Dissertation

submitted to the

Combined Faculty for the Natural Sciences and Mathematics

of

Heidelberg University, Germany

for the degree of

Doctor of Natural Sciences

Put forward by

Diplom-Mathematiker Ole Klein

Born in Frankenthal (Pfalz)

Date of oral examination: 12.07.2016

Preconditioned and Randomized Methods for Efficient Bayesian Inversion of Large Data Sets

and their Application to Flow and Transport in Porous Media

Supervisor:	Prof. Dr. Peter Bastian
Second supervisor:	Prof. Dr. Olaf A Cirpka

Abstract

The efficient and reliable estimation of model parameters is important for the simulation and optimization of physical processes. Most models contain variables that have to be adjusted, e.g. in the form of material properties, and the uncertainty of state estimates and predictions is directly linked to the uncertainty of these parameters. Therefore, efficient methods for parameter estimation and uncertainty quantification are required. If the physical system is spatially highly heterogeneous, then the number of model parameters can be very large. At the same time, imaging techniques and time series can provide a large number of measurements for model calibration. Many of the available methods become inefficient or outright unfeasible if both the number of model parameters and the number of state observations are large.

This thesis is concerned with the development of methods that remain efficient when a large number of measurements is used to estimate an even larger number of model parameters. The main result is a special preconditioned Conjugate Gradients method that can achieve both quasilinear complexity in the number of parameters and pseudo-constant complexity in the number of measurements. The thesis also provides randomized methods that allow linearized uncertainty quantification for large systems, taking redundancy in the measurements into account if applicable.

Zusammenfassung

Die effiziente und zuverlässige Schätzung von Modellparametern ist wichtig für die Simulation und Optimierung physikalischer Prozesse. Die meisten Modelle enthalten unbekannte Größen, z.B. Materialkonstanten, und die Unsicherheit von Zustandsschätzungen und Vorhersagen wird maßgeblich durch die Unsicherheit dieser Parameter beeinflusst. Daher werden effiziente Methoden für Parameterschätzung und Unsicherheitsschätzung benötigt. Falls das physikalische System starke räumliche Heterogenität aufweist, kann die Anzahl der Modellparameter sehr groß sein. Gleichzeitig können bildgebende Verfahren und Zeitreihen große Mengen an Messungen für die Modellkalibrierung bereit stellen. Viele der zur Verfügung stehenden Methoden werden ineffizient oder völlig unbrauchbar, wenn sowohl die Anzahl an Parametern als auch die Anzahl an Zustandsbeobachtungen groß sind.

Diese Arbeit befasst sich mit der Entwicklung von Methoden, die effizient bleiben, wenn eine große Zahl Messungen genutzt wird um eine noch größere Zahl Modellparameter zu schätzen. Das Hauptresultat ist eine spezielle vorkonditionierte Variante des CG-Verfahrens, die quasilineare Komplexität in der Anzahl der Parameter und pseudo-konstante Komplexität in der Anzahl der Messungen erreichen kann. Die Arbeit stellt außerdem randomisierte Methoden zur Verfügung, die eine linearisierte Unsicherheitsschätzung für große Systeme ermöglichen und dabei eine eventuell vorhandene Redundanz in den Messungen berücksichtigen.

Acknowledgments

I would like to thank Prof. Dr. Peter Bastian for giving me the opportunity to carry out my research and write this thesis, for his encouragement and guidance along the way, and for giving me the freedom to find my own path. The cooperative and friendly atmosphere of his research group helped me a lot during my research, and its former and present members have always been willing to help and provide support. I had many fruitful discussions with Dr. Adrian Ngo, who helped me shape my ideas and work out the details of the methods I developed. Dr. Steffen Müthing was always supportive when problems with the hardware cropped up, and Dr. Olaf Ippisch provided helpful advice when I implemented the model equations.

I would also like to thank Prof. Dr. Olaf A Cirpka for his insightful comments and suggestions. His explanations were always instructive, and through his guidance I acquired a deeper understanding of the subject matter. Large parts of this thesis are based on or motivated by the research he and his colleagues conducted throughout the years. My stay at his research group was very productive, and I learned a lot through the many discussions with Dr. Wei Li and Dr. Ronnie Schwede.

I am grateful to everyone who supported me along the way. My membership in the Heidelberg Graduate School of Mathematical and Computational Methods for the Sciences (HGS MathComp) has given me the opportunity to meet many researchers and fellows with completely different academic backgrounds. The discussions I had in this environment often produced new insights and ideas, sometimes through conversations that had only a cursory connection to my own research. I would also like to thank Prof. Dr. Kurt Roth and his research group for the interesting discussions we had. They have positively influenced my research and the methods I developed.

I am thankful for the opportunity to write my thesis at the Interdisciplinary Center for Scientific Computing (IWR) of Heidelberg University, and I would like to acknowledge the support that I have received from the members and staff of IWR and the Faculty of Mathematics and Computer Science. This research has in parts been funded through the GEOTECHNOLOGIEN research and development program of the Federal Ministry of Education and Research of Germany (BMBF), and their support is gratefully acknowledged.

Contents

1 Introduction

This thesis is concerned with the estimation of model parameters from observations of dependent quantities, specifically the efficient estimation of spatially distributed parameter fields from relatively sparse observational data in the presence of noise. The task of estimating model parameters from a given model and its output is called inversion, and parameter estimation problems of this type naturally arise in various scientific areas, among them the earth sciences, material sciences and life sciences. In these fields, inversion is used to create reliable models of highly complex but only indirectly observable systems [54]. While inverse problems appear in a variety of disciplines, from image processing to weather forecasts to tomography, we focus on the type of inverse problem typically encountered in the field of subsurface hydrology.

Subsurface hydrology is concerned with the distribution and flow of water through the pore network of the soil matrix, where the flow patterns are governed by partial differential equations with highly heterogeneous parameters [68]. The large variability of parameters can only be expressed using high-resolution parameter fields, but the number of observations of the system state is typically limited, since most measurement techniques require the installation of observation wells or expensive equipment. Logistic and financial constraints therefore lead to sparse data. In addition, the measurement data typically contains noise from a variety of sources. Inverse modeling can be seen as the attempt to extract as much information about the system parameters as possible under these conditions.

Most inverse modeling approaches are designed for a small number of parameters, a small number of state observations, or both. Such methods typically fail when high-resolution parameter estimates based on the inversion of comparatively large data sets are needed [54, 84]. Some of the methods have superlinear complexity in the number of parameters, e.g. because they have to simulate the model once for each of the parameters, or because the chosen formulation becomes ill-conditioned on finer meshes. Other approaches are linear in the number of state observations, since they have to simulate an adjoint model once for each measurement.

This thesis introduces several methods for the inversion of large data sets, focusing on scenarios where a large number of state observations is used to estimate an even larger number of parameters. Under these conditions the inverse problem is inherently ill-posed and requires regularization, and the considered methods provide regularization at comparatively low cost. Their memory requirements and associated computational

cost are low, quasilinear in the number of parameters and sublinear or even pseudo-constant in the number of observations. Therefore, they are suited for the inversion of large data sets using high-resolution parameter fields.

We operate under the following two assumptions:

- The model is an adequate representation of the physical process under consideration, and the model equations are regular enough to allow differentiation with regard to the model parameters. Based on this assumption, gradient-based methods can be used.
- While the model parameters themselves are unknown, a limited amount of information is available a priori. This may be the typical order of magnitude of the parameters combined with the implicit assumption that the parameters are independent from each other, a description of the autocorrelation structure that governs the parameters, or possibly knowledge about characteristic spatial patterns and internal dependencies. This information can be used as a form of regularization and ensures the well-posedness of the considered problems.

1.1 Modeling and Inverse Modeling

We begin by defining an abstract framework of parameters, parameterized functions, system states, state observations and their interactions. This will allow us to give a precise description of the forward problem and inverse problem as we understand them in the given context, i.e. using a given model to make predictions and estimating parameters for a given model from observations. This framework is a formalization and generalization of common practice in the considered field of research [72, 52], and an example of a concrete application is discussed in section 1.2.

Let $\mathbf{P} \coloneqq (\mathbf{p}_1, \ldots, \mathbf{p}_{n_\mathbf{P}}), \mathbf{p}_i \in \mathbb{R}^{n_{\mathbf{p}_i}}$, be a tuple of parameter vectors, and let $\mathbf{Z} \coloneqq (\mathbf{z}_1, \ldots, \mathbf{z}_{n_\mathbf{Z}}), \mathbf{z}_j \in \mathbb{R}^{n_{\mathbf{z}_j}}$, be a tuple of measurements of dependent quantities. $n_\mathbf{P}$ and $n_\mathbf{Z}$ are the number of parameter fields and the number of state variables, and $n_{\mathbf{p}_i}$ and $n_{\mathbf{z}_j}$ are the number of components of the corresponding vectors. Let $\Omega \subset \mathbb{R}^d, d \in \{2, 3\}$, denote the physical domain, in the following assumed to be a convex polytope for simplicity, $\mathcal{E}_h$ a triangulation of Ω with mesh width h, and $E \in \mathcal{E}_h$ an element of the triangulation.

We assume that each parameter vector $\mathbf{p}_i$ is divided into a spatial part $\mathbf{y}_i$ and a trend part $\boldsymbol{\beta}_i$,

$$\mathbf{p}_i = \begin{pmatrix} \mathbf{y}_i \\ \boldsymbol{\beta}_i \end{pmatrix}, \quad \mathbf{y} \in \mathbb{R}^{n_\Omega}, \quad \boldsymbol{\beta}_i \in \mathbb{R}^{n_{\boldsymbol{\beta}_i}}, \quad n_{\mathbf{p}_i} = n_\Omega + n_{\boldsymbol{\beta}_i}, \tag{1.1}$$

where n_Ω is the number of elements in $\mathcal{E}_h$. Each component $(\mathbf{y}_i)_k$ of $\mathbf{y}_i$ is associated with a subdomain $E_k \in \mathcal{E}_h$ of the domain, while $\boldsymbol{\beta}_i$ consists of non-localized parameters. Furthermore, we assume a relationship of the following form exists between the

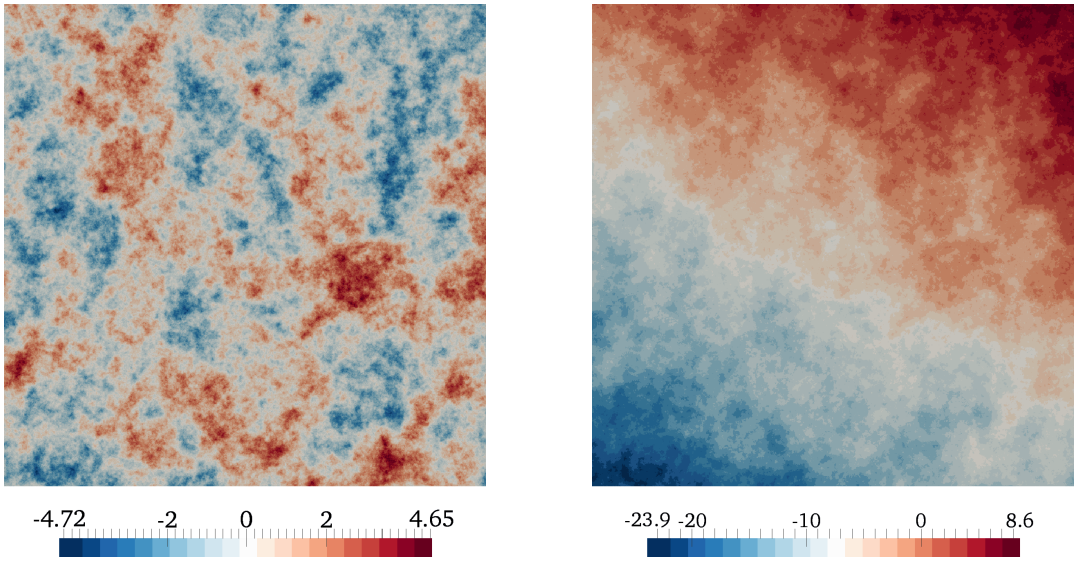

Figure 1.1: *Left:* Example of a parameter vector $\mathbf{y}$ with spatial interpretation. Each component $\mathbf{y}_k$ is assigned to one of the elements $E_k \in \mathcal{E}_h$, and neighboring parameters are highly correlated. *Right:* Parameter field $\mathbf{s}$ resulting from the combination of $\mathbf{y}$ with a trend parameter vector $\boldsymbol{\beta}$ that consists of one coefficient for the spatial mean and two coefficients associated with a linear slope.

parameter vector $\mathbf{p}_i$ and its interpretation as a physical quantity $\mathbf{s}_i$:

$$\mathbf{s}_i \coloneqq \mathbf{y}_i + \mathcal{X}_i(\boldsymbol{\beta}_i), \tag{1.2}$$

where $\mathcal{X}_i$ maps the trend coefficients to corresponding values in the discretization elements $E \in \mathcal{E}_h$. See figure 1.1 for an example of such a parameter vector together with its spatial interpretation. Given this definition of $\mathbf{s}_i$, we can evaluate it as a function on Ω:

$$s_i(\mathbf{x}) \coloneqq (\mathbf{s}_i)_k \text{ if } \mathbf{x} \in E_k \tag{1.3}$$

The functions s_i are the real physical quantities governing the system, averaged over the triangulation $\mathcal{E}_h$, and the values $(\mathbf{s}_i)_k$ their mean over the element E_k. This process can be formalized through the definition of an interpretation operator $\mathcal{I}_i \colon \mathbb{R}^{n_{\mathbf{p}i}} \to L^2(\Omega)$ that maps $\mathbf{p}_i$ to the function s_i.

For each component $\mathbf{z}_j$ of $\mathbf{Z}$ we denote the full system state of the observed quantity with u_j, the space of all possible states with V_j, and assume an observation operator $\mathcal{O}_j \colon V_j \to \mathbb{R}^{n_{\mathbf{z}_j}}$ mapping u_j to $\mathbf{z}_j$ exists. In the most basic case, this operator $\mathcal{O}_j$ simply evaluates the function u_j at a specific location $\mathbf{x}$ or its mean over a given element E to determine a component $(\mathbf{z}_i)_k$ of the measurement vector. More complex operators may take the measurement characteristic of the equipment into account.

Note that in contrast to the interpretation operator $\mathcal{I}_i$ only a small subset of the triangulation $\mathcal{E}_h$ is typically involved in the definition of $\mathcal{O}_j$.

The parameter fields s_i and state functions u_j are linked through models $\mathcal{F}_j$ that relate the physical quantities:

$$\begin{aligned} \mathcal{F}_1\left(S; u_1\right) &= 0 \\ \mathcal{F}_2\left(S, u_1; u_2\right) &= 0 \\ &\vdots \\ \mathcal{F}_j\left(S, u_1, \ldots, u_{j-1}; u_j\right) &= 0 \\ &\vdots \end{aligned} \tag{1.4}$$

or in a more concise notation:

$$\forall\, u_j \colon \mathcal{F}_j\left(S, U_{<j}; u_j\right) = 0 \tag{1.5}$$

with the tuple $U_{<j} \coloneqq (u_1, \ldots, u_{j-1})$. In this notation the semicolon is used to distinguish between the quantities placed on the left and assumed to be known, and the quantity on the right that is determined through the other variables and the implicit function $\mathcal{F}_j$. Typically $\mathcal{F}_j$ is given in the form of a partial differential equation (PDE) and u_j is the solution of the equation.

Remark 1 *The system states u_j are typically elements of Sobolev spaces, not classical continuous and differentiable functions, and their regularity depends on the model PDEs at hand and the regularity of boundary conditions and righthand sides of the equations. Therefore, the models $\mathcal{F}_j$ define mappings between Sobolev spaces, and the PDEs have to be formulated in a weak sense. Nevertheless, we will use the more familiar and succinct classical formulation of PDEs in the following, keeping in mind that it is a shorthand for the weak formulation. Constructs and situations that require special care or interpretation will be pointed out, and we will fully discuss the weak formulations in the introduction of the example model equations in chapter 4.*

A model $\mathcal{F}_j$ may be a trivial postprocessing step if u_j is a byproduct of the computation of a previous $u_k, k < j$, and it may be independent of one or more parameter fields s_i. In this sense, equation (1.4) is a very general formulation that may represent various physical processes. The different functions and operators and their effect are visualized in figure 1.2. Note that the most appropriate structure would actually be a tree, since there may be independent subsets of observations that are not linked through model equations, or measurements that depend on the same model in different ways. However, we focus on structures that are linear in the sense of figure 1.2, since more complex tree structures can be flattened to fit this representation, and modifying the presented methods for the more general case is straightforward.

$$\forall\, i \in \{1, \ldots, n_{\mathbf{P}}\} \colon \mathbf{p}_i \xrightarrow{\mathcal{I}_i} s_i$$

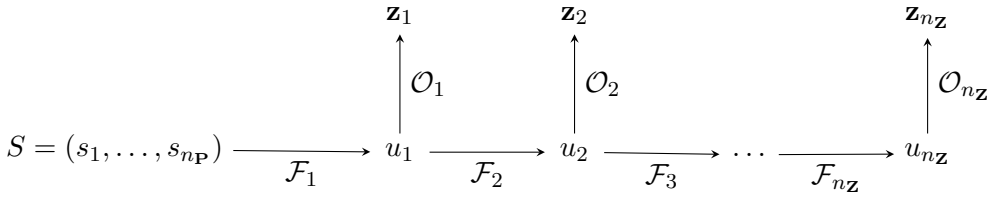

Figure 1.2: Detailed structure of the model of a physical process and its observation. Given parameter vectors $\mathbf{p}_i$ are interpreted as parameterized functions s_i and used to compute the state variables u_j.

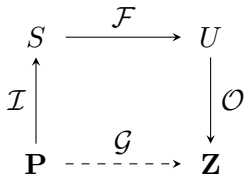

Figure 1.3: Diagram of the forward problem. The parameter-to-measurement map $\mathcal{G}$ is the concatenation of $\mathcal{I}$, $\mathcal{F}$ and $\mathcal{O}$.

Given the above definitions, one may subsume the individual operators $\mathcal{I}_i$, $\mathcal{F}_j$ and $\mathcal{O}_j$, with $1 \leq i \leq n_{\mathbf{P}}$ and $1 \leq j \leq n_{\mathbf{Z}}$, into a combined interpolation operator $\mathcal{I}$ mapping the parameter vector tuple $\mathbf{P}$ to the tuple of parameter fields $S \coloneqq (s_1, \ldots, s_{n_\mathbf{P}}) \in \left(L^2(\Omega)\right)^{n_\mathbf{P}}$, a full model $\mathcal{F}$ mapping S to the system state $U \coloneqq (u_1, \ldots, u_{n_\mathbf{Z}}) \in \prod_{j=1}^{n_\mathbf{Z}} V_j$ and a general observation operator $\mathcal{O}$ mapping U to the measurement vector tuple $\mathbf{Z}$. Figure 1.3 gives a visualization of this more abstract representation of the physical process and its observation: The discrete parameters $\mathbf{P}$ are mapped to the discrete observations $\mathbf{Z}$ via $\mathcal{I}$, $\mathcal{F}$ and $\mathcal{O}$, with the tuples of parameter functions S and state variables U being intermediate stages. This results in a parameter-to-measurement map $\mathcal{G} \coloneqq \mathcal{O} \circ \mathcal{F} \circ \mathcal{I}$.

The forward problem of modeling may now be expressed in the following way:

Problem 1 (Forward Problem)
Given a tuple of parameters $\mathbf{P}$*, a model* $\mathcal{F}$ *and operators* $\mathcal{I}$ *and* $\mathcal{O}$ *as above, determine the tuple of measurements* $\mathbf{Z}$*.*

In other words, the forward problem is equivalent to the evaluation of $\mathcal{G}(\mathbf{P})$ and typically consists of solving $n_{\mathbf{Z}}$ partial differential equations, one for each state variable in U. The corresponding inverse problem may then be expressed as:

Problem 2 (Ill-posed Inverse Problem)
Given a tuple of measurements $\mathbf{Z}$*, a model* $\mathcal{F}$ *and operators* $\mathcal{I}$ *and* $\mathcal{O}$ *as above, determine the tuple of parameters* $\mathbf{P}$*.*

The inverse problem as stated here is almost always ill-posed, mainly due to the observation operator $\mathcal{O}$. Often several different system states lead to the same measurement tuple $\mathbf{Z}$, since the number of observations is comparatively small and the operators $\mathcal{O}_j$ act as filters. Furthermore the model $\mathcal{F}$ may be over-parameterized, resulting in several different parameter fields that are associated with the same system state. And finally, the operator $\mathcal{I}$ need not be invertible. In all these cases the preimage $\mathcal{G}^{-1}(\mathbf{Z})$ contains more than one element and the inverse problem as given above doesn't have a unique solution. For these reasons the inverse problem requires regularization to become solvable, which will be described in detail in chapter 2.

1.2 Application: Flow and Transport in Porous Media

In the context of subsurface hydrology, a model that illustrates all relevant features of the structure defined above is groundwater flow combined with advective transport of solutes. The groundwater flow equation, introduced in more detail in section 4.1, is

$$\mathcal{F}_\phi(Y, Z_s; \phi) \coloneqq S_s(Z_s)\partial_t\phi + \nabla \cdot j_{\theta_w}(Y, \phi) - q_{\theta_w} = 0 \tag{1.6}$$

with the flux

$$j_{\theta_w}(Y, \phi) \coloneqq -K(Y)\nabla\phi = -\exp(Y)\nabla\phi, \tag{1.7}$$

where $K > 0$ is the hydraulic conductivity of the soil, $Y \coloneqq \ln(K)$ the log-conductivity, $S_s > 0$ the specific storativity of the soil, $Z_s \coloneqq \ln(S_s)$ the log-storativity, ϕ the hydraulic head and q_{θ_w} a source term. The conductivity is spatially highly heterogeneous, with K varying over several orders of magnitude. The behavior of the storage term S_s is less pronounced, but it typically has a spatial dependency as well. Together with corresponding initial and boundary conditions, this equation constitutes a mapping from the spatially distributed parameter fields Y and Z_s to the system state ϕ. The advection-dispersion equation, or transport equation for short, discussed in section 4.3, is

$$\mathcal{F}_c(Y, Z_s, \phi; c) \coloneqq \theta\partial_t c + \nabla \cdot j_C(Y, \phi, c) - q_C = 0 \tag{1.8}$$

with

$$j_C(Y, \phi, c) \coloneqq -\left[D(\phi)\nabla c + c j_{\theta_w}\right], \tag{1.9}$$

where θ is the porosity of the soil matrix, c the concentration of a conservative tracer or solute, D its dispersion tensor and q_C again a source term. Combined with suitable initial and boundary conditions, this is a mapping from the fields Y and Z_s and the state ϕ to the second state c. The above equation does not include the influence

of ϕ on the water content, since this contribution is negligible for confined aquifers, which means that the state c only has an indirect dependence on Z_s through ϕ. See section 4.3 for the full formulation.

Remark 2 *Like many other state equations that describe physical systems, the groundwater flow equation and the transport equation can be interpreted as statements about the conservation of a physical quantity, combined with a flux law. They are based on the general continuity equation*

$$\partial_t \rho + \nabla \cdot j_\rho - q_\rho = 0, \tag{1.10}$$

with the flux laws given above. Here ρ is a quantity governed by a law of conservation, e.g. mass, energy or momentum, and j_ρ is its flux. The conserved quantity of the groundwater flow equation is the water content of the soil, while in the case of the transport equation the total amount of solute θc is conserved. Note that the conserved quantities differ from the state variables in the case of the two PDEs above.

It may be assumed that there exists a well-defined mean value for both Y and Z_s in the modeled domain, and that both parameter fields display large structures that are adequately modeled with just a few parameters. These parameters may then be combined into vectors $\boldsymbol{\beta}_Y$ and $\boldsymbol{\beta}_{Z_s}$ with corresponding trend models $\mathcal{X}_Y$ and $\mathcal{X}_{Z_s}$. But both parameter fields, and especially the log-conductivity Y, may be expected to also show features on vastly different length scales due to the complicated processes that are involved in the creation of soil [68]. Since these features can't be explained by the trend models, they are expressed by assigning a single parameter value to each element of the discretization using parameter vectors $\mathbf{y}_Y$ and $\mathbf{y}_{Z_s}$. Appending the trend parameters to the spatial parts results in

$$\mathbf{p}_Y = \begin{pmatrix} \mathbf{y}_Y \\ \boldsymbol{\beta}_Y \end{pmatrix}, \quad \mathbf{p}_{Z_s} = \begin{pmatrix} \mathbf{y}_{Z_s} \\ \boldsymbol{\beta}_{Z_s} \end{pmatrix}, \tag{1.11}$$

and these parameters may then be interpreted as spatially distributed physical quantities $Y(\mathbf{x})$ and $Z_s(\mathbf{x})$ using equations (1.2) and (1.3).

Remark 3 *There is a discrepancy between the classical formulation used in the equations* (1.6) *and* (1.8) *and the way the parameter fields Y and Z_s are defined above, since the fields are piecewise constant on the domain Ω, and not at all defined on the intersections between elements $E \in \mathcal{E}_h$. They are therefore in $L^2(\Omega)$, and the differential equations have to be formulated in a weak sense to constitute a well-posed problem, as already mentioned in remark 1. See chapter 4 for the weak formulation of the models.*

Now assume wells have been drilled that allow for groundwater monitoring. Pumping water into or out of the ground will subject the groundwater to external stresses, and

these will lead to a system response that may be observed in the wells. This results in data about the flow process in the form of measurement values of the hydraulic head ϕ in the direct vicinity of the measurement equipment. While physically restricted to the wells, these data points may be acquired at several locations and points in time, leading to a measurement vector $\mathbf{z}_\phi$ defined by

$$(\mathbf{z}_\phi)_k = \mathcal{O}_\phi(\phi, \mathbf{x}_k, t_k), \quad \mathbf{x}_k \in \Omega, \quad t_k \in T \coloneqq [0, t_{\max}]. \tag{1.12}$$

Here $\mathbf{x}_k$ is the location of the measurement, t_k the moment when it is taken, and $\mathcal{O}_\phi$ is the observation operator for head measurements evaluating ϕ in the vicinity of $\mathbf{x}_k$ at time t_k.

Further assume that a conservative tracer component is introduced upstream and traverses the groundwater network. With suitable equipment the subsequent rise in tracer concentration in the measurement wells can be detected and recorded, resulting in additional information about the groundwater flow. This data may be collected in another measurement vector $\mathbf{z}_c$ through setting

$$(\mathbf{z}_c)_k = \mathcal{O}_c(c, \mathbf{x}_k, t_k), \quad \mathbf{x}_k \in \Omega, \quad t_k \in T, \tag{1.13}$$

with $\mathcal{O}_c$ as above. In (1.12) and (1.13) the same locations $\mathbf{x}_k$ and times t_k have been used for simplicity, but it is of course possible to use a higher sampling frequency for the concentration, and in reality it may be necessary to use different measurement locations due to the size of the equipment.

Under these conditions the groundwater flow equation $\mathcal{F}_\phi$, equation (1.6), and the advection-dispersion equation $\mathcal{F}_c$, equation (1.8), constitute a model $\mathcal{F}$ that maps the fields Y and Z_s to the state variables ϕ and c, and this in turn results in a discrete model $\mathcal{G}$ that maps the parameter vectors $\mathbf{p}_Y$ and $\mathbf{p}_{Z_s}$ to the measurement vectors $\mathbf{z}_\phi$ and $\mathbf{z}_c$. In this situation, the inverse problem 2 takes the following form:

Problem 3 (Concrete Example of Inverse Problem)
Given observations $\mathbf{z}_\phi$ and $\mathbf{z}_c$ of the system states ϕ and c in a domain Ω and a time interval T, and assuming the validity of the groundwater flow equation and transport equation as models of the observed process, determine the underlying parameters $\mathbf{p}_Y$ and $\mathbf{p}_{Z_s}$.

There is a simple configuration that directly shows that this inverse problem can't in general be well-posed: assume that there are two wells for injection and extraction of water, and that all observation points are placed on the direct line between these two wells. This situation is displayed in figure 1.4. Further assume that the parameter fields Y and Z_s are not symmetrical around the mentioned line. Under these conditions, it isn't possible to distinguish between the parameter fields and their mirror images based on information gained from the state observations alone, and therefore the inverse problem doesn't have a unique solution.

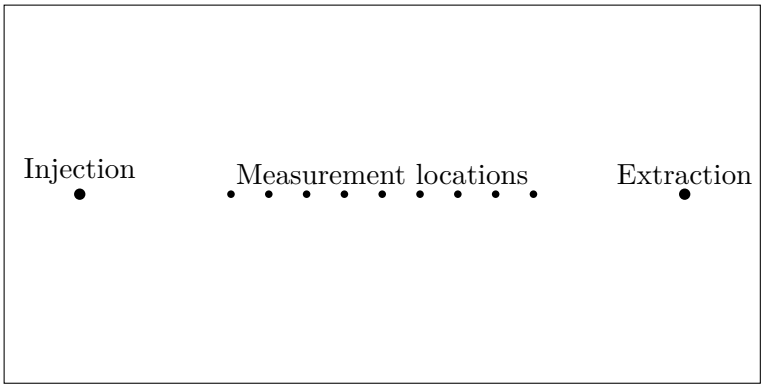

Figure 1.4: Measurement setup that will lead to ambiguity regardless of the equations that are used to model the system. Each set of parameters that is not completely symmetric with regard to the axis defined by the measurement locations will result in observations that have more than one preimage, since the mirror image of the parameters leads to the same observations.

A more general example is a flawed experimental setup where the tracer plume is transported away from the observation wells by the ambient flow instead of towards them. If no hydraulic head data is available, the information content of the measurements is effectively zero, and any number of parameter fields is a solution to the inverse problem. The opposite occurs when measurement errors produce observations that are physically impossible. In the case of steady-state flow with Dirichlet boundary conditions and without sources or sinks, i.e. $q_{\theta_w} = 0$ in Ω, the maximum principle [26] states that the values of ϕ inside the domain are bounded by those on the boundary of Ω. If one of the head observations violates this principle due to measurement errors, there is no consistent and conforming state ϕ and therefore also no inverse solution.

The above examples demonstrate that the inverse problem may be ill-posed in terms of the existence and uniqueness of the inverse solution, with the limiting cases being the complete absence of a solution on the one hand and every possible tuple of parameter vectors being a valid solution on the other hand. However, the most important aspect is typically the missing stability of the inverse solution. Classical examples in this regard are the backwards heat equation, i.e. the task of determining the initial condition of the heat equation from measurements taken at later times, and the sideways heat equation, i.e. the task of determining the boundary condition on one part of the domain boundary from information about the other parts. As is shown in [25], both problems are unstable in the sense that arbitrarily small changes in the given information lead to arbitrarily large changes in the solutions. Since the heat equation is just a specific interpretation of the diffusion equation, these findings also apply in the larger context of the inversion of diffusive models, e.g. the groundwater flow equation or the Richards equation. In this situation, regularization can be

interpreted as the introduction of smoothness constraints to stabilize the inherently anti-diffusive process of inversion. An example of a more detailed discussion from the perspective of groundwater modeling can be found in [74], while [3] describes ill-posed inverse problems from seismology and discusses regularization and solution techniques.

1.3 Major Contributions

This work discusses several numerical methods and algorithms that are, to the best of our knowledge, new developments or significant extensions of existing methods. Furthermore, a number of well-established numerical and mathematical techniques are applied in novel contexts. In this section we collect all such contributions for ease of reference.

Prior Information as Preconditioner In section 2.3, we use the inverse of the prior covariance matrix as a preconditioner to drastically improve the convergence rate of the Conjugate Gradients method. While prior information has already been used for preconditioning before, e.g. by *Bui-Thanh et al.* [15], it has to the best of our knowledge neither been applied for the kind of prior information we use, nor directly to the Conjugate Gradients scheme. Preconditioning with prior information not only provides a drastic reduction in the number of iterations, the implicit removal of spurious modes can lead to mesh-independent convergence rates.

Elimination of Inverse Prior Covariance Matrix Choosing the inverse of the prior covariance matrix as preconditioner allows us to eliminate it completely from the algorithm, as detailed in section 2.3.5. As a result, the preconditioned scheme not only requires less iterations, the eliminated matrix multiplication reduces the effort needed for an iteration of the method, i.e. the preconditioner has "negative cost".

Randomized Uncertainty Quantification While the presented PCG method is suitable for large-scale transient applications for which the assembly of Hessian information is too costly, it does not provide information about the uncertainty of the estimate. In section 2.5, we present two randomized algorithms for the calculation of the posterior covariance matrix under such conditions, one using a partial spectral decomposition and one using a partial singular value decomposition (SVD). Both are applications of the theory developed by *Halko et al.* [36], and the spectral decomposition has been adapted from *Bui-Thanh et al.* [15]. The randomized SVD is a new development that additionally provides information for statistical a posteriori analysis of the inversion results, see section 2.6.

Hessian Information as Preconditioner In section 2.7, we combine the results of the previous sections, using an estimate of the inverse of the posterior covariance matrix as preconditioner to further improve the convergence rate of the PCG method. The estimate is created by applying the uncertainty quantification algorithms of section 2.5 to the initial guess instead of the inversion result. Consequently, the method has high initialization costs, but the inclusion of model information in the preconditioner may improve convergence, especially in cases where the regularization is relatively weak. This second PCG scheme also allows eliminating the inverse of the prior covariance matrix from the algorithm, which again leads to "negative cost" of the preconditioner apart from the setup phase.

Randomized Gauss-Newton and Levenberg-Marquardt The two partial decompositions can also be used in other algorithms that rely on estimates of the posterior covariance matrix, e.g. the Gauss-Newton (GN) method. In section 3.3.2, we present a randomized variant of this scheme. This method is conceptually related to the Principal Component Geostatistical Approach (PCGA) by *Lee and Kitanidis* [49, 46]. The decomposition that is chosen here is typically more expensive than that of PCGA, but additionally includes information from the forward model and is self-calibrating in the number of singular values that are obtained. Furthermore, interpolating between this randomized method and PCG produces a variant of the Levenberg-Marquardt method similar to the one presented by *Nowak and Cirpka* [62]. This scheme is also given in section 3.3.2.

Inversion of Large Data Sets The prior preconditioned CG method has memory requirements that are independent of the number of observations and computational costs that are largely independent of that number, see section 6.1.1, while the randomized methods are based on spectral decompositions that automatically filter measurement noise that can't be reconciled with the forward model. This may make these methods useful for the inversion of large data sets, e.g. high-resolution time series or the results of imaging techniques, since it is no longer necessary to smoothen or filter the data to reduce its dimension before it is used as input for parameter estimation.

1.4 Document Outline

The remaining chapters of this document are structured in the following way:

Chapter 2 (Method Description) develops a method for the regularization of the ill-posed inverse problem 2. It gives a short introduction into Bayesian statistics from the viewpoint of geostatistical inversion, and provides both the stochastic and the Maximum A Posteriori (MAP) formulation of the inverse problem. The preconditioned Conjugate Gradients (PCG) method is introduced to solve the resulting minimization problem, and randomized algorithms for uncertainty

quantification are presented. The chapter concludes with a summary that highlights the most important aspects of the discussed methods.

Chapter 3 (Alternative Approaches) discusses possible alternatives to the approach described in chapter 2. Different methods may be chosen to regularize the ill-posed inverse problem 2, to reduce the complexity of the resulting formulations, or to solve the underlying optimization problems. An overview of popular approaches is given and their relation with the methods described in chapter 2 is discussed. Based on the decompositions of the previous chapter, randomized variants of the Gauss-Newton (GN) method are introduced.

Chapter 4 (Governing Equations) presents governing equations for flow and transport processes in porous media, which are examples of the models $\mathcal{F}_i$ relating parameters and observations above. It also derives adjoint equations, which may be used in the methods discussed in chapters 2 and 3.

Chapter 5 (Implementation Details) gives detailed information about the numerical reference implementation of the methods described in chapter 2. Included are the used spatial discretizations, time stepping schemes and flux reconstruction.

Chapter 6 (Applications) presents results obtained with the reference implementation of chapter 5. The properties of the method, as described in chapter 2, are demonstrated with the help of synthetic test cases, and both the advantages and limitations of the method are discussed.

Chapter 7 (Conclusions) summarizes the central results of the previous chapters, draws conclusions about the applicability of the method and mentions areas that require further research.

2 Method Description

In order to arrive at a well-posed problem definition, the ill-posed inverse problem 2 needs to be regularized. Several different approaches to achieve this exist, each with its own benefits, drawbacks and reasoning. The main properties such a regularization should have are the following:

- The resulting reformulation of the problem should be general enough to be applicable in a wide range of situations.
- For each admissible set of state observations there should be exactly one corresponding configuration of parameters.
- The inversion should be robust, i.e. a small perturbation in the observations should not result in a large deviation in the estimated parameters.
- The regularized problem should be "close" to the original ill-posed problem in a quantifiable way.

The first three properties correspond to Hadamard's definition of well-posedness [34, 25], while the fourth ensures that the well-posed problem is as similar as possible to the original. The different approaches typically differ in the way they achieve uniqueness of the solution, or at least a drastic reduction in the number of solutions, and in what mathematical terms the "distance" between the two problems is defined. In the field of geoscience, several different approaches have been developed in the last decades. Most of these amount to regularization through a drastic reduction in the number of parameters, often through explicit or implicit assumptions about the concrete layout of the examined area.

Among the first who used statistical information for parameter estimation in the geosciences was *D.G. Krige*, who estimated average gold rates in South Africa using probes from boreholes [47]. This approach was formalized by *Georges Matheron* [53] and is known as Kriging. While Kriging itself only estimates the spatial distribution of a variable based on direct measurements, later developments also allowed the incorporation of other quantities, provided information about the correlation structure was available. These methods typically are applications of the Bayesian method of inference. In contrast to the zonation methods mentioned above, they neither impose a fixed spatial structure on the parameters nor do they restrict the number of parameters. Instead, both the parameters and the state observations are treated as multi-dimensional random variables. Several closely related variants of this approach exist [84], of which we mention the Quasi-Linear Geostatistical Approach (QLGA)

by *Kitanidis* [45] and the Successive Linear Estimator (SLE) by *Yeh et al.* [83] as examples. As shown in [54], most of these methods may be interpreted as Maximum A Posteriori (MAP) estimation, often using variants of the Gauss-Newton algorithm or the Conjugate Gradients method for optimization.

The goal of this chapter is the development of an extension of the above approach that remains efficient when applied to scenarios that require the solution of computationally demanding models and the inversion of large data sets. In sections 2.1 and 2.2 the background of the proposed method is presented to provide context. Then the method is developed incrementally in the subsequent sections, and afterwards section 2.8 provides a summary and discussion of the introduced algorithms and their applicability.

2.1 Random Variables and Covariance Matrices

As a preparation for the discussion of Bayesian inference, we first provide a short recapitulation of random variables and their properties as they are defined in the literature, e.g. in [33]. Such a random variable $\mathbf{Y}$ is, in principle, the assignment of real values $\mathbf{Y}(\omega)$ to a given set of events ω. This set of events, called the sample space, models the outcome of an experiment or some other process with distinguishable states and associated probabilities. Since each of the events has a given probability to occur, the value of the random variable $\mathbf{Y}$ is in general uncertain as well. In many applications the concrete sample space and the assigned probabilities are ignored, and instead the random variable is simply characterized by its distribution, i.e. a map that assigns each possible value of $\mathbf{Y}$ the probability of obtaining that value.

For continuous random variables the probability of any given value is strictly speaking zero, and consequently it is more appropriate to use a probability density function (PDF) $f_{\mathbf{Y}}$ to describe such a $\mathbf{Y}$. For any interval $A \subset \mathbb{R}$, the probability of $\mathbf{Y}$ taking a value in A can then be expressed as

$$\mathbb{P}\left[\mathbf{Y} \in A\right] = \int_A f_{\mathbf{Y}}(\mathbf{x}_{\mathbf{Y}}), \tag{2.1}$$

which is an integral over the probability density of all possible values $\mathbf{x}_{\mathbf{Y}} \in A$. Such a value $\mathbf{x}_{\mathbf{Y}}$ is called a realization or sample of $\mathbf{Y}$, and we will often simply use the name of the random variable itself to designate such samples if the meaning is clear from context.

As long as the associated integral is well-defined, random variables allow the computation of their probability-weighted average, known as the expected value or mean and given by

$$\mathbf{Y}^* \coloneqq \mathbb{E}\left[\mathbf{Y}\right] \coloneqq \int_{\mathbb{R}} \mathbf{x}_{\mathbf{Y}} \cdot f_{\mathbf{Y}}(\mathbf{x}_{\mathbf{Y}}). \tag{2.2}$$

The spread around this mean is typically characterized by the variance

$$\sigma_{\mathbf{Y}}^2 \coloneqq \mathrm{Var}\left[\mathbf{Y}\right] \coloneqq \mathbb{E}\left[\left[\mathbf{Y} - \mathbb{E}\left[\mathbf{Y}\right]\right]^2\right] = \int_{\mathbb{R}} \left[\mathbf{x}_{\mathbf{Y}} - \mathbf{Y}^*\right]^2 \cdot f_{\mathbf{Y}}(\mathbf{x}_{\mathbf{Y}}) \tag{2.3}$$

or its root, the standard deviation

$$\sigma_{\mathbf{Y}} \coloneqq \left[\mathrm{Var}\left[\mathbf{Y}\right]\right]^{1/2}. \tag{2.4}$$

These three quantities are the most used measures for the description of random variables. While we will mostly focus on the mean and variance, we also provide the definitions for the higher-order moments skewness and kurtosis, which are given by

$$\mathrm{Skew}\left[\mathbf{Y}\right] \coloneqq \mathrm{Var}\left[\mathbf{Y}\right]^{-3/2} \cdot \mathbb{E}\left[\left[\mathbf{Y} - \mathbb{E}\left[\mathbf{Y}\right]\right]^3\right] \tag{2.5}$$

and

$$\mathrm{Kurt}\left[\mathbf{Y}\right] \coloneqq \mathrm{Var}\left[\mathbf{Y}\right]^{-2} \cdot \mathbb{E}\left[\left[\mathbf{Y} - \mathbb{E}\left[\mathbf{Y}\right]\right]^4\right] \tag{2.6}$$

respectively. These moments are useful for the a posteriori analysis in section 2.6.

2.1.1 Gaussian Random Vectors and Random Fields

In many applications random processes must be modeled based on limited knowledge about the true PDF $f_{\mathbf{Y}}$, and often a Gaussian distribution with a prescribed mean μ and variance σ^2 is used instead. This distribution is also known as normal distribution and is defined by its PDF

$$f_{\mathcal{N}}(\mathbf{x}_{\mathbf{Y}}; \mu, \sigma^2) \coloneqq \left[\tau\sigma^2\right]^{-1/2} \exp\left(-\frac{1}{2}\sigma^{-2}\left[\mathbf{x}_{\mathbf{Y}} - \mu\right]^2\right), \tag{2.7}$$

where $\tau \coloneqq 2\pi \approx 6.283\ldots$ is the circumference of the unit circle. The notation

$$\mathbf{Y} \sim \mathcal{N}\left(\sigma_{\mathbf{Y}}^2, \mathbf{Y}^*\right) \tag{2.8}$$

is used to express that $\mathbf{Y}$ is modeled using a Gaussian distribution, i.e. assuming equation (2.7) holds with $\sigma^2 \coloneqq \sigma_{\mathbf{Y}}^2$ and $\mu \coloneqq \mathbf{Y}^*$.

It is straightforward to generalize this definition to random vectors, i.e. vectors that have random variables as components. Such a vector $\mathbf{Y} = (\mathbf{Y}_1, \mathbf{Y}_2, \ldots, \mathbf{Y}_k)$ is said to be multivariate Gaussian distributed, or simply normally distributed, if each of its components $\mathbf{Y}_k$ follows equation (2.7). It can then be described by the multidimensional Gaussian PDF

$$f_{\mathcal{N}}(\mathbf{x}_{\mathbf{Y}}; \boldsymbol{\mu}, \mathbf{Q}) \coloneqq \left[\tau\left|\mathbf{Q}\right|\right]^{-k/2} \exp\left(-\frac{1}{2}\left[\mathbf{x}_{\mathbf{Y}} - \boldsymbol{\mu}\right]^T \mathbf{Q}^{-1}\left[\mathbf{x}_{\mathbf{Y}} - \boldsymbol{\mu}\right]\right), \tag{2.9}$$

where k is the number of components of $\mathbf{Y}$, $\boldsymbol{\mu}$ a k-dimensional vector that is the mean of the distribution and $\mathbf{Q}$ a symmetric positive definite matrix of dimension $k \times k$ known as covariance matrix. As above,

$$\mathbf{Y} \sim \mathcal{N}(\mathbf{Q}_{\mathbf{YY}}, \mathbf{Y}^*) \tag{2.10}$$

expresses that $\mathbf{Y}$ is normally distributed with mean $\boldsymbol{\mu} \coloneqq \mathbf{Y}^*$ and covariance matrix $\mathbf{Q} \coloneqq \mathbf{Q}_{\mathbf{YY}}$.

Remark 4 *There are Gaussian distributions where* $\mathbf{Q}$ *does not have full rank and is only positive semidefinite, and in this situation the above PDF is not defined. While such distributions certainly have important applications, in the given context a rank-deficient matrix* $\mathbf{Q}$ *would imply that one of the parameters is a function of the others or one of the observations is completely determined through the others, including any measurement noise. In the following we assume that these redundant entries have been removed, i.e. the covariance matrix* $\mathbf{Q}$ *is always assumed to be positive definite so that a PDF as in equation* (2.9) *exists.*

A Gaussian random field on the domain Ω is a Gaussian random vector $\mathbf{Y}$ with associated spatial information. Each of its components $\mathbf{Y}_k$ is assigned to a location in Ω, e.g. through association with a subdomain $E_k \in \mathcal{E}_h$ in analogy to the spatial parts $\mathbf{y}_i$ of the parameter vectors $\mathbf{p}_i$ on page 2. This spatial information may be used to construct physically reasonable covariance matrices based on a small number of underlying assumptions and variables [61]. One such assumption that is often employed is second-order stationarity of the random field, i.e. the assumption that the mean and the covariance structure are invariant under translation. As a consequence, all components have the same mean μ, and the covariance between two components $\mathbf{Y}_i$ and $\mathbf{Y}_j$ is given by

$$(\mathbf{Q}_{\mathbf{YY}})_{ij} \coloneqq \mathbb{E}\left[[\mathbf{Y}_i - \mu]\,[\mathbf{Y}_j - \mu]\right] = r(\mathbf{x}_i - \mathbf{x}_j), \tag{2.11}$$

where $\mathbf{x}_i$ and $\mathbf{x}_j$ are the spatial coordinates associated with $\mathbf{Y}_i$ and $\mathbf{Y}_j$ respectively, e.g. the centers of the subdomains $E_i, E_j \in \mathcal{E}_h$, and r is a covariance function that provides the covariance between two components as a function of their distance.

Two of the most often used choices for r are

$$r_{\exp}(\mathbf{x}) \coloneqq \exp\left(-\lambda^{-1} \|\mathbf{x}\|_2\right), \tag{2.12}$$

known as exponential covariance, and

$$r_{\text{Gauss}}(\mathbf{x}) \coloneqq \exp\left(-\left[\lambda^{-1} \|\mathbf{x}\|_2\right]^2\right), \tag{2.13}$$

known as Gaussian covariance. In both definitions, λ is a scale parameter that is called correlation length and defines the scale of dominant features in realizations of the random field. While this hyperparameter may in principle be treated as an

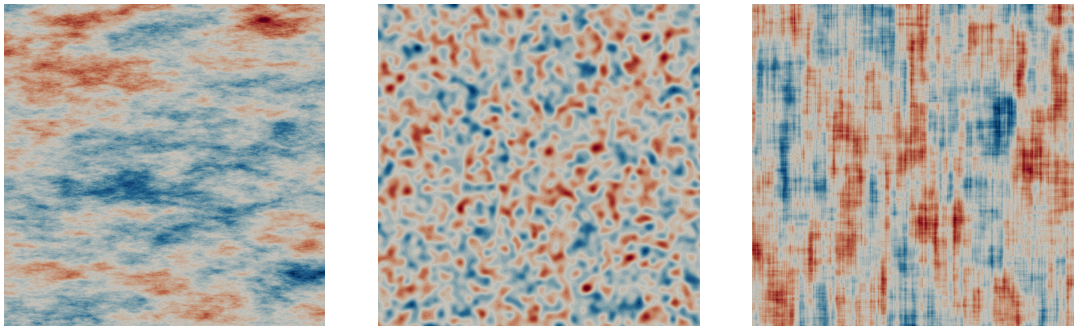

Figure 2.1: Examples of different second-order stationary covariance structures. *Left:* Exponential covariance function similar to equation (2.12), but with two different correlation lengths $\lambda_1 \neq \lambda_2$. The horizontal correlation length λ_1 is three times larger than the vertical correlation length λ_2. *Middle:* Gaussian covariance function as in equation (2.13). *Right:* Separable exponential covariance function $r(\mathbf{x}) \coloneqq \exp\left(-\left[\lambda_1^{-1}\left|\mathbf{x}_1\right| + \lambda_2^{-1}\left|\mathbf{x}_2\right|\right]\right)$. The vertical correlation length λ_2 is four times the horizontal correlation length λ_1.

unknown and estimated just as the other parameters, see e.g. *Michalak and Kitanidis* [56], we assume that it is known for simplicity. Anisotropic variants of these functions may be defined through a scale vector $(\lambda_1, \ldots, \lambda_d) \in \mathbb{R}^d$ that scales each dimension individually, or in the most general case through a symmetric positive definite transformation matrix applied to the distance $\mathbf{x}$ [22]. Examples of random fields with stationary covariance can be found in figure 2.1.

While many more covariance functions exist, we restrict ourselves to r_{exp} and r_{Gauss} for simplicity. The methods discussed in the following are also applicable when using other covariance functions, or a different covariance structure altogether. The only requirement is the availability of fast algorithms for the application of $\mathbf{Q_{YY}}$ to vectors and for decompositions of the form

$$\mathbf{Q_{YY}} = \mathbf{L}\mathbf{L}^T, \tag{2.14}$$

as we describe them in the next section. If only multiplication with $\mathbf{Q_{YY}}$ is fast, then the parameters may still be estimated, but their uncertainty can't be quantified with the presented methods, since their application requires the above decomposition.

2.1.2 Covariance Matrix Calculus

The reformulation of the ill-posed inverse problem 2 that will be discussed in the next section requires matrix operations based on the covariance matrix $\mathbf{Q_{YY}}$. If the Gaussian random field is second-order stationary, as we will always assume in the following, then this multiplication with $\mathbf{Q_{YY}}$ can be carried out efficiently using the Fast Fourier Transform (FFT) [22]. For stationary random fields $\mathbf{Y}$ associated with

a structured grid $\mathcal{E}_h$ on Ω, the covariance matrix $\mathbf{Q_{YY}}$ is block Toeplitz with blocks that are themselves Toeplitz, if the elements of $\mathcal{E}_h$ are ordered lexicographically. Embedding Ω in a larger domain Ω^{ext} that is large enough, the covariance matrix $\mathbf{Q_{YY}}$ can then be extended to a symmetric positive semidefinite block circulant matrix $\mathbf{Q}^{\text{ext}}_{\mathbf{YY}}$ with blocks that are themselves circulant. Such matrices are diagonalized by the multidimensional discrete Fourier transform. As a result, the multiplication with $\mathbf{Q_{YY}}$ can be written as

$$\mathbf{Q_{YY}Y} = \mathbf{E}^T\mathbf{F}^{-1}\mathbf{\Lambda_Y}\mathbf{FEY}, \tag{2.15}$$

where $\mathbf{\Lambda_Y}$ is a diagonal matrix containing the eigenvalues of $\mathbf{Q}^{\text{ext}}_{\mathbf{YY}}$, $\mathbf{F}$ is the discrete Fourier transform, $\mathbf{F}^{-1}$ is the inverse transform, $\mathbf{E}$ extends $\mathbf{Y}$ to the larger domain Ω^{ext} by padding it with zeros, and $\mathbf{E}^T$ restricts the result to Ω again. Using this approach, the multiplication with $\mathbf{Q_{YY}}$ can be performed at the cost of two multidimensional discrete Fourier transforms and an amount of memory that is a small multiple of that for storing a realization of $\mathbf{Y}$.

The multiplication with $\mathbf{Q}^{-1}_{\mathbf{YY}}$ is typically much more expensive. The inverse of the extended matrix $\mathbf{Q}^{\text{ext}}_{\mathbf{YY}}$ is not an extension of the matrix $\mathbf{Q}^{-1}_{\mathbf{YY}}$ onto the larger domain Ω^{ext}, since the finite extent of Ω leads to boundary effects in the inverse matrix [61]. As a consequence, the direct application of the circulant embedding technique described above to $\mathbf{Q}^{-1}_{\mathbf{YY}}$ isn't possible. Due to the large size of $\mathbf{Y}$ in realistic applications, one usually resorts to iterative methods, and in this context the matrix $\left[\mathbf{Q}^{\text{ext}}_{\mathbf{YY}}\right]^{-1}$ can be used as an efficient preconditioner. However, multiplication with $\mathbf{Q}^{-1}_{\mathbf{YY}}$ remains an expensive operation and may be inherently unstable for certain covariance structures, as discussed by *Nowak* [61]. While the reformulation of the inverse problem and the methods will initially contain multiplications with such an inverse, we will show in sections 2.3.5 and 2.7 how the algorithms can be restructured to avoid this expensive operation.

Apart from multiplication with $\mathbf{Q_{YY}}$ and its inverse $\mathbf{Q}^{-1}_{\mathbf{YY}}$, a third operation is typically needed. If a decomposition of the covariance matrix $\mathbf{Q_{YY}}$ of the form

$$\mathbf{Q_{YY}} = \mathbf{LL}^T \tag{2.16}$$

with an invertible matrix $\mathbf{L}$ is known, then equation (2.9) implies that

$$\mathbf{Y} \sim \mathcal{N}\left(\mathbf{Q_{YY}}, \mathbf{Y}^*\right) \iff \mathbf{L}^{-1}\left[\mathbf{Y} - \mathbf{Y}^*\right] \sim \mathcal{N}\left(\mathbf{I}, \mathbf{0}\right) \tag{2.17}$$

holds, where $\mathbf{I}$ is the identity matrix. This allows the generation of realizations of $\mathbf{Y}$ from samples of $\mathcal{N}\left(\mathbf{I}, \mathbf{0}\right)$ [22]. These latter samples, also known as white noise, are simple to produce due to the missing correlation, and applying the matrix $\mathbf{L}$ to the result and then adding the mean $\mathbf{Y}^*$ produces samples of $\mathbf{Y}$. For reference purposes we collect these steps in algorithm 1 (**SG**). The inverse process, i.e. subtracting the mean and multiplying with $\mathbf{L}^{-1}$, produces a realization of $\mathcal{N}\left(\mathbf{I}, \mathbf{0}\right)$ from samples of $\mathcal{N}\left(\mathbf{Q_{YY}}, \mathbf{Y}^*\right)$, which is a convenient statistical check that we will apply in the a posteriori analysis of section 2.6.

Algorithm 1: Generation of Samples from Prior Distribution (**SG**)

Input: Mean $\mathbf{Y}^*$, decomposition of covariance matrix $\mathbf{Q_{YY}} = \mathbf{L}\mathbf{L}^T$
Output: Sample of distribution $\mathbf{Y}$
$\mathbf{W} \coloneqq \mathcal{N}(\mathbf{I}, \mathbf{0})$ [generate white noise];
$\mathbf{Y} \coloneqq \mathbf{Y}^* + \mathbf{L}\mathbf{W}$ [transform to correct covariance structure];
return $\mathbf{Y}$;

Due to these two applications and others that will be discussed in the next sections, decompositions as in equation (2.16) are important. A well-known decomposition that can be used is the Cholesky decomposition, where $\mathbf{L}$ is a lower triangular matrix, i.e. all entries of $\mathbf{L}$ above the main diagonal are zero. While the Cholesky decomposition may theoretically be computed for all covariance matrices, its cost may be a limiting factor in practice [52].

Another possibility for equation (2.16) is the spectral decomposition of $\mathbf{Q_{YY}}$,

$$\mathbf{Q_{YY}} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T, \tag{2.18}$$

where $\mathbf{V}$ is an orthogonal matrix containing the eigenvectors of $\mathbf{Q_{YY}}$ and $\mathbf{\Lambda}$ is a diagonal matrix containing its eigenvalues. Since $\mathbf{Q_{YY}}$ is positive definite, all its eigenvalues are positive, and we may therefore define $\mathbf{L} \coloneqq \mathbf{V}\mathbf{\Lambda}^{1/2}$ to obtain a suitable decomposition. In realistic applications a full decomposition can't be stored due to the large number of eigenvectors, and an approximate decomposition of $\mathbf{Q_{YY}}$ as detailed in [49] or section 2.5 may prove useful.

A third possibility is choosing $\mathbf{L} \coloneqq \mathbf{Q}_{\mathbf{YY}}^{1/2}$, the positive root of $\mathbf{Q_{YY}}$. This matrix is again symmetric positive definite, and therefore

$$\mathbf{Q_{YY}} = \mathbf{Q}_{\mathbf{YY}}^{1/2}\mathbf{Q}_{\mathbf{YY}}^{1/2} = \mathbf{Q}_{\mathbf{YY}}^{1/2}\left[\mathbf{Q}_{\mathbf{YY}}^{1/2}\right]^T \tag{2.19}$$

is a decomposition that has the desired properties. For general covariance matrices, this root is even more expensive to compute than the Cholesky decomposition, but in the case of a stationary random field the circulant embedding technique can be applied. We have

$$\mathbf{Q}_{\mathbf{YY}}^{1/2}\mathbf{Y} \approx \mathbf{E}^T\mathbf{F}^{-1}\mathbf{\Lambda}_{\mathbf{Y}}^{1/2}\mathbf{F}\mathbf{E}\mathbf{Y}, \tag{2.20}$$

where the matrices $\mathbf{\Lambda_Y}$, $\mathbf{F}$ and $\mathbf{E}$ are the same as in equation (2.15), since

$$\begin{aligned}\mathbf{Q_{YY}} &= \mathbf{E}^T\mathbf{F}^{-1}\mathbf{\Lambda_Y}\mathbf{F}\mathbf{E} \\ &= \mathbf{E}^T\mathbf{F}^{-1}\mathbf{\Lambda}_{\mathbf{Y}}^{1/2}\mathbf{F}\mathbf{F}^{-1}\mathbf{\Lambda}_{\mathbf{Y}}^{1/2}\mathbf{F}\mathbf{E} \\ &\approx \mathbf{E}^T\mathbf{F}^{-1}\mathbf{\Lambda}_{\mathbf{Y}}^{1/2}\mathbf{F}\mathbf{E}\mathbf{E}^T\mathbf{F}^{-1}\mathbf{\Lambda}_{\mathbf{Y}}^{1/2}\mathbf{F}\mathbf{E}.\end{aligned} \tag{2.21}$$

Including the projection matrix $\mathbf{E}\mathbf{E}^T$ introduces a systematic error in the last line. This error is similar to the one discussed above for the multiplication with $\mathbf{Q}_{\mathbf{YY}}^{-1}$, but

typically much more benign, since multiplication with $\mathbf{Q}_{\mathbf{YY}}^{1/2}$ is normally a smoothing operation in contrast to multiplication with $\mathbf{Q}_{\mathbf{YY}}^{-1}$. If $\mathbf{Q}_{\mathbf{YY}}^{1/2}\mathbf{Y}$ is needed for a $\mathbf{Y}$ that is zero in a layer of several correlation lengths around the boundary, then this error will be negligible for the standard covariance functions due to their locality, and we may use equation (2.20) to perform multiplication with $\mathbf{Q}_{\mathbf{YY}}^{1/2}$. This property can often be guaranteed by choosing a domain Ω that is large enough. Note that this doesn't necessarily increase the cost for simulations if a third domain Ω' with $\Omega \subset \Omega' \subset \Omega^{\text{ext}}$ is introduced, where Ω and Ω^{ext} are again the physical domain and its extension, and the random field $\mathbf{Y}$ is defined on Ω' instead of Ω. For convenience, we assume that choosing $\Omega' = \Omega$ is sufficient.

Remark 5 *If the error that is incurred by using equation* (2.20) *can't be neglected, e.g. because the covariance function doesn't decay fast enough and the domain can't be chosen large enough for external reasons, then the matrix* $\mathbf{Q}_{\mathbf{YY}}^{1/2}$ *has to be replaced by its spectral decomposition*

$$\mathbf{Q}_{\mathbf{YY}}^{1/2} = \mathbf{V}\mathbf{\Lambda}^{1/2}\mathbf{V}^T, \tag{2.22}$$

where the matrices $\mathbf{V}$ *and* $\mathbf{\Lambda}$ *are the same as in equation* (2.18)*. This spectral decomposition can be constructed from the one of* $\mathbf{Q}_{\mathbf{YY}}$*, which in turn can be obtained through variants of the randomized algorithms that will be discussed in section 2.5.*

2.2 Bayesian Inference

We may use the definitions of the previous chapter to formulate a well-posed version of the inverse problem. Assume that the parameter tuple $\mathbf{P}$ is normally distributed, i.e. follows a multivariate Gaussian distribution:

$$\mathbf{P} \sim \mathcal{N}\left(\mathbf{Q}_{\mathbf{PP}}, \mathbf{P}^*\right), \tag{2.23}$$

with a given covariance matrix $\mathbf{Q}_{\mathbf{PP}}$, assumed to be symmetric positive definite in the following, and mean $\mathbf{P}^*$. This expresses prior knowledge about the system, since a high probability of a given parameter tuple $\mathbf{P}$ indicates that it is a good representative of the assumed spatial structure of the domain. A more detailed formulation of equation (2.23) is

$$\begin{pmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \vdots \\ \mathbf{p}_{n_\mathbf{P}} \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mathbf{Q}_{\mathbf{p}_1\mathbf{p}_1} & \mathbf{Q}_{\mathbf{p}_1\mathbf{p}_2} & \cdots & \mathbf{Q}_{\mathbf{p}_1\mathbf{p}_{n_\mathbf{P}}} \\ \mathbf{Q}_{\mathbf{p}_2\mathbf{p}_1} & \mathbf{Q}_{\mathbf{p}_2\mathbf{p}_2} & \cdots & \mathbf{Q}_{\mathbf{p}_2\mathbf{p}_{n_\mathbf{P}}} \\ \vdots & \vdots & & \vdots \\ \mathbf{Q}_{\mathbf{p}_{n_\mathbf{P}}\mathbf{p}_1} & \mathbf{Q}_{\mathbf{p}_{n_\mathbf{P}}\mathbf{p}_2} & \cdots & \mathbf{Q}_{\mathbf{p}_{n_\mathbf{P}}\mathbf{p}_{n_\mathbf{P}}} \end{pmatrix}, \begin{pmatrix} \mathbf{p}_1^* \\ \mathbf{p}_2^* \\ \vdots \\ \mathbf{p}_{n_\mathbf{P}}^* \end{pmatrix} \right), \tag{2.24}$$

where $\mathbf{Q}_{\mathbf{p}_i\mathbf{p}_i}$ is the covariance matrix associated with $\mathbf{p}_i$, $\mathbf{p}_i^*$ is its mean, and $\mathbf{Q}_{\mathbf{p}_i\mathbf{p}_j}$ is the cross-covariance matrix between $\mathbf{p}_i$ and $\mathbf{p}_j$. It should be noted that for simplicity

we will always assume $\mathbf{Q}_{\mathbf{p}_i\mathbf{p}_j} = 0, i \neq j$, in the applications of chapter 6. In this situation $\mathbf{Q}_{\mathbf{PP}} = \operatorname{diag}\left(\mathbf{Q}_{\mathbf{p}_1\mathbf{p}_1}, \mathbf{Q}_{\mathbf{p}_2\mathbf{p}_2}, \ldots, \mathbf{Q}_{\mathbf{p}_{n_\mathbf{P}}\mathbf{p}_{n_\mathbf{P}}}\right)$ is a block diagonal matrix. Each of the covariance matrices $\mathbf{Q}_{\mathbf{p}_i\mathbf{p}_i}$ is itself a block diagonal matrix, since it consists of the covariance matrices $\mathbf{Q}_{\mathbf{y}_i\mathbf{y}_i}$ and $\mathbf{Q}_{\boldsymbol{\beta}_i\boldsymbol{\beta}_i}$ of the spatial part $\mathbf{y}_i$ and trend part $\boldsymbol{\beta}_i$.

Remark 6 *If the parameter vectors $\mathbf{p}_i$ can't be assumed independent and cross-covariance information has to be taken into account, then the proposed method remains applicable under mild assumptions on the cross-covariance structure. Often the off-diagonal blocks of $\mathbf{Q}_{\mathbf{PP}}$ have the same structure as the covariance matrices $\mathbf{Q}_{\mathbf{p}_i\mathbf{p}_i}$, e.g. they are invariant under translation as well, and the matrix multiplication methods of section 2.1.2 may be applied to each block of $\mathbf{Q}_{\mathbf{PP}}$ individually. As a consequence, methods that only require multiplication with $\mathbf{Q}_{\mathbf{PP}}$ like the central result of section 2.3 can still be applied. However, methods that rely on the multiplication with $\mathbf{Q}_{\mathbf{PP}}^{-1}$ may become unfeasible, since this is a global operation in the sense that it mixes contributions from all the matrix blocks.*

Also note that the approach proposed for the linearized uncertainty quantification and a posteriori analysis, sections 2.5 and 2.6, is no longer applicable under these conditions, since the same reasoning applies to $\mathbf{Q}_{\mathbf{PP}}^{1/2}$. These two issues may potentially be handled by ignoring the block structure and performing a spectral decomposition of $\mathbf{Q}_{\mathbf{PP}}$, see e.g. remark 5, but then the applicability of the methods depends on the number of eigenvectors that are needed to adequately represent the cross-covariance structure.

Furthermore, we assume that the state observations are also random variables. While the result of the measurements is fixed in theory due to $\mathbf{Z} = \mathcal{G}(\mathbf{P})$, in reality the values of the measurements will fluctuate due to measurement errors. The measurement process will therefore yield $\mathbf{Z} = \mathcal{G}(\mathbf{P}) + \epsilon$ with some noise ϵ. A general formulation for the measurement error ϵ isn't available, since its distribution depends on the specifics of the accuracy and biasedness of the measurement process. The most basic approach neglects potential bias and models the measurement error as another Gaussian random variable:

$$\epsilon = [\mathbf{Z} - \mathcal{G}(\mathbf{P})] \sim \mathcal{N}(\mathbf{Q}_{\mathbf{ZZ}}, \mathbf{0}), \tag{2.25}$$

where $\mathbf{Q}_{\mathbf{ZZ}}$ is the covariance matrix of the measurement errors, again assumed to be symmetric positive definite. Although individual measurement errors may be correlated, the matrix $\mathbf{Q}_{\mathbf{ZZ}}$ is typically a diagonal matrix in practice, unless more detailed information about the correlation structure is available [17, 56]. The diagonal entries of the matrix contain the variance of the observations and therefore quantify the measurement uncertainty. Equation (2.25) states that, given a fixed instance of parameters $\mathbf{P}$, the observed quantities are normally distributed around the model outcome:

$$\mathbf{Z}|\mathbf{P} \sim \mathcal{N}(\mathbf{Q}_{\mathbf{ZZ}}, \mathcal{G}(\mathbf{P})). \tag{2.26}$$

2.2.1 Bayesian Inverse Problem

Under the assumptions of the previous section we may invoke Bayes' Theorem, which states:

$$f_{\mathbf{Z}} \cdot f_{\mathbf{P}|\mathbf{Z}} = f_{\mathbf{P}} \cdot f_{\mathbf{Z}|\mathbf{P}}. \tag{2.27}$$

For a given tuple of measurements $\mathbf{Z}$ used for parameter estimation the expression $f_{\mathbf{Z}}$ is a constant, and equation (2.27) states that

$$f_{\mathbf{P}|\mathbf{Z}} \propto f_{\mathbf{P}} \cdot f_{\mathbf{Z}|\mathbf{P}}, \tag{2.28}$$

i.e. the posterior probability is proportional to both the prior probability $f_{\mathbf{P}}$ and the likelihood $f_{\mathbf{Z}|\mathbf{P}}$.

Since the terms on the righthand side have been defined in equations (2.23) and (2.26), we have

$$f_{\mathbf{P}} \propto \exp\left(-\frac{1}{2}\left\|\mathbf{P}-\mathbf{P}^*\right\|^2_{\mathbf{Q}^{-1}_{\mathbf{PP}}}\right) \tag{2.29}$$

and

$$f_{\mathbf{Z}|\mathbf{P}} \propto \exp\left(-\frac{1}{2}\left\|\mathbf{Z}-\mathcal{G}(\mathbf{P})\right\|^2_{\mathbf{Q}^{-1}_{\mathbf{ZZ}}}\right), \tag{2.30}$$

with

$$\|v\|_A \coloneqq \left[v^T A v\right]^{1/2} \tag{2.31}$$

being the notation for the norm induced by the weighted scalar product defined for any symmetric positive definite matrix A, and therefore

$$f_{\mathbf{P}|\mathbf{Z}} \propto \exp\left(-\frac{1}{2}\left[\left\|\mathbf{P}-\mathbf{P}^*\right\|^2_{\mathbf{Q}^{-1}_{\mathbf{PP}}} + \left\|\mathbf{Z}-\mathcal{G}(\mathbf{P})\right\|^2_{\mathbf{Q}^{-1}_{\mathbf{ZZ}}}\right]\right). \tag{2.32}$$

This is an expression that describes, at least up to an unknown constant, the posterior distribution of $\mathbf{P}$ given $\mathbf{Z}$. As such, it assigns a trust value to each possible parameter tuple $\mathbf{P}$, since the above function is a measure for the plausibility of a specific $\mathbf{P}$ causing the observation of a specific $\mathbf{Z}$. Using equation (2.32), we can formulate the following inverse problem to replace problem 2:

Problem 4 (Bayesian Inverse Problem)
Given a tuple of measurements $\mathbf{Z}$, a model $\mathcal{F}$ and operators $\mathcal{I}$ and $\mathcal{O}$ as before, and in addition a prior probability distribution for $\mathbf{P}$ and a conditional distribution $\mathbf{Z}|\mathbf{P}$, determine the posterior probability distribution of $\mathbf{P}$. To do that, apply equation (2.32) *and renormalize the result to arrive at a function with integral one.*

This inverse problem fulfills the well-posedness criteria on page 13 in the following sense:

- Apart from assumptions about the stochastic structure no artificial restrictions are introduced. The given distributions may be replaced by more appropriate choices, and the problem formulation is therefore rather general.
- Since the objective of the inverse problem has been redefined as assigning a function value to each $\mathbf{P}$ instead of returning a single $\mathbf{P}$, a unique solution exists for every $\mathbf{Z}$ as long as the posterior distribution is defined.
- Small perturbations in the observations only lead to small changes in the assigned values, and therefore the inversion process is stable.
- The original ill-posed problem 2 can be seen as the limit case of uniform prior distribution of $\mathbf{P}$ and infinitesimally small measurement errors. The reformulated inverse problem is therefore a direct extension of the original one.

It should be noted that the inverse problem may remain ill-posed even after reformulation if the function defined in equation (2.32) does not have a finite integral. Such technicalities aside, problem 4 may be solved to obtain the mentioned probability distribution, with the inversion result being those parameter tuples $\mathbf{P}$ with a comparatively large PDF value, and the uncertainty of the parameter estimate corresponding to the size of that area.

Remark 7 *At this point we may revisit the groundwater flow examples from page 8. The first setup, while not allowing a solution in the sense of problem 2, will result in a posterior distribution that is symmetric around the line formed by the measurement wells, reflecting the symmetry in the problem statement. In the case of noninformative tracer measurements, the likelihood will always be zero, since the model outcome does not depend on the parameters. The posterior distribution is then equal to the prior distribution, reflecting that no additional information was gained through the measurements. Inconsistent or physically impossible measurement values, as in the third example, will simply assign a higher probability to those parameter tuples that result in measurement values that are close to the given ones, even though they can't be reproduced exactly.*

Solving the inverse problem in the strict sense is not possible, since the PDF typically doesn't allow for a closed formulation and has to be evaluated in an uncountable number of points, but finite approximations known as surrogate models may be generated. The task of assembling the full PDF may also be reduced to computing or estimating some of the moments of the random variable $\mathbf{P}|\mathbf{Z}$, as described in the next section.

2.2.2 Maximum A Posteriori

There are several possible candidates for a point parameter estimate based on the posterior PDF, the two most important being the mean of the distribution and the

mode of the PDF, i.e. the point where the PDF has its maximum value. Higher order moments, like the variance, may also be estimated to arrive at a better representation of the PDF and also acquire information about the uncertainty of the point estimate. One may sample from the distribution to compute its mean and variance, but this may again be too expensive if the cost for evaluations of equation (2.32) is too high, see chapter 3. Computing the mode, also called the Maximum A Posteriori point (MAP), can be significantly cheaper. MAP estimation searches for the maximizer $\mathbf{P}_{\text{map}}$ of $f_{\mathbf{P}|\mathbf{Z}}$, which is equivalent to the minimizer $\arg\min_{\mathbf{P}} L(\mathbf{P})$ of the objective function

$$L(\mathbf{P}) \coloneqq \frac{1}{2} \left\| \mathbf{P} - \mathbf{P}^* \right\|^2_{\mathbf{Q}^{-1}_{\mathbf{PP}}} + \frac{1}{2} \left\| \mathbf{Z} - \mathcal{G}(\mathbf{P}) \right\|^2_{\mathbf{Q}^{-1}_{\mathbf{ZZ}}}, \tag{2.33}$$

compare equation (2.32). With this objective function, the inverse problem in the context of MAP estimation can be defined in the following way:

Problem 5 (Maximum A Posteriori Inverse Problem)
Given a tuple of measurements $\mathbf{Z}$*, a model* $\mathcal{F}$ *and operators* $\mathcal{I}$ *and* $\mathcal{O}$ *as before, and in addition a prior probability distribution for* $\mathbf{P}$ *and a conditional distribution* $\mathbf{Z}|\mathbf{P}$*, determine the parameter tuple* $\mathbf{P}_{map}$ *that maximizes the posterior PDF of* $\mathbf{P}$ *given* $\mathbf{Z}$*, i.e. find the minimizer* $\arg\min_{\mathbf{P}} L(\mathbf{P})$ *of the objective function given in equation* (2.33)*.*

The MAP estimate may be compared to the Maximum Likelihood (ML) estimate [69], which tries to maximize the likelihood $f_{\mathbf{Z}|\mathbf{P}}$ and therefore minimize the norm

$$\widetilde{L}(\mathbf{P}) \coloneqq \frac{1}{2} \left\| \mathbf{Z} - \mathcal{G}(\mathbf{P}) \right\|_{\mathbf{Q}^{-1}_{\mathbf{ZZ}}}. \tag{2.34}$$

In contrast to ML, MAP estimation also incorporates a priori information about the parameters through the term $\left\| \mathbf{P} - \mathbf{P}^* \right\|_{\mathbf{Q}^{-1}_{\mathbf{PP}}}$, and in this sense it can be seen as a regularization of the Maximum Likelihood approach [69]. The differences between MAP and ML estimation are considered in more detail in chapter 3.

Problem 5 has the structure of a Least Squares problem [48]. Such problems are easy to solve if $\mathcal{G}$ is linear, since this implies a convex minimization problem [11], with a unique global minimum that may be found using standard techniques. Unfortunately, the model $\mathcal{G}$ is almost never linear, even if it consists of linear PDEs [25]. Since almost all optimization schemes are local searches, they may stagnate in local minima far away from the global minimum if the initial guess is not close enough to the solution.

Remark 8 *While the MAP estimate is often significantly easier to determine, the posterior mean may be a better representative for the posterior PDF [9]. The MAP point is only determined through the maximum value of the PDF and not through an integration process like the mean, and can therefore be influenced more easily by outliers of the distribution. In the case of multimodal distributions, the MAP estimate may be non-unique and also uncharacteristic for the distribution as a whole. In the case of the first example on page 8 there are at least two extremal*

points, and iterative schemes may arbitrarily converge to one of them or even stagnate due to the symmetry of the objective function. In the case of the second example, the MAP estimate is equal to the mean, since the prior distribution is unimodal and symmetric. See section 2.6 for techniques that may help in detecting MAP estimates that are of poor quality.

2.3 Preconditioned Conjugate Gradients

One of the numerical schemes that may be used to solve nonlinear Least Squares problems such as problem 5 is the Conjugate Gradients (CG) method, an extension of the method of Steepest Descent [73, 40]. Steepest Descent uses the negative of the gradient of the objective function as a step direction for optimization, in the case of $L(\mathbf{P})$ from equation (2.33) therefore

$$-\nabla L = -\mathbf{Q}_{\mathbf{PP}}^{-1}\left[\mathbf{P}-\mathbf{P}^{*}\right] + \mathbf{H}_{\mathbf{ZP}}^{T}\mathbf{Q}_{\mathbf{ZZ}}^{-1}\left[\mathbf{Z}-\mathcal{G}(\mathbf{P})\right], \tag{2.35}$$

where $\mathbf{H_{ZP}}$ is called the sensitivity matrix and contains the derivatives of the measurements with regard to the parameters, i.e.

$$\begin{aligned}(\mathbf{H}_{\mathbf{ZP}})_{i,j} &\coloneqq \mathbf{H}_{\mathbf{z}_i\mathbf{p}_j} \\ (\mathbf{H}_{\mathbf{z}_i\mathbf{p}_j})_{k,l} &\coloneqq \partial_{(\mathbf{p}_j)_l}(\mathbf{z}_i)_k.\end{aligned} \tag{2.36}$$

A more detailed formulation in analogy to equation (2.24) is

$$\mathbf{H}_{\mathbf{ZP}} = \begin{pmatrix} \mathbf{H}_{\mathbf{z}_1\mathbf{p}_1} & \mathbf{H}_{\mathbf{z}_1\mathbf{p}_2} & \cdots & \mathbf{H}_{\mathbf{z}_1\mathbf{p}_{n_\mathbf{P}}} \\ \mathbf{H}_{\mathbf{z}_2\mathbf{p}_1} & \mathbf{H}_{\mathbf{z}_2\mathbf{p}_2} & \cdots & \mathbf{H}_{\mathbf{z}_2\mathbf{p}_{n_\mathbf{P}}} \\ \vdots & \vdots & & \vdots \\ \mathbf{H}_{\mathbf{z}_{n_\mathbf{Z}}\mathbf{p}_1} & \mathbf{H}_{\mathbf{z}_{n_\mathbf{Z}}\mathbf{p}_2} & \cdots & \mathbf{H}_{\mathbf{z}_{n_\mathbf{Z}}\mathbf{p}_{n_\mathbf{P}}} \end{pmatrix} \tag{2.37}$$

together with

$$\mathbf{H}_{\mathbf{z}_i\mathbf{p}_j} = \begin{pmatrix} \partial_{(\mathbf{p}_j)_1}(\mathbf{z}_i)_1 & \partial_{(\mathbf{p}_j)_2}(\mathbf{z}_i)_1 & \cdots & \partial_{(\mathbf{p}_j)_{n_{\mathbf{p}_j}}}(\mathbf{z}_i)_1 \\ \partial_{(\mathbf{p}_j)_1}(\mathbf{z}_i)_2 & \partial_{(\mathbf{p}_j)_2}(\mathbf{z}_i)_2 & \cdots & \partial_{(\mathbf{p}_j)_{n_{\mathbf{p}_j}}}(\mathbf{z}_i)_1 \\ \vdots & \vdots & & \vdots \\ \partial_{(\mathbf{p}_j)_1}(\mathbf{z}_i)_{n_{\mathbf{z}_i}} & \partial_{(\mathbf{p}_j)_2}(\mathbf{z}_i)_{n_\mathbf{z}} & \cdots & \partial_{(\mathbf{p}_j)_{n_{\mathbf{p}_j}}}(\mathbf{z}_i)_{n_{\mathbf{z}_i}} \end{pmatrix} \tag{2.38}$$

for each possible combination of parameter vector $\mathbf{p}_i$ and measurement vector $\mathbf{z}_j$, where again $(\cdot)_k$ denotes the kth component of a given vector. By construction $\mathbf{H_{ZP}}$ is the linearization of the discrete model $\mathcal{G}(\mathbf{P})$ around the point where the gradient ∇L is computed.

Algorithm 2: Nonlinear Steepest Descent (**SD**)

Input: initial value $\mathbf{P}_0$, stopping criterion
Output: estimate of MAP point $\mathbf{P}_{\text{map}}$
$i \coloneqq 0$ [set index];
repeat
  $i \to i+1$ [shift index];
  $\mathbf{R}_i \coloneqq -\nabla L|_{\mathbf{P}_{i-1}}$ [compute residual];
  $\mathbf{D}_i \coloneqq \mathbf{R}_i$ [set direction];
  $\alpha_i \coloneqq \text{linesearch}(\mathbf{P}_{i-1}, \mathbf{D}_i)$ [compute step width];
  $\mathbf{P}_i \coloneqq \mathbf{P}_{i-1} + \alpha_i \mathbf{D}_i$ [define i-th iteration];
until *converged*;
$\mathbf{P}_{\text{map}} \coloneqq \mathbf{P}_i$ [accept final iteration];
return $\mathbf{P}_{map}$;

2.3.1 Steepest Descent

Starting from an initial guess $\mathbf{P}_0$, the method of Steepest Descent generates a series of iterations $\mathbf{P}_i$, step directions $\mathbf{D}_i$ and step widths α_i using algorithm 2 (**SD**). The stopping criterion is usually a certain reduction of the norm of the gradient, e.g. by a factor of 10^3 or 10^4, combined with a maximum number of steps that should be taken if this reduction can't be reached. The algorithm may also be stopped if it stagnates and the low reduction of L from step to step suggests a bad initial guess $\mathbf{P}_0$.

If the model $\mathcal{G}$ were linear, the residual could be updated using the previous iteration instead of being recomputed, and the optimal step width would be known. The objective function would be of the form

$$L(\mathbf{P}) = \frac{1}{2}\mathbf{P}^T\mathbf{A}\mathbf{P} - \mathbf{B}^T\mathbf{P}, \tag{2.39}$$

with a symmetric positive definite matrix $\mathbf{A}$ and a vector $\mathbf{B}$, and the optimal step width would be defined by

$$\alpha_{i,\text{opt}} \coloneqq \frac{\mathbf{R}_i^T\mathbf{R}_i}{\mathbf{D}_i^T\mathbf{A}\mathbf{D}_i}. \tag{2.40}$$

Since, as mentioned above, the model must be assumed to be nonlinear, the residual has to be assembled in each iteration, and some sort of line search is necessary to find the optimal step width. Due to the structure of the objective function (2.33), it is natural to approximate it locally with a quadratic polynomial, and doing this along the search direction to optimize the step width is known as Quadratic Line Search:

$$L(\mathbf{P}_{i-1} + \alpha\mathbf{E}_i) \approx a \cdot \alpha^2 + b \cdot \alpha + L(\mathbf{P}_{i-1}), \tag{2.41}$$

where $\mathbf{E}_i$ is the unit vector in direction $\mathbf{D}_i$ and a and b are constants that need to be determined. Given the function values $L_{i;0} \coloneqq L(\mathbf{P}_{i-1})$ and $L_{i;1} \coloneqq L(\mathbf{P}_{i-1} + \epsilon\mathbf{E}_i)$

for a potential step width ϵ, we may either evaluate the additional value $L_{i;1/2} \coloneqq L(\mathbf{P}_{i-1} + \frac{\epsilon}{2}\mathbf{E}_i)$ and set

$$\begin{aligned} a &\coloneqq 2\left[L_{i;1} - 2L_{i;1/2} + L_{i;0}\right] \cdot \epsilon^{-2} \\ b &\coloneqq -\left[L_{i;1} - 4L_{i;1/2} + 3L_{i;0}\right] \cdot \epsilon^{-1}, \end{aligned} \tag{2.42}$$

or use the fact that $-\left[\mathbf{R}_i \cdot \mathbf{E}_i\right]$ is the directional derivative of L in search direction and set

$$\begin{aligned} a &\coloneqq \left[L_{i;1} - L_{i;0} - b \cdot \epsilon\right] \cdot \epsilon^{-2} \\ b &\coloneqq -\left[\mathbf{R}_i \cdot \mathbf{E}_i\right]. \end{aligned} \tag{2.43}$$

The first of the two options is more robust, while the second reuses the gradient and as a result requires one function evaluation less. Both approaches allow the calculation of the new step width and an estimate of the new objective function value through the coordinates of the vertex of the parabola defined by (2.41), namely

$$\left(-\frac{b}{2a}, L_{i;0} - \frac{b^2}{4a}\right). \tag{2.44}$$

If necessary, the line search may be repeated by setting ϵ to $-\frac{b}{2a}$ and recalculating a and b to improve the results. Note that the resulting step width is in terms of the unit direction $\mathbf{E}_i$ and must be multiplied by $\|\mathbf{D}_i\|_2^{-1}$ if it is needed in terms of $\mathbf{D}_i$. Initial guesses ϵ may be generated through linear extrapolation combined with a trust region approach to keep the parameters within a few standard deviations around the current estimate. After the first step the previous step width may be used as an initial guess.

2.3.2 Conjugate Gradients

The method of Steepest Descent produces locally optimal step directions, but the scheme may display oscillatory behavior and slow convergence [73]. The Conjugate Gradients (CG) method augments the direction given in equation (2.35) with a correction term designed to speed up the convergence. Using the additional initial values $\mathbf{R}_0 \coloneqq 0$ and $\mathbf{D}_0 \coloneqq 0$, the CG method is given in algorithm 3 (**CG**).

In the linear case, the optimal step width is again given by equation (2.40), and the conjugation factor β_i is

$$\beta_i \coloneqq \frac{\mathbf{R}_i^T \mathbf{R}_i}{\mathbf{R}_{i-1}^T \mathbf{R}_{i-1}}. \tag{2.45}$$

It can be shown that this results in a Krylov subspace method, and that all directions $\mathbf{D}_i$ are pairwise $\mathbf{A}$-orthogonal, i.e.

$$\forall\, i\, \forall\, j \neq i \colon \mathbf{D}_i^T \mathbf{A} \mathbf{D}_j = 0. \tag{2.46}$$

Algorithm 3: Nonlinear Conjugate Gradients (**CG**)

Input: initial value $\mathbf{P}_0$, $\mathbf{R}_0 = 0$, $\mathbf{D}_0 = 0$, stopping criterion
Output: estimate of MAP point $\mathbf{P}_{\text{map}}$
$i \coloneqq 0$ [set index];
repeat
  $i \to i + 1$ [shift index];
  $\mathbf{R}_i \coloneqq -\nabla L|_{\mathbf{P}_{i-1}}$ [compute residual];
  $\beta_i \coloneqq \text{orthogonalize}(\mathbf{R}_{i-1}, \mathbf{R}_i, \mathbf{D}_{i-1})$ [compute conjugation factor];
  $\mathbf{D}_i \coloneqq \mathbf{R}_i + \beta_i \mathbf{D}_{i-1}$ [set direction];
  $\alpha_i \coloneqq \text{linesearch}(\mathbf{P}_{i-1}, \mathbf{D}_i)$ [compute step width];
  $\mathbf{P}_i \coloneqq \mathbf{P}_{i-1} + \alpha_i \mathbf{D}_i$ [define i-th iteration];
until *converged*;
$\mathbf{P}_{\text{map}} \coloneqq \mathbf{P}_i$ [accept final iteration];
return $\mathbf{P}_{map}$;

This distinguishes the CG method from Steepest Descent, as the latter produces directions that are only orthogonal with respect to the standard coordinate system, not to one tailored to the bilinear form of the objective function, and only subsequent directions are guaranteed to be orthogonal. These two points are the theoretical foundation for the increased rate of convergence of algorithm 3 (**CG**) when compared with algorithm 2 (**SD**). One can show that the rate of convergence of Steepest Descent in the energy norm is bounded by

$$k_{\text{SD}}(\mathbf{A}) \coloneqq \frac{\kappa(\mathbf{A}) - 1}{\kappa(\mathbf{A}) + 1}, \tag{2.47}$$

where

$$\kappa(\mathbf{A}) \coloneqq \frac{\lambda_{\max}(\mathbf{A})}{\lambda_{\min}(\mathbf{A})} \tag{2.48}$$

is the spectral condition number of $\mathbf{A}$, the quotient of its largest and its smallest eigenvalue, while the rate of convergence of the CG method is bounded by

$$k_{\text{CG}}(\mathbf{A}) \coloneqq \frac{\kappa(\mathbf{A})^{1/2} - 1}{\kappa(\mathbf{A})^{1/2} + 1}. \tag{2.49}$$

These bounds are comparatively sharp, and for large $\kappa(\mathbf{A})$ the CG method can be orders of magnitude faster.

Since $\mathcal{G}$ is nonlinear, the above is not directly applicable in the given situation, and equation (2.45) is only one possible choice for the conjugation factor. Popular choices for β_i from the literature [35], all reducing to equation (2.45) in the linear case, are the Fletcher-Reeves formula [29]

$$\beta_i^{\text{FR}} \coloneqq \frac{\mathbf{R}_i^T \mathbf{R}_i}{\mathbf{R}_{i-1}^T \mathbf{R}_{i-1}}, \tag{2.50}$$

the Polak-Ribière formula [64]

$$\beta_i^{\mathrm{PR}} \coloneqq \frac{\mathbf{R}_i^T \left[\mathbf{R}_i - \mathbf{R}_{i-1}\right]}{\mathbf{R}_{i-1}^T \mathbf{R}_{i-1}} \tag{2.51}$$

and the Hestenes-Stiefel formula [40]

$$\beta_i^{\mathrm{HS}} \coloneqq -\frac{\mathbf{R}_i^T \left[\mathbf{R}_i - \mathbf{R}_{i-1}\right]}{\mathbf{D}_{i-1}^T \left[\mathbf{R}_i - \mathbf{R}_{i-1}\right]}. \tag{2.52}$$

In the following, we will default to using β_i^{PR} from equation (2.51).

2.3.3 Preconditioning

The Conjugate Gradients method is a significant improvement over the simple Steepest Descent, and the top row of figure 2.2 gives an example of the improvement in convergence speed that may be achieved. Unfortunately the scheme still has a strong dependence on the spectral condition and often slows down when the resolution is increased, see the upper right corner of figure 2.2 for example. This form of the algorithm may therefore quickly become unfeasible if the number of discretization cells n_Ω becomes too large.

Preconditioning can be used to transform the spectrum of the involved operators and further increase the convergence rate. This technique is based on a transformation of the underlying space of the objective function $L(\mathbf{P})$. Let $\mathbf{E}$ be an invertible matrix, then L may also be written as a function of $\widetilde{\mathbf{P}} \coloneqq \mathbf{E}\mathbf{P}$, which leads to

$$\widetilde{L}(\widetilde{\mathbf{P}}) = \left\| \widetilde{\mathbf{P}} - \widetilde{\mathbf{P}}^* \right\|^2_{[\mathbf{E}^{-1}]^T \mathbf{Q}_{\mathbf{PP}}^{-1} \mathbf{E}^{-1}} + \left\| \mathbf{Z} - \mathcal{G}(\mathbf{E}^{-1}\widetilde{\mathbf{P}}) \right\|^2_{\mathbf{Q}_{\mathbf{ZZ}}^{-1}}, \tag{2.53}$$

where $\widetilde{\mathbf{P}}^* \coloneqq \mathbf{E}\mathbf{P}^*$. The correct choice of $\mathbf{E}$ can have drastic consequences for the convergence behavior of the method. Returning again to the linear case, the transformation of (2.39) is

$$\widetilde{L}(\widetilde{\mathbf{P}}) = \frac{1}{2}\widetilde{\mathbf{P}}^T \left[\left[\mathbf{E}^{-1}\right]^T \mathbf{A}\mathbf{E}^{-1} \right] \widetilde{\mathbf{P}} - \left[\mathbf{E}^{-1} b\right]^T \widetilde{\mathbf{P}}. \tag{2.54}$$

Since $\mathbf{A}$ is symmetric positive definite, we can choose $\mathbf{E}$ to be the positive root $\mathbf{A}^{1/2}$, and the equation simplifies to

$$\widetilde{L}(\widetilde{\mathbf{P}}) = \left[\frac{1}{2}\widetilde{\mathbf{P}} - \mathbf{A}^{-1/2}\mathbf{B} \right]^T \widetilde{\mathbf{P}} \tag{2.55}$$

with the obvious minimum $\mathbf{P}_{\mathrm{map}} = 2\mathbf{A}^{-1/2}\mathbf{B}$. Both the Steepest Descent method and the Conjugate Gradients method converge in one step for this transformed system, but in practice the construction of a matrix $\mathbf{E}$ with the right properties is as expensive

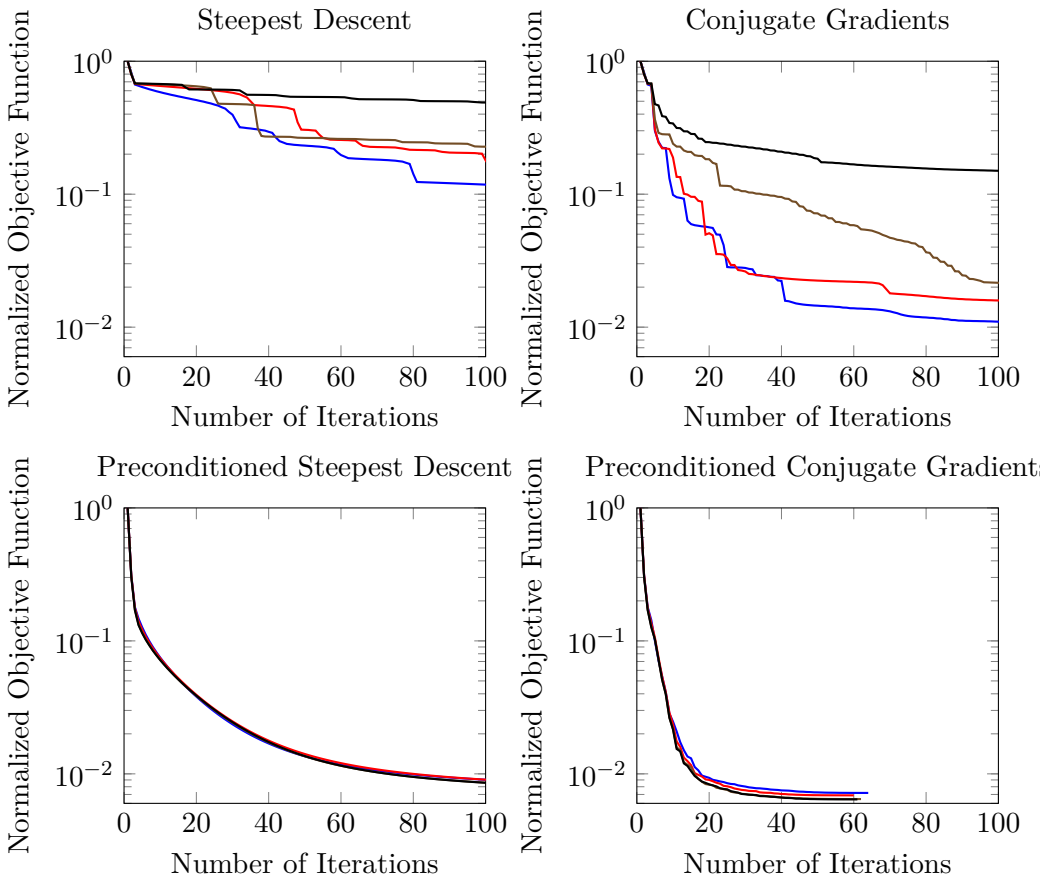

Figure 2.2: Convergence behavior of the described methods for parameters with exponential covariance function, *left:* Steepest Descent variants, *right:* Conjugate Gradients variants, *top:* original versions, *bottom:* preconditioned versions with $\mathbf{Q}_{\mathbf{PP}}^{-1}$ as preconditioner. Test case is a two dimensional groundwater problem similar to that discussed in section 6.1.1, with $n_\Omega = 64 \times 64$ (——), 128×128 (——), 256×256 (——) and 512×512 (——). Conjugate Gradients consistently converges faster than Steepest Descent. Without preconditioner the convergence behavior is erratic and alternates between stagnation and sudden jumps, while the descent of the preconditioned versions is significantly smoother. The unpreconditioned versions become slower with decreasing mesh width, while the preconditioned versions retain efficiency or even become slightly faster. *Not shown:* Additionally, iterations of the preconditioned versions are cheaper and may be significantly faster than those of the original versions, compare section 2.3.5.

Algorithm 4: Preconditioned Nonlinear Conjugate Gradients (**PCG**)

Input: initial value $\mathbf{P}_0$, $\mathbf{R}_0 = 0$, $\mathbf{T}_0 = 0$, $\mathbf{D}_0 = 0$, matrix $\mathbf{M}^{-1}$, stopping criterion
Output: estimate of MAP point $\mathbf{P}_{\text{map}}$
$i \coloneqq 0$ [set index];
repeat
  $i \to i + 1$; [shift index]
  $\mathbf{R}_i \coloneqq -\nabla L|_{\mathbf{P}_{i-1}}$; [compute residual]
  $\mathbf{T}_i \coloneqq \mathbf{M}^{-1}\mathbf{R}_i$ [compute preconditioned residual];
  $\beta_i \coloneqq \text{orthogonalize}(\mathbf{R}_{i-1}, \mathbf{T}_{i-1}, \mathbf{R}_i, \mathbf{T}_i, \mathbf{D}_{i-1})$ [compute conjugation factor];
  $\mathbf{D}_i \coloneqq \mathbf{T}_i + \beta_i \mathbf{D}_{i-1}$ [set direction];
  $\alpha_i \coloneqq \text{linesearch}(\mathbf{P}_{i-1}, \mathbf{D}_i)$ [compute step width];
  $\mathbf{P}_i \coloneqq \mathbf{P}_{i-1} + \alpha_i \mathbf{D}_i$ [define i-th iteration];
until *converged*;
$\mathbf{P}_{\text{map}} \coloneqq \mathbf{P}_i$ [accept final iteration];
return $\mathbf{P}_{map}$;

as optimizing (2.39) directly. Preconditioning therefore consists of finding a matrix $\mathbf{E}$ that is as close as possible to $\mathbf{A}^{1/2}$ but still cheap to compute, or equivalently a matrix $\mathbf{M}^{-1} \coloneqq \mathbf{E}^{-1}\left[\mathbf{E}^{-1}\right]^T$ that is as close as possible to $\mathbf{A}^{-1}$ while having low cost of assembly and application to vectors.

Applying the Conjugate Gradients method, algorithm 3 (**CG**), to the modified objective function (2.53) and transforming the result back to the original representation results in algorithm 4 (**PCG**). The conjugation factor formulas have to be modified to provide orthogonality in the transformed space and take the form

$$\beta_i^{\text{FR}} \coloneqq \frac{\mathbf{R}_i^T \mathbf{T}_i}{\mathbf{R}_{i-1}^T \mathbf{T}_{i-1}} \tag{2.56}$$

for the Fletcher-Reeves formula,

$$\beta_i^{\text{PR}} \coloneqq \frac{\mathbf{R}_i^T \left[\mathbf{T}_i - \mathbf{T}_{i-1}\right]}{\mathbf{R}_{i-1}^T \mathbf{T}_{i-1}} \tag{2.57}$$

for the Polak-Ribière formula and

$$\beta_i^{\text{HS}} \coloneqq -\frac{\mathbf{R}_i^T \left[\mathbf{T}_i - \mathbf{T}_{i-1}\right]}{\mathbf{D}_{i-1}^T \left[\mathbf{T}_i - \mathbf{T}_{i-1}\right]} \tag{2.58}$$

for the Hestenes-Stiefel formula. These more general definitions reduce to the ones given above for the special choice $\mathbf{M}^{-1} = \mathbf{I}$, where $\mathbf{I}$ is again the identity matrix, since this implies $\mathbf{T}_i = \mathbf{R}_i$. For completeness we also introduce the preconditioned Steepest Descent method in the form of algorithm 5 (**PSD**), which is the result of performing the above steps for algorithm 2 (**SD**) instead of algorithm 3 (**CG**).

Algorithm 5: Preconditioned Nonlinear Steepest Descent (**PSD**)

Input: initial value $\mathbf{P}_0$, matrix $\mathbf{M}^{-1}$, stopping criterion
Output: estimate of MAP point $\mathbf{P}_{\text{map}}$
$i \coloneqq 0$ [set index];
repeat
  $i \to i+1$ [shift index];
  $\mathbf{R}_i \coloneqq -\nabla L|_{\mathbf{P}_{i-1}}$ [compute residual];
  $\mathbf{T}_i \coloneqq \mathbf{M}^{-1}\mathbf{R}_i$ [compute preconditioned residual];
  $\mathbf{D}_i \coloneqq \mathbf{T}_i$ [set direction];
  $\alpha_i \coloneqq \text{linesearch}(\mathbf{P}_{i-1}, \mathbf{D}_i)$ [compute step width];
  $\mathbf{P}_i \coloneqq \mathbf{P}_{i-1} + \alpha_i \mathbf{D}_i$ [define i-th iteration];
until *converged*;
$\mathbf{P}_{\text{map}} \coloneqq \mathbf{P}_i$ [accept final iteration];
return $\mathbf{P}_{map}$;

As a preconditioner we propose the inverse of the prior covariance matrix $\mathbf{M} \coloneqq \mathbf{Q}_{\mathbf{PP}}^{-1}$, which leads to $\mathbf{E} = \mathbf{Q}_{\mathbf{PP}}^{-1/2}$, $\mathbf{M}^{-1} = \mathbf{Q}_{\mathbf{PP}}$ and the preconditioned residual

$$\begin{aligned} \mathbf{T}_i &= -\mathbf{Q}_{\mathbf{PP}} \nabla L|_{\mathbf{P}_{i-1}} \\ &= -\left[\mathbf{P}_{i-1} - \mathbf{P}^*\right] + \mathbf{Q}_{\mathbf{PP}} \mathbf{H}_{\mathbf{ZP}}^T \mathbf{Q}_{\mathbf{ZZ}}^{-1} \left[\mathbf{Z} - \mathcal{G}(\mathbf{P})\right] \end{aligned} \tag{2.59}$$

in the above algorithms. There are several related reasons for this choice of preconditioner:

- The locally optimal preconditioner is the Hessian of the objective function, since this is the matrix of the linearized objective function:

$$\text{Hess}(L) \approx \mathbf{Q}_{\mathbf{PP}}^{-1} + \mathbf{H}_{\mathbf{ZP}}^T \mathbf{Q}_{\mathbf{ZZ}}^{-1} \mathbf{H}_{\mathbf{ZP}}, \tag{2.60}$$

  where we have neglected second order effects, i.e. the dependency of $\mathbf{H}_{\mathbf{ZP}}$ on $\mathbf{P}$. The matrix $\mathbf{Q}_{\mathbf{PP}}^{-1}$ is the part of the Hessian that is independent of the current iteration and therefore the part that is suitable as a traditional preconditioner matrix.

- The convergence properties of the CG method are strongly influenced by the spectrum of the linearized operator and especially the clustering of its eigenvalues [78]. Since the rank of the second term of equation (2.60) is bounded by that of $\mathbf{Q}_{\mathbf{ZZ}}^{-1}$, the linearization of the preconditioned system corresponds to a low-rank perturbation of the identity matrix [15], and the scheme may achieve mesh-independent convergence rates.

- The transformation with $\mathbf{E} = \mathbf{Q}_{\mathbf{PP}}^{-1/2}$ decorrelates the random variable $\mathbf{P}$, i.e. the components of $\widetilde{\mathbf{P}} = \mathbf{Q}_{\mathbf{PP}}^{-1/2}\mathbf{P}$ are identically distributed and uncorrelated. Since

the coupling between the different components is removed, the scheme is restricted to the modes introduced by the measurement part of the preconditioned residual. As a result, spurious high-frequency modes are removed from the iterations, and the reduced set of active eigenvalues improves convergence.

- The step directions of the preconditioned schemes are, at least up to the conjugation terms, identical to that of the Gauss-Newton scheme when neglecting contributions that are quadratic in $\mathbf{H_{ZP}}$, see equation (3.12). Since these contributions have a very low rank compared to the rest of the appearing terms, the step directions are relatively close to each other. The Gauss-Newton scheme is known for its good convergence, and closeness of the step directions may therefore positively influence the convergence of the Conjugate Gradients method.

- The step directions are also similar to the Kalman Filter update formula for the same reasons, see equation (3.8). The Kalman Filter update is known to be optimal for the case of linear models that link parameters and measurements that are both normally distributed. While the models are typically nonlinear in the given context, see page 24, this suggests improved convergence properties in cases where the nonlinearities are not too strong.

2.3.4 Convergence Behavior

Figure 2.2 shows the performance of the four algorithms introduced so far when applied to a stationary two-dimensional groundwater flow problem. $\mathbf{P}$ consists of a single parameter field representing the log-conductivity of the soil, with n_Ω spatial parameters and one trend parameter for the mean value of the parameter field. The number of discretization cells n_Ω varies in steps of four between $64^2 \approx 4.1 \cdot 10^3$ and $512^2 \approx 2.6 \cdot 10^5$. The log-conductivity is assumed to be normally distributed with exponential covariance, see equation (2.12), and all four algorithms start with a homogeneous representation for the parameter field.

The graphs display the value of the objective function as a function of the number of iterations performed, divided by the value at the start of the algorithm, for all chosen values of n_Ω. As can be seen in the upper left corner, the Steepest Descent method requires several hundred iterations even for small n_Ω and stagnates far from the minimum value for larger n_Ω. The Conjugate Gradients method, seen in the top right corner, fares better and achieves satisfactory results for small n_Ω, but also becomes significantly slower when the resolution of the numerical grid increases. The preconditioned versions, shown in the bottom left and bottom right corner, converge significantly faster than the unpreconditioned ones and have a smoother convergence process. Most importantly, for these two methods the convergence behavior is independent of the number of cells n_Ω used to discretize the parameter estimation problem.

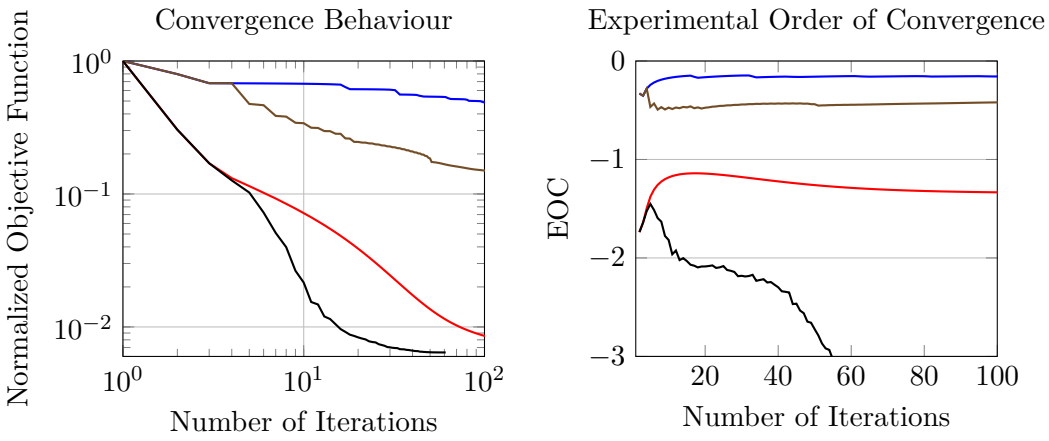

Figure 2.3: *Left:* Excerpt of the data presented in figure 2.2 for fixed numerical grid of size $n_\Omega = 512 \times 512$. *Right:* Corresponding experimental order of convergence in the objective function L with respect to its minimum value. ——: Steepest Descent, ——: Preconditioned Steepest Descent, ——: Conjugate Gradients, ——: Preconditioned Conjugate Gradients.

Under the assumption that the result $\mathbf{P}_{\text{map}}$ of the preconditioned Conjugate Gradients method is indeed $\arg\min_{\mathbf{P}} L(\mathbf{P})$, i.e. the method has converged to the global minimum, we may compute the experimental order of convergence (EOC) in the objective function values through

$$\text{EOC}(i) \coloneqq \frac{1}{\log(i+1)} \log\left(\frac{L(\mathbf{P}_i) - L(\mathbf{P}_{\text{map}})}{L(\mathbf{P}_0) - L(\mathbf{P}_{\text{map}})}\right). \tag{2.61}$$

Figure 2.3 gives a direct comparison of the convergence behavior of the four algorithms for a fixed grid of size $n_\Omega = 512 \times 512$, on the left through the evolution of the values of the objective function, which is a different view on the data already presented in figure 2.2, and on the right through the results of the EOC computation. Only the preconditioned versions achieve convergence in an acceptable number of iterations, and the preconditioned Conjugate Gradient method outperforms all other variants by a wide margin.

Figure 2.4 shows the synthetic random field that was used and the results of the algorithms 3 (**CG**) and 4 (**PCG**). The unpreconditioned scheme stagnates far away from the optimum, since the sensitivity matrix $\mathbf{H_{ZP}}$ introduces information about necessary changes mainly in the direct vicinity of the measurement locations and the locations of external forcing. Preconditioning explicitly couples parameters that are correlated, and significantly increases the speed with which the information from the observations propagates into the domain. As a result, the method converges within a comparatively small number of iterations.

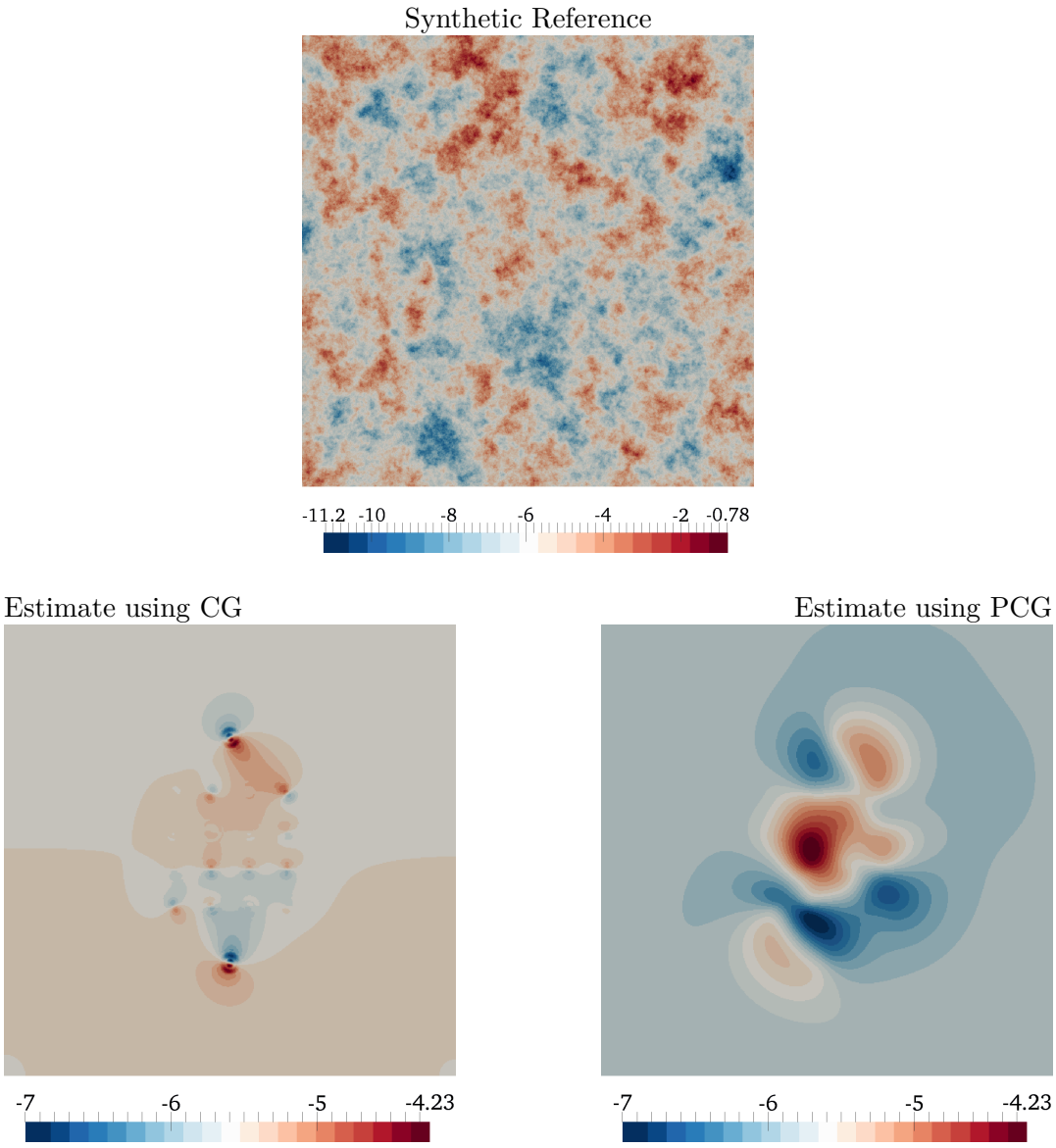

Figure 2.4: *Upper row:* Synthetic reference parameter field $\widehat{\mathbf{P}}$ with exponential covariance structure. *Lower left:* Approximate inversion result returned by the CG algorithm after 100 steps. Changes in the parameters are introduced locally at the injection well, extraction well and observation sites through the sensitivity of the system and then propagated into the rest of the domain through the covariance structure. Since this process is mainly restricted to the direct neighbors of the discretization elements, it becomes slower with increasing resolution. *Lower right:* Inversion result $\mathbf{P}_{\text{map}}$ of the PCG algorithm. Preconditioning transforms the spatially local step direction into a global one, significantly speeding up the convergence. Note that the inversion result is a smooth function and that the discrete stepping is only introduced to improve the contrast and ease visual interpretation.

Algorithm 6: Caching Prior Preconditioned Conjugate Gradients ($\mathbf{PCG}_c$)

Input: initial value $\mathbf{P}_0$, auxiliary variable $\mathbf{V}_0$, $\mathbf{R}_0 = 0$, $\mathbf{T}_0 = 0$, $\mathbf{D}_0 = 0$, $\mathbf{W}_0 = 0$, stopping criterion

Output: estimate of MAP point $\mathbf{P}_{\text{map}}$

$i \coloneqq 0$ [set index];

repeat

$i \to i + 1$ [shift index];

$(\mathbf{R}_i, \mathbf{T}_i) \coloneqq \left(-\nabla L|_{\mathbf{P}_{i-1}}, -\mathbf{Q}_{\mathbf{PP}} \nabla L|_{\mathbf{P}_{i-1}}\right)$ [compute residuals];

$\beta_i \coloneqq \text{orthogonalize}(\mathbf{R}_{i-1}, \mathbf{T}_{i-1}, \mathbf{R}_i, \mathbf{T}_i, \mathbf{D}_{i-1})$ [compute conjugation factor];

$(\mathbf{D}_i, \mathbf{W}_i) \coloneqq (\mathbf{T}_i, \mathbf{R}_i) + \beta_i (\mathbf{D}_{i-1}, \mathbf{W}_{i-1})$ [set directions];

$\alpha_i \coloneqq \text{linesearch}(\mathbf{P}_{i-1}, \mathbf{V}_{i-1}, \mathbf{D}_i, \mathbf{W}_i)$ [compute step width];

$(\mathbf{P}_i, \mathbf{V}_i) \coloneqq (\mathbf{P}_{i-1}, \mathbf{V}_{i-1}) + \alpha_i (\mathbf{D}_i, \mathbf{W}_i)$ [define i-th iterations];

until *converged*;

$\mathbf{P}_{\text{map}} \coloneqq \mathbf{P}_i$ [accept final iteration];

return $\mathbf{P}_{map}$;

2.3.5 Caching PCG Version

In addition to the drastic reduction in the number of iterations needed for convergence, the particular choice $\mathbf{M} = \mathbf{Q}_{\mathbf{PP}}^{-1}$ has consequences for the computational cost per iteration and the applicability of the method. Introducing the auxiliary variables $\mathbf{V}_i \coloneqq \mathbf{Q}_{\mathbf{PP}}^{-1}\mathbf{P}_i$ and $\mathbf{W}_i \coloneqq \mathbf{Q}_{\mathbf{PP}}^{-1}\mathbf{D}_i$ and observing that $\mathbf{R}_i = \mathbf{Q}_{\mathbf{PP}}^{-1}\mathbf{T}_i$, we may rewrite algorithm 3 (**CG**) to create two closely linked sequences, one consisting of $\mathbf{P}_i$, $\mathbf{T}_i$ and $\mathbf{D}_i$ and one consisting of $\mathbf{V}_i$, $\mathbf{R}_i$ and $\mathbf{W}_i$, as given in algorithm 6 ($\mathbf{PCG}_c$).

The second sequence is in theory redundant, since its elements can be computed from those of the original sequence through multiplication with $\mathbf{Q}_{\mathbf{PP}}^{-1}$, but this multiplication may be very expensive or numerically unstable, see section 2.1.2 and the discussion below. Storing this additional sequence allows the reformulation

$$\mathbf{R}_i = -\nabla L|_{\mathbf{P}_{i-1}} = -\left[\mathbf{V}_{i-1} - \mathbf{V}^*\right] + \mathbf{H}_{\mathbf{ZP}}\mathbf{Q}_{\mathbf{ZZ}}^{-1}\left[\mathbf{Z} - \mathcal{G}(\mathbf{P}_{i-1})\right] \tag{2.62}$$

with $\mathbf{V}^* \coloneqq \mathbf{Q}_{\mathbf{PP}}^{-1}\mathbf{P}^*$ and

$$\begin{aligned} \|\mathbf{P}_{i-1} + \alpha\mathbf{D}_i - \mathbf{P}^*\|_{\mathbf{Q}_{\mathbf{PP}}^{-1}} &= \left[\mathbf{P}_{i-1} + \alpha\mathbf{D}_i - \mathbf{P}^*\right]^T \left[\mathbf{V}_{i-1} + \alpha\mathbf{W}_i - \mathbf{V}^*\right] \\ &= \mathbf{P}_{i-1}^T\mathbf{V}_{i-1} + \alpha^2\mathbf{D}_i^T\mathbf{W}_i + \left[\mathbf{P}^*\right]^T\mathbf{V}^* \\ &\quad + 2\alpha\mathbf{W}_i^T\left[\mathbf{P}_{i-1} - \mathbf{P}^*\right] - 2\mathbf{V}_{i-1}^T\mathbf{P}^*, \end{aligned} \tag{2.63}$$

where $\mathbf{V}^*$ can be precomputed as in the case of $\mathbf{V}_0$ or dropped altogether if $\mathbf{P}^* = 0$. This allows the computation of the gradient and the evaluation of the objective function for the line search without application of $\mathbf{Q}_{\mathbf{PP}}^{-1}$. Since this also holds for the different formulas for the conjugation factors β_i, algorithm 6 ($\mathbf{PCG}_c$) can be executed without a single multiplication with $\mathbf{Q}_{\mathbf{PP}}^{-1}$, as long as $\mathbf{V}^*$ and $\mathbf{V}_0$ are known.

This is automatically the case if all spatial parameters in $\mathbf{P}_0$ respectively $\mathbf{P}^*$ are zero and the trend parameters are not correlated with the spatial parameters. Otherwise, a single multiplication with $\mathbf{Q}_{\mathbf{PP}}^{-1}$ is necessary to generate the second initial value $\mathbf{V}_0$ respectively $\mathbf{V}^*$ for the algorithm.

Storing the entries of $\mathbf{V}_i$ and $\mathbf{W}_i$ requires additional memory, but there are at least three reasons for doing so:

- The covariance matrix $\mathbf{Q}_{\mathbf{PP}}$ is an $n_{\mathbf{P}} \times n_{\mathbf{P}}$ block matrix with large and often dense submatrices. If there is only a single parameter vector or the parameter vectors $\mathbf{p}_i$ are uncorrelated, i.e. $\mathbf{Q}_{\mathbf{PP}}$ is a block diagonal matrix, then relatively efficient algorithms are available for the multiplication with $\mathbf{Q}_{\mathbf{PP}}^{-1}$, see section 2.1.2. Nevertheless, multiplication with $\mathbf{Q}_{\mathbf{PP}}^{-1}$ remains significantly more expensive than multiplication with $\mathbf{Q}_{\mathbf{PP}}$. This has the peculiar effect that the preconditioner has negative cost, since the multiplication with $\mathbf{Q}_{\mathbf{PP}}^{-1}$ for the line search is replaced by one with $\mathbf{Q}_{\mathbf{PP}}$ for the computation of the search direction.
- In reality the parameter vectors will most likely be correlated, and such information should be included in the prior covariance matrix $\mathbf{Q}_{\mathbf{PP}}$ if it is known. For $n_{\mathbf{P}} > 1$ the matrix may therefore contain nonzero off-diagonal blocks $\mathbf{Q}_{\mathbf{p}_i\mathbf{p}_j}, i \neq j$. The multiplication with $\mathbf{Q}_{\mathbf{PP}}$ may be performed as an iterative process consisting of multiplication with its constituent blocks, which can be achieved with the methods from section 2.1.2, but the multiplication with $\mathbf{Q}_{\mathbf{PP}}^{-1}$ is significantly more challenging and may become unfeasible even for a moderate number of discretization cells n_Ω.
- The matrix $\mathbf{Q}_{\mathbf{PP}}$ is symmetric positive definite and therefore guaranteed to be invertible in exact arithmetic. But if its spectrum decays to zero too fast, the multiplication with its inverse may be numerically unstable. While the prior covariance matrix should regularize the inverse problem, it introduces a secondary ill-posed inverse problem instead if multiplication with $\mathbf{Q}_{\mathbf{PP}}^{-1}$ is required. Preconditioning with $\mathbf{Q}_{\mathbf{PP}}$ and storing the auxiliary values removes this ill-conditioned operation. See below for such a situation.

Remark 9 *Note that this storage strategy is only possible for the preconditioned scheme with* $\mathbf{M} = \mathbf{Q}_{\mathbf{PP}}^{-1}$. *Neither the variant without preconditioner nor a version with a different preconditioner is compatible with this optimization. In both cases at least one multiplication with* $\mathbf{Q}_{\mathbf{PP}}^{-1}$ *is necessary for the line search. In principle, the transformation with* $\mathbf{Q}_{\mathbf{PP}}^{-1/2}$ *could be used instead and would also avoid multiplication with* $\mathbf{Q}_{\mathbf{PP}}^{-1}$, *but then two multiplications with* $\mathbf{Q}_{\mathbf{PP}}^{1/2}$ *would be required instead of one multiplication with* $\mathbf{Q}_{\mathbf{PP}}$. *We refer to remark 6 and section 2.7 in this regard.*

Figure 2.5 displays the convergence results for the same test case as figure 2.2, but with a Gaussian covariance function, equation (2.13), instead of exponential covariance. The parameter fields that follow this distribution are significantly smoother,

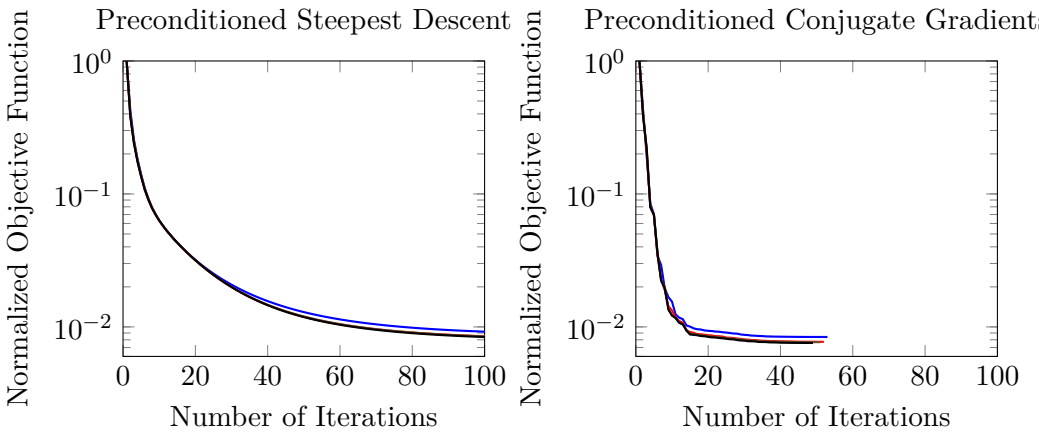

Figure 2.5: Convergence behavior of the described methods for parameters with Gaussian covariance function, *left:* Preconditioned Steepest Descent, *right:* Preconditioned Conjugate Gradients. Test case is the same as for figure 2.2 except for the covariance function. Unpreconditioned versions are missing, since they cannot perform due to the severely ill-conditioned multiplication with the inverse of the covariance matrix $\mathbf{Q_{PP}}$.

compare figures 2.4 and 2.6, which reflects the fact that the spectrum of the covariance matrix quickly decays to zero. As a result, the covariance matrix is effectively singular, and evaluating the objective function fails because the multiplication with $\mathbf{Q}_{\mathbf{PP}}^{-1}$ can't be carried out. Preconditioning with $\mathbf{Q_{PP}}$ removes exactly the high-frequency modes that are amplified by $\mathbf{Q}_{\mathbf{PP}}^{-1}$, restricting the iterations to the correct subspace of the parameter space, and the methods remain applicable.

2.4 Calculation of Sensitivities

One of the most expensive operations in the optimization schemes presented in the last section is typically the assembly of the sensitivity matrix $\mathbf{H_{ZP}}$. If the full matrix is assembled using the most basic algorithm, i.e. difference quotients created through the perturbation of single parameter values, then the cost for matrix assembly, and therefore also that for the term

$$\mathbf{H}_{\mathbf{ZP}}^{T}\mathbf{Q}_{\mathbf{ZZ}}^{-1}\left[\mathbf{Z}-\mathcal{G}(\mathbf{P})\right] \tag{2.64}$$

in equations (2.35) and (2.59), is asymptotically

$$T\left(n_{\mathbf{P}}, n_{\mathbf{Z}}, n_{\Omega}, n_{T}, n_{\boldsymbol{\beta}}, n_{\mathbf{z}}\right) = \mathcal{O}\left(\left[n_{\mathbf{P}} \cdot \left[n_{\Omega} + n_{\boldsymbol{\beta}}\right] \cdot \left[n_{\Omega} \cdot n_{T}\right] + n_{\mathbf{z}}\right] \cdot n_{\mathbf{Z}}\right), \tag{2.65}$$

where

$$n_{\boldsymbol{\beta}} \coloneqq \max_{1 \leq i \leq n_{\mathbf{P}}} n_{\boldsymbol{\beta}_i} \text{ and } n_{\mathbf{z}} \coloneqq \max_{1 \leq j \leq n_{\mathbf{Z}}} n_{\mathbf{z}_j}, \tag{2.66}$$

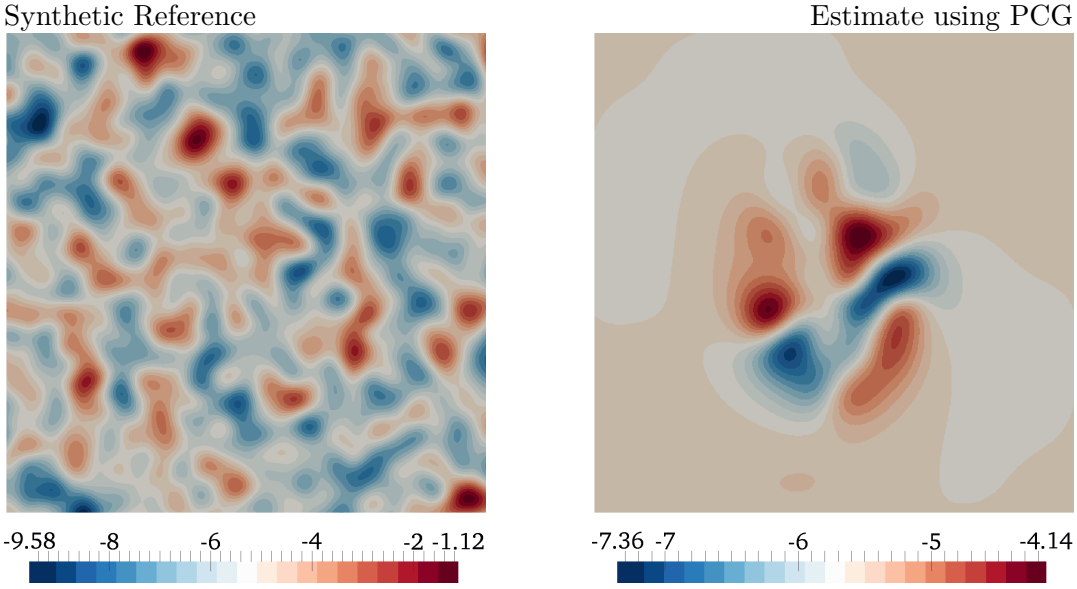

Figure 2.6: *Left:* Synthetic reference parameter field $\widehat{\mathbf{P}}$ with Gaussian covariance structure. In this case the eigenvalues of $\mathbf{Q_{PP}}$ corresponding to high-frequency modes are exceedingly small, and multiplication with $\mathbf{Q_{PP}^{-1}}$ is ill-conditioned. The unpreconditioned versions of the algorithms are therefore no longer applicable. *Right:* Preconditioning removes the ill-conditioned operation from the algorithms, and the methods are able to converge.

n_T is the number of time steps if the model formulation is transient and $n_T = 1$ else, and the rest of the numbers are defined as on page 2. The derivatives with regard to approximately $n_{\mathbf{P}} \cdot [n_\Omega + n_{\boldsymbol{\beta}}]$ parameters are needed, and each of these requires the solution of $n_{\mathbf{Z}}$ model equations, which can be obtained in $\mathcal{O}(n_\Omega \cdot n_T)$ if a PDE solver of optimal complexity is available. After these solutions are computed, approximately $n_{\mathbf{Z}} \cdot n_{\mathbf{z}}$ changes in the observations have to be evaluated to calculate the derivatives.

Remark 10 *Here we have assumed that a solver of optimal complexity, i.e. linear in n_Ω, is available, for example a Multigrid or Algebraic Multigrid (AMG) solver [13]. If such a solver can't be used for the given model equations, then the common factor $n_\Omega \cdot n_T \cdot n_{\mathbf{Z}}$ representing the effort of solving the forward problem has to be replaced by an appropriate expression in equation* (2.65) *and all following similar equations. This does not influence the validity of the argumentation. We also have assumed that the different models $\mathcal{F}_j$ are comparable in terms of effort that is needed to solve the forward problem. If this is not the case, then $n_{\mathbf{Z}}$ may be replaced by the number of states that are expensive to compute, again with no substantial consequences to the argumentation that follows. Also note that we have assumed that the computational cost of the observation operator $\mathcal{O}$ is independent of the mesh width.*

Equation (2.65) may be simplified using $n_{\boldsymbol{\beta}} \ll n_\Omega \cdot n_T$, $n_{\mathbf{z}} \ll n_\Omega \cdot n_T$ and the fact that extracting measurements from the system state is typically significantly cheaper than solving the model equations. This results in the asymptotic complexity

$$T\left(n_{\mathbf{P}}, n_{\mathbf{Z}}, n_\Omega, n_T, n_{\boldsymbol{\beta}}, n_{\mathbf{z}}\right) = \mathcal{O}\left(n_{\mathbf{P}} \cdot n_{\mathbf{Z}} \cdot n_\Omega^2 \cdot n_T\right). \tag{2.67}$$

Since this expression is quadratic in the number of elements n_Ω even under the assumption of an optimal solution strategy for the models, this approach is unfeasible for any realistic mesh width h. This section introduces adjoint states as a means of calculating the required derivatives in a more efficient way, reducing the complexity of assembling the gradient of the objective function $L(\mathbf{P})$ to $\mathcal{O}\left(n_{\mathbf{Z}} \cdot n_\Omega \cdot n_T\right)$, i.e. the complexity of solving the forward problem 1.

2.4.1 Lagrangian Formalism

We assume $\mathcal{F}_j \in L^2(\Omega)$ if the models are stationary and $\mathcal{F}_j \in L^2(\Omega \times T)$ if they are transient, and define the Lagrangian of the objective function $L(\mathbf{P})$ as

$$\mathcal{L}(\mathbf{P}, U, \Psi) = L(\mathbf{P}, U) + \sum_{j=1}^{n_{\mathbf{Z}}} \left\langle \psi_j, \mathcal{F}_j\left(S(\mathbf{P}), U_{<j}; u_j\right)\right\rangle, \tag{2.68}$$

where $\Psi \coloneqq (\psi_1, \ldots, \psi_{n_{\mathbf{Z}}})$. The ψ_j are known as Lagrange multipliers and are in $L^2(\Omega)$ for stationary models and in $L^2(\Omega \times T)$ for transient models, with $\langle \cdot, \cdot \rangle$ denoting the corresponding scalar product. Note that by definition the Lagrangian $\mathcal{L}(\mathbf{P}, U, \Psi)$ coincides with the objective function $L(\mathbf{P})$ for pairs $(\mathbf{P}, U)$ that are consistent.

Now consider a single parameter p, i.e. a component of one of the parameter vectors $\mathbf{p}_i$. The gradient of $L(\mathbf{P})$ consists of the derivatives of $L(\mathbf{P})$ with respect to all such parameters p. Assuming sufficient regularity of all appearing functions and operators, by the definition of the Lagrangian we have [63]

$$\begin{aligned} \mathrm{d}_p L = \mathrm{d}_p \mathcal{L} &= \sum_{k=1}^{n_{\mathbf{Z}}} \left\langle \partial_{u_k} L, \mathrm{d}_p u_k \right\rangle + \partial_p L \\ &+ \sum_{j=1}^{n_{\mathbf{Z}}} \left\langle \mathrm{d}_p \psi_j, \mathcal{F}_j\left(S(\mathbf{P}), U_{<j}; u_j\right)\right\rangle + \sum_{j=1}^{n_{\mathbf{Z}}} \left\langle \psi_j, \sum_{k=1}^{j} \partial_{u_k} \mathcal{F}_j \mathrm{d}_p u_k + \partial_p \mathcal{F}_j \right\rangle, \end{aligned} \tag{2.69}$$

for states U that are consistent with the parameter fields $S(\mathbf{P})$, since the added terms are zero in this case. Here $\partial_{u_k} \mathcal{F}_j$ is the linearization of the model $\mathcal{F}_j$ with respect to u_k. Again using the fact that $\mathcal{F}_j\left(S(\mathbf{P}), U_{<j}; u_j\right) = 0$, the third term of the righthand side cancels out, and we arrive at

$$\mathrm{d}_p L = \sum_{k=1}^{n_{\mathbf{Z}}} \left\langle \partial_{u_k} L, \mathrm{d}_p u_k \right\rangle + \partial_p L + \sum_{j=1}^{n_{\mathbf{Z}}} \sum_{k=1}^{j} \left\langle \psi_j, \partial_{u_k} \mathcal{F}_j \mathrm{d}_p u_k \right\rangle + \sum_{j=1}^{n_{\mathbf{Z}}} \left\langle \psi_j, \partial_p \mathcal{F}_j \right\rangle. \tag{2.70}$$

Reordering the summation results in

$$\mathrm{d}_p L = \sum_{k=1}^{n_{\mathbf{Z}}} \langle \partial_{u_k} L, \mathrm{d}_p u_k \rangle + \partial_p L + \sum_{k=1}^{n_{\mathbf{Z}}} \sum_{j=k}^{n_{\mathbf{Z}}} \langle \psi_j, \partial_{u_k} \mathcal{F}_j \mathrm{d}_p u_k \rangle + \sum_{j=1}^{n_{\mathbf{Z}}} \langle \psi_j, \partial_p \mathcal{F}_j \rangle \,, \quad (2.71)$$

with both expressions summing over all pairs (j,k) with $1 \leq j \leq n_{\mathbf{Z}}$ and $1 \leq k \leq j$. Using the adjoint $(\partial_{u_k} \mathcal{F}_j)^\dagger$ of the linear operator $\partial_{u_k} \mathcal{F}_j$, this can be written as

$$\mathrm{d}_p L = \sum_{k=1}^{n_{\mathbf{Z}}} \langle \partial_{u_k} L, \mathrm{d}_p u_k \rangle + \partial_p L + \sum_{k=1}^{n_{\mathbf{Z}}} \sum_{j=k}^{n_{\mathbf{Z}}} \left\langle (\partial_{u_k} \mathcal{F}_j)^\dagger \psi_j, \mathrm{d}_p u_k \right\rangle + \sum_{j=1}^{n_{\mathbf{Z}}} \langle \psi_j, \partial_p \mathcal{F}_j \rangle \,, \quad (2.72)$$

which finally results in

$$\mathrm{d}_p L = \sum_{k=1}^{n_{\mathbf{Z}}} \left\langle \partial_{u_k} L + \sum_{j=k}^{n_{\mathbf{Z}}} (\partial_{u_k} \mathcal{F}_j)^\dagger \psi_j, \mathrm{d}_p u_k \right\rangle + \partial_p L + \sum_{j=1}^{n_{\mathbf{Z}}} \langle \psi_j, \partial_p \mathcal{F}_j \rangle \,. \quad (2.73)$$

2.4.2 Adjoint Model and Problem

Let the adjoint model function $\mathcal{F}_j^\dagger$ with regard to the model $\mathcal{F}_j$ and the functional L be defined through

$$\mathcal{F}_j^\dagger \left(S, U, \Psi_{>j}; \psi_j\right) \coloneqq \left(\partial_{u_j} \mathcal{F}_j\right)^\dagger \psi_j + \sum_{k>j} \left(\partial_{u_k} \mathcal{F}_j\right)^\dagger \psi_k + \partial_{u_j} L, \quad (2.74)$$

where $\Psi_{>j} \coloneqq (\psi_{j+1}, \ldots, \psi_{n_{\mathbf{Z}}})$. Under the assumption that the adjoint states Ψ solve the equations

$$\begin{aligned} \mathcal{F}_{n_{\mathbf{Z}}}^\dagger \left(S, U; \psi_{n_{\mathbf{Z}}}\right) &= 0 \\ \mathcal{F}_{n_{\mathbf{Z}}-1}^\dagger \left(S, U, \psi_{n_{\mathbf{Z}}}; \psi_{n_{\mathbf{Z}}-1}\right) &= 0 \\ &\vdots \\ \mathcal{F}_j^\dagger \left(S, U, \Psi_{>j}; \psi_j\right) &= 0 \\ &\vdots \end{aligned} \quad (2.75)$$

where again the semicolon separates known and unknown quantities in the implicit function definition, or in analogy to equation (1.5)

$$\forall\, \psi_j \colon \mathcal{F}_j^\dagger \left(S, U, \Psi_{>j}; \psi_j\right) = 0, \quad (2.76)$$

equation (2.73) reduces to

$$\mathrm{d}_p L = \partial_p L + \sum_{j=1}^{n_{\mathbf{Z}}} \langle \psi_j, \partial_p \mathcal{F}_j \rangle \,. \quad (2.77)$$

This approach is called the Adjoint State Method [75, 63], since it uses adjoint states ψ_j to compute the required derivatives, and its main advantage is the fact that the adjoint models defined in equation (2.74) do not depend on the choice of the parameter p. This means it is possible to solve the system in equation (2.75) once and use the resulting adjoint states ψ_j to compute all derivatives required for the gradient of the objective function in equations (2.35) and (2.59). The process consists of the computation of

$$\partial_{u_j} L = \mathcal{O}_j^\dagger \partial_{\mathbf{z}_j} L = \left(\mathcal{O}^\dagger \mathbf{Q}_{\mathbf{ZZ}}^{-1} (\mathbf{Z} - \mathcal{G}(\mathbf{P})) \right)_j \tag{2.78}$$

where $\mathcal{O}_j^\dagger$ is the adjoint of the linearization of the observation operator $\mathcal{O}_j$ and $\mathcal{O}^\dagger$ that of the combined operator $\mathcal{O}$.

We may again formalize this by introducing a full adjoint model $\mathcal{F}^\dagger$ that maps S and U to Ψ through (2.75), and define the following adjoint problem:

Problem 6 (Adjoint Problem)
Given a tuple of parameters $\mathbf{P}$*, a tuple of measurements* $\mathbf{Z}$*, a model* $\mathcal{F}$*, operators* $\mathcal{I}$ *and* $\mathcal{O}$*, and a conditional distribution* $\mathbf{Z}|\mathbf{P}$*, determine the adjoint states* Ψ *through the adjoint model* $\mathcal{F}^\dagger$ *with the righthand side given in equation* (2.78)*. Use these adjoint states and equation* (2.77) *to compute the gradient of* L *at* $\mathbf{P}$*.*

The term

$$\sum_{j=1}^{n_\mathbf{Z}} \langle \psi_j, \partial_p \mathcal{F}_j \rangle \tag{2.79}$$

in equation (2.77) is a componentwise representation of the one in equation (2.64), but in contrast to equation (2.65), the asymptotic complexity for computing this part of the gradient is

$$T\left(n_\mathbf{P}, n_\mathbf{Z}, n_\Omega, n_T, n_{\boldsymbol{\beta}}, n_\mathbf{z}\right) = \mathcal{O}\left(\left[n_\mathbf{P} \cdot \left[n_\Omega + n_{\boldsymbol{\beta}}\right] + \left[n_\Omega \cdot n_T\right] + n_\mathbf{z}\right] \cdot n_\mathbf{Z}\right), \tag{2.80}$$

where the subtle but important difference is a plus sign that appears instead of a multiplication sign in equation (2.65). The evaluation requires the assembly of approximately $n_\mathbf{z} \cdot n_\mathbf{Z}$ contributions to the adjoint source terms $\partial_{u_j} L$, the solution of a system of adjoint PDEs, which is in $\mathcal{O}\left(n_\Omega \cdot n_T \cdot n_\mathbf{Z}\right)$ like the solution of the forward problem, and approximately $n_\mathbf{P} \cdot \left[n_\Omega + n_{\boldsymbol{\beta}}\right] \cdot n_\mathbf{Z}$ scalar products in equation (2.77).

If p is a localized parameter, i.e. refers to a component $(\mathbf{y}_i)_k$ of a spatial part $\mathbf{y}_i$, then the support of $\partial_p \mathcal{F}_j$ is contained in the associated element E_k, see page 2. Consequently, the evaluation of the scalar product is a local operation that can be neglected in comparison to the solution of the adjoint model. Using $n_{\boldsymbol{\beta}} \ll n_\Omega \cdot n_T$ and $n_\mathbf{z} \ll n_\Omega \cdot n_T$, and assuming an observation operator $\mathcal{O}$ with comparatively low cost, equation (2.80) may be simplified, which results in the asymptotic complexity

$$T\left(n_\mathbf{P}, n_\mathbf{Z}, n_\Omega, n_T, n_{\boldsymbol{\beta}}, n_\mathbf{z}\right) = \mathcal{O}\left(n_\mathbf{Z} \cdot n_\Omega \cdot n_T\right). \tag{2.81}$$

This is the same complexity as solving the forward problem and significantly cheaper than using simple difference quotients, compare equation (2.67). In practice, the effort for solving the adjoint problem is often the same or at least in the same order of magnitude as that of solving the forward problem, and under these circumstances the adjoint state method is a very efficient technique for the calculation of the gradient.

2.5 Uncertainty Quantification

The result of the MAP approach is a single point estimate $\mathbf{P}_{\text{map}} = \arg\min_{\mathbf{P}} L(\mathbf{P})$, which may be computed with the PCG method discussed in section 2.3. However, such estimates without some sort of error quantification are largely meaningless, since it is unclear to which extent they can be trusted. Furthermore, parameters are normally estimated to create a parameterized model for simulations and predictions, and both the generation of samples and the interpretation of the results require uncertainty estimates. We therefore provide a method for the estimation of the uncertainty of the MAP inversion result $\mathbf{P}_{\text{map}}$. The first half of this section follows *Bui-Thanh et al.* [15].

If the model $\mathcal{F}$ were linear, normally distributed parameter vectors $\mathbf{p}_i$ and measurement vectors $\mathbf{z}_j$ would result in a posterior distribution that is normally distributed as well [69]. Therefore, if the nonlinearity of the model equations is not too strong, and if $\mathbf{P}_{\text{map}}$ is sufficiently close to the mean, the posterior distribution can be approximated by

$$\mathbf{P}|\mathbf{Z} \overset{\text{approx.}}{\sim} \mathcal{N}\left(\mathbf{Q}_{\mathbf{PP}}^{\text{post}}, \mathbf{P}_{\text{map}}\right), \tag{2.82}$$

where $\mathbf{Q}_{\mathbf{PP}}^{\text{post}}$ is the posterior covariance matrix.

Remark 11 *It should be stressed that such an approximation is only valid if the underlying assumptions are fulfilled. If the model is too nonlinear, the variance of the posterior may be significantly overestimated or underestimated, especially if the distribution can't be adequately represented by its first and second moments, e.g. when the posterior distribution is multimodal. In principle, the same restrictions as for the application of MAP estimation apply. Just as the MAP point $\mathbf{P}_{map}$ can only be interpreted as a rough estimate in such a situation, the linearized posterior covariance should be seen as a qualitative statement at most, not an accurate estimate of uncertainty. See section 2.6 for a discussion of statistical tests that may help in assessing the quality of the approximation.*

The linearization of the posterior PDF uses the approximate Hessian of the objective function

$$\text{Hess}(L) \approx \mathbf{Q}_{\mathbf{PP}}^{-1} + \mathbf{H}_{\mathbf{ZP}}^{T}\mathbf{Q}_{\mathbf{ZZ}}^{-1}\mathbf{H}_{\mathbf{ZP}} \tag{2.83}$$

and defines the posterior covariance matrix as

$$\mathbf{Q}_{\mathbf{PP}}^{\text{post}} \coloneqq \left[\mathbf{Q}_{\mathbf{PP}}^{-1} + \mathbf{H}_{\mathbf{ZP}}^{T}\mathbf{Q}_{\mathbf{ZZ}}^{-1}\mathbf{H}_{\mathbf{ZP}}\right]^{-1}. \tag{2.84}$$

Factoring out the matrix $\mathbf{Q}_{\mathbf{PP}}^{1/2}$ on both sides, just as in the transformation for the PCG method on page 29, results in

$$\mathbf{Q}_{\mathbf{PP}}^{\text{post}} = \mathbf{Q}_{\mathbf{PP}}^{1/2}\left[\mathbf{I} + \mathbf{M}_{\text{like}}\right]^{-1}\mathbf{Q}_{\mathbf{PP}}^{1/2} \tag{2.85}$$

with

$$\mathbf{M}_{\text{like}} \coloneqq \mathbf{Q}_{\mathbf{PP}}^{1/2}\mathbf{H}_{\mathbf{ZP}}^{T}\mathbf{Q}_{\mathbf{ZZ}}^{-1}\mathbf{H}_{\mathbf{ZP}}\mathbf{Q}_{\mathbf{PP}}^{1/2}. \tag{2.86}$$

To transform this equation, we apply the Sherman-Morrison-Woodbury formula [81], also known as the Woodbury matrix identity, which states that

$$\left[\mathbf{A}^{-1} + \mathbf{B}\mathbf{C}^{-1}\mathbf{B}^{T}\right]^{-1} = \mathbf{A} - \mathbf{A}\mathbf{B}\left[\mathbf{C} + \mathbf{B}^{T}\mathbf{A}\mathbf{B}\right]^{-1}\mathbf{B}^{T}\mathbf{A} \tag{2.87}$$

holds for matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ if both $\mathbf{A}$ and $\mathbf{C}$ are invertible and $\mathbf{B}$ has the correct dimensions. For the special case with $\mathbf{A}$ the identity matrix, $\mathbf{B}$ an orthogonal matrix and $\mathbf{C}^{-1} = \operatorname{diag}(c_i)$ a diagonal matrix, we have

$$\begin{aligned}
\left[\mathbf{I} + \mathbf{B}\mathbf{C}^{-1}\mathbf{B}^{T}\right]^{-1} &= \mathbf{I} - \mathbf{B}\left[\mathbf{C} + \mathbf{B}^{T}\mathbf{B}\right]^{-1}\mathbf{B}^{T} \\
&= \mathbf{I} - \mathbf{B}\left[\mathbf{C} + \mathbf{I}\right]^{-1}\mathbf{B}^{T} \\
&= \mathbf{I} - \mathbf{B}\left[\operatorname{diag}(c_i^{-1}) + \mathbf{I}\right]^{-1}\mathbf{B}^{T} \\
&= \mathbf{I} - \mathbf{B}\operatorname{diag}\left(\frac{1+c_i}{c_i}\right)^{-1}\mathbf{B}^{T} \\
&= \mathbf{I} - \mathbf{B}\operatorname{diag}\left(\frac{c_i}{1+c_i}\right)\mathbf{B}^{T},
\end{aligned} \tag{2.88}$$

which relates the spectral decomposition of an update to the identity matrix to the spectral decomposition needed for the inverse update. Since $\mathbf{Q}_{\mathbf{ZZ}}^{-1}$ is symmetric positive definite, $\mathbf{M}_{\text{like}}$ is symmetric positive semidefinite, compare equation (2.86), and we have the spectral decomposition

$$\mathbf{M}_{\text{like}} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^{T}, \tag{2.89}$$

where $\boldsymbol{\Lambda} = \operatorname{diag}(\lambda_i)$ contains the eigenvalues of $\mathbf{M}_{\text{like}}$ and $\mathbf{V}$ is an orthogonal matrix containing the corresponding eigenvectors. Inserting this decomposition into equation (2.85) and applying the identity above leads to

$$\begin{aligned}
\mathbf{Q}_{\mathbf{PP}}^{\text{post}} &= \mathbf{Q}_{\mathbf{PP}}^{1/2}\left[\mathbf{I} + \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^{T}\right]^{-1}\mathbf{Q}_{\mathbf{PP}}^{1/2} \\
&= \mathbf{Q}_{\mathbf{PP}}^{1/2}\left[\mathbf{I} - \mathbf{V}\boldsymbol{\Upsilon}\mathbf{V}^{T}\right]\mathbf{Q}_{\mathbf{PP}}^{1/2} \\
&= \mathbf{Q}_{\mathbf{PP}} - \mathbf{Q}_{\mathbf{PP}}^{1/2}\mathbf{V}\boldsymbol{\Upsilon}\mathbf{V}^{T}\mathbf{Q}_{\mathbf{PP}}^{1/2},
\end{aligned} \tag{2.90}$$

where

$$\boldsymbol{\Upsilon} \coloneqq \operatorname{diag}\left(\frac{\lambda_i}{1+\lambda_i}\right) \tag{2.91}$$

contains the eigenvalues for the inverse update according to equation (2.88).

Since $\mathbf{M}_{\text{like}}$ contains the matrix $\mathbf{Q}_{\mathbf{ZZ}}^{-1}$ as a factor, its rank is at most that of $\mathbf{Q}_{\mathbf{ZZ}}$. Therefore, almost all eigenvalues of $\mathbf{M}_{\text{like}}$ will be zero if $n_{\mathbf{Z}} \cdot n_{\mathbf{z}} \ll n_{\mathbf{P}} \cdot [n_\Omega + n_{\boldsymbol{\beta}}]$, with $n_{\mathbf{z}}$ and $n_{\boldsymbol{\beta}}$ defined as on page 38. We may then replace the decompositions above with approximate versions that only contain the $r \leq [n_{\mathbf{Z}} \cdot n_{\mathbf{z}}]$ largest eigenvalues, i.e. for descending eigenvalues $\lambda_1 \geq \lambda_2 \geq \lambda_3 \ldots$ we use

$$\begin{aligned} \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^T &\approx \mathbf{V}_r\boldsymbol{\Lambda}_r\mathbf{V}_r^T + \mathcal{O}\left(\sum_{i>r}\lambda_i\right) \\ \mathbf{V}\boldsymbol{\Upsilon}\mathbf{V}^T &\approx \mathbf{V}_r\boldsymbol{\Upsilon}_r\mathbf{V}_r^T + \mathcal{O}\left(\sum_{i>r}\frac{\lambda_i}{1+\lambda_i}\right), \end{aligned} \tag{2.92}$$

where the matrices

$$\begin{aligned} \boldsymbol{\Lambda}_r &\coloneqq \operatorname{diag}(\lambda_i)_{i\leq r} \\ \boldsymbol{\Upsilon}_r &\coloneqq \operatorname{diag}\left(\frac{\lambda_i}{1+\lambda_i}\right)_{i\leq r} \end{aligned} \tag{2.93}$$

are $r \times r$ matrices containing the r largest eigenvalues of $\boldsymbol{\Lambda}$ and $\boldsymbol{\Upsilon}$ respectively, and $\mathbf{V}_r$ contains the eigenvectors of these r largest eigenvalues. Note that the induced order of the eigenvectors is the same for $\boldsymbol{\Lambda}$ and $\boldsymbol{\Upsilon}$, since the function

$$f(\lambda) = \frac{\lambda}{1+\lambda} \tag{2.94}$$

is strictly monotonically increasing in $\lambda \geq 0$, and $\mathbf{V}_r$ is therefore well-defined.

Remark 12 *It is clear that the choice $r = n_{\mathbf{Z}} \cdot n_{\mathbf{z}}$ results in a good approximate spectral decomposition, since in this case all nonzero eigenvalues are recovered, but depending on the concrete application a significantly smaller r may suffice. If, for example, the observations in $\mathbf{Z}$ are a high-resolution time series of measurements, many of the observations will be redundant, i.e. they will contribute little or no information that is not already contained in one of the other observations. To a lesser degree this may also happen if the spatial resolution of the observations is high. In both situations, the numerical rank of $\mathbf{M}_{like}$ may be significantly smaller than $n_{\mathbf{Z}} \cdot n_{\mathbf{z}}$, which means that a truncation index $r \ll (n_{\mathbf{Z}} \cdot n_{\mathbf{z}})$ may already result in a good approximation of $\mathbf{M}_{like}$.*

2.5.1 Randomized Spectral Decomposition

Such an approximate spectral decomposition of $\mathbf{M}_{\text{like}}$ can be constructed using the Lanczos algorithm [28]. Alternatively, it may be computed using the randomized method presented in [15], which in contrast to the Lanczos algorithm is not an iterative procedure and therefore more readily parallelizable. This algorithm generates r independent normally distributed parameter tuples $\mathbf{R}_i$ with

$$\mathbf{R}_i \sim \mathcal{N}\left(\text{diag}(\sigma_r^2), \mathbf{0}\right), \tag{2.95}$$

where the variance σ_r^2 is small compared to the variability of $\mathbf{P}_{\text{map}}$. These tuples may be interpreted as random deviations from $\mathbf{P}_{\text{map}}$, and we examine the range space of $\mathbf{M}_{\text{like}}$ by computing

$$\mathbf{T}_i \coloneqq \mathbf{M}_{\text{like}}\mathbf{R}_i = \mathbf{Q}_{\mathbf{PP}}^{1/2}\left[\mathbf{H}_{\mathbf{ZP}}^T\mathbf{Q}_{\mathbf{ZZ}}^{-1}\left[\mathbf{H}_{\mathbf{ZP}}\mathbf{Q}_{\mathbf{PP}}^{1/2}\mathbf{R}_i\right]\right] \tag{2.96}$$

for each of the $\mathbf{R}_i$. This can be accomplished by evaluating the forward model for $\mathbf{P}_{\text{map}}$ and for $\mathbf{P}_{\text{map}} + \mathbf{Q}_{\mathbf{PP}}^{1/2}\mathbf{R}_i, 1 \leq i \leq r$. If the variance σ_r^2 is chosen sufficiently small, the change in the model outcome may be assumed to be linear in the perturbations, which yields

$$\mathbf{H}_{\mathbf{ZP}}\mathbf{Q}_{\mathbf{PP}}^{1/2}\mathbf{R}_i = \mathcal{G}\left(\mathbf{P}_{\text{map}} + \mathbf{Q}_{\mathbf{PP}}^{1/2}\mathbf{R}_i\right) - \mathcal{G}\left(\mathbf{P}_{\text{map}}\right), \tag{2.97}$$

where $\mathcal{G}$ is again the discrete model mapping $\mathbf{P}$ to $\mathbf{Z}$. The second matrix-vector multiplication in equation (2.96) is structurally the same as in the computation of the gradient of the objective function L, see equation (2.35), with the expression in equation (2.97) taking the role of the measurement residual, and may therefore be computed using the techniques described in section 2.4. The result can then be multiplied by $\mathbf{Q}_{\mathbf{PP}}^{1/2}$ to obtain $\mathbf{T}_i$.

Using a stabilized Gram-Schmidt algorithm or Householder transformations, a set of orthonormal vectors $\mathbf{C}_i$ spanning the same space as $\mathbf{T}_i$ can be constructed. With such an orthonormal basis $\mathbf{C}_i$ we may define a condensed representation of the matrix $\mathbf{M}_{\text{like}}$. Following *Halko et al.* [36], let the matrices $\mathbf{R}$, $\mathbf{T}$ and $\mathbf{C}$ be the accumulation of the vectors defined above. Equation (2.96) may then also be expressed as

$$\mathbf{T} = \mathbf{M}_{\text{like}}\mathbf{R}, \tag{2.98}$$

and a surrogate

$$\mathbf{B} \coloneqq \mathbf{C}^T\mathbf{M}_{\text{like}}\mathbf{C} \tag{2.99}$$

for the full matrix $\mathbf{M}_{\text{like}}$ has to comply with this relation restricted to the examined subspace, i.e.

$$\mathbf{C}^T\mathbf{T} \approx \mathbf{B}\left[\mathbf{C}^T\mathbf{R}\right]. \tag{2.100}$$

If the $\mathbf{T}_i$ are linearly independent the matrix $\mathbf{C}^T\mathbf{T}$ will have full rank, which means $\mathbf{C}^T\mathbf{R}$ is invertible, and we arrive at

$$\mathbf{B} \approx \left[\mathbf{C}^T\mathbf{T}\right]\left[\mathbf{C}^T\mathbf{R}\right]^{-1} = \mathbf{S}\mathbf{\Lambda}_r\mathbf{S}^T \approx \mathbf{C}^T\mathbf{M}_{\text{like}}\mathbf{C}. \tag{2.101}$$

Algorithm 7: Randomized Eigenvalue Decomposition ($\mathbf{ED}_r$)

Input: MAP estimate $\mathbf{P}_{\text{map}}$, perturbation scale σ_r^2, tolerance tol $\ll 1$
Output: matrix of eigenvectors $\mathbf{V}_r$ and matrix of eigenvalues $\mathbf{\Lambda}_r$
$i \coloneqq 0$ [set index];
$\mathbf{Z}_{\text{map}} \coloneqq \mathcal{G}\left(\mathbf{P}_{\text{map}}\right)$ [compute model outcome];
create empty matrices $\mathbf{R}$ for preimages and $\mathbf{T}$ for images;
repeat
  $i \to i + 1$ [shift index];
  $\mathbf{R}_i \sim \mathcal{N}\left(\text{diag}(\sigma_r^2), \mathbf{0}\right)$ [generate uncorrelated perturbation and append to $\mathbf{R}$];
  $\mathbf{T}_i^{(0)} \coloneqq \mathbf{Q}_{\mathbf{PP}}^{1/2}\mathbf{R}_i$ [transform into correlated perturbation];
  $\mathbf{Z}_i \coloneqq \mathcal{G}\left(\mathbf{P}_{\text{map}} + \mathbf{T}_i^{(0)}\right)$ [compute model outcome];
  $\mathbf{T}_i^{(1)} \coloneqq \mathbf{H}_{\mathbf{ZP}}^T\mathbf{Q}_{\mathbf{ZZ}}^{-1}\left[\mathbf{Z}_i - \mathbf{Z}_{\text{map}}\right]$ [apply adjoint state method];
  $\mathbf{T}_i \coloneqq \mathbf{Q}_{\mathbf{PP}}^{1/2}\mathbf{T}_i^{(1)}$ [calculate image of perturbation and append to $\mathbf{T}$];
  construct orthonormal matrix $\mathbf{C}$ spanning the same space as $\mathbf{T}$;
  $\mathbf{B} \coloneqq \left[\mathbf{C}^T\mathbf{T}\right]\left[\mathbf{C}^T\mathbf{R}\right]^{-1}$ [compute condensed operator];
  determine spectral decomposition $\mathbf{B} = \mathbf{S}_i\mathbf{\Lambda}_i\mathbf{S}_i^T$;
until *smallest eigenvalue in* $\mathbf{\Lambda}_i < tol$;
set number of eigenvectors $r \coloneqq i$;
set matrix of eigenvectors $\mathbf{V}_r \coloneqq \mathbf{C}\mathbf{S}_r$;
return $\mathbf{V}_r$ *and* $\mathbf{\Lambda}_r$;

Note that $\mathbf{B}$ and all matrices used in its construction are of size $r \times r$. As a consequence, we may apply standard algorithms for dense matrices to compute the inverse of $\mathbf{C}^T\mathbf{R}$, the eigenvalues in $\mathbf{\Lambda}_r$ and the condensed eigenvectors in $\mathbf{S}$. Setting

$$\mathbf{V}_r \coloneqq \mathbf{C}\mathbf{S} \tag{2.102}$$

then transforms the eigenvectors of the condensed representation into those of the original matrix $\mathbf{M}_{\text{like}}$. With these matrices we may approximate $\mathbf{Q}_{\mathbf{PP}}^{\text{post}}$ through

$$\mathbf{Q}_{\mathbf{PP}}^{\text{post}} \approx \mathbf{Q}_{\mathbf{PP}} - \mathbf{Q}_{\mathbf{PP}}^{1/2}\mathbf{V}_r\mathbf{\Upsilon}_r\mathbf{V}_r^T\mathbf{Q}_{\mathbf{PP}}^{1/2}. \tag{2.103}$$

The steps above result in algorithm 7 ($\mathbf{ED}_r$). The eigenvectors and eigenvalues that are returned may then be used to evaluate entries of $\mathbf{Q}_{\mathbf{PP}}^{\text{post}}$ or use it in operations, as will be discussed in section 2.6. Depending on the application, it may be beneficial to multiply the acquired eigenvectors with $\mathbf{Q}_{\mathbf{PP}}^{1/2}$, see equation (2.103).

Remark 13 *While this randomized partial spectral decomposition may be significantly cheaper than a full construction, it is important to mention that the algorithm may introduce approximation errors. If the matrix* $\mathbf{C}^T\mathbf{R}$ *is ill-conditioned, its inversion can make the algorithm unstable [36]. In this situation it may be safer to first construct the orthonormal basis* $\mathbf{C}$ *through sampling and then perform steps*

similar to those above with the images of $\mathbf{C}$ *instead of those of* $\mathbf{R}$*. This allows directly using* $\mathbf{B} = \mathbf{C}^T [\mathbf{M}_{like}\mathbf{C}]$ *as the condensed representation and thereby avoids the matrix inversion. Note, however, that this requires approximately twice the effort of the approach presented above. The special structure of* $\mathbf{M}_{like}$ *allows us to use an alternative algorithm that we will describe next.*

2.5.2 Randomized Singular Value Decomposition

We may also rewrite equation (2.85) in the form

$$\mathbf{Q}_{\mathbf{PP}}^{\text{post}} = \mathbf{Q}_{\mathbf{PP}}^{1/2} \left[\mathbf{I} + \mathbf{L}_{\text{like}}\mathbf{L}_{\text{like}}^T\right]^{-1} \mathbf{Q}_{\mathbf{PP}}^{1/2}, \tag{2.104}$$

where

$$\mathbf{L}_{\text{like}} \coloneqq \mathbf{Q}_{\mathbf{PP}}^{1/2}\mathbf{H}_{\mathbf{ZP}}\mathbf{Q}_{\mathbf{ZZ}}^{-1/2} \tag{2.105}$$

provides a decomposition of the matrix $\mathbf{M}_{\text{like}}$ as discussed in section 2.1.2. If we compute the singular value decomposition (SVD) of $\mathbf{L}_{\text{like}}$, i.e.

$$\mathbf{L}_{\text{like}} = \mathbf{V}_{\mathbf{P}}\boldsymbol{\Lambda}^{1/2}\mathbf{V}_{\mathbf{Z}}^T, \tag{2.106}$$

where $\mathbf{V}_{\mathbf{P}}$ are the left-singular vectors and $\mathbf{V}_{\mathbf{Z}}$ are the right-singular vectors, then we can write

$$\begin{aligned}
\mathbf{Q}_{\mathbf{PP}}^{\text{post}} &= \mathbf{Q}_{\mathbf{PP}}^{1/2} \left[\mathbf{I} + \mathbf{V}_{\mathbf{P}}\boldsymbol{\Lambda}^{1/2}\mathbf{V}_{\mathbf{Z}}^T\mathbf{V}_{\mathbf{Z}}\boldsymbol{\Lambda}^{1/2}\mathbf{V}_{\mathbf{P}}^T\right]^{-1} \mathbf{Q}_{\mathbf{PP}}^{1/2} \\
&= \mathbf{Q}_{\mathbf{PP}}^{1/2} \left[\mathbf{I} + \mathbf{V}_{\mathbf{P}}\boldsymbol{\Lambda}\mathbf{V}_{\mathbf{P}}^T\right]^{-1} \mathbf{Q}_{\mathbf{PP}}^{1/2} \\
&= \mathbf{Q}_{\mathbf{PP}}^{1/2} \left[\mathbf{I} - \mathbf{V}_{\mathbf{P}}\boldsymbol{\Upsilon}\mathbf{V}_{\mathbf{P}}^T\right] \mathbf{Q}_{\mathbf{PP}}^{1/2} \\
&= \mathbf{Q}_{\mathbf{PP}} - \mathbf{Q}_{\mathbf{PP}}^{1/2}\mathbf{V}_{\mathbf{P}}\boldsymbol{\Upsilon}\mathbf{V}_{\mathbf{P}}^T\mathbf{Q}_{\mathbf{PP}}^{1/2}
\end{aligned} \tag{2.107}$$

in analogy to equation (2.90). This means that the left-singular vectors $\mathbf{V}_{\mathbf{P}}$ of $\mathbf{L}_{\text{like}}$ are in fact the eigenvectors $\mathbf{V}$ of $\mathbf{M}_{\text{like}}$, and the approach using equation (2.106) leads to the same decomposition as before. But in contrast to the spectral decomposition the singular value decomposition also provides the right-singular vectors $\mathbf{V}_{\mathbf{Z}}$, which is relevant in the context of the statistical analysis presented in the next section.

An approximate singular value decomposition

$$\mathbf{L}_{\text{like}} \approx \mathbf{V}_{\mathbf{P},r}\boldsymbol{\Lambda}_r^{1/2}\mathbf{V}_{\mathbf{Z},r} \tag{2.108}$$

of the matrix $\mathbf{L}_{\text{like}}$ may be obtained through steps that are similar to those for the approximate spectral decomposition in equation (2.92). In contrast to $\mathbf{M}_{\text{like}}$ the matrix $\mathbf{L}_{\text{like}}$ is not a symmetric square matrix, and this has to be taken into account in the algorithm. Starting with random tuples $\mathbf{P}_i$ results in a condensed matrix with a size in the order of $[n_{\mathbf{P}} \cdot n_{\mathbf{p}}] \times r$, while starting with random tuples $\mathbf{Z}_j$ leads to a

matrix of size $[n_{\mathbf{Z}} \cdot n_{\mathbf{z}}] \times r$. We therefore employ the latter to arrive at a matrix size that allows the application of standard algorithms.

The algorithm for the approximate singular value decomposition generates r independent normally distributed measurement tuples

$$\mathbf{R}_j \sim \mathcal{N}(\mathbf{I}, \mathbf{0}) \tag{2.109}$$

which are interpreted as random changes in the state observations. We again sample the range space, this time of $\mathbf{L}_{\text{like}}$, to obtain

$$\mathbf{T}_j \coloneqq \mathbf{L}_{\text{like}}\mathbf{R}_j = \mathbf{Q}_{\mathbf{PP}}^{1/2}\mathbf{H}_{\mathbf{ZP}}^T\mathbf{Q}_{\mathbf{ZZ}}^{-1/2}\mathbf{R}_j, \tag{2.110}$$

and collect the vectors $\mathbf{R}_j$ and $\mathbf{T}_j$ in the matrices $\mathbf{R}$ and $\mathbf{T}$, so that

$$\mathbf{T} = \mathbf{L}_{\text{like}}\mathbf{R} \tag{2.111}$$

holds. As before, an orthonormal basis $\mathbf{C}_j$ of the sampled subspace may be constructed and subsumed in a matrix $\mathbf{C}$. For each of these basis vectors we evaluate

$$\mathbf{B}_j \coloneqq \mathbf{L}_{\text{like}}^T\mathbf{C}_j = \mathbf{Q}_{\mathbf{ZZ}}^{-1/2}\mathbf{H}_{\mathbf{ZP}}\mathbf{Q}_{\mathbf{PP}}^{1/2}\mathbf{C}_j, \tag{2.112}$$

again using equation (2.97) for the linearization of the forward model. This can be accomplished by scaling down $\mathbf{Q}_{\mathbf{PP}}^{1/2}\mathbf{C}_j$ by an appropriate factor and reestablishing the correct scale after the application of the forward model.

By construction, the resulting matrix $\mathbf{B}$ containing the images $\mathbf{B}_j$ is a representation of $\mathbf{L}_{\text{like}}^T$ acting on the range space of $\mathbf{L}_{\text{like}}$. Due to the small size of $\mathbf{B}$ we can construct its singular value decomposition

$$\mathbf{B} = \mathbf{V}_{\mathbf{Z},r}\mathbf{\Lambda}_r^{1/2}\mathbf{S}^T, \tag{2.113}$$

where $\mathbf{S}$ contains the right singular vectors of $\mathbf{B}$. Note that $\mathbf{B}$ is a representation of $\mathbf{L}_{\text{like}}^T$, not $\mathbf{L}_{\text{like}}$, and that the right-singular vectors $\mathbf{V}_{\mathbf{Z},r}$ therefore appear on the left of $\mathbf{\Lambda}_r$ in the above equation. The left-singular vectors $\mathbf{V}_{\mathbf{P},r}$ of $\mathbf{L}_{\text{like}}$ may then be obtained through setting

$$\mathbf{V}_{\mathbf{P},r} \coloneqq \mathbf{CS} \tag{2.114}$$

as in the randomized spectral decomposition.

This second approach is given in algorithm 8 ($\mathbf{SVD}_r$). Note that it is in principle the same as the two-pass approach mentioned in remark 13, but applied to $\mathbf{L}_{\text{like}}$ instead of $\mathbf{M}_{\text{like}}$. The singular value decomposition typically has the same cost as the spectral decomposition, since the number of forward model runs, adjoint model runs and matrix multiplications per eigenvalue respectively singular value is the same for the two algorithms. In a general setting the SVD is therefore the preferred algorithm, since it provides additional information at approximately the same cost.

Algorithm 8: Randomized Singular Value Decomposition ($\mathbf{SVD}_r$)

Input: MAP estimate $\mathbf{P}_{\text{map}}$, perturbation scale σ_r^2, tolerance tol $\ll 1$

Output: matrices of singular vectors $\mathbf{V}_{\mathbf{P},r}$ and $\mathbf{V}_{\mathbf{Z},r}$, matrix of singular values $\mathbf{\Lambda}_r^{1/2}$

$i \coloneqq 0$ [set index];

$\mathbf{Z}_{\text{map}} \coloneqq \mathcal{G}\left(\mathbf{P}_{\text{map}}\right)$ [compute model outcome];

create empty matrices V (preimages), W (images), C (basis vectors) and B (representation);

repeat

- $i \to i + 1$ [shift index];
- $\mathbf{R}_j \sim \mathcal{N}\left(\mathbf{I}, \mathbf{0}\right)$ [generate uncorrelated perturbation and append to $\mathbf{R}$];
- $\mathbf{T}_j \coloneqq \mathbf{Q}_{\mathbf{PP}}^{1/2}\mathbf{H}_{\mathbf{ZP}}^T\mathbf{Q}_{\mathbf{ZZ}}^{-1/2}\mathbf{R}_j$ [apply adjoint state method and append to $\mathbf{T}$];
- construct orthonormal matrix $\mathbf{C}$ spanning the same space as $\mathbf{T}$;
- $\mathbf{Z}_j \coloneqq \mathcal{G}\left(\mathbf{P}_{\text{map}} + \sigma_r\mathbf{Q}_{\mathbf{PP}}^{1/2}\mathbf{C}_j\right)$ [compute model outcome];
- $\mathbf{B}_j \coloneqq \sigma_r^{-1}\mathbf{Q}_{\mathbf{ZZ}}^{-1/2}\left[\mathbf{Z}_j - \mathbf{Z}_{\text{map}}\right]$ [calculate result and append to $\mathbf{B}$];
- determine singular value decomposition $\mathbf{B} = \mathbf{V}_{\mathbf{Z},i}\mathbf{\Lambda}_i^{1/2}\mathbf{S}_i^T$;

until *smallest singular value in* $\mathbf{\Lambda}_i^{1/2} < tol^{1/2}$;

set number of singular vectors $r \coloneqq i$;

set second matrix of singular vectors $\mathbf{V}_{\mathbf{P},r} \coloneqq \mathbf{C}\mathbf{S}_r$;

return $\mathbf{V}_{\mathbf{P},r}$, $\mathbf{V}_{\mathbf{Z},r}$ *and* $\mathbf{\Lambda}_r^{1/2}$;

See figure 2.7 for an example of the spectra produced by algorithm 8 ($\mathbf{SVD}_r$). The exact shape of the spectrum depends on the forward model $\mathcal{F}$, the chosen prior distribution and the observations. The amount of correlation between the measurement locations affects the width of the spectrum, while the ratio of assumed measurement errors and prior uncertainty determines its height. The height and shape of the spectrum influence the performance of the optimization schemes, see the discussion concerning the spectral condition number in section 2.3.2. Its width determines the cost of the linearized uncertainty quantification. Figures 2.8 and 2.9 show examples of the pairs of singular vectors that are obtained.

2.6 A Posteriori Analysis and Sample Generation

The results of the parameter estimation, $\mathbf{P}_{\text{map}}$, and the uncertainty quantification, $\mathbf{Q}_{\mathbf{PP}}^{\text{post}}$, are often used to generate samples from the posterior distribution, which may in turn be used in simulations and subsequent analysis. This section describes a method to obtain such samples based on the approximate spectral decomposition that was constructed in section 2.5. It also discusses statistical tests that may be performed to assess the quality of the inversion results.

The constituents of the posterior covariance matrix $\mathbf{Q}_{\mathbf{PP}}^{\text{post}}$, computed as described in

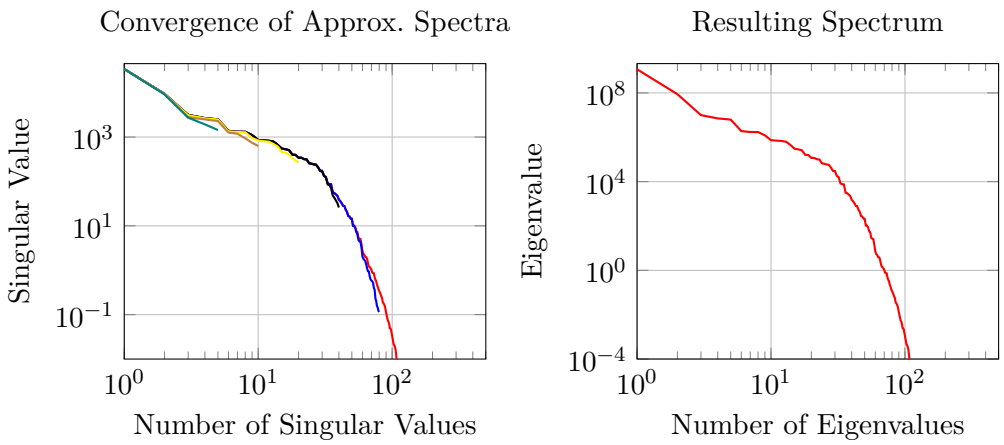

Figure 2.7: *Left:* Approximate spectra of $\mathbf{L}_{\text{like}}$ obtained with algorithm 8 ($\mathbf{SVD}_r$) for $r = 5$ (—), 10 (—), 20 (—), 40 (—), 80 (—) and 108 (—) for a synthetic example setup. The algorithm is stopped when the first singular value below 10^{-2} is encountered. Forward model is the transient groundwater flow equation with $n_\Omega = 1.64 \cdot 10^4$, $n_T = 100$, $n_\phi = 2.53 \cdot 10^3$. *Right:* The resulting spectrum of $\mathbf{M}_{\text{like}}$. Since each eigenvalue requires the solution of a forward problem and an adjoint problem, the cost for the spectral decomposition is about an order of magnitude lower than the full assembly of the sensitivity matrix $\mathbf{H}_{\mathbf{ZP}}$.

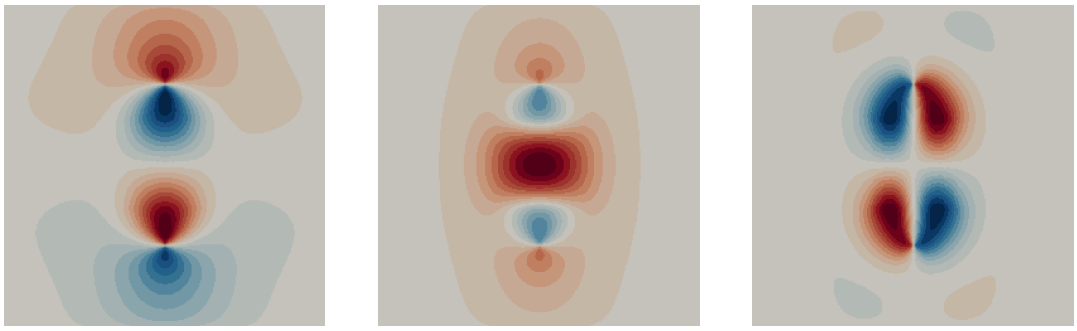

Figure 2.8: *From left to right:* The first three left-singular vectors of $\mathbf{L}_{\text{like}}$ for the steady-state dipole experiments of section 6.1.1, computed with algorithm 8 ($\mathbf{SVD}_r$). The underlying synthetic reference field is homogeneous, and the symmetry of the experimental setup is reflected in the symmetry of the singular vectors. The corresponding right-singular vectors can be found in figure 2.9.

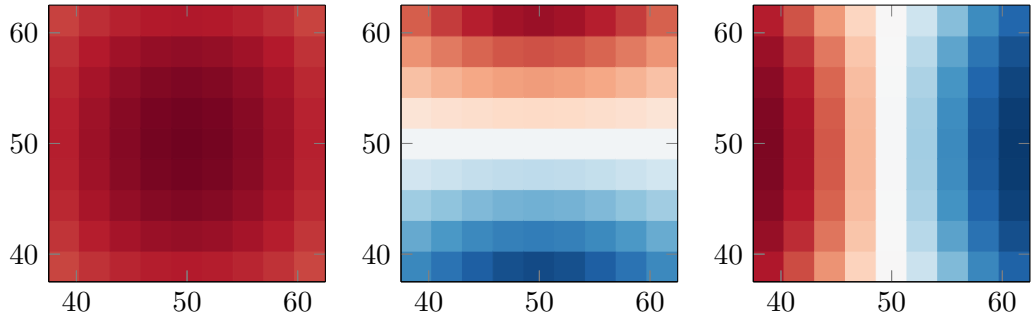

Figure 2.9: *From left to right:* The first three right-singular vectors of $\mathbf{L}_{\text{like}}$, interpreted as functions on a subdomain of Ω by plotting the mean of the observations in the four corners on each of the quadratic subdomains. See figure 2.8 for the corresponding left-singular vectors.

the previous section, may be used to give an upper bound for the posterior variance. The algorithms 7 ($\mathbf{ED}_r$) and 8 ($\mathbf{SVD}_r$) both produce a matrix of eigenvectors $\mathbf{V}_r = \mathbf{V}_{\mathbf{P},r}$ and a diagonal matrix of eigenvalues $\mathbf{\Lambda}_r$ that can be used to evaluate entries of $\mathbf{Q}_{\mathbf{PP}}^{\text{post}}$ through equation (2.103). The diagonal entries of $\mathbf{Q}_{\mathbf{PP}}^{\text{post}}$ contain the posterior variance of the individual parameters, and their evaluation provides a first indication of the quality of the inversion result in terms of reduced uncertainty of the parameters. Setting

$$\mathbf{W}_r \coloneqq \mathbf{Q}_{\mathbf{PP}}^{1/2}\mathbf{V}_r, \tag{2.115}$$

and denoting the k-th vector in the matrix $\mathbf{W}_r$ with $(\mathbf{W}_r)_k$, the entries of $\mathbf{Q}_{\mathbf{PP}}^{\text{post}}$ can be written as

$$\begin{aligned}\left(\mathbf{Q}_{\mathbf{PP}}^{\text{post}}\right)_{ij} &= \mathbf{E}_i^T\mathbf{Q}_{\mathbf{PP}}^{\text{post}}\mathbf{E}_j \\ &= \mathbf{E}_i^T\mathbf{Q}_{\mathbf{PP}}\mathbf{E}_j - \mathbf{E}_i^T\mathbf{Q}_{\mathbf{PP}}^{1/2}\mathbf{V}_r\mathbf{\Lambda}_r\mathbf{V}_r^T\mathbf{Q}_{\mathbf{PP}}^{1/2}\mathbf{E}_j \\ &= (\mathbf{Q}_{\mathbf{PP}})_{ij} - \mathbf{E}_i^T\mathbf{W}_r\mathbf{\Lambda}_r\mathbf{W}_r^T\mathbf{E}_j \\ &= (\mathbf{Q}_{\mathbf{PP}})_{ij} - \sum_{k=1}^{r}\left[\mathbf{E}_i^T(\mathbf{W}_r)_k\right]\frac{\lambda_k}{1+\lambda_k}\left[(\mathbf{W}_r)_k\right]^T\mathbf{E}_j\Big],\end{aligned} \tag{2.116}$$

where $\mathbf{E}_i$ and $\mathbf{E}_j$ are unit vectors selecting the row and column of $\mathbf{Q}_{\mathbf{PP}}^{\text{post}}$. For the diagonal entries this implies

$$\sigma^2_{\text{post},i} \coloneqq \left(\mathbf{Q}_{\mathbf{PP}}^{\text{post}}\right)_{ii} = (\mathbf{Q}_{\mathbf{PP}})_{ii} - \sum_{k=1}^{r}\frac{\lambda_k}{1+\lambda_k}\left[(\mathbf{W}_r)_k\right]^T\mathbf{E}_i\Big]^2, \tag{2.117}$$

i.e. the information gained from the observations reduces the variance of the individual parameters. If we stop algorithm 7 ($\mathbf{ED}_r$) or algorithm 8 ($\mathbf{SVD}_r$) prematurely,

e.g. due to time constraints, and only compute an approximation

$$\widetilde{\sigma}^2_{\text{post},i} \coloneqq (\mathbf{Q}_{\mathbf{PP}})_{ii} - \sum_{k=1}^{\widetilde{r}} \frac{\lambda_k}{1+\lambda_k} \left[\left[(\mathbf{W}_r)_k \right]^T \mathbf{E}_i \right]^2, \tag{2.118}$$

where $\widetilde{r} < r$ is the number of eigenvalues that have been computed, and if the resulting spectrum is accurate enough, then the error incurred in the computation of the posterior variance is

$$\sigma^2_{\text{post},i} - \widetilde{\sigma}^2_{\text{post},i} = -\sum_{k=\widetilde{r}+1}^{r} \frac{\lambda_k}{1+\lambda_k} \left[\left[(\mathbf{W}_r)_k \right]^T \mathbf{E}_i \right]^2 \leq 0. \tag{2.119}$$

Consequently, a partially computed spectrum of $\widetilde{r} < r$ eigenvalues at most leads to an overestimation of the linearized posterior variance, never an underestimation, and partial results can be used to compute upper bounds. Note that this argument is only valid for the two algorithms if the chosen number of computed eigenvalues is large enough, since the partial spectrum is not truncated but rather approximated. However, the eigenvalues tend to converge to their final values from below, as shown in figure 2.7, which suggests that this bound may also hold for coarser approximations.

2.6.1 Realizations of the Posterior Distribution

If we extend the system of orthonormal vectors contained in $\mathbf{V}_r$ to a full orthonormal basis of the parameter space and denote the resulting augmented matrix with $\mathbf{V}_r^+$, then equation (2.90) may also be written as

$$\mathbf{Q}_{\mathbf{PP}}^{\text{post}} = \mathbf{Q}_{\mathbf{PP}}^{1/2} \left[\mathbf{I} - \mathbf{V}_r^+ \boldsymbol{\Upsilon}_r^+ \left[\mathbf{V}_r^+ \right]^T \right] \mathbf{Q}_{\mathbf{PP}}^{1/2}, \tag{2.120}$$

where $\boldsymbol{\Upsilon}^+$ denotes the matrix $\boldsymbol{\Upsilon}$ padded with zeros. This may in turn be written as

$$\mathbf{Q}_{\mathbf{PP}}^{\text{post}} = \mathbf{Q}_{\mathbf{PP}}^{1/2} \mathbf{V}_r^+ \left[\mathbf{I} - \boldsymbol{\Upsilon}_r^+ \right] \left[\mathbf{V}_r^+ \right]^T \mathbf{Q}_{\mathbf{PP}}^{1/2}, \tag{2.121}$$

since by construction $\mathbf{V}_r^+ \left[\mathbf{V}_r^+ \right]^T = \mathbf{I}$. A decomposition

$$\mathbf{Q}_{\mathbf{PP}}^{\text{post}} = \mathbf{L}_{\text{post}} \mathbf{L}_{\text{post}}^T \tag{2.122}$$

for the generation of samples according to section 2.1.2 could therefore be obtained by setting

$$\mathbf{L}_{\text{post}} \coloneqq \mathbf{Q}_{\mathbf{PP}}^{1/2} \mathbf{V}_r^+ \left[\mathbf{I} - \boldsymbol{\Upsilon}_r^+ \right]^{1/2}. \tag{2.123}$$

However, this decomposition would depend on the particular choice of $\mathbf{V}_r^+$ and be relatively unwieldy. Instead, we again use that $\mathbf{V}_r^+$ is an orthogonal matrix, and therefore $\left[\mathbf{V}_r^+ \right]^T \mathbf{V}_r^+ = \mathbf{I}$, and define $\mathbf{L}_{\text{post}}$ as

$$\mathbf{L}_{\text{post}} \coloneqq \mathbf{Q}_{\mathbf{PP}}^{1/2} \mathbf{V}_r^+ \left[\mathbf{I} - \boldsymbol{\Upsilon}_r^+ \right]^{1/2} \left[\mathbf{V}_r^+ \right]^T. \tag{2.124}$$

Algorithm 9: Generation of Samples from Posterior Distribution ($\mathbf{SG}_{\text{post}}$)

Input: MAP estimate $\mathbf{P}_{\text{map}}$, matrix of eigenvectors $\mathbf{V}_r$, matrix of eigenvalues $\mathbf{\Lambda}_r$
Output: Sample of posterior distribution $\mathbf{P}$
$\mathbf{W} \coloneqq \mathcal{N}(\mathbf{I}, \mathbf{0})$ [generate white noise];
for $1 \leq i \leq r$ **do**
  $\mathbf{W}_i \coloneqq (\mathbf{V}_r)_i \left[(\mathbf{V}_r)_i\right]^T \mathbf{W}$ [project $\mathbf{W}$ onto eigenspaces];
end
$\mathbf{W}_{\text{rem}} \coloneqq \mathbf{W} - \sum_{i=1}^{r} \mathbf{W}_i$ [compute remainder that is not in span of eigenvectors];
$\mathbf{R} \coloneqq \mathbf{W}_{\text{rem}} + \sum_{i=1}^{r} \left[1 + \lambda_i\right]^{-1/2} \mathbf{W}_i$ [scale projections by appropriate factor];
$\mathbf{P} \coloneqq \mathbf{P}_{\text{map}} + \mathbf{Q}_{\mathbf{PP}}^{1/2}\mathbf{R}$ [transform result and add posterior mean];
return $\mathbf{P}$;

This second definition is particularly convenient for the production of samples from the posterior distribution. Since the eigenvectors of $\left[\mathbf{I} - \mathbf{\Upsilon}_r^+\right]$ are also the eigenvectors of $\left[\mathbf{I} - \mathbf{\Upsilon}_r^+\right]^{1/2}$, the matrix $\mathbf{V}_r^+ \left[\mathbf{I} - \mathbf{\Upsilon}_r^+\right] \left[\mathbf{V}_r^+\right]^T$ simply scales the individual eigenspaces of $\mathbf{M}_{\text{like}}$. Note that there is no need to explicitly construct the orthogonal basis in $\mathbf{V}_r^+$, which would be computationally demanding. Instead, we may decompose a given $\mathbf{P}$ into the contributions from the individual eigenspaces,

$$\mathbf{P} = \sum_{i=1}^{r} (\mathbf{V}_r)_i \left[(\mathbf{V}_r)_i\right]^T \mathbf{P} + \mathbf{P}_{\text{rem}}, \tag{2.125}$$

where $\mathbf{P}_{\text{rem}}$ is the part of $\mathbf{P}$ that is not in the span of $\mathbf{V}_r$, and write the action of $\mathbf{L}_{\text{post}}$ on $\mathbf{P}$ as

$$\begin{aligned} \mathbf{L}_{\text{post}}\mathbf{P} &= \mathbf{L}_{\text{post}} \left[\sum_{i=1}^{r} (\mathbf{V}_r)_i \left[(\mathbf{V}_r)_i\right]^T \mathbf{P} + \mathbf{P}_{\text{rem}}\right] \\ &= \mathbf{Q}_{\mathbf{PP}}^{1/2}\mathbf{V}_r^+ \left[\mathbf{I} - \mathbf{\Upsilon}_r^+\right]^{1/2} \left[\mathbf{V}_r^+\right]^T \left[\sum_{i=1}^{r} (\mathbf{V}_r)_i \left[(\mathbf{V}_r)_i\right]^T \mathbf{P} + \mathbf{P}_{\text{rem}}\right] \\ &= \mathbf{Q}_{\mathbf{PP}}^{1/2} \left[\sum_{i=1}^{r} \left[1 + \lambda_i\right]^{-1/2} (\mathbf{V}_r)_i \left[(\mathbf{V}_r)_i\right]^T \mathbf{P} + \mathbf{P}_{\text{rem}}\right]. \end{aligned} \tag{2.126}$$

The cost for the generation of samples of the posterior distribution is therefore the same as for samples of the prior distribution, except for a few scalar products and vector additions. Algorithm 9 ($\mathbf{SG}_{\text{post}}$) summarizes the steps for the generation of realizations of the posterior distribution, while figure 2.10 shows examples of the output of this algorithm.

Remark 14 *The decomposition constructed above is of the form*

$$\mathbf{Q}_{\mathbf{PP}}^{post} = \mathbf{L}_{post}\mathbf{L}_{post}^T \tag{2.127}$$

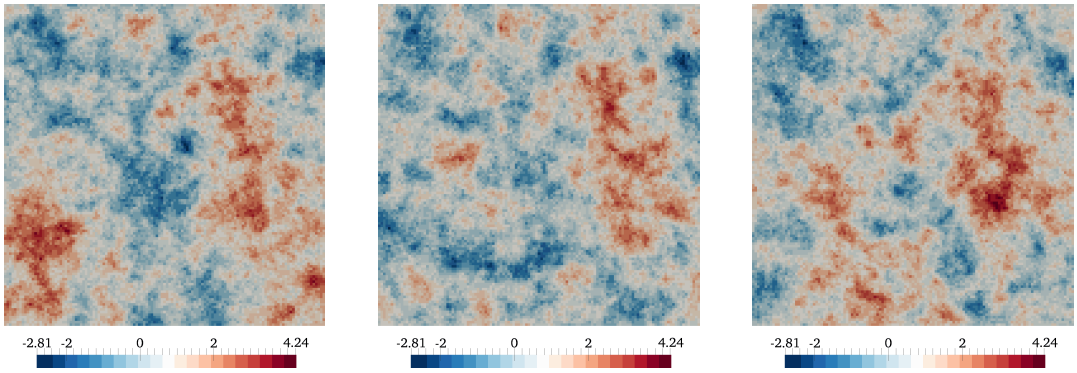

Figure 2.10: Three different realizations of the posterior distribution from section 6.3. The uncertainty of the parameters is very low in the top third of the domain and most of its right half, which leads to samples that show many similarities in that part of the domain. The parameters in the lower left of the domain remain uncertain, and this is also reflected in the realizations.

that was used in section 2.1.2. While this has been employed to generate samples of the posterior distribution, it also means that the MAP estimate $\mathbf{P}_{map}$ *and the posterior covariance matrix* $\mathbf{Q}_{\mathbf{PP}}^{post}$ *may themselves be used as prior mean and prior covariance matrix for another inversion if new data becomes available. All steps of the previous sections carry through when all occurrences of* $\mathbf{Q}_{\mathbf{PP}}^{1/2}$ *are replaced by* $\mathbf{L}_{post}$ *and* $\mathbf{L}_{post}^{T}$*, with the correct choice being clear from context. This property enables the application of the discussed methods in situations where inversion during the course of data acquisition is useful for planing purposes or where additional data becomes available after the inversion process.*

The new prior covariance matrix $\mathbf{Q}_{\mathbf{PP}}^{post}$ *ensures consistency with the previous data in the vicinity of the new prior mean* $\mathbf{P}_{map}$*, but this consistency may deteriorate in the case of nonlinear models. Consequently, any previous data needs to be included in subsequent inversions to keep the parameter estimate in the correct subspace. This requires an update of the measurement covariance matrix, which is given in section 2.6.3 as* $\mathbf{Q}_{\mathbf{ZZ}}^{post}$*.*

2.6.2 Normalized Errors

While the matrix $\mathbf{L}_{\text{post}}$ of the previous section may be used to generate samples, its inverse

$$\begin{aligned}\mathbf{L}_{\text{post}}^{-1} &= \left[\mathbf{Q}_{\mathbf{PP}}^{1/2}\mathbf{V}_r^+\left[\mathbf{I}-\boldsymbol{\Upsilon}_r^+\right]^{1/2}\left[\mathbf{V}_r^+\right]^T\right]^{-1} \\ &= \mathbf{V}_r^+\left[\mathbf{I}-\boldsymbol{\Upsilon}_r^+\right]^{-1/2}\left[\mathbf{V}_r^+\right]^T\mathbf{Q}_{\mathbf{PP}}^{-1/2} \\ &= \mathbf{V}_r^+\left[\mathbf{I}+\boldsymbol{\Lambda}_r^+\right]^{1/2}\left[\mathbf{V}_r^+\right]^T\mathbf{Q}_{\mathbf{PP}}^{-1/2},\end{aligned} \tag{2.128}$$

with the matrix of eigenvalues $\mathbf{\Lambda}_r$ extended to $\mathbf{\Lambda}_r^+$ in the same fashion, can be used to check if a given parameter vector tuple $\mathbf{P}$ is a realization of the posterior distribution. If the posterior distribution is close enough to being Gaussian and $\mathbf{P}_{\text{map}}$ and $\mathbf{Q}_{\mathbf{PP}}^{\text{post}}$ can be used as its mean and covariance matrix, then equation (2.17) gives

$$\mathbf{P} \sim \mathcal{N}\left(\mathbf{Q}_{\mathbf{YY}}^{\text{post}}, \mathbf{P}_{\text{map}}\right) \iff \mathbf{L}_{\text{post}}^{-1}\left[\mathbf{P} - \mathbf{P}_{\text{map}}\right] \sim \mathcal{N}\left(\mathbf{I}, \mathbf{0}\right). \tag{2.129}$$

Any synthetic reference $\widehat{\mathbf{P}}$ that is used to generate observations for subsequent inversion is by definition both a realization of the prior distribution and a realization of the posterior distribution, and therefore

$$\Delta_{\mathbf{P}} \coloneqq \mathbf{L}_{\text{post}}^{-1}\left[\widehat{\mathbf{P}} - \mathbf{P}_{\text{map}}\right] \sim \mathcal{N}\left(\mathbf{I}, \mathbf{0}\right) \tag{2.130}$$

has to hold for the decorrelated error $\Delta_{\mathbf{P}}$, also known as normalized error [52]. This means the individual components $(\Delta_{\mathbf{P}})_i$ of $\Delta_{\mathbf{P}}$ should be independent realizations of a random variable $\mathbf{Y} \sim \mathcal{N}\left(1, 0\right)$, which can be checked by computing the sample mean

$$\mathbf{Y}^{\star} \coloneqq \frac{1}{N}\sum_{i=1}^{N}(\Delta_{\mathbf{P}})_i, \tag{2.131}$$

the sample variance

$$s_{\mathbf{Y}}^2 \coloneqq \frac{N}{N-1}\left[m_2 - m_1^2\right], \tag{2.132}$$

the sample skewness

$$b_1^{\mathbf{Y}} \coloneqq \left[\frac{N}{N-1}m_2^{-1}\right]^{3/2} m_3, \tag{2.133}$$

and the sample kurtosis

$$b_2^{\mathbf{Y}} \coloneqq \left[\frac{N}{N-1}m_2^{-1}\right]^{2} m_4, \tag{2.134}$$

where N is the number of components and

$$m_k \coloneqq \frac{1}{N}\sum_{i=1}^{N}\left[(\Delta_{\mathbf{P}})_i - \mathbf{Y}^{\star}\right]^k \tag{2.135}$$

is the definition of the sample moments m_k [41]. The expected values are $\mathbf{Y}^{\star} = 0$, $s_{\mathbf{Y}}^2 = 1$, $b_1^{\mathbf{Y}} = 0$, and $b_2^{\mathbf{Y}} = 3$ respectively, since these are the results for the standard normal distribution $\mathcal{N}\left(1, 0\right)$. If $\mathbf{P}_{\text{map}}$ and $\mathbf{Q}_{\mathbf{PP}}^{\text{post}}$ pass the test, i.e. the sample moments are reasonably close to their expected values, then the results may be accepted as a lower-order representation of the posterior distribution. If the statistics of the components of $\Delta_{\mathbf{P}}$ deviate significantly from these values, then the inversion result should not be interpreted as the mean of the posterior distribution. The primary reasons for failures of this test are the following:

- The forward model $\mathcal{F}$ is too nonlinear and the resulting posterior distribution can't be approximated by a Gaussian distribution. If it is multimodal or heavily skewed, then the modus $\mathbf{P}_{\text{map}}$ is either non-unique or a poor estimate for the mean, as discussed in remark 8, and the posterior covariance matrix $\mathbf{Q}_{\mathbf{PP}}^{\text{post}}$ may fail to capture the covariance structure due to the linearization used in its definition. In such a case the Maximum A Posteriori approach is inadequate and a more sophisticated alternative should be applied instead, see the discussion in section 3.2.
- The optimization algorithm didn't converge to the global optimum, either because it broke down or because it converged to a local optimum that is not the global optimum. Then a different starting point or some form of relaxation should be used, and if this fails as well a different choice of optimization scheme is required.
- For data from real-world applications, an inadequate choice of forward model $\mathcal{F}$ is a third possibility. If the model isn't able to reproduce the state observations within the prescribed bounds, then the result will be biased or follow different statistics. This includes the case of measurement errors that are significantly larger or smaller than assumed, or assumptions about the parameter distribution that don't match the ground truth.

Remark 15 *The application of $\mathbf{L}_{post}^{-1}$ as presented above requires multiplication with $\mathbf{Q}_{\mathbf{PP}}^{-1/2}$, which is in most cases prohibitively expensive. Note, however, that the caching PCG method, algorithm 6 (**PCG**$_c$), provides*

$$\mathbf{V}_{map} \coloneqq \mathbf{Q}_{\mathbf{PP}}\mathbf{P}_{map}, \tag{2.136}$$

the limit of the auxiliary second sequence. Since

$$\mathbf{Q}_{\mathbf{PP}}^{-1/2}\mathbf{P}_{map} = \mathbf{Q}_{\mathbf{PP}}^{1/2}\mathbf{Q}_{\mathbf{PP}}^{-1}\mathbf{P}_{map} = \mathbf{Q}_{\mathbf{PP}}^{1/2}\mathbf{V}_{map} \tag{2.137}$$

holds, the multiplication with $\mathbf{Q}_{\mathbf{PP}}^{-1/2}$ can be replaced by one with $\mathbf{Q}_{\mathbf{PP}}^{1/2}$. This latter multiplication can often be carried out efficiently, see the discussion in section 2.1.2. This also holds for the caching variants of the Gauss-Newton and Levenberg-Marquardt methods that are briefly discussed in section 3.3.2, since both methods directly provide $\mathbf{Q}_{\mathbf{PP}}^{-1/2}\mathbf{P}_{map}$.

The transformed synthetic reference $\mathbf{Q}_{\mathbf{PP}}^{-1/2}\widehat{\mathbf{P}}$ can be computed using

$$\mathbf{Q}_{\mathbf{PP}}^{-1/2}\widehat{\mathbf{P}} = \mathbf{Q}_{\mathbf{PP}}^{-1/2}\left[\mathbf{P}^* + \mathbf{Q}_{\mathbf{PP}}^{1/2}\mathbf{W}\right] = \mathbf{Q}_{\mathbf{PP}}^{-1/2}\mathbf{P}^* + \mathbf{W}, \tag{2.138}$$

*where $\mathbf{Q}_{\mathbf{PP}}^{-1/2}\mathbf{P}^*$ is provided by the caching algorithms as described above and $\mathbf{W}$ is the white noise used in the generation of $\widehat{\mathbf{P}}$, compare algorithm 1 (**SG**). Here the discussion in section 2.1.2 has to be taken into account.*

Algorithm 10: Randomized Test of Unbiasedness (Parameter Version; $\mathbf{TU}_r^{\mathbf{P}}$)

Input: MAP estimate $\mathbf{P}_{\text{map}}$, synthetic reference $\widehat{\mathbf{P}}$, matrix of eigenvectors $\mathbf{V}_r$, matrix of eigenvalues $\mathbf{\Lambda}_r$
Output: Normalized error $\Delta_{\mathbf{P}}$
$\mathbf{E} \coloneqq \widehat{\mathbf{P}} - \mathbf{P}_{\text{map}}$ [compute error of estimate];
$\mathbf{W} \coloneqq \mathbf{Q}_{\mathbf{PP}}^{-1/2}\mathbf{E}$ [unnecessary if using cached data, see remark 15 for details];
for $1 \leq i \leq r$ **do**
  $\mathbf{W}_i \coloneqq (\mathbf{V}_r)_i \left[(\mathbf{V}_r)_i\right]^T \mathbf{W}$ [project $\mathbf{W}$ onto eigenspaces];
end
$\mathbf{W}_{\text{rem}} \coloneqq \mathbf{W} - \sum_{i=1}^{r} \mathbf{W}_i$ [compute remainder that is not in span of eigenvectors];
$\Delta_{\mathbf{P}} \coloneqq \mathbf{W}_{\text{rem}} + \sum_{i=1}^{r} \left[1 + \lambda_i\right]^{1/2} \mathbf{W}_i$ [scale projections by appropriate factor];
return $\Delta_{\mathbf{P}}$;

2.6.3 Normalized Residuals

While algorithm 10 ($\mathbf{TU}_r^{\mathbf{P}}$) may be used to assess the quality of the inversion result, it relies on the synthetic reference $\widehat{\mathbf{P}}$. In realistic scenarios this information is not available, and it is therefore not possible to decorrelate the error $\widehat{\mathbf{P}} - \mathbf{P}_{\text{map}}$. However, the measurement residual $\mathbf{Z} - \mathcal{G}(\mathbf{P}_{\text{map}})$ may also be used for a posteriori analysis [44]. Just as the posterior covariance matrix $\mathbf{Q}_{\mathbf{PP}}^{\text{post}}$ describes the posterior distribution given the observations, we may define a posterior covariance matrix $\mathbf{Q}_{\mathbf{ZZ}}^{\text{post}}$ that describes the distribution of the measurement residuals given the inversion result. In addition to the measurement errors in $\mathbf{Q}_{\mathbf{ZZ}}$, this has to take the correlation due to the inversion process into account.

In the context of the linearized posterior distribution, *Nowak* [61] derives the expression

$$\mathbf{Q}_{\mathbf{ZZ}}^{\text{post}} = \mathbf{Q}_{\mathbf{ZZ}} \left[\mathbf{H}_{\mathbf{ZP}} \mathbf{Q}_{\mathbf{PP}} \mathbf{H}_{\mathbf{ZP}}^T + \mathbf{Q}_{\mathbf{ZZ}}\right]^{-1} \mathbf{Q}_{\mathbf{ZZ}} \tag{2.139}$$

for the covariance matrix of the measurement residuals after inversion. Applying the Sherman-Morrison-Woodbury formula, equation (2.87), transforms this identity into

$$\begin{aligned} \mathbf{Q}_{\mathbf{ZZ}}^{\text{post}} &= \mathbf{Q}_{\mathbf{ZZ}} \left[\mathbf{Q}_{\mathbf{ZZ}}^{-1} - \mathbf{Q}_{\mathbf{ZZ}}^{-1} \mathbf{H}_{\mathbf{ZP}} \left[\mathbf{Q}_{\mathbf{PP}}^{-1} + \mathbf{H}_{\mathbf{ZP}}^T \mathbf{Q}_{\mathbf{ZZ}}^{-1} \mathbf{H}_{\mathbf{ZP}}\right]^{-1} \mathbf{H}_{\mathbf{ZP}}^T \mathbf{Q}_{\mathbf{ZZ}}^{-1}\right] \mathbf{Q}_{\mathbf{ZZ}} \\ &= \mathbf{Q}_{\mathbf{ZZ}} - \mathbf{H}_{\mathbf{ZP}} \left[\mathbf{Q}_{\mathbf{PP}}^{-1} + \mathbf{H}_{\mathbf{ZP}}^T \mathbf{Q}_{\mathbf{ZZ}}^{-1} \mathbf{H}_{\mathbf{ZP}}\right]^{-1} \mathbf{H}_{\mathbf{ZP}}^T \\ &= \mathbf{Q}_{\mathbf{ZZ}} - \mathbf{H}_{\mathbf{ZP}} \mathbf{Q}_{\mathbf{PP}}^{\text{post}} \mathbf{H}_{\mathbf{ZP}}^T, \end{aligned} \tag{2.140}$$

i.e. just as the additional information from the measurements reduces the uncertainty of the parameters, compare equation (2.90), the updated parameters reduce the variability of the measurement residuals. Unfortunately, the above formula is not directly applicable to compute $\mathbf{Q}_{\mathbf{ZZ}}^{\text{post}}$, since the matrix $\mathbf{H}_{\mathbf{ZP}}$ would have to be assembled. We

therefore return to equation (2.139) and instead use the root $\mathbf{Q}_{\mathbf{ZZ}}^{1/2}$ to reformulate it as

$$\begin{aligned}\mathbf{Q}_{\mathbf{ZZ}}^{\text{post}} &= \mathbf{Q}_{\mathbf{ZZ}}^{1/2}\left[\mathbf{I} + \mathbf{Q}_{\mathbf{ZZ}}^{-1/2}\mathbf{H}_{\mathbf{ZP}}\mathbf{Q}_{\mathbf{PP}}\mathbf{H}_{\mathbf{ZP}}^{T}\mathbf{Q}_{\mathbf{ZZ}}^{-1/2}\right]^{-1}\mathbf{Q}_{\mathbf{ZZ}}^{1/2} \\ &= \mathbf{Q}_{\mathbf{ZZ}}^{1/2}\left[\mathbf{I} + \mathbf{L}_{\text{like}}^{T}\mathbf{L}_{\text{like}}\right]\mathbf{Q}_{\mathbf{ZZ}}^{1/2},\end{aligned} \tag{2.141}$$

with

$$\mathbf{L}_{\text{like}} = \mathbf{Q}_{\mathbf{PP}}^{1/2}\mathbf{H}_{\mathbf{ZP}}\mathbf{Q}_{\mathbf{ZZ}}^{-1/2} \tag{2.142}$$

as in section 2.5.2. In analogy to equation (2.107), this is equivalent to

$$\begin{aligned}\mathbf{Q}_{\mathbf{ZZ}}^{\text{post}} &= \mathbf{Q}_{\mathbf{ZZ}}^{1/2}\left[\mathbf{I} + \mathbf{V}_{\mathbf{Z}}\boldsymbol{\Lambda}^{1/2}\mathbf{V}_{\mathbf{P}}^{T}\mathbf{V}_{\mathbf{P}}\boldsymbol{\Lambda}^{1/2}\mathbf{V}_{\mathbf{Z}}^{T}\right]^{-1}\mathbf{Q}_{\mathbf{ZZ}}^{1/2} \\ &= \mathbf{Q}_{\mathbf{ZZ}}^{1/2}\left[\mathbf{I} + \mathbf{V}_{\mathbf{Z}}\boldsymbol{\Lambda}\mathbf{V}_{\mathbf{Z}}^{T}\right]^{-1}\mathbf{Q}_{\mathbf{ZZ}}^{1/2} \\ &= \mathbf{Q}_{\mathbf{ZZ}}^{1/2}\left[\mathbf{I} - \mathbf{V}_{\mathbf{Z}}\boldsymbol{\Upsilon}\mathbf{V}_{\mathbf{Z}}^{T}\right]\mathbf{Q}_{\mathbf{ZZ}}^{1/2} \\ &= \mathbf{Q}_{\mathbf{ZZ}} - \mathbf{Q}_{\mathbf{ZZ}}^{1/2}\mathbf{V}_{\mathbf{Z}}\boldsymbol{\Upsilon}\mathbf{V}_{\mathbf{Z}}^{T}\mathbf{Q}_{\mathbf{ZZ}}^{1/2}\end{aligned} \tag{2.143}$$

with the matrices $\mathbf{V}_{\mathbf{Z}}$ and $\boldsymbol{\Lambda}^{1/2}$ from equation (2.106) and the matrices $\boldsymbol{\Lambda}$ and $\boldsymbol{\Upsilon}$ from equation (2.90).

While the spectral decomposition of algorithm 7 ($\mathbf{ED}_r$) can't be used in this context, the randomized singular value decomposition of algorithm 8 ($\mathbf{SVD}_r$) provides matrices $\mathbf{V}_{\mathbf{Z},r}$ and $\boldsymbol{\Lambda}_r$ that are replacements for $\mathbf{V}_{\mathbf{Z}}$ and $\boldsymbol{\Lambda}$, possibly of smaller size. If $\mathbf{V}_{\mathbf{Z},r}$ has full rank, then it contains an orthogonal basis of the observation space. If this is not the case, we extend it to contain such a basis and denote the resulting matrix with $\mathbf{V}_{\mathbf{Z},r}^{+}$. In analogy to equation (2.121), we may then write $\mathbf{Q}_{\mathbf{ZZ}}^{\text{post}}$ as

$$\mathbf{Q}_{\mathbf{ZZ}}^{\text{post}} = \mathbf{Q}_{\mathbf{ZZ}}^{1/2}\mathbf{V}_{\mathbf{Z},r}^{+}\left[\mathbf{I} - \boldsymbol{\Upsilon}_r^{+}\right]\mathbf{V}_{\mathbf{Z},r}^{T}\mathbf{Q}_{\mathbf{ZZ}}^{1/2}, \tag{2.144}$$

where $\boldsymbol{\Upsilon}_r^{+}$ is again the matrix $\boldsymbol{\Upsilon}_r$ padded with zeros, and we may again define a matrix

$$\mathbf{L}_{\text{post}} \coloneqq \mathbf{Q}_{\mathbf{ZZ}}^{1/2}\mathbf{V}_{\mathbf{Z},r}^{+}\left[\mathbf{I} - \boldsymbol{\Upsilon}_r^{+}\right]^{1/2}\left[\mathbf{V}_{\mathbf{Z},r}^{+}\right]^{T} \tag{2.145}$$

to obtain a decomposition of the form

$$\mathbf{Q}_{\mathbf{ZZ}}^{\text{post}} = \mathbf{L}_{\text{post}}\mathbf{L}_{\text{post}}^{T}. \tag{2.146}$$

The inverse of $\mathbf{L}_{\text{post}}$ is

$$\begin{aligned}\mathbf{L}_{\text{post}}^{-1} &= \left[\mathbf{Q}_{\mathbf{ZZ}}^{1/2}\mathbf{V}_{\mathbf{Z},r}^{+}\left[\mathbf{I} - \boldsymbol{\Upsilon}_r^{+}\right]^{1/2}\left[\mathbf{V}_{\mathbf{Z},r}^{+}\right]^{T}\right]^{-1} \\ &= \mathbf{V}_{\mathbf{Z},r}^{+}\left[\mathbf{I} - \boldsymbol{\Upsilon}_r^{+}\right]^{-1/2}\left[\mathbf{V}_{\mathbf{Z},r}^{+}\right]^{T}\mathbf{Q}_{\mathbf{ZZ}}^{-1/2} \\ &= \mathbf{V}_{\mathbf{Z},r}^{+}\left[\mathbf{I} + \boldsymbol{\Lambda}_r^{+}\right]^{1/2}\left[\mathbf{V}_{\mathbf{Z},r}^{+}\right]^{T}\mathbf{Q}_{\mathbf{ZZ}}^{-1/2},\end{aligned} \tag{2.147}$$

Algorithm 11: Randomized Test of Unbiasedness (Measurement Version; $\mathbf{TU}_r^{\mathbf{Z}}$)

Input: MAP estimate $\mathbf{P}_{\text{map}}$, Observations $\mathbf{Z}$, matrix of eigenvectors $\mathbf{V}_r$, matrix of eigenvalues $\mathbf{\Lambda}_r$

Output: Normalized residual $\Delta_{\mathbf{Z}}$

$\mathbf{Z}_{\text{map}} \coloneqq \mathcal{G}(\mathbf{P}_{\text{map}})$ [compute model outcome];

$\mathbf{R} \coloneqq \mathbf{Z} - \mathbf{Z}_{\text{map}}$ [compute residual of estimate];

$\mathbf{W} \coloneqq \mathbf{Q}_{\mathbf{ZZ}}^{-1/2}\mathbf{R}$ [multiply residual with $\mathbf{Q}_{\mathbf{ZZ}}^{-1/2}$];

for $1 \leq i \leq r$ **do**

  $\mathbf{W}_i \coloneqq (\mathbf{V}_r)_i \left[(\mathbf{V}_r)_i\right]^T \mathbf{R}$ [project $\mathbf{W}$ onto eigenspaces];

end

$\mathbf{W}_{\text{rem}} \coloneqq \mathbf{W} - \sum_{i=1}^{r} \mathbf{W}_i$ [compute remainder that is not in span of eigenvectors];

$\Delta_{\mathbf{Z}} \coloneqq \mathbf{W}_{\text{rem}} + \sum_{i=1}^{r} \left[1 + \lambda_i\right]^{1/2} \mathbf{W}_i$ [scale projections by appropriate factor];

return $\Delta_{\mathbf{Z}}$;

and

$$\Delta_{\mathbf{Z}} \coloneqq \mathbf{L}_{\text{post}}^{-1}\left[\mathbf{Z} - \mathcal{G}(\mathbf{P}_{\text{map}})\right] \sim \mathcal{N}(\mathbf{I}, \mathbf{0}) \tag{2.148}$$

has to hold for the decorrelated or normalized residual $\Delta_{\mathbf{Z}}$. This means the equations (2.131) to (2.134) may also be applied to $\Delta_{\mathbf{Z}}$ instead of $\Delta_{\mathbf{P}}$, and the components of the normalized residual should also have mean zero, variance one, skewness zero and a kurtosis of three.

Remark 16 *The optimization is restricted to the subspace of the parameter space that is spanned by the eigenvectors of* $\mathbf{M}_{like}$*. Since this is often only a small fraction of the full parameter space, most components of the normalized error* $\Delta_{\mathbf{P}}$ *from the previous section are solely determined by the reference* $\widehat{\mathbf{P}}$ *and therefore insensitive to the result* $\mathbf{P}_{map}$*, in particular those corresponding to high-frequency modes not present in the solution. As a consequence, the normalized error may fail to be an adequate measure of goodness-of-fit. See the applications in section 6.1.1 for an example.*

The normalized residual $\Delta_{\mathbf{Z}}$ *has a much smaller size, however, and by construction almost all of its components can be expected to be sensitive to the solution* $\mathbf{P}_{map}$*. Additionally,* $\Delta_{\mathbf{Z}}$ *is also available for real-world applications that don't rely on synthetic data, and therefore is a more appropriate choice for a measure of goodness-of-fit.*

2.7 Revisiting the Preconditioner

The decomposition of the posterior covariance matrix

$$\mathbf{Q}_{\mathbf{PP}}^{\text{post}} = \mathbf{Q}_{\mathbf{PP}}^{1/2}\left[\mathbf{I} - \mathbf{V}\mathbf{\Upsilon}\mathbf{V}^T\right]\mathbf{Q}_{\mathbf{PP}}^{1/2}, \tag{2.149}$$

as it was derived in equation (2.90), may also be computed for parameter vector tuples other than $\mathbf{P}_{\text{map}}$. The result is

$$\begin{aligned}\mathbf{Q}_{\mathbf{PP}}^{(\mathbf{P}_i)} &\coloneqq \left[\mathbf{Q}_{\mathbf{PP}}^{-1} + \left[\mathbf{H}_{\mathbf{ZP}}(\mathbf{P}_i)\right]^T \mathbf{Q}_{\mathbf{ZZ}}^{-1}\mathbf{H}_{\mathbf{ZP}}(\mathbf{P}_i)\right]^{-1} \\ &= \mathbf{Q}_{\mathbf{PP}}^{1/2}\left[\mathbf{I} - \mathbf{V}(\mathbf{P}_i)\boldsymbol{\Upsilon}(\mathbf{P}_i)\left[\mathbf{V}(\mathbf{P}_i)\right]^T\right]\mathbf{Q}_{\mathbf{PP}}^{1/2},\end{aligned} \tag{2.150}$$

where $\mathbf{P}_i$ is one of the iterations. We have temporally included $\mathbf{P}_i$ as an argument of $\mathbf{H}_{\mathbf{ZP}}$ and the matrices of the decomposition to emphasize the dependence.

The matrix $\mathbf{Q}_{\mathbf{PP}}^{(\mathbf{P}_i)}$ is an estimate of $\mathbf{Q}_{\mathbf{PP}}^{\text{post}}$ at the location $\mathbf{P}_i$, generated through linearization of the objective function L around $\mathbf{P}_i$. As such, it is used to generate the step direction of the Gauss-Newton scheme, compare section 3.3.1. Under the assumption that the Hessian of L doesn't change too drastically in the course of the optimization, the matrix

$$M \coloneqq \left[\mathbf{Q}_{\mathbf{PP}}^{(\mathbf{P}_0)}\right]^{-1} \tag{2.151}$$

may be used as a preconditioner just as $\mathbf{Q}_{\mathbf{PP}}^{-1}$ has been used in sections 2.3 and 2.3.5. In contrast to equation (2.59), the resulting preconditioned residual for the i-th iteration becomes

$$\begin{aligned}\mathbf{T}_i &= -\mathbf{Q}_{\mathbf{PP}}^{(\mathbf{P}_0)}\nabla L|_{\mathbf{P}_{i-1}} \\ &= -\mathbf{Q}_{\mathbf{PP}}^{(\mathbf{P}_0)}\left[\mathbf{Q}_{\mathbf{PP}}^{-1}\left[\mathbf{P}_{i-1} - \mathbf{P}^*\right] + \mathbf{H}_{\mathbf{ZP}}^T\mathbf{Q}_{\mathbf{ZZ}}^{-1}\left[\mathbf{Z} - \mathcal{G}(\mathbf{P}_{i-1})\right]\right] \\ &= -\mathbf{Q}_{\mathbf{PP}}^{1/2}\left[\mathbf{I} - \mathbf{V}\boldsymbol{\Upsilon}\mathbf{V}^T\right]\left[\mathbf{Q}_{\mathbf{PP}}^{-1/2}\left[\mathbf{P}_{i-1} - \mathbf{P}^*\right] + \mathbf{Q}_{\mathbf{PP}}^{1/2}\mathbf{H}_{\mathbf{ZP}}^T\mathbf{Q}_{\mathbf{ZZ}}^{-1}\left[\mathbf{Z} - \mathcal{G}(\mathbf{P}_{i-1})\right]\right].\end{aligned} \tag{2.152}$$

In section 2.3.5, multiplications with $\mathbf{Q}_{\mathbf{PP}}^{-1}$ were avoided by caching and reusing the relevant matrix-vector products, and a similar approach is also possible for this preconditioner. Setting $\mathbf{V}_i \coloneqq \mathbf{Q}_{\mathbf{PP}}^{-1/2}\mathbf{P}_i$ and $\mathbf{V}^* \coloneqq \mathbf{Q}_{\mathbf{PP}}^{-1/2}\mathbf{P}^*$, we can define

$$\begin{aligned}\mathbf{A}_i &\coloneqq \mathbf{Q}_{\mathbf{PP}}^{1/2}\mathbf{R}_i \\ &= \mathbf{Q}_{\mathbf{PP}}^{1/2}\left[-\mathbf{Q}_{\mathbf{PP}}^{-1}\left[\mathbf{P}_{i-1} - \mathbf{P}^*\right] + \mathbf{H}_{\mathbf{ZP}}^T\mathbf{Q}_{\mathbf{ZZ}}^{-1}\left[\mathbf{Z} - \mathcal{G}(\mathbf{P}_{i-1})\right]\right] \\ &= -\mathbf{Q}_{\mathbf{PP}}^{-1/2}\left[\mathbf{P}_{i-1} - \mathbf{P}^*\right] + \mathbf{Q}_{\mathbf{PP}}^{1/2}\mathbf{H}_{\mathbf{ZP}}^T\mathbf{Q}_{\mathbf{ZZ}}^{-1}\left[\mathbf{Z} - \mathcal{G}(\mathbf{P}_{i-1})\right] \\ &= -\left[\mathbf{V}_{i-1} - \mathbf{V}^*\right] + \mathbf{Q}_{\mathbf{PP}}^{1/2}\mathbf{H}_{\mathbf{ZP}}^T\mathbf{Q}_{\mathbf{ZZ}}^{-1}\left[\mathbf{Z} - \mathcal{G}(\mathbf{P}_{i-1})\right].\end{aligned} \tag{2.153}$$

If both $\mathbf{V}_{i-1}$ and $\mathbf{V}^*$ are known, $\mathbf{A}_i$ can be computed using a multiplication with $\mathbf{Q}_{\mathbf{PP}}^{1/2}$, and a multiplication with $\mathbf{Q}_{\mathbf{PP}}^{-1}$ or $\mathbf{Q}_{\mathbf{PP}}^{-1/2}$ is unnecessary. Defining $\mathbf{B}_i \coloneqq \mathbf{Q}_{\mathbf{PP}}^{-1/2}\mathbf{T}_i$ leads to

$$\mathbf{B}_i = \left[\mathbf{I} - \mathbf{V}\boldsymbol{\Upsilon}\mathbf{V}^T\right]\mathbf{A}_i \tag{2.154}$$

and

$$\mathbf{T}_i = \mathbf{Q}_{\mathbf{PP}}^{1/2}\left[\mathbf{I} - \mathbf{V}\boldsymbol{\Upsilon}\mathbf{V}^T\right]\mathbf{A}_i = \mathbf{Q}_{\mathbf{PP}}^{1/2}\mathbf{B}_i,$$

i.e. the preconditioned residual $\mathbf{T}_i$ can be computed from $\mathbf{A}_i$ through a second multiplication with $\mathbf{Q}_{\mathbf{PP}}^{1/2}$.

The definitions of $\mathbf{A}_i$ and $\mathbf{B}_i$ imply

$$\mathbf{A}_i^T \mathbf{B}_i = \mathbf{R}_i^T \mathbf{T}_i, \tag{2.155}$$

and

$$\mathbf{A}_i^T \mathbf{B}_{i-1} = \mathbf{R}_i^T \mathbf{T}_{i-1}, \tag{2.156}$$

which means that the conjugation factors can be evaluated through the formulas

$$\begin{aligned} \beta_i^{\mathrm{FR}} &= \frac{\mathbf{A}_i^T \mathbf{B}_i}{\mathbf{A}_{i-1}^T \mathbf{B}_{i-1}} \\ \beta_i^{\mathrm{PR}} &= \frac{\mathbf{A}_i^T \left[\mathbf{B}_i - \mathbf{B}_{i-1}\right]}{\mathbf{A}_{i-1}^T \mathbf{B}_{i-1}} \\ \beta_i^{\mathrm{HS}} &= -\frac{\mathbf{A}_i^T \left[\mathbf{B}_i - \mathbf{B}_{i-1}\right]}{\mathbf{D}_{i-1}^T \left[\mathbf{T}_i - \mathbf{T}_{i-1}\right]}, \end{aligned} \tag{2.157}$$

avoiding the assembly of $\mathbf{R}_i$. If further $\mathbf{W}_{i-1} \coloneqq \mathbf{Q}_{\mathbf{PP}}^{-1/2} \mathbf{D}_{i-1}$ is known for the previous step direction $\mathbf{D}_{i-1}$, then both the new step direction $\mathbf{D}_i$ and $\mathbf{W}_i \coloneqq \mathbf{Q}_{\mathbf{PP}}^{-1/2} \mathbf{D}_i$ can be computed through

$$\begin{aligned} \mathbf{D}_i &= \mathbf{T}_i + \beta_i \mathbf{D}_{i-1} \\ \mathbf{W}_i &= \mathbf{B}_i + \beta_i \mathbf{D}_{i-1}, \end{aligned} \tag{2.158}$$

where β_i is one of the conjugation factors given above. Finally, evaluation of the objective function for the line search is possible through the identity

$$\left\| \mathbf{P}_{i-1} + \alpha \mathbf{D}_i - \mathbf{P}^* \right\|_{\mathbf{Q}_{\mathbf{PP}}^{-1}} = \left\| \mathbf{V}_{i-1} + \alpha \mathbf{W}_i - \mathbf{V}^* \right\|_{\mathbf{I}}, \tag{2.159}$$

again without application of $\mathbf{Q}_{\mathbf{PP}}^{-1}$ or $\mathbf{Q}_{\mathbf{PP}}^{-1/2}$.

Consequently, the described method requires neither the assembly of the sensitivity matrix $\mathbf{H}_{\mathbf{ZP}}$ nor multiplication with $\mathbf{Q}_{\mathbf{PP}}^{-1}$, just as the PCG scheme using $\mathbf{Q}_{\mathbf{PP}}^{-1}$ as preconditioner, and most arguments given in sections 2.3 and 2.3.5 also hold for this second scheme. Algorithm 12 ($\mathbf{PCG}_c^{\mathrm{post}}$) details the computational steps of this method.

Remark 17 *Compared to the prior preconditioned CG method given by algorithm 6 ($\mathbf{PCG}_c$), this second method has considerable initial costs, since $\mathbf{Q}_{\mathbf{PP}}^{(\mathbf{P}_0)}$, the estimate of the posterior covariance matrix at $\mathbf{P}_0$, has to be decomposed before the first iteration. Subsequent iterations then again profit from the "negative cost" of the preconditioner. Whether it is advantageous to perform this initial decomposition depends on the relative convergence rates of the two methods and the number of eigenvalues that have to be retrieved.*

Algorithm 12: Caching Posterior Preconditioned Conjugate Gradients ($\mathbf{PCG}_c^{\text{post}}$)

Input: initial value $\mathbf{P}_0$, auxiliary variable $\mathbf{V}_0$, $\mathbf{R}_0 = 0$, $\mathbf{T}_0 = 0$, $\mathbf{D}_0 = 0$, $\mathbf{W}_0 = 0$, stopping criterion

Output: estimate of MAP point $\mathbf{P}_{\text{map}}$

compute decomposition $\mathbf{Q}_{\mathbf{PP}}^{(\mathbf{P}_0)} = \mathbf{Q}_{\mathbf{PP}}^{1/2}\left[\mathbf{I} - \mathbf{V}_r \boldsymbol{\Upsilon}_r \mathbf{V}_r^T\right]\mathbf{Q}_{\mathbf{PP}}^{1/2}$, e.g. using algorithm 8;

$i \coloneqq 0$ [set index];

repeat

  $i \to i+1$ [shift index];

  $\mathbf{A}_i \coloneqq -\left[\mathbf{V}_{i-1} - \mathbf{V}^*\right] + \mathbf{Q}_{\mathbf{PP}}^{1/2}\mathbf{H}_{\mathbf{ZP}}^T\mathbf{Q}_{\mathbf{ZZ}}^{-1}\left[\mathbf{Z} - \mathcal{G}(\mathbf{P}_{i-1})\right]$ [compute $\mathbf{Q}_{\mathbf{PP}}^{1/2}\mathbf{R}_i$];

  $\mathbf{B}_i \coloneqq \left[\mathbf{I} - \mathbf{V}_r \boldsymbol{\Upsilon}_r \mathbf{V}_r^T\right]\mathbf{A}_i$ [store partial preconditioned residual];

  $\mathbf{T}_i \coloneqq \mathbf{Q}_{\mathbf{PP}}^{1/2}\mathbf{B}_i$ [calculate preconditioned residual];

  $\beta_i \coloneqq \text{orthogonalize}(\mathbf{A}_{i-1}, \mathbf{B}_{i-1}, \mathbf{A}_i, \mathbf{B}_i, \mathbf{D}_{i-1})$ [compute conjugation factor];

  $(\mathbf{D}_i, \mathbf{W}_i) \coloneqq (\mathbf{T}_i, \mathbf{B}_i) + \beta_i\,(\mathbf{D}_{i-1}, \mathbf{W}_{i-1})$ [set directions];

  $\alpha_i \coloneqq \text{linesearch}(\mathbf{V}_{i-1}, \mathbf{W}_i)$ [compute step width];

  $(\mathbf{P}_i, \mathbf{V}_i) \coloneqq (\mathbf{P}_{i-1}, \mathbf{V}_{i-1}) + \alpha_i\,(\mathbf{D}_i, \mathbf{W}_i)$ [define i-th iterations];

until *converged*;

$\mathbf{P}_{\text{map}} \coloneqq \mathbf{P}_i$ [accept final iteration];

return $\mathbf{P}_{map}$;

The estimate $\mathbf{Q}_{\mathbf{PP}}^{(\mathbf{P}_0)}$ is the linearization of the Hessian, and therefore the first step of the method solves the linearization of the Least Squares problem. If the problem is linear or only weakly nonlinear, then this is equivalent to the solution of the normal equations with subsequent iterative refinement. If the problem is moderately or strongly nonlinear, then it is likely that the Hessian information has to be updated at some point, and in this case the method has to be restarted. In our experience the improved convergence rate in comparison with algorithm 6 ($\mathbf{PCG}_c$) is often not worth the additional cost of several matrix decompositions. We therefore only consider the limiting case of restarting the method in each iteration, which turns the posterior preconditioned CG method into the randomized Gauss-Newton scheme as it will be discussed in section 3.3.2.

2.8 Summary and Discussion

To summarize the content of the previous sections, the proposed method consists in performing the following steps:

Acquisition of Data In real-world applications, the measurement vector tuple $\mathbf{Z}$ is the result of field experiments or other observations of a physical system. These observations have to be accompanied by assumptions about the measurement error in the form of $\mathbf{Q}_{\mathbf{ZZ}}$, the structure of the examined domain in the form of $\mathbf{P}^*$ and $\mathbf{Q}_{\mathbf{PP}}$, and the physical process in the form of the forward model $\mathcal{F}$.

If the method is applied to synthetic test cases as in chapter 6, the artificial measurements have to follow the correct distribution. This can be guaranteed by using algorithm 1 to construct a synthetic reference $\widehat{\mathbf{P}}$, computing the model outcome $\mathcal{G}(\widehat{\mathbf{P}})$, and adding noise with distribution $\mathcal{N}(\mathbf{Q}_{\mathbf{ZZ}}, \mathbf{0})$ to the result.

Preconditioned Conjugate Gradients The observations $\mathbf{Z}$ define the value of the objective function L, equation (2.33), for any given parameter vector tuple $\mathbf{P}$. One of the methods of section 2.3 may then be used to minimize L and obtain the Maximum A Posteriori estimate $\mathbf{P}_{\text{map}}$. Of the presented methods the caching PCG method of section 2.3.5 is consistently the fastest, since it has both the highest experimental order of convergence (EOC) and the lowest computational cost per iteration.

Randomized Methods Alternatively, the posterior preconditioned scheme of section 2.7 may be used to compute $\mathbf{P}_{\text{map}}$. As discussed in remark 17, this is only useful if the improvement of the convergence rate is large enough to amortize the decomposition that is required during the initial setup. Another possible choice are the randomized Gauss-Newton and Levenberg-Marquardt schemes that will be discussed in section 3.3.2.

Uncertainty Quantification The PCG method doesn't provide information about the uncertainty of the estimate $\mathbf{P}_{\text{map}}$. In the context of Bayesian inference, a natural measure of uncertainty is the posterior covariance matrix $\mathbf{Q}_{\mathbf{PP}}^{\text{post}}$, and the two randomized algorithms in section 2.5 can be used to obtain a spectral decomposition of this matrix. This decomposition can be less expensive than a classical full assembly of $\mathbf{H}_{\mathbf{ZP}}$ if there is sufficient autocorrelation between the state observations, e.g. in time series or imaging methods.

A Posteriori Analysis If the inversion is based on synthetic data, then the spectral decomposition of $\mathbf{Q}_{\mathbf{PP}}^{\text{post}}$ can be used to decorrelate the difference between the estimate $\mathbf{P}_{\text{map}}$ and the synthetic reference $\widehat{\mathbf{P}}$, compare section 2.6.2. This decorrelated error can then be used in statistical tests to check the quality of the inversion result. As shown in section 2.6.3, the same steps can be applied to the measurement residual, and this second statistical analysis is also available in the case of real-world data.

Sample Generation for Posterior Distribution If the statistical tests didn't reject the estimate $\mathbf{P}_{\text{map}}$ and the posterior covariance matrix $\mathbf{Q}_{\mathbf{PP}}^{\text{post}}$, then they can be used to draw samples form the posterior distribution as detailed in section 2.6.1. The resulting realizations of the posterior distribution can then be used in simulations, which in turn provide information about the probability distribution of the system states.

2.8.1 Computational Costs

One step of the Conjugate Gradients method, algorithm 3 (**CG**), requires three runs of the forward model $\mathcal{F}$, one for evaluating the objective function for the current estimate and two for the quadratic line search if the gradient isn't reused for the directional derivative. For the same reason three multiplications with $\mathbf{Q}_{\mathbf{PP}}^{-1}$ are required if intermediate results aren't cached. The adjoint problem 6 is solved once to compute the search direction.

The caching PCG method, algorithm 6 ($\mathbf{PCG}_c$), eliminates all multiplications with $\mathbf{Q}_{\mathbf{PP}}^{-1}$ and uses one multiplication with $\mathbf{Q}_{\mathbf{PP}}$ instead. This reduces the cost per iteration significantly, since the effort required for multiplying with $\mathbf{Q}_{\mathbf{PP}}^{-1}$ is usually higher than that for multiplication with $\mathbf{Q}_{\mathbf{PP}}$, often by more than one order of magnitude [61]. As a result, the preconditioner has "negative cost". This is also true when comparing with a caching variant of the unpreconditioned scheme, since at least one multiplication with $\mathbf{Q}_{\mathbf{PP}}^{-1}$ per step remains.

The PCG scheme using an estimate of the posterior covariance matrix, algorithm 12 ($\mathbf{PCG}_c^{\text{post}}$), additionally requires a spectral decomposition of $\mathbf{Q}_{\mathbf{PP}}^{(\mathbf{P}_i)}$ every time the method is restarted. The fraction of iterations where this happens is a number $0 < \eta < 1$, with the exact value of η depending on the chosen strategy for restarting the method. On average, one step of the method therefore requires an additional ηr runs of the forward model $\mathcal{F}$, ηr runs of the adjoint model $\mathcal{F}^\dagger$ and $2\eta r$ multiplications with $\mathbf{Q}_{\mathbf{PP}}^{1/2}$. Furthermore, the multiplication with $\mathbf{Q}_{\mathbf{PP}}$ has to be replaced by two multiplications with $\mathbf{Q}_{\mathbf{PP}}^{1/2}$, leading to $2 + 2\eta r$ multiplications with $\mathbf{Q}_{\mathbf{PP}}^{1/2}$ in total.

Table 2.1 serves as a structured summary for this analysis. The next chapter will introduce three variants of the Gauss-Newton method, algorithms 14 (**GN**) and 15 ($\mathbf{GN}_{\text{CE}}$) in section 3.3.1 and algorithm 16 ($\mathbf{GN}_r$) in section 3.3.2. We include the computational cost of these methods in table 2.1 to make comparison easier, but refer to these later sections for details about the methods.

The classical Gauss-Newton method, algorithm 14 (**GN**), assembles the full sensitivity matrix $\mathbf{H}_{\mathbf{ZP}}$, which requires the solution of $N \coloneqq \prod_{j=1}^{n_{\mathbf{Z}}} n_{\mathbf{z}_j}$ adjoint problems. The computation of the step direction consists in multiplication with a matrix that is too large to be inverted, and therefore an inner iterative method is required. As a result, a very larger number of multiplications with $\mathbf{Q}_{\mathbf{PP}}^{-1}$ is needed, which makes the method unfeasible for parameter fields with high resolution.

The modified Gauss-Newton method, algorithm 15 ($\mathbf{GN}_{\text{CE}}$), is based on a reformulation that drastically reduces the size of the matrix that has to be inverted. The assembly of this smaller matrix requires N multiplications with $\mathbf{Q}_{\mathbf{PP}}$, and an additional multiplication is required to obtain $\mathbf{D}_i$. The randomized Gauss-Newton scheme, algorithm 16 ($\mathbf{GN}_r$), is a special case of the posterior preconditioned CG method with the choice $\eta = 1$, i.e. restarting the method after each iteration, and

	$\mathcal{F}$	$\mathcal{F}^\dagger$	$\mathbf{Q}_{\mathbf{PP}}^{-1}$	$\mathbf{Q}_{\mathbf{PP}}$	$\mathbf{Q}_{\mathbf{PP}}^{1/2}$
CG	3	1	3	0	0
$\mathbf{PCG}_c$	3	1	0	1	0
$\mathbf{PCG}_c^{\text{post}}$	$3+\eta r$	$1+\eta r$	0	0	$2+2\eta r$
GN	3	N	$\gg 1$	0	0
$\mathbf{GN}_{\text{CE}}$	3	N	0	$1+N$	0
$\mathbf{GN}_r$	$3+r$	$1+r$	0	0	$2+2r$

Table 2.1: Simulations of the forward model $\mathcal{F}$, simulations of the adjoint model $\mathcal{F}^\dagger$ and multiplications with $\mathbf{Q}_{\mathbf{PP}}^{-1}$, $\mathbf{Q}_{\mathbf{PP}}$ or $\mathbf{Q}_{\mathbf{PP}}^{1/2}$ needed for one step of each of the discussed methods. The three Gauss-Newton methods in the lower half of the table can be found in sections 3.3.1 and 3.3.2. $N \coloneqq \prod_{j=1}^{n_{\mathbf{Z}}} n_{\mathbf{z}_j}$ is the total number of observations in $\mathbf{Z}$, while $r \leq N$ is the rank of the approximate spectral decomposition of $\mathbf{M}_{\text{like}}$. η is the fraction of steps in which algorithm 12 ($\mathbf{PCG}_c^{\text{post}}$) is restarted.

therefore it needs the solution of r forward problems, the solution of r adjoint problems and $2+2r$ multiplications with $\mathbf{Q}_{\mathbf{PP}}^{1/2}$ for each iteration.

The cost of other operations can usually be neglected, with the possible exception of performing the spectral decomposition of the condensed matrix. Which of the discussed methods is fastest depends on several factors. While the prior preconditioned CG method has the lowest cost per iteration by a wide margin, the total number of iterations needed for convergence is equally important. The applications of chapter 6 demonstrate that the required number of iterations of this method can be sublinear or even pseudo-constant in the number of observations N, which means that the prior preconditioned CG method will be the most efficient when the number of observations N is large enough. A more detailed analysis has to take the specifics of the forward model $\mathcal{F}$ into account, i.e. the width, height and shape of the spectrum of $\mathbf{M}_{\text{like}}$ as discussed in section 2.5.2.

2.8.2 Memory Requirements

The Conjugate Gradients method, algorithm 3 (**CG**), requires storage for the iteration $\mathbf{P}_i$, the step direction $\mathbf{D}_i$ and the residuals $\mathbf{R}_{i-1}$ and $\mathbf{R}_i$, i.e. a total of four stored parameter vector tuples. The previous iteration $\mathbf{P}_{i-1}$ and previous direction $\mathbf{D}_{i-1}$ can use the same memory location as $\mathbf{P}_i$ and $\mathbf{D}_i$ respectively, since the old values are no longer needed once their replacement has been computed. The parameter tuple for the line search can also be stored in the location of $\mathbf{P}_i$ for similar reasons. The caching PCG method, algorithm 6 ($\mathbf{PCG}_c$), additionally stores the second sequence, i.e. the tuples $\mathbf{V}_i$, $\mathbf{W}_i$, $\mathbf{T}_{i-1}$, and $\mathbf{T}_i$. This doubles the number of parameter tuples that have to be kept in memory. The posterior preconditioned scheme, algorithm 12 ($\mathbf{PCG}_c^{\text{post}}$), has to store the iteration $\mathbf{P}_i$, the step direction $\mathbf{D}_i$,

	CG	$\mathbf{PCG}_c$	$\mathbf{PCG}_c^{\text{post}}$	**GN**	$\mathbf{GN}_{\text{CE}}$	$\mathbf{GN}_r$
Stored $\mathbf{P}$	4	8	$9+r$	$5+N$	$2+N$	$9+r$

Table 2.2: The number of parameter vector tuples that have to be kept in memory for each of the discussed methods. This includes iterations $\mathbf{P}_i$, step directions $\mathbf{D}_i$ and any auxiliary tuples that are required by the method. $N \coloneqq \prod_{j=1}^{n_{\mathbf{Z}}} n_{\mathbf{z}_j}$ is the total number of observations in $\mathbf{Z}$, while $r \leq N$ is the rank of the approximate spectral decomposition of $\mathbf{M}_{\text{like}}$. Note that each of the methods has to store one full solution of the forward problem for the Adjoint State method, and that these memory requirements may be more important than those of the parameter tuples if the models are transient.

the preconditioned residual $\mathbf{T}_i$, the auxiliary tuples $\mathbf{V}_i$, $\mathbf{W}_i$, $\mathbf{A}_{i-1}$, $\mathbf{A}_i$, $\mathbf{B}_{i-1}$ and $\mathbf{B}_i$, and the r eigenvectors of

$$\mathbf{M}_{\text{like}} \approx \mathbf{V}_r \mathbf{\Lambda}_r \mathbf{V}_r^T, \tag{2.160}$$

the spectral decomposition that is used to compute the step direction. This raises the number of parameter tuples that have to be kept in memory to $9+r$. The storage requirements for each of the methods are also listed in table 2.2.

The Gauss-Newton method, algorithm 14 (**GN**), has to store the iteration $\mathbf{P}_i$, the step direction $\mathbf{D}_i$ and the $N \coloneqq \prod_{j=1}^{n_{\mathbf{Z}}} n_{\mathbf{z}_j}$ columns of the sensitivity matrix $\mathbf{H}_{\mathbf{ZP}}$. The matrix appearing in the definition of the step direction is too large to be inverted, as discussed in the previous section. If we assume that a Conjugate Gradients method is used to compute the step direction, then at least three additional parameter tuples have to be kept in memory, i.e. $5 + N$ tuples in total. The modified Gauss-Newton method, algorithm 15 ($\mathbf{GN}_{\text{CE}}$), uses a reformulation that avoids this expensive operation. In this case the computation of the step direction $\mathbf{D}_i$ can be carried out without additional memory, i.e. $2 + N$ tuples have to be stored. The randomized Gauss-Newton method, algorithm 16 ($\mathbf{GN}_r$), is a special case of the posterior preconditioned CG method and requires the same amount of storage.

Only the Conjugate Gradients method and the caching PCG method have memory requirements that are low and constant, since the other methods store either the full sensitivity matrix $\mathbf{H}_{\mathbf{ZP}}$ or a low-rank representation of $\mathbf{Q}_{\mathbf{PP}}^{\text{post}}$. Only the caching PCG method, algorithm 6 ($\mathbf{PCG}_c$), combines these low memory requirements with mesh-independent convergence rates, compare section 2.3.4.

Remark 18 *In addition to the parameter vector tuples, each of the methods needs to store the complete results of a simulation of the model $\mathcal{F}$, since this information is required by the adjoint state method, compare section 4.4. If $\mathcal{F}$ is stationary, then such a simulation result will require the same memory as one parameter tuple or a small number of them, depending on the concrete choice of discretization. In this situation the caching PCG method, algorithm 6 ($\mathbf{PCG}_c$), will usually be able to use significantly higher grid resolutions than the Hessian-based methods. If the model*

$\mathcal{F}$ is transient, then the difference in storage requirements is smaller, since each individual parameter field needs significantly less memory than the whole transient evolution of the system states. However, this is almost automatically a situation where memory becomes scarce, and therefore the lower memory requirements of the caching PCG method may become relevant.

3 Alternative Approaches

As with any solution to a complex problem, several alternatives to the approach described in the previous chapter exist [3]. The ill-posed inverse problem 2 may be regularized in different ways, e.g. through a drastic reduction of the dimension of the parameter space, or by using another form of penalty term. The well-posed stochastic formulation of the inverse problem 4 may be treated in another way, for example arriving at other point estimates and variance estimates of the posterior distribution. And last but not least, a large number of optimization schemes that are applicable for the minimization problem 5 are available. The following sections provide examples of such alternative approaches, discussing their potential advantages and disadvantages. This presentation is not exhaustive, and additional approaches can be found in the cited literature.

3.1 Regularization Techniques

The Bayesian framework introduced in section 2.2 is not the only technique that may be used to regularize the ill-posed inverse problem 2. One of the classic regularization methods is Tikhonov regularization [77], which formulates the inverse problem as a minimization problem and stabilizes it through a penalty term:

$$\widetilde{L}(\mathbf{P}) = \frac{\gamma}{2} \|\mathbf{P}\|_{\mathbf{M}} + \frac{1}{2} \|\mathbf{Z} - \mathcal{G}(\mathbf{P})\|_2 \tag{3.1}$$

Here $\mathbf{M}$ is a matrix of the right size and γ is a factor that has to be large enough to regularize the inverse problem. In most applications $\mathbf{M}$ is the identity matrix, which is known as L_2 regularization.

A generalization of the classical Tikhonov regularization is

$$\widetilde{L}(\mathbf{P}) = \frac{\gamma}{2} \|\mathbf{P} - \mathbf{P}_0\|_{\mathbf{M}} + \frac{1}{2} \|\mathbf{Z} - \mathcal{G}(\mathbf{P})\|_{\mathbf{N}} \tag{3.2}$$

with an additional matrix $\mathbf{N}$ and a tuple of parameter vectors $\mathbf{P}_0$. Comparison with equation (2.33) shows that the objective function acquired from the posterior probability distribution is equivalent to choosing $\mathbf{P}_0 \coloneqq \mathbf{P}^*$, $\mathbf{M} \coloneqq \mathbf{Q}_{\mathbf{PP}}^{-1}$, $\mathbf{N} \coloneqq \mathbf{Q}_{\mathbf{ZZ}}^{-1}$ and $\gamma \coloneqq 1$. The main difference between the Bayesian approach and other variants of Tikhonov regularization is the fact that the norms are chosen based on the stochastic formulation of the forward problem and not imposed from the outside. Also note

that this specific choice does not require an educated guess for the scaling parameter γ, since it naturally arises from the formulation.

Two other classic regularization techniques are the Landweber iteration [37]

$$\mathbf{P}_i \coloneqq \mathbf{P}_{i-1} + \gamma \mathbf{H}_{\mathbf{ZP}}^T \left[\mathbf{Z} - \mathcal{G}(\mathbf{P}_i)\right], \tag{3.3}$$

with γ being a damping parameter, and Conjugate Gradients on the Normal Equations (CGNE) [25]. These methods are equivalent to applying Steepest Descent and Conjugate Gradients respectively to the Maximum Likelihood (ML) objective function

$$\widetilde{L}(\mathbf{P}) = \frac{\gamma}{2} \left\|\mathbf{Z} - \mathcal{G}(\mathbf{P})\right\|_2 \tag{3.4}$$

instead of the MAP objective function. Here the damping term

$$\gamma \coloneqq \sigma_{\mathbf{z}}^{-2} \tag{3.5}$$

is based on the standard deviation $\sigma_{\mathbf{z}}$ of the measurement error. This approach relies on the self-regularization properties of the iterative procedures instead of an explicit regularization term. Since Maximum Likelihood is the limiting case of MAP with uninformative prior, the methods described in chapter 2 can also be seen as an extension of this approach to nonlinear models with prior information.

Switching from MAP estimation to ML estimation consists in simply leaving out half of the objective function, compare equations (2.33) and (3.4), and therefore it is easy to present convergence results similar to those in figure 2.2 using the same implementation. Figure 3.1 shows the convergence behavior of the Landweber iteration and the CGNE method. In contrast to the MAP objective function, the ML objective function typically has a minimum that is exactly known a priori, since it has to be zero if at least one viable set of parameters $\mathbf{P}$ exists for the given observations $\mathbf{Z}$. While the methods converge to well-defined estimates, they are not physically plausible due to the lack of prior information. Figure 3.2 shows that the CGNE method tends to correct the parameter field in the direct vicinity of the observations while leaving almost all other parameters at the initial guess. This overfitting can be prevented by preconditioning with $\mathbf{Q}_{\mathbf{PP}}^{-1}$, effectively introducing the prior information by indirect means. The results are very similar to those of the PCG method, but the latter additionally provides a statistical interpretation for the estimate. Since both methods require approximately the same effort, there is no reason to drop the regularization term from the objective function.

Other regularization techniques that we can only mention in passing are maximum entropy regularization and total variation regularization. They differ from Tikhonov regularization in the choice of the penalty term that is added.

Another approach that may be used to regularize the inverse problem is a drastic reduction in the total number of parameters. If this number is sufficiently small, the solution of the inverse problem becomes over-determined instead of under-determined,

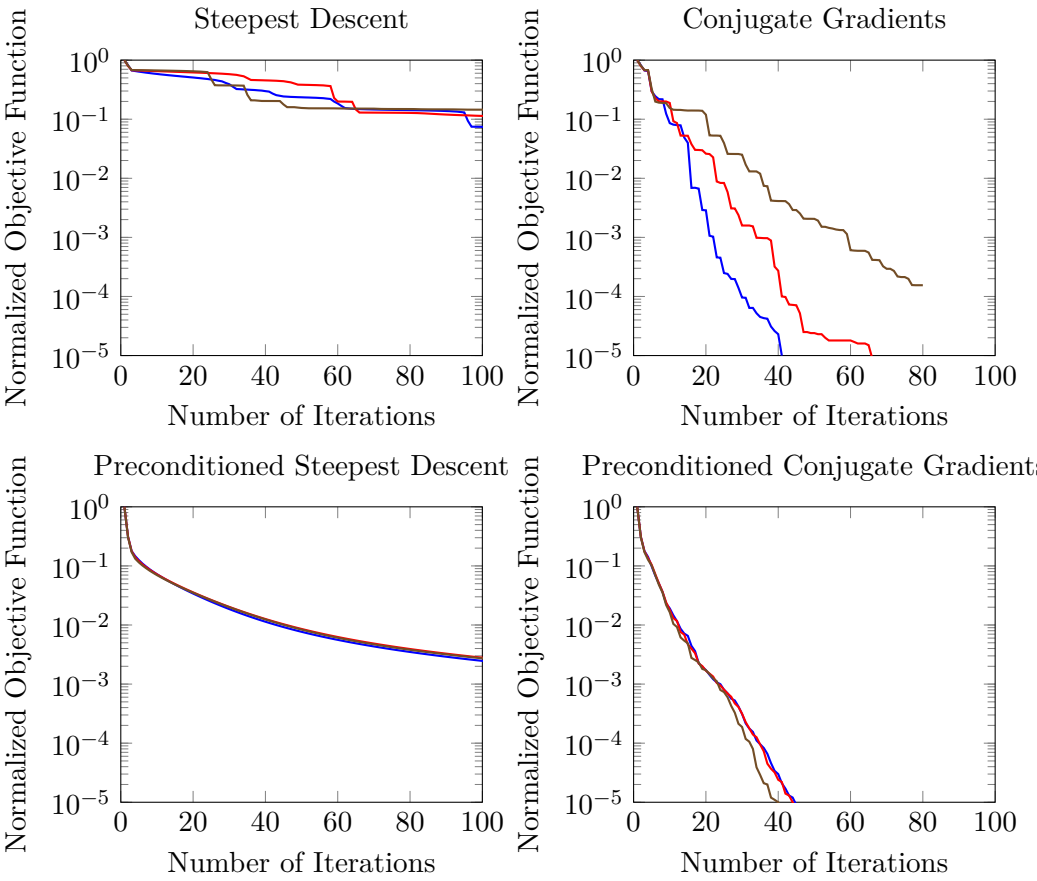

Figure 3.1: Convergence behavior of the methods from section 2.3 when applied to the Maximum Likelihood objective function, equation (3.4), *left:* Steepest Descent variants, *right:* Conjugate Gradients variants, *top:* original versions, *bottom:* preconditioned versions, with $\mathbf{Q}_{\mathbf{PP}}^{-1}$ as preconditioner. Number of discretization elements is $n_\Omega = 64 \times 64$ (——), 128×128 (——) and 256×256 (——). Convergence behavior is similar to that in figure 2.2, but the objective function converges to zero, which simplifies the analysis. The methods create sequences of iterations that may be seen as a solution to the inverse problem, but neither do the methods converge to the same set of parameters, compare figure 3.2, nor are the resulting parameter fields embedded in a context that allows interpretation as in the Bayesian framework.

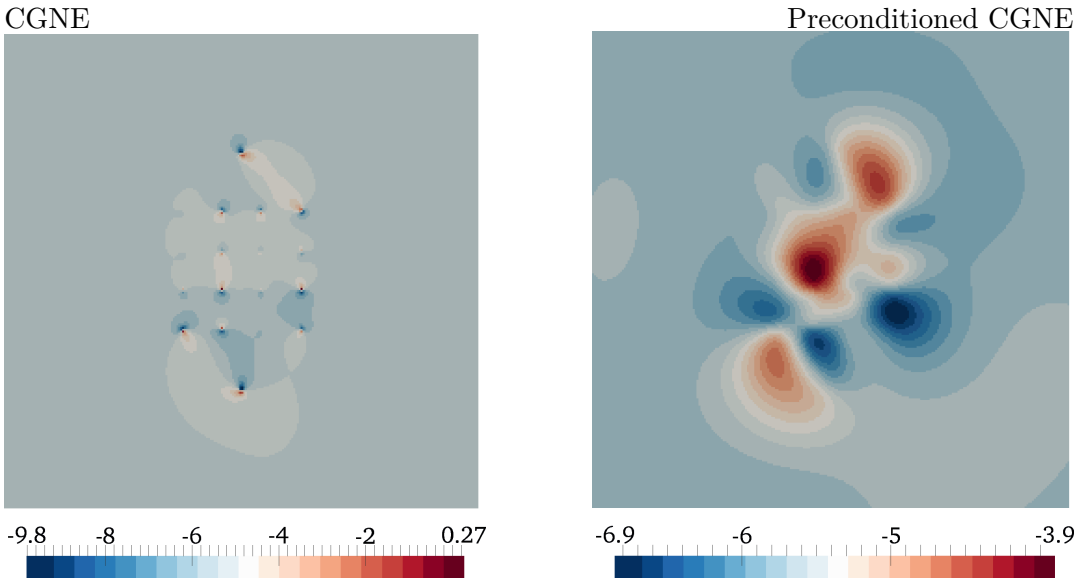

Figure 3.2: *Left:* The CGNE approach converges to a well-defined set of parameters, but due to the missing connection between individual parameters it modifies the parameter field only in the direct vicinity of the locations that have a strong influence on the observed values, which are the injection and extraction wells and the monitoring sites. *Right:* In the preconditioned version the prior information is introduced indirectly through the preconditioner $\mathbf{Q}_{\mathbf{PP}}^{-1}$. The resulting field strongly resembles the one attained through MAP estimation, compare figure 2.4, but doesn't provide error estimates.

and regularization techniques like the ones mentioned above are no longer necessary. Such a reduction of the number of parameters is always based on further assumptions about the parameter space or other simplifications.

There are several possibilities for such a reduction, and the simplest is modeling each parameter field s_i as homogeneous, i.e. with a single corresponding parameter value. There are applications of such parameterizations, but such a coarse representation will be highly inadequate if the spatial variability of the system is of relevance. A similar but more complex approach is the division of the domain Ω into a set of predetermined zones, with s_i being assumed piecewise constant in each of these zones. While this allows for more flexibility in the characterization of the parameter fields, the central issue of missing heterogeneity remains on the level of the individual zones. Another aspect that requires attention is the zonation process, since the quality of the inversion results can be highly sensitive to the subjective choice of zones.

A technique that leads to a similar reduction of the dimension of the parameter space is the introduction of pilot points [65, 23, 1]. At these locations, virtual observations of the parameter fields are introduced, and the parameters themselves are determined through interpolation of these virtual observations. This reduces the inverse prob-

lem to the estimation of virtual parameter observations from actual measurements. In contrast to the zonation method described above, the resulting parameter fields contain heterogeneity. Nevertheless, the parameter space is artificially restricted to a low dimension, with the interpolation polynomials of the pilot points as basis functions instead of the characteristic functions of the zones. Therefore, the placement of the virtual observations again introduces subjectivity and may influence the quality of the results.

A method that does not rely on the correct choice of zones or points and instead derives a low-dimensional parameter space from prior information is the Karhunen-Loève expansion, also known as principal component analysis (PCA) [51, 50]. The Principal Component Geostatistical Approach (PCGA) by *Lee and Kitanidis* [49, 46] is an example of this method. It uses the techniques described in section 2.5, but applied to the prior covariance matrix $\mathbf{Q_{PP}}$ instead of $\mathbf{M}_{\text{like}}$, to construct a low-dimensional representation of $\mathbf{Q_{PP}}$. As a result, the method requires half as many model solves as the methods described in section 2.5 for spectral decompositions, but it relies on the assumption that all relevant information about the forward model $\mathcal{F}$ is contained in its action on the first few eigenspaces of $\mathbf{Q_{PP}}$. Whether or not this approach is more efficient therefore depends on how many additional dimensions have to be considered to compensate for the fact that the decomposition doesn't take the characteristics of the model into account.

The main purpose of such low-dimensional parameterizations, apart from the resulting regularization, is the reduction of the computational cost of the inversion process through simplification of the prior. While such a reduction may be beneficial and indeed necessary for the rapid convergence of iterative methods, see the top row of figure 2.2, and the smaller number of parameters may make multiplication with $\mathbf{Q}_{\mathbf{PP}}^{-1}$ or its analogon feasible, this does not concern the PCG method as introduced in section 2.3. Synthetic tests show dimension-independent convergence of the method, see for example the bottom row of figure 2.2, and the computational cost for multiplication with $\mathbf{Q_{PP}}$ is negligible in comparison to the cost of the forward model even for the two-dimensional steady-state flow problems of section 6.1.1.

3.2 Treatment of the Inverse Problem

Several alternatives to MAP estimation exist for the handling of the stochastic inverse problem 4, and except for the Maximum Likelihood estimation from the last section these tend to be both more accurate and significantly more expensive.

The determination of the mean and higher moments of the posterior distribution can be interpreted as numerical integration, or quadrature, in the probability space. In this context MAP estimation may be compared to a low-order quadrature rule. Among the approaches that deliver more accurate results is stochastic collocation, which uses interpolation that, in contrast to the pilot point approach above, is applied

Algorithm 13: Metropolis-Hastings Markov Chain Monte Carlo (**MCMC**)

Input: initial value $\mathbf{P}_0$
Output: samples $\mathbf{P}_k, \mathbf{P}_{k+1}, \mathbf{P}_{k+2}, \ldots, k \gg 0$, from the posterior distribution
$i \coloneqq 0$ [set index];
repeat
  $i \to i+1$ [shift index];
  generate proposal $\widetilde{\mathbf{P}}$ from $q(\widetilde{\mathbf{P}}|\mathbf{P}_{i-1})$;
  calculate acceptance probability $r_0 \coloneqq \alpha(\widetilde{\mathbf{P}}|\mathbf{P}_{i-1})$;
  draw sample r from uniform distribution on $[0, 1]$;
  if $r < r_0$ **then**
    $\mathbf{P}_i \coloneqq \widetilde{\mathbf{P}}$ [accept proposal];
  else
    $\mathbf{P}_i \coloneqq \mathbf{P}_{i-1}$ [keep old iteration];
  end
until *enough samples*;

to the probability space instead of the physical domain Ω. This results in a surrogate for the posterior PDF, and information about the posterior distribution, like its mean or variance, may then be approximated by integrals involving this surrogate. Other approaches discretize the random variable in both the physical domain and the probability space, like stochastic Galerkin methods [4], or expand the random variable in a series, like generalized Polynomial Chaos (gPC) [82]. A large variety of methods based on similar or complementing ideas exists, but mentioning them all is beyond the scope of this work and we may only refer to the literature.

Apart from these deterministic methods, approaches based on the Monte Carlo method may be used to produce samples of the posterior distribution and analyze it through statistical information gathered from the samples. Since direct sampling from the posterior distribution would require a closed formulation for the posterior PDF, these methods are typically variants of the Markov-Chain Monte Carlo (MCMC) method. As a representative of these methods we may consider the Metropolis-Hastings MCMC [55, 38], see algorithm 13 (**MCMC**).

Here $q(\widetilde{\mathbf{P}}|\mathbf{P}_{i-1})$ is a proposal distribution that is used to generate a random walk, while the acceptance probability

$$\alpha(\widetilde{\mathbf{P}}|\mathbf{P}_{i-1}) \coloneqq \min\left(1, \frac{\exp(-2L(\widetilde{\mathbf{P}}))q(\mathbf{P}_{i-1}|\widetilde{\mathbf{P}})}{\exp(-2L(\mathbf{P}_{i-1}))q(\widetilde{\mathbf{P}}|\mathbf{P}_{i-1})}\right) \tag{3.6}$$

guarantees that the Markov chain converges to the desired equilibrium distribution, since $\exp(-2L(\mathbf{P}))$ is proportional to the posterior PDF as a function of $\mathbf{P}$. The generated sequence $\mathbf{P}_i$ follows the posterior distribution only after an initial phase known as burn-in period, and consecutive samples tend to be highly correlated [60].

Therefore, the generation of independent samples of the posterior distribution requires large parts of the above sequence to be discarded. Since each iteration of the algorithm requires the evaluation of the objective function L, and therefore the solution of the forward model $\mathcal{F}$, the Metropolis-Hastings MCMC is prohibitively expensive if uncorrelated samples are required.

While the high cost of MCMC methods may be significantly reduced, e.g. through multilevel techniques [43], the computational cost for both the deterministic and the nondeterministic methods mentioned tends to be several orders higher than that for the conceptually simpler MAP estimation. Consequently, Maximum A Posteriori parameter estimation may be viable in situations where the more accurate methods are too expensive.

Instead of determining a consistent set of parameters that can be used to parameterize the forward model, as it is the case with the inversion methods discussed so far, one may relax the requirements and only ask for the parameters that are most likely or most fitting at any given moment. The resulting parameter vectors are then functions of time that are updated each time new information becomes available, which is known as data assimilation [27]. While inversion methods are used to parameterize models and predict the behavior of modeled systems under different circumstances, data assimilation techniques are applied when predictions are needed for an ongoing process that is itself the source of the acquired data, e.g. in weather forecasts.

Remark 19 *In the Bayesian framework, the assimilation of new data may be interpreted as a case of parameter estimation with prior information based on the previous state of the parameters, and in this context data assimilation is also known as recursive Bayesian estimation for this reason [69]. Conversely, the inversion techniques can be seen as the application of data assimilation to all available data at once, also known as batch Bayesian estimation, and consequently Bayesian inversion methods and Bayesian data assimilation methods are closely linked.*

Data assimilation and inversion complement one another, and their advantages and disadvantages depend on the context in which they are applied. A fixed parameterization may allow a larger variety of applications, but the much larger number of constraints that have to be satisfied makes the inverse problem harder to solve. Choosing between data assimilation and inversion therefore involves considerations about the purpose of the parameter estimation, the requirements of the applications, and the extent of trust in the correctness of the forward model $\mathcal{F}$.

The most basic data assimilation method with widespread use is the Kalman Filter [42]. Expressed in the notation of the previous chapter, it updates the estimate of the mean of the posterior distribution by setting

$$\mathbf{P}_{\text{map}} \coloneqq \mathbf{P}^* + \mathbf{Q}_{\mathbf{PP}}\mathbf{H}_{\mathbf{ZP}}^T \left[\mathbf{H}_{\mathbf{ZP}}\mathbf{Q}_{\mathbf{PP}}\mathbf{H}_{\mathbf{ZP}}^T + \mathbf{Q}_{\mathbf{ZZ}}\right]^{-1} \left[\mathbf{Z} - \mathcal{G}(\mathbf{P})\right], \tag{3.7}$$

where $\mathbf{P}$ is the previous estimate for the mean, $\mathbf{Q_{PP}}$ is the previous estimate for the covariance matrix, $\mathbf{Z}$ is new data, and $\mathbf{H_{ZP}}$ is the linearized model. Afterwards, it updates the estimate of the covariance matrix according to equation (2.84). The Kalman Filter is known to be exact if the prior distribution is Gaussian and the model is linear, while a nonlinear model $\mathcal{F}$ would require linearization and an iterative approach of the form

$$\mathbf{P}_i \coloneqq \mathbf{P}_{i-1} - [\mathbf{P}_{i-1} - \mathbf{P}^*] + \mathbf{Q_{PP}}\mathbf{H}_{\mathbf{ZP}}^T \left[\mathbf{H_{ZP}}\mathbf{Q_{PP}}\mathbf{H}_{\mathbf{ZP}}^T + \mathbf{Q_{ZZ}}\right]^{-1} [\mathbf{Z} - \mathcal{G}(\mathbf{P}_{i-1})]\,. \tag{3.8}$$

Note that this iterative version is only mentioned to highlight the similarities between the Kalman update formula and the preconditioned CG scheme, compare equation (2.59). In practice, specialized filters suitable for models with nonlinearities and low regularity are preferred, e.g. the Ensemble Kalman Filter (EnKF) [27, 71].

3.3 Optimization Schemes

Possible solvers of the optimization problem of the MAP approach may be divided into two broad groups, those that restrict the system to feasible states, i.e. solve the forward problem in each iteration, and those that treat the model constraints as part of the optimization. The latter type of solver is often based on an augmented objective function, similar to the Lagrangian we have used for the derivative calculation in section 2.4, and may therefore also benefit from the preconditioner and randomized methods introduced in chapter 2. Nevertheless, we focus on methods that explicitly resolve the constraints, since these are the schemes that are typically employed in subsurface hydrology [54, 31, 72].

3.3.1 Gauss-Newton and Cokriging Equations

The Gauss-Newton method linearizes the objective function around a given parameter vector tuple $\mathbf{P}_{i-1}$,

$$L(\mathbf{P}_{i-1} + \delta_{\mathbf{P}}) \approx \frac{1}{2}\left\|\mathbf{P}_{i-1} + \delta_{\mathbf{P}} - \mathbf{P}^*\right\|^2_{\mathbf{Q}_{\mathbf{PP}}^{-1}} + \frac{1}{2}\left\|\mathbf{Z} - [\mathcal{G}(\mathbf{P}_{i-1}] + \mathbf{H_{ZP}}\delta_{\mathbf{P}})\right\|^2_{\mathbf{Q}_{\mathbf{ZZ}}^{-1}}, \tag{3.9}$$

and computes the gradient with respect to $\delta_{\mathbf{P}}$,

$$\begin{aligned}\nabla_{\delta_{\mathbf{P}}} L &\approx \mathbf{Q}_{\mathbf{PP}}^{-1}\left[\mathbf{P}_{i-1} + \delta_{\mathbf{P}} - \mathbf{P}^*\right] - \mathbf{H}_{\mathbf{ZP}}^T\mathbf{Q}_{\mathbf{ZZ}}^{-1}\left[\mathbf{Z} - [\mathcal{G}(\mathbf{P}_{i-1}) + \mathbf{H_{ZP}}\delta_{\mathbf{P}}]\right] \\ &= \left[\mathbf{Q}_{\mathbf{PP}}^{-1} + \mathbf{H}_{\mathbf{ZP}}^T\mathbf{Q}_{\mathbf{ZZ}}^{-1}\mathbf{H_{ZP}}\right]\delta_{\mathbf{P}} + \mathbf{Q}_{\mathbf{PP}}^{-1}\left[\mathbf{P}_{i-1} - \mathbf{P}^*\right] - \mathbf{H}_{\mathbf{ZP}}^T\mathbf{Q}_{\mathbf{ZZ}}^{-1}\left[\mathbf{Z} - \mathcal{G}(\mathbf{P})\right].\end{aligned} \tag{3.10}$$

The equation $\nabla_{\delta_{\mathbf{P}}} L = 0$ has to hold for the optimum of the linearized objective function, which leads to

$$\begin{aligned}\delta_{\mathbf{P}} &= -\left[\mathbf{Q}_{\mathbf{PP}}^{-1} + \mathbf{H}_{\mathbf{ZP}}^T\mathbf{Q}_{\mathbf{ZZ}}^{-1}\mathbf{H_{ZP}}\right]^{-1}\left[\mathbf{Q}_{\mathbf{PP}}^{-1}\left[\mathbf{P} - \mathbf{P}^*\right] - \mathbf{H}_{\mathbf{ZP}}^T\mathbf{Q}_{\mathbf{ZZ}}^{-1}\left[\mathbf{Z} - \mathcal{G}(\mathbf{P})\right]\right] \\ &= -\mathbf{Q}_{\mathbf{PP}}^{(\mathbf{P}_{i-1})}\left[\mathbf{Q}_{\mathbf{PP}}^{-1}\left[\mathbf{P} - \mathbf{P}^*\right] - \mathbf{H}_{\mathbf{ZP}}^T\mathbf{Q}_{\mathbf{ZZ}}^{-1}\left[\mathbf{Z} - \mathcal{G}(\mathbf{P})\right]\right],\end{aligned} \tag{3.11}$$

Algorithm 14: Gauss-Newton (**GN**)

Input: initial value $\mathbf{P}_0$, stopping criterion
Output: estimate of MAP point $\mathbf{P}_{\text{map}}$
$i \coloneqq 0$ [set index];
repeat
  $i \to i+1$ [shift index];
  calculate $\mathbf{H_{ZP}}$ [apply adjoint state method $\prod_{j=1}^{n_\mathbf{Z}} n_{\mathbf{z}_j}$ times];
  $\mathbf{Q}_{\mathbf{PP}}^{(\mathbf{P}_{i-1})} \coloneqq \left[\mathbf{Q}_{\mathbf{PP}}^{-1} + \mathbf{H}_{\mathbf{ZP}}^T \mathbf{Q}_{\mathbf{ZZ}}^{-1} \mathbf{H}_{\mathbf{ZP}}\right]^{-1}$ [calculate estimate of $\mathbf{Q}_{\mathbf{PP}}^{\text{post}}$];
  $\mathbf{D}_i \coloneqq -\mathbf{Q}_{\mathbf{PP}}^{(\mathbf{P}_{i-1})} \nabla L|_{\mathbf{P}_{i-1}}$ [compute direction];
  $\alpha_i \coloneqq \text{linesearch}(\mathbf{P}_{i-1}, \mathbf{D}_i)$ [compute step width];
  $\mathbf{P}_i \coloneqq \mathbf{P}_{i-1} + \alpha_i \mathbf{D}_i$ [define i-th iteration];
until *converged*;
$\mathbf{P}_{\text{map}} \coloneqq \mathbf{P}_i$ [accept final iteration];
return $\mathbf{P}_{map}$;

where we have used the notation $\mathbf{Q}_{\mathbf{PP}}^{(\mathbf{P}_{i-1})}$ for the estimate of $\mathbf{Q}_{\mathbf{PP}}^{\text{post}}$ at $\mathbf{P}_{i-1}$, as introduced in section 2.7. The scheme then sets

$$\mathbf{P}_i \coloneqq \mathbf{P}_{i-1} - \mathbf{Q}_{\mathbf{PP}}^{(\mathbf{P}_{i-1})} \left[\mathbf{Q}_{\mathbf{PP}}^{-1} \left[\mathbf{P}_{i-1} - \mathbf{P}^*\right] - \mathbf{H}_{\mathbf{ZP}}^T \mathbf{Q}_{\mathbf{ZZ}}^{-1} \left[\mathbf{Z} - \mathcal{G}(\mathbf{P}_{i-1})\right]\right] \tag{3.12}$$

as the next iteration, and repeats this process until a convergence criterion is met. Note that the PCG scheme we have discussed in section 2.3.5, algorithm 6 ($\mathbf{PCG}_c$), may also be interpreted as a variant of the Gauss-Newton algorithm using $\mathbf{Q_{PP}}$, the simplest estimate of $\mathbf{Q}_{\mathbf{PP}}^{\text{post}}$, instead of the local estimate $\mathbf{Q}_{\mathbf{PP}}^{(\mathbf{P}_{i-1})}$. Algorithm 14 (**GN**) summarizes the steps of the Gauss-Newton method. It is important to note that this formulation of the scheme requires the full assembly and subsequent inversion of a very large and dense matrix and therefore can't be used if the number of discretization elements n_Ω is large.

Alternatively, the method may be formulated as a fixpoint iteration:

$$\mathbf{P}_i \coloneqq \mathbf{Q}_{\mathbf{PP}}^{(\mathbf{P}_{i-1})} \left[\mathbf{Q}_{\mathbf{PP}}^{-1} \mathbf{P}^* + \mathbf{H}_{\mathbf{ZP}}^T \mathbf{Q}_{\mathbf{ZZ}}^{-1} \left[\mathbf{Z} - \mathcal{G}(\mathbf{P}_{i-1}) + \mathbf{H_{ZP}} \mathbf{P}_{i-1}\right]\right] \tag{3.13}$$

If the parameters have zero mean, $\mathbf{P}^* = 0$, this simplifies to

$$\mathbf{P}_i = \mathbf{Q}_{\mathbf{PP}}^{(\mathbf{P}_{i-1})} \mathbf{H}_{\mathbf{ZP}}^T \mathbf{Q}_{\mathbf{ZZ}}^{-1} \left[\mathbf{Z} - \mathcal{G}(\mathbf{P}_{i-1}) + \mathbf{H_{ZP}} \mathbf{P}_{i-1}\right]. \tag{3.14}$$

The application of the Sherman-Morrison-Woodbury formula

$$\left[\mathbf{A}^{-1} + \mathbf{B}^T \mathbf{C}^{-1} \mathbf{B}\right]^{-1} \mathbf{B}^T \mathbf{C}^{-1} = \mathbf{A}\mathbf{B}^T \left[\mathbf{C} + \mathbf{B}\mathbf{A}\mathbf{B}^T\right]^{-1}, \tag{3.15}$$

which is a special case of the formulas from [39] for symmetric invertible matrices $\mathbf{A}$ and $\mathbf{B}$, yields

$$\mathbf{P}_i \coloneqq \mathbf{Q_{PP}} \mathbf{H}_{\mathbf{ZP}}^T \left[\mathbf{Q_{ZZ}} + \mathbf{H_{ZP}} \mathbf{Q_{PP}} \mathbf{H}_{\mathbf{ZP}}^T\right]^{-1} \left[\mathbf{Z} - \mathcal{G}(\mathbf{P}_{i-1}) + \mathbf{H_{ZP}} \mathbf{P}_{i-1}\right]. \tag{3.16}$$

Algorithm 15: Modified Gauss-Newton (Cokriging Equations) ($\mathbf{GN}_{\mathrm{CE}}$)

Input: initial value $\mathbf{P}_0$, stopping criterion
Output: estimate of MAP point $\mathbf{P}_{\mathrm{map}}$
$i \coloneqq 0$ [set index];
repeat
  $i \to i+1$ [shift index];
  calculate $\mathbf{H_{ZP}}$ [apply adjoint state method $\prod_{j=1}^{n_\mathbf{Z}} n_{\mathbf{z}_j}$ times];
  $\mathbf{B}_i \coloneqq \mathbf{Q_{PP}}\mathbf{H}_{\mathbf{ZP}}^T \left[\mathbf{Q_{ZZ}} + \mathbf{H_{ZP}}\mathbf{Q_{PP}}\mathbf{H}_{\mathbf{ZP}}^T\right]^{-1}$ [calculate matrix for step direction];
  $\mathbf{D}_i \coloneqq -\mathbf{P}_{i-1} + \mathbf{B}_i\left[\mathbf{Z} - \mathcal{G}(\mathbf{P}_{i-1}) + \mathbf{H_{ZP}}\mathbf{P}_{i-1}\right]$ [compute direction];
  $\alpha_i \coloneqq \mathrm{linesearch}(\mathbf{P}_{i-1}, \mathbf{D}_i)$ [compute step width];
  $\mathbf{P}_i \coloneqq \mathbf{P}_{i-1} + \alpha_i \mathbf{D}_i$ [define i-th iteration];
until *converged*;
$\mathbf{P}_{\mathrm{map}} \coloneqq \mathbf{P}_i$ [accept final iteration];
return $\mathbf{P}_{map}$;

This version of the Gauss-Newton scheme is also known as the Cokriging Equations [61], and its mayor advantage is the drastically reduced size of the matrix that has to be inverted in each step of the method. A comparison with equation (2.139) shows that this transformation is equivalent to switching from an estimate of $\mathbf{Q}_{\mathbf{PP}}^{\mathrm{post}}$ to a formulation based on $\mathbf{Q}_{\mathbf{ZZ}}^{\mathrm{post}}$, the posterior covariance matrix of the measurements. Reformulated as an iterative scheme with line search, this transformation of the underlying equation leads to algorithm 15 ($\mathbf{GN}_{\mathrm{CE}}$).

3.3.2 Randomized Gauss-Newton and Levenberg-Marquardt

The randomized algorithms for uncertainty quantification presented in section 2.5 may also be applied to the matrix $\mathbf{Q}_{\mathbf{PP}}^{(\mathbf{P}_{i-1})}$ that appears in the Gauss-Newton method, algorithm 14 (**GN**), as done in section 2.7. This is equivalent to reformulating the objective function in terms of $\widetilde{\mathbf{P}} \coloneqq \mathbf{Q}_{\mathbf{PP}}^{-1/2}\mathbf{P}$ and $\widetilde{\mathbf{P}}^* \coloneqq \mathbf{Q}_{\mathbf{PP}}^{-1/2}\mathbf{P}^*$ and deriving the Gauss-Newton method for this new objective function. The linearization around $\widetilde{\mathbf{P}}$ reads

$$\widetilde{L}(\widetilde{\mathbf{P}} + \delta_{\widetilde{\mathbf{P}}}) \approx \frac{1}{2}\left\|\widetilde{\mathbf{P}} + \delta_{\widetilde{\mathbf{P}}} - \widetilde{\mathbf{P}}^*\right\|_{\mathbf{I}}^2 + \frac{1}{2}\left\|\mathbf{Z} - \left[\mathcal{G}(\mathbf{Q}_{\mathbf{PP}}^{1/2}\widetilde{\mathbf{P}}) + \mathbf{H_{ZP}}\mathbf{Q}_{\mathbf{PP}}^{1/2}\delta_{\widetilde{\mathbf{P}}}\right]\right\|_{\mathbf{Q}_{\mathbf{ZZ}}^{-1}}^2, \quad (3.17)$$

and the gradient with respect to the change in parameters becomes

$$\begin{aligned}\nabla_{\delta_{\widetilde{\mathbf{P}}}}\widetilde{L} &\approx \widetilde{\mathbf{P}} + \delta_{\widetilde{\mathbf{P}}} - \widetilde{\mathbf{P}}^* - \mathbf{Q}_{\mathbf{PP}}^{1/2}\mathbf{H}_{\mathbf{ZP}}^T\mathbf{Q}_{\mathbf{ZZ}}^{-1}\left[\mathbf{Z} - \left[\mathcal{G}(\mathbf{Q}_{\mathbf{PP}}^{1/2}\widetilde{\mathbf{P}}) + \mathbf{H_{ZP}}\mathbf{Q}_{\mathbf{PP}}^{1/2}\delta_{\widetilde{\mathbf{P}}}\right]\right] \quad (3.18)\\ &= \left[\mathbf{I} + \mathbf{Q}_{\mathbf{PP}}^{1/2}\mathbf{H}_{\mathbf{ZP}}^T\mathbf{Q}_{\mathbf{ZZ}}^{-1}\mathbf{H_{ZP}}\mathbf{Q}_{\mathbf{PP}}^{1/2}\right]\delta_{\widetilde{\mathbf{P}}} + \widetilde{\mathbf{P}} - \widetilde{\mathbf{P}}^*\\ &\qquad - \mathbf{Q}_{\mathbf{PP}}^{1/2}\mathbf{H}_{\mathbf{ZP}}^T\mathbf{Q}_{\mathbf{ZZ}}^{-1}\left[\mathbf{Z} - \mathcal{G}(\mathbf{Q}_{\mathbf{PP}}^{1/2}\widetilde{\mathbf{P}})\right].\end{aligned}$$

Algorithm 16: Randomized Gauss-Newton ($\mathbf{GN}_r$)

Input: initial value $\mathbf{P}_0$, stopping criterion
Output: estimate of MAP point $\mathbf{P}_{\text{map}}$
$i \coloneqq 0$ [set index];
repeat
  $i \to i+1$ [shift index];
  decompose $\mathbf{Q}_{\mathbf{PP}}^{(\mathbf{P}_{i-1})} = \mathbf{Q}_{\mathbf{PP}}^{1/2}\left[\mathbf{I} - \mathbf{V}_r\boldsymbol{\Upsilon}_r\mathbf{V}_r^T\right]\mathbf{Q}_{\mathbf{PP}}^{1/2}$, e.g. using algorithm 8;
  $\mathbf{R}_i \coloneqq -\nabla L|_{\mathbf{P}_{i-1}}$ [calculate residual];
  $\mathbf{D}_i \coloneqq \mathbf{Q}_{\mathbf{PP}}^{1/2}\left[\mathbf{I} - \mathbf{V}_r\boldsymbol{\Upsilon}_r\mathbf{V}_r^T\right]\mathbf{Q}_{\mathbf{PP}}^{1/2}\mathbf{R}_i$ [compute step direction];
  $\alpha_i \coloneqq \text{linesearch}(\mathbf{P}_{i-1}, \mathbf{D}_i)$ [compute step width];
  $\mathbf{P}_i \coloneqq \mathbf{P}_{i-1} + \alpha_i\mathbf{D}_i$ [define i-th iteration];
until *converged*;
$\mathbf{P}_{\text{map}} \coloneqq \mathbf{P}_i$ [accept final iteration];
return $\mathbf{P}_{map}$;

The matrix that appears is the same as in section 2.5, and consequently the new step direction is

$$\begin{aligned}\delta_{\widetilde{\mathbf{P}}} &= -\left[\mathbf{I} + \mathbf{M}_{\text{like}}\right]^{-1}\left[\widetilde{\mathbf{P}} - \widetilde{\mathbf{P}}^* - \mathbf{Q}_{\mathbf{PP}}^{1/2}\mathbf{H}_{\mathbf{ZP}}^T\mathbf{Q}_{\mathbf{ZZ}}^{-1}\left[\mathbf{Z} - \mathcal{G}(\mathbf{Q}_{\mathbf{PP}}^{1/2}\widetilde{\mathbf{P}})\right]\right] \\ &\approx -\left[\mathbf{I} - \mathbf{V}_r\boldsymbol{\Upsilon}_r\mathbf{V}_r^T\right]\left[\widetilde{\mathbf{P}} - \widetilde{\mathbf{P}}^* - \mathbf{Q}_{\mathbf{PP}}^{1/2}\mathbf{H}_{\mathbf{ZP}}^T\mathbf{Q}_{\mathbf{ZZ}}^{-1}\left[\mathbf{Z} - \mathcal{G}(\mathbf{Q}_{\mathbf{PP}}^{1/2}\widetilde{\mathbf{P}})\right]\right],\end{aligned} \tag{3.19}$$

where

$$\mathbf{M}_{\text{like}} \approx \mathbf{V}_r\boldsymbol{\Upsilon}_r\mathbf{V}_r^T \tag{3.20}$$

is again an approximate spectral decomposition, e.g. computed with algorithm 7 ($\mathbf{ED}_r$) or algorithm 8 ($\mathbf{SVD}_r$). A reformulation of $\delta_{\widetilde{\mathbf{P}}}$ in terms of $\mathbf{P}$ then yields

$$\delta_{\mathbf{P}} = -\mathbf{Q}_{\mathbf{PP}}^{1/2}\left[\mathbf{I} - \mathbf{V}_r\boldsymbol{\Upsilon}_r\mathbf{V}_r^T\right]\mathbf{Q}_{\mathbf{PP}}^{1/2}\left[\mathbf{Q}_{\mathbf{PP}}^{-1}\left[\mathbf{P} - \mathbf{P}^*\right] - \mathbf{H}_{\mathbf{ZP}}^T\mathbf{Q}_{\mathbf{ZZ}}^{-1}\left[\mathbf{Z} - \mathcal{G}(\mathbf{P})\right]\right], \tag{3.21}$$

and therefore the iterative procedure

$$\begin{aligned}\mathbf{P}_i \coloneqq \mathbf{P}_{i-1} &- \mathbf{Q}_{\mathbf{PP}}^{1/2}\left[\mathbf{I} - \mathbf{V}_r\boldsymbol{\Upsilon}_r\mathbf{V}_r^T\right]\mathbf{Q}_{\mathbf{PP}}^{1/2} \\ &\cdot\left[\mathbf{Q}_{\mathbf{PP}}^{-1}\left[\mathbf{P}_{i-1} - \mathbf{P}^*\right] - \mathbf{H}_{\mathbf{ZP}}^T\mathbf{Q}_{\mathbf{ZZ}}^{-1}\left[\mathbf{Z} - \mathcal{G}(\mathbf{P}_{i-1})\right]\right],\end{aligned} \tag{3.22}$$

as detailed in algorithm 16. This scheme is used in the synthetic test cases of section 6.1.1.

We may also introduce a scaling parameter η to interpolate between the preconditioned PCG method of section 2.3 and the randomized Gauss-Newton method. This leads to

$$\begin{aligned}\mathbf{P}_i \coloneqq \mathbf{P}_{i-1} &- \mathbf{Q}_{\mathbf{PP}}^{1/2}\left[\mathbf{I} - \eta\mathbf{V}_r\boldsymbol{\Upsilon}_r\mathbf{V}_r^T\right]\mathbf{Q}_{\mathbf{PP}}^{1/2} \\ &\cdot\left[\mathbf{Q}_{\mathbf{PP}}^{-1}\left[\mathbf{P}_{i-1} - \mathbf{P}^*\right] - \mathbf{H}_{\mathbf{ZP}}^T\mathbf{Q}_{\mathbf{ZZ}}^{-1}\left[\mathbf{Z} - \mathcal{G}(\mathbf{P}_{i-1})\right]\right],\end{aligned} \tag{3.23}$$

which can be interpreted as a randomized variant of the modified Levenberg-Marquardt scheme presented by *Nowak and Cirpka* [62], with the randomized Gauss-Newton method used instead of the Cokriging Equations.

Both the randomized Gauss-Newton method and the randomized Levenberg-Marquardt method may be reformulated to avoid multiplications with $\mathbf{Q}_{\mathbf{PP}}^{-1}$ and $\mathbf{Q}_{\mathbf{PP}}^{-1/2}$, in analogy to the caching PCG schemes of sections 2.3.5 and 2.7. This leads to a more efficient variant of the algorithms. Since the cost per iteration is typically dominated by the matrix decomposition, this cached variant is not as essential as in the case of the PCG method and is left out for the sake of brevity.

4 Governing Equations

While the formulation of the inverse problem in chapter 1 and the description and derivation of the numerical methods in chapter 2 can be carried out in a rather abstract fashion and in broad generality, the actual application of the method in test cases and real-world scenarios requires the specification of the forward model $\mathcal{F}$ describing the physical processes that are considered. In the following we describe several standard models of subsurface hydrology, namely the groundwater flow equation, the Richards equation and the convection-diffusion equation, closely following the presentation given by *Roth* [68]. We also derive adjoint models that may be used in the context of calculating the gradient of the objective function as described in section 2.4.

Note that the following models are formulated as general as possible with regard to the parameter fields. Some of the parameter fields may be assumed known for various reasons, e.g. because the model is simplified by setting one of the parameters to a constant value, or because the system can be brought into a well-defined initial state. In such a situation and the ones of chapter 6, the affected model has to be understood as the restriction of the more general model to the reduced parameter space. Also note that the straightforward extension of the presented methods for the inclusion of boundary values in the parameters requires the introduction of different types of parameter fields, those for the discretization of the domain and those for the boundary, and therefore has been left out for the sake of brevity.

4.1 Groundwater Flow Equation

The flow of water in soil and unconsolidated rock is governed by the interaction of gravity and capillary forces. Due to the complex topological arrangement of the pores and their small volume, the pore space and the water flow within can't be modeled in all details [68]. Therefore the models that are employed tend to be formulated in terms of spatially averaged quantities and effective parameters combined with material properties that are derived from theoretical considerations or scientific experiments. In this representation of the system, gravity and capillary forces define potential fields, the gravitational potential ψ_g and the matric potential ψ_m, and their sum, called the soil water potential

$$\psi_w \coloneqq \psi_g + \psi_m, \tag{4.1}$$

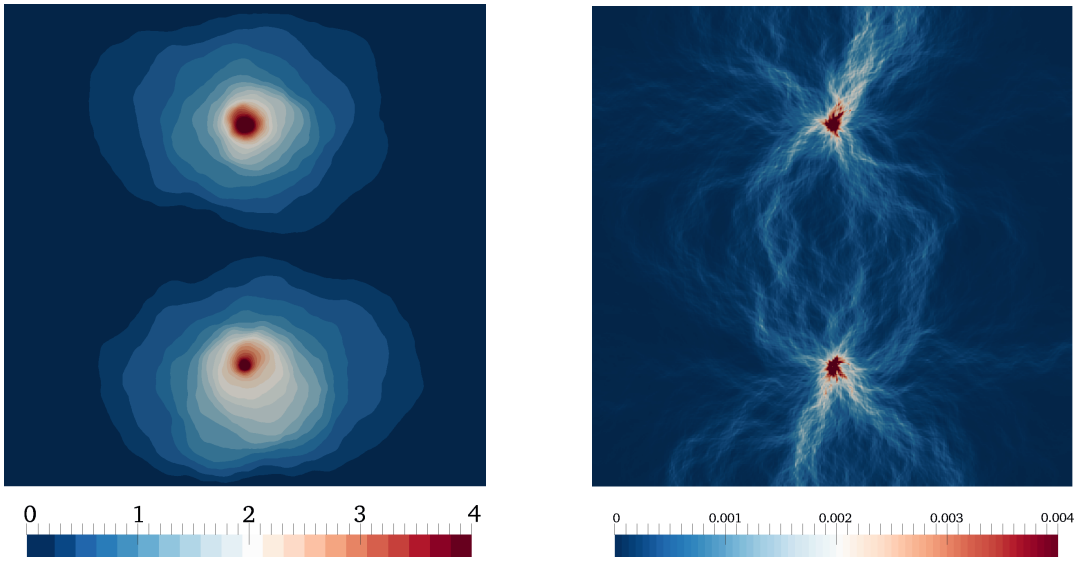

Figure 4.1: *Left:* Absolute value of the potential ϕ for a setup with an injection well in the upper half of the domain and an extraction well in the lower half of the domain. Values are larger than average in the vicinity of the injection well and lower than average near the extraction well, the absolute value is chosen to increase the range of representable values. *Right:* Flow patterns resulting from the underlying conductivity field and the distribution of the potential. Color represents norm of water flux, values inside the wells aren't represented by the scale.

governs the spatially averaged flow. Often the hydraulic head

$$\phi := [\rho g]^{-1} \psi_w \tag{4.2}$$

is used instead of ψ_w in the formulations, since it has a direct interpretation as the height of a water column. This only changes the units used but does not influence the structure of the PDEs in any other way.

The law of conservation of mass is represented by the conservation of the volumetric water content θ_w, i.e. the volume fraction that contains water, based on the fact that water has very low compressibility. Assuming the spatially averaged water content to be sufficiently smooth and applying the continuity equation yields

$$\partial_t \theta_w(\phi) + \nabla \cdot j_{\theta_w}(\phi) - q_{\theta_w} = 0, \tag{4.3}$$

where j_{θ_w} is the water content flux and q_{θ_w} is a source term that describes water being added to or removed from the system, e.g. through injection and extraction wells. See figure 4.1 for an example of the relation between the potential ϕ and the resulting flow patterns.

Remark 20 *Equation* (4.3) *includes the assumption that the water content* θ_w *is a unique function of the hydraulic head* ϕ*, and therefore of the soil water potential* ψ_w*,*

called the soil water characteristic or pressure-saturation relation. In reality, the water content exhibits hysteretic behavior, with the amount of hysteresis depending, among other things, on the pore size distribution. The water content is therefore not just a function of the current water potential, but also of its history and the path leading to its current state. In the case of the groundwater flow equation this is not of concern, since the effects of hysteresis are negligible under saturated conditions, but it becomes relevant when considering the Richards equation.

The two functions $\theta_w(\phi)$ and $j_{\theta_w}(\phi)$ in equation (4.3) relating the hydraulic head ϕ with the water content θ_w and its flux are material properties that depend on the given hydrological system. In the case of groundwater the system is under saturated conditions, i.e. all available space is filled with water and the water content may be identified with the porosity θ of the medium. In this situation an increase in water content can only occur through compression of the soil matrix, and the change in water content is assumed to be proportional to the change in potential, i.e. $\delta_{\theta_w} = S_s \delta_\phi$, where the proportionality constant S_s is known as a storage term. The temporal derivative in equation (4.3) may therefore be replaced using

$$\partial_t \theta_w(\phi) = S_s \partial_t \phi = \exp(Z_s) \partial_t \phi, \tag{4.4}$$

with $Z_s = \ln(S_s)$ being an unknown scalar parameter field. The relation between ϕ and j_{θ_w} is known as Darcy's Law [20], an empirical flux law that was formulated based on scientific experiments with sand filters. It states that the macroscopic flux is driven by the gradient of the potential,

$$j_{\theta_w} = -K \nabla \phi, \tag{4.5}$$

where K is a symmetric second rank tensor, called the conductivity of the porous medium. In addition to validation in experiments Darcy's Law may also be derived from formulations of the flow dynamics on the pore scale [80]. The law is only valid for stationary flow, but is in extension also used under transient flow conditions if the external forcing is slow enough. While the conductivity K is a tensor, it is almost always assumed to be isotropic and replaced by a scalar in practice [68]. The hydraulic conductivity is highly heterogeneous and can vary by several orders of magnitude in the same material, therefore formulations typically use the log-conductivity $Y = \ln(K)$ instead. This leads to Y being a second unknown scalar parameter field.

Combining the continuity equation (4.3) with the two constitutive equations (4.4) and (4.5) results in the groundwater flow equation:

Model 1 (Groundwater Flow Equation, Classical Formulation)
Given a domain Ω, a time interval $T = [0, t_{max}]$ and parameter fields $S \coloneqq (Z_s, Y, \phi_0)$ on Ω, the groundwater flow equation in classical formulation is the transient PDE

$$\mathcal{F}_\phi(S; \phi) \coloneqq \exp(Z_s) \partial_t \phi + \nabla \cdot j_{\theta_w}(Y, \phi) - q_{\theta_w} = 0 \tag{4.6}$$

on $\Omega \times T$ *with the flux*

$$j_{\theta_w}(Y, \phi) \coloneqq -\exp(Y)\nabla\phi \tag{4.7}$$

and initial and boundary conditions

$$\phi = \phi_0 \text{ for } t = 0, \qquad \phi = b_\phi^D \text{ on } \Gamma_\phi^D, \qquad j_{\theta_w} \cdot \mathbf{n} = b_\phi^N \text{ on } \Gamma_\phi^N. \tag{4.8}$$

Here Γ_ϕ^D *designates the Dirichlet part of the boundary where* ϕ *is prescribed, and* $\Gamma_\phi^N = \partial\Omega \setminus \Gamma_\phi^D$ *the Neumann part of the boundary where the normal component of the flux* j_{θ_w} *is prescribed. The notation* $\mathcal{F}_\phi(S; \phi)$ *is used for both the PDE in equation* (4.6) *and the full model, with the implication that suitable boundary and initial conditions as above are defined in the given context.*

As was already mentioned in remark 1, the groundwater flow equation as defined above is neither suitable for most of the real-world scenarios nor for the piecewise constant parameterization, since a solution ϕ of equation (4.6) would need to be twice continuously differentiable in Ω. Instead, a weak formulation of the groundwater flow equation has to be used.

The weak formulation uses the Sobolev spaces $H^1(\Omega)$ and $H^{1/2}(\Gamma_\phi^D)$. Due to the trace theorem [12] every $u \in H^1(\Omega)$ has a restriction $\varrho(u) \in H^{1/2}(\Gamma_\phi^D)$ to the Dirichlet boundary part and every $v \in H^{1/2}(\Gamma_\phi^D)$ has an extension $\varepsilon(v) \in H^1(\Omega)$ to the whole domain Ω. We use these mappings to define the space

$$H^1_{\Gamma_\phi^D}(\Omega) \coloneqq \left\{ u \in H^1(\Omega) \,\middle|\, \varrho(u) = 0 \in H^{1/2}\left(\Gamma_\phi^D\right) \right\}, \tag{4.9}$$

the subspace of functions with homogeneous Dirichlet boundary conditions, and in turn the Bochner space

$$L^2\left(T; H^1_{\Gamma_\phi^D}(\Omega)\right) \coloneqq \left\{ u \in L^2(\Omega \times T) \,\middle|\, \forall\, t \in T\colon u(t) \in H^1_{\Gamma_\phi^D}(\Omega) \right\}, \tag{4.10}$$

the subspace of $L^2(\Omega \times T)$ of functions with weak spatial derivatives and homogeneous Dirichlet boundary conditions [26]. For the Dirichlet and Neumann boundary parts we define the corresponding spaces

$$L^2\left(T; H^{1/2}\left(\Gamma_\phi^D\right)\right) \coloneqq \left\{ u \in L^2\left(\Gamma_\phi^D \times T\right) \,\middle|\, \forall\, t \in T\colon u(t) \in H^{1/2}\left(\Gamma_\phi^D\right) \right\} \tag{4.11}$$

and

$$L^2\left(T; H^{1/2}\left(\Gamma_\phi^N\right)\right) \coloneqq \left\{ u \in L^2\left(\Gamma_\phi^N \times T\right) \,\middle|\, \forall\, t \in T\colon u(t) \in H^{1/2}\left(\Gamma_\phi^N\right) \right\}, \tag{4.12}$$

assume the Dirichlet boundary values b_ϕ^D are in $L^2(T; H^{1/2}(\Gamma_\phi^D))$, assume the Neumann boundary values b_ϕ^N are in $L^2(T; H^{1/2}(\Gamma_\phi^N))$, and denote with $\varepsilon(b_\phi^D)$ an extension of the Dirichlet boundary values onto $L^2(T; H^1_{\Gamma_\phi^D}(\Omega))$.

Let the expression $\langle \cdot, \cdot \rangle$ denote the scalar product of $L^2(\Omega \times T)$ and $\langle \cdot, \cdot \rangle_V$ that of $L^2(V)$ for all other spaces V mentioned above, i.e. $\Gamma_\phi^D \times T$, $\Gamma_\phi^N \times T$ and Ω. The strong variational formulation of equation (4.6) is

$$\forall\, \psi_\phi \in L^2(\Omega \times T) : \langle \psi_\phi, \mathcal{F}_\phi(S;\phi) \rangle = 0 \tag{4.13}$$

with

$$\begin{aligned} \forall\, \psi_\phi \in L^2(\Omega) &: \langle \psi_\phi, \phi(0) - \phi_0 \rangle_\Omega = 0, \\ \forall\, \psi_\phi \in L^2\left(\Gamma_\phi^D \times T\right) &: \left\langle \psi_\phi, \varrho(\phi) - b_\phi^D \right\rangle_{\Gamma_\phi^D \times T} = 0, \\ \forall\, \psi_\phi \in L^2\left(\Gamma_\phi^N \times T\right) &: \left\langle \psi_\phi, j_{\theta_w} \cdot \mathbf{n} - b_\phi^N \right\rangle_{\Gamma_\phi^N \times T} = 0, \end{aligned} \tag{4.14}$$

which requires $\phi \in H^2(\Omega)$ for all times $t \in T$ and has the same mathematical structure as the Lagrange multipliers that appear in the Lagrangian, equation (2.68). Using integration by parts and shifting the spatial derivative over to ψ_ϕ results in an equivalent but more general equation:

$$\begin{aligned} \forall\, \psi_\phi \in L^2\left(T; H^1_{\Gamma_\phi^D}(\Omega)\right) : \langle \psi_\phi, \exp(Z_s)\partial_t \phi \rangle - \left\langle \nabla \psi_\phi, j_{\theta_w}(Y,\phi) \right\rangle \\ - \langle \psi_\phi, q_{\theta_w} \rangle + \left\langle \psi_\phi, j_{\theta_w} \cdot \mathbf{n} \right\rangle_{\Gamma \times T} = 0, \end{aligned} \tag{4.15}$$

with $\Gamma \coloneqq \partial\Omega$ being the boundary of Ω and j_{θ_w} defined as in equation (4.7). Since ψ_ϕ is zero on Γ_ϕ^D and $j_{\theta_w} \cdot \mathbf{n} = b_\phi^N$ holds weakly on Γ_ϕ^N, this leads to the following weak formulation:

Model 2 (Groundwater Flow Equation, Weak Formulation)
Let Ω be a given domain, $T \coloneqq [0, t_{max}]$ a time interval, $S \coloneqq (Z_s, Y, \phi_0)$ a tuple of parameter fields on Ω and $(\phi, \partial_t \phi)$ a pair of states for which

$$\phi \in \varepsilon(b_\phi^D) + L^2\left(T; H^1_{\Gamma_\phi^D}(\Omega)\right), \quad \partial_t \phi \in L^2\left(T; H^{-1}(\Omega)\right), \tag{4.16}$$

with $\partial_t \phi$ the weak temporal derivative of ϕ. The pair $(\phi, \partial_t \phi)$ is the weak solution of the groundwater flow equation if the condition

$$\forall\, \psi_\phi \in L^2(\Omega) : \langle \psi_\phi, \phi(0) - \phi_0 \rangle_\Omega = 0 \tag{4.17}$$

holds for ϕ, and both functions together solve the equation

$$\forall\, \psi_\phi \in L^2\left(T; H^1_{\Gamma_\phi^D}(\Omega)\right) : \langle \psi_\phi, \exp(Z_s)\partial_t \phi \rangle + a_\phi(\phi, \psi_\phi) + b_\phi(\psi_\phi) = 0 \tag{4.18}$$

with the bilinear form

$$a_\phi(\phi, \psi_\phi) \coloneqq -\left\langle \nabla \psi_\phi, j_{\theta_w}(Y,\phi) \right\rangle \tag{4.19}$$

and the linear form

$$b_\phi(\psi_\phi) \coloneqq -\langle \psi_\phi, q_{\theta_w} \rangle + \left\langle \psi_\phi, b_\phi^N \right\rangle_{\Gamma_\phi^N \times T}. \tag{4.20}$$

This equation, together with consistent initial and boundary conditions, gives a second definition of ϕ as a function of S. We again denote it with $\mathcal{F}_\phi(S;\phi)$, since it is clear from context which of the definitions is meant.

We may also define the stationary limit of the above equation. If the head ϕ does not change with time, both the log-storage term Z_s and the initial state ϕ_0 vanish from the formulation, and the resulting model only depends on the log-conductivity Y:

Model 3 (Stationary Groundwater Flow Equation)
Let Ω be a given domain, Y a parameter field on Ω and

$$\phi \in \varepsilon(b_\phi^D) + H^1_{\Gamma_\phi^D}(\Omega)\,. \tag{4.21}$$

The function ϕ is the weak solution of the stationary groundwater flow equation if it solves the equation

$$\forall\, \psi_\phi \in H^1_{\Gamma_\phi^D}(\Omega) : a_\phi(\phi, \psi_\phi) + b_\phi(\psi_\phi) = 0, \tag{4.22}$$

where $a_\phi(\cdot,\cdot)$ and $b_\phi(\cdot)$ result from equations (4.19) and (4.20) by replacing the scalar product of $L^2(\Omega \times T)$ with that of $L^2(\Omega)$. As above, we denote this mapping from S to ϕ with $\mathcal{F}_\phi(S;\phi)$.

4.2 Richards Equation

The Richards equation [66] extends the groundwater flow equation from the previous section to regimes that are not fully saturated. Its central assumption is a connected air phase, since this allows for the multiphase regime to be reduced to the dynamics of a single phase, and consequently it is only to a certain extent applicable in the capillary fringe, the transitional area that separates the groundwater from the vadose zone. The Richards equation applies Darcy's Law (4.5) to the water phase of the partially saturated porous medium, i.e.

$$\partial_t \theta_w(\phi_\mathrm{m}) + \nabla \cdot j_{\theta_w}(\phi_\mathrm{m}) - q_{\theta_w} = 0, \tag{4.23}$$

with the flux

$$j_{\theta_w}(\phi_\mathrm{m}) = -K\kappa(\phi_\mathrm{m})\left[\nabla\phi_\mathrm{m} - e_g\right], \tag{4.24}$$

where e_g is the unit vector in the direction of gravity.

In contrast to the groundwater flow equation, the water content θ_w varies across a wide range and can't be linearized in the temporal derivative in Darcy's Law. Additionally, the conductivity is highly dependent on the water content and consists of the parameter field $K = \exp(Y)$ for saturated conditions together with a factor $\kappa(\phi_\mathrm{m})$ that reflects the reduction in conductivity under partially saturated conditions. The relative conductivity κ is typically expressed as a function of the saturation

$$\Theta \coloneqq \frac{\theta_w(\phi_\mathrm{m}) - \theta_r}{\theta_s - \theta_r}, \tag{4.25}$$

where θ_r is the residual water content after complete drainage of the medium and θ_s is the water content under fully saturated conditions, bounded from above by the porosity θ of the medium. The full specification of the flux (4.24) therefore requires two functions $\kappa(\Theta)$ and $\Theta(\phi_\mathrm{m})$ that define material properties.

The relation $\Theta(\phi_\mathrm{m})$ is the strongly hysteretic soil water characteristic that was already mentioned in remark 20. It therefore can't be expected to be a unique function, which implies that parameterizations of $\Theta(\phi_\mathrm{m})$ are only valid on a single hysteresis branch and have to be subject to local switching conditions in areas where the flow dynamics violate this assumption. The two most popular parameterizations are the one by *Brooks and Corey* [14] and the one by *van Genuchten* [79]. In numerical codes the van Genuchten parameterization is often preferred, since the resulting functions are differentiable everywhere. This parameterization has the form

$$\Theta(\phi_\mathrm{m}) = \left[1 + \left[\alpha \left|\phi_\mathrm{m}\right|\right]^n\right]^{-m}, \tag{4.26}$$

which is often used in the simpler form

$$\Theta(\phi_\mathrm{m}) = \left[1 + \left[\alpha \left|\phi_\mathrm{m}\right|\right]^n\right]^{\frac{1-n}{n}} \tag{4.27}$$

by setting $m \coloneqq 1 - \frac{1}{n}$. Here α and n are two parameters that characterize the porous medium.

The parameterizations of the relation $\kappa(\Theta)$ are often of the form

$$\kappa(\Theta) = \Theta^a I(\Theta)^b, \tag{4.28}$$

where $I(\Theta)$ is an antiderivative of some function in ϕ_m that is renormalized, i.e. $I(1) = 1$, and a is an exponent that accounts for the tortuosity of the porous medium, which is a measure for the increase in average travel length along flow paths due to curvature. A popular parameterization is the one by *Mualem* [58], which is a modification of the equation above and has the form

$$\kappa(\Theta) = \Theta^a \left[\frac{\int_0^\Theta \phi_\mathrm{m}^{-1}}{\int_0^1 \phi_\mathrm{m}^{-1}}\right]^2, \tag{4.29}$$

where ϕ_m as a function of the saturation has to be calculated using the inverse of one of the parameterizations $\Theta(\phi_\mathrm{m})$ and a is a free parameter. Other parameterizations of the above type are the one by *Burdine* [16] and the one by *Mualem and Dagan* [59].

Combining the van Genuchten parameterization of $\Theta(\phi_\mathrm{m})$ and the Mualem parameterization of $\kappa(\Theta)$ results in the Mualem-van Genuchten parameterization

$$\begin{aligned}
\kappa(\Theta) &= \Theta^a \left[1 - \left[1 - \Theta^{\frac{n}{n-1}}\right]^{\frac{1-n}{n}}\right]^2 \\
\kappa(\phi_\mathrm{m}) &= \left[1 + \left[\alpha \left|\phi_\mathrm{m}\right|\right]^n\right]^{a \cdot \frac{1-n}{n}} \cdot \left[1 - \left[\alpha \left|\phi_\mathrm{m}\right|\right]^{n-1} \left[1 + \left[\alpha \left|\phi_\mathrm{m}\right|\right]^n\right]^{\frac{1-n}{n}}\right]^2.
\end{aligned} \tag{4.30}$$

For the water content the parameterization results from equation (4.25) and is

$$\begin{aligned} \theta_w(\Theta) &= \theta_r + \Theta \cdot [\theta_s - \theta_r] \\ \theta_w(\phi_\mathrm{m}) &= \theta_r + \left[1 + [\alpha\,|\phi_\mathrm{m}|]^n\right]^{\frac{1-n}{n}} \cdot [\theta_s - \theta_r]\,. \end{aligned} \tag{4.31}$$

Choosing constant values for these parameters leads to homogeneous media. If we instead assume that the soil is a Miller-similar porous medium, which means that the pore-space geometry is the same in each location but the average pore size may vary in space [57], then reference values $\widehat{Y}$ and $\widehat{\alpha}$ exist with

$$K \cdot [\exp(\chi)]^2 = \widehat{K}, \quad \alpha \cdot \exp(\chi) = \widehat{\alpha}, \tag{4.32}$$

where χ is the logarithm of the Miller similarity scale parameter [67]. The other parameters $n = \widehat{n}$ and $a = \widehat{a}$ are assumed to be scale invariant and therefore constant in Ω. Miller similarity can be included in the model by replacing equations (4.30) and (4.31) with

$$\begin{aligned} \kappa(\phi_\mathrm{m}) = [\exp(\chi)]^{-2} &\left[1 + \left[[\exp(\chi)]^{-1}\,\widehat{\alpha}\,|\phi_\mathrm{m}|\right]^{\widehat{n}}\right]^{a\cdot\frac{1-\widehat{n}}{\widehat{n}}} \\ &\cdot \left[1 - \left[[\exp(\chi)]^{-1}\,\widehat{\alpha}\,|\phi_\mathrm{m}|\right]^{\widehat{n}-1} \left[1 + \left[[\exp(\chi)]^{-1}\,\widehat{\alpha}\,|\phi_\mathrm{m}|\right]^{\widehat{n}}\right]^{\frac{1-\widehat{n}}{\widehat{n}}}\right]^2 \end{aligned} \tag{4.33}$$

and

$$\theta_w(\phi_\mathrm{m}) = \theta_r + \left[1 + \left[[\exp(\chi)]^{-1}\,\widehat{\alpha}\,|\phi_\mathrm{m}|\right]^{n}\right]^{\frac{1-n}{n}} \cdot [\theta_s - \theta_r]\,. \tag{4.34}$$

Under this assumption the spatial variability of the average pore diameter is the sole cause for heterogeneity of the effective hydraulic parameters, and Y and α are perfectly correlated.

For simplicity we assume that a fixed tuple (α, n, a) for each location $\mathbf{x} \in \Omega$ is enough to parameterize the unsaturated flow, e.g. because the simulation is restricted to pure infiltration or pure drainage of the pore space. This leaves three possible choices for the parameterization of the domain:

- Assume that the medium is Miller-similar and equation (4.32) holds. Then χ is the only relevant parameter field and all other parameters are constant in Ω, which can be modeled by setting the covariance matrices of their spatial parts to zero and dropping the corresponding contributions from the objective function.
- Assume that the medium doesn't exhibit Miller similarity, and all parameters have to be treated individually. Then each of them is a spatially distributed parameter field in the sense of chapter 1, which raises the question of correlation of the different parameter fields and how it is incorporated in the inversion.

- Assume that the medium is almost Miller-similar but allow for imperfections, i.e. treat the reference values themselves as spatially heterogeneous. If the variance of these reference value distributions is small, then the effective parameters will be highly correlated. Note that it isn't necessary to include cross-covariance information in the sense of remark 6, since the deviations from Miller similarity can be assumed to be independent.

From these options we choose the third one, and we refer to the reference parameters using the names of the local parameters to simplify notation. The reference parameters may then be treated as parameter fields as we have introduced them, and we denote the two constitutive functions with $\theta_w(\alpha, n, \chi, \phi_\mathrm{m})$ and $\kappa(\alpha, n, a, \chi, \phi_\mathrm{m})$ to reflect this dependency. Inserting the Mualem-van Genuchten parameterization with Miller scaling, equation (4.33) and (4.34), into the flux definition (4.24) leads to the following form of the Richards equation:

Model 4 (Richards Equation, Classical Formulation)
Given a domain Ω, a time interval $T = [0, t_{max}]$ and the tuple of parameter fields

$$S \coloneqq (Y, \alpha, n, a, \chi, \phi_{m,0}) \tag{4.35}$$

on Ω, the Richards equation in classical formulation is the transient PDE

$$\mathcal{F}_{\phi_m}(S; \phi_m) \coloneqq \partial_t \theta_w(\alpha, n, \chi, \phi_m) + \nabla \cdot j_{\theta_w}(Y, \alpha, n, a, \chi, \phi_m) - q_{\theta_w} = 0 \tag{4.36}$$

with the flux

$$j_{\theta_w}(Y, \alpha, n, a, \chi, \phi_m) \coloneqq -\exp(Y)\kappa(\alpha, n, a, \chi, \phi_m)\left[\nabla \phi_m - e_g\right] \tag{4.37}$$

and initial and boundary conditions

$$\phi_m = \phi_{m,0} \text{ for } t = 0, \qquad \phi_m = b^D_{\phi_m} \text{ on } \Gamma^D_{\phi_m}, \qquad j_{\theta_w} \cdot \mathbf{n} = b^N_{\phi_m} \text{ on } \Gamma^N_{\phi_m}. \tag{4.38}$$

Here $\Gamma^D_{\phi_m}$ and $\Gamma^N_{\phi_m}$ again refer to the Dirichlet and Neumann parts of the boundary, and we use the notation $\mathcal{F}_{\phi_m}$ to refer to both the PDE and the full model including initial and boundary conditions.

Using the definitions and steps of the previous section, the Richards equation may also be reformulated in a weak sense:

Model 5 (Richards Equation, Weak Formulation)
Let Ω be a given domain, $T \coloneqq [0, t_{max}]$ a time interval,

$$S \coloneqq (Y, \alpha, n, a, \chi, \phi_{m,0}) \tag{4.39}$$

a tuple of parameter fields on Ω and $(\phi_m, \partial_t \phi_m)$ a pair of states for which

$$\phi_m \in \varepsilon(b^D_{\phi_m}) + L^2\left(T; H^1_{\Gamma^D_{\phi_m}}(\Omega)\right), \quad \partial_t \phi \in L^2\left(T; H^{-1}(\Omega)\right), \tag{4.40}$$

with $\partial_t \phi_m$ the weak temporal derivative of ϕ_m. The pair $(\phi_m, \partial_t \phi_m)$ is the weak solution of the Richards equation if the condition

$$\forall\, \psi_{\phi_m} \in L^2(\Omega) : \langle \psi_{\phi_m}, \phi_m(0) - \phi_{m,0} \rangle_\Omega = 0 \tag{4.41}$$

holds for ϕ_m, and both functions together solve the equation

$$\begin{aligned} &\forall\, \psi_{\phi_m} \in L^2\left(T; H^1_{\Gamma^D_{\phi_m}}(\Omega)\right) : \\ &\qquad \langle \psi_{\phi_m}, \partial_t \theta_w(\alpha, n, \chi, \phi_m) \rangle + a_{\phi_m}(\phi_m, \psi_{\phi_m}) + b_{\phi_m}(\psi_{\phi_m}) = 0 \end{aligned} \tag{4.42}$$

with the form

$$a_{\phi_m}(\phi_m, \psi_{\phi_m}) \coloneqq -\left\langle \nabla \psi_{\phi_m}, j_{\theta_w}(Y, \alpha, n, a, \chi, \phi_m) \right\rangle, \tag{4.43}$$

the linear form

$$b_{\phi_m}(\psi_{\phi_m}) \coloneqq -\langle \psi_{\phi_m}, q_{\theta_w} \rangle + \left\langle \psi_{\phi_m}, b^N_{\phi_m} \right\rangle_{\Gamma^N_{\phi_m} \times T}, \tag{4.44}$$

and j_{θ_w} defined as in equation (4.37). Note that $a_{\phi_m}(\cdot, \cdot)$ uses the same notation as in model 2 but is no longer a bilinear form, since the relative conductivity κ in the definition of j_{θ_w} depends on the solution ϕ_m.

In analogy to the groundwater flow equation, we may again give an equation for the stationary limit where ϕ_m and θ_w no longer change with time:

Model 6 (Stationary Richards Equation)
Let Ω be a given domain,

$$S \coloneqq (Y, \alpha, n, a, \chi) \tag{4.45}$$

a tuple of parameter fields on Ω and

$$\phi_m \in \varepsilon(b^D_{\phi_m}) + H^1_{\Gamma^D_{\phi_m}}(\Omega)\,. \tag{4.46}$$

The function ϕ_m is the weak solution of the stationary Richards equation if it solves the equation

$$\forall\, \psi_{\phi_m} \in H^1_{\Gamma^D_{\phi_m}}(\Omega) : a_{\phi_m}(\phi_m, \psi_{\phi_m}) + b_{\phi_m}(\psi_{\phi_m}) = 0, \tag{4.47}$$

where $a_{\phi_m}(\cdot, \cdot)$ and $b_{\phi_m}(\cdot)$ result from equations (4.43) and (4.44) by replacing the scalar product of $L^2(\Omega \times T)$ with that of $L^2(\Omega)$.

4.3 Transport Equation

The convection-diffusion equation is the generic model for solute transport. It is based on the assumption that the solute concentration is low enough to neglect its influence on the water flow dynamics, and further implies that the process is limited to convection and hydrodynamic dispersion. In this context convection refers to solute migration with the water flow on the macroscopic level, while the hydrodynamic dispersion consists of the effects of both molecular diffusion and microscopic convective transport.

Similar to the groundwater flow equation and the Richards equation, the transport equation for conservative tracers is based on the conservation of mass. The total amount of solute in a given volume is represented by the total concentration

$$C \coloneqq \theta_w c, \tag{4.48}$$

where θ_w is again the percentage of volume filled with water, and c is the concentration of solute in the water phase. Since the total solute mass is conserved, the continuity equation states

$$\partial_t C(\phi) + \nabla \cdot j_C(\phi) - q_C(\phi, c) = 0, \tag{4.49}$$

where, in analogy to equation (4.3), j_C is the flux of total concentration and q_C is a term that represents concentration being added or removed from the system. Here ϕ as a function argument refers to the complete state, not a local evaluation, and therefore also represents gradients and fluxes as appropriate. In contrast to equation (4.3), q_C is a reaction term that explicitly depends on the concentration c for sinks, since solute mass can only be extracted if it is present. Assuming that solute mass is only introduced into the system where water is injected, it has the form

$$q_C(c) = \begin{cases} q_{\theta_w} c_{\text{in}} & \text{for } q_{\theta_w} \geq 0 \\ q_{\theta_w} c & \text{for } q_{\theta_w} < 0, \end{cases} \tag{4.50}$$

where c_{in} is the solute concentration in the injected water and q_{θ_w} is the source term for the water phase from the previous two sections. See figure 4.2 for an example of simulated solute transport in a heterogeneous medium.

Remark 21 *While we are using the state variable ϕ of the groundwater flow equation in these equations, it can readily be replaced by the matric head ϕ_m that we have used in the formulation of the Richards equation. The conversion of one representation to the other simply consists in adding respectively subtracting the gravity potential. While we restrict ourselves to the hydraulic head ϕ, the transport equation may also be used in conjunction with the Richards equation, either by reformulating the above equations in terms of ϕ_m, or reformulating the Richards equation in terms of ϕ, or implicitly converting between the two potentials. This only changes the parameter fields that implicitly parameterize the transport equation.*

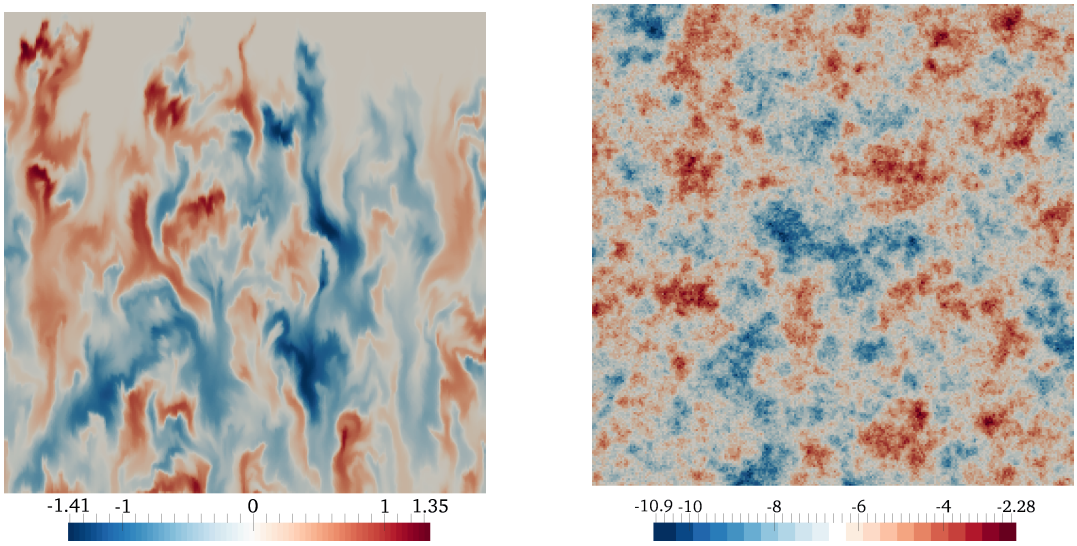

Figure 4.2: *Left:* Transport of a conservative tracer, with values denoting deviation from the mean concentration. Initial condition was a Gaussian random field with exponential covariance structure, water traverses the domain from the top boundary to the bottom boundary, tracer entering the domain at the top has homogeneous concentration. *Right:* Log-conductivity field underlying the groundwater flow equation that was used to simulate the transient solute transport.

The flux law for the comparatively simple model of solute transport described above is

$$j_C(\phi, c) \coloneqq -D(\phi)\nabla c + c j_{\theta_w}, \tag{4.51}$$

where D is the Bear-Scheidegger tensor for hydrodynamic dispersion [70], given by

$$(D(\phi))_{ij} \coloneqq [\lambda_l - \lambda_t] \frac{(j_{\theta_w})_i (j_{\theta_w})_j}{\left\| j_{\theta_w} \right\|_2} + \lambda_t \left[\left\| j_{\theta_w} \right\|_2 + \theta_w D_m \right] \delta_{ij}. \tag{4.52}$$

Here the tensor is reformulated as a function of the flux j_{θ_w} instead of the velocity to simplify notation, and D has to be divided by θ_w to arrive at the tensor from the literature. The scale parameters λ_l and λ_t are the longitudinal and transversal dispersion coefficients respectively, and D_m is the molecular diffusion tensor.

Inserting this flux law in equation (4.49) leads to the convection-diffusion equation:

Model 7 (Transport Equation, Classical Formulation)
Given a domain Ω, a time interval $T \coloneqq [0, t_{max}]$, the tuple of parameter fields

$$S \coloneqq (Z_s, Y, \phi_0, \lambda_l, \lambda_t, c_0) \tag{4.53}$$

and the resulting system state of the groundwater flow equation ϕ, the transport equation in classical formulation is the transient PDE

$$\mathcal{F}_c(S, \phi; c) \coloneqq \partial_t \left[\theta_w(Z_s, \phi) c\right] + \nabla \cdot j_C(Z_s, Y, \lambda_l, \lambda_t, \phi, c) - q_C(c) = 0 \tag{4.54}$$

with the flux

$$j_C(Z_s, Y, \lambda_l, \lambda_t, \phi, c) \coloneqq -\left[D(Z_s, Y, \lambda_l, \lambda_t, \phi)\nabla c + c j_{\theta_w}(Y, \phi)\right], \tag{4.55}$$

the dispersion tensor

$$(D(Z_s, Y, \lambda_l, \lambda_t, \phi))_{ij} \coloneqq [\lambda_l - \lambda_t] \frac{(j_{\theta_w})_i (j_{\theta_w})_j}{\left\|j_{\theta_w}\right\|_2} + \lambda_t \left[\left\|j_{\theta_w}\right\|_2 + \theta_w D_m\right] \delta_{ij}, \tag{4.56}$$

and initial and boundary conditions

$$c = c_0 \text{ for } t = 0, \qquad c = b_c^D \text{ on } \Gamma_c^D, \qquad j_C \cdot \mathbf{n} = b_c^N \text{ on } \Gamma_c^N. \tag{4.57}$$

In contrast to the groundwater flow equation and the Richards equation, which only use Dirichlet and Neumann boundary conditions, we allow an outflow boundary part $\Gamma_c^O = \Gamma \setminus \left[\Gamma_c^D \cup \Gamma_c^N\right]$. *We assume that* $j_{\theta_w} \cdot \mathbf{n} \leq 0$ *holds for* Γ_c^D *and* $j_{\theta_w} \cdot \mathbf{n} \geq 0$ *holds for* Γ_c^O, *which means that the flux transports the Dirichlet values into the domain and implicitly defines the values of c on the outflow boundary part* Γ_c^O. *The outflow boundary condition is*

$$j_C \cdot \mathbf{n} = c j_{\theta_w} \text{ on } \Gamma_c^O, \tag{4.58}$$

i.e. the diffusive flux vanishes on the outflow boundary part. If instead of the groundwater flow equation the Richards equation is used to define j_{θ_w}, *the definition of S above has to be replaced by*

$$S \coloneqq (Y, \alpha, n, a, \phi_{m,0}, \lambda_l, \lambda_t, c_0), \tag{4.59}$$

and ϕ *has to be replaced by* ϕ_m. *Note that in both cases some parameter fields in S don't appear in the formulation of the PDE. They are included in S to emphasize that the state c implicitly depends on them through the flux* j_{θ_w}.

The same procedure as for the groundwater flow equation and the Richards equation again leads to a weak formulation:

Model 8 (Transport Equation, Weak Formulation)
Let Ω *be a given domain,* $T \coloneqq [0, t_{max}]$ *a time interval,*

$$S \coloneqq (Z_s, Y, \phi_0, \lambda_l, \lambda_t, c_0) \tag{4.60}$$

a tuple of parameter fields on Ω, $(\phi, \partial_t \phi)$ *the pair of states from the groundwater flow equation, model 2, and* $(c, \partial_t c)$ *a pair of states for which*

$$c \in \varepsilon(b_c^D) + L^2\left(T; H^1_{\Gamma_c^D}(\Omega)\right), \quad \partial_t c \in L^2\left(T; H^{-1}(\Omega)\right), \tag{4.61}$$

with $\partial_t c$ *the weak temporal derivative of c. The pair* $(c, \partial_t c)$ *is the weak solution of the convection-diffusion equation if the condition*

$$\forall\, \psi_c \in L^2(\Omega) : \langle \psi_c, c(0) - c_0 \rangle_\Omega = 0 \tag{4.62}$$

holds for c, and both functions together solve the equation

$$\forall\, \psi_c \in L^2\left(T; H^1_{\Gamma_c^D}(\Omega)\right) : \langle \psi_c, \partial_t\left[\theta_w(Z_s,\phi)c\right]\rangle + a_c(c,\psi_c) + b_c(\psi_c) = 0 \tag{4.63}$$

with the bilinear form

$$a_c(c,\psi_c) \coloneqq -\langle \nabla\psi_c, j_C(Z_s, Y, \lambda_l, \lambda_t, \phi, c)\rangle - \left\langle \psi_c, q_C^{out}(c)\right\rangle + \left\langle \psi_c, c j_{\theta_w}(Y,\phi)\cdot \mathbf{n}\right\rangle_{\Gamma_c^O\times T}, \tag{4.64}$$

the linear form

$$b_c(\psi_c) \coloneqq -\left\langle \psi_c, q_C^{in}\right\rangle + \left\langle \psi_c, b_c^N\right\rangle_{\Gamma_c^N\times T}, \tag{4.65}$$

and j_C defined as in equation (4.55). Here q_C^{in} denotes the positive part of q_C according to equation (4.50), and q_C^{out} denotes the negative part. If the Richards equation is used, the parameter fields have to be replaced by

$$S \coloneqq (Y, \alpha, n, a, \phi_{m,0}, \lambda_l, \lambda_t, c_0) \tag{4.66}$$

and the pair of states $(\phi, \partial_t\phi)$ by $(\phi_m, \partial_t\phi_m)$. As before, $\mathcal{F}_c$ denotes both the different formulations of the PDE and the model as a whole, i.e. including initial and boundary conditions.

4.4 Adjoint Equations

The abstract formulation of the adjoint state method in section 2.4 assumed high regularity of the model equations for ease of presentation. We now derive the adjoint equations for the concrete model equations given in sections 4.1 to 4.3 under more realistic assumptions on the regularity of the equations and their solutions. The following approach is very similar to the one chosen in section 2.4, but uses a variational formulation and perturbation theory.

4.4.1 Groundwater Flow

As seen in sections 4.1 and 4.3, the groundwater flow equation $\mathcal{F}_\phi$ and the convection-diffusion equation $\mathcal{F}_c$ can be written in the form

$$\begin{aligned}\forall\, \psi_\phi \in L^2\left(T; H^1_{\Gamma_\phi^D}(\Omega)\right) :& \\ \langle \psi_\phi, \exp(Z_s)\partial_t\phi\rangle - \left\langle \nabla\psi_\phi, j_{\theta_w}(Y,\phi)\right\rangle - \langle \psi_\phi, q_{\theta_w}\rangle + \left\langle \psi_\phi, b_\phi^N\right\rangle_{\Gamma_\phi^N\times T}& \\ = 0&\end{aligned} \tag{4.67}$$

and

$$\begin{aligned}
&\forall\, \psi_c \in L^2\left(T; H^1_{\Gamma_c^D}(\Omega)\right): \\
&\quad \langle \psi_c, \exp(Z_s)\partial_t \phi c + \theta_w \partial_t c\rangle - \langle \nabla \psi_c, j_C(Z_s, Y, \lambda_l, \lambda_t, \phi, c)\rangle \\
&\quad - \left\langle \psi_c, q_C^{\text{out}}(c)\right\rangle + \left\langle \psi_c, c j_{\theta_w}(Y, \phi)\cdot \mathbf{n}\right\rangle_{\Gamma_c^O \times T} - \left\langle \psi_c, q_C^{\text{in}}\right\rangle + \left\langle \psi_c, b_c^N\right\rangle_{\Gamma_c^N \times T} \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad = 0, \quad (4.68)
\end{aligned}$$

where we have replaced the temporal derivative in the transport equation using the chain rule. Note that we assume that derivatives only act on the operand that follows directly and don't implicitly extend to the whole remaining expression. Furthermore, the initial conditions of ϕ and c are given by

$$\begin{aligned}
&\forall\, \psi_\phi \in L^2(\Omega)\colon \langle \psi_\phi, \phi(0) - \phi_0\rangle_\Omega = 0 \\
&\forall\, \psi_c \in L^2(\Omega)\colon \langle \psi_c, c(0) - c_0\rangle_\Omega = 0.
\end{aligned} \tag{4.69}$$

If the parameter fields are changed by adding a small perturbation δ_S,

$$\begin{aligned}
S &= (Z_s, Y, \phi_0, \lambda_l, \lambda_t, c_0) \to \widetilde{S} \\
\delta_S &\coloneqq \widetilde{S} - S = (\delta_{Z_s}, \delta_Y, \delta_{\phi_0}, \delta_{\lambda_l}, \delta_{\lambda_t}, \delta_{c_0}),
\end{aligned} \tag{4.70}$$

then the states also change,

$$\begin{aligned}
U &= (\phi, c) \to \widetilde{U} \\
\delta_U &\coloneqq \widetilde{U} - U = (\delta_\phi, \delta_c),
\end{aligned} \tag{4.71}$$

and since the forward problem is well-posed the corresponding changes δ_U are also small. Since the PDEs also hold for the perturbed pair $(\widetilde{S}, \widetilde{U})$, the changes in the individual terms of the equations have to cancel each other out. The linear forms $b_\phi(\cdot)$ and $b_c(\cdot)$, equations (4.20) and (4.65), contain neither parameters nor state variables, which means that only the bilinear forms $a_\phi(\cdot,\cdot)$ and $a_c(\cdot,\cdot)$ and the terms of the temporal derivatives have to be considered. An expansion of equations (4.67) and (4.68) up to first order in the changes δ_S and δ_U shows that

$$\begin{aligned}
&\forall\, \psi_\phi \in L^2\left(T; H^1_{\Gamma_\phi^D}(\Omega)\right): \\
&\quad \langle \psi_\phi, \exp(Z_s)\delta_{Z_s}\partial_t \phi + \exp(Z_s)\partial_t \delta_\phi\rangle - \left\langle \nabla \psi_\phi, \partial_Y j_{\theta_w}\delta_Y + \delta_{j_{\theta_w}}(\phi, \delta_\phi)\right\rangle \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad = 0 \quad (4.72)
\end{aligned}$$

and

$$\begin{aligned}
&\forall\, \psi_c \in L^2\left(T; H^1_{\Gamma_c^D}(\Omega)\right):\\
&\quad \langle \psi_c, \exp(Z_s)\delta_{Z_s}\partial_t\phi c + \exp(Z_s)\partial_t\delta_\phi c + \exp(Z_s)\partial_t\phi\delta_c\rangle\\
&\quad + \langle \psi_c, \exp(Z_s)\delta_{Z_s}\phi\partial_t c + \exp(Z_s)\delta_\phi\partial_t c + \theta_w\partial_t\delta_c\rangle\\
&\quad - \left\langle \nabla\psi_c, \partial_{Z_s}j_C\delta_{Z_s} + \partial_Y j_C\delta_Y + \partial_{\lambda_l}j_C\delta_{\lambda_l} + \partial_{\lambda_t}j_C\delta_{\lambda_t} + \delta_{j_C}(\phi,\delta_\phi) + \delta_{j_C}(c,\delta_c)\right\rangle\\
&\quad - \left\langle \psi_c, \partial_c q_C^{\text{out}}\delta_c\right\rangle + \left\langle \psi_c, \left[c\partial_Y j_{\theta_w}\delta_Y + c\delta_{j_{\theta_w}}(\phi,\delta_\phi) + \delta_c j_{\theta_w}\right]\cdot\mathbf{n}\right\rangle_{\Gamma_c^O\times T}\\
&\qquad\qquad = 0. \quad (4.73)
\end{aligned}$$

While most induced changes can be calculated directly through partial differentiation, the variations in j_{θ_w} and j_C caused by the changes in ϕ and c lack locality due to the involved gradients and are given by

$$\begin{aligned}
\delta_{j_{\theta_w}}(\phi,\delta_\phi) &\coloneqq -\exp(Y)\nabla\delta_\phi \quad (4.74)\\
\delta_{j_C}(\phi,\delta_\phi) &\coloneqq \delta_D(\phi,\delta_\phi)\nabla c\\
\delta_{j_C}(c,\delta_c) &\coloneqq D\nabla\delta_c + \delta_c j_{\theta_w}
\end{aligned}$$

and

$$\begin{aligned}
\delta_D(\phi,\delta_\phi) \coloneqq [\lambda_l-\lambda_t]\Bigg[&\frac{1}{2}\exp(Y)\left\|\nabla\phi\right\|_2^{-1}\left[\nabla\delta_\phi\left[\nabla\phi\right]^T + \nabla\phi\left[\nabla\delta_\phi\right]^T\right] \quad (4.75)\\
&- \exp(Y)\left\|\nabla\phi\right\|_2^{-3}\left[\nabla\phi\cdot\nabla\delta_\phi\right]\nabla\phi\left[\nabla\phi\right]^T\Bigg]\\
&+ \lambda_t\left[\exp(Y)\left\|\nabla\phi\right\|_2^{-1}\left[\nabla\phi\cdot\nabla\delta_\phi\right] + \exp(Z_s)D_m\delta_\phi\right]\mathbf{I},
\end{aligned}$$

where the arguments of the functions specify the cause of the perturbation. These equations can be obtained through linearization of the definitions from model 2 and model 8. Equation (4.75) and the partial derivatives of D can be obtained more easily when D is reformulated using

$$\begin{aligned}
D(\phi) &= [\lambda_l-\lambda_t]\left\|j_{\theta_w}\right\|_2^{-1} j_{\theta_w}\left[j_{\theta_w}\right]^T + \lambda_t\left[\left\|j_{\theta_w}\right\|_2 + \theta_w D_m\right]\mathbf{I} \quad (4.76)\\
&= [\lambda_l-\lambda_t]\exp(Y)\left\|\nabla\phi\right\|_2^{-1}\nabla\phi\left[\nabla\phi\right]^T + \lambda_t\left[\exp(Y)\left\|\nabla\phi\right\|_2 + \theta_w D_m\right]\mathbf{I}
\end{aligned}$$

instead of the componentwise formula in equation (4.52).

Summing the two equations (4.72) and (4.73), inserting the expressions from equa-

tion (4.74), and sorting all terms depending on δ_S to the front, we therefore have

$$\begin{aligned}
&\forall\, \psi_\phi \in L^2\left(T; H^1_{\Gamma_\phi^D}(\Omega)\right),\ \forall\, \psi_c \in L^2\left(T; H^1_{\Gamma_c^D}(\Omega)\right):\\
&\quad \langle \exp(Z_s)\psi_\phi \partial_t \phi + \exp(Z_s)\psi_c \partial_t [\phi c] + \nabla\psi_c \cdot \partial_{Z_s} D\nabla c, \delta_{Z_s}\rangle\\
&\quad + \left\langle \nabla\psi_\phi \cdot \left[-j_{\theta_w}\right] + \nabla\psi_c \cdot \left[\partial_Y D\nabla c - c j_{\theta_w}\right], \delta_Y \right\rangle\\
&\quad + \langle \nabla\psi_c \cdot \partial_{\lambda_l} D\nabla c, \delta_{\lambda_l}\rangle + \langle \nabla\psi_c \cdot \partial_{\lambda_t} D\nabla c, \delta_{\lambda_t}\rangle + \left\langle \psi_c, c\delta_Y j_{\theta_w}\cdot \mathbf{n}\right\rangle_{\Gamma_c^O\times T}\\
&\quad + \langle \psi_\phi, \exp(Z_s)\partial_t\delta_\phi\rangle + \langle \nabla\psi_\phi, \exp(Y)\nabla\delta_\phi\rangle\\
&\quad + \langle \psi_c, \exp(Z_s)\partial_t\delta_\phi c + \exp(Z_s)\partial_t\phi\delta_c + \exp(Z_s)\delta_\phi\partial_t c + \theta_w\partial_t\delta_c\rangle\\
&\quad + \left\langle \nabla\psi_c, \delta_D(\phi,\delta_\phi)\nabla c + c\exp(Y)\nabla\delta_\phi + D\nabla\delta_c - \delta_c j_{\theta_w}\right\rangle\\
&\quad + \left\langle \psi_c, -q^{\text{out}}_{\theta_w}\delta_c\right\rangle + \left\langle \psi_c, \left[-c\exp(Y)\nabla\delta_\phi + \delta_c j_{\theta_w}\right]\cdot\mathbf{n}\right\rangle_{\Gamma_c^O\times T}\\
&\qquad\qquad = 0, \quad (4.77)
\end{aligned}$$

where $q^{\text{out}}_{\theta_w}$ is the part of q_{θ_w} that is negative, compare equation (4.50).

Let $R(\delta_\phi, \delta_c)$ be a linear functional of $\delta_{\mathbf{Z}}$, i.e. a function that can be written as

$$R(\delta_\phi, \delta_c) = \langle R_\phi, \delta_\phi\rangle + \langle R_c, \delta_c\rangle, \tag{4.78}$$

where R_ϕ and R_c are suitable functions in $L^2(\Omega\times T)$. Note that

$$R(\delta_\phi, \delta_c) \coloneqq \langle \partial_\phi L, \delta_\phi\rangle + \langle \partial_c L, \delta_c\rangle \tag{4.79}$$

defines such a functional, with $R_\phi = \partial_\phi L$ and $R_c = \partial_c L$. Adding $R(\delta_\phi, \delta_c)$ to both sides of equation (4.77) results in

$$\begin{aligned}
&\forall\, \psi_\phi \in L^2\left(T; H^1_{\Gamma_\phi^D}(\Omega)\right),\ \forall\, \psi_c \in L^2\left(T; H^1_{\Gamma_c^D}(\Omega)\right):\\
&\quad \langle \exp(Z_s)\psi_\phi \partial_t \phi + \exp(Z_s)\psi_c \partial_t [\phi c] + \nabla\psi_c \cdot \partial_{Z_s} D\nabla c, \delta_{Z_s}\rangle\\
&\quad + \left\langle \nabla\psi_\phi \cdot \left[-j_{\theta_w}\right] + \nabla\psi_c \cdot \left[\partial_Y D\nabla c - c j_{\theta_w}\right], \delta_Y \right\rangle\\
&\quad + \langle \nabla\psi_c \cdot \partial_{\lambda_l} D\nabla c, \delta_{\lambda_l}\rangle + \langle \nabla\psi_c \cdot \partial_{\lambda_t} D\nabla c, \delta_{\lambda_t}\rangle + \left\langle \psi_c, \delta_Y j_{\theta_w}\cdot \mathbf{n}\right\rangle_{\Gamma_c^O\times T}\\
&\quad + \langle \psi_\phi, \exp(Z_s)\partial_t\delta_\phi\rangle + \langle \nabla\psi_\phi, \exp(Y)\nabla\delta_\phi\rangle\\
&\quad + \langle \psi_c, \exp(Z_s)\partial_t\delta_\phi c + \exp(Z_s)\partial_t\phi\delta_c + \exp(Z_s)\delta_\phi\partial_t c + \theta_w\partial_t\delta_c\rangle\\
&\quad + \left\langle \nabla\psi_c, \delta_D(\phi,\delta_\phi)\nabla c + c\exp(Y)\nabla\delta_\phi + D\nabla\delta_c - \delta_c j_{\theta_w}\right\rangle + \left\langle \psi_c, -q^{\text{out}}_{\theta_w}\delta_c\right\rangle\\
&\quad + \left\langle \psi_c, \left[-c\exp(Y)\nabla\delta_\phi + \delta_c j_{\theta_w}\right]\cdot\mathbf{n}\right\rangle_{\Gamma_c^O\times T} + \langle R_\phi, \delta_\phi\rangle + \langle R_c, \delta_c\rangle\\
&\qquad\qquad = R(\delta_\phi, \delta_c). \quad (4.80)
\end{aligned}$$

To simplify this equation, we group structurally similar terms together through the definition of

$$\begin{aligned}
r_S(\psi_\phi, \psi_c, \delta_S) \coloneqq\ & \langle \exp(Z_s)\psi_\phi \partial_t \phi + \exp(Z_s)\psi_c \partial_t [\phi c] + \nabla\psi_c \cdot \partial_{Z_s} D\nabla c, \delta_{Z_s}\rangle \quad (4.81)\\
&+ \left\langle \nabla\psi_\phi \cdot \left[-j_{\theta_w}\right] + \nabla\psi_c \cdot \left[\partial_Y D\nabla c - c j_{\theta_w}\right], \delta_Y \right\rangle\\
&+ \langle \nabla\psi_c \cdot \partial_{\lambda_l} D\nabla c, \delta_{\lambda_l}\rangle + \langle \nabla\psi_c \cdot \partial_{\lambda_t} D\nabla c, \delta_{\lambda_t}\rangle\\
&+ \langle -\exp(Z_s)\left[\psi_\phi(0) + \psi_c(0)c_0\right], \delta_{\phi_0}\rangle_\Omega + \langle -\theta_w(0)\psi_c(0), \delta_{c_0}\rangle_\Omega
\end{aligned}$$

for the parameter changes,

$$\begin{aligned} r_\phi(\psi_\phi, \psi_c, \delta_\phi) \coloneqq & \langle \psi_\phi, \exp(Z_s)\partial_t \delta_\phi \rangle + \langle \nabla \psi_\phi, \exp(Y) \nabla \delta_\phi \rangle \\ & + \langle \psi_c, \exp(Z_s)\partial_t \delta_\phi c + \exp(Z_s)\delta_\phi \partial_t c \rangle \\ & + \langle \nabla \psi_c, \delta_D(\phi, \delta_\phi)\nabla c + c \exp(Y) \nabla \delta_\phi \rangle \\ & + \langle R_\phi, \delta_\phi \rangle + \langle \exp(Z_s) \left[\psi_\phi(0) + \psi_c(0) c_0 \right], \delta_\phi(0) \rangle_\Omega \end{aligned} \tag{4.82}$$

for the hydraulic head change,

$$\begin{aligned} r_c(\psi_c, \delta_c) \coloneqq & \langle \psi_c, \exp(Z_s)\partial_t \phi \delta_c + \theta_w \partial_t \delta_c \rangle + \left\langle \nabla \psi_c, D \nabla \delta_c - \delta_c j_{\theta_w} \right\rangle \\ & + \left\langle \psi_c, -q^{\text{out}}_{\theta_w} \delta_c \right\rangle + \langle R_c, \delta_c \rangle + \langle \theta_w(0) \psi_c(0), \delta_c(0) \rangle_\Omega \end{aligned} \tag{4.83}$$

for the change in concentration, and

$$r_\Gamma(\psi_c, \delta_S, \delta_\phi, \delta_c) \coloneqq \left\langle \psi_c, \left[\delta_Y j_{\theta_w} - c \exp(Y) \nabla \delta_\phi + \delta_c j_{\theta_w} \right] \cdot \mathbf{n} \right\rangle_{\Gamma_c^Q \times T} \tag{4.84}$$

for the boundary integrals. Taking equation (4.69) into account for the terms that contain initial conditions, equation (4.80) then reads

$$\begin{aligned} \forall\, \psi_\phi \in L^2\left(T; H^1_{\Gamma_\phi^D}(\Omega)\right), \forall\, \psi_c \in L^2\left(T; H^1_{\Gamma_c^D}(\Omega)\right): \\ r_S(\psi_\phi, \psi_c, \delta_S) + r_\phi(\psi_\phi, \psi_c, \delta_\phi) + r_c(\psi_c, \delta_c) + r_\Gamma(\psi_c, \delta_S, \delta_\phi, \delta_c) \\ = R(\delta_\phi, \delta_c). \end{aligned} \tag{4.85}$$

The next step is the elimination of r_ϕ, r_c and r_Γ through careful choice of the functions ψ_ϕ and ψ_c, which can be achieved by solving the adjoint equations. Integrating the terms containing temporal derivatives by parts, using equation(4.75), and noting that the multiplication with tensors consisting of dyadic products can be reformulated as a product of scalar products, e.g.

$$[\nabla \psi_c]^T \left[\nabla \delta_\phi \left[\nabla \phi \right]^T \right] \nabla c = \left[[\nabla \psi_c]^T \nabla \delta_\phi \right] \left[[\nabla \phi]^T \nabla c \right] = \left[\nabla \psi_c \cdot \nabla \delta_\phi \right] \left[\nabla \phi \cdot \nabla c \right], \tag{4.86}$$

we arrive at

$$\begin{aligned} r_\phi(\psi_\phi, \psi_c, \delta_\phi) = & \langle -\exp(Z_s)\partial_t \psi_\phi, \delta_\phi \rangle + \langle \exp(Y) \nabla \psi_\phi, \nabla \delta_\phi \rangle \\ & + \langle -\exp(Z_s) c \partial_t \psi_c, \delta_\phi \rangle + \langle c \exp(Y) \nabla \psi_c, \nabla \delta_\phi \rangle \\ & + \left\langle \frac{1}{2} \left[\lambda_l - \lambda_t \right] \exp(Y) \left\| \nabla \phi \right\|_2^{-1} \left[\nabla \phi \cdot \nabla c \right] \nabla \psi_c, \nabla \delta_\phi \right\rangle \\ & + \left\langle \frac{1}{2} \left[\lambda_l - \lambda_t \right] \exp(Y) \left\| \nabla \phi \right\|_2^{-1} \left[\nabla \psi_c \cdot \nabla \phi \right] \nabla c, \nabla \delta_\phi \right\rangle \\ & + \left\langle - \left[\lambda_l - \lambda_t \right] \exp(Y) \left\| \nabla \phi \right\|_2^{-3} \left[\nabla \psi_c \cdot \nabla \phi \right] \left[\nabla \phi \cdot \nabla c \right] \nabla \phi, \nabla \delta_\phi \right\rangle \\ & + \left\langle \lambda_t \exp(Y) \left\| \nabla \phi \right\|_2^{-1} \left[\nabla \psi_c \cdot \nabla c \right] \nabla \phi, \nabla \delta_\phi \right\rangle \\ & + \langle \lambda_t \exp(Z_s) D_m \left[\nabla \psi_c \cdot \nabla c \right], \delta_\phi \rangle \\ & + \langle R_\phi, \delta_\phi \rangle + \langle \exp(Z_s) \left[\psi_\phi(t_{\max}) + \psi_c(t_{\max}) c(t_{\max}) \right], \delta_\phi(t_{\max}) \rangle_\Omega \end{aligned} \tag{4.87}$$

and

$$\begin{aligned} r_c(\psi_c, \delta_c) = &\langle \exp(Z_s)\partial_t\phi\psi_c, \delta_c\rangle + \langle -\partial_t\left[\theta_w\psi_c\right], \delta_c\rangle + \langle D\nabla\psi_c, \nabla\delta_c\rangle \\ &+ \left\langle \nabla\psi_c \cdot \left[-j_{\theta_w}\right], \delta_c\right\rangle + \left\langle -q_{\theta_w}^{\text{out}}\psi_c, \delta_c\right\rangle \\ &+ \langle R_c, \delta_c\rangle + \langle \theta_w(t_{\max})\psi_c(t_{\max}), \delta_c(t_{\max})\rangle_\Omega. \end{aligned} \tag{4.88}$$

If we choose ψ_c to be zero at $t_{\max}$, the end of the time interval, then the last term in the above equation vanishes, and if we choose $\psi_c = 0$ on Γ_c^O, then the same happens with the boundary integral r_Γ, equation (4.84). Since $\delta_c = 0$ holds weakly on Γ_c^D, the above equations also hold for general $\psi_c \in L^2(\Omega \times T)$ that don't necessarily vanish on Γ_c^D. Together with the homogeneous boundary conditions we have just introduced, this turns equation (4.85) into

$$\begin{aligned} \forall\, \psi_\phi \in L^2\left(T; H^1_{\Gamma_\phi^D}(\Omega)\right), \forall\, \psi_c \in L^2\left(T; H^1_{\Gamma_c^O}(\Omega)\right): \\ r_S(\psi_\phi, \psi_c, \delta_S) + r_\phi(\psi_\phi, \psi_c, \delta_\phi) + r_c(\psi_c, \delta_c) \\ = R(\delta_\phi, \delta_c). \end{aligned} \tag{4.89}$$

Enforcing $r_c = 0$ for all possible choices of δ_c then gives a PDE that defines the behavior of ψ_c as a function of space and time, starting at $t = t_{\max}$ and moving backwards in time.

We may formalize these observations by defining the adjoint equation of the convection-diffusion equation:

Model 9 (Adjoint Transport Equation)
Let Ω be a given domain, $T \coloneqq [0, t_{max}]$ a time interval,

$$S \coloneqq (Z_s, Y, \phi_0, \lambda_l, \lambda_t, c_0) \tag{4.90}$$

a tuple of parameter fields on Ω, $(\phi, \partial_t\phi)$ the pair of states from the groundwater flow equation, model 2, $(c, \partial_t c)$ the pair of states from the transport equation, model 8, and $(\psi_c, \partial_t\psi_c)$ a pair of states for which

$$\psi_c \in L^2\left(T; H^1_{\Gamma_c^O}(\Omega)\right), \quad \partial_t\psi_c \in L^2\left(T; H^{-1}(\Omega)\right), \tag{4.91}$$

with $\partial_t\psi_c$ the weak temporal derivative of ψ_c. The pair $(\psi_c, \partial_t\psi_c)$ is the weak solution of the adjoint convection-diffusion equation for the righthand side R_c, if the condition

$$\forall\, \delta_c \in L^2(\Omega) : \langle \delta_c, \psi_c(t_{max})\rangle = 0 \tag{4.92}$$

holds for ψ_c, and both functions together solve the equation

$$\forall\, \delta_c \in L^2\left(T; H^1_{\Gamma_c^O}(\Omega)\right) : \langle \delta_c, -\partial_t\left[\theta_w\psi_c\right]\rangle + a_{\psi_c}(\psi_c, \delta_c) + b_{\psi_c}(\delta_c) = 0 \tag{4.93}$$

with the bilinear form

$$a_{\psi_c}(\psi_c, \delta_c) \coloneqq \langle \nabla\delta_c, D\nabla\psi_c\rangle + \left\langle \delta_c, \nabla\psi_c \cdot \left[-j_{\theta_w}\right]\right\rangle + \left\langle \delta_c, \left[\exp(Z_s)\partial_t\phi - q^{out}_{\theta_w}\right]\psi_c\right\rangle \tag{4.94}$$

and the linear form

$$b_{\psi_c}(\delta_c) \coloneqq \langle \delta_c, R_c\rangle. \tag{4.95}$$

Here q_C^{in} denotes the positive part of q_C according to equation (4.50), and q_C^{out} denotes the negative part. If the Richards equation is used, the parameter fields have to be replaced by

$$S \coloneqq (Y, \alpha, n, a, \chi, \phi_{m,0}, \lambda_l, \lambda_t, c_0), \tag{4.96}$$

the pair of states $(\phi, \partial_t\phi)$ by $(\phi_m, \partial_t\phi_m)$, and $\exp(Z_s)\partial_t\phi$ has to be replaced with $\partial_t\theta_w$ in equation (4.94). We denote the adjoint model and its solution with $\mathcal{F}_c^\dagger$.

Note that the adjoint convection-diffusion equation is very similar to the original equation. The main differences between the two equations are the inverted time direction, the exchange of inflow and outflow parts of the boundary, the switch from the conservative formulation of the equation to the nonconservative formulation, and the modified source and reaction terms. Also note that the formulation above includes an implicit no-flow boundary condition on Γ_c^N for ψ_c, since this is the natural boundary condition of the variational formulation.

Remark 22 *We can gain further insight into the relevance and interpretation of the adjoint equation if we assume for a moment that all states are regular enough to use the classical formulation of the PDEs. Removing the integrals and multiplying the result by -1, equation (4.93) becomes*

$$\partial_t\left[\theta_w\psi_c\right] + \nabla\cdot\left[D\nabla\psi_c\right] + \nabla\psi_c\cdot j_{\theta_w} - \exp(Z_s)\partial_t\phi\psi_c + q^{out}_{\theta_w}\psi_c - R_c = 0. \tag{4.97}$$

Using the identity

$$\nabla\psi_c\cdot j_{\theta_w} = \nabla\cdot\left[\psi_c j_{\theta_w}\right] - \left[\nabla\cdot j_{\theta_w}\right]\psi_c \tag{4.98}$$

and the fact that ϕ and j_{θ_w} are coupled through the groundwater flow equation

$$\nabla\cdot j_{\theta_w} = q_{\theta_w} - \exp(Z_s)\partial_t\phi, \tag{4.99}$$

this equation can also be written in the form

$$\partial_t\left[\theta_w\psi_c\right] + \nabla\cdot\left[D\nabla\psi_c\right] + \nabla\cdot\left[\psi_c j_{\theta_w}\right] - q^{in}_{\theta_w}\psi_c - R_c = 0. \tag{4.100}$$

We may compare this with the original transport equation, which states

$$\partial_t\left[\theta_w c\right] - \nabla\cdot\left[D\nabla c\right] + \nabla\cdot\left[c j_{\theta_w}\right] - q^{in}_{\theta_w}c_{in} - q^{out}_{\theta_w}c = 0. \tag{4.101}$$

In a convection-dominated setting, the two states c and ψ_c are transported by the same water flux j_{θ_w} and therefore move along the same trajectories, but while c experiences diffusive spreading along its trajectory, the profile of ψ_c is actually sharpened. Both states are introduced into the system via the source term $q^{in}_{\theta_w}$, but while the value of c is fixed, that of ψ_c is determined implicitly through the PDE. This may be compared with the backwards heat equation on page 9, where the same anti-diffusive sharpening occurs. In essence, the adjoint state ψ_c is a measure for how sensitive the functional R_c is to changes of the state c at a given point in space and time, and therefore follows the same paths through the domain.

Applying the same considerations to the function r_ϕ, i.e. setting ψ_ϕ to zero for $t_{\max}$, introducing homogeneous Neumann conditions for ψ_ϕ on Γ^N_ϕ and requiring $r_\phi = 0$ for all possible choices of δ_ϕ, leads to the adjoint equation of the groundwater flow equation:

Model 10 (Adjoint Groundwater Flow Equation)
Let Ω be a given domain, $T \coloneqq [0, t_{max}]$ a time interval,

$$S \coloneqq (Z_s, Y, \phi_0, \lambda_l, \lambda_t, c_0) \tag{4.102}$$

a tuple of parameter fields on Ω, $(\phi, \partial_t\phi)$, $(c, \partial_t c)$ and $(\psi_c, \partial_t\psi_c)$ the solutions of model 2, model 8 and model 9 respectively, and $(\psi_\phi, \partial_t\psi_\phi)$ a pair of states for which

$$\psi_\phi \in L^2\left(T; H^1_{\Gamma^D_\phi}(\Omega)\right), \quad \partial_t\psi_\phi \in L^2\left(T; H^{-1}(\Omega)\right), \tag{4.103}$$

with $\partial_t\psi_\phi$ the weak temporal derivative of ψ_ϕ. The pair $(\psi_\phi, \partial_t\psi_\phi)$ is the weak solution of the adjoint groundwater flow equation for the righthand side R_ϕ, if the condition

$$\forall\, \delta_\phi \in L^2(\Omega) : \langle \delta_\phi, \psi_\phi(t_{max})\rangle = 0 \tag{4.104}$$

holds for ψ_ϕ, and both functions together solve the equation

$$\forall\, \delta_\phi \in L^2\left(T; H^1_{\Gamma^D_\phi}(\Omega)\right) : \langle \delta_\phi, -\exp(Z_s)\partial_t\psi_\phi\rangle + a_{\psi_\phi}(\psi_\phi, \delta_\phi) + b_{\psi_\phi}(\delta_\phi) = 0 \tag{4.105}$$

with the bilinear form

$$a_{\psi_\phi}(\psi_\phi, \delta_\phi) \coloneqq \langle \nabla\delta_\phi, \exp(Y)\nabla\psi_\phi\rangle \tag{4.106}$$

and the linear form

$$
\begin{aligned}
b_{\psi_\phi}(\delta_\phi) \coloneqq\ & \langle \nabla \delta_\phi, c \exp(Y) \nabla \psi_c \rangle \\
& + \left\langle \nabla \delta_\phi, \frac{1}{2} [\lambda_l - \lambda_t] \exp(Y) \left\| \nabla \phi \right\|_2^{-1} [\nabla \phi \cdot \nabla c] \nabla \psi_c \right\rangle \\
& + \left\langle \nabla \delta_\phi, \frac{1}{2} [\lambda_l - \lambda_t] \exp(Y) \left\| \nabla \phi \right\|_2^{-1} [\nabla \psi_c \cdot \nabla \phi] \nabla c \right\rangle \\
& + \left\langle \nabla \delta_\phi, - [\lambda_l - \lambda_t] \exp(Y) \left\| \nabla \phi \right\|_2^{-3} [\nabla \psi_c \cdot \nabla \phi] [\nabla \phi \cdot \nabla c] \nabla \phi \right\rangle \\
& + \left\langle \nabla \delta_\phi, \lambda_t \exp(Y) \left\| \nabla \phi \right\|_2^{-1} [\nabla \psi_c \cdot \nabla c] \nabla \phi \right\rangle \\
& + \langle \delta_\phi, \lambda_t \exp(Z_s) D_m [\nabla \psi_c \cdot \nabla c] - \exp(Z_s) c \partial_t \psi_c + R_\phi \rangle .
\end{aligned}
\tag{4.107}
$$

If the groundwater flow equation is considered without solute transport, it is enough to set both ψ_c and $\partial_t \psi_c$ to zero in the above equations. This only changes the source term, which takes the form

$$
b_{\psi_\phi}(\delta_\phi) \coloneqq \langle \delta_\phi, R_\phi \rangle . \tag{4.108}
$$

The adjoint model and its solution are denoted by $\mathcal{F}_\phi^\dagger$.

Again note that the adjoint equation is very similar to the original one. Apart from the inverted time direction, the only difference is the modified source term. In addition to R_ϕ, it contains contributions from the adjoint transport equation that reflect the dependency of the adjoint solution ψ_ϕ on ψ_c, which is an inversion of the dependency of c on ϕ. As in the case of the transport equation, the above formulation includes an implicit no-flow boundary condition on Γ_ϕ^N.

Solving both the adjoint transport equation $\mathcal{F}_c^\dagger(S, U; \psi_c)$ and the adjoint groundwater flow equation $\mathcal{F}_\phi^\dagger(S, U, \psi_c; \psi_\phi)$ and inserting the solutions ψ_c and ψ_ϕ into equation (4.89) yields

$$
r_S(\psi_\phi, \psi_c, \delta_S) = R(\delta_\phi, \delta_c), \tag{4.109}
$$

i.e. R as a function of δ_S can be evaluated through r_S.

If we are interested in $\mathrm{d}_p R$, i.e. the derivative of R with respect to a single component p of one of the parameter vectors $\mathbf{p}_i$, we can compute this derivative by introducing a small change δ_p in p and keeping all other components of $\mathbf{p}_i$ and all other parameter vectors fixed. This defines a small change $\delta_{\mathbf{p}_i}$ and through the linearization of equation (1.2) and equation (1.3) a corresponding change δ_{s_i}, which is zero everywhere except for a single discretization cell if p is a localized parameter, and a scaled-down linearized trend function otherwise. We may extend δ_{s_i} to a full change of parameters δ_S by setting all other entries to zero and use equation (4.109) to get

$$
\mathrm{d}_p R \approx \delta_p^{-1} r_S(\psi_\phi, \psi_c, \delta_S). \tag{4.110}
$$

A small change δ_p is only necessary if the trend model $\mathcal{X}_i$, see page 3, is nonlinear. If this is not the case, all involved equations are linear, and δ_p may be set to one to directly compute the desired derivative.

Remark 23 *In the notation of equations* (2.73) *and* (2.77)*, R is the linearization of the objective function L with regard to the states $U = (\phi, c)$, and R_ϕ and R_c are the derivatives $\partial_\phi L$ and $\partial_c L$. The adjoint operators $(\partial_\phi \mathcal{F}_\phi)^\dagger$ and $(\partial_\phi \mathcal{F}_c)^\dagger$ are contained in r_ϕ, $(\partial_c \mathcal{F}_c)^\dagger$ can be found in r_c, and $\partial_p \mathcal{F}_\phi$ and $\partial_p \mathcal{F}_c$ may be computed through r_S as given above. The main differences between the approach in section 2.4 and the one given here are the model regularity requirements and the formulation in terms of perturbations instead of partial derivatives.*

Since the adjoint equations and their solutions ψ_ϕ and ψ_c do not depend on the parameter change δ_S in any way, we may solve the adjoint equations once and reuse their solutions to compute the derivatives with regard to any number of parameters through simple postprocessing. If we are for example interested in the derivative with respect to a parameter p that belongs to the log-conductivity Y, equation (4.110) becomes

$$\mathrm{d}_p R \approx \left\langle \nabla \psi_\phi \cdot \left[-j_{\theta_w} \right] + \nabla \psi_c \cdot \left[\partial_Y D \nabla c - c j_{\theta_w} \right], \delta_p^{-1} \delta_Y \right\rangle, \tag{4.111}$$

where δ_Y is the change in Y that is introduced by a small change δ_p in p, and if we are instead interested in a parameter p belonging to the initial condition ϕ_0, it becomes

$$\mathrm{d}_p R \approx \left\langle -\exp(Z_s) \left[\psi_\phi(0) + \psi_c(0) c_0 \right], \delta_p^{-1} \delta_{\phi_0} \right\rangle_\Omega, \tag{4.112}$$

where δ_{ϕ_0} is the change in ϕ_0 that is introduced by δ_p.

4.4.2 Richards Regime

Replacing the groundwater flow equation with the Richards equation only changes the underlying parameters and leads to more terms that have to be taken into account. We have

$$\begin{aligned} \forall\, \psi_{\phi_\mathrm{m}} \in L^2\left(T; H^1_{\Gamma^D_{\phi_\mathrm{m}}}(\Omega)\right) : \left\langle \psi_{\phi_\mathrm{m}}, \partial_t \theta_w(\alpha, n, \chi, \phi_\mathrm{m}) \right\rangle \\ - \left\langle \nabla \psi_{\phi_\mathrm{m}}, j_{\theta_w}(Y, \alpha, n, a, \chi, \phi_\mathrm{m}) \right\rangle - \left\langle \psi_{\phi_\mathrm{m}}, q_{\theta_w} \right\rangle + \left\langle \psi_{\phi_\mathrm{m}}, b^N_{\phi_\mathrm{m}} \right\rangle_{\Gamma^N_{\phi_\mathrm{m}} \times T} = 0 \end{aligned} \tag{4.113}$$

instead of equation (4.67), and the parameter and state perturbations are given by

$$\begin{aligned} S &= (Y, \alpha, n, a, \chi, \phi_{m,0}, \lambda_l, \lambda_t, c_0) \to \widetilde{S} \\ \delta_S &\coloneqq \widetilde{S} - S = \left(\delta_Y, \delta_\alpha, \delta_n, \delta_a, \delta_\chi, \delta_{\phi_{m,0}}, \delta_{\lambda_l}, \delta_{\lambda_t}, \delta_{c_0} \right), \end{aligned} \tag{4.114}$$

and

$$\begin{aligned} U &= (\phi_\mathrm{m}, c) \to \widetilde{U} \\ \delta_U &\coloneqq \widetilde{U} - U = (\delta_{\phi_\mathrm{m}}, \delta_c). \end{aligned} \tag{4.115}$$

While the model equation is different, the central steps remain the same, starting with a linearized equilibrium condition for the perturbations. A Taylor expansion of the Richards equation yields

$$\begin{aligned}
&\forall\, \psi_{\phi_\mathrm{m}} \in L^2\left(T; H^1_{\Gamma^D_{\phi_\mathrm{m}}}(\Omega)\right): \\
&\quad \langle \psi_{\phi_\mathrm{m}}, \partial_t\left[\partial_\alpha \theta_w\right]\delta_\alpha + \partial_t\left[\partial_n \theta_w\right]\delta_n + \partial_t\left[\partial_\chi \theta_w\right]\delta_\chi + \partial_t\left[\partial_{\phi_\mathrm{m}} \theta_w\right]\delta_{\phi_\mathrm{m}} + \partial_{\phi_\mathrm{m}}\theta_w \partial_t \delta_{\phi_\mathrm{m}}\rangle \\
&\quad - \left\langle \nabla\psi_{\phi_\mathrm{m}}, \delta_Y j_{\theta_w} + \delta_\alpha \partial_\alpha j_{\theta_w} + \delta_n \partial_n j_{\theta_w} + \delta_\chi \partial_\chi j_{\theta_w}\right\rangle \\
&\quad - \langle \nabla\psi_{\phi_\mathrm{m}}, \exp(Y)\partial_{\phi_\mathrm{m}}\kappa\nabla\phi_\mathrm{m} - \exp(Y)\kappa\nabla\delta_{\phi_\mathrm{m}}\rangle \\
&\qquad\qquad = 0, \quad (4.116)
\end{aligned}$$

and an expansion of the transport equation gives

$$\begin{aligned}
&\forall\, \psi_c \in L^2\left(T; H^1_{\Gamma^D_c}(\Omega)\right): \\
&\quad \langle \psi_c, \partial_t\left[\partial_\alpha \theta_w c\right]\delta_\alpha + \partial_t\left[\partial_n \theta_w c\right]\delta_n + \partial_t\left[\partial_\chi \theta_w c\right]\delta_\chi\rangle \\
&\quad + \langle \psi_c, \partial_t\left[\partial_{\phi_\mathrm{m}} \theta_w c\right]\delta_{\phi_\mathrm{m}} + \partial_{\phi_\mathrm{m}}\theta_w c\partial_t\delta_{\phi_\mathrm{m}} + \partial_t\theta_w\delta_c + \theta_w\partial_t\delta_c\rangle \\
&\quad + \left\langle \nabla\psi_c, \delta_Y\left[\partial_Y D\nabla c - c j_{\theta_w}\right] + \delta_\alpha\left[\partial_\alpha D\nabla c - c\partial_\alpha j_{\theta_w}\right]\right\rangle \\
&\quad + \left\langle \nabla\psi_c, \delta_n\left[\partial_n D\nabla c - c\partial_n j_{\theta_w} + \partial_a D\nabla c - c\partial_a j_{\theta_w}\right]\right\rangle \\
&\quad + \left\langle \nabla\psi_c, \delta_a\left[\delta_\chi\left[\partial_\chi D\nabla c - c\partial_\chi j_{\theta_w}\right] + \delta_{\lambda_l}\partial_{\lambda_l} D\nabla c + \delta_{\lambda_t}\partial_{\lambda_t} D\nabla c\right]\right\rangle \\
&\quad + \left\langle \nabla\psi_c, \delta_{\phi_\mathrm{m}}\partial_{\phi_\mathrm{m}} D\nabla c + c\exp(Y)\partial_{\phi_\mathrm{m}}\kappa\nabla\phi_\mathrm{m} + c\exp(Y)\kappa\nabla\delta_{\phi_\mathrm{m}} + D\nabla\delta_c - \delta_c j_{\theta_w}\right\rangle \\
&\quad + \left\langle \psi_c, -q^{\mathrm{out}}_{\theta_w}\delta_c\right\rangle + \left\langle \psi_c, c\left[\delta_Y j_{\theta_w} + \delta_\alpha\partial_\alpha j_{\theta_w} + \delta_n\partial_n j_{\theta_w} + \delta_\chi\partial_\chi j_{\theta_w}\right]\cdot\mathbf{n}\right\rangle_{\Gamma^O_c\times T} \\
&\quad + \left\langle \psi_c, \left[-c\exp(Y)\partial_{\phi_\mathrm{m}}\kappa\delta_{\phi_\mathrm{m}}\nabla\phi_\mathrm{m} - c\exp(Y)\kappa\nabla\delta_{\phi_\mathrm{m}} + \delta_c j_{\theta_w}\right]\cdot\mathbf{n}\right\rangle_{\Gamma^O_c\times T} \\
&\qquad\qquad = 0. \quad (4.117)
\end{aligned}$$

Since these sums are zero, this also holds for the combined sum of all terms, which may again be divided into groups of similar contributions by defining

$$\begin{aligned}
r_S(\psi_{\phi_\mathrm{m}}, \psi_c, \delta_S) \coloneqq\ & \left\langle \nabla\psi_{\phi_\mathrm{m}}\cdot\left[-j_{\theta_w}\right] + \nabla\psi_c\cdot\left[\partial_Y D\nabla c - c j_{\theta_w}\right], \delta_Y\right\rangle \\
&+ \langle \partial_t\left[\partial_\alpha\theta_w\right]\psi_{\phi_\mathrm{m}} + \partial_t\left[\partial_\alpha\theta_w c\right]\psi_c, \delta_\alpha\rangle \\
&+ \left\langle \nabla\psi_{\phi_\mathrm{m}}\cdot\left[-\partial_\alpha j_{\theta_w}\right] + \nabla\psi_c\cdot\left[\partial_\alpha D\nabla c - c\partial_\alpha j_{\theta_w}\right], \delta_\alpha\right\rangle \\
&+ \langle \partial_t\left[\partial_n\theta_w\right]\psi_{\phi_\mathrm{m}} + \partial_t\left[\partial_n\theta_w c\right]\psi_c, \delta_n\rangle \\
&+ \left\langle \nabla\psi_{\phi_\mathrm{m}}\cdot\left[-\partial_n j_{\theta_w}\right] + \nabla\psi_c\cdot\left[\partial_n D\nabla c - c\partial_n j_{\theta_w}\right], \delta_n\right\rangle \\
&+ \langle \partial_t\left[\partial_\chi\theta_w\right]\psi_{\phi_\mathrm{m}} + \partial_t\left[\partial_\chi\theta_w c\right]\psi_c, \delta_\chi\rangle \\
&+ \left\langle \nabla\psi_{\phi_\mathrm{m}}\cdot\left[-\partial_\chi j_{\theta_w}\right] + \nabla\psi_c\cdot\left[\partial_\chi D\nabla c - c\partial_\chi j_{\theta_w}\right], \delta_\chi\right\rangle \\
&+ \left\langle \nabla\psi_{\phi_\mathrm{m}}\cdot\left[-\partial_a j_{\theta_w}\right] + \nabla\psi_c\cdot\left[\partial_a D\nabla c - c\partial_a j_{\theta_w}\right], \delta_a\right\rangle \\
&+ \langle \nabla\psi_c\cdot\partial_{\lambda_l} D\nabla c, \delta_{\lambda_l}\rangle + \langle \nabla\psi_c\cdot\partial_{\lambda_t} D\nabla c, \delta_{\lambda_t}\rangle \\
&+ \left\langle -\partial_{\phi_\mathrm{m}}\theta_w\left[\psi_{\phi_\mathrm{m}}(0) + \psi_c(0)c_0\right], \delta_{\phi_{m,0}}\right\rangle_\Omega + \langle -\theta_w(0)\psi_c(0), \delta_{c_0}\rangle_\Omega
\end{aligned} \quad (4.118)$$

for the parameter changes,

$$\begin{aligned} r_{\phi_\mathrm{m}}(\psi_{\phi_\mathrm{m}}, \psi_c, \delta_{\phi_\mathrm{m}}) \coloneqq & \langle \psi_{\phi_\mathrm{m}}, \partial_t \left[\partial_{\phi_\mathrm{m}} \theta_w\right] \delta_{\phi_\mathrm{m}} + \partial_{\phi_\mathrm{m}} \theta_w \partial_t \delta_{\phi_\mathrm{m}} \rangle \\ & + \langle \nabla \psi_{\phi_\mathrm{m}}, \exp(Y) \partial_{\phi_\mathrm{m}} \kappa \delta_{\phi_\mathrm{m}} \nabla \phi_\mathrm{m} + \exp(Y) \kappa \nabla \delta_{\phi_\mathrm{m}} \rangle \\ & + \langle \psi_c, \partial_t \left[\partial_{\phi_\mathrm{m}} \theta_w c\right] \delta_{\phi_\mathrm{m}} + \partial_{\phi_\mathrm{m}} \theta_w c \partial_t \delta_{\phi_\mathrm{m}} \rangle \\ & + \langle \nabla \psi_c, \delta_{\phi_\mathrm{m}} \partial_{\phi_\mathrm{m}} D \nabla c + c \exp(Y) \partial_{\phi_\mathrm{m}} \kappa \delta_{\phi_\mathrm{m}} \nabla \phi_\mathrm{m} + c \exp(Y) \kappa \nabla \delta_{\phi_\mathrm{m}} \rangle \\ & + \langle R_{\phi_\mathrm{m}}, \delta_{\phi_\mathrm{m}} \rangle + \langle \partial_{\phi_\mathrm{m}} \theta_w \left[\psi_{\phi_\mathrm{m}}(0) + \psi_c(0) c_0\right], \delta_{\phi_\mathrm{m}}(0) \rangle_\Omega \end{aligned} \tag{4.119}$$

for the matric head change, with R_{ϕ_m} defined as R_ϕ and R_c above,

$$\begin{aligned} r_c(\psi_c, \delta_c) \coloneqq & \langle \psi_c, \partial_t \theta_w \delta_c + \theta_w \partial_t \delta_c \rangle + \left\langle \nabla \psi_c, D \nabla \delta_c - \delta_c j_{\theta_w} \right\rangle \\ & + \left\langle \psi_c, -q_{\theta_w}^{\mathrm{out}} \delta_c \right\rangle + \langle R_c, \delta_c \rangle + \langle \theta_w \psi_c(0), \delta_c(0) \rangle_\Omega \end{aligned} \tag{4.120}$$

for the change in concentration, and

$$\begin{aligned} r_\Gamma(\psi_c, \delta_S, \delta_{\phi_\mathrm{m}}, \delta_c) \coloneqq & \left\langle \psi_c, c \left[\delta_Y j_{\theta_w} + \delta_\alpha \partial_\alpha j_{\theta_w} + \delta_n \partial_n j_{\theta_w} + \delta_\chi \partial_\chi j_{\theta_w}\right] \cdot \mathbf{n} \right\rangle_{\Gamma_c^O \times T} \\ & + \langle \psi_c, \left[-c \exp(Y) \partial_{\phi_\mathrm{m}} \kappa \delta_{\phi_\mathrm{m}} \nabla \phi_\mathrm{m}\right] \cdot \mathbf{n} \rangle_{\Gamma_c^O \times T} \\ & + \left\langle \psi_c, \left[-c \exp(Y) \kappa \nabla \delta_{\phi_\mathrm{m}} + \delta_c j_{\theta_w}\right] \cdot \mathbf{n} \right\rangle_{\Gamma_c^O \times T} \end{aligned} \tag{4.121}$$

for the boundary integrals. In analogy to equation (4.85),

$$\begin{aligned} & \forall\, \psi_{\phi_\mathrm{m}} \in L^2\left(T; H^1_{\Gamma^D_{\phi_\mathrm{m}}}(\Omega)\right), \forall\, \psi_c \in L^2\left(T; H^1_{\Gamma_c^D}(\Omega)\right): \\ & \quad r_S(\psi_{\phi_\mathrm{m}}, \psi_c, \delta_S) + r_{\phi_\mathrm{m}}(\psi_{\phi_\mathrm{m}}, \psi_c, \delta_{\phi_\mathrm{m}}) + r_c(\psi_c, \delta_c) + r_\Gamma(\psi_c, \delta_S, \delta_{\phi_\mathrm{m}}, \delta_c) \\ & \qquad = R(\delta_{\phi_\mathrm{m}}, \delta_c) \end{aligned} \tag{4.122}$$

holds with these definitions, and we again try to remove the parts of the lefthand side that depend on the perturbations of the system states. Integration by parts results in

$$\begin{aligned} r_{\phi_\mathrm{m}}(\psi_{\phi_\mathrm{m}}, \psi_c, \delta_{\phi_\mathrm{m}}) = & \langle \partial_t \left[\partial_{\phi_\mathrm{m}} \theta_w\right] \psi_{\phi_\mathrm{m}}, \delta_{\phi_\mathrm{m}} \rangle + \langle -\partial_t \left[\partial_{\phi_\mathrm{m}} \theta_w \psi_{\phi_\mathrm{m}}\right], \delta_{\phi_\mathrm{m}} \rangle \\ & + \langle \partial_t \left[\partial_{\phi_\mathrm{m}} \theta_w c\right] \psi_c, \delta_{\phi_\mathrm{m}} \rangle + \langle -\partial_t \left[\partial_{\phi_\mathrm{m}} \theta_w c \psi_c\right], \delta_{\phi_\mathrm{m}} \rangle \\ & + \langle \exp(Y) \kappa \nabla \psi_{\phi_\mathrm{m}}, \nabla \delta_{\phi_\mathrm{m}} \rangle + \langle \nabla \psi_{\phi_\mathrm{m}} \cdot \left[\exp(Y) \partial_{\phi_\mathrm{m}} \kappa \nabla \phi_\mathrm{m}\right], \delta_{\phi_\mathrm{m}} \rangle \\ & + \langle c \exp(Y) \kappa \nabla \psi_c, \nabla \delta_{\phi_\mathrm{m}} \rangle + \langle \nabla \phi_\mathrm{m} \cdot \left[c \exp(Y) \partial_{\phi_\mathrm{m}} \kappa \nabla \psi_c\right], \delta_{\phi_\mathrm{m}} \rangle \\ & + \langle \nabla \psi_c \cdot \partial_{\phi_\mathrm{m}} D \nabla c, \delta_{\phi_\mathrm{m}} \rangle + \langle R_{\phi_\mathrm{m}}, \delta_{\phi_\mathrm{m}} \rangle \\ & + \langle \partial_{\phi_\mathrm{m}} \theta_w \left[\psi_{\phi_\mathrm{m}}(t_{\max}) + \psi_c(t_{\max}) c(t_{\max})\right], \delta_{\phi_\mathrm{m}}(t_{\max}) \rangle_\Omega \end{aligned} \tag{4.123}$$

and

$$\begin{aligned} r_c(\psi_c, \delta_c) = & \langle \partial_t \theta_w \psi_c, \delta_c \rangle + \langle -\partial_t \left[\theta_w \psi_c\right], \delta_c \rangle + \langle D \nabla \psi_c, \nabla \delta_c \rangle \\ & + \left\langle \nabla \psi_c \cdot \left[-j_{\theta_w}\right], \delta_c \right\rangle + \left\langle -q_{\theta_w}^{\mathrm{out}} \psi_c, \delta_c \right\rangle + \langle R_c, \delta_c \rangle \\ & + \langle \theta_w(t_{\max}) \psi_c(t_{\max}), \delta_c(t_{\max}) \rangle_\Omega, \end{aligned} \tag{4.124}$$

which leads to the definition of the adjoint equations.

Since the adjoint transport equation is the same as before, except for minimal changes in notation that are necessary to accommodate the chosen formulation of the Richards equation, see model 9, it is enough to define the adjoint Richards equation:

Model 11 (Adjoint Richards Equation)
Let Ω be a given domain, $T := [0, t_{max}]$ a time interval,

$$S := (Y, \alpha, n, a, \chi, \phi_{m,0}, \lambda_l, \lambda_t, c_0) \tag{4.125}$$

a tuple of parameter fields on Ω, $(\phi_m, \partial_t \phi_m)$, $(c, \partial_t c)$ and $(\psi_c, \partial_t \psi_c)$ the solutions of model 5, model 8 and model 9 respectively, and $(\psi_{\phi_m}, \partial_t \psi_{\phi_m})$ a pair of states for which

$$\psi_{\phi_m} \in L^2\left(T; H^1_{\Gamma^D_{\phi_m}}(\Omega)\right), \quad \partial_t \psi_{\phi_m} \in L^2\left(T; H^{-1}(\Omega)\right), \tag{4.126}$$

with $\partial_t \psi_{\phi_m}$ the weak temporal derivative of ψ_{ϕ_m}. The pair $(\psi_{\phi_m}, \partial_t \psi_{\phi_m})$ is the weak solution of the adjoint Richards equation for the righthand side R_{ϕ_m}, if the condition

$$\forall\, \delta_{\phi_m} \in L^2(\Omega) : \langle \delta_{\phi_m}, \psi_{\phi_m}(t_{max}) \rangle = 0 \tag{4.127}$$

holds for ψ_{ϕ_m}, and both functions together solve the equation

$$\begin{aligned} &\forall\, \delta_{\phi_m} \in L^2\left(T; H^1_{\Gamma^D_{\phi_m}}(\Omega)\right) : \\ &\qquad \langle \delta_{\phi_m}, -\partial_t \left[\partial_{\phi_m} \theta_w \psi_{\phi_m}\right] \rangle + a_{\psi_{\phi_m}}(\psi_{\phi_m}, \delta_{\phi_m}) + b_{\psi_{\phi_m}}(\delta_{\phi_m}) = 0 \end{aligned} \tag{4.128}$$

with the bilinear form

$$\begin{aligned} a_{\psi_{\phi_m}}(\psi_{\phi_m}, \delta_{\phi_m}) := &\langle \nabla \delta_{\phi_m}, \exp(Y) \kappa \nabla \psi_{\phi_m} \rangle \\ &+ \langle \delta_{\phi_m}, \nabla \psi_{\phi_m} \cdot \left[\exp(Y) \partial_{\phi_m} \kappa \nabla \phi_m\right] \rangle \\ &+ \langle \delta_{\phi_m}, \partial_t \left[\partial_{\phi_m} \theta_w\right] \psi_{\phi_m} \rangle \end{aligned} \tag{4.129}$$

and the linear form

$$\begin{aligned} b_{\psi_{\phi_m}}(\delta_{\phi_m}) := &\langle \nabla \delta_{\phi_m}, c \exp(Y) \kappa \nabla \psi_c \rangle + \langle \delta_{\phi_m}, \nabla \phi_m \cdot \left[c \exp(Y) \partial_{\phi_m} \kappa \nabla \psi_c\right] \rangle \\ &+ \langle \delta_{\phi_m}, -\partial_{\phi_m} \theta_w c \partial_t \psi_c + \nabla \psi_c \cdot \partial_{\phi_m} D \nabla c + R_{\phi_m} \rangle. \end{aligned} \tag{4.130}$$

If ψ_{ϕ_m} and ψ_c are the solutions of the adjoint Richards equation and adjoint convection-diffusion equation respectively, equation (4.122) turns into

$$r_S(\psi_{\phi_\mathrm{m}}, \psi_c, \delta_S) = R(\delta_{\phi_\mathrm{m}}, \delta_c), \tag{4.131}$$

which may again be used to compute derivatives of R via

$$\mathrm{d}_p R \approx \delta_p^{-1} r_S(\psi_{\phi_\mathrm{m}}, \psi_c, \delta_S). \tag{4.132}$$

Remark 24 *While the Richards equation contains neither convection nor reaction terms, only a nonlinear diffusion term, the adjoint Richards equation as given above has convective and reactive components. The convection term originates in the nonlinearity of the diffusion term of the Richards equation, as can be seen in the Taylor expansion in equation* (4.116)*. The reaction term appears due to the integration by parts of the temporal derivatives and may be eliminated using the equation*

$$\partial_{\phi_m}\theta_w\partial_t\psi_{\phi_m} = \partial_t\left[\partial_{\phi_m}\theta_w\psi_{\phi_m}\right] - \partial_t\left[\partial_{\phi_m}\theta_w\right]\psi_{\phi_m}. \tag{4.133}$$

The resulting PDE is structurally simpler, since the reaction term vanishes and the temporal derivative is only applied to the solution ψ_{ϕ_m}. We nevertheless keep the adjoint Richards equation in the form given above, since it is beneficial to have the temporal derivative isolated when discretizing the PDE, see section 5.1.

To complement the transient adjoint equations that were derived above, we also provide their stationary counterparts. The adjoint stationary groundwater flow equation is obtained by leaving out all temporal derivatives in the derivation:

Model 12 (Adjoint Stationary Groundwater Flow Equation)
Let Ω be a given domain, Y a parameter field on Ω, ϕ the solution of model 3 and

$$\psi_\phi \in H^1_{\Gamma^D_\phi}(\Omega). \tag{4.134}$$

The function ψ_ϕ is the weak solution of the adjoint stationary groundwater flow equation for the righthand side R_ϕ, if it solves the equation

$$\forall\, \delta_\phi \in H^1_{\Gamma^D_\phi}(\Omega) : a_{\psi_\phi}(\psi_\phi, \delta_\phi) + b_{\psi_\phi}(\delta_\phi) = 0 \tag{4.135}$$

with the bilinear form

$$a_{\psi_\phi}(\psi_\phi, \delta_\phi) \coloneqq \langle \nabla\delta_\phi, \exp(Y)\nabla\psi_\phi\rangle_\Omega \tag{4.136}$$

and the linear form

$$b_{\psi_\phi}(\delta_\phi) \coloneqq \langle \delta_\phi, R_\phi\rangle_\Omega. \tag{4.137}$$

The adjoint stationary Richards equation can be derived the same way:

Model 13 (Adjoint Stationary Richards Equation)
Let Ω be a given domain,

$$S \coloneqq (Y, \alpha, n, a, \chi) \tag{4.138}$$

a tuple of parameter fields on Ω, ϕ_m the solution of model 6, and

$$\psi_{\phi_m} \in H^1_{\Gamma^D_{\phi m}}(\Omega). \tag{4.139}$$

The function ψ_{ϕ_m} is the weak solution of the adjoint stationary Richards equation for the righthand side R_{ϕ_m}, if it solves the equation

$$\forall\, \delta_{\phi_m} \in H^1_{\Gamma^D_{\phi m}}(\Omega) : a_{\psi_{\phi m}}(\psi_{\phi_m}, \delta_{\phi_m}) + b_{\psi_{\phi m}}(\delta_{\phi_m}) = 0 \tag{4.140}$$

with the bilinear form

$$a_{\psi_{\phi_m}}(\psi_{\phi_m}, \delta_{\phi_m}) \coloneqq \langle \nabla \delta_{\phi_m}, \exp(Y)\kappa \nabla \psi_{\phi_m} \rangle + \langle \delta_{\phi_m}, \nabla \psi_{\phi_m} \cdot [\exp(Y)\partial_{\phi_m}\kappa \nabla \phi_m] \rangle$$

and the linear form

$$b_{\psi_{\phi m}}(\delta_{\phi_m}) \coloneqq \langle \delta_{\phi_m}, R_{\phi_m} \rangle.$$

5 Implementation Details

For the computer-assisted inversion of real or synthetic experimental data the governing equations, sections 4.1 to 4.3, and their adjoint counterparts, section 4.4, have to be discretized. Several discretization strategies for PDEs exist, with the Finite Volume and Finite Element Methods and their varieties being among the most popular. In most cases the spatial and temporal discretization are separated, since computing timestep after timestep both mimics the natural flow of information and reduces the number of unknowns that have to be handled at any given time. This chapter presents the implemented discretization schemes as used for the examples in chapter 6 and discusses further implementation details.

The temporal and spatial discretization we use is an application of the Runge Kutta Discontinuous Galerkin (RKDG) method developed by *Cockburn and Shu* [18]. Section 5.1 introduces a variant of the Runge Kutta methods, and section 5.2 discusses the spatial discretization of PDEs using the discontinuous Galerkin method. The third section presents further details of the implementation and mentions the software libraries that were used.

5.1 Time Discretization

All of the considered model equations and adjoint equations are of the same type, namely stationary or transient convection-diffusion-reaction equations. With the exception of the Richards equation the considered equations are linear. It therefore makes sense to discuss the discretization of a generic convection-diffusion-reaction equation, which can be adapted for each of the individual model equations.

For a generic system state u, we consider a transient PDE of the form

$$\forall\, \psi_u \in L^2\left(T; H^1_{\Gamma_u^D}(\Omega)\right) : \langle \psi_u, \partial_t g_u(u)\rangle + a_u(u, \psi_u) + b_u(\psi_u) = 0 \qquad (5.1)$$

on a domain $\Omega \subset \mathbb{R}^d$, where $g_u(\cdot)$ depends only on the value of u itself and not its spatial derivatives. This $(d+1)$-dimensional variational formulation could in principle be discretized directly, i.e. a finite element scheme or finite volume scheme could be applied on the domain $\Omega \times T$. However, this would lead to very large discrete systems, and in practice a separation of the discretization in time and the discretization in

space is often preferred [18]. Assuming that the solution is regular enough to allow a pointwise evaluation in time, the above equation can be restated as

$$\forall\,\tau \in T,\, \forall\,\psi_u \in H^1_{\Gamma_u^D}(\Omega) : \langle \psi_u, [\partial_t g_u(u)](\tau)\rangle_\Omega + a_u^{(\tau)}(u(\tau), \psi_u) + b_u^{(\tau)}(\psi_u) = 0, \quad (5.2)$$

where the stationary counterparts $a_u^{(\tau)}(\cdot,\cdot)$ and $b_u^{(\tau)}(\cdot)$ of $a_u(\cdot,\cdot)$ and $b_u(\cdot)$ are obtained by fixing $t = \tau$ in their definitions and replacing the scalar product of $L^2(\Omega \times T)$ with that of $L^2(\Omega)$. Since ψ_u no longer depends on t in this formulation, we may pull the temporal derivative out of the scalar product and get

$$\forall\,\tau \in T,\, \forall\,\psi_u \in H^1_{\Gamma_u^D}(\Omega) : \mathrm{d}_t\left\langle \psi_u, g_u^{(\tau)}(u(\tau))\right\rangle_\Omega + a_u^{(\tau)}(u(\tau), \psi_u) + b_u^{(\tau)}(\psi_u) = 0, \quad (5.3)$$

where $g_u^{(\tau)}(\cdot)$ again refers to setting $t = \tau$ in the time-dependent functions in $g_u(\cdot)$. The temporal derivative and the spatial parts may now be discretized separately. Following the method of lines, we first discretize space. This yields

$$\begin{aligned}&\forall\,\psi_h \in V_h:\\ &\left\langle \psi_h, g_h^{(t_{k+1})}(u_h(t_{k+1}))\right\rangle_\Omega - \left\langle \psi_h, g_h^{(t_k)}(u_h(t_k))\right\rangle_\Omega + \int_{t_k}^{t_{k+1}} \left[a_h^{(\tau)}(u_h(\tau), \psi_h) + b_h^{(\tau)}(\psi_h)\right] \mathrm{d}\tau\\ &\qquad = 0 \quad (5.4)\end{aligned}$$

for a sequence of discrete times t_k with

$$0 = t_0 < \cdots < t_k < t_{k+1} < \cdots \leq t_{\max}, \quad (5.5)$$

where V_h is a discrete replacement for $H^1_{\Gamma_u^D}(\Omega)$, $u_h^k \in V_h$ for all $k \in \mathbb{N}$, and $g_h(\cdot)$, $a_h^{(t_k)}(\cdot,\cdot)$ and $b_h^{(t_k)}(\cdot)$ are suitable discretizations of $g_u(\cdot)$, $a_u^{(t_k)}(\cdot,\cdot)$ and $b_u^{(t_k)}(\cdot)$ that operate on V_h. The boundary conditions are no longer present in this discrete formulation and its space V_h, since they will be incorporated in the discretizations of the spatial operators in section 5.2.

5.1.1 Runge-Kutta Methods

The Runge-Kutta methods approximate the above time integral with quadrature formulas, with the concrete choice of quadrature formula deciding about the properties of the method, e.g. its speed or stability. The simplest variant is the explicit Euler method, which creates a sequence u_h^k through iteratively solving

$$\begin{aligned}\forall\,\psi_h \in V_h: \left\langle \psi_h, g_h^{(t_{k+1})}\left(u_h^{k+1}\right)\right\rangle_\Omega - \left\langle \psi_h, g_h^{(t_k)}\left(u_h^k\right)\right\rangle_\Omega&\\ + \delta_k\left[a_h^{(t_k)}\left(u_h^k, \psi_h\right) + b_h^{(t_k)}(\psi_h)\right] = 0,&\quad (5.6)\end{aligned}$$

where $\delta_k \coloneqq t_{k+1} - t_k$, starting with the initial condition

$$\forall\, \psi_h \in V_h \colon \left\langle \psi_h, u_h^0 - u_0 \right\rangle_\Omega = 0, \tag{5.7}$$

where u_0 is the initial value of u, and discretizing the boundary conditions in the same fashion. The implicit Euler method works the same way, but solves

$$\begin{aligned} \forall\, \psi_h \in V_h \colon \left\langle \psi_h, g_h^{(t_{k+1})}\left(u_h^{k+1}\right)\right\rangle_\Omega - \left\langle \psi_h, g_h^{(t_k)}\left(u_h^{k}\right)\right\rangle_\Omega \\ + \delta_k \left[a_h^{(t_{k+1})}\left(u_h^{k+1}, \psi_h\right) + b_h^{(t_{k+1})}\left(\psi_h\right)\right] = 0 \end{aligned} \tag{5.8}$$

instead. Both methods compute a sequence u_h^k of discrete states that are an approximation of $u(t_k)$. Note that the equations (5.6) and (5.8) can both be interpreted as discretized stationary PDEs, and that therefore techniques for the solution of such systems can directly be applied to the two iterative schemes above.

Runge-Kutta methods of higher order are usually specified with the help of Butcher tableaux, tables that are used to recursively compute increments for the discrete state u_h^k to calculate u_h^{k+1}. Following *Di Pietro and Ern* [21], we choose a different formulation based on intermediate system states:

Discretization 1 (Semi-Implicit Runge-Kutta Method)
Let s be a number of stages, $\mathbf{d} \in \mathbb{R}^{s+1}$ a vector of step widths with $d_1 = 0$ and $d_s = 1$,

$$\boldsymbol{\alpha} = \begin{pmatrix} -1 & 1 & 0 & \cdots & 0 \\ \alpha_{21} & a_{22} & 1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ \alpha_{s1} & \alpha_{s2} & \cdots & \alpha_{ss} & 1 \end{pmatrix} \tag{5.9}$$

an $s \times (s+1)$ matrix in which the sum of each row is zero, and

$$\boldsymbol{\beta} = \begin{pmatrix} \beta_{11} & \beta_{12} & 0 & \cdots & 0 \\ \beta_{21} & \beta_{22} & \beta_{23} & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ \beta_{s1} & \beta_{s2} & \cdots & \beta_{ss} & \beta_{s(s+1)} \end{pmatrix} \tag{5.10}$$

another $s \times (s+1)$ matrix. Given a previous iteration u_h^k at time t_k and a time step δ_k, the semi-implicit Runge-Kutta scheme for the parameters $(\mathbf{d}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ produces an iteration u_h^{k+1} for the time $t_{k+1} \coloneqq \delta_k$ by setting

$$\forall\, 0 \le i \le s \colon t_{k,i} \coloneqq t_k + d_{i+1}\delta_k, \tag{5.11}$$

and

$$u_h^{k,0} \coloneqq u_h^k, \tag{5.12}$$

solving the system of equations

$$\forall\, 1 \leq i \leq s,\, \forall\, \psi_h \in V_h\colon$$
$$\sum_{j=0}^{i} \alpha_{i(j+1)} \left\langle \psi_h, g_h^{(t_{k,i})}\left(u_h^{k,i}\right)\right\rangle_\Omega + \sum_{j=0}^{i} \delta_k \beta_{i(j+1)} \left[a_h^{(t_{k,j})}\left(u_h^{k,j}, \psi_h\right) + b_h^{(t_{k,j})}\left(\psi_h\right)\right] = 0 \quad (5.13)$$

for the intermediate states $u_h^{k,i}, 1 \leq i \leq s$*, with the boundary conditions evaluated in the same fashion, and finally setting*

$$u_h^{k+1} \coloneqq u_h^{k,s}. \quad (5.14)$$

Equation (5.13) is a system of s equations for the s unknowns $u_h^{k,i}$. Since the equation for a $u_h^{k,i}$ depends only on the previous $u_h^{k,j}, j < i$, and possibly $u_h^{k,i}$ itself if $\beta_{i(i+1)} \neq 0$, this system can be solved iteratively from $u_h^{k,0}$ to $u_h^{k,s}$. Note that the equations in (5.13) can again be seen as discretized stationary PDEs, which means that the task of solving a transient PDE has been reduced to repeatedly solving stationary PDEs. If $\beta_{i(i+1)} \neq 0$ for at least one $1 \leq i \leq s$, then the time discretization is implicit and the PDE to be solved in that stage is the stationary limit of the original PDE with additional reaction terms and source terms stemming from the time discretization. If $\beta_{i(i+1)} = 0$ for all $1 \leq i \leq s$, then the scheme is explicit and the resulting stationary PDEs have a significantly simpler structure, since the spatial derivatives only operate on iterations and intermediate stages that have already been computed.

The explicit Euler method is the simplest example of such a method:

Discretization 2 (Explicit Euler Method)
The explicit Euler method is a Runge-Kutta method as in discretization 1, with the choice

$$\mathbf{d} \coloneqq \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad \boldsymbol{\alpha} \coloneqq \begin{pmatrix} -1 & 1 \end{pmatrix}, \quad \boldsymbol{\beta} \coloneqq \begin{pmatrix} 1 & 0 \end{pmatrix}. \quad (5.15)$$

It is an explicit time discretization scheme of first order.

The definition of the implicit Euler method is almost identical:

Discretization 3 (Implicit Euler Method)
The implicit Euler method is a Runge-Kutta method as in discretization 1, with the choice

$$\mathbf{d} \coloneqq \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad \boldsymbol{\alpha} \coloneqq \begin{pmatrix} -1 & 1 \end{pmatrix}, \quad \boldsymbol{\beta} \coloneqq \begin{pmatrix} 0 & 1 \end{pmatrix}. \quad (5.16)$$

It is an implicit time discretization scheme of first order.

The implicit Euler method is highly diffusive and should therefore only be applied where this numerical diffusion adds stability and does not negatively impact the quality of the solution. The explicit Euler method is unsuitable for stiff problems and additionally can fail when combined with higher-order spatial discretizations [21]. We therefore require higher-order temporal discretization schemes. Normally, the benefits of such methods are limited due to the low regularity of the formulation of the PDE and its solution, and the cost per time step becomes prohibitive when the number of stages is too large. We therefore focus on methods that are of second order. One of them is a reformulation of Heun's Method:

Discretization 4 (Heun Scheme)
The Heun scheme is a Runge-Kutta method as in discretization 1, with the choice

$$\mathbf{d} \coloneqq \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}, \quad \boldsymbol{\alpha} \coloneqq \begin{pmatrix} -1 & 1 & 0 \\ -1/2 & -1/2 & 1 \end{pmatrix}, \quad \boldsymbol{\beta} \coloneqq \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1/2 & 0 \end{pmatrix}. \tag{5.17}$$

It is an explicit time discretization scheme of second order.

This method is identical to Heun's Method in the case of linear models and is an example of strong stability-preserving (SSP) Runge-Kutta methods [32]. We accompany this definition with that of a second-order implicit method, a reformulation of the strongly S-stable method by *Alexander* [2]:

Discretization 5 (Alexander Scheme)
The Alexander scheme is a Runge-Kutta method as in discretization 1, with the choice

$$\mathbf{d} \coloneqq \begin{pmatrix} 0 \\ \gamma \\ 1 \end{pmatrix}, \quad \boldsymbol{\alpha} \coloneqq \begin{pmatrix} -1 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix}, \quad \boldsymbol{\beta} \coloneqq \begin{pmatrix} 0 & \gamma & 0 \\ 0 & 1-\gamma & \gamma \end{pmatrix} \tag{5.18}$$

with $\gamma \coloneqq 1 - 2^{-1/2}$. *This method performs an implicit Euler step of smaller step width* $\gamma\delta_k$ *and uses the result to modify a second implicit Euler step with full step width* δ_k. *It is an implicit time discretization scheme of second order.*

5.1.2 Adaptive Timestepping

An important aspect of time stepping schemes is the control of the step width δ_k. If the step width is chosen too large the simulation may become unstable or fail to provide accurate results due to the accumulation of discretization errors, while a very small step width will lead to significantly increased computational costs. Additionally, the dynamic adjustment of δ_k based on criteria derived from the discrete solution and its evolution may further improve the accuracy of the method.

Explicit schemes like the explicit Euler method or the Heun scheme are limited by the Courant-Friedrichs-Lewy (CFL) condition, which states that the step width δ_k has to fulfill

$$\delta_k \leq \frac{C}{v} h, \tag{5.19}$$

where v is the characteristic propagation speed of the physical process and C is a constant that depends on the discretization. In the case of convection-dominated transport processes as described in section 4.3 the propagation speed is

$$v \coloneqq \frac{\left| j_{\theta_w} \right|}{\theta_w}. \tag{5.20}$$

For the space discretizations we discuss in the next section, the constant is given by

$$C \coloneqq \frac{1}{2l+1}, \tag{5.21}$$

where l is the degree of the polynomials used in the spatial discretization and the Runge-Kutta method is assumed to have $s = l + 1$ stages [19].

To provide a simple control mechanism for the Alexander scheme, we define the following additional method:

Discretization 6 (Scheme for Adaptive Step Control)
The method used in the control of discretization 5 is defined by the choice

$$\mathbf{d} \coloneqq \begin{pmatrix} 0 \\ \gamma \\ 1 \end{pmatrix}, \quad \boldsymbol{\alpha} \coloneqq \begin{pmatrix} -1 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix}, \quad \boldsymbol{\beta} \coloneqq \begin{pmatrix} 0 & \gamma & 0 \\ 0 & 0 & 1-\gamma \end{pmatrix} \tag{5.22}$$

with $\gamma \coloneqq 1 - 2^{-1/2}$. It is an implicit time discretization scheme of first order.

This scheme performs two implicit Euler steps, one with a step width of $\gamma\delta_k$ and one with a step width of $[1-\gamma]\,\delta_k$. Due to the intermediate stage the discretization 6 has a lower error constant than the implicit Euler method, and since the intermediate stage of discretization 5 can be reused, only one of the implicit Euler steps has to be performed.

Since the two discretizations 5 and 6 have different order, the difference of their solutions is an estimate of the error of the method with lower order if the time step δ_k is large enough. As a simple heuristic, we assume that this is always the case and use the difference of the solutions to control the step width. As in the case of embedded Runge-Kutta methods [24], we control the error of the lower-order method, discretization 6, but use the results of the higher-order method, discretization 5.

5.2 Space Discretization

Due to the results of the previous section, the spatial discretization can be handled in the same way for both the stationary and the transient models. All model equations consist of one or more operators acting on the solution u, i.e. diffusion terms of the form

$$a_u^{\text{diff}}(u, \psi_u) = \langle \nabla \psi_u, K\kappa(u)\nabla u \rangle_\Omega \tag{5.23}$$

with a solution-independent diffusion tensor K and possibly a solution-dependent contribution $\kappa(u)$, convection terms in either conservative formulation

$$a_u^{\text{con}}(u, \psi_u) = \langle \nabla \psi_u, u j_{\text{con}} \rangle_\Omega \tag{5.24}$$

or non-conservative formulation

$$a_u^{\text{con}}(u, \psi_u) = \langle \psi_u, \nabla u \cdot j_{\text{con}} \rangle_\Omega \tag{5.25}$$

with a convective flux j_{con}, or reaction terms of the form

$$a_u^{\text{rea}}(u, \psi_u) = \langle \psi_u, r_u u \rangle_\Omega. \tag{5.26}$$

Additionally, all the equations contain a source term of the form

$$b_u(\psi_u) = \langle \psi_u, q_u \rangle_\Omega. \tag{5.27}$$

In this regard, each of the considered equations can be written as

$$\forall\, \psi_u \in V\colon a_u^{\text{diff}}(u, \psi_u) + a_u^{\text{con}}(u, \psi_u) + a_u^{\text{rea}}(u, \psi_u) + b_u(\psi_u) = 0, \tag{5.28}$$

where $a_u^{\text{rea}}(\cdot, \cdot)$ and $b_u(\cdot)$ potentially contain contributions of the discretized temporal derivative, see section 5.1. The goal of this section is the construction of finite-dimensional approximations for the individual terms. Setting

$$\forall\, \psi_h \in V_h\colon a_{u,h}^{\text{diff}}(u, \psi_u) + a_{u,h}^{\text{con}}(u, \psi_u) + a_{u,h}^{\text{rea}}(u, \psi_u) + b_{u,h}(\psi_u) = 0 \tag{5.29}$$

for these approximations, the resulting finite-dimensional system may be solved for an approximate solution u_h of (5.28), which is either the solution of the discretized model or one of the stages of the Runge-Kutta schemes from the previous section.

5.2.1 Discontinuous Galerkin

We discretize these terms using an Internal Penalty discontinuous Galerkin (IPdG) approach [21]. For simplicity, we assume that the triangulation $\mathcal{E}_h$ used for the discretization of the model equations is identical to that of the parameter fields, page 2. There is typically no reason to use a coarser structured grid for the parameters, since

the only global operation on the parameters in algorithm 6 ($\mathbf{PCG}_c$) is the multiplication with $\mathbf{Q_{PP}}$, which often has negligible cost in comparison to solving the model equations. If unstructured grids are required, e.g. for local refinement or complicated domains, then transfer operators have to be defined between the grids.

The trial space V_h of the discontinuous Galerkin method is usually a broken polynomial space, i.e. a space of functions that are polynomials when restricted to individual elements. We define the finite element space

$$V_h^{(k)}(\mathcal{E}_h) \coloneqq \left\{ v \in L^2(\Omega) \,\middle|\, \forall E \in \mathcal{E}_h \colon v|_E \in Q_d^k \right\}, \tag{5.30}$$

where Q_d^k is the set of polynomials in d dimensions with maximum degree k per dimension. These functions may be discontinuous across the interfaces between elements, as the name of the method implies, and we therefore need notation to handle these discontinuities.

We refer to the $(d-1)$-dimensional intersections of the elements E_i with each other and the boundary Γ as faces F_j and group all such faces in a set $\mathcal{F}_h$. Interior faces, i.e. those faces that belong to two different elements E^+ and E^-, are collected in $\mathcal{F}_h^i$. We denote the unit normal vector on such a face F in the direction from E^- to E^+ with $\mathbf{n}_F$. Here and in the following, the orientation of the face is arbitrary but has to be kept fixed. Functions v_h from broken polynomial spaces are two-valued on internal faces $F \in \mathcal{F}_h^i$, and we use v_h^- for the value from E^- and v_h^+ for that from E^+. For points on the interface, $\mathbf{x} \in F$, we use these two function values to define the jump of v_h across F,

$$[\![v_h]\!](\mathbf{x}) \coloneqq v_h^-(\mathbf{x}) - v_h^+(\mathbf{x}), \tag{5.31}$$

and the weighted average of v_h,

$$\{v_h\}_\omega(\mathbf{x}) \coloneqq \omega^- v_h^-(\mathbf{x}) + \omega^+ v_h^+(\mathbf{x}), \tag{5.32}$$

where $\omega^- \in [0,1]$ and $\omega^+ \coloneqq 1 - \omega^-$. For boundary faces $F \in \mathcal{F}_h^b \coloneqq \mathcal{F}_h \setminus \mathcal{F}_h^i$ we use the unit normal vector pointing outside of the domain as $\mathbf{n}_F$ and set

$$[\![v_h]\!](\mathbf{x}) \coloneqq \{v_h\}_\omega(\mathbf{x}) \coloneqq v_h^-(\mathbf{x}), \tag{5.33}$$

where v_h^- is the function value from the single element that F belongs to. We also introduce the broken gradient $\nabla_h v_h$ for functions in $V_h^{(k)}$, which is defined elementwise by setting

$$\forall E \in \mathcal{E}_h \colon \left[\nabla_h v_h\right]|_E \coloneqq \nabla\left[v_h|_E\right]. \tag{5.34}$$

On the interfaces, the average $\{\nabla_h v_h\}_\omega$ and jump $[\![\nabla_h v_h]\!]$ of the broken gradient are defined through the averages and jumps of its components respectively, just as for other vector-valued discrete functions.

5.2.2 Diffusion Terms

The simplest form of discretization for diffusion terms $a_u^{\text{diff}}(\cdot,\cdot)$ as in equation (5.23) consists in inserting discrete functions u_h and ψ_h for u and ψ_u and replacing the gradient with the broken gradient:

$$a_h^{(0)}(u_h,\psi_h) \coloneqq \langle \nabla_h \psi_h, K\kappa(u_h)\nabla_h u_h\rangle_\Omega = \sum_{E\in\mathcal{E}_h} \langle \nabla_h \psi_h, K\kappa(u_h)\nabla_h u_h\rangle_E \tag{5.35}$$

However, $a_h^{(0)}(\cdot,\cdot)$ does not produce a consistent discretization. Following [21], it is modified by adding a consistency term, which leads to

$$a_h^{(1)}(u_h,\psi_h) \coloneqq a_h^{(0)}(u_h,\psi_h) - \sum_{F\in\mathcal{F}_h^i} \left\langle \kappa(u_h^{\text{upw}}) \left\{K\nabla_h u_h\right\}_\omega \cdot \mathbf{n}_F, [\![\psi_h]\!]\right\rangle_F, \tag{5.36}$$

where the weights for the average are given by

$$\omega^\pm \coloneqq \left[K(E^+) + K(E^-)\right]^{-1} K(E^\pm) \tag{5.37}$$

and the upwind value u_h^{upw} on a face F is

$$u_h^{\text{upw}} \coloneqq \begin{cases} u_h^- & \text{if } \{K\nabla_h u_h\}_\omega \cdot \mathbf{n}_F > 0 \\ u_h^+ & \text{else.} \end{cases} \tag{5.38}$$

Here and in the following we ignore contributions of the boundary faces $F \in \mathcal{F}_h^b$, since these will be discussed in section 5.2.4.

Adding an additional term reestablishes the symmetry of the expression, which results in

$$a_h^{\text{cs}}(u_h,\psi_h) \coloneqq a_h^{(1)}(u_h,\psi_h) - \sum_{F\in\mathcal{F}_h^i} \left\langle [\![u_h]\!], \kappa(u_h^{\text{upw}}) \left\{K\nabla_h \psi_h\right\}_\omega \cdot \mathbf{n}_F\right\rangle_F. \tag{5.39}$$

Finally, discrete coercivity on $V_h^{(k)}$ is required, which motivates the addition of a stability term and leads to

$$a_h^{\text{sip}}(u_h,\psi_h) \coloneqq a_h^{\text{cs}}(u_h,\psi_h) + \sum_{F\in\mathcal{F}_h^i} \eta \frac{\gamma_{K,F}}{h_F} \kappa(u_h^{\text{upw}}) \langle [\![u_h]\!], [\![\psi_h]\!]\rangle_F, \tag{5.40}$$

where the local mesh width is defined as

$$h_F \coloneqq \frac{1}{2} |F|_{d-1}^{-1} \left[\left|E^+\right|_d + \left|E^-\right|_d\right] \tag{5.41}$$

and is identical to the global mesh width h for the structured grids we are using,

$$\gamma_{K,F} \coloneqq \omega^- K(E^+) + \omega^+ K(E^-) = 2\left[K(E^+) + K(E^-)\right]^{-1} K(E^+)K(E^-) \tag{5.42}$$

is the harmonic mean of the diffusion coefficient K on the interface used as a local weight for the stabilization, and η is a user-supplied penalty parameter.

The discretized diffusion term $a_h^{\text{sip}}(\cdot,\cdot)$ is based on the numerical flux

$$\mathbf{J}_{u_h}^{\text{diff}}(F) \coloneqq -\kappa(u_h^{\text{upw}}) \left[\{K\nabla_h u_h\}_\omega \cdot \mathbf{n}_F - \eta \frac{\gamma_{K,F}}{h_F} [\![u_h]\!] \right] \tag{5.43}$$

as a discrete approximation of the normal component of the diffusive flux

$$j_u^{\text{diff}} \coloneqq -K\kappa(u)\nabla u \tag{5.44}$$

on the face F, and with this flux the discretization takes the form

$$\begin{aligned} a_h^{\text{sip}}(u_h,\psi_h) = \sum_{E\in\mathcal{E}_h} \langle \nabla_h\psi_h, K\kappa(u_h)\nabla_h u_h\rangle_E + \sum_{F\in\mathcal{F}_h^i} \left\langle \mathbf{J}_{u_h}^{\text{diff}}(F), [\![\psi_h]\!] \right\rangle_F \\ - \sum_{F\in\mathcal{F}_h^i} \left\langle [\![u_h]\!], \kappa(u_h^{\text{upw}}) \{K\nabla_h\psi_h\}_\omega \cdot \mathbf{n}_F \right\rangle_F. \end{aligned} \tag{5.45}$$

5.2.3 Convection Terms

The convection terms $a_u^{\text{con}}(\cdot,\cdot)$ as in equation (5.24) and equation (5.25) can be treated in a similar fashion. Starting point is again a simple discretization that inserts discrete functions u_h and ψ_h for u and ψ_u and replaces the gradient with the broken gradient. For the conservative formulation, equation (5.24), the steps are identical to those for diffusion terms. The result is

$$\begin{aligned} a_h^{\text{upw}}(u_h,\psi_h) \coloneqq \sum_{E\in\mathcal{E}_h} \langle \nabla_h\psi_h, u_h j_{\text{con}}\rangle_E + \sum_{F\in\mathcal{F}_h^i} \left\langle \mathbf{J}_{u_h}^{\text{con}}(F), [\![\psi_h]\!] \right\rangle_F \\ - \sum_{F\in\mathcal{F}_h^i} \left\langle [\![u_h]\!], \{\psi_h\}_\omega j_{\text{con}} \cdot \mathbf{n}_F \right\rangle_F, \end{aligned} \tag{5.46}$$

where

$$\mathbf{J}_{u_h}^{\text{con}}(F) \coloneqq \{u_h\}_\omega j_{\text{con}} \cdot \mathbf{n}_F + \frac{\eta}{2} |j_{\text{con}} \cdot \mathbf{n}_F| [\![u_h]\!] \tag{5.47}$$

is again a discrete approximation of the normal component of the convective flux

$$j_u^{\text{con}} \coloneqq u j_{\text{con}} \tag{5.48}$$

on the face F, and $\omega^- \coloneqq \omega^+ \coloneqq 1/2$. In principle, the penalty parameter η could be user-supplied again, but choosing $\eta \coloneqq 1$ leads to the usual upwind fluxes of Finite Volume schemes [21].

The discretization of the non-conservative formulation, equation (5.25), starts with

$$a_h^{(0)}(u_h,\psi_h) \coloneqq \langle \psi_h, \nabla_h u_h \cdot j_{\text{con}}\rangle_\Omega = \sum_{E\in\mathcal{E}_h} \langle \psi_h, \nabla_h u_h \cdot j_{\text{con}}\rangle_E \tag{5.49}$$

and adds a consistency term in analogy to equation (5.36), which yields

$$a_h^{\mathrm{cf}}(u_h, \psi_h) \coloneqq a_h^{(0)}(u_h, \psi_h) - \sum_{F \in \mathcal{F}_h^i} \langle [\![u_h]\!], \{\psi_h\}_\omega \, j_{\mathrm{con}} \cdot \mathbf{n}_F \rangle_F \tag{5.50}$$

with $\omega^- = \omega^+ = 1/2$ as above. Then a stabilization term is added as before, which leads to

$$\begin{aligned} a_h^{\mathrm{upw}}(u_h, \psi_h) &\coloneqq a_h^{\mathrm{cf}}(u_h, \psi_h) - \sum_{F \in \mathcal{F}_h^i} \langle [\![u_h]\!], \{\psi_h\}_\omega \, j_{\mathrm{con}} \cdot \mathbf{n}_F \rangle_F && (5.51) \\ &= \sum_{E \in \mathcal{E}_h} \langle \psi_h, \nabla_h u_h \cdot j_{\mathrm{con}} \rangle_E + \sum_{F \in \mathcal{F}_h^i} \left\langle \mathbf{J}_{u_h}^{\mathrm{con}}(F), [\![\psi_h]\!] \right\rangle_F && (5.52) \\ &\quad - \sum_{F \in \mathcal{F}_h^i} \langle [\![u_h]\!], \{\psi_h\}_\omega \, j_{\mathrm{con}} \cdot \mathbf{n}_F \rangle_F, \end{aligned}$$

with $\mathbf{J}_{u_h}^{\mathrm{con}}(F)$, η and the weights defined as for the conservative formulation. Note that the two discretizations only differ in the volume terms of the elements.

5.2.4 Remaining Terms and Boundary Conditions

The reaction terms and source terms are simple to discretize, since they are local processes and therefore interactions between different elements don't have to be considered. Instead, the integral over the domain Ω can be replaced with a sum over the integrals of the individual elements, which are defined for the discrete functions u_h and ψ_h. The discretized reaction terms have the form

$$a_h^{\mathrm{rea}}(u_h, \psi_h) = \sum_{E \in \mathcal{E}_h} \langle \psi_h, r_u u_h \rangle_E, \tag{5.53}$$

and the discretized source terms are of the form

$$b_h(\psi_h) = \sum_{E \in \mathcal{E}_h} \langle \psi_h, q_u \rangle_E. \tag{5.54}$$

Noting that the discretizations consist of volume terms evaluated on the individual elements $E \in \mathcal{E}_h$ and flux terms based on the numerical flux $\mathbf{J}_{u_h}$ evaluated on the interior faces $F \in \mathcal{F}_h^i$, boundary conditions can be incorporated by specifying $\mathbf{J}_{u_h}$ on the boundary faces $F \in \mathcal{F}_h^b$. Setting

$$\mathbf{J}_{u_h}(F) \coloneqq b_u^N \text{ for } F \subset \Gamma_u^N \tag{5.55}$$

defines the flux on the Neumann part of the boundary, while the flux on the Dirichlet part is given by the equations (5.43) and (5.47) with the jump $[\![u_h]\!]$ and average $\{u_h\}_\omega$ replaced by

$$[\![u]\!]_D \coloneqq u_h^- - b_u^D \tag{5.56}$$

respectively

$$\{u\}_{\omega,D} \coloneqq \omega^- u_h^- + \omega^+ b_u^D. \tag{5.57}$$

In principle, these two functions could be used as the definition of the jump and average of u_h on the boundary, replacing equation (5.33). This would simplify the notation and the incorporation of the boundary contributions into the discrete forms of the previous sections. However, it would also make the definition dependent upon the concrete problem at hand and the specific choice of boundary values. Note that the accompanying terms that were added to achieve consistency and symmetry of the discretization are also required for the boundary face contributions.

5.2.5 Slope Limiter and Flux Reconstruction

After discretizing the PDEs in space and time, the resulting finite-dimensional systems may be solved to get an approximate solution of the forward respectively adjoint problems. If the PDE is linear, its discretization is linear as well and may be formulated as a linear equation system for the unknowns. If the PDE is nonlinear, then Newton's method can be employed to arrive at a sequence of linear equation systems. As mentioned by *Cockburn and Shu* [19], the RKDG scheme needs to be stabilized when using an explicit timestepping scheme, and this can be achieved by applying a slope limiter to the individual stages. We refer to the work of the original authors for the discussion of slope limiters and the specifics of their implementation.

Combining the previous sections with a specific choice of trial space results in a discontinuous Galerkin method:

Discretization 7 (Discontinuous Galerkin)
The discontinuous Galerkin method uses the trial space $V_h^{(k)}(\mathcal{E}_h)$ as defined in equation (5.30)*, where $k \geq 1$ is the polynomial degree on the elements of $\mathcal{E}_h$. We will always use $k = 1$ in the applications of chapter 6. The space $V_h^{(k)}$ is used for both the discrete solution u_h and the test functions ψ_h in the relevant discretized terms, e.g. the discretized diffusion term from equation* (5.45) *or one of the discretized convection terms from equations* (5.46) *and* (5.51)*. Reaction terms as in equation* (5.53) *and source terms as in equation* (5.54) *are also included, if applicable. Boundary conditions are incorporated by choosing the correct fluxes on the boundary faces, as discussed in the previous section. The user-supplied penalty parameter η should be chosen as small as possible for diffusion terms, just large enough to guarantee the stability of the discretization, while we use the upwind flux for the convection terms. As mentioned above, a slope limiter is required to further stabilize the method if it is employed in an explicit timestepping scheme.*

Choosing $k = 0$ leads to a Cell-Centered Finite Volume scheme (CCFV). Most contributions of the discretized terms vanish, since the broken gradients $\nabla_h u_h$ and $\nabla_h \psi_h$ are by definition zero on each of the elements. The only remaining contributions are

based on jumps across faces or cell averages. This leads to a conceptually simpler method:

Discretization 8 (Cell-Centered Finite Volume)
The Cell-Centered Finite Volume scheme uses the trial space $V_h^{(0)}(\mathcal{E}_h)$, i.e. piecewise constant functions. This method uses the same discretized terms as the discontinuous Galerkin method, discretization 7, but most contributions vanish due to the broken gradients being zero. The numerical flux of the discontinuous Galerkin method becomes dominated by the broken gradients for small mesh widths h, which allows choosing the penalty parameter η to stabilize the method. This isn't possible in the case of Cell-Centered Finite Volume, since the numerical flux on the faces consists solely of jump terms. For this reason we set $\eta = 1$ for both the diffusion and the convection terms to arrive at a consistent formulation. This method is diffusive enough to be stable without additional penalty terms or slope limiters.

Up to now, the numerical flux $\mathbf{J}_{u_h}$ is only defined on faces, and only in the normal direction of the faces. This is problematic when the flux has to be evaluated at other locations, e.g. the transport equation depending on values of j_{θ_w} inside of elements, compare equations (5.46) and (5.51), or sensitivity integrals like that in equation (4.111) requiring gradients and fluxes on the whole domain. We therefore need an extension of $\mathbf{J}_{u_h}$ onto the domain Ω that is consistent with the numerical flux across the faces. The flux reconstruction should produce a global function

$$\mathbf{J}_{u_h} \in H\left(\mathrm{div}; \Omega\right) \coloneqq \left\{ \mathbf{J} \in \left[L^2\left(\Omega\right)\right]^d \,\middle|\, \nabla \cdot \mathbf{J} \in L^2\left(\Omega\right) \right\}, \tag{5.58}$$

which then can be used in integrals on the whole domain Ω or individual elements $E \in \mathcal{E}_h$.

Such a reconstruction can be accomplished using the notion of discrete gradient, as described for general meshes by *Di Pietro and Ern* [21]. On the structured grids we are using this construction can be simplified and carried out dimension by dimension. For each element $E \in \mathcal{E}_h$ and each of the dimensions $1 \leq i \leq d$, we have two faces F_1 and F_2 and corresponding jumps

$$[\![u_h]\!]\left(F_j\right) \in L^2\left(F_j\right), j \in \{1, 2\}. \tag{5.59}$$

These jumps can be extended to a function

$$\left([\![u_h]\!]\left(E\right)\right)_i \in L^2\left(E\right) \tag{5.60}$$

through linear interpolation between the two jumps $[\![u_h]\!]\left(F_1\right)$ and $[\![u_h]\!]\left(F_2\right)$ on the faces. This interpolation has to take the orientation of the jumps into account, i.e. if one of the normal vectors $\mathbf{n}_{F_j}, j \in \{1, 2\}$, doesn't point in the same direction as the dimension i, then the sign of $[\![u_h]\!]$ on that face has to be switched. This provides the components of a vector-valued function

$$[\![u_h]\!]\left(E\right) \in \left[L^2\left(E\right)\right]^d, \tag{5.61}$$

which defines a lifting of the jumps onto the element E. This function can be combined with the broken gradient $\nabla_h u_h$ to define a discrete gradient

$$\mathbf{G}_{u_h}(E) \coloneqq \nabla_h u_h - [\![u_h]\!](E) \tag{5.62}$$

on each element $E \in \mathcal{E}_h$.

The same steps can be applied to functions of the form $\eta\gamma_h [\![u_h]\!]$, where η is the penalty parameter and γ_h a weight function appearing in the numerical flux definition, compare equations (5.43) and (5.47). The same linear interpolation as above lifts $\eta\gamma_h [\![u_h]\!]$ onto the element E, leading to a function

$$\eta\gamma_h [\![u_h]\!](E) \in \left[L^2(E)\right]^d. \tag{5.63}$$

Note that this simple extension introduces an error, as the original construction is a local projection that takes the spatially varying parameters into account [21]. However, this approximation is accurate enough for our intents and purposes. The numerical flux $\mathbf{J}_{u_h}$ can then be defined on E by leaving out the weighted average of the definition for faces $F \in \mathcal{F}_h^i$ and using the lifted version of the jumps, e.g.

$$\mathbf{J}_{u_h}^{\text{diff}}(E) \coloneqq -\kappa(u_h)\left[K\nabla_h u_h - \eta\frac{\gamma_{K,F}}{h_F}[\![u_h]\!](E)\right] \tag{5.64}$$

for the diffusive flux of equation (5.43).

However, these versions of the discrete gradient $\mathbf{G}_{u_h}$ and the numerical flux $\mathbf{J}_{u_h}$ aren't in $H(\text{div};\Omega)$, since the normal component of the broken gradient $\nabla_h u_h$ isn't continuous across faces. We therefore project the two functions onto the $H(\text{div};\Omega)$-conforming space

$$W_h^{(k)}(\mathcal{E}_h) \coloneqq \left\{ v \in H(\text{div};\Omega) \,\middle|\, \forall E \in \mathcal{E}_h \colon v|_E \in RTN_d^k \right\}, \tag{5.65}$$

where

$$RTN_d^k \coloneqq \left[Q_d^k\right]^d + \mathbf{x}Q_d^k \tag{5.66}$$

is the Raviart-Thomas-Nédélec space of degree k. This space can also be written as

$$RTN_d^k = \begin{cases} Q_d^{k+1,k} \times Q_d^{k,k+1} & \text{for } d = 2 \\ Q_d^{k+1,k,k} \times Q_d^{k,k+1,k} \times Q_d^{k,k,k+1} & \text{for } d = 3, \end{cases} \tag{5.67}$$

where $Q_d^{l,m}$ and $Q_d^{l,m,n}$ are generalizations of Q_d^k with l, m and n being the maximum degree in each of the two or three dimensions. The projection onto $W_h^{(k)}(\mathcal{E}_h)$ provides a flux reconstruction for the discontinuous Galerkin method, discretization 7, with the same choice for k:

Discretization 9 (Higher Order Flux Reconstruction)
The function $\mathbf{J}_{u_h}^{proj} \in W_h^{(k)}(\mathcal{E}_h), k \geq 1$, *is a* $H\,(div;\Omega)$*-conforming reconstruction of the flux of* u_h *if*

$$\forall\, F \in \mathcal{F}_h, \psi_F \in Q_{d-1}^k \colon \left\langle \left[\mathbf{J}_{u_h}^{proj} - \mathbf{J}_{u_h}\right] \cdot \mathbf{n}_F, \psi_F \right\rangle_F = 0 \tag{5.68}$$

using the definition of $\mathbf{J}_{u_h}$ *for faces* $F \in \mathcal{F}_h$ *and*

$$\forall\, E \in \mathcal{E}_h, \psi_E \in W^\dagger \colon \left\langle \mathbf{J}_{u_h}^{proj} - \mathbf{J}_{u_h}, \psi_E \right\rangle_E = 0 \tag{5.69}$$

using the extension of $\mathbf{J}_{u_h}$ *onto elements, where*

$$W^\dagger := \begin{cases} Q_d^{k-1,k} \times Q_d^{k,k-1} & \text{for } d = 2 \\ Q_d^{k-1,k,k} \times Q_d^{k,k-1,k} \times Q_d^{k,k,k-1} & \text{for } d = 3 \end{cases} \tag{5.70}$$

is the local test space for the flux on elements $E \in \mathcal{E}_h$.

This flux reconstruction can also be used for a $H\,(\mathrm{div};\Omega)$-conforming reconstruction $\mathbf{G}_{u_h}^{\mathrm{proj}}$ of the gradient of u_h by replacing the numerical flux $\mathbf{J}_{u_h}$ with the discrete gradient $\mathbf{G}_{u_h}$ from equation (5.62). For the Cell-Centered Finite Volume scheme, discretization 8, the flux reconstruction is simpler:

Discretization 10 (First Order Flux Reconstruction)
The function $\mathbf{J}_{u_h}^{proj} \in W_h^{(0)}(\mathcal{E}_h)$ *is a* $H\,(div;\Omega)$*-conforming reconstruction of the flux of* u_h *if*

$$\forall\, F \in \mathcal{F}_h, \psi_F \in Q_{d-1}^0 \colon \left\langle \left[\mathbf{J}_{u_h}^{proj} - \mathbf{J}_{u_h}\right] \cdot \mathbf{n}_F, \psi_F \right\rangle_F = 0 \tag{5.71}$$

using the definition of $\mathbf{J}_{u_h}$ *for faces* $F \in h$. *The space* Q_{d-1}^0 *is the space of constant functions, so this is equivalent to*

$$\forall\, F \in \mathcal{F}_h \colon \int_F \left[\mathbf{J}_{u_h}^{proj} - \mathbf{J}_{u_h}\right] \cdot \mathbf{n}_F = 0. \tag{5.72}$$

In this case, the function $\mathbf{J}_{u_h}^{proj}$ *is completely determined by its face integrals and has no internal degrees of freedom on the elements. Therefore the flux reconstruction can be computed without any intermediate liftings.*

5.3 Libraries and Software Packages

The numerical methods of the previous sections have been implemented in DUNE, the Distributed and Unified Numerics Environment, a modular toolbox for the grid-based solution of PDEs [7, 6, 5]. DUNE uses abstract interfaces and generic programming techniques, which makes the implementation of very general and flexible methods comparatively easy. The discretization module PDELab builds on the interfaces

of DUNE to provide representations of higher-level mathematical constructs, like function spaces and operators [8]. This makes it possible to produce an implementation that more closely resembles the original mathematical notation, and enables the reuse of standard components. Software based on DUNE and PDELab has access to ISTL, the Iterative Solver Template Library, which contains several parallel solvers based on domain decomposition and a parallel Algebraic Multigrid (AMG) solver [10]. Furthermore, the intricacies of high-performance parallel computing are handled transparently by DUNE and PDELab, and therefore the implementation used in chapter 6 is automatically suited for parallel computation.

While the implementation is for the most part based on DUNE, a number of external libraries are used. This includes FFTW, the Fastest Fourier Transform in the West, a subroutine library that provides functions for parallel computation of the discrete Fourier transform [30], and HDF5, a library for efficient parallel I/O [76], which is used by the implementation for reading and writing parameter fields. All libraries and software frameworks that were used for this thesis are free software and freely available from their respective open-source projects. At the time of writing, a release of the implementation in the form of a DUNE module is planned.

6 Applications

In this chapter we consider several applications of the methods that were developed in chapter 2. In each of the test cases, a synthetic reference $\widehat{\mathbf{P}}$ is generated using algorithm 1 (**SG**). The forward problem 1 is solved using this reference parameter tuple, with the result being a set of observations $\widehat{\mathbf{Z}}$. To simulate the measurement process, noise $\epsilon \sim \mathcal{N}(\mathbf{Q}_{\mathbf{ZZ}}, \mathbf{0})$ is generated and added to $\widehat{\mathbf{Z}}$ to obtain synthetic observations

$$\mathbf{Z} \coloneqq \widehat{\mathbf{Z}} + \epsilon. \tag{6.1}$$

These observations are then used as input to define the inverse problem and solve it for $\mathbf{P}_{\text{map}}$.

Due to the large number of possible experimental setups, boundary conditions, model assumptions and solver choices, it isn't possible to examine every single combination in detail. For this reason, we restrict ourselves to the following subset:

- While the methods have also been tested for other covariance structures, the exponential covariance function, equation (2.12), is used in each of the examples. The covariance structure is in most cases chosen to be isotropic, and all parameter fields use the same correlation length if several are considered at once. The only trend parameter that is considered is the mean of the parameter field.

- All examples use the caching prior preconditioned CG method, algorithm 6 ($\mathbf{PCG}_c$), to compute the estimate $\mathbf{P}_{\text{map}}$, since this method has the lowest memory requirements and lowest computational cost per iteration among the methods that are suitable for high-resolution parameter fields. Occasionally the randomized Gauss-Newton scheme, algorithm 16 ($\mathbf{GN}_r$), is used for comparison purposes.

- We prioritize simplicity of the setup and reproducibility of the numerical results. For this reason all boundary conditions are constant and explicitly known, and the domain is rectangular so that the spectral methods of section 2.1.2 are applicable without interpolation or transformation. All of the test cases are discretized using the discontinuous Galerkin method, discretization 7, with $k = 1$ and second order explicit and implicit time discretization if applicable.

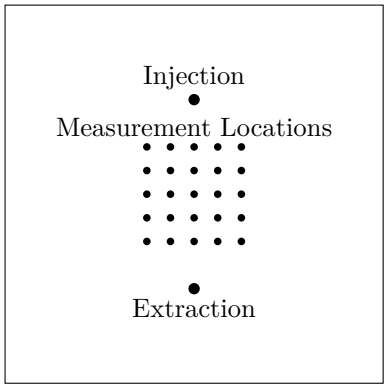

Figure 6.1: The experimental setup used in sections 6.1.1, 6.1.2 and 6.1.3. An array of measurement locations is installed between a pair of wells. Water is injected into one of the wells and extracted from the other, which induces flow patterns in the aquifer and a potential gradient at the measurement locations. The hydraulic potential is assumed to be constant on the boundary of the domain. In the three-dimensional test case this is a view from above, with no-flow boundary conditions imposed at the top and bottom of the domain.

6.1 Dipole Experiments

As a first application we consider synthetic dipole experiments. Water is injected into a confined aquifer at one location and extracted at another, see figure 6.1. This induces a dipole in the hydraulic potential, compare figure 4.1, and the resulting changes in potential can be monitored at locations between the injection and extraction wells. These observations then allow estimation of the parameters that govern the groundwater flow. We begin by considering stationary two-dimensional flow, followed by extensions to three-dimensional and transient scenarios.

6.1.1 Inversion of Stationary Flow in 2D

We consider a two-dimensional square domain Ω of size $100\,\mathrm{m} \times 100\,\mathrm{m}$. The injection and extraction well are located at $(50, 75)$ and $(50, 25)$ respectively. The measurement locations form an equidistant square grid with the lower left corner at $(37.5, 37.5)$ and the upper right corner at $(62.5, 62.5)$, compare figure 6.1. The placement of individual observations depends on the total number of measurements n_ϕ, which varies between 5×5 and 30×30. Assumptions about the distribution of the log-conductivity Y and the measurement error of the head measurements can be found in table 6.1. For simplicity we assume that the head ϕ is constant on the boundary Γ. The exact value is irrelevant, since the groundwater flow equation, model 3, is only concerned with relative changes of potential. We further choose a constant injection rate respectively

Property	Description	Value
Log-conductivity Y:		
	Covariance model	exponential
$(\lambda_x, \lambda_y, \lambda_z)$	Correlation lengths	$(5\,\mathrm{m}, 5\,\mathrm{m}, 1\,\mathrm{m})$
σ^2	Prior variance	1
$\boldsymbol{\beta}^*$	Prior mean	-5.8
$\sigma^2_{\boldsymbol{\beta}}$	Uncertainty of prior mean	0.1
Error of head observations ϕ:		
	Covariance model	uncorrelated
σ^2	Prior variance	$1 \times 10^{-4}\,\mathrm{m}^2$

Table 6.1: Assumptions about the parameter distribution and measurement errors for the two-dimensional and three-dimensional stationary dipole experiments. The correlation length λ_z describes vertical correlation and is only relevant in the three-dimensional case. The logarithm $Y = \ln(K)$ has to be interpreted with K measured in ms^{-1}.

extraction rate of $q_{\theta_w} := \pm 7.5 \times 10^{-3}\,\mathrm{s}^{-1}$, in each case evenly distributed over the area of the well.

A synthetic reference parameter field $\widehat{\mathbf{P}}$ representing the log-conductivity Y is generated, see the upper row of figure 6.2, and the spatial distribution of the head ϕ is computed using model 3 together with the given boundary conditions and source terms. We use a structured equidistant grid of size $n_\Omega = 256 \times 256 = 6.55 \times 10^4$ and $V_h^{(1)}(\mathcal{E}_h)$ as ansatz space, i.e. locally bilinear discontinuous finite elements. The resulting discretized state space for ϕ has a dimension of $6.55 \times 10^4 \cdot 4 = 2.62 \times 10^5$. Synthetic measurements are taken at $m \times m$ locations, where $m \in \{5, 6, 7, 8, 9, 10, 15, 20, 25, 30\}$, and noise is added to these measurements to simulate measurement error. This produces ten sets of data with varying size, from 25 observations for $m = 5$ to 900 observations for $m = 30$.

Each of the generated data sets can be used as input for inversion. We consider the following three choices for the optimization algorithm:

- The caching prior preconditioned CG method, algorithm 6 ($\mathbf{PCG}_c$), with a reduction of the norm of the directional derivative by a factor of 10^4 as stopping criterion
- Again the caching PCG method, algorithm 6 ($\mathbf{PCG}_c$), but with a reduction by a factor of 10^5 as stopping criterion
- The randomized Gauss-Newton method, algorithm 16 ($\mathbf{GN}_r$), using the caching variant mentioned in section 3.3.2 with a reduction by a factor of 10^4 as stopping criterion

This produces 30 different inversion results, which can be compared regarding the number of simulation runs that are required, the time it takes to compute the estimate, and the quality of the result.

The lower row of figure 6.2 shows two of the inversion results, the estimates for $n_\phi = 25$ and $n_\phi = 900$. Both estimates were obtained using the caching PCG method with tolerance 10^{-5} for the reduction of the directional derivative. The additional observations clearly increase the resolution of the estimate, as significantly more details appear in the estimate that is based on the higher number of measurements. This is confirmed by the uncertainty quantification, which produces the estimates for the posterior variance shown in figure 6.3. The posterior variance of the parameters in the direct vicinity of the measurement locations is significantly lower for the result based on the larger number of observations.

Figure 6.4 shows the effort that is needed to produce the estimates. The left image contains the total number of simulation runs that are required, while the right image shows the computational cost in terms of seconds of computing time on a parallel machine (Intel Xeon, 64 cores, 2800 MHz). Comparing the two different strategies for algorithm 6 ($\mathbf{PCG}_c$), reducing the norm of the derivative by an additional order of magnitude approximately triples the computational costs.

The randomized Gauss-Newton scheme, algorithm 16 ($\mathbf{GN}_r$), isn't suited for the considered setup. The number of eigenvalues that have to be recovered ranges from $r = n_\phi$ for small values of n_ϕ to $r = \frac{2}{3}n_\phi$ for the largest values. This means that both algorithms described in section 2.5 are more expensive than directly computing the columns of $\mathbf{H_{ZP}}$, and consequently algorithm 15 ($\mathbf{GN}_{\mathrm{CE}}$) would be more efficient.

For small values of n_ϕ the variants of the Gauss-Newton method are faster than the versions of the PCG scheme, but for values around $n_\phi = 100$ the preconditioned CG method starts to be more efficient. The computational cost of the Gauss-Newton method is linear in the number of observations, while that of the PCG scheme grows significantly slower and is almost constant ignoring the values at $n_\phi = 25$ and $n_\phi = 36$. As a result, the preconditioned Conjugate Gradients method is an order of magnitude faster than the Gauss-Newton scheme for large values of n_ϕ.

As discussed by *Nowak* [61], twice the minimum $L(\mathbf{P}_{\mathrm{map}})$ of the objective function L should follow the χ^2 distribution with n_ϕ degrees of freedom. This distribution has an expected value of n_ϕ and a variance of $2n_\phi$. Therefore

$$L(\mathbf{P}_{\mathrm{map}}) = \frac{1}{2}\left[n_\phi \pm [2n_\phi]^{1/2}\right] \tag{6.2}$$

is an estimate of $L(\mathbf{P}_{\mathrm{map}})$ in the form of its expected value and standard confidence interval. Figure 6.5 shows the expected value of the minimum as a solid line, the confidence interval bounded by two dashed lines, and the final value of each of the 30 inversions. The PCG method with tolerance 10^{-4} deviates from the confidence interval for large values of n_ϕ. The Gauss-Newton method and the PCG method

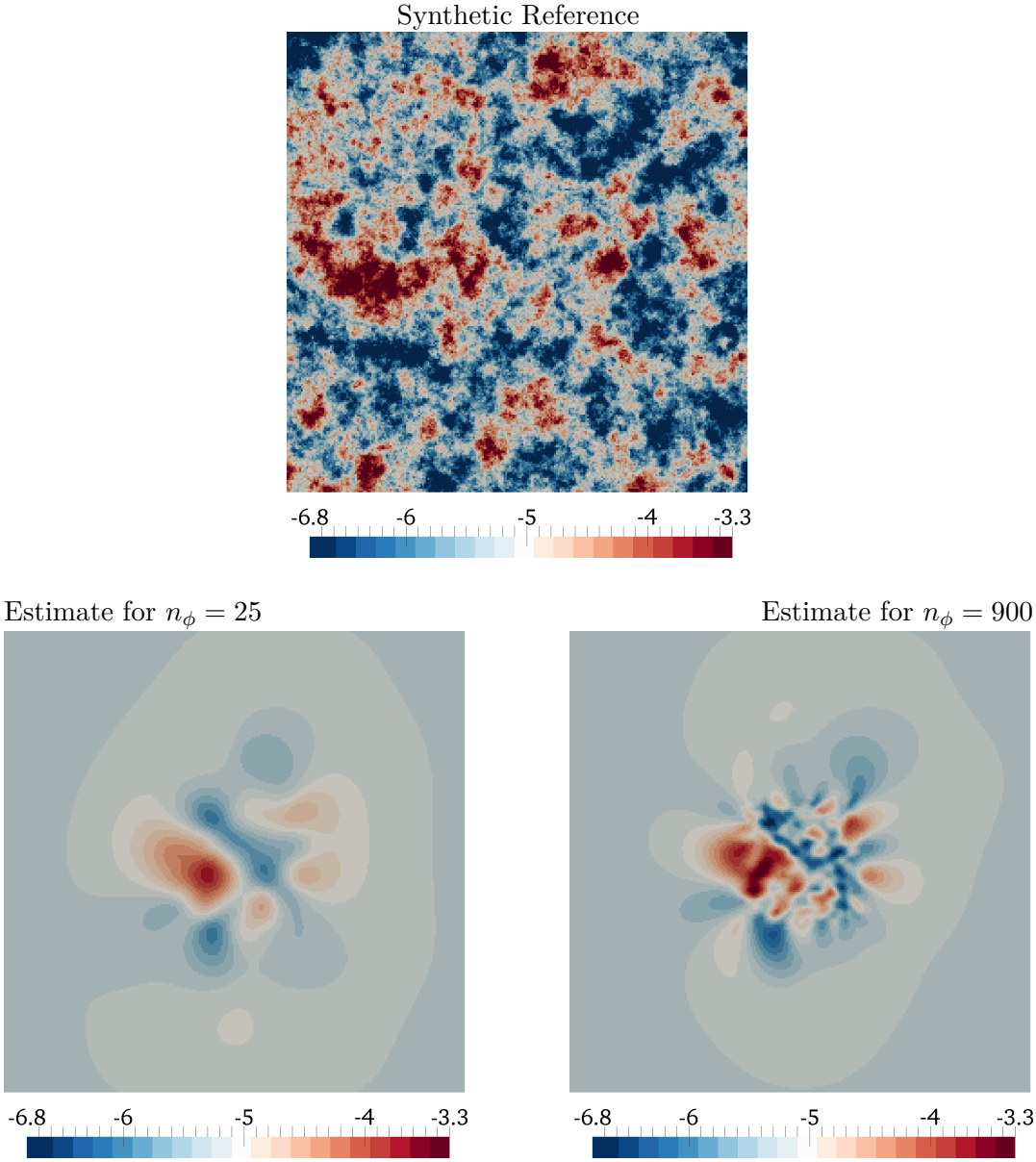

Figure 6.2: *Upper row:* Synthetic reference parameter field $\widehat{\mathbf{P}}$ with exponential covariance structure, representing the log-conductivity of the aquifer, $n_\Omega = 256 \times 256$. *Lower left:* Estimate $\mathbf{P}_{\text{map}}$ for $n_\phi = 25$ head measurements. *Lower right:* Estimate $\mathbf{P}_{\text{map}}$ for $n_\phi = 900$ head measurements. Both estimates are able to reproduce the spatial mean of the reference field in the direct vicinity of the observations but provide little information about the conductivity further away. Including additional measurements significantly improves the quality of the result, while incurring only a moderate increase in effort needed for the inversion, compare figure 6.4.

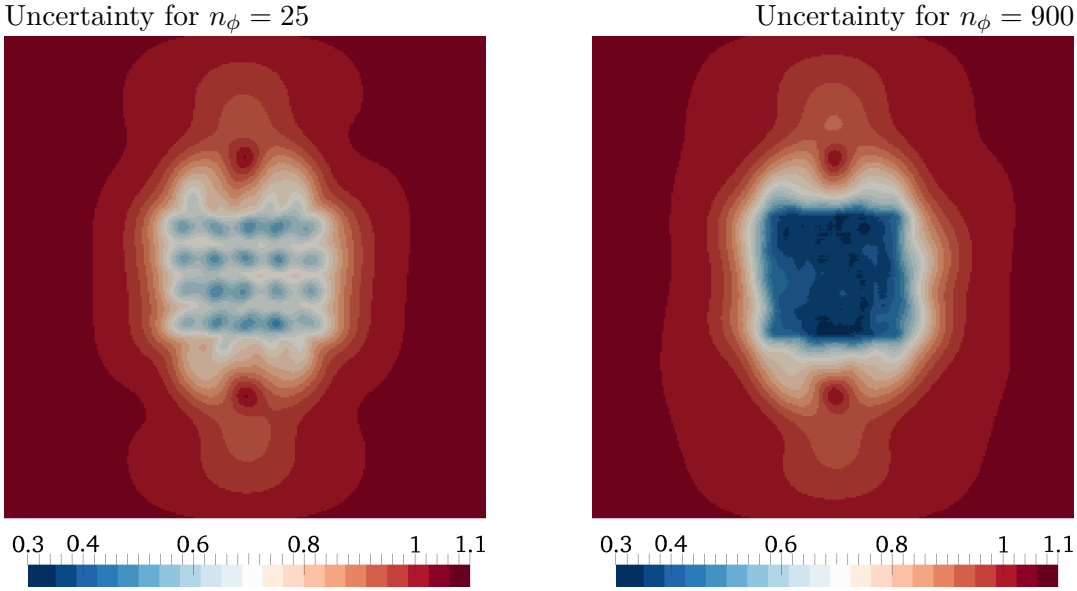

Figure 6.3: *Left:* Uncertainty of the estimate for $n_\phi = 25$ head measurements (lower left in figure 6.2). *Right:* Uncertainty of the estimate for $n_\phi = 900$ head measurements (lower right in figure 6.2). The values are the sum of local variance and trend variance. The posterior variance allows assessment of the reliability of the estimates, and a comparison confirms that the estimate using more observations is significantly more accurate.

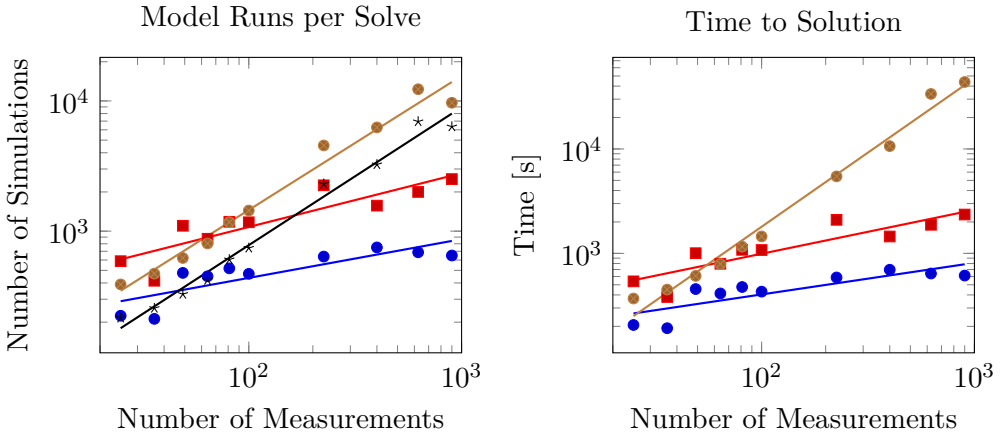

Figure 6.4: *Left:* Number of forward and adjoint model runs required by the prior preconditioned CG method ($\mathbf{PCG}_c$) with tolerance 10^{-4} for the reduction of the directional derivative (●), the prior preconditioned CG method with tolerance 10^{-5} (■) and the randomized Gauss-Newton scheme ($\mathbf{GN}_r$, ●), together with estimates for one of the classical Gauss-Newton methods ($\mathbf{GN}_{\mathrm{CE}}$, ⋆). *Right:* Time required for inversion. The Gauss-Newton methods are the best choice for small n_ϕ, but for large n_ϕ the prior preconditioned CG scheme is significantly more efficient.

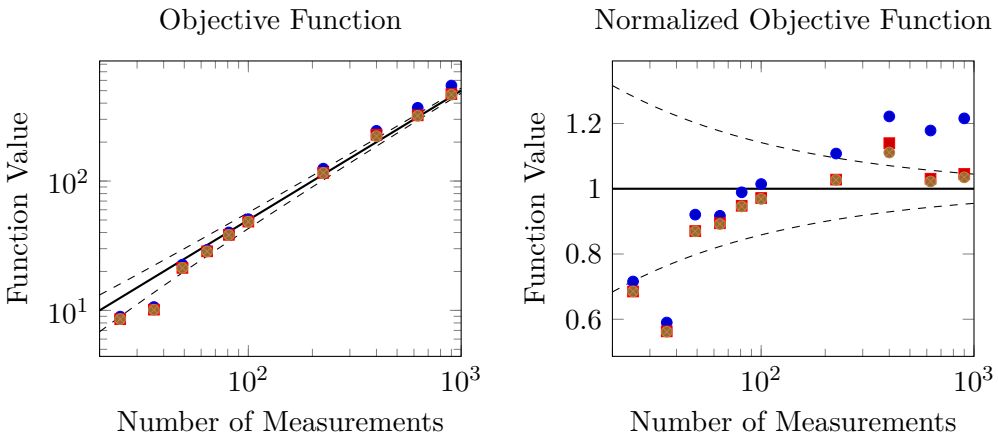

Figure 6.5: *Left:* Final values of the objective function L for the prior preconditioned CG method ($\mathbf{PCG}_c$) with tolerance 10^{-4} for the reduction of the directional derivative (●), the prior preconditioned CG method with tolerance 10^{-5} (■), and the randomized Gauss-Newton scheme ($\mathbf{GN}_r$, ●). The black line represents the expected value of the minimum of L, while the dashed lines mark the confidence interval. *Right:* Quotient of the final values and expected values as a more appropriate visualization for large values of n_ϕ. The CG method with tolerance 10^{-4} deviates from the confidence interval, while the other two schemes are in compliance.

with tolerance 10^{-5} are both in compliance. With respect to this measure the PCG method is as accurate as the Gauss-Newton method and significantly more efficient.

Comparing the final value of L to its expected value is cheap and a quick indicator of the quality of the inversion result, but the test is based on a single number. This means the test is able to assess the quality, but can't communicate additional information about underlying connections and reasons for poor results. The statistical tests derived in section 2.6.2 and section 2.6.3 are based on high-dimensional parameter errors and measurement residuals. As such, they produce several statistical indicators in the form of estimates of stochastic moments, and the normalized errors and residuals themselves may contain useful information in the form of spatial patterns.

Figure 6.6 contains the histograms of the normalized errors $\Delta_{\mathbf{P}}$ and normalized residuals $\Delta_{\mathbf{Z}}$ for the three largest data sets, $n_\phi = 400$, $n_\phi = 625$ and $n_\phi = 900$, computed with algorithm 8 ($\mathbf{SVD}_r$). The histograms of $\Delta_{\mathbf{P}}$ are virtually the same for each combination of the three data sets and the three optimization approaches. However, the histograms of $\Delta_{\mathbf{Z}}$ are highly sensitive. The histograms of the PCG method with tolerance 10^{-5} and the Gauss-Newton method generally have the correct mean value, spread and general shape, with one outlier in the case of the Gauss-Newton method. The histograms of the PCG method with tolerance 10^{-4} have the wrong mean, have

	Mean	Variance	Skewness	Kurtosis
$\widehat{\mathbf{P}}$	2.697	1.845×10^2	0.5978	2.571
$\mathbf{P}^*$	101.5	1.192×10^8	0.0503	2.031
$\mathbf{P}_{\text{rand}}$	2461	1.135×10^9	0.3934	2.187

Table 6.2: Example a posteriori statistics of the normalized residual $\Delta_{\mathbf{Z}}$ in the case of $n_\phi = 100$ for the reference parameter tuple $\widehat{\mathbf{P}}$, the prior mean $\mathbf{P}^*$ and a random sample $\mathbf{P}_{\text{rand}}$ drawn from the prior distribution. The reference $\widehat{\mathbf{P}}$ is itself an approximation of $\mathbf{P}_{\text{map}}$, with deviations of the state observations that are by construction in the order of the measurement errors. The values of the mean and variance of $\Delta_{\mathbf{Z}}$ for $\widehat{\mathbf{P}}$ can therefore be seen as the maximum values that may be tolerated for the inversion result.

a spread that is much too large and are skewed, which indicates that the scheme wasn't able to achieve full convergence. This is consistent with $L(\mathbf{P}_{\text{map}})$ being far outside of the confidence interval in the previous test.

These findings suggest that the normalized residual $\Delta_{\mathbf{Z}}$ is a better measure for goodness-of-fit than the normalized error $\Delta_{\mathbf{P}}$, as was already mentioned in remark 16. For this reason we concentrate on the evaluation of the moments of $\Delta_{\mathbf{Z}}$, which can be found in figure 6.7 and may be compared with the values of table 6.2 as reference. Two values for the variance are off the chart but can be deduced from figure 6.6. The detailed analysis confirms that the sample mean, variance and skewness deviate from their expected values in the case of the PCG scheme with tolerance 10^{-4} for larger values of n_ϕ. The sample moments for the PCG method with tolerance 10^{-5} and the Gauss-Newton scheme are in good agreement with the expected values, with the PCG method being slightly better in several instances. This confirms that the prior preconditioned Conjugate Gradients method can be as accurate as the well-established Gauss-Newton method at significantly lower computational cost and memory requirements.

6.1.2 Inversion of Stationary Flow in 3D

We extend the scenario to a full three-dimensional representation of a confined aquifer in the form of a rectangular domain Ω of size $50\,\text{m} \times 50\,\text{m} \times 5\,\text{m}$. The properties of the log-conductivity parameter field and the measurement errors are the same as in the two-dimensional case and can be found in table 6.1. The top and bottom of the domain are assumed to be impermeable, which leads to no-flow boundary conditions, while the remaining faces of the domain have a fixed potential as in the two-dimensional case. The injection well is represented by a straight line from $(25, 12.5, 1)$ to $(25, 12.5, 4)$, while the extraction well is a straight line from $(25, 37.5, 1)$

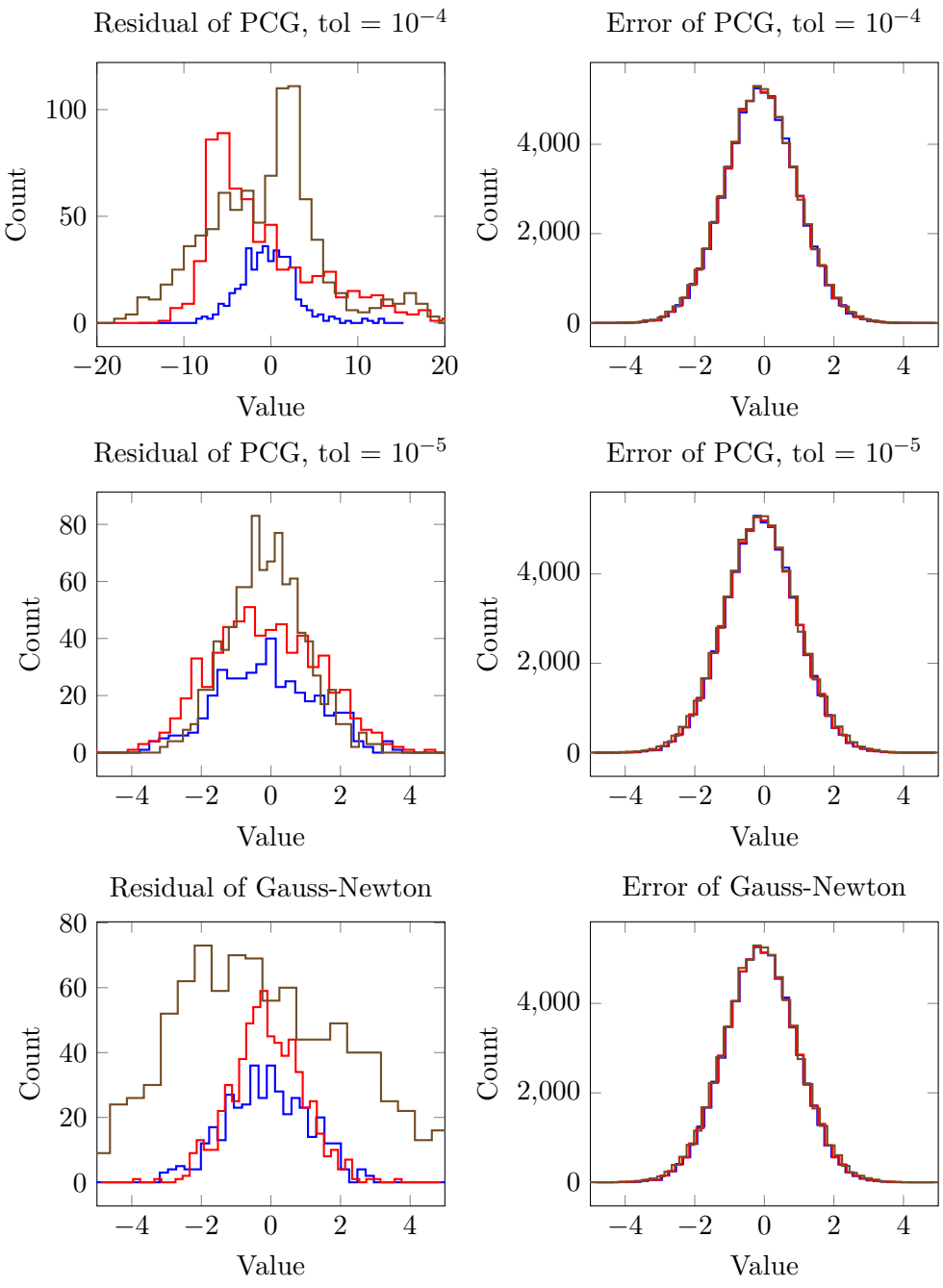

Figure 6.6: Histograms for the normalized residuals and errors for $n_\phi = 400$ (—●—), $n_\phi = 625$ (—■—) and $n_\phi = 900$ (—◆—), showing that the statistics of the residuals are a much more sensitive indicator for the quality of the estimate than those of the errors. Histograms for other values of n_ϕ look similar. Note the different scale for the normalized residuals of PCG with tolerance 10^{-4}.

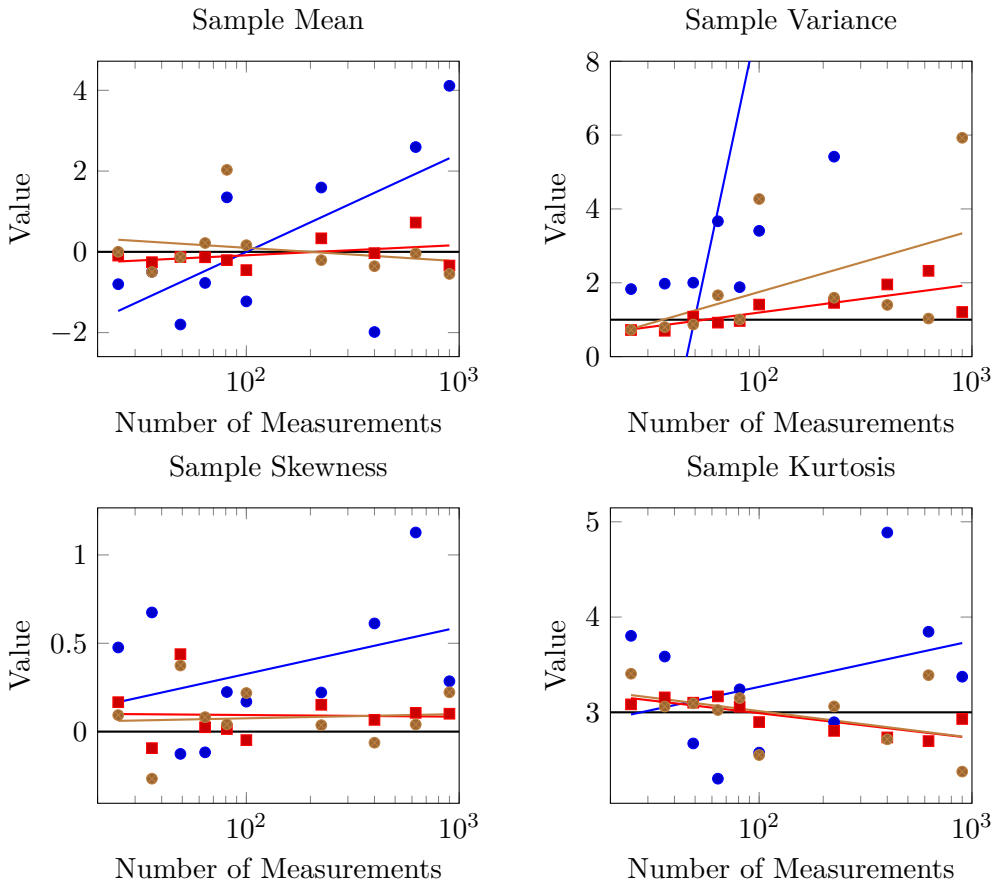

Figure 6.7: A posteriori statistics of the normalized residual $\Delta_{\mathbf{Z}}$ for the prior preconditioned CG method ($\mathbf{PCG}_c$) with tolerance 10^{-4} for the reduction of the directional derivative of L (●), the prior preconditioned CG method with tolerance 10^{-5} (■), and the randomized Gauss-Newton scheme ($\mathbf{GN}_r$, ●). The Gauss-Newton method is generally close to the expected values, indicating successful convergence and applicability of the linearization of the posterior distribution. A small number of outliers exists, suggesting an additional iteration of the algorithm would be required for full convergence in these cases. While the statistics of the CG method with tolerance 10^{-4} deviate from the expected values and indicate that further iterations are required, those of the CG method with tolerance 10^{-5} are as good as or better than those of the Gauss-Newton method.

	Model Runs	$L(\mathbf{P}_{\text{map}})$	Mean	Variance	Skewness	Kurtosis
$\mathbf{PCG}_c$	2614	101.5	−0.1945	0.9604	0.0294	2.992
$\mathbf{GN}_r$	2653	103.4	0.4032	5.5102	−0.1251	2.353

Table 6.3: Number of forward and adjoint model runs, final value of the objective function and first four sample moments for the normalized residual $\Delta_{\mathbf{Z}}$ for the three-dimensional test case solved with the PCG method with tolerance 10^{-5} and the randomized Gauss-Newton method. In the case of the PCG method all four moments are in good agreement with their expected values. The values for the Gauss-Newton scheme suggest that one or two additional iterations would be required for full convergence.

to $(25, 37.5, 4)$. The placement of the wells can be seen in figure 6.1 interpreted as a view from above.

The domain is discretized using a structured grid of size $n_\Omega = 128 \times 128 \times 16 = 2.62 \times 10^5$, equidistant in each of the dimensions. The ansatz space is again $V_h^{(1)}(E_h)$, which has a dimension of $2.62 \times 10^5 \cdot 8 = 2.10 \times 10^6$ in this case. The measurements of ϕ are located at the coordinates

$$\mathbf{x}_{i,j,k} \coloneqq \left(50 \cdot \left[\frac{3}{8} + i \cdot \frac{1}{16}\right], 50 \cdot \left[\frac{3}{8} + j \cdot \frac{1}{16}\right], 5 \cdot \left[\frac{1}{8} + k \cdot \frac{3}{4}\right]\right) \tag{6.3}$$

for $i, j \in \{0, 1, \ldots, 4\}$ and $k \in \{0, 1, \ldots, 8\}$, which forms a grid with mesh width 3.125 m in the horizontal and 0.468 75 m in the vertical direction. The measurement locations represent 25 observation wells of 9 measurements each, and their horizontal position is shown in figure 6.1. This setup results in an observation space of dimension $n_\phi = 225$.

A synthetic reference $\widehat{\mathbf{P}}$ is generated, see the upper row of figure 6.8, then synthetic measurements are obtained by simulating the stationary groundwater equation, model 3, and adding noise to the state observations. The resulting data is used as input for inversion, once with the PCG method with tolerance 10^{-5} and once with the randomized Gauss-Newton method.

The bottom row of figure 6.8 shows the inversion result $\mathbf{P}_{\text{map}}$ of the PCG scheme. The estimate is able to reproduce the coarse structure of the reference field, while smaller features are missing in the estimate due to the limited amount of data. Table 6.3 lists the number of model runs that were needed for the inversion, the final value of L and the a posteriori statistics of $\Delta_{\mathbf{Z}}$ for both algorithms. The caching prior preconditioned CG scheme achieved a significantly better result for approximately the same computational effort. This is consistent with the findings for the two-dimensional test case of the previous section.

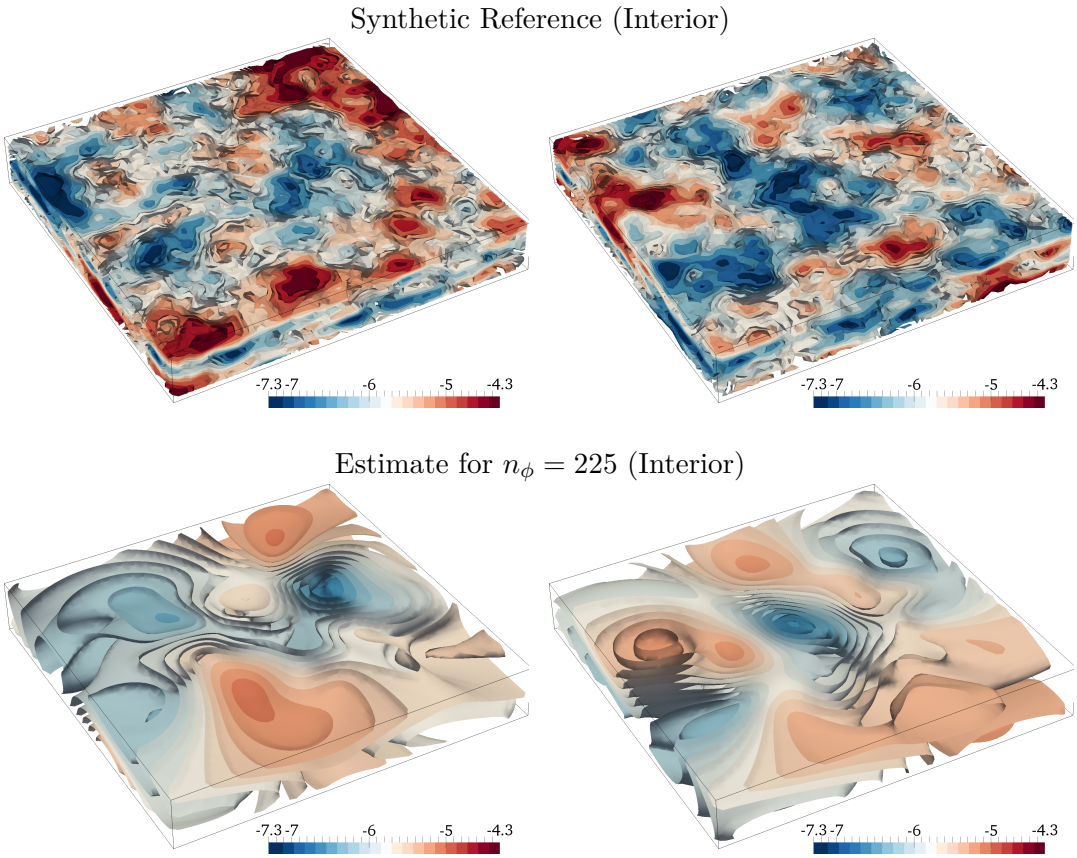

Figure 6.8: *Upper row:* Synthetic reference parameter field $\widehat{\mathbf{P}}$ with exponential covariance structure, representing the log-conductivity of a three-dimensional aquifer, $n_\Omega = 128 \times 128 \times 16$. Both images show the same domain, once from above and once from below. *Lower row:* Estimate $\mathbf{P}_{\text{map}}$ for $n_\phi = 225$ head measurements, same orientation. All images show the inner part of the domain where the parameters are sensitive, compare figure 6.2, with approximately 20% cut away in the vertical and 50% cut away in the other two dimensions. The estimate is able to reproduce the coarse features and general structure of the reference field, while smaller features can't be recovered from the data. This includes a small number of layers that are not represented in the estimate. *Not shown:* The posterior variance of the parameter field has the same structure as in the two-dimensional case, with low values around the measurement columns and intermediate values between the measurement locations.

Property	Description	Value
Log-conductivity Y:		
	Covariance model	exponential
λ	Correlation length	5 m
σ^2	Prior variance	1.6
$\boldsymbol{\beta}^*$	Prior mean	-5.8
$\sigma^2_{\boldsymbol{\beta}}$	Uncertainty of prior mean	0.1
Log-storativity Z_s:		
	Covariance model	exponential
λ	Correlation length	5 m
σ^2	Prior variance	1.6
$\boldsymbol{\beta}^*$	Prior mean	-9.2
$\sigma^2_{\boldsymbol{\beta}}$	Uncertainty of prior mean	0.1
Error of head observations ϕ:		
	Covariance model	uncorrelated
σ^2	Prior variance	$1 \times 10^{-4}\,\mathrm{m}^2$

Table 6.4: Assumptions about the parameter distribution and measurement errors for the transient dipole experiment, largely identical to those listed in table 6.1 for the stationary test cases. The logarithm $Y = \ln(K)$ has to be interpreted with K measured in ms^{-1}, and the logarithm $Z_s = \ln(S_s)$ with S_s measured in m^{-1}.

6.1.3 Inversion under Transient Conditions

To complement the test cases of the previous two sections, we simulate transient groundwater flow in a confined aquifer. The setup is identical to that of section 6.1.1, but we simulate the behavior of the system from the moment the pumps are started instead of assuming established stationary conditions. This requires additional assumptions about the log-storativity Z_s and its distribution, which can be found in table 6.4.

The domain Ω is again a square of size 100 m × 100 m, and the considered time interval is $T := [0\,\mathrm{s}, 50\,\mathrm{s}]$. While not completely realistic, we assume that the pumps are able to produce the required water flow instantaneously and keep it constant, since otherwise the fluctuating source and sink terms would have to be taken into account. We use a structured equidistant grid of size $n_\Omega = 512 \times 512 = 2.62 \times 10^5$ and $V_h^{(1)}(\mathcal{E}_h)$ as ansatz space, while the time interval is divided into $n_T = 50$ steps using the second-order implicit Alexander scheme, discretization 5. This leads to a discrete state space for ϕ of dimension $n_\Omega \cdot 4 \cdot n_T = 5.24 \times 10^7$. Two parameter vectors are required for the transient groundwater flow equation, model 2, and therefore the

Model Runs	$L(\mathbf{P}_{\text{map}})$	Mean	Variance	Skewness	Kurtosis
2082	151.8	−0.0232	1.314	−0.0551	3.007

Table 6.5: Number of forward and adjoint model runs, final value of the objective function and first four sample moments for the normalized residual $\Delta_{\mathbf{Z}}$ for the transient dipole experiment. The number of model runs required for convergence and the a posteriori statistics are similar to those obtained for the stationary test cases, compare table 6.3.

parameter space has dimension $2.62 \times 10^5 \cdot 2 = 5.24 \times 10^5$. The hydraulic head is monitored at 5×5 measurement locations, and 11 observations are recorded for each location at times $t_i = 5\,\text{s} + i \cdot 4\,\text{s}, 0 \leq i \leq 10$. The resulting observation space has dimension $n_\phi = 5 \cdot 5 \cdot 11 = 275$.

Two synthetic reference fields are generated, one for Y and one for Z_s. Then the transient groundwater flow equation is simulated with the boundary conditions from section 6.1.1 and the assumption that the hydraulic head is constant in Ω for $t = 0\,\text{s}$. The resulting head observations are combined with Gaussian noise to simulate measurement errors and then used as input for the PCG method with tolerance 10^{-5}.

The two reference fields and their estimates are shown in figure 6.9. The estimate of Y is comparable to those from the previous sections both in terms of variability and spatial resolution, see figure 6.2. The estimate of Z_s is significantly less detailed, and its upper left corner is an example of aliasing as described by *Li et al.* [52]. The synthetic reference isn't representative for the posterior distribution in this case, and the high values of Y in the affected area translate to an increase in the posterior mean of Z_s.

Both parameter estimates are accompanied by estimates of their uncertainty, see figure 6.10. These estimates of the posterior variance are computed using algorithm 8 ($\mathbf{SVD}_r$) as in the stationary case. The areas with the lowest uncertainty are also those with comparatively high spatial resolution and large variations of the posterior mean. The estimate of Z_s is significantly more uncertain than that of Y, indicating that the chosen setup is less suited for an estimation of Z_s. Table 6.5 contains the qualitative assessment of the inversion result. The required number of model simulations is similar to those of the stationary test cases, and the final value of the objective function $L(\mathbf{P}_{\text{map}})$ is one standard deviation away from its expected value. The moments of $\Delta_{\mathbf{Z}}$ are all very close to their expected values, as in the stationary case, indicating that the estimate $\mathbf{P}_{\text{map}}$ and the posterior covariance matrix $\mathbf{Q}_{\mathbf{PP}}^{\text{post}}$ are an adequate representation of the posterior distribution.

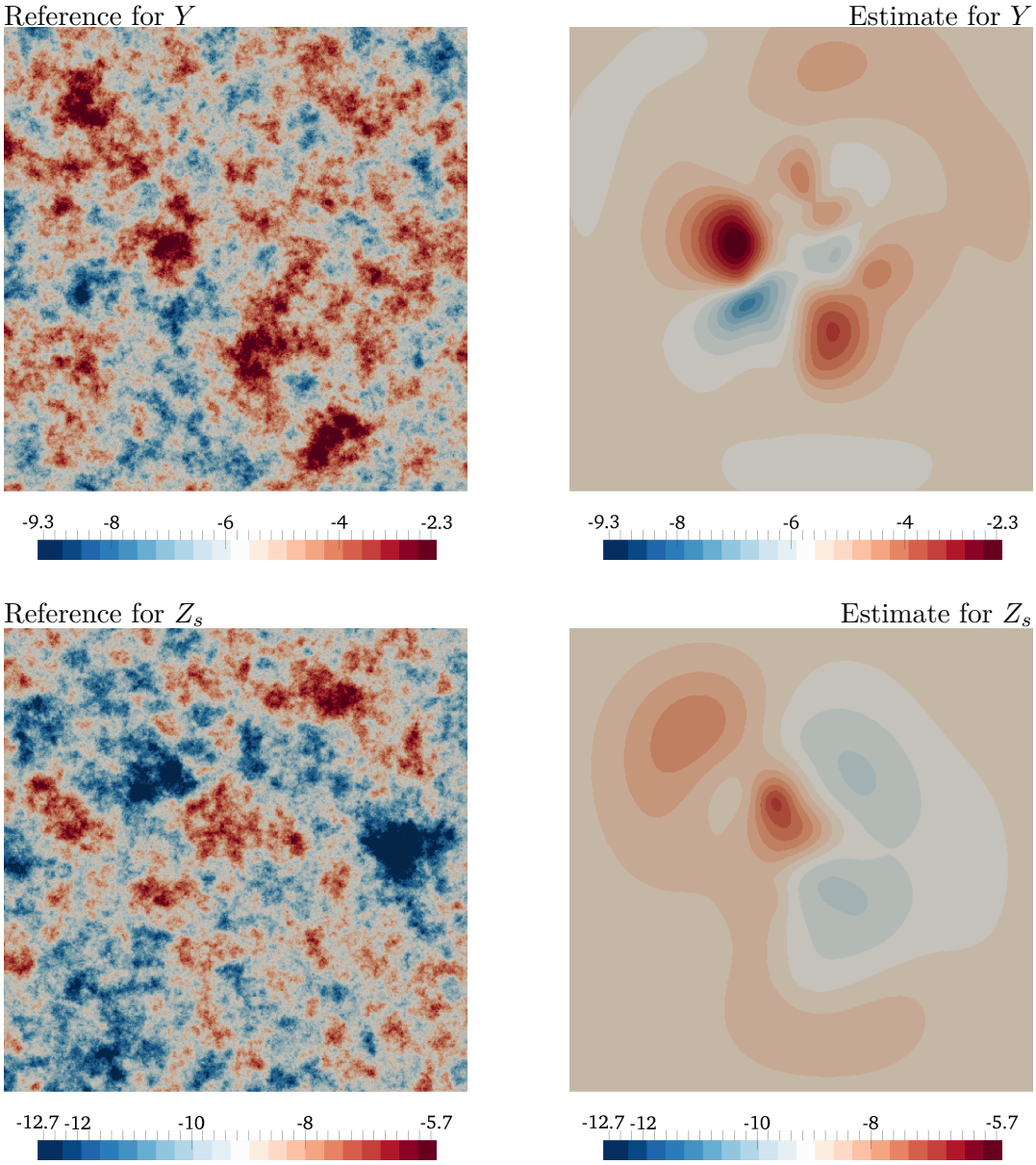

Figure 6.9: *Upper row:* Synthetic reference parameter field and estimate for the log-conductivity Y, $n_\Omega = 512 \times 512$. *Lower row:* Synthetic reference parameter field and estimate for the log-storativity Z_s. The estimate of Y has a similar structure and resolution as the estimate for $n_\phi = 5 \times 5$ in the stationary test case, compare figure 6.2, implying that the data gained from the first part of the drawdown curves contained approximately the same amount of information as the data taken from the stationary limit. The estimate of Z_s has less spatial resolution and displays aliasing in the upper left corner, caused by the synthetic reference having large values of Y in that area.

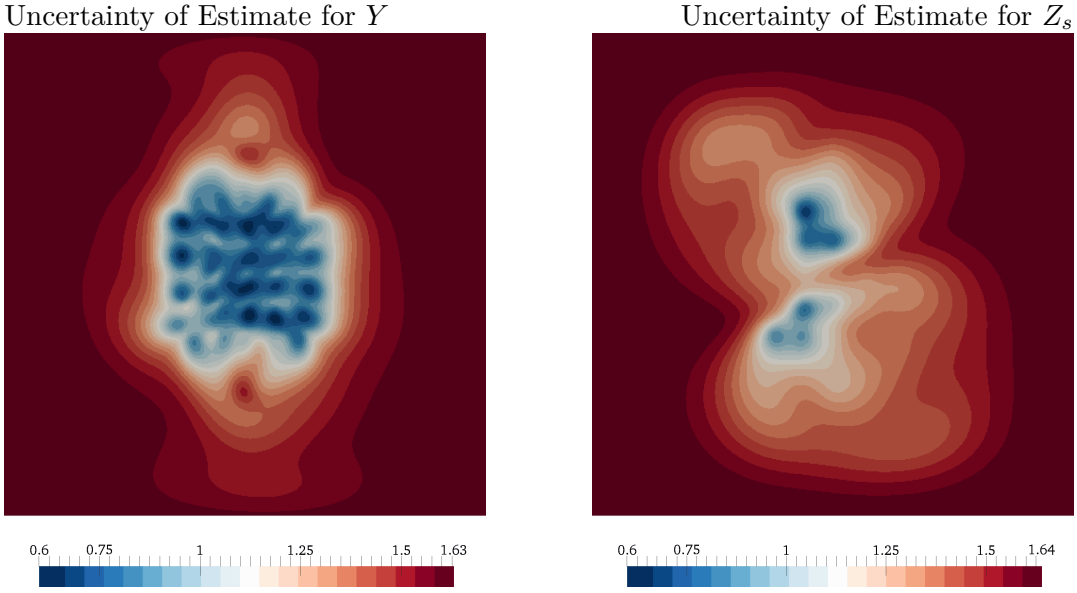

Figure 6.10: *Left:* Uncertainty for the estimate of Y. *Right:* Uncertainty for the estimate of Z_s. The posterior variance of Y is very similar to that seen in the stationary test cases. The posterior variance of Z_s is significantly higher. This is consistent with the lower spatial resolution of the estimate and the aliasing effects shown in figure 6.9.

6.2 Inversion of Solute Transport

As a second application we consider the joint inversion of head measurements and tracer concentration data. A solute is transported across a two-dimensional square domain of size $100\,\mathrm{m} \times 100\,\mathrm{m}$, see figure 6.11, with the water flow driven by a constant potential difference of $\Delta\phi = 3\,\mathrm{m}$ between the top and bottom boundaries. Both the hydraulic head ϕ and the tracer concentration c are monitored at an array of measurement locations that is spread across the lower part of the domain.

The transport equation and its adjoint are linear and the state c isn't used in any other context, which means that the unit of the state variable may be chosen freely. It is convenient to measure c in terms of an arbitrary but fixed reference concentration $\widehat{c}$, since this turns it into a unitless quantity. We also interpret c as the deviation from some given mean concentration c^*. These assumptions enable us to treat the initial condition c_0 as a unitless Gaussian random field with mean zero. The prior distribution of c_0 is described in table 6.6, as are the distributions of the log-conductivity Y and the measurements of ϕ and c. The parameters of the Bear-Scheidegger tensor are assumed to be constant for simplicity, and have the values $\lambda_l = 1 \times 10^{-3}\,\mathrm{m}$, $\lambda_t = 1 \times 10^{-5}\,\mathrm{m}$ and $D_m = 2 \times 10^{-9}\,\mathrm{ms}^{-1}$.

We discretize the domain with an equidistant structured grid of size $n_\Omega = 256 \times 256$,

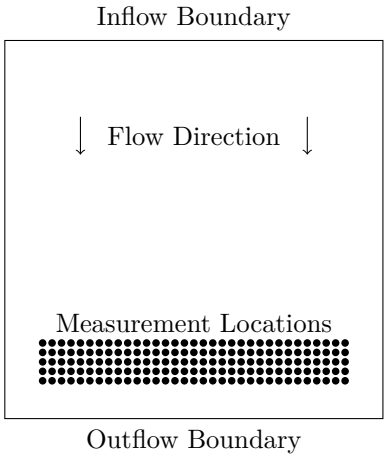

Figure 6.11: The experimental setup for the test case of section 6.2. The groundwater is driven by a constant potential difference of $\Delta\phi = 3\,\mathrm{m}$ between the top and bottom boundaries. A no-flow boundary condition is placed on the left and right boundaries. The resulting flow field transports a heterogeneously distributed tracer concentration across the domain, and the breakthrough curves are monitored at an array of measurement locations spread across the lower part of the domain.

Property	Description	Value
Initial condition c_0:		
	Covariance model	exponential
λ	Correlation length	$5\,\mathrm{m}$
σ^2	Prior variance	0.4
$\boldsymbol{\beta}^*$	Prior mean	0
$\sigma^2_{\boldsymbol{\beta}}$	Uncertainty of prior mean	0
Error of concentration measurements c:		
	Covariance model	uncorrelated
σ^2	Prior variance	2.5×10^{-3}

Table 6.6: Assumptions about the parameter distribution and measurement errors for the tracer experiment. The prior distributions for the log-conductivity Y and the measurement error of ϕ are the same as for the transient groundwater flow test case, see table 6.4. Both the tracer concentration c and its initial condition are relative values based on an arbitrary but fixed reference concentration $\widehat{c}$. The initial condition c_0 uses a fixed mean $c_0^* = 0$ instead of a trend parameter.

	Mean	Variance	Skewness	Kurtosis
$\Delta_{\mathbf{P}}$	0.202	9.396	1.439	8.287
$\Delta_{\mathbf{Z}}$	−0.827	192.5	2.229	34.63

Table 6.7: A posteriori statistics for the normalized error $\Delta_{\mathbf{P}}$ and the normalized residual $\Delta_{\mathbf{Z}}$ for the joint inversion of head measurements and tracer data. The high values of the sample variances indicate that the method didn't achieve full convergence for this test case.

leading to a parameter space of dimension $6.55 \times 10^4 \cdot 2 = 1.31 \times 10^5$. The time interval $T := [0, 5 \times 10^5\,\mathrm{s}]$ is divided into $n_T = 50$ steps of uniform size, with substeps as required by the CFL condition, equation (5.19). We use the ansatz space $V_h^{(1)}(\mathcal{E}_h)$ for both model equations, and therefore the discretized state space has dimension $n_\Omega \cdot 4 \cdot n_T \cdot 2 = 2.62 \times 10^7$. The array of measurement locations consists of 33×5 points and the system state is monitored at each of the $n_T + 1$ discrete times $t_i = i\mathrm{s}, 0 \leq i \leq 50$, which means the observation space has dimension $33 \cdot 5 \cdot 51 \cdot 2 = 1.68 \times 10^4$.

The PCG method stops after 276 iterations based on a total of 1180 simulations, producing the estimates shown in figure 6.12. The final value of the objective function is 1.23×10^4, while its expected value is $L(\mathbf{P}_{\mathrm{map}}) = 8415 \pm 92$, i.e. the final value is too large by a factor of 1.5 and deviates significantly from its expected value. This is confirmed by the statistical evaluation of the normalized error $\Delta_{\mathbf{P}}$ and the normalized residual $\Delta_{\mathbf{Z}}$, see table 6.7. In both cases the variance is significantly larger than one, indicating that the scheme didn't achieve full convergence. The directional derivative has become zero, although its theoretical value computed using the gradient, compare equation (2.43), would indicate a descent direction. This suggests that the discretization error is too large and causes inaccuracies in the adjoint state method. Possible solutions are switching to a finer resolution or using an adjoint equation that is derived directly from the discretized forward equation.

The PCG method was able to reduce the norm of the directional derivative by three orders of magnitude. This isn't enough to accept the result as an estimate of the mean of the posterior distribution, but it still allows the interpretation of the inversion result as an estimate of the synthetic reference. Figure 6.12 and figure 6.13 show that the estimate is able to predict the general structure and main features of the synthetic reference field, including the spatial mean in most parts of the domain. The inclusion of tracer data clearly improves the estimation of the log-conductivity Y, providing information about Y in parts of the domain not covered by the measurement array, despite the fact that the concentration c is itself unknown as well. The uncertainty quantification shown in figure 6.13 is based on an approximate SVD using $r = 250$ singular values.

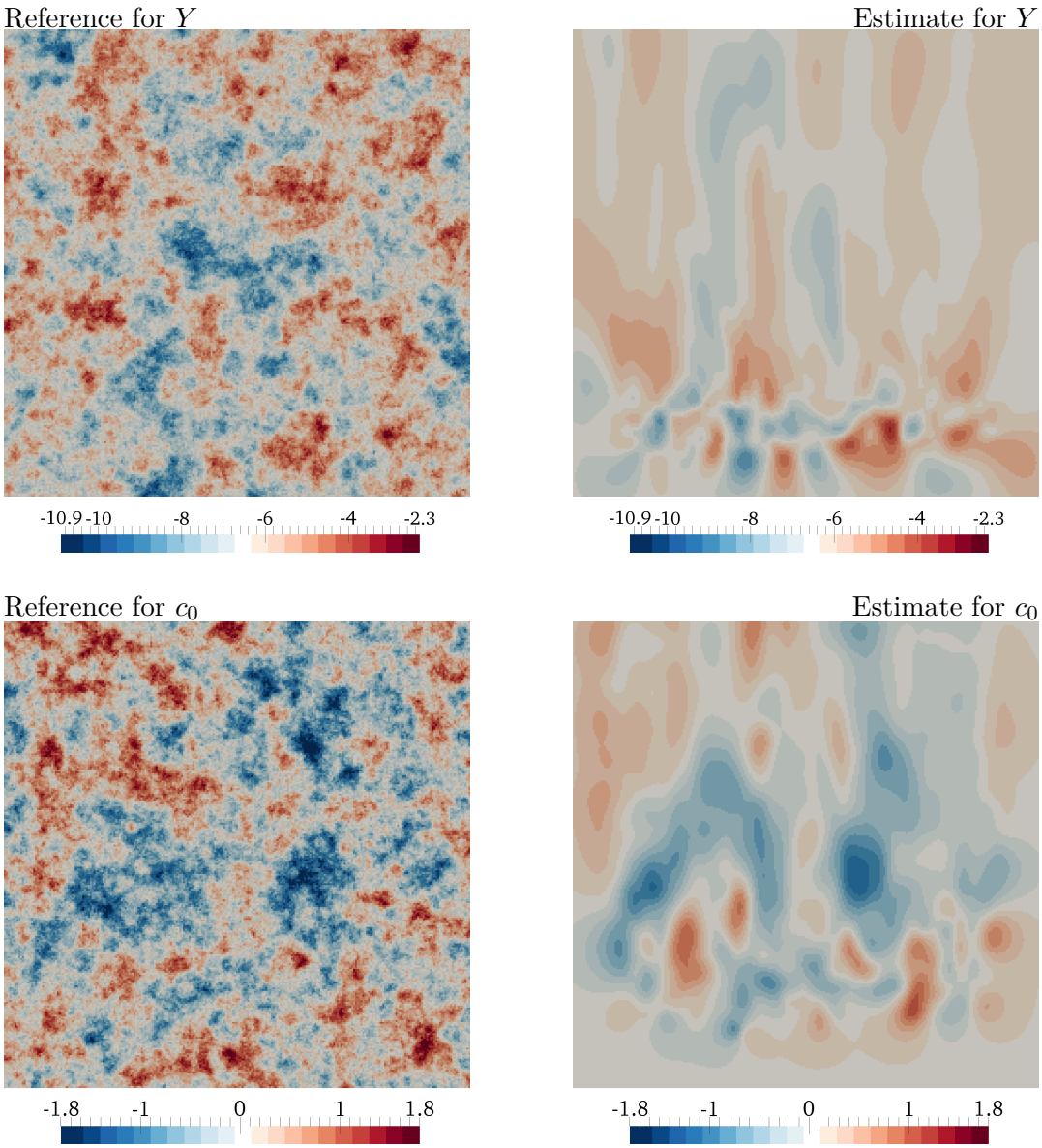

Figure 6.12: *Upper row:* Synthetic reference parameter field and estimate for the log-conductivity Y of the tracer experiment. *Lower row:* Synthetic reference parameter field and estimate for the initial value c_0. The estimate of Y is significantly more accurate in the vicinity of the measurement array than in the rest of the domain. The estimate of c_0 isn't as localized, but gradually loses accuracy when moving away from the array. The inclusion of tracer information improves the estimate of Y in comparison to those from the previous sections, especially in parts of the domain that are far away from the measurements.

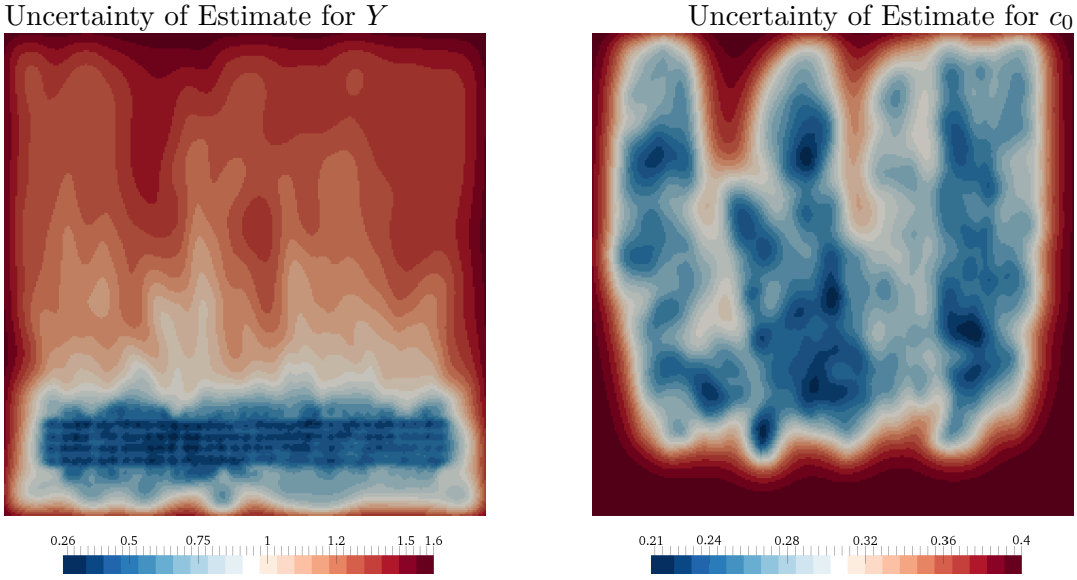

Figure 6.13: *Left:* Result of an approximate uncertainty quantification with $r = 250$ for the estimate of Y in figure 6.12. *Right:* Result of the same uncertainty quantification for the estimate of c_0. While the sensitivity of the log-conductivity Y is for the most part restricted to the direct vicinity of the measurement array, the initial value c_0 is sensitive in most parts of the domain.

6.3 Inversion of the Transient Richards Equation

The last application we consider is the inversion of the Richards equation. We assume that the domain Ω is filled with a Miller-similar medium with properties as listed in table 6.8. The parameter tuples represent the uncertain reference parameters used for Miller scaling. Their small but nonzero variance is based on the assumption that there isn't a perfect set of reference parameters describing the whole medium, i.e. imperfections and deviations in the parameterization exist and are spatially correlated. Equation (4.32) allows the definitions

$$Y_{\text{eff}} \coloneqq Y - 2 \cdot \chi, \quad \alpha_{\text{eff}} \coloneqq \alpha \left[\exp(\chi)\right]^{-1}, \quad \ln(\alpha_{\text{eff}}) = \ln(\alpha) - \chi, \tag{6.4}$$

where in a change of notation Y and α now refer to the reference values and Y_{eff} and α_{eff} to the actual local parameters. The other parameters are assumed to be scale-invariant, i.e. $n_{\text{eff}} \coloneqq n$ and $a_{\text{eff}} \coloneqq a$. A parameter estimate should first and foremost be able to reproduce these effective parameters, since they are the basis of the model. While the inversion may include measurements of both matric head and saturation, we restrict ourselves to direct measurements of ϕ_{m} for simplicity.

We consider transient unsaturated flow in a two-dimensional square domain Ω of size $2\,\text{m} \times 2\,\text{m}$, see figure 6.14. The bottom of the domain is kept at a constant potential

Property	Description	Value
Covariance structure for all parameters:		
	Covariance model	exponential
λ	Correlation length	0.2 m
Log-Miller scaling parameter χ:		
σ^2	Prior variance	1
$\boldsymbol{\beta}^*$	Prior mean	0.37
$\sigma^2_{\boldsymbol{\beta}}$	Uncertainty of prior mean	0
Saturated log-conductivity Y:		
σ^2	Prior variance	0.1
$\boldsymbol{\beta}^*$	Prior mean	-5.8
$\sigma^2_{\boldsymbol{\beta}}$	Uncertainty of prior mean	0.01
Van Genuchten parameter α:		
σ^2	Prior variance	$0.01\,\mathrm{m}^{-2}$
$\boldsymbol{\beta}^*$	Prior mean	$2\,\mathrm{m}^{-1}$
$\sigma^2_{\boldsymbol{\beta}}$	Uncertainty of prior mean	$0.1\,\mathrm{m}^{-2}$
Van Genuchten parameter n:		
σ^2	Prior variance	0.01
$\boldsymbol{\beta}^*$	Prior mean	1.6
$\sigma^2_{\boldsymbol{\beta}}$	Uncertainty of prior mean	1×10^{-3}
Mualem tortuosity parameter a:		
σ^2	Prior variance	0.01
$\boldsymbol{\beta}^*$	Prior mean	0.5
$\sigma^2_{\boldsymbol{\beta}}$	Uncertainty of prior mean	1×10^{-3}
Error of matric head observations ϕ_m:		
	Covariance model	uncorrelated
σ^2	Prior variance	$1 \times 10^{-4}\,\mathrm{m}^2$

Table 6.8: Assumptions about the parameter distribution and measurement errors for the Richards equation. All parameters except χ are defined on the reference scale. They represent the reference parameters of the Miller-similar medium and therefore have relatively low spatial variability. The log-Miller scaling parameter uses a fixed mean χ^* that essentially defines the reference scale, with the value 0.37 only chosen to show that the method can handle fixed means that aren't zero. The logarithm $Y = \ln(K)$ has to be interpreted with K measured in ms^{-1}.

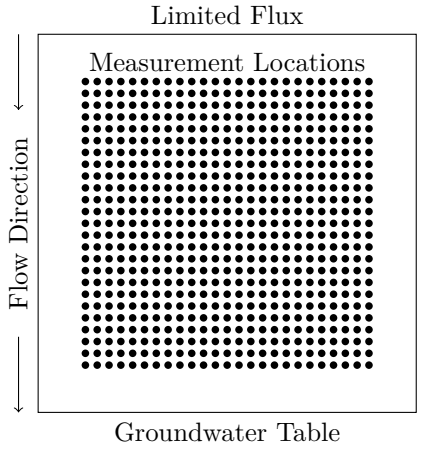

Figure 6.14: The experimental setup for the transient Richards equation. The potential at the bottom of the domain is held at a constant value of $\phi_{\mathrm{m}} = 0$. In a field experiment this describes the groundwater table, and in a laboratory setting it is a controlled boundary condition. At the top of the domain a limited flux condition is applied, which lets water propagate into the domain, while the water flux across the two sides of the domain is zero.

of $\phi_{\mathrm{m}} = 0\,\mathrm{m}$, and at $t = 0\,\mathrm{s}$ the forces are assumed to be in equilibrium, i.e. $\phi_{\mathrm{m}}(0)$ is proportional to the distance from the bottom boundary, reaching its lowest value $\phi_{\mathrm{m}} = -2\,\mathrm{m}$ at the top of the domain. The heterogeneity of the parameters implies that the actual equilibrium will deviate from this idealized initial condition, but we assume that the resulting fluxes can be neglected. A constant flux of $j_{\theta_w} = \min(\exp(Y), 3 \times 10^{-4}\,\mathrm{ms}^{-1})$ enters the domain at the top, and water starts flowing down through the domain. We limit the flux at the top of the domain to avoid boundary conditions that don't allow a solution. The dependence on Y doesn't influence the formulation of the adjoint model, since Y is treated as constant every time an adjoint equation is solved. No-flow boundary conditions are applied on the two remaining boundaries. We choose $\theta_s = 0.32$ and $\theta_r = 0.03$ for the saturated and the residual water content, and all other parameters of the Richards equation can be found in table 6.8.

The domain is discretized using a structured equidistant grid of size $n_\Omega = 128 \times 128$. The parameter field tuple is

$$S \coloneqq (\chi, Y, \alpha, n, a)\,, \tag{6.5}$$

producing a parameter space of dimension $n_\Omega \cdot 5 = 8.19 \times 10^4$. We simulate the flow dynamics in the time interval $T \coloneqq [0\,\mathrm{s}, 300\,\mathrm{s}]$, using the Alexander scheme, discretization 5, and adaptive timestepping through discretization 6. The solution of the forward model is stored with a resolution of 3 s, so that it is available for the adjoint model and the evaluation of state observations. Since the ansatz space is again $V_h^{(1)}(\mathcal{E}_h)$, the resulting discretized state space has dimension $n_\Omega \cdot 4 \cdot n_T =$

1.97×10^7. Synthetic measurements of the system state ϕ_m are generated using a grid of measurement locations of size 25×25, compare figure 6.14, and a sampling frequency of one measurement per three seconds. This produces an observation space of dimension $225 \cdot 25 \cdot 25 = 1.41 \times 10^5$, since we exclude a boundary layer of relative width 1/8, i.e. 0.25 m respectively 37.5 s, when taking measurements. In this case the observation space is larger than the parameter space, but the inverse problem still requires regularization due to the high autocorrelation of the state observations.

The PCG method is started with synthetic parameter fields and corresponding measurements as input, with Gaussian noise added to the observations to simulate measurement error. It stops after reaching the maximum number of iterations, which was set to 450 in this case. The synthetic reference fields and the resulting estimates can be seen in figures 6.15, 6.16 and 6.17. The large number of measurements has made it possible to reconstruct the log-Miller scale parameter χ in the part of the domain that was reached by the infiltrating water, apart from high-frequency fluctuations. The estimate of χ has a slightly larger mean than the reference, which is compensated by a corresponding rise in the mean of the estimates of Y and α. Due to the vastly different variances of the three prior distributions this is much more noticeable in the latter two estimates. This also explains the correlation between the estimates of Y and α, as it can be interpreted as an exchange between the correlated and uncorrelated parts of the effective parameters Y_eff and α_eff. This form of aliasing is expected, since the triple (χ, Y, α) is an overparameterization of the model. Figure 6.19 shows that the effective local parameters Y_eff and α_eff are indeed estimated correctly.

While the correlated deviations in Y and α can be attributed to interaction with the estimation of χ, this does not explain the large shift in the mean of α or the strong bias in the estimates of n and a. It is likely that these poor estimates are caused by the strongly nonlinear interdependence of the parameters, compare equations (4.30) and (4.31). The system stays relatively close to saturation, and therefore only a small part of the range $[\theta_r, \theta_s]$ of possible values for the water content θ_w is actually encountered during the simulation. This means completely different parameterizations $(\alpha_\mathrm{eff}, n, a)$ are able to explain the measurements. The system is insensitive with regard to these parameters under the chosen initial and boundary conditions.

The discussed issues are also apparent in the linearized uncertainty quantification shown in figure 6.18, which was computed using algorithm 8 ($\mathbf{SVD}_r$) with $r = 200$ singular values. It shows a drastic reduction in the uncertainty of χ, but only a moderate reduction in the uncertainty of the other parameters and almost none for the Mualem parameter a. This confirms that the method was able to reproduce the synthetic reference of χ with high accuracy and produced a relatively good estimate of Y, but couldn't constrain the other parameters to any significant degree.

The final value of the objective function is 73510, which is a significant deviation from the expected value $L(\mathbf{P}_\mathrm{map}) = 70630 \pm 270$. This indicates that the method wasn't able to converge as desired, most likely due to the low sensitivity of some

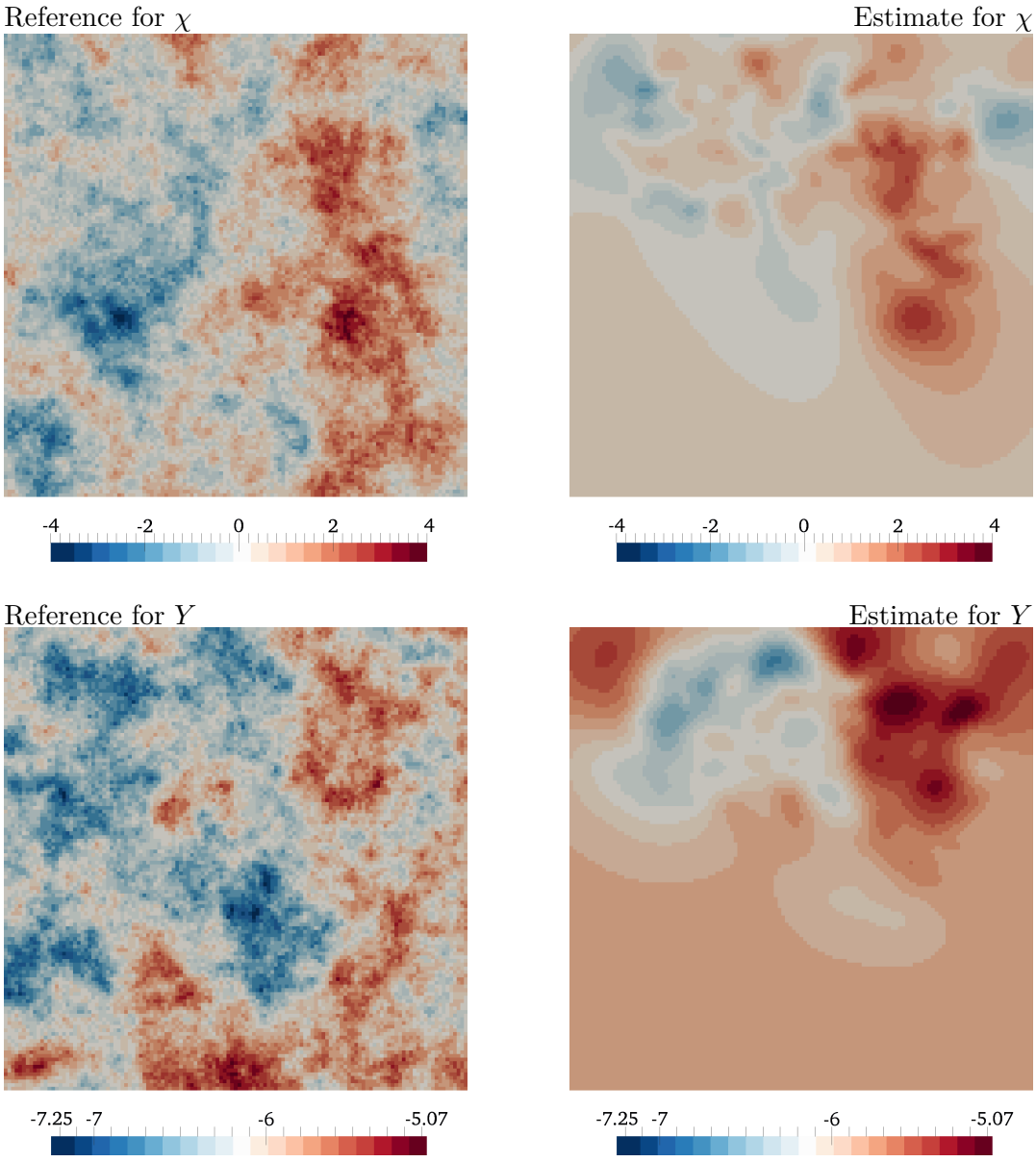

Figure 6.15: *Upper row:* Synthetic reference parameter field and estimate for the log-Miller scale parameter χ. *Lower row:* Synthetic reference parameter field and estimate for the log-conductivity Y. The large number of measurements makes it possible to almost completely recover χ in the areas that are reached by the infiltrating water. Due to the overparameterization of the model, a part of the effective log-conductivity Y_{eff} originating from χ is attributed to Y instead, leading to a biased estimate. Apart from the resulting shift in its mean, the estimate of Y is able to capture the coarse structure of the synthetic reference field.

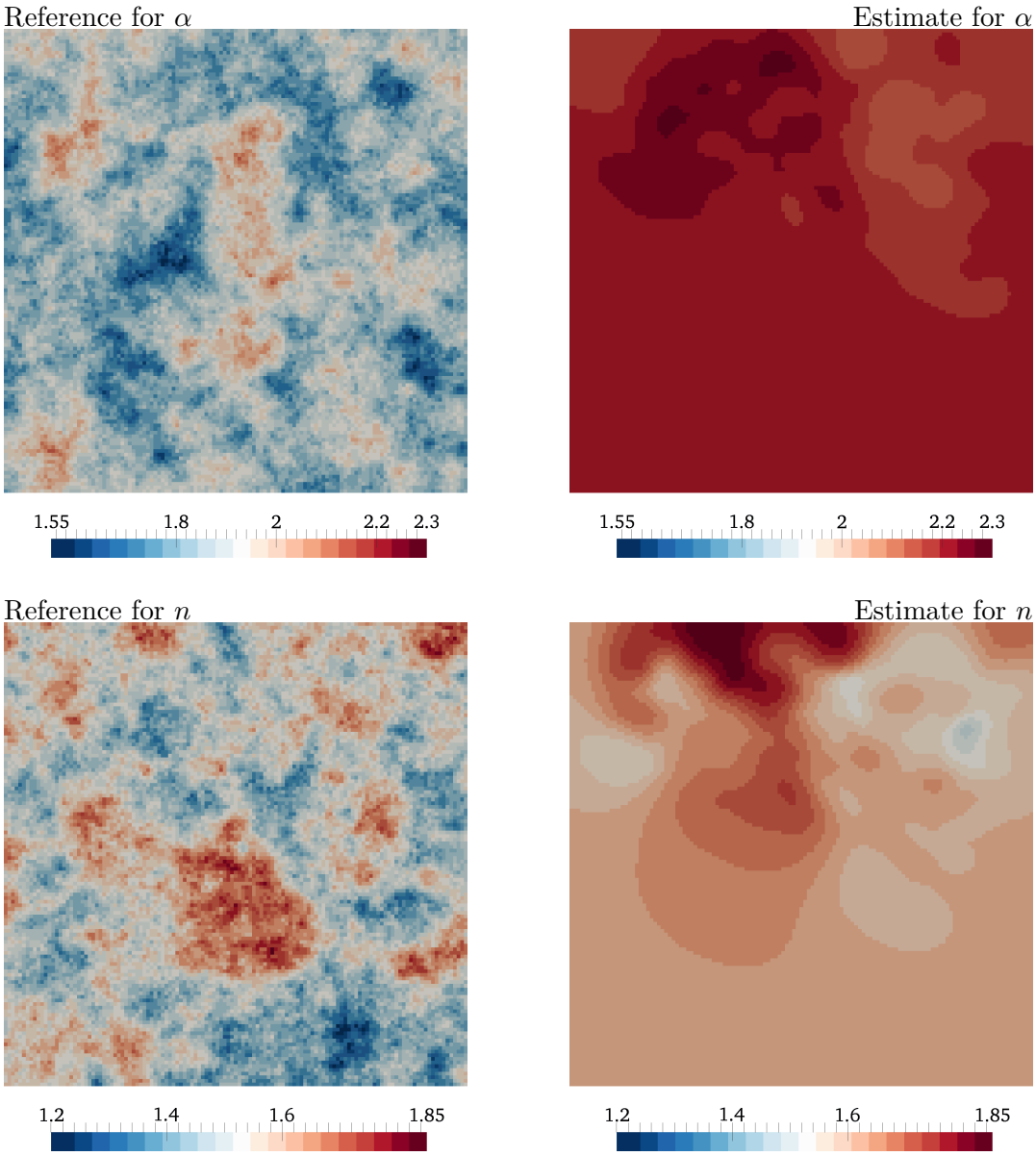

Figure 6.16: *Upper row:* Synthetic reference parameter field and estimate for the van Genuchten parameter α. *Lower row:* Synthetic reference parameter field and estimate for the van Genuchten parameter n. The estimate of α is unable to recover any information about the synthetic reference, with its spatial structure originating from the estimate of Y and its mean likely being an interaction with the estimation of n and a. The estimate of n recovers some information in the right part of the domain, but this is overshadowed by large deviations that occur on the left.

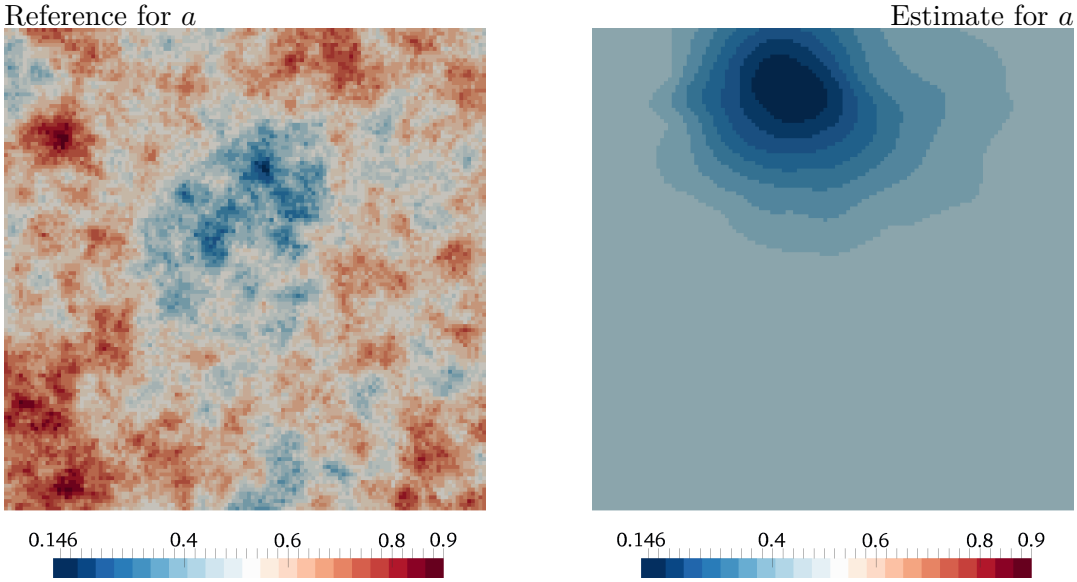

Figure 6.17: Synthetic reference parameter field and estimate for the Mualem tortuosity parameter a. The estimate is unable to recover any information about the synthetic reference, similar to the estimate of α in figure 6.16.

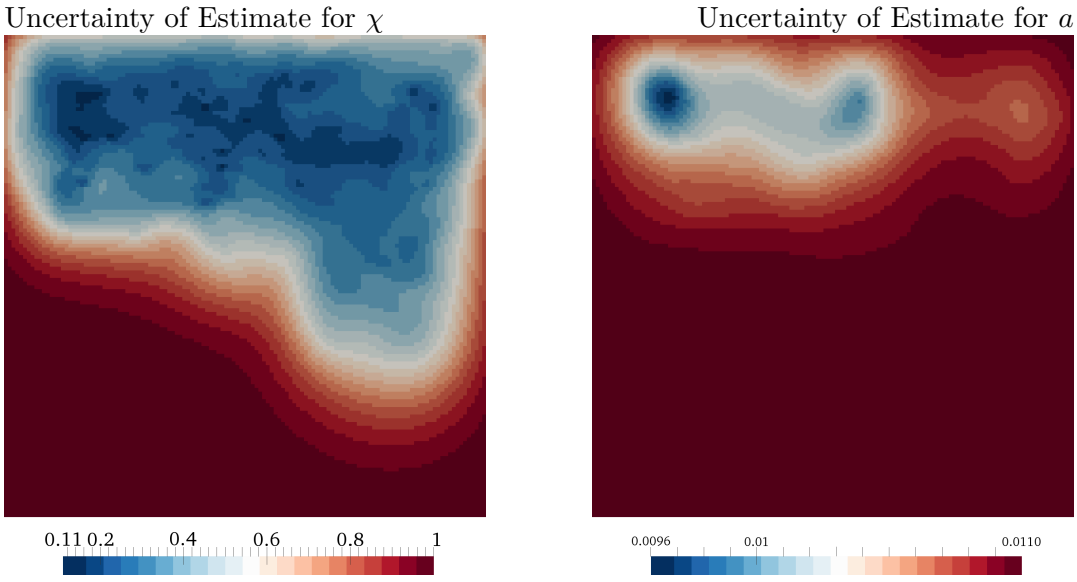

Figure 6.18: Two examples of the uncertainty quantification with $r = 200$ for the estimated parameter fields, *left:* uncertainty of the estimate of χ, *right:* uncertainty of the estimate of a. The estimate of χ has the highest reduction of uncertainty among the parameter fields, while the estimate of a has by far the lowest. The uncertainty estimates of the remaining parameter fields are structurally similar to that of χ, but do not achieve the same reduction of uncertainty.

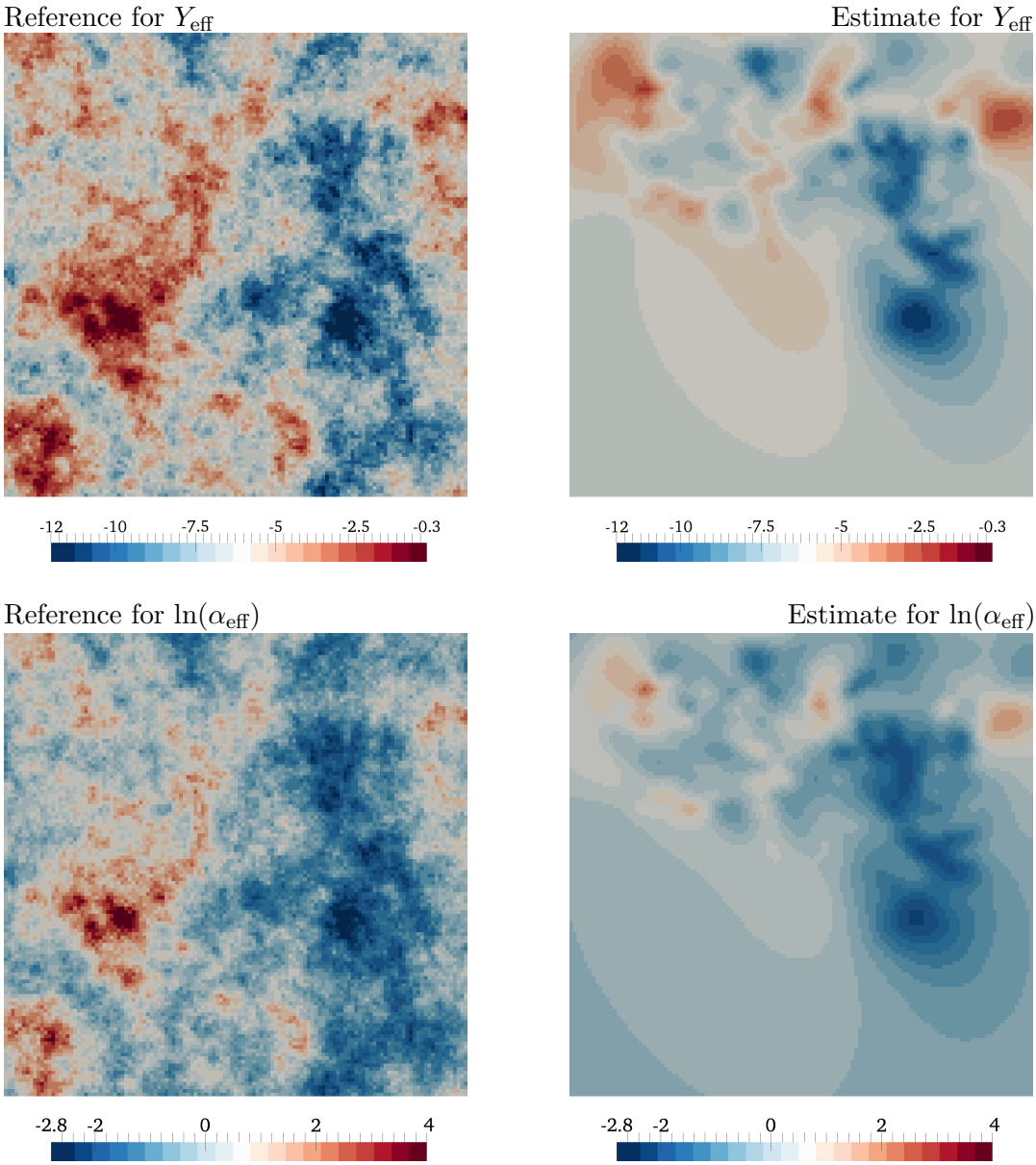

Figure 6.19: *Upper row:* Synthetic reference parameter field and estimate for the effective local log-conductivity Y_{eff}. *Lower row:* Synthetic reference parameter field and estimate for the effective local van Genuchten parameter α_{eff}. The construction based on equation (6.4) and the large variance of the log-Miller scale parameter χ yield two reference fields that are highly correlated. The two estimates capture all relevant features of the reference fields, suggesting that the bias in the underlying estimated parameters is caused by overparameterization and has no real impact on the system state.

Observations	Model Runs	$L(\mathbf{P}_{\text{map}})$	Mean	Variance	Skewness	Kurtosis
141250	2070	73510	3.928	133.9	1.778	16.56

Table 6.9: A posteriori statistics for the inversion of the transient Richards equation. The method would have required further iterations to achieve full convergence, as indicated by the final value of the objective function and the mean and variance of the normalized residual $\Delta_{\mathbf{Z}}$.

of the parameters. The same holds for the statistics of the normalized residual $\Delta_{\mathbf{Z}}$, compare table 6.9. An inspection of the evolution of the objective function and the directional derivative during optimization suggests that the iterative method may be converging to a local minimum of the objective function. If this is the case, then a different initial guess may lead to better results.

The result of the optimization isn't suitable as an estimate of the mean of the posterior distribution, but the test case demonstrates that the caching PCG method remains applicable when both the number of parameters and the number of observations are large and classical Hessian-based methods are no longer feasible. The parameter estimation could be improved through the inclusion of measurements of water content, possibly combined with modified boundary conditions, since this would put additional constraints on the Mualem-van Genuchten parameterization.

7 Conclusions

This thesis has presented several methods for the estimation of spatially resolved parameter fields, a caching prior preconditioned Conjugate Gradients method, algorithm 6 ($\mathbf{PCG}_c$), a randomized PCG method, algorithm 12 ($\mathbf{PCG}_c^{\text{post}}$), and a randomized Gauss-Newton method, algorithm 16 ($\mathbf{GN}_r$). All three methods are designed for the inversion of large data sets using high-resolution parameter fields.

The main idea behind the caching prior preconditioned CG method is the application of $\mathbf{Q}_{\mathbf{PP}}^{-1}$, the inverse of the covariance matrix of the parameters, as a preconditioner for the Conjugate Gradients method. This choice of preconditioner leads to mesh-independent convergence behavior, and at the same time it reduces the computational cost per iteration, since one of the most expensive operations can be removed from the algorithm. The other methods are based on a partial spectral decomposition or singular value decomposition of the Hessian of the preconditioned objective function, and avoid multiplication with $\mathbf{Q}_{\mathbf{PP}}^{-1}$ in a similar fashion.

The randomized partial spectral decomposition, algorithm 7 ($\mathbf{ED}_r$), and the randomized partial singular value decomposition, algorithm 8 ($\mathbf{SVD}_r$), are also used to provide linearized uncertainty quantification for the parameter estimates. The singular value decomposition can be used to normalize both estimation errors and measurement residuals, see algorithms 10 ($\mathbf{TU}_r^{\mathbf{P}}$) and 11 ($\mathbf{TU}_r^{\mathbf{Z}}$). These normalized errors and residuals provide information that makes it possible to check the inversion results for consistency and plausibility.

7.1 Summary

The caching prior preconditioned CG method, algorithm 6 ($\mathbf{PCG}_c$), is the main result of the previous chapters, as it has several important properties that are beneficial when considering high-dimensional inverse problems:

- It has the lowest cost per iteration of the considered methods, requiring three simulations of the forward model $\mathcal{F}$, one simulation of the adjoint model $\mathcal{F}^\dagger$ and a single multiplication with $\mathbf{Q}_{\mathbf{PP}}$ per step. The other methods require either significantly more expensive matrix operations or a larger number of model runs, compare section 2.8.1.

- It has the lowest memory requirements among the methods that remain efficient on meshes with high resolution, as discussed in section 2.8.2, storing eight parameter vector tuples and one discretized system state of the model $\mathcal{F}$. The only scheme with lower requirements is the CG method without preconditioner, algorithm 3 (**CG**), and this method becomes inefficient on finer meshes since its convergence rate is mesh-dependent, see section 2.3.4.
- The test cases of sections 6.1.1 and 6.1.2 show that the number of measurements doesn't have much influence on the required number of iterations, with the effort for inversion clearly sublinear in the number of observations. For the considered test cases the efficiency of the caching PCG method is comparable to the well-established Gauss-Newton method, with the Gauss-Newton scheme being faster if the number of measurements is small and the caching PCG method being more efficient for larger data sets.

Therefore the method remains applicable when other methods are no longer feasible due to memory constraints, and it is potentially orders of magnitude faster in situations where these other methods can still be applied.

We may conclude that the classical Gauss-Newton method in its compact formulation, algorithm 15 ($\mathbf{GN}_{\mathrm{CE}}$), remains the best choice if the number of state observations $N_{\mathbf{Z}} \coloneqq \prod_{j=1}^{n_{\mathbf{Z}}} n_{\mathbf{z}_j}$ is comparatively small, while the caching PCG method, algorithm 6 ($\mathbf{PCG}_c$), can become more efficient after a certain critical value of $N_{\mathbf{Z}}$. Furthermore, there is a critical value for $N_{\mathbf{P}} \coloneqq \prod_{i=1}^{n_{\mathbf{P}}} n_{\mathbf{p}_i}$ that marks the point after which the caching PCG method is the only applicable method among the ones that have been discussed. These two values will depend on the forward model $\mathcal{F}$ and the number and location of the measurements, but the test cases of chapter 6 suggest that they are in ranges that are realistic for applications, especially if the observations originate from a transient model or imaging techniques.

The randomized methods rely on partial spectral decompositions, which can be obtained with algorithm 7 ($\mathbf{ED}_r$) or algorithm 8 ($\mathbf{SVD}_r$). Both approaches require one run of the forward model $\mathcal{F}$ and one run of the adjoint model $\mathcal{F}^\dagger$ per eigenvalue. This means the number of essential eigenvalues r has to be smaller than $\frac{1}{2}N_{\mathbf{Z}}$, else the direct computation of $\mathbf{H}_{\mathbf{ZP}}$ of the classical methods would be computationally more efficient. The test cases of section 6.1 demonstrate that this isn't necessarily the case even if comparatively large values for $N_{\mathbf{Z}}$ are chosen. Therefore, the efficiency of these randomized methods depends on the shape of the spectrum of $\mathbf{M}_{\mathrm{like}}$ and ultimately on the amount of autocorrelation between the state observations.

The linearized uncertainty quantification and the two statistical tests that were introduced in section 2.6, algorithm 10 ($\mathbf{TU}_r^{\mathbf{P}}$) and algorithm 11 ($\mathbf{TU}_r^{\mathbf{Z}}$), also rely on these partial spectral decompositions, and therefore their efficiency also depends on the number of eigenvalues that have to be calculated. In these cases the results of the spectral decomposition aren't used to generate a search direction, they only provide a linearization of the model $\mathcal{F}$ in the direct vicinity of the parameter estimate $\mathbf{P}_{\mathrm{map}}$.

This means a smaller number of eigenvalues may already suffice if the resulting local linearization is accurate enough for the concrete application. The uncertainty estimates of sections 6.2 and 6.3 show that a very small number of recovered eigenvectors and eigenvalues may be enough to provide accurate results.

7.2 Outlook

While the efficiency of the proposed method has been demonstrated through the applications of chapter 6, all of the considered test cases are based on synthetic data. It is well known that good performance on artificial data doesn't automatically guarantee the same behavior for real-world applications, as real data is typically much more irregular. Therefore, the caching prior preconditioned Conjugate Gradients method, algorithm 6 ($\mathbf{PCG}_c$), needs to be tested with data from real experiments.

The randomized Gauss-Newton method, algorithm 16 ($\mathbf{GN}_r$), has only been applied to a small number of test cases in section 6.1, and in all of these tests it was less efficient than the classical Gauss-Newton method in the formulation of algorithm 15 ($\mathbf{GN}_{\mathrm{CE}}$). Its stabilized variants, the posterior preconditioned CG method of section 2.7, algorithm 12 ($\mathbf{PCG}_c^{\mathrm{post}}$), and the randomized Levenberg-Marquardt method mentioned in section 3.3.2, haven't been applied at all. These methods should be examined through synthetic test cases that are designed for their particular strengths, i.e. high-dimensional data sets with low effective dimension in terms of information content, e.g. unfiltered high-resolution time series.

The techniques that have been presented in this thesis, namely preconditioning with the inverse of the prior covariance matrix and calculating an approximate spectral decomposition for the Hessian of the preconditioned objective function, are quite general and may potentially be applied in other methods for Bayesian inversion or Bayesian data assimilation. Similarly, the potential of the presented methods as building blocks or preconditioners in more sophisticated parameter estimation methods may be of interest.

Definitions

List of Problems

List of Models

List of Discretizations

List of Algorithms

Notation

The lists of this chapter contain the symbols and acronyms used throughout the document. The symbols are grouped into three different categories for ease of reference, being Roman letters, Greek letters and remaining symbols. Any symbol that doesn't appear in the following lists, e.g. $\mathbf{R}$, $\mathbf{T}$ or $\mathbf{W}$, serves as an auxiliary variable and may have different meanings in different contexts.

With few exceptions, the following typographic conventions hold:

Bold	discrete vectors and matrices
$\mathcal{C}$alligraphic	abstract operators
Normal weight	continuous functions and scalar quantities
(...) Parentheses	function arguments and tuples
[...] Brackets	grouping of mathematical terms

A small number of symbols is used to represent more than one mathematical object, and these symbols are therefore listed several times. Care has been taken to make the intended meaning clear from context. The lists also contain the section of the first appearance for any symbol to help in the case of ambiguity or if further information is needed.

Roman Letters

Name	Dimension	Description	Section
a	$[-]$	parameter of Mualem parameterization	4.2
b_x^y		boundary value for state x and condition y (D: Dirichlet, N: Neumann)	4.1
C	$[-]$	total solute concentration	1.2
c	$[-]$	solute concentration in water	1.2
D	$[L^2T^{-1}]$	dispersion tensor	1.2
D_m	$[LT^{-1}]$	molecular diffusion tensor	4.3
d		dimension of domain	1.1
$\mathcal{E}_h$		set of elements of discretized domain	1.1
E		element of discretized domain	1.1
e_g	$[-]$	unit vector in direction of gravity	4.2
$\mathbf{F}$		discrete Fourier transform	2.1.2

Name	Dimension	Description	Section
$\mathcal{F}$		forward model	1.1
$\mathcal{F}_h$		set of faces of discretized domain	5.2.1
F		face in discretized domain	5.2.1
f_x		probability density function of x	2.1
$\mathbf{G}_x$		reconstructed discrete gradient of x	5.2.5
$\mathcal{G}$		discrete forward model	1.1
g	$[LT^{-2}]$	gravitational acceleration	4.1
$\mathbf{H}_{xy}$		sensitivity matrix of x with respect to y	2.3
h		mesh width of discretization	1.1
$\mathbf{I}$		identity matrix	2.1.2
$\mathcal{I}$		interpretation operator	1.1
$\mathbf{J}_x$		numerical flux of x, flux reconstruction	5.2.2
j_x		flux of quantity x or for state x	1.2
K	$[LT^{-1}]$	hydraulic conductivity	1.2
$\mathbf{L}$		matrix decomposition of type $\mathbf{M} = \mathbf{L}\mathbf{L}^T$	2.1.1
$\mathcal{L}$		Lagrangian of objective function	2.4.1
L		objective function	2.2.2
$\mathbf{M}_{\text{like}}$		likelihood part of preconditioned Hessian	2.5
$\mathbf{n}_x$		unit normal vector on x, or of boundary Γ	4.1
$\mathcal{N}(x, y)$		normal distribution with covariance matrix x and mean y	2.1.1
n	$[-]$	second van Genuchten parameter	4.2
n_x		number of entries or components of x	1.1
$\mathcal{O}$		observation operator	1.1
$\mathcal{O}(x)$		Landau symbol, complexity class of x	2.4
$\mathbf{P}$		tuple of parameter vectors	1.1
$\mathbf{P}_{\text{map}}$		MAP point of the objective function	2.2.2
$\mathbf{p}$		parameter vector	1.1
p		component of a parameter vector	2.4.1
$\mathbf{Q}_{xx}$		covariance matrix of x	2.1.1
$\mathbf{Q}_{xx}^{\text{post}}$		posterior covariance matrix of x	2.5
$\mathbf{Q}_{xy}$		cross-covariance matrix of x and y	2.2
Q_x^y		set of polynomials in x dimensions with maximum degree y per dimension	5.2.1
q_x		source term for quantity x or state x	1.2
r_x		reaction term for quantity x or state x	5.2
$\mathbf{s}$		parameter field, consisting of trend contributions and localized parameters	1.1
S		tuple of parameter functions	1.1
S_s	$[L^{-1}]$	specific storativity	1.2
s		parameter function on domain Ω	1.1

Name	Dimension	Description	Section
T		finite time interval $[0, t_{\max}]$	1.2
$T(x)$		complexity of given expression in terms of x	2.4
t	$[T]$	time	1.2
U		tuple of system states	1.1
u		system state, state function	1.1
u_h		discretized system state	5.1
V		space of system states	1.1
$\mathbf{x}$		physical coordinate in domain Ω	1.1
$\mathbf{x}_x$		realization of random variable x	2.1
$\mathcal{X}$		map from trend parameters to spatial interp.	1.1
$\mathbf{Y}$		random variable	2.1
$\mathbf{y}$		spatially localized part of parameter vector	1.1
Y		log-conductivity, $K = \exp(Y)$	1.2
$\mathbf{Z}$		tuple of measurement vectors	1.1
$\mathbf{z}$		measurement vector, vector of observations	1.1
Z_s		log-storativity, $S_s = \exp(Z_s)$	1.2

Greek Letters

Name	Dimension	Description	Section
α	$[L^{-1}]$	first van Genuchten parameter	4.2
$\boldsymbol{\beta}$		coefficient vector for trend parameters	1.1
Γ		domain boundary, $\Gamma = \partial\Omega$	4.1
Γ_x^y		boundary part for state x and condition y (D: Dirichlet, N: Neumann, O: Outflow)	4.1
$\Delta_{\mathbf{P}}$		normalized estimation error	2.6.2
$\Delta_{\mathbf{Z}}$		normalized measurement residual	2.6.3
δ_k		k-th time step width	5.1.1
δ_x		small change in x, perturbation	3.3.1
ϵ		measurement noise	2.2
ε		extension of boundary trace onto domain Ω	4.1
Θ	$[-]$	saturation	4.2
θ	$[-]$	porosity	1.2
θ_r	$[-]$	residual water content	4.2
θ_s	$[-]$	water content at saturation	4.2
θ_w	$[-]$	water content	1.2
κ	$[-]$	relative conductivity	4.2
$\mathbf{\Lambda}$		diagonal matrix of eigenvalues	2.1.2
λ_l	$[L]$	longitudinal dispersion coefficient	4.3

Name	Dimension	Description	Section
λ_t	$[L]$	transversal dispersion coefficient	4.3
ρ	$[ML^{-3}]$	density of water	4.1
ϱ		restriction to trace on boundary Γ	4.1
$\boldsymbol{\Upsilon}$		diagonal matrix of reciprocal eigenvalues	2.5
ϕ	$[L]$	hydraulic head	1.2
ϕ_{m}	$[L]$	matric head	4.2
χ		logarithm of Miller similarity scale parameter	4.2
Ψ		tuple of adjoint states	2.4.1
ψ		adjoint system state	2.4.1
ψ_g	$[ML^{-1}T^{-2}]$	gravity potential	4.1
ψ_m	$[ML^{-1}T^{-2}]$	matric potential	4.1
ψ_w	$[ML^{-1}T^{-2}]$	water potential	4.1
ψ_x		test function for state x	4.1
Ω		physical domain, compact subset of $\mathbb{R}^2$ or $\mathbb{R}^3$	1.1
ω		weight for averaging on faces	5.2.1

Symbols

Name	Dimension	Description	Section
x_0		initial value of state x	4.1
x^*		mean of x	2.1
x^T		transposed of matrix or vector x	2.1.1
$x^\dagger$		adjoint of operator or function x	2.4.1
$\widehat{x}$		reference value for x	2.3.4
$\widetilde{x}$		object of same type as x, "another" x	2.2.2
$\lvert x\rvert$		absolute value of x	2.1.1
$\lVert x\rVert_2$		Euklidean norm of x	2.1.1
$\lVert x\rVert_y$		norm of x induced by s.p.d. matrix y	2.2.1
$[\![x]\!]$		jump of x across face	5.2.1
$\{x\}_\omega$		weighted average of x on face	5.2.1
$\langle x, y\rangle_z$		L^2 scalar product of functions x and y on z	2.4.1
$\mathrm{d}_y x$		derivative of x with respect to y	2.4.1
$\partial_y x$		partial derivative of x with respect to y	1.2
$\nabla_y x$		gradient of x with regard to y	1.2
$\nabla_h x$		broken gradient on elements $E \in \mathcal{E}_h$	5.2.1
$\nabla \cdot x$		divergence of x	1.2

Acronyms

AMG	Algebraic Multigrid
CCFV	Cell-Centered Finite Volume
CFL	Courant-Friedrichs-Lewy condition
CG	Conjugate Gradients
CGNE	Conjugate Gradients on the Normal Equations
DUNE	Distributed and Unified Numerics Environment
EnKF	Ensemble Kalman Filter
EOC	experimental order of convergence
FFT	Fast Fourier Transform
GN	Gauss-Newton
IPdG	Interior Penalty discontinuous Galerkin
MAP	Maximum A Posteriori
MCMC	Markov-Chain Monte Carlo
ML	Maximum Likelihood
PCA	principal component analysis
PCG	preconditioned Conjugate Gradients
PCGA	Principal Component Geostatistical Approach
PDE	partial differential equation
PDF	probability density function
PSD	preconditioned Steepest Descent
QLGA	Quasi-Linear Geostatistical Approach
RKDG	Runge-Kutta discontinuous Galerkin
SD	Steepest Descent
SLE	Successive Linear Estimator
SSP	strong stability-preserving
SVD	singular value decomposition

Bibliography

[1] Andrés Alcolea, Jesús Carrera, and Agustín Medina. Pilot points method incorporating prior information for solving the groundwater flow inverse problem. *Advances in Water Resources*, 29(11):1678–1689, 2006. 72

[2] Roger Alexander. Diagonally implicit runge-kutta methods for stiff ode's. *SIAM Journal on Numerical Analysis*, 14(6):1006–1021, 1977. 113

[3] Richard C Aster, Brian Borchers, and Clifford H Thurber. *Parameter estimation and inverse problems*. Academic Press, 2013. 10, 69

[4] Ivo Babuska, Raúl Tempone, and Georgios E Zouraris. Galerkin finite element approximations of stochastic elliptic partial differential equations. *SIAM Journal on Numerical Analysis*, 42(2):800–825, 2004. 74

[5] Peter Bastian, Markus Blatt, Andreas Dedner, Christian Engwer, Robert Klöfkorn, Ralf Kornhuber, Markus Ohlberger, and Oliver Sander. A generic grid interface for parallel and adaptive scientific computing. part ii: Implementation and tests in dune. *Computing*, 82(2-3):121–138, 2008. 123

[6] Peter Bastian, Markus Blatt, Andreas Dedner, Christian Engwer, Robert Klöfkorn, Markus Ohlberger, and Oliver Sander. A generic grid interface for parallel and adaptive scientific computing. part i: abstract framework. *Computing*, 82(2-3):103–119, 2008. 123

[7] Peter Bastian, Markus Blatt, Christian Engwer, Andreas Dedner, Robert Klöfkorn, Sreejith Kuttanikkad, Mario Ohlberger, and Oliver Sander. The distributed and unified numerics environment (dune). In *Proc. of the 19th Symposium on Simulation Technique in Hannover*, 2006. 123

[8] Peter Bastian, Felix Heimann, and Sven Marnach. Generic implementation of finite element methods in the distributed and unified numerics environment (dune). *Kybernetika*, 46(2):294–315, 2010. 124

[9] James O Berger. *Statistical decision theory and Bayesian analysis*. Springer Science & Business Media, 1985. 24

[10] Markus Blatt. *A parallel algebraic multigrid method for elliptic problems with highly discontinuous coefficients*. PhD thesis, IWR, Heidelberg University, Germany., 2010. 124

[11] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004. 24

[12] Dietrich Braess. *Finite elements: Theory, fast solvers, and applications in solid mechanics*. Cambridge University Press, 2007. 84

[13] William L Briggs, Steve F McCormick, et al. *A multigrid tutorial*. SIAM, 2000. 39

[14] RH Brooks and AT Corey. Hydraulic properties of porous media. *Hydrology Papers, Colorado State University, Fort Collins, Colorado*, 1964. 87

[15] Tan Bui-Thanh, Carsten Burstedde, Omar Ghattas, James Martin, Georg Stadler, and Lucas C Wilcox. Extreme-scale uq for bayesian inverse problems governed by pdes. In *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, page 3. IEEE Computer Society Press, 2012. 10, 32, 43, 46

[16] NT Burdine et al. Relative permeability calculations from pore size distribution data. *Journal of Petroleum Technology*, 5(03):71–78, 1953. 87

[17] Olaf A Cirpka and Peter K Kitanidis. Sensitivity of temporal moments calculated by the adjoint-state method and joint inversing of head and tracer data. *Advances in Water Resources*, 24(1):89–103, 2000. 21

[18] Bernardo Cockburn and Chi-Wang Shu. The runge–kutta discontinuous galerkin method for conservation laws v: multidimensional systems. *Journal of Computational Physics*, 141(2):199–224, 1998. 109, 110

[19] Bernardo Cockburn and Chi-Wang Shu. Runge–kutta discontinuous galerkin methods for convection-dominated problems. *Journal of scientific computing*, 16(3):173–261, 2001. 114, 120

[20] Henry Darcy. *Les fontaines publiques de la ville de Dijon*. Victor Dalmont, 1856. 83

[21] Daniele A Di Pietro and Alexandre Ern. *Mathematical aspects of discontinuous Galerkin methods*, volume 69. Springer Science & Business Media, 2011. 111, 113, 115, 117, 118, 121, 122

[22] CR Dietrich and GN Newsam. Fast and exact simulation of stationary gaussian processes through circulant embedding of the covariance matrix. *SIAM Journal on Scientific Computing*, 18(4):1088–1107, 1997. 17, 18

[23] John Doherty. Ground water model calibration using pilot points and regularization. *Groundwater*, 41(2):170–177, 2003. 72

[24] John R Dormand and Peter J Prince. A family of embedded runge-kutta formulae. *Journal of computational and applied mathematics*, 6(1):19–26, 1980. 114

[25] Heinz W Engl, Martin Hanke, and Andreas Neubauer. *Regularization of inverse problems*, volume 375. Springer Science & Business Media, 1996. 9, 13, 24, 70

[26] Lawrence C Evans. *Partial Differential Equations*. Graduate studies in mathematics. American Mathematical Society, 1998. 9, 84

[27] Geir Evensen. *Data assimilation: the ensemble Kalman filter*. Springer Science & Business Media, 2009. 75, 76

[28] H Pearl Flath, Lucas C Wilcox, Volkan Akçelik, Judith Hill, Bart van Bloemen Waanders, and Omar Ghattas. Fast algorithms for bayesian uncertainty quantification in large-scale linear inverse problems based on low-rank partial hessian approximations. *SIAM Journal on Scientific Computing*, 33(1):407–432, 2011. 46

[29] Reeves Fletcher and Colin M Reeves. Function minimization by conjugate gradients. *The computer journal*, 7(2):149–154, 1964. 28

[30] Matteo Frigo and Steven G Johnson. The design and implementation of fftw3. *Proceedings of the IEEE*, 93(2):216–231, 2005. 124

[31] J Jaime Gómez-Hernánez, Andrés Sahuquillo, and José E Capilla. Stochastic simulation of transmissivity fields conditional to both transmissivity and piezometric data—i. theory. *Journal of Hydrology*, 203(1):162–174, 1997. 76

[32] Sigal Gottlieb, Chi-Wang Shu, and Eitan Tadmor. Strong stability-preserving high-order time discretization methods. *SIAM review*, 43(1):89–112, 2001. 113

[33] Allan Gut. *Probability: A Graduate Course*. Springer Science & Business Media, 2006. 14

[34] Jacques Hadamard. Sur les problèmes aux dérivées partielles et leur signification physique. *Princeton university bulletin*, 13(49-52):28, 1902. 13

[35] William W Hager and Hongchao Zhang. A survey of nonlinear conjugate gradient methods. *Pacific journal of Optimization*, 2(1):35–58, 2006. 28

[36] Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011. 10, 46, 47

[37] Martin Hanke, Andreas Neubauer, and Otmar Scherzer. A convergence analysis of the landweber iteration for nonlinear ill-posed problems. *Numerische Mathematik*, 72(1):21–37, 1995. 70

[38] W Keith Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970. 74

[39] Harold V Henderson and Shayle R Searle. On deriving the inverse of a sum of matrices. *Siam Review*, 23(1):53–60, 1981. 77

[40] Magnus R Hestenes and Eduard Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6):409–436, 1952. 25, 29

[41] DN Joanes and CA Gill. Comparing measures of sample skewness and kurtosis. *The statistician*, pages 183–189, 1998. 56

[42] Rudolph E Kalman. A new approach to linear filtering and prediction problems. *Journal of Fluids Engineering*, 82(1):35–45, 1960. 75

[43] C Ketelsen, R Scheichl, and AL Teckentrup. A hierarchical multilevel markov chain monte carlo algorithm with applications to uncertainty quantification in subsurface flow. *arXiv preprint arXiv:1303.7343*, 2013. 75

[44] Peter K Kitanidis. Orthonormal residuals in geostatistics: Model criticism and parameter estimation. *Mathematical Geology*, 23(5):741–758, 1991. 58

[45] Peter K Kitanidis. Quasi-linear geostatistical theory for inversing. *Water resources research*, 31(10):2411–2419, 1995. 14

[46] Peter K Kitanidis and Jonghyun Lee. Principal component geostatistical approach for large-dimensional inverse problems. *Water resources research*, 50(7):5428–5443, 2014. 11, 73

[47] Danie G Krige. *A statistical approach to some mine valuation and allied problems on the Witwatersrand: By DG Krige*. PhD thesis, University of the Witwatersrand, 1951. 13

[48] Charles L Lawson and Richard J Hanson. *Solving least squares problems*, volume 161. SIAM, 1974. 24

[49] Jonghyun Lee and Peter K Kitanidis. Large-scale hydraulic tomography and joint inversion of head and tracer data using the principal component geostatistical approach (pcga). *Water Resources Research*, 50(7):5410–5427, 2014. 11, 19, 73

[50] Heng Li and Dongxiao Zhang. Probabilistic collocation method for flow in porous media: Comparisons with other stochastic methods. *Water Resources Research*, 43(9), 2007. 73

[51] Wei Li and Olaf A Cirpka. Efficient geostatistical inverse methods for structured and unstructured grids. *Water resources research*, 42(6), 2006. 73

[52] Wei Li, Wolfgang Nowak, and Olaf A Cirpka. Geostatistical inverse modeling of transient pumping tests using temporal moments of drawdown. *Water resources research*, 41(8), 2005. 2, 19, 56, 138

[53] Georges Matheron. *Traité de géostatistique appliquée*. Editions Technip, 1962. 13

[54] Dennis McLaughlin and Lloyd R Townley. A reassessment of the groundwater inverse problem. *Water Resources Research*, 32(5):1131–1161, 1996. 1, 14, 76

[55] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953. 74

[56] Anna M Michalak and Peter K Kitanidis. Estimation of historical groundwater contaminant distribution using the adjoint state method applied to geostatistical inverse modeling. *Water Resources Research*, 40(8), 2004. 17, 21

[57] EE Miller and RD Miller. Physical theory for capillary flow phenomena. *Journal of Applied Physics*, 27(4):324–332, 1956. 88

[58] Yechezkel Mualem. A new model for predicting the hydraulic conductivity of unsaturated porous media. *Water Resources Research*, 12(3):513–522, 1976. 87

[59] Yechezkel Mualem and Gedeon Dagan. Hydraulic conductivity of soils: Unified approach to the statistical models. *Soil Science Society of America Journal*, 42(3):392–395, 1978. 87

[60] MEJ Newman and GT Barkema. *Monte Carlo Methods in Statistical Physics*. Oxford University Press: New York, USA, 1999. 74

[61] Wolfgang Nowak. *Geostatistical methods for the identification of flow and transport parameters in subsurface flow*. PhD thesis, Institut für Wasserbau, Universität Stuttgart, Stuttgart, Germany. (Available at http://elib.uni-stuttgart.de/opus/frontdoor.php), 2005. 16, 18, 58, 65, 78, 128

[62] Wolfgang Nowak and Olaf A Cirpka. A modified levenberg–marquardt algorithm for quasi-linear geostatistical inversing. *Advances in water resources*, 27(7):737–750, 2004. 11, 80

[63] Rene-Edouard Plessix. A review of the adjoint-state method for computing the gradient of a functional with geophysical applications. *Geophysical Journal International*, 167(2):495–503, 2006. 40, 42

[64] Elijah Polak and Gerard Ribiére. Note sur la convergence de méthodes de directions conjuguées. *ESAIM: Mathematical Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique*, 3(R1):35–43, 1969. 29

[65] Banda S RamaRao, A Marsh LaVenue, Ghislain De Marsily, and Melvin G Marietta. Pilot point methodology for automated calibration of an ensemble of conditionally simulated transmissivity fields: 1. theory and computational experiments. *Water Resources Research*, 31(3):475–493, 1995. 72

[66] Lorenzo A Richards. Capillary conduction of liquids through porous mediums. *Journal of Applied Physics*, 1(5):318–333, 1931. 86

[67] Kurt Roth. Steady state flow in an unsaturated, two-dimensional, macroscopically homogeneous, miller-similar medium. *Water Resources Research*, 31(9):2127–2140, 1995. 88

[68] Kurt Roth. *Soil Physics Lecture Notes, V2.2*. Institute of Environmental Physics, Heidelberg University, 2012. 1, 7, 81, 83

[69] Simo Särkkä. *Bayesian filtering and smoothing*. Cambridge University Press, 2013. 24, 43, 75

[70] Adrian E Scheidegger. General theory of dispersion in porous media. *Journal of Geophysical Research*, 66(10):3273–3278, 1961. 92

[71] Anneli Schöniger, Wolfgang Nowak, and Harrie-Jan Hendricks Franssen. Parameter estimation by ensemble kalman filters with transformed data: Approach and application to hydraulic tomography. *Water Resources Research*, 48(4), 2012. 76

[72] Ronnie L Schwede, Adrian Ngo, Peter Bastian, Olaf Ippisch, Wei Li, and Olaf A Cirpka. Efficient parallelization of geostatistical inversion using the quasi-linear approach. *Computers & Geosciences*, 44:78–85, 2012. 2, 76

[73] Jonathan R Shewchuk. An introduction to the conjugate gradient method without the agonizing pain, 1994. 25, 27

[74] Ne-Zheng Sun. *Inverse problems in groundwater modeling*, volume 6. Springer Science & Business Media, 2013. 10

[75] Ne-Zheng Sun and William W-G Yeh. Coupled inverse problems in groundwater modeling: 1. sensitivity analysis and parameter identification. *Water resources research*, 26(10):2507–2525, 1990. 42

[76] The HDF Group. Hierarchical Data Format, version 5, 1997-2016. http://www.hdfgroup.org/HDF5/. 124

[77] Andrey N Tikhonov. On the stability of inverse problems. *Dokl. Akad. Nauk SSSR*, 39(5):195–198, 1943. 69

[78] Abraham Van der Sluis and Henk A van der Vorst. The rate of convergence of conjugate gradients. *Numerische Mathematik*, 48(5):543–560, 1986. 32

[79] Martinus Th Van Genuchten. A closed-form equation for predicting the hydraulic conductivity of unsaturated soils. *Soil science society of America journal*, 44(5):892–898, 1980. 87

[80] Stephen Whitaker. Flow in porous media i: A theoretical derivation of darcy's law. *Transport in porous media*, 1(1):3–25, 1986. 83

[81] Max A Woodbury. Inverting modified matrices. *Memorandum report*, 42:106, 1950. 44

[82] Dongbin Xiu. Fast numerical methods for stochastic computations: a review. *Communications in computational physics*, 5(2-4):242–272, 2009. 74

[83] T-C Jim Yeh, Allan L Gutjahr, and Minghui Jin. An iterative cokriging-like technique for ground-water flow modeling. *Groundwater*, 33(1):33–41, 1995. 14

[84] DA Zimmerman, Ghislain De Marsily, Carol A Gotway, Melvin G Marietta, Carl L Axness, Richard L Beauheim, Rafael L Bras, Jesús Carrera, Gedeon Dagan, Paul B Davies, et al. 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. 1, 13