# Bayesian Probabilistic Numerical Methods for Ordinary and Partial Differential Equations

Junyang Wang

Thesis submitted for the degree of
Doctor of Philosophy


*School of Mathematics, Statistics & Physics*
*Newcastle University*
*Newcastle upon Tyne*
*United Kingdom*

June 2021

# Declaration

I declare that this thesis has been composed by myself and that the work has not been submitted for any other degree or professional qualification. I confirm that the work submitted is my own, except where work which has formed part of jointly authored publications has been included. The specific details are described below. In all cases the experimental (coding) and theoretical results contained in the thesis are my own work; the contribution of co-authors was limited to the initial design the research project and to the interpretation of the results that were reported.

This thesis contains content in the paper *A Role for Symmetry in the Bayesian Solution of Differential Equations*, published in the journal *Bayesian Analysis*, jointly written by myself (first author), Jon Cockayne and Chris Oates, as well as the paper *Bayesian Numerical Methods for Nonlinear Partial Differential Equations*, jointly written by myself (first author), Jon Cockayne, Oksana Chkrebtii, Tim Sullivan and Chris Oates, published in the journal *Statistics and Computing*. The content in this thesis relevant to *A Role for Symmetry in the Bayesian Solution of Differential Equations* is mostly contained in Chapter 4, and the content in this thesis relevant to *Bayesian Numerical Methods for Nonlinear Partial Differential Equations* in Chapter 5. The Introduction Chapter 1 and the literature review Chapter 3 also contains material from *A Role for Symmetry in the Bayesian Solution of Differential Equations*.

**Acknowledgements**

First and foremost I would like to thank my main supervisor Chris Oates for providing me with invaluable support and advice throughout my PhD and for never losing patience with me when I struggled. I have learnt a great deal about Bayesian Statistics as well as about research and academia in general from Chris.

I am extremely grateful for The EPSRC Centre for Doctoral Training in Cloud Computing for Big Data for providing the funding for my PhD. I would like to thank Paul Watson and Darren Wilkinson for giving me the opportunity of the PhD as well as their excellent management of the CDT. I also am very appreciative of the CDT administrative staff for organising various CDT events which made my PhD much more enjoyable, as well as for providing me with administrative support when I needed it. I would also like to thank the other CDT students for making my PhD experience more lively and enjoyable.

I would like to thank my supervisor Chris Oates again for his invaluable feedback on this thesis and the manuscripts I submitted during the PhD. I would also like to thank the other co-authors of the two journal papers I submitted during my PhD, Jon Cockayne, Oksana Chkrebtti and Tim Sullivan, for the useful discussions and constructive criticisms on my results and proofreading of the manuscript. In addition for the contents of Chapter 4 (which has been published in *A Role for Symmetry in the Bayesian Solution of Differential Equations*), I am grateful to Mark Craddock, François-Xavier Briol and Tim Sullivan for discussion of this work, as well as to an Associate Editor and two Reviewers from the journal Bayesian Analysis for their challenging but constructive feedback. I would also like to thank Michael Schober on useful advice on thesis writing.

Lastly, I would like to thank my parents, Jia Yang and Fude Wang, for providing me with emotional support and encouragement throughout my PhD.

**Abstract**

Differential equations provide an important mathematical framework for modelling the behaviour of quantities that evolve in continuous time and space, often within a complex system or process. Many physical phenomena, including fundamental laws such as Newton's laws of motion, are formulated as differential equations. However, most useful differential equations lack a closed form solution expressible in terms of established functions, and so in practice numerical methods are required to obtain a discrete approximation to quantities of interest.

Classical numerical methods approximate quantities of interest by taking a finite number of evaluations from some known and computationally tractable quantity, such as the gradient field, and use these within an algorithm to construct an approximation. This is similar to statistics, where a finite number of observations of some unknown, underlying process are used to infer the process itself. In this view, numerical algorithms can be interpreted as estimators, and statistical considerations can be brought to bear. Going further, on can consider probabilistic numerical methods, which output a probability distribution over the quantity of interest. In recent years, this idea has emerged into a new field of research, called Probabilistic Numerics.

In the first part of this thesis, an exact Bayesian probabilistic numerical method for ordinary differential equations (ODEs) is presented. The method is a synthesis of classical Lie group theory, to exploit underlying symmetries in the gradient field, and non-parametric regression in a transformed solution space for the ODE. The procedure is presented in detail for first and second order ODEs and relies on a certain strong technical condition – existence of a solvable Lie algebra – being satisfied. Numerical illustrations are provided for nonlinear first and second order ODEs. However, the ability to perform exact Bayesian inference comes at a high price, because the class of ODEs that admit a solvable Lie algebra is limited.

In the second part of this thesis, an approximate Bayesian probabilistic numerical method for nonlinear partial differential equations (PDEs) is presented. A Bayesian treatment of nonlinear PDEs does not yet exist, as the case of nonlinear PDEs poses substantial challenges from an inferential perspective, most notably due to the absence of explicit conditioning formula. This thesis extends earlier work on linear PDEs to a general class of initial value problems specified by nonlinear PDEs. Numerical experiments are conducted on a range of examples, and indicate the proposed method is able to provide meaningful probabilistic uncertainty quantification for the unknown solution of the PDE, while

controlling the number of times the right-hand-side of the PDE is evaluated. This is practically useful in situations where evaluation of the right-hand-side of the PDE is associated with a high computational cost.

The nascent field of Probabilistic Numerics is receiving increased attention, but fundamental questions remain regarding aims and scope of the field. The contributions of this thesis, while limited to proofs of concept, are helpful in clarifying a role for Bayesian statistics in the probabilistic solution of differential equations. The thesis concludes with a discussion, which is broadly supportive of taking a Bayesian approach to differential equations, whilst highlighting where exact Bayesian inference may not be achievable and suggesting approximation strategies in that context.

# Contents

# List of Figures

# Chapter 1

# Introduction

Differential Equations are mathematical equations which model the relationship between physical quantities of interest and their rates of change with respect to independent variables, such as time. Since the development of calculus by Newton and Leibniz in the 17th century, differential equations have become ubiquitous in many scientific fields and disciplines, including physics, engineering, biology, economics and finance. Prominent examples include: Newton's second law of motion, which famously states that the rate of change of momentum $\mathbf{p}$ of a body over time is equal to the net force $\mathbf{F}$ applied on the body:

$$\mathbf{F} = \frac{\mathrm{d}\mathbf{p}}{\mathrm{d}t}$$

Schrodinger's Equation in quantum mechanics:

$$i\hbar\frac{\partial \mathbf{\Psi}}{\partial t} = \hat{H}\mathbf{\Psi}$$

This equation describes the relationship between the wave function $\mathbf{\Psi}$, the Hamiltonian operator $\hat{H}$, which is also typically a differential operator on spatial variables. The logistic growth equation, used to model population growth in biology:

$$\frac{\mathrm{dN}}{\mathrm{d}t} = r\mathrm{N}\left(1 - \frac{\mathrm{N}}{K}\right)$$

The Black-Scholes equation (for a European call or put on an underlying stock paying no dividends):

$$\frac{\partial \mathrm{V}}{\partial t} = \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 \mathrm{V}}{\partial S^2} + rS\frac{\partial \mathrm{V}}{\partial S} - r\mathrm{V} = 0$$

where $V$ is the price of the option as a function of stock price $S$ and time $t$, $r$ is the risk-free interest rate, and $\sigma$ the volatility of the stock. While differential equations are able to provide the framework in which many physical phenomena or abstract quantities are modelled, they also bring forward new challenges. In particular, most differential equations do not have a closed form solution and therefore cannot be solved by hand symbolically. To combat this, mathematicians have developed numerical methods, algorithms that rely on numerical approximation as opposed to symbolic manipulation, in order to solve differential equations. Some of the major achievements in this area include the Runge–Kutta method and the Galerkin method. Classical numerical methods for differential equations produce an approximation to the solution of the differential equation whose error (called *numerical error*) is uncertain in general.

Suppose we have some unknown quantity of interest $\mathrm{y}^\dagger$, which could for example be the unknown solution to a differential equation. A numerical task $Q$ can be formulated as the approximation of a quantity of interest

$$Q : \mathcal{Y} \quad \to \quad \mathcal{Q},$$

subject to a finite computational budget. The true underlying state $\mathrm{y}^\dagger \in \mathcal{Y}$ is typically high- or infinite-dimensional, so that only limited information

$$A : \mathcal{Y} \quad \to \quad \mathcal{A} \tag{1.1}$$

is provided and exact computation of $Q(\mathrm{y}^\dagger)$ is prohibited. For example, numerical integration aims to approximate an integral $Q(\mathrm{y}^\dagger) = \int \mathrm{y}^\dagger(t)\mathrm{d}t$ given the values $A(\mathrm{y}^\dagger) = \{(x_i, \mathrm{y}^\dagger(x_i))\}_{i=1}^n$ of the integrand $\mathrm{y}^\dagger$ on a finite number of abscissa $\{x_i\}_{i=1}^n$. Similarly, a numerical approximation to the solution $Q(\mathrm{y}^\dagger) = \mathrm{y}^\dagger$ of a differential equation $\mathrm{d}y/\mathrm{d}x = f(x, \mathrm{y}(x))$, $\mathrm{y}(x_0) = y_0$, will typically be based on a finite number of evaluations of $f$, the gradient field. In this viewpoint a numerical method corresponds to a map $b : \mathcal{A} \to \mathcal{Q}$, as depicted in Figure 1.1a, where $b(a)$ represents an approximation to the solution of the differential equation based on the information $a \in \mathcal{A}$.

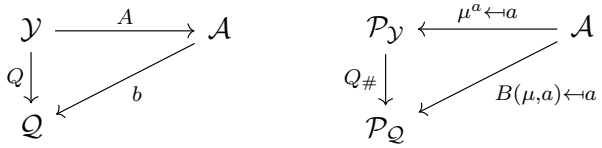
The increasing ambition and complexity of contemporary applications is such that the computational budget can be *extremely* small compared to the precision that is required at the level of the quantity of interest. As such, in many important problems it is not possible to reduce the numerical error to a negligible level. Fields acutely associated with this challenge include climate forecasting (Wedi, 2014), computational cardiology

(Chabiniok *et al.*, 2016) and molecular dynamics (Perilla *et al.*, 2015). In the presence of non-negligible numerical error, it can often be unclear how scientific interpretation of the output of computation can proceed. For example, *a posteriori* analysis of traditional numerical methods can be used to establish hard upper bounds on the numerical error, but these bounds typically depend on an unknown global constant. In the case of ODEs, this may be the maximum value of a norm $\|f\|$ of the gradient field (see e.g. Estep, 1995). If $\|f\|$ were known, it would be possible to provide a hard bound on numerical error. However, in the typical numerical context where $f$ is a black box, it may only be known is that $\|f\| < \infty$. One could attempt to approximate $\|f\|$ with cubature, but that itself requires a numerical cubature method whose error is required to obey a known bound. In general, therefore, there are no hard error bounds without global information on the task at hand being *a priori* provided (Larkin, 1974).

## 1.1 Probabilistic Numerical Methods

Probability theory provides a natural language in which uncertainty can be expressed and, since the solution of a differential equation is unknown, it is interesting to ask whether probability theory can be applied to quantify *numerical uncertainty* associated with it. This perspective, in which numerical tasks are cast as problems of statistical inference, is pursued in the nascent field of *probabilistic numerics*. The field of probabilistic numerics dates back to Larkin (1972) and a modern perspective is provided in Hennig *et al.* (2015); Oates & Sullivan (2019). Under the abstract framework just described, numerical methods can be interpreted as point estimators in a statistical context, where the state $\mathrm{y}^\dagger$ can be thought of as a latent variable in a statistical model, and the 'data' consist of information $A(\mathrm{y}^\dagger)$ that does not fully determine the quantity of interest $Q(\mathrm{y}^\dagger)$ but is indirectly related to it. Hennig *et al.* (2015) provide an accessible introduction and survey of the field. In particular, they illustrated how PNM can be used to quantify uncertainty due to discretisation in important scientific problems, such as astronomical imaging.

Let the notation $\Sigma_{\mathcal{Y}}$ denote a $\sigma$-algebra on the space $\mathcal{Y}$ and let $\mathcal{P}_{\mathcal{Y}}$ denote the set of probability measures on $(\mathcal{Y}, \Sigma_{\mathcal{Y}})$. A probabilistic numerical method (PNM) is a procedure which takes as input a 'belief' distribution $\mu \in \mathcal{P}_{\mathcal{Y}}$, representing epistemic uncertainty with respect to the true (but unknown) value $\mathrm{y}^\dagger$, along with a finite amount of information, $A(\mathrm{y}^\dagger) \in \mathcal{A}$. The output is a distribution $B(\mu, A(\mathrm{y}^\dagger)) \in \mathcal{P}_{\mathcal{Q}}$ on $(\mathcal{Q}, \Sigma_{\mathcal{Q}})$, representing epistemic uncertainty with respect to the quantity of interest $Q(\mathrm{y}^\dagger)$ after the information $A(\mathrm{y}^\dagger)$ have been processed. For example, a PNM for an ordinary differential equation (ODE) takes an initial belief distribution defined on the solution space of the differential equation, together with information arising from a finite number of evaluations of the gradient field, plus the initial condition of the ODE, to produce a distribution over either




(a) (b)

Figure 1.1: Diagrams for a numerical method. (a) The traditional viewpoint of a numerical method is equivalent to a map $b$ from a finite-dimensional information space $\mathcal{A}$ to the space of the quantity of interest $\mathcal{Q}$. (b) The probabilistic viewpoint treats approximation of $Q(\mathrm{y}^\dagger)$ in a statistical context, described by a map $B(\mu, \cdot)$ from $\mathcal{A}$ to the space of probability distributions on $\mathcal{Q}$. The probabilistic numerical method $(A, B)$ is Bayesian if and only if (b) is a commutative diagram.

the solution space of the ODE, or perhaps some derived quantity of interest. In this thesis, the measurability of $A$ and $Q$ will be assumed.

Despite computational advances in this emergent field, until recently there had not been an attempt to establish rigorous statistical foundations for PNM. In Cockayne *et al.* (2019) the authors argued that Bayesian principles can be adopted. In brief, this framework requires that the input belief distribution $\mu$ carries the semantics of a Bayesian agent's prior belief, and that the output of a PNM agrees with the inferences drawn when the agent is rational. To be more precise recall that, in this thesis, information is provided in a deterministic[1] manner through (1.1) and thus Bayesian inference corresponds to conditioning of $\mu$ on the level sets of $A$. Let $Q_\# : \mathcal{P}_\mathcal{Y} \to \mathcal{P}_\mathcal{Q}$ denote the push-forward map associated to $Q$. i.e. $Q_\#(\mu)(S) = \mu(Q^{-1}(S))$ for all $S \in \Sigma_\mathcal{Q}$. Let $\{\mu^a\}_{a\in\mathcal{A}} \subset \mathcal{P}_\mathcal{Y}$ denote the disintegration, assumed to exist[2], of $\mu \in \mathcal{P}_\mathcal{Y}$ along the map $A$.

**Definition 1.** A probabilistic numerical method $(A, B)$ with $A : \mathcal{Y} \to \mathcal{A}$ and $B : \mathcal{P}_\mathcal{Y} \times \mathcal{A} \to \mathcal{P}_\mathcal{Q}$ for a quantity of interest $Q : \mathcal{Y} \to \mathcal{Q}$ is *Bayesian* if and only if $B(\mu, a) = Q_\#(\mu^a)$ for all $\mu \in \mathcal{P}_\mathcal{Y}$ and all $a \in \mathcal{A}$.

This definition is intuitive; the output of the PNM should coincide with the marginal distribution for $Q(\mathrm{y}^\dagger)$ according to the disintegration element $\mu^a \in \mathcal{P}_\mathcal{Y}$, based on the information $a \in \mathcal{A}$ that was provided. The definition is equivalent to the statement that Figure 1.1b is a commutative diagram. In Cockayne *et al.* (2019) the map $A$ was termed an *information operator* and the map $B$ was termed a *belief update operator*; we adhere to these definitions in our work. The Bayesian approach to PNM confers several important benefits:

[1] It is of course possible to perform Bayesian inference in the noisy-data context, but for the ODEs considered in this thesis we assume that one can obtain noiseless evaluations of the gradient field.

[2] The reader unfamiliar with the concept of a disintegration can treat $\mu^a$ as a technical notion of the 'conditional distribution of y given $A(\mathrm{y}) = a$' when reading this work. The *disintegration theorem*, Thm. 1 of Chang & Pollard (1997), guarantees existence and uniqueness up to a $A_\#\mu$-null set under the weak requirement that $\mathcal{Y}$ is a metric space, $\Sigma_\mathcal{Y}$ is the Borel $\sigma$-algebra, $\mu$ is Radon, $\Sigma_\mathcal{A}$ is countable generated and $\Sigma_\mathcal{A}$ contains all singletons $\{a\}$ for $a \in \mathcal{A}$.

- The input $\mu$ and output $B(\mu, a)$ belief distributions can be interpreted, respectively, as a *prior* and (marginal) *posterior*.[3] As such, they automatically inherit the stronger formal semantics and philosophical foundations that underpin the Bayesian framework and, in this sense, are well-understood (see e.g. Gelman & Shalizi, 2013).

- The definition of Bayesian PNM is operational. Thus, if we are presented with a prior $\mu$ and information $a$ then there is a unique Bayesian PNM and it is constructively defined.

- The modern perspective on uncertainty quantification is to consider all relevant sources of uncertainty, such as discretisation error, parameter uncertainty, uncertainty due to measurement error and uncertainty due to model mis-specification. The different types of uncertainty are then integrated into inferences and predictions. The class of Bayesian PNM is closed under composition, such that uncertainty due to different sources of discretisation can be jointly modelled and rigorously propagated.

Nevertheless, the strict definition of Bayesian PNM limits scope to design convenient computational algorithms and indeed several proposed PNM are not Bayesian, including all previously proposed PNMs on differential equations. An in depth discussion of the existing PNM methods on differential equations is presented in chapter 3. The challenge is two-fold; for a Bayesian PNM, the elicitation of an appropriate prior distribution $\mu$ and the exact computation of its disintegration $\{\mu^a\}_{a\in\mathcal{A}}$ must both be addressed.

This thesis is concerned with the development of novel PNMs for differential equations and will be structured in five parts. In Part II, Chapter 2, an informal introduction of the main mathematical tools used in the thesis, Ordinary Differential Equations, Partial Differential Equations and Gaussian Processes, will be presented. In Chapter 3 a high-level overview of existing PN methods for differential equations will be presented, and we will argue a strictly Bayesian PNM for the the numerical solution of an ODE does not yet exist.

In Part III, a novel Bayesian PNM for ODEs is proposed as a proof-of-concept. The proposed Bayesian PNM utilises classical Lie group methods to exploit underlying symmetries in the gradient field, and non-parametric regression in a transformed solution space for the ODE. The procedure is presented in detail for first and second order ODEs and relies on a certain strong technical condition – existence of a solvable Lie algebra – being satisfied. The procedure is applied on example first and second order ODEs and numerical results are presented. An overview of classical Lie group methods used in the proposed Bayesian PNM is also provided.

[3]Indeed, if the set $\mathcal{Y}^a = \{\mathrm{y} \in \mathcal{Y} : A(\mathrm{y}) = a\}$ is not measure zero under $\mu$, then $\mu^a$ is the conditional distribution defined by restricting $\mu$ to the subset $\mathcal{Y}^a$; $\mu^a(\mathrm{y}) = 1[\mathrm{y} \in \mathcal{Y}^a]\mu(\mathrm{y})/\mu(\mathcal{Y}^a)$. The theory of disintegrations generalises the conditional distribution $\mu^a$ to cases where $\mathcal{Y}^a$ is a null set.

In Part IV, we present the first approximate Bayesian PNM for the numerical solution of nonlinear PDEs. The proposed method extends earlier work on linear PDEs to a general class of initial value problems specified by nonlinear PDEs. The proposed method can be viewed as exact Bayesian inference under an approximate likelihood, which is based on discretisation of the nonlinear differential operator. Experimental results are presented for example PDEs and performance is contrasted with the Crank–Nicholson scheme, a classical finite difference method under limited evaluations of the right-hand-side of the PDE, motivated by problems for which evaluation of the right-hand-side, initial or boundary conditions of the PDE is associated with a high computational cost. Theoretical analysis of the sample path properties of Matérn processes, which are used to determine a suitable prior model for the solution of the PDE, are also presented.

In the final chapter of the thesis, we will reflect upon the contributions this thesis has made to probabilistic numerics as well as the numerical solution of differential equations, and suggest potential future research directions.

# Chapter 2

# Background

The purpose of this chapter is to provide an introduction to the three main mathematical topics examined in the thesis: ordinary differential equations, partial differential equations and stochastic processes, with a particular emphasis on Gaussian Processes. Since the methods developed in this thesis are probabilistic solvers of differential equations, stochastic processes are the natural mathematical objects of choice for the solution of a probabilistic numerical method. The discussions in this section are intended to be informal and accessible, and in later sections we will introduce more in depth concepts and theorems when required.

## 2.1 Ordinary Differential Equations

An ordinary differential equation (ODE) is an equation involving a *state* variable $\mathbf{y} : \Gamma \to \mathbb{R}^m$, where $\Gamma$ is a closed interval in $\mathbb{R}$, and its derivatives with respect to a single variable $x \in \Gamma$. The *order* of a differential equation is the highest number of times $\mathbf{y}$ is differentiated. An *explicit*, $1st$ order ODE can be written in the form:

$$\frac{\mathrm{d}\mathbf{y}}{\mathrm{d}x} = \mathbf{f}(x, \mathbf{y}(x)) \tag{2.1}$$

The term $\mathbf{f}(x, \mathbf{y}(x))$ is often referred to as the *gradient field* of the differential equation. A function $\mathbf{y}^\dagger : \Gamma \to \mathbb{R}^m$ that satisfies 2.1 is known as a *solution* of 2.1. In general the existence of a solution is not guaranteed, and when a solution exists it's not necessarily unique. Fortunately, it turns out for ODEs, the existence and uniqueness of a solution is well understood:

**Theorem 2.1** (Picard-Lindelöf existence theorem)**.** *Let $\boldsymbol{y}_0 \in \mathbb{R}^m, R > 0, a < b, x_0 \in [a, b]$. Let $\overline{B_R(\boldsymbol{y}_0)}$ denote the closed m-sphere of radius $R$ around $\boldsymbol{y}_0$, i.e.:*

$$\overline{B_R(\boldsymbol{y}_0)} = \{\boldsymbol{y} \in \mathbb{R}^m : \|\boldsymbol{y} - \boldsymbol{y}_0\|_2 \leq R\}$$

*Let* $\mathbf{f} : [a, b] \times \overline{B_R(\boldsymbol{y}_0)} \to \mathbb{R}^m$ *be a continuous function that satisfies Lipschitz continuity in the second variable, meaning:*

$$\|\mathbf{f}(x, \boldsymbol{y}) - \mathbf{f}(x, \boldsymbol{z})\|_2 \leq L\|\boldsymbol{y} - \boldsymbol{z}\|_2$$

*for some fixed constant* $L > 0$ *and all* $x \in [a, b]$, $\boldsymbol{y}, \boldsymbol{z} \in \overline{B_R(\mathbf{y}_0)}$. *Then there exists an* $\epsilon > 0$ *and an unique differentiable function* $\mathbf{y}^\dagger : [x_0 - \epsilon, x_0 + \epsilon] \cap [a, b] \to \mathbb{R}^m$ *such that:*

$$\frac{\mathrm{d}\mathbf{y}^\dagger}{\mathrm{d}x} = \mathbf{f}(x, \mathbf{y}^\dagger(x))$$

*and* $\mathbf{y}^\dagger(x_0) = \boldsymbol{y_0}$

*Proof.* The proof is omitted as this is a well known result, which for example can be found in (Hartman, 1982).

□

Picard-Lindelöf guarantees both existence and uniqueness of a solution (at least locally) under relatively mild smoothness conditions imposed on the gradient field, and the addition of an initial or boundary condition. Without a specified initial or boundary condition, an ODE may have an infinite number of solutions that can be parameterised in terms of constants of integration, known as a *general solution*. When an initial or boundary condition is specified and a unique solution existed, it is referred to as a *particular solution*.

Of course, an ODE could include more than just the first derivative, in which case an *explicit*, *nth* order ODE can be written in the form:

$$\frac{\mathrm{d}^\mathrm{n}\mathbf{y}}{\mathrm{d}x^n} = \mathbf{f}\left(x, \mathbf{y}(x), \frac{\mathrm{d}\mathbf{y}}{\mathrm{d}x}, \ldots, \frac{\mathrm{d}^{\mathrm{n}-1}\mathbf{y}}{\mathrm{d}x^{n-1}}\right) \tag{2.2}$$

Note setting $\tilde{\mathbf{y}} = [\mathbf{y}, \frac{\mathrm{d}\mathbf{y}}{\mathrm{d}x}, \frac{\mathrm{d}^2\mathbf{y}}{\mathrm{d}x^2}, \ldots, \frac{\mathrm{d}^{\mathrm{n}-1}\mathbf{y}}{\mathrm{d}x^{n-1}}]^\top$, we can reformulate the problem in terms of $\tilde{\mathbf{y}}$:

$$\frac{\mathrm{d}}{\mathrm{d}x}\tilde{\mathbf{y}} = \frac{\mathrm{d}}{\mathrm{d}x}\begin{bmatrix} \mathbf{y} \\ \frac{\mathrm{d}\mathbf{y}}{\mathrm{d}x} \\ \vdots \\ \frac{\mathrm{d}^{\mathrm{n}-1}\mathbf{y}}{\mathrm{d}x^{n-1}} \end{bmatrix} = \begin{bmatrix} \frac{\mathrm{d}\mathbf{y}}{\mathrm{d}x} \\ \frac{\mathrm{d}^2\mathbf{y}}{\mathrm{d}x^2} \\ \vdots \\ \mathbf{f}(x, \mathbf{y}(x), \frac{\mathrm{d}\mathbf{y}}{\mathrm{d}x}, \ldots, \frac{\mathrm{d}^{\mathrm{n}-1}\mathbf{y}}{\mathrm{d}x^{n-1}}) \end{bmatrix} \tag{2.3}$$

which is of the form 2.1. This means Theorem 3 can be applied to higher order ODEs as well. An important classification of ODEs is linearity. An ODE 2.1 is *linear* if the

dependent variable $\mathbf{y}$ and its derivatives appears only linearly. A first order linear ODE can be written in the form:

$$\frac{\mathrm{d}\mathbf{y}}{\mathrm{d}x} = A(x)\mathbf{y}(x) + \mathbf{b}(x) \tag{2.4}$$

where $A(x)$ is some known $m \times m$ matrix of functions of $x$, and $\mathbf{b} : \Gamma \in \mathbb{R} \to \mathbb{R}^m$ also a known function. Equation 2.4 is called *homogeneous* if $\mathbf{b}(x) \equiv 0$. In general it's much easier to obtain closed form solutions of linear ODEs than nonlinear ones. For example, it's well known that the general 2nd order homogeneous ODE with constant coefficients:

$$a\frac{\mathrm{d}^2\mathrm{y}}{\mathrm{d}x^2} + b\frac{\mathrm{dy}}{\mathrm{d}x} + c\mathrm{y} = 0 \tag{2.5}$$

has general solution

$$y = a_1\exp(\lambda_1 x) + a_2\exp(\lambda_2 x)$$

where $\lambda_1$ and $\lambda_2$ are the roots of the polynomial $ax^2 + bx + c = 0$, and $a_1, a_2$ are any constants.

### 2.1.1 Numerical Solution of ODEs

However, closed form solutions of ODEs are rarely obtainable. Even just replacing the constant coefficients in 2.5 with arbitrary functions of x while still retaining linearity, causes a closed form solution to not be available in general. So often in practice it is necessary to resort to *numerical solutions*. The idea of a *numerical method* in solving a differential equation is to find an approximation to the true solution $\{\boldsymbol{y}_0, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_{n-1}\}$ on a discrete set of values of $\{x_0, x_1, \ldots, x_{n-1}\}$ in the ODE's domain. Many numerical methods for ODEs can be motivated by the simple observation that:

$$\mathbf{y}(x_{i+1}) = \mathbf{y}(x_i) + \int_{x_i}^{x_{i+1}} \mathbf{f}(x, \mathbf{y}(x))dx$$

The integral $\int_{x_i}^{x_{i+1}} \mathbf{f}(x, \mathbf{y}(x))dx$ generally cannot be directly evaluated due to the dependence of $\mathbf{f}$ on $\mathbf{y}$ (as the true solution is unknown), so many numerical methods attempt to approximate $\int_{x_i}^{x_{i+1}} \mathbf{f}(x, \mathbf{y}(x))dx$ in some way.

Perhaps the simplest approximation of $\int_{x_i}^{x_{i+1}} \mathbf{f}(x, \mathbf{y}(x))dx$ is to use a vector of rectangles of width $x_{i+1} - x_i$ and heights $\mathbf{f}(x_i, \mathbf{y}(x_i))$. This approximation gives rise to the well known Euler's method, which also assumes the widths $x_{i+1} - x_i = h$ are constant:

$$\boldsymbol{y}_{i+1} = \boldsymbol{y}_i + h\mathbf{f}(x_i, \boldsymbol{y}_i)$$

The Euler Method is typically initiated by $(x_0, \boldsymbol{y}_0), \boldsymbol{y}_0 = \mathbf{y}(x_0)$ as the true solution is known at the initial condition. The Euler Method is also an explicit method since $\boldsymbol{y}_{i+1}$ is expressed explicitly in terms of $\boldsymbol{y}_j$ for $j < i + 1$.

The accuracy of a numerical method is typically assessed via two metrics. The first is known as the *local truncation error* $\tau_n$, which measures the error against the true solution in one iteration of the numerical method. For the Euler method this is equal to:

$$\tau_i = \mathbf{y}(x_i) - \mathbf{y}(x_{i-1}) - h\mathbf{f}(x_{i-1}, \mathbf{y}(x_{i-1}))$$

The global truncation error instead measures the accumulation of local truncation error over all of the iterations, which is simply equal to $e_i = \mathbf{y}(x_i) - \boldsymbol{y}_i$. A numerical method is convergent if the $e_i$ tends to 0 as $h$ tends to 0, and is a *order* $q$ method if $|e_i| = \mathcal{O}(h^q)$. The Euler method can be shown to be a first order method, i.e. $q = 1$ (Theorem 1.1, (Iserles, 2008)), which may not perform well unless $h$ is very small. For this reason the Euler method is not typically used.

A more popular and widely used higher order method is the Runge-Kutta method. This is a family of higher order methods, which takes a weighted average of a number of gradient values between $x_{i-1}$ and $x_i$. The Euler's method is a special case of Runge-Kutta. The most well known version of Runge-Kutta is the explicit, 4th order Runge-Kutta method (Chapter 8.5 of (Conte & De Boor, 1980)):

$$\boldsymbol{y}_{i+1} = \boldsymbol{y}_i + \frac{h}{6}(\boldsymbol{k}_1 + 2\boldsymbol{k}_2 + 2\boldsymbol{k}_3 + \boldsymbol{k}_4)$$

where

$$\boldsymbol{k}_1 = \boldsymbol{f}(x_i, \boldsymbol{y}_i), \boldsymbol{k}_2 = \mathbf{f}\left(x_i + \frac{h}{2}, \boldsymbol{y}_i + \frac{h\boldsymbol{k}_1}{2}\right)$$

$$\boldsymbol{k}_3 = \boldsymbol{f}\left(x_i + \frac{h}{2}, \boldsymbol{y}_i + \frac{h\boldsymbol{k}_2}{2}\right), \boldsymbol{k}_4 = \boldsymbol{f}\left(x_i + h, \boldsymbol{y}_i + h\boldsymbol{k}_3\right)$$

The Runge-Kutta method throws away all the information before $\boldsymbol{y}_i$ when calculating $\boldsymbol{y}_{i+1}$. A class of methods that improves computational efficiency by using information from previous time steps are known as *linear multistep methods*, the most well known of which is perhaps the Adam-Bashforth Method (Chapter 2, (Iserles, 2008)). For example, the second order Adam-Bashforth method is as follows:

$$\boldsymbol{y}_{i+2} = \boldsymbol{y}_{i+1} + \frac{h}{2}(3\mathbf{f}(x_{i+1}, \boldsymbol{y}_{i+1}) - \mathbf{f}(x_i, \boldsymbol{y}_i))$$

Another perhaps more obvious improvement to Euler's method would be to use the area

of a trapezium $\frac{h}{2}(\mathbf{f}(x_{i+1}, \mathbf{y}(x_{i+1})) + \mathbf{f}(x_i, \mathbf{y}(x_i)))$ instead of a rectangle, to approximate $\int_{x_i}^{x_{i+1}} \mathbf{f}(x, \mathbf{y}(x))dx$, leading to:

$$\mathbf{y}_{i+1} = \mathbf{y}_i + \frac{h}{2}(\mathbf{f}(x_{i+1}, \mathbf{y}_{i+1}) + \mathbf{f}(x_i, \mathbf{y}_i))$$

It can be shown the *trapezoidal rule* is a second order method (Chapter 1.3, (Iserles, 2008)). Notice however the presence of $\boldsymbol{y}_{i+1}$ on both sides of the equation. This means that the trapezoidal rule is an *implict* method, and requires a nonlinear equation to be solved in order to compute it. Implicit numerical methods are more difficult to use but are often more numerically stable than explicit methods (see for example Chapter 4 of (Iserles, 2008) for a more detailed discussion), and implicit analogues of Runge-Kutta and linear multistep methods (known as 'Adam-Moulton methods') also exist.

## 2.2 Partial Differential Equations

A partial differential equation (PDE) is a natural generalisation of an ordinary differential equation, where a *state* variable $\mathbf{u} : \Gamma \to \mathbb{R}^m$, $\Gamma \subset \mathbb{R}^d$, can now depend on its partial derivatives with respect to multiple variables (slight abuse of notation here, as $\Gamma$ was previously one dimensional and used for the domain of ODEs). In general, proving the existence and uniqueness of solution for PDEs is difficult. Currently there isn't a Picard-Lindelöf existence theorem equivalent for PDEs. In fact, one of the most famous unsolved problems in mathematics and physics as well as one of the seven Millennium Prize Problems stated by the Clay Mathematics Institute, is to prove or give a counter-example of the following statement:

*In three space dimensions and time, given an initial velocity field, there exists a vector velocity and a scalar pressure field, which are both smooth and globally defined, that solve the Navier-Stokes equations.*

For a more precise problem description we refer the reader to (Fefferman, 2000). The Navier-Stokes equations are highly significant as they are the equations of motion for Newtonian fluids, and can be derived from the conservation of mass and momentum. The equations can take different forms depending on the physical assumptions. In the case of an incompressible (meaning constant fluid density) fluid, which is also the case specified in the Millennium Prize problem, the equations are:

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\frac{1}{\rho}\nabla p + \nu \nabla^2 \mathbf{u} + f(\mathbf{x}, t)$$

where $\mathbf{u}$ denotes the 3-dimensional fluid velocity field, $p$ the scalar pressure field, $f(\mathbf{x}, t)$ the external volumetric force (such as gravity) and $\nu$ a constant representing kinematic viscosity. The fact that it's not clear whether a specific PDE like the Navier-Stokes

equation has a well defined solution indicates that it's much harder to prove the existence and uniqueness of solutions of PDEs in general.

To understand why this might be the case, rather than viewing a PDE as an equation with one dependent variable and its derivative with respect to a finite number of independent variables, it can be instead viewed as an ODE with an infinite number of dependent variables. Consider for example the following simple PDE:

$$\frac{\partial \mathrm{u}(t,x)}{\partial t} = x \tag{2.6}$$

This PDE can be viewed as an ODE in $\mathrm{u}_x(t) = \mathrm{u}(t,x)$, and $\mathrm{u}_x$ viewed as an infinite dimensional 'vector' as $x$ can take an uncountably infinite number of values. This is reflected by the fact that the general solution to this example is $\mathrm{u}(t,x) = xt + c(x)$ for any arbitrary function $c(x)$, implying that there's an infinite degree of freedom for the choice of the initial condition.

This view of PDEs as *infinite dimensional* ODEs suggests establishing a general theory for the existence and uniqueness of solutions of PDEs is likely to be much more difficult. Indeed, we already know Picard-Lindelöf only guarantees existence and uniqueness for solutions of finite dimensional ODEs. Instead, the existence and uniqueness of solutions of PDEs are typically established on a case by case basis depending on the equation. Clearly, it would be desirable to have an unique solution (given sufficient initial and boundary conditions) that is smooth enough so that all the derivatives which appear in the PDE exist and are continuous. This is known as a *classical* or *strong solution* of the PDE, and is analogous to the solution of ODEs. However, in practice it is difficult to find solutions that are this smooth directly, or it may be that no strong solutions exist at all. Instead, the typical strategy in PDE theory is to first search for a *weak solution*, which does not impose smoothness requirements as strong as a classical solution. This often makes it easier to establish the existence and uniqueness of a (weak) solution, and then examine afterwards whether or not this weak solution is also smooth enough to be a strong solution (Chapter 1.3, (Evans, 1998)). So essentially, the strategy is to consider the *existence* and *smoothness* of the solution separately.

The definition of the weak solution depends on the form of the PDE. In some equations, first a 'weak form' of the differential equation needs to be derived. For example, consider the following second order PDE for $\mathrm{u} : \Gamma \in \mathbb{R}^{\mathrm{d}} \to \mathbb{R}$:

$$-\nabla \cdot (a(\boldsymbol{x})\nabla \mathrm{u}(\boldsymbol{x})) = f(\boldsymbol{x}) \quad \boldsymbol{x} \in \Gamma \tag{2.7}$$

$$\mathrm{u}(\boldsymbol{x}) = 0 \quad \boldsymbol{x} \in \partial\Gamma \tag{2.8}$$

Note Poisson's equation is a special case of this PDE with $a(\boldsymbol{x}) \equiv 1$. The weak form of the

PDE is obtained by multiplying the PDE by an arbitrary smooth function $v$ of compact support (i.e. $v$ vanishes outside some compact set in $\Gamma$), and then integrating both sides (Example 12.8 of (Sullivan, 2015)):

$$-\int_{\Gamma} \nabla \cdot (a(\boldsymbol{x})\nabla \mathrm{u}(\boldsymbol{x}))v(\boldsymbol{x})d\boldsymbol{x} = \int_{\Gamma} f(\boldsymbol{x})v(\boldsymbol{x})d\boldsymbol{x} \tag{2.9}$$

Using integration by parts, this becomes:

$$\int_{\Gamma} a(\boldsymbol{x})\nabla \mathrm{u}(\boldsymbol{x}) \cdot \nabla v(\boldsymbol{x})\mathrm{d}\boldsymbol{x} = \int_{\Gamma} f(\boldsymbol{x})v(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \tag{2.10}$$

Equation 2.10 is the weak form of the PDE 2.8, and if $\mathrm{u}(\boldsymbol{x})$ satisfies 2.10 for any smooth $v$ of compact support, then it is a weak solution of the PDE 2.8. The weak form in this case does not contain any second derivative terms, which is consistent with the idea that the weak solution imposes less of a smoothness requirement than the strong/classical solution. Weak solutions are usually defined in Sobolev spaces, which loosely speaking are $L^p$ normed vector space of functions smooth enough to possess weak derivatives up to a given order. Sometimes it is possible to prove the weak solution is in fact smooth enough to be a strong solution, via arguments such as the Sobolev embedding theorem, which roughly speaking states Sobolev spaces which contains functions with (at least) a sufficiently high order of weak derivatives also contains functions with (at least) a lower order of classical or strong derivatives. This is typically the strategy in finding classical solutions of PDEs. For a detailed overview of the theory of Sobolev spaces and inequalities, we direct the reader to Chapters 5 of (Evans, 1998).

Evidently even establishing the existence of solutions of PDEs is much more difficult than for ODEs, so perhaps it's unsurprising that obtaining closed form solutions of PDEs is in general even more difficult. So once again numerical methods are used to solve PDEs in practice, outside a small number of examples where a closed form solution is available. Classical numerical methods of PDEs are designed to approximate either the strong or the weak solution. For the strong solution, the canonical numerical method is the finite difference method, while the canonical numerical method for the weak solution is the finite element method.

### 2.2.1 Finite Difference Methods

Finite difference methods approximates the PDE by replacing the derivative terms with finite difference approximations. For example, consider the one dimensional heat equation with Dirichlet boundary conditions on the unit interval:

$$\frac{\partial \mathrm{u}(t,x)}{\partial t} = \frac{\partial^2 \mathrm{u}(t,x)}{\partial x^2} \tag{2.11}$$

$$\mathrm{u}(t,0) = \mathrm{u}(t,1) = 0 \tag{2.12}$$

$$\mathrm{u}(x,0) = g(x) \tag{2.13}$$

One way to discretise this PDE would be to fix evenly spaced time discretisation $[t_0, t_1, \ldots, t_{n-1}]$ and spatial discretisation $[x_1, x_2, \ldots, x_m]$, with $\Delta t = t_{i+1} - t_i$, and $\Delta x = x_{i+1} - x_i$, and update the numerical solution $u_j^{i+1}$ at time $t_{i+1}$ and space $x_j$ by:

$$\frac{u_j^{i+1} - u_j^i}{\Delta t} = \frac{u_{j+1}^i - 2u_j^i + u_{j-1}^i}{\Delta x^2}$$

This is known as the *Forward Time, Centered Space (FTCS)* method. It is an explicit method, as the formula for $u_j^{i+1}$ is an explicit function of terms at time $t_i$ or earlier, and is therefore easy to compute. It can also be shown the global truncation error is of order $\mathcal{O}(\Delta t + \Delta x^2)$ (see Chapter 1 of (Thomas, 1998)), consistent with the first and second order discretisation approximations used on the time and spatial derivatives, respectively. However, this numerical scheme also requires $r = \Delta t/\Delta x^2 \leq 1/2$ to be numerically stable and convergent (see Example 2.2.2 of (Thomas, 1998)). These stability conditions are often present in explicit schemes which may be difficult to achieve if the time and spatial discretisations are restrained in some way, or if there is a forcing function in the PDE that is expensive or difficult to evaluate.

The implicit version of the above scheme is the *Backward Time, Centered Space (BTCS)* method, which evaluates the discretised spatial derivative at $t_{i+1}$ instead of $t_i$:

$$\frac{u_j^{i+1} - u_j^i}{\Delta t} = \frac{u_{j+1}^{i+1} - 2u_j^{i+1} + u_{j-1}^{i+1}}{\Delta x^2} \tag{2.14}$$

This scheme has the same order of error as the explicit version, and is more difficult to compute due to the presence of $t_{i+1}$ terms on both sides of the equation, so an inverse problem needs to be solved to retrieve the numerical solution. However, the implicit method carries the advantage of being unconditionally stable (i.e. stability does not depend on $\Delta t$ or $\Delta x$, see Chapter 2.1.0 of (Morton & Mayers, 2005)).

To solve 2.14, the equation is rearranged into:

$$-ru_{j-1}^{i+1} + (1+2r)u_j^{i+1} - ru_{j+1}^{i+1} = u_j^i$$

This equation can then be cast into matrix form and solved via matrix inversion:

$$\begin{bmatrix} 1+2r & -r & \dots & \dots & \dots \\ -r & 1+2r & -r & \dots & \dots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \dots & \dots & -r & 1+2r & -r \\ \dots & \dots & \dots & -r & 1+2r \end{bmatrix} \begin{bmatrix} u_2^{i+1} \\ u_3^{i+1} \\ \vdots \\ u_{m-2}^{i+1} \\ u_{m-1}^{i+1} \end{bmatrix} = \begin{bmatrix} u_2^i + ru_1^{i+1} \\ u_3^i \\ \vdots \\ u_{m-2}^i \\ u_{m-1}^i + ru_m^{i+1} \end{bmatrix}$$

Note $u_1^{i+1}$ and $u_m^{i+1}$ are on the boundary and therefore already known via the boundary condition.

One way to 'combine' FTCS and BTSC is to take the mean of the discretised spatial derivative at time $t_i$ and $t_{i+1}$ instead of one or the other, and can also be interpreted as taking the spatial derivative at time $t_{i+1/2}$. This gives another implicit scheme, known as the *Crank-Nicholson method*:

$$\frac{u_j^{i+1} - u_j^i}{\Delta t} = \frac{1}{2}\left(\frac{u_{j+1}^i - 2u_j^i + u_{j-1}^i}{\Delta x^2} + \frac{u_{j+1}^{i+1} - 2u_j^{i+1} + u_{j-1}^{i+1}}{\Delta x^2}\right) \tag{2.15}$$

This can be similarly rearranged into:

$$-ru_{j-1}^{i+1} + (2+2r)u_j^{i+1} - ru_{j+1}^{i+1} = (2-2r)u_j^i + ru_{j-1}^i + ru_{j+1}^i \tag{2.16}$$

and solved via matrix form:

$$\begin{bmatrix} 2+2r & -r & \dots & \dots & \dots \\ -r & 2+2r & -r & \dots & \dots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \dots & \dots & -r & 2+2r & -r \\ \dots & \dots & \dots & -r & 2+2r \end{bmatrix} \begin{bmatrix} u_2^{i+1} \\ u_3^{i+1} \\ \vdots \\ u_{m-2}^{i+1} \\ u_{m-1}^{i+1} \end{bmatrix} = \begin{bmatrix} (2-2r)u_2^i + ru_1^i + ru_3^i + ru_1^{i+1} \\ (2-2r)u_3^i + ru_2^i + ru_4^i \\ \vdots \\ (2-2r)u_{m-2}^i + ru_{m-3}^i + ru_{m-1}^i \\ (2-2r)u_{m-1}^i + ru_{m-2}^i + ru_m^i + ru_m^{i+1} \end{bmatrix}$$

The advantage of Crank-Nicholson over BTCS is that the error is of order $\mathcal{O}(\Delta t^2 + \Delta x^2)$ (Chapter 2.10, (Morton & Mayers, 2005)) while remaining unconditionally stable (Chapter 2.1.0, (Morton & Mayers, 2005)) and not being much more difficult computationally to solve.

### 2.2.2 Finite Element Methods

In this thesis we will not be concerned with finite element methods as we are interested in obtaining a strong probabilistic solution of the PDE, where the posterior samples are smooth enough such that all the derivative terms in the PDE exist, almost surely. However, we still give a brief introduction to finite element methods as they are widely used in classical numerical analysis of PDEs.

Finite element methods utilise the weak form of the PDE instead (unlike finite difference methods which are based on the strong form of the PDE) to obtain a numerical solution. One such popular method is known as the *Galerkin method*. Returning to the example 2.10 and considering the one dimensional (i.e. ODE) case, the Galerkin method assumes the solution takes the form:

$$u(x) = \sum_{j=1}^{m} \lambda_j \Lambda_j(x)$$

where the $\Lambda_j(x)$ are some linearly independent basis functions (in some suitable Sobolev Space), one popular choice for $\Lambda_j(x)$ are triangle or tent functions around $x_j$:

$$\Lambda_j(x) = \begin{cases} 1 - |\frac{x-x_j}{\Delta x}| & |\frac{x-x_j}{\Delta x}| \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

Furthermore, by assuming $v(x) = \sum_{j=1}^{m} v_j \Lambda_j(x)$ and plugging both expressions of $u(x)$ and $v(x)$ into 2.10, it can be shown that:

$$A\boldsymbol{\lambda} = \boldsymbol{f}$$

where $A_{ij} = \int_\Gamma a(x) \nabla \Lambda_i(x) \cdot \Lambda_j(x) dx$, $\boldsymbol{\lambda}_j = \lambda_j$, $\boldsymbol{f}_i = \int_\Gamma f(x) \Lambda_i(x) dx$.
So $\boldsymbol{\lambda}$ can be obtained by inverting the matrix $A$.

## 2.3 Gaussian Processes

In order to develop probabilistic numerical methods of differential equations, it is necessary to consider what it means for a function to be random. Stochastic processes are therefore the ideal mathematical objects to work with. Let $I$ be some measurable index set. A *stochastic process* $X : I \times \Omega \to \mathbb{R}^m$ is a collection of $\mathbb{R}^m$ valued random variables defined on a shared probability space $(\Omega, \mathcal{F}, \mathbb{P})$, where $\Omega$ is a sample space, $\mathcal{F}$ a $\sigma$-algebra, and $\mathbb{P}$ a probability measure. If $I$ is a subset of the real line, the stochastic process is often interpreted as being indexed by time. If $I$ is a subset of $\mathbb{R}^d$, then $X$ is sometimes also called a *random field*. The *expectation* of $X$ is defined by the Lebesgue integral $\mathbb{E}[X(z, \omega)] := \int X(z, \omega) \mathrm{d}\mathbb{P}(\omega)$

For the rest of this section we assume $m = 1$ for simplicity. An important property of stochastic processes is stationarity. A stochastic process $X$ on $I = \mathbb{R}^d$ is *strongly stationary* if $F_X(x_{z_1}, \ldots, x_{z_n})) = F_X(x_{z_1+z}, \ldots, x_{z_n+z}))$ for all $z, z_1, \ldots, z_n \in \mathbb{R}^{\mathbb{d}}$, where $F_X(x_{z_1}, \ldots, x_{z_n}))$ is the joint distribution of $X(z_1), \ldots, X(z_n)$. One family of stochastic processes that of particular interest in this thesis is *Gaussian Processes*. Gaussian Processes are stochastic processes $X : I \times \Omega \to \mathbb{R}$ with a continuous index set $I$ such that for every finite set of indices $z_1, z_2, \ldots, z_k \in I$, we have $(X(z_1, .), X(z_2, .), \ldots, X(z_k, .))$ is a multivariate Gaussian random variable. The expectation and covariance (also known as kernel function) of a Gaussian Process are intuitively defined as:

$$\begin{aligned}\mu(z) &= \mathbb{E}\left[X(z, \omega)\right] \\ \Sigma(z, z') &= \mathbb{E}\left[(X(z, \omega) - \mu(z))(X(z', \omega) - \mu(z'))\right]\end{aligned}$$

Gaussian Processes possess several desirable properties that are useful in a Bayesian updating framework, which it inherits from its finite dimensional analogue. First, like its finite dimensional analogue, a Gaussian Process's statistical properties are completely specified by its mean and covariance functions (Chapter 2.2 of (Rasmussen & Williams, 2006)). Because of this Gaussian Processes are typically written using the notation $X \sim \mathcal{GP}(\mu, \Sigma)$. Secondly, if two normal random variables are jointly normally distributed, then the conditional distribution of one given the other is normally distributed as well. This property extends to Gaussian Processes (Chapter 2.2 of (Rasmussen & Williams, 2006)). For example, consider $X$ over test points $\boldsymbol{z^*} = (z_1^*, z_2^*, \ldots, z_{k_1}^*)$ and training points $\boldsymbol{z} = (z_1, z_2, \ldots, z_{k_2})$, then it has joint distribution:

$$\begin{bmatrix} X(\boldsymbol{z^*}) \\ X(\boldsymbol{z}) \end{bmatrix} \sim \mathcal{N}\left( \begin{bmatrix} \mu(\boldsymbol{z^*}) \\ \mu(\boldsymbol{z}) \end{bmatrix}, \begin{bmatrix} \Sigma(\boldsymbol{z^*}, \boldsymbol{z^*}) & \Sigma(\boldsymbol{z^*}, \boldsymbol{z}) \\ \Sigma(\boldsymbol{z}, \boldsymbol{z^*}) & \Sigma(\boldsymbol{z}, \boldsymbol{z}) \end{bmatrix} \right)$$

and conditional distribution:

$$\begin{aligned} X(\boldsymbol{z^*})|X(\boldsymbol{z}) = \boldsymbol{a} \sim \mathcal{N}(&\mu(\boldsymbol{z^*}) + \Sigma(\boldsymbol{z^*}, \boldsymbol{z})\Sigma(\boldsymbol{z}, \boldsymbol{z})^{-1}(\boldsymbol{a} - \mu(\boldsymbol{z})), \\ &\Sigma(\boldsymbol{z^*}, \boldsymbol{z^*}) - \Sigma(\boldsymbol{z^*}, \boldsymbol{z})\Sigma(\boldsymbol{z}, \boldsymbol{z})^{-1}\Sigma(\boldsymbol{z}, \boldsymbol{z^*})) \end{aligned}$$

Thirdly, Gaussian Processes are closed under linear and affine maps. This property is very useful in solving differential equations probabilistically because differentiation is a linear map, so the derivative of a Gaussian Process is again a Gaussian Process (Chapter 9.4, (Rasmussen & Williams, 2006)). For example:

$$\begin{bmatrix} X(\boldsymbol{z}^*) \\ \frac{\partial X(\boldsymbol{z})}{\partial z_1} \end{bmatrix} \sim \mathcal{N}\left( \begin{bmatrix} \mu(\boldsymbol{z}^*) \\ \frac{\partial \mu(\boldsymbol{z})}{\partial z_1} \end{bmatrix}, \begin{bmatrix} \Sigma(\boldsymbol{z}^*, \boldsymbol{z}^*) & \frac{\bar{\partial}\Sigma(\boldsymbol{z}^*, \boldsymbol{z})}{\partial \bar{z_1}} \\ \frac{\partial \Sigma(\boldsymbol{z}, \boldsymbol{z}^*)}{\partial z_1} & \frac{\partial \bar{\partial}\Sigma(\boldsymbol{z}, \boldsymbol{z})}{\partial z_1 \partial \bar{z_1}} \end{bmatrix} \right)$$

where $\frac{\bar{\partial}}{\partial \bar{z_1}}$ denotes differentiation with respect to the second variable. Therefore the conditional distribution is similarly:

$$X(\boldsymbol{z}^*)|\frac{\partial X(\boldsymbol{z})}{\partial z_1} = \boldsymbol{b} \sim \mathcal{N}\left( \mu(\boldsymbol{z}^*) + \frac{\bar{\partial}\Sigma(\boldsymbol{z}^*, \boldsymbol{z})}{\partial \bar{z_1}} \frac{\partial \bar{\partial}\Sigma(\boldsymbol{z}, \boldsymbol{z})}{\partial z_1 \partial \bar{z_1}}^{-1} (\boldsymbol{b} - \frac{\partial \mu(\boldsymbol{z})}{\partial z_1}), \right.$$
$$\left. \Sigma(\boldsymbol{z}^*, \boldsymbol{z}^*) - \frac{\bar{\partial}\Sigma(\boldsymbol{z}^*, \boldsymbol{z})}{\partial \bar{z_1}} \frac{\partial \bar{\partial}\Sigma(\boldsymbol{z}, \boldsymbol{z})}{\partial z_1 \partial \bar{z_1}}^{-1} \frac{\partial \Sigma(\boldsymbol{z}, \boldsymbol{z}^*)}{\partial z_1} \right)$$

Fourthly, a Gaussian Process $X \sim \mathcal{GP}(\mu, \Sigma)$ that is *weakly stationary*, meaning it satisfies the following:

$$\begin{aligned} \mu(z_1) &= \mu(z_1 + z_2) \\ \Sigma(z_1, z_2) &= \Sigma(z_1 - z_2, 0) \\ \mathbb{E}(X(z_1, \omega)^2) &< \infty \end{aligned}$$

for all $z_1, z_2 \in \mathbb{R}$, is also strongly stationary. This is significant as weak stationarity is in general weaker than strong stationarity, but much easier to verify.

There is a wide range of choices of the covariance function for Gaussian Processes, here we give some of the most popular examples, for index set $I \subset \mathbb{R}$. For stationary covariance functions, higher dimensional generalisations can either be obtained by replacing $|z - z'|$ with the Euclidean norm $\|z - z'\|$. Alternatively, a product of one dimensional covariance functions (one for each dimension of $I$) can be used instead.

**Definition 2** (Squared Exponential Covariance)**.**

$$C(z, z'; \rho, \sigma) := \sigma^2 \exp\left( -\frac{(z - z')^2}{2\rho^2} \right) \tag{2.17}$$

is the *Squared Exponential* covariance model.

The hyperparameters are $\sigma^2$, which is known as the amplitude parameter, and $\rho$, which is known as the length-scale parameter. Both $\sigma$ and $\rho$ are strictly positive. The length-scale parameter controls the rate of change of samples from the Gaussian Process, and how far they can extrapolate away from fixed and known function values. The smaller the length-scale, the more more quickly the sampled function can change. The amplitude

parameter determines the amount of variation of samples from the Gaussian Process from the mean, the larger the amplitude parameter, the greater the variation.

**Definition 3** (Rational Quadratic Covariance)**.**

$$C(z, z'; \rho, \sigma) := \sigma^2 \left(1 + \frac{(z - z')^2}{\rho^2}\right)^{-1} \tag{2.18}$$

is the *rational quadratic* covariance model.

The hyperparameters for the Rational Quadratic Covariance are analogous to the hyperparameters for the Squared Exponential Covariance. It can be shown Gaussian Processes with zero mean and Squared Exponential or Rational Quadratic kernels are infinitely mean squared differentiable (Chapter 4.2 of (Rasmussen & Williams, 2006)), a concept we will formally define in Chapter 4 of this thesis. Note mean squared differentiability of a Gaussian Process is not the same as differentiability of sample paths generated from the Gaussian Process, which is in general much more difficult to determine. There are also other choice of covariance functions that are much less smooth, for example:

**Definition 4** (Wiener Covariance)**.**

$$C(z, z') = \sigma^2 \min(z - \tau, z' - \tau) \tag{2.19}$$

Here $\sigma^2$ again describes the amplitude and $\tau$ is the starting point of the Brownian motion, where it is initialised. The Wiener Process is also known as Brownian Motion, and is highly significant in many areas of mathematics, including stochastic calculus and martingales. It also has many applications outside of mathematics, such as in physics in studying the diffusion of fluids. The Wiener Process is mean squared continuous but not mean squared differentiable anywhere (see Appendix B of (Rasmussen & Williams, 2006)).

**Definition 5** (Matérn Covariance)**.** Let $\nu = p + \frac{1}{2}$ where $p \in \mathbb{N}$. The Matérn covariance function is defined, for $z, z' \in \mathbb{R}$, as

$$K_\nu(z, z') = K_\nu(z - z') = \sigma^2 \exp\left(-\frac{|z - z'|}{\rho}\right) \frac{p!}{(2p)!} \sum_{k=0}^{p} \frac{(2p - k)!}{(p - k)!k!} \left(\frac{2}{\rho}\right)^k |z - z'|^k. \tag{2.20}$$

Again $\sigma^2$ and $\rho$ denote the amplitude and length-scale respectively. A Gaussian Process with zero mean and the Matérn $p + \frac{1}{2}$ kernel is $p$ times mean squared differentiable (see for example Section 2 of (Stein, 1999)).

This Background chapter laid out an accessible summary to the necessary mathematical material required for the thesis. In the next chapter, we discuss the existing probabilistic numerical methods for the solution of differential equations.

# Chapter 3

# An Overview of the State of the Art

The purpose of this chapter is to provide a high-level overview of existing PNMs for differential equations. In particular, we highlight whether existing methods constitute (approximate or exact) Bayesian PNM. Skilling (1992) introduced the first PNM (of any flavour) for the numerical solution of ODEs. Two decades later, this problem is receiving renewed critical attention as part of the active development of PNM. Nevertheless, the vast majority of existing PN methods on differential equations only apply to ODEs, and PN methods for PDEs are relatively limited, even for linear PDEs. To date, no (approximate or exact) Bayesian PNM for the numerical solution of nonlinear PDEs has been proposed.

**Notation:** The notational convention used in this thesis is that the non-italicised y denotes a generic function, whereas the italicised $y$ denotes a scalar value taken by the function y. The notation $\mathrm{y}^\dagger$ is reserved for the true solution to an ODE. We also in this section only use the notation $Y$ to refer to a Gaussian process, $m$ and $\Sigma$ to refer to the mean and covariance of the Gaussian Process respectively. Throughout, the underlying state space $\mathcal{Y}$ is taken to be a space occupied by the true solution of the ODE, i.e. $\mathrm{y}^\dagger \in \mathcal{Y}$.

### Skilling (1992)

The first paper on this topic, of which we are aware, was Skilling (1992). This will serve as a prototypical PNM for the numerical solution of an ODE. Originally described as 'Bayesian' by the author, we will argue that, at least in the strict sense of Definition 1, it is not a Bayesian PNM. Consider a generic univariate first-order initial value problem

$$\frac{\mathrm{d}\mathrm{y}}{\mathrm{d}x} = f(x, \mathrm{y}(x)), \qquad x \in [x_0, x_T], \qquad \mathrm{y}(x_0) = y_0. \tag{3.1}$$

Throughout this thesis all ODEs that we consider will be assumed to be well-defined and admit a unique solution $\mathrm{y}^\dagger \in \mathcal{Y}$ where $\mathcal{Y}$ is some pre-specified set, which is reasonable as the assumptions in Theorem  are typically satisfied. In this thesis the quantity of interest $Q(\mathrm{y}^\dagger)$ will either be the solution curve $\mathrm{y}^\dagger$ itself or the value $\mathrm{y}^\dagger(x_T)$ of the solution at a specific input (in this section it will be the former). The approach outlined in Skilling (1992) allows for a general prior $\mu \in \mathcal{P}_\mathcal{Y}$. The gradient field $f$ is treated as a 'black box' oracle that can be queried at a fixed computational cost. Thus we are provided with evaluations of the gradient field $[f(x_0, y_0), \ldots, f(x_n, y_n)]^\top \in \mathbb{R}^{n+1}$ for certain input pairs $\{(x_i, y_i)\}_{i=0}^n$.

This approach of treating evaluations of the gradient field as 'data' will be seen to be a common theme in existing PNM for ODEs and theoretical support for this framework is rooted in the field of information-based complexity (Traub & Woźniakowski, 1992). Let $a_i = f(x_i, y_i)$ and $a^i = [a_0, \ldots, a_i]$. The selection of the input pairs $(x_i, y_i)$ on which $f$ is evaluated is not constrained and several possibilities, of increasing complexity, were discussed in Skilling (1992). To fix ideas, the simplest such approach is to proceed iteratively as follows:

(0.1) The first pair $(x_0, y_0)$ is fully determined by the initial condition of the ODE.

(0.2) The oracle then provides one piece of information, $a_0 = f(x_0, y_0)$.

(0.3) The prior $\mu$ is updated according to $a_0$, leading to a belief distribution $\mu_0$ which is just the disintegration element $\mu^{a^0}$.

(1) A discrete time step $x_1 = x_0 + h$, where $h = \frac{x_T - x_0}{n} > 0$, is performed and a particular point estimate $y_1 = \int \mathrm{y}(x_1) \mathrm{d}\mu_0(\mathrm{y})$ for the unknown true value $\mathrm{y}^\dagger(x_1)$ is obtained. This specifies the second pair $(x_1, y_1)$.

The process continues similarly, such that at time step $i-1$ we have a belief distribution $\mu_{i-1} = B(\mu, a^{i-1}) \in \mathcal{P}_\mathcal{Y}$, where the general belief update operator $B$ is yet to be defined, and the following step is performed:

($i$) Let $x_i = x_{i-1} + h$ and set $y_i = \int \mathrm{y}(x_i) \mathrm{d}\mu_{i-1}(\mathrm{y})$ .

The final output is a probability distribution $\mu_n = B(\mu, a^n) \in \mathcal{P}_\mathcal{Y}$. Now, strictly speaking, the method just described is *not* a PNM in the concrete sense that we have defined. Indeed, the final output $\mu_n$ is a deterministic function of the values $a^n$ of the gradient field that were obtained. However, in the absence of additional assumptions on the global smoothness of the gradient field, the values of $f(x, y)$ outside any open neighbourhood of the true solution curve $\mathcal{C} = \{(x, y) : y = \mathrm{y}^\dagger(x),\ x \in [x_0, x_T]\}$ do not determine the solution of the ODE and, conversely, the solution of the ODE provides no information about the values of the gradient field outside any open neighbourhood of the true solution curve $\mathcal{C}$.

Thus it is not possible, in general, to write down an information operator $A : \mathcal{Y} \to \mathcal{A}$ that reproduces the information $a^n$ when applied to the solution curve $\mathrm{y}^\dagger(\cdot)$ of the ODE.

The approach taken in Skilling (1992) was therefore to posit an *approximate* information operator $\hat{A}$ and a particular belief update operator $B$, which are now described. The approximate information operator is motivated by the intuition that if $\mathrm{y}^\dagger(x_i)$ is well-approximated by $y_i$ at the abscissa $x_i$ then $\frac{\mathrm{dy}^\dagger}{\mathrm{d}x}(x_i)$ should be well-approximated by $f(x_i, y_i)$. That is, the following approximate information operator $\hat{A}$ was constructed:

$$\hat{A}(\mathrm{y}) \quad = \quad \left[\frac{\mathrm{dy}}{\mathrm{d}x}(x_0), \ldots, \frac{\mathrm{dy}}{\mathrm{d}x}(x_n)\right]^\top. \tag{3.2}$$

Of course, $\hat{A}(\mathrm{y}^\dagger) \neq a^n$ in general. To acknowledge the approximation error, Skilling (1992) proposed to model the information with an approximate likelihood:

$$\frac{\mathrm{d}\mu_n}{\mathrm{d}\mu_0}(\mathrm{y}) \quad = \quad \prod_{i=1}^{n} \frac{\mathrm{d}\mu_i}{\mathrm{d}\mu_{i-1}}(\mathrm{y}) \tag{3.3}$$

$$\frac{\mathrm{d}\mu_i}{\mathrm{d}\mu_{i-1}}(\mathrm{y}) \quad \propto \quad \exp\left(-\frac{1}{2\sigma^2}\left(\frac{\mathrm{dy}}{\mathrm{d}x}(x_i) - f(x_i, y_i)\right)^2\right) \tag{3.4}$$

Where $\frac{\mathrm{d}\mu_i}{\mathrm{d}\mu_{i-1}}(\mathrm{y})$ denotes the Radon-Nikodym derivative. This was referred to in Skilling (1992) as simply a "likelihood" and, together with $\mu_0 = \mu^{a^0}$, the output $\mu_n$ is completely specified. Here $\sigma$ is a fixed positive constant, however in principle a non-diagonal covariance matrix can also be considered. The negative consequences of basing inferences on an approximate information operator $\hat{A}$ are potentially twofold. First, recall that values of the gradient field that are not contained on the true solution curve of the ODE do not, in principle, determine the true solution curve $\mathrm{y}^\dagger$. It is therefore unclear if these values should be taken into account at all. Second, in the special case where the gradient field $f$ does not depend the second argument then the quantities $\frac{\mathrm{dy}}{\mathrm{d}x}(x_i)$ and $f(x_i, y_i)$ are identical. From this perspective, $\mu_n$ represents inference under a mis-specified likelihood, since information is treated as erroneous when it is in fact exact. The use of a mis-specified likelihood violates the *likelihood principle* and implies violation of the Bayesian framework. This confirms that the approach of Skilling (1992) cannot be Bayesian in the strict sense of Definition 1. After Skilling (1992), several authors have proposed improvements to the above method.

#### Schober *et al.* (2014, 2019); Teymur *et al.* (2016, 2018)

The approach of Schober *et al.* (2014) considered Eq. (3.4) in the $\sigma \downarrow 0$ limit. In order that exact conditioning can be performed in this limit, the input belief distribution $\mu$ was restricted to be a $k$-times integrated Wiener measure on the solution space of the ODE.

The tractability of the integrated Wiener measure leads to a closed-form characterisation of the posterior and enables computation to be cast as a Kalman filter.

Schober *et al.* (2014) makes the observation that the sequential update formulae of the mean of a Gaussian Process is similar to the sequential updating of Runge–Kutta methods, in the sense that Runge–Kutta methods use a linear combination of gradient field observations at previous time-steps in order to estimate the solution at the current time-step. For example, Schober *et al.* (2014) notes that the first update of the Euler's method $y_0 + (x_1 - x_0)f(x_0, y_0)$, coincides with the update of a Gaussian Process prior $Y \sim \mathcal{GP}(m, \Sigma)$ conditioned on the observation $[Y(x_0), \frac{dY(x_0)}{dx}]^\top = [y_0, f(x_0, y_0)]^\top$:

$$m^0(x_1) = m(x_1) + [\Sigma(x_1, x_0), \Sigma_{\bar{\partial}x}(x_1, x_0)] \begin{bmatrix} \Sigma(x_0, x_0) & \Sigma_{\bar{\partial}x}(x_0, x_0) \\ \Sigma_{\partial x}(x_0, x_0) & \Sigma_{\partial x \bar{\partial} x}(x_0, x_0) \end{bmatrix} \begin{bmatrix} y_0 \\ f(x_0, y_0) \end{bmatrix}$$

if the prior mean $m = 0$ everywhere, and the covariance function $K$ chosen to be the once integrated Wiener process.

Furthermore, for $k \in \{1, 2\}$ the authors prove that if the input pair $(x_1, y_1)$ is taken as $y_1 = m^0(x_1) = \int \mathrm{y}(x_1)\mathrm{d}\mu_0(\mathrm{y})$, as indicated in Section 3, then the smoothing estimate $\hat{y}_1 = m^1(x_1) = \int \mathrm{y}(x_1)\mathrm{d}\mu_1(\mathrm{y})$, i.e. the posterior mean for $\mathrm{y}(x_1)$ based on information $a^1$, coincides with the deterministic approximation to $\mathrm{y}^\dagger(x_1)$ that would be provided by a $k$-th order Runge-Kutta method. As such, theoretical guarantees such as local convergence order are inherited. For $k = 3$ it was shown that the same conclusion can be made to hold, *provided* that the input pair $(x_1, y_1)$ is selected in a manner that is no longer obviously related to $\mu_0$.

In all cases however, the identification with a classical Runge-Kutta method does not extend beyond iteration $n = 1$. As a workaround, Schober *et al.* (2014) proposed the method of 'naive chaining', which restarts the algorithm at every time step as if it's a new initial value problem, only taking the value of $\hat{y}_i$ from the previous time step. However, this carries the downside of not producing a global posterior distribution.

The approach of Schober *et al.* (2014) is not Bayesian in the sense of Definition 1, which can again be deduced from dependence on values of the gradient field away from the true solution curve, so that the likelihood principle is violated.

In a subsequent related work, Schober *et al.* (2019) showed for the once integrated Wiener process prior, the mean after each step can be made to coincide with the trapezoidal rule if it takes an additional evaluation of $f$ at the end of each step. Furthermore, Schober *et al.* (2019) showed for the twice integrated Wiener Process, the mean coincides with a third-order Nordsieck method when the predictive distribution has reached steady state.

Similar to Schober *et al.* (2019), Teymur *et al.* (2016, 2018) constructed algorithms

where the posterior distribution in one iteration matches the Adam–Bashforth and Adam–Moulton linear multistep predictors, by using bespoke prior covariance functions constructed from Lagrange polynomials.

### Kersting & Hennig (2016), Magnani *et al.* (2017)

The work of Kersting & Hennig (2016) attempted to elicit an appropriate non-zero covariance matrix for use in Eq. (3.4), in order to encourage uncertainty estimates to be better calibrated. Their proposal consisted of the approximate likelihood

$$\frac{\mathrm{d}\mu_i}{\mathrm{d}\mu_{i-1}}(\mathrm{y}) \propto \exp\left(-\frac{1}{2}\left(\frac{\frac{\mathrm{dy}}{\mathrm{d}x}(x_i) - m_i}{\sigma_i}\right)^2\right) \tag{3.5}$$

$$m_i = \int f(x_i, \mathrm{y}(x_i))\mathrm{d}\mu_{i-1}(\mathrm{y}) \tag{3.6}$$

$$\sigma_i^2 = \int \left(f(x_i, \mathrm{y}(x_i)) - m_i\right)^2 \mathrm{d}\mu_{i-1}(\mathrm{y}). \tag{3.7}$$

This can be viewed as the predictive marginal likelihood for the value $f(x_i, \mathrm{y}(x_i))$ based on $\mu_{i-1}$. From a practical perspective, the approach is somewhat circular as the integrals in Eq. (3.6) and (3.7) involve the black-box gradient field $f$ and are therefore cannot be computed. The authors suggested a number of ways that these quantities could be numerically approximated[1], which involve evaluating $f(x_i, y_i)$ at one or more values $y_i$ that must be specified. The overall approach again violates the likelihood principle and is therefore not Bayesian in the sense of Definition 1.

Magnani *et al.* (2017) examines experimentally ODE filters that use an integrated Ornstein–Uhlenbeck Process prior instead of an integrated Wiener process prior, and argues that an integrated Ornstein–Uhlenbeck process prior is better suited for trajectories with bounded derivatives.

### Chkrebtii *et al.* (2016), Chkrebtii & Campbell (2019)

In Chkrebtii *et al.* (2016) the authors constructed an approximate Bayesian PNM for the solution of initial value problems specified by either a nonlinear ODE or a linear PDE. Instead of using the mean of the current posterior as input to the gradient field as in Kersting & Hennig (2016), the input pair $(x_i, y_i)$ was selected by sampling $y_i$ from the marginal distribution for $\mathrm{y}(x_i)$ implied by $\mu_{i-1}$. The approximate likelihood in this

[1] One such method is *Bayesian quadrature*, another PNM wherein the integrand $f$ is modelled as uncertain until it is evaluated. This raises separate philosophical challenges, as one must then ensure that the statistical models used for $\mathrm{y}(\cdot)$ and $f(x_i, \cdot)$ are logically consistent. In Kersting & Hennig (2016) these functions were simply modelled as independent.

approach was taken as follows:

$$\begin{aligned} \frac{\mathrm{d}\mu_i}{\mathrm{d}\mu_{i-1}}(\mathrm{y}) &\propto \exp\left(-\frac{1}{2}\left(\frac{\frac{\mathrm{dy}}{\mathrm{d}x}(x_i) - f(x_i, y_i)}{\sigma_i}\right)^2\right) \\ m_i &= \int \frac{\mathrm{dy}}{\mathrm{d}x}(x_i)\mathrm{d}\mu_{i-1}(\mathrm{y}) \\ \sigma_i^2 &= \int \left(\frac{\mathrm{dy}}{\mathrm{d}x}(x_i) - m_i\right)^2 \mathrm{d}\mu_{i-1}(\mathrm{y}). \end{aligned}$$

Compared to Eq. (3.5), (3.6) and (3.7), this approach does not rely on integrals over the unknown gradient field. However, the approach also relies on the approximate information operator in Eq. (3.2) and is thus not Bayesian according to Definition 1.

Chkrebtii *et al.* (2016) also presented a nonlinear PDE (Navier–Stokes), but a pseudo-spectral projection in Fourier space was applied at the outset to transform the PDE into a system of first order ODEs - an approach that exploited the specific form of that PDE.

In a later work, Chkrebtii & Campbell (2019) generalises the method of Chkrebtii *et al.* (2016) by allowing for adaptive time stepping. The adaptive timesteps are chosen sequentially by an information theoretic approach, where the value of the next timestep is chosen to be the value which maximises a Monte Carlo estimate of the Kullback–Leibler entropy.

#### Conrad *et al.* (2017); Abdulle & Garegnani (2018)

The approaches proposed in Conrad *et al.* (2017); Abdulle & Garegnani (2018) are not motivated in the Bayesian framework, but instead seek to introduce a stochastic perturbation into a classical numerical method. Both methods focus on the quantity of interest $Q(\mathrm{y}^\dagger) = \mathrm{y}^\dagger(x_T)$. In the simple context of Eq. (3.1), the method of Conrad *et al.* (2017) augments the explicit Euler method with a stochastic perturbation:

$$y_i \;=\; y_{i-1} + hf(x_{i-1}, y_{i-1}) + h^2\epsilon_i, \qquad x_i \;=\; x_{i-1} + h, \qquad i = 1, \ldots, n$$

The distribution of the sequence $(\epsilon_i)_{i=1}^n$ must be specified. In the simplest case where the $\epsilon_i$ are modelled as independent, say with $\epsilon_i \sim \rho$, the canonical flow map $\Phi_i : \mathbb{R} \to \mathbb{R}$ of the explicit Euler method, defined as $\Phi_i(z) = z + hf(x_i, z)$, is replaced by the probabilistic counterpart $\Psi_i : \mathcal{P}_\mathbb{R} \to \mathcal{P}_\mathbb{R}$ given by

$$\Psi_i(\nu)(\mathrm{d}z) = \int \rho\left(\frac{\mathrm{d}z - \Phi_i(\tilde{z})}{h^2}\right)\nu(\mathrm{d}\tilde{z})$$

through which stochasticity can be propagated. The output of the method of Conrad *et al.* (2017) is then $B = \Psi_n \circ \cdots \circ \Psi_1\delta(y_0)$, where $\delta(z)$ denotes an atomic distribution on

$z \in \mathbb{R}$. For the case where each $\rho_i$ has zero mean, it can be shown that the mean of $B$ equals $\Phi_n \circ \cdots \circ \Phi_1(y_0)$, which is exactly the deterministic approximation produced with the explicit Euler method.

This framework can be practically problematic, since $\epsilon_i$ is charged with modelling the extrapolation error and such errors are not easily modelled as independent random variables – Section 2.8 of Higham (2002) is devoted to this point. Thus, if for example $f(x, y) = y$, the true linearisation error at step $i$ is $e^{x_i} - e^{x_{i-1}}$ so that the 'true' sequence $(\epsilon_i)_{i=1}^n$ in this case is monotonic and exponentially unbounded. The challenge of designing a stochastic model for the sequence $(\epsilon_i)_{i=1}^n$ that reflects the highly structured nature of the error remains unresolved. On the other hand, the mathematical properties of this method are now well-understood (Lie *et al.*, 2018, 2019). The proposal of Abdulle & Garegnani (2018) was to instead consider randomisation of the inputs $\{x_i\}_{i=0}^T$ in the context of a classical numerical method, also outside of the Bayesian framework.

#### Cockayne *et al.* (2016)

Linear elliptic PDEs were considered in Cockayne *et al.* (2016), where a Gaussian Process prior was defined on the unknown solution, and the conjugacy of Gaussian measures under linear operators was used to construct an exactly Bayesian PNM. However, the method is limited to the case of linear elliptic PDEs without time dependence.

#### Cockayne *et al.* (2019); Tronarp *et al.* (2019); Kersting *et al.* (2018)

The survey just presented begs the question of whether a Bayesian PNM for ODEs can exist at all. A first step toward this goal was taken in Cockayne *et al.* (2019), where it was argued that an information operator can be constructed if the vector field $f$ is brought to the left-hand-side in Eq. (3.1). Specifically, they proposed the information operator

$$\tilde{A}(\mathrm{y}) \quad = \quad \left[\frac{\mathrm{d}\mathrm{y}}{\mathrm{d}x}(x_0) - f(x_0, \mathrm{y}(x_0)), \ldots, \frac{\mathrm{d}\mathrm{y}}{\mathrm{d}x}(x_n) - f(x_n, \mathrm{y}(x_n))\right]^\top$$

for which the 'data' are trivial; $\tilde{a}^n = 0$. It was rigorously established that the approximate likelihood

$$\frac{\mathrm{d}\mu_{i,\sigma}}{\mathrm{d}\mu_{i-1,\sigma}}(\mathrm{y}) \quad = \quad \exp\left(-\frac{1}{2\sigma^2}\left(\frac{\mathrm{d}\mathrm{y}}{\mathrm{d}x}(x_i) - f(x_i, \mathrm{y}(x_i))\right)^2\right)$$

leads to an exact Bayesian PNM in the limit: $\mu_{n,\sigma} \overset{\mathcal{F}}{\to} \mu^{\tilde{a}^n}$ as $\sigma \downarrow 0$ for $\tilde{A}_{\#}\mu$-almost all $\tilde{a}^n \in \mathbb{R}^{n+1}$. Here $\overset{\mathcal{F}}{\to}$ denotes convergence in an integral probability metric defined by a suitable set $\mathcal{F}$ of test functions (see Sec. 4 of Cockayne *et al.*, 2019). However, the dependence of the information operator $\tilde{A}$ on $f$ means that this cannot be used as the

basis for a practical method. Indeed, unless $f$ depends linearly on its second argument and conjugacy properties of the prior can be exploited, the posterior cannot easily be characterised. Approximate techniques from nonlinear filtering were proposed to address this challenge in Tronarp *et al.* (2019). Tronarp *et al.* (2019) demonstrated how nonlinear filtering techniques can be used to obtain low-cost approximations to the solution of ODEs. In particular, Tronarp *et al.* (2019) treats the IVP 3.1 as a nonlinear Bayesian filtering problem, by conditioning a Gaussian Process prior $Y \sim \mathcal{GP}(\mu, \Sigma)$ sequentially on the observation that $\frac{\mathrm{d}Y^i}{\mathrm{d}x}(x_i) - f(x_i, Y^i(x_i)) = 0$, where $Y^i$ is the prior after $i$ updates. This is generally intractable but approximation techniques in nonlinear Bayesian filtering can be utilised. For example, by Taylor expanding $Y^i$ around its predictive mean in the observation $\frac{\mathrm{d}Y^i}{\mathrm{d}x}(x_i) - f(x_i, Y^i(x_i)) = 0$, Tronarp *et al.* (2019) shows that the zeroth order Taylor approximation is precisely the update scheme in Schober *et al.* (2019), and the first order Taylor approximation corresponds to the extended Kalman Filter. In a later paper, Tronarp *et al.* (2021) provides theoretical convergence rates of the *maximum a posteriori* estimator (MAP) of $Y^i$. Similarly, Kersting *et al.* (2018) also provides local and global convergence rates for ODE filters, but for the filtering mean instead of the MAP.

#### Bosch *et al.* (2020); Krämer & Hennig (2020)

Bosch *et al.* (2020) extends the work of Tronarp *et al.* (2021) by considering an integrated Wiener Process prior with time dependent covariance for the approximate likelihood:

$$\frac{\mathrm{d}\mu_{i,\sigma}}{\mathrm{d}\mu_{i-1,\sigma}}(\mathrm{y}) \quad = \quad \exp\left(-\frac{1}{2\sigma_i^2}\left(\frac{\mathrm{dy}}{\mathrm{d}x}(x_i) - f(x_i, \mathrm{y}(x_i))\right)^2\right)$$

and using a local quasi-maximum likelihood to calibrate $\sigma_i^2$. Also, Bosch *et al.* (2020) considers the residual $Y^i - \mathrm{y}^\dagger(x_i)$ for local error control. Under the assumption the residual is unbiased, Bosch *et al.* (2020) uses the standard deviations of the residual, i.e. the diagonal of the approximate likelihood 3 to adaptively select step sizes to be as large as possible while keeping the standard deviations under a certain tolerance, in order to increase computational efficiency.

Using the same approximate likelihood, Krämer & Hennig (2020) suggests strategies for improving stability, particularly in the case of small step-sizes and using higher order Taylor approximations to $\frac{\mathrm{d}Y^i}{\mathrm{d}x}(x_i) - f(x_i, Y^i(x_i)) = 0$ in the sense of Tronarp *et al.* (2019). Strategies include using accurate initialisation of the prior mean, a coordinate change that makes numerical stability independent of step size, and square root implementation of the Kalman Filter to avoid negative eigenvalues.

**Lie *et al.* (2018); Matsuda & Miyatake (2021)**

These papers do not attempt to solve an ODE probabilistically but instead aim to solve a Bayesian inverse problem to infer an unknown parameter in an ODE model. In general, the likelihood function is intractable when the ODE cannot be analytically solved. This leads to approximate likelihoods, based on numerical methods, being used instead. Lie *et al.* (2018) shows that the posterior distributions of probabilistic ODE solvers such as Schober *et al.* (2014) and Conrad *et al.* (2017) can be used to construct approximate likelihoods for solving inverse problems in the ODE context. Matsuda & Miyatake (2021) instead introduce latent Gaussian random variables to model the discrepancy between the numerical and the true solution of the ODE. Estimates of both the latent variables and parameter are then jointly obtained via maximum (approximate) likelihood. However, the variance of the discretisation error is assumed to monotonically increase with time, which may not be an appropriate assumption if both the true solution and the numerical solution converges to a stationary state as time increases.

This completes our overview of existing PNMs for differential equations. We've seen that the only Bayesian PNM which currently exist are for linear PDEs and linear elliptic PDE, and all other published methods are either approximate Bayesian PNMs or non Bayesian. This thesis will aim to develop exact Bayesian PNMs for certain classes of nonlinear ODEs in chapter 4, and an approximate Bayesian PNM for nonlinear PDEs in chapter 5.

# Chapter 4

# Exact Bayesian Inference for Ordinary Differential Equations?

The literature review in the previous chapter suggests that no Bayesian PNM has yet been proposed outside of very specific, linear differential equations, and also that such an endeavour may be fundamentally difficult. Indeed, a theme that has emerged with existing PNM for ODEs, which can be traced back to Skilling (1992), is the use of approximate and subjective forms for the likelihood. The complex, implicit relationship between the latent ODE solution $\mathrm{y}^{\dagger}$ and the data $f(x_i, y_i)$ arising from the gradient field appears to preclude use of an exact likelihood. Of course, violation of the likelihood principle is not traditionally a concern in the design of a numerical method, yet if the strictly richer output that comes with a Bayesian PNM is desired, then clearly adherence to the likelihood principle is important. It is therefore natural to ask the question, "under what conditions can exact Bayesian inference for ODEs be made?".

This chapter of the thesis presents a proof-of-concept PNM for the numerical solution of a (limited) class of ODEs that is both (a) Bayesian in the sense of Definition 1 and (b) can in principle be implemented. The method being proposed is indicated in Figure 4.1 and its main properties are as follows:

- The classical theory of Lie groups is exploited, for the first time in the context of PNM, to understand when an ODE of the form in Eq. (3.1) can be transformed into an ODE whose gradient field is a function of the independent state variable only, reducing the ODE to an integral.

- For ODEs that admit a solvable Lie algebra, our proposal can be shown to simultaneously perform exact Bayesian inference on both the original and the Lie-transformed ODE. Crucially, as we explain later, to identify a Lie algebra only high-level *a priori* information about the ODE is required. The case of first- and second-order ODEs is presented in detail, but the method itself is general.

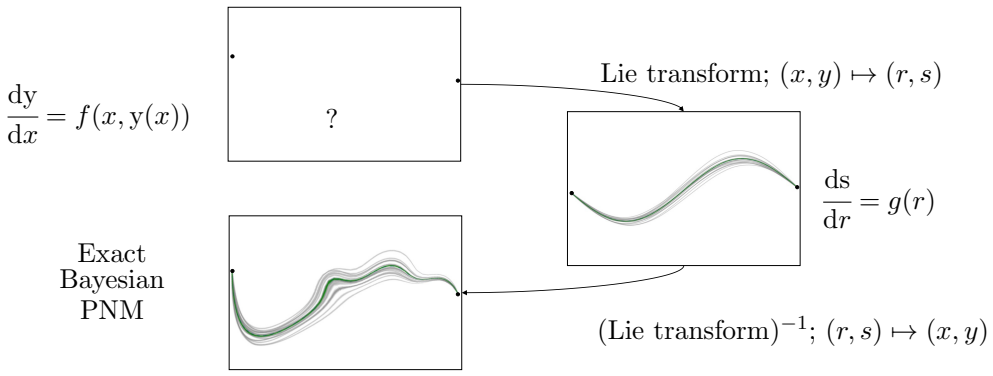



Figure 4.1: Schematic of our proposed approach. An $n$th order ODE that admits a solvable Lie algebra can be transformed into $n$ integrals, to which exact Bayesian probabilistic numerical methods can be applied. The posterior measure on the transformed space is then pushed back through the inverse transformation onto the original domain of interest.

- In general the specification of prior belief can be difficult. The prior distributions that we construct are guaranteed to respect aspects of the structure of the ODE. As such, our priors are, to some extent, automatically adapted to the ODE at hand as opposed to being arbitrarily posited.
- In addition to the benefits conferred in the Bayesian framework, detailed in Section 1.1 and in Cockayne *et al.* (2019), the method being proposed can be computationally realised. On the other hand, there is a cost in terms of the run-time of the method that is substantially larger than existing, non-Bayesian approaches (especially classical numerical methods). As such, we consider this work to be a proof-of-concept rather than an applicable Bayesian PNM.

The remainder of the chapter is structured as follows: Section 4.1 is dedicated to a succinct review of Lie group methods for ODEs. In Section 5.2, Lie group methods are exploited to construct priors over the solution space of the ODE whenever a solvable Lie algebra is admitted and exact Bayesian inference is performed on a transformed version of the original ODE which takes the form of an integral. Numerical experiments are reported in Section 4.3. Finally, some conclusions and recommendations for future research are drawn in Section 4.4.

## 4.1 Overview of Lie group methods

This section provides a succinct overview of classical Lie group methods, introduced in the 19th century by Sophus Lie in the differential equation context (Hawkins, 2012). Lie developed the fundamental notion of a *Lie group of transformations*, which roughly correspond to maps that take one solution of the ODE to another. This provided a formal

generalisation of certain algebraic techniques, such as dimensional analysis and transformations based on spatial symmetries, that can sometimes be employed to algebraically reduce the order of an ODE.

This section proceeds as follows: First, in Section 4.1.1 we introduce a one-parameter Lie group of transformations and then, in Section 4.1.2, we explain what it means for a curve or a surface to be transformation-invariant. In Secion 4.1.3 we recall consequences of Lie's theory in the ODE context. Last, in Section 4.1.4 the generalisation to multi-parameter Lie groups is indicated. Our development is heavily influenced by Bluman & Anco (2002) and we refer the reader to their book when required.

### 4.1.1 One-Parameter Lie Groups of Transformations

The purpose of this section is to recall essential definitions, together with the *first fundamental theorem of Lie*, which relates a Lie group of transformations to its infinitesimal generator. In what follows we consider a fixed domain $D \subset \mathbb{R}^d$ and denote a generic state variable as $x = (x_1, \ldots, x_d) \in D$.

**Definition 6** (One-Parameter Group of Transformations)**.** A *one-parameter group of transformations* on $D$ is a map $X : D \times S \to D$, defined on $D \times S$ for some set $S \subset \mathbb{R}$, together with a bivariate map $\phi : S \times S \to S$, such that the following hold:

(1) For each $\epsilon \in S$, the transformation $X(\cdot, \epsilon)$ is a bijection on $D$.

(2) $(S, \phi)$ forms a group with law of composition $\phi$.

(3) If $\epsilon_0$ is the identity element in $(S, \phi)$, then $X(\cdot, \epsilon_0)$ is the identity map on $D$.

(4) For all $x \in D$, $\epsilon, \delta \in S$, if $x^* = X(x, \epsilon)$, $x^{**} = X(x^*, \delta)$, then $x^{**} = X(x^*, \phi(\epsilon, \delta))$.

In what follows we continue to use the shorthand notation $x^* = X(x, \epsilon)$. The notion of a *Lie* group additionally includes smoothness assumptions on the maps that constitute a group of transformations. Recall that a real-valued function is *analytic* if it can be locally expressed as a convergent power series.

**Definition 7** (One-Parameter Lie Group of Transformations)**.** Let $X$, together with $\phi$, form a one-parameter group of transformations on $D$. Then we say that $X$, together with $\phi$, form a *one-parameter Lie group of transformations* on $D$ if, in addition, the following hold:

(5) $S$ is a (possibly unbounded) interval in $\mathbb{R}$.

(6) For each $\epsilon \in S$, $X(\cdot, \epsilon)$ is infinitely differentiable in $D$.

(7) For each $x \in D$, $X(x, \cdot)$ is an analytic function on $S$.

(8) $\phi$ is analytic in $S \times S$.

Without the loss of generality it will be assumed, through re-parametrisation if required, that $S$ contains the origin and $\epsilon = 0$ is the identity element in $(S, \phi)$. The definition is illustrated through three examples:

**Example 1** (Translation in the x-Axis)**.** The one-parameter transformation $x_1^* = x_1 + \epsilon$, $x_2^* = x_2$ for $\epsilon \in \mathbb{R}$ forms a Lie group of transformations on $D = \mathbb{R}^2$ with group composition law $\phi(\epsilon, \delta) = \epsilon + \delta$.

**Example 2** (Rotation Group)**.** The one-parameter transformation $x_1^* = x_1 \cos(\epsilon) - x_2 \sin(\epsilon)$, $x_2^* = x_1 \sin(\epsilon) + x_2 \cos(\epsilon)$ for $\epsilon \in \mathbb{R}$ again forms a Lie group of transformations on $D = \mathbb{R}^2$ with group composition law $\phi(\epsilon, \delta) = \epsilon + \delta$.

**Example 3** (Cyclic group $C_p$)**.** Let $D = \{1, 2, 3, \ldots, p\}$. Let $S = \mathbb{Z}$. For $n \in D$ and $m \in S$, let $X(n, m) = n + m \pmod p$. Then $X$, together with $\phi(a, b) = a + b$, defines a one parameter group of transformations on $D$, but is not a Lie group of transformations since (5) is violated.

The first fundamental theorem of Lie establishes that a Lie group of transformations can be characterised by its infinitesimal generator, defined next:

**Definition 8** (Infinitesimal Transformation)**.** Let $X$ be a one-parameter Lie group of transformations. Then the transformation $x^* = x + \epsilon\xi(x)$,

$$\xi(x) \quad = \quad \left.\frac{\partial X(x, \epsilon)}{\partial \epsilon}\right|_{\epsilon=0},$$

is called the *infinitesimal transformation* associated to $X$ and the map $\xi$ is called an *infinitesimal*.

**Definition 9** (Infinitesimal Generator)**.** The *infinitesimal generator* of a one-parameter Lie group of transformations $X$ is defined to be the operator $\mathrm{X} = \xi \cdot \nabla$ where $\xi$ is the infinitesimal associated to $X$ and $\nabla = (\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, \ldots, \frac{\partial}{\partial x_n})$ is the gradient.

**Example 4** (Ex. 1, continued)**.** For Ex. 1, we have

$$\xi(x) \quad = \quad \left( \left.\frac{\mathrm{d}x_1^*}{\mathrm{d}\epsilon}\right|_{\epsilon=0}, \left.\frac{\mathrm{d}x_2^*}{\mathrm{d}\epsilon}\right|_{\epsilon=0} \right) \quad = \quad (1, 0)$$

so the infinitesimal generator for translation in the x-axis is $\mathrm{X} = \frac{\partial}{\partial x_1}$.

**Example 5** (Ex. 2, continued)**.** Similarly, the infinitesimal generator for the rotation group is $\mathrm{X} = -x_2 \frac{\partial}{\partial x_1} + x_1 \frac{\partial}{\partial x_2}$.

The first fundamental theorem of Lie provides a constructive route to obtain the infinitesimal generator from the transformation itself:

**Theorem 4.1** (First Fundamental Theorem of Lie; see pages 39-40 of Bluman & Anco (2002))**.** *A one parameter Lie group of transformations $X$ is characterised by the initial value problem:*

$$\frac{\mathrm{d}x^*}{\mathrm{d}\tau} \;=\; \xi(x^*), \qquad x^* = x \text{ when } \tau = 0, \tag{4.1}$$

*where $\tau(\epsilon)$ is a parametrisation of $\epsilon$ which satisfies $\tau(0) = 0$ and, for $\epsilon \neq 0$,*

$$\tau(\epsilon) \;=\; \int_0^\epsilon \frac{\partial \phi(a,b)}{\partial b}\Big|_{(a,b)=(\delta^{-1},\delta)} \mathrm{d}\delta.$$

*Here $\delta^{-1}$ denotes the group inverse element for $\delta$.*

Since Eq. (4.1) is translation-invariant in $\tau$, it follows that without loss of generality we can assume a parametrisation $\tau(\epsilon)$ such that the group action becomes $\phi(\tau_1, \tau_2) = \tau_1 + \tau_2$ and, in particular, $\tau^{-1} = -\tau$. In the remainder of the chapter, for convenience we assume that all Lie groups are parametrised such that the group action is $\phi(\epsilon_1, \epsilon_2) = \epsilon_1 + \epsilon_2$.

The next result can be viewed as a converse to Theorem 4.1, as it shows how to obtain the transformation from the infinitesimal generator.

**Theorem 4.2.** *A one parameter Lie group of transformations with infinitesimal generator* $\mathrm{X}$ *is equivalent to $x^* = e^{\epsilon \mathrm{X}} x$, where $e^{\epsilon \mathrm{X}} = \sum_{k=0}^{\infty} \frac{1}{k!} \epsilon^k \mathrm{X}^k x$.*

*Proof of Theorem 4.2.* From Taylor's theorem we have that

$$\begin{aligned} x^* &= X(x, \epsilon) \\ &= \sum_{k=0}^{\infty} \frac{\epsilon^k}{k!} \frac{\partial^k X(x,\epsilon)}{\partial \epsilon^k}\Big|_{\epsilon=0} \end{aligned}$$

For any differentiable function $F$ we have that

$$\frac{\mathrm{d}F(x^*)}{\mathrm{d}\epsilon} \;=\; \sum_{i=1}^{d} \frac{\partial F(x^*)}{\partial x_i^*} \frac{\mathrm{d}x_i^*}{\mathrm{d}\epsilon} \;=\; \sum_{i=1}^{d} \xi_i \frac{\partial F(x^*)}{\partial x_i^*} \;=\; \mathrm{X}F(x^*)$$

and similarly

$$\frac{\mathrm{d}^k F(x^*)}{\mathrm{d}\epsilon^k} \;=\; \mathrm{X}^k F(x^*).$$

Thus

$$\left.\frac{\partial^k X(x,\epsilon)}{\partial \epsilon^k}\right|_{\epsilon=0} = \mathrm{X}^k x$$

so that the stated result is recovered. □

The following is immediate from the proof of Theorem 4.2:

**Corollary 4.1.** *If $F$ is infinitely differentiable, then $F(x^*) = e^{\epsilon \mathrm{X}} F(x)$.*

### 4.1.2 Invariance Under Transformation

In this section we explain what it means for a curve or a surface to be invariant under a Lie group of transformations and how this notion relates to the infinitesimal generator.

**Definition 10** (Invariant Function)**.** A function $F : D \to \mathbb{R}$ is said to be *invariant* under a one parameter Lie group of transformations $x^* = X(x, \epsilon)$ if $F(x^*) = F(x)$ for all $x \in D$ and $\epsilon \in S$.

Based on the results in Section 4.1.1, one might expect that invariance to a transformation can be expressed in terms of the infinitesimal generator of the transformation. This is indeed the case:

**Theorem 4.3.** *A differentiable function $F : D \mapsto \mathbb{R}$ is invariant under a one parameter Lie group of transformations with infinitesimal generator $\mathrm{X}$ if and only if $\mathrm{X}F(x) = 0$ for all $x \in D$.*

*Proof of Theorem 4.3.* The result is established as follows:

$$\begin{aligned}
F \text{ invariant} &\Leftrightarrow F(x^*) = 0 \text{ whenever } F(x) = 0 \\
&\Leftrightarrow e^{\epsilon \mathrm{X}} F(x) = 0 \text{ whenever } F(x) = 0 \quad \text{(Cor. 4.1)} \\
&\Leftrightarrow F(x) + \epsilon \mathrm{X}F(x) + O(\epsilon^2) = 0 \text{ whenever } F(x) = 0 \quad \text{(Taylor)} \\
&\Leftrightarrow \mathrm{X}F(x) = 0 \text{ whenever } F(x) = 0
\end{aligned}$$

where the last line follows since the coefficient of the $O(\epsilon)$ term in the Taylor expansion must vanish. This completes the proof. □

**Theorem 4.4.** *For a function $F : D \mapsto \mathbb{R}$ and a one parameter Lie group of transformations $x^* = X(x, \epsilon)$, the relation $F(x^*) = F(x) + \epsilon$ holds for all $x \in D$ and $\epsilon \in S$ if and only if $\mathrm{X}F(x) = 1$ for all $x \in D$.*

*Proof of Theorem 4.4.* From Cor. 4.1, we have that $F(x^*) = F(x) + \epsilon \mathrm{X}F(x) + O(\epsilon^2)$. The result follows from inspection of the $\epsilon$ coefficient. □

The following definition is fundamental to the method proposed in Section 5.2:

**Definition 11** (Canonical Coordinates)**.** Consider a coordinate system $r = (r_1(x), \ldots, r_n(x))$ on $D$. Then any one parameter Lie group of transformations $x^* = X(x, \epsilon)$ induces a transformation of the coordinates $r_i^* = r_i(x^*)$. The coordinate system $r$ is called *canonical* for the transformation if $r_1^* = r_1, \ldots, r_{n-1}^* = r_{n-1}$ and $r_n^* = r_n + \epsilon$.

**Example 6** (Ex. 2, continued)**.** For the rotation group in Ex. 2, we have canonical coordinates $r_1(x_1, x_2) = \sqrt{x_1^2 + x_2^2}$ , $r_2(x_1, x_2) = \arctan(x_2/x_1)$.

In canonical coordinates, a one parameter Lie group of transformations can be viewed as a straight-forward translation in the $r_n$-axis. The existence of canonical coordinates is established in Thm. 2.3.5-2 of Bluman & Anco (2002). Note that Thms. 4.3 and 4.4 imply that $\mathrm{X}r_i^* = 0$ for $i = 1, 2, ..., n-1$, $\mathrm{X}r_n^* = 1$.

**Definition 12** (Invariant Surface)**.** For a function $F : D \to \mathbb{R}$, a surface defined by $F(x) = 0$ is said to be *invariant* under a one parameter Lie group of transformation $x^* = X(x, \epsilon)$ if and only if $F(x^*) = 0$ whenever $F(x) = 0$ for all $x \in D$ and $\epsilon \in S$.

The invariance of a surface, as for a function, can be cast in terms of an infinitesimal generator:

**Corollary 4.2.** *A surface $F(x) = 0$ is invariant under a one parameter Lie group of transformations with infinitesimal generator* X *if and only if* $\mathrm{X}F(x) = 0$ *whenever* $F(x) = 0$.

### 4.1.3 Symmetry Methods for ODEs

The aim of this section is to relate Lie transformations to ODEs for which these transformations are admitted. These techniques form the basis for our proposed method in Section 5.2.

For an ODE of the form in Eq. (3.1), one can consider the action of a transformation on the coordinates $(x, y)$; i.e. a special case of the above framework where the generic coordinates $x_1$ and $x_2$ are respectively the independent $(x)$ and dependent $(y)$ variables of the ODE. It is clear that such a transformation also implies some kind of transformation of the derivatives $y_m := \frac{\mathrm{d}^m y}{\mathrm{d}x^m}$. Indeed, consider a one-parameter Lie group of transformations $(x^*, y^*) = (X(x, y; \epsilon), Y(x, y; \epsilon))$. Then we have from the chain rule that $y_m^* := \frac{\mathrm{d}^m y^*}{\mathrm{d}(x^*)^m}$ is a function of $x, y, y_1, \ldots, y_m$ and we denote $y_m^* = Y_m(x, y, y_1, \ldots, y_m; \epsilon)$. As an explicit example:

$$y_1^* \;=\; \frac{\mathrm{d}y^*}{\mathrm{d}x^*} \;=\; \frac{\frac{\partial Y(x,y;\epsilon)}{\partial x} + y_1 \frac{\partial Y(x,y;\epsilon)}{\partial y}}{\frac{\partial X(x,y;\epsilon)}{\partial x} + y_1 \frac{\partial X(x,y;\epsilon)}{\partial y}} \;=:\; Y_1(x, y, y_1; \epsilon)$$

In general:

$$y_m^* = \frac{\frac{\partial y_{m-1}^*}{\partial x} + y_1 \frac{\partial y_{m-1}^*}{\partial y} + y_2 \frac{\partial y_{m-1}^*}{\partial y_1} + ... + y_m \frac{\partial y_{m-1}^*}{\partial y_{m-1}}}{\frac{\partial X(x,y;\epsilon)}{\partial x} + y_1 \frac{\partial X(x,y;\epsilon)}{\partial y}} =: Y_m(x, y, y_1, \ldots, y_m; \epsilon)$$

In this sense a transformation defined on $(x, y)$ can be naturally extended to a transformation on $(x, y, y_1, y_2, \ldots)$ as required.

**Definition 13** (Admitted Transformation)**.** An $m$th order ODE $F(x, y, y_1, \ldots, y_m) = 0$ is said to *admit* a one parameter Lie group of transformations $(x^*, y^*) = (X(x, y; \epsilon), Y(x, y; \epsilon))$ if the surface $F$ defined by the ODE is invariant under the Lie group of transformations, i.e. if $F(x^*, y^*, y_1^*, \ldots, y_m^*) = 0$ whenever $F(x, y, y_1, \ldots, y_m) = 0$.

**Example 7.** Clearly any ODE of the form $\frac{\mathrm{d}y}{\mathrm{d}x} = F(x)$ admits the transformation $(x^*, y^*) = (x, y + \epsilon)$.

Our next task is to understand how the infinitesimal generator of a transformation can be extended to act on derivatives $y_m$.

**Definition 14** (Extended Infinitesimal Transformation)**.** The $m$th *extended infinitesimals* of a one parameter Lie group of transformations $(x^*, y^*) = (X(x, y; \epsilon), Y(x, y; \epsilon))$ are defined as the functions $\xi, \eta, \eta^{(1)}, \ldots, \eta^{(m)}$ for which the following equations hold:

$$\begin{aligned}
x^* = X(x, y; \epsilon) &= x + \epsilon\xi(x, y) + O(\epsilon^2) \\
y^* = Y(x, y; \epsilon) &= y + \epsilon\eta(x, y) + O(\epsilon^2) \\
y_1^* = Y_1(x, y, y_1; \epsilon) &= y_1 + \epsilon\eta^{(1)}(x, y, y_1) + O(\epsilon^2) \\
&\vdots \\
y_m^* = Y_m(x, y, y_1, \ldots, y_m; \epsilon) &= y_m + \epsilon\eta^{(m)}(x, y, y_1, y_2, \ldots, y_m) + O(\epsilon^2)
\end{aligned}$$

It can be shown straightforwardly via induction that

$$\eta^{(m)}(x, y, y_1, y_2, \ldots, y_m) = \frac{\mathrm{d}^m \eta}{\mathrm{d}x^m} - \sum_{k=0}^{m} \frac{m!}{(m-k)!k!} y_{m-k-1} \frac{\mathrm{d}^k \xi}{\mathrm{d}x^k} \tag{4.2}$$

where $\frac{\mathrm{d}}{\mathrm{d}x}$ denotes the full derivative with respect to $x$, i.e. $\frac{\mathrm{d}}{\mathrm{d}x} = \frac{\partial}{\partial x} + y_1 \frac{\partial}{\partial y} + \sum_{k=2}^{m+1} y_k \frac{\partial}{\partial y_{k-1}}$. It follows that $\eta^{(m)}$ is a polynomial in $y_1, y_2, \ldots, y_m$ with coefficients linear combinations of $\xi, \eta$ and their partial derivatives up to the $m$th order.

**Definition 15** (Extended Infinitesimal Generator)**.** The $m$th *extended infinitesimal gen-*

*erator* is defined as

$$\begin{aligned} \mathrm{X}^{(m)} &= \xi_m(x, y, y_1, \ldots, y_m) \cdot \nabla \\ &= \xi(x,y)\frac{\partial}{\partial x} + \eta(x,y)\frac{\partial}{\partial y} + \eta^{(1)}(x,y)\frac{\partial}{\partial y_1} + \cdots + \eta^{(m)}(x, y, y_1, \ldots, y_m)\frac{\partial}{\partial y_m} \end{aligned}$$

where $\nabla = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial y_1}, \ldots, \frac{\partial}{\partial y_m})$ is the extended gradient.

The following corollaries are central to the actual computation of the admitted Lie groups of an ODE.

**Corollary 4.3.** *A differentiable function $F : D_m \to \mathbb{R}$ where $D_m$ is the phase space containing elements of the form $(x, y, y_1, \ldots, y_m)$, is invariant under a one parameter Lie group of transformations with an extended infinitesimal generator $\mathrm{X}^{(m)}$ if and only if $\mathrm{X}^{(m)}F(x, y, y_1, \ldots, y_m) = 0$ for all $(x, y, y_1, \ldots, y_m) \in D_m$.*

**Corollary 4.4** (Infinitesimal Criterion for Symmetries Admitted by an ODE)**.** *A one parameter Lie group of transformations is admitted by the $m$th order ODE $F(x, y, y_1, .., y_m) = 0$ if and only if its extended infinitesimal generator $\mathrm{X}^{(m)}$ satisfies $\mathrm{X}^{(m)}F(x, y, y_1, \ldots, y_m) = 0$ whenever $F(x, y, y_1, \ldots, y_m) = 0$.*

### 4.1.4 Multi-Parameter Lie Groups and Lie Algebras

To leverage the full power of Lie symmetry methods for ODEs of order $m \geq 2$, we need to consider multiple Lie symmetries which are collectively described by a *Lie algebra*. Fortunately, the notion of a multi-parameter Lie group of transformations is a natural generalisation from the one parameter case. Thus, this last section of background material concerns the generalisation of the definitions in Section 4.1.1 to the case of a multi-parameter Lie group. The associated Lie algebra will also be defined.

**Definition 16** (Multi-Parameter Lie Group of Transformations)**.** The set of transformations $x^* = X(x, \epsilon)$ where $x_i^* = X_i(x, \epsilon)$ and $\epsilon = (\epsilon_1, \epsilon_2, \ldots, \epsilon_r) \in S \subset \mathbb{R}^r$ is called a *$r$-parameter Lie group of transformations* if it satisfies the same axioms as in the one parameter case, but with law of composition $\phi(\epsilon, \delta) = (\phi_1(\epsilon, \delta), \ldots, \phi_r(\epsilon, \delta))$, and (without loss of generality) $\epsilon = (0, 0, ..., 0)$ as the group identity element.

**Definition 17** (Infinitesimal Matrix)**.** The appropriate generalisation for the infinitesimal transformation is the infinitesimal matrix $\Xi = [\xi_{ij}]$, whose entries are defined as $\xi_{ij}(x) = \left.\frac{\partial X_j(x,\epsilon)}{\partial \epsilon_i}\right|_{\epsilon=0}$.

**Definition 18** (Infinitesimal Generator)**.** An $r$-parameter Lie group of transformations is associated with $r$ infinitesimal generators, $\mathrm{X_i}$, defined as $\mathrm{X}_i = \mathrm{X}_i(x) = \sum_{j=1}^{d} \xi_{ij}(x)\frac{\partial}{\partial x_j}$.

The first fundamental theorem of Lie can be generalised to the multi-parameter case. In particular, it can be shown that an $r$-parameter Lie group of transformations is characterised by the set of its $r$ infinitesimal generators. The generalisation is straight-forward and so, for brevity, we refer the reader to pages 39-40 of Bluman & Anco (2002).

Next we explain how the collection of infinitesimal generators forms a Lie algebra. This relies on the basic facts that the set $\mathcal{D}$ of differential operators on $D$ is a vector space over $\mathbb{R}$ (i.e. $\lambda \mathrm{X} + \mu \mathrm{Y} \in \mathcal{D}$ for all $\mathrm{X}, \mathrm{Y} \in \mathcal{D}$ and $\lambda, \mu \in \mathbb{R}$) and that differential operators can be composed (i.e. $\mathrm{XY} \in \mathcal{D}$ for all $\mathrm{X}, \mathrm{Y} \in \mathcal{D}$).

**Definition 19** (Commutator)**.** The *commutator* of two infinitesimal generators $\mathrm{X}_i$ and $\mathrm{X}_j$ is defined as $[\mathrm{X}_i, \mathrm{X}_j] = \mathrm{X}_i\mathrm{X}_j - \mathrm{X}_j\mathrm{X}_i$.

**Theorem 4.5** (Second Fundamental Theorem of Lie; see page 78 of Bluman & Anco (2002))**.** *Consider an $r$-parameter Lie group of transformations and let $\mathcal{L}$ denote the linear span of the infinitesimal generators $\mathrm{X}_1, \ldots, \mathrm{X}_r$ in $\mathcal{D}$. Let $[\cdot, \cdot] : \mathcal{L} \times \mathcal{L} \to \mathcal{D}$ denote the unique bilinear operator that agrees with Def.* (19) *on the set of infinitesimal generators. i.e.*

$$\left[\sum_{i=1}^{r} \lambda_i \mathrm{X}_i, \sum_{j=1}^{r} \mu_j \mathrm{X}_j\right] = \sum_{i=1}^{r}\sum_{j=1}^{r} \lambda_i \mu_j (\mathrm{X}_i\mathrm{X}_j - \mathrm{X}_j\mathrm{X}_i). \tag{4.3}$$

*Then $[\cdot, \cdot]$ maps into $\mathcal{L}$. i.e. the right hand side of Eq.* (4.3) *belongs to $\mathcal{L}$ for all $\lambda, \mu \in \mathbb{R}^r$.*

**Example 8.** Consider the two parameter group of transformations on $D = \mathbb{R}^2$ given by $(x^*, y^*) = (x + x\epsilon + x^2\delta, y + y\epsilon + y^2\delta)$. The infinitesimal generators corresponding to $\delta$ and $\epsilon$, respectively, are $\mathrm{X}_1 = x^2\frac{\partial}{\partial x} + y^2\frac{\partial}{\partial y}$, $\mathrm{X}_2 = x\frac{\partial}{\partial x} + y\frac{\partial}{\partial y}$. It can be directly verified that $[\mathrm{X}_1, \mathrm{X}_2] = -\mathrm{X}_1$.

The space $\mathcal{L}$, defined in Thm. 4.5, satisfies the properties of an $r$-dimensional Lie algebra $\mathcal{L}$, defined next:

**Definition 20** (Lie Algebra)**.** An $r$-dimensional vector space $\mathcal{L}$ over $\mathbb{R}$ together with a bilinear operator $[\cdot, \cdot] : \mathcal{L} \times \mathcal{L} \to \mathcal{L}$ is called an $r$-dimensional *Lie algebra* if the following hold:

(1) Alternativity: $[\mathrm{X}, \mathrm{X}] = 0$ for all $\mathrm{X} \in \mathcal{L}$

(2) Jacobi Identity: $[\mathrm{X}, [\mathrm{Y}, \mathrm{Z}]] + [\mathrm{Y}, [\mathrm{Z}, \mathrm{X}]] + [\mathrm{Z}, [\mathrm{X}, \mathrm{Y}]] = 0$ for all $\mathrm{X}, \mathrm{Y}, \mathrm{Z} \in \mathcal{L}$

In general, for the methods presented in Section 5.2 to be applied, existence of an $n$-parameter Lie group of transformations is not in itself sufficient; we require the existence of an $n$-dimensional *solvable* Lie sub-algebra, defined next:

**Definition 21** (Normal Lie Sub-algebra)**.** Consider a Lie sub-algebra $\mathcal{J}$ of a Lie algebra $\mathcal{L}$ with bilinear operator $[\cdot,\cdot]$, i.e. a subset $\mathcal{J} \subset \mathcal{L}$ such that, when equipped with the restriction of $[\cdot,\cdot]$ to $\mathcal{J} \times \mathcal{J}$, is itself a Lie algebra and, in particular, $[\mathrm{X},\mathrm{Y}] \in \mathcal{J}$ for all $\mathrm{X},\mathrm{Y} \in \mathcal{J}$. Then $\mathcal{J}$ is said to be *normal* if, in addition, $[\mathrm{X},\mathrm{Y}] \in \mathcal{J}$ for all $\mathrm{X} \in \mathcal{J}, \mathrm{Y} \in \mathcal{L}$.

**Definition 22** (Solvable Lie Algebra)**.** An $r$-dimensional Lie algebra $\mathcal{L}$ is called *solvable* if there exists a chain of sub-algebras $\mathcal{L}^1 \subset \mathcal{L}^2 \subset ... \subset \mathcal{L}^{q-1} \subset \mathcal{L}^r =: \mathcal{L}$ such that $\mathcal{L}^{i-1}$ is a normal sub-algebra of $\mathcal{L}^i$ for $i = 2, 3, ..., r$.

For low-order ODEs, the existence requirement for an admitted Lie group of transformations is more restrictive than the requirement that the associated Lie algebra is solveable. Indeed, we have the following result:

**Theorem 4.6.** *All two-dimensional Lie Algebras are solvable.*

*Proof of Theorem 4.6.* Suppose $\mathcal{L}$ is a two dimensional Lie Algebra generated by linearly independent infinitesimal generators $\mathrm{X}_1$ and $\mathrm{X}_2$. Let $\mathrm{Y} = [\mathrm{X}_1, \mathrm{X}_2] = a\mathrm{X}_1 + b\mathrm{X}_2$ and let $\mathcal{J}$ be the one dimensional subalgebra generated by Y. Suppose $\mathrm{Z} = c\mathrm{X}_1 + d\mathrm{X}_2$ is some element of $\mathcal{L}$, then

$$\begin{aligned}
[\mathrm{Y},\mathrm{Z}] &= [Y, c\mathrm{X}_1 + d\mathrm{X}_2] \\
&= c[\mathrm{Y},\mathrm{X}_1] + d[\mathrm{Y},\mathrm{X}_2] \\
&= cb[\mathrm{X}_2,\mathrm{X}_1] + da[\mathrm{X}_1,\mathrm{X}_2] \\
&= (ad - bc)\mathrm{Y} \in \mathcal{J}
\end{aligned}$$

So $\mathcal{J} \subset \mathcal{L}$ is normal, thus $\mathcal{L}$ is solvable, as claimed. □

This completes our review of background material. The exact Bayesian PNM developed in Section 5.2 for an $n$th order ODE require the existence of an admitted $n$-parameter Lie group of transformations with a solvable Lie algebra. In practice we therefore require some high-level information on the gradient field $f$, in order to establish which transformations of the ODE may be admitted. In addition, the requirement of a solvable Lie algebra also limits the class of ODEs for which our exact Bayesian methods can be employed. Nevertheless, this class of ODEs is sufficiently broad to have merited extensive theoretical research (Bluman & Anco, 2002) and the development of software (Baumann, 2013).

## 4.2 Methods

In this section our novel Bayesian PNM is presented. The method relies on high-level information about the gradient field $f$ and, in Section 4.2.1, we discuss how such information can be exploited to identify any Lie transformations that are admitted by the ODE. In the

case of a first order ODE, any non-trivial transformation is sufficient for our method and an explicit information operator is provided for this case, together with recommendations for prior construction, in Section 4.2.2. Together, the prior and the information operator uniquely determine a Bayesian PNM, as explained in Section 1.1. In the general case of an $m$th order ODE, we require that an $m$-dimensional solvable Lie algebra is admitted by the ODE. The special case $m = 2$ is treated in detail, with an explicit information operator and guidance for prior construction provided in Section 4.2.3. In Section A.1 of the Appendix the selection of input pairs $(x_i, y_i)$ to the gradient field is discussed.

### 4.2.1 From an ODE to its Admitted Transformations

For the methods proposed in this chapter, transformations admitted by the ODE, together with their infinitesimal generators, must first be obtained. The algorithm for obtaining infinitesimal generators follows as a consequence of Cor. 4.4. Indeed, suppose we have a $m$th order ODE of the form $y_m - f(x, y, y_1, \ldots, y_{m-1}) = 0$. Then, by Cor. 4.4, a transformation with infinitesimal generator X is admitted by the ODE if and only if:

$$\mathrm{X}^{(m)}(y_m - f(x, y, y_1, \ldots, y_{m-1})) = 0 \quad \text{whenever} \quad y_m = f(x, y, y_1, \ldots, y_{m-1}). \tag{4.4}$$

In infinitesimal notation, Eq. (4.4) is equivalent to

$$\eta^{(m)}(x, y, y_1, \ldots, y_{m-1}, y_m) = \xi \frac{\partial f}{\partial x} + \eta \frac{\partial f}{\partial y} + \sum_{k=1}^{m-1} \eta^{(k)} \frac{\partial f}{\partial y_k}. \tag{4.5}$$

The direct solution of Eq. (4.5) recovers any transformations that are admitted.

In the common scenario where $f(x, y, y_1, \ldots, y_{m-1})$ is a polynomial in $y_1, y_2, \ldots, y_{m-1}$, the algorithm just described, for identification of admitted transformations, can be fully automated (c.f. Baumann, 2013). Indeed, from Def. 14 it follows that the extended infinitesimals $\eta^{(k)}$ for $k \in 1, 2, 3, \ldots, m$ are polynomial in $y_1, y_2, \ldots, y_k$. Thus, by substituting $y_m = f(x, y, y_1, \ldots, y_{m-1})$, Eq. (4.4) too must be a polynomial in $y_1, y_2, \ldots, y_{m-1}$. Moreover, the coefficients of this polynomial must vanish because (4.4) holds for arbitrary values of $x, y, y_1, \ldots, y_{m-1}$. This argument, of setting coefficients to zero, leads to a system of linear partial differential equations (overdetermined when $m \geq 2$) for $\xi(x, y)$ and $\eta(x, y)$, which can be exactly solved to retrieve all the infinitesimal generators of the ODE. The same strategy can often be applied beyond the polynomial case and explicit worked examples of this procedure are now provided:

**Example 9** (First Order ODE)**.** Consider the class of all first order ODEs of the form $\frac{\mathrm{dy}}{\mathrm{d}x} = f(x, \mathrm{y}(x))$, $f(x, y) = F\left(\frac{y}{x}\right)$. From Eq. (4.2), we have $\eta^{(1)} = \eta_x + (\eta_y - \xi_x)y_1 - \xi_y(y_1)^2$ so Eq. (4.5) becomes $\eta_x + (\eta_y - \xi_x)f - \xi_y(f)^2 = \xi\frac{\partial f}{\partial x} + \eta\frac{\partial f}{\partial y}$ and thus $\eta_x + (\eta_y -$

$\xi_x)F\left(\frac{y}{x}\right) - \xi_y F\left(\frac{y}{x}\right)^2 = -\xi F'\left(\frac{y}{x}\right)\frac{y}{x^2} + \eta F'\left(\frac{y}{x}\right)\frac{1}{x}$. For this equation to hold for general $F$, the coefficients of $F$, $F^2$ and $F'$ must vanish: $\eta_x = 0$, $\eta_y - \xi_x = 0$, $\xi_y = 0$, $-\xi\frac{y}{x^2} + \eta\frac{1}{x} = 0$. This is now a linear system of partial differential equations in $(\xi, \eta)$ which is easily solved; namely $\xi = x, \eta = y$. The associated infinitesimal generator is $\mathrm{X} = x\frac{\partial}{\partial x} + y\frac{\partial}{\partial y}$.

**Example 10** (Second Order ODE)**.** The infinitesimal generators for the second order ODE

$$(x - \mathrm{y}(x))\frac{\mathrm{d}^2\mathrm{y}}{\mathrm{d}x^2} + 2\frac{\mathrm{dy}}{\mathrm{d}x}\left(\frac{\mathrm{dy}}{\mathrm{d}x} + 1\right) + \left(\frac{\mathrm{dy}}{\mathrm{d}x}\right)^{3/2} = 0 \tag{4.6}$$

are derived in 13 of the Appendix.

### 4.2.2 The Case of a First Order ODE

In this section we present our approach for a first order ODE. This allows some of the more technical details associated to the general case to be omitted, due to the fact that any one-dimensional Lie algebra is trivial. The main result that will allow us to construct an exact Bayesian PNM is as follows:

**Theorem 4.7** (Reduction of a First Order ODE to an Integral)**.** *If a first order ODE*

$$\frac{\mathrm{dy}}{\mathrm{d}x} = f(x, \mathrm{y}(x)) \tag{4.7}$$

*admits a one parameter Lie group of transformations, then there exists coordinates $r(x, y)$, $s(x, y)$ such that*

$$\frac{\mathrm{ds}}{\mathrm{d}r} = G(r) \tag{4.8}$$

*for some explicit function $G(r)$.*

*Proof of Theorem 4.7.* Let the infinitesimal generator associated with the Lie group of transformations be denoted X. From the remarks after Def. 11, we can obtain canonical coordinates by solving the pair of first order partial differential equations $\mathrm{X}r = 0$, $\mathrm{X}s = 1$. By the chain rule we have

$$\frac{\mathrm{d}s}{\mathrm{d}r} = \frac{s_x + s_y y'}{r_x + r_y y'} =: G(r, s)$$

From the definition of canonical coordinates, the Lie group of transformations is $r^* = r$, $s^* = s + \epsilon$ in the transformed coordinate system, so

$$\frac{\mathrm{d}s^*}{\mathrm{d}r^*} = G(r^*, s^*) \quad \Longrightarrow \quad \frac{\mathrm{d}s}{\mathrm{d}r} = G(r, s + \epsilon)$$

for all $\epsilon$, which implies $G(r, s) = G(r)$ and thus Eq. (4.7) becomes

$$\frac{\mathrm{d}s}{\mathrm{d}r} \quad = \quad G(r)$$

as claimed. □

Note that the transformed ODE in Eq. (4.8) is nothing more than an integral, for which exact Bayesian PNM have already been developed (e.g. Briol *et al.*, 2019; Karvonen *et al.*, 2018). At a high level, as indicated in Fig. 4.1, our proposed Bayesian PNM performs inference for the solution $\mathrm{s}(r)$ of Eq. (4.8) and then transforms the resultant posterior back into the original $(x, y)$-coordinate system. Our PNM is therefore based on the information operator

$$A(\mathrm{y}) \quad = \quad [G(r_0), \ldots, G(r_n)]^\top \in \mathcal{A} = \mathbb{R}^{n+1} \tag{4.9}$$

which corresponds indirectly to $n+1$ evaluations of the original gradient field $f$ at certain input pairs $(x_i, y_i)$. The selection of the inputs $r_i$ is discussed in Section A.1 of the Appendix.

The transformation of a first order ODE is clearly illustrated in the following:

**Example 11** (Ex. 9, continued)**.** Consider the first order ODE $\frac{\mathrm{dy}}{\mathrm{d}x} = f(x, \mathrm{y}(x))$, $f(x, y) = F\left(\frac{y}{x}\right)$. Recall from Ex. 9 that this ODE admits the one parameter Lie group of transformations $x^* = \alpha x$, $y^* = \alpha y$ for $\alpha \in \mathbb{R}$ and the associated infinitesimal generator is $\mathrm{X} = x\frac{\partial}{\partial x} + y\frac{\partial}{\partial y}$. Solving the pair of partial differential equations $\mathrm{X}r = 0, \mathrm{X}s = 1$ yields the canonical coordinates $s = \log y$, $r = \frac{y}{x}$. The transformed ODE is then $\frac{\mathrm{d}s}{\mathrm{d}r} = \frac{F(r)}{-r^2 + rF(r)} =: G(r)$. Thus an evaluation $G(r)$ corresponds to an evaluation of $f(x, y)$ at an input $(x, y)$ such that $r = \frac{y}{x}$.

Two important points must now be addressed: First, the approach just described cannot be Bayesian unless it corresponds to a well-defined prior distribution $\mu \in \mathcal{P}_{\mathcal{Y}}$ in the original coordinate system $\mathcal{Y}$. This precludes standard (e.g. Gaussian process) priors in general, as such priors assign mass to functions in $(r, s)$-space that do not correspond to well-defined functions in $(x, y)$-space (see Fig. 4.2). Second, any prior that is used ought to be consistent with the Lie group of transformations that the ODE is known to admit. To address each of these important points, we propose two general principles for prior construction in this work. The first principle is the *implicit prior* principle. This ensures that a prior specified in the transformed coordinates $(r, s)$ can be safely transformed into a well-defined distribution on $\mathcal{Y}$. For such an implicit prior to be well-defined we need to understand when a function in $(r, s)$ space maps to a well-defined function in the original $(x, y)$ domain of interest. Let $\mathcal{S}$ denote the image of $\mathcal{Y}$ under the canonical coordinate

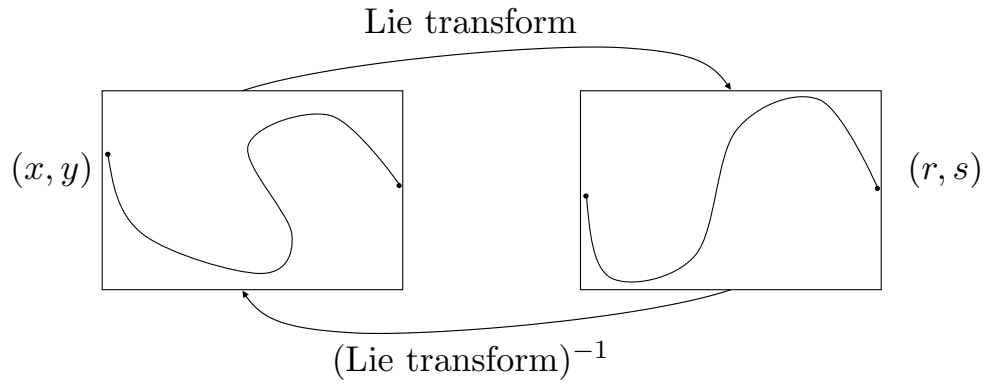



Figure 4.2: Illustration of the implicit prior principle: A prior elicited for the function $\mathrm{s}(r)$ in the transformed coordinate system $(r, s)$ must be supported on functions $\mathrm{s}(r)$ that correspond to well-defined functions $\mathrm{y}(x)$ in the original coordinate system $(x, y)$. Thus the situation depicted would not be allowed.

transformation.

**Principle 1** (Implicit Prior)**.** A distribution $\nu \in \mathcal{P}_{\mathcal{S}}$ on the transformed solution space $\mathcal{S}$ corresponds to a well-defined *implicit prior* $\mu \in \mathcal{P}_{\mathcal{Y}}$ provided that $x(r, \mathrm{s}(r))$ is strictly monotone as a function of $r$.

**Example 12** (Ex. 11, continued)**.** For the ODE in Ex. 11, with canonical coordinates $s = \log y$, $r = \frac{y}{x}$, if $x \in [x_0, x_T] = [1, x_T]$ and $y \in (0, \infty)$, then the region in the $(r, s)$ plane corresponding to $[1, x_T] \times (0, \infty)$ in the $(x, y)$ plane is $(0, \infty) \times \mathbb{R}$. Now,

$$\frac{\mathrm{d}x(r, \mathrm{s}(r))}{\mathrm{d}r} = \frac{\partial x}{\partial r} + \frac{\partial x}{\partial s}\frac{\mathrm{d}\mathrm{s}(r)}{\mathrm{d}r} = \frac{r\mathrm{s}'(r)\exp(\mathrm{s}(r)) - \exp(\mathrm{s}(r))}{r^2}.$$

Thus $\frac{\mathrm{d}x}{\mathrm{d}r} > 0$ if and only if $\mathrm{s}'(r) > \frac{1}{r}$ and the domain restriction $x \in [x_0, x_T]$ requires that we respect the constraint $\log(r) \leq \mathrm{s}(r) \leq \log(r) + \log(x_T)$ for all $r > 0$. The set $\mathcal{S}$ must therefore consists of differentiable functions s defined on $r \in (0, \infty)$ and satisfying $\log(r) \leq \mathrm{s}(r) \leq \log(r) + \log(x_T)$.

Now we turn to the second important point, namely that the prior ought to encode knowledge about any Lie transformations that are known to be admitted by the ODE. In working on the transformed space $\mathcal{S}$, it become clear how to construct a prior measure in which this knowledge is encoded. Our second principle for prior specification states that equal prior weight should be afforded to all curves that are identical up to a Lie transformation:

**Principle 2** (Invariant Prior)**.** A distribution $\nu \in \mathcal{P}_{\mathcal{S}}$ on the transformed solution space $\mathcal{S}$ is said to be *invariant* provided that $\nu(S) = \nu(S + \epsilon)$ where the elements of $S + \epsilon$ are the elements of $S$ after a vertical translation; i.e. $\mathrm{s}(\cdot) \mapsto \mathrm{s}(\cdot) + \epsilon$ and both $S, S + \epsilon \in \Sigma_{\mathcal{S}}$.

Our recommendation is that, when possible, both the implicit prior principle and the invariant prior principle should be enforced. However, in practice it seems difficult to satisfy both principles and our empirical results in Section 4.3 are based on implicit priors that are not invariant.

### 4.2.3 The Case of a Second Order ODE

In this section we present our approach for a second order ODE. The study of second order ODEs is particularly important, since Newtonian mechanics is based on ODEs of second order. The presentation is again simplified relative to the general case of an $m$th order ODE, this time by virtue of the fact that any two dimensional Lie algebra is guaranteed to be solveable (Thm. 4.6). The main result that will allow us to construct an exact Bayesian PNM is as follows:

**Theorem 4.8** (Reduction of a Second Order ODE to Two Integrals)**.** *If a second order ODE*

$$\frac{\mathrm{d}^2\mathrm{y}}{\mathrm{d}x^2} = f\left(x, \mathrm{y}(x), \frac{\mathrm{dy}}{\mathrm{d}x}\right) \tag{4.10}$$

*admits a two parameter Lie group of transformations, then there exists coordinates $r(x, y)$, $s(x, y)$ such that*

$$\frac{\mathrm{ds}}{\mathrm{d}r} = G(r) \tag{4.11}$$

*for some implicitly defined function $G$. The function $G$ is explicitly related to the solution of a second equation of the form*

$$\frac{\mathrm{d\tilde{s}}}{\mathrm{d}\tilde{r}} = H(\tilde{r}) \tag{4.12}$$

*for some explicit function $H(\tilde{r})$.*

*Proof of Theorem 4.8.* Let the infinitesimal generators associated with the Lie group of transformations be denoted $\mathrm{X}_1$ and $\mathrm{X}_2$. Recall from Thm. 4.6 that any two dimensional Lie algebra is solvable. Thus, without loss of generality we may assume

$$[\mathrm{X}_1, \mathrm{X}_2] = \lambda \mathrm{X}_1 \tag{4.13}$$

for some $\lambda \in \mathbb{R}$. The infinitesimal generators $\mathrm{X}_1$ and $\mathrm{X}_2$ each correspond to a one parameter Lie group of transformations, denoted $x^* = X_1(x, \epsilon_1)$ and $x^\dagger = X_2(x, \epsilon_2)$. Let $v(x, y)$, $w(x, y, y_1)$ be invariant functions of $\mathrm{X}_1$ and its extension $\mathrm{X}_1^{(1)}$, respectively, so $v(x^*, y^*) = v(x, y)$ and $w(x^*, y^*, y_1^*) = w(x, y, y_1)$, where $w$ has a non-trivial dependence on its third

argument. It follows from the definition of invariance that

$$\frac{\mathrm{d}w^*}{\mathrm{d}v^*} = \frac{\mathrm{d}w}{\mathrm{d}v},$$

which is equivalent to

$$\mathrm{X}_1^{(1)}\frac{\mathrm{d}w}{\mathrm{d}v} = 0 \tag{4.14}$$

by Cor. 4.3. Now because Eq. (4.14) is a homogeneous partial differential equation, the general solution $\frac{\mathrm{d}w}{\mathrm{d}v}$ can be expressed as a function of the two solutions $v(x, y)$ and $w(x, y, y_1)$. Therefore

$$\frac{\mathrm{d}w}{\mathrm{d}v} = Z(v, w) \tag{4.15}$$

for some undetermined function $Z$.

Since Eq. (4.10) admits $\mathrm{X}_2$, and Eq. (4.15) is the same ODE when expressed in terms of $x, y, y_1$, it must be the case that

$$\mathrm{X}_2^{(2)}\left(\frac{\mathrm{d}w}{\mathrm{d}v} - Z(v, w)\right) = 0 \qquad \text{whenever} \qquad \frac{\mathrm{d}w}{\mathrm{d}v} = Z(v, w).$$

Then from Cor. 4.4 it follows that $\mathrm{X}_2^{(1)}$ is admitted by the first order ODE in Eq. (4.15). Thus we are now faced with a first order ODE that admits a one parameter Lie group of transformations, as in Thm. 4.7.

Now, from Eq. (4.13), we have $\mathrm{X}_1\mathrm{X}_2 v = \mathrm{X}_2\mathrm{X}_1 v + \lambda \mathrm{X}_1 v = 0$. Thus $\mathrm{X}_2 v$ is an invariant of $\mathrm{X}_1$ and so $\mathrm{X}_2 v = h(v)$ for some function $h$. Similarly $\mathrm{X}_1^{(1)}\mathrm{X}_2^{(1)} v = \mathrm{X}_2^{(1)}\mathrm{X}_1^{(1)} v + \lambda \mathrm{X}_1^{(1)} v = 0$, so that $\mathrm{X}_2^{(1)} w = g(v, w)$, for some function $g$. This implies $\mathrm{X}_2^{(1)} = h(v)\frac{\partial}{\partial v} + g(v, w)\frac{\partial}{\partial w}$.

Proceeding as in Thm. 4.7, denote the canonical coordinates of $\mathrm{X}_2^{(1)} = h(v)\frac{\partial}{\partial v} + g(v, w)\frac{\partial}{\partial w}$ by $\tilde{r}(v, w)$, $\tilde{s}(v, w)$ such that $\mathrm{X}_2^{(1)}\tilde{r} = 0$, $\mathrm{X}_2^{(1)}\tilde{s} = 1$. In canonical coordinates, Eq. (4.15) becomes:

$$\frac{\mathrm{d}\tilde{\mathrm{s}}}{\mathrm{d}\tilde{r}} = H(\tilde{r}) \tag{4.16}$$

This is again an integral, with solution

$$\tilde{\mathrm{s}}(\tilde{r}) = \int^{\tilde{r}} H(t)\mathrm{d}t + C. \tag{4.17}$$

We can rewrite Eq. (4.17) in terms of $v, w$ to obtain an equation of the form

$$I(v, w) = 0 \tag{4.18}$$

which satisfies $\mathrm{X}_1^{(1)}(I(v,w)) = 0$ whenever $I(v,w) = 0$, since recall $v, w$ are invariants of $X_1^{(1)}$. For the final step, we recall that $v = v(x,y)$ and $w = w(x,y,y_1)$, so that Eq. (4.18) represents a first order ODE in $y$, which admits $\mathrm{X}_1$. Thus we can apply Thm. 4.7 a second time to obtain canonical coordinates $r(x,y)$, $s(x,y)$ for $X_1$. In these coordinates, Eq. (4.18) reduces into the form

$$\frac{\mathrm{d}s}{\mathrm{d}r} \quad = \quad G(r)$$

where $G$ is implicitly defined. □

Note that the ODE in Eq. (4.10) is reduced to two integrals, namely Eq. (4.11) and Eq. (4.12). At a high level, our proposed Bayesian PNM performs inference for the solution $\mathrm{s}(r)$ of Eq. (4.11) and then transforms the resultant posterior back into the original $(x,y)$-coordinate system. However, because $G$ in Eq. (4.11) depends on the solution $\tilde{\mathrm{s}}(\tilde{r})$ of Eq. (4.12), we must also estimate $\tilde{\mathrm{s}}(\tilde{r})$ and for this we need to evaluate $H$. Our PNM is therefore based on the information operator

$$A(\mathrm{y}) \quad = \quad [G(r_0), \ldots, G(r_n), H(\tilde{r}_0), \ldots, H(\tilde{r}_n)]^\top \in \mathcal{A} = \mathbb{R}^{2(n+1)}$$

which corresponds indirectly to $2(n+1)$ evaluations of $f$, the original gradient field. The extension of our approach to a general $m$th order ODE proceeds analogously, with $\mathcal{A} = \mathbb{R}^{m(n+1)}$. The use of Thm. 4.8 is illustrated in Example 14 in the Appendix.

The two principles of prior construction that we advocated in the case of a first order ODE apply equally to the case of a second- and higher-order ODE. It therefore remains only to discuss the selection of the specific inputs $r_i$ (and $\tilde{r}_i$ in the case of a second order ODE) that are used to define the information operator $A$. This discussion is again reserved for Section A.1 of the Appendix.

## 4.3 Numerical Illustration

In this section the proposed Bayesian PNM is empirically illustrated. Recall that we are not advocating these methods for practical use, rather they are to serve as a proof-of-concept for demonstrating that exact Bayesian inference can *in principle* be performed for ODEs, albeit at considerable effort; a non-trivial finding that helps to shape ongoing research and discussion in this nascent field.

The case of a first order ODE is considered in Section 4.3.1 and the second order case is contained in Section 4.3.2. In both cases, scope is limited to verifying the correctness of the procedures, as well as indicating how conjugate prior distributions can be constructed.

Code to reproduce the numerical results in this section can be found at

https://github.com/jwang727/Thesiscode. Examples were implemented in Matlab.

### 4.3.1 A First Order ODE

This section illustrates the empirical performance of the proposed method for a first order ODE.

**ODE:** To limit scope we consider first order ODEs of the form

$$\frac{\mathrm{dy}}{\mathrm{d}x} = F\left(\frac{\mathrm{y}(x)}{x}\right), \qquad x \in [1, x_T], \qquad \mathrm{y}(1) = y_0. \tag{4.19}$$

Note that admitted transformation and associated canonical coordinates for this class of ODE have already been derived in Ex. 9, Ex. 11 and Ex. 12.

**Prior:** In constructing a prior $\mu \in \mathcal{P}_\mathcal{Y}$ we refer to the implicit prior principle in Sec. 4.2.2. Indeed, recall from Ex. 11 that the ODE in Eq. (4.19) can be transformed into an ODE of the form

$$\frac{\mathrm{ds}}{\mathrm{d}r} = G(r), \qquad r \in (0, \infty), \qquad \mathrm{s}(y_0) = \log(y_0).$$

Then our approach constructs a distribution $\nu \in \mathcal{P}_\mathcal{S}$ where, from Ex. 12, $\mathcal{S}$ is the set of differentiable functions s defined on $r \in (0, \infty)$ and satisfying

$$\log(r) \;\leq\; \mathrm{s}(r) \;\leq\; \log(r) + \log(x_T). \tag{4.20}$$

To ensure monotonicity in the implicit prior principle, we take $\frac{\mathrm{d}x}{\mathrm{d}r} > 0$, which translates into the requirement that

$$\frac{\mathrm{ds}}{\mathrm{d}r} \;>\; \frac{1}{r}. \tag{4.21}$$

If Eq. (4.21) holds, then $\nu$ induces a well-defined distribution $\mu \in \mathcal{P}_\mathcal{Y}$. Note that the constraints in Eq. (4.20) and Eq. (4.21) preclude the direct use of standard prior models, such as Gaussian processes. However, it is nevertheless possible to design priors that are convenient for a given set of canonical coordinates. Indeed, for the canonical coordinates $r, s$ in our example, we can consider a prior of the form

$$\mathrm{s}(r) \;=\; \log(r) + \log(x_T)\zeta(r)$$

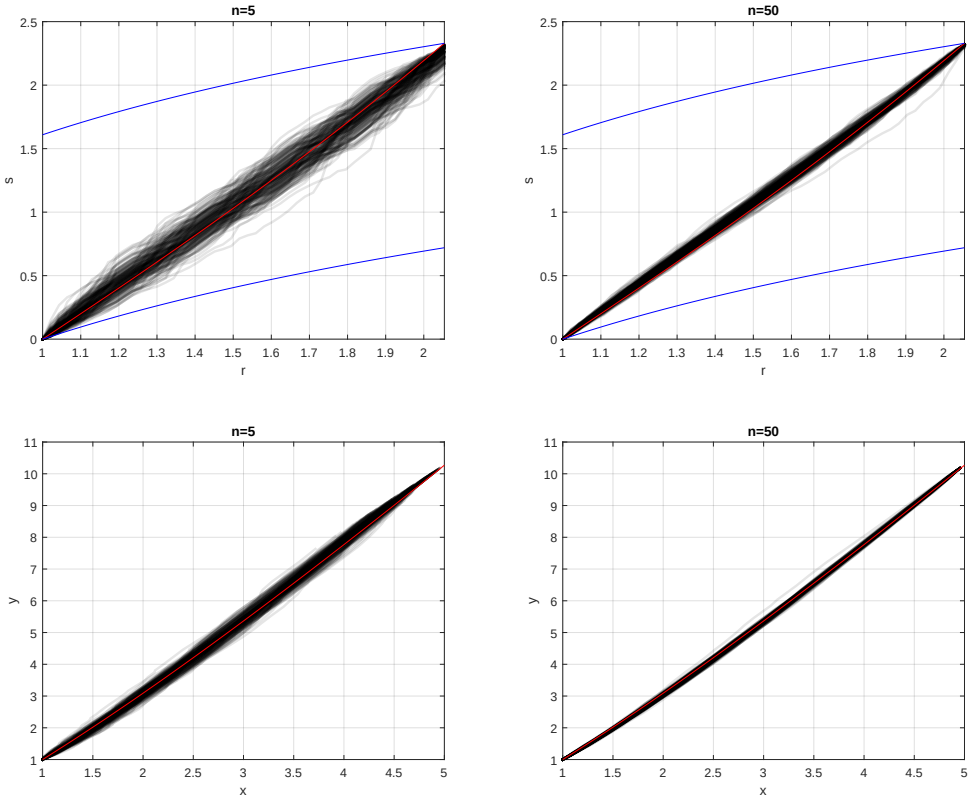



Figure 4.3: Experimental results, first order ODE: The black curves represent samples from the posterior, whilst the exact solution is indicated in red. The blue curves represent a constraint on the $(r, s)$ domain that arises when the implicit prior principle is applied. The number $n$ of gradient evaluations is indicated. Top: results in the $(r, s)$ domain. Bottom: results in the $(x, y)$ domain.

where the function $\zeta : (0, \infty) \to \mathbb{R}$ satisfies

$$\zeta(y_0) = 0, \qquad \zeta(r) \leq 1, \qquad \frac{\mathrm{d}\zeta}{\mathrm{d}r} > 0. \tag{4.22}$$

For this experiment, the approach of López-Lopera *et al.* (2018) was used as a prior model for the monotone, bounded function $\zeta$; this requires that a number, $N$, of basis functions is specified - for brevity we defer the detail to 4.3.3.

The prior just described incorporates the symmetric structure of the ODE, in the sense that the independent variable $r = \frac{y}{x}$ is the first canonical coordinate of the infinitesimal generator of the Lie group of transformations of the original ODE in Eq. 11. In other words, $r$ is a variable fixed by the Lie group of transformations of the ODE (in this case $x^* = \alpha x$, $y^* = \alpha y$, so $r^* = r$). Because the prior is defined on functions $s(r)$ of $r$, this means the prior itself is also unchanged by the Lie group of transformations of the ODE, so that the prior effectively incorporates the symmetric structure of the ODE.

**Results:** To obtain empirical results we consider the ODE with $F(r) = r^{-1} + r$ and $y_0 = 1, x_T = 5$. The posterior distributions that were obtained as the number $n$ of data points was increased were sampled and plotted in the $(r, s)$ and $(x, y)$ planes in Fig. 4.3. In each case a basis of size $N = 2n$ was used. Observe that the implicit prior principle ensures that all curves in the $(x, y)$ plane are well-defined functions (i.e. there is at most one $y$ value for each $x$ value). Observe also that the posterior mass appears to contract to the true solution $\mathrm{y}^\dagger$ of the ODE as the number of evaluations $n$ of the gradient field is increased.

### 4.3.2 A Second Order ODE

This section illustrates the empirical performance of the proposed method for a second order ODE.

**ODE:** Consider again the second order nonlinear ODE in Eqn. A.1 together with the initial condition $y(x_0) = y_0$, $\frac{\mathrm{dy}}{\mathrm{d}x}(x_0) = y_0'$.

**Prior:** It is shown in Example 14 in the Appendix that Eq. A.1 can be reduced to a first order ODE in $(s, r)$ with $-\frac{1}{x_0} - r \leq s \leq -\frac{1}{x_T} - r$. The implicit prior principle in this case requires that $\frac{\mathrm{ds}}{\mathrm{d}r} > -1$. Thus we are led to consider a parametrisation of the form

$$\mathrm{s}(r) \quad = \quad -\frac{1}{x_0} - r + \left(\frac{1}{x_0} - \frac{1}{x_T}\right)\zeta(r)$$

where the function $\zeta$ again satisfies the conditions in Eq. (4.22). The approach of López-Lopera *et al.* (2018) was therefore again used as a prior model.

For this example an additional level of analytic tractability is possible, as described in detail in Example 14 in the Appendix. Thus we need only consider an information operator of the form $A(\mathrm{y}) = [G(r_0), \ldots, G(r_n)]$.

**Results:** The posterior distributions that were obtained are plotted in the $(r, s)$ plane and the $(x, y)$ plane in Fig. 4.4. A basis of size $N = 2n$ was used, with $[y_0, y_0'] = [-10, 1], [x_0, x_T] = [5, 10]$. Observe that the implicit prior principle ensures that all curves in the $(x, y)$ plane are well-defined functions (i.e. there is at most one $y$ value for each $x$ value). The true solution appears to be smoother than the samples, even for 50 gradient evaluations, which suggests that the prior was somewhat conservative in this context.

### 4.3.3 Computational Detail

Recall that for both the first order ODE example in Sec. 4.3.1 and the second order ODE example in Sec. 4.3.2 we required a non-parametric prior over functions $\zeta$ which satisfy

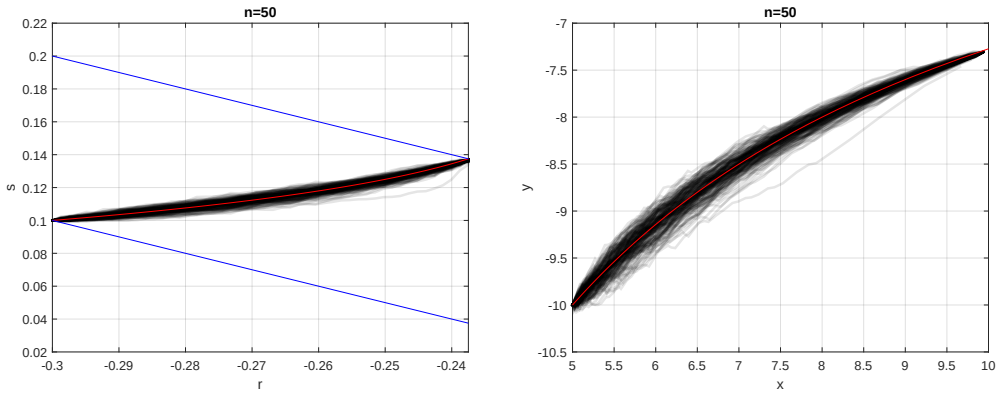



Figure 4.4: Experimental results, second order ODE: The black curves represent samples from the posterior in the $(r, s)$ plane (left) and $(x, y)$ plane (right), whilst the exact solution is indicated in red. The blue curves represent a constraint on the domain that arises when the implicit prior principle is applied. The number of gradient evaluations was $n = 50$.

the constraints given in Eq. 4.22, namely that

$$\zeta(r_0) = 0, \qquad \zeta(r) \leq 1, \qquad \frac{\mathrm{d}\zeta}{\mathrm{d}r} \geq 0. \tag{4.23}$$

Moreover, bearing in mind the posterior computation that is to follow, we require in addition that the prior conveniently facilitates the conditioning calculations involved. It is clear that standard non-parametric priors such as Gaussian processes do not satisfy the boundedness or monotonicity constraints, whilst a nonlinear transformation of such a process would fail to make conditioning on data straight-forward. In fact, the construction of such flexible priors remains an active area of research.

To proceed, we adopted an approach recently proposed in López-Lopera *et al.* (2018). In brief, the main idea is to construct an $N$-dimensional parametric distribution over functions for which Eq. 4.23 is satisfied. This distribution, being finite-dimensional, allows for the possibility of tractable conditioning operations, whilst the flexibility to take $N$ arbitrarily large provides a means of ensuring that the salient uncertainty is accurately represented. More specifically, the function $\zeta$ is parametrised as

$$\zeta(r) = \sum_{j=1}^{N} z_j \phi_j(r) \tag{4.24}$$

where the $\phi_j$ are basis functions

$$\phi_j(r) = \begin{cases} 1 - |\frac{r-t_j}{h}| & |\frac{r-t_j}{h}| \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

for equally spaced points $t_j$ with increment $h$, as recommended in López-Lopera *et al.* (2018). A prior on $\zeta$ can be induced via a prior on the coefficients $z_1, \ldots, z_N$, with $N$ taken to be substantially larger than the number $n$ of datapoints on which the $z_i$ are to be conditioned. The specific construction of a prior on the coefficients is required to encode the constraints in Eq. 4.23 and to admit tractable conditioning; these issues are discussed in the remainder.

First, we consider the boundedness and monotonicity constraints in Eq. 4.23. At the level of the coefficients, it is straight-forward to check that this requires that the prior support is restricted to the set

$$\mathcal{Z} = \{z \in \mathbb{R}^N : 0 < z_1 \leq z_2 \leq \cdots \leq z_N \leq 1\}.$$

For convenience, we elected to use a prior that was obtained by restricting a standard Gaussian measure $\mathcal{N}(0, I)$ on $\mathbb{R}^N$ to the set $\mathcal{Z}$.

Second, we consider how to condition on a dataset. Recall that information is provided on the values of the gradient $\zeta'(r_i) = b_i$ of the function $\zeta$, evaluated at a finite number of locations $r_i$ of the canonical coordinate $r$, together with the initial condition $\zeta(r_0) = b_0$ . Thus the information can be described by the linear system of constraints

$$\Phi z = b$$

where

$$\Phi = \begin{bmatrix} \phi_1(r_0) & \ldots & \phi_N(r_0) \\ \phi_1'(r_1) & \ldots & \phi_N'(r_1) \\ \vdots & \vdots & \\ \phi_1'(r_n) & \ldots & \phi_N'(r_n) \end{bmatrix}, \qquad b = \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{bmatrix}.$$

The posterior can therefore be characterised as the restriction of $\mathcal{N}(0, I)$ to the set $\mathcal{Z} \cap \mathcal{D}$ where $\mathcal{D} = \{z \in \mathbb{R}^N : \Phi z = b\}$.

Finally, we discuss how posterior computation was performed. The key observation is that an equivalent characterisation of the posterior is first to restrict $\mathcal{N}(0, I)$ to $\mathcal{D}$ and then to further restrict to $\mathcal{Z}$. This is advantageous since the linear nature of the data implies that the restriction $z|\mathcal{D}$ of $\mathcal{N}(0, I)$ to $\mathcal{D}$ is again a Gaussian with a closed form, denoted $\mathcal{N}(\mu, \Sigma)$. It is important to note that $\Sigma$ is singular (rank $\rho = N - n - 1$) and so $\Sigma = U\Lambda^2 U^\top$, where $U$ is an orthogonal matrix and $\Lambda$ is a diagonal matrix with $\rho$ non-zero entries on the diagonal. Thus we can express $z|\mathcal{D}$ in the form

$$z = \mu + U\Lambda\tilde{z}$$

where $\tilde{z} \sim \mathcal{N}(0, I)$ is a standard Gaussian on $\mathbb{R}^\rho$. Let $M = U\Lambda$ and let $m_i = [m_{i,1}, \ldots m_{i,\rho}]$

denote the $i$th row of $M$. Then we have the relation

$$z \in \mathcal{Z} \iff \tilde{z} \in \tilde{\mathcal{Z}}$$

where

$$\tilde{\mathcal{Z}} = \{\tilde{z} \in \mathbb{R}^\rho : 0 \leq m_1 \tilde{z} + \mu_1, 0 \leq (m_{i+1} - m_i)\tilde{z} + (\mu_{i+1} - \mu_i), 0 \leq -m_N \tilde{z} + (1 - \mu_N)\}$$

or equivalently

$$\tilde{\mathcal{Z}} = \{\tilde{z} \in \mathbb{R}^\rho : F\tilde{z} + g \geq 0\}$$

for

$$F = \begin{bmatrix} m_1 \\ m_2 - m_1 \\ \vdots \\ m_N - m_{N-1} \\ -m_N \end{bmatrix}, \quad g = \begin{bmatrix} \mu_1 \\ \mu_2 - \mu_1 \\ \vdots \\ \mu_N - \mu_{N-1} \\ 1 - \mu_N \end{bmatrix}.$$

The computational task is thus reduced to sampling the restriction of the $\rho$-dimensional standard Gaussian random variable $\tilde{z}$ to the (non-null) set $\tilde{\mathcal{Z}}$. The development of computational methods to sample from such (potentially high-dimensional) distributions is itself an active area of research, and for this work we employed the Hamiltonian Monte Carlo method of Pakman & Paninski (2014), as recommended specifically for this purpose in López-Lopera *et al.* (2018).

## 4.4 Discussion

This chapter presented a foundational perspective on PNM. It was first argued that there did not exist a Bayesian PNM for the numerical solution of ODEs. Then, to address this gap, a prototypical Bayesian PNM was developed. The Bayesian perspective that we have put forward sheds light on foundational issues which will need to be addressed going forward:

**Foundation of PNM:** As explained in chapter 1 existing PNM for ODEs each take the underlying state space $\mathcal{Y}$ to be the solution space of the ODE. This appears to be problematic, in the sense that a generic evaluation $f(x_i, y_i)$ of the gradient field cannot be cast as information $A(\mathrm{y}^\dagger)$ about the solution $\mathrm{y}^\dagger$ of the ODE unless the point $(x_i, y_i)$ lies exactly on the solution curve $\{(x, \mathrm{y}^\dagger(x)) : x \in [x_0, x_T]\}$. As a consequence, all existing PNM of which we are aware violate the likelihood principle and are therefore not strictly Bayesian, as discussed earlier in chapter 3. The assumption of a solvable Lie algebra, used

in this work, can be seen as a mechanism to ensure the existence of an exact information operator $A$, so that the likelihood principle can be obeyed. However, for a general ODE it might be more natural to take the underlying state space to be a set $\mathcal{F}$ of permitted gradient fields and the quantity of interest $Q(f)$ to map a gradient field $f$ to the solution of the associated ODE. This would make the information operator $A$ trivial but evaluation of the push-forward $Q_{\#}\mu^a$ would require the exact solution operator of the ODE. However, the reliance on access to an oracle solver $Q$ makes this philosophically somewhat distinct from PNM.

**Limitations of Bayesian PNM:** The proposed method was intended as a proof-of-concept, not a practical numerical method. It is therefore useful to highlight the aspects in which it is limited. First, when an $m$th order ODE admits an $r$-parameter Lie group of transformations with $r > m$, there is an arbitrariness to the particular $m$-dimensional subgroup of transformations that are selected. Second, the route to obtain transformations admitted by the ODE demands that some aspects of the gradient field $f$ are known, in contrast to other work in which $f$ is treated as a black-box. For instance, in Ex. 11 we used the fact that $f$ can be expressed as $f(x, y) = F(\frac{y}{x})$, although knowledge of the form of $F$ was not required. Third, the class of ODEs for which a solvable Lie algebra is admitted is relatively small. On the other hand, references such as Bluman & Anco (2002) document important cases where our method could be applied. Fourth, the principles for prior construction that we identified do not entail a unique prior and, as such, the question of prior elicitation must still be addressed.

**Outlook:** The goal of providing rigorous and exact statistical uncertainty quantification for the solution of an ODE is, we believe, important and will continue to be addressed. Traditional numerical methods have benefitted from a century of research effort and, in comparison, Bayesian PNM is an under-developed field. For example, the limited existing work on PNM for ODEs, such as Skilling (1992); Schober *et al.* (2014); Chkrebtii *et al.* (2016); Kersting & Hennig (2016); Schober *et al.* (2019); Kersting *et al.* (2018); Tronarp *et al.* (2019), does not attempt to provide adaptive error control (though we note promising ongoing research in that direction by Chkrebtii & Campbell, 2019). Nevertheless, the case for developing Bayesian numerical methods - which shares some parallels with the case for Bayesian statistics as opposed to other inferential paradigms - is clear, as argued in Diaconis (1988) and Hennig *et al.* (2015). The insights we have provided in this chapter serve to highlight the foundational issues pertinent to Bayesian PNM for ODEs. Indeed, our proof-of-concept highlights that performing exact Bayesian inference for ODEs may be extremely difficult. This in turn provides motivation for the continued development of 'approximately Bayesian' approaches to PNM, which in chapter 3 we surveyed in detail.

# Chapter 5

# Approximate Bayesian Inference for Partial Differential Equations

## 5.1 Introduction

In the previous chapter, an exact Bayesian PNM for the numerical solution of ODEs was presented, for the case when the ODE can be reduced to quadrature by exploiting its underlying symmetry through Lie group methods. However, the class of ODEs for which a solvable Lie algebra is admitted is limited. This motivates the development of approximate Bayesian PNMs, which aim to approximate the differential equation in such a way that exact Bayesian inference can be performed, motivated in particular by challenging numerical tasks for which numerical uncertainty cannot be neglected.

As discussed previously in chapter 3, the cases of (mostly approximate) PNMs for nonlinear ODEs and linear PDEs have each been studied. However, to date, no (approximate or exact) Bayesian PNM for the numerical solution of nonlinear PDEs has been proposed. This chapter therefore focuses on nonlinear partial differential equations, though the proposed method is also applicable to linear PDEs as well as ODEs. Computationally, we focus on PDEs whose governing equations must be evaluated pointwise at high computational cost.

This chapter presents the first (approximate) Bayesian PNM for numerical uncertainty quantification in the setting of nonlinear PDEs. Our strategy is based on local linearisation of the nonlinear differential operator, in order to perform conjugate Gaussian updating in an approximate Bayesian framework. Broadly speaking, our approach is a natural generalisation of the approach taken by Chkrebtii *et al.* (2016) for ODEs, but with local linearisation to address the additional challenges posed by nonlinear PDEs. The aim is to quantify numerical uncertainty with respect to the unknown solution of the PDE. An important point is that we consider only PDEs for which evaluation of either the right-hand

side or the initial or boundary conditions is associated with a high computational cost; we do not aim to numerically solve PDEs for which a standard numerical method can readily be employed to drive numerical error to a negligible level, nor do we aim to compete with standard numerical methods in terms of CPU requirement. Such problems occur in diverse application areas, such as modelling of ice sheets, carbon and nitrogen cycles (Hurrell *et al.*, 2013), species abundance and ecosystems (Fulton, 2010), each in response to external forcing from a meteorological model, or in solving PDEs that themselves depend on the solution of an auxiliary PDE, which occur both when *operator splitting* methods are used (MacNamara & Strang, 2016) and when *sensitivity equations*, expressing the rate of change of the solution of a PDE with respect to its parameters, are to be solved (Petzold *et al.*, 2006; Cockayne & Duncan, 2020). These applications provide strong motivation for PNM, since typically it will not be possible to obtain an accurate approximation to the solution of the PDE and the rich, probabilistic description of numerical uncertainty provided by a PNM can be directly useful (e.g. Oates *et al.*, 2019*a*).

The remainder of the chapter is structured as follows: In Section 5.2 the proposed method is presented. The choice of prior is driven by the mathematical considerations described in Section 5.3. A detailed experimental assessment is performed in Section 5.4. Concluding remarks are contained in Section 5.5.

## 5.2 Methods

In Section 5.2.1 we present the general form of the nonlinear PDE that we aim to solve using PNM. The use of finite differences for local linearisation is described in Section 5.2.2. Then, in Section 5.2.3 we present our proposed approximate Bayesian PNM, discussing how computations are performed and how the associated uncertainty is calibrated.

### 5.2.1 Set-Up and Notation

For a set $S \subseteq \mathbb{R}^d$, let $C^0(S)$ denote the vector space of continuous functions $c\colon S \to \mathbb{R}$. For two *multi-indices* $\alpha, \beta \in \mathbb{N}_0^d$, we write $\alpha \leq \beta$ if $\alpha_i \leq \beta_i$ for each $i = 1, \ldots, d$. For a multi-index $\beta \in \mathbb{N}_0^d$, we let $|\beta| = \beta_1 + \cdots + \beta_d$ and let $C^\beta(S) \subseteq C^0(S)$ denote those functions $c$ whose partial derivatives

$$\partial^\alpha c \coloneqq \partial_{z_1}^{\alpha_1} \ldots \partial_{z_d}^{\alpha_d} c(z) \coloneqq \frac{\partial^{|\alpha|} c(z)}{\partial z_1^{\alpha_1} \ldots \partial z_d^{\alpha_d}}, \qquad \alpha \leq \beta$$

exist and are continuous in $S$.

Let $T \in (0, \infty)$ and let $\Gamma$ be an open and bounded set in $\mathbb{R}^d$, whose boundary is denoted

$\partial\Gamma$. Let $\beta \in \mathbb{N}_0^d$ be a multi-index and consider a differential operator

$$D\colon C^{\beta}([0,T]\times\Gamma) \to C^{0}([0,T]\times\Gamma)$$

and the associated initial value problem with Dirichlet boundary conditions

$$\begin{aligned} Du(t,x) &= f(t,x), && t\in[0,T],\ x\in\Gamma \\ u(0,x) &= g(x), && x\in\Gamma \\ u(t,x) &= h(t,x), && t\in[0,T],\ x\in\partial\Gamma \end{aligned} \tag{5.1}$$

whose unique classical (i.e. strong) solution $u \in C^{\beta}([0,T]\times\Gamma)$ is assumed to exist[1]. The task considered in this chapter is to produce a probability distribution over functions that (approximately) carries the semantics of Bayesian inference for $u$; i.e. we seek to develop an (approximate) Bayesian PNM for the numerical solution of (5.1) (Cockayne *et al.*, 2019). In particular, we are motivated by the problems described in Section 5.1, for which evaluation of $f$, $g$ and $h$ are associated with a high computational cost. Such problems provide motivation for a careful quantification of uncertainty regarding the unknown solution $u$, since typically it will not be possible to obtain a sufficient number of evaluations of $f$, $g$ and $h$ in order for $u$ to be precisely identified.

**Why Not Emulation?**

Given that the dominant computational cost is assumed to be evaluation of $f$, $g$ and $h$, it is natural to ask whether the uncertainty regarding these functions can be quantified using a probabilistic model, such as an *emulator* (Kennedy & O'Hagan, 2001). This would in principle provide a straight-forward Monte Carlo solution to the problem of quantifying uncertainty in the solution $u$ of (5.1), where first one simulates an instance of $f$, $g$ and $h$ from the emulator and then applies a classical numerical method to solve (5.1) to high numerical precision. The problem with this approach is that construction of a defensible emulator is difficult; the functions $f$, $g$ and $h$ are coupled together by the nonlinear PDE in (5.1) and, for example, it cannot simultaneously hold that each of $f$, $g$ and $h$ are Gaussian processes. In fact, the challenge of ensuring that samples of $f$, $g$ and $h$ are consistent with the existence of a solution to (5.1) poses a challenge that is comparable with solving the PDE itself. This precludes a straight-forward emulation approach to (5.1) and motivates our focus on PNM in the remainder, where uncertainty is quantified in the solution space of (5.1).

[1]The existence of a strong solution is a nontrivial assumption, since several PDEs admit only a weak solution; see Section 1.3.2 of Evans (1998) for definitions and background. A well-known class of classical numerical methods that also presuppose the existence of a strong solution are the radial basis function methods (Fornberg & Flyer, 2015). In Section 5.4.2 we consider, empirically, the performance of the method developed in this chapter when applied to a PDE for which a strong solution does not exist.

### 5.2.2 Finite Difference Approximation of Differential Operators

If $D$ is linear then the differential equation in (5.1) is said to be *linear* and one or more of the Bayesian PNM of Chkrebtii *et al.* (2016); Cockayne *et al.* (2016); Chkrebtii & Campbell (2019) may be applied (assuming any associated method-specific requirements are satisfied). If $D$ is *nonlinear* then at most we can express $D = P + Q$, where $P$ is linear and $Q$ is nonlinear (naturally such representations are non-unique in general). For example, for Burgers' equation

$$Du = \frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} - \varepsilon\frac{\partial^2 u}{\partial x^2} = 0$$

we have both $P = \partial_t - \varepsilon\partial_x^2$, $Q = u\partial_x$ and also the trivial $Q = D$, $P = 0$. In this chapter we aim, given a decomposition of $D$ in terms of $P$ and $Q$, to adaptively approximate $Q$ by a linear operator, in order that exact Gaussian conditioning formulae can be exploited. Although we do not prescribe how to select $P$ and $Q$, one should bear in mind that we aim to construct a linear approximation of $Q$, meaning that a decomposition should be identified that renders $Q$ as close to linear as possible, to improve the quality of the approximation. The effect of different selections for $P$ and $Q$ is investigated in Section 5.4.2.

To adaptively construct linear approximations to the nonlinear differential operator $Q$, we propose to exploit traditional finite difference formulae (Strikwerda, 2004). Note that our conceptualisation of these approximations as linear *operators* for Gaussian conditioning is somewhat non-traditional. Define a time discretisation grid $\mathbf{t} = [t_0, t_1 \dots t_{n-1}]$, where $0 = t_0 < t_1 < \cdots < t_{n-1} \leq T$ with the increment $\delta \coloneqq t_i - t_{i-1}$ fixed. For concreteness, consider Burgers' equation with $P = \partial_t - \varepsilon\partial_x^2$, $Q = u\partial_x$. The following discussion is intended only to be informal. Suppose that the unknown solution $u(t_{i-1}, \cdot)$ at time $t_{i-1}$ has been approximated to accuracy $\mathcal{O}(\delta)$ by $u_{i-1}(\cdot)$, as quantified by a norm $\|\cdot\|$ on $C^\beta([0,T] \times \Gamma)$. Then we could adaptively build a linear approximation to $Q$ at time $t_i$ as

$$Q_i u(t_i, x) \coloneqq u_{i-1}(x)\frac{\partial u}{\partial x}(t_i, x). \tag{5.2}$$

This provides an approximation $D_i = P + Q_i$ to the original differential operator $D$, at time $t_i$, with accuracy $\mathcal{O}(\delta)$. To achieve higher order accuracy, we can use higher order approximations of $Q$. For example, letting $\frac{\partial u}{\partial t}\big|_{i-1}(x)$ denote an approximation to $\frac{\partial u}{\partial t}(t_{i-1}, x)$, we could take

$$Q_i u(t_i, x) \coloneqq \left[u_{i-1}(x) + \delta \left.\frac{\partial u}{\partial t}\right|_{i-1}(x)\right]\frac{\partial u}{\partial x}(t_i, x).$$

The only requirement that we impose on finite difference approximations is that $Q_i$ uses (only) data that were gathered at earlier time points $t_{i-1}, t_{i-2}, \dots$, analogous to *backward*

*difference* formulae. This is to ensure that the approximations $Q_i$ are well-defined before they are used in our method, which is described next.

Henceforth we assume that an appropriate representation $D = P + Q$ has been identified and an appropriate linear approximation to $Q$ has been selected. The next section describes how probabilistic inference for $u$ can then proceed.

### 5.2.3 Proposed Approach

In this section we describe our proposed method. Recall that we assume there exists a unique $u \in C^\beta([0,T] \times \Gamma)$ for which (5.1) is satisfied. Since (5.1) represents an infinite number of constraints, it is not generally possible to recover $u$ exactly with a finite computational budget. Our proposed method mirrors a general approach used to construct Bayesian PNM (Cockayne *et al.*, 2019), in that we consider conditioning on only a finite number of the constraints in (5.1) and reporting the remaining uncertainty as our posterior. The case of nonlinear PDEs presents an additional challenge in that a subset of the constraints are nonlinear, and are therefore not amenable to exact Gaussian conditioning. To circumvent this issue, we condition on linear approximations to the constraints following the ideas developed in Section 5.2.2.

#### Prior Distribution

The starting point of any Bayesian analysis is the elicitation of a suitable prior distribution. In our case, it would be desirable to elicit a prior that is supported on $C^\beta([0,T]\times\Gamma)$, since we *a priori* know that the solution $u$ to (5.1) has this level of regularity. Our approach is rooted in Gaussian conditioning and thus the regularity of Gaussian process sample paths must be analysed. This analysis is somewhat technical and we therefore defer the discussion of prior elicitation to Section 5.3.

For the remainder of this section we assume that a suitable Gaussian process prior $U \sim \mathcal{GP}(\mu, \Sigma)$ has been elicited. Here $\mu\colon [0,T] \times \Gamma \to \mathbb{R}$, $\mu(t,x) := \mathbb{E}[U(t,x)]$ is the *mean function* and $\Sigma\colon ([0,T] \times \Gamma) \times ([0,T] \times \Gamma) \to \mathbb{R}$, $\Sigma((t,x),(t',x')) := \mathbb{E}[(U(t,x) - \mu(t,x))(U(t',x') - \mu(t',x'))]$ is the *covariance function*; see Rasmussen & Williams (2006) for background. The random variable notation $U$ serves to distinguish the true solution $u$ of (5.1) from our probabilistic model for it. The specific choices of $\mu$ and $\Sigma$ discussed in Section 5.3 have sufficient regularity for the subsequent derivations in this section to be well-defined.

#### Initialisation

At the outset we fix a time discretisation $\mathbf{t} = [t_0, t_1 \dots t_{n-1}]$, where $0 = t_0 < t_1 < \cdots < t_{n-1} \leq T$, and a spatial discretisation $\mathbf{x} = [x_1, x_2, \dots, x_m] \in (\Gamma \cup \partial\Gamma)^m$ where the $x_i$ are

required to be distinct. It will sometimes be convenient to interpret $\mathbf{x}$ as a set $\{x_1, \ldots, x_n\}$; for instance we will write $\mathbf{x} \setminus \partial\Gamma$ to denote $\{x_1, \ldots, x_n\} \setminus \partial\Gamma$.

Our first task is to condition on (or *assimilate*) a finite number of constraints that encode the initial condition $u(0, x) = g(x)$, $x \in \Gamma$. For this purpose we use the spatial discretisation $\mathbf{x}$, and condition on the data $U(0, x) = g(x)$ at each $x \in \mathbf{x} \setminus \partial\Gamma$. (For example, if $\Gamma = [0, 1]$ and $0 = x_1 < x_2 < \cdots < x_{m-1} < x_m = 1$, then we condition on $U(0, x_i) = g(x_i)$ for $i = 2, \ldots, m-1$. The two boundary locations $x_1, x_m \in \partial\Gamma$ are excluded since these constraints are assimilated as part of the boundary condition, which will shortly be discussed.) To perform conditioning, we use the following vectorised shorthand:

$$
\begin{aligned}
\boldsymbol{v}_i &\coloneqq (t_i, \mathbf{x}) \coloneqq [(t_i, x_1), (t_i, x_2), \ldots, (t_i, x_m)]^\top \in ([0, T] \times \Gamma)^m \\
U(\boldsymbol{v}_i) &\coloneqq [U(t_i, x_1), \ldots, U(t_i, x_m)]^\top \in \mathbb{R}^{m \times 1} \\
g(\boldsymbol{v}_i) &\coloneqq [g(x_1), \ldots, g(x_m)]^\top \in \mathbb{R}^{m \times 1} \\
\Sigma((t, x), \boldsymbol{v}_i) &\coloneqq [\Sigma((t, x), (t_i, x_1)), \ldots, \Sigma((t, x), (t_i, x_m))] \in \mathbb{R}^{1 \times m} \\
\Sigma(\boldsymbol{v}_i, (t, x)) &\coloneqq \Sigma((t, x), \boldsymbol{v}_i)^\top \in \mathbb{R}^{m \times 1} \\
\Sigma(\boldsymbol{v}_i, \boldsymbol{v}_j) &\coloneqq \begin{bmatrix} \Sigma((t_i, x_1), (t_j, x_1)) & \ldots & \Sigma((t_i, x_1), (t_j, x_m)) \\ \vdots & & \vdots \\ \Sigma((t_i, x_m), (t_j, x_1)) & \ldots & \Sigma((t_i, x_m), (t_j, x_m)) \end{bmatrix} \in \mathbb{R}^{m \times m}
\end{aligned}
$$

Then let $\boldsymbol{a}_0 \coloneqq (t_0, \mathbf{x} \setminus \partial\Gamma)^\top$ denote the locations in $[0, T] \times \Gamma$ where the initial condition is to be assimilated. At $\boldsymbol{a}_0$ we have the *initial data* $\mathbf{y}^0 \coloneqq g(\boldsymbol{a}_0)$. These initial data are assimilated into the Gaussian process model according to the standard conditioning formulae (eq. 2.19; Rasmussen & Williams, 2006)

$$
\begin{aligned}
U^0 &\coloneqq (U | U(\boldsymbol{a}_0) = \mathbf{y}^0) \sim \mathcal{GP}(\mu^0, \Sigma^0) \\
\mu^0(r) &\coloneqq \mu(r) + \Sigma(r, \boldsymbol{a}_0)\Sigma(\boldsymbol{a}_0, \boldsymbol{a}_0)^{-1}(g(\boldsymbol{a}_0) - \mu(\boldsymbol{a}_0)) \\
\Sigma^0(r, s) &\coloneqq \Sigma(r, s) - \Sigma(r, \boldsymbol{a}_0)\Sigma(\boldsymbol{a}_0, \boldsymbol{a}_0)^{-1}\Sigma(\boldsymbol{a}_0, s)
\end{aligned}
$$

where $r, s \in [0, T] \times \Gamma$.

### Time Stepping

Having assimilated the initial data, we now turn to the remaining constraints in (5.1). Following traditional time-stepping algorithms, we propose to proceed iteratively, beginning at time $t_0$ and then advancing to $t_1$, $t_2$, and ultimately to $t_{n-1}$. At each iteration $i$ we aim to condition on a finite number of constraints that encode the boundary condition $u(t_i, x) = h(t_i, x)$, $x \in \partial\Gamma$, and the differential equation itself $Du(t_i, x) = f(t_i, x)$, $x \in \Gamma$.

For this purpose we again use the spatial discretisation $\mathbf{x}$, and condition on the *boundary data* $U(t_i, x) = h(t_i, x)$ at each $x \in \mathbf{x} \cap \partial\Gamma$ and the *differential data* $DU(t_i, x) = f(t_i, x)$ at each $x \in \mathbf{x}$. Since $D$ is nonlinear, there are no explicit formulae that can be used in general to assimilate the differential data, so instead we propose to condition on the approximate constraints $D_i U(t_i, x) = f(t_i, x)$, $x \in \mathbf{x}$ where $D_i = P + Q_i$ is an adaptively defined linear approximation to $D$, which will be problem-specific and chosen in line with the principles outlined in Section 5.2.2.

For a univariate function such as $\mu$ and a linear operator $L$, we denote $\mu_L(r) = (L\mu)(r)$. For a bivariate function such as $\Sigma$, we denote $\Sigma_L(r, s) = L_r \Sigma(r, s)$, where $L_r$ denotes the action of $L$ on the $r$ argument. In addition, we denote $\Sigma_{\bar{L}}(r, s) = L_s \Sigma(r, s)$ and we allow subscripts to be concatenated, such as $\Sigma_{L,L'} = (\Sigma_L)_{L'}$ for another linear operator $L'$.

Fix $i \in \{0, 1, \ldots, n-1\}$. Let $\boldsymbol{b}_i = (t_i, \mathbf{x} \cap \partial\Gamma)$ denote the locations in $[0, T] \times \partial\Gamma$ where the boundary conditions at time $t_i$ are to be assimilated. At $\boldsymbol{b}_i$ we have boundary data $h(\boldsymbol{b_i})$. Correspondingly, we have differential data $f(\boldsymbol{v}_i)$ and we concatenate all data at time $i$ into a single vector $\mathbf{y}^i \coloneqq [h(\boldsymbol{b}_i)^\top, f(\boldsymbol{v}_i)^\top]^\top$, so that $\mathbf{y}^i$ represents all the information on which (approximate) conditioning is to be performed. Upon assimilating these data we obtain

$$
\begin{aligned}
U^{i+1} &\coloneqq (U^i | [U(\boldsymbol{b}_i), D_i U(\boldsymbol{v}_i)] = \mathbf{y}^i) \sim \mathcal{GP}(\mu^{i+1}, \Sigma^{i+1}) \\
\mu^{i+1}(r) &\coloneqq \mu^i(r) + [\Sigma^i(r, \boldsymbol{b}_i), \Sigma^i_{\bar{D}_i}(r, \boldsymbol{v}_i)] A_i^{-1} \begin{bmatrix} h(\boldsymbol{b}_i) - \mu^i(\boldsymbol{b}_i) \\ f(\boldsymbol{v}_i) - \mu^i_{D_i}(\boldsymbol{v}_i) \end{bmatrix} \\
\Sigma^{i+1}(r, s) &\coloneqq \Sigma^i(r, r') - [\Sigma^i(r, \boldsymbol{b}_i), \Sigma^i_{\bar{D}_i}(r, \boldsymbol{v}_i)] A_i^{-1} \begin{bmatrix} \Sigma^i(\boldsymbol{b}_i, s) \\ \Sigma^i_{D_i}(\boldsymbol{v}_i, s) \end{bmatrix} \\
A_i &\coloneqq \begin{bmatrix} \Sigma^i(\boldsymbol{b}_i, \boldsymbol{b}_i) & \Sigma^i_{\bar{D}_i}(\boldsymbol{b}_i, \boldsymbol{v}_i) \\ \Sigma^i_{D_i}(\boldsymbol{v}_i, \boldsymbol{b}_i) & \Sigma^i_{D_i \bar{D}_i}(\boldsymbol{v}_i, \boldsymbol{v}_i) \end{bmatrix}
\end{aligned}
$$

The result of performing $n$ time steps of the algorithm just described is a Gaussian process $\mathcal{GP}(\mu^n, \Sigma^n)$, to which we associate the semantics of an (approximate) posterior in a Bayesian PNM for the solution of (5.1).

The Bayesian interpretation of $\mathcal{GP}(\mu^n, \Sigma^n)$ is reasonable since this distribution arises from the conditioning of the prior $\mathcal{GP}(\mu, \Sigma)$ on a finite number of constraints that are (approximately) satisfied by the solution $u$ of (5.1). This is clarified in the following statement:

**Lemma 5.1.** *The stochastic process $U^n$ obtained above is identical to the distribution*

*obtained when* $U \sim \mathcal{GP}(\mu, K)$ *is conditioned on the dataset*

$$\begin{bmatrix} U(\boldsymbol{a}_0) \\ [U(\boldsymbol{b}_0), D_0 U(\boldsymbol{v}_0)]^\top \\ \vdots \\ [U(\boldsymbol{b}_{n-1}), D_{n-1} U(\boldsymbol{v}_{n-1})]^\top \end{bmatrix} = \begin{bmatrix} \mathbf{y}^0 \\ \mathbf{y}^1 \\ \vdots \\ \mathbf{y}^n \end{bmatrix}.$$

*Proof.* This follows immediately from the self-consistency property of Bayesian inference (invariance to the order in which data are conditioned), but for completeness we demonstrate their algebraic equivalence in Appendix B.1. □

**Remark 1** (Computational Complexity)**.** The computational cost of our algorithm is not competitive with that of a standard numerical method.[2] However, we are motivated by problems for which $f$, $g$ and $h$ are associated with a high computational cost, for which the auxiliary computation required to provide probabilistic uncertainty quantification is inconsequential. Thus we merely remark that the iterative algorithm we presented is gated by the inversion of the matrix $A_i$ at the $i^{\text{th}}$ time step, the size of which is $O(m)$, independent of $i$, and therefore the complexity of predicting the final state $u(T, \cdot)$ of the PDE by performing $n$ iterations of the above algorithm is $O(nm^3)$. For comparison, direct Gaussian conditioning on the information in Lemma 5.1 would incur a higher computational cost of $O(n^3m^3)$, but would provide the joint distribution over the solution $u(t, \cdot)$ at all times $t \in [0, T]$. Although we do not pursue it in this chapter, in the latter case the grid structure present in $\mathbf{t}$ and $\mathbf{x}$ could be exploited to mitigate the $O(n^3m^3)$ cost; for example, a compactly supported covariance model $\Sigma$ would reduce the cost by a constant factor (Gneiting, 2002), or if the preconditions of Schäfer *et al.* (2021) are satisfied then their approach would reduce the cost to $O(nm \log(nm) \log^{d+1}(nm/\epsilon))$ at the expense of introducing an error of $O(\epsilon)$. See also the recent work of de Roos *et al.* (2021).

**Remark 2.** The posterior mean $\mu^{i+1}$ can be interpreted as a particular instance of a radial basis method (Fornberg & Flyer, 2015), as a consequence of the representer theorem for kernel interpolants (Schölkopf *et al.*, 2001). For brevity we do not explore this connection further, but we note that a similar connection was explored in detail in Cockayne *et al.* (2016).

[2] Technically, the computational complexity of our algorithm the same as that of a traditional numerical method that performs forward Euler increments in the temporal component and symmetric collocation in the spatial component (Fasshauer, 1999; Cockayne *et al.*, 2016). However such methods are rarely used, with one factor for this being the computational cost.

### Calibration of Uncertainty

The principal advantage of PNM over classical numerical methods is that they provide probabilistic quantification of uncertainty, in our case expressed in the Bayesian framework, which can be integrated along with other sources of uncertainty to facilitate inferences and decision-making in a real-world context. In order for our posterior distribution to faithfully reflect the scale of uncertainty about the solution of (5.1), we must allow the hyper-parameters of the prior model to adapt to the dataset. However, we do not wish to sacrifice the sequential nature of our algorithm and thus we seek an approach to hyper-parameter estimation that operates in real-time as the algorithm is performed.

To achieve this we focus on a covariance model $\Sigma(\cdot,\cdot;\sigma)$ with a scalar hyper-parameter denoted $\sigma > 0$, which is assumed to satisfy $\Sigma(\cdot,\cdot;\sigma) = \sigma^2\Sigma(\cdot,\cdot;1)$. Such a $\sigma$ is sometimes called a *scale* or *amplitude* hyper-parameter of the covariance model. From Lemma 5.1 it follows that $\sigma$ directly controls the spread of the posterior and it is therefore essential that $\sigma$ is estimated from data in order that the uncertainty reported by the posterior can be meaningful. To estimate $\sigma$, we propose to maximise the predictive likelihood of the "differential data" $f(\boldsymbol{v}_i)$, given the information collected up to iteration $i-1$, for $i \in \{0,\ldots,n-1\}$, which can be considered as an *empirical Bayes* approach based on just those factors in the likelihood that correspond to the differential data. The reasons for focussing on the differential data (as opposed to also including the initial and boundary data) are twofold; first, the differential data constitutes the vast majority of the dataset, and second, this simplifies the computational implementation, described next.

At iteration $i$, the predictive likelihood for $U_{D_i}(\boldsymbol{v}_i)$ is $\mathcal{N}(\mu^i_{D_i}(\boldsymbol{v}_i), \Sigma^i_{D_i\bar{D}_i}(\boldsymbol{v}_i,\boldsymbol{v}_i;\sigma))$, and the observed differential data are $f(\boldsymbol{v}_i)$. Thus we select $\sigma$ to maximise the full predictive likelihood of the differential data

$$\prod_{i=0}^{n-1} \mathcal{N}(f(\boldsymbol{v}_i); \mu^i_{D_i}(\boldsymbol{v}_i), \Sigma^i_{D_i\bar{D}_i}(\boldsymbol{v}_i,\boldsymbol{v}_i;\sigma)). \tag{5.3}$$

Crucially, the linear operators $D_i$ that we constructed do not directly depend on $\sigma$, and it is a standard property of Gaussian conditioning that $\Sigma^i_{D_i\bar{D}_i}(\boldsymbol{v}_i,\boldsymbol{v}_i;\sigma) = \sigma^2\Sigma^i_{D_i\bar{D}_i}(\boldsymbol{v}_i,\boldsymbol{v}_i;1)$. These facts permit a simple closed form expression for the maximiser $\hat{\sigma}$ of (5.3), namely

$$\hat{\sigma}^2 = \frac{1}{n}\sum_{i=0}^{n-1} \left\| (\Sigma^i_{D_i\bar{D}_i}(\boldsymbol{v}_i,\boldsymbol{v}_i);1)^{-\frac{1}{2}}(f(\boldsymbol{v}_i) - \mu^i_{D_i}(\boldsymbol{v}_i)) \right\|^2 \tag{5.4}$$

where $M^{-1/2}$ denotes an inverse matrix square root; $(M^{1/2})^2 = M$. Furthermore, it is clear from (5.4) that in practice one can simply run our proposed algorithm with the prior covariance model $\Sigma(\cdot,\cdot;1)$ and then report the posterior covariance $\hat{\sigma}^2\Sigma^{i+1}(\cdot,\cdot;1)$, so that hyper-parameter estimation is performed in real-time without sacrificing the sequential

nature of the algorithm.

Closed form expressions such as (5.4) are not typically available for other hyper-parameters that may be included in the covariance model, and we therefore assume in the sequel that any other hyper-parameters have been expert-elicited. This limits the applicability of our method to situations where some prior expert insight can be provided. However, we note that data-driven estimation of the amplitude parameter $\sigma$ is able to compensate to a degree for mis-specification of other parameters in the covariance model.

**Relation to Earlier Work**

Here we summarise how the method just proposed relates to existing literature on Bayesian PNM and beyond.

The sequential updating procedure that we have proposed is similar to that of Chkrebtii *et al.* (2016) in the special case of a linear PDE. It is not identical in these circumstances though, for two reasons: First, Chkrebtii *et al.* (2016) incorporated the initial condition $u(0, x) = g(x)$, $x \in \Gamma$, into the prior model, whereas we explicitly conditioned on initial data $g(\boldsymbol{a}_0)$ during the initialisation step of the method. This direct encoding of the initial condition in Chkrebtii *et al.* (2016) relies on $g$ being analytically tractable in order that a suitable prior can be derived by hand. Our treatment of $g$ as a black-box function from which initial data are provided is therefore more general. Second, in Chkrebtii *et al.* (2016) the authors advocated the use of an explicit *measurement error* model, whereas our conditioning formula assume that the differential data $\mathbf{y}^i$ are exact measurements of $U$, as clarified in Lemma 5.1. For linear PDEs this assumption is correct, but it is an approximation in the case a nonlinear PDE. Our decision not to employ a measurement error model here is due to the fact that the scale of the measurement error cannot easily be estimated in an online manner as part of a sequential algorithm, without further approximations being introduced.

To limit scope, the adaptive selection of the $t_i$ or $x_j$ was not considered, but we refer the reader to Chkrebtii & Campbell (2019) for an example of how this can be achieved using Bayesian PNM. Note, however, that adaptive selection of a time grid may be problematic when evaluation of either $f$ or $h$ is associated with a high computational cost, since the possibility of taking many small time steps relinquishes control of the computational budget. For this reason, non-adaptive methods may be preferred in this context, since the run-time of the PNM can be provided up-front.

The choice of linearisation $Q_i$ was left as an input to the proposed method, with some guidelines (only) provided in Section 5.2.2. This can be contrasted with recent work for ODEs in Tronarp *et al.* (2019, 2021); Bosch *et al.* (2020), where first order Taylor series were used to automatically linearise a nonlinear gradient field. It would be possible to also consider the use of Taylor series methods for nonlinear PDEs. However, their use

assumes that the gradient field is analytically tractable and can be differentiated, while for the method developed in this chapter, we are motivated by situations in which $f$ is a black-box that can (only) be point-wise evaluated. The use of linearisations in PNM was also explored Chen *et al.* (2021), in the *maximum a posteriori* estimation context.

The combination of local linearisation and Gaussian process conditioning was also studied by Raissi *et al.* (2018), who considered initial value problems specified by PDEs, where the initial condition was random and the goal was to approximate the implied distribution over the solution space of the PDE. The authors observed that if the initial condition was a Gaussian process, then approximate conjugate Gaussian computation is possible when a finite difference approximation to the differential operator was employed. This provided a one-pass, cost-efficient alternative to the Monte Carlo approach of repeatedly sampling an initial condition and then applying a classical numerical method. Our work bears a superficial similarity to Raissi *et al.* (2018) and related work on *physics-informed* Gaussian process regression (e.g. Wheeler *et al.*, 2014; Wang & Berger, 2016; Jidling *et al.*, 2017; Chen *et al.*, 2020), in that finite difference approximations enable approximate Gaussian conditioning to be performed. However, these authors are addressing a *fundamentally different* problem to that addressed in this chapter; we aim to quantify numerical uncertainty for a single (i.e. non-random) PDE. Accordingly, we emphasise issues that are critical to the performance of PNM, such as explicitly assessing the error of point estimation and quality of the credible sets provided by our PNM (Section 5.4).

## 5.3 Prior Construction

This section is dedicated to studying the sample path properties of Gaussian Processes with the Matérn covariance function, motivated by attempting to construct a prior whose samples are elements of elements of $C^\beta([0,T] \times \Gamma)$, the set in which a solution to (5.1) is sought. First, in Section 5.3.1 we introduce the technical notions of *sample continuity* and *sample differentiability*, clarifying what properties of the prior are required to hold. These sample-path properties are distinct from mean-square properties, the latter being more commonly studied. Then, in Section 5.3.2 for the univariate case, we formally prove that a Gaussian Process with the Matérn covariance function have sample paths with the required degree of smoothness. For the multivariate case, we speculate that the required properties holds for a particular Matérn tensor product, which we then advocate as a default choice for our PNM.

### 5.3.1 Mathematical Properties for the Prior

The strong solution of (5.1) is assumed to be an element of $C^\beta([0,T] \times \Gamma)$, therefore it is logical to construct a prior distribution whose samples also belong to this set. In

particular, if the true solution has $\beta_i$ derivatives in the variable $z_i$ (for instance because the PDE features a term $\partial_{z_i}^{\beta_i} u$, where we have set $z = (t, x)$), it would be appropriate to construct a prior (and hence a posterior) whose samples also have $\beta_i$ derivatives in the variable $z_i$. In the following, we prove a prior with this property for the univariate case, and speculate how the same can be achieved in the multivariate case.

To make this discussion precise, we make explicit a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ and recall the fundamental definitions of sample continuity and sample differentiability for a random field $X \colon I \times \Omega \to \mathbb{R}$ defined on an open, pathwise-connected set $I \subseteq \mathbb{R}^d$ (i.e. $I$ is an interval when $d = 1$):

**Definition 23** (Sample Continuity)**.** $X$ is said to be *sample continuous* if, for $\mathbb{P}$-almost all $\omega \in \Omega$, the sample path $X(\cdot, \omega)$ is continuous (everywhere) in $I$.

**Definition 24** (Sample Differentiability)**.** Let $v^1, \ldots, v^p \in \mathbb{R}^d$ be a sequence of directions and $v = (v^1, \ldots, v^p)$. Then $X$ is said to be *sample partial differentiable in the sequence of directions* $v$ if for $\mathbb{P}$-almost all $\omega \in \Omega$, the following limit exists for all $z \in I$

$$\mathcal{D}^p X(z, v, \omega) = \lim_{h_1 \to 0} \ldots \lim_{h_p \to 0} \frac{\Delta^p X(z, v, h, \omega)}{\prod_{i=1}^p h_i} < \infty$$

where

$$\Delta^p X(z, v, h, \omega) \coloneqq \sum_{r \in \{0,1\}^p} (-1)^{p - \sum_{i=1}^p r_i} X\left(z + \sum_{i=1}^p r_i h_i v^i, \omega\right).$$

The limits above are taken sequentially from left to right. In the discussions that follow, we take $v^i \in \{e_1, e_2, \ldots, e_d\}$, the standard Cartesian unit basis vectors of $\mathbb{R}^d$, in which case the usual partial derivatives are retrieved, and we use the shorthand $\mathcal{D}^p X(z, v, \omega) = \partial^\alpha X(z, \omega)$ to denote sample partial derivatives, where $\alpha = (\alpha_1, \ldots, \alpha_p)$, $|\alpha| = p$, and $\alpha_i$ denotes the number of times the variable $z_i$ is differentiated.

A similar property, which is more easily studied than sample continuity (resp. sample differentiability), is mean-square continuity (resp. mean-square differentiability). This property is recalled next, since we will make use of mean-square properties en route to establishing sample path properties in Section 5.3.2.

**Definition 25** (Mean-Square Continuity)**.** $X$ is said to be *mean-square continuous at* $z \in I$ if

$$\mathbb{E}\left[X(z, \omega)^2\right] < \infty, \qquad \lim_{z' \to z} \mathbb{E}\left[(X(z', \omega) - X(z, \omega))^2\right] = 0.$$

**Definition 26** (Mean-Square Differentiability)**.** Let $v^1, \ldots, v^p \in \mathbb{R}^d$ be a sequence of directions and $v = (v^1, \ldots, v^p)$. Then $X$ is said to be *mean-square partial differentiable at* $z \in I$ *in the sequence of directions* $v$ if there exists a finite random field $\omega \mapsto \mathcal{D}^p_{\mathrm{MS}} X(z, v, \omega)$

such that

$$\lim_{h_1 \to 0} \dots \lim_{h_p \to 0} \mathbb{E}\left[\left(\frac{\Delta^p X(z, v, h, \omega)}{\prod_{i=1}^p h_i} - \mathcal{D}^p_{\text{MS}} X(z, v, \omega)\right)^2\right] = 0$$

is well-defined.

For a mean-square differentiable Gaussian processes $X$, with mean function $\mu \in C^\alpha(I)$ and covariance function $\Sigma \in C^{(\alpha,\alpha)}(I \times I)$, one has

$$\partial^\alpha_{\text{MS}} X \sim \mathcal{GP}(\partial^\alpha \mu, \partial^\alpha \bar{\partial}^\alpha \Sigma)$$

where we use the shorthand $\mathcal{D}^p_{\text{MS}} X(z, v, \omega) = \partial^\alpha_{\text{MS}} X(z, \omega)$ to denote mean-square partial derivatives, where again the $v^i$ are unit vectors parallel to the coordinate axes, $|\alpha| = p$, and $\alpha_i$ denotes the number of times the variable $z_i$ is differentiated.[3] See Stein (1999, Section 2.6). If $X$ is mean-square continuous (resp. mean-square differentiable for all $\alpha \leq \beta$) at all $z \in I$, then we say simply that $X$ is *mean-square continuous* (resp. *order $\beta$ mean-square differentiable* ). In contrast to sample path properties, mean-square properties are often straight-forward to establish. In particular, if $X$ is weakly stationary with autocovariance function $\Sigma(z) = \Sigma(z, 0)$, then

$$\mathbb{E}\left[(X(z, \omega) - X(z', \omega))^2\right] = 2(\Sigma(0) - \Sigma(z - z')), \tag{5.5}$$

meaning that $X$ is mean-square continuous whenever its autocovariance function $\Sigma$ is continuous at 0 (Stein, 1999, Section 2.4).

### 5.3.2 Matérn Covariance Function

Surprisingly, we are unable to find explicit results in the literature for the sample path properties of commonly used covariance models; this is likely due to the comparative technical difficulty in establishing sample path properties compared to mean-square properties. In this section, we furnish a gap in the literature by rigorously establishing the sample differentiability properties of Gaussian processes defined by the Matérn covariance function, in the univariate case.

**Definition 27** (Matérn Covariance)**.** Let $\nu = p + \frac{1}{2}$ where $p \in \mathbb{N}$. The Matérn covariance function is defined, for $z, z' \in \mathbb{R}$, as

$$K_\nu(z, z') = K_\nu(z - z') = \sigma^2 \exp\left(-\frac{|z - z'|}{\rho}\right) \frac{p!}{(2p)!} \sum_{k=0}^{p} \frac{(2p - k)!}{(p - k)!k!} \left(\frac{2}{\rho}\right)^k |z - z'|^k. \tag{5.6}$$

[3] The shorthand notation surpresses the order in which derivatives are taken, and can therefore only be applied in situations where partial derivatives are continuous, to ensure that their order can be interchanged without affecting the result. In the following, the notation $\partial^\alpha_{\text{MS}} X$ is used only for Gaussian processes with $\partial^\alpha \bar{\partial}^\alpha \Sigma \in C(I \times I)$.

**Proposition 5.1** (Mean-Square Differentiability of Univariate Matérn)**.** *Let $I \subseteq \mathbb{R}$ be an open set and let $\mu \in C^p(I)$. Then any process $X \sim \mathcal{GP}(\mu, K_\nu)$, with $K_\nu$ as in* (5.6) *with $\nu = p + \frac{1}{2}$, is order $p$ mean-square differentiable. Furthermore, $\partial^p_{\text{MS}} X$ is mean-square continuous.*

*Proof of Proposition 5.1.* The assumed regularity of $\mu$, being an element of $C^p(I)$, implies that $X$ and $X - \mu$ have identical differentiability properties up to order $p$, and we therefore assume $\mu = 0$ for simplicity in the remainder.

The mean square differentiability of the Matérn covariance function has been well-documented. In particular, because of the stationarity of the Matérn covariance function, $X$ is order $p$ mean-square differentiable if and only if $K_\nu^{(p,p)}(0,0) = (-1)^p K_\nu^{(2p)}(0)$ exists and is finite, and the Matérn covariance function with parameter $\nu$ is $2p$ times differentiable if and only if $\nu > p$; see Section 2 of Stein (1999). This establishes existence of the mean-square derivative $\partial^{(p)}_{\text{MS}} X$.

It remains to prove that $\partial^{(p)}_{\text{MS}} X$ is mean-square continuous. From the discussion in (5.5), $\partial^{(p)}_{\text{MS}} X$ is mean-square continuous if and only if its autocovariance function, $K_\nu^{(2p)}$, is continuous at 0. Let $h = z - z'$ and $K_\nu(h) = f(h)g(h)$ where

$$f(h) = \sigma^2 \exp\left(-\frac{|h|}{\rho}\right) \frac{p!}{(2p)!}, \qquad g(h) = \sum_{k=0}^{p} \frac{(2p-k)!}{(p-k)!k!} \left(\frac{2}{\rho}\right)^k |h|^k$$

so that, by Leibniz's generalised product rule, for $m \in \mathbb{N}_0$,

$$K_\nu^{(m)}(h) = \sum_{n=0}^{m} \binom{m}{n} f^{(m-n)}(h) g^{(n)}(h).$$

One can verify that, for $n \in \{0, 1, \ldots, m\}$,

$$f^{(m-n)}(h) = \begin{cases} \frac{(-1)^{m-n}}{\rho^{m-n}} \sigma^2 \exp(-\frac{h}{\rho}) \frac{p!}{(2p)!} & h > 0 \\ \frac{1}{\rho^{m-n}} \sigma^2 \exp(-\frac{-h}{\rho}) \frac{p!}{(2p)!} & h < 0 \end{cases}$$

$$g^{(n)}(h) = \begin{cases} \sum_{k=0}^{p-n} \frac{(2p-n-k)!}{(p-n-k)!k!} (\frac{2}{\rho})^{n+k} h^k & h > 0, n \leq p \\ (-1)^n \sum_{k=0}^{p-n} \frac{(2p-n-k)!}{(p-n-k)!k!} (\frac{2}{\rho})^{n+k} (-h)^k & h < 0, n \leq p \\ 0 & h \neq 0, n > p \end{cases}$$

from which it follows that

$$\lim_{h\downarrow 0} K_\nu^{(2p)}(h) = \frac{p!}{(2p)!}\sigma^2 \sum_{n=0}^{p} \binom{2p}{n} \frac{(2p-n)!}{(p-n)!} \frac{(-1)^{2p-n}}{\rho^{2p-n}} \left(\frac{2}{\rho}\right)^n$$
$$= \frac{\sigma^2}{\rho^{2p}} \sum_{n=0}^{p} \binom{p}{n} (-1)^{2p-n} 2^n = (-1)^p \frac{\sigma^2}{\rho^{2p}} (2-1)^p = (-1)^p \frac{\sigma^2}{\rho^{2p}}$$

and an analogous calculation shows

$$\lim_{h\uparrow 0} K_\nu^{(2p)}(h) = (-1)^p \frac{\sigma^2}{\rho^{2p}}.$$

Finally, we must check that the value $K_\nu^{(2p)}(0)$ agrees with the two limits just derived. $K_\nu^{(2p-1)}(h)$ is continuously differentiable, so $K_\nu^{(2p-1)}(0) = 0$ because it is an odd function (as it is an odd derivative of an even function $K_\nu$). Thus we have that

$$K_\nu^{(2p)}(0) = \lim_{h\to 0} \frac{K_\nu^{(2p-1)}(h) - K_\nu^{(2p-1)}(0)}{h} = \lim_{h\to 0} \frac{K_\nu^{(2p-1)}(h)}{h}$$
$$= \lim_{h\to 0} \frac{p!}{(2p)!}\sigma^2 \exp\left(-\frac{|h|}{\rho}\right) \sum_{n=0}^{p-1} \binom{2p-1}{n} \frac{(2p-n-1)!}{(p-n-1)!} \frac{(-1)^{2p-1-n}}{\rho^{2p-1-n}} \left(\frac{2}{\rho}\right)^{n+1} + O(h)$$
$$= (-1)^p \sigma^2 \sum_{n=0}^{p-1} \binom{p-1}{n} \frac{(-1)^{p-1-n} 2^n}{\rho^{2p}} = (-1)^p \frac{\sigma^2}{\rho^{2p}} (2-1)^{p-1} = (-1)^p \frac{\sigma^2}{\rho^{2p}}$$

as required. □

Following a general approach outlined in Potthoff (2010), and focussing initially on the univariate case, our first step toward establishing sample differentiability is to establish sample continuity of the mean-square derivatives. Recall that, for two stochastic processes $X, \tilde{X}$ on a domain $I$, we say $\tilde{X}$ is a *modification of* $X$ if, for every $z \in I$, $\mathbb{P}(X(z,\omega) = \tilde{X}(z,\omega)) = 1$. A modification of a stochastic process does not change its mean square properties, but sample path properties need not be invariant to modification.[4] For Gaussian processes, which are characterised up to modifications by their finite dimensional distributions, it is standard practice to work with continuous modifications when they exist (see for example Dudley, 1967; Marcus & Shepp, 1972).

**Proposition 5.2.** *Let $X$ be as in Proposition 5.1. Then $\partial^i_{\text{MS}} X$ has a modification that is sample continuous for all $0 \leq i \leq p$.*

[4] To build intuition into the role of modifications, let $X\colon [0,1]\times\Omega \to \mathbb{R}$ be a sample continuous stochastic process and consider the process $\tilde{X}(z,\omega) \coloneqq X(z,\omega) + 1[z = Z(\omega)]$ where $Z \sim \mathcal{U}(0,1)$, independent of $X$. Then $\tilde{X}$ is a modification of $X$ whose finite dimensional distributions (and hence mean square properties) are identical to those of $X$, but $\tilde{X}$ is almost surely *not* sample continuous. In such circumstances it is convenient (and standard practice) to work with the sample continuous process, $X$, as opposed to $\tilde{X}$.

As in the proof of Proposition 5.1, the assumed regularity of $\mu$, being an element of $C^p(I)$, implies that $X$ and $X-\mu$ have identical differentiability properties up to $p$th order, and we may therefore assume $\mu = 0$.

Our main tool is the *Kolmogorov continuity theorem* (see for example Kunita (1997), Section 1.4):

**Theorem 5.1** (Kolmogorov's Continuity Theorem)**.** *Let $I \subseteq \mathbb{R}^d$ be an open set, and let $\mathbf{z}$ be a dense subset of $I$. Let $X : I \times \Omega \to \mathbb{R}$ be a random field. If there exists constants $\alpha, \beta, C > 0$ such that*

$$\mathbb{E}\left[|X(z,\omega) - X(z',\omega)|^\alpha\right] \leq C\|z - z'\|^{d+\beta}$$

*for all $z, z' \in \mathbf{z}$, then there exists a modification of $X$ that is sample continuous.*

**Lemma 5.2.** *Let $I \subseteq \mathbb{R}^d$ be an open set and suppose that a positive definite function $\Sigma : I \times I \to \mathbb{R}$ satisfies*

$$\Sigma(z,z) + \Sigma(z',z') - 2\Sigma(z,z') \leq C\|z - z'\|^\gamma$$

*for some $\gamma, C \in (0,\infty)$ and all $z, z' \in I$. Let $X \sim \mathcal{GP}(0,\Sigma)$. Then there exists a modification of $X$ that is sample continuous.*

*Proof of Lemma 5.2.* Notice that

$$\Sigma(z,z) + \Sigma(z',z') - 2\Sigma(z,z') = \mathbb{E}[(X(z,\omega) - X(z',\omega))^2],$$

so if $\gamma > d$ then the required result follows from Kolmogorov's continuity theorem (Theorem 5.1, with $\alpha = 2$). If not, then we can consider higher order moments via Isserlis's theorem: For $n \in \mathbb{N}$,

$$\mathbb{E}[(X(z,\omega) - X(z',\omega))^{2n}] = \frac{(2n)!}{2^n n!}(\mathbb{E}[(X(z,\omega) - X(z',\omega))^2])^n$$

and thus, with any $n > d/\gamma$, we have

$$\mathbb{E}[(X(z,\omega) - X(z',\omega))^{2n}] \leq \frac{(2n)!}{2^n n!}C^n\|z - z'\|^{\gamma n} \leq \tilde{C}\|z - z'\|^{d+\beta}$$

with $\tilde{C} = \frac{(2n)!}{2^n n!}C^n$ and $\beta = \gamma n - d$. The result then follows from Kolmogorov's continuity theorem (Theorem 5.1, with $\alpha = 2n$). □

*Proof of Proposition 5.2.* Our aim is to show that $\partial_{\text{MS}}^{(i)} X$ satisfies the preconditions of Lemma 5.2. This process has covariance function $K_\nu^{(i,i)}(z,z') = (-1)^i K_\nu^{(2i)}(z - z')$. From

stationarity we have, with $h = z - z'$,

$$K_\nu^{(2i)}(z,z) + K_\nu^{(2i)}(z',z') - 2K_\nu^{(2i)}(z,z') = 2K_\nu^{(2i)}(0) - 2K_\nu^{(2i)}(h).$$

From similar calculations to those performed in the proof of Proposition 5.1, we have that for all $0 \leq i \leq p$,

$$\begin{aligned} -2K_\nu^{(2i)}(h) &= \frac{2\sigma^2 p!}{(2p)!} \exp\left(-\frac{|h|}{\rho}\right) \sum_{n=0}^{\min(2i,p)} \sum_{k=0}^{p-n} \binom{2i}{n} \frac{(2p-n-k)!}{(p-n-k)!k!} \frac{(-1)^{2i-n+1}}{\rho^{2i-n}} \left(\frac{2}{\rho}\right)^{n+k} |h|^k \\ &= \exp\left(-\frac{|h|}{\rho}\right) \sum_{k=0}^{p} a_k |h|^k \end{aligned}$$

for some real coefficients $a_k$. Therefore:

$$\begin{aligned} |2K_\nu^{(2i)}(0) - 2K_\nu^{(2i)}(h)| &= \left| a_0 - a_0 \exp\left(-\frac{|h|}{\rho}\right) - \exp\left(-\frac{|h|}{\rho}\right) \sum_{k=1}^{p} a_k |h|^k \right| \\ &\leq \left| a_0 - a_0 \exp\left(-\frac{|h|}{\rho}\right) \right| + \left| \exp\left(-\frac{|h|}{\rho}\right) \sum_{k=1}^{p} a_k |h|^k \right| \end{aligned} \tag{5.7}$$

This final term can be upper bounded by an expression of the form $C|h|^\gamma$ for sufficiently large $C > 0$ and $\gamma = 1$. Indeed, as $h \to 0$ the behaviour of (5.7) is $O(|h|)$. As $|h| \to \infty$ the exponential term dominates and (5.7) decays to $a_0$. In the region $0 < |h| < \infty$, (5.7) is smooth. Thus we can use Lemma 5.2 to conclude that $\partial_{\text{MS}}^{(i)} X$ has a modification that is sample continuous. □

The second step is to leverage a fundamental result on the sample path properties of stochastic processes.

**Theorem 5.2** (Criterion for Sample Differentiability; Theorem 3.2 of Potthoff (2010))**.** *Let $I \subseteq \mathbb{R}^d$ be an open, pathwise connected set, and consider a random field $X \colon I \times \Omega \to \mathbb{R}$ such that $\mathbb{E}[X(z,\omega)^2] < \infty$ for all $z \in I$. Suppose $X$ is first order mean-square differentiable, with mean-square partial derivatives $\mathcal{D}^1_{\text{MS}} X(\cdot, e_k, \omega)$, $1 \leq k \leq d$, themselves being mean-square continuous and having modifications that are sample continuous. Then $X$ has a modification $\tilde{X}$ that is first order sample partial differentiable, with partial derivatives $\mathcal{D}^1 \tilde{X}(\cdot, e_k, \omega)$, $1 \leq k \leq d$, themselves being sample continuous and satisfying, almost surely, $\mathcal{D}^1 \tilde{X}(\cdot, e_k, \omega) = \mathcal{D}^1_{\text{MS}} X(\cdot, e_k, \omega)$, $1 \leq k \leq d$.*

Since continuity of partial derivatives implies differentiability, the conclusion of Theorem 5.2 implies that $X$ is first order sample differentiable. In particular, the existence of continuous mean square partial derivatives, and sample continuous modifications of those

mean square partial derivatives is sufficient to imply the existence of a modification with continuous sample partial derivatives.

Iterative application of Theorem 5.2 to higher order derivatives provides the following.

**Corollary 5.1.** *Fix $p \in \mathbb{N}$. Let $I \subseteq \mathbb{R}^d$ be an open, pathwise connected set and consider $X \sim \mathcal{GP}(\mu, \Sigma)$ with $\mu \in C^p(I)$ and $\Sigma \in C^{(p,p)}(I \times I)$, so that $X$ has mean-square partial derivatives $\partial^\beta_{\text{MS}} X$, $\beta \in \mathbb{N}_0^d, |\beta| \leq p$. Suppose $\partial^\beta_{\text{MS}} X$ is mean-square continuous and sample continuous for all $|\beta| \leq p$. Then $X$ has continuous sample partial derivatives $\partial^\beta X$, and they satisfy $\partial^\beta X = \partial^\beta_{\text{MS}} X$ almost surely, for all $|\beta| \leq p$.*

For $p = 1$ the result is immediate from Theorem 5.2, so in what follows we concentrate on $p > 1$.

The main technical challenge of this proof is to deal with modifications, which arise with each application of Theorem 5.2. Recall that two stochastic processes $X$ and $\tilde{X}$ are said to be *indistinguishable* if $\mathbb{P}(X(z, \omega) = \tilde{X}(z, \omega)\ \forall z \in I) = 1$. If $X$ and $\tilde{X}$ are modifications of each other and each is sample continuous, then $X$ and $\tilde{X}$ are indistinguishable (Jeanblanc *et al.*, 2009, Section 1.1).

*Proof of Corollary 5.1.* We first present a proof for $d = 1$, to improve transparency of the argument, then we present the argument for the general case $d \geq 1$. Note that, since we are considering Gaussian processes, the requirement for a second moment in Theorem 5.2 is automatically satisfied.

For each $0 \leq i < p$, it is assumed that $\partial^i_{\text{MS}} X$ has mean square derivative $\partial^{i+1}_{\text{MS}} X$ that is mean square continuous and sample continuous. Theorem 5.2 therefore implies that each $\partial^i_{\text{MS}} X$ has a modification, denoted $\psi_i$, that is sample continuously differentiable, and satisfying $\partial \psi_i = \partial^{i+1}_{\text{MS}} X$ almost surely. Since $\psi_i$ and $\partial^i_{\text{MS}} X$ are sample continuous they are indistinguishable; i.e. almost surely $\psi_i = \partial^i_{\text{MS}} X$. It follows that, for each $0 \leq i < p$, we have almost surely that $\partial^i \psi_0 = \partial^{i-1}(\partial \psi_0) = \partial^{i-1} \psi_1 = \cdots = \psi_i$, while for $i = p$ we have that $\partial^p \psi_0 = \partial \psi_{p-1} = \partial^p_{\text{MS}} X$.

The case $d \geq 1$ is analogous with more notation is involved; though, since we assumed $\Sigma \in C^{(p,p)}(I \times I)$ we may employ the shorthand notation $\partial^\beta_{\text{MS}} X$ for all $|\beta| \leq p$ (since the order of derivatives can be freely interchanged). For each $0 \leq i < p$ and $|\beta| = i$, it is assumed that $\partial^\beta_{\text{MS}} X$ has mean square partial derivatives $\partial^{\beta+\gamma}_{\text{MS}} X$, $|\gamma| = 1$, that are mean square continuous and sample continuous. Theorem 5.2 therefore implies that each $\partial^\beta_{\text{MS}} X$ has a modification, denoted $\psi_\beta$, that is sample continuously differentiable, and satisfying $\partial^\gamma \psi_\beta = \partial^{\beta+\gamma}_{\text{MS}} X$ almost surely for all $|\gamma| = 1$. Since $\psi_\beta$ and $\partial^\beta_{\text{MS}} X$ are sample continuous they are indistinguishable; i.e. almost surely $\psi_\beta = \partial^\beta_{\text{MS}} X$. It follows that, for each $0 \leq i < p$ and $|\beta| = i$, we have almost surely that $\partial^\beta \psi_0 = \psi_\beta$, while for and $|\beta| = p$ we have that $\partial^\beta \psi_0 = \partial^\beta_{\text{MS}} X$. □

This provides a strategy to establish sample properties of Matérn processes, such as the following:

**Corollary 5.2** (Sample Differentiability of Univariate Matérn)**.** *Let $I \subseteq \mathbb{R}$ be an open interval and let $X \sim \mathcal{GP}(\mu, K_\nu)$, with $\mu \in C^p(I)$ and $K_\nu$ as in* (5.6)*. Then there exists a modification $\tilde{X}$ of $X$ such that $\mathbb{P}(\tilde{X} \in C^p(I)) = 1$.*

*Proof.* By Proposition 5.1, $X$ is order $p$ mean-square differentiable and $\partial^i_{\text{MS}} X$ is mean-square continuous for $0 \leq i \leq p$. By Proposition 5.2, we may work with a modification $\tilde{X}$ of $X$ such that $\partial^i_{\text{MS}} \tilde{X} (= \partial^i_{\text{MS}} X)$ is sample continuous for each $0 \leq i \leq p$. One can directly verify that $K_\nu \in C^{(p,p)}(I \times I)$; see the calculations in Proposition 5.1. The result then follows from Corollary 5.1. □

Corollary 5.2 is stronger than existing results in the literature, the most relevant of which is Scheuerer (2010, Theorem 5), who showed that samples from $\mathcal{GP}(\mu, K_{\nu+\epsilon})$ are $C^p(I)$ for any $\epsilon > 0$.

Finally, we conjecture a multivariate version of the previous result, which intuitively allow for different smoothness in the different variables, which is necessary to properly capture the regularity of solutions to PDEs.

**Conjecture 5.1** (Sample Differentiability of Matérn Tensor Product)**.** *Let $I = (a_1, b_1) \times \cdots \times (a_d, b_d)$ be a bounded hyper-rectangle in $\mathbb{R}^d$. Fix $\beta \in \mathbb{N}_0^d$. Let $\mu \in C^\beta(I)$ be bounded in $I$, and consider a covariance function $\Sigma \colon I \times I \to \mathbb{R}$ of the form*

$$\Sigma(z, z') = \prod_{i=1}^{d} K_{\nu_i}(z_i - z_i')$$

*where $z = (z_1, z_2, \ldots, z_d)$, $z' = (z_1', z_2', \ldots, z_d')$ and $\nu_i = \beta_i + \frac{1}{2}$ for each $i = 1, \ldots, d$. Then a Gaussian process of the form $X \sim \mathcal{GP}(\mu, \Sigma)$ has a modification $\tilde{X}$ that satisfies $\mathbb{P}(\tilde{X} \in C^\beta(I)) = 1$.*

We do not provide a proof to this result. However, we note that the Matérn Tensor Product process possess continuous (in the mean square sense) mean square partial derivatives of the desired orders by construction, and those mean squared partial derivatives also possess continuous modifications by Kolmogorov's continuity theorem, which is straightforward to verify using the univariate calculations performed in the proofs of Proposition 5.2. These two conditions were sufficient to imply the existence of a modification with continuous sample partial derivatives via Theorem 5.2, in the case of first order partial derivatives. So we speculate with a suitable analogue of Theorem 5.2, one can similarly show the samples of the Matérn Tensor Product almost surely lies in $C^\beta(I)$. While we do not offer a proof for Conjecture 5.1, we hope that the theoretical analysis in

this section motivates our use of the Matérn Tensor Product 'in spirit' as prior choice for our proposed probabilistic numerical method for PDEs.

## 5.4 Experimental Assessment

In this section, proof-of-concept numerical studies of three different initial value problems are presented. The first and simplest case is a homogeneous Burger's equation, a PDE with one nonlinear term and a solution that is known to be smooth. The second case is a porous medium equation, with two nonlinear terms appearing in the PDE and a solution that is known to be piecewise smooth, so that a classical solution does not exist and our modelling assumptions are violated. The third case returns to Burger's equation but now with forcing, to simulate a scenario where the right hand side $f$ is a black box function that may be evaluated at a high computational cost. All three experiments are *synthetic*, in the sense that the functions $f$, $g$ and $h$, which in our motivating task are considered to be black boxes associated with a high computational cost, are in actual fact simple analytic expressions, enabling a thorough empirical assessment to be performed.

In order to assess the empirical performance of our algorithm, two distinct performance measures were employed. The first of these aims to assess the accuracy of the posterior mean, which is analogous to how classical numerical methods are assessed. For this purpose the $L^\infty$ error was considered:

$$E_\infty \coloneqq \sup_{t\in[0,T],x\in\Gamma} |\mu^n(t,x) - u(t,x)| \tag{5.8}$$

In practice the value of (5.8) is approximated by taking the maximum over the grid $\mathbf{t}\times\mathbf{x}$ on which the data $\mathbf{y}^0,\ldots,\mathbf{y}^{n-1}$ were obtained. Accuracy that is comparable to a classical numerical method is of course desirable, but it is not our goal to compete with classical numerical methods in terms of $L^\infty$ error. The second statistic that we consider assesses whether the distributional output from our PNM is *calibrated*, in the sense that the scale of the Gaussian posterior is comparable with the difference between the posterior mean $\mu^n$ and the true solution $u$ of (5.1):

$$Z \coloneqq \sup_{t\in[0,T],x\in\Gamma} \frac{|\mu^n(t,x) - u(t,x)|}{\hat{\sigma}\Sigma^n(t,x)^{1/2}} \tag{5.9}$$

This performance measure will be called a *Z-score*, in analogy with traditional terminology from statistics. For the purpose of this exploratory work, values of $Z$ that are orders of magnitude smaller than 1 are interpreted as indicating that the distributional output from the PNM is under-confident, while values that are orders of magnitude greater than 1 indicate that the PNM is over-confident. A PNM that is neither under nor over confident is

said to be *calibrated* (precise definitions of the term "calibrated" can be found in Karvonen *et al.*, 2020; Cockayne *et al.*, 2021, but the results we present are straight-forward to interpret using the informal approach just described). Our goal in this work is to develop an approximately Bayesian PNM for nonlinear PDEs that is both accurate and calibrated. Again, in practice the supremum in (5.9) is approximated by the maximum over the $\mathbf{t} \times \mathbf{x}$ grid.

For all experiments below, we consider uniform temporal and spatial grids of respective sizes $n = 2^i + 1$, $m = 2^j + 1$, where $i, j \in \{2, 3, 4, 5, 6, 7\}$. This ensures that the grid points at which data are obtained are strictly nested as either the temporal exponent $i$ or the spatial exponent $j$ are increased. The prior mean $\mu(t, x) = 0$ for all $t \in [0, T]$, $x \in \Gamma$, will be used throughout.

Code to reproduce the numerical results in this section can be found at https://github.com/jwang727/Thesiscode. Examples were implemented in Python.

### 5.4.1 Homogeneous Burger's Equation

Our first example is the homogeneous *Burger's equation*

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} - \alpha\frac{\partial^2 u}{\partial x^2} = 0, \qquad t \in [0, T],\ x \in [0, L]$$

with initial and boundary conditions

$$\begin{aligned} u(0, x) &= 2\alpha\Big(\frac{ak\sin(kx)}{b + a\cos(kx)}\Big), && x \in [0, L] \\ u(t, 0) = u(t, 2\pi) &= 0, && t \in [0, T] \end{aligned}$$

and, for our experiments, $\alpha = 0.02$, $a = 1$, $b = 2$, $k = 1$, $T = 30$ and $L = 2\pi$. These initial and boundary conditions were chosen because they permit a closed-form solution

$$u(t, x) = 2\alpha\Big(\frac{ak\exp(-\alpha k^2 t)\sin(kx)}{b + a\exp(-\alpha k^2 t)\cos(kx)}\Big) \tag{5.10}$$

that can be used as a ground truth for our assessment.

To linearise the differential operator Burger's equation we consider approximations of the form in (5.2), i.e.

$$Q_i u(t_i, x) \coloneqq u_{i-1}(x)\frac{\partial u}{\partial x}(t_i, x)$$

where $u_{i-1}(x)$ was taken equal to the predictive mean $\mu^{i-1}(t_i, x)$ arising from the Gaussian process approximation $U^{i-1}$.

**Default Prior:** Burger's equation has a first order temporal derivative term and a second order spatial derivative term, so following the discussion in Section 5.3 we consider as a default a Gaussian process prior with covariance function $\Sigma$ that is a product between a Matérn 3/2 kernel $K_{3/2}(t,t')$ for the temporal component, and a Matérn 5/2 kernel $K_{5/2}(x,x')$ for the spatial component:

$$\Sigma((t,x),(t',x')) = K_{3/2}(t,t';\rho_1,\sigma_1)K_{5/2}(x,x';\rho_2,\sigma_2) \tag{5.11}$$

The notation in (5.11) makes explicit the dependence on the amplitude hyper-parameters $\sigma_1,\sigma_2$ and the length-scale hyper-parameters $\rho_1,\rho_2$; note that only the product $\sigma \coloneqq \sigma_1\sigma_2$ of the two amplitide parameters is required to be specified. For the experiments below $\sigma$ was estimated as per (5.4), while the length-scale parameters were fixed at values $\rho_1 = 6/\sqrt{3}$, $\rho_2 = 3/\sqrt{5}$ (not optimised; these were selected based on a *post-hoc* visual check of the credible sets in Figure 5.1). Typical output from our PNM equipped with the default prior is presented in Figure 5.1.

**An Alternative Prior:** The Matérn covariance models assume only the minimal amount of smoothness required for the PDE to be well-defined. However, in this assessment the ground truth $u$ is available (5.10) and is seen to be infinitely differentiable in $(0,T]\times[0,2\pi]$. It is therefore interesting to explore whether a prior that encodes additional smoothness can improve on the default prior in (5.11). A prototypical example of such a prior is

$$\Sigma((t,x),(t',x')) = C(t,t';\rho_3,\sigma_3)C(x,x';\rho_4,\sigma_4)$$

where

$$C(z,z';\rho,\sigma) \coloneqq \sigma\left(1+\frac{(z-z')^2}{\rho^2}\right)^{-1}$$

is the *rational quadratic* covariance model. For the experiments below $\sigma$ was estimated as per (5.4), while the length-scale parameters were fixed at values $\rho_3 = \sqrt{3}$, $\rho_4 = \sqrt{3}$.

**Results:** The error $E_\infty$ was computed at 36 combinations of temporal and spatial grid sizes $(n,m)$ and results for the default prior are displayed in the top row of Figure 5.2. It can be seen that the error $E_\infty$ is mostly determined, in this example, by the finite length $n$ of the temporal grid rather than the length $m$ of the spatial grid. The slope of the curves in Figure 5.2 is consistent with a convergence rate of $O(n^{-1})$ for the error $E_\infty$ when spatial discretisation is neglected. The $Z$-scores associated with the default prior (bottom row of Figure 5.2) appear to be bounded as $(n,m)$ are increased, tending toward 0 but taking values of order 1 for all regimes, except for the smallest value ($m = 5$) of the spatial grid. This provides evidence that our proposed PNM, equipped with the default prior, is either

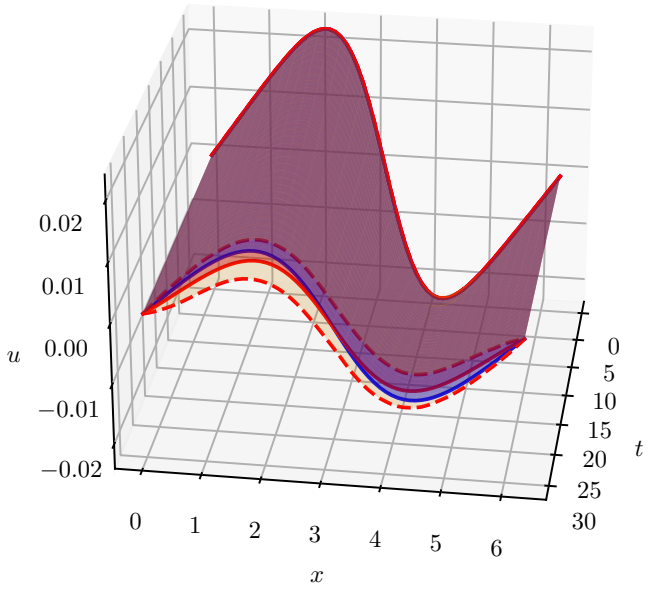



Figure 5.1: Homogeneous Burger's equation: For each point $(t, x)$ in the domain we plot: the analytic solution $u(t, x)$ (blue), the posterior mean $\mu^n(t, x)$ (red) from the proposed probabilistic numerical method, and 0.025 and 0.975 quantiles of the posterior distribution at each point (orange). Here the default prior was used, with a spatial grid of size $m = 65$ and a temporal grid of size $n = 65$.

calibrated or slightly under-confident but, crucially from a statistical perspective, it is not over-confident.

Equivalent results for the alternative covariance model are presented for the error $E_\infty$ in the top row of Figure 5.3 and for the $Z$-score in the bottom row of Figure 5.3. Here the error $E_\infty$ is again gated by the size $n$ of the temporal grid and decreases at a faster rate compared to when the default prior was used. The $Z$-scores associated with the alternative covariance model are larger for small values of $N$ and $M$, but also taking values of order 1 for large values of $N$ and $M$, while remaining slightly larger than the default prior. This suggests the rational quadratic covariance model could be preferable for the homogeneous Burger's equation as it achieves lower error while being only slightly more overconfident for large values of $N$ and $M$, perhaps because the rational quadratic covariance model reflects the true smoothness of the solution better than the Matérn model.

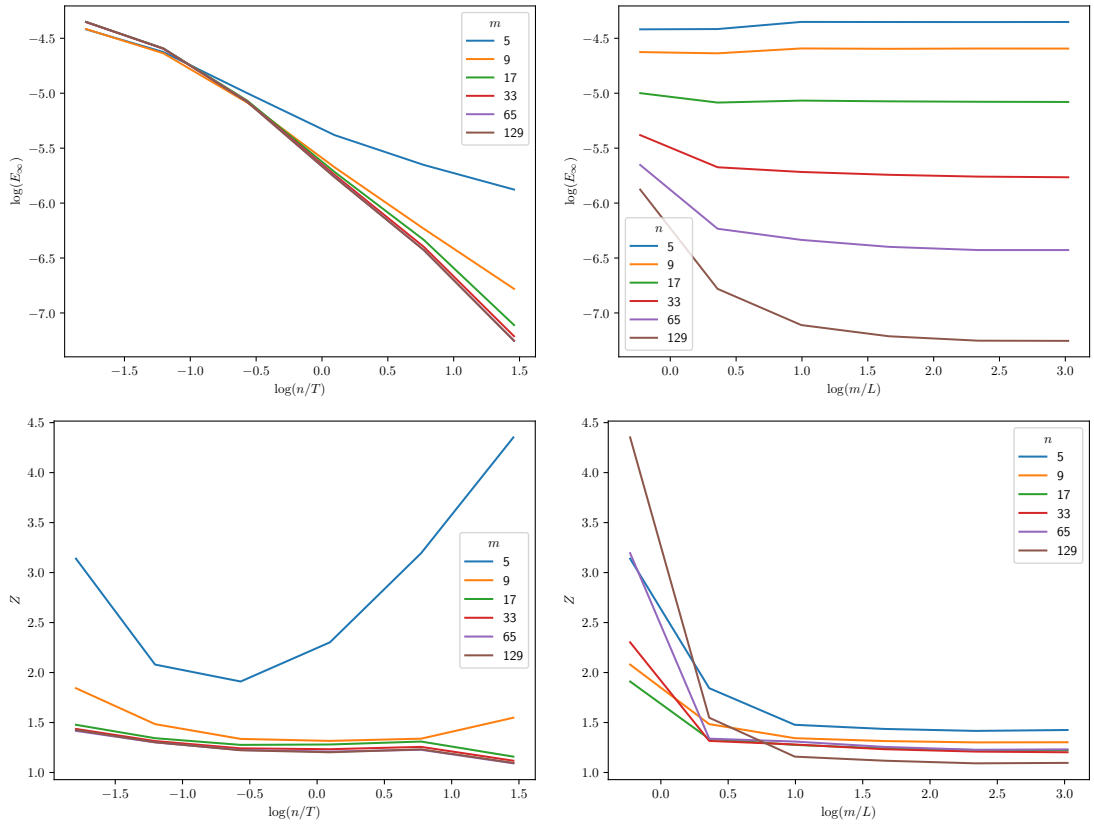



Figure 5.2: Homogeneous Burger's equation, default prior: For each pair $(n, m)$ of temporal $(n)$ and spatial $(m)$ grid sizes considered, we plot: (top left) the error $E_\infty$ for fixed $m$ and varying $n$; (top right) the error $E_\infty$ for fixed $n$ and varying $m$; (bottom left) the $Z$-score for fixed $m$ and varying $n$; (bottom right) the $Z$-score for fixed $n$ and varying $m$.

### 5.4.2 Porous Medium Equation

Our second example is the *porous medium equation*

$$\frac{\partial u}{\partial t} - \frac{\partial^2 (u^k)}{\partial x^2} = 0, \qquad t \in [t_0, t_0 + T],\ x \in [-L/2, L/2], \tag{5.12}$$

which is more challenging compared to Burger's equation because the solution is only piecewise smooth, meaning that a strong solution does not exist and our modelling assumptions are violated. Furthermore, there are two distinct nonlinearities in the differential operator, allowing us to explore the impact of the choice of linearisation on the performance of the PNM. For our experiment we fix $k = 2$, so that the porous medium equation becomes

$$\frac{\partial u}{\partial t} - 2\left(\frac{\partial u}{\partial x}\right)^2 - 2u\frac{\partial^2 u}{\partial x^2} = 0, \qquad t \in [t_0, t_0 + T],\ x \in [-L/2, L/2],$$

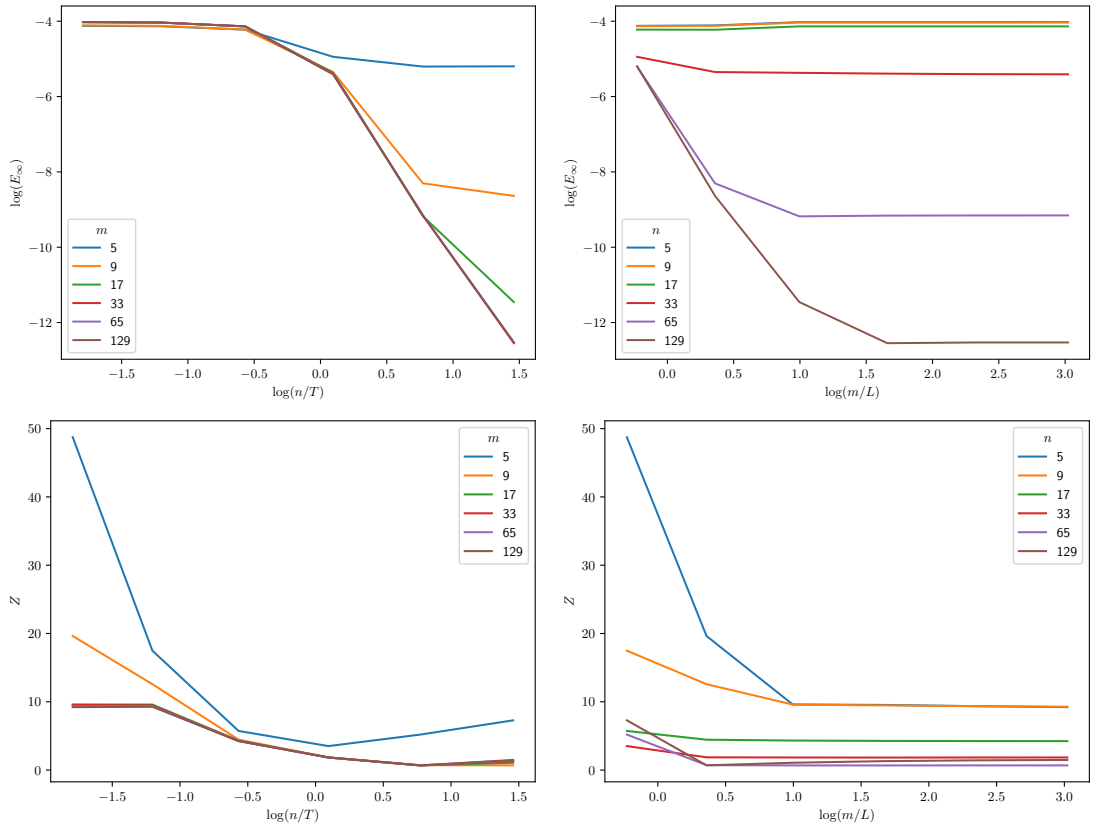



Figure 5.3: Homogeneous Burger's equation, alternative prior: For each pair $(n, m)$ of temporal $(n)$ and spatial $(m)$ grid sizes considered, we plot: (top left) the error $E_\infty$ for fixed $m$ and varying $n$; (top right) the error $E_\infty$ for fixed $n$ and varying $m$; (bottom left) the $Z$-score for fixed $m$ and varying $n$; (bottom right) the $Z$-score for fixed $n$ and varying $m$.

and we consider the initial and boundary conditions

$$\begin{aligned} u(t_0, x) &= t_0^{-1/3} \max(0, 1 - x^2/(12 t_0^{2/3})), && x \in [-L/2, L/2] \\ u(t, -L/2) = u(t, L/2) &= 0, && t \in [t_0, t_0 + T] \end{aligned}$$

with $t_0 = 2$, $T = 8$, $L/2 = 10$. These initial and boundary conditions were chosen because they permit a (unique) closed-form solution, due to Barenblatt (1952):

$$u(t, x) = \max\left(0, \frac{1}{t^{1/3}}\left(1 - \frac{1}{12}\frac{x^2}{t^{2/3}}\right)\right)$$

The solution is therefore only piecewise smooth, with discontinuous first derivatives at $x^2 = 12t^{2/3}$, which are inside of the domain $[-L/2, L/2]$ for all $t \in [t_0, t_0 + T]$.

**Prior:** Henceforth we consider the default prior advocated in Section 5.3 and Section 5.4.1, with amplitude $\sigma$ estimated using maximum likelihood and length-scale pa-

rameters fixed at values $\rho_1 = 1/\sqrt{3}$, $\rho_2 = 2/\sqrt{5}$ (not optimised; based on a simple *post-hoc* visual check).

**Choice of Linearisation:** The differential operator here contains the nonlinear component $Qu = (\partial_x u)^2 + u\partial_x^2 u$ that must be linearised. The first term $(\partial_x u)^2$ is the product of two identical terms, so we linearise it by fixing one of the $\partial_x u$ to suitable constant values adaptively based on quantities that have been pre-computed. The second term $u\partial_x^2 u$ can be linearised in at least two distinct ways, fixing either $u$ or $\partial_x^2 u$ to suitable constant values adaptively based on quantities that have been pre-computed. Thus we consider the two linearisations

$$Q_i^{(1)} u(t_i, x) \coloneqq \frac{\partial \mu^{i-1}}{\partial x}(t_i, x)\frac{\partial u}{\partial x}(t_i, x) + \mu^{i-1}(t_i, x)\frac{\partial^2 u}{\partial x^2}(t_i, x)$$
$$Q_i^{(2)} u(t_i, x) \coloneqq \frac{\partial \mu^{i-1}}{\partial x}(t_i, x)\frac{\partial u}{\partial x}(t_i, x) + u(t_i, x)\frac{\partial^2 \mu^{i-1}}{\partial x^2}(t_i, x)$$

where we recall that $\mu^{i-1}(t_i, x)$ is the predictive mean arising from the Gaussian process approximation $U^{i-1}$. Through simulation we aim to discover which (if either) linearisation is more appropriate for use in our PNM.

**Conservation of Mass:** In addition to admitting multiple linearisations, we consider the porous medium equation because when $k > 1$ it exhibits a *conservation law*, which is typical of many nonlinear PDEs that are physically-motivated. Specifically, integrating (5.12) with respect to $x$ gives

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{-L/2}^{L/2} u(t, x)\,\mathrm{d}x - \frac{\partial(u^k)}{\partial x}\Big|_{-L/2}^{L/2} = 0$$

and, from the fact that $u = 0$ for all $x^2 \geq 12t^{2/3}$, it follows that $\partial_x(u^k) = ku^{k-1}\partial_x u = 0$ for all $x^2 \geq 12t^{2/3}$ and thus $\int_{-L/2}^{L/2} u(t, x)\mathrm{d}x$ is $t$-invariant. A desirable property of a numerial method is that it respects conservations law of this kind; as exemplified by the *finite volume* methods (LeVeque, 2002) and *symplectic integrators* (Sanz-Serna, 1992). Interestingly, it is quite straight-forward to enforce this conservation law in our PNM by adding additional linear constraints to the system in Lemma 5.1. Namely, we add the linear constaints

$$\int_{-L/2}^{L/2} u(t_i, x)\mathrm{d}x = \int_{-L/2}^{L/2} u(t_0, x)\mathrm{d}x = 4(3^{\frac{1}{2}} - 3^{-\frac{1}{2}})$$

at each point $i \in \{1, \ldots, n-1\}$ on the temporal grid. The performance of our PNM both with and without conservation of mass will be considered.

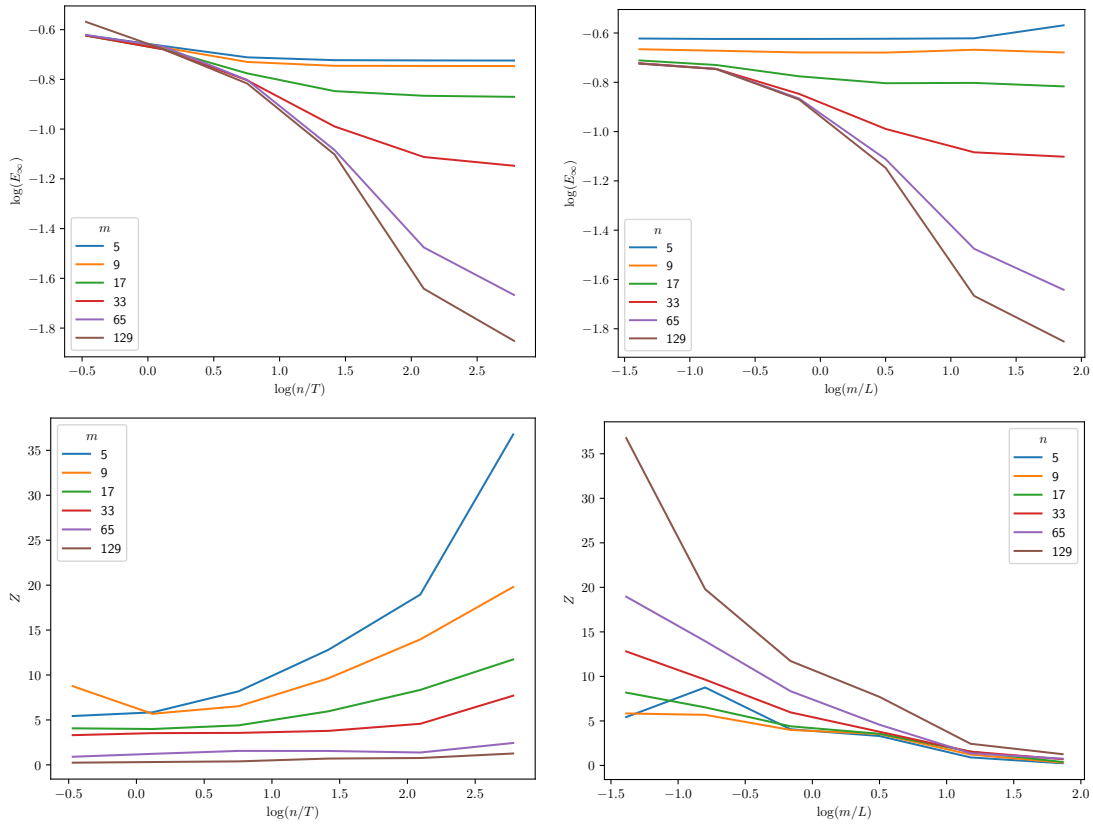



Figure 5.4: Porous medium equation, with linearisation $Q^{(1)}$: For each pair $(n, m)$ of temporal $(n)$ and spatial $(m)$ grid sizes considered, we plot: (top left) the error $E_\infty$ for fixed $m$ and varying $n$; (top right) the error $E_\infty$ for fixed $n$ and varying $m$; (bottom left) the $Z$-score for fixed $m$ and varying $n$; (bottom right) the $Z$-score for fixed $n$ and varying $m$.

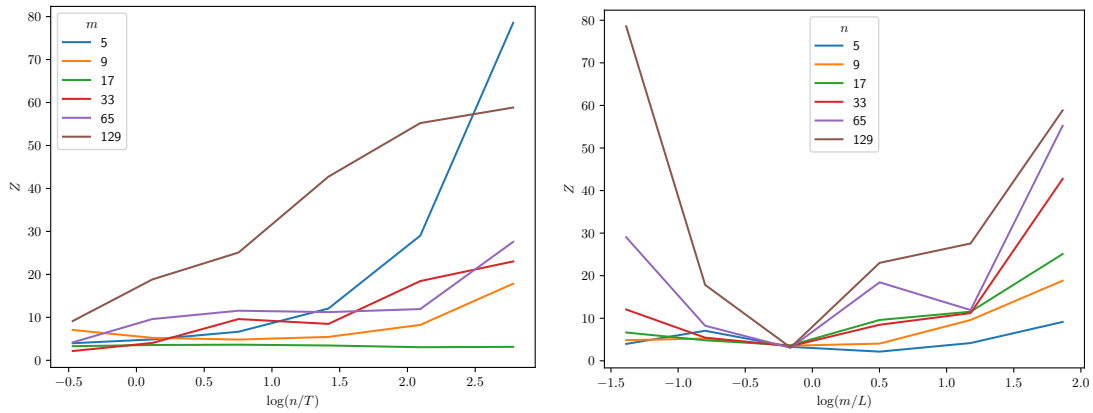



Figure 5.5: Porous medium equation, with linearisation $Q^{(2)}$: For each pair $(n, m)$ of temporal $(n)$ and spatial $(m)$ grid sizes considered, we plot: (left) the $Z$-score for fixed $m$ and varying $n$; (right) the $Z$-score for fixed $n$ and varying $m$.

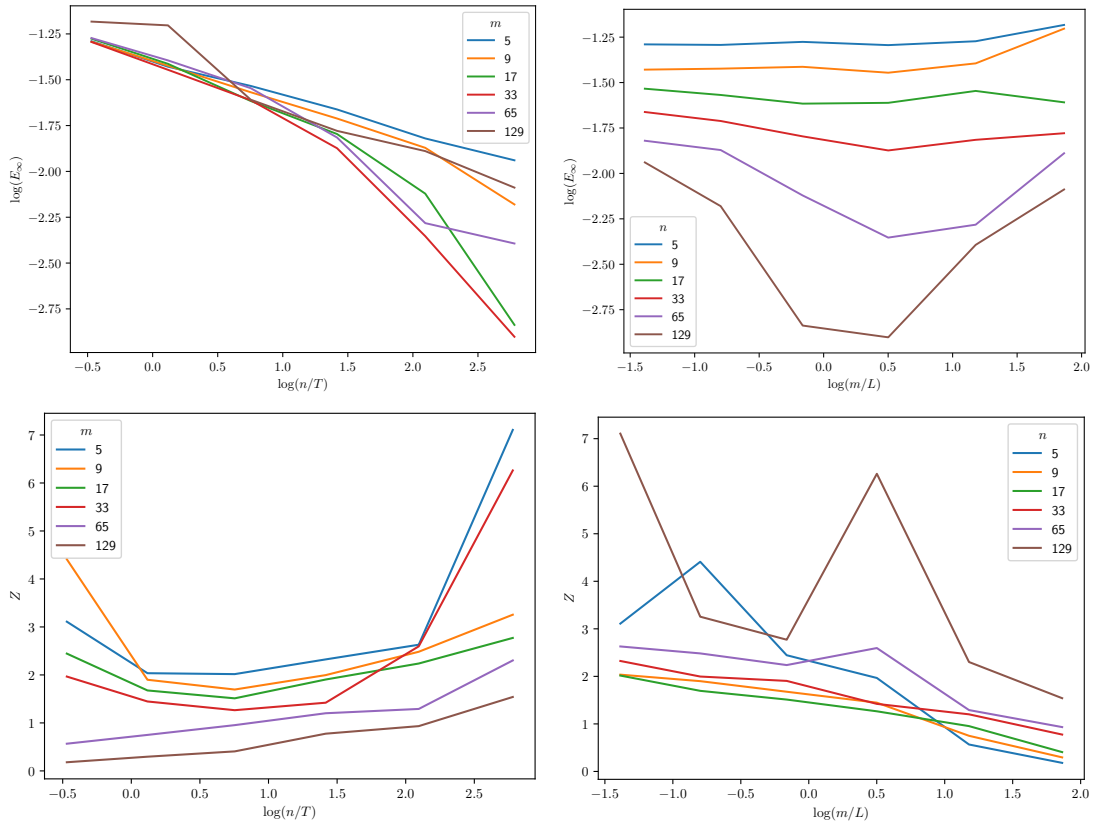



Figure 5.6: Porous medium equation, with mass conserved: For each pair $(n, m)$ of temporal $(n)$ and spatial $(m)$ grid sizes considered, we plot: (top left) the error $E_\infty$ for fixed $m$ and varying $n$; (top right) the error $E_\infty$ for fixed $n$ and varying $m$; (bottom left) the $Z$-score for fixed $m$ and varying $n$; (bottom right) the $Z$-score for fixed $n$ and varying $m$.

**Results:** Empirical results based on the linearisation $Q^{(1)}$ (without conservation of mass) are contained in Figure 5.4. In this case (and in contrast to our results for Burger's equation in Section 5.4.1), the error $E_\infty$ is seen to be gated by the smaller of the finite length $n$ of the temporal grid and the length $m$ of the spatial grid. The $Z$-score values appear to be of order 1 as $(n, m)$ are simultaneously increased, but are higher than for Burger's equation, which may reflect the fact that the solution to the porous medium equation is only piecewise smooth. For increasing $n$ with $m$ fixed the PNM appears to become over-confident, while for increasing $m$ with $n$ fixed the PNM appears to become under-confident; a conservative choice would therefore be to take $m \geq n$.

Next we compared the performance of the linearisation $Q^{(1)}$ with the linearisation $Q^{(2)}$. The error $E_\infty$ associated to $Q^{(2)}$ (not shown) was larger than the error of $Q^{(1)}$, and the $Z$-scores for $Q^{(2)}$ are displayed in Figure 5.5. Our objective is to quantify numerical uncertainty, so it is essential that output from the PNM is calibrated. Unfortunately, it can be seen that the $Z$-scores associated with $Q^{(2)}$ are unsatisfactory; for large $m$ the scores

are two orders of magnitude larger than 1, indicating that the PNM is over-confident. The failure of $Q^{(2)}$ to provide calibrated output is likely due to the fact that approximation of the second order derivative term $\partial_x^2 u$ is more challenging compared to approximation of the solution $u$, since $\partial_x^2$ is less regular than $u$ and since our initial and boundary data relate directly to $u$ itself.

Finally we considered inclusion of the conservation law into the PNM. For this purpose we used the best-performing linearisation $Q^{(1)}$. The errors $E_\infty$ and $Z$-scores are shown in Figure 5.6 and can be compared to the equivalent results without the conservation law applied, in Figure 5.4. It can be seen that the error $E_\infty$ is lower when the conservation law is applied, and moreover the $Z$-scores are slightly reduced, remaining order 1. These results agree with the intuition that incorporating additional physical constraints, when they are known, can have a positive impact on the performance of our PNM.

### 5.4.3 Forced Burger's Equation

Our final experiment concerns a nonlinear PDE whose right hand side $f$ is considered to be a black box, associated with a substantial computational cost. To avoid confounding due to the choice of differential operator, we consider again the differential operator from Burger's equation

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} - \alpha\frac{\partial^2 u}{\partial x^2} = f(t,x), \qquad t \in [0,T],\ x \in [0,L], \tag{5.13}$$

for which the behaviour of our PNM was studied in Section 5.4.1 (in the case $f = 0$). The initial and boundary conditions are

$$\begin{aligned} u(0,x) &= 0, \qquad x \in [0,L] \\ u(t,0) = u(t,L) &= 0, \qquad t \in [0,T] \end{aligned}$$

and we set $\alpha = 1$, $T = 30$, $L = 1$. The aims of this experiment are two-fold: Our first aim is to evaluate the performance of our PNM when the function $f$ is non-trivial (e.g. involving oscillatory behaviour), to understand whether the output from our PNM remains calibrated or not. Recall that our experiments are *synthetic*, meaning that the black box $f$ is in actual fact an analytic expression, in this case

$$f(t,x) \coloneqq 10\sin(6\pi x)\cos(\pi t/10) + 2\left|\sin(3\pi x)\cos(\pi t/10)\right|,$$

enabling a thorough assessment to be performed. This forcing term is deliberately chosen to have some non-smoothness (from the absolute value function) and oscillatory behaviour, as might be encountered in output from a complex computer model. Our second aim is to compare the accuracy of our PNM against a classical numerical method whose compu-

tational budget (as quantified by the number of times $f$ is evaluated) is identical to our PNM.

The solution to (5.13) does not admit a closed form, so for our ground truth we used a numerical solution computed using the `MATLAB` function `pdepe`, which implements Skeel & Berzins (1990) based on a uniform spatial grid of size 512 and an adaptively selected temporal grid. Our PNM was implemented with the same linearisation used in Section 5.4.1.

**Prior:** Again we consider the default prior advocated in Section 5.3 and Section 5.4.1, with amplitude $\sigma$ estimated using maximum likelihood and length-scale parameters fixed at values $\rho_1 = 0.5/\sqrt{3}$, $\rho_2 = 0.5/\sqrt{5}$ (not optimised; based on a simple *post-hoc* visual check).

**Crank–Nicolson Benchmark:** In this scenario, where the black box function $f$ is associated with a high computational cost, non-adaptive numerical methods are preferred, to control the total computational cost. As discussed in Section 2.2 of Chapter 2, from a classical perspective, the finite difference methods are natural candidates for the numerical solution of (5.13). Finite difference methods are classified into explicit and implicit schemes. Explicit schemes are much easier to solve but typically require certain conditions to be met for numerical stability (Thomas, 1998, Table 5.3.1). For example, for the 2D heat equation, it is required that $\Delta t/(\alpha \Delta x^2) + \Delta t/(\alpha \Delta y^2) \leq 1/2$, where $\alpha$ is the diffusivity constant, $\Delta t$ the time resolution, and $\Delta x, \Delta y$ the spatial resolutions (Thomas, 1998, page 158). Such conditions, which require the spatial resolution to be much finer than the time resolution, may be difficult to establish when $f$ is a black box or when manual selection of the solution grid is not possible. Implicit methods in general are more difficult to solve, but stability is often guaranteed. For example, for the same 2D heat equation problem, the *Crank–Nicolson* scheme (a second order, implicit method) is unconditionally stable (Thomas, 1998, page 159). For these reasons, we considered the *Crank–Nicolson* finite difference method (Crank & Nicolson, 1947) as a classical numerical method that is well-suited to the task at hand. In the implementation of Crank-Nicolson on the inhomogeneous Burger's equation, the nonlinear term is approximated via the use of *lag nonlinear terms* (Thomas, 1998, page 140), as follows:

$$\begin{aligned}\frac{u_j^{i+1}-u_j^i}{\Delta t} =& -\frac{1}{4\Delta x}[u_j^i(u_{j+1}^{i+1}-u_{j-1}^{i+1})+u_j^{i+1}(u_{j+1}^i-u_{j-1}^i)] \\ &+\frac{\alpha}{2}\left(\frac{u_{j+1}^i-2u_j^i+u_{j-1}^i}{\Delta x^2}+\frac{u_{j+1}^{i+1}-2u_j^{i+1}+u_{j-1}^{i+1}}{\Delta x^2}\right) \\ &+\frac{1}{2}(f((i+1)\Delta t,j\Delta x)+f(i\Delta t,j\Delta x))\end{aligned} \tag{5.14}$$

This can be rearranged into:

$$\begin{aligned}&-(r+pu_j^i)u_{j-1}^{i+1}+(2r+1+pu_{j+1}^i-pu_{j-1}^i)u_j^{i+1}+(-r+pu_j^i)u_{j+1}^{i+1} \\ &=ru_{j+1}^i+(1-2r)u_j^i+ru_{j-1}^i+\frac{\Delta t}{2}(f((i+1)\Delta t,j\Delta x)+f(i\Delta t,j\Delta x))\end{aligned} \tag{5.15}$$

where $r=\frac{\alpha\Delta t}{2\Delta x^2}$, $p=\frac{\Delta t}{4\Delta x}$, and can be solved via matrix form:

$$\begin{pmatrix} 2r+1+pu_3^i-pu_1^i & -r+pu_2^i & \dots & \dots & \dots \\ -(r+pu_3^i) & (2r+1+pu_4^i-pu_2^i) & -r+pu_3^i & \dots & \dots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \dots & \dots & -(r+pu_{m-2}^i) & (2r+1+pu_{m-1}^i-pu_{m-3}^i) & -r+pu_{m-2}^i \\ \dots & \dots & \dots & -(r+pu_{m-1}^i) & 2r+1+pu_m^i-pu_{m-2}^i \end{pmatrix} \begin{pmatrix} u_2^{i+1} \\ u_3^{i+1} \\ \vdots \\ u_{m-2}^{i+1} \\ u_{m-1}^{i+1} \end{pmatrix} =$$

$$\begin{pmatrix} (r+pu_2^i)u_1^{i+1}+ru_3^i+(1-2r)u_2^i+ru_1^i+\frac{\Delta t}{2}(f((i+1)\Delta t,2\Delta x)+f(i\Delta t,2\Delta x)) \\ ru_4^i+(1-2r)u_3^i+ru_2^i+\frac{\Delta t}{2}(f((i+1)\Delta t,3\Delta x)+f(i\Delta t,3\Delta x)) \\ \vdots \\ ru_{m-1}^i+(1-2r)u_{m-2}^i+ru_{m-3}^i+\frac{\Delta t}{2}(f((i+1)\Delta t,(m-2)\Delta x)+f(i\Delta t,(m-2)\Delta x)) \\ -(-r+pu_{m-1}^i)u_m^{i+1}+ru_m^i+(1-2r)u_{m-1}^i+ru_{m-2}^i+\frac{\Delta t}{2}(f((i+1)\Delta t,(m-1)\Delta x)+f(i\Delta t,(m-1)\Delta x)) \end{pmatrix}$$

The same regular temporal grid $\mathbf{t}$ and regular spatial grid $\mathbf{x}$ were employed in both Crank–Nicolson and our PNM, so that the computational costs for both methods (as quantified in terms of the number of evaluations of $f$) are identical.

**Results:** The error $E_\infty$ and $Z$-scores for our PNM are displayed in Figure 5.7. The error $E_\infty$ is seen to be gated by the size $m$ of the spatial grid and decreases as $(n,m)$ are simultaneously increased. The $Z$-score values appear to be of order 1 as $(n,m)$ are simultaneously increased, but for increasing $n$ with $m$ fixed the PNM appears to become over-confident; a conservative choice would be to take $m \geq n$, which is also what we concluded from the porous medium equation. These results suggest the output from our PNM is reasonably well-calibrated. Finally, we considered the accuracy of our PNM compared to the Crank–Nicolson benchmark. The error $E_\infty$ for Crank–Nicolson is jointly

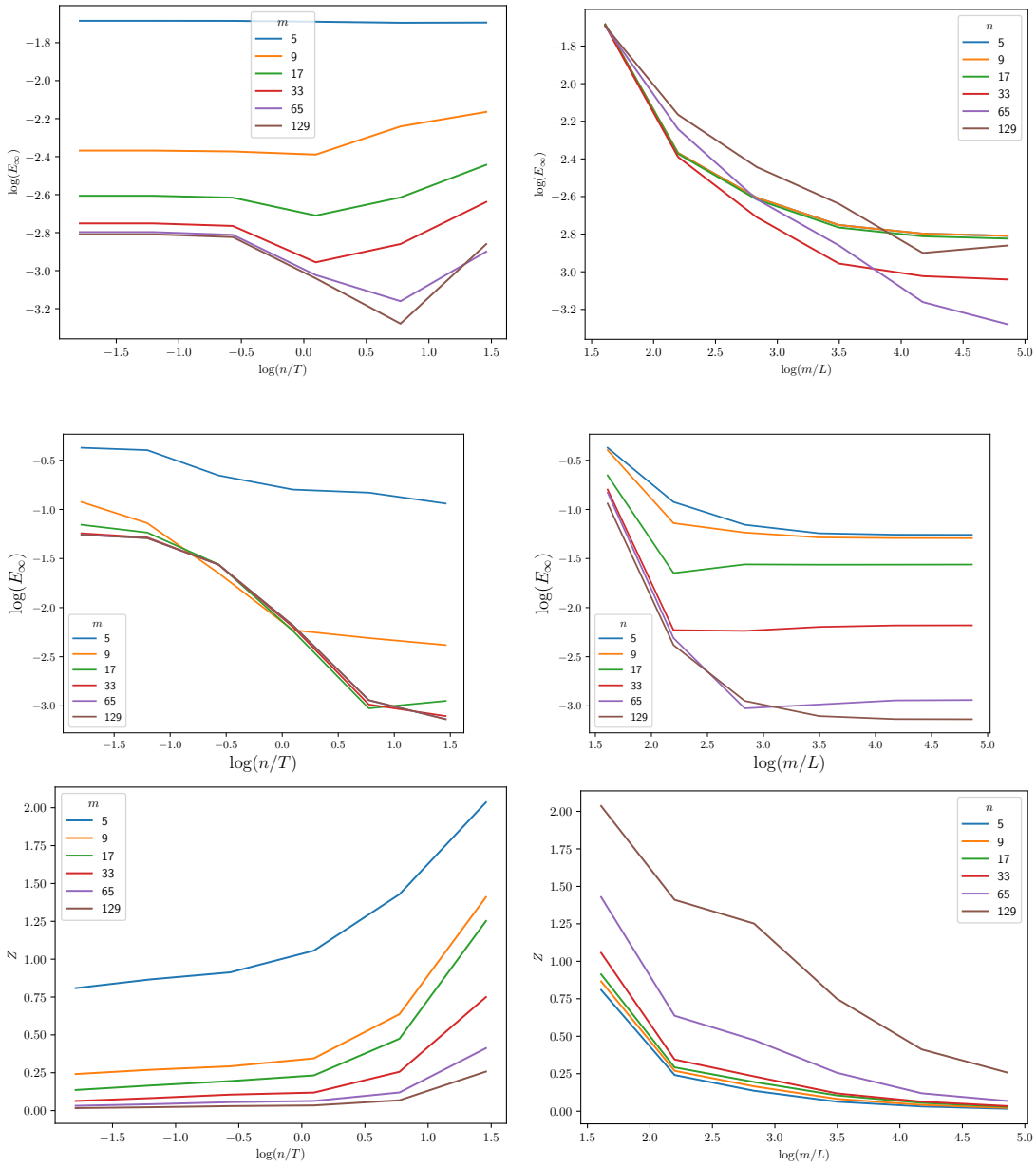



Figure 5.7: Forced Burger's equation: For each pair $(n, m)$ of temporal $(n)$ and spatial $(m)$ grid sizes considered, we plot: (top left) the error $E_\infty$ for fixed $m$ and varying $n$ for our PNM; (top right) the error $E_\infty$ for fixed $n$ and varying $m$ for our PNM; (middle left) the error $E_\infty$ for fixed $m$ and varying $n$, Crank–Nicolson method; (middle right) the error $E_\infty$ for fixed $n$ and varying $m$, Crank–Nicolson method; (bottom left) the $Z$-score for fixed $m$ and varying $n$; (bottom right) the $Z$-score for fixed $n$ and varying $m$.

displayed in Figure 5.7, below the error plots for our PNM, and interestingly, it is generally larger than the error obtained with our PNM. This provides reassurance that our PNM is as accurate as could reasonably be expected.

## 5.5 Conclusion

In this chapter, we presented the first approximate probabilistic numerical method for nonlinear partial differential equations. Our method extends that of Chkrebtii *et al.* (2016) by linearising the nonlinear differential operator. The proposed method also addressed an important and under-studied problem in numerical analysis; the numerical solution of a PDE under severe restrictions on evaluation of the initial, boundary and/or forcing terms $f$, $g$ and $h$ in (5.1). Such restrictions occur when $f$, $g$ and/or $h$ are associated with a computational cost, such as being output from a computationally intensive computer model (Fulton, 2010; Hurrell *et al.*, 2013) or arising as the solution to an auxiliary PDE (MacNamara & Strang, 2016; Cockayne & Duncan, 2020). In many such cases it is not possible to obtain an accurate approximation of the solution of the PDE, and at best one can hope to describe trajectories that are compatible with the limited information available on the PDE. To provide a principled resolution, we cast the numerical solution of a nonlinear PDE as an inference problem within the Bayesian framework and proposed a probabilistic numerical method (PNM) to infer the unknown solution of the PDE. This approach enables formal quantification of *numerical uncertainty*, in such settings where the solution of the PDE cannot be easily approximated.

# Chapter 6

# Conclusion

In this final chapter, we will examine and reflect upon the contributions made in the thesis as well as potential future research directions.

## 6.1 Discussion and Reflection

As discussed in chapter 3, no exact Bayesian PNM had yet been previously proposed in the sense of 1 outside of very specific, linear differential equations. To address this gap, in chapter 4 we presented a proof-of-concept PNM for the numerical solution of ODEs. However, the method is restricted to ODEs which admits a solvable Lie algebra, which is a relatively small class of ODEs. Furthermore, despite the transformed ODE in the canonical coordinates system is of the form where the gradient field no longer depends on the solution, a linear Gaussian Process model could not be deployed as the Implicit Prior Principle 1 is needed to ensure sample solutions in the canonical coordinates space transformed back to well defined functions in the original ODE space. Our proof-of-concept therefore indicates that exact Bayesian inference for ODEs may be extremely difficult. This in turn provided further motivation for the continued development of 'approximate Bayesian' PNMs for the solution of differential equations, which encompasses the majority of the work discussed in chapter 3.

In chapter 5 therefore, we presented the first approximate probabilistic numerical method for nonlinear partial differential equations. Our method provides formal statistical uncertainty quantification for nonlinear PDEs, which is desirable when the initial, boundary or RHS of the PDE are associated with a high computational cost, where it is not possible to obtain an accurate approximation of the solution of the PDE. Our contribution extends an active line of research into the development of PNM for a range of challenging numerical tasks (see the survey in Hennig *et al.*, 2015). A common feature of these tasks is that their difficulty justifies the use of sophisticated statistical machin-

ery, such as Gaussian processes, that themselves may be associated with a computational cost. The PNM developed in chapter 5 has a complexity $O(nm^3)$ to approximate the final state of the PDE, or $O(n^3m^3)$ to approximate the full solution trajectory of the PDE. This renders our PNM computationally intensive – potentially orders of magnitude slower than a classical numerical method – but such increase in cost can be justified when the demands of evaluating $f$, $g$ and $h$ (the right hand side of the PDE as well as the initial and boundary conditions) exceed those of running the PNM (for example, when evaluation of $f$ requires simulation from a climate model Fulton, 2010; Hurrell *et al.*, 2013). In some ways the increased computational cost is unsurprising however, as we are computing not just point estimates of the solution at $nm$ points, but also the posterior covariance on those points as well, which is a matrix of size $(nm, nm)$. The theoretical results developed on the sample path properties of Matérn processes enables the construction of a prior with a pre specified degree of smoothness in the univariate case.

## 6.2 Future work

Further work will be required to establish our approach as a general purpose numerical tool for nonlinear PDEs: First, the non-unique partitioning of the differential operator $D$ into linear and nonlinear components, $P$ and $Q$, together with the non-unique linearisation of $Q$, necessitates some expert input. This is analogous to the selection of a suitable numerical method in the classical setting, but the classical literature has benefitted from decades of research and extensive practical guidance is now available in that context. Here we took a first step to automation by proposing a Matérn tensor product covariance model $\Sigma$, along with presenting a closed-form maximum likelihood estimator for the amplitude of $\Sigma$. The user is left to provide suitable length-scale parameter(s), which is roughly analogous to requiring the user to specify a mesh density in a finite element method (an accepted reality in that context). Second, an extensive empirical assessment will be required to systematically assess the performance of the method; our focus in the present chapter was methodology and theory, providing only an experimental proof-of-concept. In particular, it will be important to assess diagnostics for *failure* of the method; it seems plausible that statistically-motivated diagnostics, such as held-out predictive likelihood, could be used to indicate the quality of the output from the PNM. Thirdly, when applying the PNM on a more realistic example, where the right hand side function $f$ and the initial and boundary conditions $g$ and $h$ are associated with a high computational cost, additional difficulties may arise. One possible difficulty could be if the dimension of the spatial variables $d$ were to significantly increase. Assuming we discretise each spatial coordinate equally with $m$ points, the covariance matrix would then have dimensions $(nm^d, nm^d)$. This potentially makes computing the full posterior distribution very expensive using the

algorithm outlined in Section 5.2.3 for even $d = 3$, without compromising by reducing $m$, though this may still be dwarfed by the cost of evaluating $f$, $g$ or $h$. To mitigate this, one could use a Markovian approximation for the matrix $A_i$, so only a tridiagonal matrix needs to be inverted at each iteration. However, this obviously means the true posterior is not obtained. In general, approximate Gaussian Process methods rely on using some smaller dimension projection of the likelihood to reduce the computational cost (see e.g. Quiñonero-Candela & Rasmussen (2005). Another direction could be to use a Variational Bayes approach, where a distribution that is much easier to compute, and is in some sense close to the true posterior is used instead (e.g. in terms of the Kullback-Leibler divergence).

Finally, we acknowledge that the problem we considered in (5.1) represents only one class of nonlinear PDEs and further work will be required to develop PNM for other classes of PDEs, such as boundary value problems and PDEs defined on more general domains.

## 6.3 Concluding remarks

While the field of probabilistic numerics is still emerging, it is quickly gaining traction and attention, evident from the recent publications in the area. In this thesis we have made novel contributions to PNMs of differential equations by presenting a proof-of-concept exact Bayesian PNM for ODEs, as well as a more practical approximate Bayesian PNM for nonlinear PDEs. While the methods presented are not yet competitive in terms of speed with classical numerical methods in general, which have had hundreds of years of development, we hope that continued research in the field of probabilistic numerics will enable probabilistic numerics to gain increasing practical prominence in science as well as theoretical value in mathematics and statistics.

# Appendix A

# Appendices for Chapter 4

This appendix section contains some additional theoretical results and discussions relevant to chapters 4 and 5 that are not included in the main part of the thesis.

**Example 13** (Deriving the Infinitesimal Generators for the Second Order ODE in Eq. 10)**.** Consider the second order nonlinear ODE

$$(x - \mathrm{y}(x))\frac{\mathrm{d}^2\mathrm{y}}{\mathrm{d}x^2} + 2\frac{\mathrm{dy}}{\mathrm{d}x}\left(\frac{\mathrm{dy}}{\mathrm{d}x} + 1\right) + \left(\frac{\mathrm{dy}}{\mathrm{d}x}\right)^{3/2} = 0. \tag{A.1}$$

Using Corollary 4.4, we have:

$$\left(\xi\frac{\partial}{\partial x} + \eta\frac{\partial}{\partial y} + \eta^{(1)}\frac{\partial}{\partial y_1} + \eta^{(2)}\frac{\partial}{\partial y_2}\right)\left(y_2 + \frac{2y_1(y_1+1) + y_1^{3/2}}{x-y}\right) = 0$$

which implies

$$-\xi\frac{2y_1(y_1+1) + y_1^{3/2}}{(x-y)^2} + \eta\frac{2y_1(y_1+1) + y_1^{3/2}}{(x-y)^2} + \eta^{(1)}\left(\frac{4y_1 + 2 + \frac{3}{2}y_1^{1/2}}{x-y}\right) + \eta^{(2)} = 0$$

Recall

$$\eta^{(1)} = \eta_x + (\eta_y - \xi_x)y_1 + \xi_y y_1^2$$

and

$$\eta^{(2)} = \eta_{xx} + (2\eta_{xy} - \xi_{xx})y_1 + (\eta_{yy} - 2\xi_{xy})y_1^2 - \xi_{yy}y_1^3 + (\eta_y - 2\xi_x)y_2 - 2\xi_y y_1 y_2$$

Also notice we can replace $y_2$ via the original differential equation, i.e.

$$y_2 = -\frac{2y_1(y_1+1) + y_1^{3/2}}{x-y}.$$

Substituting for $\eta^{(1)}$, $\eta^{(2)}$ and $y_2$ via the above expressions, multiplying both sides by $(x-y)^2$ and rearranging the terms as powers of $y_1$ yields the rather long equation:

$$\begin{aligned}
&(2x\eta_x - 2y\eta_x + x^2\eta_{xx} - 2xy\eta_{xx} + y^2\eta_{xx}) \\
+y_1&\begin{pmatrix} -2\xi + 2\eta + 4x\eta_x - 4y\eta_x + 2x\eta_y - 2y\eta_y - 2x\xi_x + 2y\xi_x \\ +2x^2\eta_{xy} - 2xy\eta_{xy} + 2y^2\eta_{xy} - x^2\xi_{xx} + 2xy\xi_{xx} \\ -y^2\xi_{xx} - 2x\eta_y + 4x\xi_x + 2y\eta_y - 4y\xi_x \end{pmatrix} \\
+y_1^2&\begin{pmatrix} -2\xi + 2\eta + 4x\eta_y - 4y\eta_y - 4x\xi_x + 4y\xi_x + 2x\xi_y - 2y\xi_y \\ +x^2\eta_{yy} - 2xy\eta_{yy} + y^2\eta_{yy} - 2x^2\xi_{xy} + 4xy\xi_{xy} \\ -2y^2\xi_{xy} - 2x\eta_y + 4x\xi_x + 2y\eta_y - 4y\xi_x + 4x\xi_y - 4y\xi_y \end{pmatrix} \\
&+y_1^3(4x\xi_y - 4y\xi_y - x^2\xi_{yy} + 2xy\xi_{yy} - y^2\xi_{yy} + 4x\xi_y - 4y\xi_y) \\
&+y_1^{1/2}\left(\frac{3}{2}x\eta_x - \frac{3}{2}y\eta_x\right) \\
+y_1^{3/2}&\left(-\xi + \eta + \frac{3}{2}x\eta_y - \frac{3}{2}y\eta_y - \frac{3}{2}x\xi_x + \frac{3}{2}y\xi_x - x\eta_y + 2x\xi_x + y\eta_y - 2y\xi_x\right) \\
&+y_1^{5/2}\left(\frac{3}{2}x\xi_y - \frac{3}{2}y\xi_y + 2x\xi_y - 2y\xi_y\right) = 0
\end{aligned}$$

This expression on the left hand side must vanish, so comparing the coefficients of powers of $y_1$ gives the determining equations:

$$\begin{aligned}
2x\eta_x - 2y\eta_x + x^2\eta_{xx} - 2xy\eta_{xx} + y^2\eta_{xx} &= 0 \quad \text{(A.2)} \\
-2\xi + 2\eta + 4x\eta_x - 4y\eta_x + 2x\eta_y - 2y\eta_y - 2x\xi_x + 2y\xi_x & \\
+2x^2\eta_{xy} - 2xy\eta_{xy} + 2y^2\eta_{xy} - x^2\xi_{xx} + 2xy\xi_{xx} & \\
-y^2\xi_{xx} - 2x\eta_y + 4x\xi_x + 2y\eta_y - 4y\xi_x &= 0 \\
-2\xi + 2\eta + 4x\eta_y - 4y\eta_y - 4x\xi_x + 4y\xi_x + 2x\xi_y - 2y\xi_y + x^2\eta_{yy} & \\
-2xy\eta_{yy} + y^2\eta_{yy} - 2x^2\xi_{xy} + 4xy\xi_{xy} - 2y^2\xi_{xy} & \\
-2x\eta_y + 4x\xi_x + 2y\eta_y - 4y\xi_x + 4x\xi_y - 4y\xi_y &= 0 \\
4x\xi_y - 4y\xi_y - x^2\xi_{yy} + 2xy\xi_{yy} - y^2\xi_{yy} + 4x\xi_y - 4y\xi_y &= 0 \quad \text{(A.3)} \\
\frac{3}{2}x\eta_x - \frac{3}{2}y\eta_x &= 0 \quad \text{(A.4)} \\
-\xi + \eta + \frac{3}{2}x\eta_y - \frac{3}{2}y\eta_y - \frac{3}{2}x\xi_x + \frac{3}{2}y\xi_x - x\eta_y + 2x\xi_x + y\eta_y - 2y\xi_x &= 0 \\
\frac{3}{2}x\xi_y - \frac{3}{2}y\xi_y + 2x\xi_y - 2y\xi_y &= 0 \quad \text{(A.5)}
\end{aligned}$$

It is immediately obvious from (A.4) that $\eta_x = 0$ and from (A.5) that $\xi_y = 0$. Consequently

(A.2) and (A.3) vanishes. The remaining determining equations simplify to:

$$-2\xi + 2\eta + 2x\xi_x - 2y\xi_x - x^2\xi_{xx} + 2xy\xi_{xx} - y^2\xi_{xx} = 0 \quad (A.6)$$

$$-2\xi + 2\eta + 2x\eta_y - 2y\eta_y + x^2\eta_{yy} - 2xy\eta_{yy} + y^2\eta_{yy} = 0 \quad (A.7)$$

$$-\xi + \eta + \frac{1}{2}x\eta_y - \frac{1}{2}y\eta_y + \frac{1}{2}x\xi_x - \frac{1}{2}y\xi_x = 0 \quad (A.8)$$

These remaining partial differential equations in $\xi(x, y)$ and $\eta(x, y)$ are linear, and recall $\eta(x, y)$ is independent of $x$ and $\xi(x, y)$ is independent of $y$ respectively. To solve these partial differential equations we can therefore express $\xi(x) = \sum_{n=0}^{\infty} a_n x^n$ and $\eta(y) = \sum_{m=0}^{\infty} b_m y^m$. Consequently (A.6) becomes:

$$-2\sum_{n=0}^{\infty} a_n x^n + 2\sum_{m=0}^{\infty} b_m y^m + 2\sum_{n=1}^{\infty} n a_n x^n - 2y\sum_{n=1}^{\infty} n a_n x^{n-1}$$
$$-\sum_{n=2}^{\infty} n(n-1)a_n x^n + 2y\sum_{n=2}^{\infty} n(n-1)a_n x^{n-1} - y^2\sum_{n=2}^{\infty} n(n-1)a_n x^{n-2} = 0$$

Comparing the constant term implies $b_0 = a_0$. Comparing the terms containing $y$ implies:

$$2b_1 - 2\sum_{n=1}^{\infty} n a_n x^{n-1} + 2\sum_{n=2}^{\infty} n(n-1)a_n x^{n-1} = 0$$

Comparing coefficients of $x^n$ gives $b_1 = a_1$, $n = n(n-1)$ or $a_n = 0$ for $n \geq 2$. Of course, $n = n(n-1)$ has solution $n = 2$ for $n \geq 2$, so $a_n = 0$ for $n \geq 3$. Comparing the terms containing $y^2$ implies $b_2 = a_2$. Notice (A.7) is symmetric with (A.6) in the sense that swapping $\xi$ with $\eta$ and $x$ with $y$ in (A.6) gives (A.7). So by symmetry (A.7) gives $b_0 = a_0$, $b_1 = a_1$, $b_2 = a_2$ and $b_n = 0$ for $n \geq 3$. (A.8) gives no additional solutions. Therefore, the example ODE admits a three parameter Lie group of transformations with infinitesimals:

$$\xi = a_0 + a_1 x + a_2 x^2$$

$$\eta = a_0 + a_1 y + a_2 y^2$$

where $a_2$, $a_1$ and $a_0$ are arbitrary constants. The infinitesimal generators corresponding to $a_2$, $a_1$ and $a_0$ are respectively

$$\mathrm{X}_1 = x^2\frac{\partial}{\partial x} + y^2\frac{\partial}{\partial y} \quad (A.9)$$

$$\mathrm{X}_2 = x\frac{\partial}{\partial x} + y\frac{\partial}{\partial y}$$

$$\mathrm{X}_3 = \frac{\partial}{\partial x} + \frac{\partial}{\partial y}, \quad (A.10)$$

which generate a three dimensional Lie algebra.

**Example 14** (Ex. 13, continued)**.** Recall from Ex. 13 that the second order nonlinear ODE in Eq. (A.1) admits a three parameter Lie group of transformations with infinitesimal generators $\mathrm{X}_1$, $\mathrm{X}_2$, $\mathrm{X}_3$ defined in Eqs. (A.9)-(A.10). These generators can be verified to satisfy $[\mathrm{X}_1, \mathrm{X}_2] = -\mathrm{X}_1$, $[\mathrm{X}_1, \mathrm{X}_3] = -\mathrm{X}_2$, $[\mathrm{X}_2, \mathrm{X}_3] = -\mathrm{X}_3$. The pairs $\mathrm{X}_1, \mathrm{X}_2$ and $\mathrm{X}_2, \mathrm{X}_3$ form a two dimensional (and therefore solvable by Thm. 4.6) Lie sub-algebra and can be used as the basis for our method. For the derivations below we proceed with arbitrary choice $\mathrm{X}_1$, $\mathrm{X}_2$.

Following the proof of Thm. 4.8, first we seek a solution $v = v(x, y)$ to the first order linear PDE $\mathrm{X}_1 v = 0$. i.e. we must solve

$$x^2 \frac{\partial v}{\partial x} + y^2 \frac{\partial v}{\partial y} = 0$$

This has general solution $v = f(\frac{1}{y} - \frac{1}{x})$ for some arbitrary function $f$, and we pick a particular solution $v(x, y) = \frac{1}{y} - \frac{1}{x}$. Next we seek a solution $w = w(x, y, y_1)$ to the first order linear PDE $\mathrm{X}_1^{(1)} w = 0$. i.e. we must solve

$$x^2 \frac{\partial w}{\partial x} + y^2 \frac{\partial w}{\partial y} + 2(y - x) y_1 \frac{\partial w}{\partial y_1} = 0.$$

Again, we pick a particular solution $w(x, y, y_1) = y_1 (\frac{x}{y})^2$. In accordance with Eq. (4.15), we can re-write the original ODE (A.1) in terms of the coordinates $v$ and $w$ to obtain

$$\frac{\mathrm{d}w}{\mathrm{d}v} = \begin{cases} \dfrac{w^{3/2} + 2w(w+1)}{v(w-1)}, & \text{for } \frac{x}{y} \geq 0 \\ \dfrac{-w^{3/2} + 2w(w+1)}{v(w-1)}, & \text{for } \frac{x}{y} < 0 \end{cases} \tag{A.11}$$

Next we express $\mathrm{X}_2^{(1)}$ in terms of $v$ and $w$ find its canonical coordinates $\tilde{r}(v, w)$, $\tilde{s}(v, w)$. To this end, we have $\mathrm{X}_2^{(1)} = x \frac{\partial}{\partial x} + y \frac{\partial}{\partial y} = -v \frac{\partial}{\partial v}$, which has canonical coordinates $\tilde{r}(v, w) = w$, $\tilde{s}(v, w) = -\log(v)$. Re-writing Eq. (A.11) in terms of $\tilde{r}$, $\tilde{s}$ leads to the analogue of Eq. (4.16):

$$\frac{\mathrm{d}\tilde{\mathrm{s}}}{\mathrm{d}\tilde{r}} = \frac{1 - \tilde{r}}{\pm \tilde{r}^{3/2} + 2\tilde{r}(\tilde{r} + 1)} =: H(\tilde{r}) \tag{A.12}$$

This example exhibits the convenient feature that Eq. (A.12) can be directly integrated to give $\tilde{\mathrm{s}}(\tilde{r}) = -\log(2\tilde{r} \pm \sqrt{\tilde{r}} + 2) + \log(\tilde{r})/2 + C$, which can be re-written in terms of $x, y, y_1$

to give

$$\log\left(\frac{1}{y}-\frac{1}{x}\right) = \log\left(2\sqrt{y_1\frac{x^2}{y^2}} \pm 1 + \frac{2}{\sqrt{y_1\frac{x^2}{y^2}}}\right) - C \tag{A.13}$$

for some integration constant $C$.

The final step, to remove the $y_1$ independence, requires canonical coordinates for $\mathrm{X}_1$. These can be selected as $r(x,y) = \frac{1}{y} - \frac{1}{x}$, $s(x,y) = -\frac{1}{y}$. Then Eq. (A.13) becomes

$$\frac{r \mp \exp(-C)}{2\exp(-C)} = \left(\frac{\frac{\mathrm{d}s}{\mathrm{d}r}}{1+\frac{\mathrm{d}s}{\mathrm{d}r}}\right)^{1/2} + \left(\frac{1+\frac{\mathrm{d}s}{\mathrm{d}r}}{\frac{\mathrm{d}s}{\mathrm{d}r}}\right)^{1/2}$$

which is equivalent to Eq. (4.11) for some function $G$.

## A.1 Design of the Training Set

The performance of the proposed Bayesian PNM is not our main focus in this work, as we consider the method to be (only) a proof-of-concept. However, for completeness we acknowledge that performance will depend on the locations at which the gradient field is evaluated; the so-called *training set*. In this section we discuss how these inputs could be optimally selected. To simplify the presentation, we focus on the case of a first order ODE, as in Eq. (4.7), where the inputs $r_0, \ldots, r_n$ must be selected.

The design of a PNM can be viewed as an instance of statistical experimental design (Chaloner & Verdinelli, 1995). In Sec. 3 of Cockayne *et al.* (2019) a connection between PNM and decision-theoretic experimental design was exposed. Such methods require that a loss function $L : \mathcal{Q} \times \mathcal{Q} \to \mathbb{R}$ is provided, where $L(q, q^\dagger)$ quantifies the loss when $q$ is used as an estimate for the true quantity of interest $q^\dagger$. Further detail was provided in Oates *et al.* (2019*b*). To avoid repetition, in the remainder we focus instead on approximate experimental design, where a loss function is not explicitly needed.

Recall that the output of a PNM is the distribution $\mu_n = B(\mu, a^n) \in \mathcal{P}_\mathcal{Q}$. Then one can specify a functional $\ell : \mathcal{P}_\mathcal{Q} \to \mathbb{R}$ and compute

$$\tau(r_0, \ldots, r_n) \;=\; \int \ell(B(\mu, A(\mathrm{y}; r_0, \ldots, r_n)))\mathrm{d}\mu(\mathrm{y}) \tag{A.14}$$

where $A(\cdot; r_0, \ldots, r_n)$ is the information operator in Eq. (4.9) with the dependence on $r_0, \ldots, r_n$ made explicit. For the choice $\ell(\nu) = \log\det(\mathrm{Cov}_{\tilde{Q}\sim\nu}[\tilde{Q}])$, a configuration $(r_0, \ldots, r_n)$ for which $\tau(r_0, \ldots, r_n)$ is minimised is said to be *D-optimal*. The functional $\ell$ plays the role of an approximation to posterior expected loss, and other choices for $\ell$ lead to other approximate notions of optimal experimental design. For instance, an

*A-optimal* design was used for the Bayesian solution of a partial differential equation in Cockayne *et al.* (2016). For further background on experimental design we refer the reader to Chaloner & Verdinelli (1995).

Importantly, Eq. (A.14) does not depend on the information $A(\mathrm{y}^\dagger)$ and can therefore be evaluated prior to the experiment being performed. However, in general the numerical approximation of Eq. (A.14), and the task of finding a minimal configuration, is practically difficult. The reader is referred to Overstall *et al.* (2019) for further discussion of experimental design in the PNM context.

# Appendix B

# Appendices for Chapter 5

## B.1 Proof of Lemma 5.1

*Proof of Lemma 5.1.* Our starting point is the stochastic process defined in Lemma 5.1, and from this we aim to derive the iterative formulation in the main text. The derivation requires several items of notation to be introduced. First, let

$$L^{i+1}(r) := \begin{bmatrix} \Sigma(r, \boldsymbol{a}_0) \\ \Sigma(r, \boldsymbol{b}_0) \\ \Sigma_{\bar{D}_0}(r, \boldsymbol{v}_0) \\ \vdots \\ \Sigma(r, \boldsymbol{b}_i) \\ \Sigma_{\bar{D}_i}(r, \boldsymbol{v}_i) \end{bmatrix}^\top, \quad R^{i+1}(r') := \begin{bmatrix} \Sigma(\boldsymbol{a}_0, r') \\ \Sigma(\boldsymbol{b}_0, r') \\ \Sigma_{D_0}(\boldsymbol{v}_0, r') \\ \vdots \\ \Sigma(\boldsymbol{b}_i, r') \\ \Sigma_{D_i}(\boldsymbol{v}_i, r') \end{bmatrix}, \quad F_{i+1} := \begin{bmatrix} g(\boldsymbol{a}_0) - \mu(\boldsymbol{a}_0) \\ h(\boldsymbol{b}_0) - \mu(\boldsymbol{b}_0) \\ f(\boldsymbol{v}_0) - \mu_{D_0}(\boldsymbol{v}_0) \\ \vdots \\ h(\boldsymbol{b}_i) - \mu(\boldsymbol{b}_i) \\ f(\boldsymbol{v}_i) - \mu_{D_i}(\boldsymbol{v}_i) \end{bmatrix}$$

and

$$M_{i+1} := \begin{bmatrix} \Sigma(\boldsymbol{a}_0, \boldsymbol{a}_0) & \Sigma(\boldsymbol{a}_0, \boldsymbol{b}_0) & \Sigma_{\bar{D}_0}(\boldsymbol{a}_0, \boldsymbol{v}_0) & \dots & \Sigma(\boldsymbol{a}_0, \boldsymbol{b}_i) & \Sigma_{\bar{D}_i}(\boldsymbol{a}_0, \boldsymbol{v}_i) \\ \Sigma(\boldsymbol{b}_0, \boldsymbol{a}_0) & \Sigma(\boldsymbol{b}_0, \boldsymbol{b}_0) & \Sigma_{\bar{D}_0}(\boldsymbol{b}_0, \boldsymbol{v}_0) & \dots & \Sigma(\boldsymbol{b}_0, \boldsymbol{b}_i) & \Sigma_{\bar{D}_i}(\boldsymbol{b}_0, \boldsymbol{v}_i) \\ \Sigma_{D_0}(\boldsymbol{v}_0, \boldsymbol{a}_0) & \Sigma_{D_0}(\boldsymbol{v}_0, \boldsymbol{b}_0) & \Sigma_{D_0\bar{D}_0}(\boldsymbol{v}_0, \boldsymbol{v}_0) & \dots & \Sigma_{D_i}(\boldsymbol{v}_0, \boldsymbol{b}_i) & \Sigma_{D_i\bar{D}_i}(\boldsymbol{v}_0, \boldsymbol{v}_i) \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ \Sigma(\boldsymbol{b}_i, \boldsymbol{a}_0) & \Sigma(\boldsymbol{b}_i, \boldsymbol{b}_0) & \Sigma_{\bar{D}_i}(\boldsymbol{b}_i, \boldsymbol{v}_0) & \dots & \Sigma(\boldsymbol{b}_i, \boldsymbol{b}_i) & \Sigma_{\bar{D}_i}(\boldsymbol{b}_i, \boldsymbol{v}_i) \\ \Sigma_{D_i}(\boldsymbol{v}_i, \boldsymbol{a}_0) & \Sigma_{D_i}(\boldsymbol{v}_i, \boldsymbol{b}_0) & \Sigma_{D_i\bar{D}_i}(\boldsymbol{v}_i, \boldsymbol{v}_0) & \dots & \Sigma_{D_i}(\boldsymbol{v}_i, \boldsymbol{b}_i) & \Sigma_{D_i\bar{D}_i}(\boldsymbol{v}_i, \boldsymbol{v}_i) \end{bmatrix}.$$

The mean and covariance of $U^{i+1}$, as defined in Lemma 5.1, are equal to

$$\mu^{i+1}(r) = \mu(r) + L^{i+1}(r)(M_{i+1})^{-1}F_{i+1}$$
$$\Sigma^{i+1}(r, r') = \Sigma(r, r') - L^{i+1}(r)(M_{i+1})^{-1}R^{i+1}(r')$$

Similarly, introduce the notation

$$L_{D_i}^{i+1}(r) := \begin{bmatrix} \Sigma_{D_i}(r, \boldsymbol{a}_0) \\ \Sigma_{D_i}(r, \boldsymbol{b}_0) \\ \Sigma_{D_i\bar{D}_0}(r, \boldsymbol{v}_0) \\ \vdots \\ \Sigma_{D_i}(r, \boldsymbol{b}_i) \\ \Sigma_{D_i\bar{D}_i}(r, \boldsymbol{v}_i) \end{bmatrix}^\top, \qquad R_{D_i}^{i+1}(r') := \begin{bmatrix} \Sigma_{\bar{D}_i}(\boldsymbol{a}_0, r') \\ \Sigma_{\bar{D}_i}(\boldsymbol{b}_0, r') \\ \Sigma_{D_0\bar{D}_i}(\boldsymbol{v}_0, r') \\ \vdots \\ \Sigma_{\bar{D}_i}(\boldsymbol{b}_i, r') \\ \Sigma_{D_i\bar{D}_i}(\boldsymbol{v}_i, r') \end{bmatrix},$$

so that the application of $D_i$ to $\mu^{i+1}$ and $\Sigma^{i+1}$ may be expressed as

$$\begin{aligned} \mu_{D_i}^{i+1}(r) &= \mu_{D_i}(r) + L_{D_i}^{i+1}(r)(M_{i+1})^{-1}F_{i+1} \\ \Sigma_{D_i}^{i+1}(r, r') &= \Sigma_{D_i}(r, r') - L_{D_i}^{i+1}(r)(M_{i+1})^{-1}R^{i+1}(r') \\ \Sigma_{\bar{D}_i}^{i+1}(r, r') &= \Sigma_{\bar{D}_i}(r, r') - L^{i+1}(r)(M_{i+1})^{-1}R_{D_i}^{i+1}(r') \\ \Sigma_{D_i\bar{D}_i}^{i+1}(r, r') &= \Sigma_{D_i\bar{D}_i}(r, r') - L_{D_i}^{i+1}(r)(M_{i+1})^{-1}R_{D_i}^{i+1}(r') \end{aligned}$$

Notice that we have a recursive partitioning of $M_{i+1}$ into blocks of the form

$$M_{i+1} = \begin{bmatrix} M_i & \beta_i \\ \gamma_i & \delta_i \end{bmatrix}$$

where

$$\beta_i := \begin{bmatrix} R^i(\boldsymbol{b}_i) & R_{D_i}^i(\boldsymbol{v}_i) \end{bmatrix}, \quad \gamma_i := \begin{bmatrix} L^i(\boldsymbol{b}_i) \\ L_{D_i}^i(\boldsymbol{v}_i) \end{bmatrix}, \quad \delta_i := \begin{bmatrix} \Sigma(\boldsymbol{b}_i, \boldsymbol{b}_i) & \Sigma_{\bar{D}_i}(\boldsymbol{b}_i, \boldsymbol{v}_i) \\ \Sigma_{D_i}(\boldsymbol{v}_i, \boldsymbol{b}_i) & \Sigma_{D_i\bar{D}_i}(\boldsymbol{v}_i, \boldsymbol{v}_i) \end{bmatrix}.$$

Thus we may use the block matrix inversion formula to deduce that

$$M_{i+1}^{-1} = \begin{bmatrix} M_i^{-1}(I + \beta_i(\delta_i - \gamma_i M_i^{-1}\beta_i)^{-1}\gamma_i M_i^{-1}) & -M_i^{-1}\beta_i(\delta_i - \gamma_i M_i^{-1}\beta_i)^{-1} \\ -(\delta_i - \gamma_i M_i^{-1}\beta_i)^{-1}\gamma_i M_i^{-1} & (\delta_i - \gamma_i M_i^{-1}\beta_i)^{-1} \end{bmatrix}$$

Setting $A_i := \delta_i - \gamma_i M_i^{-1}\beta_i$, we observe that

$$A_i = \begin{bmatrix} \Sigma^i(\boldsymbol{b}_i, \boldsymbol{b}_i) & \Sigma_{\bar{D}_i}^i(\boldsymbol{b}_i, \boldsymbol{v}_i) \\ \Sigma_{D_i}^i(\boldsymbol{v}_i, \boldsymbol{b}_i) & \Sigma_{D_i\bar{D}_i}^i(\boldsymbol{v}_i, \boldsymbol{v}_i) \end{bmatrix},$$

so our definition of $A_i$ coincides with that in the main text, and enables us to simplify $M_{i+1}^{-1}$ into

$$M_{i+1}^{-1} = \begin{bmatrix} M_i^{-1}(I + \beta_i A_i^{-1}\gamma_i M_i^{-1}) & -M_i^{-1}\beta_i A_i^{-1} \\ -A_i^{-1}\gamma_i M_i^{-1} & A_i^{-1} \end{bmatrix}.$$

Therefore we have that

$$\begin{aligned}
\mu^{i+1}(r) &= \mu(r) + L^{i+1}(r)(M_{i+1})^{-1}F_{i+1} \\
&= \mu(r) + L^i(r)M_i^{-1}F_i + L^i(r)M_i^{-1}\beta_i A_i^{-1}\gamma_i M_i^{-1}F_i - [\Sigma(r,\boldsymbol{b}_i), \Sigma_{\bar{D}_i}(r,\boldsymbol{v}_i)]A_i^{-1}\gamma_i M_i^{-1}F_i \\
&\quad - L^i(r)M_i^{-1}\beta_i A_i^{-1}\begin{bmatrix} h(\boldsymbol{b}_i) - \mu(\boldsymbol{b}_i) \\ f(\boldsymbol{v}_i) - \mu_{D_i}(\boldsymbol{v}_i)\end{bmatrix} + [\Sigma(r,\boldsymbol{b}_i), \Sigma_{\bar{D}_i}(r,\boldsymbol{v}_i)]A_i^{-1}\begin{bmatrix} h(\boldsymbol{b}_i) - \mu(\boldsymbol{b}_i) \\ f(\boldsymbol{v}_i) - \mu_{D_i}(\boldsymbol{v}_i)\end{bmatrix} \\
&= \mu^i(r) + \{L^i(r)M_i^{-1}\beta_i - [\Sigma(r,\boldsymbol{b}_i), \Sigma_{\bar{D}_i}(r,\boldsymbol{v}_i)]\}A_i^{-1}\gamma_i M_i^{-1}F_i \\
&\quad + \{[\Sigma(r,\boldsymbol{b}_i), \Sigma_{\bar{D}_i}(r,\boldsymbol{v}_i)] - L^i(r)M_i^{-1}\beta_i\}A_i^{-1}\begin{bmatrix} h(\boldsymbol{b}_i) - \mu(\boldsymbol{b}_i) \\ f(\boldsymbol{v}_i) - \mu_{D_i}(\boldsymbol{v}_i)\end{bmatrix}
\end{aligned}$$

which can be simplified by noting that

$$\begin{aligned}
L^i(r)M_i^{-1}\beta_i - [\Sigma(r,\boldsymbol{b}_i), \Sigma_{\bar{D}_i}(r,\boldsymbol{v}_i)] &= L^i(r)M_i^{-1}\begin{bmatrix} R^i(\boldsymbol{b}_i) & R^i_{D_i}(\boldsymbol{v}_i)\end{bmatrix} - [\Sigma(r,\boldsymbol{b}_i), \Sigma_{\bar{D}_i}(r,\boldsymbol{v}_i)] \\
&= [-\Sigma^i(r,\boldsymbol{b}_i), -\Sigma^i_{\bar{D}_i}(r,\boldsymbol{v}_i)] \\
\gamma_i M_i^{-1}F_i = \begin{bmatrix} L^i(\boldsymbol{b}_i) \\ L^i_{D_i}(\boldsymbol{v}_i)\end{bmatrix} M_i^{-1}F_i &= \begin{bmatrix} \mu^i(\boldsymbol{b}_i) - \mu(\boldsymbol{b}_i) \\ \mu^i_{D_i}(\boldsymbol{v}_i) - \mu_{D_i}(\boldsymbol{v}_i)\end{bmatrix}
\end{aligned}$$

to produce the iterative formulation for the mean in the main text:

$$\begin{aligned}
\mu^{i+1}(r) &= \mu^i(r) + [\Sigma^i(r,\boldsymbol{b}_i), \Sigma^i_{\bar{D}_i}(r,\boldsymbol{v}_i)]A_i^{-1}\left(\begin{bmatrix} h(\boldsymbol{b}_i) - \mu(\boldsymbol{b}_i) \\ f(\boldsymbol{v}_i) - \mu_{D_i}(\boldsymbol{v}_i)\end{bmatrix} - \begin{bmatrix} \mu^i(\boldsymbol{b}_i) - \mu(\boldsymbol{b}_i) \\ \mu^i_{D_i}(\boldsymbol{v}_i) - \mu_{D_i}(\boldsymbol{v}_i)\end{bmatrix}\right) \\
&= \mu^i(r) + [\Sigma^i(r,\boldsymbol{b}_i), \Sigma^i_{\bar{D}_i}(r,\boldsymbol{v}_i)]A_i^{-1}\begin{bmatrix} h(\boldsymbol{b}_i) - \mu^i(\boldsymbol{b}_i) \\ f(\boldsymbol{v}_i) - \mu^i_{D_i}(\boldsymbol{v}_i)\end{bmatrix}
\end{aligned}$$

For the covariance we have

$$\begin{aligned}
\Sigma^{i+1}(r,r') &= \Sigma(r,r') - L^{i+1}(r)(M_{i+1})^{-1}R^{i+1}(r') \\
&= \Sigma(r,r') - L^i(r)M_i^{-1}R^i(r') - L^i(r)M_i^{-1}\beta_i A_i^{-1}\gamma_i M_i^{-1}R^i(r') \\
&\quad + [\Sigma(r,\boldsymbol{b}_i), \Sigma_{\bar{D}_i}(r,\boldsymbol{v}_i)]A_i^{-1}\gamma_i M_i^{-1}R^i(r') \\
&\quad + L^i(r)M_i^{-1}\beta_i A_i^{-1}\begin{bmatrix} \Sigma(\boldsymbol{b}_i,r') \\ \Sigma_{D_i}(\boldsymbol{v}_i,r')\end{bmatrix} - [\Sigma(r,\boldsymbol{b}_i), \Sigma_{\bar{D}_i}(r,\boldsymbol{v}_i)]A_i^{-1}\begin{bmatrix} \Sigma(\boldsymbol{b}_i,r') \\ \Sigma_{D_i}(\boldsymbol{v}_i,r')\end{bmatrix} \\
&= \Sigma^i(r,r') - \{L^i(r)M_i^{-1}\beta_i - [\Sigma(r,\boldsymbol{b}_i), \Sigma_{\bar{D}_i}(r,\boldsymbol{v}_i)]\}A_i^{-1}\gamma_i M_i^{-1}R^i(r') \\
&\quad - \{[\Sigma(r,\boldsymbol{b}_i), \Sigma_{\bar{D}_i}(r,\boldsymbol{v}_i)] - L^i(r)M_i^{-1}\beta_i\}A_i^{-1}\begin{bmatrix} \Sigma(\boldsymbol{b}_i,r') \\ \Sigma_{D_i}(\boldsymbol{v}_i,r')\end{bmatrix}.
\end{aligned}$$

This can be simplified by noting that

$$\begin{aligned} L^i(r)M_i^{-1}\beta_i - [\Sigma(r,\boldsymbol{b}_i), \Sigma_{\bar{D}_i}(r,\boldsymbol{v}_i)] &= L^i(r)M_i^{-1}\begin{bmatrix} R^i(\boldsymbol{b}_i) & R^i_{D_i}(\boldsymbol{v}_i)\end{bmatrix} - [\Sigma(r,\boldsymbol{b}_i), \Sigma_{\bar{D}_i}(r,\boldsymbol{v}_i)] \\ &= [-\Sigma^i(r,\boldsymbol{b}_i), -\Sigma^i_{\bar{D}_i}(r,\boldsymbol{v}_i)] \\ \gamma_i M_i^{-1}R^i(r') = \begin{bmatrix} L^i(\boldsymbol{b}_i) \\ L^i_{D_i}(\boldsymbol{v}_i)\end{bmatrix} M_i^{-1}R^i(r') &= \begin{bmatrix} -\Sigma^i(\boldsymbol{b}_i,r') + \Sigma(\boldsymbol{b}_i,r') \\ -\Sigma^i_{D_i}(\boldsymbol{v}_i,r') + \Sigma_{D_i}(\boldsymbol{v}_i,r')\end{bmatrix} \end{aligned}$$

to obtain

$$\begin{aligned} \Sigma^{i+1}(r,r') &= \Sigma^i(r,r') - [\Sigma^i(r,\boldsymbol{b}_i), \Sigma^i_{\bar{D}_i}(r,\boldsymbol{v}_i)]A_i^{-1}\left(\begin{bmatrix} \Sigma^i(\boldsymbol{b}_i,r') - \Sigma(\boldsymbol{b}_i,r') \\ \Sigma^i_{D_i}(\boldsymbol{v}_i,r') - \Sigma_{D_i}(\boldsymbol{v}_i,r')\end{bmatrix} + \begin{bmatrix} \Sigma(\boldsymbol{b}_i,r') \\ \Sigma_{D_i}(\boldsymbol{v}_i,r')\end{bmatrix}\right) \\ &= \Sigma^i(r,r') - [\Sigma^i(r,\boldsymbol{b}_i), \Sigma^i_{\bar{D}_i}(r,\boldsymbol{v}_i)]A_i^{-1}\begin{bmatrix} \Sigma^i(\boldsymbol{b}_i,r') \\ \Sigma^i_{D_i}(\boldsymbol{v}_i,r')\end{bmatrix}, \end{aligned}$$

identical to the iterative formulation in the main text. □

# Bibliography


Abdulle, A. & Garegnani, G. 2018 Random time step probabilistic methods for uncertainty quantification in chaotic and geometric numerical integration. *arXiv:1801.01340* .

Barenblatt, G. 1952 On some unsteady motions of a liquid or gas in a porous medium. *Prikladnaja Matmatika i Mekhanika* **16**, 67–78.

Baumann, G. 2013 *Symmetry Analysis of Differential Equations with Mathematica®*. Springer Science & Business Media.

Bluman, G. & Anco, S. 2002 *Symmetry and Integration Methods for Differential Equations*. Springer.

Bosch, N., Hennig, P. & Tronarp, F. 2020 Calibrated adaptive probabilistic ODE solvers. *arXiv:2012.08202* .

Briol, F.-X., Oates, C. J., Girolami, M., Osborne, M. A. & Sejdinovic, D. 2019 Probabilistic integration: A role in statistical computation? (with discussion and rejoinder). *Statistical Science* **34** (1), 1–42.

Chabiniok, R., Wang, V. Y., Hadjicharalambous, M., Asner, L., Lee, J., Sermesant, M., Kuhl, E., Young, A. A., Moireau, P., Nash, M. P. *et al.* 2016 Multiphysics and multiscale modelling, data–model fusion and integration of organ physiology in the clinic: ventricular cardiac mechanics. *Interface Focus* **6** (2), 20150083.

Chaloner, K. & Verdinelli, I. 1995 Bayesian experimental design: A review. *Statistical Science* pp. 273–304.

Chang, J. T. & Pollard, D. 1997 Conditioning as disintegration. *Statistica Neerlandica* **51** (3), 287–317.

Chen, J., Chen, Z., Zhang, C. & Wu, C. F. J. 2020 APIK: Active physics-informed kriging model with partial differential equations. *arXiv:2012.11798* .

Chen, Y., Hosseini, B., Owhadi, H. & Stuart, A. M. 2021 Solving and learning nonlinear PDEs with Gaussian processes. *arXiv:2103.12959* .

Chkrebtii, O., Campbell, D. A., Girolami, M. A. & Calderhead, B. 2016 Bayesian uncertainty quantification for differential equations (with discussion). *Bayesian Analysis* **11** (4), 1239–1267.

Chkrebtii, O. A. & Campbell, D. A. 2019 Adaptive step-size selection for state-space probabilistic differential equation solvers. *Statistics and Computing* **29** (6), 1285–1295.

Cockayne, J. & Duncan, A. B. 2020 Probabilistic gradients for fast calibration of differential equation models. *arXiv:2009.04239* .

Cockayne, J., Graham, M. M., Oates, C. J. & Sullivan, T. J. 2021 Testing whether a learning procedure is calibrated. *arXiv:2012.12670* .

Cockayne, J., Oates, C., Sullivan, T. & Girolami, M. 2016 Probabilistic meshless methods for partial differential equations and bayesian inverse problems. *arXiv:1605.07811* .

Cockayne, J., Oates, C., Sullivan, T. & Girolami, M. 2019 Bayesian probabilistic numerical methods. *SIAM Review* To appear.

Conrad, P. R., Girolami, M., Särkkä, S., Stuart, A. & Zygalakis, K. 2017 Statistical analysis of differential equations: introducing probability measures on numerical solutions. *Statistics and Computing* **27** (4), 1065–1082.

Conte, S. & De Boor, C. 1980 *Elementary Numerical Analysis: An Algorithmic Approach*. McGraw-Hill.

Crank, J. & Nicolson, P. 1947 A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type. *Proceedings of the Cambridge Philosophical Society* **43** (1), 50–67.

Diaconis, P. 1988 Bayesian numerical analysis. *Statistical Decision Theory and Related Topics IV(1)* p. 1988.

Dudley, R. 1967 The sizes of compact subsets of hilbert space and continuity of gaussian processes. *Journal of Functional Analysis* **1** (3), 290–330.

Estep, D. 1995 A posteriori error bounds and global error control for approximation of ordinary differential equations. *SIAM Journal on Numerical Analysis* **32** (1), 1–48.

Evans, L. C. 1998 *Partial Differential Equations*, *Graduate Studies in MathematicsGraduate Studies in Mathematics*, vol. 19. American Mathematical Society.

Fasshauer, G. E. 1999 Solving differential equations with radial basis functions: multilevel methods and smoothing. *Adv. Comput. Math.* **11** (2-3), 139–159.

Fefferman, C. L. 2000 Existence and smoothness of the navier?stokes equation.

Fornberg, B. & Flyer, N. 2015 Solving PDEs with radial basis functions. *Acta Numerica* **24**, 215–258.

Fulton, E. A. 2010 Approaches to end-to-end ecosystem models. *Journal of Marine Systems* **81** (1-2), 171–183.

Gelman, A. & Shalizi, C. R. 2013 Philosophy and the practice of Bayesian statistics. *British Journal of Mathematical and Statistical Psychology* **66** (1), 8–38.

Gneiting, T. 2002 Compactly supported correlation functions. *Journal of Multivariate Analysis* **83** (2), 493–508.

Hartman, P. 1982 *Ordinary Differential Equations: Second Edition*. Society for Industrial and Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104).

Hawkins, T. 2012 *Emergence of the theory of Lie groups: An essay in the history of mathematics 1869–1926*. Springer Science & Business Media.

Hennig, P., Osborne, M. A. & Girolami, M. 2015 Probabilistic numerics and uncertainty in computations. *Proceedings of the Royal Society A* **471** (2179), 20150142.

Higham, N. J. 2002 *Accuracy and Stability of Numerical Algorithms*. SIAM.

Hurrell, J. W., Holland, M. M., Gent, P. R., Ghan, S., Kay, J. E., Kushner, P. J., Lamarque, J.-F., Large, W. G., Lawrence, D., Lindsay, K. *et al.* 2013 The community earth system model: a framework for collaborative research. *Bulletin of the American Meteorological Society* **94** (9), 1339–1360.

Iserles, A. 2008 *A First Course in the Numerical Analysis of Differential Equations*, 2nd edn. Cambridge University Press.

Jeanblanc, M., Yor, M. & Chesney, M. 2009 *Mathematical Methods for Financial Markets*. Springer Science & Business Media.

Jidling, C., Wahlström, N., Wills, A. & Schön, T. B. 2017 Linearly constrained Gaussian processes. In *Proceedings of the 31st Conference on Neural Information Processing Systems*.

Karvonen, T., Oates, C. J. & Särkkä, S. 2018 A bayes-sard cubature method. In *Proceedings of the 32nd Annual Conference on Neural Information Processing Systems (NeurIPS2018)*, pp. 5882–5893.

Karvonen, T., Wynne, G., Tronarp, F., Oates, C. J. & Särkkä, S. 2020 Maximum likelihood estimation and uncertainty quantification for gaussian process approximation of deterministic functions. *Journal of Uncertainty Quantification* **8** (3), 926–958.

Kennedy, M. C. & O'Hagan, A. 2001 Bayesian calibration of computer models. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **63** (3), 425–464.

Kersting, H. & Hennig, P. 2016 Active uncertainty calibration in Bayesian ODE solvers. In *Proceedings of the 32nd Conference on Uncertainty in Artificial Intelligence (UAI 2016)*, pp. 309–318.

Kersting, H., Sullivan, T. & Hennig, P. 2018 Convergence rates of gaussian ode filters. *arXiv:1807.09737* .

Krämer, N. & Hennig, P. 2020 Stable implementation of probabilistic ode solvers.

Kunita, H. 1997 *Stochastic Flows and Stochastic Differential Equations*. Cambridge University Press.

Larkin, F. 1972 Gaussian measure in Hilbert space and applications in numerical analysis. *The Rocky Mountain Journal of Mathematics* **2**, 379–421.

Larkin, F. M. 1974 Probabilistic error estimates in spline interpolation and quadrature. In *Information Processing 74 (Proc. IFIP Congress, Stockholm, 1974)*, pp. 605–609.

LeVeque, R. J. 2002 *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press.

Lie, H., Stuart, A. & Sullivan, T. 2019 Strong convergence rates of probabilistic integrators for ordinary differential equations. *Statistics and Computing* To appear.

Lie, H. C., Sullivan, T. & Teckentrup, A. L. 2018 Random forward models and log-likelihoods in bayesian inverse problems. *SIAM/ASA Journal on Uncertainty Quantification* **6** (4), 1600–1629.

López-Lopera, A. F., Bachoc, F., Durrande, N. & Roustant, O. 2018 Finite-dimensional Gaussian approximation with linear inequality constraints. *SIAM/ASA Journal of Uncertainty Quantification* **6** (3), 1224–1255.

MacNamara, S. & Strang, G. 2016 Operator splitting. In *Splitting Methods in Communication, Imaging, Science, and Engineering*, pp. 95–114. Springer.

Magnani, E., Kersting, H., Schober, M. & Hennig, P. 2017 Bayesian filtering for odes with bounded derivatives.

Marcus, M. B. & Shepp, L. A. 1972 Sample behavior of gaussian processes. In *Proceedings of the 6th Berkeley Symposium on Mathematical Statistics and Probability, Volume 2: Probability Theory*, pp. 423–441. Berkeley, Calif.: University of California Press.

Matsuda, T. & Miyatake, Y. 2021 Estimation of ordinary differential equation models with discretization error quantification.

Morton, K. W. & Mayers, D. F. 2005 *Numerical Solution of Partial Differential Equations: An Introduction*, 2nd edn. Cambridge University Press.

Oates, C. J., Cockayne, J., Aykroyd, R. G. & Girolami, M. 2019*a* Bayesian probabilistic numerical methods in time-dependent state estimation for industrial hydrocyclone equipment. *Journal of the American Statistical Association* **114** (528), 1518–1531.

Oates, C. J., Cockayne, J., Prangle, D., Sullivan, T. J. & Girolami, M. 2019*b* *Multivariate Algorithms and Information-Based Complexity*, chap. Optimality Criteria for Probabilistic Numerical Methods. Berlin/Boston De Gruyter, to appear.

Oates, C. J. & Sullivan, T. J. 2019 A modern retrospective on probabilistic numerics. *Statistics and Computing* To appear.

Overstall, A., Woods, D. & Parker, B. 2019 Bayesian optimal design for ordinary differential equation models with application in biological science. *Journal of the American Statistical Association* To appear.

Pakman, A. & Paninski, L. 2014 Exact hamiltonian monte carlo for truncated multivariate gaussians. *Journal of Computational and Graphical Statistics* **23**, 518–542.

Perilla, J. R., Goh, B. C., Cassidy, C. K., Liu, B., Bernardi, R. C., Rudack, T., Yu, H., Wu, Z. & Schulten, K. 2015 Molecular dynamics simulations of large macromolecular complexes. *Current Opinion in Structural Biology* **31**, 64–74.

Petzold, L., Li, S., Cao, Y. & Serban, R. 2006 Sensitivity analysis of differential-algebraic equations and partial differential equations. *Computers & Chemical Engineering* **30** (10-12), 1553–1559.

Potthoff, J. 2010 Sample properties of random fields III: Differentiability. *Communications on Stochastic Analysis* **4** (3), 335–353.

Quiñonero-Candela, J. & Rasmussen, C. E. 2005 A unifying view of sparse approximate gaussian process regression. *Journal of Machine Learning Research* **6** (65), 1939–1959.

Raissi, M., Perdikaris, P. & Karniadakis, G. E. 2018 Numerical gaussian processes for time-dependent and nonlinear partial differential equations. *SIAM Journal on Scientific Computing* **40** (1), A172–A198.

Rasmussen, C. E. & Williams, C. K. 2006 Gaussian processes for machine learning. *The MIT Press, Cambridge, MA, USA* **38**, 715–719.

de Roos, F., Gessner, A. & Hennig, P. 2021 High-dimensional Gaussian process inference with derivatives. *arXiv:2102.07542* .

Sanz-Serna, J. M. 1992 Symplectic integrators for Hamiltonian problems: An overview. *Acta Numerica* **1**, 243–286.

Schäfer, F., Sullivan, T. J. & Owhadi, H. 2021 Compression, inversion, and approximate PCA of dense kernel matrices at near-linear computational complexity. *Multiscale Modelling and Simulation* To appear.

Scheuerer, M. 2010 Regularity of the sample paths of a general second order random field. *Stochastic Processes and their Applications* **120** (10), 1879–1897.

Schober, M., Duvenaud, D. K. & Hennig, P. 2014 Probabilistic ODE solvers with Runge-Kutta means. In *Proceedings of the 28th Annual Conference on Neural Information Processing Systems (NIPS 2014)*, pp. 739–747.

Schober, M., Särkkä, S. & Hennig, P. 2019 A probabilistic model for the numerical solution of initial value problems. *Statistics and Computing* To appear.

Schölkopf, B., Herbrich, R. & Smola, A. J. 2001 A generalized representer theorem. In *Proceedings of the 14th International Conference on Conference on Computational Learning Theory*, pp. 416–426. Springer.

Skeel, R. D. & Berzins, M. 1990 A method for the spatial discretization of parabolic equations in one space variable. *SIAM J. Sci. Statist. Comput.* **11** (1), 1–32.

Skilling, J. 1992 Bayesian solution of ordinary differential equations. In *Maximum Entropy and Bayesian Methods*, pp. 23–37. Springer.

Stein, M. L. 1999 *Interpolation of Spatial Data*. Springer-Verlag New York.

Strikwerda, J. C. 2004 *Finite Difference Schemes and Partial Differential Equations*. SIAM.

Sullivan, T. 2015 *Introduction to Uncertainty Quantification*. Springer.

Teymur, O., Lie, H. C., Sullivan, T. & Calderhead, B. 2018 Implicit probabilistic integrators for ODEs. In *Proceedings of the 32nd Annual Conference on Neural Information Processing Systems (NeurIPS 2018)*.

Teymur, O., Zygalakis, K. & Calderhead, B. 2016 Probabilistic linear multistep methods. In *Proceedings fo the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016)*, pp. 4321–4328.

Thomas, J. 1998 *Numerical Partial Differential Equations: Finite Difference Methods*. Springer Science & Business Media.

Traub, J. & Woźniakowski 1992 Perspectives on information-based complexity. *Bulletin of the American Mathematical Society* **26** (1), 29–52.

Tronarp, F., Kersting, H., Särkkä, S. & Hennig, P. 2019 Probabilistic solutions to ordinary differential equations as non-linear Bayesian filtering: A new perspective. *Statistics and Computing* To appear.

Tronarp, F., Särkkä, S. & Hennig, P. 2021 Bayesian ode solvers: the maximum a posteriori estimate. *Statistics and Computing* **31** (3), 23.

Wang, X. & Berger, J. O. 2016 Estimating shape constrained functions using Gaussian processes. *Journal on Uncertainty Quantification* **4** (1), 1–25.

Wedi, N. P. 2014 Increasing horizontal resolution in numerical weather prediction and climate simulations: illusion or panacea? *Philosophical Transactions of the Royal Society A* **372** (2018), 20130289.

Wheeler, M. W., Dunson, D. B., Pandalai, S. P., Baker, B. A. & Herring, A. H. 2014 Mechanistic hierarchical gaussian processes. *Journal of the American Statistical Association* **109** (507), 894–904.