Bayesian Probabilistic Numerical Methods

by

Jonathan Paul Cockayne

Thesis

Submitted to the University of Warwick

for the degree of

Doctor of Philosophy in Statistics

Department of Statistics

July 2019

THE UNIVERSITY OF
WARWICK

Contents

List of Figures

Acknowledgments

First and foremost I would like to thank my wife Josie, who has supported me both in my decision to start this PhD and throughout it. I would neither have started nor finished without her. I would also like to thank my supervisor Mark Girolami, who gave me the opportunity to take my career in this surprising and rewarding direction. He has opened doors for me by introducing me to many of my collaborators.

No colleague deserves more thanks than Chris Oates. I would not be the researcher I am today without his abundant and generous support, guidance and patience. I also owe huge thanks to Tim Sullivan, whose guidance has led my research interests in directions that I would not have expected prior to starting the doctorate. Though they are more senior researchers, my work with them has always felt collaborative rather than supervisory and I cannot imagine what this thesis would have looked like without their collaboration. Lastly I would like to thank FX Briol, who has been a supportive colleague and friend throughout.

The University of Warwick has been an incredibly supportive host to my PhD, and I would like to thank all of the staff there who have supported my ongoing endeavours. Other institutions that have hosted me for a time during my PhD such as the Free University of Berlin, the Alan Turing Institute, the Max Planck Institute in Tübingen, CalTech and the University of Cambridge also have my thanks.

Declarations

The novel research in this thesis has predominantly resulted from collaborative work with co-authors on papers that have been submitted for publication. The extent to which this is my own work is now detailed.

Chapter 5 is based upon work which appears in Cockayne et al. [2016] and Cockayne et al. [2017]. The former paper was written collaboratively, predominantly with Prof. Oates, while the latter was written by myself. Prof. Oates and I contributed approximately equally to the written material and theoretical developments, while I provided the experimental results that appear.

Chapters 3, 6 and 7 are based upon the research that appears in Cockayne et al. [2019a]. This paper was proposed by myself, and predominantly written by myself and Prof. Oates with input from other co-authors. Again, Prof. Oates and myself contributed approximately equally to the written text of the paper. The theoretical results presented in this thesis are the subset of those in Cockayne et al. [2019a] to which I contributed heavily. The numerical results in these chapters are entirely my own work.

Section 7.2 features numerical results that also appear in Oates et al. [2019]. These numerical results were generated entirely by myself.

Chapter 4 is based on work in Cockayne et al. [2019b]. This paper was predominantly written by myself with input from co-authors. I also derived the theoretical results and produced the numerical results. Limited results from Bartels et al. [2019] also appear; this paper was predominantly written by myself and Simon Bartels, but the results which appear in this thesis were developed independently by myself.

Abstract

The increasing complexity of computer models used to solve contemporary inference problems has been set against a decreasing rate of improvement in processor speed in recent years. As a result, in many of these problems numerical error is a challenge for practitioners. However, while there has been a recent push towards rigorous quantification of uncertainty in inference problems based upon computer models, numerical error is still largely required to be driven down to a level at which its impact on inferences is negligible. Probabilistic numerical methods have been proposed to alleviate this; these are a class of numerical methods that return probabilistic uncertainty quantification for their numerical error. The attraction of such methods is clear: if numerical error in the computer model and uncertainty in an inference problem are quantified in a unified framework then careful tuning of numerical methods to mitigate the impact of numerical error on inferences could become unnecessary.

In this thesis we introduce the class of *Bayesian* probabilistic numerical methods, whose uncertainty has a strict and rigorous Bayesian interpretation. A number of examples of *conjugate* Bayesian probabilistic numerical methods are presented before we present analysis and algorithms for the general case, in which the posterior distribution does not posess a closed form. We conclude by studying how these methods can be rigorously composed to yield Bayesian pipelines of computation. Throughout we present applications of the developed methods to real-world inference problems, and indicate that the uncertainty quantification provided by these methods can be of significant practical use.

Part I

Introduction

Chapter 1

Introduction

> I may not have gone where I intended to go, but I think I have ended up where I intended to be.
>
> —*Douglas Adams*

The subject of the thesis, and its core research questions, will first be introduced. In this chapter mathematical statements are somewhat sparse and important concepts will be introduced rigorously in Chapter 2. The goal here is instead to provide intuition into the subject of, and main goals of the thesis.

1.1 Introduction to the Introduction

Since the advent of the digital age, the applied sciences have increasingly depended on computer models to simulate from complex physical processes. These models are formed of sets of equations that rarely have an analytical solution, and so the equations are *discretised* to produce a *numerical method* that yields an approximation to the solution. The error incurred by discretising such systems is referred to as *discretisation error*. In recent years there has also been a blossoming interest in the field of *uncertainty quantification* (UQ), which seeks to quantify, often probabilistically, the uncertainty incurred when synthesising computer models with imperfect or uncertain data. *Probabilistic numerical methods* (PNM) bring together the fields of numerical analysis and uncertainty quantification by providing a *probabilistic quantification of discretisation error.*

The roots of PNM can be traced back to the start of the 20th century [Oates and Sullivan, 2019], but recent years have seen a surge in their development. Great leaps forward have been made, but the field has been constrained by a lack of underlying theoretical principles. In particular, such methods are often labelled "Bayesian", suggesting that the uncertainty output has a rigorous and well-understood interpretation. However, until the work in this thesis, there was no common definition of what makes a PNM Bayesian, and the particular features of the setting in which PNM are generally constructed make such a definition surprisingly complicated to elicit.

This thesis sets out to answer this question. In particular, the research questions answered in this thesis are:

1. What makes a PNM Bayesian?
2. When are Bayesian PNM well-defined?
3. When can Bayesian PNM be usefully composed?

To answer these questions, we will construct a number of simple Bayesian PNM in tractable settings to identify their common characteristics, and then proceed to a rigorous treatment of the general setting. At the conclusion of the thesis, we will have established general conditions for a Bayesian PNM to have a well-defined output, have established algorithms for sampling from the output, and studied the composition of Bayesian PNM.

1.2 Numerical Analysis as Inference

This section will introduce the background for the thesis and the core conceit of PNM: that problems in numerical analysis can be phrased as *inference problems.*

1.2.1 Numerical Methods and Discretisation Error

Many equations that are of significant practical interest in the applied sciences do not have a closed-form solution. To provide a canonical example, let D be an open subset of $\mathbb{R}^d$ with boundary ∂D and consider the following elliptic partial differential equation (PDE):

$$\begin{aligned} -\nabla \cdot \Big(\kappa(\boldsymbol{x}) \nabla u^\dagger(\boldsymbol{x})\Big) &= g(\boldsymbol{x}) & \qquad \boldsymbol{x} &\in D \\ u^\dagger(\boldsymbol{x}) &= b(\boldsymbol{x}) & \qquad \boldsymbol{x} &\in \partial D \end{aligned} \tag{1.1}$$

This PDE is a simplified model for the flow of a quantity through a porous medium, described by the domain D. The function $\kappa(\boldsymbol{x})$ describes the permeability of the domain, while $u^\dagger(\boldsymbol{x})$ is the pressure field for the quantity, with the superscript $\dagger$

used to indicate that it is the "true" solution to the PDE. The applications of this PDE are numerous; it has been used as a model for the steady state in groundwater flow problems [Wang and Anderson, 1982] and also for medical imaging [see e.g. Holder, 2004]. However, while it can be proven that under certain not particularly restrictive conditions on D, κ, g and b a suitable notion of a solution exists and has certain regularity properties, no closed-form for the solution can be determined apart from in pathological cases.

The practical importance of this and similar systems cannot be overstated. Indeed, modelling physical processes using intractable equations is increasingly central to the applied sciences and society, as discussed in detail in the recent Blackett report [Government Office for Science, 2018]. Numerical analysis is concerned with the construction of and analysis of algorithms for producing approximate solutions to such equations. This unspecified approximation is termed here $\hat{u}$. To produce the approximation, generally some kind of *discretisation* of the equations is employed.

For example, in the finite difference method (FDM) for PDEs the continuous domain D is replaced with a finite set of points, defined on a regular grid, and the continuous equations replaced by finite difference approximations thereof, resulting in a finite-dimensional linear system that can be solved to approximate $u^\dagger(\boldsymbol{x})$. Alternatively the discretisation could be of the function space that $u^\dagger$ occupies; in the finite element method (FEM) this space is replaced by a finite-dimensional subset thereof, whose basis functions are defined on small cells (called *elements*) of the domain.

Regardless of how the discretisation is constructed, the translation from a problem that is continuous and infinite-dimensional to one that is finite-dimensional generally results in an error, $\|u^\dagger - \hat{u}\|$, where $\|\cdot\|$ is an unspecified norm adapted to the problem. This error is called *discretisation error*, and controlling that error is again the focus of much of numerical analysis [Higham, 2002]. Control over this error as a function of the discretisation resolution is a basic consistency requirement for numerical algorithms, and more detailed descriptions of the error can be used to adaptively refine the discretisation at the most critical locations in the domain.

Non-Negligible Error

There exist numerous applications in which current computational capacity is insufficient to allow a discretisation detailed enough to make discretisation error negligible. Two examples in which this is a serious and present challenge are:

- **Climate Modelling**. The importance of climate modelling [IPCC, 2009] hardly needs to be stated in the modern day. The domain of the problem is the earth, and the long timescales over which simulations of future climate is required composed with the fine-scale on which the phenomena modelled occur is such that a suitably fine discretisation is impossible.

- **Computational Biology**. Patient-specific models [Niederer et al., 2011] are a tool of emerging importance in medicine. In cardiology, patient-specific models are being developed to assist cardiac surgeons in procedures. For example, in the treatment of cardiac arrhythmias, a patient-specific model of electrical conductivity could be used to determine where in the heart to ablate in order to treat the arrhythmia. To be of use, it must be possible to simulate from these models on clinical timescales. The resolution of the discretisation, and thus the size of discretisation error, competes with the solution speed required for this practical application.

Many more such examples exist, but even from these two instances it is clear that discretisation error is not a "solved problem" in numerical analysis. Rather, this is a significant present challenge that affects the ability of applied scientists to tackle significant, real-world problems in society.

1.2.2 Inverse Problems and Uncertainty Quantification

Of particular relevance to this thesis are *inverse problems*. Consider a parameter θ, to be determined, and let $\mathcal{G}$ denote a map known as the *parameter to observation map*. Experimental data y is related to θ by

$$y = \mathcal{G}(\theta) + \xi$$

where ξ represents some noise process. To be concrete, here $\mathcal{G}$ might compose some observation operator with the solution of a computer model, such as Eq. (1.1), while θ represents unknown parameters of that model which must be inferred, perhaps to study some physical aspect of the problem such as the permeability field in Eq. (1.1), or perhaps to make predictions from the model. The inverse problem is the problem of "inverting" $\mathcal{G}$ to obtain an estimate of θ.

The reason for placing inverted commas around "inverting", above, is that the inversion procedure is usually ill-posed. This could be for any of a number of reasons, expounded in Dashti and Stuart [2017]:

1. If the dimension of the spaces which θ and y occupy differs, then the problem is either *over-determined* or *under-determined* and so a single θ representing the unique inverse of $\mathcal{G}$ cannot be determined.

2. $\mathcal{G}$ might not be a linear map, in which case a unique solution is not guaranteed in any case.

3. The observed instance of the noise ξ might be such that y lies outside of the image of $\mathcal{G}$.

As a result, some regularisation of the problem is needed to ensure that a solution to it exists. Bayesian methods [Stuart, 2010] have emerged as a popular method by which to perform this regularisation.

In a Bayesian inverse problem, a *prior* distribution $p(\theta)$ is placed on θ. This reflects the prior beliefs of the user about the parameter. This is then combined with a *likelihood*, $p(y|\theta)$ that describes how likely the data is to be observed under a given parameter, to obtain a *posterior* $p(\theta|y)$ that describes the user's beliefs given the data that they have observed. Such problems take their name from *Bayes theorem* which, assuming the parameters and data are finite-dimensional, provides an explicit form for the posterior distribution:

$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{p(y)} \tag{1.2}$$

where

$$p(y) = \int p(y|\theta)p(\theta)\mathrm{d}\theta.$$

A subtlety of the Bayesian formalism is the way in which *uncertainty* enters the equation. The most obvious way in which uncertainty arises is through the noisy observations of y, represented by ξ. However, a second source of uncertainty arises from the often *underdetermined* nature of such problems. Even if ξ is constant at 0, resulting in exact observations, the fact that y does not contain enough information to uniquely determine θ can still be interpreted as a kind of uncertainty. Here the uncertainty represents the extent to which θ *can* be determined from what has been observed. This interpretation is central to PNM, which posit that numerical problems can be viewed as a kind of *noiseless* inverse problem, and thus are amenable to a statistical treatment.

1.2.3 Probabilistic Numerical Methods

Probabilistic numerical methods are numerical methods that return a probability distribution, whose purpose is to quantify uncertainty in the solution due to discretisation error. This section will provide a literature review on the roots of PNM, up to present day, and place them in the context of the material thus far presented in this chapter. Note that more thorough literature reviews for the particular problems considered in this thesis are provided in the relevant chapters; for for linear solvers in Chapter 4 and for PDEs in Chapter 5.

The idea to apply a statistical methodology to numerical problems is by no means a modern creation. Perhaps the earliest reference to an approach that we now understand as a PNM appears in Poincaré [1912][1]. In that work, after examining several different perspectives on the problem of function approximation, Poincaré ultimately arrives at a Bayesian approach to the problem that we would now understand as Gaussian process regression, many years before the theory on Gaussian processes was formalised. Function approximation is in many ways a prototypical numerical method, and so this is perhaps the earliest instance of a mathematician proposing that a statistical approach be applied to a numerical problem.

The modern perspective on PNM first appeared in a remarkable and prescient series of papers by Frederick Michael (F.M.) Larkin (particularly Larkin [1972]; see also Larkin [1969, 1970, 1974, 1979b,a]; Kuelbs et al. [1972]). In those papers, Larkin proposed modelling an unknown function using a Gaussian measure on a Hilbert space, and producing a numerical method by conditioning that measure on knowledge of a finite number of functionals evaluated on the unknown. In addition to applying this to function approximation, the technique we now know as Bayesian quadrature was discussed in Larkin [1972]. Other early proponents of this view include Kadane and Wasilkowski [1985] and Diaconis [1988]. In his paper, Diaconis again introduces Bayesian quadrature, motivating this approach by way of the quote included at the start of Chapter 6. His argument is that, even though one has an analytical expression for a function f, many properties of it are nevertheless unknown, and so an approach which acknowledges that uncertainty is justified.

Bayesian optimization was developed at the end of the 1980s, in Mockus [1989] and Törn and Žilinskas [1989]. This approach augments a function approximation procedure with minimization[2]; the uncertainty in the function is used both to acknowledge resulting uncertainty in its minimum, and to develop a procedure for sequentially selecting locations at which to interrogate the function based on the belief about where its minimum *might* lie. Research on Bayesian optimization is slightly at odds with other PNM, in the sense that the probability is predominantly used to construct an evaluation strategy rather than forming a fundamental output of the method. Nevertheless, to this day it remains among the most popular

[1] Noted in Diaconis [1988]

[2] Equivalently, maximization; we assume minimization for simplicity.

and successful PNM, and is widely used to optimize hyperparameters in machine learning methods [Snoek et al., 2012].

Interest in PNM has surged in recent years, spurred by the positioning paper of Hennig et al. [2015]. Bayesian quadrature methods continue to be developed, with O'Hagan [1991] in the 1990s, and recent developments including Briol et al. [2019]; Xi et al. [2018]; Karvonen and Särkkä [2017]; Karvonen et al. [2018]. New areas in which PNM have been developed include numerical linear algebra [Hennig, 2015; Cockayne et al., 2019a; Bartels et al., 2019, see also Chapter 4], ordinary differential equations [Schober et al., 2014; Conrad et al., 2017; Kersting and Hennig, 2016; Chkrebtii et al., 2016, see also the discussion in Section 1.3] and partial differential equations Owhadi [2015]; Cockayne et al. [2016, see also Chapter 5]. Development in Bayesian optimization also continues at a rapid pace; see Snoek et al. [2012] for a review. Yet, literature on foundational principles of PNM has thus far been surprisingly sparse, a gap which this thesis seeks to address.

1.3 Towards Bayesian Probabilistic Numerical Methods

Despite the surge in interest in PNM, foundational theoretical contributions have not matched the pace of new developments. A particular open question is when PNM output a distribution can truly be interpreted as a Bayesian posterior. To take ODEs as an example, consider an ODE of the form

$$\begin{aligned} u' &= f(u(t)) & t &\in [0, T] \\ u(0) &= u_0. \end{aligned}$$

The works of Schober et al. [2018] and Chkrebtii et al. [2016] each begin with a Gaussian prior, and discretise the domain with a grid of points $0 < t_1 < \cdots < t_n = T$. A simplified version of one approach from Schober et al. [2018] proposes to obtain data by sequentially evaluating $y_i = f(m_i(t_i))$, and conditioning the prior on $u'(t_i) = y_i$. Here m_i is the posterior mean at iteration i. Conversely, Chkrebtii et al. [2016] proposes to obtain data by *sampling* $u_i \sim \mu_i$, for μ_i the posterior at iteration i, and then conditioning on $u'(t_i) = f(u_i)$. Both approaches claim to be Bayesian, and both involve a prior and a Bayesian conditioning procedure. Yet, the work of this thesis will show that *neither* approach has a strictly Bayesian interpretation. Thus, it is clearly the case that a rigorous definition of a *Bayesian* PNM is required.

Similarly, for PDEs, Cockayne et al. [2016] proposed a method for linear PDEs based upon conditioning a Gaussian prior that will be presented in Chapter 5, while Conrad et al. [2017] proposed a method based upon perturbing basis functions in classical solvers with a small amount of noise, based upon the known convergence order of the solver. There is an intuitive and fundamental distinction between these two approaches, and yet no framework in which they can be compared yet exists.

A more profound issue is the question of when PNM can be *composed* and still yield a meaningful output. Computer models often involve the solution of multiple interlinked systems of equations, often using distinct numerical methods. For an example in computational biology, see Niederer et al. [2011]. The analysis of error in the composed system can be nontrivial [Babuška and Söderlind, 2018]. This has often been listed as a potential area of high impact for PNM [see e.g. Hennig et al., 2015; Conrad et al., 2017; Cockayne et al., 2019a], owing to the richer description of error that they provide. However, conditions under which PNM, even those with a Bayesian interpretation, can be composed meaningfully have yet to be elicited, and again turn out to be surprisingly nontrivial.

1.4 Outline of the Thesis

The remainder of the thesis proceeds as follows. In the rest of this part, Chapter 2 provides the necessary mathematical and statistical background for the developments in later chapters, and Chapter 3 introduces an intuitive definition of a BPNM, leaving technical details for later in the thesis.

In Part II, BPNM are explored in a *conjugate* setting, with Gaussian priors and linear information. Chapter 4 introduces a PNM for the solution of finite-dimensional linear systems, while Chapter 5 focuses on partial differential equations.

Part III departs from the conjugate setting. In Chapter 6 we describe conditions for the existence of a posterior distribution for a generic prior and potentially nonlinear information, as well as introducing connections to decision theory and algorithms for sampling from these intractable posteriors. In Chapter 7 we explore the composition of PNM, and introduce conditions under which composed BPNM yield a distribution with a rigorous Bayesian interpretation. Finally, Chapter 8 summarises the contributions and discusses important next steps for this emerging field.

Chapter 2

Background

> "Mathematics is a game played according to certain simple rules with meaningless marks on paper."
>
> —*David Hilbert*

This chapter provides background essential for the thesis. Note that some additional basic background material is included in Appendix A. The chapter proceeds as follows. In Section 2.1 we introduce some of the basic notation required. In Section 2.2 we introduce some of the relevant PDE theory for the thesis and discuss numerical solution of PDEs, then Section 2.3 introduces relevant probability theory. Lastly, Section 2.4 introduces the idea of a *Bayesian inverse problem* and presents an example of a PDE-constrained inverse problem that will serve as a test problem in several sections of this thesis.

2.1 Notation

$\mathbb{R}$ will denote the set of all real numbers. Vectors in $\mathbb{R}^d$ will usually be denoted with bold, lower-case Latin symbols, i.e. $\boldsymbol{x} \in \mathbb{R}^d$. Throughout, we will assume that $D \subseteq \mathbb{R}^d$ for some $d < \infty$, and D will be taken to be an open set with boundary ∂D. The notation $\overline{D} = D \cup \partial D$ will be used. We use the notation $L^p(D)$ to denote the set of all functions $u : D \to \mathbb{R}$ with the property that $\|u\|_p < \infty$, where

$$\|u\|_p = \left(\int_D |u(\boldsymbol{x})|^p \, \mathrm{d}\boldsymbol{x} \right)^{\frac{1}{p}} .$$

Functions will usually be denoted with lower-case Latin symbols, i.e. $u \in L^p(D)$. When $p = \infty$, let $\|u\|_\infty = \sup_{x \in D} |u(x)|$; thus $L^\infty(D)$ is a space consisting of all bounded functions. The notation $C(D)$ will be used to denote the set of all continuous functions on D. Similarly $C^n(D)$ denotes the set of all continuous functions with n continuous derivatives on D. Occasionally we will make use of L^p spaces with respect to a measure other than the Lebesgue measure. Let μ denote some measure on D; then $L^p(D, \mu)$ is the set of all functions $u : D \to \mathbb{R}$ with the property that $\|u\|_{p,\mu} < \infty$, where

$$\|u\|_{p,\mu} = \left(\int_D |u(\boldsymbol{x})|^p \, \mu(\mathrm{d}\boldsymbol{x}) \right)^{\frac{1}{p}} .$$

The evaluation functional will be denoted $\delta_{\boldsymbol{x}}$ for $\boldsymbol{x} \in D$; for any function $f : D \to \mathbb{R}$, $\delta_{\boldsymbol{x}}(f) = f(\boldsymbol{x})$.

For a sequence $u = (u_i)$, $\|u\|_p = (\sum_{i=1}^{\infty} |u_i|^p)^{\frac{1}{p}}$. The space ℓ^p is the set of all sequences u with $\|u\|_p < \infty$. When $p = \infty$, ℓ^∞ is the set of all bounded sequences. Both the spaces $L^2(D)$ and ℓ^2 have the special property that they are *Hilbert spaces*.

2.2 Partial Differential Equations

2.2.1 Numerical Solution of PDEs

The solution of PDEs is a fundamental task in numerical analysis, that is of interest across the applied sciences. So many processes are modelled using PDEs that attempting to provide an exhaustive list is an impossible task. The canonical text on the theoretical analysis of PDEs is Evans [2010]. Brezis and Browder [1998] provide a brief history of the analysis of PDEs, noting that they are used modelling the physics of such diverse quantities as "vibrating strings, elasticity, the Newtonian gravitational field of extended matter, electrostatics, fluid flows,... heat conduction, electricity and magnetism." As such, their importance can hardly be overstated, and it is no surprise that many computer models have PDEs as their backbone. Nevertheless, as discussed in Chapter 1, PDEs of interest seldom admit closed-form solutions, and so the numerical solution of PDEs is associated with an equally vast body of research as their theory and applications.

A surprisingly nuanced point is what is meant by the *solution* of a PDE. For reference, consider again the following elliptic PDE from Eq. (1.1):

$$\begin{aligned} -\nabla \cdot \kappa(\boldsymbol{x}) \nabla u(\boldsymbol{x}) &= g(\boldsymbol{x}) && \boldsymbol{x} \in D \\ u(\boldsymbol{x}) &= b(\boldsymbol{x}) && \boldsymbol{x} \in \partial D \end{aligned} \tag{2.1}$$

The most intuitive meaning of "solution" here would be a function, $u(\boldsymbol{x})$, which satisfies the governing equations at each point $\boldsymbol{x} \in \overline{D}$. Such a solution is known as a *strong* solution, and Eq. (2.1) is referred to as the *strong form* of the PDE. However, this definition proves to be restrictive for many systems which are nevertheless of significant practical interest. To borrow an example from Evans [2010, Section 1.3], the following PDE is often used to describe the propagation of shock waves:

$$\frac{\partial u}{\partial t} + \nabla_{\boldsymbol{x}}(F(u)) = 0$$

where $F(u)$ is a potentially nonlinear function of u. The shock wave is represented by a discontinuity in u that lies on a Lebesgue-null set of the domain D (for example, a line or curve if $D = \mathbb{R}^2$), and so there are locations $\boldsymbol{x} \in D$ at which $\nabla_{\boldsymbol{x}} u$ is not defined, rendering a strong solution meaningless. As a result, a solution to the system is often sought which is *distributional* in nature, in the sense that it satisfies the governing equations *almost-everywhere* with respect to the Lebesgue measure.

Suppose that $u \in \mathcal{B}(D)$ for some separable Banach space $\mathcal{B}(D)$, and let $\mathcal{V} := \{\varphi_i\}_{i\in\mathbb{N}}$ denote a basis of $\mathcal{B}(D)$. Then the *weak form* of Eq. (2.1) is obtained by post-multiplying the equations by φ_i and then integrating over the domain:

$$\begin{aligned} -\int_D \nabla \cdot \kappa(\boldsymbol{x}) \nabla u(\boldsymbol{x}) \varphi_i(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} &= \int_D g(\boldsymbol{x}) \varphi_i(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \qquad && \boldsymbol{x} \in D \\ \int_{\partial D} u(\boldsymbol{x}) \varphi_i(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} &= \int_{\partial D} b(\boldsymbol{x}) \varphi_i(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \qquad && \boldsymbol{x} \in \partial D. \end{aligned} \tag{2.2}$$

A solution $u(\boldsymbol{x}) = \sum_{i\in\mathbb{N}} u_i \varphi_i(\boldsymbol{x})$ which satisfies Eq. (2.2) for all $i \in \mathbb{N}$ is known as a *weak solution* to the PDE.

Today there exist a plethora of numerical schemes for approximating the solution of such equations, among the most well-known of which are finite difference methods (FDM) [T. and Smith, 1987] and finite element methods (FEM) [Zienkiewicz et al., 2013; Mitchell, 1988]. Each of these approaches discretises the domain in some sense. In the case of FDM, this is by constructing a regular grid of points and approximating derivatives using finite differences. This results in a finite-dimensional linear system of equations that can be solved to produce an approximate solution to the strong form of the PDE on the grid.

FEM instead discretise the weak form of the PDE, by first dividing the domain into cells or volumes in a procedure referred to as "meshing". The solution is then approximated using a procedure known as *Galerkin's method*, which involves selecting an appropriate $\mathcal{V}_m \subset \mathcal{V}$ with $\dim(\mathcal{V}_m) = m < \infty$ and solving Eq. (2.2) in $\mathcal{V}_m$ instead of in $\mathcal{V}$. In the case of a linear PDE, this once again yields a linear system of equations that can be solved to produce the coefficients of a projection of the solution into $\mathcal{V}_m$. The space $\mathcal{V}_m$ is typically constructed by defining a basis that is compactly supported on the cells of the mesh. Then, if the PDE is governed by a compact linear operator, the resulting linear system is sparse.

2.2.2 Sobolev Spaces

The amount of PDE theory required for the thesis is limited, however *Sobolev spaces* are of significant importance and will be introduced in this section. For a more detailed treatment, see Leoni [2017] for an accessible introduction and Demengel and Demengel [2012] for the general case.

Central to the concept of a Sobolev space is a weak derivative, a derivative that exists in the same weak sense as the weak solution described in the previous section. The weak derivative allows definition of the Sobolev norm, and in turn a Sobolev space.

Definition 2.2.1 (Weak Derivative)**.** Let $D \subseteq \mathbb{R}^d$ be open, $\boldsymbol{\alpha} \in \mathbb{N}^d$, and let $|\boldsymbol{\alpha}| = \sum_{i=1}^d \alpha_i$. The vector $\boldsymbol{\alpha}$ is known as a *multi-index.* Let

$$\partial^{\boldsymbol{\alpha}} = \frac{\partial^{|\boldsymbol{\alpha}|}}{\partial x_1^{\alpha_1} \cdots \partial x_d^{\alpha_d}}.$$

For a function $u : D \to \mathbb{R}$, the function $v : D \to \mathbb{R}$ is known as the $\boldsymbol{\alpha}$*-weak-derivative* of u if it holds that

$$\int_D \phi(\boldsymbol{x}) \partial^{\boldsymbol{\alpha}} u(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} = (-1)^{|\boldsymbol{\alpha}|} \int_D \partial^{\boldsymbol{\alpha}} \phi(\boldsymbol{x}) v(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}$$

for all $\phi \in C^\infty(D)$ such that ϕ is supported on a compact subset of D. We will use the notation $D^{\boldsymbol{\alpha}} u = v$.

Definition 2.2.2 (Sobolev norm)**.** Fix $k, p \in \mathbb{N}$, For a function $u \in L^p(D)$, the *Sobolev norm* of u is defined as

$$\|u\|_{k,p} \coloneqq \sum_{|\boldsymbol{\alpha}| \leq k} \|D^{\boldsymbol{\alpha}} u\|_p.$$

Sometimes the norm

$$\|u\|_{k,p} = \left(\sum_{|\boldsymbol{\alpha}| \leq k} \|D^{\boldsymbol{\alpha}} u\|_p \right)^{\frac{1}{|\boldsymbol{\alpha}|}}$$

is used, rather than that in Definition 2.2.2, however this norm is equivalent[1].

Definition 2.2.3 (Sobolev Space)**.** The *Sobolev Space* $\mathcal{W}^{k,p}(D)$ is defined as:

$$\mathcal{W}^{k,p}(D) \coloneqq \{u \in L^p(D) : \|u\|_{k,p} < \infty\}.$$

The space $\mathcal{W}^{k,p}(D)$ is separable whenever $p < \infty$ and is a Hilbert space whenever $p = 2$. The notation $\mathcal{W}^{k,2}(D) = \mathcal{H}^k(D)$ will be used in this special case.

2.3 Probability Theory

We now turn to an exposition of essential concepts from probability theory. We begin this section by establishing some notation. The notation established here is introduced more thoroughly in Appendix A.2.

For a measurable space $\mathcal{X}$ equipped with σ-algebra $\mathcal{B}_{\mathcal{X}}$, the notation $\mathcal{P}_{\mathcal{X}}$ will be used to denote the set of all probability measures on $\mathcal{X}$. Probability measures will also sometimes be called *distributions*. Typically the σ-algebra used will be the Borel σ-algebra. For any $B \in \mathcal{B}_{\mathcal{X}}$, the notation $\mathbb{I}[B] : \mathcal{X} \to \{0, 1\}$ will be used to denote an indicator function on B. We will generally denote probability measures using Greek letters, i.e. $\mu, \nu \in \mathcal{P}_{\mathcal{X}}$. Absolute continuity of μ with-respect-to ν is denoted $\mu \ll \nu$. For functions $f : \mathcal{X} \to \mathbb{R}$ and measures $\mu \in \mathcal{P}_{\mathcal{X}}$ we will use the notation

$$\mu(f) \coloneqq \int_{\mathcal{X}} f(x)\, \mu(\mathrm{d}x).$$

The notation $\delta(x)$ will be used to denote a Dirac measure on the point $x \in \mathcal{X}$.

For a map $T : \mathcal{X} \to \mathcal{Y}$ the associated pushforward of the measure $\mu \in \mathcal{P}_{\mathcal{X}}$ is denoted $T_{\#}\mu$, and defined as $[T_{\#}\mu](B) = \mu(T^{-1}B)$, for each B in the image of $\mathcal{B}_{\mathcal{X}}$ under T. Random variables will be denoted using capital letters, i.e. X, U. When X, Y, Z are random variables on a space $\mathcal{X}$, conditional independence of X and Y given Z is denoted $X \perp\!\!\!\perp Y | Z$.

2.3.1 Construction of Measures on Function Spaces

Another important concept for this thesis is the construction of measures on function spaces, which will now be introduced. The natural setting for probability measures on function spaces is separable Banach spaces, for reasons discussed in detail in Dashti and Stuart [2017], and so we restrict attention to construction on such spaces

[1] In the sense that it induces the same topology on the Sobolev spaces introduced next.

here. To this end, assume that $\mathcal{X}$ is a separable Banach space, and let $\{\phi_i\}_{i\in\mathbb{N}}$ denote a basis of $\mathcal{X}$. Then, it holds that any $u \in \mathcal{X}$ can be represented as

$$u = \sum_{i=1}^{\infty} u_i \phi_i$$

for some sequence of coefficients (u_i). A distribution on $\mathcal{X}$ is obtained by randomising the coefficients, i.e. by defining the random variable

$$U = u_0 + \sum_{i=1}^{\infty} \gamma_i \xi_i \phi_i. \tag{2.3}$$

Here (ξ_i) is a sequence of IID random variables with mean zero, while (γ_i) is a second sequence introduced to ensure that the series converges almost-surely. The element $u_0 \in \mathcal{X}$ is some element of $\mathcal{X}$ to allow for distributions with nonzero expectation.

Different random variables (ξ_i) result in different function-space distributions, where the ℓ^p space in which (γ_i) must lie to ensure almost-sure convergence depends on the choice of (ξ_i). *Uniform distributions* are obtained when $\xi_i \sim U[-1, 1]$ and $(\gamma_i) \in \ell^1$ [Dashti and Stuart, 2017, Section 2.2]. *Gaussian distributions* are obtained when $\xi_i \sim \mathcal{N}(0, 1)$ and $(\gamma_i) \in \ell^2$. *Cauchy distributions* arise[2] when $\xi_i \sim \mathrm{Cauchy}(0, 1)$ and $(\gamma_i) \in \ell^2$.

Generally speaking, this view of constructing measures on function spaces requires that Eq. (2.3) be *truncated* when used in computation, that is, computation proceeds based on the following random variable defined on a finite-dimensional subspace of $\mathcal{X}$:

$$U^N = u_0 + \sum_{i=1}^{N} \gamma_i \xi_i \phi_i. \tag{2.4}$$

However, in the case of a Gaussian distribution an important alternate view of the distribution arises.

2.3.2 Gaussian Measures

For a Gaussian distribution it is possible to choose the γ_i and ϕ_i in such a way as to yield a more tractable distribution, commonly referred to as a *Gaussian process* (GP). While the term "Gaussian process" can be used to refer to *any* Gaussian measure, in this work it is used to refer to the presentation in this section. Comprehensive works on GPs include Bogachev [1998] and Rasmussen and Williams [2006];

[2] In fact, this holds only for the case for $\mathcal{B}$ a *Hilbert* space; the case for general Banach spaces is more complicated, see Sullivan [2017].

the former gives a measure-theoretic treatment, while the latter is more practical.

To define a GP we must first introduce some notation and definitions. For an arbitrary domain D, let $X = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_n\} \subset D$ and $X' = \{\boldsymbol{x}'_1, \dots, \boldsymbol{x}'_{n'}\} \subset D$. Then, for a function $m : D \to \mathbb{R}$, $m(X) \in \mathbb{R}^n$ is the vector with

$$[m(X)]_i = m(\boldsymbol{x}_i).$$

Similarly, for a function $k : D \times D \to \mathbb{R}$, $k(X, X') \in \mathbb{R}^{n \times n'}$ is the matrix with

$$[k(X, X')]_{ij} \coloneqq k(\boldsymbol{x}_i, \boldsymbol{x}'_j).$$

The matrix $k(X, X)$ is sometimes referred to as $k(X)$.

Definition 2.3.1 (Positive-definite function)**.** The function $k : D \times D \to \mathbb{R}$ is said to be *positive-definite* if, for any finite set $X \subset D$, $k(X, X)$ is a positive-definite matrix. Similarly, k is said to be *positive-semidefinite* if $k(X, X)$ is a positive-semidefinite matrix for each X.

Definition 2.3.2 (Gaussian process)**.** A random variable U on a domain D is said to be a *Gaussian process* if there exists a function $m : D \to \mathbb{R}$ and a symmetric, positive-definite kernel $k : D \times D \to \mathbb{R}$ such that, for any set $X = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_n\} \subset D$, it holds that

$$U(X) \sim \mathcal{N}(m(X), k(X))$$

where $U(X) = [U(\boldsymbol{x}_1, \dots, \boldsymbol{x}_n)]^\top$. We use the notation $U \sim \mathcal{GP}(m, k)$. The function m is referred to the *mean* of the GP, while k is referred to as its *covariance function* or *kernel*.

This view of GPs is more tractable than that in Eq. (2.3) because the only truncation error in the representation of the process is through the fact that only a finite number of evaluation locations X can be stored in memory. However, for each finite set X the finite-dimensional marginals $U(X)$ can be computed *exactly* in this formulation.

GPs have many other convenient properties. Much like their finite-dimensional counterparts, there is an explicit formula for the projection of a Gaussian process through an arbitrary bounded linear operator. Let $\mathcal{L}$ be a bounded linear operator, and assume that m lies in the domain of $\mathcal{L}$ while k lies in the domain of $\mathcal{L}\bar{\mathcal{L}}$, where $\bar{\mathcal{L}}$ denotes the adjoint of $\mathcal{L}$. Then, it holds that

$$[\mathcal{L}U](X) \sim \mathcal{GP}(\mathcal{L}m, \mathcal{L}\bar{\mathcal{L}}k).$$

While this involves application of the adjoint of $\mathcal{L}$, note that the application of $\bar{\mathcal{L}}$ to a positive-definite bivariate function $k(\boldsymbol{x}, \boldsymbol{x}')$ is equivalent to applying $\mathcal{L}$ to the *second* argument of k, rather than the first; i.e. if $D = \mathbb{R}$ and $\mathcal{L} = \frac{\mathrm{d}}{\mathrm{d}x}$:

$$\overline{\mathcal{L}}k(x, x') = \frac{\mathrm{d}}{\mathrm{d}x'}k(x, x')$$

A last significant property of GPs for this thesis is the presence of a closed-form conditioning formula; that is, for any bounded linear operator $\mathcal{L}$ with finite-dimensional codomain, $U | \mathcal{L}U = \boldsymbol{y}$ is again a Gaussian process provided U is supported on the domain of $\mathcal{L}$. This formula is critical for the material developed in Chapter 5, and will be presented in detail there.

Reproducing Kernel Hilbert Spaces

An object of profound importance in the analysis of Gaussian measures is the *reproducing kernel Hilbert space* (RKHS) associated with its covariance function. This concept will now be introduced. The definitions in this section follow Berlinet and Thomas-Agnan [2004], and proofs of the stated theorems can be found therein, in Chapters 2 and 3.

Definition 2.3.3 (Reproducing Kernel Hilbert Space)**.** A separable Hilbert space $\mathcal{H}(D)$ with inner product $\langle\,\cdot\,,\,\cdot\,\rangle$ is said to be a *reproducing kernel Hilbert space* (RKHS) if there exists a function $k : D \times D \to \mathbb{R}$ with the following properties:

1. For all $\boldsymbol{x} \in D$, $k(\,\cdot\,, \boldsymbol{x}) \in \mathcal{H}(D)$.
2. For all $\boldsymbol{x} \in D$, $u \in \mathcal{H}(D)$, $\langle u, k(\,\cdot\,, \boldsymbol{x})\rangle = u(\boldsymbol{x})$.

We say that k is a *reproducing kernel* associated with $\mathcal{H}(D)$.

Theorem 2.3.4. *A Hilbert space has a reproducing kernel if and only if all evaluation functionals are continuous on $\mathcal{H}(D)$.*

Theorem 2.3.4 reveals that the reproducing kernel of an RKHS is in fact the *representer* of the evaluation operator in that space, in the sense of the Riesz representation theorem [Demengel and Demengel, 2012, Theorem 1.38]. Indeed, RKHS can be characterised as Hilbert spaces in which the evaluation operator is continuous. The following theorem is due to Moore–Aronszajn, and identifies all positive definite functions with an RKHS; see also Berlinet and Thomas-Agnan [2004, Chapter 3].

Theorem 2.3.5 (Moore–Aronszajn Theorem)**.** *For a function $k : D \times D \to \mathbb{R}$, the following two statements are equivalent:*

1. *k is a positive semidefinite function.*
2. *There exists an RKHS with reproducing kernel k.*

This justifies a notation that will frequently be used in this thesis: we will often emphasise the kernel k associated with an RKHS with the notation $\mathcal{H}_k(D)$. Similarly, the inner product associated with this RKHS will often be denoted $\langle \cdot, \cdot \rangle_k$ and the norm $\| \cdot \|_k$. Note that, since a vector space may be endowed with many inner product structures that bestow a Hilbert structure, the reproducing kernel of an RKHS depends on the inner product used. Thus, the same *set* of functions can be the underlying set associated with many RKHS with different kernels and inner product structures.

The Moore–Aronszajn theorem provides a much needed connection between RKHS and GPs. Since each GP is associated with a positive-definite covariance function, each GP is *also* associated with an RKHS. For a GP $\mathcal{GP}(m, k)$, the RKHS $\mathcal{H}_k(D)$ is often referred to as the *native space* or *Cameron–Martin Space* of the GP. The Cameron–Martin space of a GP can alternately be characterised as the set of functions by which a Gaussian measure can be translated to obtain an equivalent measure.

Theorem 2.3.6 (Theorem 2.4.5 of Bogachev [1998])**.** *Let $\mu = \mathcal{GP}(m, k)$ and let μ_h be such that, if U is a random variable with law μ, μ_h is the law of $U + h$. Then $\mathcal{H}_k(D)$ can be characterised as*

$$\mathcal{H}_k(D) = \{h \in \mathcal{X} : \mu_h \ll \mu\}.$$

Note that the RKHS associated with a GP is not the same as the set of functions on which the GP is *supported*. In fact, we have the following theorem:

Theorem 2.3.7 (Theorem 2.4.7 of Bogachev [1998])**.** *Let $\mu \sim \mathcal{GP}(m, k)$ and let $\mathcal{H}_k(D)$ denote the RKHS associated with k. Then it holds that if $\mathcal{H}_k(D)$ is infinite-dimensional, then it is a null-set of μ, i.e. $\mu(\mathcal{H}_k(D)) = 0$.*

A last important property of kernels that needs to be introduced in this section is a *Mercer kernel.*

Definition 2.3.8 (Mercer Kernel)**.** A function $k : D \times D \to \mathbb{R}$ is said to be a *Mercer kernel* if k is continuous, symmetric and positive semi-definite.

The importance of a Mercer kernel lies in Mercer's theorem, which provides a representation of the kernel in terms of eigenvalues and eigenfunctions of its associated integral operator, and is of significant theoretical importance. The following theorem was originally due to Mercer [1909] and extended in Steinwart and Scovel [2012]. The version below is sufficiently general for this thesis, and is a synthesis of results from Steinwart and Scovel [2012, Lemma 2.12] and Sullivan [2015, Theorem 11.3].

Theorem 2.3.9. *Let $k : D \times D \to \mathbb{R}$ be a Mercer kernel with RKHS $\mathcal{H}_k(D)$, and assume that k is bounded. Consider the operator T_k, defined as*

$$T_k(f)(x') := \int_D f(x)k(x, x')\, \mathrm{d}x.$$

Then there is a sequence of eigenfunctions (e_i) and eigenvalues λ_i, $i \in \mathbb{N}$ such that:

- *The eigenfunctions are such that $e_i \in L_2(D)$ are $L_2(D)$-orthonormal; $\langle e_i, e_j \rangle_2 = \delta_{ij}$.*
- *When ordered, the eigenvalues $\lambda_i \in \mathbb{R}^+$ are non-negative and convergent to zero.*
- *The set $\{\sqrt{\lambda_i} e_i\}$, $i \in \mathbb{N}$ is an orthonormal basis of $\mathcal{H}_k(D)$.*

The following result, again from Sullivan [2015, Theorem 11.3], characterises k in terms of the eigenvalues and eigenvectors from Theorem 2.3.9.

Theorem 2.3.10 (Mercer's Theorem)**.** *Let $k : D \times D \to \mathbb{R}$ be a bounded Mercer kernel. Let (e_i), (λ_i) be the eigenfunctions and eigenvectors from Theorem 2.3.9. Then, for all $x, x' \in D$ it holds that*

$$k(x, x') = \sum_{i=1}^{\infty} \lambda_i e_i(x) e_i(x')$$

where convergence is absolute and uniform.

The last result in this section concerns the ability to represent stochastic processes using an expansion based on the eigendecomposition of their covariance function, when it is a Mercer kernel. The theorem was discovered independently in Karhunen [1947] and Loève [1978], but the presentation below is as in Sullivan [2015, Theorem 11.4].

Theorem 2.3.11 (Karhunen–Loève Theorem)**.** *Let $U : D \to \mathbb{R}$ be a square-integrable random variable with mean zero, and a covariance function that is a Mercer kernel. Then we have that*

$$U(x) = \sum_{i=1}^{\infty} Z_i \phi_i(x)$$

where the ϕ_i are the eigenfunctions of the covariance operator of U, while the Z_i are given by

$$Z_i := \int_D U(x)\phi_i(x)\mathrm{d}x. \tag{2.5}$$

Furthermore, $\mathbb{E}(Z_i) = 0$ and $\mathbb{E}(Z_i Z_j) = \lambda_i \delta_{ij}$, for δ_{ij} the Kronecker delta.

As a last remark, since the projection in Eq. (2.5) is linear, it holds that Z_i is Gaussian distributed if U is Gaussian.

Prior Mean and Covariance

The selection of a prior mean and covariance function are of course critical; in particular because the prior covariance determines the RKHS associated with the GP, and thus describes the smoothness properties of functions that are in the support of the distribution. Some important choices for this thesis will now be discussed. All of the covariance functions introduced in this section are *stationary*, in that the are of the form $k(\boldsymbol{x}, \boldsymbol{x}') = k(\|\boldsymbol{x} - \boldsymbol{x}'\|)$ for some norm. In this thesis, the norm adopted is the Euclidean norm, meaning that all covariance functions introduced are also *isotropic*. This assumption can naturally be relaxed, and the covariance functions introduced here represent only a few of the most widely used covariance functions known in the literature; for more information see Berlinet and Thomas-Agnan [2004] and Fasshauer [2007, Appendix D]. Duvenaud [2014] describes how covariance functions can be composed to yield new covariance functions.

Squared Exponential Covariance The *squared exponential* or *exponentiated quadratic* covariance function is given by

$$k(\boldsymbol{x}, \boldsymbol{x}'; \sigma, \ell) := \sigma^2 \exp\left(-\frac{1}{2\ell^2}\|x - x'\|_2^2\right). \tag{2.6}$$

This choice has proven popular in the machine learning and applied statistics communities owing to its tractability. The parameter $\sigma \in \mathbb{R}^+$ controls the *amplitude* of the prior, while $\ell \in \mathbb{R}^+$ is known as the *length-scale* and controls the scale on which functions drawn from the distribution vary. Functions in the support of Gaussian distributions with this covariance function are in $C^\infty(D)$.

Matérn Covariance Functions The family of *Matérn* covariance functions are also widely used, particularly when the high level of smoothness given by the squared exponential covariance function is undesirable. These covariance functions are given by

$$k(\boldsymbol{x}, \boldsymbol{x}'; \sigma, \ell, \nu) := \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu} \frac{\|\boldsymbol{x} - \boldsymbol{x}'\|_2}{\ell} \right)^\nu K_\nu \left(\sqrt{2\nu} \frac{\|\boldsymbol{x} - \boldsymbol{x}'\|_2}{\ell} \right). \tag{2.7}$$

Here $\Gamma(\,\cdot\,)$ is the Gamma function while $K_\nu(\,\cdot\,)$ is the modified Bessel function of the second kind. The parameters $\sigma, \ell \in \mathbb{R}^+$ are amplitude and length-scale parameters, as before, while $\nu \in \mathbb{R}^+$ is a smoothness parameter. Functions in the support of a Gaussian distribution with a Matérn covariance with smoothness parameter ν will have $\lceil \nu \rceil - 1$ derivatives. The squared exponential covariance function arises as a limiting case of the Matérn covariance, in the limit as $\nu \to \infty$. Commonly members of the Matérn family for specific values of ν are referred to as *Matérn-ν* covariance functions.

The functional form in Eq. (2.7) can be simplified when $\nu = p + \frac{1}{2}$ for some $p \in \mathbb{N}$. The general form is unimportant for this thesis and can be found in Rasmussen and Williams [2006, Section 4.2], however since the case of $p = 2$ is used in several of the experiments, this is presented here:

$$k\left(\boldsymbol{x}, \boldsymbol{x}'; \sigma, \ell, \frac{5}{2}\right) := \sigma^2 \left(1 + \sqrt{5}\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|_2}{\ell} + \frac{5}{3}\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|_2^2}{\ell^2}\right) \exp\left(-\sqrt{5}\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|_2}{\ell}\right). \tag{2.8}$$

Wendland Covariance Functions The Wendland covariance functions are again a family of covariance functions with varying smoothness. Compared to the Matérn family, however, the Wendland functions are both piecewise polynomial and compactly supported. However, the form of these functions now also depends upon the dimension of the input domain. Since in this thesis the dimension of the domain D of covariance functions will never exceed 2, to ensure appropriate differentiability the covariance functions for $\dim(D) = 3$ were used throughout. The forms presented below are those used in this thesis, from Fasshauer [2007, Appendix D]; these are strictly positive definite for any $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^3$:

$$k_0(\boldsymbol{x}, \boldsymbol{x}'; \sigma, \epsilon) := \sigma \max\left(0, 1 - \frac{\|\boldsymbol{x} - \boldsymbol{x}'\|_2}{\epsilon}\right)^2 \tag{2.9}$$

$$k_1(\boldsymbol{x}, \boldsymbol{x}'; \sigma, \epsilon) := \sigma \max\left(0, 1 - \frac{\|\boldsymbol{x} - \boldsymbol{x}'\|_2}{\epsilon}\right)^4 \left(\frac{4\|\boldsymbol{x} - \boldsymbol{x}'\|_2}{\epsilon} + 1\right). \tag{2.10}$$

The covariance function k_0 results in a measure supported on functions in $C^0(D)$, while k_1 gives a measure supported on twice differentiable functions. The parameter $\sigma \in \mathbb{R}^+$ again controls the amplitude, while $\epsilon \in \mathbb{R}^+$ is analogous to the length-scale parameter for the other kernels introduced. The kernels above have support wherever $\|\boldsymbol{x} - \boldsymbol{x}'\|_2 < \epsilon$.

Kernel Parameters The covariance functions introduced in this section each depend on parameters that have an enormous impact on the properties of the distribution, and so careful treatment of them receives an enormous amount of attention in the literature on GPs. In this section we describe two treatments of those parameters. For more detail see Rasmussen and Williams [2006, Chapter 5].

Perhaps the most widely followed approach, particularly in the machine learning community, is to estimate these parameters by maximising the marginal likelihood of the data on which the process will be conditioned, as a function of the parameters. This process is often called *empirical Bayes.* A second approach pursued in this thesis is to consider the hyperparameters as additional parameters to be learned from the data. These parameters can then be endowed with "hyperpriors", and their posteriors can be determined in the Bayesian framework. From a statistical perspective this is appealing, as the parameters can then be marginalised in the posterior distribution to obtain inferences that are independent of a particular assumed value. The downside of this approach is that the attractive conjugacy properties of the Gaussian process are generally not maintained when parameters of the kernel are treated in this way, so that a far more computationally expensive inference procedure is required to sample from the joint posterior over the joint distribution of the function u and the parameters of the prior.

2.4 Bayesian Inverse Problems

In this section, Bayesian inverse problems will be rigorously introduced. The presentation follows the seminal work of Stuart [2010]; Dashti and Stuart [2017] also provides a thorough and accessible treatment.

As in Section 1.2.2, we we will introduce Bayesian inverse problems by examining the abstract model problem

$$\boldsymbol{y} = \mathcal{G}(\theta^{\dagger}) + \xi. \tag{2.11}$$

Here $\mathcal{G} : \Theta \to \mathcal{Y}$, where Θ and $\mathcal{Y}$ are each separable Banach spaces equipped with their respective Borel σ-algebras, and ξ is a random variable supported on $\mathcal{Y}$ with law ν_0. The goal is to reconstruct $\theta^{\dagger} \in \Theta$ from the noisy observations $\boldsymbol{y} \in \mathcal{Y}$. We will make no assumptions on the dimensionality of Θ, and in particular the case $\dim(\Theta) = \infty$ is an important one for this thesis. However, to eliminate some technical detail we will assume that $\dim(\mathcal{Y}) < \infty$, and it will usually be assumed that $\mathcal{Y} = \mathbb{R}^d$ for some $d < \infty$. Similarly, it will often be the case that in fact the domain and image of $\mathcal{G}$ are subsets of Θ, $\mathcal{Y}$ respectively, but this amounts to small additional technical detail not introduced here.

The traditional notion of Bayes theorem introduced in Eq. (1.2) is not appropriate when $\dim(\Theta) = \infty$, as an equivalent of the Lebesgue density does not exist in such settings. Instead, inference must be performed with respect to some other, well-defined reference measure. The natural reference measure in Bayesian inference problems is the prior, and this is the approach followed in Stuart [2010]. Let μ denote a prior measure on Θ. Introduce the *translated* random variable $\xi_\theta = \xi + G(\theta)$, and denote the law of ξ_θ by ν_θ. We assume that $\nu_\theta \ll \nu_0$ for μ-almost-all $\theta \in \Theta$; thus, by the Radon–Nikodym theorem we have that:

$$\frac{\mathrm{d}\nu_\theta}{\mathrm{d}\nu_0}(\boldsymbol{y}) = \exp(-\Phi(\theta; \boldsymbol{y}))$$

for some function $\Phi : \Theta \times \mathcal{Y} \to \mathbb{R}$. This function is referred to as the *potential* in the Bayesian inversion literature, and is commonly known as the *negative log-likelihood* in Bayesian statistics. The function $\exp(-\Phi(\theta; \boldsymbol{y}))$ is referred to as the *likelihood*.

Under appropriate measurability assumptions on Φ [see Dashti and Stuart, 2017, section 3.2], the posterior measure $\mu^{\boldsymbol{y}}$ can be defined through its Radon–Nikdoym derivative with respect to μ:

Theorem 2.4.1 (Bayes Theorem)**.** *Let*

$$Z^{\boldsymbol{y}} = \int_{\Theta} \exp(-\Phi(\theta; \boldsymbol{y}))\mu(\mathrm{d}\theta).$$

Assume that $Z^{\boldsymbol{y}} > 0$, for almost-all[3] *$\boldsymbol{y} \in \mathcal{Y}$. Then, the conditional distribution $\mu^{\boldsymbol{y}}$*

[3] This “almost-all” statement, and the others in this theorem, are in fact with respect to the product measure of μ and ν_u; see Stuart [2010].

exists for almost-all $\boldsymbol{y} \in \mathcal{Y}$*, we have that* $\mu^{\boldsymbol{y}} \ll \mu$*, and*

$$\frac{\mathrm{d}\mu^{\boldsymbol{y}}}{\mathrm{d}\mu}(\theta) = \frac{1}{Z^{\boldsymbol{y}}} \exp(-\Phi(\theta; \boldsymbol{y})).$$

This notion of a Bayesian posterior is well-defined even in the case of infinite-dimensional Θ, and so the Bayesian framework has emerged as a popular framework in which to perform inference on such quantities. As discussed in Dashti and Stuart [2017, Section 1.1], the probabilistic interpretation is an elegant way to overcome the following difficulties:

- The precise realisation of the noise η that corrupts the data is not known, so *distributional* information about η must be incorporated into the solution to the inverse problem. This induces stochasticity.

- The noise η may be such that the corrupted data is not in the image of $\mathcal{G}$, making the inversion ill-posed unless η is properly incorporated into the inversion.

- Since $\dim(\Theta) \gg \dim(\mathcal{Y})$, the inference problem is inherently underdetermined. Here stochasticity arises from the fact that, since $\theta^\dagger$ cannot be determined completely from the data, uncertainty remains about its value.

It was shown in some detail in Stuart [2010] that under suitable regularity assumptions, Bayesian inverse problems are well-posed in the sense of Hadamard [1903].

We will now present an algorithm which is well-adapted to such inference problems, before presenting a motivating example of an infinite-dimensional inference problem that will appear repeatedly in this thesis.

2.4.1 The Preconditioned Crank–Nicolson Algorithm

Markov chain Monte-Carlo (MCMC) is one of the most successful techniques for sampling from the posterior distributions that arise in Bayesian statistics. Some common MCMC techniques are outlined in Appendix A.3; for a detailed introduction see Brooks et al. [2011]. The popularity of MCMC is in part due to its flexibility; at a basic level, all that is required to apply an algorithm such as random-walk Metropolis–Hastings (RWM) is the ability to evaluate both the likelihood and prior densities at arbitrary locations in the domain[4].

[4] Though of course, to prove convergence, more rigorous and restrictive conditions are required; see Roberts and Rosenthal [2004]; Meyn and Tweedie [1993].

In Bayesian inverse problems it is common to need to sample from a distribution that is mathematically defined on an infinite-dimensional space. Although for computational purposes the distribution must be discretised in some way, a desirable property is that the sampling algorithm employed is dimensionally robust, so that the acceptance rate does not decay to zero as the discretisation is refined. It is well-known that most classical algorithms, such as RWM, do not have this property, and so the number of samples required to sample from a distribution diverges as the dimension increases [Cotter et al., 2013]. To address this, Cotter et al. [2013] introduced the *preconditioned Crank–Nicolson* (pCN) algorithm which, for Gaussian priors, has the required dimension-robustness property. This algorithm will now be introduced.

Let μ denote a Gaussian reference measure, typically the prior for the Bayesian inference problem at hand. The pCN algorithm employs proposals obtained by discretising a stochastic differential equation that is invariant for μ. For a parameter $\beta \in (0, 1)$, given a current state $\theta_n \in \Theta$ proposals are of the form:

$$\tilde{\theta}_{n+1} = \sqrt{1 - \beta^2}\theta_n + \beta\xi_n.$$

Here ξ_n is distributed according to μ. The parameter β is user-specified, and is typically tuned to achieve a target acceptance rate. Proposed moves are then accepted or rejected according to a standard RWM criterion; let $\alpha(\theta, \theta')$ be given by:

$$\alpha(\theta, \theta') = \exp(\Phi(\theta; \boldsymbol{y}) - \Phi(\theta'; \boldsymbol{y})).$$

Then, with probability $\alpha(\theta_n, \tilde{\theta}_{n+1})$ the next state in the algorithm is set to $\theta_{n+1} = \tilde{\theta}_{n+1}$; otherwise the chain remains at $\theta_{n+1} = \theta_n$. This is reported as an algorithm in Algorithm 2.1. The pCN algorithm can be generalised in several directions to produce variants of the Metropolis-adjusted Langevin algorithm (MALA) and the Hamiltonian Monte–Carlo (HMC) algorithms; see Beskos et al. [2017]. These extensions were not utilised in this thesis, however.

2.4.2 Example: Electrical Impedance Tomography

We will now introduce an important Bayesian inverse problem that serves as a test problem in several chapters of this thesis.

Electrical Impedance Tomography (EIT) is a medical imaging technique in which the interior conductivity of a patient is recovered by passing small currents through electrodes attached to the patient and measuring the induced voltages. The recovered conductivity can be used to detect abnormalities such as brain tumours;

Algorithm 2.1 pCN algorithm for sampling from $\mu^{\boldsymbol{y}}$, defined as in Theorem 2.4.1. The input θ_0 is an arbitrary initial state in the support of μ, β is a tuning parameter used to control the acceptance probability, M is the number of samples required, Φ is the potential and μ a means of sampling from the prior. The notation $\mathcal{U}(0,1)$ is used to denote a uniform distribution on $[0,1]$. The output is the samples from the posterior $\theta_1, \dots, \theta_M$; if M is sufficiently large then this can be considered a sample from $\mu^{\boldsymbol{y}}$.

```
procedure pCN($\theta_0, \beta, M, \Phi, \mu$)
    for $i = 1, \dots, M$ do
        $w_i \sim \mu$
        $\tilde{\theta}_i \leftarrow \sqrt{1-\beta^2}\theta_{i-1} + \beta w_i$
        $\log \alpha \leftarrow \Phi(\theta_{i-1}; \boldsymbol{y}) - \Phi(\tilde{\theta}_i; \boldsymbol{y})$
        $U_i \sim \mathcal{U}(0,1)$
        if $\log U_i \leq \log \alpha$ then
            $\theta_i \leftarrow \tilde{\theta}_i$
        else
            $\theta_i \leftarrow \theta_{i-1}$
        end if
    end for
return $\theta_1, \dots, \theta_M$
end procedure
```

see Holder [2004] for a detailed introduction. EIT has also been proposed as a tool for monitoring machines known as *hydrocyclones*, which are pieces of industrial machinery used for separating particulates from fluids in which they are suspended [Gutierrez et al., 2000].

As a Bayesian inverse problem, the parameter of interest is a *conductivity field* $\kappa(\boldsymbol{x})$, $\boldsymbol{x} \in D$, where D models the domain of interest; perhaps the interior of a patient, or perhaps of a hydrocyclone. As such, the parameter is a function, and so the theoretical framework for Bayesian inversion on infinite-dimensional spaces is essential. It is assumed that N_e distinct electrodes are attached to the boundary ∂D. Typically multiple patterns of currents are passed through the electrodes to increase the amount of data for the recovery, and it will be assumed that the maximum number of linearly independent stimulations is applied so that there are a total of $N_e - 1$ distinct stimulations. The pattern of data collection is then defined by a *stimulation pattern* and a *measurement pattern*.

The stimulation pattern describes which currents are applied to which electrodes, and consists of a set of applied currents, $\{C_{ij}\}$, $i = 1, \dots, N_e - 1$, $j = 1, \dots, N_e$, summarised by the matrix $C \in \mathbb{R}^{(N_e-1)\times N_e}$. The rows of this matrix correspond to distinct stimulation patterns, while the columns correspond to electrodes.

The output is a set of measured voltages $\{V_{ij}\}$, $i = 1, \dots, N_e - 1$, $j = 1, \dots, N_m$, where the row indices again correspond to the stimulation patterns, while the columns describe the measurements. Often the measurements taken are not voltage measurements at electrodes, but a voltage differential *between* electrodes, and this is captured in a *measurement pattern*, described by a matrix $M \in \mathbb{R}^{N_m \times N_e}$. Columns correspond to electrodes, and rows correspond to measurements. If the precise voltages at the electrodes (across all stimulation patterns) are summarised in a matrix $\tilde{V} \in \mathbb{R}^{(N_e-1) \times N_e}$, the actual measurements obtained from the experiments are given by $V = M\tilde{V}^\top$.

Two formulations of EIT, and its inverse problem, will be presented here. The first, referred to as the *point electrode model* (PEM), is a variant of the original formulation of the problem due to Calderón [1980] in which the voltage is assumed to be applied continuously over ∂D. The second formulation, the *complete electrode model* (CEM) of Cheng et al. [1989], uses more physically realistic boundary observations by explicitly modelling the boundary electrodes. Lastly in this section, experimental data which is used in two chapters of the thesis will be presented.

The Point Electrode Model

In the point electrode model the voltage is assumed to be applied continuously over ∂D, and the electrodes are modelled as a single point at which the applied voltage has been measured, $\boldsymbol{e}_i \in \partial V$, $i = 1, \dots, N_e$. The model is a variant of that originally posited by Calderón [1980]. For each fixed stimulation pattern $i = 1, \dots, N_e - 1$ it is given by:

$$\begin{aligned} -\nabla \cdot (\kappa(\boldsymbol{x}) \nabla u_i(\boldsymbol{x})) &= 0 & \boldsymbol{x} &\in D \\ \kappa(\boldsymbol{e_j}) \frac{\partial u_i}{\partial \boldsymbol{n}}(\boldsymbol{e}_j) &= C_{ij} & j &= 1, \dots, N_e. \end{aligned} \tag{2.12}$$

Here $\boldsymbol{n}$ denotes the outward pointing normal vector on ∂D and $\frac{\partial}{\partial \boldsymbol{n}}$ is a shorthand for the directional derivative in the direction of $\boldsymbol{n}$, i.e. $\frac{\partial}{\partial \boldsymbol{n}} = \boldsymbol{n} \cdot \nabla$. The index i has been added to the voltage field u_i to emphasise its dependence on the stimulation pattern.

For the purposes of inference, let $\tilde{U} \in \mathbb{R}^{(N_e-1) \times N_e}$ be the matrix with entries $\tilde{U}_{ij} = u_i(\boldsymbol{e}_j)$, and let $U = M\tilde{U}^\top$ Let U_i denote the i^{th} row of U, and let V_i denote the i^{th} row of V. Throughout this thesis a Gaussian measurement error model will be assumed for the voltages V_i, with noise covariance Γ. The noise is assumed to be

IID across stimulation patterns, yielding the Gaussian likelihood:

$$\begin{aligned} p(V|\kappa) &\propto \exp\left(-\Phi(V|\kappa)\right) \\ \Phi(V|\kappa) &= \frac{1}{2}\sum_{i=1}^{N_e-1} \|V_i - U_i\|_\Gamma^2 \end{aligned} \tag{2.13}$$

This formulation differs somewhat from that of Calderón [1980], which considered the problem of recovering the *Dirichlet-to-Neumann* map. For the purposes of this thesis, the PEM is often more tractable than the CEM, as the need to calculate boundary integrals is eliminated.

The Complete Electrode Model

The CEM is a more physically realistic model for EIT that was first described in Cheng et al. [1989], and has been shown to be well-posed as a Bayesian inverse problem [Dunlop and Stuart, 2016]. In the CEM, electrodes are explicitly modelled as subsets of the domain. Let E_i, $i = 1, \ldots, N_e$, be such that $E_i \subset \partial D$ and $E_i \cap E_j = \emptyset$ whenever $i \neq j$. Each electrode is additionally associated with a *contact impedance* ζ_i, $i = 1, \ldots, N_e$ which models that the connection between the electrode and the boundary is not perfect. In all applications in this thesis, the contact impedances are assumed to be known *a-priori*. The CEM is then given by:

$$\begin{aligned} -\nabla \cdot (\kappa(\boldsymbol{x})\nabla u_i(\boldsymbol{x})) &= 0 & \boldsymbol{x} &\in D \\ \int_{E_j} \kappa(\boldsymbol{x})\frac{\partial u_i}{\partial \boldsymbol{n}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x} &= C_{ij} & j &= 1, \ldots, N_e \\ \kappa(\boldsymbol{x})\frac{\partial u_i}{\partial \boldsymbol{n}}(\boldsymbol{x}) &= 0 & \boldsymbol{x} &\in \partial D \setminus \bigcup_{k=1}^{N} E_k \\ u_i(\boldsymbol{x}) + \zeta_j\kappa(\boldsymbol{x})\frac{\partial u_i}{\partial \boldsymbol{n}}(\boldsymbol{x}) &= \tilde{U}_{ij} & \boldsymbol{x} &\in E_j,\ j = 1, \ldots, N_e. \end{aligned} \tag{2.14}$$

Note that, unlike in the PEM, a solution comprises both the function $u_i(\boldsymbol{x})$ and the voltages on the electrodes $\tilde{U}_{ij}$, $j = 1, \ldots, N_e$. However the details of the likelihood model otherwise follow the description in the previous section.

Experimental Data

A description of an experimental dataset that will be used in several sections of this thesis is now provided. This data was taken from the EIDORS suite of contributed

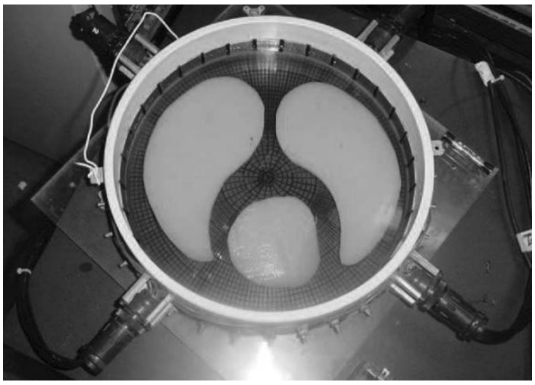


Figure 2.1: Agar targets from which the measurements described in Section 2.4.2 were obtained. The two large lung-shaped targets each have a lower conductivity than the surrounding saline, while the smaller heart-shaped target has a higher conductivity.

data[5], and is due to Isaacson et al. [2004]. To obtain the data, $N_e = 32$ equispaced electrodes were placed around the perimeter of a circular tank filled with saline solution. Three agar targets were placed into the tank, as depicted in Fig. 2.1. Two of the targets are roughly "lung shaped", and the third is roughly "heart shaped"; the lung shaped targets have a lower conductivity than the saline, while the heart shaped target has a higher conductivity. The stimulation patterns were defined, for each $i = 1, \dots, N_e - 1$, by:

$$C_{ij} = \begin{cases} A\cos(i\phi_j) & i < N_e/2 \\ A\cos(\pi j) & i = N_e/2 \\ A\sin((i - N_e/2)\,\phi_j) & i > N_e/2 \end{cases}.$$

Here A is the amplitude of the current and

$$\phi_j = \frac{4\pi j}{N_e}.$$

This serves as a useful test problem; the data was obtained in the real-world and is not simulated, but the conditions for the experiment were nevertheless carefully controlled and it was conducted in laboratory conditions. Bayesian inversion for this problem will be performed in Sections 4.6.2 and 5.4.2.

[5] See http://eidors3d.sourceforge.net/data_contrib/jn_chest_phantom/jn_chest_phantom.shtml.

2.5 Conclusion

This concludes the background material required for the thesis. The next short chapter gives an intuitive definition of a *Bayesian* probabilistic numerical method, setting the stage for the remainder of the thesis.

Chapter 3

Bayesian Probabilistic Numerical Methods

> "We demand rigidly defined areas of doubt and uncertainty!"
>
> —*Douglas Adams, The Hitchhiker's Guide to the Galaxy*

This chapter introduces the concept of a Bayesian probabilistic numerical method, and gives several examples of such methods that will be examined in more detail later in the thesis.

3.1 Bayesian Probabilistic Numerical Methods

Bayesian probabilistic numerical methods can be thought of as Bayesian inversion problems applied to problems in numerical analysis. To formalise this, we must first provide a framework in which to describe problems from numerical analysis in a framework amenable to inference.

Abstractly, consider an unknown quantity $u^\dagger \in \mathcal{X}$, where $\mathcal{X}$ is a separable Hilbert space that may be finite- or infinite-dimensional. Next, a notion of finite-dimensional *information* about the unknown $u^\dagger$ must be defined. We restrict to finite-dimensional information because BPNM are fundamentally *numerical methods*, and therefore must operate in a computational framework of finite memory. To this end, let $\mathcal{Y}$ be a second separable Hilbert space, this time explicitly finite-dimensional. We will generally assume that $\mathcal{Y} \subseteq \mathbb{R}^d$. Information about the unknown $u^\dagger$ is taken to be provided by an *information operator* $A : \mathcal{X} \to \mathcal{Y}$. Lastly, it is often necessary to define an additional *quantity-of-interest* (QoI) operator, to capture the fact that sometimes a derived quantity is of more interest than the object of inference. Let $\mathcal{Q}$ be a QoI space, and let $Q : \mathcal{X} \to \mathcal{Q}$ be a QoI operator.

Thus, to summarise, interest is in describing problems in numerical analysis using two operators:

- The *information operator* $A : \mathcal{X} \to \mathcal{Y}$, describing how the unknown $u^\dagger$ is linked to computable information $\boldsymbol{y} \in \mathcal{Y}$.
- The *QoI operator* $Q : \mathcal{X} \to \mathcal{Y}$, describing the derived quantity of mathematical interest.

It should be emphasised that the vector $\boldsymbol{y} \in \mathcal{Y}$ will generally be used to refer to the "true" information that has been computed about the unknown $u^\dagger$. While the QoI operator is an important component of the problem, it is the *information equation* that allows numerical methods to be posed as inference problems:

$$A(u^\dagger) = \boldsymbol{y}. \tag{3.1}$$

Viewed through the lens of Eq. (2.11) the analogy is clear. From the terminology in that section, A is the parameter to observation map, and the noise ξ is a Dirac distribution centred on 0. The terminology here is similar to that adopted in the literature on the average-case analysis of numerical methods [see e.g. Ritter, 2000]. Examples of particular numerical problems, and how they can be adapted to this framework, will now be given.

Example 3.1.1 (Solution of Linear Systems)**.** Let $\mathcal{X} = \mathbb{R}^d$ for some $d < \infty$, and consider solution of the linear system

$$A\boldsymbol{x}^\dagger = \boldsymbol{b}$$

where $A \in \mathbb{R}^{d \times d}$ is assumed to be invertible. Here the unknown is the vector $\boldsymbol{x}^\dagger$. Note the switch in notation; when considering the solution of finite-dimensional linear systems of equations we will use $\boldsymbol{x}$ instead of u to refer to objects in the solution space $\mathcal{X}$. This is to adhere to the conventional notation $A\boldsymbol{x} = \boldsymbol{b}$ used when studying such systems in the numerical analysis community.

Since $\mathcal{X}$ is finite-dimensional, the system could be solved explicitly in finite time (assuming exact arithmetic). There nevertheless exist a plethora of so-called *iterative* methods for the solution of such systems. These are methods for which terminating the method before convergence provides a meaningful solution to the problem, in the sense that the error incurred is small; more detail will be provided in Chapter 4. Iterative methods are often constructed by constructing a sequence of *search directions* $\boldsymbol{s}_1, \dots, \boldsymbol{s}_m$, $m < d$, and left-multiplying the system by these directions to obtain information, i.e.

$$\boldsymbol{s}_i^\top A \boldsymbol{x}^\dagger = \boldsymbol{s}_i^\top \boldsymbol{b}$$

for $i = 1, \dots, m$. This principle can be used to define an *information operator*. Let $\mathcal{Y} = \mathbb{R}^m$, and let $S_m \in \mathbb{R}^{d \times m}$ be defined by

$$S_m = [\boldsymbol{s}_1, \dots, \boldsymbol{s}_m].$$

Then, the information equation is given by

$$A(\boldsymbol{x}) = S_m^\top A \boldsymbol{x}$$

and the information $\boldsymbol{y}$ can be computed as

$$\boldsymbol{y} = S_m^\top \boldsymbol{b}.$$

In this example the object of interest is the solution $\boldsymbol{x}^\dagger$; thus, the QoI operator is simply the identity operator $Q(\boldsymbol{x}) = \boldsymbol{x}$, and the QoI space is $\mathcal{Q} = \mathcal{X}$. This problem was considered in Cockayne et al. [2019b] and Bartels et al. [2019], and will be studied in considerable detail in Chapter 4.

Example 3.1.2 (Integration)**.** Let Π be a measure on the domain D, and let $\mathcal{X}$ now be a Hilbert space of Π-integrable functions $u : D \to \mathbb{R}$. Consider computation of the following integral:

$$\Pi(u^\dagger) := \int_D u^\dagger(\boldsymbol{x}) \Pi(\mathrm{d}\boldsymbol{x}) \tag{3.2}$$

In this setting $\mathcal{X}$ is typically an infinite-dimensional space. Somewhat counter-intuitively, $u^\dagger$ is now a *given* function, but nevertheless for arbitrary $u^\dagger$ and Π the integral $\Pi(u^\dagger)$ often does not have a closed-form. The probabilistic interpretation here is used to capture the fact that, although one can interrogate $u^\dagger$ pointwise, the desired QoI can *not* be computed explicitly. Imperfect information about $u^\dagger$ must therefore be used to estimate the integral, owing to the finite nature of computation.

Many standard numerical methods for solving this problem involve implicitly constructing some approximation to the function $u^\dagger$ based on evaluation at a finite set of points $\{\boldsymbol{x}_1, \dots, \boldsymbol{x}_m\} \subset D$. Classical examples of such methods include the trapezium rule and Gaussian quadrature, but this also encompasses more advanced methods such as quasi Monte-Carlo [Niederreiter, 1992]. Again, this can be used to construct an information operator. Let $\mathcal{Y} = \mathbb{R}^m$, and consider the information operator

$$A(u) = \begin{bmatrix} u(\boldsymbol{x}_1) \\ \vdots \\ u(\boldsymbol{x}_m) \end{bmatrix}$$

with the information naturally given by

$$\boldsymbol{y} = \begin{bmatrix} u^\dagger(\boldsymbol{x}_1) \\ \vdots \\ u^\dagger(\boldsymbol{x}_m) \end{bmatrix}$$

Since an explicit closed-form exists for $u^\dagger$, $\boldsymbol{y}$ is computable. Here, however, the QoI operator is nontrivial. The space $\mathcal{Q} = \mathbb{R}$, and $Q(u)$ is defined by:

$$Q(u) = \int_D u(\boldsymbol{x})\Pi(\mathrm{d}\boldsymbol{x}).$$

This problem is not considered in detail in this thesis, but is known as *Bayesian quadrature* in the literature; see Briol et al. [2019], and references therein, for a comprehensive introduction.

Example 3.1.3 (Partial Differential Equations)**.** The last example of translating a numerical problem into an inference problem is of a partial differential equation. We will again take Eqs. (1.1) and (2.1) as a motivating example of a PDE; thus, $\mathcal{X}$ is a Hilbert space of functions that are suitably differentiable to serve as solutions to the PDE [1]. Let D be an open subset of $\mathbb{R}^d$ with boundary ∂D, and suppose that the function $\kappa(\boldsymbol{x})$ is given. Then, we have the following system:

$$\begin{aligned} -\nabla \cdot (\kappa(\boldsymbol{x})\nabla u(\boldsymbol{x})) &= g(\boldsymbol{x}) & \boldsymbol{x} &\in D \\ u(\boldsymbol{x}) &= b(\boldsymbol{x}) & \boldsymbol{x} &\in \partial D \end{aligned} \tag{3.3}$$

Numerical methods for PDEs obtain information about the solution by interrogating the functions g and b in a multitude of ways. In this thesis, as in Example 3.1.2, we will focus on pointwise interrogation of these functions. Let $\{\boldsymbol{x}^{\mathrm{I}}_1, \dots, \boldsymbol{x}^{\mathrm{I}}_m\} \subset \partial D$ and $\{\boldsymbol{x}^{\mathrm{B}}_1, \dots, \boldsymbol{x}^{\mathrm{B}}_n\} \subset D$ be finite subsets of the interior and boundary, respectively. Then $\mathcal{Y} = \mathbb{R}^{m+n}$ and the information operator and information can each be defined

[1] This will be made formal in Chapter 5.

by

$$A(u) = \begin{bmatrix} -\nabla \cdot (\kappa(\boldsymbol{x}_1^{\mathrm{I}}) \nabla u(\boldsymbol{x}_1^{\mathrm{I}})) \\ \vdots \\ -\nabla \cdot (\kappa(\boldsymbol{x}_m^{\mathrm{I}}) \nabla u(\boldsymbol{x}_m^{\mathrm{I}})) \\ u(\boldsymbol{x}_1^{\mathrm{B}}) \\ \vdots \\ u(\boldsymbol{x}_n^{\mathrm{B}}) \end{bmatrix} \qquad \boldsymbol{y} = \begin{bmatrix} g(\boldsymbol{x}_1^{\mathrm{I}}) \\ \vdots \\ g(\boldsymbol{x}_m^{\mathrm{I}}) \\ b(\boldsymbol{x}_1^{\mathrm{B}}) \\ \vdots \\ b(\boldsymbol{x}_n^{\mathrm{B}}) \end{bmatrix}.$$

This choice of information operator and information is less common than other choices, for reasons that will be explained in Chapter 5, which examines this problem in detail. However, numerical methods using such a choice of information are nevertheless widely applied, with a prime example being the symmetric collocation method of Fasshauer [1999].

Having established a framework in which to view numerical problems as inference problems, it is straightforward to state a rigorous definition of both a PNM, and a BPNM.

Definition 3.1.4 (PNM)**.** A *probabilistic numerical method* is defined by an information operator A, a QoI operator Q and a *belief update rule* $\mathfrak{A} : \mathcal{P}_{\mathcal{X}} \times \boldsymbol{y} \to \mathcal{P}_{\mathcal{X}}$. The method itself is an operator $M : \mathcal{P}_{\mathcal{X}} \times \mathcal{Y} \to \mathcal{P}_{\mathcal{Q}}$, given by

$$M(\mu, \boldsymbol{y}) = Q_{\#}\mathfrak{A}(\mu, \boldsymbol{y})$$

The *belief update rule* is a rule which takes a prior distribution $\mu \in \mathcal{P}_{\mathcal{X}}$ and updates it to a *posterior belief* $\mathfrak{A}(\mu, \boldsymbol{y}) \in \mathcal{P}_{\mathcal{X}}$. No restriction is placed on the form of this update rule, but the specification of a Bayesian PNM is then straightforward.

Definition 3.1.5 (Bayesian PNM)**.** A probabilistic numerical method is said to be *Bayesian* if its update rule $\mathfrak{A}(\mu, \boldsymbol{y})$ represents conditioning of μ on $\boldsymbol{y}$, i.e.

$$\mathfrak{A}(\mu, \boldsymbol{y}) = \mu^{\boldsymbol{y}}$$

for $A_{\#}\mu$-almost-all $\boldsymbol{y} \in \mathcal{Y}$.

This definition seems natural. Yet, examining many existing PNM, surprisingly few methods are truly Bayesian by this definition. A breakdown of existing PNM based upon this definition can be found in Appendix B.

Note that the definition of the conditional distribution $\mu^{\boldsymbol{y}}$, while intuitively understood, has been left abstract. This is because the rigorous definition of that conditional distribution when observations are made *without noise* introduces considerable additional technical detail, which will be presented in Part III. Nevertheless, for certain choices of prior μ and operators A and Q the required conditional distribution can be constructed explicitly. We refer to these settings as *conjugate*, and they will be examined in detail in Part II.

We conclude this chapter by discussing a special case of these definitions.

3.2 Classical Methods as PNM

It is convenient to be able to express classical numerical methods in the same framework as PNM, as this allows PNM and classical numerical methods to be compared in a single mathematical framework, such as will be introduced in Section 6.1.4. Many numerical methods can be viewed as obtaining information through an information equation as in Eq. (3.1). Such numerical methods can then be abstracted as an operator $\mathsf{NM} : \mathcal{Y} \to \mathcal{Q}$, which takes the information $\boldsymbol{y}$ and outputs an estimate of $q^\dagger = Q(u^\dagger)$ formed by applying the numerical method to the computed information. Thus, one can define the degenerate PNM

$$M_{\mathsf{NM}}(\mu, \boldsymbol{y}) = \delta(\mathsf{NM}(\boldsymbol{y}))$$

which simply outputs a Dirac distribution centred on $\mathsf{NM}(\boldsymbol{y})$. Note that this is independent of the prior μ.

Note that not *all* numerical methods can be set in the form of Eq. (3.1). For example, solvers of ordinary differential equations are often adaptive, meaning that the algorithm iteratively selects information by reflecting on the current solution estimate. Such solvers lie within a more general class of numerical methods, and while an extension of Eq. (3.1) to encompass such solvers may be possible, it is not considered in this thesis.

3.3 Conclusion

This concludes the intuitive definition of a Bayesian PNM. To reiterate, the existence of the posterior distribution $\mu^{\boldsymbol{y}}$ in Definition 3.1.5 has thus far been *assumed*. This will be addressed rigorously in Chapter 6. The next part is instead concerned with the conjugate setting in which a closed-form for the posterior distribution is available.

Part II

Conjugate Methods

Chapter 4

The Bayesian Conjugate Gradient Method

> "The practising Bayesian is well advised to become friends with as many numerical analysts as possible."
>
> —*James Berger, Statistical Decision Theory and Bayesian Analysis*

This chapter presents a PNM for the solution of finite-dimensional systems of linear equations of the form

$$A\boldsymbol{x}^\dagger = \boldsymbol{b} \tag{4.1}$$

where the matrix $A \in \mathbb{R}^{d\times d}$ and the vector $\boldsymbol{b} \in \mathbb{R}^d$ are each assumed to be given, while the vector $\boldsymbol{x}^\dagger \in \mathbb{R}^d$ is the unknown to be determined. In the notation of Chapter 3, $\mathcal{X} = \mathbb{R}^d$. This method is known as the *Bayesian conjugate gradient method* and was first presented in Cockayne et al. [2019b].

In many PNMs the rationale for adopting a probabilistic approach to the problem is the finite nature of computation. When $\mathcal{X}$ is infinite-dimensional the unknown can never be perfectly identified, and using probability to describe residual uncertainty in its value can be justified. In this case, since $\mathcal{X}$ is finite-dimensional this justification no longer holds. Nevertheless, a probabilistic approach to these problems is of interest for several reasons. Firstly, the structure exposed by the analysis in this chapter serves as an illuminating introduction to PNMs. Secondly and more practically, most naive inversion methods for linear systems such as that in Eq. (4.1) incur an $\mathcal{O}(d^3)$ cost. If an approximate but still accurate solution can be obtained with reduced computational effort, this may be of value when d is large.

Such high-dimensional linear systems arise frequently in the applied sciences. An example of particular relevance to this thesis is in the approximate solutions of systems of PDEs. The finite element and finite difference discretisations of PDEs discussed in Section 2.2.1 each yield large, sparse linear systems which can have billions of degrees of freedom. In some cases specialised algorithms have been developed to allow practical solution of these systems [e.g. Reinarz et al., 2018]. Another example familiar to statisticians arises in computation with Gaussian measures (see Section 2.3.2). Linear systems based upon the covariance function of the GP must be solved to work with the conditional distributions of these measures, and so efficient solution of such systems is vital for any method involving generation of spatial random fields [Besag and Green, 1993; Parker and Fox, 2012; Schäfer et al., 2017]. This includes many PNMs, such as that which will be presented in Chapter 5. In some such applications, such as in models of tropical ocean surface winds [Wikle et al., 2001], the resulting systems may again have billions of degrees of freedom. Thus, it is clear that there exist many important situations in which exact solution of a linear system is not practical.

The structure of this chapter is as follows: Section 4.1 introduces classical methods for solving linear systems. In Section 4.2 a prototypical PNM for Eq. (4.1) is presented and its inputs discussed. Its correspondence with the conjugate gradient method (CG) is also established for a particular choice of prior. Section 4.3 presents the Bayesian conjugate gradient method (BayesCG), demonstrates that its posterior mean lies in a particular Krylov subspace, and presents convergence analysis. In Section 4.4 the critical issue of prior choice is addressed. Several choices of prior covariance are discussed, and a hierarchical prior is introduced to allow BayesCG to adapt to the scale of the problem. Section 4.5 contains implementation details, while in Section 4.6 the method is applied to a challenging problem in medical imaging, which requires repeated solution of a linear system arising from discretisation of a PDE. Most of the theoretical results from this chapter are proven in the main text, but some lengthier proofs are presented in Appendix C.

4.1 Classical Techniques for Linear Systems

Broadly speaking there are two main categories of method for solving Eq. (4.1) numerically: *iterative methods* and *direct methods*. Direct methods seek to compute the solution $\boldsymbol{x}^\dagger$ exactly, assuming exact arithmetic. The most naïve direct method involves computing the matrix inverse A^{-1} and applying it to $\boldsymbol{b}$. This method is widely known to be a highly numerically unstable way to compute the solution.

More prudent direct methods for solving Eq. (4.1) involve factorising the matrix A into two factors. For example, in the LU decomposition [see Section 3.2 of

Golub and Van Loan, 2013], two matrices L and U are constructed so that $A = LU$, while L is lower triangular and U is upper triangular. Once this decomposition has been computed, the solution to the problem can be computed by first solving the system $L\boldsymbol{y} = \boldsymbol{b}$, and then solving $U\boldsymbol{x} = \boldsymbol{y}$, which incurs only an $\mathcal{O}(d^2)$ cost owing to the fact that L and U are triangular. However, the cost of computing the factors L and U remains $\mathcal{O}(d^3)$. Furthermore, even when A is sparse, in that it has many zero entries, the factors L and U are generally dense and so $\mathcal{O}(d^2)$ additional storage would be required.

Iterative methods take a different approach to direct methods. In an iterative method, the goal is to construct a sequence $(\boldsymbol{x}_m)$ which approaches $\boldsymbol{x}^\dagger$ in some suitable norm, as m increases. CG is a well-known example of an iterative method for approximate solution of linear systems based on positive-definite matrices A. It was first introduced in Hestenes and Stiefel [1952]. Unlike the LU factorisation, the CG algorithm is *designed* to be terminated after performing $m < d$ iterations, at a potentially substantially lower cost of $\mathcal{O}(md^2)$. Furthermore, when A is sparse, the sparsity structure is reflected in the cost of the algorithm, in that if $\mathrm{nnz}(A)$ denotes the number of nonzero entries of A, the cost of CG applied to A is just $\mathcal{O}(m \cdot \mathrm{nnz}(A))$. Thus CG, or variants thereof, are the standard methods for solving many problems governed by sparse, symmetric-positive definite matrices, such as in solving systems arising from FEA.

A detailed introduction to CG will now be presented. The material in the remainder of this section follows Golub and Van Loan [2013] to some degree; see also Liesen and Strakos [2012]. To proceed, we first introduce the inner product and norm induced by a positive-definite matrix M.

Definition 4.1.1. For a positive-definite matrix M, the *inner product induced by* M, denoted $\langle \boldsymbol{x}, \boldsymbol{y} \rangle_M$, is given by

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle_M = \boldsymbol{x}^\top M \boldsymbol{y}.$$

The norm induced by this inner product is denoted $\|\boldsymbol{x}\|_M$ and given by

$$\|\boldsymbol{x}\|_M := \sqrt{\langle \boldsymbol{x}, \boldsymbol{x} \rangle_M}.$$

When $\langle \boldsymbol{x}, \boldsymbol{y} \rangle_M = 0$ we say that $\boldsymbol{x}$ and $\boldsymbol{y}$ are *M-orthogonal*, or alternatively *conjugate with-respect-to M*. When additionally each of $\boldsymbol{x}$ and $\boldsymbol{y}$ have $\|\boldsymbol{x}\|_M = \|\boldsymbol{y}\|_M = 1$ we say they are *M-orthonormal*.

Second, we will often abuse notation to describe *affine spaces*. If the linear subspace $\mathbb{S} \subset \mathbb{R}^d$ has basis $\{\boldsymbol{s}_1, \dots, \boldsymbol{s}_m\}$, $m < d$, then for a vector $v \in \mathbb{R}^d$ and a matrix $M \in \mathbb{R}^{d\times d}$ the affine space $\boldsymbol{v} + M\mathbb{S}$ is defined as:

$$\boldsymbol{v} + M\mathbb{S} := \mathrm{span}(\{\boldsymbol{v} + M\boldsymbol{s}_1, \dots, \boldsymbol{v} + M\boldsymbol{s}_m\}.$$

4.1.1 Gradient Descent

A class of iterative methods arise from solving the following optimization problem in a sequence of affine spaces $\mathbb{K}_m \subset \mathbb{R}^d$, where $\dim(\mathbb{K}_m) = m$:

$$\boldsymbol{x}_m = \underset{\boldsymbol{x}\in\mathbb{K}_m}{\arg\min} \|\boldsymbol{x}^\dagger - \boldsymbol{x}\|_A. \tag{4.2}$$

Denote by $\boldsymbol{e}_m$ the *error* at iteration m and by $\boldsymbol{r}_m$ the *residual*. These are given by:

$$\begin{aligned}\boldsymbol{e}_m &: = \boldsymbol{x}^\dagger - \boldsymbol{x}_m \\ \boldsymbol{r}_m &: = A(\boldsymbol{x}^\dagger - \boldsymbol{x}_m) = \boldsymbol{b} - A\boldsymbol{x}_m \\ &= A\boldsymbol{e}_m\end{aligned}$$

While both of these are valid ways of describing the difference between $\boldsymbol{x}_m$ and the truth, $\boldsymbol{r}_m$ has the advantage of being computable, as $\boldsymbol{x}^\dagger$ is unknown while $\boldsymbol{b}$ is given.

If one seeks to solve Eq. (4.1) by iteratively minimising Eq. (4.2), a natural approach is to perform this minimisation by *gradient descent* starting from a user-supplied initial point $\boldsymbol{x}_0$. It is convenient here to consider the equivalent problem of minimising the quadratic objective function $f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^\top A\boldsymbol{x} - \boldsymbol{x}^\top\boldsymbol{b}$, whose gradient can be computed as:

$$\nabla f(\boldsymbol{x}) = A\boldsymbol{x} - \boldsymbol{b}$$

so that $\nabla f(\boldsymbol{x}_m) = -\boldsymbol{r}_m$. Then, from $\boldsymbol{x}_m$, the next estimate $\boldsymbol{x}_{m+1}$ is found by moving in the direction which decreases $f(\boldsymbol{x}_m)$ most rapidly, i.e.:

$$\begin{aligned}\boldsymbol{x}_{m+1} &= \boldsymbol{x}_m - \alpha_m\nabla f(\boldsymbol{x}_m) \\ &= \boldsymbol{x}_m + \alpha_m\boldsymbol{r}_m\end{aligned}$$

where α_m is a coefficient to be determined. Thus, the residual $\boldsymbol{r}_m$ is the m^{th} search direction for gradient descent methods.

The coefficient α_m can be determined by minimising $f(\boldsymbol{x}_{m+1})$:

$$\begin{aligned}\frac{\partial}{\partial \alpha_m} f(\boldsymbol{x}_m + \alpha_m \boldsymbol{r}_m) &= \boldsymbol{r}_m^\top \nabla f(\boldsymbol{x}_m + \alpha_m \boldsymbol{r}_m) \\ &= \boldsymbol{r}_m^\top (A(\boldsymbol{x}_m + \alpha_m \boldsymbol{r}_m) - \boldsymbol{b}) \\ &= \alpha_m \boldsymbol{r}_m^\top A \boldsymbol{r}_m - \boldsymbol{r}_m^\top \boldsymbol{r}_m\end{aligned}$$

where the first line is from application of the chain rule. Setting this to zero, we obtain

$$\alpha_m = \frac{\boldsymbol{r}_m^\top \boldsymbol{r}_m}{\boldsymbol{r}_m^\top A \boldsymbol{r}_m}.$$

In practise, gradient descent is rarely used to solve linear systems, the primary reason being that it does not generally converge in d iterations and thus has a worst-case cost exceeding that of standard direct solvers. The conjugate gradient (CG) method, which will be introduced in the next section, addresses this issue.

4.1.2 Conjugate Gradients

CG augments gradient descent by adding the requirement that the search directions be conjugate with-respect-to A, hence the name *conjugate gradient*. This can be enforced by the Gram-Schmidt orthogonalisation procedure

$$\tilde{\boldsymbol{s}}_m := \boldsymbol{r}_{m-1} - \sum_{i=1}^{m-1} \langle \boldsymbol{s}_i, \boldsymbol{r}_{m-1} \rangle_A \cdot \boldsymbol{s}_i$$

where $\boldsymbol{s}_i := \tilde{\boldsymbol{s}}_i / \|\tilde{\boldsymbol{s}}_i\|_A$. However, the orthogonality properties of the search directions are such that a simpler form exists. The following proposition characterises the CG search directions.

Proposition 4.1.2 (CG Search Directions)**.** *Let the first* un-normalised CG search direction *be* $\tilde{\boldsymbol{s}}_1 = \boldsymbol{r}_0$. *Let the first* normalised CG search direction *be given by*

$$\boldsymbol{s}_1 = \frac{\tilde{\boldsymbol{s}}_1}{\|\tilde{\boldsymbol{s}}_1\|_A}.$$

For each $m > 1$, *let the* m^{th} *CG search direction be given by*

$$\tilde{\boldsymbol{s}}_m = \boldsymbol{r}_{m-1} - \langle \boldsymbol{s}_{m-1}, \boldsymbol{r}_{m-1} \rangle_A \cdot \boldsymbol{s}_{m-1}$$

and let the m^{th} normalised CG search direction be given by

$$\boldsymbol{s}_m = \frac{\tilde{\boldsymbol{s}}_m}{\|\tilde{\boldsymbol{s}}_m\|_A}.$$

Then, for each $m = 1, \dots, d$ it holds that the un-normalised CG search directions $\{\tilde{\boldsymbol{s}}_1, \dots, \tilde{\boldsymbol{s}}_m\}$ form an A-orthogonal set. Similarly, the normalised CG search directions $\{\boldsymbol{s}_1, \dots, \boldsymbol{s}_m\}$ form an A-orthonormal set.

Proof. See Golub and Van Loan [2013, Section 10.2.4], particularly Corollary 10.2.4. □

Note that this result is also a special case of the later Proposition 4.3.2.

To avoid ambiguity, where the meaning is not clear from the context the CG search directions will be referred to using the superscript "CG", i.e. S_m^{CG}. This proposition is useful as computing the next search direction using this formula reduces the storage cost[1] and computational complexity associated with the algorithm. The resulting algorithm computes $\boldsymbol{x}_m$ using only a single matrix-vector product in each iteration, and so has complexity $\mathcal{O}(md^2)$. CG is presented as an algorithm in Algorithm 4.2.

Algorithm 4.2 Implementation of the CG algorithm. Note that the only matrix-vector multiplication required in iteration i is $A\boldsymbol{s}_i$. Furthermore note that only the vectors $\boldsymbol{x}_i$, $\boldsymbol{r}_i$ and $\boldsymbol{s}_i$ need be stored.

```
procedure CG($\boldsymbol{x}_0, A$)
    $\boldsymbol{r}_0 \leftarrow \boldsymbol{b} - A\boldsymbol{x}_0$
    $\boldsymbol{s}_1 \leftarrow \boldsymbol{r}_0$
    for $i = 1, \dots, d$ do
        $\alpha_i \leftarrow \frac{\boldsymbol{r}_{i-1}^\top \boldsymbol{r}_{i-1}}{\|\boldsymbol{s}_i\|_A}$
        $\boldsymbol{x}_i \leftarrow \boldsymbol{x}_{i-1} + \alpha_i \boldsymbol{s}_i$
        $\boldsymbol{r}_i \leftarrow \boldsymbol{r}_{i-1} - \alpha_i A\boldsymbol{s}_i$
        if $\boldsymbol{r}_i$ "sufficiently small" then
            break
        end if
        $\beta_i \leftarrow \frac{\boldsymbol{r}_i^\top \boldsymbol{r}_i}{\boldsymbol{r}_{i-1}^\top \boldsymbol{r}_{i-1}}$
        $\boldsymbol{s}_{i+1} \leftarrow \boldsymbol{r}_i + \beta_i \boldsymbol{s}_i$
    end for
    return $x_i$
end procedure
```

[1] In fact this can be simplified further to remove the requirement to store $\boldsymbol{s}_{m-1}$, but that simplification is not critical to this presentation.

Note that the solution $\boldsymbol{x}_m$ from CG can also be expressed as

$$\boldsymbol{x}_m = \boldsymbol{x}_0 + S_m S_m^\top \boldsymbol{r}_0 \tag{4.3}$$

where S_m is the matrix whose columns are $\boldsymbol{s}_1, \dots, \boldsymbol{s}_m$. This can be shown by first noting, from Algorithm 4.2, that $\boldsymbol{x}_m \in \boldsymbol{x}_0 + \operatorname{span}(\boldsymbol{s}_1, \dots, \boldsymbol{s}_m)$, and so can be written as

$$\boldsymbol{x}_m = \boldsymbol{x}_0 + S_m^\top \boldsymbol{\alpha}$$

for some $\boldsymbol{\alpha} \in \mathbb{R}^m$. Minimising $\|\boldsymbol{x}_0 + S_m^\top \boldsymbol{\alpha} - \boldsymbol{x}^\dagger\|_A$ yields the expression in Eq. (4.3).

There are several important results about CG which must now be mentioned. First among these is the characterisation of CG as a *Krylov subspace* method.

Definition 4.1.3 (Krylov Subspace)**.** The *Krylov subspace* of order m generated by a matrix $A \in \mathbb{R}^{d\times d}$ and a vector $\boldsymbol{b} \in \mathbb{R}^d$ is given by

$$K_m(A, \boldsymbol{b}) := \operatorname{span}\{\boldsymbol{b}, A\boldsymbol{b}, A^2\boldsymbol{b}, \dots, A^{m-1}\boldsymbol{b}\}.$$

The next theorem identifies the search directions in Proposition 4.1.2 as a basis of $K_m(A, \boldsymbol{b})$.

Theorem 4.1.4. *The search directions S_m from Proposition 4.1.2 have the property*

$$\operatorname{range}(S_m) = K_m(A, \boldsymbol{r}_0)$$

Proof. See Golub and Van Loan [2013, Theorem 10.2.3]. □

An immediate corollary of Theorem 4.1.4 and Eq. (4.3) is the following:

Corollary 4.1.5 (CG as a Krylov Subspace Method)**.** *CG is a Krylov Subspace method, in that $\boldsymbol{x}_m \in \boldsymbol{x}_0 + K_m(A, \boldsymbol{r}_0)$. Furthermore $\boldsymbol{x}_m$ is A-optimal in $\boldsymbol{x}_0 + K_m(A, \boldsymbol{r}_0)$, in that*

$$\boldsymbol{x}_m = \operatorname*{arg\,min}_{\boldsymbol{x} \in \boldsymbol{x}_0 + K_m(A, \boldsymbol{r}_0)} \|\boldsymbol{x} - \boldsymbol{x}^\dagger\|_A.$$

Proof. From Theorem 4.1.4, $\boldsymbol{x}_m \in K_m(A, \boldsymbol{b})$, and $\boldsymbol{x}_m$ is optimal in this space as it seeks to minimise Eq. (4.2). □

This result is also a consequence of Corollary 4.3.4, which will be proven later in this chapter.

Lastly, CG can be shown to converge exponentially fast in m.

Definition 4.1.6 (Condition Number)**.** For an arbitrary invertible matrix M, the *condition number* of M is given by

$$\kappa(M) := \|M\|_2 \|M^{-1}\|_2$$

where $\|M\|_2$ denotes that *matrix* 2*-norm* or *Frobenius norm.* When M is positive-definite, let $\lambda_{\max}$ denote the largest eigenvalue of M and $\lambda_{\min}$ denote the smallest eigenvalue of M. Then it holds that

$$\kappa(M) = \frac{\lambda_{\max}}{\lambda_{\min}}.$$

When $\kappa(M)$ is very large we will often say that M is *ill-conditioned.*

Theorem 4.1.7 (Convergence of CG)**.**

$$\frac{\|\boldsymbol{x}_m - \boldsymbol{x}^\dagger\|_A}{\|\boldsymbol{x}_0 - \boldsymbol{x}^\dagger\|_A} \leq 2 \left(\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1} \right)^m$$

Proof. See Golub and Van Loan [2013, Theorem 10.2.6]. □

Thus, the convergence rate of CG is driven by how well-conditioned the matrix A is. In practise, the procedure in Proposition 4.1.2 can be shown to be numerically unstable (see Liesen and Strakos [2012, Section 5.9]), and so even though the convergence shown in Theorem 4.1.7 is exponentially fast, conjugacy of the search directions often breaks down much faster than the rate at which the error reduces. Thus, an important practise is that of *preconditioning* the matrix A to accelerate convergence. This will be introduced next.

4.1.3 Preconditioning

Theorem 4.1.7 motivates the practise of *preconditioning* [Allaire and Kaber, 2008], in which an equivalent system to Eq. (4.1) is constructed and solved by using the auxiliary matrix P, known as a *preconditioner.* Two main preconditioning strategies exist; in left-preconditioning, one solves the system $P^{-1}A\boldsymbol{x}^\dagger = P^{-1}\boldsymbol{b}$, while in right-preconditioning the system $AP^{-1}P\boldsymbol{x}^\dagger = \boldsymbol{b}$ is solved. P is typically chosen to satisfy:

1. $\kappa(P^{-1}A) < \kappa(A)$ (or $\kappa(AP^{-1}) < \kappa(A)$).

2. The solution of systems $P\boldsymbol{x}' = \boldsymbol{b}'$ is computationally inexpensive for arbitrary $\boldsymbol{x}'$ and $\boldsymbol{b}'$.

From Theorem 4.1.7 it is clear that CG applied to a preconditioned system will converge faster than standard CG if P is chosen well.

The optimal preconditioner is $P = A^{-1}$, but this is clearly impractical. Practical preconditioners tend to be problem-specific. There exist reasonably generic approaches for sparse matrices that involve approximate decompositions of the matrix. Often these are modifications of standard matrix decompositions as used in direct methods, which aim to also preserve some measure of sparsity. Examples are the incomplete LU or incomplete Cholesky decomposition [e.g. Ajiz and Jennings, 1984; Saad, 1994]. Another, more application-specific example is in numerical solution of PDEs, where a coarse discretisation of the domain can be used to construct a preconditioner for a finer discretisation [e.g. Bramble et al., 1990]. A more detailed survey of preconditioning methods can be found in many standard texts, such as Benzi [2002] and Saad [2003]. There is no generic preconditioning method, however, and constructing one can be challenging. Indeed, this has been described as "a combination of art and science" [Saad, 2003, Chapter 10].

4.2 A Probabilistic Linear Solver

In this section a generic PNM for solving Eq. (4.1) is presented. First, in Section 4.2.1, the generic method is presented. Then, in Section 4.2.2, a particular choice of prior is described, for which the probabilistic method coincides with the solution produced from CG.

4.2.1 Inference with a Gaussian Prior

In a finite-dimensional setting multivariate Gaussian distributions are often defined by their Lebesgue density. However, in this chapter we will often work with Gaussian distributions that are *singular*, in the sense that they are concentrated on a linear subspace of $\mathbb{R}^d$. Since (strict) linear subspaces are of measure zero under the Lebesgue measure on $\mathbb{R}^d$, no Lebesgue density exists in this case. As a result we will instead say that a random vector $X \sim \mathcal{N}(\boldsymbol{x}, \Sigma)$ with mean $\boldsymbol{x} \in \mathbb{R}^d$ and positive semidefinite covariance $\Sigma \in \mathbb{R}^{d \times d}$ if it holds that $\lambda^\top X \sim \mathcal{N}(\lambda^\top \boldsymbol{x}, \lambda^\top \Sigma \lambda)$ for each $\lambda \in \mathbb{R}^d$. This definition is valid even when X is singular, though note that in this case $\lambda^\top X$ will be a degenerate Gaussian with zero variance whenever λ lies outside the subspace on which X is concentrated.

Now, let X describe the prior level of uncertainty in the solution to Eq. (4.1). It will be assumed that

$$X \sim \mathcal{N}(\boldsymbol{x}_0, \Sigma_0). \tag{4.4}$$

Here $\boldsymbol{x}_0 \in \mathbb{R}^d$ and $\Sigma_0 \in \mathbb{R}^{d\times d}$ are each assumed to be fixed, and furthermore for convenience Σ_0 is assumed to be positive-definite so that its inverse exists.

We seek a Bayesian method in accordance with Definition 3.1.5. As described in Example 3.1.1, let $\boldsymbol{y}_m \in \mathbb{R}^d$ and $S_m \in \mathbb{R}^{d\times m}$ each be given by

$$\boldsymbol{y}_m = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix} \qquad S_m = \begin{bmatrix} \uparrow & & \uparrow \\ \boldsymbol{s}_1 & \dots & \boldsymbol{s}_m \\ \downarrow & & \downarrow \end{bmatrix}$$

where the search directions are assumed to be linearly independent and initially assumed to be given *a-priori*. In Section 4.3.1 a particular choice will be presented which defines BayesCG. To associate with the notation in Chapter 3, the information operator is given by

$$A_m(\boldsymbol{x}) = S_m^\top A\boldsymbol{x}.$$

Thus, the information is $\boldsymbol{y}_m = S_m^\top \boldsymbol{b}$, and $\mathcal{Y}_m = \mathbb{R}^m$. Here the subscript m has been used to emphasise that the information is considered to be generated iteratively over a number of iterations. The QoI in this example is $\boldsymbol{x}^\dagger$ itself, so $Q(\boldsymbol{x}) = \boldsymbol{x}$ and $\mathcal{Q} = \mathbb{R}^d$.

For this choice of prior, information operator and QoI operator, the BPNM $M(\mu, \boldsymbol{y}_m)$ outputs a closed-form posterior thanks to the conjugacy properties of Gaussian distributions.

Proposition 4.2.1 (Probabilistic Linear Solver)**.** *Let* $\Lambda_m = S_m^\top A\Sigma_0 A^\top S_m$ *and* $\boldsymbol{r}_0 = \boldsymbol{b} - A\boldsymbol{x}_0$*. Then the posterior distribution is given by*

$$p(\boldsymbol{x}|\boldsymbol{y}_m) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{x}_m, \Sigma_m)$$

where

$$\boldsymbol{x}_m = \boldsymbol{x}_0 + \Sigma_0 A^\top S_m \Lambda_m^{-1} S_m^\top \boldsymbol{r}_0 \tag{4.5}$$

$$\Sigma_m = \Sigma_0 - \Sigma_0 A^\top S_m \Lambda_m^{-1} S_m^\top A\Sigma_0 \tag{4.6}$$

Proof. Note that the joint distribution of $\boldsymbol{x}$ and $\boldsymbol{y}_m$ is given by

$$\begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y}_m \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{x}_0 \\ S_m^\top A\boldsymbol{x}_0 \end{bmatrix}, \begin{bmatrix} \Sigma_0 & \Sigma_0 A^\top S_m \\ S_m^\top A\Sigma_0 & S_m^\top A\Sigma_0 A^\top S_m \end{bmatrix} \right)$$

from which the conditional distribution can be derived directly. □

The posterior from Proposition 4.2.1 describes residual uncertainty in $\boldsymbol{x}^\dagger$

given the information in $\boldsymbol{y}_m$, but is impractical for use as a probabilistic numerical method. This is primarily due to the requirement to invert the matrix $\Lambda_m \in \mathbb{R}^{m\times m}$. Though this comes at a lower cost than inverting A itself ($\mathcal{O}(m^3)$ as opposed to $\mathcal{O}(d^3)$), if many search directions are required to reach a desired level of accuracy the cost of computing the posterior is still prohibitive. This will be addressed by construction of a particular set of search directions which *diagonalise* Λ_m, in Section 4.3.1.

An important remark is that the posterior covariance matrix is singular, i.e. $\det(\Sigma_m) = 0$; in particular, $S_m^\top A \Sigma_m = 0$, since:

$$\begin{aligned} S_m^\top A \Sigma_m &= S_m^\top A \Sigma_0 - \underbrace{S_m^\top A \Sigma_0 A^\top S_m \Lambda_m^{-1}}_{=I} S_m^\top A \Sigma_0 \\ &= 0. \end{aligned}$$

This is due to the fact that observations are not corrupted with noise, so $\boldsymbol{x}^\dagger$ has been completely determined in $\mathrm{range}(S_m^\top A)$; thus, the posterior after m iterations is supported on the complement of this linear subspace. This makes certain posterior quantities, such as probabilities, difficult to compute, but we note that since its null-space is known to be $\mathrm{range}(S_m^\top A)$ it can still be sampled efficiently. This fact is related to the general property that typically the output of BPNM is supported on a null set of the prior, as will be discussed in more detail in Chapter 6.

A basic result, derived from the optimality properties of the conditional mean of Gaussian distributions, gives a sense in which $\boldsymbol{x}_m$ is optimal. This result is well-known but will prove useful when comparing the posterior mean from probabilistic linear solvers with other classical numerical methods.

Proposition 4.2.2. *Let $\mathbb{S}_m = \mathrm{range}(S_m)$. Then, the posterior mean from Proposition 4.2.1 satisfies:*

$$\boldsymbol{x}_m = \operatorname*{arg\,min}_{\boldsymbol{x} \in \boldsymbol{x}_0 + \Sigma_0 A^\top \mathbb{S}_m} \|\boldsymbol{x} - \boldsymbol{x}^\dagger\|_{\Sigma_0^{-1}} \tag{4.7}$$

Proof. First, by inspection the posterior mean from Proposition 4.2.1 satisfies $\boldsymbol{x}_m \in \boldsymbol{x}_0 + \Sigma_0 A^\top \mathbb{S}_m$. Thus, we will solve the optimisation problem in Eq. (4.7), and show that the optimum is equal to $\boldsymbol{x}_m$.

Note that all $\boldsymbol{x} \in \boldsymbol{x}_0 + \Sigma_0 A^\top \mathbb{S}_m$ are of the form

$$\boldsymbol{x} = \boldsymbol{x}_0 + \Sigma_0 A^\top S_m \boldsymbol{\alpha}$$

for some $\boldsymbol{\alpha} \in \mathbb{R}^m$. Inserting this into the norm from the Proposition, note that it is equivalent to solve the following minimisation problem:

$$\begin{aligned}\boldsymbol{\alpha}^* &= \operatorname*{arg\,min}_{\boldsymbol{\alpha}\in\mathbb{R}^m} J(\boldsymbol{\alpha})\\ J(\boldsymbol{\alpha}) &= \frac{1}{2}\|\boldsymbol{x}_0 + \Sigma_0 A^\top S_m \boldsymbol{\alpha} - \boldsymbol{x}^\dagger\|^2_{\Sigma_0^{-1}}\end{aligned}$$

and then set $\boldsymbol{x}^* = \boldsymbol{x}_0 + \Sigma_0 A^\top \boldsymbol{\alpha}^*$. Differentiating with-respect-to $\boldsymbol{\alpha}$ and setting to zero gives:

$$\begin{aligned}\nabla J(\boldsymbol{\alpha}) &= S_m^\top A \Sigma_0 A^\top S_m \boldsymbol{\alpha} + S_m^\top A(\boldsymbol{x}_0 - \boldsymbol{x}^\dagger) = 0\\ \implies \boldsymbol{\alpha^*} &= (S_m^\top A \Sigma_0 A^\top S_m)^{-1} S_m^\top \boldsymbol{r}_0\end{aligned}$$

and so

$$\boldsymbol{x}^* = \boldsymbol{x}_0 + \Sigma_0 A^\top S_m (S_m^\top A \Sigma_0 A^\top S_m)^{-1} S_m^\top \boldsymbol{r}_0.$$

which is equal to $\boldsymbol{x}_m$ in Proposition 4.2.1, as required. □

The next result bounds the rate at which the posterior mean converges to the truth with a function of the posterior covariance, thus showing that this covariance provides a meaningful representation of the error.

Proposition 4.2.3.

$$\frac{\|\boldsymbol{x}_m - \boldsymbol{x}^\dagger\|_{\Sigma_0^{-1}}}{\|\boldsymbol{x}_0 - \boldsymbol{x}^\dagger\|_{\Sigma_0^{-1}}} \leq \sqrt{\operatorname{trace}(\Sigma_m \Sigma_0^{-1})}$$

Proof. Let $\boldsymbol{\ell} \in \mathbb{R}^d$ be an arbitrary vector. Then

$$\begin{aligned}\boldsymbol{\ell}^\top \boldsymbol{x}_m - \boldsymbol{\ell}^\top \boldsymbol{x}^\dagger &= \boldsymbol{\ell}^\top(\boldsymbol{x}_0 - \boldsymbol{x}^\dagger) + \boldsymbol{\ell}^\top \Sigma_0 A^\top S_m \Lambda_m^{-1} S_m^\top A(\boldsymbol{x}^\dagger - \boldsymbol{x}_0) && \text{(from Eq. (4.5))}\\ &= \boldsymbol{\ell}^\top(\Sigma_0 - \Sigma_0 A^\top S_m \Lambda_m^{-1} S_m^\top A \Sigma_0)\Sigma_0^{-1}(\boldsymbol{x}_0 - \boldsymbol{x}^\dagger)\\ &= \langle \Sigma_m \boldsymbol{\ell}, \boldsymbol{x}_0 - \boldsymbol{x}^\dagger \rangle_{\Sigma_0^{-1}} && \text{(from Eq. (4.6))}\end{aligned}$$

and so we have that:

$$\begin{aligned}|\boldsymbol{\ell}^\top \boldsymbol{x}_m - \boldsymbol{\ell}^\top \boldsymbol{x}^\dagger| &= |\langle \Sigma_m \boldsymbol{\ell}, \boldsymbol{x}_0 - \boldsymbol{x}^\dagger \rangle_{\Sigma_0^{-1}}|\\ &\leq \|\boldsymbol{x}_0 - \boldsymbol{x}^\dagger\|_{\Sigma_0^{-1}} \underbrace{\|\Sigma_m \boldsymbol{\ell}\|_{\Sigma_0^{-1}}}_{(*)}.\end{aligned} \tag{4.8}$$

where the last line is from application of the Cauchy–Schwarz inequality. Now, by expanding the term $(*)$ and simplifying, we see that

$$\begin{aligned}\|\Sigma_m \boldsymbol{\ell}\|^2_{\Sigma_0^{-1}} &= \boldsymbol{\ell}^\top (\Sigma_0 - \Sigma_0 A^\top S_m \Lambda_m^{-1} S_m^\top A \Sigma_0)^\top \Sigma_0^{-1} (\Sigma_0 - \Sigma_0 A^\top S_m \Lambda_m^{-1} S_m^\top A \Sigma_0)\boldsymbol{\ell} \\ &= \boldsymbol{\ell}^\top \big(\Sigma_0 - 2\Sigma_0 A^\top S_m \Lambda_m^{-1} S_m^\top A \Sigma_0 \\ &\qquad\qquad + \Sigma_0 A^\top S_m \Lambda_m^{-1} \underbrace{S_m^\top A \Sigma_0 A^\top S_m}_{=\Lambda_m} \Lambda_m^{-1} S_m^\top A \Sigma_0\big)\boldsymbol{\ell} \\ &= \boldsymbol{\ell}^\top (\Sigma_0 - \Sigma_0 A^\top S_m \Lambda_m^{-1} S_m^\top A \Sigma_0)\boldsymbol{\ell} \qquad (4.9) \\ &= \boldsymbol{\ell}^\top \Sigma_m \boldsymbol{\ell}\end{aligned}$$

which follows from Eq. (4.6)

Finally let $\boldsymbol{e}_i$ denote the vector whose j^{th} entry is δ_{ij} and note that

$$\begin{aligned}\|\boldsymbol{x}_m - \boldsymbol{x}^\dagger\|_{\Sigma_0^{-1}} &= \|\Sigma_0^{-\frac{1}{2}}(\boldsymbol{x}_m - \boldsymbol{x}^\dagger)\|_2 \\ &= \left(\sum_{i=1}^d |\boldsymbol{e}_i^\top \Sigma_0^{-\frac{1}{2}} \boldsymbol{x}_m - \boldsymbol{e}_i^\top \Sigma_0^{-\frac{1}{2}} \boldsymbol{x}^\dagger|^2\right)^{\frac{1}{2}} \\ &\leq \|\boldsymbol{x}_0 - \boldsymbol{x}^\dagger\|_{\Sigma_0^{-1}} \left(\sum_{i=1}^d \boldsymbol{e}_i^\top \Sigma_0^{-\frac{1}{2}} \Sigma_m \Sigma_0^{-\frac{1}{2}} \boldsymbol{e}_i\right)^{\frac{1}{2}} \qquad \text{(from Eq. (4.8), (4.9))} \\ &= \|\boldsymbol{x}_0 - \boldsymbol{x}^\dagger\|_{\Sigma_0^{-1}} \sqrt{\operatorname{tr}\left(\Sigma_0^{-\frac{1}{2}} \Sigma_m \Sigma_0^{-\frac{1}{2}}\right)} \\ &= \|\boldsymbol{x}_0 - \boldsymbol{x}^\dagger\|_{\Sigma_0^{-1}} \sqrt{\operatorname{tr}(\Sigma_m \Sigma_0^{-1})}\end{aligned}$$

where the last line uses the fact that the trace is invariant under cyclic permutation of the argument. □

Note that this result is extremely conservative, particularly when compared to the later contraction result in Proposition 4.3.5. This is due to the later result exploiting structure in the search directions constructed in Section 4.3.1. Nevertheless, it is also possible to bound the rate of contraction of the posterior covariance itself:

Proposition 4.2.4.

$$\operatorname{trace}(\Sigma_m \Sigma_0^{-1}) = d - m$$

Proof. We have that

$$\begin{aligned}
\operatorname{trace}(\Sigma_m \Sigma_0^{-1}) &= \operatorname{trace}(I - \Sigma_0 A^\top S_m \Lambda_m^{-1} S_m^\top A) \\
&= \operatorname{trace}(I) - \operatorname{trace}(\Sigma_0 A^\top S_m \Lambda_m^{-1} S_m^\top A) \\
&= \operatorname{trace}(I) - \operatorname{trace}(\underbrace{S_m^\top A \Sigma_0 A^\top S_m}_{=\Lambda_m} \Lambda_m^{-1}) \\
&= d - m
\end{aligned}$$

where the third line uses the fact that the trace of a matrix is invariant under cyclic permutation of the argument. □

This result highlights an intuitive but somewhat disappointing property of the PNM in Proposition 4.2.1. Since the search directions are arbitrary, after observing m search directions $\boldsymbol{x}^\dagger$ has been identified perfectly in an m-dimensional linear subspace, and so it seems intuitive that uncertainty about $\boldsymbol{x}^\dagger$ should decrease at a linear rate after adjusting for the weighting of the space provided by Σ_0. Nevertheless, in light of the exponential rate in Theorem 4.1.7, this linear convergence rate seems unsatisfying.

4.2.2 Correspondence with the Conjugate Gradient Method

In this section we examine the correspondence of the posterior mean $\boldsymbol{x}_m$ described in Proposition 4.2.1 with the CG method. It is frequently the case that Bayesian probabilistic numerical methods have some classical numerical method as their posterior mean, due to the characterisation of the conditional mean of a probability distribution as an optimal element of the underlying space. In this finite-dimensional setting this is made clear in Proposition 4.2.2. By comparing this to the optimality property obtained in Theorem 4.1.7, the following result is clear:

Corollary 4.2.5. *Assume A is symmetric and positive-definite. Let $\boldsymbol{x}_0 = \boldsymbol{0}$ and $\Sigma_0 = A^{-1}$. Then, taking $S_m = S_m^{CG}$ to be the search directions from CG, Eq. (4.5) reduces to $\boldsymbol{x}_m = \boldsymbol{x}_m^{CG}$.*

Proof. The proof is immediate from inserting $\Sigma_0 = A^{-1}$ into Proposition 4.2.2, and comparing with the optimality condition in Corollary 4.1.5. □

The dependence of the prior covariance on A^{-1} makes this choice impractical, but the probabilistic interpretation of CG is nevertheless interesting. In the following section a set of search directions is constructed which both produces a CG-like conjugacy for an arbitrary prior covariance, and recovers the CG search directions in the case $\Sigma_0 = A^{-1}$.

4.3 BayesCG

In this section the algorithm referred to as BayesCG will be introduced and examined in detail. BayesCG is defined by a specific set of search directions, chosen to diagonalise Λ_m, and these will be introduced in Section 4.3.1. Then, in Section 4.3.2 it will be established that BayesCG is a Krylov subspace method, and a theoretical analysis will be conducted.

4.3.1 BayesCG Search Directions

Search directions which diagonalise Λ_m and define BayesCG will now be introduced. Note that unlike CG, the BayesCG search directions *do not* require A to be positive-definite, and are valid for arbitrary invertible A. We first present a simplification of the posterior from Proposition 4.2.1 under $A\Sigma_0A^\top$-orthonormal search directions[2], arising from the fact that under these search directions $\Lambda_m = I$.

Proposition 4.3.1 (Conjugate Search Directions $\implies$ Iterative Method)**.** *Assume that the search directions are $A\Sigma_0A^\top$-orthonormal. Then, $\boldsymbol{x}_m$ in Eq. (4.5) simplifies to*

$$\boldsymbol{x}_m = \boldsymbol{x}_{m-1} + \Sigma_0 A^\top \boldsymbol{s}_m (\boldsymbol{s}_m^\top \boldsymbol{r}_{m-1}).$$

Similarly, Σ_m can be computed iteratively as follows:

$$\Sigma_m = \Sigma_{m-1} - \Sigma_0 A^\top \boldsymbol{s}_m \boldsymbol{s}_m^\top A \Sigma_0.$$

Furthermore, to compute Σ_m post-hoc *it suffices to store the only the matrix $\Sigma_0 A^\top S_m \in \mathbb{R}^{d\times m}$.*

Proof. First, note that $\Lambda_m = I$ as the search directions $\{\boldsymbol{s}_i\}$, $i = 1, \dots, m$ are

[2] Note that this does not require that the search directions from Proposition 4.3.2 are used, but is valid for those search directions.

Q-orthonormal, where $Q = A\Sigma_0 A^\top$. Then, from Eq. (4.5):

$$\begin{aligned}
\boldsymbol{x}_m &= \boldsymbol{x}_0 + \Sigma_0 A^\top S_m S_m^\top \boldsymbol{r}_0 \\
&= \boldsymbol{x}_0 + \Sigma_0 A^\top \begin{bmatrix} S_{m-1} & \boldsymbol{s}_m \end{bmatrix} \begin{bmatrix} S_{m-1}^\top \\ \boldsymbol{s}_m^\top \end{bmatrix} \boldsymbol{r}_0 \\
&= \underbrace{\boldsymbol{x}_0 + \Sigma_0 A^\top S_{m-1} S_{m-1}^\top \boldsymbol{r}_0}_{=\boldsymbol{x}_{m-1}} + \Sigma_0 A^\top \boldsymbol{s}_m \boldsymbol{s}_m^\top \boldsymbol{r}_0.
\end{aligned}$$

It remains to show that $\boldsymbol{s}_m^\top \boldsymbol{r}_0 = \boldsymbol{s}_m^\top \boldsymbol{r}_{m-1}$. To this end, from Eq. (4.5) we have

$$\begin{aligned}
\boldsymbol{s}_m^\top \boldsymbol{r}_{m-1} &= \boldsymbol{s}_m^\top \boldsymbol{b} - \boldsymbol{s}_m^\top A \boldsymbol{x}_{m-1} \\
&= \boldsymbol{s}_m^\top \boldsymbol{b} - \boldsymbol{s}_m^\top \boldsymbol{x}_0 - \underbrace{\boldsymbol{s}_m^\top A \Sigma_0 A^\top S_{m-1}^\top}_{=0} \boldsymbol{r}_0 \\
&= \boldsymbol{s}_m^\top \boldsymbol{r}_0.
\end{aligned}$$

For the posterior covariance, note that with $\Lambda_m = I$ we have, from Eq. (4.6):

$$\begin{aligned}
\Sigma_m &= \Sigma_0 - \Sigma_0 A^\top S_m S_m^\top A \Sigma_0 \\
&= \Sigma_0 - \sum_{i=1}^{m} \Sigma_0 A^\top \boldsymbol{s}_i \boldsymbol{s}_i^\top A \Sigma_0 \\
&= \Sigma_{m-1} - \Sigma_0 A^\top \boldsymbol{s}_m \boldsymbol{s}_m^\top A \Sigma_0
\end{aligned}$$

as required. Further, from the first line it is clear that storage of $\Sigma_0 A^\top S_m$ is sufficient to compute Σ_m. This completes the proof. □

Note that this posterior differs slightly from that in Proposition 4.2.1. In Proposition 4.2.1 information was provided by $\boldsymbol{s}_m^\top \boldsymbol{r}_0$, while in Proposition 4.3.1 it is provided by $\boldsymbol{s}_m^\top \boldsymbol{r}_{m-1}$. It is straightforward to show that when search directions are conjugate, the two expressions are equivalent:

$$\begin{aligned}
\boldsymbol{s}_m^\top \boldsymbol{r}_{m-1} &= \boldsymbol{s}_m^\top \boldsymbol{b} - \boldsymbol{s}_m^\top A \boldsymbol{x}_{m-1} \\
&= \boldsymbol{s}_m^\top \boldsymbol{b} - \boldsymbol{s}_m^\top A \boldsymbol{x}_0 - \underbrace{\boldsymbol{s}_m^\top A \Sigma_0 A^\top S_{m-1}}_{=0} \boldsymbol{r}_0 = \boldsymbol{s}_m^\top \boldsymbol{r}_0. \qquad (4.10)
\end{aligned}$$

Use of $\boldsymbol{s}_m^\top \boldsymbol{r}_{m-1}$ reduces the memory requirements of computing Eq. (4.5) and is thus slightly preferred.

The next result provides an iterative method for constructing a set of $A\Sigma_0 A^\top$-conjugate search directions, which are termed the *BayesCG search directions*.

Proposition 4.3.2 (Bayesian Conjugate Gradient Method)**.** *Denote* $\tilde{\boldsymbol{s}}_1 = \boldsymbol{b} - A\boldsymbol{x}_0$ *and* $\boldsymbol{s}_1 = \tilde{\boldsymbol{s}}_1 / \|\tilde{\boldsymbol{s}}_1\|_{A\Sigma_0 A^\top}$*. For* $m > 1$ *let*

$$\tilde{\boldsymbol{s}}_m = \boldsymbol{r}_{m-1} - \langle \boldsymbol{s}_{m-1}, \boldsymbol{r}_{m-1} \rangle_{A\Sigma_0 A^\top} \boldsymbol{s}_{m-1} \tag{4.11}$$

and let $\boldsymbol{s}_m = \tilde{\boldsymbol{s}}_m / \|\tilde{\boldsymbol{s}}_m\|_{A\Sigma_0 A^\top}$*. Then for each* m*, the set* $\{\boldsymbol{s}_i\}_{i=1}^m$ *is* $A\Sigma_0 A^\top$*-orthonormal, and as a result* $\Lambda_m = I$*.*

Proof. See Appendix C.1. □

BayesCG is termed a *conjugate gradient* method because the search directions arise from gradient descent on a particular function, subject to a conjugacy requirement. The search directions used are *not* the same as those from CG, apart from in the special case when $\Sigma_0 = A^{-1}$. The posterior mean may also be thought of as a generalised conjugate gradient method in the sense of Gutknecht [1993].

Note that the search directions in Proposition 4.3.2 depend on $\boldsymbol{x}^\dagger$ through their dependence on $\boldsymbol{b}$. Specifically, assuming that $\boldsymbol{x}_0 = \boldsymbol{0}$ the first search direction $\boldsymbol{s}_1 = \boldsymbol{b} = A\boldsymbol{x}^\dagger$. This means that the first piece of information is given by

$$\boldsymbol{s}_1^\top A\boldsymbol{x}^\dagger = (\boldsymbol{x}^\dagger)^\top A^\top A\boldsymbol{x}^\dagger$$

which is *nonlinear* in $\boldsymbol{x}^\dagger$. Thus, the conditioning procedure that is followed in Proposition 4.3.1 is not strictly correct, as the information used is not technically linear. The impact of this disconnect will be explored in detail in Section 4.6.1, and represents an important line of future research for this method.

4.3.2 BayesCG as a Krylov Subspace Method

In this section it will be shown that BayesCG is a Krylov subspace method, which will result in a faster convergence rate for the posterior mean than that elicited in Proposition 4.2.3. For convenience, let $K_m^* := \boldsymbol{x}_0 + \Sigma_0 A^\top K_m(A\Sigma_0 A^\top, \boldsymbol{r}_0)$. Then, we have the following result.

Proposition 4.3.3. *The search directions from Proposition 4.3.2 satisfy*

$$\mathrm{range}(S_m) = K_m(A\Sigma_0 A^\top, \boldsymbol{r}_0).$$

Proof. See Appendix C.2. □

Corollary 4.3.4. *The BayesCG mean $\boldsymbol{x}_m$ satisfies*

$$\boldsymbol{x}_m = \operatorname*{arg\,min}_{\boldsymbol{x} \in K_m^*} \|\boldsymbol{x} - \boldsymbol{x}^\dagger\|_{\Sigma_0^{-1}}.$$

Proof. This is immediate from application of Proposition 4.3.3 in Proposition 4.2.2. □

Note that these results also provide a new perspective on the previous observation that Proposition 4.2.1 has a posterior mean that coincides with CG when $\Sigma_0 = A^{-1}$. Since, for this choice of Σ_0, both the affine space K_m^* and the norm minimised coincide with those from CG, it is clear that the posterior mean under this prior should coincide with the CG estimate.

The last theoretical result in this section establishes a convergence rate for BayesCG which echoes that from CG, given in Theorem 4.1.7.

Proposition 4.3.5.

$$\frac{\|\boldsymbol{x}_m - \boldsymbol{x}^\dagger\|_{\Sigma_0^{-1}}}{\|\boldsymbol{x}_0 - \boldsymbol{x}^\dagger\|_{\Sigma_0^{-1}}} \leq 2 \left(\frac{\sqrt{\kappa(\Sigma_0 A^\top A)} - 1}{\sqrt{\kappa(\Sigma_0 A^\top A)} + 1} \right)^m .$$

Proof. See Appendix C.3. □

Note that this rate is identical to that from CG, but with $\kappa(A)$ replaced with $\kappa(\Sigma_0 A^\top A)$. However, since we have that $\kappa(A^\top A) \geq \kappa(A)$, the convergence rate for BayesCG may be worse than that for CG unless Σ_0 is chosen to reduce the condition number of $\kappa(\Sigma_0 A^\top A)$. This idea will be examined further in the next section.

Also note that the rate of convergence in Proposition 4.3.5 is significantly faster than that elicited in Proposition 4.2.3, owing to the fact that the search directions have been chosen in such a way as to accelerate convergence. However, the equality in Proposition 4.2.4 remains unchanged by this choice of search directions; thus while the posterior *mean* will converge at an exponential rate, the posterior *covariance* contracts linearly when its size is measured by $\operatorname{trace}(\Sigma_m \Sigma_0^{-1})$. This suggests that the posterior will be conservative in general and this is verified empirically in Section 4.6.1.

4.4 Prior Choice

Having introduced the search directions which define BayesCG, and thus identified the information which will be used, it remains to describe the prior. Since, owing to the fact that observations are noiseless, the prior distribution completely determines posterior uncertainty, an appropriate choice is critical. In Section 4.4.1 we will begin by discussing the prior covariance structure. This will be followed in Section 4.4.2 by the introduction of a hierarchical prior which is designed to automatically scale to the problem at hand.

4.4.1 Covariance Structure

As already discussed, the prior choice $\Sigma_0 = A^{-1}$ yields a posterior mean which coincides with the solution estimate produced by CG. However from a statistical perspective, correspondence with a classical numerical method does not in itself justify the use of A^{-1} as the prior covariance. In this section we will discuss some alternative choices of Σ_0 that are more probabilistically justified.

Natural Prior

Information about $\boldsymbol{x}^\dagger$ is only available through interrogation of $\boldsymbol{b}$, so taking inspiration from Owhadi [2015] it seems natural to place a prior on $\boldsymbol{b}$ rather than on $\boldsymbol{x}^\dagger$. This is motivated by the fact that $\boldsymbol{b}$ is the object about which we obtain information, and so is perhaps easier to reason about than $\boldsymbol{x}^\dagger$. Furthermore, placing a prior on $\boldsymbol{b}$ is equivalent to placing a prior on $\boldsymbol{x}^\dagger$ in the Gaussian case, since basic properties of linear projections of Gaussian distributions coupled with the relationship in Eq. (4.1) imply that

$$\boldsymbol{b} \sim \mathcal{N}(\boldsymbol{b}_0, \Sigma_0) \iff \boldsymbol{x}^\dagger \sim \mathcal{N}(A^{-1}\boldsymbol{b}_0, A^{-1}\Sigma_0 A^{-\top}).$$

Since $\boldsymbol{b}$ is *a-priori* unknown, a natural prior to use would be $\boldsymbol{b} \sim \mathcal{N}(\boldsymbol{0}, I)$, which implies that $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{0}, (A^\top A)^{-1})$. This prior is just as impractical as taking $\Sigma_0 = A^{-1}$, but has the interesting property that with the BayesCG search directions from Proposition 4.3.2, convergence occurs in a single iteration:

$$\begin{aligned} \boldsymbol{s}_1 &= \frac{\boldsymbol{r}_0}{\|\boldsymbol{r}_0\|_{A\Sigma_0 A^\top}} = \frac{\boldsymbol{r}_0}{\|\boldsymbol{r}_0\|_2} \\ \implies \boldsymbol{x}_1 &= \boldsymbol{x}_0 + \frac{(A^\top A)^{-1} A^\top \boldsymbol{r}_0 (\boldsymbol{r}_0^\top \boldsymbol{r}_0)}{\|\boldsymbol{r}_0\|_2^2} \\ &= \boldsymbol{x}_0 + A^{-1}(\boldsymbol{b} - A\boldsymbol{x}_0) = \boldsymbol{x}^\dagger \end{aligned}$$

Here the first line uses the fact that the search directions are $A\Sigma_0 A^\top$-orthonormal, while the second line applies the form for $\boldsymbol{x}_1$ from Proposition 4.3.1.

Preconditioner Prior

A more practical choice of prior covariance uses the intuition above, but replaces A^{-1} with a *preconditioner* for system, as discussed in Section 4.1.3. Recall that a preconditioner is a matrix P which serves as an approximate inverse for A, in that P^{-1} can be computed in significantly less than $\mathcal{O}(d^3)$ operations and has the property $\kappa(P^{-1}A) \ll \kappa(A)$. Where such a preconditioner is available this suggests the prior $\Sigma_0 = (P^\top P)^{-1}$.

Krylov Subspace Prior

A second practical approach uses results from numerical analysis to place the majority of the prior probability mass on a subspace of $\mathbb{R}^d$ in which we expect the solution to lie. In the present setting, this is accomplished by placing mass on a particular Krylov subspace. Consider

$$\boldsymbol{x}_K = \sum_{i=1}^{n} w_i M^{i-1} \boldsymbol{b} \tag{4.12}$$

where $n < d$ and the weights are taken to be $\boldsymbol{w} \sim \mathcal{N}(0, \Phi)$, with $\Phi \in \mathbb{R}^{n\times n}$ a positive-definite matrix. Equivalently $\boldsymbol{x}_K = Q_n \boldsymbol{w}$, where $Q_n \in \mathbb{R}^{d\times n}$ is a basis for $K_n(M, \boldsymbol{b})$, such as given by Arnoldi iteration [Liesen and Strakos, 2012, Section 2.4]. Noting that $\mathbb{E}(\boldsymbol{x}_K) = \boldsymbol{0}$, the covariance of $\boldsymbol{x}_K$ is given by

$$\mathbb{E}(\boldsymbol{x}_K \boldsymbol{x}_K^\top) = Q_n \Phi Q_n^\top$$

so that $\boldsymbol{x}_K \sim \mathcal{N}(\boldsymbol{0}, Q_n \Phi Q_n^\top)$.

Arnoldi iteration to generate Q_n has the same computational complexity as application of n iterations of BayesCG, so to prevent this cost from dominating the procedure, it is necessary to take $n < m \ll d$. Furthermore we note that no probability mass is placed outside of $K_n(M, \boldsymbol{b})$ by following this procedure, which is problematic as generally $\boldsymbol{x} \notin K_n(M, \boldsymbol{b})$ for $n < d$. To ensure that $\boldsymbol{x}^\dagger$ lies in the support of the prior, let $K_n^\perp(\boldsymbol{b}, M) = \mathbb{R}^d \setminus K_n(\boldsymbol{b}, M)$, and let $Q_n^\perp \in \mathbb{R}^{d-n\times d}$ denote a matrix with $\text{range}(Q_n^\perp) = K_n^\perp(M, \boldsymbol{b})$. Let $\boldsymbol{x}_K^\perp = Q_n^\perp \boldsymbol{w}^\perp$, where $\boldsymbol{w}^\perp \sim \mathcal{N}(\boldsymbol{0}, \varphi I)$ for some scaling parameter $\varphi \in \mathbb{R}$. Then, the Krylov subspace prior is given by:

$$\begin{aligned} \boldsymbol{x} &= \boldsymbol{x}_0 + \boldsymbol{x}_K + \boldsymbol{x}_K^\perp \\ &\sim \mathcal{N}\left(\boldsymbol{x}_0, Q_n \Phi Q_n^\top + \varphi Q_n^\perp (Q_n^\perp)^\top\right). \end{aligned}$$

There are several parameter choices associated with this approach:

Choice of M Given the analysis in Section 4.3.2, the most natural choice of Krylov subspace in which to place prior mass is $\Sigma_0 A^\top K_n(\Sigma_0 A^\top A, \Sigma_0 A^\top \boldsymbol{r}_0)$. However since this depends itself on the prior covariance which is to be determined, an alternative choice must be made. Setting $M = A$ also seems natural due to the rapid convergence of CG shown in Theorem 4.1.7. Using this choice, the Krylov subspace prior can loosely be thought of as encoding a numerical analyst's intuition that projection of $\boldsymbol{x}^\dagger$ into the Krylov subspace $K_m(A, \boldsymbol{r}_0)$ results in a small error in an appropriate norm.

Selection of Φ and φ With $M = A$, the rate of convergence of CG shown in Proposition 4.2.1 can be used to decide how much mass to place on each direction in $K_n(A, \boldsymbol{b})$, and thus determine Φ and φ. Let $\xi < 1$. Owing to the assumed orthogonality of the columns of Q_n, Φ is taken to be a diagonal matrix with $\Phi_{ii} = \left[2\sigma\xi^i\right]^2$. Like ξ, $\sigma \in \mathbb{R}$ is a scaling parameter. Proposition 4.2.1 suggests fixing the scale parameters introduced to $\xi = \frac{\kappa(A)-1}{\kappa(A)+1}$ and $\sigma = \|\boldsymbol{x}^\dagger\|_A$. Note however that since these quantities are not computable without significantly more computational effort than required to run BayesCG, some approximation must be used, or values chosen based on the user's prior belief. The choice of this approximation was not explored herein, however.

The remaining parameter, φ, describes how much mass is placed on $K_m^\perp(A, \boldsymbol{b})$. The argument above suggests the constraint $\varphi < [2\sigma\xi^{i+2}]^2$, but the precise choice of φ should again be based on the user's prior belief on how rapidly CG will converge in practise for a particular problem.

Computation of $Q_n^\perp$ Note that

$$K_n^\perp(A, \boldsymbol{b}) = \{\boldsymbol{v} : Q_n^\top \boldsymbol{v} = 0\}$$

so that computing $Q_n^\perp$ is equivalent to finding the null-space of $Q_n^\top$. The QR-decomposition of a (non-square) matrix $Q_n^\top$ consists of two matrices $Q \in \mathbb{R}^{d\times d}$ and $R \in \mathbb{R}^{d\times n}$ such that $Q_n^\top = QR$. The matrix Q can be used to determine the null space of $Q_n^\top$. Split Q as $Q = [Q_1, Q_2]$, where $Q_1 \in \mathbb{R}^{d\times(n+1)}$ and $Q_2 \in \mathbb{R}^{d\times(d-n-1)}$. Then it holds that Q_2 is an orthonormal matrix with range$(Q_2) = K_n^\perp(A, \boldsymbol{b})$. The computational complexity of this procedure is $\mathcal{O}(d^3)$, however, so in practise a more expedient method for computing or estimating the required null space would need to be used; again, this was not explored.

4.4.2 Covariance Scale

In this section, calibration of the prior will be discussed. For the output of BayesCG to be useful the prior must be appropriately calibrated. This is particularly important for BayesCG owing to the data-driven nature of the search directions and the poor calibration exhibited in Section 4.6.1. Loosely speaking, it is desirable that the posterior covariance should reflect the distance between $\boldsymbol{x}_m$ and $\boldsymbol{x}^\dagger$. The approaches described here each consider the introduction of a hyperparameter ν into the prior specification, so the following modified prior is used:

$$p(\boldsymbol{x}|\nu) = \mathcal{N}(\boldsymbol{x}_0, \nu\Sigma_0) \tag{4.13}$$

with $\boldsymbol{x}_0, \Sigma_0$ as before, while $\nu \in \mathbb{R}^+$ is a scale parameter to be estimated. Thus we obtain a generalised version of the prior in Eq. (4.4), which is recovered when $\nu = 1$. Even more generally, the entire posterior covariance could be treated as unknown and endowed with a prior on positive-definite matrices, such as the inverse-Wishart prior, but this approach was not considered.

Here two approaches to estimation of ν are discussed. First, it is proposed that the prior scale should be treated as as a hyperparameter and learned from the observed data, similar to an approach from Bayesian linear regression [Gelman et al., 2014]. Secondly, we discuss empirically calibrating of the covariance scale to match an estimate of the error at iteration m. This approach is philosophically unsatisfying, but seems to yield better-calibrated UQ than the hierarchical approach.

Hierarchical Prior

In this section $\nu \in \mathbb{R}^+$ is endowed with Jeffreys' (improper) reference prior:

$$p(\nu) \propto \nu^{-1}.$$

This "hyperprior" has conjugacy properties with Eq. (4.13), so that both the posterior marginal distributions $p(\nu|\boldsymbol{y}_m)$ and $p(\boldsymbol{x}|\boldsymbol{y}_m)$ can be obtained in closed-form. For the following proposition, IG denotes an inverse-gamma distribution, while MVT_m denotes a multivariate t distribution with m degrees of freedom.

Proposition 4.4.1 (Hierarchical BayesCG)**.** *When $p(\boldsymbol{x}|\nu) = \mathcal{N}(\boldsymbol{x}_0, \nu\Sigma_0)$ and $p(\nu) \propto \nu^{-1}$, the posterior marginal for ν is given by*

$$p(\nu|\boldsymbol{y}_m) = \mathsf{IG}\left(\frac{m}{2}, \frac{1}{2}\boldsymbol{r}_0^\top S_m \Lambda_m^{-1} S_m^\top \boldsymbol{r}_0\right)$$

while the posterior marginal for $\boldsymbol{x}$ is given by

$$p(\boldsymbol{x}|\boldsymbol{y}_m) = \mathsf{MVT}_m\left(\boldsymbol{x}_m, \frac{\boldsymbol{r}_0^\top S_m \Lambda_m^{-1} S_m^\top \boldsymbol{r}_0}{m}\Sigma_m\right).$$

When the search directions are $A\Sigma_0 A^\top$-orthonormal, this simplifies to

$$\begin{aligned} p(\nu|\boldsymbol{y}_m) &= \mathsf{IG}\left(\frac{m}{2}, \frac{m}{2}\nu_m\right) \\ p(\boldsymbol{x}|\boldsymbol{y}_m) &= \mathsf{MVT}_m\left(\boldsymbol{x}_m, \nu_m\Sigma_m\right) \end{aligned}$$

where $\nu_m := \|S_m^\top \boldsymbol{r}_0\|_2^2/m$.

Proof. We first compute the posterior marginal for ν. Note that

$$p(\nu|\boldsymbol{y}) \propto p(\boldsymbol{y}|\nu)p(\nu)$$

where

$$\boldsymbol{y}|\nu \sim \mathcal{N}(S_m^\top A\boldsymbol{x}_0, \nu\Lambda_m).$$

We thus have that:

$$p(\nu|\boldsymbol{y}) \propto \nu^{-\frac{m}{2}-1}\exp\left(-\frac{1}{2\nu}\boldsymbol{r}_0^\top S_m\Lambda_m^{-1}S_m^\top\boldsymbol{r}_0\right)$$

which is the unnormalised density of an $\mathsf{IG}\left(\frac{m}{2}, \frac{1}{2}\boldsymbol{r}_0^\top S_m\Lambda_m^{-1}S_m^\top\boldsymbol{r}_0\right)$ distribution, as required. Now to determine the posterior marginal for $\boldsymbol{x}$, we have the following:

$$\begin{aligned} p(\boldsymbol{x}|\boldsymbol{y}) &= \int_0^\infty p(\boldsymbol{x}|\nu,\boldsymbol{y})p(\nu|\boldsymbol{y})\,\mathrm{d}\nu \\ &\propto \int_0^\infty \nu^{-1-(m+d)/2}\exp\left(-\nu^{-1}K(\boldsymbol{x})\right)\,\mathrm{d}\nu \qquad (4.14) \end{aligned}$$

where

$$K(\boldsymbol{x}) := \frac{1}{2}\left[\boldsymbol{r}_0^\top S_m\Lambda_m^{-1}S_m^\top\boldsymbol{r}_0 + (\boldsymbol{x}-\boldsymbol{x}_m)^\top\Sigma_m^{-1}(\boldsymbol{x}-\boldsymbol{x}_m)\right]$$

Eq. 4.14 is recognised as the integral of an unnormalised inverse-Gamma density, so that we can immediately find:

$$\begin{aligned} p(\boldsymbol{x}|\boldsymbol{y}) &\propto \Gamma(m+d)K(\boldsymbol{x})^{-\frac{1}{2}(m+d)} \\ &\propto \left[1 + \frac{1}{m}(\boldsymbol{x}-\boldsymbol{x}_m)^\top\left\{\frac{\boldsymbol{r}_0^\top S_m\Lambda_m^{-1}S_m^\top\boldsymbol{r}_0}{m}\Sigma_m\right\}^{-1}(\boldsymbol{x}-\boldsymbol{x}_m)\right]^{-\frac{1}{2}(m+d)}. \end{aligned}$$

This is recognised as an unnormalised multivariate t distribution, so that

$$p(\boldsymbol{x}|\boldsymbol{y}) = \mathsf{MVT}_m \left(\boldsymbol{x}_m, \frac{\boldsymbol{r}_0^\top S_m \Lambda_m^{-1} S_m^\top \boldsymbol{r}_0}{m} \Sigma_m \right)$$

as required. □

Since $\boldsymbol{r}_0$ reflects the initial error $\boldsymbol{x}_0 - \boldsymbol{x}^\dagger$, the quantity ν_m can be thought of as describing the difficulty of the problem. Thus in this approach the scale of the posterior should be automatically calibrated. However, as will be shown in Section 4.6.1, this nevertheless yields a poorly calibrated posterior.

Empirical Calibration

In this section we discuss an empirical procedure for calibrating ν. This is designed to compensate for the mismatch between the exponential convergence rate exhibited in Proposition 4.3.5 and the linear rate of covariance contraction in Proposition 4.2.4. The proposed approach is to construct an error indicator over the course of the algorithm, and then use this to adjust an appropriate measure of spread of the posterior to match that error prediction. It should be emphasised that this approach is *ad-hoc*, and should not be considered as the unique, best approach to calibration; it is presented here only to demonstrate that a calibration procedure that provides more realistic uncertainty quantification can be constructed.

Constructing the Error Indicator The aim here is to construct a proxy for the true error by constructing a computable upper bound for the error $\|\boldsymbol{x}_m - \boldsymbol{x}^\dagger\|_2$. Let

$$z_i := \|\boldsymbol{x}_i - \boldsymbol{x}_{i-1}\|_2 \, .$$

The proposed approach is to perform a simple regression on the values $\{z_i\}_{i=1}^m$, and use the fitted model $\nu(i)$ to extrapolate for the error required. Justified by the exponential convergence rate of BayesCG, a log-linear function $\nu(i) = \exp(a + bi)$ has been used for the regression model.

To derive the error indicator we use the following bound derived from the triangle inequality:

$$\begin{aligned} \|\boldsymbol{x}_m - \boldsymbol{x}^\dagger\|_2 &\leq \sum_{i=m+1}^{d} \|\boldsymbol{x}_i - \boldsymbol{x}_{i-1}\|_2 \\ &\approx \sum_{i=m+1}^{d} \nu(i) =: \alpha_m. \end{aligned}$$

Thus α_m provides an approximate upper-bound for $\|\boldsymbol{x}_m - \boldsymbol{x}^\dagger\|_2$.

Fitting the Posterior Next we adjust the spread of the posterior based on the approximate upper-bound α_m on the true error. This requires some measure of the posterior spread, and for the ease of computability the measure $\text{trace}(\nu_m \Sigma_m)$ was used. Thus, to be concrete, ν_m is selected so that:

$$\begin{aligned} \text{trace}(\nu_m \Sigma_m) &= \alpha_m \\ \implies \nu_m &= \frac{\alpha_m}{\text{trace}(\Sigma_m)}. \end{aligned}$$

Note that, since α_m appears in the numerator and provides an approximate upper bound for the true error, the UQ provided will still be conservative in general.

4.5 Implementation

BayesCG is presented as an algorithm in Algorithm 4.3. A Python implementation can be found at github.com/jcockayne/bcg.

There are several issues and complications which arise when implementing BayesCG in practise, which will now be discussed.

4.5.1 Further Simplication of BayesCG

Several simplifications are exploited in Algorithm 4.3. These are described here in detail. First, for stability it is recommended to compute $A\Sigma_0 A^\top$-*orthogonal* directions rather than enforcing orthonormality; this is due to the tendency for $\|\tilde{\boldsymbol{s}}_m\|_{A\Sigma_0 A^\top}$ to become very small over the course of the iteration. Second, two coefficients must be calculated: one for the purposes of updating $\boldsymbol{x}_m$, and one for updating $\tilde{\boldsymbol{s}}_m$. Let $Q = A\Sigma_0 A^\top$, and express these quantities as

$$\begin{aligned} \boldsymbol{x}_m &= \boldsymbol{x}_{m-1} + \alpha_m \Sigma_0 A^\top \tilde{\boldsymbol{s}}_m \\ \tilde{\boldsymbol{s}}_m &= \boldsymbol{r}_{m-1} + \beta_{m-1} \tilde{\boldsymbol{s}}_{m-1} \end{aligned}$$

Algorithm 4.3 Computation of the posterior distribution described in Proposition 4.3.1 with the optimisations described in Section 4.5.1. For clarity, all required matrix-vector multiplications have been made explicit, but for efficiency these should be calculated once-per-loop and stored. From the output, Σ_m computed as $\Sigma_m = \Sigma_0 - \Sigma_F \Sigma_F^\top$.

```
procedure BayesCG($A, \boldsymbol{b}, \boldsymbol{x}_0, \Sigma_0, \epsilon, m_{\max}$)    ▷ ($\epsilon$ the tolerance)
    $\Sigma_F$ initialised to a matrix of size $(d \times 0)$    ▷ ($m_{\max}$ the maximum # iterations)
    $\boldsymbol{r}_0 \leftarrow \boldsymbol{b} - A\boldsymbol{x}_0$
    $\tilde{\boldsymbol{s}}_1 \leftarrow \boldsymbol{r}_0$
    $\tilde{\nu}_0 \leftarrow 0$
    for $m = 1, \dots, m_{\max}$ do
        $E^2 \leftarrow \tilde{\boldsymbol{s}}_m^\top A \Sigma_0 A^\top \tilde{\boldsymbol{s}}_m$
        $\alpha_m \leftarrow \frac{\boldsymbol{r}_{m-1}^\top \boldsymbol{r}_{m-1}}{E^2}$
        $\boldsymbol{x}_m \leftarrow \boldsymbol{x}_{m-1} + \alpha_m \Sigma_0 A^\top \tilde{\boldsymbol{s}}_m$
        $\boldsymbol{r}_m \leftarrow \boldsymbol{r}_{m-1} - A\boldsymbol{x}_m$
        $\Sigma_F \leftarrow [\Sigma_F, \Sigma_0 A^\top \tilde{\boldsymbol{s}}_m / E]$
        $\tilde{\nu}_m \leftarrow \tilde{\nu}_{m-1} + \frac{\boldsymbol{r}_{m-1}^\top \boldsymbol{r}_{m-1}}{E^2}$
        if $\|r_m\|_2 < \epsilon$ then
            break
        end if
        $\beta_m \leftarrow \frac{\boldsymbol{r}_m^\top \boldsymbol{r}_m}{\boldsymbol{r}_{m-1}^\top \boldsymbol{r}_{m-1}}$
        $\tilde{\boldsymbol{s}}_{m+1} \leftarrow \boldsymbol{r}_m + \beta_m \tilde{\boldsymbol{s}}_m$
    end for
    $\nu_m \leftarrow \tilde{\nu}_m / m$
    return $\boldsymbol{x}_m, \Sigma_F, \nu_m$
end procedure
```

where

$$
\begin{aligned}
\alpha_m &= \frac{\tilde{\boldsymbol{s}}_m^\top \boldsymbol{r}_{m-1}}{\|\tilde{\boldsymbol{s}}_m\|_Q^2} \\
\beta_m &= -\frac{\boldsymbol{r}_m Q \tilde{\boldsymbol{s}}_m}{\|\tilde{\boldsymbol{s}}_m\|_Q^2}.
\end{aligned}
$$

Now, using the expression for $\tilde{\boldsymbol{s}}_m$ from Eq. (4.11), note that

$$
\begin{aligned}
\alpha_m &= \frac{\boldsymbol{r}_{m-1}^\top (\boldsymbol{r}_{m-1} - \beta_m \tilde{\boldsymbol{s}}_{m-1})}{\|\tilde{\boldsymbol{s}}_m\|_Q^2} \\
&= \frac{\boldsymbol{r}_{m-1}^\top \boldsymbol{r}_{m-1}}{\|\tilde{\boldsymbol{s}}_m\|_Q^2}
\end{aligned}
$$

since we have from Lemma C.1.1 that $\tilde{\boldsymbol{s}}_m^\top \boldsymbol{r}_m = 0$. Furthermore, from Eq. (C.3), we have that

$$\begin{aligned}
\boldsymbol{r}_m^\top Q \boldsymbol{s}_m &= \frac{\boldsymbol{r}_m^\top \boldsymbol{r}_{m-1} - \boldsymbol{r}_m^\top \boldsymbol{r}_m}{\boldsymbol{s}_m^\top \boldsymbol{r}_{m-1}} \\
&= -\frac{\boldsymbol{r}_m^\top \boldsymbol{r}_m}{\boldsymbol{s}_m^\top \boldsymbol{r}_{m-1}} \\
&= -\frac{\boldsymbol{r}_m^\top \boldsymbol{r}_m}{\boldsymbol{r}_{m-1}^\top \boldsymbol{r}_{m-1}} \|\tilde{\boldsymbol{s}}_m\|_Q^2
\end{aligned}$$

so that

$$\beta_m = \frac{\boldsymbol{r}_m^\top \boldsymbol{r}_m}{\boldsymbol{r}_{m-1}^\top \boldsymbol{r}_{m-1}}$$

These two simplifications have been found (empirically) to improve the stability of computation of Proposition 4.3.1 in Algorithm 4.3.

4.5.2 Numerical Breakdown of Conjugacy

It is well known [see Liesen and Strakos, 2012] that, after a certain iteration, the CG search directions exhibit a breakdown of conjugacy when computed numerically. This is in spite of the mathematical conjugacy which can be proven to hold. The reason for this phenomenon is an accumulation of floating point error, and since the procedure by which the BayesCG search directions are computed is essentially the same as that by which the CG search directions are constructed, they share this property. The breakdown of conjugacy is mitigated to some extent by using the alternative information described in Eq. (4.10), which exploits "local" conjugacy [Meurant, 2006], but ultimately this only delays the conjugacy breakdown. A side-effect is that, while mathematically convergence is guaranteed in d iterations, computationally $m > d$ iterations may required.

While the impact of this on CG is well-known, in the present setting its impact on the posterior covariance must also be discussed. When the search directions are not conjugate $\Lambda_m \neq I$, and so the simplification exploited in Proposition 4.3.1 no longer hold, inducing *overconfidence* in the resulting posterior. How this interacts with the conflicting rates from Proposition 4.3.5 and Proposition 4.2.4 will be examined numerically in Section 4.6, but an analytical treatment of floating point error in the context of BayesCG is beyond the scope of this thesis. However, to provide a benchmark, "exact" posterior against which to compare the *batch-computed search directions* are also introduced here. These are obtained by the full Gram-Schmidt orthogonalisation procedure:

$$\begin{aligned}\tilde{\boldsymbol{s}}_m^C &:= \boldsymbol{r}_{m-1} - \sum_{i=1}^{m-1} \langle \boldsymbol{s}_i^C, \boldsymbol{r}_{m-1} \rangle_{A\Sigma_0 A^\top} \boldsymbol{s}_{m-1}^C \\ \boldsymbol{s}_m^C &:= \tilde{\boldsymbol{s}}_m^C / \|\tilde{\boldsymbol{s}}_m^C\|_{A\Sigma_0 A^\top}\end{aligned}$$

Note that while these search directions are mathematically equivalent to BayesCG search directions, they maintain their conjugacy when computed.

4.5.3 Computational Cost

Depending on how it is implemented, BayesCG requires either one or two additional matrix-vector multiplications per iteration over CG[3]. Regardless, the cost of BayesCG is a constant factor higher than the cost of CG, i.e. $\mathcal{O}(md^2)$. For the batch-computed search directions an additional loop of complexity $\mathcal{O}(m)$ must be performed, and so the cost of BayesCG with these search directions is $\mathcal{O}(m^2d^2)$. This assumes that A and Σ_0 are dense matrices; if they are sparse the cost is driven instead by the number of nonzero entries in each matrix, rather than their dimension.

4.5.4 Termination Criteria

An additional use of the posterior distribution might be to determine when the posterior has contracted to a desired tolerance, thus providing a probabilistic termination criterion. Recall from Proposition 4.2.3 that $\boldsymbol{x}_m$ approaches $\boldsymbol{x}^*$ at a rate bounded by $\sigma_m := \sqrt{\text{trace}(\Sigma_m \Sigma_0^{-1})}$, and from Proposition 4.2.4 that $\text{trace}(\Sigma_m \Sigma_0^{-1}) = d - m$. This leads to the following termination criterion based upon Proposition 4.4.1:

$$\sigma_m^2 := \text{trace}(\Sigma_m \Sigma_0^{-1}) \times \nu_m = (d-m)\nu_m. \tag{4.15}$$

The algorithm would then terminate when $\sigma_m^2 < \epsilon$, for some user-specified tolerance. However, as discussed, Proposition 4.2.3 is extremely conservative, and since Proposition 4.3.5 establishes a much faster rate of convergence for $\|\boldsymbol{x}_m - \boldsymbol{x}^\dagger\|_{\Sigma_0^{-1}}$ this is likely to be an overcautious stopping criterion. Furthermore, note that ν_m is not uniformly decreasing with m, which further casts doubt on the suitability of Eq. (4.15) as a termination criterion. Thus, in practise a standard CG termination criterion is used; see Golub and Van Loan [2013, Section 11.3.8] for more detail. Fur-

[3] Note that if storage is not a limiting factor, the matrix $A\Sigma_0$ can be computed and stored, reducing the number of matrix-vector multiplications required by 1 per iteration

ther research is needed to establish whether a termination criterion can be derived from the posterior distribution.

4.6 Numerical Results

In this section two numerical studies are presented. In Section 4.6.1, a simulation study is presented in which theoretical results presented are verified empirically. In Section 4.6.2, BayesCG is applied to the EIT problem introduced in Section 2.4.2.

4.6.1 Simulation Study

This first experiment is a simulation study, with the goal of verifying numerically the theoretical properties proven earlier in this chapter.

For the simulation study a matrix A was generated by randomly drawing its eigenvalues $\lambda_1, \dots, \lambda_d$ from an exponential distribution with parameter $\gamma = 10$. The dimension d was fixed to $d = 100$. These were used as input to the MATLAB function `sprandsym`, which generates a random sparse symmetric matrix with the supplied eigenvalues. The sparsity parameter $\mathrm{nnz}(A)$ was set to 20%. After constructing A, many random solutions $\boldsymbol{x}^\dagger$ were drawn independently from a reference distribution $\mu_{\mathrm{ref}} = \mathcal{N}(0, I)$, after which $\boldsymbol{b} = A\boldsymbol{x}^\dagger$ was computed to provide the full system required.

The BayesCG algorithm was then run for $m = d = 100$ iterations on each of the random systems. Several choices of prior discussed in Section 4.4 were used:

- $\Sigma_0 = I$.
- $\Sigma_0 = A^{-1}$.
- $\Sigma_0 = (P^\top P)^{-1}$ for P a preconditioner obtained by computing an incomplete Cholesky decomposition with zero fill-in [Ajiz and Jennings, 1984]. This decomposition is simply a Cholesky decomposition in which an (approximate) factor $\hat{L}$ is constructed, and enforced to have same sparsity structure as A. The preconditioner is then given by $P = \hat{L}\hat{L}^\top$.
- Σ_0 the Krylov prior from Section 4.4.1. Here the parameters were set to $n = 20$, $\sigma = \|\boldsymbol{x}^\dagger\|_A$, $\xi = \frac{\kappa(A)-1}{\kappa(A)+1}$ and $\varphi = 0.01$ to give low weight to the complement space.

Note that not all of these choices are practical; the choice $\Sigma_0 = A^{-1}$ is impractical for reasons already discussed, while the Krylov prior involves computing the condition number of A and the norm of $\boldsymbol{x}^\dagger$. While these quantities might be estimated efficiently, this was not explored in this section. However, it should be emphasised that the choice $\Sigma_0 = (P^\top P)^{-1}$ *is* practical, as the incomplete Cholesky decomposition can be computed at substantially lower cost than the $\mathcal{O}(d^3)$ cost of computing the complete Cholesky. Three sets of search directions are used: the sequentially and batch computed search directions are compared with a set of orthonormal search directions selected at random to provide a benchmark. These random directions were obtained by sampling an orthogonal matrix from a uniform distribution using the algorithm of Diaconis and Shahshahani [1987].

Posterior Mean

In Figure 4.1 the convergence of the posterior mean $\boldsymbol{x}_m$ from BayesCG under the choices of prior discussed above, is contrasted with the solution estimate from CG for the test problems described above. When $\Sigma_0 = I$, the convergence rate of the BayesCG mean is significantly slower than the CG solution estimate. This is to be expected, since with this choice of prior the condition number that governs convergence in Corollary 4.3.4 is $\kappa(A^\top A) = \kappa(A)^2$, so slower convergence is natural. The randomly selected search directions also exhibit slower convergence that more closely matches the slower rate elicited in Proposition 4.2.3. For $\Sigma_0 = A^{-1}$, the posterior mean is identical to the estimate for $\boldsymbol{x}_m$ obtained from CG. The fastest rate of convergence is achieved when using the preconditioner prior $\Sigma_0 = (P^\top P)^{-1}$, providing a strong motivation for such priors when preconditioners are available. Note however that while the rate achieved for this prior is faster than the convergence rate of CG, if a preconditioned CG algorithm were employed convergence would be at a faster rate governed by $\kappa(P^{-1}A)$.

Batch computed directions are examined in the lower row of Fig. 4.1. The main difference appears to be that convergence is achieved in $m = d$ iterations, which is not the case for the sequentially computed directions, owing to the aforementioned breakdown of conjugacy. Lastly, note that with the Krylov subspace prior significant numerical instability is observed starting at $m = 20$. This does not occur with the batch computed directions, where a jump in the convergence rate is seen at this iteration.

Posterior Covariance

We now turn to an evaluation of the posterior covariance, plotted in Fig. 4.2. The statistic $\operatorname{trace}(\Sigma_m)$ is plotted for each test problem and prior covariance choice, to

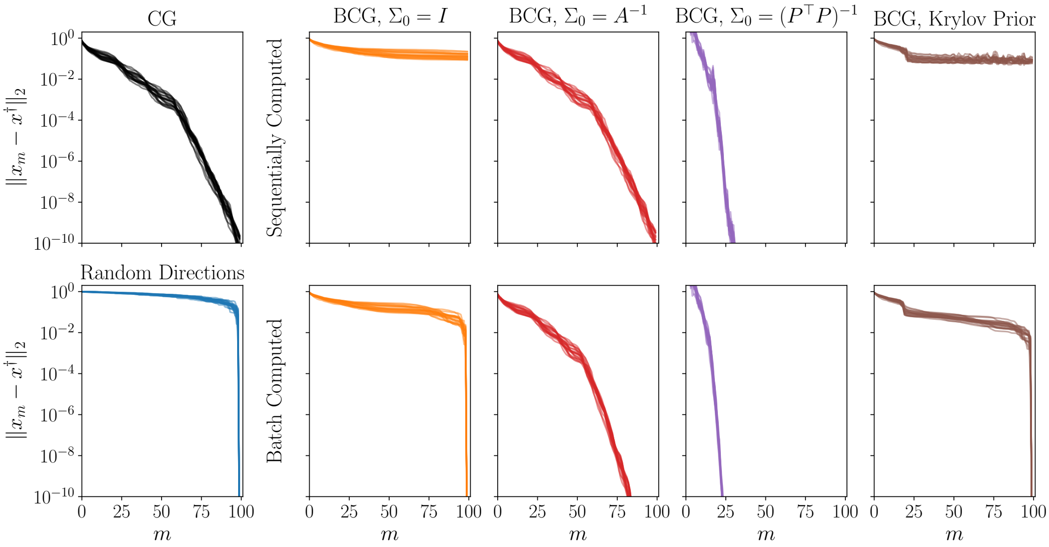


Figure 4.1: Convergence in mean of BayesCG. Computation of the error $\|\boldsymbol{x}_m - \boldsymbol{x}^\dagger\|_2$ for the test problems described in Section 4.6.1. CG (top left) was compared to variants of BayesCG (right) with different prior covariances Σ_0. The search directions are either sequentially computed (top row) or batch computed (bottom row). For comparison, random search directions are displayed in the bottom left panel.

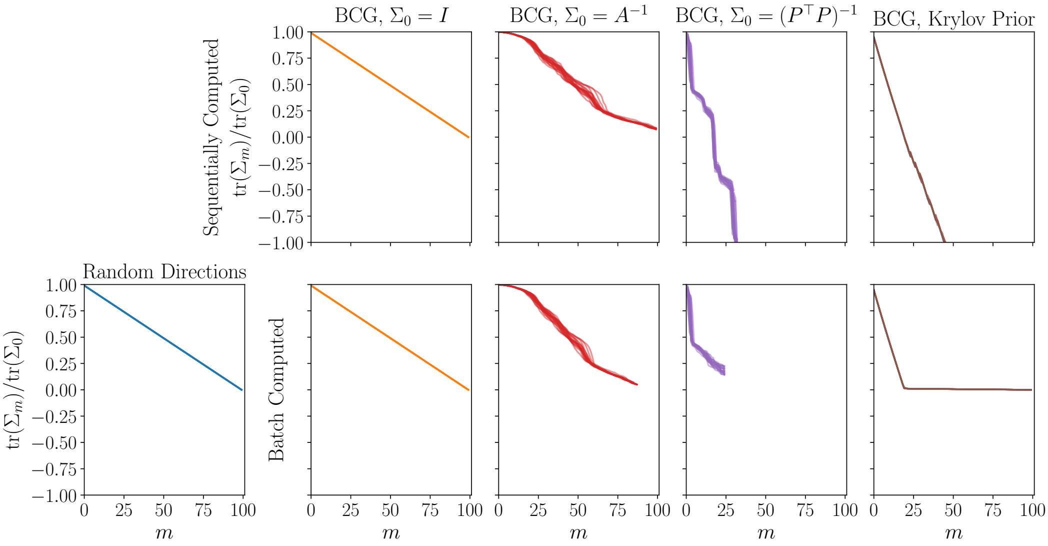


Figure 4.2: Width of the posterior covariance from BayesCG, as measured by trace(Σ_m). The experimental setup was as described in Figure 4.1, but the statistic plotted is instead trace(Σ_m)/trace(Σ_0).

empirically verify Proposition 4.2.4. Clearly, the faster convergence in the mean exhibited when the BayesCG search directions are used does not transfer to a faster rate of convergence in the posterior covariance. Throughout, a roughly linear rate of convergence is observed, as expected from Proposition 4.2.4. Furthermore, comparing batch computed and sequentially computed directions, the impact of the breakdown of conjugacy is clear in the right two columns, when the posterior covariance is shown to take *negative* values at around $m = 20$.

Uncertainty Quantification As discussed in Section 4.3.2, the UQ provided by BayesCG is expected to be conservative in general. As a result, in this section we assess the quality of the UQ using a statistical test. The same experimental setup was used, but BayesCG was run for $m = 10$ iterations rather than $m = d$ to ensure that residual uncertainty is present. Batch computed search directions were used throughout, as the main interest is in the uncertainty quantification properties of the posterior rather than the numerical properties of the algorithm.

We first consider the version of BayesCG from Proposition 4.3.1, with the Gaussian posterior. To assess the UQ, we make the *ansatz* that, if the posterior is to be considered well-calibrated, we should expect $\boldsymbol{x}^\dagger$ to be a plausible draw from the posterior distribution. The posterior covariance Σ_m is singular, of rank $d - m$. However assessing uncertainty in its null space is irrelevant, as in this space $\boldsymbol{x}^\dagger$ is known exactly. Since Σ_m is positive semidefinite, it has the singular-value decomposition

$$\Sigma_m = U \begin{bmatrix} D & 0_{d-m,m} \\ 0_{m,d-m} & 0_{m,m} \end{bmatrix} U^\top$$

where $0_{m,n}$ denotes an $m \times n$ matrix of zeroes, $D \in \mathbb{R}^{(d-m)\times(d-m)}$ is diagonal and $U \in \mathbb{R}^{d\times d}$ is an orthogonal matrix. The first $d - m$ columns of U, denoted U_{d-m}, form a basis of $\operatorname{range}(\Sigma_m) \subset \mathbb{R}^d$, the space in which $\boldsymbol{x}^\dagger$ is still uncertain. Under the *ansatz* we can then show that

$$\begin{aligned} &U_{d-m} D^{-\frac{1}{2}} U_{d-m}^\top (\boldsymbol{x}^\dagger - \boldsymbol{x}_m) \sim \mathcal{N}(\mathbf{0}, I_{d-m}) \\ \implies \quad &Z(\boldsymbol{x}^\dagger) := \| D^{-\frac{1}{2}} U_{d-m}^\top (\boldsymbol{x}^\dagger - \boldsymbol{x}_m) \|_2^2 \sim \chi^2_{d-m} \end{aligned}$$

where here $I_n \in \mathbb{R}^{n\times n}$ is the identity matrix. Note that the pre-factor U_{d-m} is dropped from the final expression as the Euclidean norm is unitarily invariant.

The procedure for evaluation of the UQ is then to draw many test problems $\boldsymbol{x}^\dagger \sim \mu_{\text{ref}}$, evaluate $Z(\boldsymbol{x}^\dagger)$ for each and compare the empirical distribution of this statistic to χ^2_{d-m}. When the posterior distribution is well-calibrated, the empirical

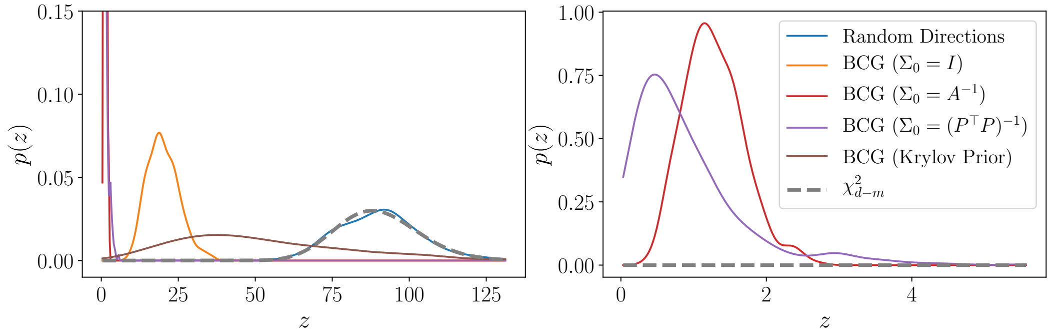


Figure 4.3: Evaluation of the uncertainty quantification from Gaussian BayesCG, under different choices of prior covariance and search directions. Kernel density estimates of Z are displayed based on 500 sampled test problems. The theoretical distribution of Z is plotted for comparison. The right panel zooms in on the area of the z-axis in which the statistics from $\Sigma_0 = A^{-1}$ and $\Sigma_0 = (P^\top P)^{-1}$ are concentrated.

distribution should resemble a χ^2_{d-m} distribution. When the posterior is conservative the distribution will exhibit a “left-shift” in its density, as $\boldsymbol{x}_m$ is closer to $\boldsymbol{x}^\dagger$ than indicated by the posterior covariance. An excessively confident posterior will be right-shifted.

In Figure 4.3 the empirical distribution of Z is presented as a kernel density estimate based upon 500 sampled problems, for the same range of priors as before. The random search directions are the only choice which provides well-calibrated UQ, owing to the fact that these directions do not have an implicit dependence on $\boldsymbol{x}^\dagger$. Conversely, the UQ is the most conservative for the prior covariances $\Sigma_0 = I$, A^{-1} and $(P^\top P)^{-1}$, i.e. when the prior contains the most information about the solution and, consequently, the convergence rate in Proposition 4.3.5 is fastest. The Krylov subspace prior seems to provide better calibrated UQ; while it does not exactly match the empirical distribution, it at least places a larger amount of its mass near that distribution. Thus, this prior goes some way towards addressing the poor UQ provided.

Similar arguments to above can be used to assess the multivariate t posterior from Proposition 4.4.1. When $S \sim \mathcal{N}(\mathbf{0}, I)$, $T \sim \mathsf{MVT}_m(\boldsymbol{\mu}, \Sigma)$ and $U \sim \chi^2_m$, it holds that:

$$\begin{aligned} & \frac{1}{\sqrt{m}} U_{d-m} D^{-\frac{1}{2}} U_{d-m}^\top (T - \boldsymbol{\mu}) \overset{d}{=} \frac{S}{\sqrt{U}} \\ \implies \quad & \frac{1}{m} \| D^{-\frac{1}{2}} U_{d-m}^\top (T - \boldsymbol{\mu}) \|_2^2 \overset{d}{=} \frac{\|S\|_2^2}{U} \end{aligned}$$

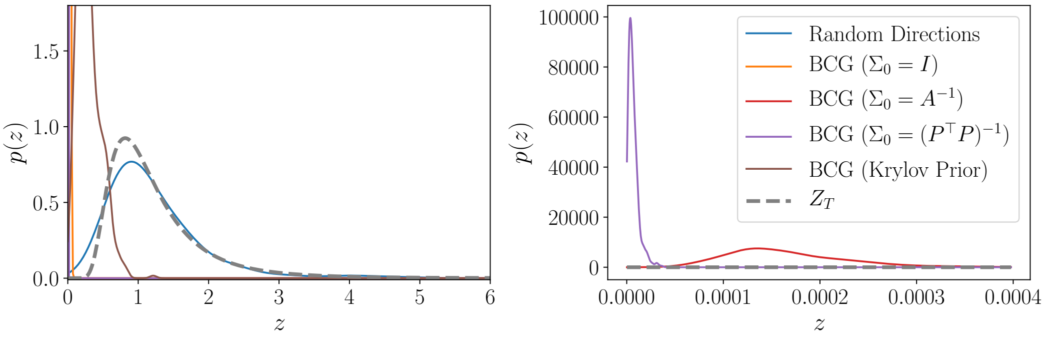


Figure 4.4: Assessment of the uncertainty quantification from the multivariate t version of BayesCG described in Proposition 4.4.1. The same prior covariances and search directions were used as in Figure 4.3.

Applied to the posterior from Proposition 4.4.1, we have $\boldsymbol{\mu} = \boldsymbol{x}_m$ and $\Sigma = \Sigma_m$. Furthermore $\|S\|_2^2 \sim \chi^2_{d-m}$. Lastly, multiplying both sides by $m/(d-m)$ we have

$$Z(\boldsymbol{x}^\dagger) := \frac{1}{d-m}\|D^{-\frac{1}{2}}U_{d-m}^\top(\boldsymbol{x}_m - \boldsymbol{x}^\dagger)\|_2^2 \overset{d}{=} \frac{\frac{\|S\|_2^2}{(d-m)}}{\frac{U}{m}}.$$

The ratio on the right-hand-side can be shown to follow an $F(d-m, m)$ distribution. Empirical distributions of this statistic are again plotted in Fig. 4.4, however the quality of the posterior calibration appears to be much the same as for the Gaussian posterior, with the BayesCG posterior providing the UQ closest to the theoretical distribution being that obtained from the Krylov prior.

Lastly, Fig. 4.5 shows the uncertainty quantification obtained when ν_m is calibrated empirically using the procedure described in Section 4.4.2. Compared to Fig. 4.3 and Fig. 4.4 the posterior appears to be generally better-calibrated for the more practical priors $\Sigma_0 = (P^\top P)^{-1}$ and the Krylov prior, but the quality of the calibration is still poor when $\Sigma_0 = A^{-1}$. Nevertheless this suggests that empirical calibration procedures could be used to compensate for the implicit dependence of the search directions on $\boldsymbol{x}^\dagger$.

4.6.2 Electrical Impedance Tomography

We now proceed to a more practical application of BayesCG, to the EIT problem described in Section 2.4.2. The EIT model used in this section is the CEM, and the experimental setup setup and data is the same as from Isaacson et al. [2004], as described in that section.

A finite-element discretisation was used to solve the weak form of Eq. (2.14).

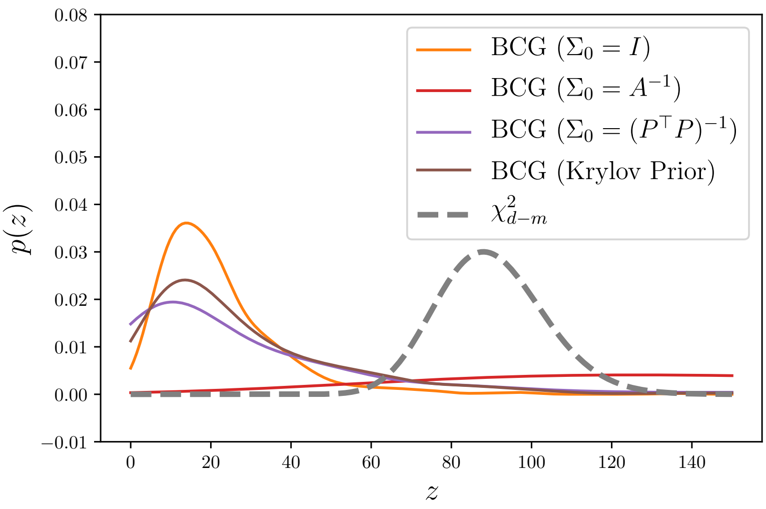


Figure 4.5: Uncertainty quantification from the empirical calibration procedure described in Section 4.4.2.

The tank was modelled as a unit circle, and the electrodes were assumed to be equispaced with each occupying precisely $1/64^{\text{th}}$ of the boundary. Thus, each electrode had length $\pi/32$ and there was a distance of $\pi/32$ between each neighbouring pair of electrodes on ∂D. The contact impedances were taken to be $\zeta_i = 1$, $i = 1, \dots, 32$ owing to a lack of information about the actual impedances.

The triangulations required for FEA were generated using the Python package `meshpy`, which was configured to ensure N_d equally sized elements were present on the boundary. N_d is always taken to be a multiple of the number of boundary electrodes; this ensures that each electrode is supported on an equal number of boundary electrodes after discretisation. Figure 4.6 shows an example of a triangulation with $N_d = 64$.

FEA yields a sparse linear system $A\boldsymbol{x}^\dagger = \boldsymbol{b}$, where A is a positive-definite *stiffness matrix*. Standard piecewise linear basis functions were used, and the computations were perfomed using the `FEniCS` finite-element package. Since the discretisation error incurred by FEA is known to be driven by some measure of the mesh size, an extremely fine discretisation of the domain might be required to eliminate discretisation error. When BayesCG is used, the system resulting from a very fine mesh can be solved for a small number of iterations $m \ll d$. In this regime the discretisation error from the linear solver is dominant, and it will be demonstrated that the UQ provided by BayesCG can be of benefit in the inverse problem of estimating the conductivity field $\kappa(\boldsymbol{x})$.

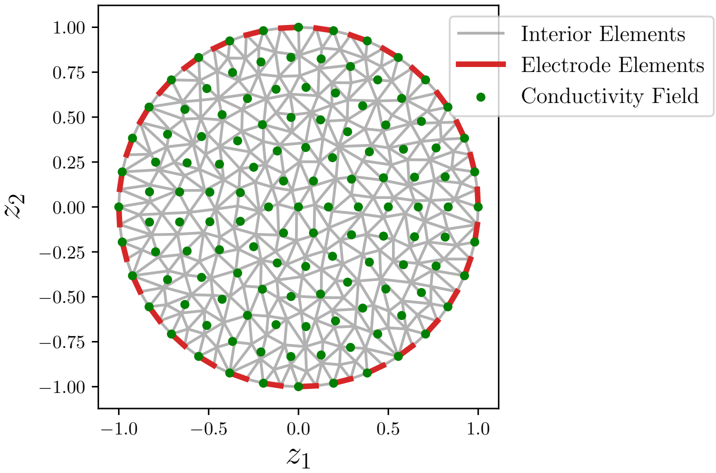


Figure 4.6: Finite-element discretisation used for the EIT experiment described in Section 4.6.2, for $N_d = 64$. Red lines indicate the elements which correspond to electrodes. Green dots show the locations at which the posterior conductivity field was sampled.

Forward Solution

We first examine the application of BayesCG to the forward problem, for an arbitrary fixed stimulation pattern. Similar to the figures in the previous section, in Fig. 4.7 error $\|\boldsymbol{x}_m - \boldsymbol{x}^\dagger\|$ is computed for the (practical) covariance choices $\Sigma_0 = I$ and $\Sigma_0 = (P^\top P)^{-1}$. As in the previous section, an incomplete Cholesky factorisation was used to compute the preconditioner P. Three mesh resolutions are presented: $N_d = 64, 128$ and 256. The matrix A naturally depends on a conductivity field κ, and for this both samples from the prior distribution over κ , and the inferred posterior mean in the inverse problem were each used. These are described in more detail in the next section.

As before, when $\Sigma_0 = I$ convergence is slow, but this is accelerated when using $\Sigma_0 = (P^\top P)^{-1}$. Since this problem is obtained from a practical example rather than sampled arbitrarily, it is useful to know that the same observations transfer.

Inverse Problem

We now turn to the problem of propagating uncertainty from BayesCG from the *forward* solution into the inverse problem of inferring κ. The Bayesian inversion framework described in Section 2.4 was used. Since κ is required to be strictly

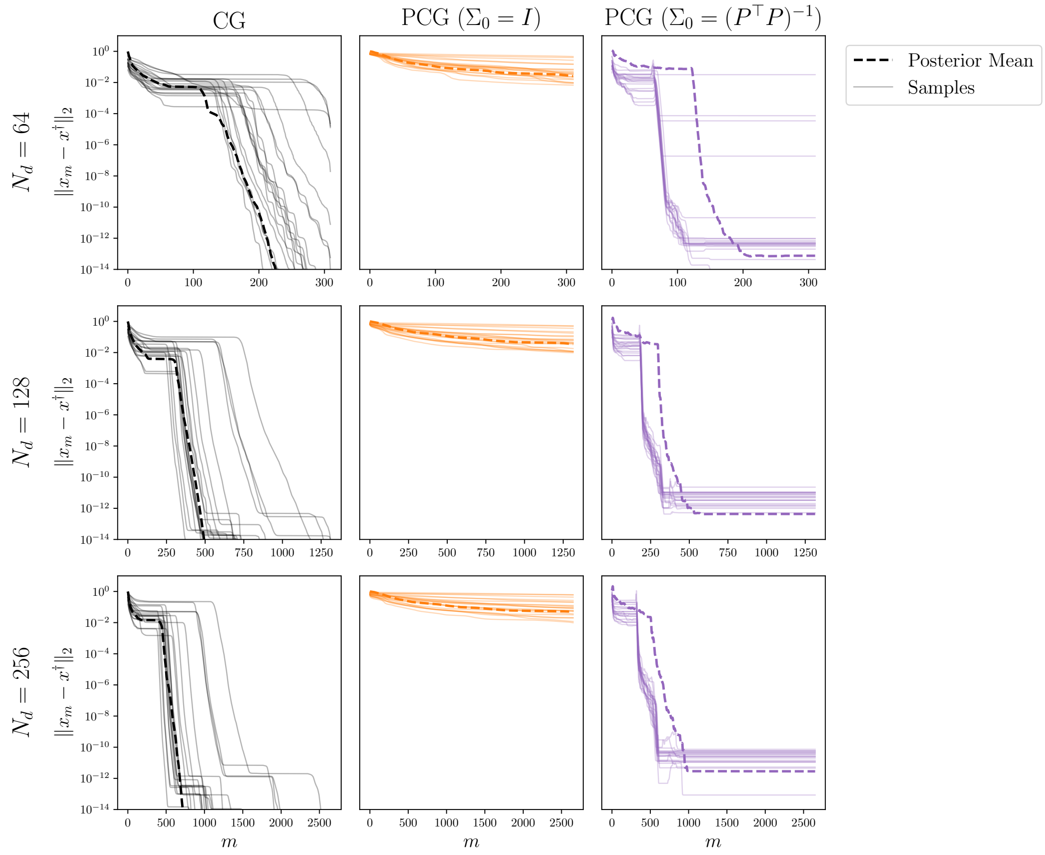
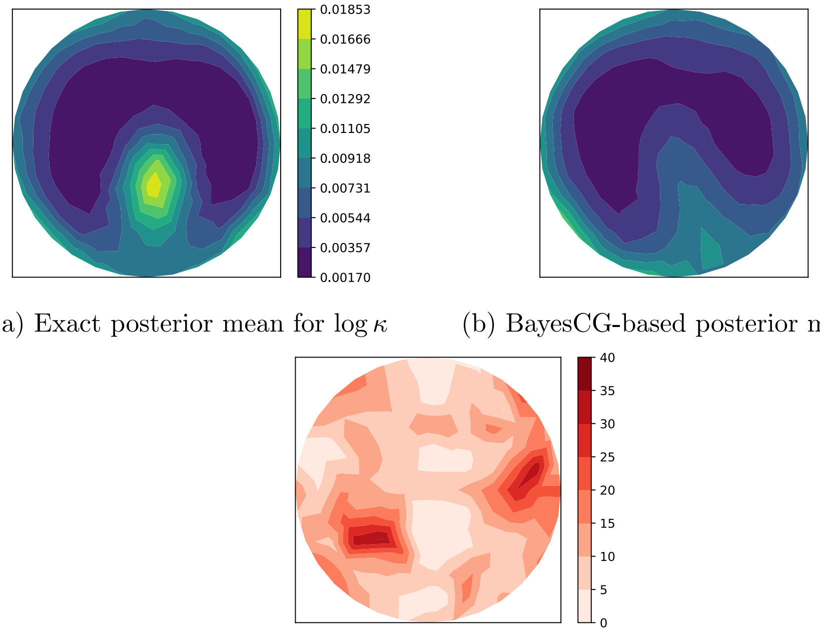


Figure 4.7: Convergence of the posterior mean for the linear system arising from FEA discretisation of the PDE in Eq. (2.14), for a number of different conductivity fields and discretisation resolutions. The solid lines represent the convergence of the BayesCG posterior mean for conductivity fields sampled from the prior μ_κ. The dashed lines are for the conductivity field obtained as the the mean of μ_κ^V.

positive, a log-Gaussian prior was placed upon it:

$$\log(\kappa(\boldsymbol{z})) = \theta(\boldsymbol{z}) \sim \mathcal{GP}(0, k)$$

where k was taken to be a Matérn 5/2 covariance function as given in Eq. (2.8). The parameters were set to $\ell = 1.0$ and $\sigma = 9.0$, to ensure that the posterior distribution lies in a region of high prior mass. Note that in general the conductivity field would not be expected to be smooth, as the boundaries of the agar displayed in Fig. 2.1 has hard boundaries at the edges of the targets. To accommodate hard boundaries a technique such as Bayesian level set invesion [Dunlop et al., 2016] could be used, but this was not explored here as the smoother prior was still found to provide reasonable reconstructions.



(a) Exact posterior mean for $\log \kappa$

(b) BayesCG-based posterior mean for $\log \kappa$

(c) Ratio of point-wise posterior standard deviation, for BayesCG-based compared to exact.

Figure 4.8: Comparison of the posterior distribution over the conductivity field κ, when using the modified potential $\hat{\Phi}_m$ with a BayesCG forward solver, compared to the standard potential Φ with the forward problem solved using CG.

To propagate uncertainty, the proposed approach is to derive a new potential $\hat{\Phi}$ by marginalising the posterior distribution output from BCG in the likelihood used in the inverse problem. It is straightforward to show that, when a Gaussian likelihood is used, this results in the new potential:

$$\hat{\Phi}_m(V|\kappa) = \frac{1}{2}\sum_{i=1}^{N_e-1} \|V_i - U_{i,m}\|^2_{(\Gamma+\Sigma^U_{i,m})^{-1}} + C$$

Here, $U_{i,m}$ and $\Sigma^U_{i,m}$ are the portions of $\boldsymbol{x}_m$ and Σ_m output from BayesCG that correspond to the electrode voltages U_i from the CEM, at iteration m of BayesCG. C is a constant independent of V that does not affect the inferences obtained, and so can be ignored. Thus, the new likelihood $\exp(-\hat{\Phi}_m(V|\kappa))$ is still proportional to a Gaussian, but with a covariance inflated by $\Sigma^U_{i,m}$ to account for the precision of the solver. It will be shown that replacing Φ from Eq. (2.13) with $\hat{\Phi}_m$ leads in turn to a posterior distribution for the conductivity field that is widened to account for the forward solver accuracy.

In Fig. 4.8 the posterior distribution over the conductivity field is displayed, under both potentials Φ and $\hat{\Phi}_m$, with $m = 80$ iterations. In both cases the discretisation with $N_d = 64$ was used. Comparing Fig. 4.8a with Fig. 4.8b, it is clear that qualitatively many of the features of the posterior are visible. In Fig. 4.8c the ratio of the pointwise posterior standard deviation between the BayesCG posterior and the exact posterior is plotted. Clearly this is uniformly larger throughout the domain, so the use of $\hat{\Phi}_m$ has resulted in a posterior distribution which is wider to account for the solver inaccuracy. Overall, the integrated standard deviation over the domain is 0.0365 for BayesCG, while for the exact posterior it is 0.0046.

4.7 Discussion

In this chapter we have presented a PNM for the solution of finite-dimensional linear systems. While the rate of convergence of the posterior mean from BayesCG was shown to be near to the convergence speed of CG, the fact that the search directions depend upon $\boldsymbol{x}^\dagger$ results in a posterior covariance that is overly conservative. This is the primary deficiency with this method, and forms the focus of future research in these methods.

The study of PNM for linear systems is of profound importance, as such systems are among the most ubiquitous in numerical analysis. Understanding how PNM can be constructed in this setting gives valuable insight and has a vast set of potential applications, including the application to PDEs presented in Section 4.6.2. A different approach to PNM for PDEs will be introduced in the next chapter; this is based on the generalisation of the methods introduced in this chapter to infinite-dimensional linear systems.

Chapter 5

The Probabilistic Meshless Method

> "Perhaps some day in the dim future it will be possible to advance the computations faster than the weather advances... But that is a dream."
>
> —*Lewis Fry Richardson, 1922*

In this chapter the *probabilistic meshless method* (PMM) will be introduced. The PMM is a novel BPNM for the solution of the *strong form* of linear elliptic PDEs, and as such is a method defined on function space. It follows the typical pattern of conjugate BPNM on function spaces, in that a Gaussian prior is posited and conditioned on linear projections of the solution. Once again the extension to Bayesian inverse problems is explored through an application to the EIT example described in Section 2.4.2, albeit in the form of the simplified PEM, rather than the CEM.

This chapter is structured as follows: In Section 5.1, related work will be outlined, and in Section 5.2 the PMM is introduced. In Section 5.3 we discuss the use of the PMM in inverse problems, in a similar manner as discussed in Section 4.6.2. In Section 5.4 we present two numerical examples: a simulation study based on a simple one-dimensional PDE, and another application to EIT. We conclude in Section 5.5 with some discussion.

5.1 Introduction

In this section we begin by introducing meshfree methods in Section 5.1.1. These methods are the closest methods in the literature on numerical solution of PDEs to the BPNM presented in this chapter. In Section 5.1.2 we describe other PNM for solving PDEs that have been proposed.

5.1.1 Meshfree Methods

While the methods for solving PDEs introduced in Section 2.2.1 are widely used in practise, they are nevertheless not uniformly the most appropriate solvers. A particular challenge can be the procedure by which the domain is discretised. When D has a complex geometry, the procedure of specifying a suitable grid or mesh over it can become an extremely delicate, often manual process. This is amplified in problems when the domain evolves over time, such as in cracking or warping problems, as the domain might need to be regularly re-meshed to avoid degeneracy [Rabczuk and Belytschko, 2007].

To combat these challenges, the class of *meshless* or *meshfree* methods has recently emerged [Fasshauer, 1997; Liu, 2002]. These methods are defined, somewhat ambiguously, by their not relying on the construction of a regular mesh over the solution domain. A non-exhaustive list of such methods includes collocation methods [Fasshauer, 1999; Kansa, 1990], element-free Galerkin methods [Belytschko et al., 1994], meshless Petrov-Galerkin methods [Atluri and Shen, 2005] and smoothed-particle hydrodynamics [Gingold and Monaghan, 1977], though this chapter will focus on collocation methods. In addition to the advantage of not relying on construction of a mesh, meshless methods often also have the virtue of yielding significantly simpler computer code, as can be seen in Fasshauer [2007]. Conversely, prominent texts on FEM go so far as to advocate that interested users should prefer using professional software to attempting to implement the methods themselves [Johnson, 1988, Section 1.9].

5.1.2 Existing PNMs for PDEs

Several other PNM for the solution of differential equations have been proposed. The literature has focussed more on the solution of ordinary differential equations than partial differential equations. The principle challenge here is that linear ODEs are generally so straightforward to solve as to be considered trivial. Thus, the nonlinear setting has been the focus of attention, in which the conjugacy properties of Gaussian distributions cannot be exploited and so approximations must be made. Schober et al. [2014], Schober et al. [2018] and Kersting and Hennig [2016] follow a series of approximations to pose the probabilistic solution of ODEs as a filtering method [Law et al., 2015], and output a Gaussian distribution over the solution to the ODE.

However, these methods are non-Bayesian as described in Appendix B, and the relationship between the distribution output by the PNM and the Bayesian posterior has not yet been established. Chkrebtii et al. [2016] constructs a method that similarly resembles a filter, but outputs a *nonparametric* posterior distribution whose relationship to the Bayesian posterior is nevertheless still unclear. Conrad et al. [2017] and Lie et al. [2019] take a different approach, instead introducing stochastic perturbations to a numerical approximation to the flow map resulting from classical numerical methods applied over small time intervals δt. The perturbations are calibrated to ensure that the convergence order of the underlying numerical method is maintained; however, this approach is fundamentally different from the others mentioned here in that no Bayesian interpretation is claimed.

Chkrebtii et al. [2016] also applied their method to parabolic (time-dependent) PDEs, but did not consider standard linear elliptic PDEs as in this chapter. Conrad et al. [2017] also considered application to PDEs, by instead introducing stochastic perturbations to finite-element basis functions, which has a similar interpretation to the approach that the authors pursued for ODEs, but nevertheless has a different philosophical interpretation to what is presented here. Wang et al. [2018] developed a fully Bayesian PNM for the solution of a particular class of ODEs using Lie group theory, but applicability was limited to those ODEs for which suitable Lie transformations exist. Furthermore, extensive manual computation was required to construct the solver and the resulting sampling algorithm was numerically challenging.

The approach described in this chapter was independently discovered in the works of Bilionis [2016] and Raissi et al. [2017], though from a more empirical standpoint and without the convergence analyses or applications to inverse problems described here. Raissi et al. [2018] develops their earlier work with applications to nonlinear PDEs which resemble the filtering approach in Chkrebtii et al. [2016], though again no theoretical guarantees were provided. Owhadi [2015] and Owhadi [2017] follow similar arguments, though the focus of these methods is on application to PDEs with *rough coefficients*, i.e. in which $\kappa(\boldsymbol{x})$ in Eq. (2.1) is a $C^0(D)$ function. Those works showed that, when information is obtained following a specific hierarchical procedure, recovery of the solution to the PDE can be obtained with near-linear computational cost.

5.2 Probabilistic Meshless Method

In this section the probabilistic meshless method (PMM) will be introduced. First, symmetric collocation method will be introduced in Section 5.2.1. This is a classical numerical method which is recovered as the posterior mean of the PMM for a particular choice of prior, echoing the relationship between CG and BayesCG presented in Chapter 4. In Section 5.2.2 the PMM will be introduced. The choice of prior will be discussed in Section 5.2.3, and theoretical analysis of the posterior will be presented in Section 5.2.4.

First some notation will be established. Consider an abstraction of the PDE in question. Let $D \subset \mathbb{R}^d$ with boundary ∂D be such that $\overline{D}$ is compact[1]. Let $\mathcal{H}(D)$, $\mathcal{H}_{\mathcal{A}}(D)$ and $H_{\mathcal{B}}(\partial D)$ each be separable Hilbert spaces of functions with inner products $\langle \cdot, \cdot \rangle$, $\langle \cdot, \cdot \rangle_{\mathcal{A}}$ and $\langle \cdot, \cdot \rangle_{\mathcal{B}}$ respectively. Introduce the bounded linear and elliptic operators $\mathcal{A} : \mathcal{H}(D) \to \mathcal{H}_{\mathcal{A}}(D)$ and $\mathcal{B} : \mathcal{H}(D) \to \mathcal{H}_{\mathcal{B}}(\partial D)$. Let $g \in \mathcal{H}_{\mathcal{A}}(D)$ and $b \in \mathcal{H}_{\mathcal{B}}(\partial D)$. Then, the problem of interest is the solution $u^\dagger \in \mathcal{H}(D)$ to the system of operator equations

$$\begin{aligned} \mathcal{A}u^\dagger(\boldsymbol{x}) &= g(\boldsymbol{x}) \quad && \boldsymbol{x} \in D \\ \mathcal{B}u^\dagger(\boldsymbol{x}) &= b(\boldsymbol{x}) \quad && \boldsymbol{x} \in \partial D. \end{aligned} \tag{5.1}$$

For concreteness, in the context of Eq. (2.1) the operator $\mathcal{A}$ can be associated with the operator $-\nabla \cdot \kappa(\boldsymbol{x})\nabla$, while $\mathcal{B}$ is the boundary trace operator which restricts $u(\boldsymbol{x})$ to ∂D.

5.2.1 Symmetric Collocation

We now introduce the symmetric collocation method, a method that has much in common with the PMM introduced in this chapter. For more details see the presentation in Fasshauer [1999, Section 3.1] and the references therein.

Symmetric collocation seeks an approximation $\hat{u}$ to $u^\dagger$ of the form:

$$\hat{u}(\boldsymbol{x}) := \sum_{i=1}^{m_{\mathcal{A}}} c_i^{\mathcal{A}} \bar{\mathcal{A}} k(\boldsymbol{x}, \boldsymbol{x}_i^{\mathcal{A}}) + \sum_{i=1}^{m_{\mathcal{B}}} c_i^{\mathcal{B}} \bar{\mathcal{B}} k(\boldsymbol{x}, \boldsymbol{x}_i^{\mathcal{B}}) \tag{5.2}$$

where

$$X^{\mathcal{A}} = \{\boldsymbol{x}_1^{\mathcal{A}}, \dots, \boldsymbol{x}_{m_{\mathcal{A}}}^{\mathcal{A}}\} \qquad X^{\mathcal{B}} = \{\boldsymbol{x}_1^{\mathcal{B}}, \dots, \boldsymbol{x}_{m_{\mathcal{B}}}^{\mathcal{B}}\}$$

and $X^{\mathcal{A}} \subset D$, $X^{\mathcal{B}} \subset \partial D$. Eq. (5.2) can be expressed more compactly by introducing

[1] i.e. closed and bounded

some notation. This notation is cumbersome to define, but amounts to "vectorizing" or "broadcasting" functions $f : D \to \mathbb{R}$ so that when an argument in D^n is passed, a vector in $\mathbb{R}^n$ is returned. Analogously, functions with two arguments will be broadcast into matrices. The fact that the PDE consists of multiple operator equations complicates this somewhat, but this intuition is nevertheless helpful.

For sets $X = \{\boldsymbol{x}_j\}_{j=1}^n$ and $X' = \{\boldsymbol{x}'_j\}_{j=1}^{n'}$, let $\mathcal{A}k(X, X')$, $\bar{\mathcal{A}}k(X, X')$ and $\mathcal{A}\bar{\mathcal{A}}k(X, X')$ each in $\mathbb{R}^{n \times n'}$ be defined as

$$\begin{aligned}
[\mathcal{A}k(X, X')]_{ij} &= \mathcal{A}k(\boldsymbol{x}_i, \boldsymbol{x}'_j) \\
[\bar{\mathcal{A}}k(X, X')]_{ij} &= \bar{\mathcal{A}}k(\boldsymbol{x}_i, \boldsymbol{x}'_j) \\
[\mathcal{A}\bar{\mathcal{A}}k(X, X')]_{ij} &= \mathcal{A}\bar{\mathcal{A}}k(\boldsymbol{x}_i, \boldsymbol{x}'_j).
\end{aligned}$$

Let

$$\mathcal{L} := \begin{bmatrix} \mathcal{A} \\ \mathcal{B} \end{bmatrix}, \quad \bar{\mathcal{L}} := \begin{bmatrix} \bar{\mathcal{A}} & \bar{\mathcal{B}} \end{bmatrix}.$$

Let $\mathcal{L}\bar{\mathcal{L}}k(X^{\mathcal{AB}}) \in \mathbb{R}^{(m_\mathcal{A}+m_\mathcal{B})\times(m_\mathcal{A}+m_\mathcal{B})}$ be given by

$$\mathcal{L}\bar{\mathcal{L}}k(X^{\mathcal{AB}}) := \begin{bmatrix} \mathcal{A}\bar{\mathcal{A}}k(X^\mathcal{A}, X^\mathcal{A}) & \mathcal{A}\bar{\mathcal{B}}k(X^\mathcal{A}, X^\mathcal{B}) \\ \bar{\mathcal{A}}\mathcal{B}k(X^\mathcal{B}, X^\mathcal{A}) & \mathcal{B}\bar{\mathcal{B}}k(X^\mathcal{B}, X^\mathcal{B}) \end{bmatrix}$$

and let $\mathcal{L}k(\boldsymbol{x}, X^{\mathcal{AB}}), \bar{\mathcal{L}}k(\boldsymbol{x}, X^{\mathcal{AB}}) \in \mathbb{R}^{m_\mathcal{A}+m_\mathcal{B}}$ be given by

$$\mathcal{L}k(\boldsymbol{x}, X^{\mathcal{AB}}) := \begin{bmatrix} \mathcal{A}k(\boldsymbol{x}, X^\mathcal{A}) \\ \mathcal{B}k(\boldsymbol{x}, X^\mathcal{B}) \end{bmatrix} \qquad \bar{\mathcal{L}}k(\boldsymbol{x}, X^{\mathcal{AB}}) := \begin{bmatrix} \bar{\mathcal{A}}k(\boldsymbol{x}, X^\mathcal{A}) \\ \bar{\mathcal{B}}k(\boldsymbol{x}, X^\mathcal{B}) \end{bmatrix}.$$

Lastly, let $\mathcal{L}k(X^{\mathcal{AB}}, \boldsymbol{x}) = \mathcal{L}k(\boldsymbol{x}, X^{\mathcal{AB}})^\top$ and $\bar{\mathcal{L}}k(X^{\mathcal{AB}}, \boldsymbol{x}) = \bar{\mathcal{L}}k(\boldsymbol{x}, X^{\mathcal{AB}})^\top$.

Using this notation, Eq. (5.2) can equivalently be expressed as

$$\begin{aligned}
\hat{u}(\boldsymbol{x}) &= \bar{\mathcal{A}}k(\boldsymbol{x}, X^\mathcal{A})\boldsymbol{c}^\mathcal{A} + \bar{\mathcal{B}}k(\boldsymbol{x}, X^\mathcal{B})\boldsymbol{c}^\mathcal{B} \\
&= \bar{\mathcal{L}}k(\boldsymbol{x}, X^{\mathcal{AB}})\boldsymbol{c}
\end{aligned}$$

where $\boldsymbol{c}^\mathcal{A}, \boldsymbol{c}^\mathcal{B}$ are each column vectors with $[\boldsymbol{c}^\mathcal{A}]_i = c_i^\mathcal{A}$, $[\boldsymbol{c}^\mathcal{B}]_i = c_i^\mathcal{B}$, and $\boldsymbol{c}$ is a column vector formed by concatenating $\boldsymbol{c}^\mathcal{A}$ and $\boldsymbol{c}^\mathcal{B}$. The coefficients $\boldsymbol{c}$ are determined by demanding that $\hat{u}$ should satisfy the PDE exactly at the locations in $X^\mathcal{A}$ and $X^\mathcal{B}$. Let $\boldsymbol{g} = g(X^\mathcal{A})$ and $\boldsymbol{b} = b(X^\mathcal{B})$; then the interpolation equations that determine $\boldsymbol{c}$

are:

$$\mathcal{A}\bar{\mathcal{L}}k(X^{\mathcal{A}}, X^{\mathcal{AB}})\boldsymbol{c}^{\mathcal{A}} = \boldsymbol{g}$$
$$\mathcal{B}\bar{\mathcal{L}}k(X^{\mathcal{B}}, X^{\mathcal{AB}})\boldsymbol{c}^{\mathcal{B}} = \boldsymbol{b}$$

or

$$\mathcal{L}\bar{\mathcal{L}}k(X^{\mathcal{AB}}, X^{\mathcal{AB}})\boldsymbol{c} = \begin{bmatrix}\boldsymbol{g}\\ \boldsymbol{b}\end{bmatrix}.$$

Since this is a finite-dimensional linear system, it is then straightforward to determine that

$$\begin{aligned}\boldsymbol{c} &= \left[\mathcal{L}\bar{\mathcal{L}}k(X^{\mathcal{AB}}, X^{\mathcal{AB}})\right]^{-1}\begin{bmatrix}\boldsymbol{g}\\ \boldsymbol{b}\end{bmatrix}\\ \implies \hat{u}(\boldsymbol{x}) &= \bar{\mathcal{L}}k(\boldsymbol{x}, X^{\mathcal{AB}})\left[\mathcal{L}\bar{\mathcal{L}}k(X^{\mathcal{AB}}, X^{\mathcal{AB}})\right]^{-1}\begin{bmatrix}\boldsymbol{g}\\ \boldsymbol{b}\end{bmatrix}.\end{aligned}$$

A detailed theoretical analysis of the symmetric collocation method is beyond the scope of this work; it suffices to know the form of the estimator.

5.2.2 Probabilistic Meshless Method

The probabilistic meshless method will now be introduced. The proposed approach is similar in spirit to Cialenco et al. [2012] and also bears resemblance to the approach exposed in Chapter 4, thought with some additional technical detail to account for the fact that the spaces in question are now infinite-dimensional rather than finite-dimensional. To this end, endow $u(\boldsymbol{x})$ with the Gaussian prior $u(\boldsymbol{x}) \sim \mathcal{GP}(m(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}'))$. In this work we will generally assume that $m(\boldsymbol{x}) = 0$. Choice of $k(\boldsymbol{x}, \boldsymbol{x}')$ will be discussed in Section 5.2.3. The BPNM is then obtained by conditioning this prior on information obtained by projecting Eq. (5.1) against a set of search directions. Let $S_{\mathcal{A}} = \{s_i^{\mathcal{A}}\}_{i=1}^{m_{\mathcal{A}}} \subset H_{\mathcal{A}}(D)$ and $S_{\mathcal{B}} = \{s_i^{\mathcal{B}}\}_{i=1}^{m_{\mathcal{B}}} \subset H_{\mathcal{B}}(\partial D)$. Then information is obtained by projecting Eq. (5.1) against this set of search directions, i.e.

$$\begin{aligned}\left\langle s_i^{\mathcal{A}}, \mathcal{A}u^\dagger\right\rangle_{\mathcal{A}} &= \left\langle s_i^{\mathcal{A}}, g\right\rangle_{\mathcal{A}} && i = 1, \dots, m_{\mathcal{A}}\\ \left\langle s_i^{\mathcal{B}}, \mathcal{B}u^\dagger\right\rangle_{\mathcal{B}} &= \left\langle s_i^{\mathcal{B}}, b\right\rangle_{\mathcal{B}} && i = 1, \dots, m_{\mathcal{B}}\end{aligned}$$

Such an approach was explored in Owhadi [2015], with the search directions therein a hierarchy of basis functions defined on increasingly small subsets of the domain. However this requires computation of inner products which incurs additional discretisation error in general, and so we focus on a different class of search directions which are more analytically tractable. To be specific, we will take $s_i^{\mathcal{A}}$ and $s_i^{\mathcal{B}}$ be the representers of the appropriate evaluation operators for $H_{\mathcal{A}}(D)$ and $H_{\mathcal{B}}(D)$. Let $X^{\mathcal{A}} = \{\boldsymbol{x}_i^{\mathcal{A}}\}_{i=1}^{m_{\mathcal{A}}}$ and $X^{\mathcal{B}} = \{\boldsymbol{x}_i^{\mathcal{B}}\}_{i=1}^{m_{\mathcal{B}}}$, with $X^{\mathcal{A}} \subset D$ and $X^{\mathcal{B}} \subset \partial D$. Then in a slight abuse of notation we will take $s_i^{\mathcal{A}} = \delta_{\boldsymbol{x}_i^{\mathcal{A}}}$ for $i = 1, \dots, m_{\mathcal{A}}$, and $s_i^{\mathcal{B}} = \delta_{\boldsymbol{x}_i^{\mathcal{B}}}$ for $i = 1, \dots, m_{\mathcal{B}}$ (recalling that $\delta_{\boldsymbol{x}}$ is the evaluation operator). The collection $X^{\mathcal{AB}} = (X^{\mathcal{A}}, X^{\mathcal{B}})$ will be referred to as the *design points*. The solution which results from such information is an approximate solution to the strong form of the PDE.

The conditional distribution is again Gaussian; recalling the the notation introduced in Section 5.2.1, we have the following:

Proposition 5.2.1 (PMM)**.** *Under the Gaussian prior $u \sim \mathcal{GP}(m, k)$ the posterior distribution $u|\boldsymbol{g}, \boldsymbol{b} \sim \mathcal{GP}(m_1, k_1) := \mu_u^{\boldsymbol{g},\boldsymbol{b}}$, where*

$$m_1(\boldsymbol{x}) = m(\boldsymbol{x}) + \bar{\mathcal{L}}k(\boldsymbol{x}, X^{\mathcal{AB}})[\mathcal{L}\bar{\mathcal{L}}k(X^{\mathcal{AB}})]^{-1} \begin{bmatrix} \boldsymbol{g} - \mathcal{A}m(X^{\mathcal{A}}) \\ \boldsymbol{b} - \mathcal{B}m(X^{\mathcal{B}}) \end{bmatrix} \tag{5.3}$$

$$k_1(\boldsymbol{x}, \boldsymbol{x}') = k(\boldsymbol{x}, \boldsymbol{x}') - \bar{\mathcal{L}}k(\boldsymbol{x}, X^{\mathcal{AB}})[\mathcal{L}\bar{\mathcal{L}}k(X^{\mathcal{AB}})]^{-1}\mathcal{L}k(X^{\mathcal{AB}}, \boldsymbol{x}'). \tag{5.4}$$

Furthermore, $k_1(\boldsymbol{x}, \boldsymbol{x})$ will be abbreviated to $\sigma(\boldsymbol{x})^2$.

Proof. This is a straightforward consequence of the conditioning formula for Gaussian distributions; see Särkkä [2011, Section 3]. □

Note that $m_1(\boldsymbol{x})$ is identical to the estimate for the solution of the PDE obtained from symmetric collocation in Section 5.2.1 when $m(\boldsymbol{x}) = 0$. Here we consider the output of the algorithm to be the full Gaussian posterior distribution, so that the posterior variance $k(\boldsymbol{x}, \boldsymbol{x}')$ enables quantification of discretisation error.

An additional remark is that direct computation of the posterior incurs a cost of $\mathcal{O}((m_{\mathcal{A}} + m_{\mathcal{B}})^3)$, owing to the requirement to invert the matrix $\mathcal{L}\bar{\mathcal{L}}k(X^{\mathcal{AB}})$. Unlike in Chapter 4, this chapter will not focus on reduction of this cost. If the information operator were not restricted to consist of evaluation functionals, similar conjugacy arguments to those described in Section 4.3.1 could be followed to reduce this cost. However, this would introduce a requirement to compute $\langle \cdot, \cdot \rangle_{\mathcal{A}}$ and $\langle \cdot, \cdot \rangle_{\mathcal{B}}$, and since in the present setting those inner products do not have an explicit closed form this approach was not pursued. Note that many approaches for rapid computation of similar matrices arising from Gaussian process regression have been presented in the literature and could be applied here; see Snelson and Ghahramani [2006]; Schäfer et al. [2017] and the references therein.

We now turn to a discussion of the choice of prior.

5.2.3 Prior Choice

Compared to Chapter 4, more structure is demanded in the present setting owing to the fact that the derivatives required by $\mathcal{A}$ and $\mathcal{B}$ must each exist for all functions in the support of the prior. Here several specific choices are considered. In Section 5.2.3 a prior measure is introduced, with covariance inspired by the natural prior $\Sigma_0 = (A^\top A)^{-1}$ from Section 4.4. In Section 5.2.3 a second prior covariance is proposed which does not directly relate to one of those considered in Chapter 4, but which proves useful for the development of theory. Note that while arbitrary prior covariance functions can be used, the selection is subject to the restriction that the spaces $\mathcal{H}_{\mathcal{A}}(D)$ and $\mathcal{H}_{\mathcal{B}}(\partial D)$ are RKHS. In practise, when $\mathcal{A}$ and $\mathcal{B}$ are differential operators as in the present section, this amounts to a requirement that k be suitably differentiable.

A Natural Prior Measure

The natural prior is derived by assuming that $g(\boldsymbol{x})$, rather than $u(\boldsymbol{x})$, is endowed with a Gaussian prior, and deriving the implied distribution on $u(\boldsymbol{x})$. For simplicity we will make the restrictive assumption in this section that the PDE has homogeneous boundary conditions, i.e. $\mathcal{B} = \mathcal{I}$ and that $b(\boldsymbol{x}) = 0$. This assumption could be generalised, but computing the covariance function required for the prior measure in this section is so challenging that such generality is unlikely to be useful. Furthermore assume that for all $g \in \mathcal{H}_{\mathcal{A}}(D)$, the PDE with forcing g has a unique solution.

The construction in this section makes use of the deep connections between kernels and Green's functions, described in Fasshauer and Ye [2011]. To this end, suppose that Eq. (2.1) admits a Green's function $G(\boldsymbol{x}, \boldsymbol{x}')$ satisfying

$$\begin{aligned} \mathcal{A}G(\boldsymbol{x}, \boldsymbol{x}') &= \delta(\boldsymbol{x} - \boldsymbol{x}') && \boldsymbol{x} \in D \\ \mathcal{B}G(\boldsymbol{x}, \boldsymbol{x}') &= 0 && \boldsymbol{x}' \in \partial D. \end{aligned}$$

Recall that the Green's function defines an integral operator that is essentially the inverse of $\mathcal{A}$ in that, for each $\boldsymbol{x} \in D$, we have:

$$\int_D G(\boldsymbol{x}, \boldsymbol{x}')g(\boldsymbol{x})\mathrm{d}\boldsymbol{x}' = u(\boldsymbol{x}).$$

Now following the same argument as in Section 4.4, since g is the object about which information is obtained, placing a prior on g is natural in some sense. Thus, suppose that $g \sim \mathcal{GP}(0, \Lambda)$ for some positive-definite covariance function $\Lambda(\boldsymbol{x}, \boldsymbol{x}')$, $\boldsymbol{x}, \boldsymbol{x}' \in D$. Placing such a distribution over g implies a requirement that $H_\Lambda(D) \subseteq H_{\mathcal{A}}(D)$.

Now we derive the prior on u implied by this distribution on g. Define the inner product space $(H_{\mathrm{nat}}(D), \langle\,\cdot\,,\,\cdot\,\rangle_{\mathrm{nat}})$ by

$$\begin{aligned} H_{\mathrm{nat}}(D) &:= \{v \in H(D) \mid \mathcal{A}v \in H_\Lambda(D),\ v = 0 \text{ on } \partial D\}, \\ \langle u, v\rangle_{\mathrm{nat}} &:= \langle \mathcal{A}u, \mathcal{A}v\rangle_\Lambda. \end{aligned}$$

Note that by definition functions in $H_{\mathrm{nat}}(D)$ encode the boundary conditions of the PDE, and so when computing the posterior from Proposition 5.2.1 no collocation points need be allocated on the boundary, i.e. $X^{\mathcal{B}}$ may be taken to be the empty set. Under this definition $\|u\|_{\mathrm{nat}}^2 := \langle u, u\rangle_{\mathrm{nat}} = \|g\|_\Lambda^2$. We now establish that $H_{\mathrm{nat}}(D)$ is an RKHS.

Proposition 5.2.2. *Introduce the* natural kernel*, defined as:*

$$k_{\mathrm{nat}}(\boldsymbol{x}, \boldsymbol{x}') := \int_D \int_D G(\boldsymbol{x}, \boldsymbol{z})G(\boldsymbol{x}', \boldsymbol{z}')\Lambda(\boldsymbol{z}, \boldsymbol{z}')\mathrm{d}\boldsymbol{z}\mathrm{d}\boldsymbol{z}'. \tag{5.5}$$

Assume that $k_{\mathrm{nat}}(\boldsymbol{x}, \boldsymbol{x})$ is bounded. Then $H_{\mathrm{nat}}(D)$ is an RKHS with reproducing kernel k_{nat}.

Proof. First we verify the reproducing property. For each $u \in H_{\mathrm{nat}}(D)$, we have

$$\begin{aligned} \langle u, k_{\mathrm{nat}}(\,\cdot\,, \boldsymbol{x})\rangle_{\mathrm{nat}} &= \langle \mathcal{A}u, \mathcal{A}k(\,\cdot\,, \boldsymbol{x})\rangle_\Lambda \\ &= \left\langle \mathcal{A}u, \mathcal{A}\iint_D G(\,\cdot\,, \boldsymbol{z})G(\boldsymbol{x}, \boldsymbol{z}')\Lambda(\boldsymbol{z}, \boldsymbol{z}')\mathrm{d}\boldsymbol{z}\mathrm{d}\boldsymbol{z}'\right\rangle_\Lambda . \end{aligned}$$

Bringing the operator $\mathcal{A}$ inside the integral and using properties of the Green's function, we have:

$$\begin{aligned}\langle u, k_{\mathrm{nat}}(\,\cdot\,,\boldsymbol{x})\rangle_{\mathrm{nat}} &= \left\langle \mathcal{A}u, \iint_D \delta(\,\cdot\, - \boldsymbol{z})G(\boldsymbol{x},\boldsymbol{z'})\Lambda(\boldsymbol{z},\boldsymbol{z}')\mathrm{d}\boldsymbol{z}\mathrm{d}\boldsymbol{z}'\right\rangle_\Lambda \\ &= \left\langle \mathcal{A}u, \int_D G(\boldsymbol{x},\boldsymbol{z'})\Lambda(\,\cdot\,,\boldsymbol{z}')\mathrm{d}\boldsymbol{z}'\right\rangle_\Lambda \\ &= \int_D G(\boldsymbol{x},\boldsymbol{z'})\left\langle \mathcal{A}u, \Lambda(\,\cdot\,,\boldsymbol{z}')\right\rangle_\Lambda \mathrm{d}\boldsymbol{z}'\end{aligned}$$

where we have exploited the linearity of inner products to bring the inner product inside the integral. Thus, applying the reproducing property in Λ we find

$$\begin{aligned}\langle u, k_{\mathrm{nat}}(\,\cdot\,,\boldsymbol{x})\rangle_{\mathrm{nat}} &= \int_D G(\boldsymbol{x},\boldsymbol{z'})g(\boldsymbol{z}')\mathrm{d}\boldsymbol{z}' \\ &= u(\boldsymbol{x})\end{aligned}$$

as required.

It remains to establish that $H_{\mathrm{nat}}(D)$ is an RKHS. From Theorem 2.3.4, this is equivalent to continuity of the evaluation functional on $H_{\mathrm{nat}}(D)$. Recall that a linear operator between normed spaces is continuous if and only if it is a bounded, so we proceed to verify that the evaluation functional is a bounded linear operator. Linearity is clear. To establish boundedness, note that by the Cauchy–Schwarz inequality, for each $v \in H_{\mathrm{nat}}(D)$ we have:

$$\begin{aligned}|v(\boldsymbol{x})| &= |\langle v, k_{\mathrm{nat}}(\,\cdot\,,\boldsymbol{x})\rangle_{\mathrm{nat}}| \\ &\leq \langle v, v\rangle_{\mathrm{nat}}^{1/2}\langle k_{\mathrm{nat}}(\,\cdot\,,\boldsymbol{x}), k_{\mathrm{nat}}(\,\cdot\,,\boldsymbol{x})\rangle_{\mathrm{nat}}^{1/2} \\ &= \|v\|_{\mathrm{nat}}\, k_{\mathrm{nat}}(\boldsymbol{x},\boldsymbol{x})^{1/2}\end{aligned}$$

which proves that the evaluation functional is a bounded linear operator. Thus, $H_{\mathrm{nat}}(D)$ is an RKHS with reproducing kernel k_{nat}. □

This result implies that it that placing a prior measure on g is *equivalent* to placing a prior measure on u, as shown by the following proposition:

Proposition 5.2.3. *It holds that $g \sim \mathcal{GP}(0,\Lambda)$ if and only if $u \sim \mathcal{GP}(0,k_{\mathrm{nat}})$.*

Proof. Note that since both $\mathcal{A}$ and the operator $v(\boldsymbol{x}) \mapsto \int G(\boldsymbol{x},\boldsymbol{z})v(\boldsymbol{z})\mathrm{d}\boldsymbol{z}$ are linear, the pushforward of a Gaussian measure through either operator is again Gaussian. It thus suffices to show that the first and second moments are equal.

First, suppose $g \sim \mathcal{GP}(0,\Lambda)$. From the properties of Green's functions we have that

$$\begin{aligned} u(\boldsymbol{x}) &= \int_D G(\boldsymbol{x},\boldsymbol{z})g(\boldsymbol{z})\mathrm{d}\boldsymbol{z} \\ \implies \mathbb{E}[u(\boldsymbol{z})] &= \mathbb{E}\left[\int_D G(\boldsymbol{x},\boldsymbol{z})g(\boldsymbol{z})\mathrm{d}\boldsymbol{z}\right] \\ &= \int_D G(\boldsymbol{x},\boldsymbol{z})\mathbb{E}[g(\boldsymbol{z})]\mathrm{d}\boldsymbol{z} = 0 \end{aligned}$$

and furthermore

$$\begin{aligned} \mathbb{E}[u(\boldsymbol{x})u(\boldsymbol{x}')] &= \mathbb{E}\left[\int_D\int_D G(\boldsymbol{x},\boldsymbol{z})g(\boldsymbol{z})G(\boldsymbol{x}',\boldsymbol{z}')g(\boldsymbol{z}')\mathrm{d}\boldsymbol{z}\mathrm{d}\boldsymbol{z}'\right] \\ &= \int_D\int_D G(\boldsymbol{x},\boldsymbol{z})G(\boldsymbol{x}',\boldsymbol{z}')\mathbb{E}[g(\boldsymbol{z})g(\boldsymbol{z}')]\mathrm{d}\boldsymbol{z}\mathrm{d}\boldsymbol{z}' \\ &= \int_D\int_D G(\boldsymbol{x},\boldsymbol{z})G(\boldsymbol{x}',\boldsymbol{z}')\Lambda(\boldsymbol{z},\boldsymbol{z}')\mathrm{d}\boldsymbol{z}\mathrm{d}\boldsymbol{z}' = k_{\mathrm{nat}}(\boldsymbol{x},\boldsymbol{x}'). \end{aligned}$$

and thus $u \sim \mathcal{GP}(0, k_{\mathrm{nat}})$.

For the converse suppose that $u \sim \mathcal{GP}(0, k_{\mathrm{nat}})$. Then we have that

$$\begin{aligned} \mathbb{E}[g(\boldsymbol{x})] &= \mathbb{E}[\mathcal{A}u(\boldsymbol{x})] \\ &= \mathcal{A}\mathbb{E}[u(\boldsymbol{x})] = 0 \end{aligned}$$

and furthermore, letting $\tilde{\mathcal{A}}$ denote $\mathcal{A}$ when acting upon $\boldsymbol{x}'$:

$$\begin{aligned} \mathbb{E}[g(\boldsymbol{x})g(\boldsymbol{x}')] &= \mathcal{A}\tilde{\mathcal{A}}\mathbb{E}[u(\boldsymbol{x})u(\boldsymbol{x}')] \\ &= \mathcal{A}\tilde{\mathcal{A}}k_{\mathrm{nat}}(\boldsymbol{x},\boldsymbol{x}') \\ &= \mathcal{A}\tilde{\mathcal{A}}\int_D\int_D G(\boldsymbol{x},\boldsymbol{z})G(\boldsymbol{x}',\boldsymbol{z}')\Lambda(\boldsymbol{z},\boldsymbol{z}')\mathrm{d}\boldsymbol{z}\mathrm{d}\boldsymbol{z}' \\ &= \int_D\int_D \delta(\boldsymbol{x}-\boldsymbol{z})\delta(\boldsymbol{x}'-\boldsymbol{z}')\Lambda(\boldsymbol{z},\boldsymbol{z}')\mathrm{d}\boldsymbol{z}\mathrm{d}\boldsymbol{z}' \\ &= \Lambda(\boldsymbol{x},\boldsymbol{x}') \end{aligned}$$

where for the third line the properties of Green's functions were again used. □

Note that Green's functions can seldom be computed in practise. Thus, this choice of prior covariance is interesting but impractical, as was the natural prior in Section 4.4. However, unlike in that section it is not the case that the natural prior implies convergence in a single iteration. This is because the search directions for the PMM have not been constructed using $g(\boldsymbol{x})$, but have instead been restricted to be evaluation functionals. A more practical choice of prior measure will now be presented.

A Practical Prior Measure

To elicit a practical method, in contrast to the previous section it will now be *assumed* that $H(D)$ is an RKHS with some reproducing kernel $\tilde{k}$, i.e. $H(D) = H_{\tilde{k}}(D)$. As commented in Section 2.3.2, Gaussian measures assign zero mass to their RKHS, so it is necessary to construct a Gaussian measure using a covariance function *derived from* $\tilde{k}$, rather than $\tilde{k}$ itself. Thus, for inference, the prior covariance used will be a kernel $\hat{k}$ which corresponds to a Gaussian measure with $H_{\tilde{k}}(D)$ in its support, to ensure that $u^\dagger$ does not lie in a null set of the prior. We will assume that $H_{\hat{k}}(D)$ is embedded in $H_{\tilde{k}}(D)$ (see Definition A.1.6). Then natural requirements are both that $H_{\hat{k}}(D)$ be dense in $H_{\tilde{k}}(D)$, and that the support of the prior is $H_{\tilde{k}}(D)$. A result from [Cialenco et al., 2012, Lemma 2.2], gives a construction for $\hat{k}$ derived from $\tilde{k}$ which satisfies these two requirements.

Proposition 5.2.4. *For the covariance function*

$$\hat{k}(\boldsymbol{x}, \boldsymbol{x}') := \int_D \tilde{k}(\boldsymbol{x}, \boldsymbol{z})\tilde{k}(\boldsymbol{z}, \boldsymbol{x}')\mathrm{d}\boldsymbol{z}$$

it holds that $H_{\hat{k}}(D)$ is dense in $H_{\tilde{k}}(D)$, and a (centered) Gaussian distribution with covariance $\hat{k}$ is supported on $H_{\tilde{k}}(D)$.

Proof. Recall the following. Since $\overline{D}$ is compact (by assumption) and $\tilde{k}$ is symmetric and positive definite, Theorem 2.3.10 states that there exists an eigendecomposition of $\tilde{k}$ which is countable, with eigenvalues $\{\lambda_i\}$ and eigenfunctions $\{e_i\}$, $i \in \mathbb{N}$, where we assume that the eigenvalues are ordered so that $\lambda_1 \geq \lambda_2 \geq \cdots > 0$. Furthermore $\{e_i\}$ is an orthonormal basis of $L^2(D)$ and $\tilde{k}$ can be represented in terms of its eigendecomposition as

$$\tilde{k}(\boldsymbol{x}, \boldsymbol{x}') = \sum_i \lambda_i e_i(\boldsymbol{x}) e_i(\boldsymbol{x}').$$

Lastly, for any $v \in H_{\tilde{k}}(D)$ it holds that

$$v = \sum_{i \in \mathbb{N}} c_i \sqrt{\lambda_i} e_i \tag{5.6}$$

for a unique sequence $(c_i) \in \ell^2$, and the $\tilde{k}$-norm of v is given by

$$\|v\|_{\tilde{k}}^2 = \sum_i c_i^2.$$

We begin by showing that $\hat{k}$ is also a Mercer kernel with eigenvalues $\{\lambda_i^2\}$ and eigenfunctions $\{e_i\}$, $i \in \mathbb{N}$. Since $\tilde{k}$ is positive-definite and bounded on $\bar{D}$ and $\bar{D}$ is compact, so too must $\hat{k}$ be, and furthermore we have that:

$$\begin{aligned}
\hat{k}(\boldsymbol{x}, \boldsymbol{x}') &= \int_D \sum_{i\in\mathbb{N}} \sum_{j\in\mathbb{N}} \lambda_i \lambda_j e_i(\boldsymbol{x}) e_i(\boldsymbol{z}) e_j(\boldsymbol{x}') e_j(\boldsymbol{z}) \mathrm{d}z \\
&= \sum_{i\in\mathbb{N}} \sum_{j\in\mathbb{N}} \lambda_i \lambda_j e_i(\boldsymbol{x}) e_j(\boldsymbol{x}') \underbrace{\int_D e_i(\boldsymbol{z}) e_j(\boldsymbol{z}) \mathrm{d}z}_{=\delta_{ij}} \\
&= \sum_{i\in\mathbb{N}} \lambda_i^2 e_i(\boldsymbol{x}) e_i(\boldsymbol{x}')
\end{aligned}$$

We now show that $H_{\hat{k}}(D)$ is dense in $H_{\tilde{k}}(D)$. For any $v \in H_{\tilde{k}}(D)$, we have that $v = \sum_{i\in\mathbb{N}} c_i \sqrt{\lambda_i} e_i$ and $(c_i) \in \ell^2$. Consider the partial sums $v^N = \sum_{i=1}^N c_i \sqrt{\lambda_i} e_i$, which converge to v in $\| \cdot \|_{\tilde{k}}$, since v has finite $\tilde{k}$-norm. Note that $v^N \in H_{\hat{k}}(D)$, since

$$v_N = \sum_{i=1}^N c_i \sqrt{\lambda_i} e_i = \sum_{i=1}^N \frac{c_i}{\sqrt{\lambda_i}} \lambda_i e_i.$$

It thus follows that for any $v \in H_{\tilde{k}}(D)$ and each $\epsilon > 0$ there is a $v' \in H_{\hat{k}}(D)$ with $\|v - v'\|_{\tilde{k}} < \epsilon$, and so $H_{\hat{k}}(D)$ is dense in $H_{\tilde{k}}(D)$.

To see that $\mu(H_{\tilde{k}}(D)) = 1$, let V be a random variable with law μ and note that from Theorem 2.3.11 V can be represented as

$$V = \sum_{i\in\mathbb{N}} \xi_i \lambda_i e_i$$

where $\xi_i \sim N(0,1)$ IID. Examining Eq. (5.6) we must have $c_i = \xi_i \sqrt{\lambda_i}$, and so the $\tilde{k}$-norm of $\hat{v}$ is given by:

$$\mathbb{E}(\|V\|_{\tilde{k}}^2) = \mathbb{E}\left(\sum_{i\in\mathbb{N}} \xi_i^2 \lambda_i \right) = \sum_{i\in\mathbb{N}} \lambda_i < \infty$$

since the sum of eigenvalues converges. Thus $\hat{v}$ lies in $H_{\tilde{k}}(D)$ almost-surely. □

5.2.4 Theoretical Results for the Forward Problem

In this section theoretical results for Proposition 5.2.1 will be presented, under the prior covariance function $\hat{k}$. Let ρ denote the maximum differential order of $\mathcal{A}$ and $\mathcal{B}$. Throughout this section we will assume that the space $H_{\hat{k}}(D)$ is norm-equivalent to the Sobolev space $\mathcal{H}^\beta(D)$, for $\beta > d/2$ (see Definition A.1.7, and furthermore that $\rho < \beta - d/2$. This ensures that $\mathcal{H}^\beta(D)$ embeds in a space of appropriately differentiable functions, thanks to Sobolev embedding theorem. This was originally problem in Sobolev [1938], and is presented in detail in in Evans [2010, Section 5.6]. See also Cialenco et al. [2012, Theorem 3.1].

The first proposition relates the pointwise error in the posterior mean to the posterior variance from Proposition 5.2.1:

Proposition 5.2.5 (Local accuracy)**.** *For all $\boldsymbol{x} \in D$ we have that*

$$|m_1(\boldsymbol{x}) - u^\dagger(\boldsymbol{x})| \leq \sigma(\boldsymbol{x}) \|u^\dagger\|_{\hat{k}}$$

Proof. We begin by extending the inner product $\langle \cdot, \cdot \rangle_{\hat{k}}$ to vector-valued functions in $H_{\hat{k}}(d)$. Introduce the function $\eta : (H_{\hat{k}}(d))^n \times (H_{\hat{k}}(d))^m \to \mathbb{R}^{m\times n}$, given by

$$[\eta(f, g)]_{ij} = \langle f_i, g_j \rangle_{\hat{k}}.$$

Clearly $\eta(f, g) = \eta(g, f)^\top$. Furthermore from linearity of the inner product we have that for any matrix $A \in \mathbb{R}^{k\times n}$, $\eta(Af, g) = A\eta(f, g)$. In a slight abuse of notation, for $g' \in H_{\hat{k}}(d)$ we will assume $\eta(f, g') \in \mathbb{R}^n$ is given by

$$[\eta(f, g')]_i = \langle f_i, g' \rangle_{\hat{k}}.$$

Now, recall that

$$m_1(\boldsymbol{x}) = \hat{k}(\boldsymbol{x}, X^{\mathcal{AB}})\mathcal{L}\bar{\mathcal{L}}\hat{k}(X^{\mathcal{AB}})\boldsymbol{v}$$

where

$$\begin{aligned} \boldsymbol{v} &= \begin{bmatrix} \boldsymbol{g} \\ \boldsymbol{b} \end{bmatrix} \\ &= \begin{bmatrix} \mathcal{A}\eta(\hat{k}(X^{\mathcal{A}}, \cdot), u^\dagger) \\ \mathcal{B}\eta(\hat{k}(X^{\mathcal{B}}, \cdot), u^\dagger) \end{bmatrix} \\ &= \begin{bmatrix} \eta(\mathcal{A}\hat{k}(X^{\mathcal{A}}, \cdot), u^\dagger) \\ \eta(\mathcal{B}\hat{k}(X^{\mathcal{B}}, \cdot), u^\dagger) \end{bmatrix} \\ &= \eta(\mathcal{L}\hat{k}(X^{\mathcal{AB}}, \cdot), u^\dagger) \end{aligned}$$

where on the second line we have used the reproducing property. Letting $K = \hat{k}(\boldsymbol{x}, X^{\mathcal{AB}})(\mathcal{L}\bar{\mathcal{L}}\hat{k}(X^{\mathcal{AB}}))^{-1}$, we then have that for fixed $\boldsymbol{x}$, since $K\mathcal{L}\hat{k}(X^{\mathcal{AB}}, \cdot) \in$

$H^{\hat{k}}(D)$:

$$\begin{aligned} m_1(\boldsymbol{x}) &= \eta(K\mathcal{L}\hat{k}(X^{\mathcal{AB}}, \cdot), u^\dagger) \\ &= \langle K\mathcal{L}\hat{k}(X^{\mathcal{AB}}, \cdot), u^\dagger \rangle_{\hat{k}} \\ \implies |u^\dagger(\boldsymbol{x}) - m_1(\boldsymbol{x})| &= \left| \left\langle \hat{k}(\boldsymbol{x}, \cdot) - K\mathcal{L}\hat{k}(X^{\mathcal{AB}}, \cdot), u^\dagger \right\rangle_{\hat{k}} \right| \\ &\leq \|u^\dagger\|_{\hat{k}} \underbrace{\left\| \hat{k}(\boldsymbol{x}, \cdot) - K\mathcal{L}\hat{k}(X^{\mathcal{AB}}, \cdot) \right\|_{\hat{k}}}_{(*)} \end{aligned}$$

by the Cauchy–Schwartz inequality. Now, we have

$$\begin{aligned} (*)^2 &= \left\langle \hat{k}(\boldsymbol{x}, \cdot) - K\mathcal{L}\hat{k}(X^{\mathcal{AB}}, \cdot), \hat{k}(\boldsymbol{x}, \cdot) - K\mathcal{L}\hat{k}(X^{\mathcal{AB}}, \cdot) \right\rangle_{\hat{k}} \\ &= \hat{k}(\boldsymbol{x}, \boldsymbol{x}) - 2\eta\left(\hat{k}(\boldsymbol{x}, \cdot), K\mathcal{L}\hat{k}(X^{\mathcal{AB}}, \cdot)\right) + \eta\left(K\mathcal{L}\hat{k}(X^{\mathcal{AB}}, \cdot), K\mathcal{L}\hat{k}(X^{\mathcal{AB}}, \cdot)\right). \end{aligned}$$

Applying the reproducing property and recalling the definitions of η, $\bar{\mathcal{L}}\hat{k}$ and $\mathcal{L}\bar{\mathcal{L}}\hat{k}$, it is then clear that

$$\begin{aligned} (*)^2 &= \hat{k}(\boldsymbol{x}, \boldsymbol{x}) - 2\bar{\mathcal{L}}\hat{k}(\boldsymbol{x}, X^{\mathcal{AB}})K^\top + K\mathcal{L}\bar{\mathcal{L}}\hat{k}(X^{\mathcal{AB}})K^\top \\ &= \hat{k}(\boldsymbol{x}, \boldsymbol{x}) - \bar{\mathcal{L}}\hat{k}(\boldsymbol{x}, X^{\mathcal{AB}})(\mathcal{L}\bar{\mathcal{L}}\hat{k}(X^{\mathcal{AB}}))^{-1}\mathcal{L}\hat{k}(X^{\mathcal{AB}}, \boldsymbol{x}) \\ &= \sigma(\boldsymbol{x})^2. \end{aligned}$$

and thus

$$|u^\dagger(\boldsymbol{x}) - m_1(\boldsymbol{x})| \leq \sigma(\boldsymbol{x})\|u^\dagger\|_{\hat{k}}$$

as required. □

Thus, the error in Proposition 5.2.5 is locally controlled by the posterior variance. This bound can be directly linked to the set of design points $X^{\mathcal{AB}}$. To accomplish this, introduce *fill distance* for the (finite) set $X \subset D$, defined as:

$$h(X) := \sup_{\boldsymbol{x} \in D} \min_{\boldsymbol{x}' \in X} \|\boldsymbol{x} - \boldsymbol{x}'\|_2.$$

Then the following proposition from Cialenco et al. [2012], quoted here without proof, bounds $\sigma(\boldsymbol{x})$ in terms of the fill distance of the design points $h(X^{\mathcal{AB}})$. Dependence of $h(X^{\mathcal{AB}})$ on the design points will generally be suppressed except where it is relevant.

Proposition 5.2.6 (Lemma 3.4 of Cialenco et al. [2012])**.** *For all* $\boldsymbol{x} \in D$ *and* *whenever $h > 0$ is sufficiently small, it holds that*

$$\sigma(\boldsymbol{x}) \leq C^F h^{\beta-\rho-d/2}$$

where C^F is a constant independent of $\boldsymbol{x}$ and $X^{\mathcal{AB}}$.

The last result in this section uses Proposition 5.2.6 to establish consistency results for the solver as the fill distance decreases:

Theorem 5.2.7. *Let $Z(\epsilon) = \{u \in \mathcal{X} : \|u - u^\dagger\|_2^2 > \epsilon\}$. For all h sufficiently small, it holds that the posterior measure $\mu_u^{\boldsymbol{g},\boldsymbol{b}}$ from Proposition 5.2.1 satisfies:*

$$\mu_u^{\boldsymbol{g},\boldsymbol{b}}(Z(\epsilon)) \leq \frac{C}{\epsilon} h^{2\beta-2\rho-d}$$

where C is a constant independent of h and $h = h(X^{\mathcal{AB}})$.

Proof. By applying the triangle inequality, we find:

$$\begin{aligned}\int_{\mathcal{X}} \|u - u^\dagger\|_2^2 \, \mu_u^{\boldsymbol{g},\boldsymbol{b}}(\mathrm{d}u) \leq & \underbrace{\int_{\mathcal{X}} \|u - m_1\|_2^2 \, \mu_u^{\boldsymbol{g},\boldsymbol{b}}(\mathrm{d}u)}_{(1)} + \underbrace{\int_{\mathcal{X}} \|m_1 - u^\dagger\|_2^2 \, \mu_u^{\boldsymbol{g},\boldsymbol{b}}(\mathrm{d}u)}_{(2)} \\ & + \underbrace{\int_{\mathcal{X}} \|u - m_1\|_2 \|m_1 - u^\dagger\|_2 \, \mu_u^{\boldsymbol{g},\boldsymbol{b}}(\mathrm{d}u)}_{(3)} .\end{aligned}$$

Beginning with (1):

$$\begin{aligned}\int_{\mathcal{X}} \|u - m_1\|_2^2 \, \mu_u^{\boldsymbol{g},\boldsymbol{b}}(\mathrm{d}u) &= \int_D \int_{\mathcal{X}} |u(\boldsymbol{x}) - m_1(\boldsymbol{x})|^2 \, \mu_u^{\boldsymbol{g},\boldsymbol{b}}(\mathrm{d}u) \, \mathrm{d}\boldsymbol{x} \\ &= \int_D \sigma^2(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} \qquad (5.7)\end{aligned}$$

where in the first time we used Fubini's theorem to change the order of integration and in the second line we used the definition of $\sigma(\boldsymbol{x})$. For (2):

$$\begin{aligned}\int_{\mathcal{X}} \|m_1 - u^\dagger\|_2^2 \, \mu_u^{\boldsymbol{g},\boldsymbol{b}}(\mathrm{d}u) &= \|m_1 - u^\dagger\|_2^2 \\ &= \int_D |m_1(\boldsymbol{x}) - u^\dagger(\boldsymbol{x})|^2 \, \mathrm{d}\boldsymbol{x} \\ &\leq \|u^\dagger\|_{\hat{k}}^2 \int_D \sigma^2(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} \qquad (5.8)\end{aligned}$$

by applying Proposition 5.2.5. For (3):

$$\begin{aligned}\int_{\mathcal{X}} \|u - m_1\|_2 \|m_1 - u^\dagger\| \, \mu_u^{\boldsymbol{g},\boldsymbol{b}}(\mathrm{d}u) &= \left[\|m_1 - u^\dagger\|_2^2 \left(\int_{\mathcal{X}} \|u - m_1\|_2 \, \mu_u^{\boldsymbol{g},\boldsymbol{b}}(\mathrm{d}u) \right)^2 \right]^{\frac{1}{2}} \\ &\leq \left[\|m_1 - u^\dagger\|_2^2 \int_{\mathcal{X}} \|u - m_1\|_2^2 \, \mu_u^{\boldsymbol{g},\boldsymbol{b}}(\mathrm{d}u) \right]^{\frac{1}{2}} \\ &\leq \|u^\dagger\|_{\hat{k}} \int_{\mathcal{D}} \sigma^2(\boldsymbol{x}) \, \mathrm{d}x\end{aligned}$$

where the second line is from application of Jensen's inequality and the third line is from application of the bounds in Eqs. (5.7) and (5.8). Combining these and applying Proposition 5.2.6, we then find, when h is sufficiently small:

$$\int_{\mathcal{X}} \|u^\dagger - u\|_2^2 \, \mu_u^{\boldsymbol{g},\boldsymbol{b}}(\mathrm{d}u) \leq [C^F]^2 (1 + \|u^\dagger\|_{\hat{k}} + \|u^\dagger\|_{\hat{k}}^2) h^{2\beta - 2\rho - d}.$$

Now recall that Markov's inequality states that if X is a nonnegative random variable and $\epsilon > 0$, then

$$\mathbb{P}(X \geq \epsilon) \leq \frac{\mathbb{E}(X)}{\epsilon}.$$

Applying this with $X = \|u^\dagger - U\|_2^2$ where U is a random variable with law $\mu_u^{\boldsymbol{g},\boldsymbol{b}}$, we find that:

$$\begin{aligned}\mu_u^{\boldsymbol{g},\boldsymbol{b}}(Z(\epsilon)) &= \mathbb{P}(X \geq \epsilon) \\ &\leq \frac{1}{\epsilon} \int_{\mathcal{X}} \|u^\dagger - u\|_2^2 \, \mu_u^{\boldsymbol{g},\boldsymbol{b}}(\mathrm{d}u) \\ &\leq \frac{1}{\epsilon} [C^F]^2 (1 + \|u^\dagger\|_{\hat{k}} + \|u^\dagger\|_{\hat{k}}^2) h^{2\beta - 2\rho - d}\end{aligned}$$

as required. □

This concludes the analysis of the forward solver. In the next section we will analyse the use of the PMM as a forward solver in Bayesian inverse problems.

5.3 PMM and Bayesian Inverse Problems

The focus of this section is on establishing the properties of the posterior distribution in Bayesian inverse problems, when PMM are used as the forward solver. Let Θ be a separable Banach space and let $\theta^\dagger \in \Theta$ be a parameter to be inferred. For simplicity, Eq. (2.1) is assumed to depend on θ through the operators $\mathcal{A}$ and $\mathcal{B}$, implying a PDE

$$\begin{aligned}\mathcal{A}[\theta]u^\dagger(\boldsymbol{x};\theta) &= g(\boldsymbol{x}) \quad \boldsymbol{x}\in D\\ \mathcal{B}[\theta]u^\dagger(\boldsymbol{x};\theta) &= b(\boldsymbol{x}) \quad \boldsymbol{x}\in \partial D.\end{aligned} \tag{5.9}$$

Note that this could be straightforwardly generalised to allow $g(\boldsymbol{x})$ and $b(\boldsymbol{x})$ to depend on θ. The operators $\mathcal{A}$ and $\mathcal{B}$ will be assumed to be linear and elliptic for each $\theta\in\Theta$, and note that the dependence of $u^\dagger$ on θ has now been emphasised; for each $\theta\in\Theta$, since the operators vary with θ the true solution to the PDE differs. Recall from Section 2.4 that we assume $\boldsymbol{y}\in\mathcal{Y}$ is observed data, with $\mathcal{Y}$ assumed to be finite-dimensional and linked to u via a parameter-to-observation map $\mathcal{G} : H(D)\to\mathbb{R}^d$ as follows:

$$\boldsymbol{y} = \mathcal{G}(u^\dagger(\,\cdot\,;\theta^\dagger)) + \xi.$$

In this section it will be assumed that $\mathcal{G}$ is a bounded linear operator obtained by evaluating $u^\dagger(\,\cdot\,,\theta)$ at some (unspecified) set of points in D, and also that $\xi\sim\mathcal{N}(0,\Gamma)$, though this latter requirement could be relaxed. Note that the inverse problem may nevertheless be nonlinear, as no assumptions have been made on the linearity of u as a function of θ. We thus have a potential given by

$$\Phi(\boldsymbol{y};\theta,u) = \frac{1}{2}\|\mathcal{G}(u(\,\cdot\,,\theta)) - \boldsymbol{y}\|_{\Gamma^{-1}}$$

where it has been emphasised that Φ depends on both θ and u. Note that while each $\theta\in\Theta$ defines an exact solution $u^\dagger(\,\cdot\,,\theta)$, this notation is used to allow for the fact that this solution must generally be replaced with the output from a numerical method. For a prior μ_θ on θ, the posterior over θ is then defined by:

$$\begin{aligned}\frac{\mathrm{d}\mu_\theta^{\boldsymbol{y}}}{\mathrm{d}\mu_\theta}(\theta) &= \frac{1}{Z}\exp(-\Phi(\boldsymbol{y};\theta,u^\dagger))\\ Z &= \int_\Theta \exp(-\Phi(\boldsymbol{y};\theta,u^\dagger))\mu_\theta(\mathrm{d}\theta).\end{aligned}$$

The approach proposed in this section is to use the output from the PMM as a forward solver in a Bayesian inverse problem by marginalising the likelihood over $\mu_u^{\boldsymbol{g},\boldsymbol{b}}$, much as in Section 4.6.2. Note that the output from Proposition 5.2.1 then depends upon θ due to the dependence of $\mathcal{A},\mathcal{B}$ on θ. Following the same arguments as in that section, this gives the PMM potential:

$$\Phi_h(\boldsymbol{y};\theta) := \frac{1}{2}\|\boldsymbol{m}(\theta) - \boldsymbol{y}\|_{(\Gamma+\Sigma(\theta))^{-1}}$$

where $\boldsymbol{m}(\theta)$ and $\Sigma(\theta)$ are defined as

$$\begin{aligned}\boldsymbol{m}(\theta) &= \mathcal{G}(m_1(\,\cdot\,;\theta))\\ \Sigma(\theta) &= \mathcal{G}(\bar{\mathcal{G}}(k_1(\,\cdot\,,\,\cdot\,;\theta)))\end{aligned}$$

for m_1 and k_1 as given in Proposition 5.2.1, with their dependence on θ emphasised. When the potential Φ_h is used, the posterior distribution is then given by:

$$\begin{aligned}\frac{\mathrm{d}\mu_\theta^{\boldsymbol{y},h}}{\mathrm{d}\mu_\theta}(\theta) &= \frac{1}{Z_h}\exp(-\Phi_h(\boldsymbol{y};\theta,u^\dagger))\\ Z_h &= \int_\Theta \exp(-\Phi_h(\boldsymbol{y};\theta,u^\dagger))\mu_\theta(\mathrm{d}\theta).\end{aligned}$$

The theoretical result proven in this section is a consistency result for the posterior $\mu_\theta^{\boldsymbol{y},h}$. In the spirit of Stuart [2010], it will be shown that $\mu_\theta^{\boldsymbol{y},h}$ converges to $\mu_\theta^{\boldsymbol{y}}$ as $h \to 0$. The metric used to demonstrate this convergence will be the *Hellinger metric*. For two measure $\mu, \nu \in \mathcal{P}_\mathcal{X}$ which are such that $\mu \ll \nu$, this is given by:

$$d_H(\mu,\nu)^2 := 1 - \int_\mathcal{X}\left(\frac{\mathrm{d}\mu}{\mathrm{d}\nu}(u)\right)^{\frac{1}{2}}\mu(\mathrm{d}u). \tag{5.10}$$

Note that the theoretical results in the previous section now also have a dependence on θ; in particular, the result from Proposition 5.2.6 now has a constant depending on θ, i.e.

$$\sigma(\boldsymbol{x}) \leq C_\theta^F h^{\beta-\rho-d/2}$$

Assumption 5.3.1. There exists a function $C(\|\theta\|_\Theta)$ such that, for each $\theta \in \Theta$,

$$\max\{C_\theta^F, C_\theta^F\|u(\,\cdot\,,\theta)\|_{\hat{k}}, C_\theta^F\|u(\,\cdot\,,\theta)\|_{\hat{k}}^2\} \leq C(\|\theta\|_\Theta)$$

and $\int C(\|\theta\|_\Theta)^4\,\mu_\theta(\mathrm{d}\theta) < \infty$.

Then, we have the following result, which guarantees that when using the new potential defined above to solve the inverse problem, the incurred error compared to the true posterior is bounded by the accuracy of the PMM.

Theorem 5.3.2 (Robustness to Approximation Error)**.** *For each fixed $\boldsymbol{y}$, the posterior distribution $\mu_\theta^{\boldsymbol{y},h}$ satisfies $d_H(\mu_\theta^{\boldsymbol{y},h}, \mu_\theta^{\boldsymbol{y}}) = \mathcal{O}(h^{\beta-\rho-d/2})$.*

Proof. See Appendix D.1. □

Note that this result only guarantees that $\mu_\theta^{\boldsymbol{y},h}$ is close to $\mu_\theta^{\boldsymbol{y}}$ in Hellinger metric, but says nothing about the properties of $\mu_\theta^{\boldsymbol{y}}$ itself. Thus, if $\mu_\theta^{\boldsymbol{y}}$ is particularly complex or challenging to interrogate, there is no guarantee that using the PMM will ameliorate this.

5.4 Numerical Results

Two numerical experiments are now introduced. In Section 5.4.1, a simple simulation study is constructed. Then in Section 5.4.2 an application to EIT is presented.

5.4.1 Illustrative Example

In this section we will apply the PMM to a simple one-dimensional test problem. Both the forward and inverse problems will be considered.

Forward Problem

Consider Poisson's equation in one dimension:

$$\begin{aligned} -\nabla^2 u(x) &= g(x) & \quad x &\in (0,1) \\ u(x) &= 0 & \quad x &\in \{0,1\}. \end{aligned}$$

The simplicity of this problem is such that the Green's function has an explicit closed-form, which can be computed by direct integration:

$$G(x,x') = \begin{cases} x(x'-1) & \text{for } x > x' \\ x'(x-1) & \text{for } x < x'. \end{cases}$$

We will solve this problem using PMM for two choices of prior covariance: k_{nat} and $\hat{k}$ from Section 5.2.3 respectively, computation of which will be discussed now.

Computation of the Natural Kernel To ensure computability of the natural kernel, the prior covariance for g was taken to be the compactly supported polynomial covariance function of Wendland [1995]:

$$\Lambda(x,x') = \sigma_0 \max(1 - \epsilon^{-1}|x-x'|, 0)^2.$$

Here ϵ is a parameter controlling the width of the support, so that Λ is nonzero wherever $|x - x'| < \epsilon$. The amplitude parameter σ_0 controls the prior width. The natural kernel

$$k_{\text{nat}}(x, x') = \int_0^1 \int_0^1 G(x, z)G(x', z')\Lambda(z, z')\mathrm{d}z\mathrm{d}z'$$

is available in closed form since G and Λ are each piecewise polynomial. Computations are lengthy however, and were performed automatically using symbolic integration software.

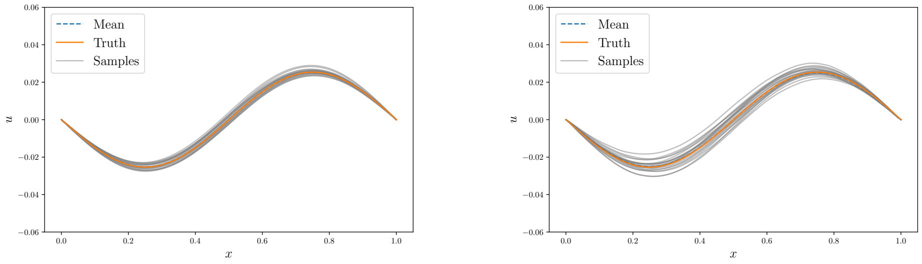
Computation of $\hat{k}$ To illustrate performance for a more practical choice of covariance, $\hat{k}$ is also computed. The base covariance function $\tilde{k}$ was taken to be a higher-order Wendland covariance function

$$\tilde{k}(x, x') = \sigma_0 \max(1 - \epsilon^{-1}|x - x'|, 0)^4 \cdot (4\epsilon^{-1}|x - x'| + 1).$$

This kernel conforms to the differential order of the PDE in question, as $\tilde{k}$ is twice differentiable at the origin. Again, despite of the clear computability of $\hat{k}$ as the integral of a polynomial, the computations required are lengthy and were performed using symbolic integration software.

Results In Fig. 5.1 the posterior mean and samples from the full posterior distribution are plotted for each choice of prior covariance, with $g(x) = \sin(2\pi x)$. The exact solution, $u^\dagger(x) = (2\pi)^{-2}\sin(2\pi x)$, is also shown. Both the covariance functions $\Lambda(x, x')$ and $\tilde{k}$ were assigned a support of $\epsilon = 0.4$, and the prior amplitude σ_0 was estimated using the empirical Bayes procedure described in Rasmussen and Williams [2006, Section 5.4]. The design points $\mathcal{X}^{\mathcal{A}}$ were taken to be take $m_{\mathcal{A}} = 39$ equi-spaced points in $(0, 1)$. For the prior covariance $\hat{k}$ this is augmented with $X^{\mathcal{B}} = \{0, 1\}$ so that the conditional measure satisfies the boundary conditions almost-surely.

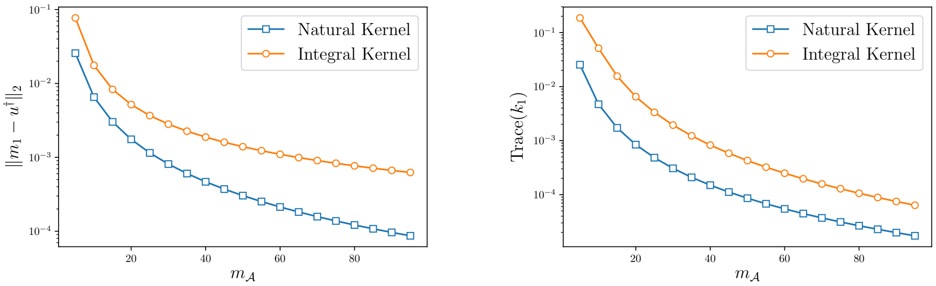
Fig. 5.2 shows convergence of the posterior mean $m_1(\boldsymbol{x})$ for each prior as the number of design points is increased. Note that the natural kernel clearly exhibits a reduction in the error incurred, and the convergence *rate* also appears to be somewhat faster. Fig. 5.2b plots the convergence of the trace of the posterior covariance, as a measure of the posterior width. The fact that this is of approximately the same magnitude as $\|m_1 - u^\dagger\|_2$ for each $m_{\mathcal{A}}$ suggests that the posterior provides a reasonable estimate of the error, even when $m_{\mathcal{A}}$ is small.



(a) Posterior based on k_{nat}.

(b) Posterior based on $\hat{k}$.

Figure 5.1: Comparison of posterior distributions from Proposition 5.2.1 based both upon the natural kernel k_{nat} and on the kernel $\hat{k}$.



(a) Error in conditional mean, $\|m_1 - u^\dagger\|_2$

(b) Residual uncertainty, $\|\sigma^2\|_1$

Figure 5.2: Convergence of posterior mean and covariance as the number $m_{\mathcal{A}}$ of design points is increased.

Inverse Problem

The application of the PMM to inverse problems will now be examined by introducing a parameter into the forward problem as follows:

$$\begin{aligned} -\nabla \cdot (\theta \nabla u(x)) &= g(x) \quad \text{for } x \in (0,1) \\ u &= 0 \qquad\;\; \text{for } x \in \{0,1\}. \end{aligned} \tag{5.11}$$

Again, take $g(x) = \sin(2\pi x)$. With true parameter value $\theta^\dagger = 1$, the observation operator $\mathcal{G}(\,\cdot\,)$ was a vector of two evaluation operators at $x = 0.25$ and $x = 0.75$, i.e.

$$\mathcal{G}(u(\,\cdot\,;\theta)) \coloneqq \begin{bmatrix} u(0.25,\theta) \\ u(0.75,\theta) \end{bmatrix}$$

The noise covariance was taken to be $\Gamma = 0.001^2 I$. The prior over theta was taken to be log-Gaussian, so that $\log\theta \sim \mathcal{N}(0,1)$. Fig. 5.3 shows posteriors for a number of values of $m_{\mathcal{A}}$ using both symmetric collocation and the PMM as the forward solver, with the former referred to as the "standard" approach and the latter as the "probabilistic" approach. Fig. 5.4 shows convergence as $m_{\mathcal{A}}$ is increased.

Comparing the posteriors in Fig. 5.3 and Fig. 5.4, note that when the standard approach is used (right column), the posterior variance does not change as the discretisation resolution $m_{\mathcal{A}}$ is varied for either the natural prior covariance or the prior covariance $\hat{k}$. The posteriors are highly peaked and 1 s.d. credible intervals do not place significant mass in the region of $\theta^\dagger = 1$. Conversely, when the PMM forward solver is used (left column) we see that the posterior is wider and places more mass around $\theta^\dagger$. Comparing the posteriors from k_{nat} and $\hat{k}$, note that faster convergence is achieved with the natural kernel, which should be expected considering the reduced variance exhibited for the forward problem.

5.4.2 Application to Electrical Impedance Tomography

We now turn again to EIT, as introduced in Section 2.4.2. In this section we work with the PEM; while both the complete-electrode model and the point-electrode model are linear for fixed κ (as a function of u), the boundary integrals required for the electrodes are analytically intractable. Thus the point electrode model serves as a convenient test-bed and removes a potential source of discretisation error. As in Section 4.6.2, the interest is in whether the quantification of uncertainty provided by PNMs, specifically the PMM introduced in this chapter, can be used to reduce the amount of computational effort expended to sample from the posterior while

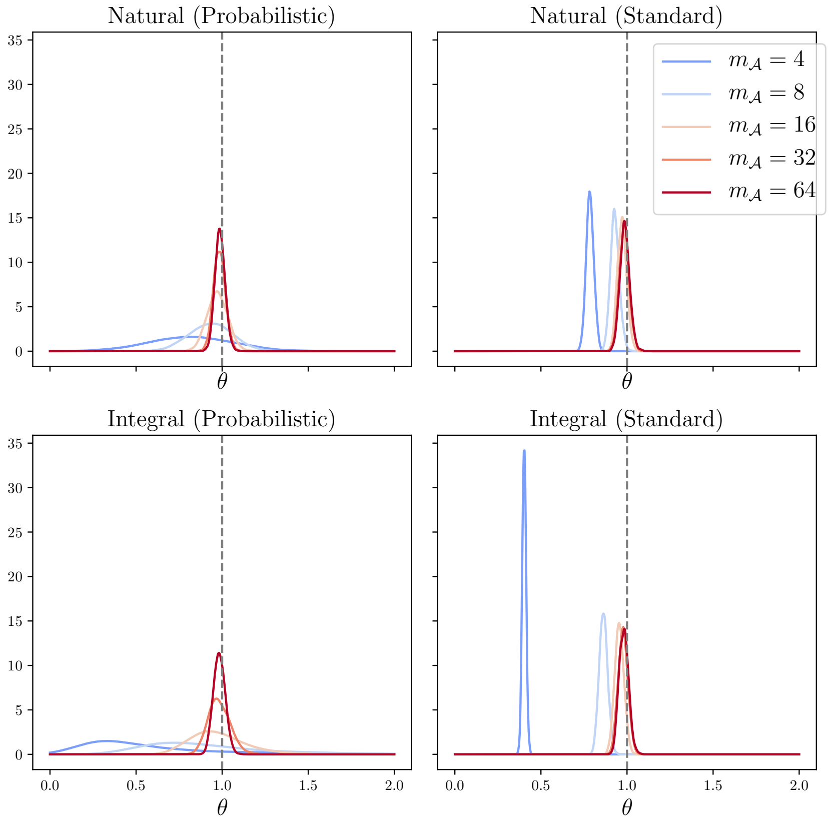


Figure 5.3: Posteriors $\mu_\theta^{\boldsymbol{y},h}$ for the parameter θ from Section 5.4.1, with a PMM forward solver (left column) versus posteriors generated using a symmetric collocation forward solver (right-column), at various discretisation resolutions and for the two prior covariance choices k_{nat} and $\hat{k}$.

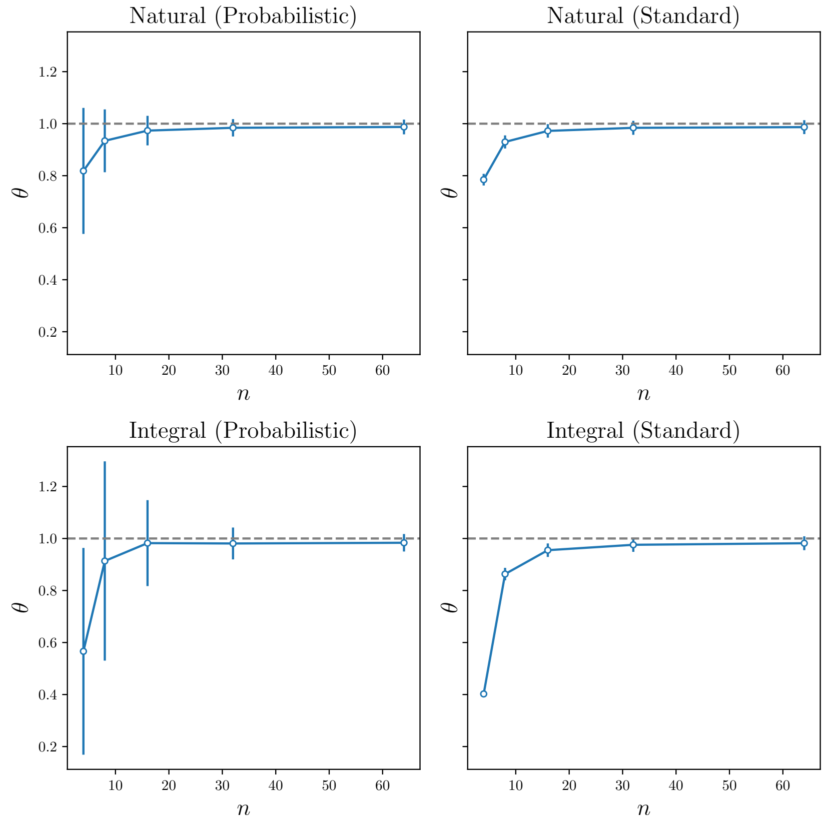
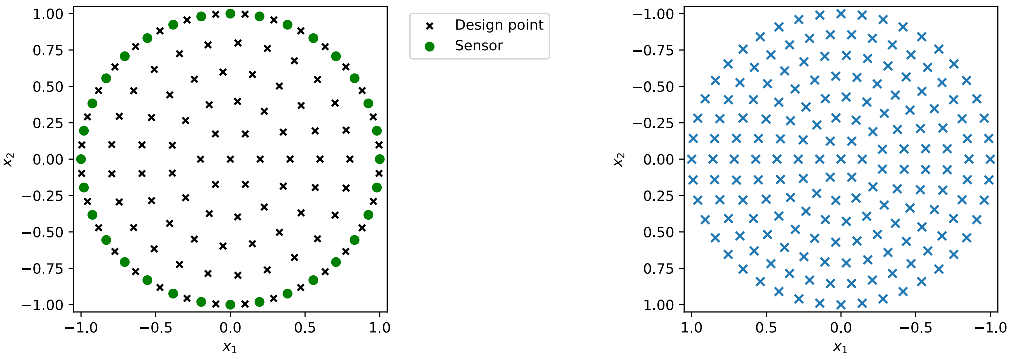


Figure 5.4: 1 s.d. credible intervals for the parameter θ from Section 5.4.1 as a function of the number of design points. When the discretisation is very course (small $m_{\mathcal{A}}$) the posterior distributions are biased and overconfident with the standard forward solver, while with the PMM forward solver they are widened.



(a) Discretisation of $u(\boldsymbol{x})$, with $m_{\mathcal{A}} = 96$.

(b) Discretisation of $\theta(\boldsymbol{x})$

Figure 5.5: Designs used in the EIT experiment in Section 5.4.2.

maintaining statistically valid inferences.

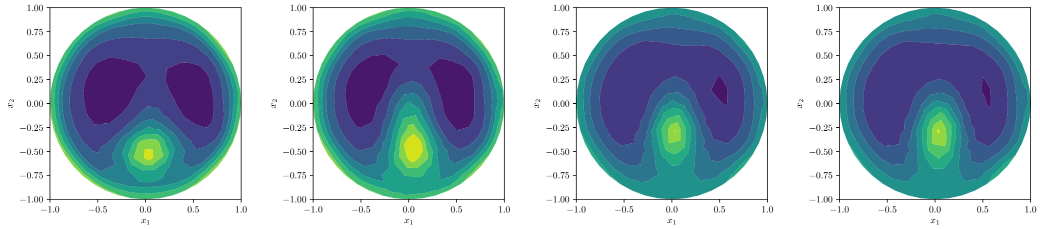
Throughout, the data introduced in Section 2.4.2 is used. The voltage measurements were assumed to have been corrupted with Gaussian noise with standard deviation 5.0, chosen heuristically owing to the fact that the measurement error in the experiment is unknown. The domain was again taken to be a unit disc, and for the purposes of the forward solver this was discretised using a regular design for different $m_{\mathcal{A}}$. One example of such a design is depicted in Fig. 5.5a.

The prior μ_u was taken to be $\mu_u \sim \mathcal{GP}(0, \hat{k})$ for $\hat{k}$ an exponentiated quadratic covariance as given in Eq. (2.6). Note that the conjugacy properties of the squared exponential are such that this is equivalent to taking $\tilde{k}$ to be squared exponential. The amplitude parameter was fixed to $\sigma = 100$ to match the width of the prior to the scale of boundary voltage observations. The length-scale parameter ℓ was endowed with a half-range Cauchy prior (as recommended in Gelman [2006]) and marginalised in the MCMC procedure.

A log-Gaussian prior was assigned to the conductivity field, so that $\kappa(\boldsymbol{x}) = \log(\theta(\boldsymbol{x}))$ and $\mu_\theta = \mathcal{GP}(m_\theta, k_\theta)$. The prior covariance k_θ was again taken to be squared exponential, with fixed length-scale and amplitude $\ell = 0.3$ and $\sigma = 1.0$. The prior mean m_θ was taken to be a constant function, with the constant chosen by maximising the log-likelihood of the observations over constant conductivity fields.

Samples from the posterior distribution were produced using the preconditioned Crank–Nicolson (pCN) method described in Section 2.4.1. The conductivity field was discretised to a grid of 177 points, depicted in Fig. 5.5b. Posterior samples from this section are based on $5,000,000$ iterations of pCN, after $5,000,000$ iterations of burn-in.

Fig. 5.6 shows the posterior conductivity fields $\kappa(\boldsymbol{x})$ obtained for $m_{\mathcal{A}} = 96$,



(a) $m_{\mathcal{A}} = 96$ (b) $m_{\mathcal{A}} = 127$ (c) $m_{\mathcal{A}} = 165$ (d) $m_{\mathcal{A}} = 209$

Figure 5.6: Mean of $\theta(\boldsymbol{x})$ for Section 5.4.2. Each figure shows the posterior mean for the PMM forward solver.

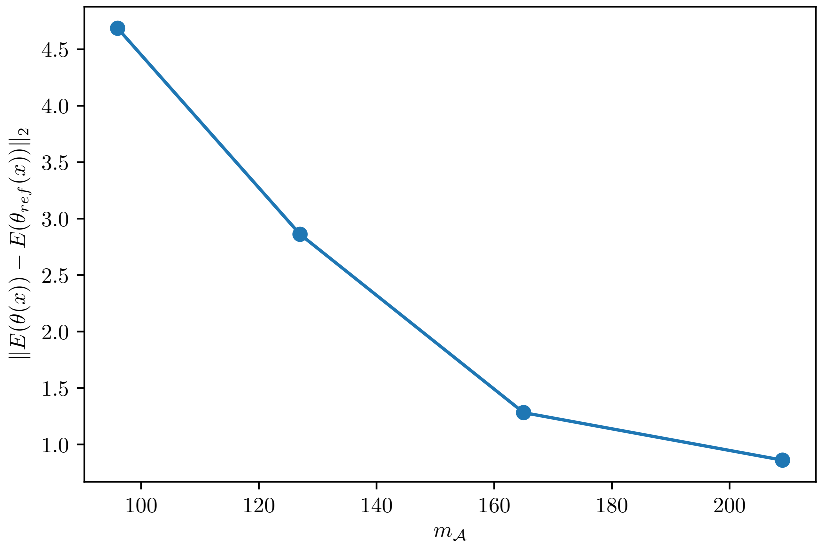
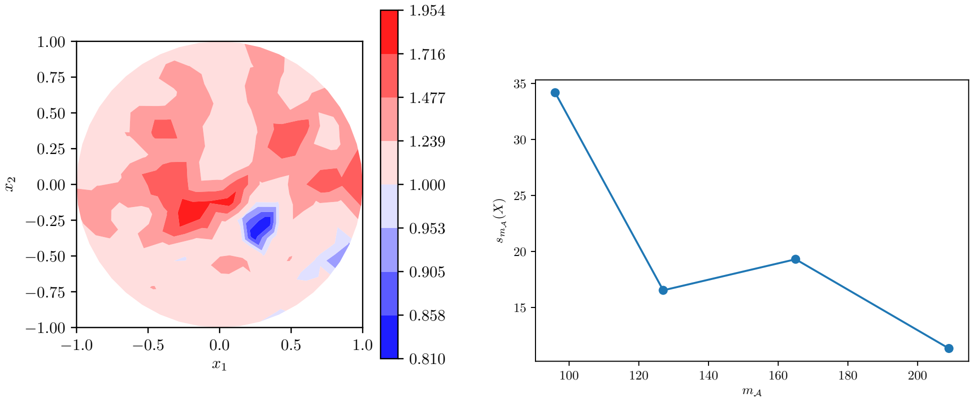


Figure 5.7: Convergence of the posterior mean to the reference field θ_{ref} in Section 5.4.2.

127, 165 and 209 design points. Note that these are qualitatively similar to the agar targets displayed in Fig. 2.1, even at the coarsest discretisation level. The convergence of these fields to a reference conductivity field θ_{ref} is displayed in Fig. 5.7, where θ_{ref} was obtained by using a high-quality symmetric collocation forward solver with $m_{\mathcal{A}} = 259$.

While the posterior mean is an important statistic of the posterior, naturally the focus should be on the posterior UQ provided. In Fig. 5.8a the pointwise posterior variance obtained the posteriors over θ using a PMM forward solver are compared to those using the reference solver. Note that the posterior variance is generally larger throughout the domain.

Inspired by the UQ evaluation performed in Section 4.6.1, a summary statis-



(a) Ratio of the variance in the posterior distribution arising from using a PMM forward solver, compared to a symmetric collocation forward solver, at $m_{\mathcal{A}} = 96$ points.

(b) Convergence of the statistic $s_{m_{\mathcal{A}}}$ as a function of $m_{\mathcal{A}}$.

Figure 5.8: Posterior variance analysis for the analysis in Section 5.4.2.

tic was also computed to evaluate the UQ. Let

$$s_{m_{\mathcal{A}}} := \int_{\Theta} \|\Sigma_{\text{ref}}^{-1/2}(\theta - m_{\text{ref}})\|_2^2 \, \mu_\theta^{\boldsymbol{y},h}(\mathrm{d}\theta)$$

where $\mu_\theta^{\boldsymbol{y},h}$ is the posterior for θ using a PMM forward solver with $m_{\mathcal{A}}$ design points, while the mean m_{ref} and covariance Σ_{ref} are the posterior mean and covariance over θ obtained from the symmetric reference forward solver. Since the posterior distribution over θ is non-Gaussian, no theoretical distribution for this statistic can be derived. However, this statistic assesses “how much overlap” there is between the reference posterior and $\mu_\theta^{\boldsymbol{y},h}$. When $s_{m_{\mathcal{A}}} < 1$ the PMM posterior $\Pi_\theta^{\boldsymbol{y},h}$ is interpreted as being over-confident, suggesting a failure to properly account for discretisation uncertainty, though $s_{m_{\mathcal{A}}} > 1$ does not necessarily imply that the UQ is well-calibrated. The statistic is plotted, for different $m_{\mathcal{A}}$, in Fig. 5.8, and as expected the results show that more conservative inferences are obtained when using the PMM forward solver.

5.5 Discussion

In this chapter we have thoroughly explored the construction of conjugate BPNM in an infinite-dimensional setting. While many of the details resemble the finite-dimensional setting described in Chapter 4, there were some technicalities arising from the extension to function spaces. In particular, it is difficult to construct a highly informative set of search directions as in Chapter 4 since the computation of inner products requires evaluation of integrals that cannot be computed in closed-form. On the other hand, because the choice of information was limited to evaluation functionals and was thus independent of $u^\dagger$, the issues with poor posterior UQ were not present. Application of the PMM to inverse problems was also considered, with similar results to Section 4.6.2.

In the next part of the thesis we will depart from the conjugate setting, and explore the construction of BPNM for generic priors and information operators. We will then formalise the composition of BPNM in pipelines of computation.

Part III

Non-Conjugate Methods

Chapter 6

Beyond Conjugacy

> "What does it mean to 'know' a function? The formula says some things (e.g. f is smooth, positive, and bounded by 20 on $[0, 1]$), but there are many other facts about f that we don't know (e.g. is f monotone, unimodal or convex?).
>
> Once we allow that we don't know f, but do know some things, it becomes natural to take a Bayesian approach..."
>
> —*Persi Diaconis, 1988*

In the previous chapters, only conjugate PNM were considered. However the restriction to conjugate problems would rule out application of PNM to many of the most challenging applications in numerical analysis, in which discretisation error is genuinely a limiting factor. In particular, for the two examples given in Section 1.2.1 (climate modelling and electrical conductivity in the heart) the most realistic models used are nonlinear, and so the methods from Part II cannot be applied directly.

Other authors have introduced PNM for nonlinear problems. As mentioned in Section 5.1.2, the literature on PNM for ODEs such as Chkrebtii et al. [2016]; Schober et al. [2014]; Conrad et al. [2017] generally assumes nonlinearity. Similarly, Raissi et al. [2018] constructs PNM for nonlinear PDEs. However, it is far from clear that these methods are *Bayesian* in the sense of Definition 3.1.5, as while they frequently employ Bayes' rule at certain points in the procedure, it is not the case that the output distribution is the conditional of the prior on data obtained.

In this chapter we will first present conditions under which a Bayesian PNM is well-defined, in Section 6.1. In that section we will also present connections between BPNM and decision theory. Sampling schemes for nonconjugate BPNM will be discussed in Section 6.2, and some experimental results will be presented in Section 6.3.

6.1 Bayesian Probabilistic Numerical Methods

In this section we begin by recapping the notation from Chapter 3 in Section 6.1.1. We then introduce the core theoretical challenge behind defining posterior distributions in this setting in Section 6.1.2, before introducing *disintegrations* in Section 6.1.3 to address this challenge. Lastly, connections between BPNM and decision theory are presented in Section 6.1.4.

6.1.1 Notation: A Recap

Recall the following from Chapter 3. Let $\mathcal{X}$ and $\mathcal{Y}$ be measureable spaces with Borel sigma-algebras $\mathcal{B}_{\mathcal{X}}$, $\mathcal{B}_{\mathcal{Y}}$, and assume that $\dim(\mathcal{Y}) < \infty$. Let $u^\dagger \in \mathcal{X}$ denote an unknown that we wish to recover, and let let $\mu \in \mathcal{P}_{\mathcal{X}}$ be a distribution referred to as the *prior* that reflects the user's prior beliefs about $u^\dagger$. Let $A : \mathcal{X} \to \mathcal{Y}$ be a measurable *information operator* such that, for each $u \in \mathcal{X}$, $A(u)$ can be computed without knowledge of u. Lastly, let $\mathcal{Q}$ be a measureable space with Borel sigma-algebra $\mathcal{B}_{\mathcal{Q}}$ and let $Q : \mathcal{X} \to \mathcal{Q}$ be a measurable *quantity of interest operator*.

From Definition 3.1.4, a *probabilistic numerical method* (PNM) is then defined by an update rule $\mathfrak{A} : \mathcal{P}_{\mathcal{X}} \times \mathcal{Y} \to \mathcal{P}_{\mathcal{X}}$ which updates a user's prior belief to a posterior belief. The method itself, $M : \mathcal{P}_{\mathcal{X}} \times \mathcal{Y} \to \mathcal{P}_{\mathcal{Q}}$, is defined by pushing the output from the update rule through the QoI operator, i.e.:

$$M(\mu, \boldsymbol{y}) = Q_{\#}\mathfrak{A}(\mu, \boldsymbol{y})$$

According to Definition 3.1.5 such a method is a *Bayesian probabilistic numerical method* if $\mathfrak{A}(\mu, \boldsymbol{y}) = \mu^{\boldsymbol{y}}$ for each $\boldsymbol{y} \in \mathcal{Y}$, where $\mu^{\boldsymbol{y}}$ denotes the conditional distribution of μ on $\boldsymbol{y}$. The set $\mathcal{X}^{\boldsymbol{y}}$ is a level set of A or, alternatively, the preimage $A^{-1}(\boldsymbol{y})$, i.e.

$$\mathcal{X}^{\boldsymbol{y}} = \{u \in \mathcal{X} : A(u) = \boldsymbol{y}\}.$$

6.1.2 Conditioning on Null Sets

A core issue, alluded to in Chapter 3, is that most definitions of a conditional distribution involve some variant of Bayes' theorem. In the present setting the more general version given in Section 2.4 is the most appropriate, so that the posterior would be defined as

$$\frac{\mathrm{d}\mu^{\boldsymbol{y}}}{\mathrm{d}\mu}(u) = \frac{1}{Z}\mathbb{I}[\mathcal{X}^{\boldsymbol{y}}](u)$$
$$Z = \int_{\mathcal{X}} \mathbb{I}[\mathcal{X}^{\boldsymbol{y}}](u)\mu(\mathrm{d}u) = \mu(\mathcal{X}^{\boldsymbol{y}}).$$

This construction of the posterior relies on the assumptions of the Radon–Nikdodym theorem, namedly that $\mu^{\boldsymbol{y}} \ll \mu$. However, in the case that $\mathcal{X}^{\boldsymbol{y}}$ is a null-set of the prior, the entire support of $\mu^{\boldsymbol{y}}$ is a null-set of μ and absolute continuity does not hold.

This setting is the most common setting for Bayesian PNM, as the submanifold of $\mathcal{X}$ defined by $\mathcal{X}^{\boldsymbol{y}}$ is generally a null-set of μ. Even in the simple Gaussian process regression setting this is clear, since if μ is a Gaussian measure supported on all of $\mathcal{X}$, then for an arbitrary $\boldsymbol{x} \in D$, $y \in \mathbb{R}$, the set

$$\mathcal{X}^y = \{u \in \mathcal{X} : u(\boldsymbol{x}) = y\}$$

clearly has the property $\mu(\mathcal{X}^y) = 0$, provided that $\mathcal{X}$ is a strict superset of $\mathcal{X}^y$. While verifying this fact for general μ and A is more challenging, empirically it is generally the case for the priors and information operators used in PNM. Nevertheless, the posterior distribution in this setting *is* well-defined, as can be seen in Rasmussen and Williams [2006]. Thus we now turn to the problem of establishing a more general notion of a conditional distribution which supports this setting.

6.1.3 Disintegrations and the Disintegration Theorem

The challenge for establishing the existence of posterior distributions from BPNM is first to introduce an appropriate generalisation of a conditional distribution which allows for the support of conditionals to be null-sets of the prior. Conditioning on null sets is a part of Kolmogorov's foundational measure-theoretic construction of probability, in Kolmogorov [1933]. The means by which this was formalised in that work was *regular conditional probability*. In this work we instead work with *disintegrations*, as argued for in Chang and Pollard [1997], defined below following Dellacherie and Meyer [1978].

Definition 6.1.1 (Disintegration)**.** A collection $\{\mu^{\boldsymbol{y}}\}_{\boldsymbol{y}\in\mathcal{Y}} \subset \mathcal{P}_{\mathcal{X}}$ is a *disintegration* of μ with respect to A if:

1. (Support:) The measure $\mu^{\boldsymbol{y}}$ is supported on $\mathcal{X}^{\boldsymbol{y}}$ for $A_{\#}\mu$-almost all $\boldsymbol{y} \in \mathcal{Y}$, i.e. $\mu^{\boldsymbol{y}}(\mathcal{X} \setminus \mathcal{X}^{\boldsymbol{y}}) = 0$.

and for each measurable $f\colon \mathcal{X} \to [0, \infty)$ it holds that

2. (Measurability:) The map $\boldsymbol{y} \mapsto \mu^{\boldsymbol{y}}(f)$ is measurable.
3. (Conditioning:) $\mu(f) = \int \mu^{\boldsymbol{y}}(f) A_{\#}\mu(\mathrm{d}\boldsymbol{y})$.

Properties (1) and (2) are each natural conditions for the required conditional distribution to have. In particular, property (1) is what separates a disintegration from a regular conditional distribution[1]. Property (3) is the condition that makes disintegrations interpretable as conditional distributions, as this is in essence the law of total probability. Thus, disintegrations serve as an appropriate notion of a posterior distribution for BPNM, when the property $\mu(\mathcal{X}^{\boldsymbol{y}})$ holds for $A_{\#}\mu$-almost-all $\boldsymbol{y} \in \mathcal{Y}$. It remains to discuss conditions on the map A and the measure μ under which a disintegration can be said to exist, and be unique. These are provided by the Disintegration theorem, quoted here from Chang and Pollard [1997].

Theorem 6.1.2 (Disintegration Theorem [Chang and Pollard, 1997, Theorem 1])**.** *Let the following conditions hold:*

1. *$\mathcal{X}$ is a metric space with Borel σ-algebra $\mathcal{B}_{\mathcal{X}}$.*
2. *$\mu \in \mathcal{P}_{\mathcal{X}}$ is a Radon measure.*
3. *$\mathcal{B}_{\mathcal{Y}}$ is countably generated with $\{\boldsymbol{y}\} \in \mathcal{B}_{\mathcal{Y}}$ for each $\boldsymbol{y} \in \mathcal{Y}$.*

Then there exists a disintegration $\{\mu^{\boldsymbol{y}}\}_{\boldsymbol{y}\in\mathcal{Y}}$ of μ with respect to A. Moreover, if $\{\nu^{\boldsymbol{y}}\}_{\boldsymbol{y}\in\mathcal{Y}}$ is another such disintegration, then $\{\boldsymbol{y} \in \mathcal{Y} : \mu^{\boldsymbol{y}} \neq \nu^{\boldsymbol{y}}\}$ is a $A_{\#}\mu$-null set.

Property (1) is a natural condition to hold for the present application; most numerical methods require the definition of some metric on their solution space in order to measure convergence. Property (2) holds whenever $\mathcal{X}$ is both separable and complete; again, these are mild conditions and are generally required for the definition of a useful probability measure on $\mathcal{X}$. Such a space is an example of a *Radon space*. Property (3) is easily satisfied in the present setting when $\mathcal{Y} = \mathbb{R}^n$, and $\mathcal{B}_{\mathcal{Y}}$ is the Borel σ-algebra. Thus, under very mild conditions BPNMs are well-defined, in the sense that the required disintegration exists and is essentially unique.

[1] Recall that for regular conditional distributions, non-uniqueness results in many different “versions” of a conditional measure, each of which satisfies the definition. According to Chang and Pollard [1997], the additional requirement (1) amounts to “a careful selection of versions of the conditional expectations (in Kolmogorov’s sense)”, resulting in a more intuitive theoretical construct.

6.1.4 Decision-Theoretic Treatment

In this section we apply results from decision theory to define an appropriate notion of *contraction* for PNM, as well as to explain the frequently observed property that PNM often reproduce classical numerical methods as their posterior mean. Both of these ideas were discussed in both Chapter 4 and Chapter 5. The fact that PNMs often reproduce classical numerical methods has also been observed frequently in the literature, in Briol et al. [2019], Karvonen and Särkkä [2017] and Karvonen et al. [2018] for quadrature rules.

Contraction of PNM

Let $L\colon \mathcal{Q} \times \mathcal{Q} \to \mathbb{R}$ denote an integrable loss function, where $L(q^\dagger, q)$ assigns an abstract numerical loss to the setting when the QoI $q^\dagger = Q(u^\dagger)$ is incorrectly estimated with another $q \in \mathcal{Q}$.

A BPNM can be regarded as providing a *randomised decision rule* for selection of q through its posterior $M(\mu, \boldsymbol{y}) = Q_{\#}\mu^{\boldsymbol{y}}$. To assess such rules, the loss function is generally averaged over the output of the randomised decision rule to produce a *risk function*. Letting $\nu \in \mathcal{P}_{\mathcal{Q}}$ denote an arbitrary randomised decision rule for q, the risk function $r\colon \mathcal{Q} \times \mathcal{P}_{\mathcal{Q}} \to \mathbb{R}$ is then defined as:

$$r(q^\dagger, \nu) = \int_{\mathcal{Q}} L(q^\dagger, q)\, \nu(\mathrm{d}q) \; .$$

A distribution $\mu \in \mathcal{P}_{\mathcal{X}}$ can be thought of as describing a class of problems which the user expects to need to solve using a numerical method such as a PNM. It is then natural to consider the *average risk* or, when μ is the prior, the *Bayes risk* incurred over all problems described by μ, as a measure of how well the BPNM estimates q on average. For a PNM, this is defined as

$$R(\mu, M, A) = \int_{\mathcal{X}} r(Q(u), M(\mu, A(u)))\, \mu(\mathrm{d}u). \tag{6.1}$$

The Bayes risk can be used to introduce a notion of *contraction* of BPNM under a sequence of information operators (A_n), for $n \in \mathbb{N}$, where each $A_n : \mathcal{X} \to \mathcal{Y}_n$. These information operators might be thought of as representing an increasing amount of information, so that $\dim(\mathcal{Y}_n) < \dim(\mathcal{Y}_{n+1})$. Thus, in the limit as $n \to \infty$, informally an “infinite amount of information” is obtained, so that u can be recovered exactly (since $\mathcal{X}$ is assumed to be separable). We will now define what it means for this sequence of operators to *contract* at a certain rate when used to estimate $q^\dagger$.

Definition 6.1.3 (Contraction)**.** Let $\varphi(n) : \mathbb{N} \to \mathbb{R}^+$ be such that $\varphi(n) \to 0$ as $n \to \infty$. A sequence (A_n, M_n) of information operators and update rules *contracts* at a rate $\varphi(n)$ under a prior μ if $R(\mu, M_n, A_n) = C\varphi(n)$, for some C independent of n.

Note that a rate of contraction for the PMM was presented in Chapter 5. A similar rate also appeared in Chapter 4, though the rate presented in that section had a somewhat different interpretation; owing to the finite-dimensionality of $\mathcal{X}$, the sequence of information operators (A_n) was defined for $n = 1, \dots, d$ rather than for $n \in \mathbb{N}$. Similar rates have appeared in other literature on BPNMs, such as Briol et al. [2019].

Bayes Decision Rules

We now introduce the concept of a *Bayes rule.*

Definition 6.1.4 (Bayes Rule)**.** A decision rule is said to be a *Bayes rule* if it achieves the minimum Bayes risk among all decision rules. Formally, let $\mathcal{M}$ denote the set of all PNM, i.e.

$$\mathcal{M} = \{M : \mathcal{P}_{\mathcal{X}} \times \mathcal{Y} \to \mathcal{P}_{\mathcal{Q}}\}.$$

Then, for fixed information A, $M^* \in \mathcal{M}$ is a Bayes rule if, for all $M \in \mathcal{M}$

$$R(\mu, M^*, A) < R(\mu, M, A).$$

We denote the set of all Bayes rules by $\mathcal{M}_{\mathrm{B}}$.

Note that this definition encompasses classical numerical methods through Section 3.2. This idea is also closely linked to the concept of an *average-case optimal* numerical method; see Ritter [2000] and Cockayne et al. [2019a] for more detail on the connection between BPNMs and such methods.

Given this definition it is natural to ask whether it is possible for PNMs to be Bayes rules. However, a basic result from Bayesian decision theory states that if $\mathcal{M}_{\mathrm{B}}$ is non-empty then it contains at least one non-random decision rule. This means that while there may be PNM which are Bayes rules, the posterior distribution offers no benefit over a point estimate when assessed in this framework.

The question of whether BPNMs have optimality properties when interpreted as decision rules remains open. The ideas introduced above clearly do not ascribe value to the UQ that these methods provide, but analysis of the UQ enables more informed decisions based upon whether the output is sufficiently accurate that *any* action should be taken, or whether more computational effort should first be spent on the problem. Nevertheless, the exposition of Bayes rules allows us to establish the following result concerning the posterior mean of BPNMs:

Theorem 6.1.5. *Suppose that M is a BPNM and suppose that $\mathcal{Q} \subseteq \mathbb{R}^d$. Suppose that squared-error loss is used, so that*

$$L(q^\dagger, q) = \|q^\dagger - q\|_2^2.$$

Then, for each $\boldsymbol{y} \in \mathcal{Y}$ it holds that if $X^{\boldsymbol{y}}$ is a random variable with law $M(\mu, \boldsymbol{y})$, then $\mathbb{E}(X^{\boldsymbol{y}})$ is a Bayes rule for estimation of $q^\dagger$.

Proof. This is a straightforward result from Bayesian decision theory. From Berger [1985, Section 4.4.1], to determine a Bayes act a it is equivalent to minimise the posterior expected loss for each $\boldsymbol{y} \in \mathcal{Y}$, i.e.

$$J(q) := \int_{\mathcal{Q}} \|q - a\|_2^2 \, \nu^{\boldsymbol{y}}(\mathrm{d}q)$$

where $\nu^{\boldsymbol{y}} = M(\mu, \boldsymbol{y})$. Let $\overline{\nu^{\boldsymbol{y}}}$ denote the expectation of $\nu^{\boldsymbol{y}}$. We then have that

$$\begin{aligned} \frac{1}{2}\nabla J(q) &= \int_{\mathcal{Q}} (q - a) \, \nu^{\boldsymbol{y}}(\mathrm{d}q) \\ &= \overline{\nu^{\boldsymbol{y}}} - a. \end{aligned}$$

Thus, taking $a = \overline{\nu^{\boldsymbol{y}}}$ causes the derivative to be zero. Furthermore by inspection, the Hessian of $J(q)$ is an identity matrix which is positive-definite; hence this is a minimum. This completes the proof. □

This result explains the fact, observed in both Chapter 4 and Chapter 5 as well as throughout the PNM literature, that PNM frequently have a posterior mean that coincides with a classical numerical method. This is owing to the equivalence between minimising squared-error loss, a natural objective in the construction of numerical methods, and conditional expectation. However, it should be emphasised that while this is an interesting and attractive property of BPNM, it is not considered to be a fundamental requirement, and recent research has revealed new BPNMs without an existing numerical method as their counterpart [e.g. Karvonen et al., 2018; Xi et al., 2018]. Further, it has been noted [Berger, 1985, Section 2.4.2] that the regularity with which squared-error loss appears in the literature has more to do with its tractability than any other merit. Thus, the above theorem should be interpreted only as explaining why it is so often the case that PNM coincide with classical numerical methods, rather than as an advocation of this property as a design criterion.

6.2 Numerical Disintegration

The proof of the disintegration theorem in Section 6.1.3 is unfortunately non-constructive. Thus, while it is guaranteed that the required disintegration will exist, interrogating it remains a challenging numerical problem. In this section we describe a method for approximately sampling from elements of the disintegration. The approach pursued is based on constructing a sequence of distributions $(\mu_\delta^{\boldsymbol{y}})$, for $\delta \in \mathbb{R}^+$, each of which which can be sampled from using standard Monte Carlo methods, and which provably approach $\mu^{\boldsymbol{y}}$ in the limit as $\delta \downarrow 0$.

Disintegrations and regular conditional probabilities have received limited attention in the literature and are generally only treated as objects of mathematical interest. The approach described here is the first method for approximately sampling from disintegrations, and is similar in spirit to analysis which appears in Ackerman et al. [2017]. It borrows from sampling techniques in the literature on rare event simulation [Cérou et al., 2011], and could also be described as a kind of approximate Bayesian computation (ABC) [Del Moral et al., 2012].

It should be noted that while the method described in this section does provide a means to approximately sample from posterior distributions resulting from arbitrary BPNM, the cost incurred is high compared to standard numerical methods for such problems. This is owing to the cost incurred by the Monte Carlo methods employed. As a result, the approach described in this section should be thought of as a proof of concept, to demonstrate that the intractable posterior distributions arising from BPNM *can* be approximately sampled from and to provide a benchmark against which more efficient approximation schemes can be compared.

6.2.1 Approximate Sampling from Disintegrations

We now introduce the sequence of distributions which will be used to approximate the element of the disintegration. Suppose that $\mathcal{Y} = \mathbb{R}^n$. Let $\phi : \mathbb{R}^+ \to \mathbb{R}^+$ denote a function with the following properties:

1. $\phi(0) = 1$.

2. $\phi(r) \to 0$ as $r \to \infty$.

3. ϕ is decreasing, and continuous at 0.

For $\delta > 0$, introduce the *relaxation function* $\phi_\delta^{\boldsymbol{y}} : \mathcal{X} \to \mathbb{R}^+$, defined as

$$\phi_\delta^{\boldsymbol{y}}(u) = \phi\left(\frac{\|A(u) - \boldsymbol{y}\|_{\mathcal{Y}})}{\delta}\right).$$

It is assumed that $\mu(\phi_\delta^{\boldsymbol{y}}) > 0$ for all $\delta > 0$. Then, the *δ-relaxed distribution* $\mu_\delta^{\boldsymbol{y}}$ is defined through its Radon–Nikodym derivative with respect to μ as

$$\frac{\mathrm{d}\mu_\delta^{\boldsymbol{y}}}{\mathrm{d}\mu}(u) = \frac{\phi_\delta^{\boldsymbol{y}}(u)}{Z_\delta^{\boldsymbol{y}}}$$
$$Z_\delta^{\boldsymbol{y}} = \mu(\phi_\delta^{\boldsymbol{y}}).$$

Informally it is clear that $\phi_\delta^{\boldsymbol{y}} \to \mathbb{I}[\mathcal{X}^{\boldsymbol{y}}]$ as $\delta \downarrow 0$. Thus, one might expect that $\mu_\delta^{\boldsymbol{y}} \to \mu^{\boldsymbol{y}}$ as $\delta \downarrow 0$. To make this intuition formal, a notion of convergence on $\mathcal{P}_{\mathcal{X}}$ is required. For the sake of generality, integral probability metrics will be used. While integral probability metrics certainly appeared in earlier works (e.g. Zolotarev [1984] and earlier editions of Dudley [2002]), Müller [1997] provides a thorough characterisation. See also Sriperumbudur et al. [2012].

Definition 6.2.1 (Integral Probability Metric)**.** Let $\mathcal{F}$ be a set of bounded and measurable functions $f : \mathcal{X} \to \mathbb{R}$. Then the *integral probability metric* induced by $\mathcal{F}$ is defined as

$$d_{\mathcal{F}}(\nu, \nu') = \sup_{f\in\mathcal{F}} |\nu(f) - \nu'(f)|$$

Many, though not all, common probability metrics can be defined in this way; in particular, the total variation distance is obtained when $\mathcal{F}$ is the set of all functions with $\|f\|_\infty \leq 1$, while the Wasserstein distance arises from the class of functions with both $\|f\|_\infty \leq 1$ and $\mathrm{Lip}(f) \leq 1$, where $\mathrm{Lip}(f)$ denotes the Lipschitz constant of f. Conversely, the Hellinger metric introduced in Eq. (5.10) is an example of a probability metric that is not a member of this class. Note that the total variation distance is not a useful notion of convergence for disintegrations, as all elements of the disintegration have disjoint support and the total variation distance between two measures with disjoint support is always 1.

It is also important to note that, depending on the function class $\mathcal{F}$, $d_{\mathcal{F}}$ may not satisfy the full definition of a metric given in Definition A.1.1. In paritcular, the property $d_{\mathcal{F}}(\mu, \nu) = 0 \iff \mu = \nu$ may not be satisfied. In such cases $d_{\mathcal{F}}$ is merely a pseudometric. However this is not critical for our analysis.

We now introduce the set of assumptions under which it can be proven that $d_{\mathcal{F}}(\mu_\delta^a, \mu^a) \to 0$ as $\delta \downarrow 0$.

Assumption 6.2.2. Assume that $A_{\#}\mu$ admits a positive Lipschitz density p_A with respect to the Lebesgue measure on $\mathcal{Y}$ and with Lipschitz constant L_A. Further assume that $p_A(\boldsymbol{y}) > 0$ for all $\boldsymbol{y} \in \mathcal{Y}$.

Assumption 6.2.3. Let $C_\phi^m := \int r^m \phi(r) \mathrm{d}r$. Then we assume that $C_\phi^n < \infty$ and $C_\phi^{n-1} < \infty$, where $n = \dim(\mathcal{Y})$.

Assumption 6.2.4. The map $\boldsymbol{y} \mapsto \mu^{\boldsymbol{y}}$ is $A_{\#}\mu$-almost-everywhere Lipschitz in $d_{\mathcal{F}}$, so that

$$d_{\mathcal{F}}(\mu^{\boldsymbol{y}}, \mu^{\boldsymbol{z}}) \leq C_\mu \|\boldsymbol{y} - \boldsymbol{z}\|_{\mathcal{Y}}$$

for $A_{\#}\mu$-almost-all $\boldsymbol{y}, \boldsymbol{z} \in \mathcal{Y}$, and for some constant C_μ that depends on μ.

Assumption 6.2.4 can be relaxed to allow $a \mapsto \mu^a$ to be merely α-Hölder, as presented in Cockayne et al. [2019a], but for simplicity that relaxation was not presented here.

Theorem 6.2.5. *Let $\bar{C}_\phi^n := C_\phi^n / C_\phi^{n-1}$, with $n = \dim(\mathcal{Y})$ Then, for $\delta > 0$ sufficiently small,*

$$d_{\mathcal{F}}(\mu_\delta^{\boldsymbol{y}}, \mu^{\boldsymbol{y}}) \leq C_\mu(\bar{C}_\phi^n + 1)\delta$$

for all $\boldsymbol{y}$ such that $\|y\|_{\mathcal{Y}} < P$, where $P < \infty$ is an arbitrary positive constant.

Proof. See Appendix E.1. □

This result justifies the use of samples from $\mu_\delta^{\boldsymbol{y}}$ to approximate $\mu^{\boldsymbol{y}}$, for δ sufficiently small. The constant P affects how small δ must be for the result to hold. Thus, posterior distributions from numerical disintegration could be thought of as being *approximate* BPNM for the numerical problem at hand, in the sense that the error between $\mu_\delta^{\boldsymbol{y}}$ and $\mu^{\boldsymbol{y}}$ is small and controlled by δ. Results demonstrating the convergence of such approximations under a continuity assumption on the disintegration have appeared in Tjur [1980] and Ackerman et al. [2017]. This result is stronger, in that it establishes transfer of the Lipschitz continuity of the disintegration to a rate of convergence in the relaxed disintegration. The result is also valid for ϕ other than indicator functions, which may be more computationally expedient.

Not captured in Theorem 6.2.5 is the impact of *prior truncation* on inferences in BPNM. Recall from Section 2.3.1 that priors on (separable) function spaces are typically constructed by randomising weights of a basis of that space in such a way as to ensure almost-sure convergence. For practical computation this basis must be truncated, as in Eq. (2.4). The results in Cockayne et al. [2019a] attempt to bound the error resulting from this truncation, but assume that the prior can be sampled without error and is truncated within the *likelihood*, which differs from the computational setup. Owhadi et al. [2015] show that in the general case, the error incurred by performing inference with a truncated prior can be arbitrarily large. What assumptions can be made to ensure robustness to prior truncation remains to be established.

6.2.2 Sampling Methods

Having established that approximating μ^a with a distribution μ^a_δ is valid, and that μ^a_δ is suitably well-defined with respect to the prior, methods such as the pCN method, introduced in Section 2.4.1, can be used to obtain samples from μ^a_δ. However, when δ is small the posterior is highly concentrated and samplers can converge very slowly.

To accelerate the convergence of the sampler a tempering scheme inspired by the literature on rare event simulation was adopted. Consider a finite set $\{\delta_1, \dots, \delta_N\}$, $N \in \mathbb{N}$, with $\delta_i > 0$ for $i = 1, \dots, N$ and $\delta_i > \delta_{i+1}$. For convenience, abbreviate $\mu^{\boldsymbol{y}}_{\delta_i} = \mu^a_i$. Then the intuition behind tempering schemes is that if δ_1 is sufficiently large it can be sampled from straightforwardly using standard Monte-Carlo methods, and furthermore if $\delta_i - \delta_{i+1}$ is sufficiently small then $\mu^{\boldsymbol{y}}_i$ is a useful importance sampling distribution for $\mu^{\boldsymbol{y}}_{i+1}$. Thus, the ordered set $\{\mu^{\boldsymbol{y}}_1, \dots, \mu^{\boldsymbol{y}}_{N-1}\}$ is a set of latent distributions on which sampling procedures can be constructed to permit more straightforward sampling from the true target distribution $\mu^{\boldsymbol{y}}_N$.

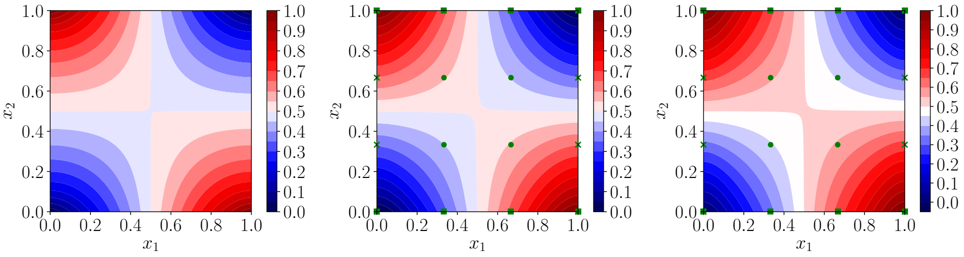
Several schemes for sampling from tempered distributions exist in the literature, most prominent among them sequential Monte Carlo (SMC; Del Moral et al. [2006], see Appendix A.3.2) and parallel tempering (Geyer [1991], see Appendix A.3.3).

6.3 Numerical Experiments

We now turn to a numerical evaluation of the procedure described in the previous section. This is applied to two examples: a simple linear PDE in two dimensions, and a more complex nonlinear boundary value problem in one dimension.

6.3.1 Poisson Equation

Our first illustrative example returns to Poisson's equation, as seen in Section 5.4.1, though in this case the equation is in two spatial dimensions rather than one. To be specific, for $D = [0, 1]^2$ let $\boldsymbol{x} \in D$ be such that $\boldsymbol{x} = [x_1, x_2]^\top$. Then, the PDE



(a) Model solution

(b) Gaussian prior

(c) Cauchy prior

Figure 6.1: (a) Model solution $u(\boldsymbol{x})$ to Poisson's equation, constructed using a fine mesh of 50 × 50 elements. (b, c) Posterior mean for the solution $u(\boldsymbol{x})$ of Poisson's equation with $n = 16$, and under a Gaussian and Cauchy prior. The design points are plotted; green dots represent interior points, green squares represent Dirichlet points and green crosses represent Neumann points.

considered is:

$$-\nabla^2 u(\boldsymbol{x}) = 0 \qquad \boldsymbol{x} \in (0,1)^2 \tag{6.2}$$

$$u(\boldsymbol{x}) = x_1 \qquad x_1 \in [0,1] \qquad x_2 = 0 \tag{6.3}$$

$$u(\boldsymbol{x}) = 1 - x_1 \qquad x_1 \in [0,1] \qquad x_2 = 1 \tag{6.4}$$

$$\partial u / \partial x_2 = 0 \qquad x_2 \in (0,1) \qquad x_1 \in \{0,1\} \tag{6.5}$$

Fig. 6.1a shows a model solution to this system, generated using FEM with a fine mesh. Note that since this PDE is linear in u, under a Gaussian prior an explicit closed-form posterior can be constructed using the methods in Chapter 5. Here the numerical disintegration procedure is used for illustrative purposes.

To admit different prior specifications than the Gaussian, a truncated series prior was used, with the basis functions taken to be given by tensor products of orthogonal polynomials in one spatial dimension. The base polynomials used were normalised Chebyshev polynomials of the first kind as described in Appendix A.4. The basis functions in Eq. (2.4) then given by:

$$\phi_i(\boldsymbol{x}) = T_{i_1}(2x_1 - 1) T_{i_2}(2x_2 - 1)$$

where $i \in \mathbb{N}^2$ is now interpreted as a multi-index. The total degree of polynomials used in the truncated representation of $u(\boldsymbol{x})$ was restricted to N_C, so that the prior distribution is given by the random variable:

$$U^{N_C} = \sum_{|i| \leq N_C} \xi_i \gamma_i \phi_i.$$

Here the scaling parameters were set to $\gamma_i = (|i| + 1)^{-2}$, to guarantee almost-sure convergence, while the random variables ξ_i are taken to be IID, and either unit Gaussian or Cauchy. Throughout, the maximum total order of the polynomials was set to $N_C = 8$, resulting in a total of 45 terms in the summation that defines U^{N_C}.

Pointwise information was used, by defining a regular grid of points $\{\boldsymbol{x}_i\}_{i=1}^n \subset [0,1]^2$ and enforcing either the interior or one of the boundary conditions at each point depending on its location. To be explicit, let X^{L} denote the set of points at which the interior condition (i.e. the Laplacian) is enforced, X^{D} denote the set of points lying on one of the boundaries at which a Dirichlet condition is enforced, and X^{N} denote the set of points lying on a boundary where a Neumann condition is enforced. Let $|X^{\mathsf{L}}| = N_{\mathsf{L}}$, $|X^{\mathsf{D}}| = N_{\mathsf{D}}$ and $|X^{\mathsf{N}}| = N_{\mathsf{N}}$, and let the total number of points $n = N_{\mathsf{L}} + N_{\mathsf{D}} + N_{\mathsf{N}}$. Then, the information operator has the form:

$$A(u) = \begin{bmatrix} A^{\mathsf{L}}(u) \\ A^{\mathsf{D}}(u) \\ A^{\mathsf{N}}(u) \end{bmatrix}$$

where

$$A^{\mathsf{L}}(x) = \begin{bmatrix} -\nabla^2 u(\boldsymbol{x}_1^{\mathsf{L}}) \\ \vdots \\ -\nabla^2 u(\boldsymbol{x}_{N_{\mathsf{L}}}^{\mathsf{L}}) \end{bmatrix}, \quad A^{\mathsf{D}}(x) = \begin{bmatrix} u(\boldsymbol{x}_1^{\mathsf{D}}) \\ \vdots \\ u(\boldsymbol{x}_{N_{\mathsf{D}}}^{\mathsf{D}}) \end{bmatrix}, \quad A^{\mathsf{N}}(x) = \begin{bmatrix} \frac{\partial u}{\partial \boldsymbol{n}}(\boldsymbol{x}_1^{\mathsf{N}}) \\ \vdots \\ \frac{\partial u}{\partial \boldsymbol{n}}(\boldsymbol{x}_{N_{\mathsf{N}}}^{\mathsf{N}}) \end{bmatrix}.$$

Three different information operator resolutions were considered; $n = 16, 25$ and 36.

The posterior distribution was obtained by use of the parallel tempering algorithm described in Appendix A.3.3. In the notation of that section, the distributions μ_i are defined by the relaxation parameters δ_i, which were taken to be equally spaced on a logarithmic scale between 10^{-2} and 10^{-4}. At each temperature, ten iterations of a preconditioned MALA sampler were used to provide the transition kernel. This sampler is described in Appendix A.3.1, and can be implemented efficiently owing to the ease with which gradients of each log-likelihood can be obtained in the present setting. The preconditioner Γ was taken to be $\Gamma = \operatorname{diag}(\gamma)$, to ensure that the scale of proposals matches the scaling of the prior coefficients. The total number of swaps, P in the notation of Appendix A.3.3, was adapted to the resolution of the information operator. When $n = 16$, $P = 10^6$, and when $n = 25$ or $n = 36$, $P = 10^7$.

Posterior means from the numerical disintegration procedure with $\delta = 10^{-4}$ and $n = 16$ are reported in Fig. 6.1b (for the Gaussian prior) and Fig. 6.1c (for

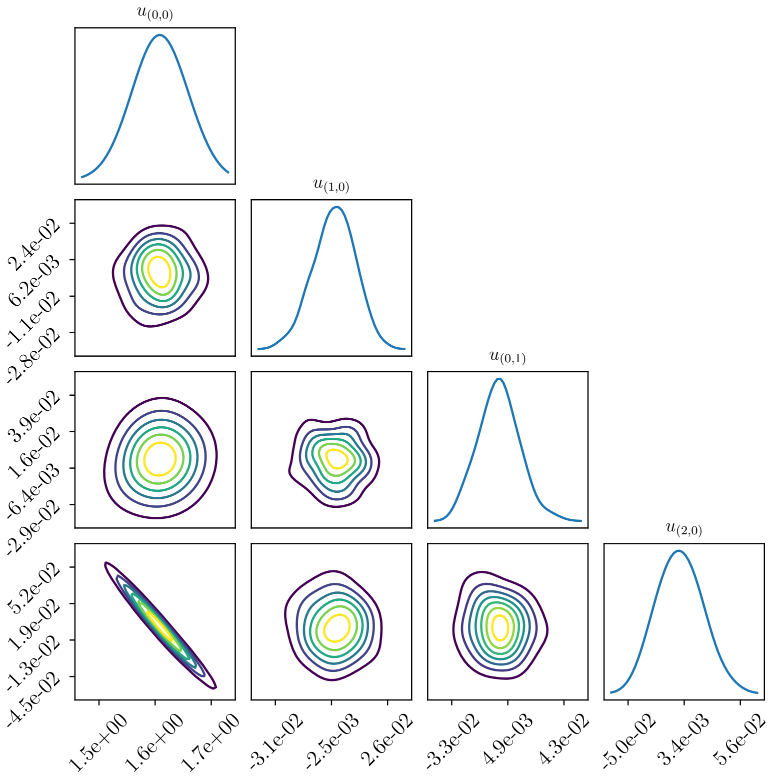
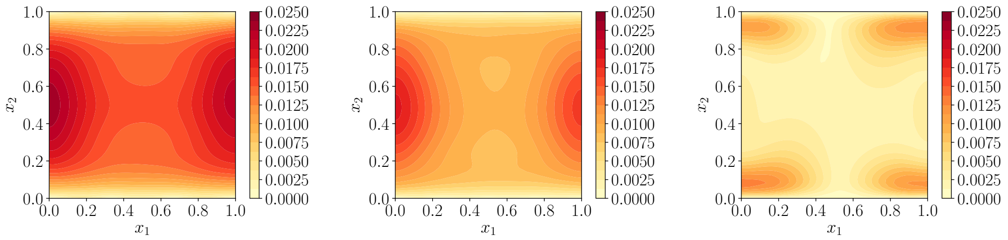


Figure 6.2: Posterior distribution over the first four coefficients in the spectral expansion for the solution $u(\boldsymbol{x})$ to the Poisson equation. The posteriors plotted are from numerical disintegration with a Gaussian prior, where $\delta = 0.0008$, $n = 16$.

the Cauchy prior). Little qualitative difference can be observed between these two figures and the model solution shown in Fig. 6.1a.

The spectrum of the posterior distribution, i.e. the posterior distribution over the coefficients u_i, $|i| < N_C$, is reported in Fig. 6.2, in this case for the Gaussian prior. Note that the posterior distributions appear to be approximately Gaussian, as expected due to the linearity of the problem.

Lastly, the posterior variance is examined in Fig. 6.3. In the first panel, Fig. 6.3a, it is clear that the posterior variance is lowest at the boundaries where Dirichlet conditions have been enforced, where the solution is known explicitly. Posterior variance rises as the solution deviates from these boundaries, and peaks near the boundaries where the Neumann condition has been enforced, reflecting that the Neumann condition provides significantly less information about the value of the solution. This behavior is also seen in Fig. 6.3b, but with diminished variance owing to the finer discretisation of the domain. Fig. 6.3c displays more pathological behavior, likely due to the fact that the amount of information is now approaching



(a) $n = 16$ (b) $n = 25$ (c) $n = 36$

Figure 6.3: Point-wise standard deviation for the posterior over Poisson's equation under a Gaussian prior, as the number of design points n is varied.

the number of degrees of freedom in the truncated prior. This demonstrates that many more than n degrees of freedom in the prior are needed to ensure that the uncertainty quantification provided is not affected by prior truncation.

6.3.2 The Painlevé ODE

In this section, the numerical disintegration procedure is applied to produce a BPNM for solution of Painlevé's first transcendental, a boundary value problem in one dimension:

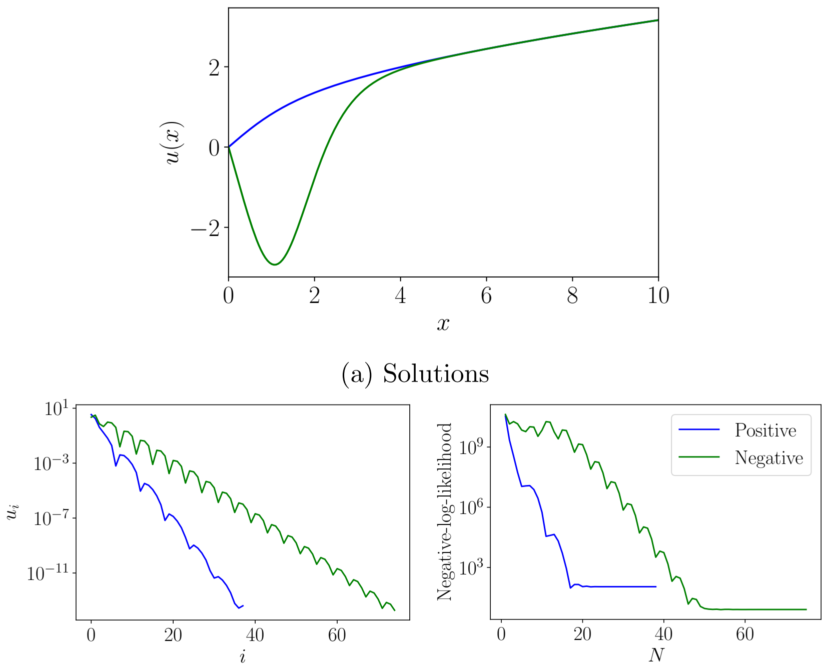
$$\begin{aligned} u'' &= u^2 - u, && x \in [0, \infty) \\ u(0) &= 0 \\ u(x) &\to \sqrt{x} && \text{as } x \to \infty \ . \end{aligned}$$

For computational purposes, the domain of the ODE was truncated to $[0, 10]$, so that in fact the following modified system was solved:

$$\begin{aligned} u'' &= u^2 - u, && x \in [0, 10] \\ u(0) &= 0 \\ u(10) &= \sqrt{10} \end{aligned}$$

This problem is of particular interest as a test-case for BPNM, as it admits two distinct solutions, shown in Fig. 6.4a. Systems in which multiple solutions exist have been used as motivation for PNM before, in Chkrebtii et al. [2016] and Cockayne et al. [2016]. A probabilistic solver provides the notable advantage that all solutions can be naturally represented by a PNM, as a multimodal posterior, whereas such a representation is difficult to provide from a classical numerical procedure.

The spectrum plot in Fig. 6.4b shows the coefficients $\{u_i\}$ obtained when



(a) Solutions

(b) Spectra

(c) Likelihoods

Figure 6.4: (a) The solutions to the Painlevé ODE, generated using the deflation technique of Farrell et al. [2015]. The underlying ODE solver employed was MATLAB's `chebfun` package, so that the solutions are in the same polynomial basis as used to sample the posteriors in this section. (b) Coefficients $\{u_i\}$ corresponding to each solution. (c) Negative-log-likelihoods corresponding to each solution, as a function of the truncation level N.

each solution is estimated in the span of a set of normalised Chebyshev polynomials. This was computed using the disintegration technique of Farrell et al. [2015] and the `chebfun` package in MATLAB. Orthonormality of the Chebyshev polynomials means that the slower decay for the negative solution compared to the positive solution is equivalent to the negative solution having a larger L_2 norm. This explains the general preference that optimisation-based numerical solvers — and the results now presented — have for the positive solution.

As in the previous section, a prior was constructed using a truncated series expansion, so that

$$U^N = \sum_{i=1}^{N} \xi_i \gamma_i \phi_i$$

The truncation order was set to $N = 40$ terms. The ϕ_i were again taken to be normalised Chebyshev polynomials of the first kind as described in Appendix A.4, so that $\phi_i(x) = T_{i-1}(\frac{1}{2}(x-5))$. The ξ_i were taken to be either unit Gaussian or Cauchy, while the decay parameters were set to $\gamma_i = \alpha\beta^{-i}$, for $\alpha = 8$ and $\beta = 1.5$. This scaling was chosen based on the exponential convergence seen in Fig. 6.4b, with values of α and β chosen by inspection of the true spectra to ensure that the prior was adequately supported near the truth.

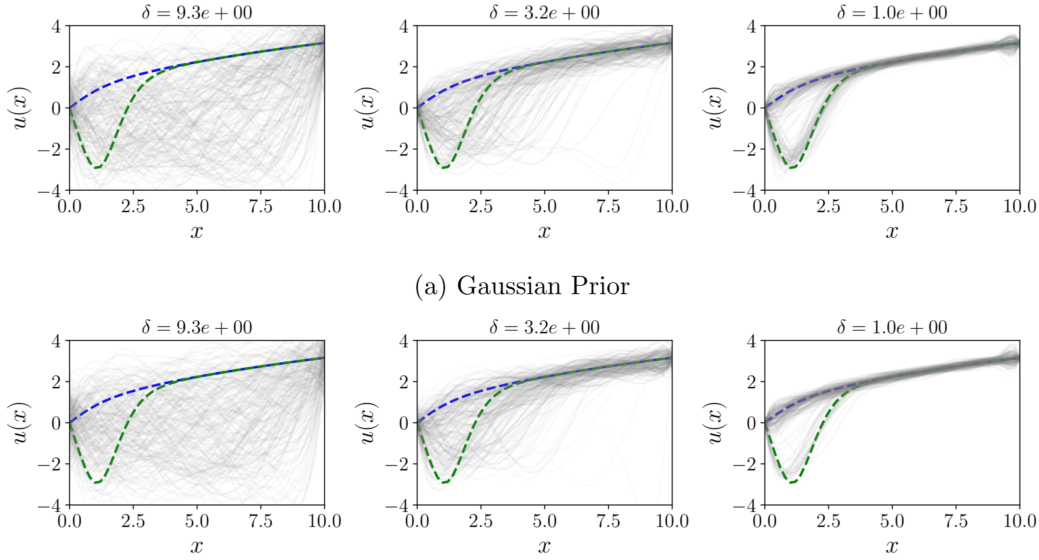
Pointwise information was again used. A regular grid of m spatial points, $\{x_1, \dots, x_m\}$, was placed in $(0, 10)$ and augmented with boundary information, so that the information operator and corresponding information $\boldsymbol{y} \in \mathbb{R}^n$, $n = m + 2$, are given by

$$A(u) = \begin{bmatrix} u''(x_1) - (u(x_1))^2 \\ \vdots \\ u''(x_m) - (u(x_m))^2 \\ u(0) \\ u(10) \end{bmatrix} \qquad \boldsymbol{y} = \begin{bmatrix} -x_1 \\ \vdots \\ -x_m \\ 0 \\ \sqrt{10} \end{bmatrix}$$

The posterior distribution was sampled by SMC (see Appendix A.3.2), based on 1600 temperatures equally spaced on a logarithmic scale between 10 and 10^{-4}. An ensemble of $P = 200$ particles was used, and the transition kernel at each temperature was provided by $10,000$ iterations of MALA at $n = 12$ and $n = 17$, and $40,000$ iterations at $n = 22$ (see Appendix A.3.1). With such a large number of temperatures and such a large number of iterations at each temperature, computation was expensive, highlighting the importance of further work to reduce the cost of BPNM outside of the conjugate framework.

Results for a selection of bandwidths δ, with $n = 17$, are shown in Fig. 6.5. Significantly more mass is placed around the positive than the negative node for the smallest value of δ, reflecting a preference for the solution with the smaller norm even in the probabilistic approach. Fig. 6.6 shows spectral posterior distributions, over the coefficients u_i, at $n = 17$ and $\delta = 1$. Note that these plots exhibit strong multimodality and a highly skewed correlation structure.

In Fig. 6.7 the amount of information n is varied, while δ is held fixed. For $n = 12$ we note that a new mode appears under a Gaussian prior, and posterior mass spans the range between the positive and negative solutions under both the Gaussian and Cauchy priors. However at $n = 22$ both posteriors settle on the positive solution to the ODE, again reflecting that the solution has lower L_2 norm. This may be due to the truncation of the prior; in Fig. 6.4c the log-likelihood of the negative solution increases at a slower rate than that of the positive solution as a function of the truncation level. This suggests that prior truncation might bias the BPNM in favour of one solution over the other, even when without truncation



(a) Gaussian Prior

(b) Cauchy Prior.

Figure 6.5: Posterior samples for the Painlevé system for $n = 17$. Blue and green dashed lines show the positive and negative solutions determined with `chebfun`. Grey lines are samples from an approximation to the posterior provided by numerical disintegration (bandwidth parameter δ).

neither solution is preferred.

6.4 Discussion

This concludes the discussion of the non-conjugate setting. We have established very general conditions for the existence and uniqueness of posterior distributions of BPNM, and have proposed a Monte-Carlo algorithm for approximately sampling from the posterior. There remain several open questions for this setting. Principal among these are well-posedness results for the posterior distributions. Theorem 6.2.5 establishes the approximation properties of numerical disintegration under an *assumption* of a level of smoothness of the disintegration with-respect-to the data, but when that level of smoothness should be expected has yet to be established and is a subject of ongoing research. Similarly, the computational cost of the proposed numerical disintegration algorithm is too high for it to be considered practically useful, and more computationally expedient methods must be developed if BPNM in the non-conjugate setting are ever to be considered a viable alternative to standard techniques in numerical analysis.

It should be noted that we do not insist that all PNM *must* be Bayesian;

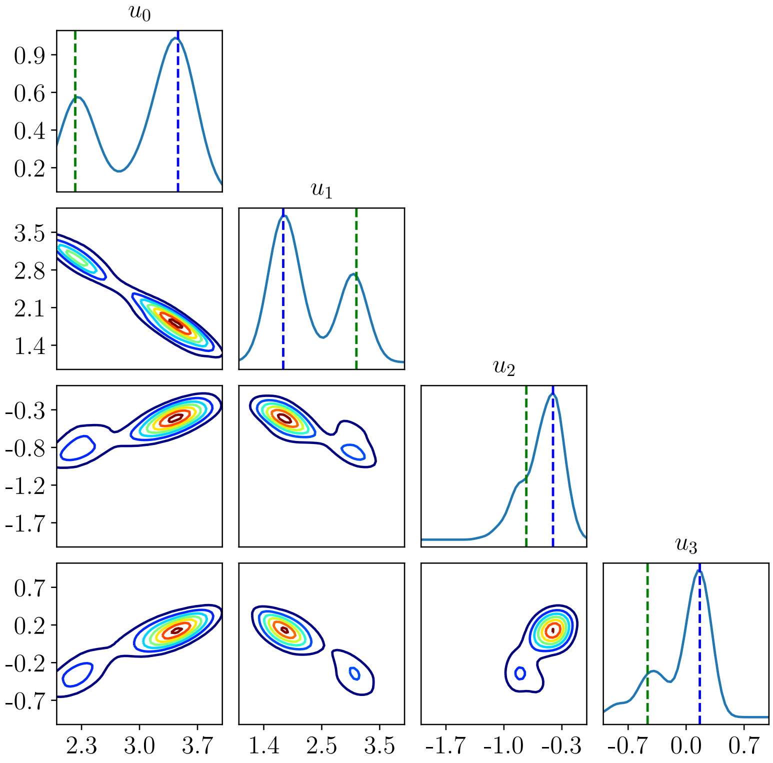


Figure 6.6: Posterior distributions for the first four coefficients obtained with numerical disintegration (bandwidth $\delta = 1$), at $n = 17$. Dashed lines show the coefficient values for the positive (blue) and negative (green) solutions determined with `chebfun`. The Gaussian prior was used.

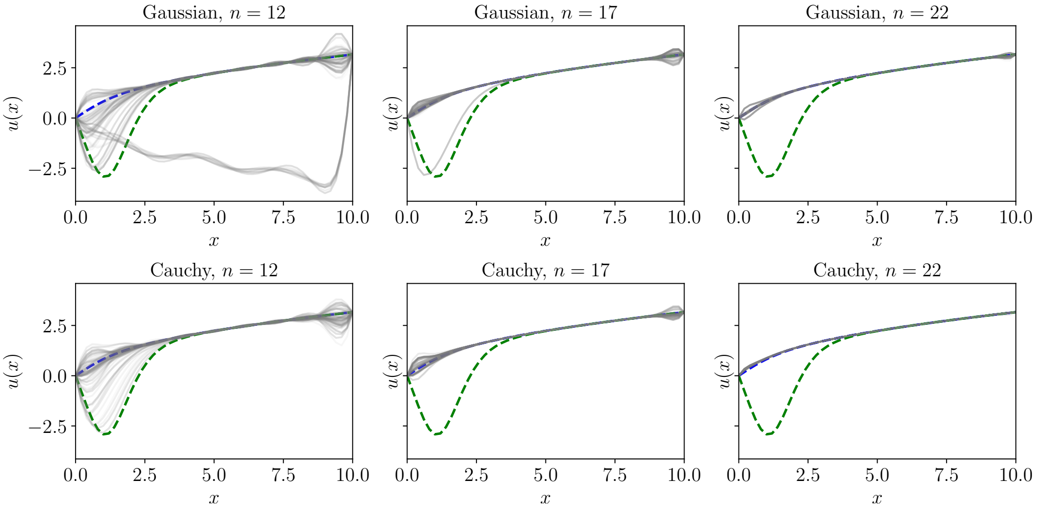


Figure 6.7: Convergence for the numerical disintegration scheme as n is increased. Top: Gaussian prior. Bottom: Cauchy prior. In all cases $\delta = 10^{-4}$.

BPNM are a subclass of PNM that produce UQ with a well-understood and widely studied meaning, but this does not dismiss other approaches as long as the meaning of the UQ produced is similarly understood. Nevertheless, a compelling argument for Bayesian PNM will be presented in the next chapter.

The last contribution of this thesis, in the next chapter, will be a study of the composition of BPNM. Specifically, we will discuss when composed BPNM yield an output with a strictly Bayesian interpretation, and present an application from industrial process monitoring.

Chapter 7

Pipelines of PNM

> “This leads to a somewhat disturbing argument of an endless chain of PNMs.”
>
> —*Anonymous reviewer of the paper ‘Bayesian probabilistic numerical methods’*

The final chapter in this thesis concerns the composition of PNM. For many of the most challenging models of physical processes, simulating from the model requires the application of multiple numerical methods, rather than a single one. For example, climate models involve large systems of coupled ODEs and PDEs for modelling the interlinked processes that determine climate evolution, such as sea temperature, air temperature and ice sheets. Heart models [Niederer et al., 2011] couple ODEs for modelling the electrical conductivity within cells to PDEs that govern the propagation of electrical pulses through the heart. In each case, otherwise independent numerical methods for solving each problem must be composed with each other to approximate a solution to the overall problem. Such a composition of numerical methods for estimating a quantity of interest is termed, in this this thesis, a *pipeline of computation.*

While a single numerical method can often be analysed theoretically, for example by deriving worst- or average-case error bounds, studying the propagation of numerical error through pipelines of computation is substantially more complicated. Nevertheless, accumulated discretisation error can have a significant impact on the output of a pipeline [Roy, 2010; Anderson, 2011; Babuška and Söderlind, 2018].

The prospect of using PNMs for this task has been repeatedly used as a motivation for the development of these methods (e.g. Hennig et al. [2015]; Conrad et al. [2017]; Cockayne et al. [2019a]). If the output of pipelines of PNM can be composed rigorously, then the posterior distribution output by the pipeline natively describes the discretisation error of the output. Furthermore, as suggested in Hennig et al. [2015], the pipeline allows for an *analysis of variance* to identify the dominant source of discretisation error in the pipeline, so that the discretisation used in that method can be refined. Some works have made limited attempts to incorporate PNM in specific pipelines (e.g. Chkrebtii et al. [2016]; Cockayne et al. [2019a,b], as well as Section 4.6.2 and Section 5.4.2). This chapter presents a detailed theoretical analysis of composition of PNMs.

The chapter proceeds as follows. In Section 7.1 we introduce the definition of a pipeline of computation, and present new theory related to to pipelines BPNM. In Section 7.2 we consider the use of a pipeline of BPNM in an application to the use of EIT in the monitoring of pieces of industrial machinery known as hydrocyclones.

7.1 Pipelines of PNM

In this section the basic definitions of pipelines of PNM are introduced. Note that the developments in this section are not specific to *Bayesian* PNM and hold for the any other PNM, including the trivial PNM constructed from classical numerical methods described in Section 3.2. Consider a sequence of n PNM, $M_1, \dots, M_n$ each identified by its information operator $A_i : \mathcal{X} \to \mathcal{Y}_i$ and its QoI $Q_i : \mathcal{Y}_i \to \mathcal{Q}_i$. The PNM are assumed to share a common state space without loss of generality. If necessary, this state space might be a tensor product space consisting of the individual state spaces required for each PNM. The definition of a pipeline of PNM is now introduced.

Definition 7.1.1 (Pipeline of Computation)**.** A *pipeline of computation* is described by a directed acyclic graph (DAG), with two kinds of nodes:

- *Method nodes*, represented by ■. These represent the PNM which the pipeline is composed of, and are labelled with integers, $i = 1, \dots, n$.
- *Information nodes*, represented by □. Root information nodes represent the information input to the pipeline, while child information nodes represent intermediate information as given by the QoI computed by the preceding nodes.

The graph is bipartite, so that edges appear only between method nodes and information nodes. Root and sink nodes of the graph must each be information nodes. The graph may have many root nodes, but we assume for simplicity that there is a single sink node representing a single QoI which it is the goal of the pipeline to compute, referred to as the *principal QoI*. Furthermore:

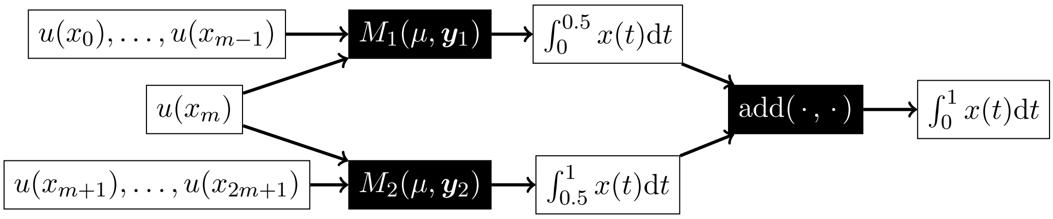

$u(x_0), \dots, u(x_{m-1})$
$u(x_m)$
$u(x_{m+1}), \dots, u(x_{2m+1})$
$M_1(\mu, \boldsymbol{y}_1)$
$M_2(\mu, \boldsymbol{y}_2)$
$\int_0^{0.5} x(t)\mathrm{d}t$
$\int_{0.5}^{1} x(t)\mathrm{d}t$
$\mathrm{add}(\cdot,\cdot)$
$\int_0^1 x(t)\mathrm{d}t$

Figure 7.1: Pipeline representation of Example 7.1.2.

- Information nodes are either root nodes, or have a single parent method node.
- Method nodes may have many parent information nodes but connect to a single output information node.

Note that the above definition allows for a method node to have multiple information nodes as its input. To accommodate this, we allow information operators A_i and information $\boldsymbol{y}_i$ to be decomposed into multiple components. Suppose that method node i has $m(i)$ information nodes as its parents. Its information node and information are then assumed to constructed as:

$$A_i(u) = \begin{bmatrix} A_i^1(u) \\ \vdots \\ A_i^{m(i)}(u) \end{bmatrix} \qquad \boldsymbol{y}_i = \begin{bmatrix} \boldsymbol{y}_i^1 \\ \vdots \\ \boldsymbol{y}_i^{m(i)} \end{bmatrix}.$$

Here $\boldsymbol{y}_i^j$ corresponds to the input supplied from input edge j of method node i, while A_i^j describes the information operator associated with that information. Thus, when the in-edges to a node are ordered, input edge j of method node i encodes the information $A_i^j(u) = \boldsymbol{y}_i^j$. This notational complexity will generally be suppressed except for where it is significant. To build intuition, a simple example of the formalisation of a sequence of computations into a pipeline is now presented.

Example 7.1.2 (Parallelised Integration)**.** In this example, a computational pipeline will be motivated by the task of parallelising the computation of an integral by splitting the domain. Let $\mathcal{X}$ be a (currently unspecified) set of functions $u : [0,1] \to \mathbb{R}$. The principal QoI that we wish to compute is then given by the map $u \mapsto \int_0^1 u(x)\mathrm{d}x$, and so $\mathcal{Q} = \mathbb{R}$. Information will be provided by pointwise evaluation of the true integrand $u^\dagger$, and for the purposes of this example we suppose that $u^\dagger$ is expensive to evaluate so that the goal is to produce a pipeline of computation that parallelises computation of the integral.

A simple parallelisation is given by splitting $[0,1]$ into $[0,0.5]$ and $[0.5,1]$, and summing the integral over each subinterval, i.e.:

$$\underbrace{\int_0^1 u^\dagger(x)\mathrm{d}x}_{\text{“Principal QoI”}} = \underbrace{\int_0^{0.5} u^\dagger(x)\mathrm{d}x}_{(1)} + \underbrace{\int_{0.5}^1 u^\dagger(x)\mathrm{d}x}_{(2)} \tag{7.1}$$

For simplicity, suppose that the points at which $u^\dagger$ is evaluated are given by $2m+1$ equally spaced points inside the domain; thus, integrals (1) and (2) will each be computed based on the information provided by $m+1$ equally spaced points inside $[0, 0.5]$ and $[0.5, 1]$ respectively, with the central location $x = 0.5$ used in both computations. Since each integral shares the central location, there are thus two information operators for each method:

$$\begin{aligned} A_1^1(u) &= [u(0), \ldots, u(x_{m-1}))]^\top & A_1^2 = A_2^1 = [u(x_m)] \\ A_2^2(u) &= [u(x_{m+1}), \ldots, u(1)]^\top . \end{aligned}$$

Similarly, two pieces of information are provided to each method:

$$\begin{aligned} \boldsymbol{y}_1^1 &= [u^\dagger(0), \ldots, u^\dagger(x_{m-1})]^\top & \boldsymbol{y}_1^2 = \boldsymbol{y}_2^1 = [u^\dagger(x_m)] \\ \boldsymbol{y}_2^2 &= [u^\dagger(x_{m+1}), \ldots, u^\dagger(1)]^\top & . \end{aligned}$$

The QoI operators are defined as

$$Q_1(u) = \int_0^{0.5} u(x)\mathrm{d}x \qquad Q_2(u) = \int_{0.5}^1 u(x)\mathrm{d}x$$

BPNM M_1, M_2 are then uniquely defined by the information operators and QoI operators. A BPNM for computing these integrals is described in Briol et al. [2019]. To sum the integrals an additional “dummy” method is required:

$$\mathrm{add}(y_1, y_2) = \delta(y_1 + y_2).$$

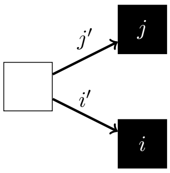
This dummy method satisfies the definition of a PNM with the state space $\mathcal{X} = \mathbb{R}$ and information space $\mathcal{Y} = \mathbb{R}^2$.

The DAG representing the pipeline is displayed in Fig. 7.1. Note that the two nodes associated with the computation of (1) and (2) in Eq. (7.1) each take input from two information nodes, as they share the evaluation $u^\dagger(0.5)$.

We now introduce the notion of a collection of PNM being *compatible* with a pipeline of computation. Informally, such a collection is compatible if the input and output spaces from the PNM match the graph's structure. This is a basic consistency condition which makes it possible for the methods represented by nodes in the graph.

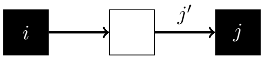
Definition 7.1.3 (Compatible)**.** A collection of PNM is said to be *compatible* with a pipeline P if:

(i) For nodes of the form



we have that $A_{i,i'} = A_{j,j'}$ and $\mathcal{Y}_{i,i'} = \mathcal{Y}_{j,j'}$.

(ii) For nodes of the form



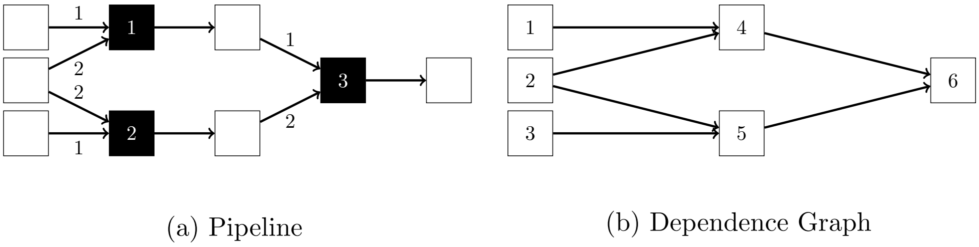
we have that $\mathcal{Q}_i = \mathcal{Y}_{j,j'}$.

Property (i) ensures that, where method nodes share information, the information "means the same" to each method node, in that the information spaces and information operators coincide. Property (ii) ensures that for a method node whose output is input to a descendent method node, the QoI space for the former method matches the information space of the latter.

Lastly, we introduce the concept of the *computation* associated with a pipeline P:

Definition 7.1.4 (Computation)**.** Consider PNMs $M_1, \ldots, M_n$ assumed to be compatible with a pipeline P. The *computation* $P[M_1, \ldots, M_n]$ associated with the pipeline and the PNM is itself a PNM, with an information space defined by the collection of root nodes of P, and output QoI space $\mathcal{Q}_n$. The information operator is implicit, and defined by the graph structure and the composite PNMs. The PNM associated with P is obtained by composing $M_1, \ldots, M_n$ in the manner described by P.

The material in this section has established definitions and conditions under which PNM can be composed into pipelines. The next section will focus on the properties of pipelines of *Bayesian* PNM.



(a) Pipeline

(b) Dependence Graph

Figure 7.2: Translation from a pipeline to a dependency graph. (a) An abstraction of the pipeline P from Example 7.1.2. (b) Dependence graph $G(P)$ corresponding to the pipeline P. The nodes are indexed with a topological ordering.

7.1.1 Bayesian Computational Pipelines

We now assume that the PNM $M_1, \dots, M_n$ are Bayesian PNM. Supposing that $M_1, \dots, M_n$ are compatible with P, it is natural to ask when the PNM $P[M_1, \dots, M_n]$ also has a Bayesian interpretation. The natural answer, according to Definition 3.1.5, is that $P[M_1, \dots, M_n](\mu, \boldsymbol{y})$ is Bayesian when it outputs an appropriate pushforward of $\mu^{\boldsymbol{y}}$. However, since both the QoI operator and information operator corresponding to the pipeline are complex and implicit, this is difficult to verify. Nevertheless, maintaining a Bayesian interpretation for $P[M_1, \dots, M_n]$ is essential if BPNM are to be meaningfully composed. Fortunately, it is possible to elicit conditions on the *graph* associated with the pipeline that guarantee a Bayesian output when the composite methods are Bayesian.

Definition 7.1.5 (Dependency Graph)**.** The *dependency graph* associated with a pipeline P, denoted $D(P)$, is the DAG obtained by deleting the method nodes from P, and for each deleted node connecting each of its parent information nodes to its child information node.

To provide intuition, the dependency graph for Example 7.1.2 is presented in Fig. 7.2.

The n' nodes in the dependency graph do not have a labelling in the present notation, and so will be associated with a topological ordering subject to the basic consistency requirements that the I root nodes should be labelled $1, \dots, I$, and the terminal node should be labelled n'. With the method nodes removed, each node of a dependency graph can be associated with a random variable Y_i, $i = 1, \dots, n'$ by a process that will now be described.

First, let $U \sim \mu$. Then, we assign Y_i based on whether i is the index of a root node. If this is the case, set $Y_i = A_{j,k}(U)$, where j, k are the indices of an information operator corresponding to the information represented by node i. Otherwise, if i is not a root node, set $Y_i = Q_j(U)$ for j the index of a QoI operator associated with node i. Thus, each the random variables Y_i is distributed according to the pushforward of the prior into the information space corresponding to the information node i. Note that while each Y_i may correspond to multiple $A_{j,k}$ and Q_j, the fact that the pipeline is assumed to be compatible ensures that the choice of $A_{j,k}$ or Q_j is arbitrary, and the random variables are uniquely defined.

The dependency graph allows us to analyse the dependency structure between the different pieces of information represented by the Y_i, and thus to establish a coherence condition for a prior and a pipeline of composed BPNM. We first introduce some notation. For a dependency graph $D(P)$, let $\pi(j) \subseteq \{1, \dots, j-1\}$ denote the parent nodes of node j, and let $\pi^{\complement}(j) = \{1, \dots, j-1\} \setminus \pi(j)$, i.e. the antecedent nodes of j in the topological ordering, which are not its parent nodes.

Definition 7.1.6 (Coherence)**.** Consider a pipeline P with dependency graph $D(P)$, and compatible BPNM $M_1, \dots, M_n$. Then a distribution $\mu \in \mathcal{P}_{\mathcal{X}}$ is said to be *coherent* for $P[M_1, \dots, M_n]$ if, for all $j = I+1, \dots, n'$, we have that

$$Y_j \perp\!\!\!\perp Y_{\pi^{\complement}(j)} \mid Y_{\pi(j)}.$$

Thus, a prior is said to be coherent for a pipeline of BPNM if, for each node in $D(P)$, the information represented by that node is independent of all other information in the pipeline *conditional* upon knowledge of the information represented by its parents. This rules out any possibility that contradictory states of knowledge can be represented in the pipeline. Note that this condition specifically does not depend upon the implicitly defined information operator and QoI operator of $P[M_1, \dots, M_n]$.

The following result relates coherency of a prior to a Bayesian interpretation of $P[M_1, \dots, M_n]$.

Theorem 7.1.7. *Let $M_1, \dots, M_n$ be BPNMs compatible with P, and let $\mu \in \mathcal{P}_{\mathcal{X}}$ be coherent with $P[M_1, \dots, M_n]$. Then $P[M_1, \dots, M_n]$ is a Bayesian PNM for the QoI Q_n under the prior μ.*

Proof. To simplify notation, for integer multi-indices $\alpha = (\alpha_1, \dots, \alpha_m)$ and $\beta = (\beta_1, \dots, \beta_n)$, where $1 \leq \alpha_i, \beta_i \leq J$, let μ_β^α be the law of $(Y_{\beta_1}, \dots, Y_{\beta_n}) | (Y_{\alpha_1}, \dots, Y_{\alpha_m})$. Furthermore, for integers i, j with $i < j$ we will use the notation $i : j = (i, i+1, \dots, j-1, j)$. Now, the output of the pipeline is $\mu_J^{1:I}$ and can be alternatively expressed as:

$$\mu_J^{1:I}(\mathrm{d}y_J) = \int_{\mathcal{Y}_{I+1:J-1}} \mu_{I+1:J}^{1:I}(\mathrm{d}y_{I+1:J})$$

where $\mathcal{Y}_\alpha = \mathcal{Y}_{\alpha_1} \times \cdots \times \mathcal{Y}_{\alpha_n}$ and $\mathrm{d}y_\alpha = \mathrm{d}y_{\alpha_1} \dots \mathrm{d}y_{\alpha_n}$. Now, by repeated application of the conditioning property from Definition 6.1.1 it follows that

$$\begin{aligned}
&\int_{\mathcal{Y}_{I+1:J-1}} \mu^{1:I}_{I+1:J}(\mathrm{d}y_{I+1:J}) \\
&\qquad = \int_{\mathcal{Y}_{J-1}} \cdots \int_{\mathcal{Y}_{I+1}} \mu^{1:I}_{I+1}(\mathrm{d}y_{I+1}) \mu^{1:I+1}_{I+2}(\mathrm{d}y_{I+2}) \cdots \mu^{1:J-1}_{J}(\mathrm{d}y_J) \\
&\qquad = \int_{\mathcal{Y}_{J-1}} \cdots \int_{\mathcal{Y}_{I+1}} \mu^{\pi(I+1)}_{I+1}(\mathrm{d}y_{I+1}) \mu^{\pi(I+2)}_{I+2}(\mathrm{d}y_{I+2}) \cdots \mu^{\pi(J)}_{J}(\mathrm{d}y_J)
\end{aligned}$$

where the last line is from application of the assumed coherency property. This last line we recognise as the identical to the computation associated with the pipeline of computation, described in Definition 7.1.4. □

This theorem establishes concrete conditions for a meaningful composition of BPNM. Note that this does not rule out meaningful composition of *non-Bayesian* PNM, but no other common frameworks for PNM have yet been developed. We now examine the impact of Theorem 7.1.7 on Example 7.1.2.

Example 7.1.2, continued. Consider a generic prior on the function space $\mathcal{X}$, represented by the random variable U. Following the ordering in Fig. 7.2b, the random variables $Y_1, \dots, Y_5$ are given by:

$$\begin{aligned}
Y_1 &= \{U(x_0), \dots, U(x_{m-1})\} & \qquad Y_4 &= \int_0^{0.5} U(x)\mathrm{d}x \\
Y_2 &= \{U(x_m)\} & Y_5 &= \int_{0.5}^{1} U(x)\mathrm{d}x \\
Y_3 &= \{U(x_{m+1}, \dots, U(x_{2m+1}))\}.
\end{aligned}$$

For μ to be a coherent distribution for the pipeline, the only conditional dependence relations which must be verified are:

$$\begin{aligned}
Y_4 &\perp\!\!\!\perp Y_3 \mid \{Y_1, Y_2\} \\
Y_5 &\perp\!\!\!\perp Y_1 \mid \{Y_2, Y_3\}
\end{aligned}$$

i.e.

$$\begin{aligned}
\int_0^{0.5} U(x)\mathrm{d}x \;\perp\!\!\!\perp\; \{U(t_i)\}_{i=m+1}^{2m+1} \mid \{U(t_i)\}_{i=0}^{m}, \\
\int_{0.5}^{1} U(x)\mathrm{d}x \;\perp\!\!\!\perp\; \{U(t_i)\}_{i=0}^{m-1} \mid \{U(t_i)\}_{i=m}^{2m+1}.
\end{aligned}$$

Whether this holds depends strongly on μ. If μ is the law of a Wiener process on $\mathcal{X}$ then it is straightforward to verify that the conditional independence conditions are satisfied, since for a U so distributed it is well known that $U(x^+) \perp\!\!\!\perp U(x^-)|U(x)$, whenever $x^+ > x$ and $x^- < x$. However it is straightforward to elicit other choices of μ that are not coherent for the pipeline. An example is a prior on functions whose first derivative is Wiener distributed. In this case, $U(x^+) \not\perp\!\!\!\perp U(x^-)|U(x)$ since knowledge of $U(x^-)$ conveys information about the *derivative* $U'(x)$ that is significant for the prediction of $U(x^+)$. This example thus shows that the additional coherence condition required to ensure that the output of the pipeline has a meaningful Bayesian interpretation is nontrivial.

Having established conditions under which a pipeline is Bayesian, we now turn to an example of a pipeline of PNM used in an industrial application.

7.2 Application to Industrial Process Monitoring

The final numerical results in this thesis concern application of pipelines of BPNM to the monitoring of pieces of industrial equipment called *hydrocyclones*.

Hydrocyclones are used in manufacturing and materials processing either to separate particulates in suspension, or to separate liquids of different densities [Bradley, 2013]. This is accomplished by injecting fluids into a large tank at high pressure, creating a vortex. Centrifugal force causes denser materials to be drawn to the outside of the tank, while lighter materials remain in the centre. When sufficiently separated, the two materials can be extracted from the tank and processed.

The contents of the hydrocyclone must be monitored to enable control of the input flow rate and ensure sufficient separation of the contents. This is a challenge since the tank walls are usually opaque, and internal sensors could disrupt the induced vortex. It has therefore been proposed to use EIT as an unintrusive means to monitor the tank's contents [Gutierrez et al., 2000]. This translates the problem considered in Section 4.6.2 into one with a temporal component, yielding a pipeline of computation.

In this section the simplified PEM will be used to solve the tomography problem. Practical experimental data was used, as provided in West et al. [2005]. To provide a brief description of the experimental setup, eight electrodes were placed on the surface of a cylindrical perspex tank which was filled with water. A mixing impeller was used to create a vortex in the tank and then removed, at which point data collection began. Data is denoted $\boldsymbol{y}_\tau$, with $\tau \in \{\tau_1, \ldots, \tau_N\}$ the collection times. Data was collected at regular intervals, however since information on the precise timings is unavailable it was assumed that $\tau_i = i$ for $i = 1, \ldots, N$. After a few seconds, concentrated potassium chloride solution was injected into the water, and it is the behaviour of this solution rotating within the tank over time that we seek to recover using EIT.

The stimulation pattern used was constructed by first choosing a reference electrode, for simplicity assumed to be the first electrode $i = 1$. A current was then passed between this electrode and each other electrode in turn, resulting in 7 stimulation patterns given by

$$C_{ij} = \begin{cases} A & j = 1 \\ -A & i = j - 1 \\ 0 & \text{otherwise} \end{cases}$$

where A is a fixed amplitude. Measurements obtained were the voltage differential over the two stimulated electrodes, so that

$$M_{ij} = \begin{cases} 1 & j = 1 \\ -1 & i = j - 1 \\ 0 & \text{otherwise} \end{cases} .$$

For modelling purposes, the tank was assumed to be a unit circle and the electrodes were assumed to be equally spaced points around its circumference.

Unlike in previous sections, the conductivity field is now taken to be time dependent, and the goal is recovery of the function $\kappa(\boldsymbol{x}, \tau)$ for $\boldsymbol{x} \in D$ and $\tau \in [0, T)$. The log-conductivity field was endowed with a separable centered Gaussian prior:

$$\begin{aligned} \theta(\boldsymbol{x}, \tau) &= \log \kappa(\boldsymbol{x}, \tau) \\ \theta(\boldsymbol{x}, \tau) &\sim \mathcal{GP}(0, k(\boldsymbol{x}, \tau; \boldsymbol{x}', \tau')) \\ k(\boldsymbol{x}, \tau; \boldsymbol{x}', \tau') &= \sigma k_\tau(\tau, \tau') k_{\boldsymbol{x}}(\boldsymbol{x}, \boldsymbol{x}'). \end{aligned}$$

The parameter σ was fixed to $\sigma = 10^{-3}$ based on the level of variation in the data. The composite covariance functions were set to $k_\tau = \min(\tau, \tau')$, resulting in a Brownian motion over time, while $k_{\boldsymbol{x}}$ was set to be squared-exponential as in Eq. (2.6), with unit amplitude and length scale $\ell = 0.3$.

To sample from the posterior over κ SMC was used, as described in Appendix A.3.2. This was motivated by the fact that the conditioning problem can be viewed as a filtering problem over τ. In brief, let μ_0 denote the prior. Each piece of

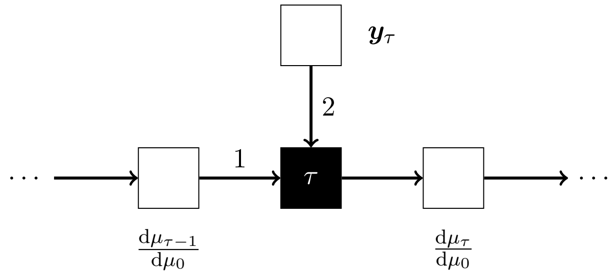

$\boldsymbol{y}_\tau$
2
1
τ
$\cdots$
$\cdots$
$\frac{\mathrm{d}\mu_{\tau-1}}{\mathrm{d}\mu_0}$
$\frac{\mathrm{d}\mu_\tau}{\mathrm{d}\mu_0}$

Figure 7.3: Pipeline representation of the computation in Section 7.2. The method node (black) represents the use of the PMM from Chapter 5 as the forward solver for evaluating the likelihood in a sequential Monte-Carlo procedure.

information $\boldsymbol{y}_t$ yields a posterior μ_τ, defined through

$$\frac{\mathrm{d}\mu_\tau}{\mathrm{d}\mu_{\tau-1}}(\theta) = \exp(-\Phi(\theta; \boldsymbol{y}_\tau)).$$

Here recall that evaluation of Φ requires solution of the PEM, as described in Eq. (2.12). To obtain this solution the PMM from Chapter 5 was applied, and the marginalisation procedure described in Section 5.4.2 was followed to incorporate the UQ for discretisation error from the forward solution into inferences. A pipeline of computation arises from the fact that this must be performed for each value of τ. The formal representation of the pipeline in the notation of Section 7.1 is depicted in Fig. 7.3. The Brownian motion form of k_τ ensures that the output of the pipeline is Bayesian for estimation of μ_T.

The SMC scheme was applied using an ensemble of $P = 100$ particles, based on a forward solver with $m_\mathcal{A} + m_\mathcal{B} = 119$ design points. Posterior means over $\kappa(\boldsymbol{x}, \tau)$ for $\tau = 1, \dots, 8$ are reported in Fig. 7.4. The high conductivity region containing the solution can clearly be seen rotating inside the domain through these frames.

The posterior variance is examined in Fig. 7.5. The figure depicts the integrated pointwise standard deviation over the domain for two distinct approaches to the problem: that just described, and one in which a symmetric collocation solver with the same discretisation resolution is used as a forward solver for the PDE involved in calculating Φ. Note that the left panel of this figure shows some structural periodicity, perhaps owing to certain values of τ happening to result in a conductivity field that is easier to infer. Nevertheless, in the figure on the right a clear upward trend in the variance can be seen, so that clearly discretisation uncertainty has been captured and propagated through the pipeline.

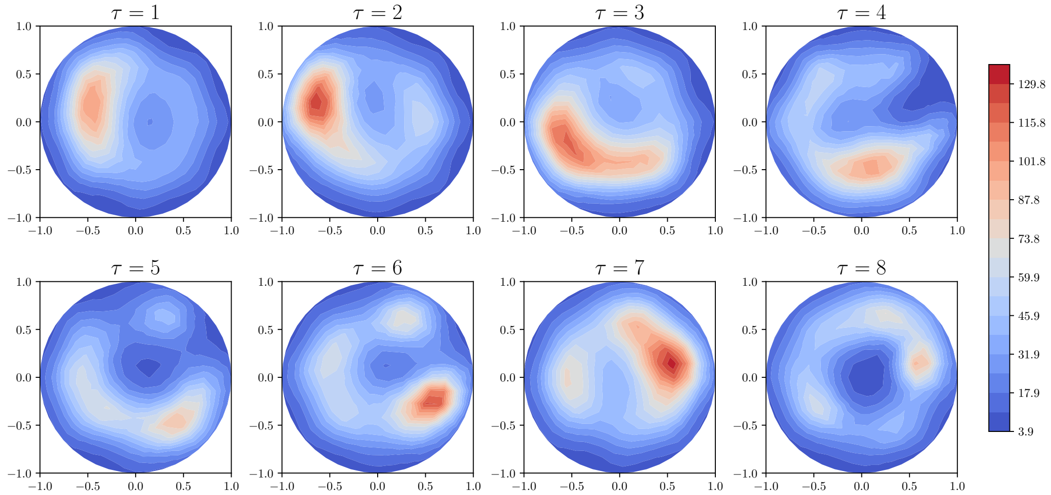


Figure 7.4: Posterior mean conductivity fields recovered in the hydrocyclone experiment, for $\tau = 1, \ldots, 8$.

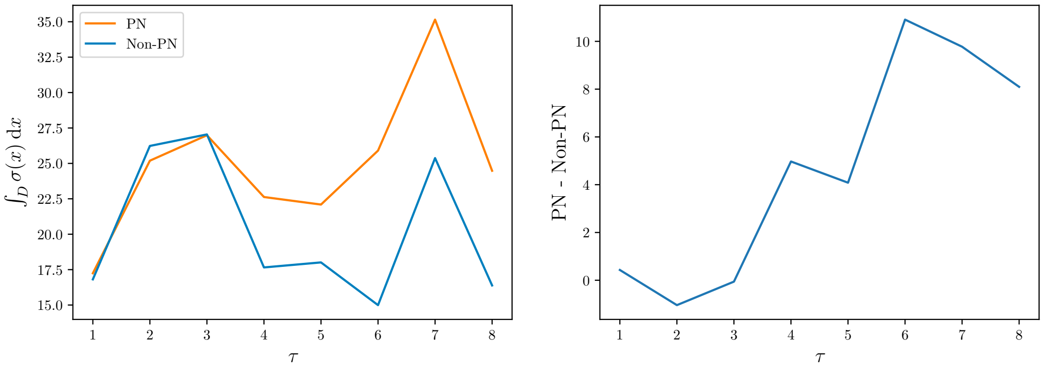


Figure 7.5: Left: Integrated standard deviation $\int_D \sigma(\boldsymbol{x};\tau)\mathrm{d}\boldsymbol{x}$ for the times $\tau = 1, \ldots, 8$. Both the probabilistic ("PN") and non-probabilistic ("Non-PN") approaches are depicted, and these are described in more detail in the text. Right: The difference between these two quantities.

7.3 Discussion

In this chapter we have established conditions under which the composition of BP-NMs yields an output with a strictly Bayesian interpretation, and studied application to the problem of monitoring hydrocyclones. While the result gives a condition for the pipeline that can be straightforwardly evaluated, the restriction imposed on a prior for the output of a pipeline to be Bayesian is surprisingly strong. This raises a natural question of whether the demand for a rigorously Bayesian output can be relaxed, and what the impact of such a relaxation might be.

The final chapter of this thesis will summarise the contributions, and discuss the future of this nascent area of research.

Chapter 8

Conclusion and Outlook

In this final chapter we will summarize the contributions of the thesis in Section 8.1, and discuss the outlook for PNM, and BPNM, in Section 8.2.

8.1 Contributions

The contributions of this thesis can be separated into two categories.

In Part I we provided necessary background and introduced the central definition of the thesis: that of a Bayesian PNM. In Part II, two novel Bayesian PNM were introduced. Chapter 4 introduced BayesCG, a PNM for solving finite-dimensional linear systems of equations. BayesCG is an iterative method in the sense of Saad [2003] that consists of updating a Gaussian belief over the solution to the system based on information obtained through a series of search directions. While the dependence of the search directions on the solution $\boldsymbol{x}^\dagger$ means that the method is not strictly Bayesian, this also results in a rate of convergence and a level of computational complexity that is competitive with classical iterative methods. The choice of prior was discussed in detail, and a prior inspired by preconditioners for the system of interest was presented and shown to be reasonably practical. However, in assessing the numerical performance of BayesCG it was shown that the uncertainty quantification provided tended to exhibit a tradeoff between a rapidly converging mean and well-calibrated uncertainty quantification. This was due to the dependence of the search directions on the true solution to the linear system, $\boldsymbol{x}^\dagger$, which induces nonlinearity in the information operator. Correct uncertainty quantification and a rapid rate of convergence could perhaps be achieved by applying the numerical disintegration algorithm discussed in Chapter 6 to estimate the posterior, though this would come at the expense of efficient computation. Nevertheless, BayesCG was tested on the EIT problem described in Section 2.4.2, and the results indicated that BayesCG might be useful in relaxing the computational effort required in solving this problem.

Chapter 5 continued the analysis of conjugate methods, introducing the PMM for the solution of PDEs that is defined on the infinite-dimensional solution space. The choice of information in this problem was more restricted, in that the difficulty of computing inner products led to a restriction to information based on evaluation functionals. On the other hand, this removed the issues with posterior UQ experienced in Chapter 4, as the search directions were then independent of $u^\dagger$ and so the PMM is truly Bayesian by the definition in Chapter 3. A detailed convergence analysis was again presented and the choice of prior was examined in detail. The PMM was once again applied to the EIT example, and here the application was more straightforward owing to there being no need for UQ calibration.

The contributions in Part III were more abstract and theoretical. In Chapter 6 we presented existence and uniqueness results, given by the disintegration theorem, for BPNM posteriors under mild conditions on the information operator and prior. These results provided no means by which to access the required posterior, however, and so we also introduced a novel numerical method, *numerical disintegration*, that can be used to approximately sample from the posterior of BPNM in this general setting. While methods based on ND are not strictly Bayesian PNM as they do not exactly sample from the required element of the disintegration, such methods can be thought of being *approximately* Bayesian, in the sense that they provide samples from a distribution that was proven to be close to the Bayesian posterior in a particular metric, under an assumption of Lipschitz continuity of the disintegration. We explored application of ND to a number of challenging but less applied numerical problems, and while the algorithm typically performs well it is associated with an extremely high numerical cost.

The last chapter, Chapter 7, studied the composition of BPNM. This is an important setting as modern inference problems frequently involve the composition of multiple numerical methods. The rigorous Bayesian interpretation of the posteriors from BPNM introduces some additional technical burden to ensure that composed BPNM carry this same interpretation. We introduced a mathematical framework in which composed BPNM can be represented with a directed acyclic graph, and presented conditions that guarantee that the output of the pipeline is Bayesian and can be easily verified from this graph. We then presented an application of composed BPNM for an applied inference problem arising from monitoring of hydrocyclones.

8.2 Outlook

The field of probabilistic numerics is still in its infancy, and while evidence of the usefulness of these methods has been presented in this thesis, there are still a great many challenges that need to be faced before widespread adoption of BPNM would be possible. Several of these challenges will be described here.

Computational Cost The greatest challenge for PNM, and especially for Bayesian PNM, remains their computational cost. Since PNM provide additional output over classical numerical methods, it is natural that their cost should be higher. However, if PNM are ever to be used in place of standard numerical methods in challenging contemporary settings in which discretisation error is a genuine concern, the increase in cost must be marginal. The only setting in this thesis in which a marginal cost increase was realised was in BayesCG, in which the cost was only a constant factor higher than in CG, and a comparable convergence speed was achieved when the prior was selected carefully.

For the PMM inversion of a dense Gramian matrix is required, and unless specialised inversion techniques are exploited this incurs a cost of $\mathcal{O}(n^3)$ where n is the amount of information obtained. However the inversion of similar matrices is a subject of active research (e.g. Snelson and Ghahramani [2006]; Schäfer et al. [2017]), and such techniques might make inference more practical. Another avenue that should be explored is exploitation of sparsity by using compactly supported covariance functions.

Outside of the conjugate setting, the cost of the numerical disintegration algorithm described in Chapter 6 is so high that this algorithm could never be considered practical. This would be even further compounded when BPNM must be composed in piplines as in Chapter 7. The ND algorithm should be seen only as a proof of concept demonstration that approximate sampling from the posterior distribution of BPNM can be accomplished, and we defer the development of specialised algorithms for the nonconjugate setting for future work. What direction those developments should take is an interesting question. One attractive approach involves approximating the information operator; one might imagine replacing the nonlinear operator A with a linearised operator $\tilde{A}$. If $\tilde{A}$ is "close" to A in some sense, one might expect that the posterior based on $\tilde{A}$ would be close to that based on A, in a similar way to how the relaxed posterior from ND is close to the element of the disintegration it approximates. This raises further questions about the well-posedness of BPNM, as in order to ensure that this is well-defined, smoothness of the disintegration in its operator is required. Thus, conditions for the well-posedness of BPNM is an important open theoretical question that needs attention to enable further development in the non-conjugate setting.

Prior Elicitation Another important challenge is the practical elicitation of priors for BPNMs. While this was addressed in some detail in Chapters 4 and 5, the known properties of the solution that could be encoded while still allowing for efficient inferences was very limited. A simple example is positivity; in Section 2.4.2 it is known that the solution $u(\boldsymbol{x})$ to EIT is everywhere positive, yet encoding this in a Gaussian prior is not possible. There thus remains a substantial gap between analytical knowledge of problems possessed by numerical analysts and what can be encoded efficiently into priors for BPNMs. Dismissing these concerns using subjectivist arguments places a barrier between those developing PNMs and the numerical analysts who must be convinced in order to promote widespread adoption, and so it is encumbant on us to either provide practical solutions, or convincing arguments for why such issues are not of concern.

The break from conjugacy and move towards approximate methods mentioned above would address this to some degree, though one motivation for the linearisation of the information operator discussed above is so that conjugacy inference with a Gaussian prior can still be employed. However, other efficient approximation techniques for the posterior such as discussed in Schillings and Schwab [2016] could be employed. Alternatively, to some degree a restriction such as positivity could be encoded into the *information operator* rather than in the prior, so that linearisation techniques could still be employed. A further possibility would be to employ variational techniques to find Gaussian priors that were *close* to the desired prior, and perform approximate Bayesian inference with this Gaussian prior again in a conjugate framework.

8.3 Closing Remarks

This thesis has defined and developed the theoretical basis of Bayesian probabilistic numerical methods, and developed methodology for two specific problem classes. Though there are still challenges to be addressed, BPNMs show great promise. It is my sincere hope that with more development these methods can become serious, practical numerical methods, and begin to be applied to real-world inference problems to begin quantifying uncertainty associated with discretisation error. In future, I believe that we will see a shift towards more approximate techniques with theoretical guarantees. This will allow the benefits of BPNM to be obtained, at least in an approximate sense, while also allowing for a reduction in the numerical cost of these algorithms.

Bibliography

Assyr Abdulle and Giacomo Garegnani. Random time step probabilistic methods for uncertainty quantification in chaotic and geometric numerical integration, 2018. preprint, arXiv:1801.01340.

N. L. Ackerman, C. E. Freer, and D. M. Roy. On computability and disintegration. *Math. Struct. Comp. Sci.*, 27(8):1287–1314, 2017. doi:10.1017/s0960129516000098.

M A Ajiz and A Jennings. A robust incomplete Choleski-conjugate gradient algorithm. *Int. J. Numer. Meth. Eng.*, 20(5):949–966, May 1984.

Grégoire Allaire and Sidi Mahmoud Kaber. *Numerical Linear Algebra*, volume 55 of *Texts in Applied Mathematics*. Springer New York, 2008.

Travis V. Anderson. Efficient, accurate, and non-Gaussian error propagation through nonlinear, closed-form, analytical system models. Master's thesis, Department of Mechanical Engineering, Brigham Young University, 2011.

Satya N. Atluri and Shengping Shen. The basis of meshless domain discretization: the meshless local Petrov–Galerkin (MLPG) method. *Adv. Comput. Math*, 23 (1-2):73–93, jul 2005. doi:10.1007/s10444-004-1813-9.

Ivo Babuška and Gustaf Söderlind. On roundoff error growth in elliptic problems. *ACM T. Math. Software*, 44(3):1–22, mar 2018. doi:10.1145/3134444.

Simon Bartels and Phillipp Hennig. Probabilistic approximate least-squares. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pages 676–684, Cadiz, Spain, 09–11 May 2016. PMLR. URL http://proceedings.mlr.press/v51/bartels16.html.

Simon Bartels, Jon Cockayne, Ilse C. F. Ipsen, and Philipp Hennig. Probabilistic linear solvers: A unifying view. *Stat. Comput.*, 2019. to appear.

T. Belytschko, Y. Y. Lu, and L. Gu. Element-free Galerkin methods. *Int. J. Numer. Meth. Eng.*, 37(2):229–256, jan 1994. doi:10.1002/nme.1620370205.

Michele Benzi. Preconditioning techniques for large linear systems: A survey. *J. Comp. Phys.*, 182(2):418–477, November 2002. doi:10.1006/jcph.2002.7176.

James O. Berger. *Statistical Decision Theory and Bayesian Analysis*. Springer Series in Statistics. Springer-Verlag, New York, second edition, 1985. doi:10.1007/978-1-4757-4286-2.

Alain Berlinet and Christine Thomas-Agnan. *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. Kluwer Academic Publishers, Boston, MA, 2004. doi:10.1007/978-1-4419-9096-9.

Dennis S. Bernstein. *Matrix mathematics*. Princeton University Press, Princeton, NJ, second edition, 2009. doi:10.1515/9781400833344. Theory, facts, and formulas.

Julian Besag and Peter J. Green. Spatial statistics and bayesian computation. 55 (1):25–37, September 1993. doi:10.1111/j.2517-6161.1993.tb01467.x.

Alexandros Beskos, Mark Girolami, Shiwei Lan, Patrick E. Farrell, and Andrew M. Stuart. Geometric MCMC for infinite-dimensional inverse problems. *J. Comp. Phys.*, 335:327–351, apr 2017. doi:10.1016/j.jcp.2016.12.041.

Ilias Bilionis. Probabilistic solvers for partial differential equations, 2016. preprint, arXiv:1607.03526.

Vladimir Igorevich Bogachev. *Gaussian Measures*, volume 62. American Mathematical Society Providence, 1998.

Douglas Bradley. *The Hydrocyclone*, volume 4 of *International Series of Monographs in Chemical Engineering*. Elsevier, 2013.

James H Bramble, Joseph E Pasciak, and Jinchao Xu. Parallel Multilevel Preconditioners. *Math. Comput.*, 55(191):1–22, July 1990.

Haïm Brezis and Felix Browder. Partial differential equations in the 20th century. *Adv. Math.*, 135(1):76–144, apr 1998. doi:10.1006/aima.1997.1713.

François-Xavier Briol, Chris J. Oates, Mark Girolami, Michael A. Osborne, and Dino Sejdinovic. Probabilistic integration: A role in statistical computation?, February 2019.

Steve Brooks, Andrew Gelman, Galin Jones, and Xiao-Li Meng. *Handbook of Markov Chain Monte Carlo*. Chapman and Hall/CRC, may 2011. doi:10.1201/b10905.

Alberto-P. Calderón. On an inverse boundary value problem. In *Seminar on Numerical Analysis and its Applications to Continuum Physics (Rio de Janeiro, 1980)*, pages 65–73. Soc. Brasil. Mat., Rio de Janeiro, 1980.

F. Cérou, P. Del Moral, T. Furon, and A. Guyader. Sequential Monte Carlo for rare event estimation. *Stat. Comput.*, 22(3):795–808, apr 2011. doi:10.1007/s11222-011-9231-6.

J. T. Chang and D. Pollard. Conditioning as disintegration. *Statist. Neerlandica*, 51(3):287–317, 1997. doi:10.1111/1467-9574.00056.

Kuo-Sheng Cheng, David Isaacson, J C Newell, and David G Gisser. Electrode models for electric current computed tomography. *IEEE T. Bio.-Med. Eng.*, 36 (9):918–924, 1989. doi:10.1109/10.35300.

Oksana A. Chkrebtii, David A. Campbell, Ben Calderhead, and Mark A. Girolami. Bayesian solution uncertainty quantification for differential equations. *Bayesian Anal.*, 11(4):1239–1267, 2016. doi:10.1214/16-BA1017.

Igor Cialenco, Gregory E. Fasshauer, and Qi Ye. Approximation of stochastic partial differential equations by a kernel-based collocation method. *Int. J. Comput. Math.*, 89(18):2543–2561, 2012. doi:10.1080/00207160.2012.688111.

Jon Cockayne, Chris Oates, T. J. Sullivan, and Mark Girolami. Probabilistic meshless methods for partial differential equations and Bayesian inverse problems, 2016. preprint, arXiv:1605.07811v1.

Jon Cockayne, Chris Oates, Tim Sullivan, and Mark Girolami. Probabilistic numerical methods for PDE-constrained bayesian inverse problems. In *Proceedings of the 36th International Workshop on Bayesianf Inference and Maximum Entropy Methods in Science and Engineering (MaxEnt 2016)*, volume 1853, page 060001. AIP Conference Proceedings, jun 2017. doi:10.1063/1.4985359.

Jon Cockayne, Chris Oates, Tim Sullivan, and Mark Girolami. Bayesian probabilistic numerical methods. *SIAM Rev.*, 2019a. to appear.

Jon Cockayne, Chris J. Oates, Ilse C.F. Ipsen, and Mark Girolami. A Bayesian conjugate gradient method (with discussion). *Bayesian Anal.*, 14(3):937–1012, 2019b. doi:10.1214/19-ba1145.

Patrick R. Conrad, Mark Girolami, Simo Särkkä, Andrew M. Stuart, and Konstantinos C. Zygalakis. Statistical analysis of differential equations: Introducing probability measures on numerical solutions. *Stat. Comput.*, 27(4):1065–1082, 2017. doi:10.1007/s11222-016-9671-0.

Simon L. Cotter, Gareth O. Roberts, Andrew M. Stuart, and David. White. MCMC methods for functions: Modifying old algorithms to make them faster. *Statistical Science*, 28(3):424–446, 2013. doi:10.1214/13-STS421.

Masoumeh Dashti and Andrew M. Stuart. The Bayesian approach to inverse problems. In *Handbook of Uncertainty Quantification*, pages 311–428. Springer International Publishing, 2017. doi:10.1007/978-3-319-12385-1_7.

Pierre Del Moral, Arnaud Doucet, and Ajay Jasra. Sequential Monte Carlo samplers. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 68(3):411–436, 2006. doi:10.1111/j.1467-9868.2006.00553.x.

Pierre Del Moral, Arnaud Doucet, and Ajay Jasra. An adaptive sequential Monte Carlo method for approximate Bayesian computation. *Stat. Comput.*, 22(5):1009–1020, 2012. doi:10.1007/s11222-011-9271-y.

Claude Dellacherie and Paul-André Meyer. *Probabilities and Potential*. North-Holland Publishing Co., Amsterdam-New York, 1978. doi:10.1016/s0304-0208(08)x7141-5.

Françoise Demengel and Gilbert Demengel. *Functional Spaces for the Theory of Elliptic Partial Differential Equations*. Springer London, 2012. doi:10.1007/978-1-4471-2807-6.

Persi Diaconis. Bayesian numerical analysis. In *Statistical Decision Theory and Related Topics IV*, volume 2, pages 163–175. Springer-Verlag, New York, 1988. doi:10.1007/978-1-4613-8768-8_20.

Persi Diaconis and Mehrdad Shahshahani. The subgroup algorithm for generating uniform random variables. *Probab. Eng. Inform. Sc.*, 1(1):15–32, January 1987. doi:10.1017/s0269964800000255.

R. M. Dudley. *Real Analysis and Probability*. Cambridge University Press, 2002. doi:10.1017/cbo9780511755347.

Matthew M. Dunlop and Andrew M. Stuart. The Bayesian formulation of EIT: Analysis and algorithms. *Inverse Probl. Imag.*, 10:1007–1036, 2016. doi:10.3934/ipi.2016030.

Matthew M. Dunlop, Marco A. Iglesias, and Andrew M. Stuart. Hierarchical Bayesian level set inversion. *Stat. Comput.*, 2016. doi:10.1007/s11222-016-9704-8.

David Duvenaud. *Automatic Model Construction with Gaussian Processes*. PhD thesis, Computational and Biological Learning Laboratory, University of Cambridge, 2014.

Lawrence C. Evans. *Partial Differential Equations*, volume 19 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, RI, second edition, 2010. doi:10.1090/gsm/019.

P. E. Farrell, Á. Birkisson, and S. W. Funke. Deflation techniques for finding distinct solutions of nonlinear partial differential equations. *SIAM J. Sci. Comput.*, 37(4): A2026–A2045, 2015. doi:10.1137/140984798.

Gregory E Fasshauer. Solving partial differential equations by collocation with radial basis functions. In Alain Le Méhauté, Christophe Rabut, and Larry L. Schumaker, editors, *Surface Fitting and Multiresolution Methods. Vol. 2 of the Proceedings of the 3rd International Conference on Curves and Surfaces held in Chamonix–Mont-Blanc, June 27–July 3, 1996*, pages 131–178. Vanderbilt University Press, Nashville, TN, 1997.

Gregory E. Fasshauer. Solving differential equations with radial basis functions: multilevel methods and smoothing. *Adv. Comput. Math.*, 11(2-3):139–159, 1999. doi:10.1023/A:1018919824891.

Gregory E Fasshauer. *Meshfree Approximation Methods with Matlab*. World Scientific, 2007. doi:10.1142/6437.

Gregory E. Fasshauer and Qi Ye. Reproducing kernels of generalized Sobolev spaces via a Green function approach with distributional operators. *Numer. Math.*, 119 (3):585–611, 2011. doi:10.1007/s00211-011-0391-2.

Andrew Gelman. Prior distributions for variance parameters in hierarchical models (comment on article by Browne and Draper). *Bayesian Anal.*, 1(3):515–533 (electronic), 2006. doi:10.1214/06-BA117A.

Andrew Gelman, John B Carlin, Hal S Stern, David B Dunson, Aki Vehtari, and Donald B Rubin. *Bayesian Data Analysis*. CRC press Boca Raton, FL, 3 edition, 2014.

C. J. Geyer. Markov chain Monte Carlo maximum likelihood. In *Proceedings of the 23rd Symposium on the Interface of Computing Science and Statistics*. Interface Foundation of North America, 1991.

R. A. Gingold and J. J. Monaghan. Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Mon. Not. R. Aston. Soc.*, 181(3):375–389, dec 1977. doi:10.1093/mnras/181.3.375.

Gene H. Golub and Charles F. Van Loan. *Matrix computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, Baltimore, MD, fourth edition, 2013.

Government Office for Science. Computational modelling: Technological futures. Technical report, 2018. URL https://www.gov.uk/government/publications/computational-modelling-blackett-review.

Tom Gunter, Michael A. Osborne, Roman Garnett, Philipp Hennig, and Stephen J. Roberts. Sampling for inference in probabilistic models with fast Bayesian quadrature. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pages 2789–2797, 2014.

J. A. Gutierrez, T. Dyakowski, M. S. Beck, and R. A. Williams. Using electrical impedance tomography for controlling hydrocyclone underflow discharge. *Powder Technol.*, 108(2):180–184, 2000. doi:10.1016/s0967-0661(97)00233-5.

Martin H. Gutknecht. Changing the norm in conjugate gradient type algorithms. *SIAM J. Numer. Anal.*, 30(1):40–56, feb 1993. doi:10.1137/0730003.

Jacques Hadamard. Sur les problèmes aux dérivées partielles et leur signification physique. *Princeton University Bulletin*, 13:49–52, 1903.

Philipp Hennig. Probabilistic interpretation of linear solvers. *SIAM J. Optim.*, 25 (1):234–260, 2015. doi:10.1137/140955501.

Philipp Hennig and Martin Kiefel. Quasi-Newton methods: A new direction. *J. Mach. Learn. Res.*, 14:843–865, 2013.

Philipp Hennig, Michael A. Osborne, and Mark Girolami. Probabilistic numerics and uncertainty in computations. *J. R. Stat. Soc. A Stat.*, 471(2179):20150142, 17, 2015. doi:10.1098/rspa.2015.0142.

M.R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *J. Res. Nat. Bur. Stand.*, 49(6):409, dec 1952. doi:10.6028/jres.049.044.

Nicholas J. Higham. *Accuracy and Stability of Numerical Algorithms*. Society for Industrial and Applied Mathematics, jan 2002. doi:10.1137/1.9780898718027.

David S Holder. *Electrical Impedance Tomography: Methods, History and Applications*. CRC Press, 2004. doi:10.1201/9781420034462.

M. Horstein. Sequential transmission using noiseless feedback. *IEEE Trans. Inf. Theory*, 9(3):136–143, jul 1963. doi:10.1109/tit.1963.1057832.

T. E. Hull and J. R. Swenson. Tests of probabilistic models for the propagation of roundoff errors. *Comm. ACM*, 9:108–113, 1966. doi:10.1145/365170.365212.

IPCC, editor. *Climate Change 2013 - The Physical Science Basis*. Cambridge University Press, 2009. doi:10.1017/cbo9781107415324.

D. Isaacson, J. L. Mueller, J. C. Newell, and S. Siltanen. Reconstructions of chest phantoms by the d-bar method for electrical impedance tomography. *IEEE Trans. Med. Imag.*, 23(7):821–828, July 2004. doi:10.1109/TMI.2004.827482.

Majnu John and Yihren Wu. Confidence intervals for finite difference solutions. *Commun. Stat.-Simul. C*, 47(7):2102–2118, jul 2017. doi:10.1080/03610918.2017.1335409.

Claes Johnson. *Numerical Solution of Partial Differential Equations by the Finite Element Method*. Cambridge University Press, 1988.

Joseph B. Kadane and Grzegorz W. Wasilkowski. *Average Case ϵ-Complexity in Computer Science: A Bayesian View*, pages 361–374. Elsevier, North-Holland, 1985.

Edward J. Kansa. Multiquadrics — A scattered data approximation scheme with applications to computational fluid-dynamics — II Solutions to parabolic, hyperbolic and elliptic partial differential equations. *Comput. Math. Appl.*, 19(8): 147–161, 1990. doi:10.1016/0898-1221(90)90271-K.

Kari Karhunen. Über lineare methoden in der wahrscheinlichkeitsrechnung. *Ann. Acad. Sci. Fenn.-M.*, (37):1–79, 1947.

Toni Karvonen and Simo Särkkä. Classical quadrature rules via Gaussian processes. In *27th IEEE International Workshop on Machine Learning for Signal Processing*, Sep 2017.

Toni Karvonen, Chris J. Oates, and Simo Särkkä. A Bayes-Sard cubature method. In *Advances in Neural Information Processing Systems 31*, pages 5882–5893. Curran Associates, Inc., 2018.

Hans Kersting and Philipp Hennig. Active uncertainty calibration in Bayesian ODE solvers. In Ihler Janzing, editor, *Uncertainty in Artificial Intelligence (UAI)*, volume 32, 2016. URL http://www.auai.org/uai2016/proceedings/papers/163.pdf.

A. N. Kolmogorov. *Foundations of Probability.* Ergebnisse Der Mathematik, 1933.

A. Kong, P. McCullagh, X.-L. Meng, D. Nicolae, and Z. Tan. A theory of statistical models for Monte Carlo integration. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 65 (3):585–618, 2003. doi:10.1111/1467-9868.00404.

Augustine Kong, Peter McCullagh, Xiao-Li Meng, and Dan L. Nicolae. Further explorations of likelihood theory for Monte Carlo integration. In *Advances in Statistical Modeling and Inference*, volume 3 of *Ser. Biostat.*, pages 563–592. World Sci. Publ., Hackensack, NJ, 2007. doi:10.1142/9789812708298_0028.

Johannes T.N. Krebs. Consistency and asymptotic normality of stochastic Euler schemes for ordinary differential equations. *Stat. Probabil. Lett.*, 125:1–8, jun 2017. doi:10.1016/j.spl.2017.01.016.

Jim Kuelbs, F. M. Larkin, and John A. Williamson. Weak probability distributions on reproducing kernel Hilbert spaces. *Rocky Mt. J. Math.*, 2(3):369–378, 1972. doi:10.1216/RMJ-1972-2-3-369.

F. M. Larkin. Estimation of a non-negative function. *BIT*, 9(1):30–52, 1969.

F. M. Larkin. Optimal approximation in Hilbert spaces with reproducing kernel functions. *Math. Comput.*, 24(112):911–921, 1970. doi:10.2307/2004625.

F. M. Larkin. Gaussian measure in Hilbert space and applications in numerical analysis. *Rocky Mt. J. Math.*, 2(3):379–421, 1972. doi:10.1216/RMJ-1972-2-3-379.

F. M. Larkin. Probabilistic error estimates in spline interpolation and quadrature. In *Information processing 74 (Proc. IFIP Congress, Stockholm, 1974)*, pages 605–609. North-Holland, Amsterdam, 1974.

F. M. Larkin. A modification of the secant rule derived from a maximum likelihood principle. *BIT*, 19(2):214–222, 1979a. doi:10.1007/BF01930851.

F.M Larkin. Probabilistic estimation of poles or zeros of functions. *J. Approx. Theory*, 27(4):355–371, dec 1979b. doi:10.1016/0021-9045(79)90124-2.

Kody Law, Andrew Stuart, and Konstantinos Zygalakis. *Data Assimilation.* Springer International Publishing, 2015. doi:10.1007/978-3-319-20325-6.

Reimar H. Leike and Torsten A. Enßlin. Towards information-optimal simulation of partial differential equations. *Phys. Rev. E*, 97(3), mar 2018. doi:10.1103/physreve.97.033314.

Giovanni Leoni. *A First Course in Sobolev Spaces.* American Mathematical Society, nov 2017. doi:10.1090/gsm/181.

H. C. Lie, A. M. Stuart, and T. J. Sullivan. Strong convergence rates of probabilistic integrators for ordinary differential equations. *Stat. Comput.*, 2019. to appear.

Jörg Liesen and Zdenek Strakos. *Krylov Subspace Methods.* Principles and Analysis. Oxford University Press, October 2012. doi:10.1093/acprof:oso/9780199655410.001.0001.

G Liu. *Mesh Free Methods.* CRC Press, jul 2002. doi:10.1201/9781420040586.

Michel Loève. *Probability Theory*, volume II. Springer-Verlag, fourth edition, 1978.

Maren Mahsereci and Philipp Hennig. Probabilistic line searches for stochastic optimization. In *Proceedings of Advances In Neural Information Processing Systems (NIPS)*, 2015.

J Mason and D Handscomb. *Chebyshev Polynomials.* Chapman and Hall/CRC, sep 2002. doi:10.1201/9781420036114.

J. Mercer. Functions of positive and negative type, and their connection with the theory of integral equations. *Philos. T. R. Soc. A*, 209(441-458):415–446, jan 1909. doi:10.1098/rsta.1909.0016.

Gérard Meurant. *The Lanczos and Conjugate Gradient Algorithms.* Society for Industrial and Applied Mathematics, Philadelphia, PA, Jan 2006.

Sean P. Meyn and Richard L. Tweedie. *Markov Chains and Stochastic Stability.* Springer London, 1993. doi:10.1007/978-1-4471-3267-7.

A. R. Mitchell. Numerical solutions of partial differential equations by the finite element method. *Commun. Appl. Numer. M.*, 4(4):598–598, jul 1988. doi:10.1002/cnm.1630040426.

Jonas Mockus. *Bayesian Approach to Global Optimization: Theory and Applications.* Springer Science & Business Media, 1989.

Sebastian Mosbach and Amanda G. Turner. A quantitative probabilistic investigation into the accumulation of rounding errors in numerical ODE solution. *Comput. Math. Appl.*, 57(7):1157–1167, 2009. doi:10.1016/j.camwa.2009.01.020.

Alfred Müller. Integral probability metrics and their generating classes of functions. *Adv. in Appl. Probab.*, 29(2):429–443, 1997. doi:10.2307/1428011.

Steven Niederer, Lawrence Mitchell, Nicolas Smith, and Gernot Plank. Simulating human cardiac electrophysiology on clinical time-scales. *Front. in Physiol.*, 2:14, 2011. doi:10.3389/fphys.2011.00014.

Harald Niederreiter. *Random Number Generation and Quasi-Monte Carlo Methods.* Society for Industrial and Applied Mathematics, jan 1992. doi:10.1137/1.9781611970081.

Otton Nikodym. Sur une généralisation des intégrales de M. J. Radon. *Fund. Math.*, 15:131–179, 1930. doi:10.4064/fm-15-1-131-179.

Chris J. Oates and Tim J. Sullivan. A modern retrospective on probabilistic numerics. *Stat. Comput.*, 2019. to appear.

Chris J. Oates, Jon Cockayne, Robert G. Aykroyd, and Mark Girolami. Bayesian probabilistic numerical methods in time-dependent state estimation for industrial hydrocyclone equipment. *J. Amer. Statist. Assoc.*, pages 1–27, February 2019. doi:10.1080/01621459.2019.1574583.

CJ Oates, S Niederer, A Lee, F-X Briol, and M Girolami. Probabilistic models for integration error in assessment of functional cardiac models, 2017. URL http://papers.nips.cc/paper/6616-probabilistic-models-for-integration-error-in-the-assessment-of-functional-cardiac-models.

A. O'Hagan. Bayes–Hermite quadrature. *J. Statist. Plann. Inference*, 29(3):245–260, 1991. doi:10.1016/0378-3758(91)90002-V.

Michael Osborne, Roman Garnett, Zoubin Ghahramani, David K. Duvenaud, Stephen J Roberts, and Carl E. Rasmussen. Active learning of model evidence using Bayesian quadrature. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 46–54. Curran Associates, Inc., 2012a.

URL http://papers.nips.cc/paper/4657-active-learning-of-model-evidence-using-bayesian-quadrature.

Michael A Osborne, Roman Garnett, Stephen J Roberts, Christopher Hart, Suzanne Aigrain, Neale Gibson, and Suzanne Aigrain. Bayesian quadrature for ratios. In *Proceedings of Artificial Intelligence and Statistics (AISTATS)*, 2012b.

Houman Owhadi. Bayesian numerical homogenization. *Multiscale Model. Simul.*, 13(3):812–828, 2015. doi:10.1137/140974596.

Houman Owhadi. Multigrid with rough coefficients and multiresolution operator decomposition from hierarchical information games. *SIAM Rev.*, 59(1):99–149, 2017. doi:10.1137/15M1013894.

Houman Owhadi, Clint Scovel, and T. J. Sullivan. On the brittleness of Bayesian inference. *SIAM Rev.*, 57(4):566–582, 2015. doi:10.1137/130938633.

Albert Parker and Colin Fox. Sampling Gaussian distributions in Krylov spaces with conjugate gradients. *SIAM J. Sci. Compu.*, 34(3):B312–B334, Jan 2012. doi:10.1137/110831404.

Henri Poincaré. *Calcul des Probabilités*. Gauthier-Villars, 1912.

T. Rabczuk and T. Belytschko. A three-dimensional large deformation meshfree method for arbitrary evolving cracks. *Comput. Method. Appl. M.*, 196(29-30): 2777–2799, may 2007. doi:10.1016/j.cma.2006.06.020.

Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Inferring solutions of differential equations using noisy multi-fidelity data. *J. Comp. Phys.*, 335: 736–746, apr 2017. doi:10.1016/j.jcp.2017.01.060.

Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Numerical gaussian processes for time-dependent and nonlinear partial differential equations. *SIAM J. Sci. Comput.*, 40(1):A172–A198, jan 2018. doi:10.1137/17m1120762.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, 2006.

Anne Reinarz, Tim Dodwell, Tim Fletcher, Linus Seelinger, Richard Butler, and Robert Scheichl. Dune-composites – a new framework for high-performance finite element modelling of laminates. *Compos. Struct.*, 184:269–278, Jan 2018. doi:10.1016/j.compstruct.2017.09.104.

Klaus Ritter. *Average-Case Analysis of Numerical Problems*, volume 1733 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin, 2000. doi:10.1007/BFb0103934.

Gareth O. Roberts and Jeffrey S. Rosenthal. General state space markov chains and MCMC algorithms. *Probab. Surv.*, 1(0):20–71, 2004. doi:10.1214/154957804100000024.

Gareth O. Roberts and Richard L. Tweedie. Exponential convergence of langevin distributions and their discrete approximations. *Bernoulli*, 2(4):341, December 1996. doi:10.2307/3318418.

Christopher Roy. Review of discretization error estimators in scientific computing. In *Proceedings of AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, 2010.

Yousef Saad. ILUT: A dual threshold incomplete LU factorization. *Numer. Linear Algebr.*, 1(4):387–402, July 1994.

Yousef Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, second edition, 2003.

Simo Särkkä. Linear operators and stochastic partial differential equations in Gaussian process regression. In Timo Honkela, Włodzisław Duch, Mark Girolami, and Samuel Kaski, editors, *Artificial Neural Networks and Machine Learning – ICANN 2011: 21st International Conference on Artificial Neural Networks, Espoo, Finland, June 14-17, 2011, Proceedings, Part II*, pages 151–158. Springer, Berlin, Heidelberg, 2011. doi:10.1007/978-3-642-21738-8_20.

Florian Schäfer, T. J. Sullivan, and Houman Owhadi. Compression, inversion, and approximate PCA of dense kernel matrices at near-linear computational complexity, 2017. preprint, arXiv:1706.02205.

Claudia Schillings and Christoph Schwab. Scaling limits in computational bayesian inversion. *ESAIM-Math. Model. Num.*, 50(6):1825–1856, October 2016. doi:10.1051/m2an/2016005.

Michael Schober, David K Duvenaud, and Philipp Hennig. Probabilistic ODE solvers with Runge–Kutta means. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 739–747. Curran Associates, Inc., 2014. URL http://papers.nips.cc/paper/5451-probabilistic-ode-solvers-with-runge-kutta-means.

Michael Schober, Simo Särkkä, and Philipp Hennig. A probabilistic model for the numerical solution of initial value problems. *Stat. Comput.*, jan 2018. doi:10.1007/s11222-017-9798-7.

Jonathan Richard Shewchuk. An introduction to the conjugate gradient method without the agonizing pain. Technical report, 1994.

John Skilling. Bayesian solution of ordinary differential equations. In C. Ray Smith, Gary J. Erickson, and Paul O. Neudorfer, editors, *Maximum Entropy and Bayesian Methods: Seattle, 1991*, pages 23–37. Springer, Dordrecht, 1992. doi:10.1007/978-94-017-2219-3_2.

Edward Snelson and Zoubin Ghahramani. Sparse Gaussian processes using pseudo-inputs. In *Advances in neural information processing systems*, pages 1257–1264, 2006.

Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. Practical Bayesian optimization of machine learning algorithms. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 2951–2959. Curran Associates, Inc., 2012. URL https://papers.nips.cc/paper/4522-practical-bayesian-optimization-of-machine-learning-algorithms.

Sergei Sobolev. Sur un théorème d'analyse fonctionnelle. *Rec. Math. [Mat. Sbornik] N.S.*, pages 471–497, 1938.

Bharath K. Sriperumbudur, Kenji Fukumizu, Arthur Gretton, Bernhard Schölkopf, and Gert R. G. Lanckriet. On the empirical estimation of integral probability metrics. *Electronic Journal of Statistics*, 6(0):1550–1599, 2012. doi:10.1214/12-ejs722.

Ingo Steinwart and Clint Scovel. Mercer's theorem on general domains: On the interaction between measures, kernels, and RKHSs. *Constr. Approx.*, 35(3):363–417, 2012. doi:10.1007/s00365-012-9153-3.

Andrew M. Stuart. Inverse problems: a Bayesian perspective. *Acta Numer.*, 19: 451–559, 2010. doi:10.1017/S0962492910000061.

T. J. Sullivan. *Introduction to Uncertainty Quantification*, volume 63 of *Texts in Applied Mathematics*. Springer, Cham, 2015. doi:10.1007/978-3-319-23395-6.

T. J. Sullivan. Well-posed Bayesian inverse problems and heavy-tailed stable quasi-Banach space priors. *Inverse Probl. Imaging*, 11(5):857–874, 2017. doi:10.3934/ipi.2017040.

V. T. and G. D. Smith. Numerical solution of partial differential equations, finite difference methods. *Math. Comput.*, 48(178):834, apr 1987. doi:10.2307/2007849.

Zhiqiang Tan. On a likelihood approach for Monte Carlo integration. *J. Amer. Statist. Assoc.*, 99(468):1027–1036, 2004. doi:10.1198/016214504000001664.

Onur Teymur, Kostas Zygalakis, and Ben Calderhead. Probabilistic linear multistep methods. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 4321–4328. Curran Associates, Inc., 2016. URL http://papers.nips.cc/paper/6356-probabilistic-linear-multistep-methods.

Tue Tjur. *Probability based on Radon measures.* Wiley Series in Probability and Statistics. John Wiley & Sons, Inc., Hoboken, NJ, 1980.

Aimo Törn and Antanas Žilinskas. *Global Optimization*, volume 350 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, 1989. doi:10.1007/3-540-50871-6.

Rolf Waeber, Peter I. Frazier, and Shane G. Henderson. Bisection search with noisy responses. *SIAM J. Control Optim.*, 51(3):2261–2279, 2013. doi:10.1137/120861898.

Herbert F. Wang and Mary P. Anderson. *Introduction to Groundwater Modeling: Finite Difference and Finite Element Methods*. Academic Press Inc., 1982.

Junyang Wang, Jon Cockayne, and Chris Oates. On the Bayesian solution of differential equations, 2018. preprint, arXiv:1805.07109.

Holger Wendland. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Adv. Comput. Math.*, 4(4):389–396, 1995. doi:10.1007/BF02123482.

Robert M. West, Sha Meng, Robert G. Aykroyd, and Richard A. Williams. Spatial-temporal modeling for electrical impedance imaging of a mixing process. *Rev. Sci. Instrum.*, 76(7):073703, 2005. doi:10.1063/1.1947882.

Christopher K Wikle, Ralph F Milliff, Doug Nychka, and L Mark Berliner. Spatiotemporal hierarchical bayesian modeling tropical ocean surface winds. *J. Am. Stat. Assoc.*, 96(454):382–397, Jun 2001. doi:10.1198/016214501753168109.

Xiaoyue Xi, François-Xavier Briol, and Mark Girolami. Bayesian quadrature for multiple related integrals. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018.

O.C. Zienkiewicz, R.L. Taylor, and J.Z. Zhu. *The Finite Element Method: its Basis and Fundamentals*. Elsevier, 2013. doi:10.1016/b978-1-85617-633-0.00001-0.

V. M. Zolotarev. Probability metrics. *Theory Probab. Appl.*, 28(2):278–302, January 1984. doi:10.1137/1128025.

Appendices

Appendix A

Additional Background

A.1 Analysis

We will begin by introducing a number of concepts from analysis and functional analysis which will be useful throughout the thesis. All definitions are set in the context of an underlying vector space $\mathcal{V}$ with $\mathbb{R}$ as the underlying scalar field, though some definitions can be made more general.

Definition A.1.1 (Metric, Pseudometric)**.** A *metric* on $\mathcal{V}$ is a function $d : \mathcal{V} \times \mathcal{V} \to \mathbb{R}^+$ with the properties that, for any $v, v', v'' \in \mathcal{V}$:

1. $d(v, v') \geq 0$.
2. $d(v, v') = 0 \iff v = v'$
3. $d(v, v') = d(v', v)$
4. $d(v, v'') \leq d(v, v') + d(v', v'')$ (*triangle inequality*)

The pair $(\mathcal{V}, d)$ is known as a *metric space.* A function $d : \mathcal{V} \times \mathcal{V} \to \mathbb{R}^+$ which satisfies all above properties except 2 is known as a *pseudometric.*

Definition A.1.2 (Norm)**.** A *norm* is a function $\|\cdot\|_\mathcal{V} : \mathcal{V} \to \mathbb{R}^+$ which has the properties

1. $\|v\|_\mathcal{V} \geq 0$ for all $v \in \mathcal{V}$ such that $v \neq 0$.
2. $\|\alpha v\|_\mathcal{V} = \alpha \|v\|_\mathcal{V}$ for each $v \in \mathcal{V}$, $\alpha \in \mathbb{R}$
3. $\|v + v'\|_\mathcal{V} \leq \|v\|_\mathcal{V} + \|v'\|_\mathcal{V}$ for each $v, v' \in \mathcal{V}$ (*triangle inequality*)

The pair $(\mathcal{V}, \|\cdot\|_\mathcal{V})$ is known as a *normed vector space.*

Note that in a normed vector space, the norm also induces a metric $d(v, v') = \|v - v'\|_{\mathcal{V}}$.

Definition A.1.3 (Banach Space)**.** A *Banach space* $(\mathcal{B}, \|\cdot\|_{\mathcal{B}})$ is a normed vector space which is complete with respect to its norm, meaning that each Cauchy sequence of points in $\mathcal{B}$ converges to a point in $\mathcal{B}$.

Definition A.1.4 (Inner Product)**.** An *inner product* on a vector space $\mathcal{V}$ is a function $\langle\,\cdot\,,\,\cdot\,\rangle_{\mathcal{V}} : \mathcal{V} \times \mathcal{V} \to \mathbb{R}$ with the properties, for each $v, v', v'' \in \mathcal{V}$:

1. $\langle v, v'\rangle_{\mathcal{V}} = \langle v', v\rangle_{\mathcal{V}}$
2. $\langle \alpha v, v'\rangle_{\mathcal{V}} = \alpha\langle v, v'\rangle_{\mathcal{V}}$
3. $\langle v + v, v''\rangle_{\mathcal{V}} = \langle v, v''\rangle_{\mathcal{V}} + \langle v', v''\rangle_{\mathcal{V}}$
4. $\langle v, v\rangle \geq 0$ with $\langle v, v\rangle = 0 \iff v = 0$.

The inner product also induces a norm through $\|v\|_{\mathcal{V}} = \sqrt{\langle v, v\rangle_{\mathcal{V}}}$.

Definition A.1.5 (Hilbert Space)**.** A *Hilbert space* $(\mathcal{H}, \langle\,\cdot\,,\,\cdot\,\rangle_{\mathcal{H}})$ is a vector space equipped with an inner product, which is complete with respect to the norm induced by its inner product.

Note that Definition A.1.5 implies that any Hilbert space is also a Banach space.

Definition A.1.6 (Embedding)**.** Let $\mathcal{X}$, $\mathcal{Y}$ be vector spaces. We say that $\mathcal{X}$ is *embedded* in $\mathcal{Y}$ if $\mathcal{X} \subset \mathcal{Y}$ and the map $i : \mathcal{X} \to \mathcal{Y}$ given by $i(x) = x$ is continuous. Equivalently, since i is linear and bounded linear operators are continuous, $\mathcal{X}$ is embedded in $\mathcal{Y}$ if there is a constant $C > 0$ such that

$$\|x\|_{\mathcal{Y}} \leq C\|x\|_{\mathcal{X}}$$

for all $x \in \mathcal{X}$.

Definition A.1.7 (Norm-Equivalence)**.** We say that two spaces $\mathcal{X}$ and $\mathcal{Y}$ are *norm-equivalent* if each is continuously embedded in the other.

A.1.1 Some Useful Spaces

In this section some examples of the spaces introduced in the previous section will be presented. The presented examples are of particular relevance for this thesis.

Definition A.1.8 (ℓ^p space)**.** Let $\mathbb{R}^\infty$ denote the set of all sequences indexed by $\mathbb{N}$ with elements in $\mathbb{R}$; that is, $s \in \mathbb{R}^\infty$ is a sequence $(s_1, s_2, \dots)$ with $s_i \in \mathbb{R}$ for each $i \in \mathbb{N}$.

For each $p \in \mathbb{R}$, $p \geq 1$, define the *p-norm* on $\mathbb{R}^\infty$ by:

$$\|s\|_p = \left(\sum_{i=1}^{\infty} |s_i|^p \right)^{\frac{1}{p}}$$

and for $p = \infty$, define the ∞*-norm* or *sup-norm* by

$$\|s\|_\infty = \sup_{i \in \mathbb{N}} |s_i|$$

The ℓ_p *space* is then defined, for $p \in \mathbb{R} \cup \{\infty\}$, $p \geq 1$, by

$$\ell^p = \{s \in \mathbb{R}^\infty : \|s\|_p < \infty\}.$$

For each $p \geq 1$, ℓ^p is a Banach space. When $p = 2$, ℓ^p is a Hilbert space with inner product:

$$\langle s, s' \rangle_2 = \sum_{i=1}^{\infty} s_i s'_i$$

A.2 Probability

Definition A.2.1 (Measureable Space)**.** For a set $\mathcal{X}$ and a σ-algebra $\mathcal{B}_\mathcal{X}$, the pair $(\mathcal{X}, \mathcal{B}_\mathcal{X})$ is called a *measurable space.*

Definition A.2.2 (Measure, Distribution)**.** A *measure* on a measurable space $(\mathcal{X}, \mathcal{B}_\mathcal{X})$ is a function $\mu : \mathcal{B}_\mathcal{X} \to \mathbb{R}^+$ with the properties:

1. $\mu(X) \geq 0$ for all $X \in \mathcal{B}_\mathcal{X}$
2. $\mu(\emptyset) = 0$
3. For each collection $\{X_i\}_{i \in \mathbb{N}}$ of pairwise disjoint sets $X_i \in \mathcal{B}_\mathcal{X}$, we have

$$\mu\left(\bigcup_{i=1}^{\infty} X_i \right) = \sum_{i=1}^{\infty} \mu(X_i).$$

A measure with the additional property that $\mu(\mathcal{X}) = 1$ is known as a *probability measure* or *distribution.*

Definition A.2.3 (Pushforward Measure)**.** Let $(\mathcal{X}, \mathcal{B}_{\mathcal{X}})$ and $(\mathcal{Y}, \mathcal{B}_{\mathcal{Y}})$ be measurable spaces and let $f : \mathcal{X} \to \mathcal{Y}$ be a measurable function. Let μ be a measure on $\mathcal{X}$. Then the *pushforward* of the measure μ through f, denoted $f_{\#}\mu$, is a measure on $\mathcal{Y}$ defined by

$$[f_{\#}\mu](A) = \mu(f^{-1}(A))$$

for each $A \in \mathcal{B}_{\mathcal{Y}}$, where $f^{-1}(A)$ is to be understood as the preimage of A; $f^{-1}(A) = \{f^{-1}(a) : a \in A\}$.

Definition A.2.4 ($L^p(\mathcal{X}, \mu)$ space)**.** For a measurable space $(\mathcal{X}, \mathcal{B}_{\mathcal{X}})$ and measure μ on $(\mathcal{X}, \mathcal{B}_{\mathcal{X}})$, consider the set of all measurable functions $f : \mathcal{X} \to \mathbb{R}$. For each $p \in [1, \infty)$, $p \geq 1$ define the (p, μ)*-norm* on this set by

$$\|f\|_{p,\mu} = \left(\int_{\mathcal{X}} |f(x)|^p \mu(\mathrm{d}x) \right)^{\frac{1}{p}}$$

The $L^p(\mathcal{X}, \mu)$*-space* is the set of all measurable functions with finite (p, μ)-norm.

Again, for all $p \in \mathbb{R} \cup \{\infty\}$, $p \geq 1$, $L^p(\mathcal{X}, \mu)$ is a Banach space. When $p = 2$, L^p is a Hilbert space with inner product:

$$\langle f, f' \rangle_{2,\mu} = \int_{\mathcal{X}} f(x) f'(x) \mu(\mathrm{d}x)$$

When $\mathcal{X} \subseteq \mathbb{R}^d$ for some d and μ is the Lebesgue measure we will simply call the (p, μ)-norm the p-norm, and the set $L^p(\mathcal{X}, \mu)$ will be called $L^p(\mathcal{X})$. When $p = \infty$, the norm $\| \cdot \|_{\infty,\mu}$ is given by

$$\|f\|_{\infty,\mu} = \inf(\{M \in \mathbb{R}^+ : \mu(\{x \in \mathcal{X} : |f(x)| \geq M\}) = 0\})$$

and the space $L^\infty(\mathcal{X}, \mu)$ is defined analogously.

Definition A.2.5 (Absolute Continuity, Singularity of Measures)**.** Consider two measures μ, ν on the measureable space $(\mathcal{X}, \mathcal{B}_{\mathcal{X}})$. We say that μ is *absolutely continuous* with respect to ν if, for each $A \in \mathcal{B}_{\mathcal{X}}$ it holds that

$$\nu(A) = 0 \implies \mu(A) = 0.$$

In this case we write $\mu \ll \nu$. If $\mu \not\ll \nu$ we say that μ is *singular* with respect to ν.

If $\mu \ll \nu$ and $\nu \ll \mu$ then we say μ and ν are *equivalent*. Similarly if $\mu \not\ll \nu$ and $\nu \not\ll \mu$ we say μ and ν are *mutually singular*.

Definition A.2.6 (Radon–Nikodym Derivative [Nikodym, 1930])**.** Suppose μ, ν are measures and $\mu \ll \nu$. Then there exists a measurable function $\frac{\mathrm{d}\mu}{\mathrm{d}\nu} : \mathcal{X} \to \mathbb{R}^+$ such that, for each $A \in \mathcal{B}_{\mathcal{X}}$:

$$\mu(A) = \int_A \frac{\mathrm{d}\mu}{\mathrm{d}\nu}(u)\, \nu(\mathrm{d}u).$$

The function $\frac{\mathrm{d}\mu}{\mathrm{d}\nu}$ is referred to as the *Radon–Nikodym derivative* of μ with-respect-to ν.

A.3 Monte–Carlo Sampling

In this section we summarise and introduce notation for the Monte–Carlo procedures used in the main text. Basic knowledge of MCMC procedures is assumed in this section; only the more esoteric procedures will be described in detail.

A.3.1 Metropolis-Adjusted Langevin Algorithm

The Metropolis-adjusted Langevin algorithm (MALA, Roberts and Tweedie [1996]) is an MCMC procedure with proposals that are derived from discretisation of a stochastic differential equation that has the target distribution as its invariant measure. Let μ be the target measure and assume that it is supported on $\Theta \subseteq \mathbb{R}^d$. Further assume that μ admits a density π with respect to the Lebesgue measure. The assumption of a Lebesgue density can be relaxed; see Beskos et al. [2017]. Proposals are then given by

$$\hat{\theta}_{i+1} = \theta_i + \Gamma\nabla \log \pi(\theta_i) + \sqrt{2}\Gamma^{\frac{1}{2}}\xi_k.$$

Here $\Gamma \in \mathbb{R}^{d\times d}$ is a preconditioner matrix that must be tuned to achieve reasonable acceptance probability, while $\xi_k \sim \mathcal{N}(0, I)$ IID. Since proposals are now asymmetric, the proposed move is accepted with probability $\alpha(\theta_k, \hat{\theta}_{k+1})$, where

$$\begin{aligned} \alpha(\theta, \theta') &= \min\left(1, \frac{\pi(\theta')q(\theta, \theta')}{\pi(\theta)q(\theta', \theta)}\right) \\ q(\theta, \theta') &= \exp\left(-\frac{1}{4}\left\|\theta - \theta' - \Gamma\nabla \log \pi(\theta')\right\|^2_{\Gamma^{-1}}\right). \end{aligned}$$

The introduction of gradient information into the proposals yields an MCMC procedure that tends to converge more rapidly; this was found to be useful in many of the challenging sampling problems presented in this thesis. MALA is presented as an algorithm in Algorithm A.4.

Algorithm A.4 MALA algorithm for sampling from measure μ with density π. Here θ_0 is an arbitrary initial state (assumed to be in the support of μ), P is the number of iterations to perform and Γ is a preconditioner matrix.

```
procedure MALA($\theta_0, P, \Gamma$)
    for $i = 1, \ldots, P$ do
        $\hat{\theta}_i \leftarrow \theta_i + \Gamma\nabla \log \pi(\theta_i) + \sqrt{2}\Gamma^{\frac{1}{2}}\xi_k$        ▷ $\xi_k \sim \mathcal{N}(0, 1)$
        if $\mathcal{U}(0, 1) < \alpha(\theta_{i-1}, \hat{\theta}_i)$ then
            $\theta_i \leftarrow \hat{\theta}_i$
        else
            $\theta_i \leftarrow \theta_{i-1}$
        end if
    end for
end procedure
return $\theta_1, \ldots, \theta_P$
```

A.3.2 Sequential Monte–Carlo

Let $\{\mu_i\}$, $i = 1, \ldots, N$ denote a set of distributions on a single measureable space Θ equipped with σ-algebra $\mathcal{B}_\Theta$. Each distribution is assumed to be absolutely continuous with respect to a common reference measure $\mu_0 \in \mathcal{P}_\Theta$. Suppose that μ_N is some target distribution, and that the intermediate μ_i, $i = 1, \ldots, N-1$ are related in some way, so that μ_i is "close" to μ_{i+1} in an appropriate (but unspecified) sense. Let $\{K_i\}$, $i = 1, \ldots, N$ be a set of *transition kernels* $K_i : \Theta \times \mathcal{B}_\Theta \to [0, 1]$ be such that $K_i(\,\cdot\,, B)$ is measureable for each $B \in \mathcal{B}_\theta$ and $K_i(\theta, \,\cdot\,)$ is a probability distribution on Θ for each $\theta \in \Theta$. Further assume that each K_i has μ_i as its invariant distribution. For intuition, in the context of the previous section a single iteration of any MCMC procedure is a valid transition kernel, as is any number of iterations of such a procedure. For each $\theta \in \Theta$, $K(\theta, \,\cdot\,)$ is a distribution over possible new locations when the transition kernel is applied, starting from θ.

In Sequential Monte–Carlo [Del Moral et al., 2006], the distribution μ_N is approximated by an ensemble of P particles that are evolved over the course of N iterations. The particles will be denoted $\{\theta_j^i\}$, $i = 1, \ldots, N$ and $j = 1, \ldots, P$. Each is associated with an importance weight w_j^i that is updated over the course of the procedure. The algorithm is initialised with an ensemble of particles $\theta_1^0, \ldots, \theta_P^0$ and uniform weights $w_j^i = P^{-1}$. A total of N iterations are performed, one for each distribution, and at each $i = 1, \ldots, N$ the following steps are performed:

1. The particles are *evolved* by sampling $\theta_j^i \sim K_i(\theta_j^{i-1}, \,\cdot\,)$.

2. Each weight is *updated* according to:

$$w_j^i = \frac{\mathrm{d}\mu_i}{\mathrm{d}\mu_{i-1}}(\theta_j^{i-1})w_j^{i-1}.$$

   The new weights are then normalised so that $\sum_{j=1}^P w_j^i = 1$.

3. The particles are *re-sampled* from the discrete distribution over $\{\theta_1^i, \ldots, \theta_P^i\}$ defined by the weights $\{w_1^i, \ldots, w_P^i\}$.

Commonly the re-sampling step is performed only when some measure of the sample quality decreases below a threshold, but this detail is not presented here. SMC is presented as an algorithm in Algorithm A.5.

Algorithm A.5 Sequential Monte Carlo. Here $\theta^0 = \{\theta_1^0, \ldots, \theta_P^0\}$ is the initial state, and $K = \{K_1, \ldots, K_N\}$ is the set of transition kernels.

```
procedure SMC($\theta^0, K$)
    $w_j^0 \leftarrow P^{-1}$ for $j = 1, \ldots, P$
    for $i = 1, \ldots, M$ do
        Sample $\hat{\theta}_j^i \sim K_i(\theta_j^{i-1}, \cdot)$ for $j = 1, \ldots, P$
        $\hat{w}_j^i \leftarrow \frac{\mathrm{d}\mu_i}{\mathrm{d}\mu_{i-1}}(\theta_j^{i-1})w_j^{i-1}$ for $j = 1, \ldots, P$
        $w_j^i \leftarrow \hat{w}_j^i(\sum_{j=1}^P w_j^i)^{-1}$ for $j = 1, \ldots, P$
        $\theta_j^i \sim \mathrm{Discrete}(\{\hat{\theta}_j^i\}_{i=1}^P; \{w_j^i\}_{i=1}^P)$, for $j = 1, \ldots, P$.
    end for
end procedure
```

A.3.3 Parallel Tempering

As in the previous section, suppose that μ_i are distributions and let K_i denote a set of appropriate transition kernels. The parallel tempering algorithm [Geyer, 1991] runs N Markov chains in parallel by alternating between application of K_i and proposing "swaps" between the states of adjacent chains. To be specific, at iteration j let k be selected uniformly at random from $\{1, \ldots, N-1\}$, and let θ^i denote the state of chain i. A swap is proposed between states θ^i and θ^{i+1}. To maintain the correct invariant distribution of the ensemble of Markov chains, this swap is accepted with probability

$$\alpha(\theta^k, \theta^{k+1}) = \frac{\pi_k(\theta^{k+1})\pi_{k+1}(\theta^k)}{\pi_k(\theta^k)\pi_{k+1}(\theta^{k+1})} \tag{A.1}$$

where π_k denotes the density of the target distribution μ_k with respect to some reference measure that the μ_i are mutually absolutely continuous with-respect-to, for $i = 1, \dots, N$.

The parallel tempering algorithm is described in Algorithm A.6.

Algorithm A.6 Parallel Tempering. Here $\theta_0 = \{\theta_0^1, \dots, \theta_0^N\}$ is the initial state, $K = \{K_1, \dots, K_M\}$ is the set of transition kernels and M is the number of iterations to perform.

```
procedure PT($\theta_0, K, P$)
    for $j = 1, \dots, P$ do
        Sample $\hat{\theta}_j^i \sim K_i(\theta_{j-1}^i, \cdot)$ for $i = 1, \dots, N$
        Sample $k \sim \mathrm{Uniform}(0, N-1)$
        if $U(0,1) < \alpha(\theta_j^k, \theta_j^{k+1})$ then
            Set $\theta_j^k = \hat{\theta}_j^{k+1}$ and $\theta_j^{k+1} = \hat{\theta}_j^k$
        else
            Set $\theta_j^k = \hat{\theta}_k^k$ and $\theta_j^{k+1} = \hat{\theta}_j^{k+1}$
        end if
        For $i \neq k, k+1$, set $\theta_j^i = \hat{\theta}_j^i$
    end for
    return $\{\theta_1^N, \dots, \theta_P^N\}$
end procedure
```

A.4 Chebyshev Polynomials

In this section, Chebyshev polynomials will be introduced. The exposition follows Sullivan [2015, Chapter 8]. We focus on Chebyshev polynomials of the first kind, as these are the polynomials used throughout the thesis.

The Chebyshev polynomials are a set of polynomials defined on $[-1, 1]$, that are uniformly bounded in this interval. Thus, $T_n : [-1, 1] \to [-1, 1]$ is a polynomial of degree n. The polynomials satisfy a three-term recurrence relation:

$$\begin{aligned} T_0(x) &\coloneqq 1 \\ T_1(x) &\coloneqq x \\ T_{n+1}(x) &\coloneqq 2xT_n(x) - T_{n-1}(x). \end{aligned}$$

Constructed thus, the polynomials have the following orthogonality property with-respect-to the measure μ on $[-1, 1]$ with density $p(x) = (1 - x^2)^{-\frac{1}{2}}$:

$$\langle T_n, T_m \rangle_{2,\mu} = \begin{cases} \frac{\pi}{2} & n = m = 0 \\ \delta_{nm}\pi & \text{otherwise} \end{cases} .$$

From this it is straightforward to instead make the Chebyshev polynomials orthonormal with respect to $\langle \cdot , \cdot \rangle_{2,\mu}$. The polynomials also have a closed form, given in Mason and Handscomb [2002, Section 1.4.2]:

$$T_n(x) = \frac{1}{2}\left[\left(x + \sqrt{x^2 - 1}\right)^m + \left(x - \sqrt{x^2 - 1}\right)^m\right]. \tag{A.2}$$

Appendix B

Dichotomy of Bayesian and Non-Bayesian PNM

Table B.1 originally appeared in Cockayne et al. [2019a], and presents a dichotomy of PNM based on Definition 3.1.5. For a more comprehensive and always up-to-date list, please see http://probnum.org/

Table B.1: Dichotomy of PNM.

Method	**QoI** $Q(x)$	**Information** $A(x)$	**Non- (or Approximate) BPNM**	**BPNM**
Integrator	$\int x(t)\nu(\mathrm{d}t)$	$\{x(t_i)\}_{i=1}^n$	Osborne et al. [2012b,a]; Gunter et al. [2014]	Bayesian Quadrature [Larkin, 1974; Diaconis, 1988; O'Hagan, 1991]
	$\int f(t)x(\mathrm{d}t)$	$\{t_i\}_{i=1}^n$ s.t. $t_i \sim x$	Kong et al. [2003]; Tan [2004]; Kong et al. [2007]	
	$\int x_1(t)x_2(\mathrm{d}t)$	$\{(t_i, x_1(t_i))\}_{i=1}^n$ s.t. $t_i \sim x_2$		Oates et al. [2017]
	$\{\int x_i(t)\nu(\mathrm{d}t)\}_{i=1}^n$	$\{x_i(t_j)\}_{i=1}^n, j = 1, \ldots, m$		Xi et al. [2018]
Optimiser	$\arg\min x(t)$	$\{x(t_i)\}_{i=1}^n$		Bayesian Optimisation [Mockus, 1989]
		$\{\nabla x(t_i)\}_{i=1}^n$		Hennig and Kiefel [2013]
		$\{(x(t_i), \nabla x(t_i)\}_{i=1}^n$		Probabilistic Line Search [Mahsereci and Hennig, 2015]
		$\{\mathbb{I}[t_{\min} < t_i]\}_{i=1}^n$		Probabilistic Bisection Algorithm [Horstein, 1963]
		$\{\mathbb{I}[t_{\min} < t_i] + \text{error}\}_{i=1}^n$	Waeber et al. [2013]	
Linear Solver	$x^{-1}b$	$\{xt_i\}_{i=1}^n$		Probabilistic Linear Solvers [Hennig, 2015; Bartels and Hennig, 2016]
	x	$\{s_i^\top x\}_{i=1}^n$		BayesCG [Cockayne et al., 2019b]
ODE Solver	x	$\{\nabla x(t_i)\}_{i=1}^n$	[Skilling, 1992]	

Method	**QoI** $Q(x)$	**Information** $A(x)$	**Non- (or Approximate) BPNM**	**BPNM**
			Filtering Methods for IVPs [Schober et al., 2014; Chkrebtii et al., 2016; Kersting and Hennig, 2016; Teymur et al., 2016; Schober et al., 2018]	
			Finite Difference Methods [John and Wu, 2017]	
		∇x + rounding error	Hull and Swenson [1966]; Mosbach and Turner [2009]	
		$\{\nabla x(H_i)\}_{i=1}^n$, H_i random	Abdulle and Garegnani [2018]	
		∇s, s obtained by Lie transformation of x		Wang et al. [2018]
	$x(t_{\text{end}})$	$\{\nabla x(t_i)\}_{i=1}^n$	Stochastic Euler [Krebs, 2017]	
PDE Solver	x	$\{Dx(t_i)\}_{i=1}^n$	Chkrebtii et al. [2016]; Raissi et al. [2018]	Probabilistic Meshless Methods [Owhadi, 2015, 2017; Cockayne et al., 2016; Raissi et al., 2017]
		Dx + discretisation error	Conrad et al. [2017]	
	$R_t(x)$	$R_{t_0}(x)$	Leike and Enßlin [2018]	

Appendix C

Proofs from Chapter 4

C.1 Proof of Proposition 4.3.2

The proof relies on the following lemma:

Lemma C.1.1. *Assume that the search directions $\{\boldsymbol{s}_i\}$ are $A\Sigma_0 A^\top$-orthogonal. Then it holds that at iteration m, the residual $\boldsymbol{r}_m = \boldsymbol{b} - A\boldsymbol{x}_m$ satisfies $\boldsymbol{r}_m^\top \boldsymbol{s}_i = 0$ for $i = 1, \ldots, m$.*

Proof. From the definitions of $\boldsymbol{r}_m$ and $\boldsymbol{x}_m$, we have that

$$\begin{aligned} \boldsymbol{s}_i^\top \boldsymbol{r}_m &= \boldsymbol{s}_i^\top \boldsymbol{b} - \boldsymbol{s}_i^\top A\boldsymbol{x}_m \\ &= \boldsymbol{s}_i^\top \boldsymbol{b} - \boldsymbol{s}_i^\top A\boldsymbol{x}_0 - \boldsymbol{s}_i^\top A\Sigma_0 A^\top S_m \Lambda_m^{-1} S_m^\top \boldsymbol{r}_0. \end{aligned}$$

Now, note that $\boldsymbol{s}_i^\top A\Sigma_0 A^\top S_m \Lambda_m^{-1} = \boldsymbol{e}_i^\top$, the vector with $[\boldsymbol{e}_i]_j = \delta_{ij}$, since $\boldsymbol{s}_i^\top A\Sigma_0 A^\top S_m$ is the i^{th} row of Λ_m whenever $i \leq m$. Thus, $\boldsymbol{s}_i^\top \boldsymbol{r}_m = \boldsymbol{s}_i^\top \boldsymbol{r}_0 - \boldsymbol{e}_i^\top S_m^\top \boldsymbol{r}_0 = 0$, which completes the proof. □

Proof of Proposition 4.3.2. Let $\tilde{\boldsymbol{t}}_1 := \boldsymbol{r}_0$, and for each $m > 1$, define $\tilde{\boldsymbol{t}}_m$ as

$$\tilde{\boldsymbol{t}}_m := \boldsymbol{r}_{m-1} - \sum_{i=1}^{m-1} \left(\boldsymbol{r}_{m-1}^\top Q \boldsymbol{t}_i \right) \boldsymbol{t}_i \tag{C.1}$$

where $Q = A\Sigma_0 A^\top$. Let $\boldsymbol{t}_m = \tilde{\boldsymbol{t}}_m / \|\tilde{\boldsymbol{t}}_m\|_Q$.

The proof is by induction. It will be shown that for each m the set of search directions $\{\boldsymbol{t}_i\}_{i=1}^m$ is Q-orthonormal, and further that each $\boldsymbol{t}_i = \boldsymbol{s}_i$, as defined in the proposition statement.

For $m = 1$ the set $\{\boldsymbol{t}_1\}$ is trivially Q-orthonormal and $\boldsymbol{t}_1 = \boldsymbol{s}_1$. For $m > 1$, we make the inductive hypothesis that $\{\boldsymbol{t}_i\}_{i=1}^{m-1}$ is Q-orthonormal and such that $\boldsymbol{t}_i = \boldsymbol{s}_i$, for $i = 1, \ldots, m-1$. Then, for each $j < m$

$$\begin{aligned} \boldsymbol{t}_j^\top Q \tilde{\boldsymbol{t}}_m &= \boldsymbol{t}_j^\top Q \boldsymbol{r}_{m-1} - \sum_{i=1}^{m-1} \boldsymbol{r}_{m-1}^\top Q \boldsymbol{t}_i \cdot \underbrace{\boldsymbol{t}_j^\top Q \boldsymbol{t}_i}_{=\delta_{ij}} \quad \text{(by the inductive hypothesis)} \\ &= \boldsymbol{t}_j^\top Q \boldsymbol{r}_{m-1} - \boldsymbol{t}_j^\top Q \boldsymbol{r}_{m-1} = 0 \end{aligned} \tag{C.2}$$

and therefore the set $\{\boldsymbol{t}_i\}_{i=1}^m$ is Q-orthonormal.

As a result, the assumptions of Proposition 4.3.1 are satisfied, allowing us to apply this proposition to find

$$\begin{aligned} \boldsymbol{r}_j &= \boldsymbol{b} - A\boldsymbol{x}_j \\ &= \boldsymbol{b} - A\boldsymbol{x}_{j-1} - Q\boldsymbol{t}_j(\boldsymbol{t}_j^\top \boldsymbol{r}_{j-1}) \\ \implies Q\boldsymbol{t}_j &= \frac{\boldsymbol{r}_{j-1} - \boldsymbol{r}_j}{\boldsymbol{t}_j^\top \boldsymbol{r}_{j-1}} \\ \implies \boldsymbol{r}_{m-1}^\top Q\boldsymbol{t}_j &= \frac{\boldsymbol{r}_{m-1}^\top \boldsymbol{r}_{j-1} - \boldsymbol{r}_{m-1}^\top \boldsymbol{r}_j}{\boldsymbol{t}_j^\top \boldsymbol{r}_{j-1}}. \end{aligned} \tag{C.3}$$

Since the set $\{\boldsymbol{t}_i\}_{i=1}^m$ is Q-orthonormal, we have from Lemma C.1.1 that for each $j \le m$, $\boldsymbol{r}_m^\top \boldsymbol{t}_j = 0$. Thus, by taking $m = j$ in Eq. (C.1) and then left-multiplying by $\boldsymbol{r}_m^\top$ it holds that for each $j \le m$:

$$0 = \boldsymbol{r}_m^\top \tilde{\boldsymbol{t}}_j := \boldsymbol{r}_m^\top \boldsymbol{r}_{j-1} - \sum_{i=1}^{m-1} \boldsymbol{r}_{m-1}^\top Q \boldsymbol{t}_i \cdot \underbrace{\boldsymbol{r}_m^\top \boldsymbol{t}_i}_{=0}. \tag{C.4}$$

from which we conclude that $\boldsymbol{r}_m^\top \boldsymbol{r}_j = 0$ whenever $j < m$. Applying this result to Eq. (C.3), we find that:

$$\boldsymbol{r}_{m-1} Q \boldsymbol{t}_j = 0 \quad \forall \quad j < m.$$

Returning to Eq. (C.1), we have:

$$\begin{aligned} \tilde{\boldsymbol{t}}_m &= \boldsymbol{r}_{m-1} - \sum_{i=1}^{m-1} \left(\boldsymbol{r}_{m-1}^\top Q \boldsymbol{t}_i\right) \boldsymbol{t}_i \\ &= \boldsymbol{r}_{m-1} - \left(\boldsymbol{r}_{m-1}^\top Q \boldsymbol{t}_{m-1}\right) \boldsymbol{t}_{m-1} - \sum_{i=1}^{m-2} \underbrace{\left(\boldsymbol{r}_{m-1}^\top Q \boldsymbol{t}_i\right)}_{=0} \boldsymbol{t}_i \\ &= \boldsymbol{r}_{m-1} - (\boldsymbol{r}_{m-1}^\top Q \boldsymbol{t}_{m-1}) \boldsymbol{t}_{m-1} \end{aligned}$$

which is equal to $\tilde{\boldsymbol{s}}_m$ for each $m > 1$, completing the proof. □

C.2 Proof of Proposition 4.3.3

Throughout this section, let $Q = A\Sigma_0 A^\top$ and let $\bar{K}_m = K_m(Q, \boldsymbol{r}_0)$. The proof relies on the following lemma:

Lemma C.2.1. *Suppose that* $\operatorname{span}(\{\boldsymbol{s}_1, \ldots, \boldsymbol{s}_m\}) = \bar{K}_m$ *and* $\{\boldsymbol{s}_1, \ldots, \boldsymbol{s}_m\}$ *is a Q-orthonormal set. Then* $\boldsymbol{r}_m = \boldsymbol{b} - A\boldsymbol{x}_m$ *is such that* $\boldsymbol{r}_m \in \bar{K}_{m+1}$.

Proof. Proof is by induction. Clearly $\boldsymbol{r}_0 \in \bar{K}_1$. Assume that $\boldsymbol{r}_i \in \bar{K}_{i+1}$ for all $i = 1, \ldots, m-1$. Then from Proposition 4.3.1 we have that, since $\{\boldsymbol{s}_1, \ldots, \boldsymbol{s}_m\}$ are Q-orthonormal:

$$\begin{aligned}
\boldsymbol{r}_m &= \boldsymbol{b} - A\boldsymbol{x}_m \\
&= \boldsymbol{b} - A\boldsymbol{x}_{m-1} + Q\boldsymbol{s}_m\boldsymbol{s}_m^\top\boldsymbol{r}_{m-1} \\
&= \underbrace{\boldsymbol{r}_{m-1}}_{\in\bar{K}_m} + \underbrace{Q\boldsymbol{s}_m}_{\in\bar{K}_{m+1}} \boldsymbol{s}_m^\top\boldsymbol{r}_{m-1}
\end{aligned}$$

so that $\boldsymbol{r}_m \in \bar{K}_{m+1}$ as required. □

Proof of Proposition 4.3.3. For $m = 1$, the first search direction is given by

$$\boldsymbol{s}_1 = \frac{\boldsymbol{r}_0}{\|\boldsymbol{r}_0\|_Q}$$

so that clearly $\operatorname{span}(\{\boldsymbol{s}_1\}) = \bar{K}_1$. Now for the inductive step, assume that

$$\operatorname{span}(\{\boldsymbol{s}_1, \ldots, \boldsymbol{s}_{m-1}\}) = \bar{K}_{m-1}.$$

From Proposition 4.3.2 we have that

$$\tilde{\boldsymbol{s}}_m = \underbrace{\boldsymbol{r}_{m-1}}_{\in\bar{K}_m} - (\boldsymbol{r}_{m-1}^\top Q\boldsymbol{s}_{m-1}) \underbrace{\boldsymbol{s}_{m-1}}_{\in\bar{K}_{m-1}}$$

where $\boldsymbol{s}_{m-1} \in \bar{K}_{m-1}$ by the inductive assumption, and $\boldsymbol{r}_{m-1} \in \bar{K}_m$ from Lemma C.2.1, noting that the assumptions of that Lemma are satisfied by the inductive assumption and Proposition 4.3.2. Thus, $\tilde{\boldsymbol{s}}_m \in \bar{K}_m$, as required, and so to must $\boldsymbol{s}_m \in \bar{K}_m$. □

C.3 Proof of Proposition 4.3.5

Throughout, let $Q = A\Sigma_0 A^\top$.

Proof of Proposition 4.3.5. We begin by introducing the *operator norm* induced by the energy norm $\|\cdot\|_A$, which is a norm on matrices $M \in \mathbb{R}^{d\times d}$

$$\|M\|_A^{\mathrm{op}} = \sup\{\|M\boldsymbol{v}\|_A : \|\boldsymbol{v}\|_A = 1\}.$$

From Proposition 4.3.3, it follows that

$$\boldsymbol{x}_m \in \boldsymbol{x}_0 + \Sigma_0 A^\top K_m(A\Sigma_0 A^\top, \boldsymbol{r}_0) =: K_m^*$$

Thus, it holds that there exists a polynomial $\tilde{P}_{m-1}$ of degree $m-1$ such that

$$\begin{aligned}\boldsymbol{e}_m := \boldsymbol{x}_m - \boldsymbol{x}^\dagger &= \boldsymbol{x}_0 - \boldsymbol{x}^\dagger + \Sigma_0 A^\top \tilde{P}_{m-1}(Q)\boldsymbol{r}_0\\ &= \boldsymbol{e}_0 + \Sigma_0 A^\top \tilde{P}_{m-1}(Q)A\boldsymbol{e}_0\\ &= P_m(\Sigma_0 A^\top A)\boldsymbol{e}_0\end{aligned}$$

where P_m is some polynomial of degree m. From Corollary 4.3.4 we have that $P_m \in \mathbb{P}_m$ is constructed to minimise the error:

$$\begin{aligned}\|\boldsymbol{e}_m\|_{\Sigma_0^{-1}} &\le \|P_m(\Sigma_0 A^\top A)\|_{\Sigma_0^{-1}}^{\mathrm{op}} \cdot \|\boldsymbol{e}_0\|_{\Sigma_0^{-1}}\\ &= \|\Sigma_0^{-\frac{1}{2}} P_m(\Sigma_0 A^\top A)\Sigma_0^{\frac{1}{2}}\|_I^{\mathrm{op}} \cdot \|\boldsymbol{e}_0\|_{\Sigma_0^{-1}}\\ &= \|P_m(\Sigma_0^{\frac{1}{2}} A^\top A \Sigma_0^{\frac{1}{2}})\|_I^{\mathrm{op}} \cdot \|\boldsymbol{e}_0\|_{\Sigma_0^{-1}}\end{aligned}$$

Now, note that $\Sigma_0^{\frac{1}{2}} A^\top A \Sigma_0^{\frac{1}{2}}$ is symmetric. Thus, let Γ be the matrix with the eigenvalues of $\Sigma_0^{\frac{1}{2}} A^\top A \Sigma_0^{\frac{1}{2}}$ on its diagonal, and let V be the orthonormal matrix whose columns are its eigenvectors. Thus, we have that

$$\Sigma_0^{\frac{1}{2}} A^\top A \Sigma_0^{\frac{1}{2}} = V\Gamma V^\top.$$

Furthermore note that $\Sigma_0 A^\top A = \Sigma_0^{\frac{1}{2}}[\Sigma_0^{\frac{1}{2}} A^\top A \Sigma_0^{\frac{1}{2}}]\Sigma_0^{-\frac{1}{2}}$. Hence, $\Sigma_0 A^\top A$ is similar to $\Sigma_0^{\frac{1}{2}} A^\top A \Sigma_0^{\frac{1}{2}}$, and so the matrices share the same eigenvalues.

Now, clearly $P_m(V\Gamma V^\top) = V P_m(\Gamma) V^\top$ since V is orthonormal. Thus

$$\begin{aligned}\|\boldsymbol{e}_m\|_I &\le \underbrace{\|V\|_I^{\mathrm{op}}\|V^\top\|_I^{\mathrm{op}}}_{=1} \|P_m(\Gamma)\|_I^{\mathrm{op}} \cdot \|\boldsymbol{e}_0\|_{\Sigma_0^{-1}}\\ &= \|P_m(\Gamma)\|_I^{\mathrm{op}} \cdot \|\boldsymbol{e}_0\|_{\Sigma_0^{-1}} \qquad \text{(C.5)}\end{aligned}$$

where $\|V\|_I^{\mathrm{op}}\|V^\top\|_I^{\mathrm{op}} = 1$ follows since V is unitary and $\|\cdot\|_I^{\mathrm{op}}$ coincides with the matrix 2-norm, which is unitarily invariant. Let $\mathbb{P}_m$ denote the set of all polynomials P of order m with $P(0) = 1$. This requirement ensures that if A is singular, $\|\boldsymbol{e}_m\|_{\Sigma_0^{-1}} = \|\boldsymbol{e}_0\|_{\Sigma_0^{-1}}$ for all m. Let $\bar{\Gamma}$ denote the set of eigenvalues of $\Sigma_0^{\frac{1}{2}} A^\top A \Sigma_0^{\frac{1}{2}}$. Then

$$\begin{aligned}
\|P_m(\Gamma)\|_I^{\mathrm{op}} &= \min_{P \in \mathbb{P}_m} \max_{\gamma \in \bar{\Gamma}} \sup_{\|\boldsymbol{v}\|_2 = 1} \|P(\gamma)\boldsymbol{v}\|_2 \\
&= \min_{P \in \mathbb{P}_m} \max_{\gamma \in \bar{\Gamma}} |P(\gamma)| \\
&\leq \min_{P \in \mathbb{P}_m} \max_{\gamma \in [\gamma_{\min}, \gamma_{\max}]} |P(\gamma)|. \qquad \text{(C.6)}
\end{aligned}$$

Lemma C.3.1, proven below, establishes that the polynomial minimising this expression is

$$P(\gamma) = \frac{T_m\left(\frac{\gamma_{\max} + \gamma_{\min} - 2\gamma}{\gamma_{\max} - \gamma_{\min}}\right)}{T_m\left(\frac{\gamma_{\max} + \gamma_{\min}}{\gamma_{\max} - \gamma_{\min}}\right)}$$

where $T_m(\,\cdot\,)$ is the m^{th} Chebyshev polynomial of the first kind; see Appendix A.4 for a detailed introduction.

Let $\kappa = \gamma_{\max}/\gamma_{\min}$. Now, $T_m(z) \in [-1, 1]$ for all m and all $z \in [-1, 1]$; thus the numerator takes maximum value 1. Therefore

$$\|P_m(\Gamma)\|_{\Sigma_0^{-1}}^{\mathrm{op}} \leq \left| T_m\left(\frac{\kappa + 1}{\kappa - 1}\right)\right|^{-1}.$$

Lastly, note from Eq. (A.2) that:

$$T_m(z) = \frac{1}{2}\left[\left(z + \sqrt{z^2 - 1}\right)^m + \left(z - \sqrt{z^2 - 1}\right)^m\right]$$

so that

$$\begin{aligned}
\|P_m(\Gamma)\|_2^{\mathrm{op}} &\leq 2\left[\left(\frac{\sqrt{\kappa} + 1}{\sqrt{\kappa} - 1}\right)^m + \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1}\right)^m\right]^{-1} \\
&\leq 2\left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1}\right)^m.
\end{aligned}$$

Inserting this into Eq. (C.5) and noting that since $\Sigma_0^{\frac{1}{2}} A^\top A \Sigma_0^{\frac{1}{2}}$ has the same eigenvalues as $\Sigma_0 A^\top A$, it also has the same condition number, completes the proof. □

Lemma C.3.1 (Appendix S3 of Shewchuk [1994]). *Eq.* (C.6) *is minimised by*

$$P(\gamma) = \frac{T_m\left(\frac{\gamma_{\max}+\gamma_{\min}-2\gamma}{\gamma_{\max}-\gamma_{\min}}\right)}{T_m\left(\frac{\gamma_{\max}+\gamma_{\min}}{\gamma_{\max}-\gamma_{\min}}\right)}$$

where T_m is the m^{th} Chebyshev polynomial of the first kind.

Proof. For convenience let

$$\gamma_0 := \frac{\gamma_{\max}+\gamma_{\min}}{\gamma_{\max}-\gamma_{\min}}.$$

Note that $\gamma_0 > 1$. Further note that

$$\gamma \in [\gamma_{\min}, \gamma_{\max}] \implies \frac{\gamma_{\max}+\gamma_{\min}-2\gamma}{\gamma_{\max}-\gamma_{\min}} \in [-1, 1].$$

Now recall the following properties of Chebyshev polynomials (see Mason and Handscomb [2002]):

C1 $T_m(z) \in [-1, 1]$ for all $z \in [-1, 1]$.

C2 $T_m(1) = 1$, and $T_m(-1) = (-1)^m$.

C3 Let $Z = \{z_i\}, i = 1, \dots, m$ denote the ordered zeros of $T_m(z)$. Then, $Z \subset [-1, 1]$.

C4 $T_m(z)$ attains the value $(-1)^{m+i}$ in the range $[z_i, z_{i+1}]$ for $i = 1, \dots, m-1$.

First, note that $T_m(\gamma_0) > 1$; this is because $\gamma_0 > 1$, $T_m(1) = 1$ from **C2**, and T_m attains all its zeros in $[-1, 1]$ from **C3**; thus T_m is a strictly increasing function in $(1, \infty)$. From this it is also clear that $P(0) = 1$, since $T_m(\gamma_0) \neq 0$. Thus, $P(\gamma) \in \mathbb{P}_m$ as required. Further, note that

$$\max_{\gamma \in [\gamma_{\min}, \gamma_{\max}]} |P(\gamma)| = T_m(\gamma_0)^{-1}$$

since the denominator is strictly positive from the argument above, while the numerator attains its maximum value 1 in $[-1, 1]$ from **C1** and **C3**.

Proof that $P(\gamma)$ minimizes Eq. (C.6) is by contradiction. Suppose there is a $Q(\gamma) \in \mathbb{P}_m$ with

$$\max_{\gamma \in [\gamma_{\min}, \gamma_{\max}]} |Q(\gamma)| < T_m(\gamma_0)^{-1}. \tag{C.7}$$

Now consider the polynomial $P(\gamma) - Q(\gamma)$, which is a polynomial of degree m. From **C1**, $P(\gamma) \in [-T_m(\gamma_0)^{-1}, T_m(\gamma_0)^{-1}]$, and $P(\gamma)$ has m zeros in $[\gamma_{\min}, \gamma_{\max}]$.

From Eq. (C.7) it is clear that $P(\gamma) - Q(\gamma)$ also has m zeros in $[\gamma_{\text{min}}, \gamma_{\text{max}}]$, as to prevent $P(\gamma)$ from crossing zero between its extrema in this range would require $|Q(\gamma)| > T_m(\gamma_0)^{-1}$ (by **C4**).

However, since $P(0) = Q(0) = 1$, $P - Q$ has an additional zero outside $[\gamma_{\text{min}}, \gamma_{\text{max}}]$. Therefore, $P - Q$ is a polynomial of degree m with at least $m + 1$ zeros, which is a contradiction. Thus $P(\gamma)$ minimises Eq. (C.6).

□

Appendix D

Proofs from Chapter 5

D.1 Proof of Proposition 5.3.2

For convenience we introduce the Löwner ordering on positive semidefinite matrices. For $A, B \in \mathbb{R}^{d\times d}$ we say that $A \preceq B$ if $B - A$ is positive semidefinite. A natural corollary of this is that if $A \preceq B$ then $\boldsymbol{x}^\top A\boldsymbol{x} < \boldsymbol{x}^\top B\boldsymbol{x}$ for all $\boldsymbol{x} \in \mathbb{R}^d$. Furthermore, if $A \preceq B$ and each of A, B are nonsingular, then $B^{-1} \preceq A^{-1}$. For more information, see Bernstein [2009].

From Dashti and Stuart [2017, Theorem 4.9], it is sufficient to show that the two potentials $\Phi_h(\boldsymbol{y};\theta)$ and $\Phi(\boldsymbol{y};\theta,u^\dagger)$ are asymptotically identical. Let $\Phi_{\text{coll}}(\theta) = \Phi(\boldsymbol{y};\theta,m_1(\,\cdot\,,\theta))$ be the potential when a symmetric collocation forward solver is used with the same set of design points $X^{\mathcal{AB}}$. Then, suppressing dependence on θ and $u^\dagger$, the triangle inequality yields:

$$|\Phi_h(\theta) - \Phi(\theta)| \leq \underbrace{|\Phi_h(\theta) - \Phi_{\text{coll}}(\theta)|}_{(1)} + \underbrace{|\Phi_{\text{coll}}(\theta) - \Phi(\theta)|}_{(2)}. \tag{D.1}$$

Beginning with (2) and letting $\boldsymbol{u}(\theta) = \mathcal{G}(u^\dagger(\,\cdot\,;\theta))$ and $\Gamma^{-1} = G$, we have

$$\begin{aligned}
&2(\Phi_{\text{coll}}(\theta) - \Phi(\theta))\\
&\qquad = \|\boldsymbol{y} - \boldsymbol{m}(\theta)\|_G^2 - 2\Phi(\theta)\\
&\qquad \leq \|\boldsymbol{y} - \boldsymbol{u}(\theta)\|_G^2 + 2\|\boldsymbol{y} - \boldsymbol{u}(\theta)\|_G\|\boldsymbol{u}(\theta) - \boldsymbol{m}(\theta)\|_G\\
&\qquad\qquad + \|\boldsymbol{u}(\theta) - \boldsymbol{m}(\theta)\|_G^2 - 2\Phi(\theta)\\
&\qquad = 2\|\boldsymbol{y} - \boldsymbol{u}(\theta)\|_G\|\boldsymbol{u}(\theta) - \boldsymbol{m}(\theta)\|_G + \|\boldsymbol{u}(\theta) - \boldsymbol{m}(\theta)\|_G^2
\end{aligned}$$

where on the final line we have used the fact that $\Phi(\theta) = \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{u}(\theta)\|_G^2$. Noting that all of the terms on the final line are positive we therefore have

$$\begin{aligned}
&2|\Phi_{\text{coll}}(\theta) - \Phi(\theta)| \\
&\qquad \leq \underbrace{\|\boldsymbol{u}(\theta) - \boldsymbol{m}(\theta)\|_G^2}_{(a)} + 2\underbrace{\|\boldsymbol{y} - \boldsymbol{u}(\theta)\|_G \|\boldsymbol{u}(\theta) - \boldsymbol{m}(\theta)\|_G}_{(b)}.
\end{aligned}$$

Now let $\gamma = (\lambda_{\min}[\boldsymbol{\Gamma}])^{-1}$ and note that γ is the maximal eigenvalue of G. Thus $\|\boldsymbol{x}\|_G^2 \leq \gamma \|\boldsymbol{x}\|_2^2$. Further, recall that $\|\cdot\|_2 \leq \sqrt{n}\|\cdot\|_\infty$. Applying this first to (a):

$$\|\boldsymbol{u}(\theta) - \boldsymbol{m}(\theta)\|_G \leq \sqrt{\gamma}\|\boldsymbol{u}(\theta) - \boldsymbol{m}(\theta)\|_2 \leq \sqrt{\gamma\, n}\|\boldsymbol{u}(\theta) - \boldsymbol{m}(\theta)\|_\infty. \tag{D.2}$$

Similarly for (b), we can show that:

$$\|\boldsymbol{y} - \boldsymbol{u}(\theta)\|_G \leq \sqrt{\gamma}\|\boldsymbol{y} - \boldsymbol{u}(\theta)\|_2$$

This yields the bound

$$\begin{aligned}
2|\Phi_{\text{coll}}(\theta) - \Phi(\theta)| &\leq \gamma n \|\boldsymbol{u}(\theta) - \boldsymbol{m}(\theta)\|_\infty^2 + 2\gamma\sqrt{n}\|\boldsymbol{u}(\theta) - \boldsymbol{m}(\theta)\|_\infty \|\boldsymbol{y} - \boldsymbol{u}(\theta)\|_2 \\
&\leq \gamma n H^2 \bar{C}_\theta^2 + 2\gamma\sqrt{n} H \bar{C}_\theta \|\boldsymbol{y} - \boldsymbol{u}(\theta)\|_2
\end{aligned}$$

by application of Proposition 5.2.5 and Proposition 5.2.6, and with $H = h^{\beta-\rho-d/2}$ and $\bar{C}_\theta = C_\theta^F \|u^\dagger(\,\cdot\,;\theta)\|_k$.

Now returning to (1) from Eq. (D.1)

$$2|\Phi_h(\theta) - \Phi_{\text{coll}}(\theta)| = \left| \|\boldsymbol{y} - \boldsymbol{m}(\theta)\|_{(\Sigma(\theta)+\Gamma)^{-1}}^2 - \|\boldsymbol{y} - \boldsymbol{m}(\theta)\|_G^2 \right|.$$

By applying the Woodbury identity and letting $M(\theta) := G(\Sigma^{-1}(\theta) + G)^{-1}G$, note that

$$\begin{aligned}
&(\boldsymbol{y} - \boldsymbol{m}(\theta))^\top (\Sigma(\theta) + \Gamma)^{-1} (\boldsymbol{y} - \boldsymbol{m}(\theta)) \\
&\quad = (\boldsymbol{y} - \boldsymbol{m}(\theta))^\top G (\boldsymbol{y} - \boldsymbol{m}(\theta)) - (\boldsymbol{y} - \boldsymbol{m}(\theta))^\top G(\Sigma(\theta)^{-1} + G)^{-1} G (\boldsymbol{y} - \boldsymbol{m}(\theta)) \\
&\quad = \|\boldsymbol{y} - \boldsymbol{m}(\theta)\|_G^2 - \|\boldsymbol{y} - \boldsymbol{m}(\theta)\|_{M(\theta)}^2
\end{aligned}$$

Thus, we have

$$\begin{aligned}
&2|\Phi_h(\theta) - \Phi_{\text{coll}}(\theta)| \\
&\qquad = \|\boldsymbol{y} - \boldsymbol{m}(\theta)\|^2_{M(\theta)} \\
&\qquad \leq \underbrace{\|\boldsymbol{y} - \boldsymbol{u}(\theta)\|^2_{M(\theta)}}_{(c)} + \underbrace{\|\boldsymbol{u} - \boldsymbol{m}(\theta)\|^2_{M(\theta)}}_{(d)} + 2\|\boldsymbol{y} - \boldsymbol{u}(\theta)\|_{M(\theta)}\|\boldsymbol{u} - \boldsymbol{m}(\theta)\|_{M(\theta)}.
\end{aligned}$$

For (c), note that

$$\begin{aligned}
\Sigma^{-1}(\theta) + G &\succeq \Sigma^{-1}(\theta) \\
\implies (\Sigma^{-1}(\theta) + G)^{-1} &\preceq \Sigma(\theta) \\
\implies M(\theta) &\preceq G\Sigma(\theta)G
\end{aligned}$$

so that

$$\begin{aligned}
\|\boldsymbol{y} - \boldsymbol{u}(\theta)\|_{M(\theta)} &\leq \|\boldsymbol{y} - \boldsymbol{u}(\theta)\|_{G\Sigma(\theta)G} \\
&\leq \gamma \sqrt{\text{trace}(\Sigma(\theta))}\|\boldsymbol{y} - \boldsymbol{u}(\theta)\|_2
\end{aligned}$$

where we have used the fact that for any positive-definite $A \in \mathbb{R}^{d\times d}$, $A \preceq \text{trace}(A)I$. For (d) we have

$$\begin{aligned}
\Sigma^{-1}(\theta) + G &\succeq G \\
\implies (\Sigma^{-1}(\theta) + G)^{-1} &\preceq G \\
\implies M(\theta) &\preceq GG^{-1}G = G
\end{aligned}$$

and so

$$\begin{aligned}
\|\boldsymbol{u}(\theta) - \boldsymbol{m}(\theta)\|_{M(\theta)} &\leq \|\boldsymbol{u}(\theta) - \boldsymbol{m}(\theta)\|_G \\
&\leq \sqrt{\gamma\, n}\|\boldsymbol{u}(\theta) - \boldsymbol{m}(\theta)\|_\infty
\end{aligned}$$

from Eq. (D.2). This yields the bound

$$\begin{aligned}
2|\Phi_h(\theta) - \Phi_{\text{coll}}(\theta)| &\leq \gamma\ \text{trace}(\Sigma(\theta))\|\boldsymbol{y} - \boldsymbol{u}(\theta)\|_2^2 + \gamma n\|\boldsymbol{u}(\theta) - \boldsymbol{m}(\theta)\|_\infty^2 \\
&\quad + 2\gamma\sqrt{n\text{trace}(\Sigma(\theta))}\|\boldsymbol{y} - \boldsymbol{u}(\theta)\|_2\|\boldsymbol{u}(\theta) - \boldsymbol{m}(\theta)\|_\infty \\
&\leq \gamma H^2\|\boldsymbol{y} - \boldsymbol{u}(\theta)\|_2^2 + \gamma n H^2 \bar{C}_\theta^2 + 2\gamma\sqrt{n}H^2\bar{C}_\theta\|\boldsymbol{y} - \boldsymbol{u}(\theta)\|_2
\end{aligned}$$

Combining these bounds, we obtain

$$\begin{aligned}
&2|\Phi_h(\theta) - \Phi(\theta)| \\
&\qquad \leq \gamma H\Bigg(H\|\boldsymbol{y} - \boldsymbol{u}(\theta)\|_2^2 + 2nH\bar{C}_\theta^2 + 2\sqrt{n}H\bar{C}_\theta\|\boldsymbol{y} - \boldsymbol{u}(\theta)\|_2 \\
&\qquad\qquad\qquad + 2\sqrt{n}\bar{C}_\theta\|\boldsymbol{y} - \boldsymbol{u}(\theta)\|_2\Bigg) \\
&\qquad \leq \gamma H\left(\|\boldsymbol{y} - \boldsymbol{u}(\theta)\|_2^2 + 2n\bar{C}_\theta^2 + 4\sqrt{n}\bar{C}_\theta\|\boldsymbol{y} - \boldsymbol{u}(\theta)\|_2\right)
\end{aligned}$$

for all h sufficiently small and such that $h < 1$.

Having established this bound, we now turn to verifying that the assumptions required for Dashti and Stuart [2017, Theorem 4.9] hold. It is required that there exist functions $M_1, M_2 : \mathbb{R}^+ \to \mathbb{R}^+$ so that

D1: $\Phi(\theta) \geq -M_1(\|\theta\|_\Theta)$,

D2: $\Phi_h(\theta) \geq -M_1(\|\theta\|_\Theta)$,

D3: $|\Phi_h(\theta) - \Phi(\theta)| \leq M_2(\|\theta\|_\Theta)\varphi(h)$,

D4: $\exp(M_1(\|\theta\|_\Theta)\left(1 + M_2(\|\theta\|_\Theta)^2\right)$ is integrable in θ with respect to μ.

where $\varphi(h) \to 0$ as $h \to 0$. If (D1-4) hold then it can be concluded that, for h sufficiently small, $d_{\text{Hell}}(\mu_\theta^{\boldsymbol{y},h}, \mu_\theta^{\boldsymbol{y}}) \leq C\varphi(h)$ for some constant C, as required.

Taking $M_1(\|\theta\|_\Theta) = 0$ satisfies both (D1) and (D2), as $\Phi(\theta) \geq 0$ and $\Phi_h(\theta) \geq 0$ for all $\theta \in \Theta$. For (D3) and (D4), take $\varphi(h) = h^{\beta-\rho-d/2}$ and let

$$\eta = \sup\{\|\mathcal{G}(u)\|_2 : u \in H_{\hat{k}}(D)\ ,\ \|u\|_{\hat{k}} \leq 1\}$$

Note that $\eta < \infty$ since $\mathcal{G}$ is a bounded linear operator by assumption. To define $M_2(\|\theta\|_\Theta)$, first note:

$$\begin{aligned}
\bar{C}_\theta\|\boldsymbol{y} - \boldsymbol{u}(\theta)\|_2 &= C_\theta^F\|u^\dagger(\,\cdot\,;\theta)\|_{\hat{k}}\|\boldsymbol{y} - \boldsymbol{u}(\theta)\|_2 \\
&\leq C_\theta^F\|u^\dagger(\,\cdot\,;\theta)\|_{\hat{k}}(\|\boldsymbol{y}\|_2 + \|\boldsymbol{u}(\theta)\|_2) \\
&\leq C_\theta^F\|u^\dagger(\,\cdot\,;\theta)\|_{\hat{k}}(\|\boldsymbol{y}\|_2 + \eta\|u(\,\cdot\,;\theta)\|_{\hat{k}}) \\
&\leq (\|\boldsymbol{y}\|_2 + \eta)C(\|\theta\|_\Theta)
\end{aligned}$$

where in the last line we have applied Assumption 5.3.1. Further:

$$\begin{aligned}
\|\boldsymbol{y} - \boldsymbol{u}(\theta)\|_2^2 + 2n\bar{C}_\theta^2 &= \|\boldsymbol{y} - \boldsymbol{u}(\theta)\|_2^2 + 2n\left[C_\theta^F \|u^\dagger(\,\cdot\,;\theta)\|_{\hat{k}}\right]^2 \\
&\leq \|\boldsymbol{y}\|_2^2 + \eta^2 \|u^\dagger(\,\cdot\,;\theta)\|_{\hat{k}}^2 + 2\eta\|\boldsymbol{y}\|_2 \|u^\dagger(\,\cdot\,;\theta)\|_{\hat{k}} + 2n\left[C_\theta^F \|u^\dagger(\,\cdot\,;\theta)\|_{\hat{k}}\right]^2 \\
&\leq \|\boldsymbol{y}\|_2^2 + \left[\eta^2 + 2\eta\|\boldsymbol{y}\|_2\right] C(\|\theta\|_\Theta) + 2nC(\|\theta\|_\Theta)^2
\end{aligned}$$

having again applied Assumption 5.3.1 in the last line. Lastly, define

$$M_2(\|\theta\|_\Theta) \coloneqq 4\gamma\sqrt{n}\|\boldsymbol{y}\|_2^2 + \gamma\left[\eta^2 + 4\sqrt{n}\eta + (2\eta+1)\|\boldsymbol{y}\|_2\right] C(\|\theta\|_\Theta) + 2\gamma n C(\|\theta\|_\Theta)^2$$

Then by construction (D3) is satisfied. Furthermore (D4) is satisfied by the integrability assumption in Assumption 5.3.1. This completes the proof.

Appendix E

Proofs from Chapter 6

E.1 Proof of Theorem 6.2.5

Throughout, let $\nu = A_{\#}\mu$. Fix $f \in \mathcal{F}$ and $\boldsymbol{y} \in \mathcal{Y}$. Then we have that:

$$\begin{aligned}
\mu_\delta^{\boldsymbol{y}}(f) &= \frac{1}{Z_\delta^{\boldsymbol{y}}} \int_{\mathcal{X}} f(u)\, \phi_\delta^{\boldsymbol{y}}(u)\, \mu(\mathrm{d}u) \\
&= \frac{1}{Z_\delta^{\boldsymbol{y}}} \int_{\mathcal{X}} \int_{\mathcal{Y}} f(u)\, \phi\left(\frac{\|\tilde{\boldsymbol{y}} - \boldsymbol{y}\|_{\mathcal{Y}}}{\delta}\right) \mu^{\tilde{\boldsymbol{y}}}(\mathrm{d}u)\, \nu(\mathrm{d}\tilde{\boldsymbol{y}}) \\
&= \frac{1}{Z_\delta^{\boldsymbol{y}}} \int_{\mathcal{Y}} \phi\left(\frac{\|\tilde{\boldsymbol{y}} - \boldsymbol{y}\|_{\mathcal{Y}}}{\delta}\right) \mu^{\tilde{\boldsymbol{y}}}(f)\, \nu(\mathrm{d}\tilde{\boldsymbol{y}}) \\
&= \int_{\mathcal{Y}} \mu^{\tilde{\boldsymbol{y}}}(f)\, [A_{\#}\mu_\delta^{\boldsymbol{y}}](\mathrm{d}\tilde{\boldsymbol{y}}).
\end{aligned}$$

Where the second line follows from application of property (3) in Definition 6.1.1, the third from an application of Fubini's theorem (recalling that ϕ is bounded, and $f \in \mathcal{F}$ so f is also bounded) and the final line is by the definition of $\mu_\delta^{\boldsymbol{y}}$. Thus

$$\begin{aligned}
|\mu_\delta^{\boldsymbol{y}}(f) - \mu^{\boldsymbol{y}}(f)| &= \left| \int_{\mathcal{Y}} \left(\mu^{\tilde{\boldsymbol{y}}}(f) - \mu^{\boldsymbol{y}}(f)\right)\, [A_{\#}\mu_\delta^{\boldsymbol{y}}](\mathrm{d}\tilde{\boldsymbol{y}}) \right| \\
&\leq \int_{\mathcal{Y}} \left|\mu^{\tilde{\boldsymbol{y}}}(f) - \mu^{\boldsymbol{y}}(f)\right|\, [A_{\#}\mu_\delta^{\boldsymbol{y}}](\mathrm{d}\tilde{\boldsymbol{y}}) \\
&\leq C_\mu \int_{\mathcal{Y}} \|\tilde{\boldsymbol{y}} - \boldsymbol{y}\|_{\mathcal{Y}}\, [A_{\#}\mu_\delta^{\boldsymbol{y}}](\mathrm{d}\tilde{\boldsymbol{y}}) \qquad \text{(E.1)}
\end{aligned}$$

where the last line follows from application of Assumption 6.2.4. From this it follows that

$$\begin{aligned} d_{\mathcal{F}}(\mu^{\boldsymbol{y}}_\delta, \mu^{\boldsymbol{y}}) &= \sup_{f\in\mathcal{F}} |\mu^{\boldsymbol{y}}_\delta(f) - \mu^{\boldsymbol{y}}(f)| \\ &\leq C_\mu \int_{\mathcal{Y}} \|\tilde{\boldsymbol{y}} - \boldsymbol{y}\|_{\mathcal{Y}} \, [A_{\#}\mu^{\boldsymbol{y}}_\delta](\mathrm{d}\tilde{\boldsymbol{y}}) \end{aligned}$$

Now, consider the function

$$g^{\boldsymbol{y}}_\delta(u) := \frac{\|A(u) - \boldsymbol{y}\|_{\mathcal{Y}}}{\delta}$$

Let $\nu_r = [g^{\boldsymbol{y}}_\delta]_{\#}\nu$. From Assumption 6.2.2, $A(X)$ has a Lipschitz Lebesgue density on $\mathcal{Y}$. Thus, ν_r also has a continuous and positive Lebesgue density on $[0, \infty)$, denoted $p_{r,\delta}$. Thus

$$\nu_r([a, b]) = \int_a^b p_{r,\delta}(r) \, \mathrm{d}r.$$

Furthermore by the definition of the pushforward measure, $\nu_r(B) = \nu([g^{\boldsymbol{y}}_\delta]^{-1}(B))$ for each B in the σ-algebra associated with ν_r (note that this is given by the image of the σ-algebra $\mathcal{B}_{\mathcal{X}}$ under $g^{\boldsymbol{y}}_\delta \circ A$). Thus

$$\begin{aligned} \nu_r([a, b]) &= \int_a^b \int_{S_{\delta r}(\boldsymbol{y})} p_A(\boldsymbol{y}') \, \mathrm{d}\boldsymbol{y}' \, \mathrm{d}r \\ \implies p_{r,\delta}(r) &= \int_{S_{\delta r}(\boldsymbol{y})} p_A(\boldsymbol{y}') \, \mathrm{d}\boldsymbol{y}' \end{aligned}$$

where $S_r(\boldsymbol{y})$ denotes the surface of a sphere of radius r centered at $\boldsymbol{y}$. Since p_A is positive and Lipschitz we then have

$$\begin{aligned} p_{r,\delta}(r) &= \int_{S_{\delta r}(\boldsymbol{y})} |p_A(\boldsymbol{y}) + p_A(\boldsymbol{y}') - p_A(\boldsymbol{y})| \, \mathrm{d}\boldsymbol{y}' \\ &\leq \int_{S_{\delta r}(\boldsymbol{y})} \big(|p_A(\boldsymbol{y})| + |p_A(\boldsymbol{y}') - p_A(\boldsymbol{y})|\big) \, \mathrm{d}\boldsymbol{y}' \\ &\leq \int_{S_{\delta r}(\boldsymbol{y})} \big(p_A(\boldsymbol{y}) + L_A\|\boldsymbol{y} - \boldsymbol{y}'\|_{\mathcal{Y}}\big) \, \mathrm{d}\boldsymbol{y}' \\ &= (p_A(\boldsymbol{y}) + L_A\delta r) \, \mathrm{SR}(\delta r) \end{aligned}$$

where $\mathrm{SR}(r)$ is the surface area of a $\|\cdot\|_{\mathcal{Y}}$-sphere of radius r. Recall that we have $\mathrm{SR}(r) = r^{n-1}\mathrm{SR}(1)$. We can similarly elicit a lower bound:

$$\begin{aligned} p_{r,\delta}(r) &= \int_{S_{\delta r}(\boldsymbol{y})} |p_A(\boldsymbol{y}) + p_A(\boldsymbol{y}') - p_A(\boldsymbol{y})| \,\mathrm{d}\boldsymbol{y}' \\ &\geq \int_{S_{\delta r}(\boldsymbol{y})} \big| p_A(\boldsymbol{y}) - |p_A(\boldsymbol{y}') - p_A(\boldsymbol{y})| \big| \,\mathrm{d}\boldsymbol{y}' \end{aligned}$$

where the second line is from the reverse triangle inequality. Now note that using the fact that p_A is Lipschitz, we have that whenever $\boldsymbol{y}' \in S_{\delta r}(\boldsymbol{y})$:

$$|p_A(\boldsymbol{y}') - p_A(\boldsymbol{y})| \leq L_A \|\boldsymbol{y} - \boldsymbol{y}'\|_{\mathcal{Y}} = L_A \delta r$$

where the second equality follows from the fact that $\boldsymbol{y}'$ lies on the sphere $S_{\delta r}(\boldsymbol{y})$ so its distance from $\boldsymbol{y}$ is known. Now, let $M(P) = \inf\{\frac{p_A(\boldsymbol{y})}{L_A} : \|\boldsymbol{y}\|_{\mathcal{Y}} < P\}$ and note that $M(P) > 0$ from Assumption 6.2.2, since $p_A(\boldsymbol{y}) > 0$ by assumption for all $\boldsymbol{y} \in \mathcal{Y}$. Then we have that whenever $r < M(P)/\delta$

$$\begin{aligned} p_{r,\delta}(r) &\geq \int_{S_{\delta r}(\boldsymbol{y})} (p_A(\boldsymbol{y}) - L_A \delta r) \,\mathrm{d}r \\ &= (p_A(\boldsymbol{y}) - L_A(\delta r))\mathrm{SR}(\delta r). \end{aligned}$$

We then proceed to elicit a bound on Eq. (E.1):

$$\int_{\mathcal{Y}} \|\tilde{\boldsymbol{y}} - \boldsymbol{y}\|_{\mathcal{Y}} A_{\#}\mu_\delta^a(\mathrm{d}\tilde{\boldsymbol{y}}) = \frac{\int_{\mathcal{Y}} \|\tilde{\boldsymbol{y}} - \boldsymbol{y}\|_{\mathcal{Y}} \phi\left(\frac{\|\tilde{\boldsymbol{y}} - \boldsymbol{y}\|_{\mathcal{Y}}}{\delta}\right) \nu(\mathrm{d}\tilde{\boldsymbol{y}})}{\int_{\mathcal{Y}} \phi\left(\frac{\|\tilde{\boldsymbol{y}} - \boldsymbol{y}\|_{\mathcal{Y}}}{\delta}\right) \nu(\mathrm{d}\tilde{\boldsymbol{y}})}$$

Letting $r = \frac{1}{\delta}\|\tilde{\boldsymbol{y}} - \boldsymbol{y}\|_{\mathcal{Y}}$:

$$\begin{aligned} &\int_{\mathcal{Y}} \|\tilde{\boldsymbol{y}} - \boldsymbol{y}\|_{\mathcal{Y}} A_{\#}\mu_\delta^a(\mathrm{d}\tilde{\boldsymbol{y}}) \\ &\quad = \frac{\int_0^\infty \delta r \phi(r) p_{r,\delta}(r) \,\mathrm{d}r}{\int_0^\infty \phi(r) p_{r,\delta}(r) \,\mathrm{d}r} \\ &\quad \leq \frac{\int_0^\infty \delta r \phi(r) \mathrm{SR}(\delta r)(p_A(\boldsymbol{y}) + L_A \delta r) \mathrm{d}r}{\int_0^{M(P)/\delta} \phi(r)(p_A(\boldsymbol{y}) - L_A(\delta r))\mathrm{SR}(\delta r) \,\mathrm{d}r + \int_{M(P)/\delta}^\infty \phi(r) p_{r,\delta}(r) \,\mathrm{d}r} \end{aligned}$$

Now, since both ϕ and $p_{r,\delta}$ are strictly positive, the denominator can be lower-bounded by $\int_0^{M(P)/\delta} \phi(r) p_{r,\delta}(r)\, \mathrm{d}r$. We obtain:

$$\begin{aligned}\int_{\mathcal{Y}} \|\tilde{\boldsymbol{y}} - \boldsymbol{y}\|_{\mathcal{Y}} A_{\#}\mu_\delta^a(\mathrm{d}\tilde{\boldsymbol{y}}) &\leq \frac{\int_0^\infty \delta r \phi(r) \mathrm{SR}(\delta r)(p_A(\boldsymbol{y}) + L_A \delta r)\mathrm{d}r}{\int_0^{M(P)/\delta} \phi(r) p_{r,\delta}(r)\, \mathrm{d}r} \\ &\leq \frac{\delta \mathrm{SR}(\delta) \int_0^\infty r^n \phi(r)(p_A(\boldsymbol{y}) + L_A \delta r)\, \mathrm{d}r}{\mathrm{SR}(\delta) \int_0^{M(P)/\delta} r^{n-1} \phi(r)(p_A(\boldsymbol{y}) - L_A \delta r)\, \mathrm{d}r} \\ &= \delta \frac{\int_0^\infty r^n \phi(r)(p_A(\boldsymbol{y}) + L_A \delta r)\, \mathrm{d}r}{\int_0^{M(P)/\delta} r^{n-1} \phi(r)(p_A(\boldsymbol{y}) - L_A \delta r)\, \mathrm{d}r}. \qquad \text{(E.2)}\end{aligned}$$

Now note that in the limit as $\delta \to 0$, we have that

$$\begin{aligned}\int_0^\infty r^n \phi(r)(p_A(\boldsymbol{y}) + L_A \delta r)\, \mathrm{d}r &\to p_A(\boldsymbol{y}) \int_0^\infty r^n \phi(r) \mathrm{d}r = p_A(\boldsymbol{y}) C_\phi^n \\ \int_0^{M(P)/\delta} r^{n-1} \phi(r)(p_A(\boldsymbol{y}) - L_A \delta r)\, \mathrm{d}r &\to p_A(\boldsymbol{y}) \int_0^\infty r^{n-1} \phi(r) \mathrm{d}r = p_A(\boldsymbol{y}) C_\phi^{n-1}\end{aligned}$$

from Assumption 6.2.3. Thus the ratio in Eq. (E.2) is such that

$$\frac{\int_0^\infty r^n \phi(r)(p_A(\boldsymbol{y}) + L_A \delta r)\, \mathrm{d}r}{\int_0^{M(P)/\delta} r^{n-1} \phi(r)(p_A(\boldsymbol{y}) - L_A \delta r)\, \mathrm{d}r} \xrightarrow[\delta \downarrow 0]{} \frac{C_\phi^n}{C_\phi^{n-1}}.$$

Thus, for δ sufficiently small, (E.2) can be bounded above by $\delta(\epsilon + \bar{C}_\phi^n)$ where $\bar{C}_\phi^n := C_\phi^n / C_\phi^{n-1}$ and ϵ is an arbitrary positive constant. Arbitrarily taking $\epsilon = 1$ establishes that

$$|\mu_\delta^a(f) - \mu^a(f)| \leq C_\mu (1 + \bar{C}_\phi^n)\, \delta$$

for δ sufficiently small, completing the proof. □