is used in the simulation of $\mathbf{m}^{k+1}$. [5] Second, the function 'hd$(n_k, \mathcal{M}^k, \widetilde{\Sigma}^k, D(P^k||Q))$' computes a set of n_k hard conditioning couples $\mathbf{hd}^k = (I, \mathbf{v})$. [6 − 7] Then, the conditional sampling algorithm implemented within 'model$(\mathbf{hd}^k)$' uses $\mathbf{hd}^k$ to produce a new realization $\mathbf{m}^k$, for which the likelihood value $L(\mathbf{m}^{k+1})$ is computed. [8] At the

Algorithm 1 PoPEx (serial)

Input: $n_{\max}$, N, and Q
$k \leftarrow 0$ and $P^0 \leftarrow Q$
while $k < N$ **do**
  sample $n_k \sim U(0, n_{\max})$
  $\mathbf{hd}^k \leftarrow \mathrm{hd}(n_k, \mathcal{M}^k, \widetilde{\Sigma}^k, D(P^k||Q))$
  $\mathbf{m}^{k+1} \leftarrow \mathrm{model}(\mathbf{hd}^k)$
  $L(\mathbf{m}^{k+1}) \leftarrow \mathrm{likelihood}(\mathbf{m}^{k+1})$
  update $\mathcal{M}^k$, $\widetilde{\Sigma}^k$ and $D(P^k||Q)$
  $k \leftarrow k + 1$
end while

end of each iteration, the new model $\mathbf{m}^{k+1}$ and its likelihood value $L(\mathbf{m}^{k+1})$ are used to update $\mathcal{M}^k \to \mathcal{M}^{k+1}$, $\widetilde{\Sigma}^k \to \widetilde{\Sigma}^{k+1}$ and $D(P^k||Q) \to D(P^{k+1}||Q)$.

The great flexibility of the PoPEx algorithm can already be appreciated from the above description. The two most important computations, i.e. the generation of a model [6] and the computation of its likelihood value [7], are implemented as function calls. This opens the door to use PoPEx for a broad range of inverse problems in many different fields. For changing the problem, the only adjustments happen inside the functions 'model(·)' and 'likelihood(·)'. The easiest way to illustrate the main lines of the PoPEx sampling algorithm is to consider a simple example. For this reason, it can be beneficial to go forward to the section 5.1 and study the example and the visualization of the algorithm once again.

4.2.2 Parallel PoPEx Sampling

Every loop in the PoPEx algorithm consists of four main steps: derive a set of hard conditioning points, generate a new model, compute its likelihood value, and compute the updates for the next iteration. One strategy to parallelize this procedure is to encapsulate the first three steps in a subprocess and separate them from the last one. A master process is then able to launch such subprocesses in parallel on different CPU's. Each subprocess is fed by the current available maps and performs the enclosed steps independently. After the result of a subprocess is communicated back to the master process, this latter updates the maps and launches another subprocess. A brief overview of this workflow is presented in figure 4.4. This strategy does not quite follow the common way of organizing parallelism. It would be more standard to simultaneously launch batches of simulations. The PoPEx algorithm however, is designed differently and controls the subprocesses individually. The reason is that a batch-wise implementation may suffer from an inefficient usage of the available CPU resources because the computational time of a subprocess can vary (large amount of data in $\mathbf{hd}^k$, unrealistic model parameters, etc.). This

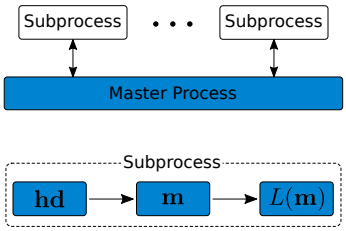


Figure 4.4: Overview of the parallelized PoPEx procedure.

means that faster computations would leave some CPU's idle until the last job of the batch has terminated. Treating each subprocess individually fully utilizes the available capacities. The pseudocodes of the **parallelized PoPEx** algorithm and the corresponding subprocess are given in the algorithms 2 and 3, respectively. In contrast to the serial PoPEx algorithm, there is one more

Algorithm 2 PoPEx (parallel)

Input: $n_{\max}$, n_{par}, N, and Q
$k \leftarrow 0$ and $P^0 \leftarrow Q$
manager $\leftarrow$ empty queue # FIFO queue
while $k < N$ **do**
  $n_m \leftarrow$ length(manager)
  if $n_m < n_{\text{par}}$ **and** $k + n_m < N$ **then**
    p $\leftarrow$ **new** subprocess
    p.start$\big($Subprocess$(\mathcal{M}^k, \widetilde{\Sigma}^k, D(P^k||Q), n_{\max})\big)$
    manager.append(p)
  end if
  p $\leftarrow$ manager.pop()
  if p.ready() **then**
    $\big(\mathbf{m}^{k+1}, L(\mathbf{m}^{k+1})\big)$ = p.get()
    update $\mathcal{M}^k$, $\widetilde{\Sigma}^k$ and $D(P^k||Q)$
    $k \leftarrow k + 1$
  else
    manager.append(p)
  end if
end while

input variable n_{par} that defines the number of parallel subprocesses that run simultaneously. The variable 'manager' appearing within the main algorithm is a FIFO ('first in first out') queue of maximal length n_{par} that maintains the communication towards the subprocesses. FIFO stands for queues where new elements are appended at the tail [9, 17] and removed from its head [11].

Algorithm 3 Subprocess

1: **Input:** $\mathcal{M}^k$, $\widetilde{\Sigma}^k$, $D(P^k||Q)$, and $n_{\max}$
2: **Output:** $\mathbf{m}^{k+1}$ and $L(\mathbf{m}^{k+1})$
3: sample $n_k \sim U(0, n_{\max})$
4: $\mathbf{hd}^k \leftarrow \mathrm{hd}(n_k, \mathcal{M}^k, \widetilde{\Sigma}^k, D(P^k||Q))$
5: $\mathbf{m}^{k+1} \leftarrow \mathrm{model}(\mathbf{hd}^k)$
6: $L(\mathbf{m}^{k+1}) \leftarrow \mathrm{likelihood}(\mathbf{m}^{k+1})$

In this regard, the lines $[5-10]$ of algorithm 2 are designed to launch n_{par} subprocesses [8] and retain corresponding handles [9]. The lines $[11-18]$ on the other hand, check the status of the first subprocess in the queue [12] and react accordingly. If it has terminated, their outputs are received [13] and the corresponding variables are updated $[14-15]$. If it is still running however, the handle is sent to the back of the queue [17]. The main motivation for appending the running subprocesses at the end is to rapidly detect and remove other jobs that have been completed. But as a consequence, reproducibility of the algorithm is not guaranteed. If reproducibility is crucial, we could simply change line [17] such that the processes are re-appended at the head of the queue and ensure that the first n_{par} workers are launched before lines $[11-18]$ may apply. The computational work in each subprocess corresponds to the lines $[4-7]$ of algorithm 1. This means that the function implementations of 'hd$(\cdot)$', 'model$(\cdot)$', and 'likelihood$(\cdot)$' are as before.

Although each subprocess obtains a KLD map for generating a new realization and computing its likelihood value, it does not update the Kullback-Leibler divergence map. This is done sequentially on the master process. In other words, whenever a subprocess has completed its computations, the results are transmitted to the master process, which then immediately updates the KLD map before starting a new subprocess. What is crucial is that the maps in P^k can be *updated* and do not need to be recomputed from scratch. This means that the required computational cost is negligible and therefore does not delay the algorithm significantly (cf. chapter 6). The advantage of this choice is that the subprocesses always obtain the most recent KLD map. Especially in the beginning of the PoPEx sampling this can be beneficial.

A very important feature of this algorithm is its *independence* of any physical parameterization. It only involves random variables that are characteristic indicator functions of categories, and therefore, the algorithm is independent of the model values and all the physical parameters they are associated with. As every new model is depending on all the previous ones, any approximation of integrals with respect to the posterior measure must take these correlations into account.

4.3 Posterior Event Prediction

Solving an inverse problem, not only serves to represent the posterior measure function, but also aims to compute the posterior measure of events $A \subset \mathcal{M}$ as in equation (4.1). These integrals can be approximated by the self-normalized estimator

$$\hat{\mu}_{\text{sn}} = \sum_{k=1}^{N} f(\mathbf{m}^k)\tilde{w}_k^N, \tag{4.10}$$

where $w_k = \frac{\sigma(\mathbf{m}^k)}{\phi_k(\mathbf{m}^k)}$ is the adaptive importance sampling ratio. The tilde notation was again used to indicate normalized weights such that

$$\tilde{w}_k^N = \frac{w_k}{\sum_{j=1}^{N} w_j}.$$

In this normalization process, a total number of N weights is taken into account. This is indicated by the upper index in $\tilde{w}_k^N$ and signifies that for any fixed upper bound $k_0 = 1 \ldots, N$ we have $\sum_{k=1}^{k_0} \tilde{w}_k^{k_0} = 1$. Due to the self-normalization in equation (4.10), it is not necessary to compute exact posterior measure values what means that the constant c in equation (2.7) can be ignored. Comparing two weights $\tilde{w}_j^N$ and $\tilde{w}_k^N$ must be understood as the relative comparison of the ratio values $\frac{\sigma(\mathbf{m}^j)}{\phi_j(\mathbf{m}^j)}$ and $\frac{\sigma(\mathbf{m}^k)}{\phi_k(\mathbf{m}^k)}$ regardless of the normalization bound N.

Formally, the main idea of the PoPEx sampling scheme is to consider a discrete random process $\mathbf{M}^1, \mathbf{M}^2, \ldots$ where each random vector $\mathbf{M}^{k+1}$ deduces from a conditional simulator and depends on the k previous realizations. The dependence is carried from $\{\mathbf{m}^1, \ldots, \mathbf{m}^k\}$ into $\mathbf{M}^{k+1}$ through a synthetic set of hard conditioning data $\mathbf{hd}^k$ such that $\mathbf{M}^{k+1}$ is implicitly defined by the density measure $\rho(\cdot \,|\, \mathbf{hd}^k)$. The joint sampling distribution $\phi_k(\mathbf{m}, \mathbf{hd})$ is therefore a composition of selecting $\mathbf{hd}$ and the conditional simulation tool. It can be written as

$$\phi_k(\mathbf{m}, \mathbf{hd}) = \rho(\mathbf{m} \,|\, \mathbf{hd})\tau_k(\mathbf{hd}). \tag{4.11}$$

The marginal density $\phi_k(\mathbf{m})$ follows immediately by summation over all possible conditioning sets $\mathbf{hd}$ such that

$$\phi_k(\mathbf{m}) = \sum_{\mathbf{hd}} \rho(\mathbf{m} \,|\, \mathbf{hd})\tau_k(\mathbf{hd}).$$

In this expression, the sum is finite because I and $\mathbf{v}$ are deduced from a finite number of possibilities. From the notation it is clear that τ_k fully characterizes the selection procedure of $\mathbf{hd}$ such that the marginal density reads $\phi_k(\mathbf{hd}) = \tau_k(\mathbf{hd})$. Similarly it can be deduced that the conditional density (in the sense of equation (2.4)) is given by $\phi_k(\mathbf{m} \,|\, \mathbf{hd}) = \rho(\mathbf{m} \,|\, \mathbf{hd})$. This second equality is important because it states that once the synthetic hard conditioning is imposed, the sampling process only depends on the conditional simulation tool.

The major strength of most Monte Carlo applications is that they produce random estimators $\hat{\mu}$ with expectation $\mathbb{E}(\cdot)$ and variance $\mathrm{Var}(\cdot)$ such that

$$\begin{aligned} &\text{(a)} \quad \mathbb{E}(\hat{\mu}) \;=\; \mu \\ &\text{(b)} \quad \mathrm{Var}(\hat{\mu}) \;\to\; 0 \quad \text{as} \quad N \to +\infty. \end{aligned} \tag{4.12}$$

These two properties signify that the estimator $\hat{\mu}$ is unbiased (a) and it's error goes to 0 when the number of samples grows large (b). More technically speaking, (a) and (b) together with Chebyshev's inequality [Durrett, 2010] are sufficient to conclude that $\hat{\mu}$ converges to μ in probability. Thus, if the number of samples is sufficiently large, it is reasonable to compute $\hat{\mu}$ as an approximation of μ. The main goal of the section 4.3.1 is to show that under some common regularity assumptions, the properties in (4.12) also hold for the self-normalized PoPEx estimator in equation (4.10) and therefore $\hat{\mu}_{\mathrm{sn}}$ converges to μ in probability. In section 4.3.2 we will provide a technique to compute the sampling weights w_k efficiently.

4.3.1 Convergence of the Estimator

For showing the convergence of the estimator $\hat{\mu}_{\mathrm{sn}}$, it is required that condition (4.3) holds in every iteration, i.e.

$$f(\mathbf{m})\sigma(\mathbf{m}) \neq 0 \quad \Longrightarrow \quad \phi_k(\mathbf{m}) > 0, \quad \text{for all } k > 0. \tag{4.13}$$

Furthermore, we assume that for all $k > 0$ "the proposal ϕ_k has heavier tails than ρ". More precisely, for a random vector $\mathbf{M}^k$ with density ϕ_k we assume that the adaptive importance sampling ratio $\frac{\rho(\mathbf{M}^k)}{\phi_k(\mathbf{M}^k)}$ has finite second moment (cf. equation 4.6). For showing the convergence of $\hat{\mu}_{\mathrm{sn}}$, it is sufficient to show that

$$\hat{\mu} = \frac{1}{N}\sum_{k=1}^{N} f(\mathbf{m}_k) w_k \;\to\; \mu.$$

The convergence of the self-normalized estimator $\hat{\mu}_{\mathrm{sn}}$ then follows with $f \equiv 1$ and from Slutky's theorem.

As mentioned above, τ_k incorporates two different probability densities. One is defined over the set of all possible subsets $I \subset \{1, \dots n\}$ and the second characterizes the extraction of $\mathbf{v}$ from $\mathbf{m}^1, \dots, \mathbf{m}^k$. The cardinality of the index set I, denoted by n_k, is selected from an uniform probability over $\{0, \dots n_{\max}\}$. This mainly assures two key properties. The number of conditions in **hd** is bounded from above and, more importantly, that it is always possible to sample $I = \emptyset$. If the latter is true, $\phi_k(\mathbf{m}) = 0$ implies that $\rho(\mathbf{m}) = 0$, what is sufficient for (4.13) to hold. For simplicity, $f(\mathbf{m})\sigma(\mathbf{m})$ is assumed to be non-zero for any $\mathbf{m} \in \mathcal{M}$. If this is not the case, the corresponding modifications can easily be made. For any $k > 0$, a random vector $\mathbf{X}^k$ is defined as

$$\mathbf{X}^k = f(\mathbf{M}^k)\frac{\sigma(\mathbf{M}^k)}{\phi_k(\mathbf{M}^k)}.$$

By the equation of change of variable [Durrett, 2010], the expected value of $\mathbf{X}^k$ is μ because

$$\mathbb{E}\big(\mathbf{X}^k\big) = \int_{\mathcal{E}^k} f(\mathbf{m}) \frac{\sigma(\mathbf{m})}{\phi_k(\mathbf{m})} \phi_k(\mathbf{m}) d\mathbf{m} = \mu,$$

where $\mathcal{E}^k = \{\mathbf{m} \in \mathcal{M} \ : \ \phi_k(\mathbf{m}) > 0\}$ is the support of the proposal density at iteration k. From the linearity of the operator $\mathbb{E}\big(\cdot\big)$ it is obtained that

$$\mathbb{E}\big(\hat{\mu}\big) = \frac{1}{N} \sum_{i=1}^{N} \mathbb{E}\big(\mathbf{X}^k\big) = \mu,$$

which satisfies (4.12a) and proves that $\hat{\mu}$ is unbiased. However, we also want the quality of the estimation to increase when the number of models grows. With $||\cdot||$ denoting the usual matrix norm, an upper bound for $\big\|\mathrm{Var}\big(\hat{\mu}\big)\big\|$ reads

$$\big\|\mathrm{Var}\big(\hat{\mu}\big)\big\| \leq \frac{1}{N^2} \sum_{j=1}^{N} \big\|\mathrm{Var}\big(\mathbf{X}^j\big)\big\| + \frac{2}{N^2} \sum_{k=2}^{N} \sum_{j=1}^{k-1} \big\|\mathrm{Cov}\big(\mathbf{X}^j, \mathbf{X}^k\big)\big\|, \tag{4.14}$$

in which $\mathrm{Cov}(\,\cdot\,,\,\cdot\,)$ stands for the covariance operator. Before continuing, let's consider another set random variables. For each integer pair $j < k$ we define

$$Y^{jk} : \Omega \to \{0, \ldots, n_{\max}\}$$

to count the number of conditioning values that have been extracted from $\mathbf{m}^j$ and imposed in $\mathbf{m}^k$. By the definition of conditional expectation the covariance of $\mathbf{X}^j$ and $\mathbf{X}^k$ can be rewritten as

$$\begin{aligned} \mathrm{Cov}\big(\mathbf{X}^j, \mathbf{X}^k\big) &= \mathbb{E}\big((\mathbf{X}^j - \mu)(\mathbf{X}^k - \mu)^{\mathrm{T}}\big) \\ &= \mathbb{E}\Big(\mathbb{E}\big((\mathbf{X}^j - \mu)(\mathbf{X}^k - \mu)^{\mathrm{T}} \mid Y^{jk}\big)\Big) \\ &= \sum_{s=0}^{n_{\max}} \mathbb{P}(Y^{jk} = s) \mathbb{E}\big((\mathbf{X}^j - \mu)(\mathbf{X}^k - \mu)^{\mathrm{T}} \mid Y^{jk} = s\big). \end{aligned} \tag{4.15}$$

We continue with two important assumptions that mainly rely on the function f and the involved conditional simulator:

Assumption 4.3.1.

1. *There is a constant $C_1 < +\infty$ such that $||\mathrm{Var}(\mathbf{X}^k)|| \leq C_1$ for all $k = 1, 2, \ldots$*
2. *There is a function $C_2 : \{0, 1, \ldots, n_{\max}\} \longrightarrow [0, +\infty)$ such that*

   (i) $C_2(0) = 0$
   (ii) $r \leq s \iff C_2(r) \leq C_2(s)$, and
   (iii) for all $j < k$ and all $s \in \{0, \ldots, n_{\max}\}$

$$\big\|\mathbb{E}\big((\mathbf{X}^j - \mu)(\mathbf{X}^k - \mu)^{\mathrm{T}} \mid Y^{jk} = s\big)\big\| \ \leq \ C_2(s).$$

The first assumption is very common and prevents the marginal variability to become increasingly large. This is sufficient for the first part in equation (4.14) to vanish as $N \to +\infty$. The three conditions in the second assumption will be used for bounding the norm of the covariance terms. In particular, they assure that there is a non-decreasing function C_2 that passes through the origin and controls the conditional covariance of $\mathbf{X}^j$ and $\mathbf{X}^k$ when Y^{jk} is known. In other words, if we know that the model $\mathbf{m}^j$ delivered s conditioning values for the construction of $\mathbf{m}^k$, the covariance of $\mathbf{X}^j$ and $\mathbf{X}^k$ is bounded by $C_2(s)$.

So far, only the selection procedure of I has been considered. The second distribution within τ_k characterizes the extraction of a vector $\mathbf{v}$ from the previously generated realizations $\mathbf{m}^1, \dots, \mathbf{m}^k$. This process uses a probability measure $\widetilde{\Sigma}^k = \{\tilde{\sigma}_1^k, \dots, \tilde{\sigma}_k^k\}$ that favors realizations with high posterior values. For each index $i \in I$ the distribution $\widetilde{\Sigma}^k$ is used to randomly select a realization $\mathbf{m}^j$ and copy-paste the model value m_i^j into $\mathbf{v}$. If $\{\tilde{\sigma}_1^k, \dots, \tilde{\sigma}_k^k\}$ is chosen properly, this procedure generates a conditioning vector $\mathbf{v}$ by mainly learning from 'good' models. Regarding the equation (4.15), we want to quantify the probability of the sets $\{Y^{jk} = s\}$. Note that for Y^{jk} with $j < k$ we consider the PoPEx iteration $k-1$ in which $\mathbf{m}^k$ is simulated. At that point, the algorithm generates a normalized set of weights $\widetilde{\Sigma}^{k-1} = \{\tilde{\sigma}_1^{k-1}, \dots, \tilde{\sigma}_{k-1}^{k-1}\}$ such that $\sum_{j=1}^{k-1} \tilde{\sigma}_j^{k-1} = 1$. The computation of $\mathbb{P}(Y^{jk} = s)$ can be split into the probability of selecting the number of conditioning $n_k \geq s$ and putting exactly s values from $\mathbf{m}^j$ into $\mathbf{m}^k$. The second of which follows a binomial distribution $B(n_k, \tilde{\sigma}_j^{k-1})$ with density $\binom{n_k}{s}(\tilde{\sigma}_j^{k-1})^s(1-\tilde{\sigma}_j^{k-1})^{n_k-s}$ such that

$$\mathbb{P}(Y^{jk} = s) = \sum_{r=s}^{n_{\max}} \mathbb{P}(n_k = r)\binom{r}{s}(\tilde{\sigma}_j^{k-1})^s(1-\tilde{\sigma}_j^{k-1})^{r-s},$$

where $\binom{n}{k} := \frac{n!}{k!(n-k)!}$ denotes the binomial coefficient. Substituting this equation into (4.15) and using $\mathbb{P}(n_k = r) \leq 1$ together with $(2iii)$ of assumption 4.3.1, we obtain

$$\left\|\mathrm{Cov}\left(\mathbf{X}^j, \mathbf{X}^k\right)\right\| \leq \sum_{s=0}^{n_{\max}} \sum_{r=s}^{n_{\max}} \binom{r}{s}(\tilde{\sigma}_j^{k-1})^s(1-\tilde{\sigma}_j^{k-1})^{r-s} C_2(s).$$

Because C_2 vanishes at 0, all terms with $s = 0$ may be dropped. Furthermore, after reordering the sum and using $(2ii)$, the inequality becomes

$$\begin{aligned}\left\|\mathrm{Cov}\left(\mathbf{X}^j, \mathbf{X}^k\right)\right\| &\leq \sum_{r=1}^{n_{\max}} \sum_{s=1}^{r} \binom{r}{s}(\tilde{\sigma}_j^{k-1})^s(1-\tilde{\sigma}_j^{k-1})^{r-s} C_2(s) \\ &\leq C_2(n_{\max}) \sum_{r=1}^{n_{\max}} \sum_{s=1}^{r} \binom{r}{s}(\tilde{\sigma}_j^{k-1})^s(1-\tilde{\sigma}_j^{k-1})^{r-s}.\end{aligned}$$

Using $\tilde{\sigma}_j^{k-1} \in [0,1]$ together with the totality of the binomial sum, it can be

observed that

$$\begin{aligned}\sum_{s=1}^{r}\binom{r}{s}(\tilde{\sigma}_j^{k-1})^s(1-\tilde{\sigma}_j^{k-1})^{r-s} &= 1-(1-\tilde{\sigma}_j^{k-1})^r \\ &\leq 1-(1-\tilde{\sigma}_j^{k-1})^{n_{\max}} \\ &\leq n_{\max}\tilde{\sigma}_j^{k-1}.\end{aligned}$$

The last inequality can be shown by recurrence over n such that $1-(1-x)^n \leq nx$ for all $x \in [0,1]$ and $n \geq 1$. Almost at the end of the algebraic work, we may limit the covariance between $\mathbf{X}^j$ and $\mathbf{X}^k$ by

$$\begin{aligned}\left\|\mathrm{Cov}\left(\mathbf{X}^j,\mathbf{X}^k\right)\right\| &\leq C_2(n_{\max})\sum_{r=1}^{n_{\max}} n_{\max}\tilde{\sigma}_j^{k-1} \\ &= C_2(n_{\max})n_{\max}^2\tilde{\sigma}_j^{k-1}.\end{aligned}$$

It is now straightforward to conclude the property (b) of equation (4.12) by substituting the last expression into (4.14) such that

$$\begin{aligned}\left\|\mathrm{Var}(\hat{\mu})\right\| &\leq \frac{1}{N^2}\sum_{j=1}^{N}\left\|\mathrm{Var}\left(\mathbf{X}^j\right)\right\| + \frac{2}{N^2}\sum_{k=2}^{N}\sum_{j=1}^{k-1}\left\|\mathrm{Cov}\left(\mathbf{X}^j,\mathbf{X}^k\right)\right\| \\ &\leq \frac{C_1}{N} + \frac{2C_2(n_{\max})n_{\max}^2}{N^2}\sum_{k=2}^{N}\underbrace{\sum_{j=1}^{k-1}\tilde{\sigma}_j^{k-1}}_{=1} \\ &\leq \frac{C_1+2C_2(n_{\max})n_{\max}^2}{N} \quad \longrightarrow\ 0, \quad \text{as } N \longrightarrow +\infty.\end{aligned} \tag{4.16}$$

This ends the proof and we conclude that the PoPEx estimators $\hat{\mu}$ and $\hat{\mu}_{\mathrm{sn}}$ both converge to μ in probability.

It was mentioned that the adaptive importance sampling weights $\frac{\sigma(\mathbf{m}^k)}{\phi_k(\mathbf{m}^k)}$ can be computed in a convenient and natural way such that the normalization constants of ρ and ϕ_k disappear. The next section explains how.

4.3.2 Computation of the Weights

Let us recall that the solution of the inverse problem is quantified by the posterior density function in equation (2.7) and reads

$$\sigma(\mathbf{m}) = cL(\mathbf{m})\rho(\mathbf{m}).$$

It is also important to keep in mind that we consider self-normalized estimators as in equation (4.10) so that the normalization constant c can still be ignored. From the above equation we note that the computation of the weights $w_k =$

$\frac{\sigma(\mathbf{m}^k)}{\phi_k(\mathbf{m}^k)}$ can be simplified by using the factorization of σ. Each likelihood value $L(\mathbf{m}^k)$ is evaluated during the PoPEx procedure, so that it remains to consider the ratios

$$\frac{\rho(\mathbf{m}^k)}{\phi_k(\mathbf{m}^k)}, \qquad \text{for } \; k = 1, \dots, N. \tag{4.17}$$

Recall that the conditional simulation tool randomly generates realizations from the prior measure ρ. This means that the above ratios compare the probability measures of generating a model $\mathbf{m}$ outside ($\rho(\mathbf{m})$) and inside ($\phi_k(\mathbf{m})$) a PoPEx chain or similarly, with and without the definition of an inverse problem. The PoPEx random process iteratively picks an index-value pair $\mathbf{hd} = (I, \mathbf{v})$ and generates the next realization. We will show that the ratios in equation (4.17) only depend on the selection process of $\mathbf{hd}$. Let us distinguish two events that henceforth will be noted similarly:

1. The conditional sampler is assumed to follow the prior probability measure ρ. Recall that such a tool iteratively informs model parameters with simulation values from a training image. A set $\mathbf{hd} = (I, \mathbf{v})$ appearing within ρ will refer to the event where "the first n_k indices (met during the simulation) were $\{i \in I\}$ and they obtained the values $\{v_1, \dots, v_{n_k}\}$." Following this line, $\rho(\mathbf{m} \,|\, \mathbf{hd})$ expresses the conditional measure to produce $\mathbf{m}$, when the first n_k assignments imposed the values $\{v_1, \dots, v_{n_k}\}$ at the locations $\{i \in I\}$. Note that within the PoPEx algorithm, the selection of n_k is randomized. Within the prior measure ρ however, n_k is given by the set $\mathbf{hd}$ and therefore considered to be fixed.

2. The sampling distribution on the other hand takes the PoPEx iterations into account. The set of constraints $\mathbf{hd}$ appears within ϕ_k for considering the event where "during the k-th PoPEx iteration $\mathbf{hd}$ has been produced and used as hard conditioning data." Accordingly, $\phi_k(\mathbf{m} \,|\, \mathbf{hd})$ measures the probability of sampling $\mathbf{m}$ at iteration k, knowing that $\mathbf{hd}$ has been imposed.

By definition, the density measure $\phi_k(\cdot \,|\, \mathbf{hd})$ is zero on the subset $\{\mathbf{m} \in \mathcal{M} : \mathbf{m}_I \neq \mathbf{v}\}$ because the hard conditioning data is fully honored by the conditional simulator. Without loss of generality, we therefore only consider coinciding pairs $\mathbf{m}$ and $\mathbf{v}$ (i.e. where the values on the n_k locations coincide) with non-zero measures $\phi_k(\mathbf{m})$ and $\phi_k(\mathbf{m} \,|\, \mathbf{hd})$. Let's recall that the simulation process behind $\rho(\cdot \,|\, \mathbf{hd})$ and $\phi_k(\cdot \,|\, \mathbf{hd})$ is the same, so that the two conditional measure values are equal and

$$\frac{\rho(\mathbf{m})}{\phi_k(\mathbf{m})} = \frac{\rho(\mathbf{m})}{\rho(\mathbf{m} \,|\, \mathbf{hd})} \frac{\phi_k(\mathbf{m} \,|\, \mathbf{hd})}{\phi_k(\mathbf{m})}.$$

Thanks to these ratio expressions, any (unknown) normalization constant of ρ and ϕ_k naturally disappear because they involve similar conditional simulators.

Using the properties of conditional densities, we obtain

$$\frac{\rho(\mathbf{m})}{\phi_k(\mathbf{m})} = \frac{\rho(\mathbf{hd})}{\rho(\mathbf{hd}\,|\,\mathbf{m})}\frac{\phi_k(\mathbf{hd}\,|\,\mathbf{m})}{\phi_k(\mathbf{hd})}. \tag{4.18}$$

What is convenient in the latter equation, is that the involved quantities can be expressed from standard techniques of combinatorial probability theory. As mentioned in chapter 3, conditional simulations build samples by treating the parameter indices sequentially in a random order. Therefore, $\rho(\mathbf{hd}^k|\mathbf{m}^k)$ measures the probability of informing the first n_k pixels according to $\mathbf{hd}^k$, when the sampled model is known. But knowing $\mathbf{m}^k$ implies that the conditioning values in $\mathbf{hd}^k$ are given, so that we only need to compute the probability to meet the n_k conditioning locations (in any order) in the very beginning of the conditional simulation. If there are n model parameters, the simulation of the n_k first locations is independent of the $n-n_k$ remaining model parameters and $\rho(\mathbf{hd}^k|\mathbf{m}^k)$ reads

$$\rho(\mathbf{hd}^k|\mathbf{m}^k) = \frac{n_k!(n-n_k)!}{n!}.$$

Recall that although n_k is randomized withing the PoPEx algorithm, inside the prior measure ρ it is considered to be fixed and given by $\mathbf{hd}$. Let's write the hard data such that $\mathbf{hd}^k = \{hd_1^k, \dots, hd_{n_k}^k\}$ where hd_i^k is the i-th index-value pair of the hard conditioning. If the model $\mathbf{m}^k$ is unknown, the probabilities of the values in $\mathbf{v}$ must be taken into account so that

$$\rho(\mathbf{hd}^k) = \frac{n_k!(n-n_k)!}{n!}\frac{1}{n_k!}\sum_{\varsigma\in S_{n_k}}\prod_{i=1}^{n_k}\rho\big(hd_{\varsigma(i)}^k \,\big|\, hd_{\varsigma(1)}^k, \dots, hd_{\varsigma(i-1)}^k, \varsigma\big)$$

and

$$\frac{\rho(\mathbf{hd}^k)}{\rho(\mathbf{hd}^k\,|\,\mathbf{m}^k)} = \frac{1}{n_k!}\sum_{\varsigma\in S_{n_k}}\prod_{i=1}^{n_k}\rho\big(hd_{\varsigma(i)}^k \,\big|\, hd_{\varsigma(1)}^k, \dots, hd_{\varsigma(i-1)}^k, \varsigma\big). \tag{4.19}$$

In this notation, $\rho\big(hd_{\varsigma(i)}^k \,\big|\, hd_{\varsigma(1)}^k, \dots, hd_{\varsigma(i-1)}^k, \varsigma\big)$ is the prior probability measure of encountering the hard conditioning pair $hd_{\varsigma(i)}^k$ when knowing the $i-1$ previous ones $hd_{\varsigma(1)}^k, \dots, hd_{\varsigma(i-1)}^k$ as well as the first part of the random simulation path $\varsigma \in S_{n_k}$.

Every PoPEx iteration selects a number n_k, *independently* samples conditioning locations from $\widetilde{D}(P^k||Q)$ and extracts values according to $\widetilde{\Sigma}^k$. Therefore, the measure value $\tau_k(\mathbf{hd}^k)$ can be understood as

$$\tau_k(\mathbf{hd}^k) = \tau_k(\mathbf{v}\,|\,I, n_k)\tau_k(I\,|\,n_k)\tau_k(n_k).$$

From the definition of the proposal in equation (4.11) we have $\phi_k(\mathbf{hd}^k) = \tau_k(\mathbf{hd}^k)$. Hence, the conditional density for a known model $\mathbf{m}^k$ then becomes

$\phi_k(\mathbf{hd}^k \,|\, \mathbf{m}^k) = \tau_k(I \,|\, n_k)\tau_k(n_k)$ and it follows that

$$\frac{\phi_k(\mathbf{hd}^k)}{\phi_k(\mathbf{hd}^k \,|\, \mathbf{m}^k)} = \tau_k(\mathbf{v} \,|\, I, n_k).$$

This quantity measures the probability of selecting conditioning values $\mathbf{v}$ when the set of conditioning indices is known. Using similar arguments as above but noticing that within a PoPEx chain, the conditioning values are selected *independently*, we have

$$\tau_k(\mathbf{v} \,|\, I, n_k) = \prod_{i=1}^{n_k} \tau_k\big(hd_i^k\big),$$

and the adaptive importance sampling ratio such that

$$\frac{\rho(\mathbf{m}^k)}{\phi_k(\mathbf{m}^k)} = \frac{1}{n_k!} \sum_{\varsigma \in S_{n_k}} \prod_{i=1}^{n_k} \frac{\rho\big(hd_{\varsigma(i)}^k \,\big|\, hd_{\varsigma(1)}^k, \ldots, hd_{\varsigma(i-1)}^k, \varsigma\big)}{\tau_k\big(hd_{\varsigma(i)}^k\big)}. \tag{4.20}$$

Although from a practical point of view, this expression is not very handy, it can already be seen that the ratio in equation (4.17) only depends on the conditioning data $\mathbf{hd}^k$. In many situations it is reasonable to assume that all the conditioning binomials in $\mathbf{hd}^k$ are independent of each other. When working with the simulator in 3.2.2 this assumption is acceptable if the conditioning locations are well separated. It can therefore be perceptive to choose the length of $\mathbf{hd}^k$ to be adequate with respect to the grid size. In the case of independent hard conditioning data, the ratio reads

$$\frac{\rho(\mathbf{m}^k)}{\phi_k(\mathbf{m}^k)} = \prod_{i=1}^{n_k} \frac{\rho\big(hd_i^k\big)}{\tau_k\big(hd_i^k\big)}.$$

In chapter 3, the simulation values are categorized by a set $\{f_1, \ldots, f_s\}$. For a given $\mathbf{hd}^k = (I, \mathbf{v})$, an index-to-index map $r = r(i)$ is defined such that $f_{r(i)}$ identifies the category of the value v_i. An approximation to $\rho\big(hd_i^k\big)$ is obtained from the categorical prior probabilities in Q by $\rho\big(hd_i^k\big) \approx q_{r(i)}(i)$. This simply suggests to find the map $q_{r(i)}$ that corresponds to the category of v_i and extract the probability value at the corresponding parameter index. A similar approximation is issued for $\tau_k(hd_i^k)$. The categorical distribution P^k relies on the same weights as the procedure that selects conditioning values $\mathbf{v}$ (i.e. the weights in $\widetilde{\Sigma}^k$), and therefore it is reasonable to accept $\tau_k(hd_i^k) \approx p_{r(i)}^k(i)$. It is worthwhile to note that when working with discrete models, where the categories $\{f_1, \ldots, f_s\}$ define facies values, these approximations are exact. Finally, a computable ratio is provided by

$$\frac{\rho(\mathbf{m}^k)}{\phi_k(\mathbf{m}^k)} = \prod_{i \in I} \frac{q_{r(i)}(i)}{p_{r(i)}^k(i)}. \tag{4.21}$$

This expression is very practical. All the quantities in equation (4.21) are assembled during the PoPEx algorithm, so that the required effort for evaluating the ratio is negligible. Moreover, the expression is easily translated into log-probabilities, what can simplify the floating-point representation of the values. Although it only represents an approximation of the true ratio, often the assumptions are not too strongly violated, and the usage of the above equation is feasible. Finally, the weights w_k are computed by correcting the likelihood measure according to the hard conditioning data:

$$w_k \;=\; L(\mathbf{m}^k)\frac{\rho(\mathbf{m}^k)}{\phi_k(\mathbf{m}^k)} \;=\; L(\mathbf{m}^k)\prod_{i\in I}\frac{q_{r(i)}(i)}{p^k_{r(i)}(i)}. \tag{4.22}$$

When working with self-normalized estimators, the normalization constant c can be ignored.

4.3.3 Degeneracy of the Weights

A common issue in the importance sampling framework is that the distribution of the weights in $W^N = \{w_1, \ldots, w_N\}$ may become increasingly skewed when the dimension of $\mathcal{M}$ is large (e.g. Remark 5.7.1 of Rubinstein and Kroese [2016]). As the weights w_k are non-negative and have expectation equal to one, this means that they may take small value with high probability but occasionally become very large. Using such weights in equation (4.10) produces estimators $\hat{\mu}_{\text{sn}}$ that are dominated by very few samples. Several preventive techniques exist and they often try to consider a reduced dimensionality in the computation of the weights [Doucet et al., 2001]. The expression (4.21) reduces the computation of the ratio to the hard conditioning data but represents only one part of the weights in equation (4.22), so that the degeneracy problem may still exists. A statistical evidence of this effect is obtained from a measure called **Kish's effective sample size** [Kish, 1965], denoted by $n_e\left(W^N\right)$ and reading

$$n_e\left(W^N\right) = \frac{\left(\sum_{k=1}^N w_k\right)^2}{\sum_{k=1}^N w_k^2} = \frac{\left\|W^N\right\|_1^2}{\left\|W^N\right\|_2^2}. \tag{4.23}$$

Here, $\left\|W^N\right\|_1 = \sum_{k=1}^N w_k$ and $\left\|W^N\right\|_2 = \left(\sum_{k=1}^N w_k^2\right)^{1/2}$ are the usual l_1 and l_2 norms, respectively. It is clear that there must be at least one strictly positive weight for n_e to make sense. Henceforth, this requirement will be assumed to be true. For the continuation it is important to note that n_e is independent of any normalization constant because $n_e\left(W^N\right) = n_e\left(cW^N\right)$ for any constant c.

There is an obvious link between n_e and the variance of W^N. It is sufficient to notice, that an estimator of the variance is obtained by $\frac{1}{N}\left\|W^N\right\|_2^2 - \frac{1}{N^2}\left\|W^N\right\|_1^2$. Strongly varying weights give values such that $\frac{1}{N}\left\|W^N\right\|_2^2 >> \frac{1}{N^2}\left\|W^N\right\|_1^2$ and therefore, $n_e << N$. In general, lowering the variance in W^N

increases the effective number $n_e(W^N)$. However, it is often hard to specify a bound under which n_e is alarmingly small, because this strongly depends on the application. The ordering of the weights in equation (4.23) is not important. Without loss of generality, we may assume that there are two integers $k_1 \geq 0, k_2 > 0$ with $k_1 \leq N - k_2$ and such that

$$\begin{aligned} 0 = w_1 = \cdots = w_{k_1} \\ < w_{k_1+1} \leq \cdots \leq w_{N-k_2} \\ < w_{N-k_2+1} = \cdots = w_N. \end{aligned} \tag{4.24}$$

While $(N - k_1)$ counts the number of non-vanishing weights, k_2 indicates how many times the maximum value $w_N > 0$ is reached.

If the distribution of the weights in W^N is highly skewed, its effective number $n_e\left(W^N\right)$ is very small. An idea to manipulate n_e is to chose $\alpha > 0$ and form the **power set** W_α^N as

$$W_\alpha^N = \left\{(w_1)^\alpha, \ldots, (w_N)^\alpha\right\}. \tag{4.25}$$

This manipulation is quite intuitive. Let's consider a random set of values $V^N = \left\{v_1, \ldots, v_N\right\}$ with $0 \leq v_i \leq 1$ for all $i = 1, \ldots, N$. For $\alpha = 2$ and by passing from V^N to V_α^N, every value v_i is multiplied by itself and therefore reduced by a factor of v_i. Proportionally speaking, this means that small values are decreased more heavily than large ones and the distribution in V_α^N is becoming more dispersed. The same reasoning applies for any integer $\alpha = 3, 4, 5, \ldots$. If on the other hand $\alpha = \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \ldots$, we may argue in the exact opposite direction so that the distribution in V_α^N becomes less dispersed. But the power function v^α is continuous in v and in α so that the above arguments provide an intuition for any $v > 0$ and $\alpha > 0$.

With a power set W_α^N, any approximation $\hat{\mu}$ and $\hat{\mu}_{\text{sn}}$ is then computed by using the weights in W_α^N rather than W^N. The manipulated estimators are henceforth denoted by $\hat{\mu}_\alpha$ and $\hat{\mu}_{\alpha,\text{sn}}$ whilst the latter reads

$$\hat{\mu}_{\alpha,\text{sn}} = \frac{\sum_{k=1}^N f(\mathbf{m}^k) w_k^\alpha}{\sum_{j=1}^N w_j^\alpha}.$$

Similarly to the weights in W^N, the effective sample size of W_α^N can be computed by

$$n_e\left(W_\alpha^N\right) = \frac{\left(\sum_{k=1}^N w_k^\alpha\right)^2}{\sum_{k=1}^N w_k^{2\alpha}}.$$

If α is chosen properly, this signifies that the computation of $\hat{\mu}_{\alpha,\text{sn}}$ is effectively based on a larger number of samples than $\hat{\mu}_{\text{sn}}$, what can increase its statistical significance. Furthermore, for a fixed $\alpha > 0$ the power function x^α is strictly increasing on $[0, +\infty)$ and therefore, the ordering of the weights does not change when passing from W^N to W_α^N. This means that the largest contribution

in $\hat{\mu}_{\text{sn}}$ comes from the same model that also provides the most significant contribution in $\hat{\mu}_{\alpha,\text{sn}}$. On the downside however, we note that $\hat{\mu}_{\alpha,\text{sn}}$ might be biased. The main goal of this section is to state lemma 4.3.3, in which some conditions are derived for $\hat{\mu}_{\alpha,\text{sn}}$ to asymptotically drop its bias and converge to μ in probability. It is clear that the convergence of the self-normalized estimator $\hat{\mu}_{\alpha,\text{sn}}$ directly follows from the convergence of $\hat{\mu}_\alpha$. But first, let us show some properties of $n_e\left(W_\alpha^N\right)$.

Proposition 4.3.2. *If W_α^N, $k_1 \geq 0$, and $k_2 > 0$ are defined as above, then*

(i) $k_2 \leq n_e\left(W_\alpha^N\right) \leq N - k_1$, $\forall \alpha > 0$

(ii) *if* $k_2 < N - k_1$ *then* $\alpha < \beta \Leftrightarrow n_e\left(W_\beta^N\right) < n_e\left(W_\alpha^N\right)$

(iii) $n_e\left(W_\alpha^N\right) \uparrow N - k_1$, *as* $\alpha \to 0$

(iv) $n_e\left(W_\alpha^N\right) \downarrow k_2$, *as* $\alpha \to \infty$

Proof. (i) From the Cauchy-Schwarz inequality we may derive the relation $\left\|W_\alpha^N\right\|_1^2 \leq (N-k_1)\left\|W_\alpha^N\right\|_2^2$ and thus, $n_e\left(W_\alpha^N\right) \leq N - k_1$. For proving the second inequality of (i), let us simplify the following notations. We assume that the index in each of the sums goes from k_1+1 up to $N-k_2$; the expression $\sum_k$ therefore signifies $\sum_{k=k_1+1}^{N-k_2}$ and the result is obtained by

$$\begin{aligned} n_e\left(W_\alpha^N\right) &= \frac{\left(k_2 w_N^\alpha + \sum_k w_k^\alpha\right)^2}{k_2 w_N^{2\alpha} + \sum_k w_k^{2\alpha}} \\ &= \frac{k_2^2 w_N^{2\alpha} + 2k_2 w_N^\alpha \sum_k w_k^\alpha + \left(\sum_k w_k^\alpha\right)^2}{k_2 w_N^{2\alpha} + \sum_k w_k^{2\alpha}} \\ &\geq \frac{k_2^2 w_N^{2\alpha} + k_2 w_N^\alpha \sum_k w_k^\alpha}{k_2 w_N^{2\alpha} + \sum_k w_k^{2\alpha}} \\ &\geq \frac{k_2^2 w_N^{2\alpha} + k_2 w_N^\alpha \sum_k w_k^\alpha}{k_2 w_N^{2\alpha} + w_N^\alpha \sum_k w_k^\alpha} = k_2. \end{aligned}$$

(ii) Again, the readability of the following equations is improved by simplifying the notation of the sums. For the proofs of points (ii) and (iv) however, we assume the sum index to go from k_1+1 up to N and write $\sum_k$ for $\sum_{k=k_1+1}^N$, $\sum_{j,k}$ for $\sum_{j,k=k_1+1}^N$, $\sum_{j<k}$ for $\sum_{k=k_1+2}^N \sum_{j=k_1+1}^{k-1}$, etc. By the assumptions $k_2 < N - k_1$ and $k_2 > 0$ there are at least two non-zero weights w_{k_1+1} and w_{N-k_2+1} such that $0 < w_{k_1+1} < w_{N-k_2+1}$. Furthermore, the function $n_e\left(W_\alpha^N\right)$ is continuously differentiable with respect to α in the entire open set $(0,\infty)$ and therefore, it is sufficient to show that the derivative is strictly negative for any $\alpha \in (0,\infty)$. Using

the quotient rule together with

$$\frac{d}{d\alpha}\left\|W_\alpha^N\right\|_1^2 = 2\left\|W_\alpha^N\right\|_1\left(\sum_k \log(w_k)w_k^\alpha\right),$$
$$\frac{d}{d\alpha}\left\|W_\alpha^N\right\|_2^2 = \sum_k 2\log(w_k)w_k^{2\alpha},$$

the derivative becomes

$$\frac{d}{d\alpha}n_e\left(W_\alpha^N\right) = 2\frac{||W_\alpha^N||_1}{||W_\alpha^N||_2^4}\underbrace{\left[\sum_{j,k}\log(w_j)w_j^\alpha w_k^\alpha\left(w_k^\alpha - w_j^\alpha\right)\right]}_{(S)}.$$

Let us show that (S) is strictly negative. By simply reordering the sums, we obtain

$$\begin{aligned}(S) &= \sum_{j<k}\log(w_j)w_j^\alpha w_k^\alpha\left(w_k^\alpha - w_j^\alpha\right) + \sum_{k<j}\log(w_j)w_j^\alpha w_k^\alpha\left(w_k^\alpha - w_j^\alpha\right)\\ &= \sum_{j<k}\Big[\log(w_j)w_j^\alpha w_k^\alpha\left(w_k^\alpha - w_j^\alpha\right) + \log(w_k)w_k^\alpha w_j^\alpha\left(w_j^\alpha - w_k^\alpha\right)\Big]\\ &= \sum_{j<k}w_j^\alpha w_k^\alpha\left(w_k^\alpha - w_j^\alpha\right)\log\left(\frac{w_j}{w_k}\right).\end{aligned}$$

Whenever an index j is strictly smaller than k, the weight w_j^α must be smaller or equal to w_k^α and therefore, $w_j^\alpha w_k^\alpha\left(w_k^\alpha - w_j^\alpha\right) \geq 0$ and $\log\left(w_j/w_k\right) \leq 0$. For $j = k_1 + 1$ and $k = N - k_2 + 1$ these inequalities are strict. It follows that any summand above is non-positive and

$$(S) \leq w_{k_1+1}^\alpha w_{N-k_2+1}^\alpha \underbrace{\left(w_{N-k_2+1}^\alpha - w_{k_1+1}^\alpha\right)}_{>0}\underbrace{\log\left(\frac{w_{k_1+1}}{w_{N-k_2+1}}\right)}_{<0} < 0.$$

What ends the proof of (ii).

(iii) For all $k = k_1 + 1, \ldots, N$ we have $w_k > 0$ so that

$$\lim_{\alpha\to 0} w_k^\alpha = \lim_{\alpha\to 0} w_k^{2\alpha} = 1$$

and therefore,

$$n_e\left(W_\alpha^N\right) \;\to\; \frac{(N-k_1)^2}{N-k_1} \;=\; N - k_1, \qquad \text{as } \alpha \to 0.$$

The monotonicity comes from (ii).

(*iv*) The effective number of weights in W_α^N can be rewritten such as

$$n_e\left(W_\alpha^N\right) \;=\; \frac{1}{\sum_k \left(\frac{w_k^\alpha}{\sum_j w_j^\alpha}\right)^2},$$

with the sum index going from k_1+1 up to N. It is possible to bound the terms $w_k^\alpha / \sum_j w_j^\alpha$ from above and below by

$$0 \;\leq\; \frac{w_k^\alpha}{\sum_j w_j^\alpha} \;\leq\; \frac{1}{k_2}\left(\frac{w_k}{w_N}\right)^\alpha.$$

Therefore, $\lim_{\alpha\to\infty} w_k^\alpha / \sum_j w_j^\alpha = 0$, for any w_k that is strictly smaller than w_N. In the case where $w_k = w_N$, the same equation can be used for deducing

$$\lim_{\alpha\to\infty} \frac{w_k^\alpha}{\sum_j w_j^\alpha} \;\leq\; \frac{1}{k_2}.$$

But for $w_k = w_N$ we obtain

$$\begin{aligned}\frac{w_k^\alpha}{\sum_j w_j^\alpha} \;&=\; 1 - \frac{\sum_{r\neq k} w_r^\alpha}{\sum_j w_j^\alpha} \;=\; 1 - \frac{(k_2-1)w_N^\alpha}{\sum_j w_j^\alpha} \;-\; \sum_{r\leq N-k_2} \frac{w_r^\alpha}{\sum_j w_j^\alpha} \\ &\geq\; 1 - \frac{k_2-1}{k_2} \;-\; \sum_{r\leq N-k_2} \frac{w_r^\alpha}{\sum_j w_j^\alpha},\end{aligned}$$

and

$$\lim_{\alpha\to\infty} \frac{w_k^\alpha}{\sum_j w_j^\alpha} \;\geq\; 1 - \frac{k_2-1}{k_2} \;=\; \frac{1}{k_2}.$$

Finally, the limits are

$$\lim_{\alpha\to\infty} \frac{w_k^\alpha}{\sum_j w_j^\alpha} = \begin{cases} 0 \text{ if } w_k < w_N \\ 1/k_2 \text{ if } w_k = w_N, \end{cases}$$

what ends the proof, because

$$\lim_{\alpha\to\infty} n_e\left(W_\alpha^N\right) = \frac{1}{k_2\left(\frac{1}{k_2}\right)^2} = k_2,$$

while the monotonicity is again given by (*ii*).

□

Together with the properties 4.3.2, it is possible to prove the asymptotic convergence of the PoPEx estimator $\hat{\mu}_{\alpha,\mathrm{sn}}$. This important result is stated in lemma 4.3.3 and motivates the usage of the PoPEx prediction scheme in algorithm 4.

4.3.4 Convergence of the PoPEx Estimator

It was mentioned earlier, that the choice of $\alpha > 0$ is important. But selecting an appropriate power set W_α^N can be very challenging and may depend on the application. A natural choice is to fix a lower bound l_0 and choose α such that $n_e(W_\alpha^N) \geq l_0$. In this case, the computation of $\hat{\mu}_\alpha$ is effectively based on at least l_0 models. One straightforward idea is therefore to choose α such that

$$n_e(W_\alpha^N) = \max\left\{l_0, n_e(W^N)\right\}, \tag{4.26}$$

what signifies that $\alpha = 1$ for $n_e(W^N) \geq l_0$ and $\alpha < 1$ (s.t. $n_e(W_\alpha^N) = l_0$) otherwise. However, there are many other possibilities to select good values for α. In section 4.3.1 it is shown that the estimator $\hat{\mu}$ is unbiased and converges to μ. It is also clear that $\hat{\mu}_\alpha$ is equal to $\hat{\mu}$ for $\alpha = 1$. Therefore, we suggest that any selection scheme replacing equation (4.26) fulfills at least the following asymptotic behaviour: the value of α depends on the number $n_e(W^N)$ such that

$$n_e(W^N) \to +\infty \quad \Longrightarrow \quad \alpha \to 1. \tag{4.27}$$

In words, this means that the modification between W^N and W_α^N vanishes with the effective number of weights in W^N becoming increasingly large. Equation (4.26) satisfies this requirement, because $\alpha = 1$ whenever $n_e(W^N) \geq l_0$. Before moving to lemma 4.3.3 let us define $L_k \in \mathbb{R} \cup \{+\infty\}$ to be an upper bound for w_k such that

$$L_k = \inf\left\{\delta > 0 \ : \ \mathbb{P}\left(\frac{\sigma(\mathbf{M}^k)}{\phi_k(\mathbf{M}^k)} < \delta\right) = 1\right\}, \quad \text{for any } \ k > 0.$$

There are two important requirements for the constants L_k. First of all, it must be assured that they can not grow to infinity. But it was already mentioned that σ needs to have sharper tails than any proposal ϕ_k. This is sufficient to bound the weights w_k and the constants L_k from above. On the other hand, it is also necessary that the algorithm regularly generates significant weights so that $n_e(W^N)$ may grow to $+\infty$. Therefore, there must be a sub-sequence $(L_{k_j})_{j>0}$ of $(L_k)_{k>0}$ that stays away from 0. These requirements are stated more clearly in the following lemma.

Lemma 4.3.3. *Let $c, C \in \mathbb{R}$ be two constants with $0 < c < C$ such that*

(i) $L_k < C$ for any $k > 0$, and

(ii) there is a sub-sequence $(L_{k_j})_{j>0} \subset (L_k)_{k>0}$ such that $L_{k_j} > c$ for any $j > 0$.

If α is chosen as in equation (4.27), then

$$\hat{\mu}_{\alpha,\mathrm{sn}} \ \to \ \mu$$

in probability.

Proof. As mentioned above, it is sufficient to show the convergence of the estimator $\hat{\mu}_\alpha$ from which the convergence of $\hat{\mu}_{\alpha,\mathrm{sn}}$ then follows with $f \equiv 1$ and from Slutky's theorem. But if the condition in equation (4.27) is satisfied, it is sufficient to show that $n_e(W^N) \to +\infty$ in probability. From the set of weights $W^N = \{w_1, \ldots, w_N\}$ we form a new set $V^N = \{v_1, \ldots, v_N\}$ by

$$v_k = \begin{cases} 0 & \text{if } w_k < c \\ w_k & \text{otherwise.} \end{cases}$$

Without loss of generality and for the rest of the proof we may reorder the weights and assume the existence of $r \in \{0, \ldots, N-1\}$ with

$$\underbrace{w_1 \leq \cdots \leq w_r}_{<c} < \underbrace{w_{r+1} \leq \cdots \leq w_N}_{\geq c},$$

such that $w_r < c$ and $w_{r+1} \geq c$. A lower bound of $n_e\left(V^N\right)$ follows immediately and reads

$$n_e\left(V^N\right) = \frac{\left(\sum_{k=r+1}^N w_k\right)^2}{\sum_{k=r+1}^N w_k^2} \geq (N-r)\left(\frac{c}{C}\right)^2 \tag{4.28}$$

The existence of a sub-sequence $(L_{k_j})_{j>0}$ with $L_{k_j} > c$ forces the algorithm to regularly generate weights being larger than c. It follows that $(N-r) \to +\infty$ in probability and thus $n_e\left(V^N\right) \to +\infty$. Furthermore, $n_e\left(W^N\right)$ can be rewritten as

$$\begin{aligned} n_e\left(W^N\right) &= \frac{\left(\sum_{k=1}^r w_k + \sum_{k=r+1}^N w_k\right)^2}{\sum_{k=1}^N w_k^2} \\ &= \frac{1}{\sum_{k=1}^N w_k^2}\left[\Big(\sum_{k=r+1}^N w_k\Big)^2 + R\right] \\ &= \frac{1}{\sum_{k=1}^N w_k^2}\left[\left(\sum_{k=r+1}^N w_k^2\right) n_e\left(V^N\right) + R\right]. \end{aligned}$$

with $R = 2\Big(\sum_{k=r+1}^N w_k\Big)\Big(\sum_{k=1}^r w_k\Big) + \Big(\sum_{k=1}^r w_k\Big)^2$. From property 4.3.2($i$) it can be seen that $1 \leq n_e\left(V^N\right) \leq N - r$ and therefore

$$n_e\left(W^N\right) \geq \frac{n_e\left(V^N\right)}{\sum_{k=1}^N w_k^2}\left[\sum_{k=r+1}^N w_k^2 + \frac{R}{N-r}\right].$$

But from the non-negativity and the ordering of the weights, it follows that

$\sum_{k=r+1}^{N} w_k \geq (N-r)w_r$ and

$$\begin{aligned} R &\geq \Big(\sum_{k=r+1}^{N} w_k\Big)\Big(\sum_{k=1}^{r} w_k\Big) \\ &\geq (N-r)w_r \sum_{k=1}^{r} w_k \;\geq\; (N-r)\sum_{k=1}^{r} w_k^2. \end{aligned}$$

We finally obtain the bound

$$n_e\left(W^N\right) \;\geq\; \frac{n_e\left(V^N\right)}{\sum_{k=1}^{N} w_k^2}\left[\sum_{k=r+1}^{N} w_k^2 + \sum_{k=1}^{r} w_k^2\right] \;=\; n_e\left(V^N\right).$$

It was shown above that $n_e\left(V^N\right) \to +\infty$ from where we can now deduce that $n_e\left(W^N\right) \to +\infty$. □

The idea of equation (4.26) is to ensure that the computation of $\hat{\mu}_\alpha$ is based on at least l_0 significant models. Furthermore, it assures that the growth rate of $n_e(W^N)$ and $n_e(W_\alpha^N)$ are equal for $n_e(W^N) > l_0$ which (by lemma 4.3.3) is important for the asymptotic behavior of the method. Finally the method for computing posterior probabilities of events is proposed by the pseudo code in the algorithm 4. The computation of α can be translated into a smooth, 1-

Algorithm 4 Prediction

1: **Input:** l_0 and $f(\cdot)$
2: **Output:** $\hat{\mu}_{\alpha,\mathrm{sn}}$
3: compute α such that $n_e(W_\alpha^N) = \max\left\{l_0,\, n_e(W^N)\right\}$
4: **for** $k = 1, \ldots, N$ **do**
5: **if** $(w_k)^\alpha > 0$ **then**
6: compute $f(\mathbf{m}^k)$
7: **end if**
8: **end for**
9: compute $\hat{\mu}_{\alpha,\mathrm{sn}}$

dimensional optimization problem, and does not require a considerable amount of computational time. The most important effort usually goes into the evaluation of $f(\mathbf{m}^k)$. All the weights are known in advance and therefore we can omit computations that are associated with zero weights. Furthermore, the iterations in the algorithm 4 are independent and can be performed simultaneously in parallel.

Chapter 5

Applications of PoPEx

The PoPEx method is very promising in being a powerful and highly parallelizable Monte Carlo sampling scheme. Its asymptotic behavior is demonstrated by the lemma 4.3.3. In the present chapter, several test examples are established and used for assessing the performance of PoPEx. The main purpose of these examples is to set up test conditions where uncontrollable error sources are minimized and the sampling scheme can be evaluated by an empirical convergence analysis. More precisely, the modeling uncertainties are reduced by generating a synthetic reference domain that provides a set of observations. In this sense, the reference domain is an artificial representations of reality and is unknown to the PoPEx algorithm. It is generated by the same conditional simulation tool that also defines the model space $\mathcal{M}$ in the PoPEx algorithm. Therefore, the modelization uncertainties are negligible and the likelihood measure is given as in equation (2.6). The posterior density function reads

$$\sigma(\mathbf{m}) = c\rho\big(\mathbf{m}\big)\nu\big(\mathbf{g}(\mathbf{m})\big),$$

where ρ and ν describe the prior and the data space uncertainties.

Furthermore, the problem sizes are chosen sufficiently large to encompass interesting model spaces, but still small enough to compute 'exact' posterior solutions. In example 4.1.3 it was suggested to produce N unconditioned simulations and approximate posterior estimates such that

$$\hat{\mu}_{\mathrm{sn}} = \sum_{k=1}^{N} f(\mathbf{m}^k)\tilde{w}_k,$$

where

$$\tilde{w}_k = \frac{L(\mathbf{m}^k)}{\sum_{j=1}^{N} L(\mathbf{m}^j)}, \qquad \text{for all} \quad k = 1\ldots, N.$$

The drawback of this method is its inefficiency. This means that for obtaining a reasonable approximation of μ, a large number of samples are needed. But if the

problem is not too complex, this idea can be used to produce a vast number of unconditioned models and compute estimators from the above equation. It is clear that the model ensemble must be sufficiently large such that the degeneracy problem of section 4.3.3 does not exist. The resulting predictions are considered to be the exact posterior solutions and are denoted by μ^{ex}. Although the number of realizations is very large, it is not unsoiled to call the solutions to be exact. Nevertheless these are very accurate approximations of the true quantity such that, in this work, we will call them 'exact prediction' or 'exact solution'. They can be used for an empirical convergence analysis of the PoPEx algorithm.

Lemma 4.3.3 states that for a sufficiently smooth function f, the estimator $\hat{\mu}_{\alpha,\mathrm{sn}}$ converges to the true quantity μ. We want to see this result in practice on concrete examples. After each iteration $k = 1, \dots, N$, it is possible to compute a PoPEx estimator $\hat{\mu}^k_{\alpha,\mathrm{sn}}$. Such estimators often represent posterior probability maps. A convenient distance between $\hat{\mu}^k_{\alpha,\mathrm{sn}}$ and μ^{ex} is obtained by the Jensen-Shannon divergence (JSD) (e.g. Lin [2006]), reading

$$J(\hat{\mu}^k_{\alpha,\mathrm{sn}} \,||\, \mu^{\mathrm{ex}}) = \frac{1}{2}\Big(D\big(\hat{\mu}^k_{\alpha,\mathrm{sn}} \,||\, \overline{\mu}\big) + D\big(\mu^{\mathrm{ex}} \,||\, \overline{\mu}\big)\Big). \tag{5.1}$$

In this expression $\overline{\mu} = (\hat{\mu}^k_{\alpha,\mathrm{sn}} + \mu^{\mathrm{ex}})/2$ denotes an average map and $D(\cdot||\cdot)$ the Kullback-Leibler divergence (cf. equation (4.9)). The JSD is computed pointwise for every model index and therefore defines one distance value per model parameter. A (scalar) error value is obtained by computing the average of the Jensen-Shannon divergence map over all parameter indices. Considering such error measures for each $k = 1, \dots, N$, gives an approximation curve that is expected to converge to zero.

In the following, we discuss three different problems that use categorical MPS simulations (cf. section 3.2.2) to characterize petrophysical subsurface properties. The example in section 5.1 considers a standard 2-facies setup that often serves for testing algorithms. This example is also used to provide a visualization of the PoPEx learning scheme (cf. section 5.1.1). The section 5.2 provides an extension of the first problem by employing multiple training images. Finally, the last example in section 5.3 addresses a 4-facies problem with a transient forward operation.

5.1 Groundwater Production

This first setup was originally introduced by Mariethoz et al. [2010a]. It considers stationary groundwater production in a 2-dimensional domain where the model parameters describe channelized petrophysical properties. A synthetic reference domain provides 9 hydraulic head measurements that are used in the inverse procedure and represent the data set.

Model Space

We consider a single model type setup that describes the spatial heterogeneities of a subsurface region. The realizations $\mathbf{m}$ are defined in a simulation grid of 100×100 pixels and obtained from the DeeSse algorithm with the training image in figure 3.3. This means that any model contains $n = 10\,000$ parameters that are categorized into the facies values 'blue' and 'red'. Two unconditioned random realizations are illustrated in the upper part of figure 5.1. The prior

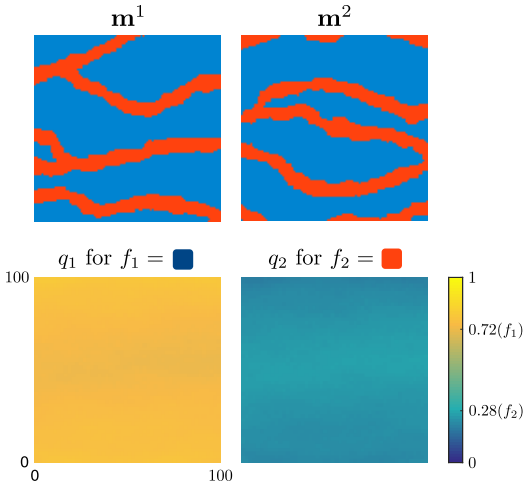


Figure 5.1: Two unconditioned model parameter maps (top) and the corresponding categorical prior distribution Q (bottom).

categorical density Q is approximated from a large number of unconditioned realizations and is shown in the lower part of figure 5.1. Their average probability values of 0.28 and 0.72 match the corresponding facies proportion in the training image.

Data Space

Let the 100×100 model parameters represent a computational domain of $100(m) \times 100(m)$. The two model categories stand for uniform transmissivity

values of $10^{-2}(m^2/s)$ ('red' or 'channels') and $10^{-4}(m^2/s)$ ('blue' or 'matrix'), respectively. It is assumed that the system is closed on the upper and the lower domain limit, where no-flow boundary conditions apply. A generic flow direction is imposed from the left to the right by fixing the respective boundary head values to $1(m)$ and $0(m)$. One realization with an arbitrary seed (for the MPS random process) was generated and is considered to be the reference domain (c.f. left figure of 5.2). A pumping well is located in the center

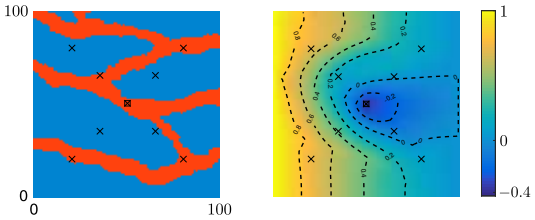


Figure 5.2: Reference domain (left) and the corresponding hydraulic head values (right).

of the domain (square) and extracts groundwater with a rate of $3(l/s)$. At nine locations (crosses), the hydraulic head values of the numerical reference solution are retained (c.f. right figure of 5.2); these observations represent the only data constraints that are used for solving the inverse problem. Thus, the data set reads $\mathbf{d}^{\text{obs}} = \{d_1^{\text{obs}}, \ldots, d_9^{\text{obs}}\}$. A link between the model and the data space is established by the GroundWater subsurface modeling software [Arpat and Caers, 2007]. This is a numerical solver for partial differential equations and can be used to compute the hydraulic head values for any point in the computational domain. For this reason, it forms the main ingredient in the definition of the forward operator $\mathbf{g}$. We assume the measurement errors to be independent and to follow a 9-dimensional Gaussian distribution. In this case, the likelihood value can be measured such as

$$L(\mathbf{m}) \propto \exp\left(-\frac{\text{MSE}(\mathbf{m})}{2\sigma^2}\right), \tag{5.2}$$

where $\text{MSE}(\mathbf{m}) = \frac{1}{9}\sum_{i=1}^{9}\left(d_i^{\text{obs}} - g_i(\mathbf{m})\right)^2$ denotes the mean square error (MSE) between the predictions and the reference values. The standard deviation is set to $\sigma = 0.05(m)$, what reasonably matches measurement errors met in practice.

5.1.1 Visualization of the PoPEx Sampling

Before analyzing the PoPEx results let's use the above problem to illustrate the key steps of the PoPEx algorithm. We recall that the method iteratively learns the maps P^k and $\widetilde{D}(P^k||Q)$ from the previous models and constructs

a set of synthetic hard conditioning data that is imposed in the simulation of $\mathbf{m}^{k+1}$. Figure 5.3 shows the evolution of the maps P^k and $\widetilde{D}(P^k||Q)$ as well as some conditioned realizations $\mathbf{m}^{k+1}$ for $k = 200$, 500, and 2000. This

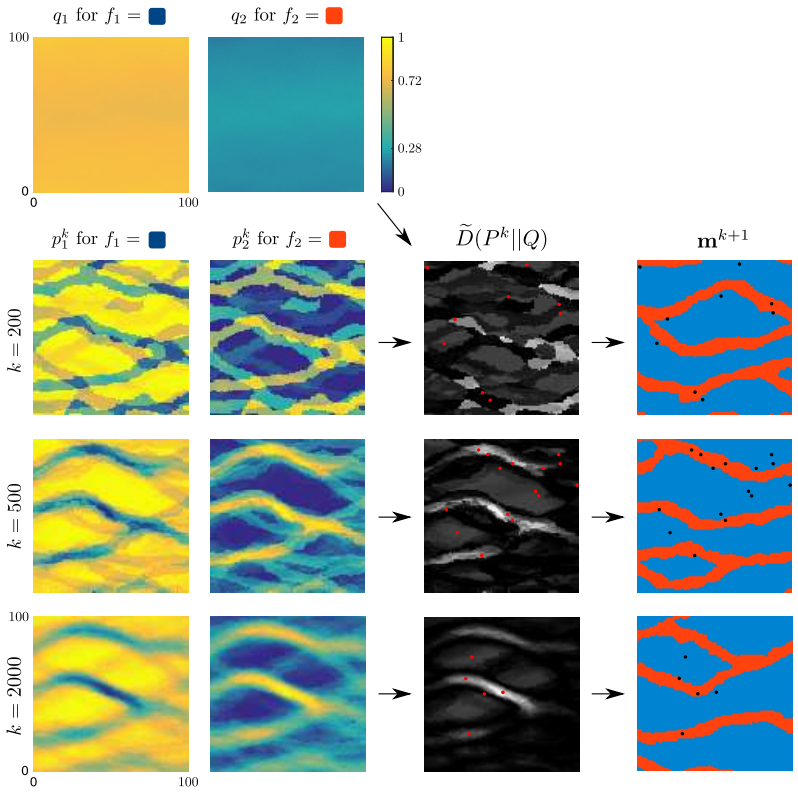


Figure 5.3: Illustration of the PoPEx workflow. The categorical distributions Q and P^k together with the resulting Kullback-Leibler divergence map $\widetilde{D}(P^k||Q)$ are shown for the iterations $k = 200$, 500, and 2000. The last column shows the new realization that was obtained from the synthetic hard conditioning data $\mathbf{hd}^k$ (indicated by the black dots).

figure is separated into multiple parts. The two images in the top row are the (fixed) categorical prior densities for each facies type. The maps q_1 and q_2 take constant values of approximately 0.72 and 0.28, respectively. Then, the first two columns of the matrix below illustrate the categorical densities p_1^k and p_2^k after the iterations $k = 200$, 500, and 2000. The third column is the normalized Kullback-Leibler divergence $\widetilde{D}(P^k||Q)$. In each iteration this map is used for sampling n_k hard conditioning indices and form the set I. The selected locations are indicated with red dots. As q_1 has higher values than q_2,

large probabilities in the map p_2^k contain more information than large values in p_1^k. This is because they are more surprising with respect to the prior probability densities q_1 and q_2. It can be observed that this information content is well represented by the Kullback-Leibler divergence $\widetilde{D}(P^k||Q)$. Afterwards, the algorithm uses $\widetilde{\Sigma}^k = \left\{\tilde{\sigma}_1^k, \ldots, \tilde{\sigma}_k^k\right\}$ to pick n_k conditioning values from $\mathbf{m}^1, \ldots, \mathbf{m}^k$ and generate $\mathbf{m}^{k+1}$ according to $\mathbf{hd}^k = (I, \mathbf{v})$. The new realizations are shown in the last column, where the black dots indicate the location where synthetic hard conditioning data was applied. It can be seen that the number n_k, the locations in I as well as the hard conditioning values $\mathbf{v}$ change in each iteration.

The following results are split in tree sections 'solution', 'prediction', and 'convergence'. In the first part, we analyze the efficiency of the learning scheme for the proposal densities ϕ_k and show the reproduction quality (i.e. the 'fit') of the data $\mathbf{d}^{\text{obs}}$. Then we are interested in producing categorical probability predictions, before we finally examine the parallel scaling of the method by a convergence analysis.

5.1.2 Solution of the Inverse Problem

The PoPEx sampling scheme produced a chain of 10 000 models and was limited to use not more than $n_{\max} = 20$ conditioning data pairs. The categorical prior distribution Q was approximated from 1000 unconditioned simulations and is shown in the top row of figure 5.3. It was mentioned that the proposal distribution ϕ_k must be closely inspired by the posterior density. If this is true, the adaptive importance sampling scheme regularly generates models $\mathbf{m}$ with important posterior measure value $\sigma(\mathbf{m})$. This can be difficult because the regions that have important posterior density are usually very sparse with respect to the complexity of the entire space $\mathcal{M}$. This is precisely the drawback of the prior sampling scheme in example 4.1.3. Similarly to a common rejection sampler (e.g. Winkler [2012]) it generates many realizations with unimportant posterior measure value and is therefore very inefficient. As a measure of quality for the sampling distribution, we can compute the frequency of generating 'good' models. A (rather arbitrary) quality threshold is

$$\text{RMSE}(\mathbf{m}^k) \leq 0.07(m), \tag{5.3}$$

with $\text{RMSE}(\mathbf{m}^k) = \sqrt{\text{MSE}(\mathbf{m}^k)}$ being the root mean squared error. This corresponds to the 95% confidence interval of the data distribution defined by 9 independent observations with Gaussian errors. The prior density ρ is fairly flat so that the likelihood and the posterior density functions are almost proportional to each other. Therefore, it is reasonable to associate the quality of the sampling distributions to the frequency of generating models that satisfy equation (5.3). As a quality comparison, we consider two existing McMC schemes and unconditioned prior sampling. In their study, Mariethoz et al. [2010a] presented two Markov chain Monte Carlo methods that are entitled

Iterative Spatial Resampling (ISR) and *Interrupted Markov chain Monte Carlo (IMcMC)*. Both algorithms use MPS schemes for the generation of a chain of models $\mathbf{m}^1, \ldots, \mathbf{m}^N$. Their evolution mainly depends on the likelihood ratio $L(\mathbf{m}^{k+1})/L(\mathbf{m}^k)$ of two consecutive realizations. In each iteration k, the ISR method extracts facies values from $\mathbf{m}^k$ and uses them as hard conditioning data for the generation of a candidate model $\mathbf{m}^*$. This model is then accepted with a probability of $\min\{1, L(\mathbf{m}^*)/L(\mathbf{m}^k)\}$. As suggested by Mariethoz et al. [2010a], the correlation between consecutive models is reduced by only considering realizations that are at a sampling distance of at least 12. Conversely, the IMcMC method also extracts hard conditioning data from the previous realization but accepts a new model whenever $L(\mathbf{m}^*) \geq L(\mathbf{m}^k)$. If a realization satisfies equation (5.3) the chain is interrupted and restarted. For both McMC methods we fix the number of conditioning points to 100.

In figure 5.4 it can be observed that for generating 200 models that satisfy equation (5.3), PoPEx only needed 3919 simulations, while the McMC methods required 15 698 (ISR) and 21 052 (IMcMC), and the unconditioned sampler 42 136 realizations. This means that the PoPEx algorithm was roughly 4 (resp.

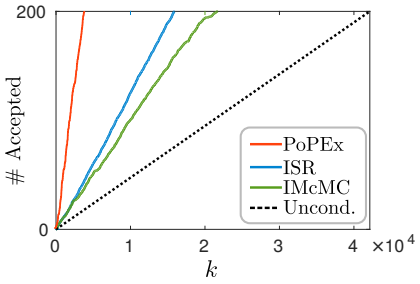


Figure 5.4: Acceptance rate of 'good' models (in the sense of equation (5.3)) for PoPEx, ISR, IMcMC, and unconditioned prior sampling versus the number of iterations k.

5.3) times faster than ISR (resp. IMcMC) and more that 10 times more efficient than prior sampling. It suggests that the proposals ϕ_k generate important models with great regularity. The quality of the realizations is often measured by their ability to 'fit the data'. This terminology is used to call for models $\mathbf{m}$ where the data prediction $\mathbf{g}(\mathbf{m})$ coincides with the observations in $\mathbf{d}^{\text{obs}}$. The probabilistic inversion however does not try to reproduce the measurements but characterizes the data uncertainty when $\mathbf{d}^{\text{obs}}$ was observed. It can therefore be used to characterize the marginal posterior uncertainty of single observations in the neighborhood of $\mathbf{d}^{\text{obs}}$. For visualizing the posterior information content, it is often useful to compare the posterior data distributions to their prior equivalents. Figure 5.5 shows the prior (left boxplot) and the posterior (right boxplots) marginal data distributions for the 9 hydraulic head measurements. The left square represents the computational domain and shows the locations

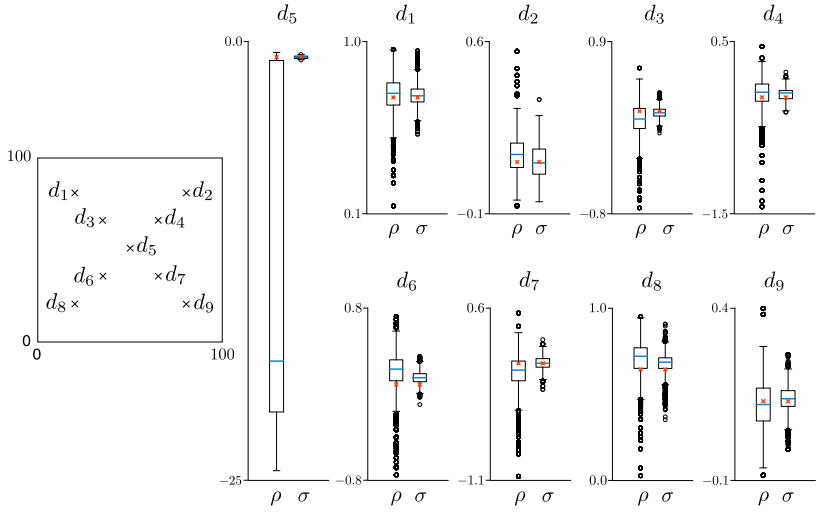


Figure 5.5: Prior (left) and posterior (right) marginal data distributions for the 9 hydraulic head measurements.

where the hydraulic head values have been measured. For each observation, the figure shows a boxplot pair that compares the prior and posterior marginal distributions of head values. The red crosses represent the observed values in $\mathbf{d}^{\text{obs}}$, while the blue line is the prior and the posterior median, respectively. It can be observed that the prior value range heavily changes from one observation to the other and is largest at the pumping well (observation d_5). This is not surprising because the extraction rate of $3(l/s)$ if fixed regardless the underlying model. It means that in a (physically unrealistic) setup where the pumping well is surrounded by low permeable material, it still extracts the same quantity of groundwater, what results in a considerable drop of the hydraulic head values around the extraction location. If we are interested in comparing the information content in the prior and the posterior density, we can compare the relative value ranges of the two boxplots. It can be observed that when passing from the prior to the posterior distribution, the marginal uncertainty is reduced. The interquartile ranges (IQR) and the respective reductions are presented in table 5.1. It can be seen that the narrowing effect, measured by $\frac{\text{IQR}_\sigma}{\text{IQR}_\rho}$, is the more important the closer the observation is to the extraction well. Roughly speaking, the reduction of the uncertainty can be interpreted as an importance measure of the data point in the contribution to the likelihood value. When comparing observation pairs that are at the same distance and the same height (i.e. $d_1 \leftrightarrow d_2$, $d_3 \leftrightarrow d_4$, $d_6 \leftrightarrow d_7$, and $d_8 \leftrightarrow d_9$), it is conspicuous that three out of four times the left data point shows a larger uncertainty reduction and therefore seems to be 'more sensitive' for large posterior mea-

Table 5.1: Prior and posterior interquartile ranges (IQR) for the marginal hydraulic head values.

Observation	IQR_ρ	IQR_σ	$\frac{\mathrm{IQR}_\sigma}{\mathrm{IQR}_\rho}$
d_1	0.115	0.067	58.7%
d_2	0.102	0.100	97.9%
d_3	0.235	0.079	33.9%
d_4	0.210	0.103	49.2%
d_5	21.101	0.084	0.4%
d_6	0.202	0.078	38.9%
d_7	0.193	0.087	44.9%
d_8	0.117	0.059	49.9%
d_9	0.102	0.047	46.1%

sure values. The reason for this can be found in the boundary conditions that induce a general groundwater flow from the left to the right side. But there is one exception: the observation d_9 is located on a channel that is directly linked to the pumping well (cf. figure 5.2) which makes the problem more vulnerable to this observation.

5.1.3 Predictions

When working with categorical facies models, we are often interested in computing the posterior probability map for each category. In other words, we want to quantifying the integrals

$$\mu = \int_{\mathcal{M}} \mathbf{1}_{f_i}(\mathbf{m})\sigma(\mathbf{m})d\mathbf{m}, \tag{5.4}$$

for any facies type f_i. If, as in the present case, the models $\mathbf{m}$ are represented by 2-dimensional facies maps, the above integral also defines a posterior probability diagram on the computational domain. In the introduction of this chapter it was mentioned that it is possible to compute reference maps μ^{ex} from a large number of unconditioned simulations. For the present example this can be done in reasonable time and we computed the exact solution from 300 000 unconditioned random realizations. At the same time, the PoPEx chain was used to produce facies probability predictions $\hat{\mu}_{\alpha,\mathrm{sn}}$ with α being deduced from $l_0 = 100$ (cf. equation (4.26)). Again, we are interested in comparing the performance of PoPEx not only with the exact solution μ^{ex} but also with the predictions of ISR and IMcMC. The two McMC predictions $\hat{\mu}_{\mathrm{ISR}}$ and $\hat{\mu}_{\mathrm{IMcMC}}$ are computed by running comparable amounts of forward operations as in the PoPEx chain. Here, we are working in a 2-facies setup and therefore it is sufficient to only consider probability values for one of the categories. Figure 5.6 illustrates the posterior facies predictions for the category 'red' by using unconditioned prior

sampling (top left; the exact solution), PoPEx (top right), ISR (bottom left), and IMcMC (bottom right). There are significant differences to be observed.

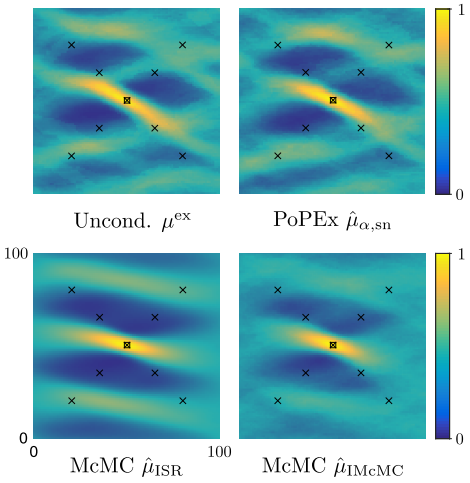


Figure 5.6: Comparison of the posterior facies probability maps for category f_2 ('red'), obtained by unconditioned prior sampling (exact solution), PoPEx, ISR, and IMcMC.

Comparing them visually, the best approximation was clearly obtained by the PoPEx chain. All estimators succeeded in putting a strong probability for the pumping well to be placed in a highly permeable region. However, the IMcMC map does almost completely fail to show sharp probability contrasts and regions where channels are unlikely to be found. The ISR picture on the other hand, mainly shows three parallel downward structures. There is only a very small probability for them to be interconnected, what does not correspond to the reference map. The lesser contrasted small bifurcations are only discovered by the PoPEx algorithm. Therefore, from a visual examination, we may agree that the differences are remarkable. Let us compare them computationally by an error measure as in equation (5.1).

5.1.4 Convergence Analysis and Parallel Scaling

The asymptotic behavior of the PoPEx chains is demonstrated by lemma 4.3.3. Here, we use the above bi-category problem to provide an empirical evidence for the convergence of $\hat{\mu}_{\alpha,\text{sn}}$. We analyze two PoPEx setups with $n_{\max} = 10$ resp. $n_{\max} = 20$ and compare both to the predictions of ISR and IMcMC. For each of the methods and after each iteration k we use $\mathbf{m}^1, \ldots, \mathbf{m}^k$ to predict facies probability maps for f_1 and f_2. These maps are defined in the simulation grid and therefore describe 10 000 probability values. As proposed above,

a scalar error measure is obtained by the average Jensen-Shannon divergence (cf. equation (5.1)) between the predictions $\hat{\mu}^k_{\alpha,\mathrm{sn}}$, $\hat{\mu}^k_{\mathrm{ISR}}$, $\hat{\mu}^k_{\mathrm{IMcMC}}$ and μ^{ex}. For obtaining more significant outcomes, all of the following convergence results represent an average performance of 5 runs with different initial seeds. Figure 5.7 shows the error curves for the two PoPEx and the two McMC schemes. The PoPEx curves are quite similar to each other and converge very fast with

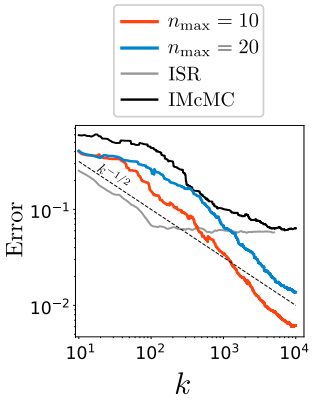


Figure 5.7: Error between $\hat{\mu}^k_{\alpha,\mathrm{sn}}$, $\hat{\mu}^k_{\mathrm{ISR}}$, $\hat{\mu}^k_{\mathrm{IMcMC}}$ and μ^{ex} after each realization k. The PoPEx estimators have been defined by $l_0 = 100$ and $n_{\max} = 10$ or $n_{\max} = 20$, respectively.

a consistent rate. They both greatly outperform the convergence of the McMC schemes where the approximation quality reaches a plateau (ISR) or is improving very slowly (IMcMC). These issues are both quite common when McMC schemes are used for exploring highly complex and multi-modal densities. The PoPEx chains do not show any of those difficulties. For independent samples according to σ, the central limit theorem (CLT) [Durrett, 2010] predicts a convergence rate of $k^{-1/2}$. Although the PoPEx chains are not independent, their convergence rates coincide with the CLT-prediction. This is an important result, because it can be understood as an empirical evidence for the ergodicity of the PoPEx algorithm and therefore for lemma 4.3.3.

The parallelization of the PoPEx algorithm described in section 4.2.2 has great potential in terms of scalability. The main workload of a PoPEx loop consists in the four tasks of deriving a synthetic hard conditioning set, generating a new model, computing its likelihood value and updating the KLD map. The first three out of those four steps are encapsulated in subprocesses. Considering Amdahl's law in equation (1.1), this means that the fraction p of serial work consists of the computations that are related to the Kullback-Leibler divergence. There is no need to recompute this map from scratch in every iteration but it can simply be updated. The KLD is defined by a vector of length n (being the number of model parameters) and the computational

cost for updating the KLD map is of the order of n (i.e. $p = O(n)$). The associated workload is therefore negligible with respect to the remaining three tasks and the PoPEx parallelization *scales perfectly* (cf. chapter 6).

Let us consider two parallel PoPEx setups with 15 and 30 subprocesses (i.e. $n_{\text{par}} = 15$ and $n_{\text{par}} = 30$), respectively. The first chain is carried out on a 32 CPU facility ($17.2(Tflop/s)$) while the second runs on 64 CPU's ($34.4(Tflop/s)$). This means that between the two tasks, the computational resources have been doubled. The total runtime was $1.02 \pm 0.259(h)$ and $0.52 \pm 0.093(h)$, respectively, what signifies an overall speedup factor of 1.95 ± 0.602. This fully satisfies the request for perfect parallel scalability. A second test considers the speedup factor for obtaining the same prediction accuracy. This means that for a fixed error value (i.e. the approximation quality) we measure the time that is needed to produce facies predictions that reach the required accuracy. Figure 5.8a shows the two convergence curves versus the elapsed time. Again, we notice that the two convergence curves are very similar and

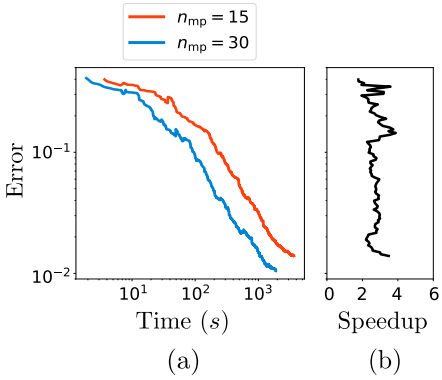


Figure 5.8: Speedup factor when comparing the computational time for similar facies predictions.

separated by a regular gap in computational time. This becomes even more evident when considering the part 5.8b that shows the speedup factor (x-axis) for each accuracy level (y-axis). We observe that for most of the exercise, the speedup factor is reasonably close to 2 what fully satisfies our expectations. Small variations should not be overestimated because the statistics are based on the (relatively small) set of 5 chains. It is therefore save to say that based on the considered examples, the PoPEx parallelism scales perfectly.

5.2 Multiple Training Images

For this example we consider the same problem as in section 5.1 but with a larger model space. The main goal is to run similar exercises and compare the performances.

Model Space

Here, the model space embeds two different model types. One of them, as in the previous example, describes spatial heterogeneities on a subsurface region and is simulated by DeeSse on a grid with 100×100 pixels. The second model type however, is used to select the training image for the MPS simulations. 12 different pictures are generated from the image in figure 3.3 by rotating it counter-clockwise for $i \cdot 30°$ degrees with $i = 0, 1, \ldots, 11$ (cf. figure 5.9). The corresponding model type is a single discrete variable with values in

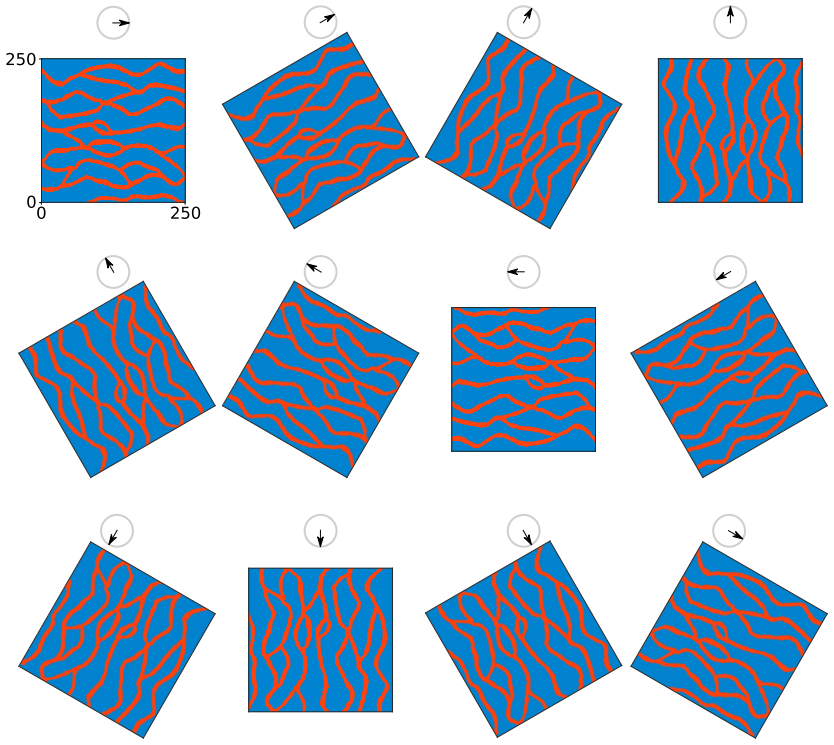


Figure 5.9: Possible training images that are obtained by rotating it clockwise in steps of 30° degrees.

$\{1, 2, \ldots, 12\}$ that indicates the choice of the TI. Each model $\mathbf{m}$ contains two

model types $\{\mathbf{m}_1, \mathbf{m}_2\}$ and totally $n = 10\,001$ model parameters. The spatial heterogeneities are still categorized into 'blue' and 'red' with prior categorical distribution as in the top row of figure 5.3. The TI-model type has 12 categories $f_j = \{j\}$ for $j = 1, \ldots, 12$, with uniform prior probability. In summary, this means that a model $\mathbf{m}$ is obtained by combining the two conditional simulators in the sections 3.2.1 and 3.2.2.

Data Space

As above, the 100×100 model values parameterize the subsurface heterogeneity map of a $100(m)$ by $100(m)$ computational domain where 'blue' is associated to low and 'red' to high transmissivity. The forward problem $\mathbf{g}$ involves the same boundary conditions and pumping rates so that the likelihood measure is defined as in equation 5.2.

Again, the PoPEx sampling scheme runs for 10 000 models. It was mentioned that the conditioning bound $n_{\max}$ is defined model type wise and therefore must have two components; we set $n_{\max} = \{1, 20\}$. This means that the selection of the training image can be conditioned by 1 value (which is sufficient to fully describe this model type) and the spatial heterogeneities can maximally be influenced by 20 synthetic hard data pairs. Note that the randomly drawn set of conditioning numbers also has two components, one for each model type. It can therefore be denoted by $n_k = \{n_{k,1}, n_{k,2}\}$ where $n_{k,1}$ is either 0 or 1 and $n_{k,2}$ is an integer between 0 and 20.

5.2.1 Predictions

Our first interest is again to consider facies prediction maps (cf. equation (5.4)). Figure 5.10 illustrates the PoPEx prediction (right) with $l_0 = 100$ and compares it the to exact solution that was obtained from 300 000 unconditioned models (left). We observe results that are very similar to the PoPEx predic-

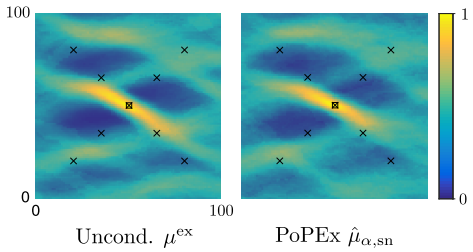


Figure 5.10: Comparison of the posterior facies probability maps for category f_2 ('red').

tions in figure 5.6. This means that the exact solution is closely approximated with the main structures being quite similar. Again, the method has no issue

to almost surely deduce that the pumping location is surrounded by highly permeable material. But also the less contrasted probability structures are greatly matching the reference ones in the left image. This is very convenient because the PoPEx sampling produced the same total number of samples but had 12 (instead of 1) possible training images to choose from. One of the advantages of PoPEx is that this choice is not completely blind. Hence, by conditioning the TI selection procedure, it is possible to prefer some images over others. Setups that are very unlikely to produce high posterior values will automatically be detected and mostly avoided. The prior probability of selecting a TI is uniformly set to $\frac{1}{12}$. In figure 5.11 these values are represented for each orientation by the blue bar on the left side of each histogram. The learning process of PoPEx can

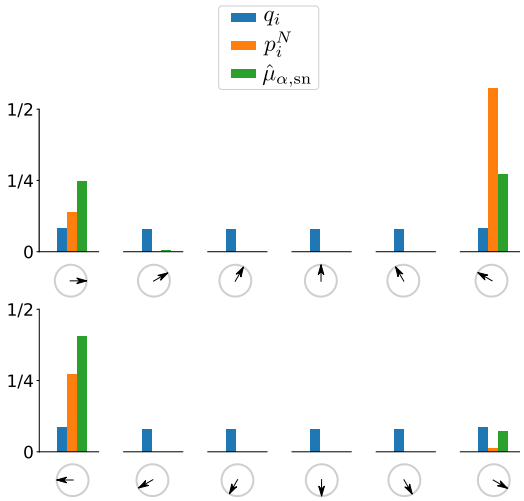


Figure 5.11: Categorical probability values of Q and P^N together with the posterior training image prediction $\hat{\mu}_{\alpha,\mathrm{sn}}$.

best be illustrated by considering the maps P^N. For the model type that is associated with the training image selection, the categorical probability values in P^N represent the importance that the algorithm associate with each picture. These values are represented by the central orange bars in figure 5.11. The algorithm only gives significant potential to the images with orientation 0°, 150°, and 180°. We recall that the reference domain was obtained by an arbitrary seed together with the first training image. This explains why this picture and its 180°-reversion have great importance. It is striking however, that the 6-th training image (with orientation equal to 150°) is very likely to produce 'good' models as well. But when considering the reference domain in figure 3.3 it can be seen that the most important central channel has a downward orientation. This explains why the downward oriented 6-th picture is also very important for the posterior measure function. We even observe that the 12-th TI (which

represents a 180° rotation of the 6-th picture) has still a non-zero significance. The PoPEx models can be used to compute an estimator $\hat{\mu}_{\alpha,\mathrm{sn}}$ that predicts the posterior importance of each picture. These probability values are represented by the green bars on the right hand side in the histograms of figure 5.11. Again, it is obvious that the picture 1 and 6 with their respective reversals (7 and 12) are most likely. Considering the two figures 5.10 and 5.11 we may conclude that the PoPEx learning scheme was very efficient.

5.2.2 Convergence Analysis and Parallel Scaling

The last statement is further enhanced by the following experiment, where we compare convergence curves of the single and the multiple-TI setups. As in figure 5.7, we compute the prediction error for the categorical facies probabilities after each iteration k and compare it to the exact solution μ^{ex}. Figure 5.12 shows the prediction accuracy for the single (red) and the multiple (blue) picture problem. The similarity of the two curves is remarkable and means

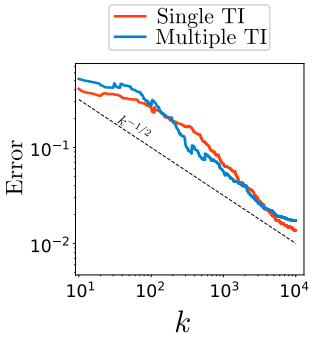


Figure 5.12: Comparison of the convergence curves for predicting categorical facies probabilities in the single and the multiple-TI setup.

that adding multiple choices of the training image does not significantly postpone the prediction quality. The reason for that can again be found in figure 5.11. The model type that is associated to the selection of the training image has only one discrete model parameter. The corresponding values in the type specific probability maps $\{p_1^k, \ldots, p_{12}^k\}$ measure the training image frequency in $\mathbf{m}^1, \ldots, \mathbf{m}^k$ according to $\widetilde{\Sigma}^k$. Therefore, it can be understood that whenever the choice of the TI is conditioned, PoPEx selects a picture according to $\{p_1^k, \ldots, p_{12}^k\}$, what only chooses among 1, 6, 7, and 12 (orange bars). All of these training images have high posterior values (green bars) so that it is understandable that PoPEx generated 'good' models with similar frequency as in section 5.1. Thus, the learning scheme in P^k is able to quickly sort out useless setups and produce rapidly convergent predictions.

Let's examine again the scaling behavior of the random sampling. As before, we consider two PoPEx chains with $n_{\text{par}} = 15$ and $n_{\text{par}} = 30$ that are produced on computer facilities with 32 CPU's ($17.2(Tflop/s)$) and 64 CPU's ($34.4(Tflop/s)$), respectively. The total runtime was $1.08 \pm 0.181(h)$ and $0.54 \pm 0.102(h)$ what signifies a speedup of 2.02 ± 0.509 and perfectly matches the expectations for embarrassingly parallelizable algorithms. In addition we, compare again the computational time that is requested for obtaining similar prediction accuracy. Figure 5.13a shows the error curves for the two PoPEx setups versus the elapsed time. The two graphs are nicely parallel with

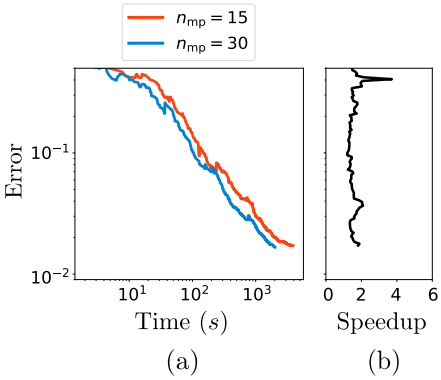


Figure 5.13: Speedup factor when comparing the computational time for similar facies predictions.

an reasonably constant speedup gap in between. Figure 5.13b shows the acceleration factor (x-axis) for any accuracy level (y-axis). We observe that the speedup is very close to 2 for most of the exercise and suggests that the PoPEx parallelization scales perfectly.

5.3 Dye Tracing

In this section, we solve a problem that involves a tracer test in a highly heterogeneous fluvial aquifer. The inverse solution is used to predict the capture zone of a pumping well.

Model Space

As in section 5.1, we consider again a single model type setup that characterizes subsurface heterogeneities in an aquifer. In this example however, the model space is much more complex because we consider 4-facies types and realizations $\mathbf{m}$ that describe $n = 20\,000$ parameter values. The conceptual training image for the geological heterogeneities is derived from a 3D simulation by the FLUMY software [Lopez et al., 2009]. This tool combines geological processes with a stochastic component. A meandering river crosses an alluvial valley with a given slope and causes erosion and deposition of sediments. Over time, the river migrates and alters the topography of the alluvial plain. This evolution generates complex geological patterns with a realistic and highly heterogeneous architecture. However, the thickness of the alluvial sediments is usually negligible with respect to the horizontal dimensions of the plain. It is therefore reasonable to reduce the complexity of the problem and neglect the vertical component of the flow.

Following this approach, a 2-dimensional training image was generated and is illustrated in figure 5.14. The picture represents a domain of $5000(m) \times$

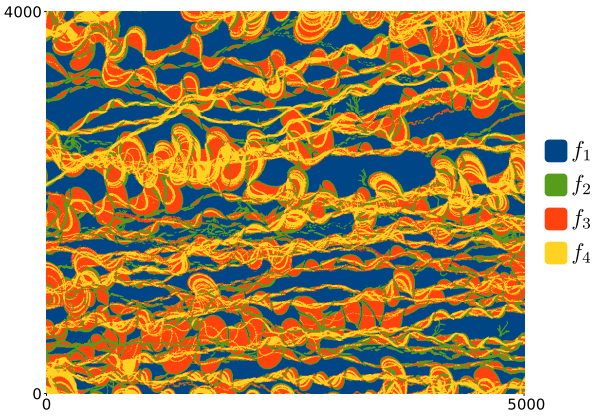


Figure 5.14: Training image

$4000(m)$ and is subdivided into 1000×800 quadratic pixels. It was obtained by truncating the vertical integration of a 3D model into four facies types f_1 ('blue'), f_2 ('green'), f_3 ('red'), and f_4 ('yellow'). Models $\mathbf{m}$ are obtained from

DeeSse simulations on a computational grid of 200×100 quadratic pixels that represents a domain of $1000(m) \times 500(m)$. Two unconditioned realizations and the corresponding categorical prior densities are shown in figure 5.15.

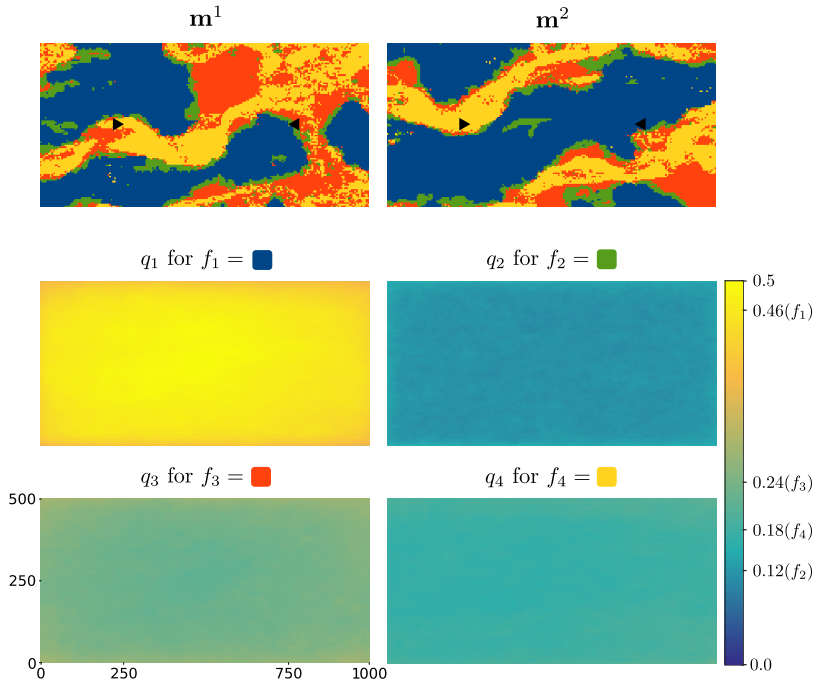


Figure 5.15: Two unconditioned model parameter maps (top) and the corresponding categorical prior distribution Q (bottom).

Data Space

The four facies categories represent transmissivity values of 10^{-5} ('blue'), 10^{-3} ('green'), 10^{-2} ('red'), and $10^{-1}(m^2/s)$ ('yellow'); the drainage porosity and the specific storage were fixed uniformly to 0.2 and 10^{-6}, respectively. A pumping well is installed at location $(750, 250)$. It produces groundwater at a rate of $15(l/s)$ for a total duration of 20 days. The terrain is exposed to a natural slope of 4‰ in the x-direction, while the basin is closed at $y = 0(m)$ and $y = 500(m)$. Corresponding boundary conditions are: fixed head values of $4(m)$ (left) and $0(m)$ (right) together with no-flow on the upper and lower boundaries. A constant tracer concentration of $1(kg/m^3)$ is enforced at $(250, 250)$ throughout the time period.

For any given model, the subsurface water flow together with the tracer expansion is computed by the GroundWater simulation software [Arpat and

Caers, 2007]. At days 2, 4, 6, 8, 10, 12, 15, and 20 the solute concentration is recorded at the pumping well. This provides a set of 8 observations and represents all the data constraints in $\mathbf{d}^{\text{obs}}$. The conditional simulation tool together with an arbitrary seed was used to generate the reference domain in figure 5.16 (left). Black triangles indicate the tracer injection (left, pointing right) and

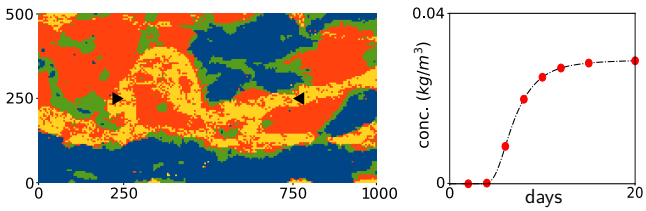


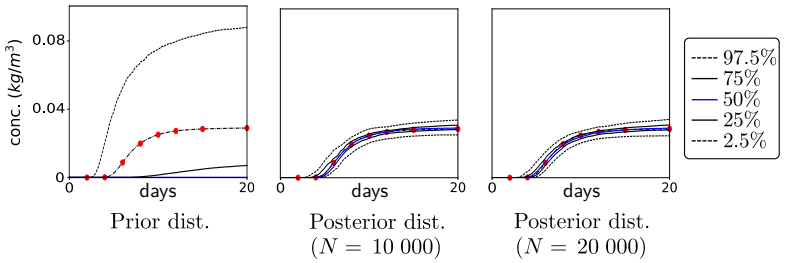
Figure 5.16: The reference domain (left figure) with the tracer injection (left arrow) and the pumping well (right arrow) together with the observed tracer concentration at the pumping well (right figure).

the pumping well location (right, pointing left). The tracer concentration at the pumping well resulting from this reference domain is shown in figure 5.16 (right). Red dots indicate the extracted data that was used in the inverse procedure. This means that the entire reference domain in figure 5.16 is unknown to the PoPEx algorithm. Its only task is to represent an unknown subsurface model and provide the sparse set of tracer concentrations. For constructing $L(\mathbf{m})$, we assume the observations to be independent and consider a multivariate normal distribution between the predictions $\mathbf{g}(\mathbf{m}) = \{g_1(\mathbf{m}), \ldots, g_8(\mathbf{m})\}$ and observations $\mathbf{d}^{\text{obs}} = \{d_1^{\text{obs}}, \ldots, d_8^{\text{obs}}\}$ with uniform standard deviation of $\sigma_L = 0.0015(kg/m^3)$. This represents 1.5‰ of the concentration at the injection point, and roughly 5% of the maximal concentration (cf. figure 5.16). The subscript L in σ_L distinguishes the standard deviation of the likelihood measure from the posterior density σ. Assuming an uniform and independent Gaussian behavior of $\mathbf{g}(\mathbf{m})$ around $\mathbf{d}^{\text{obs}}$, gives a likelihood function that is proportional to $\exp\left(-\frac{1}{2\sigma_L^2}\sum_i (g_i(\mathbf{m}) - d_i^{\text{obs}})^2\right)$.

5.3.1 Solution of the Inverse Problem

PoPEx was trained to run the above problem for a total of $N = 20\,000$ models with $n_{\max} = 25$. The prior probabilities in Q were approximated from 1000 unconditioned MPS simulations. For each realization in the chain, the algorithm computed the tracer concentration at the pumping well, extracted 8 data points and compared them to the reference data in figure 5.16. As for any prediction, the posterior distribution of the tracer breakthrough curve can be approximated by a self-normalized estimator $\hat{\mu}_{\alpha,\text{sn}}$ with $l_0 = 100$ and α as

in equation (4.26). Figure 5.17 shows the 2.5% – 97.5% (dashed), 25% – 75% (full) and average (blue) curves of the prior and the posterior tracer concentration at the pumping well. The red dots indicate the extracted reference data.



Prior dist. Posterior dist. ($N = 10\,000$) Posterior dist. ($N = 20\,000$)

Figure 5.17: Prior and posterior concentration probabilities at the pumping well. The curves indicate 2.5% – 97.5% region (dashed), 25% – 75% region (full) and average value (blue), while the red dots represent the reference concentration.

It is clear that for any sampling strategy, a critical measure is the required computational effort, which usually is proportional to the number of samples. For this reason, all results are shown for two different stages in the sampling procedure: after 10 000 and after 20 000 realizations. At a first glance, both estimations are quite similar. This may be surprising when keeping in mind that the computational effort for the second estimation is twice as high. However, it can be seen that the probability lines are steadier and smoother in the last image. Both estimations of the 50% region (between the full lines) embed the red reference data and follow the shape of the reference curve very precisely. The estimation of the posterior expectation (blue) almost matches the entire curve. The higher density of data points in the first 10 days, increases the relative importance of this period with respect to the second half. Thus, it is reasonable to allow less uncertainty in the beginning of the simulation. The more generous 95% regions (between the dashed curves) are still appropriate in reproducing the shape of the reference curve. This is even more significant when realizing that the prior distribution is far from being centered around the reference curve.

5.3.2 Predictions

In practice, when producing freshwater from an aquifer, it is often crucial to protect the source and determine its capture zone [Leeuwen et al., 1998]. Here, we use the results of the PoPEx model chain for predicting the posterior probabilities of the 10-days capture zone. It means that for each location in the simulation grid, we compute a Bernoulli probability value for the water to be captured within 10 days. Figure 5.18 shows the prior distribution and for the predictions after 10 000 and 20 000 iterations, respectively. As expected, since

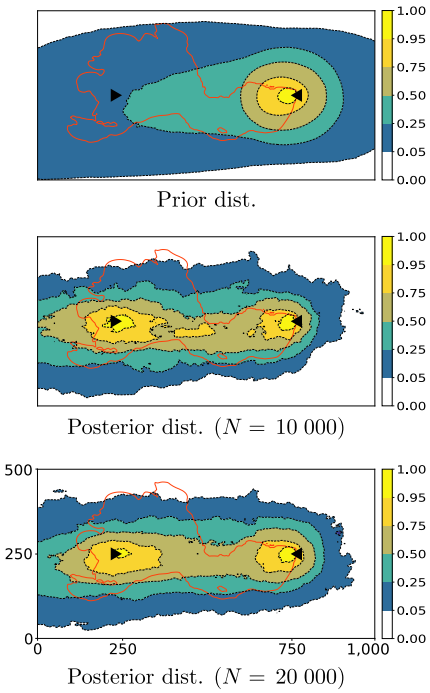


Figure 5.18: Prior and posterior 10 days capture zone probabilities. The curves indicate 5%, 25%, 50%, 75% and 95% regions, while the red lines circumscribes the capture zone in the reference domain.

the tracer is arriving in less than 10 days at the pumping well, the injection point is located within a region having a high probability to belong to the 10-days capture zone. This is already clearly visible in the map generated from 10 000 realizations. These results show the existence of a connected path of high transmissivity between the injection point and the pumping well. However, zones of lower probability are located in between. This indicates that the position of the channel is not well identified from the tracer data alone. In the reference domain, shown in figure 5.16, we can see that the yellow facies (with the largest transmissivity value) first shows a very tight upwards bend before heading almost directly towards the extraction well. The injected tracer will mostly follow the region with the largest transmissivity. Therefore, it does not take a direct path towards the well and its arrival time is delayed. The main information that can be extracted from the observations is this delay and the final concentration. From the available data, it is therefore impossible to predict precise water pathways that are far from the tracer injection and the algorithm is correctly informing us about that uncertainty.

It is interesting that the reference capture zone (red line) slightly passes

outside the 95% region in the top section of the computational domain. This should not be interpreted as an inaccuracy of the PoPEx method, because it similarly happens for the exact solution in figure 5.19. However, it indicates that the training image (prior knowledge) together with the available observations (likelihood function) make the upwards extension of the reference zone very unlikely.

5.3.3 Convergence Analysis and Parallel Scaling

For computing exact solutions, an empirical reference set of 1 000 000 unconditioned realizations was generated. The corresponding prediction of the 10-days capture zone probability is shown in figure 5.19. We are again interested in the

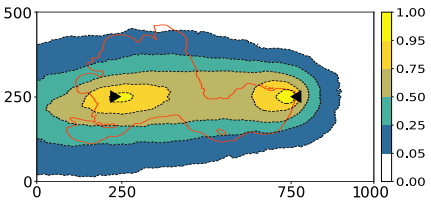


Figure 5.19: Exact prediction of the 10-days capture zone probability. The red line circumscribes the capture zone in the reference domain.

convergence speed of the PoPEx predictions $\hat{\mu}^k_{\alpha,\mathrm{sn}}$ when compared to μ^{ex}. As mentioned earlier, these two maps define Bernoulli probability values for each point in the computational domain. It determines whether the groundwater at the corresponding location belongs to the 10 days capture zone or not. A convenient distance between two Bernoulli probability maps $\hat{\mu}^k_{\alpha,\mathrm{sn}}$ and μ^{ex} is given in equation (5.1). Again, a scalar error value is obtained by considering the spatial average of the divergence map.

Figure 5.20 shows the evolution of error between $\hat{\mu}^k_{\alpha,\mathrm{sn}}$ and μ^{ex} versus the iteration number k. In figure 5.20a the minimum number of effective weights has been fixed to $l_0 = 100$ and we varied the maximum number of conditioning values $n_{\max} \in \{10, 25\}$. It can be seen that the two convergence curves are quite similar. This is not surprising, because the PoPEx algorithm is designed to correct the influence of the hard conditioning by using the weights in equation (4.22). On the other hand, it can be seen from the blue curve that for $n_{\max} = 10$ and $k > 9000$ the error reaches a 'plateau'. This signifies that for a certain time, the PoPEx algorithm is not able to further improve the prediction or in other words, that the method can not find sufficiently important realizations. From such behavior it is deduced that for an efficient learning scheme, the parameter $n_{\max}$ should not be chosen too small. However, what is important is that the overall convergence rate of both curves well compares with the dashed line. This is significant because this line represents the convergence rate that

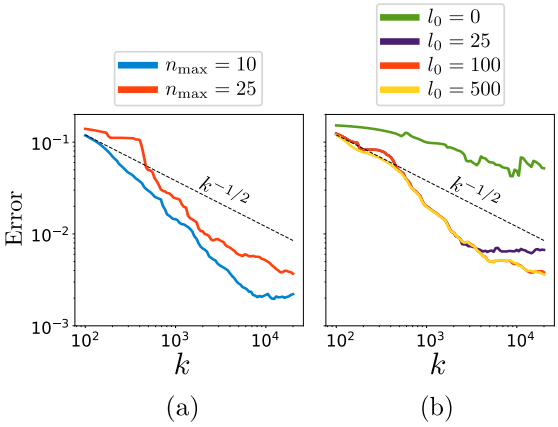


Figure 5.20: Error between $\hat{\mu}^k_{\alpha,\text{sn}}$ and μ^{ex} for a fixed $l_0 = 100$ and variable $n_{\max}$ (a), and fixed $n_{\max} = 25$ and variable l_0 (b).

is predicted by the CLT when directly sampling from the posterior probability distribution. Because the error curves represent the average performance of 5 PoPEx runs, it is reasonable that they slightly fluctuate and do not reproduce the theoretical rate of $k^{-1/2}$ precisely.

For the second experience, we fixed $n_{\max} = 25$ and varied the number $l_0 \in \{0, 25, 100, 500\}$. Recall that the choice of a large value for l_0 generally increases the effective number of weights but risks to produce biased predictions. On the other hand, when the effective number of weights is too low, the predictions will be based on very few models and may be significantly wrong. It is therefore not surprising that for $l_0 = 0$ the approximation accuracy is very bad (green curve in figure 5.20b). However, the remaining three convergence curves are highly similar for $k \leq 4000$ where the magenta curve ($l_0 = 25$) reaches a 'plateau' and has difficulties to further improve the approximations. As in the previous figure, the curves represent the average performance of 5 similar PoPEx runs with different initial seeds. It follows that small fluctuations may arise and should not be evaluated too strictly. However, the stagnation of the curve with $l_0 = 25$ is due to of a different reason. Whenever the parameter l_0 is small, the weights in W^k_α are more sensible to highly dominant values. This means that a model $\mathbf{m}^{k_0}$ with very large weight w_{k_0} dominates the prediction $\hat{\mu}^k_{\alpha,\text{sn}}$ for many iterations $k > k_0$ and therefore, the approximation error only slightly changes. Thus, such a behavior indicates that l_0 should not be chosen too small. We can conclude by highlighting that for a reasonably large l_0, the overall convergence rate compares very well with the theoretical rate of $k^{-1/2}$.

The last part of the results section is dedicated to a short analysis of the parallel scalability of PoPEx. We repeat the same exercise by first using $n_{\text{par}} = 15$ on a 64 CPU's facility ($34.4(Tflop/s)$), and then changing to $n_{\text{par}} = 75$

on 320 CPU's ($172(Tflop/s)$). Therefore, between the first and the second procedure, the computational capacity is increased by a factor of 5. The total runtime for 20 000 models was $27.51 \pm 1.521(h)$ and $5.00 \pm 0.397(h)$, respectively. This signifies an overall speedup factor of 5.5 ± 0.74 and scales perfectly.

We are again interested in the speedup factor for obtaining the same approximation accuracy. This means that in each iteration k, the approximation error is again computed by a Jensen-Shannon divergence between the prediction and the exact solution. In figure 5.21a it can be seen that the convergence rate of both curves are very similar. Figure 5.21b shows the observed speedup

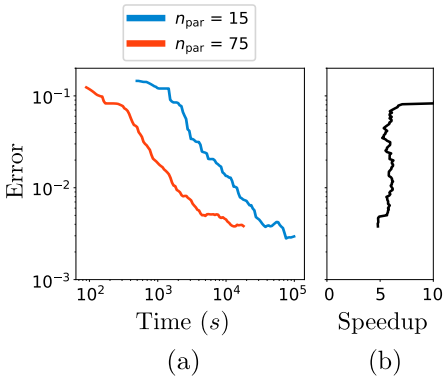


Figure 5.21: Error between $\hat{\mu}^k_{\alpha,\text{sn}}$ and μ^{ex} for a fixed $n_{\max} = 25$, $l_0 = 100$ and variable n_{par} (a), and the speedup factor for obtaining the same error accuracy (b).

in time for obtaining the same approximation accuracy. This means that for any error value (y-axis) we compute the corresponding speedup factor (x-axis) for reaching the considered approximation accuracy. It is evident that the curve significantly matches the predicted speedup factor of 5 and therefore underlines the exceptional scaling behavior of the PoPEx algorithm.

Chapter 6

Discussion and Conclusion

In this work we present, test, and discuss the **Posterior Population Expansion (PoPEx)**; a fast and efficient sampling method for solving inverse problems with complex prior information. Its ergodic correctness is demonstrated analytically by lemma 4.3.3. The algorithm is parallelized and scales perfectly. This means that the number of parallel chains is equal to the time reduction factor, without compromising the quality of the results. PoPEx is incredibly flexible and simple to use.

An important number of inverse schemes are based on the minimization of a misfit function. They are particularly efficient in a setup where the model space and the data spaces are linear, their marginal uncertainty is Gaussian, and they are connected through a linear forward operator [Arulampalam et al., 2002, Mosegaard and Tarantola, 2002]. In the hydrogeological framework however, usually none of these assumptions are satisfied. In order to overcome such restrictions, the usage of non-linear transformations into Gaussian spaces [Zhou et al., 2011, Li et al., 2012, Xu et al., 2013, Xu and Gómez-Hernández, 2015], linear approximations of the forward operator [Chen et al., 2009], and simplified parametrizations from Gaussian subspaces [Doherty, 2003, Alcolea et al., 2006, Tonkin and Doherty, 2009] have been tested. Unfortunately, it is not possible in general to meet all of the three simplifications simultaneously.

The PoPEx algorithm however, has been designed for handling complex prior information which, in the geostatistical environment, is often expressed by randomized simulation tools [Caers et al., 2003, Liu et al., 2004, Okabe and Blunt, 2007, Ronayne et al., 2008]. As the PoPEx method only uses categorical indicator functions, it is independent of the model parameterization. This is especially convenient when working with Jeffrey's parameters that are strictly positive variables with a bijective map to their inverse (e.g. 'period' and 'frequency', 'resistivity' and 'conductivity', 'velocity' and 'slowness', etc.). Moreover, PoPEx is capable to handle all the four different types of uncertainty, distinguished by Sagar et al. [1975]: spatial heterogeneities, initial conditions,

boundary conditions and sources/sinks. The usage of PoPEx effectively requires the implementation of only two functions:

- $\mathbf{m} \leftarrow \text{model}(\mathbf{hd})$, and
- $L(\mathbf{m}) \leftarrow \text{likelihood}(\mathbf{m})$.

The first, 'model($\mathbf{hd}$)', takes a set of synthetic hard conditioning data $\mathbf{hd}$ and must return a member $\mathbf{m}$ of the model space $\mathcal{M}$. This parameter set contains sufficient information for evaluating the second function; the likelihood measure $L(\mathbf{m}) = \text{likelihood}(\mathbf{m})$. Therefore, the model space $\mathcal{M}$, the data space $\mathcal{D}$, the forward operator $\mathbf{g} : \mathcal{M} \rightarrow \mathcal{D}$, and the likelihood measure L are completely detached from the PoPEx sampling scheme and may involve many different techniques and ideas. Overall, this makes the algorithm very dynamic and suitable for a broad range of problems, even beyond the field of geostatistics.

The performance of PoPEx was also demonstrated empirically based on three different examples (cf. sections 5.1-5.3). All of which have been solved very efficiently in reasonable time. It was shown that the predictions converged very fast with convergence speed being comparable to the theoretical rate of $k^{-1/2}$ (predicted by the central limit theorem when samples are drawn independently from the posterior distribution). Furthermore, the accuracy was not compromised by the parallelization of the algorithm. Perfect scaling behavior was demonstrated based on several exercises. In the sections 5.1 and 5.2, the computational capacity was increased by a factor of 2 (from 17.2 to 34.4($Tflop/s$)). After sampling 10 000 models, the estimated speedup was 1.95 ± 0.602 and 2.02 ± 0.509, respectively. An even larger expansion in computational power, namely by a factor of 5 (from 34.4 to 172($Tflop/s$)), was used to sample 20 000 realizations in section 5.3. Also this exercise fully satisfied our expectation by showing an overall speedup factor of 5.5 ± 0.74. Furthermore, we even showed that similar speedup results are obtained when comparing prediction accuracies with respect to an empirical exact solution (cf. figures 5.8, 5.13, and 5.21).

Qualitatively, the above results are very valuable. They are obtained by considering different examples with various problem sizes. Nevertheless there are several aspects that need to be discussed in more detail.

Learning Scheme

It was mentioned that PoPEx is an adaptive importance sampler that implicitly learns and adapts the proposal ϕ_k. During this procedure, the learning scheme faces three main challenges:

- it must be easy to produce random samples according to ϕ_k,
- the tails of ϕ_k must be heavier than the tails of the posterior density σ, and
- ϕ_k should mimic σ.

The first property depends on the conditional simulation tool that is entrained in generating models $\mathbf{m}$. The second is usually satisfied, because it is common that the regions with high posterior measure are very sparse and the sampling density ϕ_k still allows some 'bad' models that lie outside those areas. It suggests that ϕ_k is flatter than σ. The success or failure of the third condition heavily relies on the sets $\widetilde{\Sigma}^k$. They dominate the selection of the indices I and the values $\mathbf{v}$ of $\mathbf{hd}^k$. In the above definition, $\widetilde{\Sigma}^k$ is directly deduced from the likelihood values $L(\mathbf{m}^j)$ and therefore mimics σ quite well. However, the link between L and ϕ_k must not be too strong. Let's assume that for a given $r \geq 1$, the sampling density ϕ_k is approximately proportional to L^r, in which case we have

$$\frac{\sigma}{\phi_k} \propto \frac{\rho}{L^{r-1}}.$$

For a flat prior information ρ and a large model space $\mathcal{M}$, this ratio (and with it the variance of the weights in W^k) might become increasingly large.

The most important drawback of the PoPEx method however, is that the likelihood values must be evaluated and represented by floating-point numbers (cf. equation (4.22)). If the dimension of the data space is very large (i.e. the set $\mathbf{d}^{\text{obs}}$ is huge), it might happen that the likelihood measure is very small for most realizations. Floating point numbers can not represent arbitrarily small values and would be zero almost everywhere. In such cases, the categorical probability distribution P^k is based on very few realizations and the learning process of the method is slow or even inexistent. But if the number of observations is large, it is not uncommon that many of the data points are highly correlated. One idea to reduce the dimension of the data space is to project it on a smaller subspace where the component dependency is minimized. This can be beneficial for preventing an overfit of the observations [Russell and Norvig, 2010, Murphy, 2012]. But doing so requests a careful and intensive analysis of the data set and is far from being trivial.

Alternatively, the likelihood function often defines a *Gibbs field* (or *Gibbs measure*) [Winkler, 2012], i.e.

$$L(\mathbf{m}) = \frac{1}{C} \exp\big(-H(\mathbf{m})\big),$$

where C is a normalization constant and H the energy function. The latter is unique up to an additive constant and in many applications only takes non-negative values. Usually, for floating point operations, it is more convenient to handle energy values $H(\mathbf{m})$ rather than the full Gibbs measure values $\exp\{-H(\mathbf{m})\}$. During the PoPEx algorithm, whenever P^k is computed, an alternative solution is to select $r \geq 1$ and use weights $\tilde{\sigma}_i^k$ such that

$$\tilde{\sigma}_j^k \propto \exp\Big(-\big(H(\mathbf{m}^j)\big)^{\frac{1}{r}}\Big).$$

Similar to the idea in section 4.3.3, this aims to reduce the skewness of the values. However, for the computation of the weights in equation (4.22) (and

therefore for any prediction $\hat{\mu}_{\alpha,\mathrm{sn}}$), it is still requested to compute the unbiased Gibbs measures. However, it can be advantageous to learn from a smoothed measure function $\exp\Big(-\big(H(\mathbf{m})\big)^{\frac{1}{r}}\Big)$ in order to obtain a sufficiently large number of non-zero likelihood values $\exp\big(-H(\mathbf{m}^k)\big)$.

Computation of the Weights

Consider a model $\mathbf{m} = \{m_1, \ldots, m_n\}$ that consists of n parameters. Any joint probability measure function

$$\rho(\mathbf{m}) = \rho(m_1, \ldots, m_n)$$

describes the correlations between the parameters m_i in the manifold $\mathcal{M}$. For a given permutation over the model indices $\varsigma \in S_n$, the density function ρ allows a sequential decomposition such as

$$\rho(\mathbf{m}) \;=\; \rho(m_{\varsigma(1)})\rho(m_{\varsigma(2)} \,|\, m_{\varsigma(1)}) \cdot \ldots \cdot \rho(m_{\varsigma(n)} \,|\, m_{\varsigma(1)}, \ldots, m_{\varsigma(n-1)}).$$

This follows immediately from the multivariate extension of the conditional rule $\rho(s,t) = \rho(s\,|\,t)\rho(t) = \rho(t\,|\,s)\rho(s)$. It can be interpreted as a sequential run through the model parameters for quantifying the measure values based on the previous parameters.

Sequential simulators are often very convenient because they manage to take advantage of such decomposition [Goovaerts, 1997]. In other words, this means that for producing a realization $\mathbf{m}$, they select a random path ς and sequentially provide simulation values to the components of $\mathbf{m}$. In step i, the value $m_{\varsigma(i)}$ is randomly drawn from $\rho(\,\cdot\,|\,m_{\varsigma(1)}, \ldots, m_{\varsigma(i-1)})$ and therefore independent of the unsimulated model parameters $m_{\varsigma(i+1)}, \ldots, m_{\varsigma(n)}$. For a synthetic set of hard conditioning data $\mathbf{hd} = \{hd_1, \ldots, hd_s\}$ we can therefore choose a path ς that meets the conditioning locations in the very beginning. When having a model $\mathbf{m}$ with $\mathbf{m}_I = \mathbf{v}$ it follows that

$$\rho(\mathbf{m}, \mathbf{hd}\,|\,\varsigma) = \rho(\mathbf{hd}\,|\,\varsigma)\rho(\mathbf{m}\,|\,\mathbf{hd}, \varsigma)$$

with

$$\begin{aligned}\rho(\mathbf{hd}\,|\,\varsigma) &= \rho(hd_1) \cdot \ldots \cdot \rho(hd_s \,|\, hd_1, \ldots, hd_{s-1})\\ \rho(\mathbf{m}\,|\,\mathbf{hd}, \varsigma) &= \rho(m_{\varsigma(s+1)}, \ldots, m_{\varsigma(n)}\,|\,\mathbf{hd}).\end{aligned}$$

Therefore, whenever sequential simulators are used, equation (4.19) is suitable. However, it is important to note that not all geostatistical simulation tools work sequentially. For non-sequential simulators, only a careful study can decide whether equation (4.20) is still applicable.

Furthermore, it was assumed that the synthetic hard conditioning data is independent, such that

$$\rho(\mathbf{hd}\,|\,\varsigma) = \prod_{i=1}^{s} \rho(hd_{\varsigma(i)}).$$

For MPS simulations that produce spatial realizations on a computational grid, this assumption is reasonable if the conditioning locations are well separated. It is therefore very important that the maximum number of conditioning points $n_{\max}$ is adequate with respect to the number of model parameters n.

Parallel Scaling

A complete analysis of the parallel behavior must either include a large number of upscaling factors or an analytical examination in the sense of Amdahl's law (cf. equation 1.1). Algorithm 2 encapsulates three out of four computation steps into subprocesses. Considering Amdahl's law, the fraction p of serial workload mainly consists of recomputing $D(P^k||Q)$, what inherently includes the update of $P^k = \{p_1^k, \dots, p_s^k\}$. If n is the number of model parameters, the inherent computational cost can be quantified as follows:

- *Updating P^k:* The values in $\widetilde{\Sigma}^k$ are not normalized one-by-one. In each iteration we only store the normalization constant $\left\|\Sigma^k\right\|_1 = \sum_{j=1}^{k} \sigma_j$ that can be changed to $\left\|\Sigma^{k+1}\right\|_1 = \left\|\Sigma^k\right\|_1 + \sigma_{k+1}$ by using one floating point operation. Furthermore, there is no need to recompute the maps p_i^k from scratch in each iteration. Instead, we have that (cf. equation (4.8))

$$p_i^{k+1} = \frac{p_i^k \left\|\Sigma^k\right\|_1 + \mathbf{1}_{f_i}(\mathbf{m}^{k+1})\sigma_{k+1}}{\left\|\Sigma^{k+1}\right\|_1}.$$

  The evaluation of the categorical indicator functions $\mathbf{1}_{f_i}(\mathbf{m}^{k+1})$ is outsourced to the subprocesses and computed directly after the generation of $\mathbf{m}^{k+1}$. Once the fractions $\frac{\left\|\Sigma^k\right\|_1}{\left\|\Sigma^{k+1}\right\|_1}$ and $\frac{\sigma_{k+1}}{\left\|\Sigma^{k+1}\right\|_1}$ are computed, the update of p_i^k requests $3n$ operations ($2n$ multiplications and n sums) so that the overhead for passing from P^k to P^{k+1} is proportional to $3ns + 3$.

- *Computing $D(P^{k+1}||Q)$:* If the logarithm of a value is obtained by C operations, each term $p_i^{k+1} \log\left(\frac{p_i^{k+1}}{q_i}\right)$ is associated with a effort of $n(C+2)$ actions (n divisions, n evaluations of the logarithm, and n multiplications) so that the total computation of $D(P^{k+1}||Q)$ comes at the cost of $sn(C+2) + n(s-1) = n\big((C+3)s - 1\big)$ operations (s times $n(C+2)$ and $s-1$ sums of n parameters).

The sequential workload p of the algorithm 2 is therefore determined by

$$p \propto n\big(s(C+6) - 1\big) + 3.$$

As mentioned earlier, the forward operation (and therefore the evaluation of the likelihood measure) often involves a numerical solution of partial differential equations. It is known that such computations involve matrix-vector multiplications. If the number of degrees of freedom in the finite element formulation

is proportional to n, they come at least at a cost of n^2. It follows that for applications with complex model spaces (where the number of parameters is large), PoPEx scales perfectly up to a considerable number of parallel chains where $n << \frac{n^2}{n_{\text{par}}}$ or equivalently $n_{\text{par}} << n$.

Commonly used parallelization strategies slightly differ from the one that is implemented in PoPEx. Parallelism is typically organized batch-wise, where a package of n_{par} subprocess are opened and closed simultaneously. Figure 6.1 provides a conceptual overview of such a strategy. In this case, the Kullback-

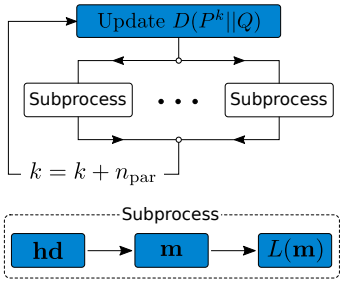


Figure 6.1: Conceptual overview of a batch-wise parallelization.

Leibler divergence is updated batch-wise from $D(P^{kn_{\text{par}}}||Q)$ to $D(P^{(k+1)n_{\text{par}}}||Q)$ after all the n_{par} simulations are terminated. However, this approach risks an inefficient usage of the available CPU resources, because the computational time of the subprocesses may vary (due to large hard conditioning data sets **hd**, unrealistic model parameters, etc.). The PoPEx parallelization aims to fully utilize the computational potential. On the downside of the implemented strategy, we notice that it is not fully reproducible (while the batch-wise organization is). But ensuring the reproducibility by either a batch-wise organization or as mentioned in section 4.2.2, results in a reduced sampling frequency.

What is next...?

...for the study of geostatistical inverse problems, there are still plenty of chapters, papers, dissertations and books to be filled. Among all the future investigations, there might be one that (in some sense) is a continuation of the present work. Today, there are several ideas that might be worth trying:

- *Real world:* The above tests asked for synthetic problems where it was possible to compute exact solutions and empirical convergence curves. A very interesting study could be to use the PoPEx sampling scheme for solving a real world inverse problem. This would face challenges like the generation of geologically realistic models, the assessment of modeling uncertainties, the collection of real data, etc.

- *Large data sets:* It was mentioned that large data sets can entail difficulties. An issue that could be worth to be examined is to measure the impact of different data spaces. The choices of $\mathcal{D}$ and the likelihood measure L (that involves $\mathbf{d}^{\text{obs}}$) might significantly influence the predictive quality of the inverse solution.
- *Model types:* In the above examples, we only considered one or two different model types. But with PoPEx it is possible to simultaneously include many different types of uncertainties, such as initial and boundary conditions, recharge time series, sinks and sources, etc.
- *Comparison:* It was claimed several times that the discrete representations of sharp subsurface structures (e.g. channels, lenses, karst conduits, faults, etc.) is often very important. An extensive comparison between different methods could produce interesting insights into this problematic.

But a potentially very promising idea is described in the following. It aims to take full advantage of the dynamic parallelization scheme and further accelerate the PoPEx method. During a Monte Carlo sampling, the most important computational effort usually goes into the evaluation of the likelihood measure $L(\mathbf{m})$. It is important to note that any model with zero likelihood measure has no effect in the PoPEx learning scheme nor for the evaluation of predictions. In the equations for P^k and $\hat{\mu}_{\alpha,\text{sn}}$, all information associated to $\mathbf{m}^j$ with $L(\mathbf{m}^j) = 0$ is multiplied by 0 and therefore irrelevant.

When considering the PoPEx parallelization illustrated in figure 4.4, we see that any shortcut that avoids some (expensive) evaluations of $L(\mathbf{m})$ directly accelerates the sampling scheme. From a *supervised learning* point of view [Hastie et al., 2009, Murphy, 2012], the two sets $\{\mathbf{m}^1, \ldots, \mathbf{m}^k\}$ and $\{L(\mathbf{m}^1), \ldots, L(\mathbf{m}^k)\}$ can be interpreted as k individuals with n attributes $m_1, \ldots, m_n$ and one response variable $L(\mathbf{m})$ (c.f. table 6.1). Many advanced

Table 6.1: Supervised learning for a set of k realizations.

ID	Features			Response
1	m_1^1	$\cdots$	m_n^1	$L(\mathbf{m}^1)$
2	m_1^2	$\cdots$	m_n^2	$L(\mathbf{m}^2)$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
k	m_1^k	$\cdots$	m_n^k	$L(\mathbf{m}^k)$

classification or regression schemes are currently available and might be trained to learn (abstract) relations between a model and its likelihood measure. Such techniques can be used for each model $\mathbf{m}^{k+1}$ to predict its 'potential' for generating a sufficiently important likelihood value. If the supervised learning scheme predicts very low potential, the algorithm could avoid the computation of the likelihood measure by puting $L(\mathbf{m}^{k+1}) = 0$ and immediately continue

with the next sample. For complex likelihood measures with sparse importance regions, this technique could significantly accelerate the sampling procedure.

"Thank you for taking your time and reading this manuscript.
I personally hope that your future journey will be as exciting
and interesting as it was for me to produce this work."

Appendix A

Conditional Probabilities on Submanifolds

Let's assume that there is probability density h on a n-dimensional manifold $\mathcal{X}$. If there is a parameterization $\mathbf{x} = \{x_1, \ldots, x_n\}$ of $\mathcal{X}$, the probability of any event $A \subset \mathcal{X}$ can be computed by

$$\mathbb{P}(A) = \int_A d\mathbb{P} = \int_A h(\mathbf{x}) d\mathbf{x},$$

where $d\mathbf{x} = dx_1 \cdot \ldots \cdot dx_n$. Let $\mathcal{B} \subset \mathcal{X}$ be a submanifold with zero probability measure, i.e. $\mathbb{P}(\mathcal{B}) = 0$. For many applications it is needed to define 'conditional density functions given $\mathcal{B}$' that are commonly denoted by $h(\mathbf{x} \,|\, \mathcal{B})$. From the field of measure theory it is possible to derive one (intuitive but inconsistent) possible method to do so. This technique considers a sequence of submanifolds $\mathcal{B}_1, \mathcal{B}_2, \ldots, \subset \mathcal{X}$ with $\mathbb{P}(\mathcal{B}) > 0$ that 'converges' to $\mathcal{B}$. Conditional density measures 'knowing $\mathcal{B}_i$' are defined as

$$h_i(\mathbf{x}) = h\big(\mathbf{x} \,|\, \mathcal{B}_i\big) = \frac{h(\mathbf{x})}{\mathbb{P}(\mathcal{B}_i)},$$

and $h(\mathbf{x} \,|\, \mathcal{B})$ is considered to be the limit (in terms of measure function) of the sequence $\big(h_i\big)_{i>0}$. This method however is inconsistent because the sequence of submanifolds $\big(\mathcal{B}_i\big)_{i>0}$ is not unique in general (cf. section A.2).

A consistent technique for obtaining conditional density functions is obtained by the conjunction of two states of information. This method avoids sequences of submanifolds but requires to define appropriate volume measures on $\mathcal{X}$ and $\mathcal{B}$. If the volume of any subset $A \subset \mathcal{X}$ can be measured by

$$V(A) = \int_A dV = \int_A u(\mathbf{x}) d\mathbf{x},$$

then the map u is called a volumetric density function. The **homogeneous probability density**, denoted by μ, is a measure that assigns equal probability value for regions with same volumes. Therefore, in a parameterization $\mathbf{x}$ with volume density $u(\mathbf{x})$, the homogeneous probability density is proportional to u such that

$$\mu(\mathbf{x}) = cu(\mathbf{x}).$$

If the total volume of $\mathcal{X}$ is finite, μ can be defined as $\mu(\mathbf{x}) = u(\mathbf{x})/V(\mathcal{X})$. The physical units, denoted by square brackets $[\cdot]$, are important for distinguishing the different measure functions. A probability value has no unit and therefore

$$[h(\mathbf{x})] = \frac{1}{[d\mathbf{x}]}.$$

Following the same idea for the remaining quantities, we can see that $[u(\mathbf{x})] = \frac{[dV]}{[d\mathbf{x}]}$, $[c] = \frac{1}{[dV]}$, and $[\mu(\mathbf{x})] = \frac{1}{[d\mathbf{x}]}$. Although μ is proportional to u, we can see the difference in their physical units. The measure u quantifies volume ($[dV]$) in a given parameterization $\mathbf{x}$ ($[dx]$) while μ (having similar values as u) measures probability ($[\,]$) by using the same parameterization ($[dx]$).

From a probability density function h, it is possible to define a **volumetric probability density** $H(\mathbf{x})$ by

$$H(\mathbf{x}) = \frac{h(\mathbf{x})}{\mu(\mathbf{x})}.$$

Such densities are *independent* of the parameterization $\mathbf{x}$ or alternatively, independent of physical units because $[H(\mathbf{x})] = \frac{[d\mathbf{x}]}{[d\mathbf{x}]} = [\,]$. For any event $A \subset \mathcal{X}$ the probability measure $\mathbb{P}(A)$ can alternatively be obtained from the volumetric probability density H such as

$$\int_A H(\mathbf{x})\mu(\mathbf{x})d\mathbf{x} = \int_A h(\mathbf{x})d\mathbf{x} = \mathbb{P}(A).$$

The measure $\mu(\mathbf{x})d\mathbf{x}$ has no physical unit and can therefore be considered as a **unitless volume measure** for the parameterization $\mathbf{x}$. In most of the present work, we assumed $d\mathbf{x}$ to not signify the parametric expression $dx_1 \cdot \ldots \cdot dx_n$, but the unitless volumetric measure $\mu(\mathbf{x})dx_1 \cdot \ldots \cdot dx_n$. However, for the remaining part of this section we will consider $d\mathbf{x}$ to signify standard parametric Borel measures.

A.1 Probabilities on Metric Manifolds

We will use Einstein's summation convention which states that "if an index appears twice in a term, it implies summation of that term over all possible values of the index". Let $\mathcal{X}$ be a smooth manifold with coordinate system

$\mathbf{x} = (x_1, \ldots, x_n)$. Let $\mathbf{x} \in \mathcal{X}$ and $g_{\mathbf{x}} : T_{\mathbf{x}}\mathcal{X} \times T_{\mathbf{x}}\mathcal{X} \to \mathbb{R}$ be a Riemannian metric such that

$$g_{\mathbf{x}} = g_{ij}(\mathbf{x}) dx^i \otimes dx^j. \tag{A.1}$$

This expression can be considered as a two-dimensional matrix so that its determinant is equal to the local volume spanned by the tangent vectors dx^i. Therefore, a homogeneous probability measure in a Riemannian manifold is obtained by

$$\mu(\mathbf{x}) = \sqrt{|\det(g_{\mathbf{x}})|}.$$

For a given probability density h on $\mathcal{X}$, the corresponding volumetric measure H is such that

$$H(\mathbf{x}) \propto \frac{h(\mathbf{x})}{\sqrt{|\det(g_{\mathbf{x}})|}}.$$

Let $m \leq n$ and $\mathcal{Y}$ be a manifold with coordinates $\mathbf{y} = (y_1, \ldots, y_m)$ and let $F : \mathcal{Y} \to \mathcal{X}$ be a differentiable map $F(\mathbf{y}) = \big\{F_1(\mathbf{y}), \ldots, F_n(\mathbf{y})\big\}$ such that $F(\mathcal{Y})$ is a smooth submanifold of $\mathcal{X}$. With the formula of *change of parameterization* (e.g. Tu [2010])

$$dx^i = \frac{\partial x^i}{\partial y^j} dy^j, \qquad i = 1, \ldots, n$$

and equation (A.1), the corresponding metric tensor $\tilde{g}_{\mathbf{y}} : T_{\mathbf{y}}\mathcal{Y} \times T_{\mathbf{y}}\mathcal{Y} \to \mathbb{R}$ at $\mathbf{y} \in \mathcal{Y}$ is given by

$$\tilde{g}_{\mathbf{y}} = \tilde{g}_{kr}(\mathbf{y}) dy^k \otimes dy^r,$$

with

$$\tilde{g}_{kr}(\mathbf{y}) = g_{ij}\big(F(\mathbf{y})\big) \frac{\partial x^i}{\partial y^k} \frac{\partial x^j}{\partial y^r}, \qquad k, r = 1, \ldots, m. \tag{A.2}$$

As the transition function F is differentiable, the submanifold $F(\mathcal{Y})$ has dimension m and homogeneous density $\mu_{\mathcal{Y}}(\mathbf{y}) = \sqrt{|\det(\tilde{g}_{\mathbf{y}})|}$. If h is again a probability density on $\mathcal{X}$ and H its volumetric equivalent, the volume measure $d\tilde{V}(F(\mathbf{y}))$ can be used to define a consistent conditional probability density $h\big(\mathbf{x} \,|\, F(\mathcal{Y})\big)$ on the submanifold $F(\mathcal{Y})$. For any $A \subset F(\mathcal{Y})$, the probability measure $\mathbb{P}\big(A \,|\, F(\mathcal{Y})\big)$ is computed as

$$\begin{aligned} \mathbb{P}\big(A \,|\, F(\mathcal{Y})\big) &= k \int_{F^{-1}(A)} H(F(\mathbf{y})) \mu_{\mathcal{Y}}(\mathbf{y}) d\mathbf{y} \\ &= k \int_{F^{-1}(A)} \frac{h\big(F(\mathbf{y})\big)}{\sqrt{|\det(g_{F(\mathbf{y})})|}} \sqrt{|\det(\tilde{g}_{\mathbf{y}})|} d\mathbf{y} \end{aligned}$$

where k is proportionality constants without physical dimension. Therefore, a consistent conditional probability density $h\big(\mathbf{x} \,|\, F(\mathcal{Y})\big)$ on $F(\mathcal{Y})$ (where $\mathbf{x} = F(\mathbf{y})$) is obtained by

$$h\big(\mathbf{x} \,|\, F(\mathcal{Y})\big) \propto h(\mathbf{x}) \sqrt{\frac{|\det(\tilde{g}_{\mathbf{y}})|}{|\det(g_{\mathbf{x}})|}}. \tag{A.3}$$

Using the terminologies above, we observe that the probability density of h defined over the manifold $\mathcal{X}$ and conditioned on the submanifold $F(\mathcal{Y})$ is given by

$$h\big(\mathbf{x} \,|\, F(\mathcal{Y})\big) \;\propto\; h(\mathbf{x}) \frac{\mu_{\mathcal{Y}}\big(\mathbf{y}\big)}{\mu_{\mathcal{X}}(\mathbf{x})}$$

where $\mu_{\mathcal{X}}$ and $\mu_{\mathcal{Y}}$ are the homogeneous probability densities of $\mathcal{X}$ and $\mathcal{Y}$, respectively.

Example A.1.1 (Constant probability density)**.** *Let* $\mathcal{X} = \mathbb{R}^2$, $\mathcal{Y} = \mathbb{R}$, $F(y) = \begin{pmatrix} y \\ y^2 \end{pmatrix}$, *and* $g(\mathbf{x}) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$. *With* $\frac{\partial \mathbf{x}}{\partial y} = \begin{pmatrix} 1 \\ 2y \end{pmatrix}$ *it can be seen that*

$$\begin{aligned} \tilde{g}(y) &= \begin{pmatrix} 1 & 2y \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 2y \end{pmatrix} = \big(1 + 4y^2\big) \\ \mu_{\mathcal{X}}(\mathbf{x}) d\mathbf{x} &= dx^1 dx^2 \\ \mu_{\mathcal{Y}}(\mathbf{y}) d\mathbf{y} &= \sqrt{1 + 4y^2}\, dy. \end{aligned}$$

The conditioned probability $h\big(\mathbf{x} \,|\, F(\mathbb{R})\big)$ *of an uniform measure* $h(\mathbf{x}) = c$ *over* $\mathbb{R}$ *is such that*

$$h\big(\mathbf{x} \,|\, F(\mathbb{R})\big) \propto c\sqrt{1 + 4y^2}.$$

Note that the volumetric equivalent of $h\big(\mathbf{x} \,|\, F(\mathbb{R})\big)$ *in* $\mathbb{R}$ *reads*

$$H\big(\mathbf{x} \,|\, F(\mathbb{R})\big) = \frac{h\big(\mathbf{x} \,|\, F(\mathbb{R})\big)}{\sqrt{1 + 4y^2}} \propto c,$$

and is constant as well.

Example A.1.2 (Volume measure on coordinates)**.** *Let* $1 \leq t < n$ *and* $\mathbf{y} = (x_1, \ldots, x_t)$ *be the first* t *coordinates of* $\mathbf{x}$. *For a given point* $\mathbf{x} = F(\mathbf{y}) = (\mathbf{y}, f_1(\mathbf{y}), \ldots, f_{n-t}(\mathbf{y}))$ *and* $D = \frac{\partial f}{\partial \mathbf{y}}$ *we have that*

$$\frac{\partial \mathbf{x}}{\partial \mathbf{y}} = \begin{pmatrix} I \\ D \end{pmatrix}.$$

The metric g *can be subdivided into blocks such that*

$$g = \begin{pmatrix} g_{yy} & g_{yf} \\ g_{fy} & g_{ff} \end{pmatrix}$$

and therefore

$$\tilde{g} = \begin{pmatrix} I \\ D \end{pmatrix}^{\mathrm{T}} \begin{pmatrix} g_{yy} & g_{yf} \\ g_{fy} & g_{ff} \end{pmatrix} \begin{pmatrix} I \\ D \end{pmatrix} = g_{yy} + D^{\mathrm{T}} g_{fy} + g_{yf} D + D^{\mathrm{T}} g_{ff} D$$

defines the metric on the submanifold $\mathbf{x} = F(\mathbf{y})$.

Example A.1.3 (Separable manifold)**.** *Let $\mathcal{X}$ be separable such that $\mathcal{X} = U \times V$ with coordinate system $\mathbf{x} = \{\mathbf{u}, \mathbf{v}\}$. The conditional probability when knowing that $\mathbf{v} = \mathbf{v}_0$ is fixed, reads*

$$\frac{\partial \mathbf{x}}{\partial \mathbf{y}} = \begin{pmatrix} I \\ 0 \end{pmatrix},$$

and by the previous example

$$\tilde{g} = g_{uu}.$$

Let us also assume that the homogeneous measure $\mu_{\mathcal{X}}$ is separable such that

$$\mu_{\mathcal{X}}(\mathbf{u}, \mathbf{v}) = \mu_U(\mathbf{u})\mu_V(\mathbf{v}).$$

Therefore, the conditional probability density of h reads

$$h\big((\mathbf{u}, \mathbf{v}) \,|\, \mathbf{v} = \mathbf{v}_0\big) \propto h(\mathbf{u}, \mathbf{v}_0) \frac{\sqrt{|\det(\tilde{g}_{uu})|}}{\sqrt{|\det(g_{\mathbf{x}})|}} \propto \frac{h(\mathbf{u}, \mathbf{v}_0)}{\mu_V(\mathbf{v})}.$$

A.2 Borel's Paradox

> "As any other 'mathematical paradox', the 'Borel paradox' is not a paradox, it is just the result of inconsistent calculation, with an arbitrary definition of conditional probability density." *Mosegaard and Tarantola (2002)*

This part does not provide new material and almost fully follows the ideas of Mosegaard and Tarantola [2002]. However, Borel's paradox is very instructional for illustrating possible issues when working with conditional probability densities that are not volumetric. Let us consider an uniform volumetric probability density H over the sphere $\mathbb{S}^2$. By definition, a *great circle* is the intersection of $\mathbb{S}^2$ with any plane passing through its center. If the density H is constant over the sphere, it is clear that the conditional probability density over any great circle must be uniform too. Let us assume that the spherical coordinates $\{\rho, \vartheta\}$ are used. In our setup, $0 \leq \rho < 2\pi$ denotes the *longitude* and $-\frac{\pi}{2} \leq \vartheta \leq \frac{\pi}{2}$ is the *latitude*. As H is uniform, we know that the probability in $\mathbb{S}^2$ is measured by $H(\rho, \vartheta) = \frac{1}{4\pi}$ such that $d\mathbb{P} = \frac{1}{4\pi}\mu(\rho, \vartheta)d\rho d\vartheta$. The metric tensor on the spherical coordinates can be deduces by a change of coordinates

$$\begin{aligned} x &= \cos(\vartheta)\cos(\rho) \\ y &= \cos(\vartheta)\sin(\rho) \\ z &= \sin(\vartheta) \end{aligned}$$

such that

$$g_{\rho,\vartheta} = M^{\mathrm{T}} I_3 M$$

with

$$M = \begin{pmatrix} -\cos(\vartheta)\sin(\rho) & -\sin(\vartheta)\cos(\rho) \\ \cos(\vartheta)\cos(\rho) & -\sin(\vartheta)\sin(\rho) \\ 0 & \cos(\vartheta) \end{pmatrix} \qquad \text{and} \qquad I_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

so that

$$g_{\rho,\vartheta} = \begin{pmatrix} \cos^2(\vartheta) & 0 \\ 0 & 1 \end{pmatrix}. \tag{A.4}$$

It follows that in the spherical coordinate system, the probability of an event A can be computed by

$$\mathbb{P}(A) = \int_A \frac{1}{4\pi} |\cos(\vartheta)| d\rho d\vartheta,$$

and the uniform probability distribution over the surface of a sphere reads

$$h(\rho, \vartheta) = \frac{1}{4\pi} |\cos(\vartheta)|. \tag{A.5}$$

We observe that in this coordinate system, the density function is no longer constant. This is in the very heart of Borel's paradox, that we will discuss in the next section.

A.2.1 Inconsistent Approach

The result referred by 'Borel's paradox' is obtained because of defining conditional probability densities as an arbitrary limit. We will point out that in this case, the analytical expression of the conditional density depends on the limit what means that the result is not unique in general. Mosegaard and Tarantola [2002] claim that usual parameter spaces accept a natural definition of distance and that any 'limiting operation' must be considered with respect to an *uniform convergence* associated with this measure.

Let us define the conditional probability $h\big(\rho \,|\, \vartheta = 0\big)$ as a limit of $h(\rho, -\epsilon < \vartheta < \epsilon)$ with ϵ going to 0, i.e.

$$\begin{aligned} h\big(\rho \,|\, \vartheta = 0\big) &= \lim_{\epsilon \to 0} \frac{\mathbb{P}(\rho, -\epsilon < \vartheta < \epsilon)}{\mathbb{P}(-\epsilon < \vartheta < \epsilon)} \\ &= \lim_{\epsilon \to 0} \frac{\int_{-\epsilon}^{\epsilon} h(\rho, \vartheta) d\vartheta}{\int_0^{2\pi} \int_{-\epsilon}^{\epsilon} h(\rho, \vartheta) d\vartheta d\rho} = \frac{1}{2\pi}. \end{aligned} \tag{A.6}$$

As expected, this density function is uniform on the great circle $\vartheta = 0$. But if we use the same strategy to compute the conditional probability density knowing that $\rho = 0$, we obtain

$$\begin{aligned} h\big(\vartheta \,|\, \rho = 0\big) &= \lim_{\epsilon \to 0} \frac{\mathbb{P}(-\epsilon < \rho < \epsilon, \vartheta)}{\mathbb{P}(-\epsilon < \rho < \epsilon)} \\ &= \lim_{\epsilon \to 0} \frac{\int_{-\epsilon}^{\epsilon} h(\rho, \vartheta) d\rho}{\int_{-\pi/2}^{\pi/2} \int_{-\epsilon}^{\epsilon} h(\rho, \vartheta) d\rho d\vartheta} = \frac{1}{2} |\cos(\vartheta)|. \end{aligned} \tag{A.7}$$

which is clearly not uniform. This is a contradiction because the original density is uniform over the entire sphere and therefore should also be constant when conditioned onto any subdomain of $\mathbb{S}^2$. Note that if we define a slightly different parametrization such that $0 \leq \rho < \pi$ and $-\pi \leq \vartheta < \pi$, the conditional probability in equation (A.6) and (A.7) would change into

$$h\big(\rho \,|\, \vartheta = 0\big) = \frac{1}{\pi}$$
$$h\big(\vartheta \,|\, \rho = 0\big) = \frac{1}{4}|\cos(\vartheta)|$$

what also results in a paradox.

A.2.2 Consistent Approach

Let $\mathbf{x} = (\rho, \vartheta)$, $y = \rho$ and $F(\rho) = (\rho, 0)$. From equations (A.2) and (A.4) the metric tensor on the submanifold reads

$$\tilde{g}_y = \begin{pmatrix} 1 & 0 \end{pmatrix} g_{\rho,\vartheta=0} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = 1.$$

Together with equations (A.3) and (A.5) we deduce that

$$h\big(\rho \,|\, \vartheta = 0\big) \propto h(\rho, 0) \frac{1}{|cos(0)|} \propto \frac{1}{4\pi}.$$

and similarly,

$$h\big(\vartheta \,|\, \rho = 0\big) \propto h(0, \vartheta) \frac{1}{|cos(\vartheta)|} = \frac{1}{4\pi}.$$

Both conditional densities are uniform. No paradox appears. No matter if the great circle is a meridian or the equator. This is due to the fact that in this second ('consistent') approach, we used the conjunction of two *volumetric* probability densities as in equation (2.4).

Appendix B

Symbols and Abbreviations

Symbols

$\mathrm{Cov}(\cdot\,,\cdot)$	Covariance operator
$\mathcal{D}$	Data space
$D\big(P^k \| \| Q\big)$	Kullback-Leibler divergence; $D\big(P^k \| \| Q\big) : \{1, \ldots, n\} \to [0, +\infty)$
$\mathbf{d}$	Synthetic observations; $\mathbf{d} = \big\{d_1, \ldots, d_r\big\} \in \mathcal{D}$
$\mathbf{d}^{\mathrm{obs}}$	Measured observations; $\mathbf{d}^{\mathrm{obs}} = \big\{d_1^{\mathrm{obs}}, \ldots, d_r^{\mathrm{obs}}\big\} \in \mathcal{D}$
$\mathbb{E}(\cdot)$	Expectation operator
$\mathcal{F}$	Partition of the image of $\mathbf{M}$; $\mathcal{F} = \big\{f_1, \ldots, f_s\big\}$
f_i	Categories of the image of $\mathbf{M}$
$f\big(\mathbf{m}\big)$	Measure of interest for $\mathbf{m} \in \mathcal{M}$
$\mathbf{g}$	Forward operator; $\mathbf{g} : \mathcal{M} \to \mathcal{D}$
$\mathbf{hd}$	Hard conditioning data; $\mathbf{hd} = \big(I, \mathbf{v}\big)$
$\mathbf{hd}^k$	Synthetic hard conditioning data at iteration k
I	Set of hard conditioning indices
$L\big(\mathbf{m}\big)$	Likelihood measure of $\mathbf{m} \in \mathcal{M}$
l_0	Lower bound for the number of effective weights
$\mathcal{M}$	Model space
$\mathcal{M}^k$	Subset of models; $\mathcal{M}^k = \big\{\mathbf{m}^1, \ldots, \mathbf{m}^k\big\}$
$\mathbf{M}$	Random vector; $\mathbf{M} : \Omega \to \mathcal{M}$
$\mathbf{m}$	Set of model parameters; $\mathbf{m} = \big\{m_1, \ldots, m_n\big\} \in \mathcal{M}$
$\mathbf{m}^k$	The k-th model in a PoPEx chain
μ	Exact posterior prediction

$\hat{\mu}$	Estimator for μ
$\hat{\mu}_{\mathrm{sn}}$	Self-normalized estimator for μ
$\hat{\mu}_{\alpha,\mathrm{sn}}$	Self-normalized PoPEx estimator for μ
N	Length of the PoPEx chain
$n_e(\cdot)$	Kish's effective sample size
n_k	Number of synthetic hard conditioning data at iteration k
$n_{\max}$	Maximum number of synthetic hard conditioning data
n_{par}	Number of parallel subprocesses
$\nu(\mathbf{d})$	Observational measure of $\mathbf{d} \in \mathcal{D}$
Ω	Abstract probability space
$\mathbb{P}$	Probability measure
$p \wedge q$	Conjunction of two states of information p and q
P^k	Categorical 'posterior' distribution; $P^k = \{p_1^k, \ldots, p_s^k\}$
p_i^k	Categorical 'posterior' density; $p_i^k : \{1, \ldots, n\} \to [0,1]$
$\phi_k(\mathbf{m})$	Adaptive proposal measure of $\mathbf{m} \in \mathcal{M}$
Q	Categorical prior distribution; $Q = \{q_1, \ldots, q_s\}$
q_i	Categorical prior probability map; $q_i : \{1, \ldots, n\} \to [0,1]$
$\rho(\mathbf{m})$	Prior information of $\mathbf{m} \in \mathcal{M}$
Σ^k	Set of learning weights; $\Sigma^k = \{\sigma_1^k, \ldots, \sigma_k^k\}$
$\sigma(\mathbf{m})$	Posterior information; $\sigma(\mathbf{m}) = cL(\mathbf{m})\rho(\mathbf{m})$
$\tau_k(\mathbf{hd})$	Measure of selecting $\mathbf{hd}$ at iteration k
$\theta(\mathbf{d} \mid \mathbf{m})$	Forward relation measure of $\mathbf{d} \in \mathcal{D}$ for a given $\mathbf{m} \in \mathcal{M}$
$\mathrm{Var}(\cdot)$	Variance operator
$\mathbf{v}$	Set of hard conditioning values
W^N	Set of adaptive importance sampling weights; $W^N = \{w_1, \ldots, w_N\}$
W_α^N	Power set of adaptive importance sampling weights; $W_\alpha^N = \{(w_1)^\alpha, \ldots, (w_N)^\alpha\}$
w_k	Adaptive IS weight; $w_k = \sigma(\mathbf{m}^k) \,/\, \phi_k(\mathbf{m}^k)$
$\mathbf{1}_{f_i}$	Categorical indicator function; $\mathbf{1}_{f_i} : \mathcal{M} \to \{0,1\}^n$

Abbreviations

AIS	Adaptive importance sampling
CLT	Central limit theorem
CPU	Central processing unit
DS	Direct sampling
EnKF	Ensemble Kalman filter
FIFO	First in first out
Flop	Floating point operations
IMcMC	Interrupted Markov chain Monte Carlo
IQR	Interquartile range
IS	Importance sampling
ISR	Iterative spatial resampling
JSD	Jensen-Shannon divergence
KLD	Kullback-Leibler divergence
LLN	Law of large numbers
MAP	Maximum a-posteriori
MC	Monte Carlo
ML	Maximum likelihood
MPS	Multiple-point statistics
MSE	Mean squared error
PoPEx	Posterior Population Expansion
RMSE	Root mean squared error
TI	Training image

Bibliography

Andres Alcolea and Philippe Renard. Blocking Moving Window algorithm: Conditioning multiple-point simulations to hydrogeological data. *Water Resources Research*, 46(8), 2010. ISSN 1944-7973. W08511.

Andres Alcolea, Jesus Carrera, and Agustin Medina. Inversion of heterogeneous parabolic-type equations using the pilot points method. *International Journal for Numerical Methods in Fluids*, 51(9-10):963–980, 2006. ISSN 1097-0363.

G. Burc Arpat and Jef Caers. Conditional Simulation with Patterns. *Mathematical Geology*, 39(2):177–203, 2007. ISSN 1573-8868.

M. Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp. A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking. *Transactions on Signal Processing*, 50(2):174–188, 2002. ISSN 1053-587X.

R. Bailey and D. Baù. Ensemble smoother assimilation of hydraulic head and return flow data to estimate hydraulic conductivity distribution. *Water Resources Research*, 46(12), 2010.

Gerrit Burgers, Peter Jan van Leeuwen, and Geir Evensen. Analysis Scheme in the Ensemble Kalman Filter. *Monthly Weather Review*, 126(6):1719–1724, 1998.

Jef Caers. *Modeling Uncertainty in the Earth Sciences*. Wiley, 2011. ISBN 9781119995937.

Jef Caers, Sebastien Strebelle, and Karen Payrazyan. Stochastic integration of seismic data and geologic scenarios: a West Africa submarine channel saga. *The Leading Edge*, 22(3):192–196, 2003.

Jesús Carrera, Andres Alcolea, Agustín Medina, Juan Hidalgo, and Luit Slooten. Inverse problem in hydrogeology. *Hydrogeology Journal*, 13:206–222, 03 2005.

Guillaume Caumon. *Geological Objects and Physical Parameter Fields in the Subsurface: A Review*, pages 567–588. Springer International Publishing, Cham, 2018.

Yan Chen and Dongxiao Zhang. Data assimilation for transient flow in geologic formations via ensemble Kalman filter. *Advances in Water Resources*, 29(8): 1107–1122, 2006.

Yan Chen, Dean S. Oliver, and Dongxiao Zhang. Data assimilation for nonlinear problems by ensemble Kalman filter with reparameterization. *Journal of Petroleum Science and Engineering*, 66(1–2):1–14, 2009. ISSN 0920-4105.

Nicolas Cherpeau, Guillaume Caumon, Jef Caers, and Bruno Lévy. Method for stochastic inverse modeling of fault geometry and connectivity using flow data. *Mathematical Geosciences*, 44(2):147–168, 2012.

Jean-Paul Chilès and Pierre Delfiner. *Geostatistics: Modeling Spatial Uncertainty*, volume 497. John Wiley & Sons, 2009.

G. de Marsily, J. P. Delhomme, A. Coudrain-Ribstein, and A. M. Lavenue. Four decades of inverse problems in hydrogeology. *Geological Society of America Special Papers*, 348:1–17, 2000.

John Doherty. Ground Water Model Calibration Using Pilot Points and Regularization. *Ground Water*, 41(2):170–177, 2003. ISSN 1745-6584.

Arnaud Doucet, Nando de Freitas, and Neil Gordon. *Sequential Monte Carlo Methods in Practice*. Springer, New York, USA, 2001.

Rick Durrett. *Probability: Theory and Examples*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2010. ISBN 9781139491136.

Geir Evensen. The Ensemble Kalman Filter: theoretical formulation and practical implementation. *Ocean Dynamics*, 53(4):343–367, 2003.

Geir Evensen. *Data Assimilation: The Ensemble Kalman Filter*. Springer-Verlag New Yord, Inc., Secaucus, NJ, USA, 2006.

Luc Feyen and Jef Caers. Quantifying geological uncertainty for flow and transport modeling in multi-modal heterogeneous formations. *Advances in Water Resources*, 29(6):912–929, 2006.

Jianlin Fu and Jaime Gómez-Hernández. Preserving spatial structure for inverse stochastic simulation using blocking Markov chain Monte Carlo method. *Inverse Problems in Science and Engineering*, 16(7):865–884, 2008.

Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional Computing Series. Pearson Education, 1994.

J. Jaime Gómez-Hernández and Xian-Huan Wen. To be or not to be multi-Gaussian? A reflection on stochastic hydrogeology. *Advances in Water Resources*, 21(1):47–61, 1998.

Pierre Goovaerts. *Geostatistics for Natural Resources Evaluation*. Applied geostatistics series. Oxford University Press, 1997.

Yaqing Gu and Dean S. Oliver. An iterative ensemble Kalman filter for multiphase fluid data assimilation. *Comput. Geosci.*, 12(4):438–446, 2007.

Felipe Guardiano and R. Mohan Srivastava. *Multivariate Geostatistics: Beyond Bivariate Moments*, pages 133–144. Springer Netherlands, Dordrecht, 1993.

Thomas Mejer Hansen, Klaus Mosegaard, and Knud Skou Cordua. Using geostatistics to describe complex a priori information for inverse problems. In *8th International Geostatistics Congress*, pages 329–338. Gecamin, 2008.

Thomas Mejer Hansen, Knud Skou Cordua, and Klaus Mosegaard. Inverse problems with non-trivial priors: efficient solution through sequential Gibbs sampling. *Computational Geosciences*, 16(3):593–611, Jun 2012.

Trevor Hastie, Robert Tibshirani, and Jerome H. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer series in statistics. Springer, 2009.

Christian Heimes et al. python-multiprocessing, 2008. URL `https://pypi.org/project/multiprocessing/`.

Maurice Herlihy and Nir Shavit. *The Art of Multiprocessor Programming*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2008.

Timothy Classen Hesterberg. *Advances in Importance Sampling*. PhD thesis, Stanford University, California, 2003.

Mehrdad Honarkhah and Jef Caers. Stochastic Simulation of Patterns Using Distance-Based Pattern Modeling. *Mathematical Geosciences*, 42(5):487–517, 2010.

Christoph Jäggli, Julien Straubhaar, and Philippe Renard. Posterior population expansion for solving inverse problems. *Water Resources Research*, 53 (4):2902–2916, 2017. ISSN 1944-7973.

Christoph Jäggli, Julien Straubhaar, and Philippe Renard. Parallelized Adaptive Importance Sampling for Solving Inverse Problems. *Frontiers in Earth Science*, 6:203, 2018.

Andre Journel and Tuanfeng Zhang. The Necessity of a Multiple-Point Prior Model. *Mathematical Geology*, 38(5):591–610, Jul 2006.

Jaouher Kerrou, Philippe Renard, Harrie-Jan Hendricks Franssen, and Ivan Lunati. Issues in characterizing heterogeneity and connectivity in non-multigaussian media. *Advances in Water Resources*, 31(1):147–159, 2008.

Andreas Kirsch. *An Introduction to the Mathematical Theory of Inverse Problems*. Springer-Verlag, Berlin, Heidelberg, 1996.

Leslie Kish. *Survey sampling*. Wiley Classics Library. Wiley, 1965.

Peter K. Kitanidis. *On Stochastic Inverse Modeling*, pages 19–30. American Geophysical Union (AGU), 2013.

Solomon Kullback and Richard A. Leibler. On Information and Sufficiency. *Ann. Math. Statist.*, 22(1):79–86, 03 1951.

Eric Laloy, Niklas Linde, Diederik Jacques, and Grégoire Mariethoz. Merging parallel tempering with sequential geostatistical resampling for improved posterior exploration of high-dimensional subsurface categorical fields. *Advances in Water Resources*, 90:57–69, 2016.

Eric Laloy, Romain Hérault, Diederik Jacques, and Niklas Linde. Training-Image Based Geostatistical Inversion Using a Spatial Generative Adversarial Neural Network. *Water Resources Research*, 54(1):381–406, 2018.

John M. Lee. *Introduction to Riemannian Manifolds*. Graduate Texts in Mathematics. Springer International Publishing, 2019. ISBN 9783319917542.

Matthijs Leeuwen, Chris B M te Stroet, Adrian P Butler, and Jacob A Tompkins. Stochastic determination of well capture zones. *Water Resources Research*, 34(9):2215–2223, 1998.

L. Li, H. Zhou, H. J. Hendricks Franssen, and J. J. Gómez-Hernández. Groundwater flow inverse modeling in non-MultiGaussian media: performance assessment of the normal-score Ensemble Kalman Filter. *Hydrology and Earth System Sciences*, 16(2):573–590, 2012.

Liangping Li, Haiyan Zhou, J Jaime Gómez-Hernández, and Sanjay Srinivasan. A Comparison of EnKF and EnPAT Inverse Methods: Non-Gaussianity. In *Geostatistics Valencia 2016*, pages 837–841. Springer, 2017.

Jianhua Lin. Divergence Measures Based on the Shannon Entropy. *IEEE Trans. Inf. Theor.*, 37(1):145–151, September 2006.

Niklas Linde, Philippe Renard, Tapan Mukerji, and Jef Caers. Geological realism in hydrogeological and geophysical inverse modeling: A review. *Advances in Water Resources*, 86:86–101, 2015.

Jun S. Liu. *Monte Carlo Strategies in Scientific Computing*. Springer Publishing Company, Incorporated, 2008.

Yuhong Liu, Andrew Harding, William Abriel, and Sebastien Strebelle. Multiple-point simulation integrating wells, three-dimensional seismic data, and geology. *AAPG bulletin*, 88(7):905–921, 2004.

Simon Lopez, Isabelle Cojan, Jacques Rivoirard, and Alain Galli. *Process-Based Stochastic Modelling: Meandering Channelized Reservoirs*, pages 139–144. Wiley-Blackwell, 2009.

Gregoire Mariethoz and Jef Caers. *Front Matter*. Wiley-Blackwell, 2014.

Grégoire Mariethoz, Philippe Renard, and Jef Caers. Bayesian inverse problem and optimization with iterative spatial resampling. *Water Resources Research*, 46(W11530), 2010a.

Grégoire Mariethoz, Philippe Renard, and Julien Straubhaar. The Direct Sampling method to perform multiple-point geostatistical simulations. *Water Resources Research*, 46(W11536), 2010b.

Robert C. Martin. *Clean Code: A Handbook of Agile Software Craftsmanship*. Robert C. Martin Series. Pearson Education, 2008.

Dennis McLaughlin and Lloyd R. Townley. A Reassessment of the Groundwater Inverse Problem. *Water Resources Research*, 32(5):1131–1161, 1996.

Carmen G. Moles, Pedro Mendes, and Julio R. Banga. Parameter estimation in biochemical pathways: a comparison of global optimization methods. *Genome research*, 13(11):2467–2474, 2003.

Klaus Mosegaard. Resolution analysis of general inverse problems through inverse Monte Carlo sampling. *Inverse Problems*, 14(3):405, 1998.

Klaus Mosegaard and Malcolm Sambridge. Monte Carlo analysis of inverse problems. *Inverse Problems*, 18(3):R29, 2002.

Klaus Mosegaard and Albert Tarantola. 16 - probabilistic approach to inverse problems. In Paul C. Jennings William H.K. Lee, Hiroo Kanamori and Carl Kisslinger, editors, *International Handbook of Earthquake and Engineering Seismology, Part A*, volume 81, Part A of *International Geophysics*, pages 237 – 265. Academic Press, 2002.

Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. Taylor & Francis, Cambridge, MA, 2012.

J.C. Naylor and A.F.M. Smith. Econometric illustrations of novel numerical integration strategies for Bayesian inference. *Journal of Econometrics*, 38 (1):103 – 125, 1988.

Man-Suk Oh and James O. Berger. Adaptive importance sampling in Monte Carlo integration. *Journal of Statistical Computation and Simulation*, 41: 143–168, 07 1992.

Hiroshi Okabe and Martin J. Blunt. Pore space reconstruction of vuggy carbonates using microtomography and multiple-point statistics. *Water Resources Research*, 43(12), 2007.

Travis E. Oliphant. *Guide to NumPy*. CreateSpace Independent Publishing Platform, USA, 2nd edition, 2015.

Dean S. Oliver, Luciane B. Cunha, and Albert C. Reynolds. Markov chain Monte Carlo methods for conditioning a permeability field to pressure data. *Mathematical Geology*, 29(1):61–91, 1997.

Henning Omre and Håkon Tjelmeland. Petroleum geostatistics. In *Scho (eds), Geostatistics Wollongong '96*, pages 41–52. Kluwer Academic Publishers, 1996.

Art B. Owen. *Monte Carlo theory, methods and examples*. 2013.

Victor M. Panaretos. *Statistics for Mathematicians: A Rigorous First Course*. Compact Textbooks in Mathematics. Birkhäuser, 2016.

Michael J. Pyrcz and Clayton V. Deutsch. *Geostatistical Reservoir Modeling*. OUP USA, 2014.

Luciano Ramalho. *Fluent Python: Clear, Concise, and Effective Programming*. O'Reilly Media, 2015.

Mickaële Le Ravalec-Dupin and Benoît Nœtinger. Optimization with the Gradual Deformation Method. *Mathematical Geology*, 34(2):125–142, Feb 2002.

Christian P. Robert and George Casella. *Monte Carlo Statistical Methods*. Springer Texts in Statistics. Springer-Verlag New York, 2004. ISBN 978-0-387-21239-5.

R. Robert Johansson. *Numerical Python: A Practical Techniques Approach for Industry*. Apress, 2015.

Thomas Romary. History matching of approximated lithofacies models under uncertainty. *Computational Geosciences*, 14(2):343–355, Mar 2010.

Michael J. Ronayne, Steven M. Gorelick, and Jef Caers. Identifying discrete geologic structures that produce anomalous hydraulic response: An inverse modeling approach. *Water resources research*, 44(8), 2008.

Reuven Y. Rubinstein and Dirk P. Kroese. *Simulation and the Monte Carlo Method*. Wiley Series in Probability and Statistics. Wiley, 3 edition, 2016.

Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall series in artificial intelligence. Prentice Hall, 2010.

Budhi Sagar, Sidney Yakowitz, and Lucien Duckstein. A direct method for the identification of the parameters of dynamic nonhomogeneous aquifers. *Water Resources Research*, 11(4):563–570, 1975.

Malcolm Sambridge and Klaus Mosegaard. Monte Carlo Methods in Geophysical Inverse Problems. *Reviews of Geophysics*, 40(3), 2002.

Julien Straubhaar. DeeSse Technical Reference Guide. Technical Report, Centre d'hydrogéologie et géothermie, University of Neuchâtel, Neuchâtel, 2011.

Julien Straubhaar, Philippe Renard, Grégoire Mariethoz, Roland Froidevaux, and Olivier Besson. An Improved Parallel Multiple-Point Algorithm Using a List Approach. *Mathematical Geosciences*, 43(3):305–328, 2011.

Julien Straubhaar, Alexandre Walgenwitz, and Philippe Renard. Parallel Multiple-Point Statistics Algorithm Based on List and Tree Structures. *Mathematical Geosciences*, 45(2):131–147, Feb 2013.

Sebastien Strebelle. Conditional Simulation of Complex Geological Structures Using Multiple-Point Statistics. *Mathematical Geology*, 34(1):1–21, 2002.

Albert Tarantola. *Inverse Problem Theory and Methods for Model Parameter Estimation*. Society for Industrial and Applied Mathematics, Philadelphia, 2005.

Matthew Tonkin and John Doherty. Calibration-constrained monte carlo analysis of highly parameterized models using subspace techniques. *Water Resources Research*, 45(12), 2009. ISSN 1944-7973.

Loring W. Tu. *An Introduction to Manifolds*. Universitext. Springer New York, 2010.

John Wainwright and Mark Mulligan. *Environmental modelling: finding simplicity in complexity*. John Wiley & Sons, 2005.

Gerhard Winkler. *Image Analysis, Random Fields and Markov Chain Monte Carlo Methods: A Mathematical Introduction*. Stochastic Modelling and Applied Probability. Springer Berlin Heidelberg, 2012. ISBN 9783642557606.

Teng Xu and J. Jaime Gómez-Hernández. Inverse sequential simulation: A new approach for the characterization of hydraulic conductivities demonstrated on a non-gaussian field. *Water Resources Research*, 51(4):2227–2242, 2015.

Teng Xu, J. Jaime Gómez-Hernández, Haiyan Zhou, and Liangping Li. The power of transient piezometric head data in inverse modeling: An application of the localized normal-score EnKF with covariance inflation in a heterogenous bimodal hydraulic conductivity field. *Advances in Water Resources*, 54: 100–118, 2013.

Haiyan Zhou, J. Jaime Gómez-Hernández, Harrie-Jan Hendricks Franssen, and Liangping Li. An approach to handling non-Gaussianity of parameters and state variables in ensemble Kalman filtering. *Advances in Water Resources*, 34(7):844–864, 2011.

Haiyan Zhou, J. Jaime Gómez-Hernández, and Liangping Li. Inverse methods in hydrogeology: Evolution and recent trends. *Advances in Water Resources*, 63:22–37, 2014.

D. A. Zimmerman, G. de Marsily, C. A. Gotway, M. G. Marietta, C. L. Axness, R. L. Beauheim, R. L. Bras, J. Carrera, G. Dagan, P. B. Davies, D. P. Gallegos, A. Galli, J. Gómez-Hernández, P. Grindrod, A. L. Gutjahr, P. K. Kitanidis, A. M. Lavenue, D. McLaughlin, S. P. Neuman, B. S. RamaRao, C. Ravenne, and Y. Rubin. A comparison of seven geostatistically based inverse approaches to estimate transmissivities for modeling advective transport by groundwater flow. *Water Resources Research*, 34(6):1373–1413, 1998.