TOPICS IN THE PROBABILISTIC SOLUTION OF ORDINARY DIFFERENTIAL EQUATIONS

Onur Teymur

A thesis submitted in partial fulfilment of the requirements of the degree of Doctor of Philosophy

— to the —

Department of Mathematics
Imperial College London

2018

DECLARATIONS

ORIGINALITY

This thesis is my own work, conducted during the period of my enrolment at Imperial College London. Parts of the material are related to collaborative work I have undertaken with my advisor Dr Ben Calderhead and, to a lesser extent, with outside collaborators Dr Konstantinos Zygalakis, Dr Han Cheng Lie and Dr Tim Sullivan. Their specific contributions are noted as they arise.

Onur Teymur

v1: 30.10.2018
v2: 27.02.2019

COPYRIGHT

ABSTRACT

This thesis concerns several new developments in the probabilistic solution of ordinary differential equations. Probabilistic numerical methods are differentiated from their classical counterparts through the key property of returning a probability measure as output, rather than simply a point value. When properly calibrated, this measure can then be taken to probabilistically represent the output uncertainty arising from the application of the numerical procedure.

After giving some introductory context, we start with a concise survey of the still-developing field of probabilistic ODE solvers, highlighting how several different paradigms have developed somewhat in parallel. One of these, established by Conrad et al. [Con16], defines randomised one-step solvers for initial value problems, where the outputs are empirical measures arising from Monte Carlo repetitions of the algorithm. We extend this to multistep solvers of Adams–Bashforth type using a novel Gaussian process construction. The properties of this method are explored and its convergence is rigorously proved.

We continue by defining a class of implicit probabilistic ODE solvers, the first in the literature. Unlike explicit methods, these modified Adams–Moulton algorithms incorporate information from the ODE dynamics beyond the current time-point, and as such are able to enhance the accuracy of the probabilistic model of numerical error. In their full form, they output a non-parametric description of the stepwise error, though we also propose a parametric approximation that aids computation. Once again, we explore the properties of the method and prove its convergence in the small step-size limit.

We follow with a discussion on the problem of calibration for these classes of algorithms, and generalise a proposal from Conrad et al. in order to implement it for our methods. We then apply the new integrators to two test differential equation models, first in the solution of the forward model, then later in the setting of a Bayesian inverse problem. We contrast the effect of using probabilistic integrators instead of classical ones on posterior inference over the model parameters, as well as derived functions of the forward solution.

We conclude with a brief discussion on the advantages and shortcomings of the proposed methods, and posit several suggestions for potential future research.

ACKNOWLEDGEMENTS

Thanks to Ben Calderhead for his support and encouragement; to Konstantinos Zygalakis, Han Cheng Lie and Tim Sullivan for essential insight in key places; to David van Dyk, Daniel Mortlock, Alistair Young and Chris Oates for valuable feedback; to François-Xavier Briol and Nikolas Nüsken for their consistent intellectual generosity; and to my mum, for absolutely everything.

This is dedicated to my dad, who would have loved to have seen it.

CONTENTS

GLOSSARY OF NOTATION

Throughout this work we attempt to maintain certain consistent notations and indexing conventions which we now summarise here. On occasion these are eased where this locally benefits the clarity of the exposition—such instances are highlighted in the text.

Where possible we use lower-case Latin letters ($x, z, f, \dots$) to refer to continuous objects and upper-case Latin letters ($X, Z, F, \dots$) to refer to corresponding discrete objects. In each case these may be deterministic variables, such as a function $f(x(t), t)$, or random variables such as Z_i. The context should make this clear, and this ambiguity is unavoidable when much of our work concerns the probabilistic treatment of quantities usually thought of as deterministic.

We use t to denote time, the independent variable in an ODE. Subscript indices usually refer to time ordinates. $x(t_i)$, the value of the function x at time t_i, may also be denoted by X_i. A sequence of variables $X_i, X_{i+1}, \dots, X_{j-1}, X_j$ will be written $X_{i:j}$ and, where context is clear, sometimes simply as X. We occasionally employ the notation $X_{\leq i}$, meaning the sequence of variables up to and including X_i. When considering multivariate problems we use d for the dimension and superscripts in round brackets to index them. Such vectors are written as columns; thus $X_i \equiv (X_i^{(1)}, \dots, X_i^{(d)})^T$.

In several situations, multiple instantiations of a particular algorithm produce copies of variables—in the stochastic case this corresponds to the repeated realisation of a random variable. We denote these with superscripts in square brackets, so that the k'th sample from the measure induced by the random variable X is denoted $X^{[k]}$.

We tend to use lower-case Greek letters ($\alpha, \phi, \theta, \ldots$) for parameters. In an inverse problem setting, models are parameterised by θ, which in general is q-dimensional. Experimental data is denoted by Y and is a vector of M elements or, where Y is multidimensional of dimension d_Y, a $d_Y \times M$ matrix.

When considering the approximate solution of an ODE, Z_i refers to the approximation to X_i, the value of the true solution x at time t_i. $\dot{X}_i$ refers to the true derivative at t_i, *i.e.* $\dot{X}_i \equiv \mathrm{d}X_i/\mathrm{d}t = f(X_i, t_i)$. Passing the approximate solution Z_i to the function f gives an approximation $f(Z_i, t_i)$ to the derivative which we denote F_i, avoiding the confusing use of $\dot{Z}_i$ since with inexact inputs f does not act as a differential operator.

In initial value problems, where time starts at $t = 0$, it makes sense to use zero-indexing, so that X_0 refers to $x(t_0)$ where t_0 is simply 0. We usually divide the interval of integration $[t_0, t_{\text{end}}]$ into N segments so that $X_N = x(t_{\text{end}})$ and $X \equiv X_{0:N}$ has cardinality $N+1$. In a multivariate problem, X is an $d \times (N+1)$ matrix.

In iterative procedures the current iteration is denoted by index i, so that the output of a time-stepping method returns a state approximation at index $i+1$. In general we try to use i, j to index across time; v, w to index across dimensions; and k, m to index across iterations.

A number of other symbols are reserved for particular concepts throughout. We list some of them here:

h	time-step in iterative algorithms
s	number of steps in multistep methods
$\beta^{AB}_{j,s}$	Adams–Bashforth coefficients
$\beta^{AM}_{j,s}$	Adams–Moulton coefficients
e_i	(classical) local truncation error at step i
E_i	(classical) global error at step i
α	integrator calibration constant
ω	elementary event
Ω	sample space
ξ	set of perturbations of randomised integrators
Φ_t	(exact) flow map
Ψ^h	numerical flow map
L_f	Lipschitz constant of f
J_f	Jacobian matrix of f

1

INTRODUCTION & BACKGROUND

1.1 INVERSE PROBLEMS

In most of science, experimental observations are not direct, unmediated representations of the quantity of interest. For example, a biologist may wish to know the rate at which some hormone is secreted by a certain gland in an animal's body. This is, for practical purposes, impossible to observe directly. But if the biologist has a scientific model for the effects of this hormone—say that higher concentrations make the hair on the animal's body grow faster—measurements of hair length may provide information about the hormone secretion rate.

Drawing conclusions about an inaccessible underlying process based on a set of indirect observations, along with a mathematical model connecting the two, is an instance of an inverse problem. The term is used in opposition to the forward problem, in which the outcome of an experiment is deduced given knowledge of the real process. (If the biologist were able to alter the secretion rate of the hormone in a controlled manner, what change in hair growth would be expected?) Mathematically, this set-up is written:

$$y = \mathcal{G}(x) + \varepsilon \tag{1.1}$$

Here, x represents the underlying physical process and y represents the collected data. $\mathcal{G}$ is the observation operator, which encodes the model assumed to connect the two. ε represents the random noise that it is accepted will inevitably corrupt any measurement. The inverse problem seeks to recover x given y, $\mathcal{G}$, and some statistical assumptions on ε.[1]

This formulation is extremely general, and we now restrict it to the more specific format we will consider in this work. Firstly, we will assume throughout that the data is a discrete object of cardinality M. This corresponds to the reasonable real-world assumption that the number of measurements is finite. In line with our intention to represent discrete objects by capital letters, we will henceforth term the dataset Y.

Each element Y_j of Y is a real-valued vector of dimension d_Y. We also assume that the points where the data are collected can be indexed by a one-dimensional continuous scalar variable $t \in \mathbb{R}$, which we identify with time. Such a restriction is justified because we focus on methods for ODEs rather than those for PDEs. For each time t_{Y_j} for which a datum Y_j exists, we say that there is a corresponding random noise vector $\varepsilon_j \in \mathbb{R}^{d_Y}$, each of which is independent and identically distributed according to some probability measure.

The nature of x we leave more general. Its dimension d is not necessarily equal to d_Y. Typically x would be a function (of time t) and depend on a q-dimensional parameter θ. Each component of this parameter represents some quantitative degree of freedom in the model specification, the value of which is unknown. In our earlier example, it may be that while the biologist's model states that a linear relationship exists between hormone secretion and hair growth, the actual quantity of hair growth that can be attributed to a change of one unit of hormone is not known. This unknown quantity becomes a parameter of the model.

In parametric models, the inverse problem is in effect the attempt to deduce the value of the parameter θ since, given this, we can completely specify the structure of the model. In our set-up, we make this dependence clear by writing the underlying true process as $x_\theta(t)$.[2]

Lastly we consider the observation operator $\mathcal{G}$. We assume this is independent of t (except through its argument x) and therefore acts as a sort of projection operator from $\mathbb{R}^{d_Y}$ to $\mathbb{R}^d$. In other words, at a particular time t^*, it acts as a function mapping $x_\theta(t^*) \in \mathbb{R}^{d_Y}$ to a vector in $\mathbb{R}^d$. We generally assume that this mapping is linear, and can therefore be represented by a matrix G.[3]

[1] We have separated the stochastic noise ε from the observation operator, though some treatments combine them and define a stochastic observation operator $\mathcal{G}_\varepsilon$ instead. We have also made the common assumption that the measurement noise is additive—for our purposes this suffices, but naturally some models do assume a functionally more complicated error than this.

[2] The situation in which the range of the parameters does not cover all plausible models is suggestive of inadequacy of the model itself. Resolving this is a complex problem and is often not possible to do in a systematic way. Furthermore, even the meaning of the 'true value' of a parameter can be subtle. For example, if the model itself is a poor one, the use of a supposedly known parameter value—such as some universal physical constant—may result in predictions inferior to the use of some alternative 'not exactly true' value. Thus model inadequacy can also manifest as parameter uncertainty. These and other related issues are discussed in depth by Kennedy & O'Hagan [Ken01].

Summarising the above, we can rewrite (1.1) more precisely as

$$Y_j = Gx_\theta(t_{Y_j}) + \varepsilon_j\,, \qquad j = 1, 2, \ldots, M \tag{1.2}$$

Recall that given the data $Y \equiv Y_{1:M}$, the matrix G, the distribution of ε, and the assumed form of the model x_θ, we seek to infer the parameter θ.

1.1.1 *What does it mean to 'solve' an inverse problem?*

The question of how to describe the 'solution' of a parametric inverse problem does not in itself have an obvious answer. The standard approach is to report an estimate $\widehat{\theta}$ for the unknown parameter θ, calculated using some pre-agreed statistical estimation procedure. For example, in a least squares approach, the estimator

$$\widehat{\theta} = \operatorname*{argmin}_{\theta\in\Theta} \sum_{j=1}^{M} \left(Y_j - G(x_\theta(t_{Y_j}))\right)^2 \tag{1.3}$$

is calculated (approximately—using conjugate gradients, say [Kai06, §5.5]). If the errors ε_j are independent zero-mean Gaussian, and G and x_θ are linear mappings, this L^2 minimisation returns the maximum likelihood estimator, by a result known as the Gauss–Markov Theorem [Gre12, §4.4]. Other estimators can be considered—we may seek to minimise the L^1 residuals instead, and this estimator can be shown to be more robust to outliers in the data [Ast11, §1.2].

The problem with such approaches is that inverse problems are typically ill-posed [Had02]. In this situation, small changes in the data Y can produce abrupt—and sometimes arbitrarily large—changes in the inferred parameter values [Stu10]. This means firstly that the calculation itself can be unstable. However, even if the minimisation can be successfully performed, the scientific value of reporting a point estimate is questionable when there is such a strong sensitivity to the specific dataset.

The classical approach to mitigating this issue is regularisation [Tik95] where, for example, penalty terms are added to the right-hand side of (1.3) to improve stability, or some type of dimension reduction is performed to simplify the computation. While this strategy is in widespread use—and has a solid theory behind it—it introduces bias to the estimator [Ast11, §1.4], adds further parameters which it is often unclear how to set, and essentially constitutes nothing other than an *ad hoc* removal of as much ill-posedness from the problem as is necessary to report an answer [Kai06, §3].

[3] Our focus in this thesis is on methodological issues rather than the analysis of a particular problem, and we can justify this simplification on those grounds. That being said, Aster et al. [Ast11] give several examples from geophysics where the assumption of linearity is in fact warranted in practice.

1.1.2 *Statistical inverse problems*

A different approach to inverse problems is possible using the framework of Bayesian statistical theory. In short, this paradigm requires us to treat *all* variables in (1.2) as random variables, with probability measures representing our degree of knowledge about their values. In particular, this approach reinterprets the 'solution' of an inverse problem to mean a characterisation of the probability distribution of θ given the data Y, rather than simply a point estimate for it [Tar05, §1.5]. The formal theoretical framework underpinning this paradigm is that of Bayesian probability theory [Jay03]. An instructive viewpoint is given by Kaipio & Somersalo [Kai06, §3], who point out that rather than being viewed as a reduction in ill-posedness, this paradigm actually reformulates the problem as a well-posed problem but in the larger space of probability distributions. The rigorous mathematical foundations of this view are explored by Stuart [Stu10].

We will introduce some of the basic theory and methods of Bayesian probability in the coming sections, highlighting in particular the ways in which this extremely general theory of probability applies to the setting of parameter inference in inverse problems. Before doing so we note that this application falls squarely within the realm of Uncertainty Quantification (UQ) [Smi13; Sul15], an emerging field using precisely these tools—amongst others—to enrich the reporting of scientific conclusions with a rigorous description of sources of error. The cardinal aim of Uncertainty Quantification—which is the foundation for its increasing propagation through so many parts of applied science—is that of enabling rational decision-making under uncertainty [Tan07].

1.2 BAYESIAN INFERENCE

Statistical inference is usually defined as the process of deducing a probability model for some phenomenon based on the analysis of data generated by that same phenomenon [Rob07, §1]. The focus can be on trying to explain the mechanism that produced the data at hand, or on the prediction of future behaviour, or both.

For the parametric inverse problem described in Section 1.1, the set of possible probability models is assumed to be indexed by an unknown finite-dimensional parameter θ taking values in a space $\Theta \subseteq \mathbb{R}^q$, such that the aim of the inference procedure is to deduce a data-informed probability measure over Θ. In the Bayesian approach, this is done by defining a prior distribution $p(\theta)$ representing our knowledge of θ before the data are considered, and a parametric sampling distribution $p(Y|\theta)$ which encodes the assumed form of the forward model as well as information about various sources of error in the data generating process. (The way in which errors are represented in

this distribution is one of the central points of our later study.) Together, these form a joint probability model over (θ, Y) that can then be conditioned on Y which, having already been observed, is known. This last step—of conditioning on the data in hand in a formal probabilistic sense—is the cornerstone of Bayesian analysis.[4]

The resulting distribution is the posterior probability distribution $p(\theta|Y)$ and inference about θ is made based on this. The 'solution' of the inverse problem in this context is thus taken to be given by this posterior distribution, or well-chosen derived quantities—for example the mean—calculated from it. The fundamental relationship between these quantities is given by Bayes' Theorem

$$p(\theta|Y) = \frac{p(Y|\theta)p(\theta)}{p(Y)} \tag{1.4}$$

1.2.1 *Practical Bayesian analysis*

We have now established that in order to analyse the inverse problem arising from equation (1.2) in a Bayesian way, all we have to do is choose a prior distribution $p(\theta)$, encode our scientific model into the distribution $p(Y|\theta)$ and out spits the posterior $p(\theta|Y)$—which we have called our 'solution'. What could be simpler?!

The truth is that no part of this suggested programme is trivial to implement in reality—indeed each is the primary subject of significant and ongoing research endeavours in contemporary statistics. We briefly examine them in turn, and highlight where our thesis makes a contribution.

Choosing the prior

Historically, the choice of prior distribution has been—particularly for non-Bayesians—one of the most controversial elements of the Bayesian approach to statistical inference [Gel08]. Firstly, a note on interpretation. The prior distribution requires us to encode our lack of knowledge of θ in mathematical form. The measure $p(\theta)$ is thus a quantification of epistemic uncertainty about θ, and does not mean that θ is truly random in a frequentist sense.

As pointed out by Kaipio & Somersalo [Kai06, §3.3], prior knowledge is often qualitative in nature, and the challenge often consists in adequately translating this qualitative

[4] A rigorous measure-theoretic approach would demand that we define some π to represent a probability measure on the measurable space Θ, with $p(\theta)$—considered as a function of the variable θ—representing its density. Our exposition will conflate these where no ambiguity results, because for most of it such a distinction is needlessly fiddly. During some of the later analysis we will be more careful. We will also write $p(\theta^*)$ for the probability of a specific element $\theta^* \in \Theta$. In general we assume all spaces are appropriately measurable and all measures have densities, unless otherwise noted.

information into the form of a probability measure. In the absence of strong prior information, other approaches may be used, such as conjugate priors (for mathematical convenience), or uninformative priors (in an attempt to be objective) [You05, §3.6].[5]

Beyond noting these issues, we will not consider prior choice deeply in this work. Where the stability of the simulations is not compromised by such a choice, we will often resort to the improper prior $p(\theta) \propto 1$, which does not represent a proper probability measure if the parameter space Θ is such that its Lebesgue measure is not finite. This will commonly be the case, since Θ is often a non-compact subset of Euclidean subspace $\mathbb{R}^q$.

Regardless of the prior we use, we assume throughout that it is easily and exactly evaluable pointwise—that is, given a particular element $\theta^* \in \Theta$, its probability—the value of $p(\theta^*)$, with $p(\cdot)$ treated as a function—is trivial to calculate without error.

Encoding the model

The sampling distribution $p(Y|\theta)$ encodes the assumed mathematical relationship between the data Y and the ground truth. The choice of model is a key task for the statistician, and comparison of different models—to find out which best explains the data—is one of the main endeavours of statistics in practice [Cox06, Appendix B].

Since θ is unknown but Y is known, this distribution is often treated as a function of θ, in which case it is known as the likelihood and written $L(Y;\theta)$. We can identify the likelihood with the operator $\mathcal{G}$ in equation (1.2), though it also carries information about the structure of the measurement error ε. In both Bayesian and frequentist statistics, the central status of the likelihood function is expressed by the Likelihood Principle, which states that all the information in the data that is to be used for posterior inference must be contained in the likelihood function. Jaynes [Jay03, §8.5] discusses the interpretation of this principle at length.

In the common situation (assumed throughout this work) that each datum is corrupted by an independent measurement error, the likelihood function will take the form of a product of their individual likelihoods. Thus we have

$$L(Y;\theta) = \prod_{j=1}^{M} p(Y_j|\theta) \tag{1.5}$$

For our purposes, we will assume that the *form* of the likelihood function $L(Y;\theta)$ is fully specified in advance of the analysis. That is, the assumed likelihood can be straightforwardly written down—even if evaluating it is not always straightforward.

[5] Though a committed Bayesian would invariably contend that a claim asserting the absence of prior information is itself an informative statement of the state of prior knowledge.

Evaluating the likelihood

Equation (1.5) is the canonical example of the near-universal situation in which the likelihood function is not a normalised probability density function. This means that, even if the likelihood is defined in closed form and we are given $\theta^* \in \Theta$, the value $L(Y; \theta^*)$ to which it evaluates is not the probability of θ^* given Y. As a result, we cannot simply apply Bayes' Theorem (1.4). This is the first of several practical issues that complicate our calculation.

The subtleties of evaluating the likelihood are various and different problems arise in different situations. The core material of this thesis addresses the case where the likelihood is not evaluable exactly, even given an exact value θ^* and even when its form is fully specified, due to inescapable mathematical impediments. We focus on differential equation models, for which the likelihood function is defined implicitly, and which fall into this category. We expand on this setup in considerable detail starting in Section 1.3.

In passing, we give some other examples of difficulties that may arise in likelihood evaluation. Where the quantity of data is very large, it may be that that the full product in equation (1.5) is too expensive to calculate. Choosing a strict subset of the data—randomly, but in a particular fashion—and evaluating only the likelihood contributions of this subset, can result in an unbiased estimator of the log-likelihood. If such an estimator is used in place of the full likelihood in a Monte Carlo procedure (see Section 1.2.2), then under the pseudo-marginal framework [And09], posterior inference may still be possible without introducing bias.

Sometimes the likelihood is completely intractable and alternative methods are required which are able to circumvent the need to evaluate it completely. A broad class of methods called approximate Bayesian computation (ABC) seeks to directly generate samples from the posterior distribution without evaluating the likelihood at all. Such methods are the subject of considerable contemporary research—Sisson et al. [Sis18] give a comprehensive survey.

Exploring and interpreting the posterior

The dual problem to encoding prior and model assumptions into probability measures is that of decoding inferences from the posterior [Kai06, §3.5]. In the previous paragraphs we described how, given a particular value θ^*, the prior and likelihood (or at least some approximation to it) can be evaluated pointwise. This is of course insufficient to perform inference by itself—the parameter space Θ needs to be explored to reveal the structure of the posterior measure $p(\theta|Y)$. In effect this means that a procedure needs to be designed to choose a set of values $\theta_1, \theta_2, \ldots$ for which the posterior should be evaluated.

A simple approach that may work when the dimension of Θ is small (and where the subset $\Theta^\dagger \subset \Theta$ in which $p(\theta|Y)$ takes high values is known or easily found) is evaluating a pre-determined set of values—chosen on a grid, say—in a procedure sometimes termed quantisation. A much more general approach is to construct an algorithm to sample from $p(\theta|Y)$. We expand on this idea in Section 1.2.2.

Assuming now that the posterior distribution $p(\theta|Y)$ has been characterised in some way, the next step is to interpret it. While we have noted that the core principle of the Bayesian approach to inverse problems is that of returning a full posterior probability distribution rather than simply a point estimate as in equation (1.3), probability distributions are difficult to interpret in themselves.

As a result, typically some estimator $\widehat{\theta}$ is chosen to summarise the posterior measure—commonly the maximum a posteriori estimator $\widehat{\theta}_{\text{MAP}} = \text{argmax}_{\theta\in\Theta}\, p(\theta|Y)$ or some estimator based on minimising a pre-specified loss function, such as the mean estimator $\widehat{\theta}_{L^2} = \mathbb{E}_\theta(\theta|Y)$ (which minimises L^2 loss) or the median estimator $\widehat{\theta}_{L^1}$ (which minimises L^1 loss).[6] One clear advantage of having access to the full posterior distribution is that other statistics can also be calculated, from which it is possible to make statements about the degree of confidence in the reported estimator, such as $\text{Var}(\theta|Y)$. If the aim is to track the uncertainty in the conclusion, these additional quantities are essential.

It is also worth noting that if the inference procedure forms part of a larger chain of computations, it may in fact be the posterior parameter samples *themselves* that are the useful inputs to the next stage of the process.

1.2.2 *Approximate inference and Monte Carlo methods*

As alluded to in the previous section, an ever-present feature of practical Bayesian analysis is the requirement to deal with well-defined but computationally intractable quantities. The effective approximation of such quantities is one of the central fields of research in modern computational Bayesian statistics. Let us first consider the canonical example. Consider the parameter inference problem described in Section 1.2. We assume that, given a particular value θ^*, the prior $p(\theta^*)$ is exactly evaluable. Let us also assume here that the likelihood function $L(Y;\theta^*)$ is also evaluable.

Substituting the likelihood $L(Y;\theta)$ for the sampling distribution $p(Y|\theta)$ in equation (1.4) and noting that, as probability measures, the integral over the parameter space

[6] The median estimator (for univariate θ) can be expressed mathematically by defining the cumulative distribution function $F(\theta|Y) := \int_{-\infty}^{\theta} p(\theta'|Y)\,\mathrm{d}\theta'$ and taking the estimator to be $\widehat{\theta}_{L^1} = F^{-1}(0.5)$.

Θ of both sides must equal unity, we can write

$$p(\theta|Y) = \frac{L(Y;\theta)p(\theta)}{\int_\Theta L(Y;\theta)p(\theta)d\theta} \tag{1.6}$$

From this it is clear that the seemingly innocuous task of determining the posterior $p(\theta|Y)$ actually requires the evaluation of an integral. Furthermore, except for in extremely limited cases—when the prior and likelihood form a conjugate pair—this integral has no closed form solution. It must therefore be approximated, and if the dimension q of Θ is even moderate, classical quadrature procedures become unfeasible, since the number of grid points required grows exponentially with q [Rob04, §4.3]. The standard approach is therefore based on Monte Carlo simulation.

The Monte Carlo method is a sampling-based approximation method for integrals which can be cast as expectations with respect to a probability measure. In the case of Bayesian inference, the derived statistics of the posterior measure $p(\theta|Y)$ that are used to summarise it, such as its expectation or variance, are all expressible in terms of such expectations. Specifically, for some square-integrable function g, we have by the definition of expectation

$$\mathbb{E}_\theta(g(\theta)|Y) = \int_\Theta g(\theta)p(\theta|Y)\,\mathrm{d}\theta \tag{1.7}$$

(For $g(\theta) = \theta$, we recover an expression for the posterior mean.) The Monte Carlo method follows from the Law of Large Numbers, which states that given a set of independent and identically distributed samples $\theta^{[1]}, \theta^{[2]}, \ldots, \theta^{[K]}$ drawn from a probability measure, their ergodic average converges to the mean of this measure as $K \to \infty$. For our posterior measure $p(\theta|Y)$ we therefore have, given K samples $\theta^{[1]}, \theta^{[2]}, \ldots, \theta^{[K]} \sim p(\theta|Y)$, the approximation

$$\mathbb{E}_\theta(g(\theta)|Y) \approx \frac{1}{K}\sum_{k=1}^{K} g(\theta^{[k]}) \tag{1.8}$$

This is all very well if we are able to draw samples at will from the posterior, but the expression (1.6) makes it clear that since all we can do is evaluate it in unnormalised form, this is not completely trivial. The canonical solution is to implement a Markov chain Monte Carlo (MCMC) algorithm [Rob04, §7], a class of procedures which are able to generate (correlated) samples from $p(\theta|Y)$ given only an ability to evaluate an unnormalised version of its density.

The prototypical example of MCMC is the Metropolis–Hastings algorithm [Met53; Has70], which works by constructing an ergodic Markov chain on the space Θ that has stationary distribution $p(\theta|Y)$, and relies on defining a transition kernel $q(\cdot|\theta)$ that

generates candidate samples at each step which are then accepted or rejected according to a specific rule. Comprehensive studies of the foundations of this principle are given in Robert & Casella [Rob04, §7], Liu [Liu01, §5], and Gelman et al. [Gel13, §11].

Another MCMC algorithm, often employed when the target distribution is multidimensional and conditional densities of subsets of the variables are available, is Gibbs sampling [Gem84]. Gibbs sampling can be combined with (and even nested in) the Metropolis–Hastings algorithm in various ways, producing a large class of Markov chain-based sampling algorithms. Furthermore, an enormous field of research exists on improving the efficacy of these methods—for example to reduce correlation in the chain, or to allow the chain to reach its stationary distribution more quickly, or to allow it to explore the entire parameter space Θ without getting either 'stuck' or 'lost'. The references in the previous paragraph all give extensive discussion of these topics.

In Section 4.4.3, we describe an MCMC algorithm which is a modified form of the preconditioned Crank–Nicolson algorithm of Cotter et al. [Cot13]. We use this method to undertake sampling in a different application—within a numerical algorithm itself, rather than in parameter space. There we discuss the reasons for choosing this method, and an assessment of its performance for the task at hand.

We return once more to the subject of MCMC sampling in Section 5.4.1, where we discuss the challenges of sampling from the posterior $p(\theta|Y)$ in the case of the specific type of problem we are tackling there, and consider various strategies for overcoming these difficulties.

1.3 DIFFERENTIAL EQUATION MODELS

The primary objects of our study are inverse problems in which evaluating the likelihood involves the solution of a implicit equation. This almost universally implies that the likelihood cannot be exactly evaluated pointwise. A common case found in many models in the applied sciences is where the defining implicit relation is in the form of an system of ordinary differential equations (ODEs).

We will proceed in a more mathematically precise fashion, and relate our description to the notation introduced in Section 1.1. Consider a time-dependent dynamical system from which instantaneous experimental measurements can be taken. The usual scientific interpretation is to model such a phenomenon as if there were some underlying process $x(t)$, with $x : \mathbb{R} \to \mathbb{R}^d$ a function and $t \in \mathbb{R}$, evolving according to governing laws—of physics, chemistry, economics, biology—able to be succinctly described by a differential equation, such as the following:

$$\frac{\mathrm{d}}{\mathrm{d}t}x(t) = f(x(t), t, \theta) \qquad x(0) = X_0 \tag{1.9}$$

It is typical for such a model to have free parameters $\theta \in \mathbb{R}^q$, the true values of which are unknown, and which make a consequential difference to the model's output. Per the discussion in Section 1.1, we assume that these parameters fully specify the model, and that as such model inadequacy does not come into play. The usual model based on this setup is a regression model which takes the solution function $x_\theta(t)$ of equation (1.9) as the regressor, and the data Y as a set of measurements of this process, each independently corrupted by additive noise.[7]

Note that we have exactly specified the value X_0 of the solution x_θ at $t = 0$. This makes the ODE into an initial value problem (IVP). From the perspective of scientific modelling, this is a useful and widespread type of ODE which can be thought of as tracking the evolution of a time-dependent system with known starting condition. For example, many common predator-prey models typically assume that the initial populations are known, and then evolve according to the dynamics defined by the model [Fre80, §II; Bra11, §1.1].

The initial value is also required mathematically, since without this additional information the ODE is not guaranteed to have a unique solution. Deuflhard & Bornemann [Deu02, §2] point out that, while long-established scientific models may be well-understood to possess unique solutions (it may "appear to be obvious in the particular scientific context"), this mathematical certainty is required in order that new models can be proposed and analysed with confidence. Proposed models which are mathematically badly-behaved are unlikely to be of much use to the scientist. Furthermore, if the aim is parameter inference, we want to be assured that the existence or uniqueness of solutions is not affected simply by varying θ.

1.3.1 *Basic theory of initial value problems*

The discussion above motivates us to state the following central existence and uniqueness result for initial value problems, known as the Picard-Lindelöf Theorem. We first state precisely what we mean by a solution to a differential equation [Cod55, §1] and recall the definition of Lipschitz continuity for real functions.

DEFINITION 1.1 *A solution to the initial value problem* (1.9) *is a continuously differentiable function* $x_\theta : I \to \mathbb{R}^d$ *making* (1.9) *into an identity, where* I *is an interval in* $\mathbb{R}$ *containing the origin.*

DEFINITION 1.2 *A function* $f : \mathbb{R}^a \to \mathbb{R}^b$ *is (globally) Lipschitz continuous if there exists a number* $L_f > 0$ *such that for all* $v, w \in \mathbb{R}^a$*, it holds that* $|f(v) - f(w)| < L_f|v - w|$.

[7] The notation $x(t)$ in equation (1.9) indicates a function considered as an input variable of the differential equation, whereas $x_\theta(t)$ represents the solution function *given* a particular θ. Later, we will often simply write x and x_θ respectively.

THEOREM 1 (PICARD–LINDELÖF THEOREM) *Fix $\theta \in \Theta \subseteq \mathbb{R}^q$ and define the function $f_\theta : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^d$ by $(x, t) \mapsto f(x, t, \theta)$. Now suppose f_θ is Lipschitz continuous in its first argument and continuous in its second. Then the initial value problem* (1.9) *has a unique solution.*

Proof. Rigorous proofs of this technical result are given in Arnol'd [Arn92, §31] and Coddington & Levinson [Cod55, Thm. 3.1]. □

✦ REMARK 1.1 An extension to the Picard–Lindelöf Theorem shows that any solution on the interval I can be continued into the future and the past up to any finite t, which is sufficient for our purposes. Details are available in the given references. ✦

✦ REMARK 1.2 Existence of *at least* one solution is implied by mere continuity as a result of the Peano Existence Theorem [Cod55, Thm. 1.2]. By contrast, the uniqueness result does require Lipschitz continuity—it is easy to find non-Lipschitz functions f for which multiple solutions exist. Conversely it is also true that many common models based on non-Lipschitz functions do possess unique solutions. All of the technical results in this thesis assume Lipschitz continuity, though many of our experiments involve non-Lipschitz functions.[8] It is usually the case that the proof techniques required to give equivalent results in the non-Lipschitz class are more complex. Since this type of analysis is not our central focus, we simply note this fact with this remark. ✦

✦ REMARK 1.3 Though the existence of the solution x_θ to (1.9) has now been established, in the overwhelming majority of cases this solution is not available in closed form. What this means is that in general we cannot explicitly exhibit the mapping $t \mapsto x_\theta(t)$ such that, given a particular value t^*, we can return its exact value $x_\theta(t^*)$. This fact is of supreme importance in the remainder of our study. ✦

Before returning to the statistical context which is our central focus, we make some further minor mathematical observations relating to equation (1.9). We introduced this equation as a first-order, non-autonomous differential equation and the following results account for the apparent lack of generality in this definition.

PROPOSITION 1.1 *An nth order initial value problem of the form*

$$\begin{gathered} \frac{\mathrm{d}^n x}{\mathrm{d}t^n} = f\left(\frac{\mathrm{d}^{n-1}x}{\mathrm{d}t^{n-1}}, \cdots, \frac{\mathrm{d}x}{\mathrm{d}t}, x, t, \theta\right) \\ x(0) = X_0\,, \quad \frac{\mathrm{d}}{\mathrm{d}t}(0) = X_0'\,, \quad \ldots, \quad \frac{\mathrm{d}^{n-1}}{\mathrm{d}t^{n-1}}(0) = X_0^{(n-1)} \end{gathered} \tag{1.10}$$

can be rewritten as a system of first order equations of the form (1.9).

[8] Griffiths & Higham [Gril0, p. vii] assert that the Lipschitz condition "fails to be satisfied by most realistic ODE models".

Proof. Define variables $x_1, x_2, \ldots, x_n$ such that

$$\frac{\mathrm{d}x_1}{\mathrm{d}t} = x_2\,, \quad \frac{\mathrm{d}x_2}{\mathrm{d}t} = x_3\,, \quad \frac{\mathrm{d}x_{n-1}}{\mathrm{d}t} = x_n\,, \quad \frac{\mathrm{d}x_n}{\mathrm{d}t} = f(x_1, \ldots, x_n, t, \theta)$$
$$x_1(0) = X_0\,, \quad x_2(0) = X_0'\,, \quad \ldots, \quad x_n(0) = X_0^{(n-1)}$$

Then the vector $x = (x_1, x_2, \ldots, x_n)^T$ satisfies (1.9). □

PROPOSITION 1.2 *Any non-autonomous IVP (one in which f depends on the variable t independently of x) can be transformed into a system of autonomous IVPs (where an augmented function f^* only depends on t through x).*

Proof. Let $\mathrm{d}x/\mathrm{d}t = f(x, t, \theta)$ with $x(0) = X_0$ as in (1.9). If we write $u = (x, t, \theta)^T$, $f^*(u) = (f(x, t, \theta), 1)^T$ and $u(0) \equiv U_0 = (X_0, 0)^T$, then we have the equivalent autonomous system $\mathrm{d}u/\mathrm{d}t = f^*(u)$. □

These two simple results justify our continued consideration of first-order problems only and, furthermore, allow us to consider only autonomous problems without loss of generality.[9] As a consequence of Proposition 1.2, henceforth we will usually write the function defining the ODE with two arguments, *i.e.* as $f(x, \theta)$. We now return to the statistical model we intend to consider.

1.4 PARAMETER INFERENCE IN ODE MODELS

The statistical model given in Section 1.3 is described mathematically as follows:

$$Y_j = x_\theta(t_{Y_j}) + \varepsilon_j\,, \quad \frac{\mathrm{d}x}{\mathrm{d}t} = f(x, \theta)\,, \quad x(0) = X_0\,, \quad \varepsilon_j \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2) \tag{1.11}$$

Here, the vectors of measurement error ε_i are taken to be realisations of zero-mean Gaussian random variables, and the observation operator G from equation (1.2) is simply the identity operator. Recall that given data Y and model function f, the inverse problem is to ask for the value of θ which best explains Y. A generalisation is possible to the situation in which X_0 is unknown—in this case it should be treated as an additional parameter of the model and appended to θ. Note that this does not contradict the existence result in Section 1.3.1—we would merely be modelling our prior uncertainty about the value of X_0 and attempting to draw inference about it in the standard Bayesian way.

[9] There is a sense in which the generalisation to non-autonomous IVPs would be superfluous anyway, since when discussing the probabilistic interpretation of ODEs from Chapter 2, we assume that t (as the independent variable) is always known. As a result, we do not in fact need to consider non-autonomous IVPs in the service of our main contribution.

This particular parameter inference problem has been extensively studied, due in part to its challenging nature. Discussion of the issues involved, and surveys of various available approaches, are given by Ramsay et al. [Ram07], Campbell [Cam07] and Xue et al. [Xue10]. The standard approach (*cf.* Section 1.1.1) is a form of non-linear least squares, where parameter values are optimised [Bie86] or sampled [Gel96]—depending on whether a non-Bayesian or Bayesian approach is taken—by examining the goodness of fit for each value. The obvious problem with this is that, as noted in Remark 1.3, the function x_θ is not available in closed form, even if θ is fully specified.

Clearly, some numerical method has to be employed in order to approximate the solution x_θ. Such methods fall into two main categories [Gea71, §1.2]. The first type approximate $x_\theta(t)$ by the sum of a finite number of independent functions, for example Chebyshev polynomials [Cle57]. These procedures produce a *function* $z(t)$ approximating the solution $x_\theta(t)$.

The second type produce a discrete path $Z \equiv (Z_0, Z_1, \ldots, Z_N)$ approximating the solution function $x_\theta(t)$ at only the discrete points $(t_0, t_1, \ldots, t_N)$, though interpolation is possible to form a continuous approximation $z(t)$. In this thesis we focus on this second category of iterative numerical method, and survey several classes of them in Section 3.1 in preparation for our own contribution.

Returning to the inverse problem, the naive response is to simply proceed as if the approximate solution were equal to the exact one. However, this approach is fraught with problems. Plainly, the residual sum-of-squares (1.3) calculated using this approach will not be exact. Furthermore, if the numerical approximation Z is not defined at the same time points T_Y as the data Y, interpolation will be required, introducing further inaccuracy.

Both of these issues will result in errors in whichever optimisation or sampling procedure is to be used, with knock-on effects for inference on θ. A study of the latter phenomenon in the Bayesian context, complete with illustrative plots, is given in Conrad et al. [Con16], and we recreate and extend some of these results in Chapter 5. Some quantitative bounds on the total variation distance between posterior probability measures for θ (and bias in derived estimators $\widehat{\theta}$) arising from the numerical approximation of the forward solution $x_\theta(t)$ are given by Donnet & Samson [Don07] and Capistrán et al. [Cap16].

A further problem is the difficulty of actually implementing an algorithm to effectively explore the parameter space Θ, which is required for the reasons given in Section 1.2.1. For a typical ODE-defined forward model, the residual sum-of-squares (1.3) will depend very sensitively—and usually unpredictably—on θ, posing challenges for any search algorithm [Cam07]. Marlin [Mar00; quoted in Ram07] warns that an error level of the order 25% should be expected in parameter estimates inferred using this

approach. Campbell [Cam07, §2.3] gives a concise survey of several methods, both Bayesian and non-Bayesian, specifically targeted at solving this problem.

A parallel issue is that this type of procedure can be computationally very expensive. In order to evaluate the goodness of fit of a single parameter value, a full numerical integration of the entire differential system is required.[10] Much of the literature relating to inverse problems in ODE models is focused on avoiding this expense by defining a cheaper surrogate model which can approximate the result of integrating the system without actually doing so. This general idea goes at least as far back as Varah [Var82] and several papers are still published on it every year. Xue et al. [Xue10] contains numerous references. The common shortcoming is that the error introduced by working with a surrogate system rather than the true one is either ignored or only considered asymptotically.

Approaches which define probabilistic surrogate models also exist—for example Graepel [Gra03] or Calderhead et al. [Cal09]. While these methods are also primarily about reducing computational cost, the probabilistic structure at least allows uncertainties to be considered non-asymptotically, and permits the straightforward treatment of partial-data systems. We will touch on some of these methods in Section 1.4.2. Finally, we note an interesting sequential approach, in which the surrogate model is formed with the use of a particle filter, proposed by Arnold et al. [Arn13].

Despite the discussion above, it is not the case that the only response to the high computational cost of the naive approach is to try to avoid solving the differential system entirely. It also motivates the desire to properly quantify the error introduced by the numerical integration. In this case, the hope is that the scale of the approximation error can in some way be balanced against the other unavoidable sources of error in the problem. This having been done, it may (for example) be possible to justify running the numerical computation at a coarser resolution. This thought forms the starting point for the field of probabilistic numerical methods (PN) [Hen15] and is central to our work. We survey these methods extensively in Chapter 2.

1.4.1 Modelling the numerical uncertainty

In this section we formalise the statistical structure of the parameter inference problem arising from the forward model given in (1.11). We first address the setup corresponding to the naive Bayesian solution described in Section 1.4, and then highlight the way in which our approach will be different.

[10] If an MCMC approach is employed, this problem may be exacerbated by the regular rejection of expensively calculated model solutions—a significant amount of computation can be wasted in this manner.

In any Bayesian analysis, inference must be performed over all unknown variables after conditioning on all known variables. For the time being, let us assume that knowledge of all of θ, X_0 and (the functional form of) f is analogous to complete knowledge of the solution x_θ. Assuming that X_0 and f are both fixed, this is equivalent to the statement that knowledge of the regressor function x_θ of the model is equivalent to knowledge of the parameter θ. This gives the Bayesian model decomposition

$$p(\theta, \sigma|Y) \propto L(Y; \theta, \sigma)p(\theta)p(\sigma) \tag{1.12}$$

In accordance with (1.11) and the assumptions in the previous paragraph, the term $L(Y; \theta, \sigma)$ is Gaussian and proportional to $\prod_{j=1}^{M} \exp(-(Y_j - x_\theta(t_{Y_j}))^2/2\sigma^2)$.

We now return to our primary contention—that the premise that x_θ is fully specified by (θ, X_0, f) is incorrect and that as a result the setup (1.12) is inadequate. In an ODE model, x_θ *cannot* be fully specified, due to the approximation error which inevitably arises in its numerical evaluation, and we must therefore be uncertain about its true value. The Bayesian approach should be to model this uncertainty rather than ignore it. We therefore specify a non-degenerate probability measure $p(x|\theta)$ which mathematically represents our lack of knowledge about the true value of x_θ. The result is a hierarchical structure with separate distributions representing the measurement error of the data Y and the approximation error of the regressor x_θ. The rest of this thesis will be concentrated on the subtle issue of characterising the latter measure.[11]

As we will come to see, several different approaches are possible for treating $p(x|\theta)$. In each case, we will be required to take certain decisions affecting the algorithms we use to approximate it. Such decisions could include choosing the step-size h in an iterative integrator, or indeed which method we choose in the first place. We represent these algorithmic degrees of freedom, broadly construed, by a parameter ϕ. In other words, while θ constitutes the parameters of the model, ϕ are the parameters representing the numerical procedure used to approximate x. (The forthcoming sections will provide much greater detail on this abstract description.)

If the algorithmic degrees of freedom were not fixed in advance, ϕ would theoretically need to be included as a full part of the posterior model. However, we typically *will* fix ϕ in advance of running a simulation. This is because we are interested in understanding the effect of a specified numerical method on the solution of the forward problem

[11] An interesting half-and-half approach is suggested by Tarantola [Tar05, §1.3], who uses a Bayesian setup to directly characterise $L(Y; \theta, \sigma)$ while explicitly acknowledging that this distribution represents two different sources of error—measurement error and 'theoretical uncertainties', the latter corresponding to model inadequacy, numerical error, etc. They posit that these two sources of error "generally produce uncertainties with the same order of magnitude" and models them as Gaussian distributions scaled in accordance with this principle, but theoretical justifications for this statement are not provided.

and, eventually, the inverse problem. A data-conditioned posterior $p(\phi|Y)$ is not a meaningful object—experimental data manifestly does not provide information about the numerical method to be used in a statistical procedure implemented to analyse it after its collection. As a result, we give the problem the following structure:

$$p(\theta, \sigma|Y, \phi) \propto L(Y; x, \sigma)p(x|\theta, \phi)p(\theta)p(\sigma). \tag{1.13}$$

✦ Remark 1.4 For convenience, we will often refer to these distributions verbally as follows: $p(\theta, \sigma|Y, \phi)$ is the 'full posterior', $L(Y; x, \sigma)$ is the 'data likelihood', $p(x|\theta, \phi)$ is the 'measure over the numerical solution' or similar, while the remaining distributions are parameter priors. ✦

At this stage it is worth reinforcing the conceptual point that $p(x|\theta, \phi)$ is a probability measure representing our inescapable lack of knowledge of x_θ given the computational tools at our disposal, represented by ϕ. The solution function x_θ is a deterministic object—not random in a truly stochastic sense—yet this setup as a description of epistemic uncertainty is completely valid within the Bayesian framework. In fact, not only is it valid, but it is to be formally treated in exactly the same way as uncertainty arising from true randomness. This principle is discussed—in the related but simpler context of round-off error occurring in the digital representation of real numbers—in exactly this way as far back as Hull & Swenson [Hul66] and Henrici [Hen62, §1.6].

We will examine and compare a number of approaches to forming a statistical model for the measure $p(x|\theta, \phi)$. Each is the product of a slightly different branch of academic research, and the aims of each—along with their respective shortcomings—are also different. Our contribution, which starts from Chapter 3, is the extension and generalisation of one of these paradigms.

1.4.2 Gradient matching

Before covering in detail in Chapter 2 the way in which the emerging theory of probabilistic numerical methods treats this problem, we take a brief diversion to consider a separate but related strand of research. As we will see, the ultimate shortcoming of this approach will be instructive in understanding the way in which PN is different.

The concept of defining a distribution like $p(x|\theta, \phi)$ as part of a Bayesian analysis of the form (1.13) has not always taken as its starting point the desire to model numerical error. A series of articles—Calderhead et al. [Cal09], Dondelinger et al. [Don13], Barber & Wang [Bar14] and Macdonald et al. [Mac15]—are all concerned primarily with surrogate modelling, by which is meant a desire to perform parameter inference without actually solving the ODE.

The central principle in these papers is to match the gradients between the full and surrogate models and, following a Bayesian approach, include the parameters of the surrogate model as part of the inference process.[12] A consequence of this is the introduction of a probability distribution $p(X|\theta, \phi)$ akin to the measure over numerical solutions found in (1.13), though we again stress that the origin of this is not the explicit modelling of numerical uncertainty. (In fact, we will argue that it cannot hope to do this even in theory.) The actual construction is subtly different across the four articles—we give a rough description closest to that in Dondelinger et al. [Don13].

Firstly we note that the theory proceeds in a discrete setting, so the object of interest is X—the set of exact values of $x_\theta(t)$ at times $t_0, \ldots, t_N$. Starting from the familiar premise that X is not exactly calculable, a Gaussian process (GP)[13] prior $p(X|\psi)$ is defined for it, and the data Y used to derive a posterior GP interpolation [Ras06, §2.2], giving $p(X|Y, \sigma, \psi)$. This is done in the standard Bayesian way, namely

$$p(X|Y, \sigma, \psi) \propto p(Y|X, \sigma)p(X|\psi) \tag{1.14}$$

In this expression, as in equation (1.2), σ is the variance of the Gaussian measurement error. ψ represents the covariance structure of the Gaussian process prior $p(X|\psi)$. Due to the linearity of the differential operator, a Gaussian process model for the derivative $\dot{X}$ can then immediately be written down as $p(\dot{X}|\psi)$—the required transformations for the mean and covariance functions under this operation are given by Adler [Adl81, Thm. 2.2.2]. (In the context of ODE models, this fact was previously exploited by Solak et al. [Sol03].)

In parallel, the GP-generated posterior values for X can be passed to the function $f(\cdot, \theta)$ defining the ODE. Along with the significant assumption that these are distributed around the true derivatives with Gaussian error of variance γ, a second model for the derivative can be written down as $p(\dot{X}|X, \theta, \gamma) = \mathcal{N}(f(X, \theta), \gamma \cdot \mathbb{I}_d)$. These two models can then be combined by simply multiplying the densities together—this procedure is called 'product-of-experts' [Hin02].

[12] This contrasts with earlier surrogate modelling approaches which are either based on deterministic surrogates, such as splines or basis function expansions [Var82; Ram07], or are probabilistic but require regularisation parameters to be set in advance [Gra03].

[13] We omit a long introduction to Gaussian processes, since they are now very well known. The standard reference for foundational Gaussian process methodology in the context of Bayesian inference is Rasmussen & Williams [Ras06]. However it is will be useful for us to note here a flexibility in terminology which we will continue to use later when describing our own contribution in Chapter 3. A Gaussian process is defined as a distribution over functions, so that $x(t) \sim \mathcal{GP}(m(t), k(t, t'))$ has covariance function $k(\cdot, \cdot)$. However the articles referenced in this section employ the slight abuse $X \sim \mathcal{GP}(m, K)$ for discrete X, where in this setting K is a matrix of covariance function values calculated from the kernel (the 'Gram matrix'). Of course, this latter formulation is simply a finite-dimensional Gaussian and could be notated as $\mathcal{N}(m, K)$, but the specific kernel-derived covariance structure means it is convenient to continue using the language of Gaussian processes.

The resulting distribution is $p(\dot{X}|X, \theta, \psi, \gamma)$. Multiplying by the GP prior $p(X|\psi)$ and then marginalising over the derivatives $\dot{X}$ gives $p(X|\theta, \psi, \gamma)$. This is simply our familiar distribution $p(x|\theta, \phi)$, with x considered discretely and with our general algorithm parameter ϕ having been taken to include both ψ and γ, respectively the algorithm parameters for the two constituent parts of the model. We can then use this as part of the procedure to infer θ, as in (1.13).

This idea—and minor variations of it suggested in the other referenced articles—has one significant methodological shortcoming, which in our view recommends an alternative approach. Closer inspection reveals that the fit of the surrogate model occurs having conditioned on the data Y. This is theoretically problematic, since the structure of the model (1.13) now contains a cycle, in the sense of probabilistic belief networks [Kol09, §3] or Bayesian networks [Pea85]. In words, the data is being used to fit the hyperparameters of a likelihood model, which are subsequently used to assess data fit. An obvious corollary of this point is that integrating an IVP in isolation—the forward model alone, in a setup in which no experimental data is available—is not possible using this method.

This reasoning explains why the distribution $p(X|\theta, \phi)$ cannot here be taken to be a pure model of numerical error since, as explained in Section 1.4.1, inference about the uncertainty in an entirely numerical procedure should not be dependent on experimental data.[14]

We have seen that, as an approach to reducing computational expense, there is some merit in working with a surrogate model and attempting to simply minimise the discrepancy between it and the true solution. However if we wish to tackle the problem of modelling the numerical error itself, we require a different approach. The latter strand of thinking is a core feature of the PN paradigm, which we explore in detail in the next chapter.

[14] Separately to the discussion in this section, the specific statistical setup in Barber & Wang [Bar14] also throws up an interesting consistency issue as a by-product of the way the joint model connecting the two expressions for the gradient is formed (this paper does not adopt the product-of-experts approach of Calderhead et al. [Cal09] and Dondelinger et al. [Don13] for this task). Macdonald et al. [Mac15] motivate their own work with a scrupulous analysis of this issue, exposing the source of the problem. Careful thought reveals that this inconsistency may also apply to the methods we introduce in Chapter 3—indeed, the resolution of this problem is one of the main motivations for the the improved concept to be described in Chapter 4. We return to this specific point in Section 4.1.1.

2

PROBABILISTIC FORMULATION OF ODE SOLVERS

In this chapter, we consider in detail a framework for the probabilistic modelling of the uncertainty arising from numerical computations. In the context of ODE solvers, this idea was introduced by Skilling [Ski91], building on the more general ideas of Diaconis [Dia88] and O'Hagan [OHa92]. Skilling argues that the process of integrating an ODE should return a probability distribution over numerical solutions rather than simply a point value, and suggests a way to do it within a Bayesian framework. Note that, unlike in the case of surrogate models discussed in Section 1.4.2, this distribution is explicitly intended to represent the uncertainty arising from the application of the approximate numerical method.

The core idea is that outputs of a numerical procedure are considered as 'data' and the desired distribution over the inaccessible numerical solution $p(x|\theta, \phi)$ as a posterior formed by combining this data with a functional prior for x. This is approached in the standard Bayesian way—namely by defining a likelihood and performing an inversion using Bayes' Theorem. The difference between this setup and that described by equation (1.14) is that a PN procedure is undertaken *without* conditioning on experimental data, preserving the directionality of the hierarchical model (1.13), and avoiding the clearly unsatisfactory situation where experimental data is used to draw inference about the effect of a purely numerical procedure.

The way in which this numerical data is collected and then incorporated—equivalent to specifying the likelihood model—throws up a number of subtleties, and in this chapter we will consider some approaches in detail. The issues to be considered

include foundational ones—do algorithms which purport to return a measure over the numerical solution have appropriate technical properties?—but also ones of interpretability and usefulness.

The first of these might ask if can we make statements about limiting behaviours which would satisfy both numerical analysts and statisticians. For example, does the measure contract properly in the limit of infinite computation? Is the output in the *non-limiting* case satisfactory, measured against some appropriate metric—does the finite-computation behaviour match some expected and interpretable notion of uncertainty?

The second might ask whether we can construct algorithms which are based on classical iterative numerical methods, collecting our data by following the path of such a method. Furthermore, could we do this such that, for instance, the mean of the derived probability measure would correspond to the path of the classical algorithm. Can we make the description of uncertainty encoded in $p(x|\theta, \phi)$ correspond in some intuitive way to the numerical error of the classical method? The complex nature of ODE theory, and the plethora of numerical methods available to solve them, means that these are highly non-trivial questions.[15]

Questions of this nature have evolved into an active research area of the field known as probabilistic numerical methods (PN) [Hen15]. In the context of the probabilistic solution of differential equations, the main frameworks have been developed—somewhat in parallel—by articles such as Hennig & Hauberg [Hen14], Chkrebtii et al. [Chk13; Chk16] and Schober et al. [Sch14]. We will consider these approaches in detail in Section 2.1. A further strand, originated by Conrad et al. [Con16], has hitherto been considered in the same group as these, though in our view it takes an approach different enough to be considered separately. Our own contributions extend this latter paradigm, so we also consider it in detail, in Section 2.2 .

In passing, we note that similar ideas are also being used to study other numerical algorithms, for example in the topics of integration (*e.g.* quadrature/cubature), linear algebra (*e.g.* matrix inversion/solution of linear systems) and optimisation (*e.g.* gradient descent). The common theme is the same—that the output of numerical algorithms is treated as data and quantities of interest are inferred from them using statistical techniques.

A survey article summarising this work has been published by Hennig et al. [Hen15], and a review with comprehensive references is given by Cockayne et al. [Coc17]. A

[15] Contrast this situation with the modelling of round-off error in Hull & Swenson [Hul63] mentioned in Section 1.4.1, in which the error ε_k in a real number rounded to k decimal places is modelled with reasonable heuristic justification as uniformly distributed over the interval of precision, *i.e.* $\varepsilon_k \sim \mathcal{U}[-5 \times 10^{-(k+1)}, 5 \times 10^{-(k+1)}]$.

connection between these algorithms is also made by Hennig et al. [Hen15]—in particular they state a 'general recipe' for a probabilistic numerical algorithms consisting of a 'generative model' (which encodes the likelihood model for the numerical data) and a 'decision rule' (which determines how the algorithm generates new data).

2.1 REVIEW OF PROBABILISTIC ODE SOLVERS

Diaconis [Dia88] was amongst the first to suggest the use of Bayesian probability theory to interpret the output of numerical algorithms as data and perform inference on quantities of interest within this framework.[16] The focus of his paper is on a Bayesian approach to numerical quadrature. Skilling [Ski91] fleshed out the concept in the specific context of ODE solvers for the first time, in particular by considering the values produced by repeated evaluation of the objective function $f(\cdot, \theta)$ as the procedure's data, then formally defining a prior on the space of solution functions and a likelihood encoding the way in which the generated data should be assimilated. This general setup has been picked up by a number of recent papers, which we now consider closely.

✦ AUTHOR'S NOTE The order of presentation of the following literature review does not strictly correspond to the chronology of the articles explored. The aim is not to provide a comprehensive historical record, nor focus on issues of precedence. For example, the publication date of a journal article is in some cases significantly later than the release of its first version online. Furthermore as the field is seeing rapid advances even at the time of writing, the 'graph of dependencies' between these articles and various parts of our own contribution in later chapters is complicated. We will try to make these clear where appropriate. Nevertheless despite its description as a 'review', it is important to stress that the novel contributions of this thesis occurred concurrently, rather than strictly after, some of the work included in this section. ✦

HENNIG & HAUBERG (2014)

Hennig & Hauberg [Hen14] effectively translate the ideas of Skilling [Ski91] into the modern language of Gaussian processes. Here, and in the reviews of the following articles, we will strip away the specific application, and simplify each method to one-dimensional first-order IVPs. We will also translate into our own notation.

Firstly, a Gaussian prior measure ρ_0 is set up on the function space $\mathcal{X}$ containing possible solution functions x. Since differentiation is linear and Gaussian measures

[16] Similar ideas are also present in the earlier work by Larkin [Lar72]. While the formal statistical structure of the problem is not developed there, the concept is advanced (at least in principle) of deriving a probability measure over the space of possible solutions of a numerical procedure.

are closed under linear transformations, it is helpful to think of this as a joint Gaussian measure over the space $(\mathcal{X}, \dot{\mathcal{X}})$ encapsulating prior information about both x and its first derivative $\dot{x}$—the reason for this will become clear in the next paragraph. The particular choice made in the article is to specify ρ_0 to be a squared-exponential Gaussian process defined by

$$\begin{aligned} x(t) \sim \rho_0 &\equiv \mathcal{GP}(\mu_0(t), k(t,t')) \\ \mu_0(t) = X_0\,, \qquad & k(t,t') = \alpha \exp\left(-\frac{(t-t')^2}{2\lambda^2}\right) \end{aligned} \tag{2.1}$$

Here, in writing $\mu_0(t) = X_0$ we mean that μ_0 is a constant function taking the value X_0 for all t. This mean function serves to condition on the initial value X_0. Then, for each $i \in [0, N-1]$, an 'observation' $F_{i+1} \equiv f(\mu_i(t_{i+1}), \theta)$ is collected, where the evaluation point $\mu_i(t_{i+1})$ is the mean of the measure ρ_i—representing the current state of knowledge of x—at time t_{i+1}.[17]

Following the key idea in Skilling [Ski91], a Gaussian error model is then assumed for these observations with respect to the true solution derivative $\dot{x}(t)$. This once again exploits the linearity of differentiation, and the likelihood for each point follows as

$$F_{i+1}|\rho_i \sim \mathcal{N}(\dot{x}(t_{i+1}), \Lambda_{i+1}) \tag{2.2}$$

where Λ_{i+1} is a variance parameter in derivative space. Finally, this likelihood term is combined with the prior in a Bayesian inversion to give the $(i+1)$-times updated posterior ρ_{i+1} over $\mathcal{X}$ by $\rho_{i+1} \propto p(F_{i+1}|\rho_i) \cdot \rho_i$.

After N points have been collected and the measure ρ_N determined, its variance at time t is then interpreted as the uncertainty in $x(t)$ arising from the computation. Since everything is Gaussian, the mean and variance of this distribution can be given explicitly.

The parameters $\{\Lambda_i\}_{i=1}^N$ and λ are set using very rough calibration arguments—the topic of method calibration is one of our primary concerns and will be discussed further in Section 4.3. In fact, in Hennig et al. the calibration procedure is undertaken by conditioning on experimental data Y and hence no method is proposed which would be able to calibrate the uncertainty in the forward model alone. This means the methodological inconsistency described in Section 1.4.2 persists.

The method of choosing the input points $t_1, \ldots, t_N$ at which to evaluate $f(x(t), \theta)$ is also only briefly remarked upon. A pre-determined grid is suggested, but the ordering of the calculations—which, due to the way the algorithm is designed, will have an impact on the eventual inference—is not addressed.

[17] An extension by Hauberg et al. [Hau15] generalises to the case where the function is evaluated with noise; *i.e* the observation F_{i+1} is a non-degenerate random variable.

Chkrebtii et al. (2013/2016)

In the PhD thesis by Chkrebtii [Chk13] and the subsequent paper [Chk16], a variant algorithm is given which also builds on Skilling's original formulation but contrasts from Hennig et al.'s paper by adopting an explicitly sequential approach and a clearer principle for setting the derivative observation model (2.2). Furthermore a proof is given, absent from Hennig et al., that the mean of the measure ρ_N defined over the numerical solutions contracts in the $N \to \infty$ limit to the true solution $x(t)$.

In this scheme, the initial functional prior is defined over the derivative space $\dot{\mathcal{X}}$ then integrated to give the prior on the solution space $\mathcal{X}$, rather than the other way round. As noted in the previous section, the two formulations are equivalent, so we persist with the ρ notation. Here, a different covariance structure is suggested—one defined in terms of the convolution of a square-integrable kernel function $R_\lambda : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ with itself. (In the paper, several possible kernels R_λ are posited and compared.) Specifically, the prior model is given as

$$\begin{gathered} \dot{x}(t) \sim \rho_0 \equiv \mathcal{GP}(\dot{\mu}_0(t), \dot{k}_0(t,t')) \\ \dot{\mu}_0(t) = * \,, \qquad \dot{k}_0(t,t') = \alpha \int_{\mathbb{R}} R_\lambda(t,z) R_\lambda(t',z)\, dz \end{gathered} \tag{2.3}$$

and then,[18] integrating,

$$x(t) \sim \mathcal{GP}\left(\int_0^t \dot{\mu}_0(z)\, dz, \int_{z=0}^{t} \int_{z'=0}^{t'} \dot{k}_0(z,z')\, dz\, dz' \right) \tag{2.4}$$

Here, as before, the zero subscripts denote that these are the GP measures at the opening step—having conditioned on the initial value but with no further updates having been made. The procedure continues sequentially, using the same concept of treating function evaluations F_i as observations—though here they are called 'model interrogations'. A key difference is that the next datum F_{i+1} is determined by *sampling* from the predictive posterior measure $\rho_i(t_{i+1})$ and passing this sample to the function f, rather than simply evaluating at the current mean.

As in Hennig et al., a Gaussian error model is assumed for the derivative, but here the covariance is taken to be the uncertainty of the derivative in the current update of the model. Thus we have

$$F_{i+1} | \rho_i \sim \mathcal{N}\left(\dot{\mu}_i(t_{i+1}), \dot{k}_i(t_{i+1}, t_{i+1}) \right) \tag{2.5}$$

This approach resolves the problem of how to set Λ_i in (2.2), though naturally the resolution is merely a modelling choice (and indeed is justified by the authors only on

[18] The $*$ in (2.3) arises because the issue of incorporating the initial value X_0 when the prior is initially defined over the derivative requires an extra constraint. Details are straightforward and are given in Supplement D.1 of [Chk16] and as such we omit the full discussion.

those terms). A further adjustment is that the likelihood model is centred around the predictive mean (which is known) rather than the true derivative (which is unknown). As before, this error model is then used to update the predictive distribution via $\rho_{i+1} \propto p(F_{i+1}|\rho_i) \cdot \rho_i$.

The result is a posterior Gaussian measure $\rho_N^{[1]}$, arrived at by conditioning on one complete set of sampled observations $F_{1:N}^{[1]}$. The square-bracketed superscripts refer to the fact that this is the first run of the algorithm. Multiple runs produce multiple such measures $\rho_N^{[2]}, \ldots, \rho_N^{[K]}$, each different since the sampled observations F_i are different each time. The measure ρ_N formed by the uniform mixture of these individual measures then represents the posterior over the solution function $x(t)$.

A convergence result is also given, which states that as the step-size h ($\equiv t_{i+1} - t_i$), spread parameter λ and prior variance scale α all tend to zero, the posterior distribution returned by the probabilistic integrator converges in mean (L^1) to the true solution. In other words, if x_θ is the true solution, it holds for all $t \in [0, T]$ that

$$\mathbb{E}_F|\rho_N(t) - x_\theta(t)| \to 0 \qquad \text{as } h, \lambda, \alpha \to 0 \tag{2.6}$$

✦ Remark 2.1 In the preceding discussion, we introduced the more concise notation ρ_i to represent measures over the function space $\mathcal{X}$ of solution functions $x(t)$. This allowed us to concisely write ρ_0 for the prior—equivalent in the notation used earlier to $p(x|\phi)$; and ρ_N for the posterior—equivalent to $p(x|F_1, F_2, \ldots, F_N, \phi)$.[19] In these expressions, the algorithm parameter ϕ is taken to include the prior modelling choices made in (2.1) and (2.3), including the form of covariance function $k(\cdot, \cdot)$ and hyperparameters α and λ, as well as the hyperparameter Λ in equation (2.2).

The way in which the posterior measure ρ_N relates to the 'measure over numerical solutions' $p(x|\theta, \phi)$ in our original model formulation (1.13) can be seen by noting that all the information about θ gathered by the algorithm is carried by the N model interrogations $F_1, \ldots, F_N$ and thus the posterior $\rho_N \equiv p(x|F_1, F_2, \ldots, F_N, \phi)$ is in fact the same distribution as $p(x|\theta, \phi)$.

✦ Remark 2.2 We reiterate that the terms *prior*, *likelihood* and *posterior* in this paradigm refer to an 'inner' Bayesian procedure focused on inferring the distribution of $x(t)$, in contrast to the 'outer' Bayesian inversion required to solve the inverse problem represented by the model (1.13). ✦

[19] F_i can be interpreted as $\bigcup_{k=1}^{K} F_i^{[k]}$ in the case of Chkrebtii et al., where multiple instantiations are run, each generating their own set of model interrogations.

The algorithms compared

One consequence of the differing ways that the two schemes assimilate the data F into their statistical models for x is that the process required to output an interpretable posterior is of a fundamentally different character in the two cases. In the case of Hennig et al., the method of collecting the F_i by evaluating $f(\cdot, \theta)$ at the mean of the current measure means a full distributional posterior $p(x(t)|F, \phi) \equiv \rho_N$ is available. Once the prior ρ_0 and evaluation ordinates $t_1, \ldots, t_N$ are set and the algorithm proceeds, the F here is fully specified by the subsequent deterministic calculation.

By contrast, the sampling step is key to the method of Chkrebtii et al. Its effect is that one run of the algorithm returns a measure $p(x(t)|F^{[k]}, \phi)$ which is a posterior for the solution conditional on a single realisation k of numerical data F. Multiple realisations—computed by repeating the calculation with different random seeds, in the manner of Monte Carlo—are combined to give a mixture distribution which is then taken to be the F-conditioned posterior ρ_N. This is undoubtedly a richer model, with a much greater degree of feedback from the ODE itself—in statistical terms, a more informative likelihood—resulting in a non-parametric posterior measure.

Naturally, the Monte Carlo repetitions add computational expense. The question of whether this is a price worth paying is simply a version of the usual speed/accuracy trade-off familiar to algorithm designers everywhere. That said, a theoretical point noted by Kersting & Hennig [Ker16] is that, since the contraction of measure (2.6) is an asymptotic result, the nature of the relationship between the empirical posterior and the true solution is not clear after a finite number of Monte Carlo repetitions.

Schober et al. (2014)

While the contributions of Hennig et al. and Chkrebtii et al. describe ODE-solving algorithms which return probability distributions instead of point estimates, a key shortcoming in both works is the lack of a relationship between their constructions and established, classical IVP solvers. In particular, the choice of covariance function in the Gaussian process priors entirely determines the solutions generated, and yet none of the choices in these two papers correspond in any interpretable sense to classical ODE methods.

This issue was considered by Schober at al. [Sch14], who pointed out that particular choices of covariance function in the Gaussian process prior ρ_0 for $x(t)$ give rise to distributional solutions for $x(t + h)$ with mean exactly equal to the output of a classical Runge–Kutta method. They supply these covariances explicitly for orders 1 (corresponding to the forward Euler method; see Section 3.1.1), 2 and 3—though in the latter two cases this requires an additional trick of taking the limit of the initial time $t_0 \to -\infty$. The covariances are based on repeatedly integrating the covariance of

Brownian motion $k_{\text{Br}}(t,t') = \sigma^2 \min(t,t')$ for some $\sigma^2 > 0$ so that, for example, the following once-integrated prior gives rise to a forward Euler step:

$$\begin{aligned} x(t) &\sim \rho_0 \equiv \mathcal{GP}(0, k_{\text{Br}}^{\uparrow 1}(t,t')) \\ k_{\text{Br}}^{\uparrow 1}(t,t') &= \int_{z=0}^{t} \int_{z'=0}^{t'} k_{\text{Br}}(z,z') \, \mathrm{d}z \, \mathrm{d}z' \\ &= \sigma^2 \left(\frac{\min^3(t,t')}{3} + |t-t'| \frac{\min^2(t,t')}{2} \right) \end{aligned} \tag{2.7}$$

This approach only guarantees a correspondence with the classical method for a single time-step, since all subsequent time-steps can be viewed as solving a modified IVP resulting from the uncertain output of the first step. This point is made in the paper by Schober et al. [Sch14] themselves. Because of this restriction to a single time step, they are unable to make claims about the global error properties of their probabilistic solver. The covariance structure is chosen specifically to align with the Runge–Kutta estimate after one step and—but for this criterion—is of an unusual non-standard form.[20] This observation calls into question the usefulness of the method to model the second and subsequent steps of a longer integration.

Kersting & Hennig (2016) & Schober et al. (2018)

In the article by Kersting & Hennig [Ker16], extended and formalised by Schober et al. [Sch18], an alternative approach is considered generalising in some respects the earlier work of Schober et al. [Sch14]. Once again a prior model is proposed for the solution $x(t)$ and updated based on function evaluations, but here the construction is explicitly cast as a stochastic filtering problem [Øks98, §6].

In these papers, $x(t)$ and its first q time derivatives are assumed to follow a Gauss–Markov process that solves a pre-specified stochastic differential equation (SDE). The parameter q—along with the SDE—can be chosen in order to make the output of the filter tally in expectation with a classical numerical integrator, in the manner of the Runge–Kutta integrator of Schober et al. [Sch14]. In brief, by writing x and its derivatives as a vector $(x, \dot{x}, \ddot{x}, \ldots, x^{(q)})^T$ and forming the Itô-type relation (2.8), standard results [Sär06, Thm. 2.9] can be used to give the mean and covariance of x.

$$\begin{pmatrix} \mathrm{d}x \\ \mathrm{d}\dot{x} \\ \vdots \\ \mathrm{d}x^{(q)} \end{pmatrix} = \begin{pmatrix} 0 & 1 & \cdots & 0 \\ & \ddots & \ddots & \\ & & 0 & 1 \\ a_0 & a_1 & \cdots & a_q \end{pmatrix} \begin{pmatrix} x \\ \dot{x} \\ \vdots \\ x^{(q)} \end{pmatrix} \mathrm{d}t + \begin{pmatrix} 0 \\ 0 \\ \vdots \\ \sigma^2 \end{pmatrix} \mathrm{d}B_t \tag{2.8}$$

[20] This is even clearer in the case of the twice– or three times–integrated Brownian motion covariances given for higher-order Runge–Kutta methods, which involve higher and higher powers of $\min(t,t')$ and $\max(t,t')$.

Here $\sigma^2 > 0$, B_t is a standard Brownian motion and $(x(0), \dot{x}(0), \ddot{x}(0), \ldots, x^{(q)}(0))^T$, the vector of initial values of x and its derivative, is assumed known. Choosing $a_j = 0$ for all $j \in [0, q]$ models $x(t)$ as a q-times integrated Brownian motion, as in Schober et al. [Sch14]. An alternative choice, that of taking $a_q = -c$ for some positive $c > 0$, instead results in a q-times integrated Ornstein-Uhlenbeck process and is considered in Magnani et al. [Mag17].

The prior having been set, data is collected by the algorithm in the form of evaluations of $f(\cdot, \theta)$, which are assimilated into the model as observations of the first derivative $\dot{x}$, in a manner similar to Chkrebtii et al. [Chk16]. The fact that all distributions are Gaussian means this can be done as in a Kalman filter [Kal60]. At each step, the algorithm returns a mean and covariance for the solution $x(t)$ consistent with all data collected so far. This can be updated iteratively, meaning that the calculation does not get more expensive as the integrator advances and more data are collected. The Kalman update equations can be found in Särkkä [Sär13, §4].

Relating this concept back to the sequential model updating approach of Chkrebtii et al., it can be seen to constitute a structurally similar model, though both model update and data generation steps are in this case performed in accordance with the Kalman equations rather than by sampling predictive distributions. Details are given in Kersting & Hennig [Ker16, §2.2] and Schober et al. [Sch18, §2.4].

The papers which have developed this paradigm give various connections to classical methods, though they are not always completely intuitive. For $q = 1, 2$ and 3, the integrated Brownian motion prior tallies after one iteration (in mean) with the q-stage Runge–Kutta method, as described in the previous section [Sch14]. For $q = 1$, and with an additional function evaluation at the end of each step, the method is equivalent to the trapezoidal rule [Sch18, Prop. 1] and for $q = 2$ it is equivalent to a third order Nordsieck method, but only after an initial stabilisation period [Sch18, Prop. 2]. In each case, however, these connections require the algorithm to assume that the data assimilated into the model should be error-free. This is an unrealistic assumption, since the input to f is inexact at all but the first iteration, and as such the output cannot be expected to represent the value of the derivative $x'(t)$ exactly.

The primary benefit of the filtering construction is that, with careful programming, it can be competitive in terms of speed with standard algorithms. This is due to the fact that at each iteration, a single function evaluation is made, and all remaining computation is made up of straightforward linear operations. This minimal cost overhead is stressed by the authors of these papers—this is a settlement to the cost/accuracy trade-off in favour of the former, a point that they explicitly recognise. Finally, we note that several rigorous convergence statements for integrators constructed using the filtering approach are presented in a very recent paper by Kersting et al. [Ker18].

2.1.1 *Discussion*

The common theme in all of the approaches discussed so far is the functional nature of the random variable $x(t)$ upon which inference is to be drawn. In other words, while the algorithms are structurally different in the specific ways they incorporate numerical data from the ODE, the input and output in all cases is a measure over some space $\mathcal{X}$ of functions, of which $x(t)$ is assumed to be an element. This feature is formalised in Schober et al. [Sch18], in which the authors state an express desire to avoid "an analysis gap between statistical and numerical computations."

While this view is consistent and attractive, a number of practical issues arise, with no consensus on their resolution. Firstly, how does one approach the thorny issue of prior choice ρ_0 for the ODE solution?

The usual Bayesian answer to this question falls into one of three categories. The first is to suggest that the prior should attempt to formally encode meaningful knowledge about the variable of interest—in the case of an ODE solution function x, this could relate to its high-level features, such as its smoothness or whether or not it is expected to be periodic.[21]

Alternatively high-level heuristic arguments may be used to motivate prior choice—examples of this approach include uninformative or maximum-entropy priors [Jay03, §12]. Finally the choice can be made for mathematical convenience—the canonical example in elementary Bayesian analysis being conjugate priors. It is not clear how either of these would be applied to the complex models being considered here.

Those PN methods presented in Section 2.1 which aim to match the posterior mean $\mathbb{E}(x|\theta, \phi)$ to the output of a particular classical method [Sch14; Sch18] choose priors specifically to achieve this . The earlier methods [Hen14; Chk16] barely consider the interpretation of their prior choice at all, and consequently say little about its eventual effect on the posterior distribution over solutions.

Neither approach seems satisfactory if the prior choice is primarily intended to reflect meaningful information about the variable at hand. However, it is arguably even more difficult to encode such informative prior assumptions in distributional form than in a typical (experimental) Bayesian analysis, due to the absence of convincing intuitive reasoning for how to model the epistemic uncertainty in a numerical procedure.

Schober et al. [Sch18] have recently tackled this issue head on, by explicitly reinterpreting the posterior measure merely as an "external analysis of the effects of the [prior] assumptions". They specifically claim that the aim of their method is to act as an 'infer-

[21] Hennig et al. [Hen15, §2(c)] argue that in the numerical setting, this sort of higher-level information should in principle be available to the numerical algorithm, at runtime. Whether and how this could be implemented in a practical manner is unclear and not explored.

ence agent', a view which formally absolves the practitioner from having to consider what information is encoded in the prior measure. Notably, the prior chosen for the method in that paper—multiply-integrated Brownian motion—implies that the solution x, if considered as a draw from the posterior measure ρ_N, has properties such as differentiability class at odds with the true solution of the ODE x_θ. The 'inference agent' interpretation allows them to circumvent this inconsistency, since under this view the solution is not required to be considered as a sample from the posterior measure.[22]

A second problematic issue is calibration. We will discuss this in much greater depth in Section 4.3 but in short, what we mean is the process of setting the tuning parameters ϕ of the algorithms used to derive the posterior measures within each framework. A complete Bayesian analysis would require that each such hyperparameter be included as a full part of the inference procedure. This is both computationally challenging—some of the reasons were considered in Section 1.4.1—and furthermore throws up the non-trivial question of how to choose hyperpriors. As a result, a variety of non-Bayesian approaches to calibration are suggested in the articles summarised in this section, including hyperprior optimisation—otherwise known as empirical Bayes [Rob56]—which in some cases involves the use of a data-dependent marginal distribution.

It is important to clarify that these considerations do not in themselves invalidate any of the methods explored in this section. However, they do call into question the merit of obstinately insisting on a purist Bayesian approach. To be clear, what we mean here is not that Bayesian reasoning itself is at fault, but that non-Bayesian approaches can be justified for practical purposes if strict Bayesian approaches result in the sorts of intractable difficulties just described. With this in mind, we turn to a fundamentally different paradigm for constructing uncertainty estimates over solutions of IVPs.

2.2 RANDOMISED NUMERICAL METHODS

Conrad et al. (2016)

The article by Conrad et al. [Con16] proposes an entirely different construction—one which modifies existing non-probabilistic ODE solvers to create probabilistic ones, by introducing stochastic perturbations ξ_i at each iteration. The scale of perturbation that ensures the method remains convergent is rigorously stated and proved.[23] As such, this framework addresses the issue of the theoretical performance of probabilistic integrators in relation to their non-probabilistic counterparts for the first time.

[22] Some further discussion on the subtleties of interpretation of the prior and posterior distributions in probabilistic numerical methods is presented by Cockayne et al. [Coc17, Appendix A].

[23] The convergence we refer to is in the $h \to 0$ sense of numerical analysis—a formal definition is given as Definition 3.1 in Chapter 3.

Though structurally different to the methods in Section 2.1, the fundamental aim—to return a probability measure over the solution of a numerical procedure in a way that represents computational uncertainty—remains the same.

The construction proceeds as follows. Treating the problem in a discrete setting, the algorithm produces a sequence $Z \equiv Z_{0:N}$ of values approximating $X \equiv X_{0:N} \equiv x(t_{0:N})$, with $F_{i+1} \equiv f(Z_{i+1}, \theta)$ constituting the observations and Z_{i+1} determined by an iterative relation depending on previous estimates Z_i and F_i. This procedure closely resembles a standard iterative IVP-solving algorithm—the main difference being that the iterative relation generating the sequence Z is non-deterministic. (To stress this point we will sometimes write $Z(\xi)$ to reflect the dependence of Z on the set of random perturbations ξ.)

Note that, in contrast to the approaches in Section 2.1, there is no attempt here to continuously assimilate the generated values into a functional model for the unknown continuous solution x or $\dot{x}$ during the run of the algorithm. Instead, the justification for the method comes *post hoc* in the form of a convergence theorem bounding the worst-case expected squared-error $\max_i \mathbb{E}_\xi \|Z_i(\xi) - X_i\|^2$. The precise statement of this result requires some further definitions and technical background, and we therefore defer it to Section 3.2.

This approach is intuitive, allowing for modified versions of standard algorithms which inherit known useful properties, and giving provable probabilistic error bounds for the output. It is effectively a form of randomised numerical method, where the introduced stochasticity at each step aims to reflect the error of the underlying method. The resulting posterior distribution is non-parametric, and it relies on Monte Carlo sampling to give empirical approximations to it. It is only defined on the grid $t_{0:N}$, though in some limited contexts it is possible to extend the formulation to the entire (continuous) range $[t_0, t_N]$, thus giving a posterior output more qualitatively similar to those given by the methods in Section 2.1 [Con16, Thm. 2.2]. We will expand on these observations in Chapter 3.

A key difference in philosophy is that, from a statistical viewpoint, the method explicitly defines a distribution over numerical solutions Z rather than an uncertainty centred around the true solution x (or X). The relationship of the measure over Z to the true solution X is then guaranteed by the convergence analysis. The interpretation of Z as a sample from a discrete measure representing the uncertainty in X is then thought of as an explicit modelling choice. Our contention is that this is a setup qualitatively much more similar to classical IVP integrators, and one better suited to analysing the error in the numerical procedure itself—subject of course to the implementation of an effective calibration process.

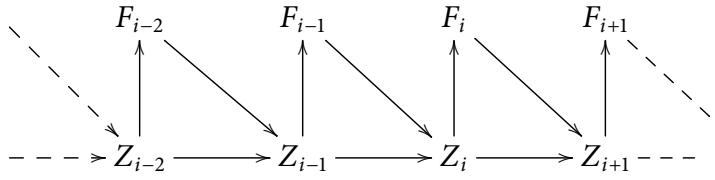


Figure 2.1: Bayesian network representation of an indicative section of the joint distribution $p(Z_{1:N}, F_{0:N-1}|\theta, \phi)$ as decomposed in Equation (2.10).

To see the structure of the model, consider the distribution representing numerical uncertainty $p(x|\theta, \phi)$, from equation (1.13). In accordance with the discussion in this section, we are now considering the alternative distribution $p(Z|\theta, \phi)$. We decompose this joint distribution to make explicit the sequential nature of its calculation.

$$p(Z_{1:N}|\theta, \phi) = \int p(Z_{1:N}, F_{0:N-1}|\theta, \phi)\, \mathrm{d}F_{0:N-1} \tag{2.9}$$

$$= \int \left[\prod_{i=0}^{N-1} p(F_i|Z_i, \theta) p(Z_{i+1}|Z_i, F_i, \phi) \right] \mathrm{d}F_{0:N-1} \tag{2.10}$$

The term $p(F_i|Z_i, \theta)$ simply describes the deterministic transformation of applying the function $f(\cdot, \theta)$—this could be written in distributional form as $\delta_{F_i}(f(Z_i, \theta))$. The term $p(Z_{i+1}|Z_i, F_i, \phi)$ is given as a Gaussian centred around the output of any classical single-step IVP solver, with its variance controlling the scale of the perturbations—this variance is constrained in accordance with the main convergence result of Conrad et al. [Con16, Thm. 2.2], which we give in Section 3.2 as Theorem 2. We will sometimes refer to $p(Z_{i+1}|Z_i, F_i, \phi)$ and its later generalisations as the 'stepping' or 'step-forward' distribution.

The Bayesian network representation [Pea85] given in Figure 2.1 highlights the dependence structure of the joint distribution $p(Z_{1:N}, F_{0:N-1}|\theta, \phi)$ visually. When we come to generalising and modifying the statistical model for the sequence $Z_{1:N}$ in later chapters, this representation will be a useful way of contrasting the different approaches.

Note that, using the terminology of Hennig et al., [Hen15, §3a], the two constituent components of the decomposition in the right-hand side of (2.10) respectively correspond to the 'decision rule'—how the algorithm generates a new data-point F_i; and the 'generative model'—which encodes the likelihood model for Z.

A further key conceptual point is that there is genuine stochastic randomness in the method, as in Chkrebtii et al. [Chk16] but unlike the other integrators described in Section 2.1. This means that in order to actually evaluate the distribution over numerical solutions given in equation (2.9), multiple runs are required with different random seeds $\omega \in \Omega$, and a Monte Carlo estimator must then be employed to marginalise out

this stochasticity. We can express this (in a non-rigorous way) by extending equation (2.10) to

$$p(Z_{1:N}|\theta,\phi) = \iint \left[\prod_{i=0}^{N-1} p(F_i|Z_i,\theta)p(Z_{i+1}|Z_i,F_i,\phi,\xi)\right] \mathrm{d}F_{0:N-1}\,\mathrm{d}\mathbb{P}_\xi \tag{2.11}$$

A typical run involves sampling an element $\omega^{[k]}$ of the sample space Ω, using this to generate a realised set of Gaussian perturbation $\xi_1(\omega^{[k]}),\ldots,\xi_n(\omega^{[k]})$, then evaluating the decomposition in (2.11) to give a sample $Z^{[k]}$ from $p(Z|\theta,\phi,\xi)$. This process is then repeated with multiple different seeds $\omega^{[k]}$, and the introduced randomness subsequently marginalised out, resulting in ensemble of samples from which Monte Carlo estimates for statistics of $p(Z|\theta,\phi)$ can be derived.[24]

An experimental analysis of this method is also undertaken in by Conrad et al. One crucial consideration is how to calibrate the scale of the perturbations which, as the authors themselves state, completely controls the apparent uncertainty indicated by the method. This issue arises due to the presence of an unspecified constant α in the permitted variance implied by Theorem 2. In the paper, an approach is suggested which seeks to replicate the expected scale of the global error in the underlying classical method. In our own extension to this method, to be introduced in Chapter 3, we adopt a similar approach—we therefore defer the full details of their calibration scheme to Section 4.3.

2.3 OTHER RECENT DEVELOPMENTS

Before continuing to a complete exposition of our own contributions, we briefly note here several other pieces of work related to the probabilistic solution of ODEs, some of which are very recent.

Firstly, in relation to the theoretical underpinnings of the functional approach to inferring the ODE solution x, an even stricter framework to that proposed by Schober et al. [Sch18] is developed by Cockayne et al. [Coc17, §2.3]. This paper defines so-called 'Bayesian probabilistic numerical methods' to be those in which the prior-likelihood-posterior structure of the inner inversion central to the methods outlined in Section 2.1 can be interpreted in a rigorous measure-theoretic sense as a Bayesian procedure.

[24] In fact, there is no methodological problem with the random perturbations ξ depending on the parameter θ and, in the method we introduce in Chapter 4, we require this generalisation. In this case, equation (2.11) can be further generalised to

$$p(Z_{1:N}|\theta,\phi) = \iint \left[\prod_{i=0}^{N-1} p(F_i|Z_i,\theta)p(Z_{i+1}|Z_i,F_i,\phi,\xi,\theta)\right] \mathrm{d}F_{0:N-1}\,\mathrm{d}\mathbb{P}_{\xi|\theta} \tag{2.12}$$

Specifically, they object to the likelihood formulations in those articles—since the nature of the 'data' F is numerical, they contend that the information it provides should be considered noise-free—unlike the usual case of noise-corrupted experimental data. They argue that this causes Bayes' Theorem to be ill-defined and suggest an alternative construction to circumvent this issue. While they provide examples for other classes of numerical problems, *none* of the ODE methods described in this thesis satisfy their strict definition.

Very recent work by Wang et al. [Wan18] substantiates the claim that such methods do not yet exist, though they provide an interesting proof-of-concept Bayesian probabilistic ODE solver which may be applicable for the sub-class of ODEs to which a solvable Lie algebra of transformations can be associated [Blu02, §2-3]. For further details, the reader is directed to the aforementioned papers.

Within the filtering paradigm, rigorous convergence statements of the type given in Conrad et al. [Con16]—and also later in this thesis—were not previously available. The first results of this kind have recently been published by Kersting et al. [Ker18]. Even more recently, Tronarp et al. [Tro18] have begun to develop a general framework in which the filtering-based methods described here manifest as special cases.

Finally, an idea with parallels to the paradigm of randomised numerical methods—which we will cover in detail starting in Chapter 3—is introduced in recent paper of Abdulle & Garegnani [Abd18]. The core idea here is to randomise the *time step* h of an iterative algorithm, rather than the output estimate, and in doing so with repetition generate output measures analogous to our $p(x|\theta, \phi)$. Implemented in a particular way, this concept additionally enables interesting higher-level features of the underlying dynamical system to be preserved, such as symplecticity.

3

RANDOMISED METHODS FOR THE PROBABILISTIC SOLUTION OF ODES

The framework suggested by Conrad et al. [Con16] and introduced in Section 2.2 is the first that attempts to construct probabilistic ODE solvers from classical, deterministic ones. In simple terms, the approach centres around the introduction of stepwise stochasticity to iterative integrators. These random perturbations are intended to cumulatively reflect the uncertainty arising in the underlying method, but evidently their scale and the way in which they are introduced must be constrained in a way that does not materially alter the necessary analytical properties of the algorithm.

In this chapter we discuss these methods in greater detail, including several novel contributions. Before doing so, we provide a concise introductory survey of the families of classical initial value problem solvers we will consider, giving important definitions as we go. Furthermore, we will summarise some of the key technical properties of these methods—such as convergence—as they are thought of in conventional numerical analysis. This will allow us to analyse the modified, probabilistic versions that will be discussed later using the correct language, and compare them to long-accepted theoretical benchmarks.

3.1 CLASSICAL NUMERICAL METHODS FOR IVPS

In this thesis, we restrict attention to iterative algorithms based on the 'difference method' [Gea71, §1] *i.e.* those where the solution is approximated at a sequence of discrete mesh points, with estimates calculated sequentially on the mesh. (We briefly

mentioned the other main class of methods, based on orthogonal function expansions, in Section 1.4.) In fact, we restrict ourselves further to the class of constant step-size, non-adaptive algorithms—those where the scheme to be used is fixed in advance and remains unchanged throughout the run.

Methods with step-size control based on solution tolerances, or which adapt in other ways during the course of the solve, are ubiquitous in modern computational packages for ODEs, but a comprehensive survey of such solvers is beyond our scope. A large body of literature exists on this rich subject [Ise09; Stu98; Hai08; But08; Jac09; Gea71; Atk09; Pal09; Sül03]. We begin our survey with the following central concept:

DEFINITION 3.1 (CONVERGENCE) *Recall that $X_i \equiv x(t_i) \in \mathbb{R}^d$ is the exact solution of the differential equation* (1.9) *at time t_i, and Z_i is a numerical approximation to X_i generated by some time-stepping numerical method. Such a method is convergent if for every $t_{\text{end}} > 0$ and Lipschitz function $f(\cdot, \theta) : \mathbb{R}^d \to \mathbb{R}^d$, it holds that*

$$\lim_{h \to 0} \max_i \|Z_i - X_i\| = 0 \tag{3.1}$$

where i runs through all discrete time-steps in the range $0, 1, \ldots, N$ with $N \equiv \lfloor h^{-1} t_{\text{end}} \rfloor$, and $\|\cdot\|$ is the standard Euclidean norm. Furthermore, the method is convergent of order p if there exists an integer $p \geq 1$ and constant $C > 0$ (independent of h but possibly dependent on t_{end}) such that

$$\max_i \|Z_i - X_i\| \leq C h^p \tag{3.2}$$

✦ REMARK 3.1 As noted by Iserles [Ise09, §1.2], convergence is an obligatory minimum requirement for a time-stepping numerical scheme. This thought arises from the reasonable assumption that more computation should result in increasing proximity to the correct solution—a non-convergent method violates this principle. Thus, while such further properties as stability, rate of convergence, consistency etc. may be balanced against one another to compare the merits of different schemes, if a method is not at the very least convergent, it is *de facto* useless. As a result, when we introduce probabilistic integrators that are based on classical methods later in this work, we provide proofs of (stochastic) convergence analogous to these definitions. ✦

✦ REMARK 3.2 Stuart & Humphries [Stu98, §3.1] write that the types of questions that are typically asked of numerical methods for differential equations can be divided into *(i)* those examining the relationship between the large N behaviour of the discrete scheme and the large t behaviour of the IVP itself, in the limit $h \to 0$, and *(ii)* those asking which qualitative dynamic features of the IVP are replicated by the numerical scheme for a wide range of values of h. The first of these concerns convergence statements and the like, which our analysis focuses on. The second concerns issues

of stability. While we consider implicit methods later and remark on their apparent improved stability, we do not aim to give a rigorous treatment of the stability properties of the probabilistic methods we introduce. Our instinct is that this calls for the heavy machinery of stochastic analysis and we hope future research will be able to shed light on this subtle topic. We will revisit this discussion briefly in Section 6.2.3. ✦

3.1.1 *One-step methods*

The simplest IVP integrators combine the numerical estimate of the current point Z_i and an appropriate local approximation of the function f, to give an estimate for the next point Z_{i+1}. The forward Euler scheme is the most well known. This is given by

$$Z_{i+1} = Z_i + hF_i \tag{3.3}$$

Here, $F_i \equiv f(Z_i, \theta)$ is calculated by passing the *estimate* of the current point Z_i to the function $f(\cdot, \theta)$. Two issues are immediately apparent. The first is that, *even if* the current point is known exactly, the estimate for Z_{i+1} will not in general be exact unless $f(\cdot, \theta)$ is constant over the entire time interval $[t_i, t_{i+1}]$. This means that an error is introduced with every application of this formula, termed local truncation error.

The second issue is that since $Z_i \neq X_i \equiv x(t_i)$ on all steps after the first—and hence is *not* exact—it is also the case that $F_i \neq f(x(t_i), \theta)$. In other words, the estimate for the value of $f(\cdot, \theta)$ at time t_i is based on its evaluation at an input value known to be inexact. In this way, earlier numerical inaccuracies propagate to all subsequent iterations. It is from this fact that the concept of accumulated or global error arises.

In order to give more precise definitions of these quantities, and understand how the errors produced by a numerical method relate to its convergence properties, we introduce the following definitions:

Definition 3.2 (Flow maps) *For an initial value problem* $\mathrm{d}x/\mathrm{d}t = f(x(t), \theta)$; $x_\theta(0) = X_0$, *and given* $t > 0$, *the flow map* $\Phi_t : \mathbb{R}^d \to \mathbb{R}^d$ *is the function mapping the initial value to the solution at time* t, *i.e.* $\Phi_t(X_0) = x_\theta(t)$.

For an iterative numerical method, the numerical flow map $\Psi^h : \mathbb{R}^d \to \mathbb{R}^d$ *is the function mapping the input at the current iteration to the output at the next, so that* $Z_{i+1} = \Psi^h(Z_i)$. *For example, the numerical flow map of the forward Euler method* (3.3) *with step-size* h *is defined by* $\Psi^h(Z_i) = Z_i + hf(Z_i, \theta)$.

Definition 3.3 (Errors) *The local truncation error* e_i *of an iterative integrator advancing one step of length* h *from* Z_i *can be written in terms of the flow map and numerical flow map as*

$$e_i(h) = \Psi^h(Z_i) - \Phi_h(Z_i) \tag{3.4}$$

and the global or accumulated error E_i at time $t_i \equiv ih$ as

$$E_i(h) = (\Psi^h \circ \overset{i}{\cdots} \circ \Psi^h)(X_0) - \Phi_{ih}(X_0) \tag{3.5}$$

where $\Psi^h \circ \overset{i}{\cdots} \circ \Psi^h$ denotes the i-times repeated composition of the map Ψ^h. Particularly during convergence arguments, the norms $\|e_i\|$ and $\|E_i\|$ of these quantities will be of primary interest.

Definition 3.4 (Order) *An iterative numerical method is of order p if for every analytic $f(\cdot, \theta)$ and fixed Z_i, the local truncation error $\|e_i\|$ is $\mathcal{O}(h^{p+1})$.*

The forward Euler scheme is of order one. This can be straightforwardly determined by rewriting its defining relation (3.3) as $Z_{i+1} - [Z_i + hf(Z_i, \theta)] = 0$, substituting the true solution values $x(t_i)$ for Z_i and $x(t_{i+1})$ for Z_{i+1}, then performing a Taylor expansion around t_i. This gives

$$\begin{aligned} x(t_{i+1}) - \big[x(t_i) + hf(x(t_i), \theta)\big] \\ &= \big[x(t_i) + h\dot{x}(t_i) + \mathcal{O}(h^2)\big] - \big[x(t_i) + h\dot{x}(t_i)\big] \\ &= \mathcal{O}(h^2) \end{aligned} \tag{3.6}$$

The relationship between local and global errors is addressed by the following result.

Proposition 3.1 *Let $f(\cdot, \theta)$ be a Lipschitz-continuous function with Lipschitz constant L_f and assume that the numerical method with numerical flow map Ψ^h has local truncation error $\|e_i\| = \mathcal{O}(h^{p+1})$ for all $1 \leq i < N$. Then, given any such i, the global truncation error $\|E_i\| = \mathcal{O}(h^p)$.*

Proof. We give as an example the proof for the forward Euler method, though the generalisations to other one-step methods, as well as to multistep methods, are straightforward. Details are available in *e.g.* Quarteroni et al. [Qua00, §11.6.3]. The approach is an inductive bounding argument on the sequence e_i of local errors. We write

$$\begin{aligned} \|E_i\| \equiv \|Z_i - X_i\| &= \|Z_{i-1} + hf(Z_{i-1}, \theta) - X_i\| \\ &= \|Z_{i-1} - X_{i-1} + X_{i-1} + hf(X_{i+1}, \theta) - X_i + hf(Z_{i-1}, \theta) - hf(X_{i-1}, \theta)\| \\ &\leq \|Z_{i-1} - X_{i-1}\| + \|X_{i-1} + hf(X_{i+1}, \theta) - X_i\| + \|hf(Z_{i-1}, \theta) - hf(X_{i-1}, \theta)\| \\ &\leq \|E_{i-1}\| + \|e_{i-1}\| + hL_f\|E_{i-1}\| \\ &= \|e_{i-1}\| + (1 + hL_f)\|E_{i-1}\| \\ &= \|e_{i-1}\| + \|e_{i-2}\|(1 + hL_f) + \|e_{i-3}\|(1 + hL_f)^2 + \cdots + \|e_1\|(1 + hL_f)^{i-2} \\ &\leq Ch^{p+1}\big(1 - (1 + hL_f)^{i-1}\big)\big(1 - (1 + hL_f)\big)^{-1} \\ &= CL_f^{-1}h^p\big((1 + hL_f)^{i-1} - 1\big) \end{aligned}$$

where in the penultimate line we have used the fact that $\|e_{i-1}\| \leq Ch^{p+1}$. Now noting that $t_i = ih$, we have

$$\|E_i\| \leq CL_f^{-1}h^p\left(\left(1+\frac{t_{i-1}L_f}{i-1}\right)^{i-1}-1\right) \leq CL_f^{-1}\exp(t_{i-1}L_f)h^p \leq C'h^p \qquad \square$$

We can now finally state the result connecting the scale of a method's local errors to its convergence. Consistency and zero-stability are technical conditions which are true for all integrators considered in this thesis—precise definitions can be found in Quarteroni et al. [Qua00, §11.3]

PROPOSITION 3.2 *An consistent and zero-stable iterative numerical method of order p, applied to a Lipschitz function $f(\cdot,\theta)$ over a fixed, compact time interval $[0, t_{\text{end}}]$ and with given fixed initial condition X_0, is convergent of order p.*

Proof. This statement immediately follows from the Definitions 3.1 and 3.4, and Proposition 3.1. $\square$

It is now clear that the forward Euler method, whose local truncation error is $\mathcal{O}(h^2)$, has global error which is $\mathcal{O}(h)$.

Another important distinguishing feature of iterative numerical methods concerns the manner in which Ψ_h depends on state estimates Z. The forward Euler scheme, which only requires the current value Z_i as input, is called explicit. An example of an implicit single step method is the backward Euler method, given by

$$Z_{i+1} = Z_i + hF_{i+1} \tag{3.7}$$

Rewriting (3.7) in terms of flow maps as $\Psi^h(Z_i) = Z_i + hf(\Psi^h(Z_i),\theta)$, we see that in this case the iteration cannot be applied directly since the numerical flow map is defined implicitly. Thus a second, nested numerical scheme—typically a fixed-point iteration or a Newton–Raphson scheme [But08, §225]—is required to output Z_{i+1}.

Finally, we note that higher-order one-step methods also exist—the trapezoidal rule, another implicit method, is defined by $\Psi^h(Z_i) = Z_i + \frac{1}{2}h(f(Z_i) + \Psi^h(Z_i))$. This method advances in a single step but has local error $\mathcal{O}(h^3)$ and global error $\mathcal{O}(h^2)$.

3.1.2 *Runge–Kutta methods*

A further one-step method with its own name is the mid-point rule given by[25]

$$Z_{i+1} = Z_i + hf(Z_i + \tfrac{1}{2}hf(Z_i, t_i, \theta), t_i + \tfrac{1}{2}h, \theta) \tag{3.8}$$

Unpicking equation (3.8) shows that the calculation contains an intermediate stage, whose role is to improve the estimated value of the function f to be used in the final

evaluation. The mid-point rule has local error $\mathcal{O}(h^3)$, and is the simplest example of a more general class of *multi-stage* methods where intermediate calculation is undertaken to improve the final estimate. Another name for this class is Runge–Kutta methods, the most well-known being RK4, with four intermediate stages given sequentially by

$$\begin{aligned}
\chi_1 &= f(Z_i, t_i, \theta) \\
\chi_2 &= f(Z_i + \tfrac{1}{2}h\chi_1, t_i + \tfrac{1}{2}h, \theta) \\
\chi_3 &= f(Z_i + \tfrac{1}{2}h\chi_2, t_i + \tfrac{1}{2}h, \theta) \\
\chi_4 &= f(Z_i + h\chi_3, t_i + h, \theta) \\
Z_{i+1} &= Z_i + \tfrac{1}{6}h(\chi_1 + 2\chi_2 + 2\chi_3 + \chi_4)
\end{aligned} \tag{3.9}$$

With these intermediate—and iterative—refinements of the estimate of f, a method with local error $\mathcal{O}(h^5)$ arises. Runge–Kutta methods of even higher order exist but require disproportionately more stages to achieve these orders of convergence—for an explicit RK method of order p, the number of stages required is strictly greater than p for all $p \geq 5$ [But08, §236]. Implicit multi-stage methods can also be defined, though they can be very expensive to implement if each intermediate stage requires an internal iteration to determine the value it passes forward to the next [Sül03, §12.12]. An extensive survey of Runge–Kutta methods, including some interesting historical background, is given by Butcher & Wanner [But96].

3.1.3 *Multistep methods*

The core principle of multistep methods is that already calculated estimates $Z_{\leq i}$ and $F_{\leq i}$ from earlier time points can be used to improve the estimate at the next time-point t_{i+1}. By contrast, the one-step methods explored in Sections 3.1.1 and 3.1.2 discard all information before the present time-point t_i. If the multistep iteration uses a linear combination of past values, the method is termed a linear multistep method (LMM).

In their most general form, an s-step linear multistep method is defined by an iterative relation of the form

$$\sum_{j=-1}^{s-1} a_j Z_{i-j} = h \sum_{j=-1}^{s-1} b_j f(Z_{i-j}, \theta) \tag{3.10}$$

It is clear from this formulation that if $b_{-1} = 0$, then Z_{i+1} is obtained only from past state estimates $Z_{i-s+1:i}$ and derivative estimates $F_{i-s+1:i}$ and is thus explicit. If $b_{-1} \neq 0$ then the method is implicit. In general, the determination of appropriate coefficients

[25] For this section, we have temporarily reverted to the more general three-argument version of the function f defining the ODE, *i.e.* $f(x(t), t, \theta)$. This is because Runge–Kutta methods involve calculating estimates of f at fractional time-points, though if we restrict attention to autonomous systems $\mathrm{d}x/\mathrm{d}t = f(x, \theta)$, this is only manifested through the changing first argument.

a_j and b_j is of critical importance, and desirable properties such as stability,[26] consistency and convergence all flow from these choices [Sül03, §12]. Particular choices of a_j and b_j give rise to well-studied methods such as backward differentiation formulae, Taylor series methods, Nyström methods, and Nordsieck methods [Hai08, §III.1].

Our study focuses on a particularly important and well-used subclass of LMMs for which $a_{-1} = 1$, $a_0 = -1$, and $a_j = 0$ for all other j. This category of LMMs is known as the Adams family of integrators. These methods were introduced in 1883 by Adams [Ada83] and popularised by the 1926 book of Moulton [Mou26]. The construction is based on extrapolating from a polynomial interpolation of past function estimates. Since these methods form a core part of our contribution in this thesis, we explore them in detail.

Adams–Bashforth method

The s-step Adams–Bashforth (AB) method—the common name for the class of explicit Adams family algorithms—calculates Z_{i+1} by constructing the unique order $s-1$ polynomial $P_i(u) \in \mathbb{P}_{s-1}$ interpolating the previously-calculated function evaluations $F_i, F_{i-1}, \ldots, F_{i-s+1}$. This polynomial is given by Lagrange's method [Abr65, §25.2] as

$$P_i(u) = \sum_{j=0}^{s-1} L_j^{0:s-1}(u) F_{i-j} \qquad L_j^{0:s-1}(u) = \prod_{\substack{k=0 \\ k \neq j}}^{s-1} \frac{u - t_{i-k}}{t_{i-j} - t_{i-k}} \tag{3.11}$$

The $L_j^{0:s-1}(u)$ are known as Lagrange polynomials of order s and have the property that $L_p^{0:s-1}(t_{i-q}) = \delta_{pq}$. They form a basis for the space of polynomials $\mathbb{P}_{s-1}$ known as the Lagrange basis. Having constructed the interpolating polynomial (3.11), the Adams–Bashforth iteration then proceeds by writing the integral version of the initial value problem (1.9) as

$$x(t_{i+1}) - x(t_i) \equiv \int_{t_i}^{t_{i+1}} f(x(t), \theta)\, \mathrm{d}t \tag{3.12}$$

This exact terms in this expression are then replaced by their numerical approximations, and finally the function under the integral is approximated by extrapolating the polynomial $P_i(u)$ to time t_{i+1}. This gives

$$Z_{i+1} - Z_i \approx \int_{t_i}^{t_{i+1}} P_i(u)\, \mathrm{d}u = h \sum_{j=0}^{s-1} \beta_{j,s}^{AB} F_{i-j} \tag{3.13}$$

[26] There are several different (but related) concepts of stability for multistep methods: zero-stability, absolute stability, relative stability, A-stability and confusingly, just 'stability'. While we refer to some of these in isolation during our later analysis, we refrain from giving extended definitions here. The book by Hackbusch [Hac14] specifically concerns the analysis of stability for numerical methods, while Palais & Palais give a (partial) hierarchy of implication of these different notions in the case of multistep ODE solvers [Pal09, Appendix 1]. Though stability is not the primary focus of our study, we return to some of the open questions relating to the stability of our methods in Section 6.2.3.

The coefficients $\beta_{j,s}^{AB}$ are positive real numbers called the Adams–Bashforth coefficients of order s and are given by

$$\beta_{j,s}^{AB} = \frac{1}{h}\int_{t_i}^{t_{i+1}} L_j^{0:s-1}(u)\,\mathrm{d}u = \frac{1}{h}\int_0^h L_j^{0:s-1}(t_i+u)\,\mathrm{d}u \tag{3.14}$$

These coefficients are independent of h and satisfy $\sum_{j=0}^{s-1}\beta_{j,s}^{AB} = 1$. Their values for $1 \leq s \leq 5$ are listed in table 3.1. Note that the 1-step Adams–Bashforth method is the same algorithm as the forward Euler method.

It can be shown—by methods similar to those described in Section 3.1.1 for one-step methods—that the local truncation error of the s-step Adams–Bashforth method is $\mathrm{O}(h^{s+1})$ and the global error $\mathrm{O}(h^s)$.

Adams–Moulton method

Adams–Moulton (AM) methods are the name given to the class of implicit Adams family integrators. They are constructed using similar principles to the Adams–Bashforth methods, except that this time the order s polynomial $Q_i(u) \in \mathbb{P}_s$ interpolates the $s+1$ points $F_{i+1}, F_i, F_{i-1}, \ldots, F_{i-s+1}$. The resulting integral extrapolation equation is thus an implicit one, with the unknown Z_{i+1} appearing on both sides. Carrying out the equivalent calculations as in equations (3.11)–(3.13) gives another set of coefficients $\beta_{j,s}^{AM}$—the Adams–Moulton coefficients—the values of which are also listed in table 3.1.

As with implicit one-step methods, the implicit nature of the multistep Adams–Moulton method means that the value of Z_{i+1} implied by the equivalent relation to (3.13) can only be calculated approximately. As a consequence, Adams–Moulton methods are used in conjunction with an explicit Adams–Bashforth method of one order lower, in a 'predictor-corrector' arrangement. In this situation, a predictor value Z_{i+1}^* is calculated using an Adams–Bashforth step, this is then used to estimate $F_{i+1}^* = f(Z_{i+1}^*, \theta)$, and finally an Adams–Moulton step uses this value as its approximation to F_{i+1}, thereby calculating Z_{i+1}.

Note that in all of the schemes described in this section, the first $s-1$ steps must be made using a different method, since evidently there is not yet enough history to deploy an s-step method during this initialisation phase. It is intuitively clear—and straightforward to prove—that the method used for this initialisation must be of at least equal order to the primary method in order for the global error to behave as desired. Typically a high-order Runge–Kutta scheme is used for these starting values, though Hairer et al. [Hai08, §III.7] remark that self-starting multistep codes which simply start with the order one method implemented with very small initial step sizes are also prevalent.

Adams–Bashforth

No. of steps s	Global order	Coefficient $\beta^{AB}_{j,s}$ of F_{i-j} for $j =$ 0	1	2	3	4	Error constant
1	1	1					1/2
2	2	3/2	−1/2				5/12
3	3	23/12	−4/3	5/12			3/8
4	4	55/24	−59/24	37/24	−3/8		251/720
5	5	1901/720	−1387/360	109/30	−637/360	251/720	95/2888

Adams–Moulton

No. of steps s	Global order	Coefficient $\beta^{AM}_{j,s}$ of F_{i-j} for $j =$ −1	0	1	2	3	Error constant
0	1	1					−1/2
1	2	1/2	1/2				−1/12
2	3	5/12	2/3	−1/12			−1/24
3	4	3/8	19/24	−5/24	1/24		−19/720
4	5	251/720	323/360	−11/30	53/360	−19/720	−3/160

Table 3.1: Coefficients and error constants of the Adams–Bashforth and Adams–Moulton integrators of orders 1 to 5. Values for methods of higher order, and details of the algorithm used to derive them, are given by Butcher [But08, §241-244].

✦ Remark 3.3 We highlight here an unfortunate convention which, while occasionally confusing, is widespread in numerical analysis and which we have therefore elected to follow. For multistep methods, the step number s is taken to be equal to the total number of time ordinates *at or before the current point* t_i at which derivative values are used during the iteration. Thus the first-order Adams–Bashforth method (the forward Euler method) is a '1-step' method, whereas the first-order Adams–Moulton method (the backward Euler method) is a '0-step' method. Viewed another way, for $s \geq 1$, the s-step Adams–Bashforth and $(s+1)$-step Adams–Moulton methods both go equally far back. ✦

3.1.4 General linear methods

The difference between the higher-order integrators in the Runge–Kutta family and multistep methods is that in the former case, intermediate stages consisting of additional evaluations of $f(\cdot, \theta)$ contribute to the final estimate for Z_{i+1} but do not themselves possess any theoretical guarantees on closeness to the exact solution,

whereas in the latter case evaluations of $f(\cdot, \theta)$ calculated in previous steps (which do possess such guarantees) are reused.

These classes differ in their regions of stability [Ise09; Sül03] and their robustness to stiff or badly-behaved ODEs. They are also very different in the scale of computation they require—if stability is a problematic factor in a given problem, it may be that more expensive codes are justified. Suffice it to say that both multistage and multistep methods are important constituents of the numerical analyst's toolbox.

A natural extension is to consider both approaches simultaneously. This framework, called General Linear Methods and introduced by Butcher [But06], allows for maximum generality and includes all methods above as special cases, including coupled predictor/corrector types. A rich and elegant theory exists to describe these methods in terms of matrices—called Butcher tableaux—and analysis of the algebraic properties of these matrices provides deft methods of proof for concepts such as convergence and stability of the corresponding integrator. A more detailed description is beyond the scope of this thesis—a comprehensive reference is the book by Jackiewicz [Jac09]—but we note in passing our hope that some of the ideas we will consider later may eventually be applicable in this more general setting too.

3.1.5 Error indicators

The classical definition of numerical error for iterative numerical ODE solvers was given in Definition 3.3. Proofs verifying the valid convergence of each scheme typically compare Taylor series expansions of the true and approximate solutions and seek to cancel as many low order terms as possible, in the manner described for the forward Euler method in Section 3.1.1. The first remaining non-zero term is, of course, the leading-order term of the local truncation error. For example, the 3-step Adams–Moulton method has local truncation error $e_i = -\frac{19}{720} h^5 x^{(5)}(\tau_i)$, where $x^{(5)}(\tau_i)$ is the fifth derivative of $x(t)$ at some point τ_i in the interval $[t_{i-3}, t_{i+1}]$. The coefficient $-\frac{19}{720}$ is called the error constant of the method and these are also listed in table 3.1.

Thinking probabilistically, we can immediately identify issues with this formulation. Clearly, the exact local truncation error is not known—if it were, we would just subtract it off and get a more accurate solution. However it is even difficult to generate less precise information about this quantity, such as lower or upper bounds, since we can not typically say anything meaningful about the boundedness of the fifth derivative of x, nor of the exact value of τ. Furthermore, everything here is based on asymptotic theory—with only the first residual Taylor series term considered—and is therefore unreliable for large h [Gea81, §3.2].

Taking an even broader view, it is difficult to know how to interpret an error—an inherently unknowable quantity—represented by a single number in this way. This is particularly true for a statistician, who is typically used to a richer characterisation of error. As we argued in Chapter 1, a Bayesian would expect that an unknown quantity such as this should be assigned a probability distribution, and then some model proposed to infer it. This thought motivates the development of all probabilistic numerical methods—both those we have already seen, and those we are about to propose.

3.2 PROBABILISTIC ONE-STEP METHODS

Having surveyed the main classical families of iterative IVP solvers, we now take up where we left off in Section 2.2, describing in greater detail the randomised integrator concept of Conrad et al. [Con16]. In subsequent sections, we give a detailed account of our own contributions generalising this paradigm. The algorithm introduced in that paper centres around the following convergence result. Before stating it, we give the stochastic equivalent to Definition 3.1 that it employs.

DEFINITION 3.5 (STOCHASTIC CONVERGENCE) *Recall that $X_i \equiv x(t_i) \in \mathbb{R}^d$ is the exact solution of the differential equation* (1.9) *at time t_i. Let $Z_i(\xi)$ be a random variable representing an approximation to X_i generated by some stochastic time-stepping numerical method with random seed ξ, in accordance with the discussion of Section 2.2.*

Such a method is convergent in mean-square if for every $t_{\text{end}} > 0$ and Lipschitz function $f(\cdot, \theta) : \mathbb{R}^d \to \mathbb{R}^d$, it holds that

$$\lim_{h \to 0} \max_i \mathbb{E}_\xi \|Z_i(\xi) - X_i\|^2 = 0 \tag{3.15}$$

where i runs through all discrete time-steps in the range $0, 1, \ldots, N$ with $N \equiv \lfloor h^{-1} t_{\text{end}} \rfloor$, and $\|\cdot\|$ is the standard Euclidean norm. Furthermore, the method is convergent in mean-square to order p if there exists an integer $p \geq 1$ and constant $C > 0$ (independent of h but possibly dependent on t_{end}) such that

$$\max_i \mathbb{E}_\xi \|Z_i(\xi) - X_i\|^2 \leq C h^{2p} \tag{3.16}$$

✦ REMARK 3.4 It immediately follows from Jensen's inequality that a scheme which converges in mean-square to order p also converges in mean to order p, *i.e.* $\max_i \mathbb{E}_\xi \|Z_i(\xi) - X_i\| \leq C h^p$, for a possibly different C. In this form this statement is a clear analogue of the deterministic convergence statement (3.2). ✦

THEOREM 2 (CONRAD ET AL., 2016) *Consider the one-step (non-probabilistic) IVP solver with numerical flow map Ψ^h. Assume this numerical scheme has global order of convergence p. Let $\xi_i \sim \mathcal{N}(0, \alpha h^r \cdot \mathbb{I}_d)$, i.i.d. for each $0 \leq i \leq N \equiv \lfloor t_{\text{end}}/h \rfloor$ and $\alpha > 0$.*

Then the randomised method defined by $Z_{i+1} = \Psi^h(Z_i) + \xi_i$ has mean-square order of convergence p if and only if $r \geq 2p+1$. Specifically, for some constant $C > 0$ independent of h but possibly depending on t_{end}, it holds that

$$\max_i \mathbb{E}_\xi \left\| Z_i(\xi) - X_i \right\|^2 \leq C h^{2p} \tag{3.17}$$

✦ REMARK 3.5 As alluded to in Remark 3.4, convergence in mean-square is not the only type of stochastic convergence possible. Other types of convergence, which form a partial hierarchy of implication, including convergence in probability, convergence in distribution, and almost-sure convergence. Precise definitions and basic results connecting them are given in *e.g.* Grimmett & Stirzaker [Gri01, §7.2].

In the present context, we note that a result analogous to Theorem 2 but establishing a stronger type of convergence (still mean-square but with the time supremum inside the expectation—so that $\lim_{h\to 0} \mathbb{E}_\xi [\max_i \| Z_i(\xi) - X_i \|^2] = 0$) is given by Lie et al. [Lie17]. However, in the remainder of this thesis, we work with the mode of convergence given in Definition 3.5 and used in Theorem 2. ✦

Theorem 2 gives the conditions on the stepwise perturbations under which randomised one-step IVP integrators preserve the overall convergence rate of their classical counterparts. We have stated it in a slightly different form to that given in the original paper [Con16].

The operative point is the exponent of h in the variance of the Gaussian perturbations. Any value greater than or equal to $2p + 1$ leaves the global convergence properties of the scheme unchanged, while any value less than this breaks the convergence entirely. We will give our own reasoning for why this exponent should be chosen to be *exactly* $2p + 1$ in Chapter 5.

We note that this method is not restricted to the forward Euler method, but applies to all one-step methods even with multiple stages. Furthermore, an extension to continuous time is possible in certain circumstances, though for finite h the continued solution is typically discontinuous. We refer to the original paper [Con16] for full details. The remaining obstacle to forming a practical algorithm from this theoretical result is that of setting the scaling parameter α. We discuss this issue—which we term calibration—in Section 4.3.

3.3 PROBABILISTIC LINEAR MULTISTEP METHODS

Conrad et al. [Con16] only consider probabilistic versions of one-step methods, such as forward Euler or Runge–Kutta schemes. This is due to the nature of the convergence arguments in their proof of Theorem 2. A natural question that therefore arises is whether these ideas could be extended to multistep methods where, as outlined in Section 3.1.3, the estimate for the next point Z_{i+1} depends on more than one previous function evaluation. This question is considered in my paper [Tey16], jointly authored with Konstantinos Zygalakis and Ben Calderhead and presented at the Thirtieth Conference on Neural Information Processing Systems in Barcelona, in December 2016.

Before continuing, we briefly recall here the overall structure of the statistical model underlying the use of randomised numerical solvers. Assuming that some appropriate calibration process has already been undertaken, the randomised algorithm proceeds by drawing multiple $\omega^{[k]}$ from the sample space Ω and uses these to form realised sequences of perturbations $\xi_i(\omega^{[k]})$, which are then applied step-by-step to a classical numerical solver.

The output of each such run is a sample trajectory $Z^{[k]}$ drawn from $p(Z|\theta, \phi, \xi)$. With the collected output of the entire procedure, we are then able to marginalise ξ by Monte Carlo, giving a set of samples from $p(Z|\theta, \phi)$. We then take this ensemble to represent the measure over numerical solutions $p(X|\theta, \phi)$, reflecting the uncertainty in X arising from the inexactness of the classical method.

In this section, we retain this structure but generalise the step-forward distribution $p(Z_{i+1}|Z_i, F_i, \phi)$. We rewrite equation (2.10) in such a way that this distribution now depends on several previous function evaluations.[27] This results in the decomposition

$$p(Z_{1:N}|\theta, \phi) = \int \left[\prod_{i=0}^{N-1} p(F_i|Z_i, \theta) p(Z_{i+1}|Z_i, \underbrace{F_i, F_{i-1}, \ldots, F_{i-s+1}}_{s \text{ points}}, \phi) \right] \mathrm{d}F_{0:N-1} \quad (3.18)$$

We propose a model for the term $p(Z_{i+1}|Z_i, F_i, F_{i-1}, \ldots, F_{i-s+1}, \phi)$ which aligns in a particular sense with the s-step Adams–Bashforth method. The end result is a randomised multistep integrator, with a convergence result generalising Theorem 2.

However, we introduce the idea using an interesting novel construction starting from Gaussian process theory [Ras06]. In this way we perform the double function of both defining a randomised numerical method of the type described in Section 2.2

[27] Strictly speaking we have been somewhat loose in writing this product since this decomposition does not take account of the initialisation procedure required for the first $s - 1$ steps. We assume for now that these initial steps are calculated using some high-order classical method.

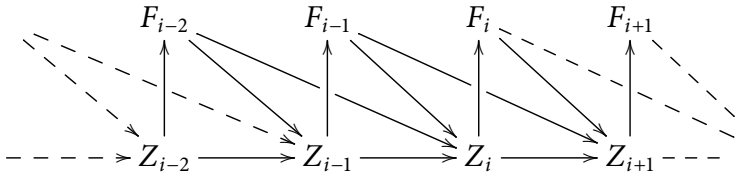


Figure 3.1: Bayesian network representation of an indicative section of the joint distribution $p(Z_{1:N}, F_{0:N-1}|\theta, \phi)$ as decomposed in Equation (3.18), with $s = 2$.

over the entire (discrete) range $(t_0, t_1, \ldots, t_{\text{end}})$, but *also* of constructing a Gaussian process model for the step-forward distribution with mean equal to a commonly-used classical method, in the manner of the probabilistic Runge–Kutta solver suggested by [Sch14] and summarised in Section 2.1.

A Bayesian network representation of the model decomposition proposed in Equation (3.18)—taking $s = 2$ for the sake of clarity—is given in figure 3.1. This can be contrasted to that previously given as figure 2.1 in Section 2.2.

3.3.1 *Gaussian process formulation of the step-forward distribution*

The construction proceeds by fixing a joint Gaussian process prior over the random variables $Z_{i+1}, Z_i, F_i, \ldots, F_{i-s+1}$.[28] This is done by first specifying a particular vector of one-variable functions $\lambda(t)$, in which $L_j^{0:s-1}$ represents the $(j+1)$th Lagrange polynomial of order $s-1$, as defined in equation (3.11).

$$\lambda(t) = \begin{pmatrix} 0 & L_0^{0:s-1}(t) & L_1^{0:s-1}(t) & \ldots & L_{s-1}^{0:s-1}(t) \end{pmatrix}^T \tag{3.19}$$

This vector is then integrated to give another vector of functions $\Lambda(t)$ as

$$\begin{aligned} \Lambda(t) &= \int_0^t \lambda(u)\,\mathrm{d}u \\ &= \begin{pmatrix} 1 & \int_0^t L_0^{0:s-1}(u)\,\mathrm{d}u & \ldots & \int_0^t L_{s-1}^{0:s-1}(u)\,\mathrm{d}u \end{pmatrix}^T \end{aligned} \tag{3.20}$$

The elements of $\lambda(t)$ (excluding the first) form a basis for $\mathbb{P}_{s-1}$, the space of polynomials of order $s-1$, and the elements of $\Lambda(t)$ form a basis for the space $\mathbb{P}_s$. The initial 0 in $\lambda(t)$ is necessary to make the dimensions of the two vectors equal, so we can correctly define products such as $\Lambda(t)^T\lambda(t)$ which will be required later. The first element of $\Lambda(t)$ can be set to any non-zero constant c—the analysis later is unaffected—and we therefore take $c = 1$.

[28] For this section, we drop the explicit dependence on θ and ϕ for notational clarity.

We now consider a Gaussian process model of the following form:

$$\begin{aligned} z(t) &\sim \mathcal{GP}(\mu(t), k(t,t')) \\ \mu(t) &= 0\,, \qquad k(t,t') = \Lambda(t)^T \Lambda(t') \end{aligned} \tag{3.21}$$

This formulation clearly recalls the Gaussian process definition of the probabilistic methods of Section 2.1. However, we choose the notation $z(t)$ rather than $x(t)$ to reinforce the point that we are defining a model for the numerical solution itself.

Recalling the statistical language of Section 2.1, we think of this as a prior distribution. In the same manner as described there, we exploit the fact that Gaussian processes are closed under differentiation [Ras06, §9.4], meaning we can immediately write down the covariance kernel of the derivative as $\lambda(t)^T \lambda(t')$.

We will show that, conditional on past evaluations, the posterior process arising from (3.21) is degenerate—*i.e.* has zero variance—and is equivalent to the Adams–Bashforth polynomial interpolator. It follows that, considered at time t_{i+1}, the posterior aligns with the usual Adams–Bashforth estimate as given by equation (3.13).

Since we will solely be interested in values of the argument t corresponding to discrete equally spaced time-steps $t_{i-j} - t_{i-j-1} \equiv h$ indexed relative to the current time-point t_i, we will make our notation more concise by writing λ_{i-j} for $\lambda(t_{i-j})$, and similarly Λ_{i-j} for $\Lambda(t_{i-j})$. This is a justifiable simplification since the Adams–Bashforth method—and indeed any time-stepping method—is a mapping between discrete objects.

We therefore define the discrete Gaussian joint distribution over $Z_{i+1}, Z_i, F_i, \ldots, F_{i-s+1}$ in a manner consistent with the process defined by (3.21). This results in the following multivariate Gaussian Gram matrix:

$$\begin{pmatrix} Z_{i+1} \\ Z_i \\ F_i \\ F_{i-1} \\ \vdots \\ F_{i-s+1} \end{pmatrix} = \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \begin{pmatrix} \Lambda_{i+1}^T \Lambda_{i+1} & \Lambda_{i+1}^T \Lambda_i & \Lambda_{i+1}^T \lambda_i & \cdots & \Lambda_{i+1}^T \lambda_{i-s+1} \\ \Lambda_i^T \Lambda_{i+1} & \Lambda_i^T \Lambda_i & \Lambda_i^T \lambda_i & \cdots & \Lambda_i^T \lambda_{i-s+1} \\ \lambda_i^T \Lambda_{i+1} & \lambda_i^T \Lambda_i & \lambda_i^T \lambda_i & \cdots & \lambda_i^T \lambda_{i-s+1} \\ \lambda_{i-1}^T \Lambda_{i+1} & \lambda_{i-1}^T \Lambda_i & \lambda_{i-1}^T \lambda_i & \cdots & \lambda_{i-1}^T \lambda_{i-s+1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \lambda_{i-s+1}^T \Lambda_{i+1} & \lambda_{i-s+1}^T \Lambda_i & \lambda_{i-s+1}^T \lambda_i & \cdots & \lambda_{i-s+1}^T \lambda_{i-s+1} \end{pmatrix} \right) \tag{3.22}$$

Recalling the decomposition (3.18), we are interested in the conditional ('posterior') distribution $p(Z_{i+1} | Z_i, F_i, \ldots, F_{i-s+1})$. This can be derived from the joint prior distribution (3.22) by applying the standard rules of multivariate Gaussian conditioning[29] [Sch17, Example 7.3], where the conditioning takes place over all variables whose value is known. This procedure leads to the following result.

[29] If $x \sim \mathcal{N}(\mu, \Sigma)$ has partition $x = \binom{x_1}{x_2}$, and $\mu = \binom{\mu_1}{\mu_2}$ and $\Sigma = \binom{\Sigma_{11}\ \Sigma_{12}}{\Sigma_{21}\ \Sigma_{22}}$ are partitioned similarly, then the conditional distribution of $x_1 | x_2$ is given by $\mathcal{N}\left(\mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(x_2 - \mu_2), \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}\right)$.

PROPOSITION 3.3 *The conditional distribution $p(Z_{i+1}|Z_i, F_{i-s+1}, \ldots, F_i)$ under the Gaussian process prior given in* (3.22)*, with covariance kernel basis functions $\lambda(t)$ and $\Lambda(t)$ as in* (3.19) *and* (3.20)*, is a δ-measure concentrated on the s-step Adams–Bashforth predictor $Z_i + h\sum_{j=0}^{s-1} \beta_{j,s}^{AB} F_{i-j}$.*

Proof. Recall that $h = t_i - t_{i-1}$ for all i and take $t_i = 0$ without loss of generality. Straightforward substitutions into (3.19) give that

$$\begin{aligned}
\lambda_i \equiv \lambda(0) &= (0,1,0,\ldots,0)^T \\
\lambda_{i-1} \equiv \lambda(-h) &= (0,0,1,\ldots,0)^T \\
&\vdots \\
\lambda_{i-s+1} \equiv \lambda(-(s-1)h) &= (0,0,0,\ldots,1)^T
\end{aligned}$$

and it follows that $\lambda_{i-p}^T \lambda_{i-q} = \delta_{pq}$, for all $0 \le p, q \le s-1$.

Similarly, $\Lambda_i \equiv \Phi(0) = (1,0,0,\ldots,0)^T$ since every component of $\Lambda(t)$ bar the first is a polynomial of degree s containing a factor t. Finally

$$\Lambda_{i+1} \equiv \Lambda(h) = \begin{pmatrix} 1 & \int_0^h L_0^{0:s-1}(u)\,\mathrm{d}u & \ldots & \int_0^h L_{s-1}^{0:s-1}(u)\,\mathrm{d}u \end{pmatrix}^T$$

In the following, we recall the notation $\Lambda_{i+1}^{(j)}$, which denotes the j'th component of Λ_{i+1}. By (3.22) and the formulae in footnote 29, we have

$$\begin{aligned}
&\mathbb{E}(Z_{i+1}|Z_i, F_{i-s+1}, \ldots, F_i) = \\
&\qquad \begin{pmatrix} \Lambda_{i+1}^T\Lambda_i \\ \Lambda_{i+1}^T\lambda_i \\ \vdots \\ \Lambda_{i+1}^T\lambda_{i-s+1} \end{pmatrix}^T \underbrace{\begin{pmatrix} \Lambda_i^T\Lambda_i & \Lambda_i^T\lambda_i & \cdots & \Lambda_i^T\lambda_{i-s+1} \\ \lambda_i^T\Lambda_i & \lambda_i^T\lambda_i & \cdots & \lambda_i^T\lambda_{i-s+1} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_{i-s+1}^T\Lambda_i & \lambda_{i-s+1}^T\lambda_{i+1} & \ldots & \lambda_{i-s+1}^T\lambda_{i-s+1} \end{pmatrix}}_{\mathbb{I}_{s+1}}{}^{-1} \begin{pmatrix} Z_i \\ F_i \\ \vdots \\ F_{i-s+1} \end{pmatrix} \\
&\qquad = (\Lambda_{i+1}^T\Lambda_i)Z_i + \sum_{j=0}^{s-1} (\Lambda_{i+1}^T\lambda_{i-j}) F_{i-j} \\
&\qquad = Z_i + \sum_{j=0}^{s-1} \Lambda_{i+1}^{(j+2)} F_{i-j} \\
&\qquad = Z_i + \sum_{j=0}^{s-1} \left[\int_0^h L_j^{0:s-1}(u)\,\mathrm{d}u \right] \cdot F_{i-j} \\
&\qquad = Z_i + h\sum_{j=0}^{s-1} \beta_{j,s}^{AB} F_{i-j} \qquad\qquad \text{since } \int_0^h L_j^{0:s-1}(u)\,\mathrm{d}u = h\beta_{j,s}^{AB}
\end{aligned}$$

which is equal to the s-step Adams–Bashforth predictor defined in (3.11) and (3.13).

Next we use the same substitutions in the formula for conditional variance.

$$\begin{aligned}
\operatorname{Var}(Z_{i+1}|Z_i, F_{i-s+1}, \ldots, F_i) &= \Lambda_{i+1}^T \Lambda_{i+1} - \begin{pmatrix} \Lambda_{i+1}^T \Lambda_i \\ \Lambda_{i+1}^T \lambda_i \\ \vdots \\ \Lambda_{i+1}^T \lambda_{i-s+1} \end{pmatrix}^T \mathbb{I}_{s+1}^{-1} \begin{pmatrix} \Lambda_i^T \Lambda_{i+1} \\ \lambda_i^T \Lambda_{i+1} \\ \vdots \\ \lambda_{i-s+1}^T \Lambda_{i+1} \end{pmatrix} \\
&= \Lambda_{i+1}^T \Lambda_{i+1} - \begin{pmatrix} 1 \\ \Lambda_{i+1}^{(2)} \\ \vdots \\ \Lambda_{i+1}^{(s+1)} \end{pmatrix}^T \begin{pmatrix} 1 \\ \Lambda_{i+1}^{(2)} \\ \vdots \\ \Lambda_{i+1}^{(s+1)} \end{pmatrix} \\
&= \Lambda_{i+1}^T \Lambda_{i+1} - \Lambda_{i+1}^T \Lambda_{i+1} \\
&= 0
\end{aligned}$$

Since a Gaussian distribution is fully specified by its mean and variance, the proposition follows. □

This construction is effectively a different derivation of the classical non-probabilistic Adams–Bashforth method. However, because of the natural probabilistic structure provided by the Gaussian process framework, we can now augment the vectors $\lambda(t)$ and $\Lambda(t)$ with an additional term to generate a conditional distribution for Z_{i+1} with non-zero variance. In Teymur et al. [Tey16], we considered an additional term of the form $\alpha h^s L_{-1}^{-1:s-1}$, *i.e.* the first Lagrange polynomial of one order higher than before. We define the vectors

$$\lambda^+(t) = \begin{pmatrix} 0 & L_0^{0:s-1}(t) & L_1^{0:s-1}(t) & \ldots & L_{s-1}^{0:s-1}(t) & \alpha h^s L_{-1}^{-1:s-1}(t) \end{pmatrix}^T \tag{3.23}$$

$$\begin{aligned}
\Lambda^+(t) &= \int_0^t \lambda^+(u)\,\mathrm{d}u \\
&= \begin{pmatrix} 1 & \int_0^t L_0^{0:s-1}(u)\,\mathrm{d}u & \ldots & \int_0^t L_{s-1}^{0:s-1}(u)\,\mathrm{d}u & \int_0^t \alpha h^s L_{-1}^{-1:s-1}(u)\,\mathrm{d}u \end{pmatrix}^T
\end{aligned} \tag{3.24}$$

Now considering the analogous Gaussian process constructions to (3.21) and (3.22), we have the following result:

PROPOSITION 3.4 *The conditional distribution $p(Z_{i+1}|Z_i, F_{i-s+1}, \ldots, F_i)$ under the Gaussian process prior given in* (3.22)*, with covariance kernel basis functions $\lambda^+(t)$ and $\Lambda^+(t)$ as in* (3.19) *and* (3.20) *replaced by their augmented versions* (3.23) *and* (3.24) *respectively, is Gaussian with mean equal to the s-step Adams–Bashforth predictor $Z_i + h\sum_{j=0}^{s-1} \beta_{j,s}^{AB} F_{i-j}$ and—setting $\alpha = \mathrm{d}^{(s+1)}x/\,\mathrm{d}t^{(s+1)}(\tau)$, where τ is a particular value of t in the range (t_{i-s+1}, t_{i+1})—standard deviation equal to the absolute value of its local truncation error.*

Proof. We follow the same reasoning as in Proposition 3.3. Since the additional basis function at the end of the augmented vector λ^+_{i-j} is zero for all $0 \leq j \leq s-1$, each inner product of the form $\lambda^{+T}\lambda^+$, $\Lambda^{+T}\lambda^+$ and $\lambda^{+T}\Lambda^+$ is equal to the corresponding inner product $\lambda^T\lambda$, $\Lambda^T\lambda$ and $\lambda^T\Lambda$, as no additional contribution from the new extended basis arises. It therefore suffices to check only the terms of the form $\Lambda^{+T}\Lambda^+$.

Integrating the additional basis function with respect to t gives a polynomial of degree $s+1$ with a constant factor t. Evaluating this at $t_i = 0$ means that the final component is also 0 in Λ^+_i. Therefore $\Lambda^{+T}_{i+1}\Lambda^+_i = \Lambda^T_{i+1}\Lambda_i$ and $\Lambda^{+T}_i\Lambda^+_i = \Lambda^T_i\Lambda_i$. It follows that the expression for $\mathbb{E}(Z_{i+1}|Z_i, F_i, \ldots, F_{i-s+1})$ is exactly the same as when using the unaugmented basis function set.

The argument in the previous paragraph means we can immediately write down that

$$\mathrm{Var}(Z_{i+1}|Z_i, F_i, \ldots, F_{i-s+1}) = \Lambda^{+T}_{i+1}\Lambda^+_{i+1} - \Lambda^T_{i+1}\Lambda_{i+1}$$

Since the first $s+1$ components of Λ^{+T}_{i+1} are equal to the $s+1$ components of Λ^T_{i+1}, this expression reduces to the contribution of the additional basis element. Therefore

$$\begin{aligned}\mathrm{Var}(Z_{i+1}|Z_i, F_i, \ldots, F_{i-s+1}) &= \left(\alpha h^s \int_0^h L_{-1}^{-1:s-1}(u)\,\mathrm{d}u\right)^2 \\ &= \left(\alpha h^{s+1}\beta^{AM}_{-1,s+1}\right)^2\end{aligned}$$

Up to sign, the Adams–Moulton coefficient $\beta^{AM}_{-1,s+1}$ is equal to the error constant for the s-step Adams–Bashforth method [But08, §244] and the proposition follows. □

Example

In order to demystify the construction, we now exhibit a concrete example for the case $s = 3$. The conditional distribution of interest is $p(Z_{i+1}|Z_i, F_i, F_{i-1}, F_{i-2})$. In the non-probabilistic case, the vectors of basis functions become

$$\begin{aligned}\lambda(t, s=3) &= \begin{pmatrix} 0 & \dfrac{(t+h)(t+2h)}{2h^2} & \dfrac{t(t+2h)}{-h^2} & \dfrac{t(t+h)}{2h^2}\end{pmatrix} \\ \Lambda(t, s=3) &= \begin{pmatrix} 1 & \dfrac{t(2t^2+9ht+h^2)}{12h^2} & \dfrac{t^2(t+3h)}{-3h^2} & \dfrac{t^2(2t+3h)}{12h^2}\end{pmatrix}\end{aligned}$$

Simple calculations now give that

$$\begin{aligned}\mathbb{E}(Z_{i+1}|Z_i, F_i, F_{i-1}, F_{i-2}) &= Z_i + h\left(\frac{23}{12}F_i - \frac{4}{3}F_{i-1} + \frac{5}{12}F_{i-2}\right) \\ \mathrm{Var}(Z_{i+1}|Z_i, F_i, F_{i-1}, F_{i-2}) &= 0\end{aligned}$$

The probabilistic version follows by setting

$$\lambda^+(t, s=3) = \begin{pmatrix} 0 & \dfrac{(t+h)(t+2h)}{2h^2} & \dfrac{t(t+2h)}{-h^2} & \dfrac{t(t+h)}{2h^2} & \dfrac{\alpha t(t+h)(t+2h)}{6} \end{pmatrix}$$

$$\Lambda^+(t, s=3) = \begin{pmatrix} 1 & \dfrac{t\,(2t^2+9ht+h^2)}{12h^2} & \dfrac{t^2\,(t+3h)}{-3h^2} & \dfrac{t^2\,(2t+3h)}{12h^2} & \dfrac{\alpha t^2(t+2h)^2}{24} \end{pmatrix}$$

and further calculation shows that

$$\mathbb{E}(Z_{i+1}|Z_i, F_i, F_{i-1}, F_{i-2}) = Z_i + h\left(\frac{23}{12}F_i - \frac{4}{3}F_{i-1} + \frac{5}{12}F_{i-2}\right)$$

$$\mathrm{Var}(Z_{i+1}|Z_i, F_i, F_{i-1}, F_{i-2}) = \left(\frac{3h^4\alpha}{8}\right)^2$$

Taking $\alpha = \mathrm{d}^{(s+1)}x/\,\mathrm{d}t^{(s+1)}(\tau)$ for some $\tau \in (t_{i-s+1}, t_{i+1})$, we see that $3h^4\alpha/8$ is equal to the absolute value of the local truncation error of the 3-step Adams–Bashforth method.

✦ Remark 3.6 It is straightforward to see that sampling Z_{i+1} from the conditional distribution $p(Z_{i+1}|Z_i, F_i, F_{i-1}, \ldots, F_{i-s+1}, \phi)$ using the model proposed in Proposition 3.4 is equivalent to running the classical Adams–Bashforth integrator and perturbing the output with a zero-mean Gaussian random variable ξ_i with standard deviation proportional to h^{s+1} at each step. The latter perspective shows that the integrator is entirely analogous structurally to the algorithm of Conrad et al. [Con16], but extended to the multistep setting. The formal argument that the construction is convergent to the same order as the underlying classical method is given in the proof of Theorem 3 in the next section. ✦

✦ Author's note In the paper Teymur et al. [Tey16] in which this construction was first introduced, it was then built upon in several ways, some of which we subsequently concluded were of limited use or even methodologically inconsistent. In fact, this realisation motivated much of the improved (and not inconsistent) work presented in Chapter 4 of this thesis—the details will be given there.

In short, the variance of the perturbations arising from the use of the augmented basis vectors (3.23) and (3.24) can be shown not to be correct, since it is a factor of h smaller than the bound allowed for by the convergence argument of Theorem 3. This is a direct consequence of the construction given in Proposition 3.4. Experimental evidence, arising from the process of calibration and presented in Chapter 5, strongly supports the view that in the practical design of a randomised integrator, the exponent of h should indeed be taken tight to the theoretical bound. The method suggested in Teymur et al. [Tey16] for setting the scalar α is therefore also moot, since we cannot treat α as an h-independent parameter if it scales a term which is itself dependent on h in the wrong way.

Secondly, an extension to implicit Adams–Moulton methods was proposed. While the convergence argument to be given in Theorem 3 *does* apply for implicit multistep methods—a fact we later rely on in our proof of Theorem 4—the specific Gaussian process construction analogous to that just given for Adams–Bashforth methods is inconsistent in a particular subtle sense that we describe in Section 4.1.1. This observation motivated the novel construction in our subsequent paper Teymur et al. [Tey18] and is covered in detail in Chapter 4 of this thesis.

Having made these points, it is worth pointing out for clarity those parts of the foregoing work which are novel and *do* work. The Adams–Bashforth construction is valid and the Gaussian process approach to deriving it is—to the best of our knowledge—new. While the particular form of the probabilistic integrator covariance kernel results in a suboptimal posterior variance, the posterior mean does align with the classical Adams–Bashforth predictor—this gives the construction the same status as algorithms such as the probabilistic Runge–Kutta solver of Schober et al. [Sch14], in which the scaling of the posterior variance is also left unresolved.

Finally, the main convergence result in Theorem 3, to be given in the next section, is valid and the bound it gives on the variance of the stepwise perturbations is strictly weaker than that required in the construction just given. Furthermore, the analysis applies to both explicit and implicit linear multistep methods, also covering those not of Adams type. This generality allows us to appeal to the result once again during the later proof of Theorem 4. ✦

3.3.2 *Convergence of the probabilistic multistep integrator*

We now give the analysis that proves the convergence of the multistep probabilistic integrator constructed in Section 3.3.1, in the same mean-square sense of Theorem 2. The version presented here is a modified version of that published as Theorem 3 in Teymur et al. [Tey16]. In fact, we give it in a significantly more general form, not tying it to the specific construction from Section 3.3.1, and furthermore encompassing the case of implicit multistep integrators as well as those not of Adams type.

THEOREM 3 *Consider the initial value problem* (1.9). *Assume the vector field* $f(\cdot, \theta)$ *is globally Lipschitz with Lipschitz constant* L_f. *For some end time* $t_{\text{end}} > 0$ *and time-step* $h > 0$ *define a grid* $t_i = ih$ *for* $0 \leq i \leq N \equiv \lfloor t_{\text{end}}/h \rfloor$. *Denote the exact solution of* (1.9) *at time* t_i *by* X_i.

For fixed $k \geq 2$, *consider the linear multistep IVP solver of the form*

$$\tilde{Z}_{i+1} = -\sum_{j=0}^{k-2} a_j \tilde{Z}_{i-j} + h \sum_{j=-1}^{k-2} b_j f(\tilde{Z}_{i-j}) \tag{3.25}$$

Assume the coefficients a_j and b_j are chosen such that this numerical scheme is of order p, i.e. the local truncation error e_i incurred in one step of length h is $\mathcal{O}(h^{p+1})$ and, furthermore, that the initialisation procedure used to calculate $\tilde{Z}_1, \ldots, \tilde{Z}_{k-2}$ is also of order p.

Let $\xi_i \sim \mathcal{N}(0, \alpha h^r \cdot \mathbb{I}_d)$, i.i.d. for each $k-1 \leq i \leq n$ and some $\alpha > 0$. Then the randomised method defined by

$$Z_{i+1} = -\sum_{j=0}^{k-2} a_j Z_{i-j} + h \sum_{j=-1}^{k-2} b_j f(Z_{i-j}) + \xi_i \tag{3.26}$$

has mean-square order of convergence p if and only if $r \geq 2p+1$. Specifically, for some constant $C > 0$ independent of h but possibly depending on t_{end}, it holds that

$$\max_i \mathbb{E} \left\| Z_i - X_i \right\|^2 \leq C h^{2p} \tag{3.27}$$

Proof. Recall that we denote the true solution of the initial value problem (1.9) at time t_i by $X_i \equiv x(t_i)$. If we substitute the true solution values for their numerical approximations into (3.26), we have

$$X_{i+1} = -\sum_{j=0}^{k-2} a_j X_{i-j} + h \sum_{j=-1}^{k-2} b_j f(X_{i-j}) + \tau_i \tag{3.28}$$

where the local truncation error $\tau_i = \mathcal{O}(h^{p+1})$, by the assumption that the underlying solver is of order p. If we now subtract (3.26) from (3.28) and denote the accumulated error at iteration i by $E_i := X_i - Z_i$, we have

$$E_{i+1} = -\sum_{j=0}^{k-2} a_j E_{i-j} + Q_i + \tau_i - \xi_i \tag{3.29}$$

where

$$Q_i := h \sum_{j=-1}^{k-2} b_j \Delta F_{i-j}, \qquad \Delta F_{i-j} := f(X_{i-j}) - f(Z_{i-j}).$$

Following Buckwar & Winkler [Buc06], we rearrange this k-step recursion to give an equivalent one-step recursion in an higher-dimensional space. Using the trivial identities

$$\begin{aligned} E_i &= E_i \\ E_{i-1} &= E_{i-1} \\ &\vdots \\ E_{i-k+2} &= E_{i-k+2} \end{aligned}$$

we obtain

$$\underbrace{\begin{pmatrix} E_{i+1} \\ E_i \\ \vdots \\ \vdots \\ E_{i-k+2} \end{pmatrix}}_{=:\mathcal{E}_{i+1}} = \underbrace{\begin{pmatrix} -a_0\mathbb{I}_d & -a_1\mathbb{I}_d & \cdots & -a_{k-1}\mathbb{I}_d \\ \mathbb{I}_d & 0 & \cdots & 0 \\ & \ddots & \ddots & \\ & & \ddots & \ddots \\ 0 & & \mathbb{I}_d & 0 \end{pmatrix}}_{=:A} \underbrace{\begin{pmatrix} E_i \\ E_{i-1} \\ \vdots \\ \vdots \\ E_{i-k+2} \end{pmatrix}}_{=:\mathcal{E}_i} + \underbrace{\begin{pmatrix} Q_i \\ 0 \\ \vdots \\ \vdots \\ 0 \end{pmatrix}}_{=:\mathcal{Q}_i} + \underbrace{\begin{pmatrix} \tau_i \\ 0 \\ \vdots \\ \vdots \\ 0 \end{pmatrix}}_{=:\mathcal{T}_i} - \underbrace{\begin{pmatrix} \xi_i \\ 0 \\ \vdots \\ \vdots \\ 0 \end{pmatrix}}_{=:\Xi_i}$$

or in compact form,

$$\mathcal{E}_{i+1} = A\mathcal{E}_i + \mathcal{Q}_i + \mathcal{T}_i - \Xi_i, \qquad i = k-1, \ldots, N, \qquad N = \left\lfloor h^{-1} t_{\text{end}} \right\rfloor \tag{3.30}$$

Note that each column vector in this equation is of dimension kd while the matrix A is $kd \times kd$.

For the subsequent argument it will be necessary to find a scalar product inducing a matrix norm such that the norm of the matrix A is less or equal to 1.

In order to do this, we will require some new definitions and several straightforward but non-trivial results from linear analysis, which we will collect here. We state them and give references to proofs, rather than reproduce lengthy rigorous arguments that can be found in several standard references.

1. The characteristic polynomial of a (not necessarily linear) multistep method is defined as $\psi(u) = \sum_{j=0}^{k-2} = a_{k-2-j}u^j$, where the coefficients a_j are as in equation (3.25) [Hac14, §5.5].

2. A multistep method is stable if all roots u_j (in general complex) of its characteristic polynomial $\psi(u)$ satisfy either $|u_j| < 1$ or $|u_j| = 1$, with those roots satisfying the latter condition required to be simple roots (*i.e.* those with multiplicity one) [Hac14, Definition 5.23]. This condition is called Dahlquist's root condition.

3. A multistep method that is linear, *i.e.* of the specific form (3.25), and where the coefficients a_j and b_j are chosen such that the method is convergent as defined in Section 3.1, is stable in the sense of point 2. In particular, this includes all Adams-type methods so far discussed [Hac14, Theorem 5.42].

4. The matrix A arising from a multistep method as in equation (3.30) is called the method's companion matrix [Hac14, Definition 5.37]. If the method is stable, this matrix is such that its eigenvalues are equal to the roots u_j of its characteristic polynomial $\psi(u)$ [Hai08, §III.4 Lemma 4.4].

5. Any square matrix A can be decomposed as PJP^{-1} where P is an invertible matrix and J is a matrix in Jordan canonical form—that is, a block matrix with each block J_j of size equal to the algebraic multiplicity of an eigenvalue u_j of A. The J_j are all such that each entry on its leading diagonal is the eigenvalue u_j and each entry in its first super-diagonal is 1 [Hor12, §3.1]. We say that A is similar to J.

6. The discrete Grönwall lemma can be stated in many forms; we use the version given by Law et al. [Law15, Lemma 1.14]. Given constants a and $b > 0$ and a sequence of positive numbers $c_0, c_1, \ldots > 0$, then for $b \neq 1$ it holds that

$$c_{i+1} \leq bc_i + a \quad \Longrightarrow \quad c_i \leq b^i c_0 + a\frac{1-b^i}{1-b}.$$

We now proceed with the main thrust of the proof. The following argument is modified from Horn & Johnson [Hor12, Lemma 5.6.10] and shows that for the case we are interested in—a convergent linear multistep method of the form (3.25)—we can construct a scalar product such that the companion matrix A has norm less than or equal to 1 in the induced matrix norm.

Using point 5, we find the matrix J in Jordan canonical form which is similar to A. For each Jordan block J_j, of size $m \times m$, we consider the diagonal matrix $D_j = \operatorname{diag}(q_j, q_j^2, \ldots, q_j^m)$ and note that

$$D_j J_j D_j^{-1} = \begin{pmatrix} u_j & q_j^{-1} & & \\ & \ddots & \ddots & \\ & & \ddots & q_j^{-1} \\ & & & u_j \end{pmatrix} = \begin{pmatrix} u_j & & & \\ & \ddots & & \\ & & \ddots & \\ & & & u_j \end{pmatrix} + \begin{pmatrix} 0 & q_j^{-1} & & \\ & \ddots & \ddots & \\ & & \ddots & q_j^{-1} \\ & & & 0 \end{pmatrix} \tag{3.31}$$

For a vector $v \in \mathbb{R}^m$, recall the definition of the Euclidean vector norm $\|v\|_2 = \sqrt{v^T v}$ and the induced matrix 2-norm $\|\|M\|\|_2 = \max_{v \neq 0} \|Mv\|_2 / \|v\|_2 = \max_{\|v\|_2 = 1} \|Mv\|_2$.

Using the triangle inequality, we have from (3.31)

$$\left\|D_j J_j D_j^{-1} v\right\|_2 \leq |u_j| \|v\|_2 + q_j^{-1} \|v\|_2 = (|u_j| + q_j^{-1}) \|v\|_2 \tag{3.32}$$

Since all eigenvalues u_j with $|u_j| = 1$ are simple, for blocks of size $m > 1$ we must have $|u_j| < 1$. Thus we can choose a q_j large enough such that $|u_j| + q_j^{-1} < 1$ and it follows that for such a q_j we have

$$\left\|\left\|D_j J_j D_j^{-1}\right\|\right\|_2 < 1$$

Repeating for each Jordan block it follows that

$$1 \geq \max_j \left\|\left\|D_j J_j D_j^{-1}\right\|\right\|_2 = \left\|\left\|DJD^{-1}\right\|\right\|_2 = \left\|\left\|DP^{-1}APD^{-1}\right\|\right\|_2 = \left\|\left\|\Lambda^{-1}A\Lambda\right\|\right\|_2 \tag{3.33}$$

where we have defined $\Lambda := PD^{-1}$. Note that equality is achieved in the first relation if and only if there exist simple unit-modulus eigenvalues u_j.

Consider now the vector $\mathcal{V} := (v_1^T, v_2^T, \dots, v_k^T)^T \in \mathbb{R}^{kd}$. This column vector, made up of k concatenated d-tuples v_j, resembles in form the vector terms appearing in (3.30).

We now choose a scalar product for $\mathcal{V}, \mathcal{W} \in \mathbb{R}^{kd}$ as

$$\langle \mathcal{V}, \mathcal{W} \rangle_* := \left\langle \Lambda^{-1}\mathcal{V}, \Lambda^{-1}\mathcal{W} \right\rangle_2 \tag{3.34}$$

where $\langle \cdot, \cdot \rangle_2$ is the standard Euclidean inner product $\langle v, w \rangle_2 = v^T w$. We can then write $\| \cdot \|_*$ for the induced vector norm and $|\!|\!| \cdot |\!|\!|_*$ for the induced matrix norm respectively, with

$$|\!|\!| A |\!|\!|_* = \max_{\|\mathcal{V}\|_*=1} \|A\mathcal{V}\|_* = \max_{\|\Lambda^{-1}\mathcal{V}\|_2=1} \left\|\Lambda^{-1}A\mathcal{V}\right\|_2 = \max_{\|\mathcal{W}\|_2=1} \left\|\Lambda^{-1}A\Lambda\mathcal{W}\right\|_2 = \left|\!\left|\!\left|\Lambda^{-1}A\Lambda\right|\!\right|\!\right|_2 \leq 1$$

as required. We also have

$$\langle \mathcal{V}, \mathcal{W} \rangle_* = \mathcal{V}^T \Lambda^{-T}\Lambda^{-1}\mathcal{W} = \mathcal{V}^T \Lambda^* \mathcal{W} \quad \text{with} \quad \Lambda^* = \Lambda^{-T}\Lambda^{-1} = (\lambda^*_{ij})_{1\leq i,j\leq k} \otimes \mathbb{I}_d \tag{3.35}$$

Due to the equivalence of norms there exist constants $c^*, c_* > 0$ such that

$$\|\mathcal{V}\|_2^2 \leq c^* \|\mathcal{V}\|_*^2 \quad \text{and} \quad \|\mathcal{V}\|_*^2 \leq c_* \|\mathcal{V}\|_\infty^2 \quad \text{for all } \mathcal{V} \in \mathbb{R}^{kd}, \tag{3.36}$$

where $\|\mathcal{V}\|_2^2 = \sum_{j=1,\dots,k} \left\|v_j\right\|^2$ and $\|\mathcal{V}\|_\infty = \max_{j=1,\dots,k} \left\|v_j\right\|$.

For the particular vectors $\tilde{\mathcal{V}} = (v^T, 0, \dots, 0)^T$ and $\tilde{\mathcal{W}} = (w^T, 0, \dots, 0)^T$ with $\tilde{\mathcal{V}}, \tilde{\mathcal{W}} \in \mathbb{R}^{kd}$ and $v, w \in \mathbb{R}^d$, one has

$$\langle \tilde{\mathcal{V}}, \tilde{\mathcal{W}} \rangle_* = \lambda^*_{11} \langle v, w \rangle_2 = \lambda^*_{11} v^T w, \tag{3.37}$$

where λ^*_{11} is as in (3.35).

Having established the various necessary analytical preliminaries, we now proceed with the core bounding argument. Applying the newly-defined norm $\| \cdot \|_*$ to (3.30), squaring and taking expectations gives

$$\begin{aligned}
\mathbb{E}\|\mathcal{E}_{i+1}\|_*^2 &= \mathbb{E}\|A\mathcal{E}_i + \mathcal{Q}_i + \mathcal{T}_i - \Xi_i\|_*^2 \\
&= \mathbb{E}\|A\mathcal{E}_i + \mathcal{Q}_i + \mathcal{T}_i\|_*^2 + \mathcal{O}(h^r) \\
&= \mathbb{E}\|A\mathcal{E}_i + \mathcal{Q}_i\|_*^2 + 2 \cdot \mathbb{E}\langle h^{1/2}(A\mathcal{E}_i + \mathcal{Q}_i), \mathcal{T}_i h^{-1/2} \rangle_* + \mathbb{E}\|\mathcal{T}_i\|_*^2 + \mathcal{O}(h^r) \\
&= \mathbb{E}\|A\mathcal{E}_i + \mathcal{Q}_i\|_*^2 + 2 \cdot \mathbb{E}\langle h^{1/2}(A\mathcal{E}_i + \mathcal{Q}_i), \mathcal{T}_i h^{-1/2} \rangle_* + \mathcal{O}(h^{2p+2}) + \mathcal{O}(h^r)
\end{aligned} \tag{3.38}$$

We now consider the term $\|A\mathcal{E}_i + \mathcal{Q}_i\|_*^2$ and expand it as

$$\|A\mathcal{E}_i + \mathcal{Q}_i\|_*^2 = \underbrace{\|A\mathcal{E}_i\|_*^2}_{(1)} + \underbrace{\|\mathcal{Q}_i\|_*^2}_{(2)} + \underbrace{2 \cdot \langle A\mathcal{E}_i, \mathcal{Q}_i \rangle_*}_{(3)}$$

For term (1) we immediately have $\|A\mathcal{E}_i\|_*^2 \leq \|\mathcal{E}_i\|_*^2$ by construction of the norm $\|\cdot\|_*$. For term (2) we have that

$$\begin{aligned}
\|\mathcal{Q}_i\|_*^2 &= \lambda_{11}^* \|Q_i\|^2 && \text{from (3.37)} \\
&= \lambda_{11}^* \left\| h \textstyle\sum_{j=-1}^{k-2} b_j \Delta F_{i-j} \right\|^2 \\
&\leq \lambda_{11}^* k h^2 \textstyle\sum_{j=-1}^{k-2} \left\| b_j \Delta F_{i-j} \right\|^2 && \text{by Cauchy-Schwarz} \\
&\leq \lambda_{11}^* k h^2 L_f^2 \textstyle\sum_{j=-1}^{k-2} b_j^2 \left\| E_{i-j} \right\|^2 && \text{since } f \text{ is Lipschitz} \\
&\leq \lambda_{11}^* k h^2 L_f^2 C_b^2 \textstyle\sum_{j=-1}^{k-2} \left\| E_{i-j} \right\|^2 && \text{where } C_b^2 = \max_{j=-1,\ldots,k-2} b_j \\
&\leq \lambda_{11}^* k h^2 L_f^2 C_b^2 c^* \|\mathcal{E}_i\|_*^2 && \text{from (3.36)} \\
&= \Gamma^2 h^2 \|\mathcal{E}_i\|_*^2 && \text{where } \Gamma^2 = \lambda_{11}^* k L_f^2 C_b^2 c^* > 0
\end{aligned}$$

For term (3) we have $2 \cdot \langle A\mathcal{E}_i, \mathcal{Q}_i \rangle_* \leq 2 \cdot \|A\mathcal{E}_i\|_* \cdot \|\mathcal{Q}_i\|_* \leq 2\Gamma h \|\mathcal{E}_i\|_*^2$ and it follows that

$$\|A\mathcal{E}_i + \mathcal{Q}_i\|_*^2 \leq (1 + \mathcal{O}(h)) \|\mathcal{E}_i\|_*^2$$

Then from (3.38) we have

$$\begin{aligned}
\mathbb{E}\|\mathcal{E}_{i+1}\|_*^2 &= \mathbb{E}\|A\mathcal{E}_i + \mathcal{Q}_i\|_*^2 + 2 \cdot \mathbb{E}\langle h^{1/2}(A\mathcal{E}_i + \mathcal{Q}_i), \mathcal{T}_i h^{-1/2} \rangle_* + \mathcal{O}(h^{2p+2}) + \mathcal{O}(h^r) \\
&\leq (1 + \mathcal{O}(h)) \mathbb{E}\|\mathcal{E}_i\|_*^2 + 2h \cdot \mathbb{E}\|A\mathcal{E}_i + \mathcal{Q}_i\|_*^2 + 2h^{-1} \cdot \mathbb{E}\|\mathcal{T}_i\|_*^2 + \mathcal{O}(h^{2p+2}) + \mathcal{O}(h^r) \\
&\leq (1 + \mathcal{O}(h)) \mathbb{E}\|\mathcal{E}_i\|_*^2 + \mathcal{O}(h^{2p+1}) + \mathcal{O}(h^r) \\
&= (1 + \mathcal{O}(h)) \mathbb{E}\|\mathcal{E}_i\|_*^2 + \mathcal{O}(h^{2p+1})
\end{aligned} \tag{3.39}$$

where the last line follows from the condition $r \geq 2p + 1$. With arbitrary constants l_1, l_2 and l_3, we then take $a = l_1 h^{2p+1}$, $b = 1 + l_2 h$ and $c_i := \mathbb{E}\|\mathcal{E}_i\|_*^2$ for $k - 1 \leq i \leq N$ (with $c_{k-1} \equiv \mathbb{E}\|\mathcal{E}_{k-1}\|_*^2 = l_3 h^{2p+1}$, by the assumption that the initialisation procedure is of order p) in the statement of the Grönwall lemma and apply its result. This gives

$$\max_i \mathbb{E}\|\mathcal{E}_i\|_*^2 \leq C h^{2p} \tag{3.40}$$

for some constant $C > 0$. Since $\mathcal{E}_i = (E_i, E_{k-i}, \cdots, E_{i-k+1})^T$ and $E_i \equiv Z_i - X_i$, we conclude that

$$\max_i \mathbb{E}\|Z_i - X_i\|^2 \leq C h^{2p} \tag{3.41}$$

□

COROLLARY 3.1 *The probabilistic integrator defined in Proposition 3.4 is convergent.*

Proof. The s-step Adams–Bashforth integrator can be written in the form (3.25) by taking $k = s+1$, $a_0 = -1$, $a_j = 0$ for $j = 1, \cdots, k-2$, and $b_j = \beta^{AB}_{j+1,s}$. The underlying integrator is of order s and the perturbations at each step are Gaussian with standard deviation proportional to the local truncation error, which is $\mathcal{O}(h^{s+1})$. This implies $r = 2p+2$ and hence the constraint in the statement of Theorem 3 is clearly seen to be satisfied, and the claim follows. □

COROLLARY 3.2 *The probabilistic s-step Adams–Bashforth integrator defined by*

$$Z_{i+1} = Z_i + \sum_{j=0}^{s-1} \beta^{AB}_{j,s} f(Z_{i-j}) + \xi_i \tag{3.42}$$

with $\xi^h_i \sim \mathcal{N}(0, \alpha h^{2s+1})$ i.i.d and $\alpha > 0$, is convergent.

Proof. Exactly as Corollary 3.1 but with $r = 2p+1$. □

COROLLARY 3.3 *The probabilistic s-step Adams–Moulton integrator defined by*

$$Z_{i+1} = Z_i + \sum_{j=-1}^{s-1} \beta^{AM}_{j,s} f(Z_{i-j}) + \xi_i \tag{3.43}$$

with $\xi^h_i \sim \mathcal{N}(0, \alpha h^{2s+3})$ i.i.d and $\alpha > 0$, is convergent.

Proof. Take $k = s+1$, $a_0 = -1$, $a_j = 0$ for $j = 1, \cdots, k-2$, and $b_j = \beta^{AM}_{j,s}$. Since the s-step Adams–Moulton integrator is of order $s+1$, we have $p = s+1$. Therefore $r = 2s+3 = 2(p-1)+3 = 2p+1$, and the result holds. □

✦ REMARK 3.7 Corollary 3.2 generalises the integrator from Proposition 3.4 to give explicitly the maximum noise scale consistent with the convergence argument of Theorem 3, *i.e.* the limiting case $r = 2p+1$. In Chapter 5 we argue that the noise scale suggested in Corollary 3.2 is the correct one, so long as there exists a method for calibrating the constant α.

Corollary 3.3 shows that no theoretical impediment exists to applying the same approach to an implicit Adams-type integrator. The convergence argument is identical. Nevertheless, as noted in Section 3.3.1 and expanded upon in Section 4.1.1, this is not a practical algorithm, due to the fact that it is defined implicitly. This observation motivates the modifications and extensions presented in the next chapter. ✦

4

IMPLICIT PROBABILISTIC ODE SOLVERS

The integrator proposed in Section 3.3 modifies explicit multistep methods by treating their stepwise truncation error as a random variable, resulting in a randomised algorithm. A natural question to ask is whether a similar construction can be found to modify implicit IVP solvers to also give a probabilistic description of solver error.

Corollary 3.3 shows that the convergence analysis of Theorem 3 remains valid for implicit Adams-type methods, however the naive modification of explicit methods to the implicit case can throw up subtle issues which either render them inconsistent or at the very least difficult to use. We will describe these issues in Section 4.1.1.

Before introducing the substance of our new construction, we take a brief diversion to discuss why implicit methods are an essential component in the numerical analyst's arsenal.

4.1 BENEFITS OF IMPLICIT METHODS

Explicit IVP solvers are intrinsically extrapolative in nature. What we mean by this is that the next point Z_{i+1} is determined independently of the evolution of the actual system dynamics beyond time t_i. In the case of Adams–Bashforth methods, the definition in Section 3.1.3 makes clear that the polynomial interpolator of the values $F_i, \ldots, F_{i-s+1}$ is extended to time t_{i+1} with no opportunity for feedback from the evolution of the defining function $f(\cdot, \theta)$. It is intuitively clear—though this statement is not precise—that this approach assumes a certain degree of smoothness

in f along with a consistency in its high-level features across a broad time interval—or at least does not account for potentially pathological behaviour of f—in order to return adequate approximations.

Implicit methods specifically account for the evolution of the dynamics of the system between time points t_i and t_{i+1}. For certain types of problems, this can be the difference between an acceptable output and a meaningless one. The term for problems where the defining dynamics are difficult to resolve for explicit solvers is 'stiff'. In reality this term is just as vague as the description in the previous paragraph, though it certainly encapsulates such features of f as smoothness, consistently-varying time scales, and other lack of pathologies.

In general it is difficult to clearly state what is meant by a stiff problem, and authors disagree on its definition. Jackiewicz [Jac09, §1.7] gives a comprehensive survey of the different definitions given by various prominent authors in well-known numerical analysis texts. We collect a small sample of these here, along with some from other sources, if only to highlight this lack of consensus.

Ramsay et al. [Ram07] posit that stiff problems are those "for which solutions beginning at varying initial values tend to converge to a common trajectory" and which "require methods that make use of the Jacobian $\partial f/\partial x$". Several authors [Ise09; Qua00; Atk09; Lam91; LeV07] define a stiff problem as one where it is the requirements of stability that constrain the step size, rather than simply those of solution accuracy. Burrage [Bur95] suggests that stiff problems are those with large values of the product $L_f(t_{\text{end}} - t_0)$[30]. Other authors [Stu98; Sha94] make reference to problems with two vastly different time-scales present, where the components varying quickly can have a significant effect on the trajectories of those varying more slowly.

Possibly the most compelling definition—though one which seems circular initially—is that given by Hairer & Wanner [Hai10, §IV.1]. They write that "stiff equations are problems for which explicit methods don't work". This recalls the definition from one of the earliest works on the topic, by Curtiss & Hirschfelder [Cur52], in which the authors state that "stiff equations are equations where certain implicit methods …perform better, usually tremendously better, than explicit ones".

Nevertheless, all authors agree that stiff problems are ubiquitous in real-world models and that methods to solve them to an acceptable degree of accuracy are required, and can often justify the increase in computation which results on these grounds. It is therefore an obvious next question whether implicit probabilistic integrators can be defined analogous to the explicit ones considered in Chapter 3.

[30] Recall that L_f denotes the Lipschitz constant of the function f defining the dynamics of the ODE, while $(t_{\text{end}} - t_0)$ is the complete time interval over which the problem is to be considered.

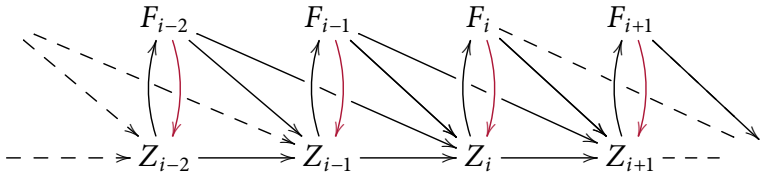


Figure 4.1: Bayesian network representation of an indicative section of the joint distribution $p(Z_{1:N}, F_{0:N-1}|\theta, \phi)$ as decomposed in Equation (4.2), with $s = 2$. Comparison with figure 3.1 highlights the inconsistency described in Section 4.1.1; the red arrows indicate the cycles now present in the model.

4.1.1 *Naive implicit probabilistic integrators*

In the paper Teymur et al. [Tey16], in which the Gaussian process formulation of a randomised Adams–Bashforth solver was introduced, an extension was mooted to the case of the implicit Adams–Moulton method. This involved defining a Gaussian process prior analogous to equation (3.22) by

$$\begin{pmatrix} Z_{i+1} \\ Z_i \\ F_{i+1} \\ F_i \\ \vdots \\ F_{i-s+1} \end{pmatrix} = \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \begin{pmatrix} \Lambda_{i+1}^T\Lambda_{i+1} & \Lambda_{i+1}^T\Lambda_i & \Lambda_{i+1}^T\lambda_{i+1} & \cdots & \Lambda_{i+1}^T\lambda_{i-s+1} \\ \Lambda_i^T\Lambda_{i+1} & \Lambda_i^T\Lambda_i & \Lambda_i^T\lambda_{i+1} & \cdots & \Lambda_i^T\lambda_{i-s+1} \\ \lambda_{i+1}^T\Lambda_{i+1} & \lambda_{i+1}^T\Lambda_i & \lambda_{i+1}^T\lambda_{i+1} & \cdots & \lambda_{i+1}^T\lambda_{i-s+1} \\ \lambda_i^T\Lambda_{i+1} & \lambda_i^T\Lambda_i & \lambda_i^T\lambda_{i+1} & \cdots & \lambda_i^T\lambda_{i-s+1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \lambda_{i-s+1}^T\Lambda_{i+1} & \lambda_{i-s+1}^T\Lambda_i & \lambda_{i-s+1}^T\lambda_{i+1} & \cdots & \lambda_{i-s+1}^T\lambda_{i-s+1} \end{pmatrix} \right) \tag{4.1}$$

with modifications, *mutatis mutandis*, to the definitions of the basis function vectors $\lambda(t)$ and $\Lambda(t)$ defined in (3.19) and (3.20). The corresponding version of the decomposition (3.18) is given by the following equation, and its Bayesian network representation in Figure 4.1.

$$p(Z_{1:N}|\theta,\phi) = \int \left[\prod_{i=0}^{N-1} p(F_i|Z_i,\theta) p(Z_{i+1}|Z_i, \underbrace{F_{i+1}, F_i, \ldots, F_{i-s+1}}_{s+1 \text{ points}}, \phi) \right] \mathrm{d}F_{0:N-1} \tag{4.2}$$

Closer inspection reveals that this approach is inconsistent, because it gives two mutually incompatible meanings to F_{i+1}—one linearly related to Z_{i+1} by the joint Gaussian prior (4.1), and one non-linearly related to Z_{i+1}, resulting from passing its value to the non-linear function $f(\cdot, \theta)$.

In a related context, a similar inconsistency is described at length in a paper by Macdonald et al. [Mac15]. This paper is the culmination of several works whose principal common idea ('gradient matching') was summarised in Section 1.4.2. As described there, these papers aim to minimise the discrepancy between the gradient

of a surrogate data interpolant modelled as a Gaussian process, and the gradient given by the derivative output from the ODE.

In an earlier paper in that paradigm, by Barber & Wang [Bar14], the joint model for these gradients is constructed in a way that treats the derivative $\dot{X}$ as a separate random variable to the solution X. The implied assumption is then that $\dot{X}$ is related to X in two different ways—one linearly through the GP model and one non-linearly through the ODE. This causes a problematic statistical inconsistency, in that X must first be marginalised out to eliminate it, but is then subsequently conditioned upon.

Macdonald et al. [Mac15], who explore this subtle detail at length, propose various convoluted strategies for dealing with this problem, including introducing a dummy variable $\tilde{X}$ to replace X in an attempt to restore the directionality of the joint model, but their approach ultimately founders due to the difficulty of defining the resulting conditional distributions such as $p(\tilde{X}|X)$, $p(X|\tilde{X}, \dot{X})$ and so forth. Ranciati et al. [Ran16, §2.3], whose surrogate model is formed of penalised splines rather than Gaussian processes, also discuss this issue and their proposal also introduces a dummy variable like $\tilde{X}$. By introducing several other assumptions, they thereby identify a restricted set of problems for which the inconsistency can be partially worked around.

Ultimately, the clash between the two meanings of $\dot{X}$ causes an insurmountable consistency issue for the gradient matching approach in the general case. The prior model suggested by (4.1) fails due to the same inconsistency. It is apparent from the product in equation (4.2) that the variable F_{i+1} is assumed both to depend on Z_{i+1}, but is also conditioned upon in the calculation of Z_{i+1}.

A second 'naive' idea would be to sidestep the Gaussian process formulation entirely and directly appeal to Corollary 3.3 by forming a randomised method based on the classical Adams–Moulton integrator. This would require calculating the usual non-probabilistic Adams–Moulton predictor for the next time step t_{i+1} and perturbing it with a realisation of a Gaussian random variable ξ_i with variance constrained in accordance with the convergence result of Theorem 3.

This approach is difficult to implement practically since it requires the exact specification of the Adams–Moulton predictor, a quantity only defined implicitly. Perturbing an inexact quantity throws up several issues around, for instance, how accurately the numerical procedure used for calculating this value is required to be, whether this affects the convergence properties of the perturbed method, and so forth.

Of course, the classical Adams–Moulton integrator must itself advance to the next step based on an approximation to the true predictor.[31] Nevertheless it is hard to justify a procedure which expends significant computation on an intermediate procedure to accurately solve an implicit relation, only to then perturb the solution away again in

an attempt to model numerical error. Lastly, while this approach would depend on the dynamics up to time t_{i+1} through the calculation of the Adams–Moulton predictor, the *error* model for the predictor is still not informed by the system dynamics, being simply a centred Gaussian calibrated in advance.

The idea that forms the core of our novel implicit probabilistic integrator resolves both of these problems. The construction was first introduced in the paper [Tey18], jointly authored with Han Cheng Lie, Tim Sullivan and Ben Calderhead, and presented at the Thirty-Second Conference on Neural Information Processing Systems in Montréal, in December 2018.

4.2 IMPLICIT PROBABILISTIC INTEGRATORS

We introduce the idea in the one-dimensional case first, then later generalise to the multidimensional setting. Consider the following distribution which directly advances the integrator one step and depends only the current point:[32]

$$p(Z_{i+1} = z | Z_i, \theta, \eta) \propto g(r(z), \eta) \tag{4.3}$$

In this equation, $r(z)$ is a positive discrepancy measure *in derivative space* designed to serve as a measure of the error incurred by an implicit multistep method. It is defined in the coming paragraphs. g is a η-scaled functional transformation which ensures that the expression on the right-hand side of (4.3) can, subject to normalisation, be made into a valid probability distribution in the variable z.

A concrete example will illuminate the definition. Consider the first-order implicit linear method—the backward Euler method. This was introduced in Section 3.1.1 and we give its defining relation (3.7) once again:

$$Z_{i+1} = Z_i + hF_{i+1} \tag{4.4}$$

As explained in Section 3.1.1, this expression can typically only be solved by an iterative calculation, since $F_{i+1} \equiv f(Z_{i+1}, \theta)$ is of course unknown. However, if the random variable Z_{i+1} has realised value z, then we may express F_{i+1} as a function of z. Specifically, by rearranging (4.4) we have

$$F_{i+1}(z) = \frac{z - Z_i}{h} \tag{4.5}$$

[31] A discussion of the effect of this phenomenon on the *de facto* global convergence order of implicit integrators can be found in Palais & Palais [Pal09, Appendix I]. In particular, it is possible to quantify the number of fixed-point iterations required to maintain the method's global convergence order. However, it is not obvious how to extend this analysis to a perturbed integrator.

[32] In this section, we use the lower case z to refer to the realised value of the random variable Z_{i+1} representing the next step of the probabilistic integrator.

The discrepancy between the value of $F_{i+1}(z)$ and the value of $f(z,\theta)$ can then be used as a measure of the error in the linear method, and penalised—this is $r(z)$. This discrepancy is effectively the explicit separating out of the two different meanings of F_{i+1} arising from the naive model (4.1). Suppressing for now the explicit dependence on θ and η, we write

$$p(Z_{i+1} = z|Z_i) = \frac{1}{K_h} \exp\left(-\frac{1}{2\eta^2}\left(\frac{z - Z_i}{h} - f(z,\theta)\right)^2\right) \tag{4.6}$$

Here, $K_h > 0$ is the normalising constant of the distribution. Comparing equations (4.3) and (4.6), the gradient discrepancy $r(z)$ is the expression $h^{-1}(z - Z_i) - f(z,\theta)$, and $g : \mathbb{R} \times \mathbb{R}^+ \to \mathbb{R}^+$ is in this case the squared-exponential transformation

$$(u,\eta) \mapsto \exp(-u^2/2\eta^2) \tag{4.7}$$

This approach directly advances the solver in a single leap, without collecting explicit numerical data as in previous approaches. It is in general non-parametric and requires either sampling or approximation to be useful—more on which later. Since $f(\cdot,\theta)$ is in general non-linear in z, clearly r is non-linear in z too. It follows that the density in equation (4.6) does not result in a Gaussian measure despite g being given as a squared-exponential transformation.

The generalisation to higher-order implicit linear multistep methods of Adams–Moulton type follows by writing

$$\begin{aligned} Z_{i+1} &= Z_i + h \sum_{j=-1}^{s-1} \beta^{AM}_{j,s} F_{i-j} \\ &= Z_i + h\left(\beta^{AM}_{-1,s} f(Z_{i+1},\theta) + \sum_{j=0}^{s-1} \beta^{AM}_{j,s} F_{i-j}\right) \end{aligned} \tag{4.8}$$

and then rearranging in the same manner, giving

$$p(Z_{i+1} = z|Z_{\leq i}) = \frac{1}{K} \exp\left(-\frac{1}{2\eta^2}\left(\frac{h^{-1}(z - Z_i) - \sum_{j=0}^{s-1} \beta^{AM}_{j,s} F_{i-j}}{\beta^{AM}_{-1,s}} - f(z,\theta)\right)^2\right) \tag{4.9}$$

By analogy with the distributions $p(Z_{i+1}|Z_i, F_{\leq i}, \phi)$ advancing the explicit probabilistic integrators by one step, we refer to the distributions (4.6) and (4.9) as the 'stepping' or 'step-forward' distribution.

The motivation behind this construction is to circumvent the two main issues outlined in Section 4.1.1. The quantity $r(z)$ measures the difference between the linear and non-linear predictors for the value of the derivative at the next time point. This not only avoids conflating these two distinct values, but specifically implies that the error

in the linear predictor—derived from the rearranged Adams–Moulton relation—can be measured using it.

The construction also means that the local truncation error can be modelled directly, without first calculating an accurate estimate for the classical Adams–Moulton predictor Z_{i+1}^{AM} and perturbing it, since the value of the exact Adams–Moulton predictor does not come into the calculation specifically.

✦ Remark 4.1 The classical Adams–Moulton predictor Z_{i+1}^{AM} is a mode of the distribution (4.9). This is straightforward to see by noting that $r(Z_{i+1}^{AM})$ makes the argument of the exponential zero, which is its maximum possible value. In fact, for small enough h, it is the unique mode of (4.9). The latter statement can be justified as a by-product of the proof of Theorem 4. ✦

4.2.1 *Extension to multidimensional systems*

The extrapolation part of a linear multistep method operates on each component of a multidimensional problem separately. Thus if $Z = (Z^{(1)}, \ldots, Z^{(d)})^T$, we have for the s-step Adams–Moulton method $Z_{i+1}^{(v)} = Z_i^{(v)} + h \sum_{j=-1}^{s-1} \beta_{j,s}^{AM} F_{i-j}^{(v)}$ for each component v in turn. Of course, this is not true of the transformation $Z_{i+1} \mapsto f(Z_{i+1}, \theta) \equiv F_{i+1}$, except in the trivial situation where f is linear in z.

Seen another way, in (3.18) the right-hand distribution in the decomposition is componentwise-independent while the left-hand one is not. All previous probabilistic integrators described in this thesis have treated the multidimensional problem in this way, as a product of one-dimensional relations. Our construction gives us an opportunity to generalise this.

In our proposal it does not make sense to consider the system of equations component-by-component, due to the presence of the non-linear $f(z, \theta)$ term, which appears as an intrinsic part of the stepping distribution $p(Z_{i+1}|Z_{\leq i})$. The multidimensional analogue of (4.9) should take account of this and be defined over all d dimensions together. For vector-valued z, Z_k, F_k, we therefore define

$$\begin{aligned} p(Z_{i+1} = z|Z_{\leq i}) &= \frac{1}{K_h} \exp\left(-\frac{1}{2} r(z)^T H^{-1} r(z)\right) \\ r(z) &= \frac{h^{-1}(z - Z_i) - \sum_{j=0}^{s-1} \beta_{j,s}^{AM} F_{i-j}}{\beta_{-1,s}^{AM}} - f(z, \theta) \end{aligned} \tag{4.10}$$

The quantity $r(z)$ is now a $d \times 1$ vector of discrepancies in derivative space, and H is a $d \times d$ matrix encoding the solver scale, generalising η.[33]

4.2.2 *Analysis of well-definedness and convergence*

The following theorem proves the well-definedness and convergence properties of this new construction. First we show that the density (4.10) is well-defined and proper, by proving the finiteness and strict positivity of its normalising constant K_h. We then describe conditions on the h-dependence of the scaling parameter H, such that an iterative integrator formed by the repeated application of this step-forward distribution possesses the desired convergence properties. In particular, we bound the scale of the second moment of the distribution with density (4.10), allowing us to write our density in a similar form to (3.26), enabling us to appeal directly to Theorem 3.

This theorem presented here is a generalisation to the multidimensional setting of that appearing in Teymur et al. [Tey18].

THEOREM 4 *Consider the initial value problem given by* (1.9)*. Assume the vector field $f(\cdot, \theta)$ is globally Lipschitz with Lipschitz constant L_f. For some end time $t_{\text{end}} > 0$ and time-step $h > 0$ define a grid $t_i = ih$ for $0 \leq i \leq N \equiv \lfloor t_{\text{end}}/h \rfloor$. Denote the solution of* (1.9) *at time t_i by X_i.*

Fix $s \in \mathbb{N} \cup \{0\}$, $\theta \in \mathbb{R}^q$, and denote by $\beta^{AM}_{-1,s}$ the first Adams–Moulton coefficient of order s. If $H = Qh^{2\rho}$ for some $\rho \geq -1$ and positive definite matrix Q, independent of h, having real eigenvalues $0 < q_1 \leq \cdots \leq q_d$, then for $h < (L_f \beta^{AM}_{-1,s} \sqrt{q_d/q_1})^{-1}$ the following statements hold:

(i) The function defined in (4.10) *is a well-defined probability density.*

(ii) Let ξ_i^h be the random variable defined by $Z_{i+1} - \widehat{Z}_{i+1}$, with $\widehat{Z}_{i+1}$ the classical s-step Adams–Moulton predictor for X_{i+1}. Then for every $r \geq 1$, there exists a constant $0 < C_r < \infty$ that does not depend on h, such that for all $i \in \mathbb{N}$, $\mathbb{E}\|\xi_i^h\|^r \leq C_r h^{(\rho+1)r}$.

(iii) If $\rho \geq s + \frac{1}{2}$, the probabilistic integrator defined by (4.10) *converges in mean-square as $h \to 0$, at the same rate as the classical s-step Adams–Moulton method.*

Proof. At iteration i, we define $\widehat{Z}_{i+1}$ to be the output of the s-step Adams–Moulton integrator. We use the index $i+1$ to emphasise that $\widehat{Z}_{i+1}$ is the Adams–Moulton estimate for X_{i+1}. ($\widehat{Z}_{i+1}$ can also be thought of as the image $\Psi_s^h(Z_i, \ldots, Z_{i-s+1})$ of the multidimensional analogue of the numerical flow-map defined in Definition 3.2.) To

[33] It is straightforward to see that if a multidimensional problem *were* treated as a product of one-dimensional distributions like (4.9), the result would be equivalent to a multivariate expression (4.10) with the matrix $H = \mathbb{I}_d$. Viewed this way, it is apparent how this formulation is more general.

be precise, for the s–step Adams–Moulton method, $\widehat{Z}_{i+1}$ is the *exact* solution of the implicit equation

$$\widehat{Z}_{i+1} = Z_i + h\left(\beta_{-1,s}^{AM} f(\widehat{Z}_{i+1}) + \sum_{j=0}^{s-1} \beta_{j,s}^{AM} F_{i-j}\right) \tag{4.11}$$

As described in Section 3.1, the implicit nature of this equation means that, in a numerical sense, $\widehat{Z}_{i+1}$ is only accessible approximately. Note that in (4.11) and in the remainder of the proof, we suppress the θ-dependence of f to improve clarity.

Given a probability space $(\Omega, \mathcal{F}, \mathbb{P})$, we now define for every $i \in \mathbb{N}$ the random variable $\xi_i : \Omega \to \mathbb{R}^d$ according to

$$Z_{i+1} = \widehat{Z}_{i+1} + \xi_i \tag{4.12}$$

where Z_{i+1} is the random variable defined by (4.10). Now choose $\omega \in \Omega$ and define $w := \xi_i(\omega)$. In other words, w is a particular realisation of ξ_i. Hereafter we will omit the explicit ω-dependence. It follows from (4.11) and (4.12) that

$$z = \widehat{Z}_{i+1} + w = Z_i + h\left(\beta_{-1,s}^{AM} f(\widehat{Z}_{i+1}) + \sum_{j=0}^{s-1} \beta_{j,s}^{AM} F_{i-j}\right) + w \tag{4.13}$$

By rearranging the terms in (4.13), we have that

$$\frac{1}{\beta_{-1,s}^{AM}}\left(\frac{z - Z_i}{h} - \sum_{j=0}^{s-1} \beta_{j,s}^{AM} F_{i-j}\right) = f(\widehat{Z}_{i+1}) + \frac{w}{h\beta_{-1,s}^{AM}} \tag{4.14}$$

We now define a norm $\|u\|_Q$ as $\sqrt{u^T Q^{-1} u}$, possible since Q is a positive definite matrix. Due to the equivalence of norms, the Lipschitz condition still holds using this norm, with a different Lipschitz constant $L_{Q,f}$.

The new Lipschitz constant $L_{Q,f}$ can be established by noting that for any $u \in \mathbb{R}^d$, it holds that $(\sqrt{1/q_d})\|u\| \leq \|u\|_Q \leq (\sqrt{1/q_1})\|u\|$, where q_1 and q_d are respectively the smallest and largest eigenvalues of Q, both necessarily positive since Q is positive-definite. It then follows that for any $u, v \in \mathbb{R}^d$,

$$\|f(u) - f(v)\|_Q \leq \sqrt{\frac{1}{q_1}}\|f(u) - f(v)\| \leq \sqrt{\frac{1}{q_1}} L_f \|u - v\| \leq \sqrt{\frac{q_d}{q_1}} L_f \|u - v\|_Q \tag{4.15}$$

and we can thus take $L_{Q,f} = \sqrt{q_d/q_1}\, L_f$.

Returning to (4.14), we subtract $f(z)$ from both sides, take the new Q-norm, and square, to obtain

$$\left\| \frac{1}{\beta_{-1,s}^{AM}}\left(\frac{z - Z_i}{h} - \sum_{j=0}^{s-1} \beta_{j,s}^{AM} f(Z_{i-j})\right) - f(z) \right\|_Q^2 = \left\| f(\widehat{Z}_{i+1}) - f(z) + \frac{w}{h\beta_{-1,s}^{AM}} \right\|_Q^2 \tag{4.16}$$

Noting that $H = Q^{2\rho}$ by assumption, we may thus rewrite (4.10) as

$$p\left(\widehat{Z}_{i+1} + w | \widehat{Z}_{i+1}, Q, h\right) = \frac{1}{K_h} \exp\left(-\frac{1}{2h^{2\rho}} \left\| f(\widehat{Z}_{i+1}) - f(\widehat{Z}_{i+1} + w) + \frac{w}{h\beta_{-1,s}^{AM}} \right\|_Q^2 \right) \tag{4.17}$$

and it follows that the normalising constant is given by

$$K_h = \int_{\mathbb{R}^d} \exp\left(-\frac{1}{2h^{2\rho}} \left\| f(\widehat{Z}_{i+1}) - f(\widehat{Z}_{i+1} + w) + \frac{w}{h\beta_{-1,s}^{AM}} \right\|_Q^2 \right) \mathrm{d}w \tag{4.18}$$

Note that we write K_h to emphasise the fact that the normalising constant depends on the step-size h.

We now bound the function described in (4.17) from above and below by un-normalised Gaussian probability densities. By the triangle inequality and the assumption of global Lipschitz continuity we have the lower bound

$$\begin{aligned} \left\| f(\widehat{Z}_{i+1}) - f(\widehat{Z}_{i+1} + w) + \frac{w}{h\beta_{-1,s}^{AM}} \right\|_Q &\geq \left\| \frac{w}{h\beta_{-1,s}^{AM}} \right\|_Q - \left\| f(\widehat{Z}_{i+1}) - f(\widehat{Z}_{i+1} + w) \right\|_Q \\ &\geq \left\| \frac{w}{h\beta_{-1,s}^{AM}} \right\|_Q - L_{Q,f} \|w\|_Q \\ &= \|w\|_Q \left((h\beta_{-1,s}^{AM})^{-1} - \sqrt{q_d/q_1} L_f \right) \end{aligned} \tag{4.19}$$

where the right-hand side can be seen to be positive since $h < (L_f \beta_{-1,s}^{AM} \sqrt{q_d/q_1})^{-1}$.

Similar reasoning yields the upper bound

$$\left\| f(\widehat{Z}_{i+1}) - f(\widehat{Z}_{i+1} + w) + \frac{w}{h\beta_{-1,s}^{AM}} \right\|_Q \leq \|w\|_Q \left((h\beta_{-1,s}^{AM})^{-1} + \sqrt{q_d/q_1} L_f \right) \tag{4.20}$$

Thus for $h > 0$, there exist constants $c_h, C_h > 0$ that do not depend on ω such that

$$c_h \|w\|_Q^2 \leq \left\| f(\widehat{Z}_{i+1}) - f(\widehat{Z}_{i+1} + w) + \frac{w}{h\beta_{-1,s}^{AM}} \right\|_Q^2 \leq C_h \|w\|_Q^2 \tag{4.21}$$

where $c_h := \left((h\beta_{-1,s}^{AM})^{-1} - \sqrt{q_d/q_1} L_f \right)^2$ and $C_h := \left((h\beta_{-1,s}^{AM})^{-1} + \sqrt{q_d/q_1} L_f \right)^2$.

Equation (4.17) now gives

$$\exp\left(-\frac{C_h w^T Q^{-1} w}{2h^{2\rho}} \right) \leq K_h \cdot p\left(\widehat{Z}_{i+1} + w | \widehat{Z}_{i+1}, Q, h\right) \leq \exp\left(-\frac{c_h w^T H^{-1} w}{2h^{2\rho}} \right) \tag{4.22}$$

Note that the lower and upper bounds do not depend on $\widehat{Z}_{i+1}$. The interpretation of (4.22) is that, up to normalisation, the random variable ξ_i has a Lebesgue density that lies between the densities of two centred Gaussian random variables.

Integrating each of the three terms in (4.22) with respect to w and using the formula for the normalising constant of a Gaussian measure on $\mathbb{R}^d$, we obtain the upper and lower bounds

$$\begin{aligned} K_h &\geq \sqrt{|Qh^{2\rho}|}\left(\frac{2\pi}{C_h}\right)^{d/2} = \sqrt{|Q|}\left(\frac{\sqrt{2\pi}h^{\rho+1}\beta_{-1,s}^{AM}}{1+L_f h\beta_{-1,s}^{AM}\sqrt{q_d/q_1}}\right)^d =: K_{C,h} \\ K_h &\leq \sqrt{|Qh^{2\rho}|}\left(\frac{2\pi}{c_h}\right)^{d/2} = \sqrt{|Q|}\left(\frac{\sqrt{2\pi}h^{\rho+1}\beta_{-1,s}^{AM}}{1-L_f h\beta_{-1,s}^{AM}\sqrt{q_d/q_1}}\right)^d =: K_{c,h} \end{aligned} \tag{4.23}$$

Note that $K_{C,h}$ and $K_{c,h}$ are the normalising constants for the Gaussian random variables $\zeta_{C,h} \sim \mathcal{N}(0, h^{2\rho}C_h^{-1}Q)$ and $\zeta_{c,h} \sim \mathcal{N}(0, h^{2\rho}c_h^{-1}Q)$ respectively. It follows from $\rho > -1$ and $|Q| > 0$ (Q being positive definite) that the upper and lower bounds in (4.23) are respectively finite and strictly positive. This proves statement *(i)*.

To prove *(ii)*, observe that (4.23) yields that, for all $0 < h < (L_f\beta_{-1,s}^{AM}\sqrt{q_d/q_1})^{-1}$ and $\widehat{Z}_{i+1} \in \mathbb{R}^d$, we have

$$1 \leq \frac{K_{c,h}}{K_h} \leq \frac{K_{c,h}}{K_{C,h}} = \left(\frac{C_h}{c_h}\right)^{d/2} = \left(\frac{1+L_f h\beta_{-1,s}^{AM}\sqrt{q_d/q_1}}{1-L_f h\beta_{-1,s}^{AM}\sqrt{q_d/q_1}}\right)^d \tag{4.24}$$

The upper bound decreases to 1 as h decreases to zero, since L_f, $\beta_{-1,s}^{AM}$, q_1, q_d and h are all strictly positive. By the second inequality in (4.22), we have for $r \geq 1$

$$\begin{aligned} \mathbb{E}\left\|\widehat{Z}_{i+1} + \xi_i\right\|^r &= \mathbb{E}\|Z_{i+1}\|^r \\ &= \int_{\mathbb{R}^d} \|z\|^r p(z|\widehat{Z}_{i+1}, H, h)\,\mathrm{d}z \\ &\leq K_{c,h}K_h^{-1}\int_{\mathbb{R}^d} \|z\|^r \exp\left(-\frac{c_h(z-\widehat{Z}_{i+1})^T H^{-1}(z-\widehat{Z}_{i+1})}{2}\right)\mathrm{d}z \\ &= K_{c,h}K_h^{-1}\,\mathbb{E}\left\|\widehat{Z}_{i+1} + \zeta_{c,h}\right\|^r \end{aligned} \tag{4.25}$$

Since the preceding inequalities hold for arbitrary $\widehat{Z}_{i+1} \in \mathbb{R}^d$, we may set $\widehat{Z}_{i+1} = 0$ in (4.17). Using this, and the fact that equation (4.24) implies that $\lim_{h\to 0} K_{c,h}K_h^{-1} = 1$, it is sufficient to show that $\mathbb{E}\|\zeta_{c,h}\|^r \leq C_r h^{(\rho+1)r}$ for some $C_r > 0$ that does not depend on h.

Now consider the change of variables $z \mapsto z' := (c_h H^{-1})^{1/2} z$. By the change of variables formula we have that $\mathrm{d}z = |H|^{1/2}c_h^{-d/2}\,\mathrm{d}z'$, and hence

$$\begin{aligned} &K_{c,h}^{-1}\int_{\mathbb{R}^d} \|z\|^r \exp\left(-\frac{c_h z^T H^{-1} z}{2}\right)\mathrm{d}z \\ &\qquad = \frac{1}{\sqrt{|H|}}\left(\frac{2\pi}{c_h}\right)^{-d/2}\int_{\mathbb{R}^d}\left(\frac{1}{c_h}\right)^{r/2}\|H^{1/2}z'\|^r \exp\left(-\frac{z'^T z'}{2}\right)\sqrt{|H|}\left(\frac{1}{c_h}\right)^{d/2}\mathrm{d}z' \end{aligned}$$

$$\leq (2\pi)^{-d/2} c_h^{-r/2} h^{\rho r} \left|\!\left|\!\left| Q^{1/2} \right|\!\right|\!\right|^r \int_{\mathbb{R}^d} \|z'\|^r \exp\left(-\frac{z'^T z'}{2}\right) \mathrm{d}z'$$

$$\leq C_r h^{(\rho+1)r} \tag{4.26}$$

where $|\!|\!| \cdot |\!|\!|$ is the induced matrix norm (satisfying the sub-multiplicative property $|\!|\!| Ax |\!|\!| \leq |\!|\!| A |\!|\!| \cdot \|x\|$), and C_r does not depend on h. This proves $\mathbb{E}\left\|\xi_i^h\right\|^r \leq C_r h^{(\rho+1)r}$.

To prove *(iii)*, we set $r = 2$ and $\rho \geq s + \frac{1}{2}$ in *(ii)* to obtain $\mathbb{E}\|\xi_i^h\|^2 \leq ch^{(2s+3)}$. Since s is the number of steps of the Adams–Moulton method of order $s+1$, the random variable ξ_i satisfies the assumption in the statement of Theorem 3. It then follows from that result that

$$\max_i \mathbb{E}\|Z_i - X_i\|^2 \leq Ch^{2(s+1)} \tag{4.27}$$

□

4.3 CALIBRATION

Theorem 4 tells us that the integrator defined by the step-forward distribution (4.10) defines an algorithm with the necessary theoretical properties. In short—for the implicit method derived from the s-step Adams–Moulton relation, if the scaling matrix H is chosen to be equal to Qh^{2s+1} for some constant positive definite matrix Q, the theorem tells us the integrator is correctly convergent in the $h \to 0$ limit. It still remains to specify the matrix Q, and to justify that no higher exponent for h should be considered.

This process of attempting to correctly capture the scale of uncertainty in the underlying numerical method by setting algorithm constants recalls the requirement to set α in the explicit multistep integrator defined by Corollary 3.2. It is necessary because the asymptotic convergence results in Theorems 2, 3 and 4 tell us nothing about how to treat their free constants in the finite h setting in which the algorithms operate in reality—indeed, as pointed out by Conrad et al. [Con16], the value of the scale parameter in their method "completely controls the apparent uncertainty in the solver". We call the process of setting these constants *calibration*.

The issue of calibration of probabilistic ODE solvers is addressed without consensus in virtually every treatment of this topic discussed in Chapter 2. Two distinct approaches can be identified. The first are of what we call 'forward' type, in which there is an attempt to directly model the theoretical uncertainty in a solver step and propagate that through the calculation. This can be thought of as trying to derive a precise and explicit generative model for the error in a single step of the classical algorithm, based on a theoretical analysis of the mathematical structure of the integrator in question.

Some summary statistic of this error model then needs to be translated into a function of the scale of the probabilistic integrator, in such a way that the global error arising after repeated application of this process correctly replicates the expected result.

For example, the algorithm suggested in Proposition 3.4 falls into this category. There, the standard deviation of the Gaussian measure representing the uncertainty in the value of Z_{i+1} is calibrated to equal the local truncation error of the underlying classical Adams–Bashforth method. This is conceptually appealing, though further thought exposes several questions which are difficult to resolve.

The local truncation error clearly represents the scale of error in the some sense, though for the reasons given in Section 3.1.5 it is unclear how to interpret this single number when thinking probabilistically. Why equate it with the standard deviation and not, say, the variance? Furthermore, does it matter that the h-scaling suggested by Proposition 3.4 is not tight to the bound provided for by Theorem 3? Does the accumulation of multiple independent local errors—as modelled by Gaussian random variables of this particular scale—resemble a recognisable notion of global error when considered at the end of the time interval of interest? These types of questions are unavoidable in the case that the error model is motivated from first principles in this way.

The second broad category of approaches are those we call 'backward' type, where the uncertainty scale is somehow matched after the computation to that suggested by some external error indicator. In this case, the specific model for the stepwise error is less important, as long as the end result is nevertheless a global uncertainty estimate which properly quantifies the lack of knowledge in the final solution. In this approach, the meaning of individual local errors is demoted in importance in the service of a global error model which is found to be appropriate overall by some specified criterion.

This strategy is by definition less precise, is almost certain to be highly problem dependent, and as a result is very likely to require 'test runs' to properly implement. Nevertheless it circumvents the subtle issue, on which much has been written, of how to tie the scale of local error to that of global error.[34]

[34] The literature being referred to here includes, for example, work on a posteriori global error estimation [Est00]. In this theory, the so-called 'stability factor' of the ODE system—a quantity analogous to the condition number of a matrix—is estimated by solving a linearised adjoint system. This factor is then used in combination with the complete sequence of local error estimates to give a global error estimate. Several other lines of research attacking the same problem also exist. The main point is that the precise connection of local to global error is in general a highly non-trivial problem—accepting this fact helps justify the proposal to calibrate global error without being too concerned that individual local errors be fully interpretable.

The proposal in Conrad et al. [Con16] falls into this category and we use a modified version of their idea in our own simulations in Chapter 5. As a consequence, we now describe this procedure—which we term scale matching—in detail.

4.3.1 *Calibration by scale matching – explicit methods*

The core principle of the scale matching approach is to calibrate a probabilistic integrator by computing a running global error estimate of the underlying classical method and setting the scaling constant in the stepping distribution of the probabilistic integrator so as to replicate the scale of error suggested by this indicator.

Consider a classical numerical method of order p with numerical flow map Ψ^h. One of the simplest global error indicators compares the output of the same method applied once with step-size h and—in parallel—twice with step-size $h/2$ [Pal09, §5.4]. The difference between the two outputs is taken to be an estimate of the error in the coarser run. Recalling the definition of error from Section 3.2, an estimate $\widehat{E}_i$ for the global error E_i of the classical method at time t_i is given by this principle as

$$\widehat{E}_i = (\Psi^h \circ \overset{i}{\cdots} \circ \Psi^h)(X_0) - (\Psi^{h/2} \circ \overset{2i}{\cdots} \circ \Psi^{h/2})(X_0) \tag{4.28}$$

Using the notation $Z_i^{\text{classic}} = (\Psi^h \circ \overset{i}{\cdots} \circ \Psi^h)(X_0)$, so that $Z^{\text{classic}} \equiv Z_{0:N}^{\text{classic}}$ is the complete discrete trajectory of the classical integrator, Conrad et al. [Con16] form a sequence of Gaussian measures, one for each step i.[35]

$$\kappa_i^{\text{classic}} = \mathcal{N}(Z_i^{\text{classic}}, \operatorname{diag}(\widehat{E}_i^2)) \tag{4.29}$$

Recall that linear methods operate on each component of a multidimensional problem independently—hence the diagonal form of the covariance matrix in (4.29).

The stepwise perturbations ξ_i of the corresponding probabilistic method, each independent and with variance $S := \alpha h^r \cdot \mathbb{I}_d$, are then scaled—by setting the constants α and r—such that the resulting global error matches the measure $\kappa_i^{\text{classic}}$ as closely as possible. This is achieved by running several repetitions of the integrator with different sample instantiations $\omega^{[1]}, \ldots, \omega^{[K]} \in \Omega$, thereby collecting a Monte Carlo ensemble of random trajectories $Z_{0:N}^{[1]}, \ldots, Z_{0:N}^{[K]}$, and then for each step forming another Gaussian measure

$$\kappa_i^{\text{prob}} = \mathcal{N}(\mathbb{E}_\xi(Z_i), \operatorname{Var}_\xi(Z_i)) \tag{4.30}$$

Here we have used the ξ subscripts to denote that these are sample statistics calculated from the Monte Carlo sample $Z_i^{[1]}(\xi), \ldots, Z_i^{[K]}(\xi)$. We remark that $\operatorname{Var}_\xi(Z_i)$ will of course depend on the scaling parameter α and the h-exponent r.

[35] We have used the slightly loose notation $\widehat{E}_i^2$ to denote the entry-wise square of the vector $\widehat{E}_i$.

We first fix r such that it equals the minimum possible value consistent with the statement of Theorem 3. For the probabilistic s-step Adams–Bashforth integrator, this gives $r = 2s + 1$.[36] The next step is to define some measure of distance between the two distributions $\kappa_i^{\text{classic}}$ and κ_i^{prob} for each time step i. The product of these distances over all time steps is then penalised, forming a probability measure over α.

In Conrad et al. [Con16], the measure of distance chosen is Bhattacharyya distance [Bha46]. For two multivariate normal distributions κ_1 and κ_2 with parameters (μ_1, Σ_1) and (μ_2, Σ_2) respectively, this is defined as

$$\delta(\kappa_1, \kappa_2) = \frac{1}{16}(\mu_1 - \mu_2)^T(\Sigma_1 + \Sigma_2)(\mu_1 - \mu_2) + \frac{1}{2}\log\frac{\det\frac{1}{2}(\Sigma_1 + \Sigma_2)}{\sqrt{\det\Sigma_1 \det\Sigma_2}} \tag{4.31}$$

The product of this expression over all time steps can then be penalised by defining a function $B(\alpha)$ by

$$B(\alpha) \propto \prod_{i=1}^{N} \exp\left(-\delta\left(\kappa_i^{\text{classic}}, \kappa_i^{\text{prob}}(\alpha)\right)\right) \tag{4.32}$$

(We have specifically highlighted the dependence of the empirical measure κ_i^{prob} on the scaling parameter α that we are in the process of setting.)

Later, we subtly modify this construction in a number of ways to better suit our purposes. Firstly, we experiment with eliminating the first term of (4.31) to remove the dependence on the means μ_i. This gives a modified expression

$$\delta'(\kappa_1, \kappa_2) = \frac{1}{2}\log\frac{\det\frac{1}{2}(\Sigma_1 + \Sigma_2)}{\sqrt{\det\Sigma_1 \det\Sigma_2}} \tag{4.33}$$

We find that solely penalising differences in variance can give a more stable calibration procedure, especially in cases where a consistent bias in the algorithm results in the term containing the means dominating the penalisation. When this occurs, $\delta(\kappa_1, \kappa_2)$ inflates in order to capture the bias, rather than scaling properly to the variance, and we find that using $\delta'(\kappa_1, \kappa_2)$ prevents this.

Secondly, we find that the expression (4.32) suffers from instability when any of the individual multiplicands $\delta\left(\kappa_i^{\text{classic}}, \kappa_i^{\text{prob}}\right)$ are close to zero, since their contribution to the product on the right-hand side becomes extremely large. We find this occasionally occurs in higher-order probabilistic methods when the variances Σ_1 and Σ_2 are very small, or sometimes near turning points in $f(\cdot, \theta)$ where locally—often for a single time ordinate t_i only—$\widehat{E}_i$ happens to be very close to zero. We therefore experiment

[36] In Chapter 5 we provide numerical evidence to support this approach.

with using the median modified Bhattacharyya distance, rather than the product, to form an alternative penaliser D' as a function of α. This gives

$$D'(\alpha) \propto \underset{1 \leq i \leq N}{\text{median}} \exp\left(-\delta'\left(\kappa_i^{\text{classic}}, \kappa_i^{\text{prob}}(\alpha)\right)\right) \tag{4.34}$$

While a complete Bayesian analysis would require α to be inferred jointly as a full unknown in the posterior model, Conrad et al. choose instead to take the maximising argument α^* of the function $D(\alpha)$ and fix it, in a manner akin to empirical Bayes [Rob56]. This greatly simplifies the implementation, and can be justified on those grounds. (The rationale for departing from a fundamentalist Bayesian approach in decisions such as this was addressed in Section 2.1.1.)

4.3.2 *Calibration by scale matching – extension to implicit methods*

We now generalise the approach of Conrad et al. in order to calibrate the implicit probabilistic integrators introduced in this chapter. Recall that for the integrator derived from the s-step Adams–Moulton method, Theorem 4 tells us that the scaling matrix H should be chosen to be equal to Qh^{2s+1} for some constant matrix Q.[37] We have applied the same heuristic here as for the explicit method—that of setting the exponent of h to be as small as possible while still remaining consistent with the theoretical bound. Once again, our simulations in Chapter 5 provide strong evidence in support of this choice.

For the explicit methods, the variance $S = \text{Var}(Z_{i+1}|Z_{\leq i}, \theta, \phi)$ of the step-forward distribution was set by forming an α-scaled diagonal matrix, determined by a scale-matching procedure that ensures that the integrator outputs a global error scale in line with expectations.

For the implicit methods, we are not able to relate such a matrix S directly to H because from the definition (4.10) it is clear that H is a scaling matrix for the spread of the *derivative* $f(Z)$, whereas the S in the explicit methods measures the spread of the *state* Z. In order to transform to the correct space without linearising the ODE, we apply the multivariate delta method [Oeh92] to give an approximation for the variance of the transformed random variable, and set H to be equal to the result. Thus

$$\begin{aligned} H &= \text{Var}(f(Z_{i+1})|Z_{\leq i}) \\ &\approx J_f(\mathbb{E}(Z_{i+1}|Z_{\leq i}))\, \text{Var}(Z_{i+1}|Z_{\leq i})\, J_f(\mathbb{E}(Z_{i+1}|Z_{\leq i}))^T \\ &= \alpha h^{2s+1} J_f(\mathbb{E}(Z_{i+1}|Z_{\leq i}))\, J_f(\mathbb{E}(Z_{i+1}|Z_{\leq i}))^T \end{aligned} \tag{4.35}$$

[37] Recall the contrasting meanings of s for explicit and implicit Adams methods—see Remark 3.3.

where J_f is the Jacobian of f defined for $U \in \mathbb{R}^d$ by

$$J_f(U) = \left(\begin{array}{ccc} \frac{\partial f^{(1)}}{\partial x^{(1)}} & \cdots & \frac{\partial f^{(1)}}{\partial x^{(d)}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f^{(d)}}{\partial x^{(1)}} & \cdots & \frac{\partial f^{(d)}}{\partial x^{(d)}} \end{array} \right) \Bigg|_{x = U} \tag{4.36}$$

We assume that the Jacobian can be evaluated in closed form—this requires analytic first derivatives of f component-wise. The evaluation point $\mathbb{E}(Z_{i+1}|Z_{\leq i}, \theta, \phi)$ required by the delta method is unknown but we can use an explicit method of equal or higher order to compute an estimate Z_{i+1}^{AB} at negligible cost, and use $J_f(Z_{i+1}^{AB})$ instead.

With reference to the latter point, we reiterate that we are roughly calibrating the method so some level of approximation is unavoidable. This comment applies equally to the case where the Jacobian is not analytically available and must be estimated numerically. It is important to note that such approximations do not affect the fundamental asymptotic convergence properties of the algorithm, since they do not affect the h-scaling of the step-forward distribution.

We also note that we are merely matching variances/spread parameters and nowhere assuming that the distributions in question are Gaussian. Specifically, the H suggested in (4.35) is a statistic representing the spread of the distribution (4.10), but since this distribution is non-Gaussian, H is not precisely its variance.

The transformation having been made and H constructed as in equation (4.35), the overall scaling constant α can then be set in the same way as in the explicit approach, via scale-matching the global error indicator to the empirical measure output by the probabilistic integrator.

This construction has the beneficial consequence of giving a non-trivial cross correlation structure to the error calibration matrix, allowing a richer description of the error in multidimensional problems, something absent from previous approaches. Furthermore, it derives this additional information via direct feedback from the ODE beyond the current time t_i, which we have claimed is a desirable attribute.

✦ Remark 4.2 The idea presented in this section bears some similarity to the original concept in Skilling [Ski91], in which a scalar 'stiffness constant' is used to transform the uncertainty scale from solution space to derivative space, in a similar way to how we have employed the Jacobian J_f. ✦

✦ Remark 4.3 Throughout the previous two sections we have assumed that the matrix $S = \mathrm{Var}(Z_{i+1}|Z_{\leq i}, \theta, \phi)$ is proportional to the identity matrix $\mathbb{I}_d$. This is in accordance with the approach of Conrad et al. [Con16]. Of course, we could seek to calibrate a more general positive definite matrix Q.

We experimented extensively with this idea, including setting Q proportional to local error indicators (such as the Milne Device [But08, §245]), the global error scale at end of the interval (to allow for multi-scale errors across dimensions), or using ideas from the theory of a posteriori error estimation [Est95; Cao04] (which combines the two). In all cases, we found at best minor improvements in performance but large increases in computational cost and programming complexity.

The first of these—using local error estimates to locally adjust the scale of the perturbations at each time-step—is intuitive, though it exhibits a problematic lack of robustness at higher orders, since local errors are often close to zero in intervals of the problem where $f(\cdot, \theta)$ happens to be locally close to polynomial. The second—calibrating different α for each dimension of a problem—is a promising approach. It may be that in a problem with wildly differing scales in different components, effort should be refocused on the use of type of multi-scale calibration. Further investigation of this topic is one of the avenues of future research we suggest in Section 6.2.2. ✦

4.4 Implementation

The stepping distribution defined by the density (4.10) is non-parametric—indeed this was highlighted as one of its strengths, since it incorporates direct feedback from the system dynamics in the interval $[t_i, t_{i+1}]$. However, the fact that it is not of a standard form means that the question of how to evaluate it—or sample from it—is not trivial. In this section we cover several possible approaches to this issue, in anticipation of our simulations in the next chapter.

4.4.1 Forward simulation

The core principle behind randomised IVP solvers—first described in Section 2.2—is to run repeated instantiations of the solver with different random seeds $\omega \in \Omega$ in order to generate an empirical measure from the ensemble of trajectories, reflecting the uncertainty in the underlying integrator. This requires us to sample from the step-forward measure at each time step. In the case of a non-parametric distribution, no exact way exists to do this. As explained in Section 1.2.2, the only viable alternative is to simulate from the distribution using a stochastic approximation method such as Monte Carlo.

Implementing such an algorithm-within-an-algorithm allows us to draw a sample from the distribution (4.10). Nevertheless, there is clearly a significant computational penalty associated with this approach, particularly since any sampling procedure needs to occur at every time-step i and, furthermore, is not paralellisable due to the inherently sequential nature of an iterative IVP integrator.

There are several ways of performing this sampling. MCMC—using Metropolis-Hastings [Met53; Has70] or some extension thereof using additional derivative information [Rob02; Gir11]—is likely to require careful tuning, adaptation [Haa01] and close diagnostic attention to ensure the Markov chain is ergodic and the samples drawn are not biased. Markov chain-based algorithms such as these can be very poor samplers if care is not taken to ensure they are properly set-up.

An alternative approach is to perform rejection sampling [Von51] using the bounding Gaussian density (4.22) which arises in the proof of part *(i)* of Theorem 4. This approach gives an explicit covering distribution for the rejection sampler, avoiding algorithm tuning issues, and moreover has the potential to output independent samples. However, the dependence of the density bound in (4.22) on the global Lipschitz constant L_f means that the 'covering constant' [Liu01, §2.2] of the rejection sampler may be very large. This will certainly be the case if L_f is itself large, and probably also if it is unknown and needs to be (conservatively) estimated. In such a situation the rejection sampler, while theoretically correct, may turn out to be an extremely inefficient way of generating samples.

Instead, we experiment with an approach based on the pre-conditioned Crank–Nicolson (pCN) algorithm [Cot13]. The nature of the distributions to be sampled from—low-dimensional and 'close' to Gaussian—allows us to employ a trick which greatly increases the sampling efficiency of the Markov chain. In order to do this, a close Gaussian approximation to the target density (4.10) must be found. We outline a method for constructing such an approximation in the next section; for this reason, we defer the detailed discussion of this method to Section 4.4.3.

4.4.2 Gaussianisation

While sampling-based methods allow for the algorithm to proceed using the exact non-parametric step-forward distribution (4.10)—at least asymptotically, since there will inevitably be stochastic errors introduced by any finite-length Monte Carlo procedure—the only way to completely avoid the additional cost penalty is by reverting to distributions of standard form, from which samples can easily be drawn. The most straightforward approach is to approximate (4.10) by a Gaussian distribution—depending on how this approximation is performed, the desideratum of maintaining information feedback from the future dynamics of the target function can be maintained.

Consider a Taylor expansion for $f(z, \theta)$, truncated after the first-order term:

$$\begin{aligned} f(z,\theta) &= f(Z_i,\theta) + f'(Z_i,\theta)(z - Z_i) + \text{higher order terms} \\ &\approx f(Z_i,\theta) + f'(Z_i,\theta)(z - Z_i) \end{aligned} \tag{4.37}$$

By Taylor's theorem for vector-valued functions, $f'(Z_i, \theta)$ is the total derivative of the vector function f with respect to vector x and so is the Jacobian $J_f(Z_i)$ given by (4.36).

In the same way as during the calibration process, we assume that the Jacobian can be evaluated in closed form, which requires analytic first derivatives of f componentwise. (As there, it may also be that a numerical approximation to the Jacobian can act as a feasible alternative.) The evaluation point Z_i is, of course, the current position of the sampler and hence known. We then have from equation (4.10)

$$\begin{aligned} r(z) &\approx \frac{h^{-1}(z - Z_i) - \sum_{j=0}^{s-1} \beta_{j,s}^{AM} F_{i-j}}{\beta_{-1,s}^{AM}} - f(Z_i,\theta) - J_f(Z_i)(z - Z_i) \\ &= \Gamma(z - w) \\ &=: \tilde{r}(z) \end{aligned} \tag{4.38}$$

where we have defined the matrix $\Gamma \in \mathbb{R}^{d\times d}$ and vector $w \in \mathbb{R}^d$ by

$$\begin{aligned} \Gamma &= (h\beta_{-1,s}^{AM})^{-1}\mathbb{I}_d - J_f(Z_i) \\ w &= Z_i + \Gamma^{-1}\left(\sum_{j=0}^{s-2} \frac{\beta_{j,s}^{AM}}{\beta_{-1,s}^{AM}} F_{i-j} + f(Z_i)\right) \end{aligned} \tag{4.39}$$

The approximation $\tilde{r}(z)$ is linear in z and hence yields a non-centred Gaussian when substituted for $r(z)$ and transformed into a probability measure via the modified transformation g mapping $(u, H) \mapsto \exp(-\frac{1}{2}u^T H^{-1}u)$ as in (4.10). Some straightforward algebra gives the moments of this approximating Gaussian measure as

$$p_{\text{approx}}(Z_{i+1} = z | Z_{\le i}) = \mathcal{N}(z; w, \Gamma^{-1}H\Gamma^{-T}) \tag{4.40}$$

We note that this procedure is merely to facilitate straightforward sampling—though $\tilde{r}(z)$ is linear in z, the inclusion of the first additional term from the Taylor expansion means that information about the non-linearity (in z) of $f(\cdot, \theta)$ is still incorporated to second order, and the generated solution Z is not jointly Gaussian across time steps i. Furthermore, since Γ^{-1} is order 1 in h, this approximation does not impact the global convergence of the integrator, as long as H is set in accordance with the principles described in Section 4.3.

Another significant benefit of this approach is that it allows a simulation to be run having pre-sampled a random seed $\omega \in \Omega$. The sampled $\omega^{[k]}$ generates the complete

sequence of step-wise perturbations $\xi(\omega^{[k]}) \equiv \xi^{[k]}$ in advance of the integration. This is indispensable in the inverse problem context, where an MCMC algorithm performing posterior inference over the model parameters θ—in the setup described in Section 1.4—will mix much better if the sampled sequence $\xi^{[k]}$ is reused from step to step. We will provide significant further detail on this issue in Chapter 5.

✦ Remark 4.4 This approximation (4.37) is similar to the method of solving implicit IVP integrators by linearising them [Pre07, §17.5]. In this way, it can be seen that the underlying method here has effectively been turned into a probabilistic version of a so-called semi-implicit multistep method of the form

$$Z_{i+1} = Z_i + h\left[\left(\frac{1}{\beta^{AM}_{-1,s}} - hJ_f(Z_i)\right)^{-1}\left(f(Z_i) + \sum_{j=0}^{s-2}\beta^{AM}_{j,s}F_{i-j}\right)\right] \tag{4.41}$$

Another name for this family of methods is the Rosenbrock family[Hai10, §4.7; Pre07, §17.5.1], though most expositions concentrate on the linearisation of multi-stage Runge–Kutta methods rather than multistep methods. ✦

✦ Remark 4.5 To avoid confusion, we point out the difference between the approximating Gaussian (4.40) just defined and the bounding Gaussian (4.22) which we noted in Section 4.4.1 could be used to construct a rejection sampler. The former is correctly normalised and is thus a true density; furthermore there are no guarantees that it bounds (or 'covers') the stepping distribution. Indeed, it cannot do so unless the two distributions are identically equal. The latter is a scaled version of a Gaussian density and is thus un-normalised. As a result, it cannot be assumed to be a good approximation of the density of the stepping distribution unless its normalising constant is known—and it may be a poor approximation regardless. ✦

4.4.3 Pre-conditioned Crank–Nicolson MCMC

In this section we introduce an MCMC algorithm which is a modification of the standard random walk Metropolis–Hastings algorithm, and that has the key property of high efficiency when the distribution being targeted is close to Gaussian. In our case, we are able to exploit the Gaussian approximation (4.40) to construct an efficient MCMC algorithm which samples exactly from the step-forward distribution (4.10), without the requirement to substitute the Gaussian approximation itself.

The underlying concept—in the context of infinite-dimensional samplers—was introduced by Beskos et al. [Bes08] under a different name, then expanded and formalised by Cotter et al. [Cot13] who labelled it the pre-conditioned Crank–Nicolson algorithm. Several relevant theoretical results are given by Pillai et al. [Pil11]. A variant,

focusing on finite-dimensional distributions, was considered in the author's Masters thesis [Tey14]. We describe and implement the latter version here.

The construction of the algorithm takes as its starting point the d-dimensional stochastic differential equation

$$\mathrm{d}U(t) = -U(t)\,\mathrm{d}t + \sqrt{2M}\,\mathrm{d}B_t \tag{4.42}$$

with $U(t)$ a stochastic process, M a positive semidefinite $d \times d$ matrix and B_t a standard Brownian motion. This defines the so-called the Ornstein–Uhlenbeck process [Kar91, §5.6A] and has stationary solution $U(t) \sim \mathcal{N}(0, M)$. Discretisation of this equation with step-size $h > 0$ is achieved using the Crank–Nicolson method [Cot13], resulting in an iterative relation

$$U_{i+1} - U_i = -h\frac{U_i + U_{i+1}}{2} + \sqrt{2h} \cdot \zeta_i \tag{4.43}$$

where $\zeta_i \overset{\text{iid}}{\sim} \mathcal{N}(0, M)$. Rearranging this expression gives

$$\begin{aligned} U_{i+1} &= \frac{1 - h/2}{1 + h/2}\,U_i + \frac{\sqrt{2h}}{1 + h/2} \cdot \zeta_i \\ &= \sqrt{1 - \gamma^2}\,U_i + \gamma \cdot \zeta_i \end{aligned} \tag{4.44}$$

where we have defined $\gamma = \sqrt{2h}(1 + h/2)^{-1}$. This relation defines a Markov chain. As long as it is possible to generate samples ζ_i from $\mathcal{N}(0, M)$, successive states U_i will be exact samples from the stationary distribution $\mathcal{N}(0, M)$.

This is clearly circular and not useful in itself, but the key insight is that the introduction of a Metropolis step allows any distribution to be sampled from exactly, and if that distribution is close to Gaussian then the $\mathcal{N}(0, M)$-stationarity of the pCN algorithm means that proposals can have a very high probability of being accepted.

Consider now the target distribution $p(Z_{i+1}|Z_{\leq i})$ from (4.10). Calculating a close Gaussian approximation $p_{\text{approx}}(Z_{i+1}|Z_{\leq i})$ to it as in (4.40), and offsetting them both by the mean w of the approximation, allows us to write the trivial equality

$$p(Z_{i+1} - w|Z_{\leq i}) = \frac{p(Z_{i+1} - w|Z_{\leq i})}{p_{\text{approx}}(Z_{i+1} - w|Z_{\leq i})}\,p_{\text{approx}}(Z_{i+1} - w|Z_{\leq i}) \tag{4.45}$$

The pCN algorithm draws samples from the centred Gaussian $p_{\text{approx}}(Z_{i+1} - w|Z_{\leq i}) \equiv \mathcal{N}(0, \Gamma^{-1}H\Gamma^{-T})$ and outputs samples from the distribution $p(Z_{i+1} - w|Z_{\leq i})$ by exploiting this relation. These samples can then be trivially transformed back to the target distribution $p(Z_{i+1}|Z_{\leq i})$ by adding w to each one individually.

pCN ALGORITHM FOR SAMPLING $p(Z_{i+1}|Z_{\leq i})$

1 INPUT Mean w and variance Λ of Gaussian approximation $\mathcal{N}(w, \Lambda)$ to $p(Z_{i+1}|Z_{\leq i})$
2 INPUT $U^{[1]}$ (w is a good choice)
3 INPUT $\gamma \in (0, 1]$
4 FOR $1 \leq k \leq K$
5 $\quad \xi^{[k]} \sim \mathcal{N}(0, \Lambda)$
6 $\quad U^* \leftarrow \sqrt{1-\gamma^2}\,(U^{[k]} - w) + w + \gamma\xi^{[k]}$
7 $\quad \alpha^{[k]} \leftarrow \min\left(1, \dfrac{p(U^*|Z_{\leq i})\mathcal{N}(U^{[k]}; w, \Lambda)}{p(U^{[k]}|Z_{\leq i})\mathcal{N}(U^*; w, \Lambda)}\right)$
8 $\quad r^{[k]} \sim \mathcal{U}[0, 1]$
9 $\quad$ IF $r^{[k]} < a^{[k]}$
10 $\quad\quad U^{[k+1]} \leftarrow U^*$
11 $\quad$ ELSE
12 $\quad\quad U^{[k+1]} \leftarrow U^{[k]}$
13 $\quad$ END
14 $\quad k \leftarrow k + 1$
15 END
16 OUTPUT $U^{[2]}, \ldots, U^{[K]}$

Algorithm 1: Pseudo-code for the modified pre-conditioned Crank–Nicholson sampler for generating K samples from the step-forward distribution (4.10) of the probabilistic Adams–Moulton integrator. The parameters w and Λ of the required Gaussian approximation are calculated using the procedure in Section 4.4.2 and given by equation (4.40).

Roughly speaking, if p_{approx} is close to p, then p/p_{approx} will be close to 1. This fact, combined with the algorithm's stationarity with respect to p_{approx}, means that almost all proposed samples are accepted. Furthermore for values of γ close to 1, successive samples will be only very weakly dependent. These features mean the algorithm results in a very efficient MCMC sampler for this class of target distributions.[38] Pseudo-code describing the implementation of the algorithm is given in Algorithm 1.

The parameter $\gamma \in (0, 1]$ controls the extent to which the proposal distribution on line 6 makes Algorithm 1 approximate a Gaussian independence sampler. On one extreme, we see that as $\gamma \to 0$, the Markov chain tends to a deterministic sequence with each successive sample equal to its antecedent. The opposite extreme, $\gamma \to 1$, diminishes the influence of the current state such that at the limit $\gamma = 1$ proposals U^* are indeed independently drawn from the Gaussian distribution $\mathcal{N}(0, \Lambda)$.

[38] Cotter et al. [Cot13, §4.2] refer to the pCN algorithm as a "natural generalisation of random walks to the setting where the target measure is defined via density with respect to a Gaussian". Under this view, the ratio p/p_{approx} can be thought of as the *de facto* target measure.

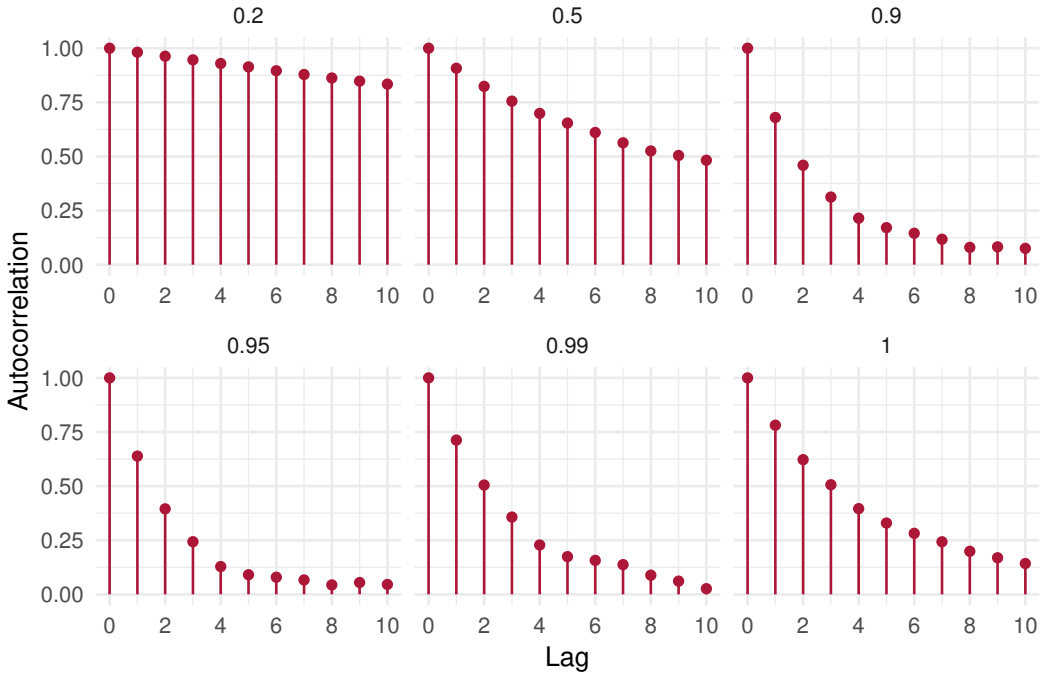
In our simulations, we find that values of γ of 0.95 or even larger still give very good performance. This indicates that the step-forward distributions we consider are close enough to Gaussian that the pCN algorithm can be effectively operated at extremely high efficiency.

We give some indicative plots of the effect of varying γ in figure 4.2. These data were generated during several runs of the probabilistic implicit Euler integrator applied to the FitzHugh–Nagumo system to be properly introduced in Section 5.1.1. The step-size is $h = 0.1$ and the data are drawn at the point when the integrator has reached $t_{60} = 6.0$ and is in the process of advancing to $t_{61} = 6.1$. For each of several values of γ, 1000 samples were generated from the step-forward distribution based on the backward Euler method (4.6) using the pCN algorithm shown in Algorithm 1.

The results are tabulated below figure 4.2. The second column gives the sample acceptance rate, while the third column shows the equivalent number of independent samples generated per iteration, as measured by dividing the effective sample size by the total number of samples collected.[39] The latter statistic is a measure of the information content of the Markov chain, in the sense that it summarises the autocorrelation sequence by a number between 0 and 1, where a sequence of independent samples would return a value of 1 and a sequence of fully correlated samples—generated by a deterministic transition kernel, for example—would return 0. The output shown relates to the first state dimension $x^{(1)}$ of the two-dimensional test problem, though results for $x^{(2)}$ are similar.

These results support the suggestion that $\gamma = 0.95$ is a reasonable value to choose for our simulations, under the assumption that broadly similar behaviour would result across iterations i of the complete integration interval. The samplers with the smallest values of γ exhibit relatively high autocorrelation, drastically increasing the number of samples required in order to generate a single one which is uncorrelated to the starting value. Those with large γ suffer from relatively poor acceptance rates which *itself* negatively affects the sample autocorrelation, since rejected samples are fully correlated with their antecedent.

[39] The effective sample size is calculated using the initial convex sequence estimator of Geyer [Gey92]. The formula for the equivalent number of independent samples per iteration is given by $\text{EIS} = (1 + 2\sum_{j=1}^{m} s_j)^{-1}$, where the s_j are the autocorrelation values at lag j, and $m \leq \infty$ is a truncation point determined in such a way as to capture as much of the signal as possible from the autocorrelation sequence, yet exclude the noise at high lags.



γ	Acceptance Rate (%)	Equivalent Independent Samples
0.2	88.3	0.023
0.5	70.2	0.042
0.9	61.6	0.168
0.95	59.6	0.229
0.99	48.6	0.170
1	37.6	0.111

Figure 4.2: Autocorrelation plots for lags 0–10 for the pCN sampler applied to the step-forward distribution (4.6) at the 60th iteration of a probabilistic backward Euler integrator solving the FitzHugh–Nagumo system (defined in Section 5.1.1) with step-size $h = 0.1$. Different values of γ are compared. The table gives the acceptance rate and equivalent number of independent samples calculated from samples of the first state variable $x^{(1)}$.

5

SIMULATION STUDIES

5.1 INTRODUCTION

In this chapter we explore the application of the methods introduced in this thesis to two simple ODE models, and discuss the benefits and shortcomings suggested by the results. Firstly we introduce these test models, then undertake the integrator calibration process described in Section 4.3, and finally apply our methods in the context of Bayesian parameter estimation.

5.1.1 FitzHugh–Nagumo model

For our first running example we choose a well-known dynamical model of two variables, the FitzHugh–Nagumo model. This model was proposed by FitzHugh [Fit61] and Nagumo et al. [Nag62] as a two-dimensional simplification of the Hodgkin–Huxley model [Hod52], which was itself introduced to model the firing mechanism of neurons in squid. Viewed a different way, the FitzHugh–Nagumo model can be seen as a generalisation of the well-known Van der Pol oscillator. It has been extensively studied both due to its important scientific interpretation but also for its interesting mathematical behaviour. Extensive details are given in the book-length study by Rocsoreanu et al. [Roc12].

We consider the following form of the model, which has been treated in several previous papers in the ODE parameter estimation and probabilistic numerics literature [Ram07; Cal09; Cam12; Jen12; Con16; Sch18]:

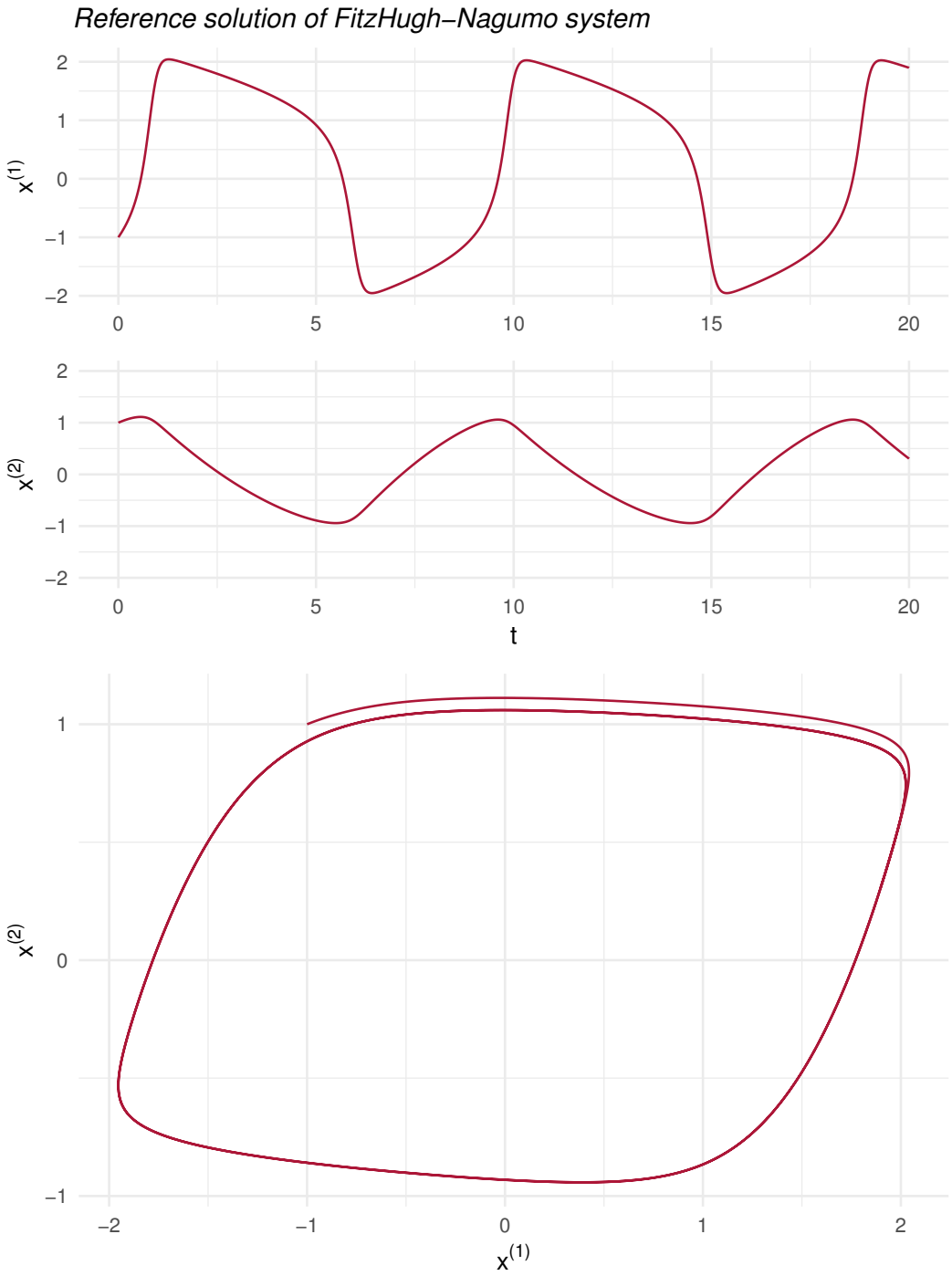


Figure 5.1: High-accuracy solution of the FitzHugh–Nagumo initial value problem defined by (5.1) with parameters $\theta = (0.2, 0.2, 3)$ and initial value $X_0 = (-1, 1)^T$ for the range for $0 \leq t \leq 20$. The top two panes show the solution as a time series while the bottom pane shows the trajectory in phase space.

$$\begin{aligned}\frac{\mathrm{d}V}{\mathrm{d}t} &= c\left(V - \frac{V^3}{3} + R\right)\\ \frac{\mathrm{d}R}{\mathrm{d}t} &= -\frac{1}{c}\big(V - a + bR\big)\end{aligned} \tag{5.1}$$

The two state variables V and R represent, respectively, the electrical potential across the cell membrane, and an auxiliary variable depending on the refractory period after the neuron's firing. The parameters a and b are related to the number of channels of the cell membrane which are opened to the Na^+ and K^+ ions [Roc12] and c adjusts the scale between the two variables. The original model in Fitzhugh [Fit61] restricts these parameters to the ranges $a \geq 0$, $0 \leq b < 1$ and $c > 0$. Stable cyclical behaviour occurs for parameter values additionally satisfying $|a| \leq 0.8$, $|b| \leq 0.8$ and $c < 8$ [Cam07, §1.2].

In our study we consider this model in an abstract way and as such to be consistent with our earlier notation we collect the state variables as $x \equiv (V, R)^T$ and the parameters as $\theta \equiv (a, b, c)$. We consider the parameter choice $\theta = (0.2, 0.2, 3)$ and initial value $X_0 = (-1, 1)^T$. These choices result in the dynamics shown in figure 5.1.

5.1.2 *Brusselator model*

Our second example is the Brusselator system, introduced by Lefever & Nicolis [Lef71]. This models the autocatalytic reaction of a mix of chemical substances which, with certain parameter choices, results in the concentrations of the individual molecules exhibiting periodicity. The equations governing the dynamics follow directly from the chemical law of mass action, though once again we consider the system abstractly. Its behaviour has been studied in, for example, Hairer et al. [Hai08, §I.16] and it was also used as an example—albeit with different parameter values to here—in a recent probabilistic numerics paper [Sch18]. The defining equations are:

$$\begin{aligned}\frac{\mathrm{d}x^{(1)}}{\mathrm{d}t} &= \theta^{(1)} + (x^{(1)})^2 x^{(2)} - (\theta^{(2)} + 1)x^{(1)}\\ \frac{\mathrm{d}x^{(2)}}{\mathrm{d}t} &= \theta^{(2)} x^{(1)} - (x^{(1)})^2 x^{(2)}\end{aligned} \tag{5.2}$$

This time we have written the equations directly in our standard notation. The system only exhibits 'interesting' dynamics[40] if $\theta^{(2)} > (\theta^{(1)})^2 + 1$. We therefore take the parameter to be $\theta = (1.4, 3)$ and the initial value to be $X_0 = (1, 2)^T$. A reference solution for these values is shown in figure 5.2.

[40] Technically-speaking, what this means is that the steady non-equilibrium state of the system is unstable with respect to space-independent infinitesimal perturbations [Lef71, §2].

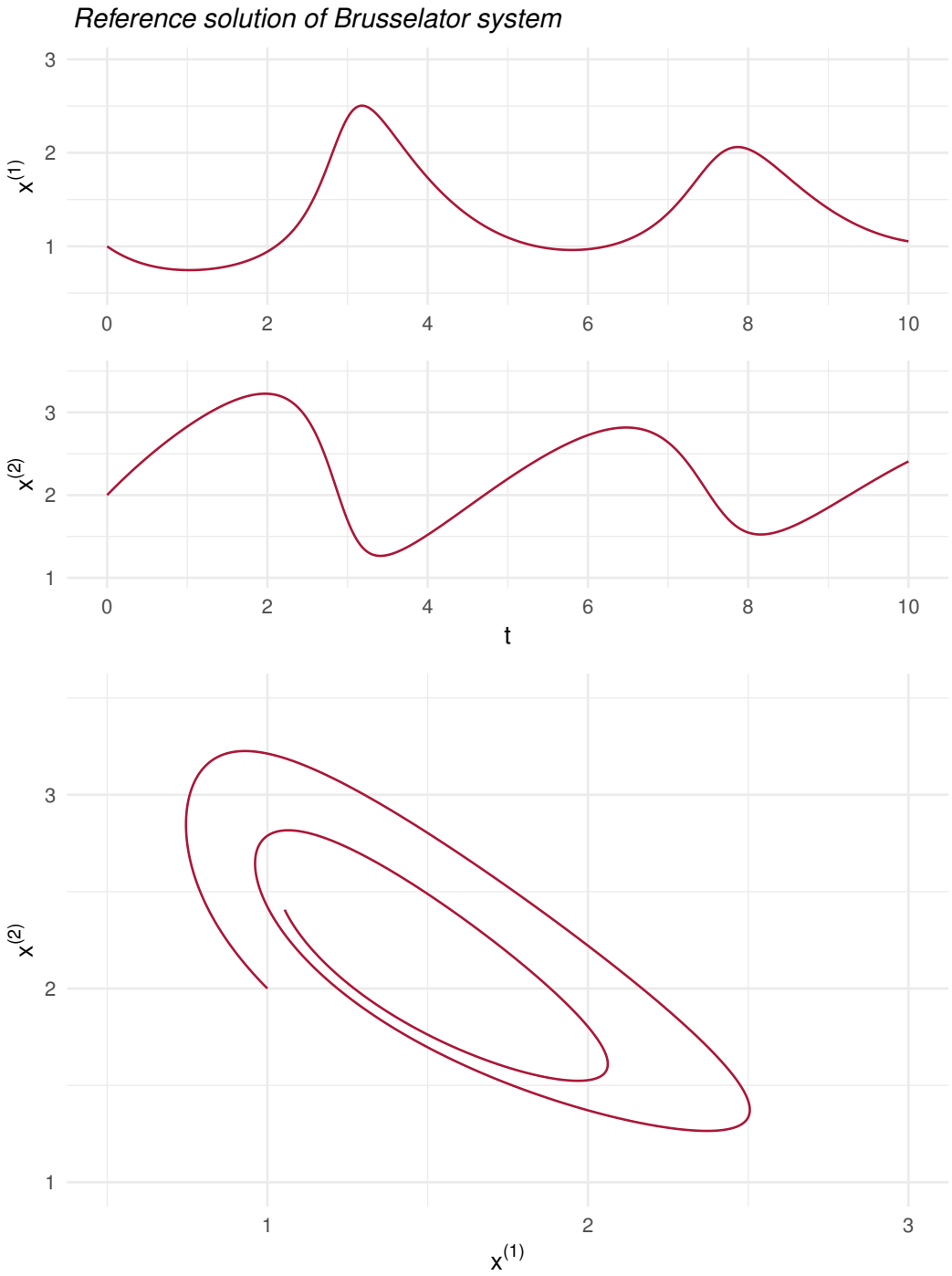


Figure 5.2: High-accuracy solution of the Brusselator initial value problem defined by (5.2) with parameters $\theta = (1.4, 3)$ and initial value $X_0 = (1, 2)^T$ for the range for $0 \leq t \leq 10$. The top two panes show the solution as a time series while the bottom pane shows the trajectory in phase space.

5.2 CALIBRATION

In this section we apply the calibration method suggested in Section 4.3 to find the scaling constants α which we will use in subsequent simulations. In Section 5.3, where we first consider the output of our probabilistic integrators, we further verify this approach to calibration by undertaking a simple goodness-of-fit check. In passing, we note that our results for the explicit first-order method—equivalent to the forward Euler method—can be compared to those given in the paper by Conrad et al. [Con16] and are similar, as expected.

Figure 5.3 plots on the vertical axis the logarithmic function values $\log B'(\alpha)$ corresponding to values of α shown on the horizontal axis, for $\log B'(\alpha)$ as in (4.34). The evaluation points are $\{1, 2, 5\} \times 10^m$ for $m = -4, -3, -2, -1, 0, 1, 2$. Since we employ an empirical Bayes approach, and are therefore only interested in the maximum probability value $\alpha^* = \operatorname*{argmax}_{\alpha \in \mathbb{R}^+} B'(\alpha)$, such a grid-search approach is adequate.[41]

The psuedo-density values are calculated by running 100 repetitions of each probabilistic integrator and forming sample statistics $\mathbb{E}_\xi(Z_i)$ and $\mathrm{Var}_\xi(Z_i)$ at each time t_i as described in Section 4.3.1. For the implicit methods *(bottom row)*, the pCN method introduced in Section 4.4.3 is employed to sample from the step-forward distribution at each step, with the parameter γ set to 0.95 and the fifth of five samples taken. As suggested by figure 4.2, the fifth sample can be expected to exhibit very low correlation with the starting position of the simulation.

Each calculation is repeated with a minimum of three different step-sizes h—usually $h = 0.01, 0.05, 0.1$, though in some of the higher-order methods we found that $h = 0.1$ caused instability so in these cases have instead plotted the results of simulations with $h = 0.005$. The thin lines connect points calculated using the same h and are intended to indicate the effect of varying α while keeping h constant.

The red data comes from simulations integrating the FitzHugh–Nagumo model while the green is from those integrating the Brusselator. Each model was run to final time $t_{\text{end}} = 10$. The top row of four panels gives the results for the probabilistic Adams–Bashforth method introduced in Chapter 3 for orders 1–4 (number of steps s = 1–4). The bottom row gives the same for the probabilistic Adams–Moulton method introduced in Chapter 4, also for orders 1–4 (in this case, number of steps s = 0–3).

The simulations all use the modified Bhattacharyya distance $\delta'(\cdot, \cdot)$ defined in (4.33) and modified penalising transformation $B'(\alpha)$ in (4.34). Using the unmodified $\delta(\cdot, \cdot)$ results in a breakdown of the calibration process for the higher-order integrators for

[41] For a full Bayesian analysis, particularly if these pseudo-densities were thought to be multi-modal, an MCMC simulation would need to be run to sample from $B'(\alpha)$.

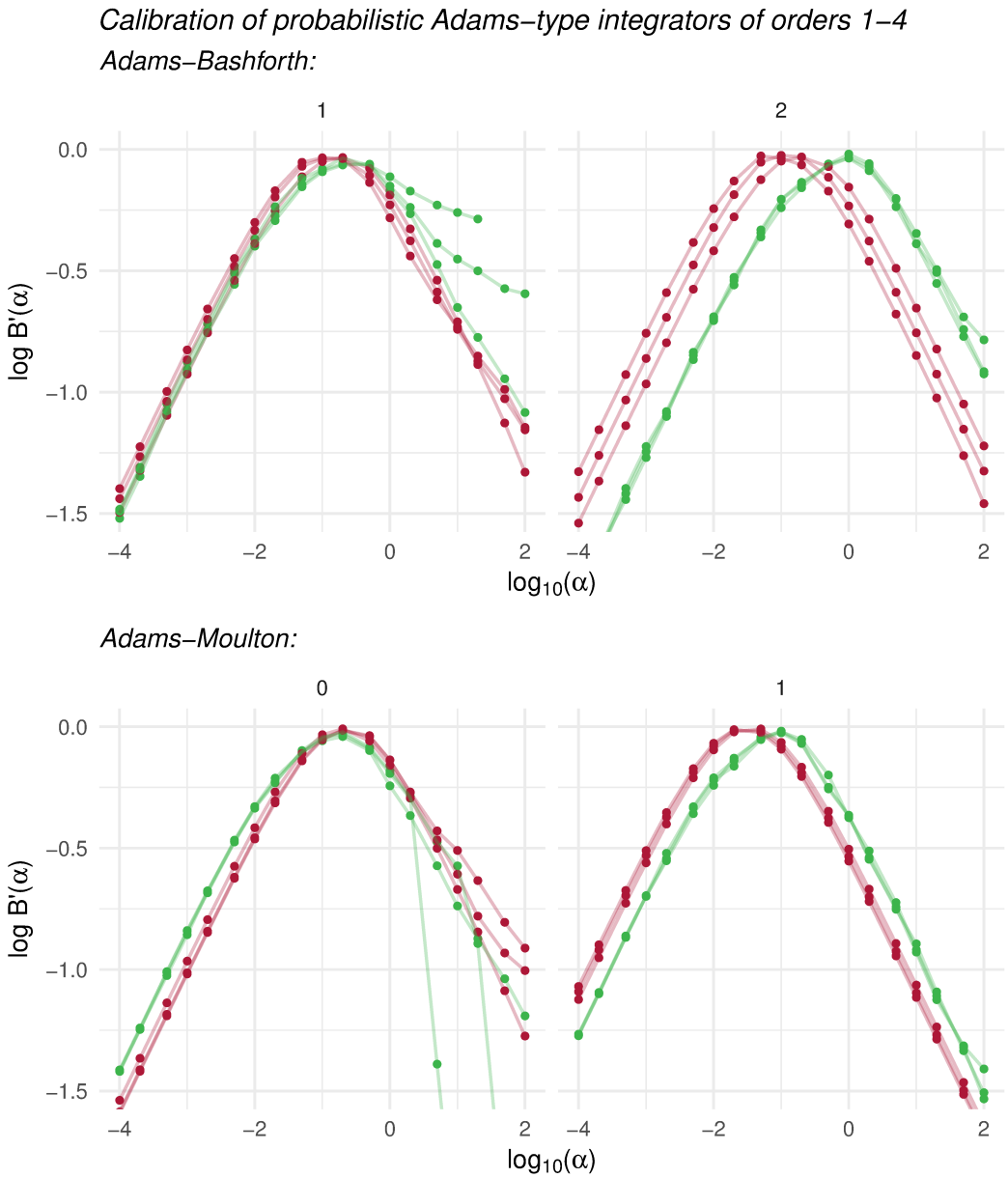
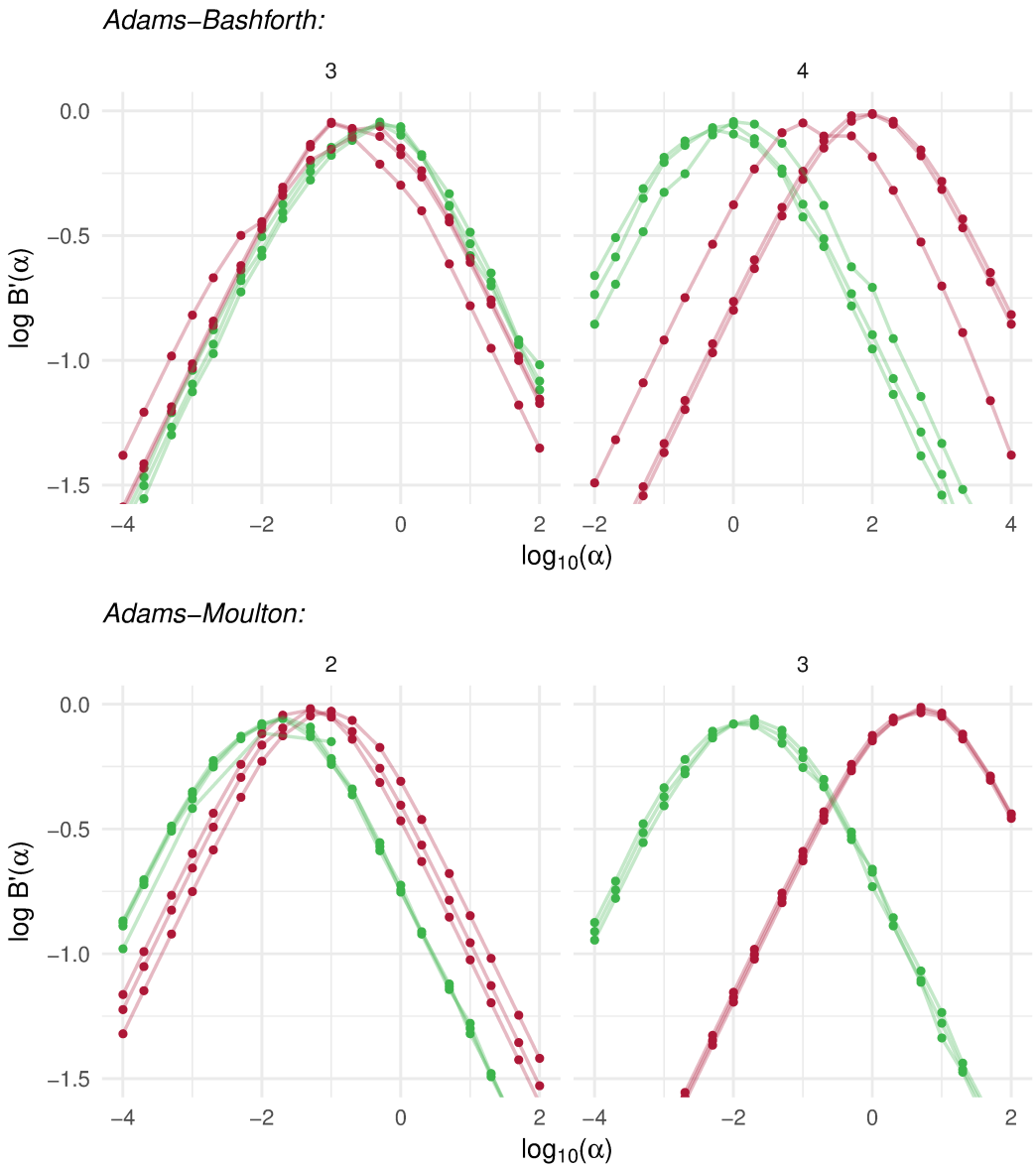


Figure 5.3: Calibration of probabilistic integrators from first to fourth order. Each pane plots the unnormalised log-density values $\log B'(\alpha)$—defined in equation (4.34)—against α. The evaluation points are $\{1, 2, 5\} \times 10^m$ for $m = -4, -3, -2, -1, 0, 1, 2$. Each datum is calculated by running 100 repetitions of each integrator and forming sample statistics $\mathbb{E}_\xi(Z_i)$ and $\mathrm{Var}_\xi(Z_i)$ at each time t_i as described in Section 4.3.1. The pCN method (Section 4.4.3) is used to sample from the step-forward distribution at each step, with the last of 5 samples taken. Each calculation is repeated with a minimum of three different step-sizes h; mostly $h = 0.01, 0.05, 0.1$, though in some of the higher-order methods we have *(cont.)*



included $h = 0.005$ instead of $h = 0.1$. The red data comes from simulations integrating the FitzHugh–Nagumo model while the green is from those integrating the Brusselator. Each model was run to final time $t_{\text{end}} = 10$. The top row of four panels gives the results for the probabilistic Adams–Bashforth method introduced in 3 for orders 1–4 (number of steps $s = 1,2,3,4$). The bottom row gives the same for the probabilistic Adams–Moulton method introduced in Chapter 4, also for orders 1–4 (in this case, number of steps $s = 0,1,2,3$). The simulations all use the modified Bhattacharyya distance $\delta'(\cdot,\cdot)$ defined in equation (4.33) and modified penalising transformation $B'(\alpha)$ in equation (4.34).

FitzHugh–Nagumo				Brusselator			
Method	α^*	Method	α^*	Method	α^*	Method	α^*
AB1	0.2	AM0	0.2	AB1	0.2	AM0	0.2
AB2	0.1	AM1	0.05	AB2	1	AM1	0.1
AB3	0.2	AM2	0.05	AB3	0.5	AM2	0.02
AB4	100	AM3	5	AB4	1	AM3	0.02

Table 5.1: Approximate values α^* maximising the function $B'(\alpha)$ defined in equation (4.34) for the FitzHugh–Nagumo and Brusselator systems. These values were read off visually from the plots in figure 5.3 and are the chosen from the grid of simulated values of α (*i.e.* with no attempt made to interpolate the grid).

the reasons described in Section 4.3.1—namely the situation where a consistent bias term dominates the scale of the variance, resulting in a failure to correctly capture the latter. Similarly we find the unmodified penaliser $B(\alpha)$ gives less consistent results as the calculation of sample statistics is skewed by the presence of near-zero terms occasionally arising at particular time-points—usually near turning points of the underlying ODE function $f(\cdot, \theta)$.

We can reason from figure 5.3 in two ways. Firstly, it provides overwhelming numerical evidence that the h-scaling suggested in Section 4.3 is the correct one. This conclusion can be justified by noting that the 'curves' for different values of h are closely superimposed in all cases—were the h-scaling incorrect, we would not expect this to be the case. In other words $B'(\alpha)$, defined in this way, is independent of h.

With this observation, we can justify the heuristic of choosing the integrators' h-dependence (corresponding to the h-scaling of ξ in the statement of Theorem 3, and of H in the statement of Theorem 4) to be *equal* to the bound permitted by the subsequent convergence analyses. For the same reason, we can state with confidence the structural deficiency of the integrator proposed in Proposition 3.4, in which the stepwise perturbations ξ are one order of h greater than this bound.

Secondly, we can read off the approximate maximising values α^* for each method and for each model. These are tabulated in table 5.1. Since we are roughly calibrating the integrators, we choose the maximum probability α amongst those in the set of points at which we evaluated $B'(\alpha)$ (which are approximately equally-spaced logarithmically) rather than attempting to interpolate these values. We will use these values in our subsequent simulations. It is immediately apparent that the calibration is highly problem dependent, particularly for the higher-order integrators. Unfortunately, this suggests that this lengthy calibration process is likely to be required each time a new model is under investigation.

It is impossible to infer from the results of calibrating these two models alone whether general rules exist for the distribution of α over a wide range of model types, depending solely on characteristics of the underlying method, such as convergence order or error constant. The emergence of any such higher-level pattern would be of supreme interest. This would be an intriguing direction for future research—this and other related ideas are briefly discussed in Section 6.2.2.

5.3 INTEGRATION OF THE FORWARD MODEL

Having calibrated the probabilistic integrators on the two chosen test problems, we now present some output exemplars of these algorithms, aiming to highlight some of their most appealing features, but also some possible shortcomings. In this section we mainly present qualitative results, aiming to give an overview of the possibilities opened up by these algorithms. We also suggest one type of goodness-of-fit check that could be performed to verify the integrator output. In Section 5.4 , where we consider parameter inference in the inverse problem, we will consider more quantitative aspects.

5.3.1 First-order methods

Figures 5.4 and 5.5 show plots of the FitzHugh–Nagumo system solved in the range $t \in [0, 10]$ using respectively the probabilistic first-order explicit (forward Euler) and first-order implicit (backward Euler) integrators. $K = 100$ repetitions were made with a step-size of $h = 0.1$. Each point represents one value $Z_i^{[k]}$ on the discrete path of one of the K instantiations of the integrator. The subscript i indexes time, whereas the bracketed superscript k indexes the Monte Carlo repetitions.

We consciously do not plot lines connecting these dots, since in the randomised integrator paradigm the individual trajectories $Z_{0:N}^{[k]}$ are *not* claimed to represent a sample from some underlying functional measure, in the manner of Chkrebtii et al. [Chk16] or Schober et al. [Sch18]. Instead, the empirical distribution of the ensemble of K values $Z_i^{[1:K]}$ is taken to represent the uncertainty in the value of the true solution $X_i \equiv x(t_i)$, corresponding to the same time t_i.

The three panes in each plot are intended to highlight the effect of calibration on these integrators. In both cases, the centre plot is correctly calibrated, meaning it shows the result of running the integrator with the scaling parameter α set equal to its maximum probability value α^* shown in table 5.1. For the sake of comparison, the top pane shows the same simulation but this time run with $\alpha = \alpha^*/10$, and in the bottom pane $\alpha = 10\alpha^*$. The dashed black line describes the trajectory $Z_{0:N}^{\text{classic}}$ of

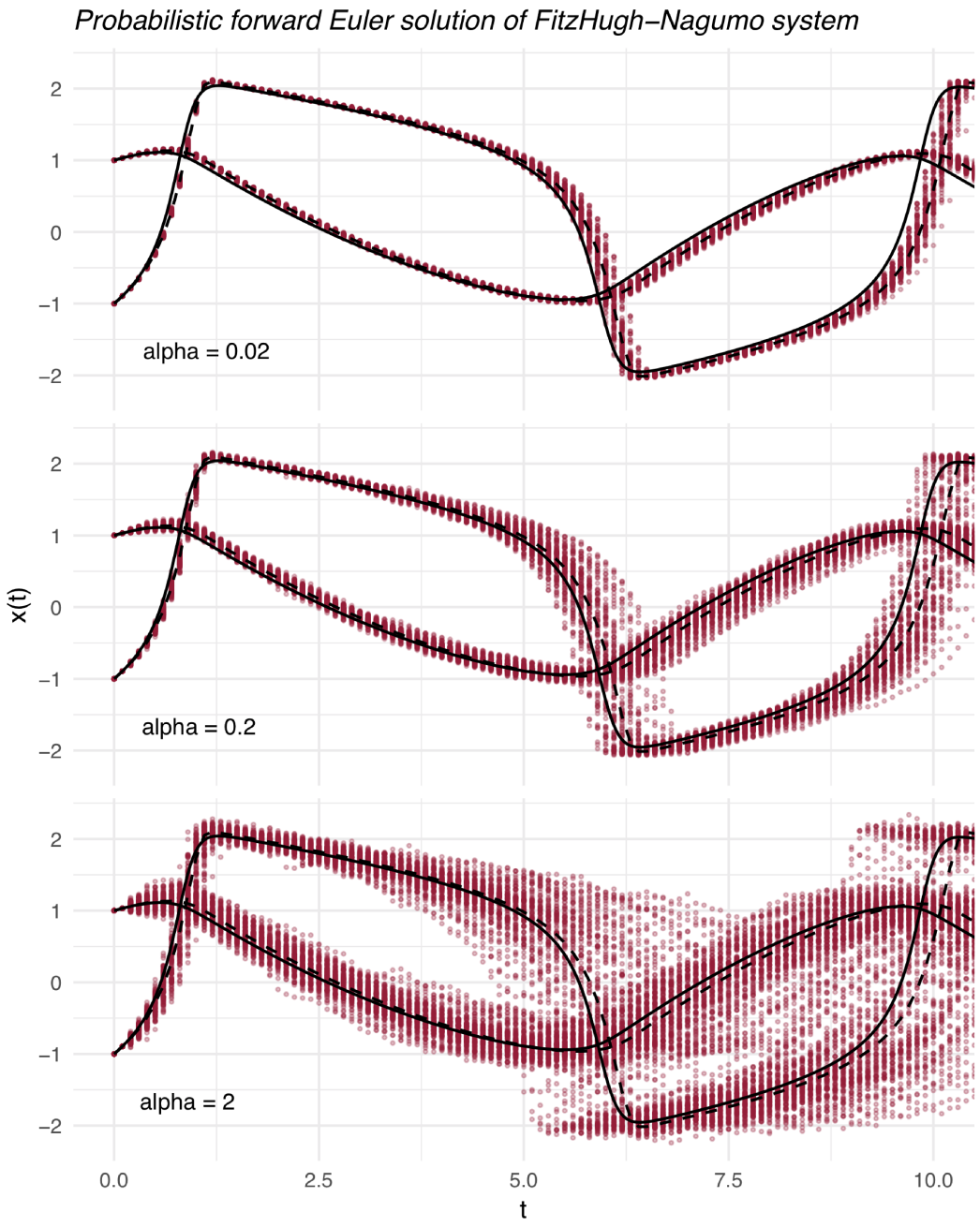


Figure 5.4: Ensemble path plot of the FitzHugh–Nagumo system solved in the range $t \in [0, 10]$ using the probabilistic forward Euler integrator. $K = 100$ repetitions were made with a step-size of $h = 0.1$. Each point represents one value $Z_i^{[k]}$ on the discrete path of one instantiation of the integrator. For this model, $\alpha^*_{\mathrm{AB1}} = 0.2$. The values of the calibration parameter α used in the simulation producing each plot are $\alpha = 0.02$ *(top)*, $\alpha = 0.2$ *(centre)*, and $\alpha = 2$ *(bottom)*. The dashed black line describes the classical trajectory $Z^{\mathrm{classic}}_{0:N}$ while the solid black line gives an reference solution calculated using a fine-mesh Runge–Kutta solver.

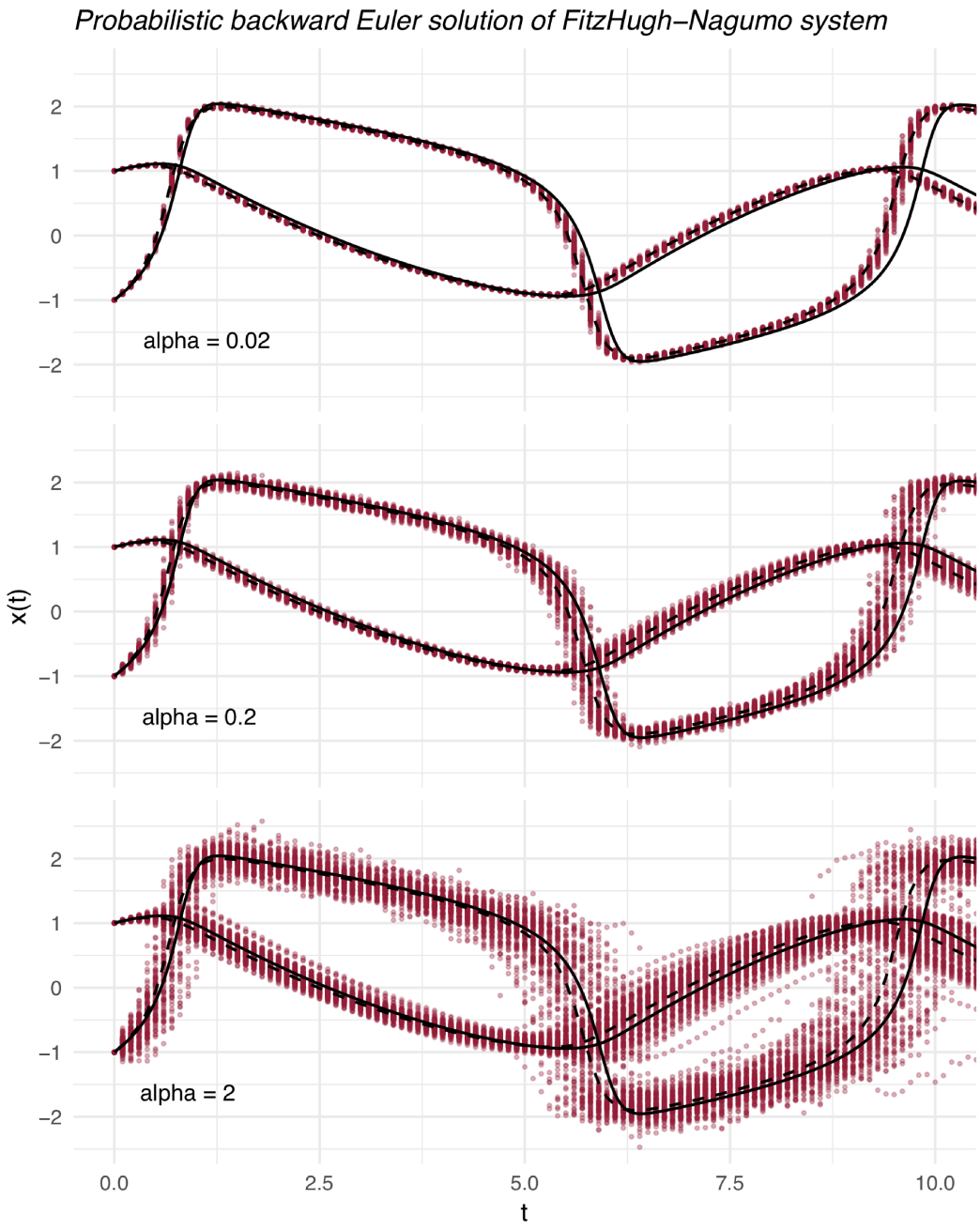


Figure 5.5: Ensemble path plot of the FitzHugh–Nagumo system solved in the range $t \in [0, 10]$ using the probabilistic backward Euler integrator. $K = 100$ repetitions were made with a step-size of $h = 0.1$. Each point represents one value $Z_i^{[k]}$ on the discrete path of one instantiation of the integrator. For this model, $\alpha^*_{\mathrm{AM0}} = 0.2$. The values of the calibration parameter α used in the simulation producing each plot are $\alpha = 0.02$ *(top)*, $\alpha = 0.2$ *(centre)*, and $\alpha = 2$ *(bottom)*. The dashed black line describes the classical trajectory $Z^{\mathrm{classic}}_{0:N}$ while the solid black line gives an reference solution calculated using a fine-mesh Runge–Kutta solver.

the classical method, while the solid black line gives an accurate reference solution calculated using a high-order Runge–Kutta solver calculated on a fine mesh.

In both cases, it is visually apparent that the centre plot captures the scale of uncertainty more appropriately than the top or bottom plots. The trajectory of the reference solution remains within the range of the ensemble of points throughout the integration interval, unlike in the top plot, and yet this range does not obviously overstate the uncertainty as is the case in both bottom plots. Obviously, this is merely a qualitative assessment of performance, but nevertheless forms a useful 'sanity check' on the output of the probabilistic algorithms.

For a somewhat more quantitative assessment, we consider the 'residuals' of the sample path—interpreted in the context of calibration as the (signed) difference between the predicted variance estimates $\widehat{E}_i^2$ and the empirical variance estimates $\mathrm{Var}_\xi(Z_i)$. These quantities were introduced and defined in equations (4.29) and (4.30). Since the dynamics of the ODE are complex and time-varying, yet the calibration constant α is set in advance and fixed, discrepancies are to be expected between the two estimates for variance when they are considered pointwise.

The signed difference between these two quantities can be expected to centre around zero if the calibration is successful, and will exhibit skew either above or below zero if the pointwise empirical variances are systematically of a different magnitude to the corresponding predictions from the classical error indicators.

Since evidently the variance of Z_i increases with i, these residuals form an intrinsically heteroscedastic sequence. It is therefore appropriate to scale each by an estimate for its standard deviation, to enable direct comparison. For this purpose we simply use the standard deviation $|\widehat{E}_i|$ given by the classical error indicator (4.28). We thus have, at each time t_i and for components $p \in \{1, 2\}$the residual quantity

$$r_i^{(p)} = \frac{\mathrm{Var}_\xi(Z_i^{(p)}) - (\widehat{E}_i^{(p)})^2}{|\widehat{E}_i^{(p)}|} \tag{5.3}$$

In figure 5.6, we plot histograms of $r_i = \{r_i^{(p)}\}_{p\in\{1,2\}}$ corresponding to each of the simulations displayed in figures 5.4 and 5.5. These show that, for $\alpha = 0.02$, the residuals r_i are consistently negative, indicating that this value of α results in output variance smaller systematically than expected, while the opposite is true of the case $\alpha = 2$. These both indicate poor model fit. The case $\alpha = 0.2$, corresponding to the calibrated value α^*, the residuals are broadly (though not completely) centred.

We note that these histograms represent very rough diagnostic test of model fit and serve as a 'sanity check' on the calibration procedure described in Section 4.3. We further remark that, in the middle panes of figure 5.6, it appears that the residuals for

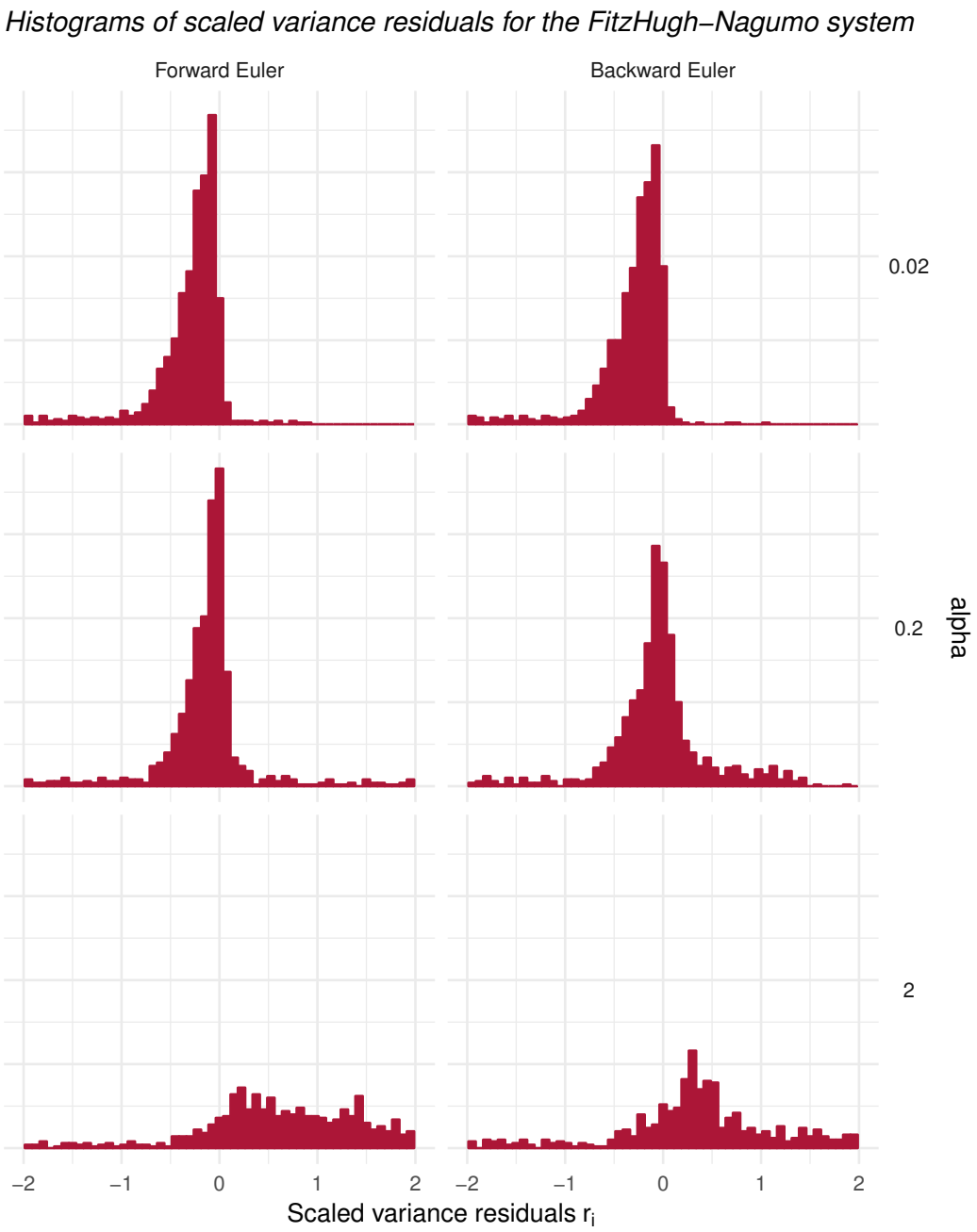


Figure 5.6: Histograms of residuals r_i, each representing the difference between classically-predicted and empirically-calculated values of the variance Z_i of the probabilistic integrator being employed. The histograms correspond to the three plots in figure 5.4 *(left-hand panes)* and the three plots in figure 5.5 *(right-hand panes)*. The residuals are scaled to remove the intrinsic heteroscedasticity present over the range of t —the expression for the scaled residuals is given in equation 5.3.

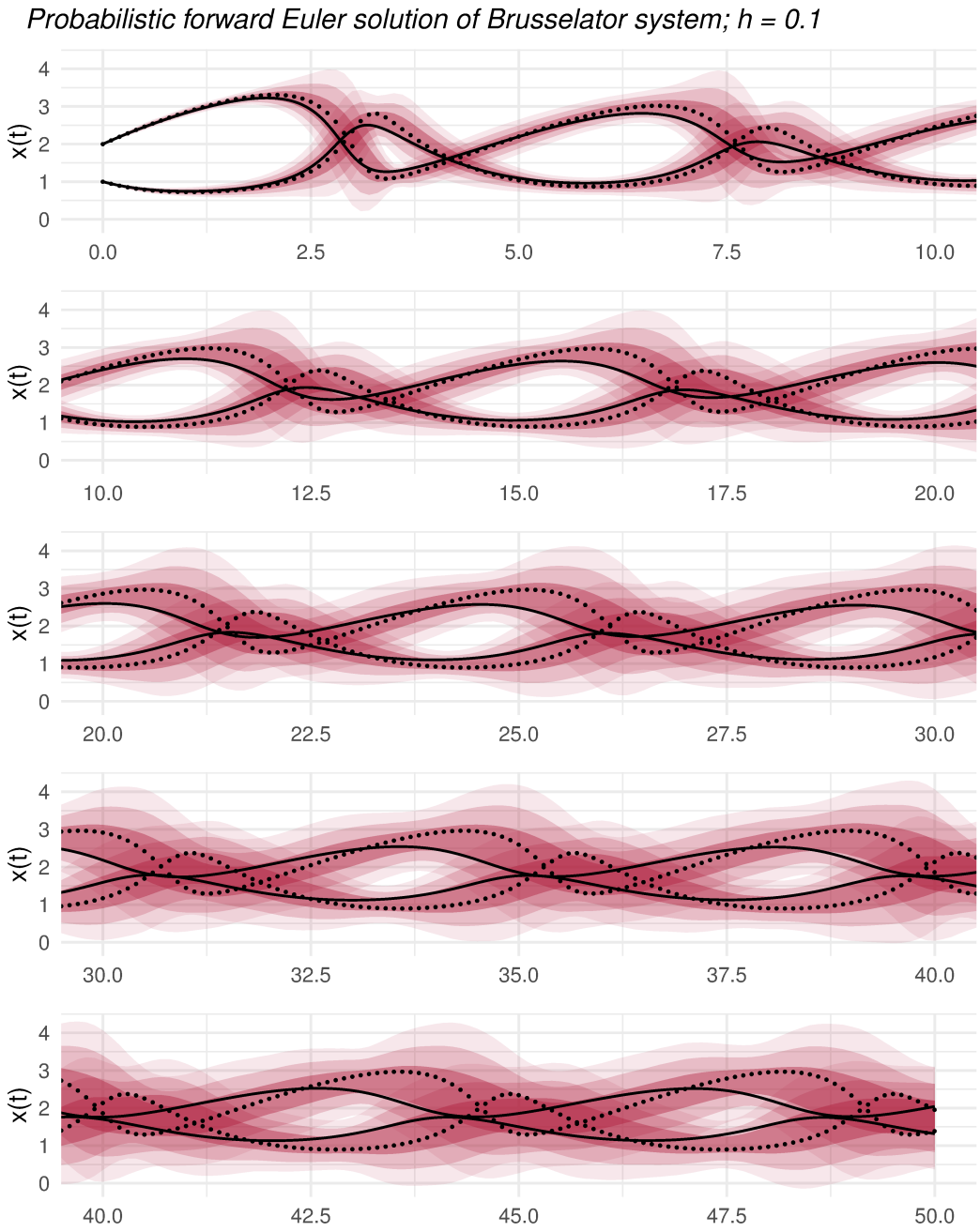


Figure 5.7: Summarised ensemble path plot of the Brusselator system solved in the range $t \in [0, 50]$ using a calibrated probabilistic first-order Adams–Bashforth integrator with step-size $h = 0.1$. The coloured bands represent 1σ, 2σ and 3σ intervals, calculated from 100 repetitions of the forward solve. The black dots describe the trajectory of the corresponding classical method, while the solid black line represents a reference solution calculated using a fine-mesh Runge–Kutta solver.

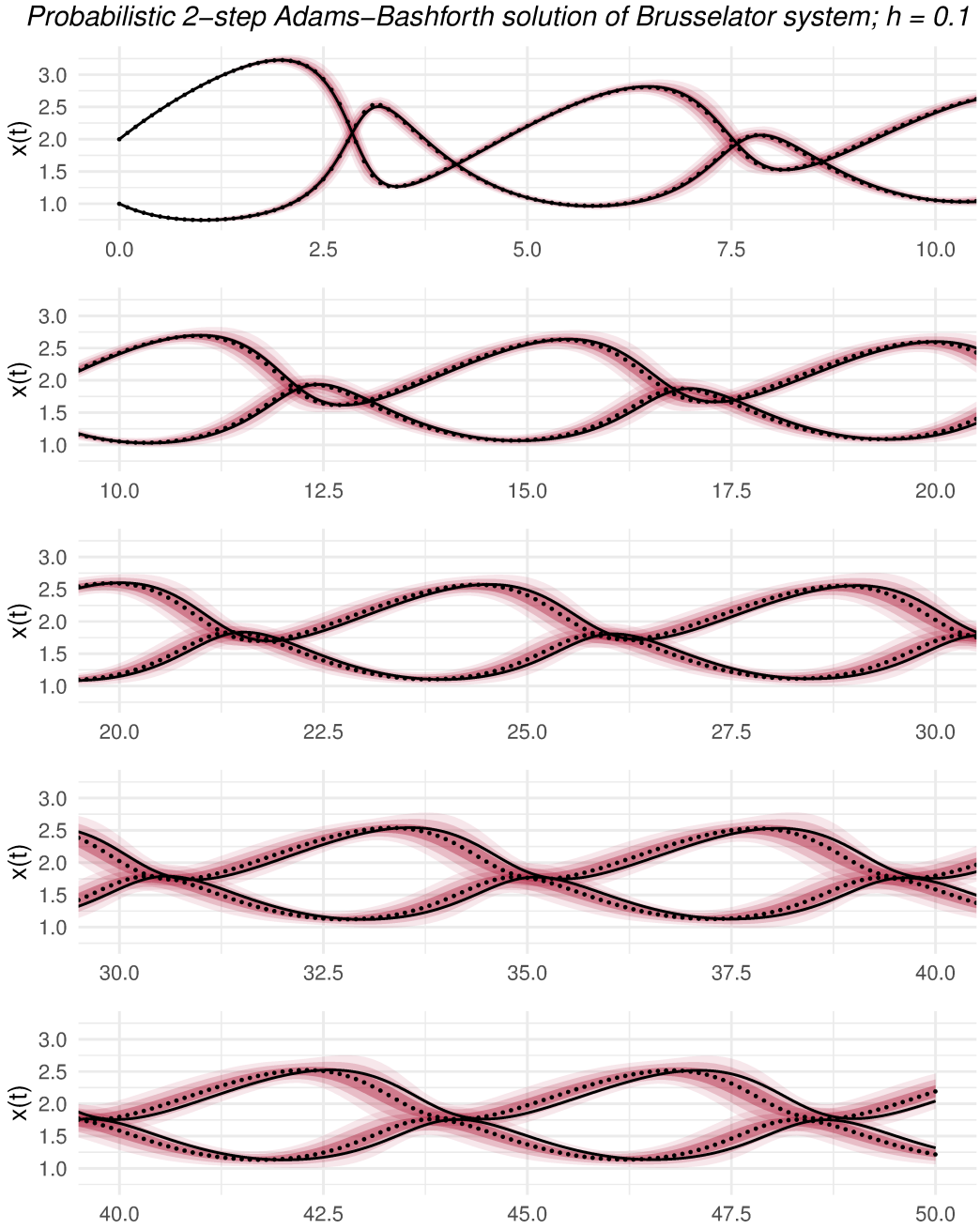


Figure 5.8: Summarised ensemble path plot of the Brusselator system solved in the range $t \in [0, 50]$ using a calibrated probabilistic second-order Adams–Bashforth integrator with step-size $h = 0.1$. The coloured bands represent 1σ, 2σ and 3σ intervals, calculated from 100 repetitions of the forward solve. The black dots describe the trajectory of the corresponding classical method, while the solid black line represents a reference solution calculated using a fine-mesh Runge–Kutta solver.

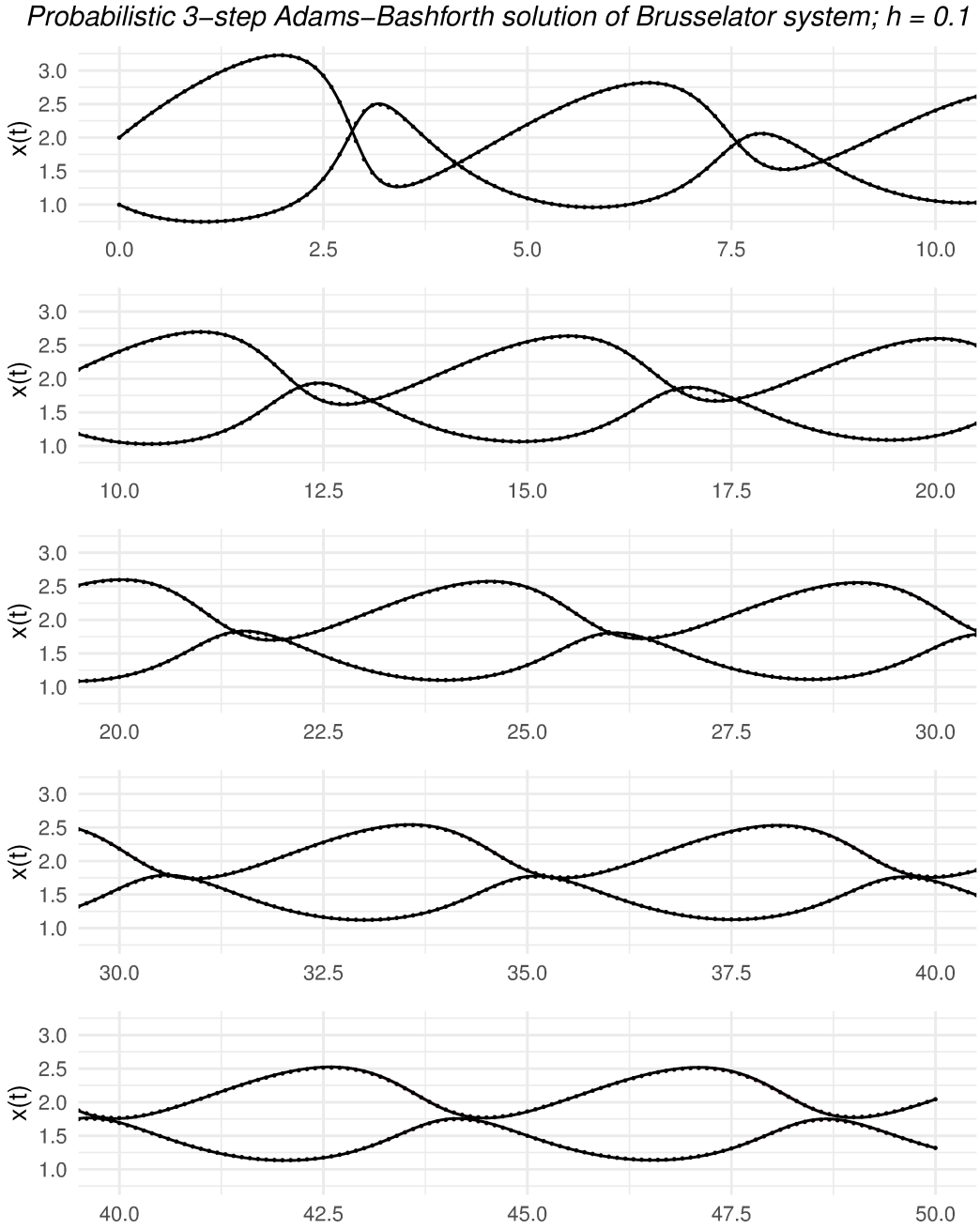


Figure 5.9: Summarised ensemble path plot of the Brusselator system solved in the range $t \in [0, 50]$ using a calibrated probabilistic third-order Adams–Bashforth integrator with step-size $h = 0.1$. The coloured bands represent 1σ, 2σ and 3σ intervals, calculated from 100 repetitions of the forward solve. The black dots describe the trajectory of the corresponding classical method, while the solid black line represents a reference solution calculated using a fine-mesh Runge–Kutta solver.

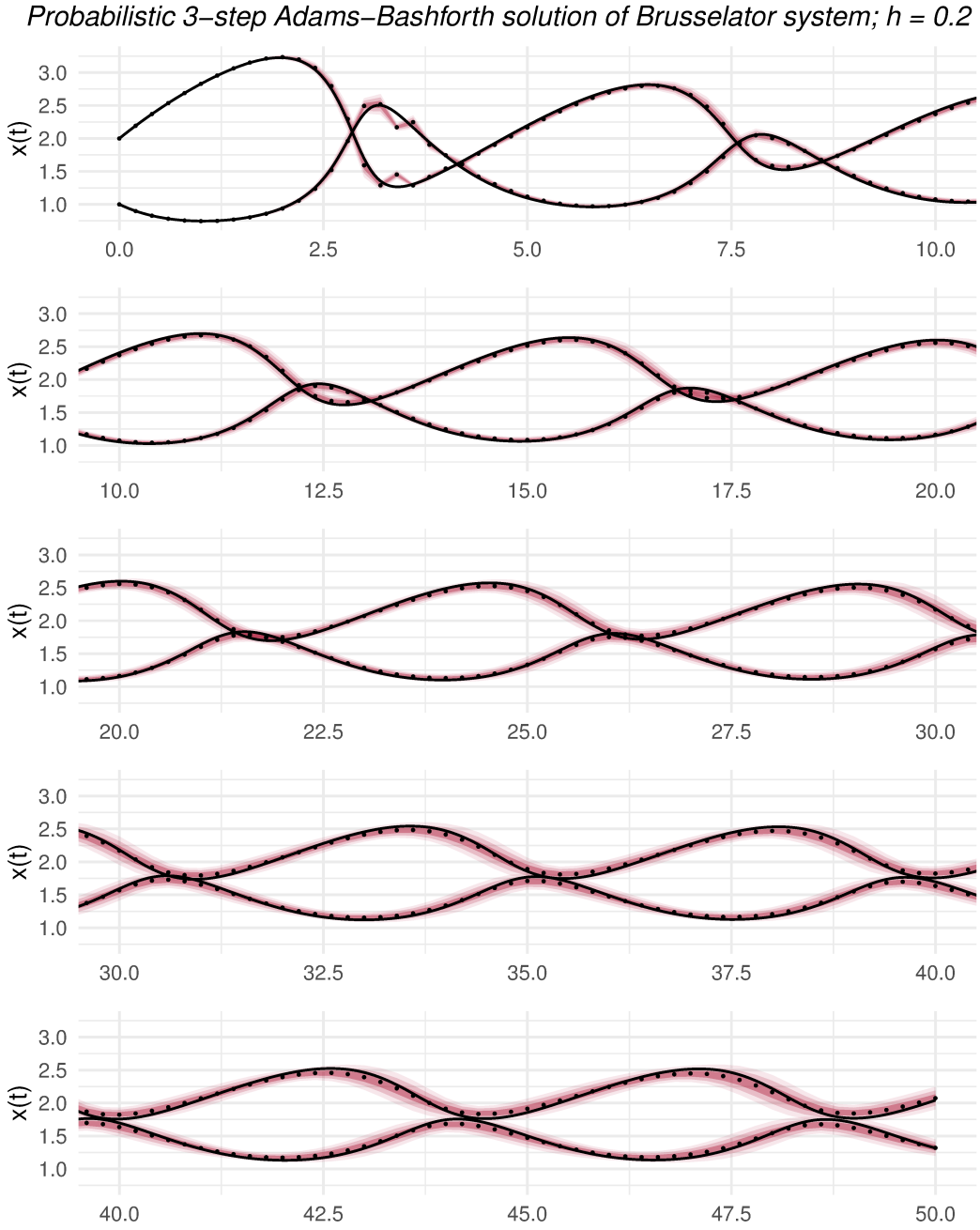


Figure 5.10: Summarised ensemble path plot of the Brusselator system solved in the range $t \in [0, 50]$ using a calibrated probabilistic third-order Adams–Bashforth integrator with step-size $h = 0.2$. The coloured bands represent 1σ, 2σ and 3σ intervals, calculated from 100 repetitions of the forward solve. The black dots describe the trajectory of the corresponding classical method, while the solid black line represents a reference solution calculated using a fine-mesh Runge–Kutta solver.

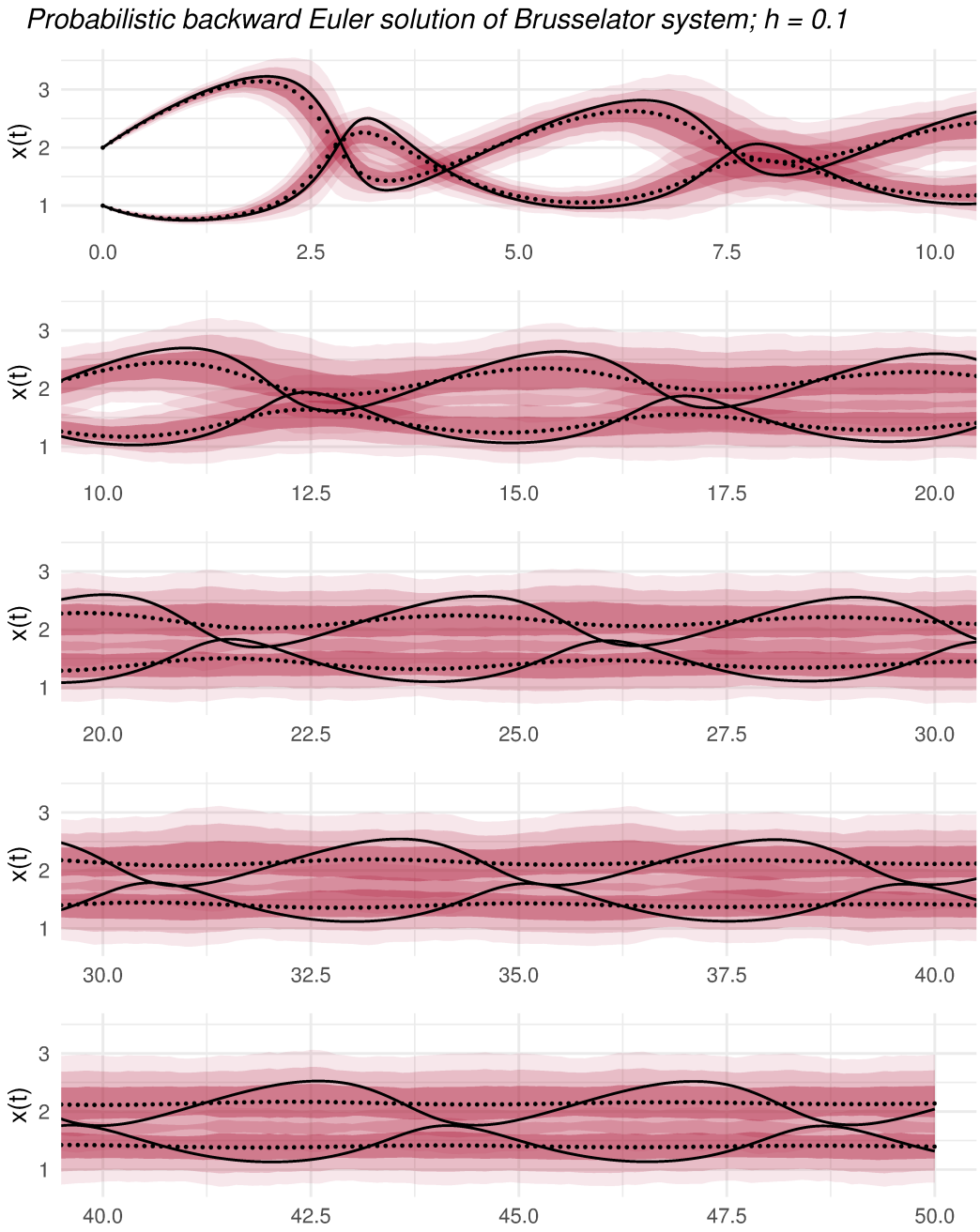


Figure 5.11: Summarised ensemble path plot of the Brusselator system solved in the range $t \in [0, 50]$ using a calibrated probabilistic first-order Adams–Moulton integrator with step-size $h = 0.1$. The coloured bands represent 1σ, 2σ and 3σ intervals, calculated from 100 repetitions of the forward solve. The black dots describe the trajectory of the corresponding classical method, while the solid black line represents a reference solution calculated using a fine-mesh Runge–Kutta solver.

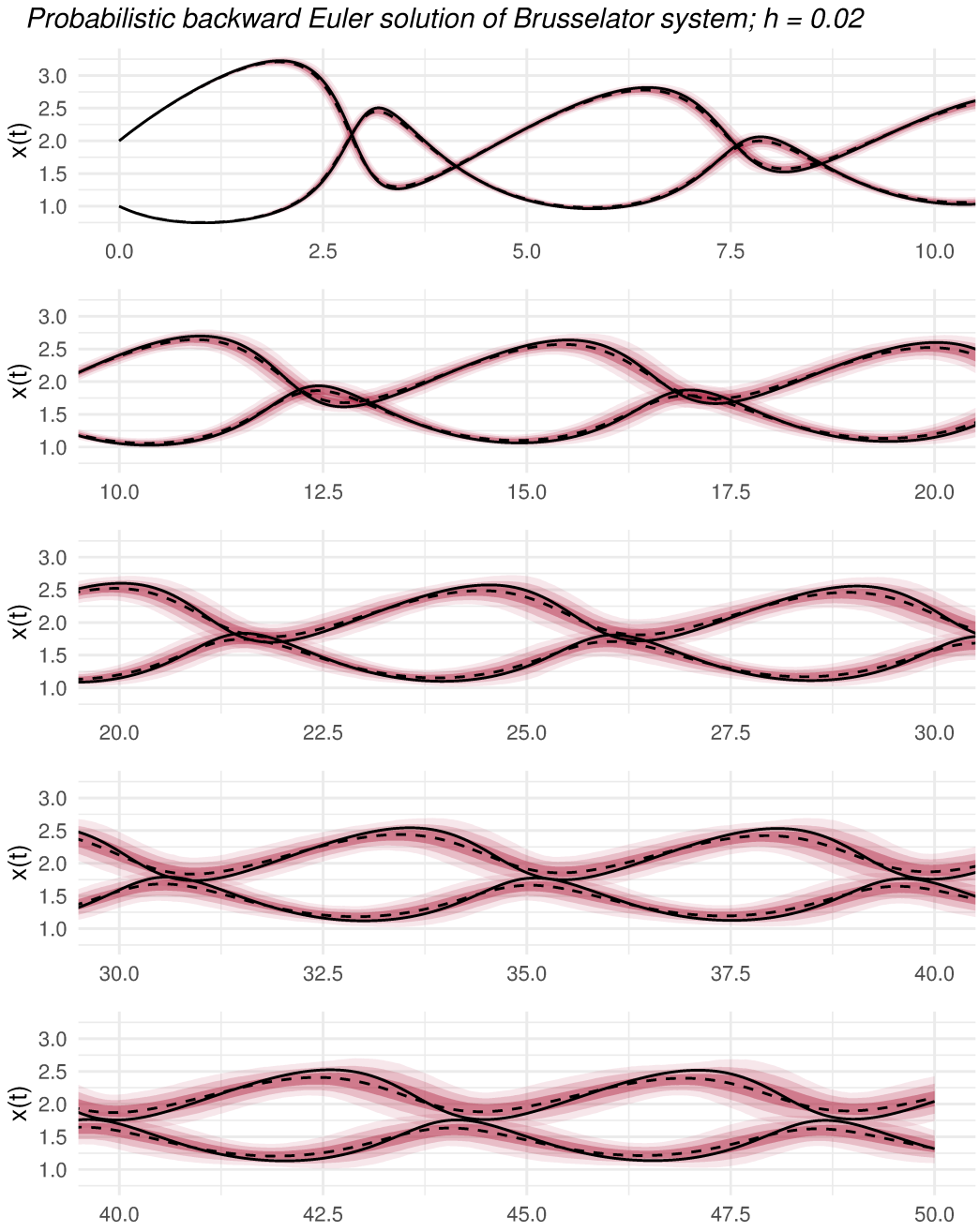


Figure 5.12: Summarised ensemble path plot of the Brusselator system solved in the range $t \in [0, 50]$ using a calibrated probabilistic first-order Adams–Moulton integrator with step-size $h = 0.02$. The coloured bands represent 1σ, 2σ and 3σ intervals, calculated from 100 repetitions of the forward solve. The dashed black line describes the trajectory of the corresponding classical method, while the solid black line represents a reference solution calculated using a fine-mesh Runge–Kutta solver. (We use a dashed line for the classical solution in this figure instead of plotting individual points, due to their extreme proximity.)

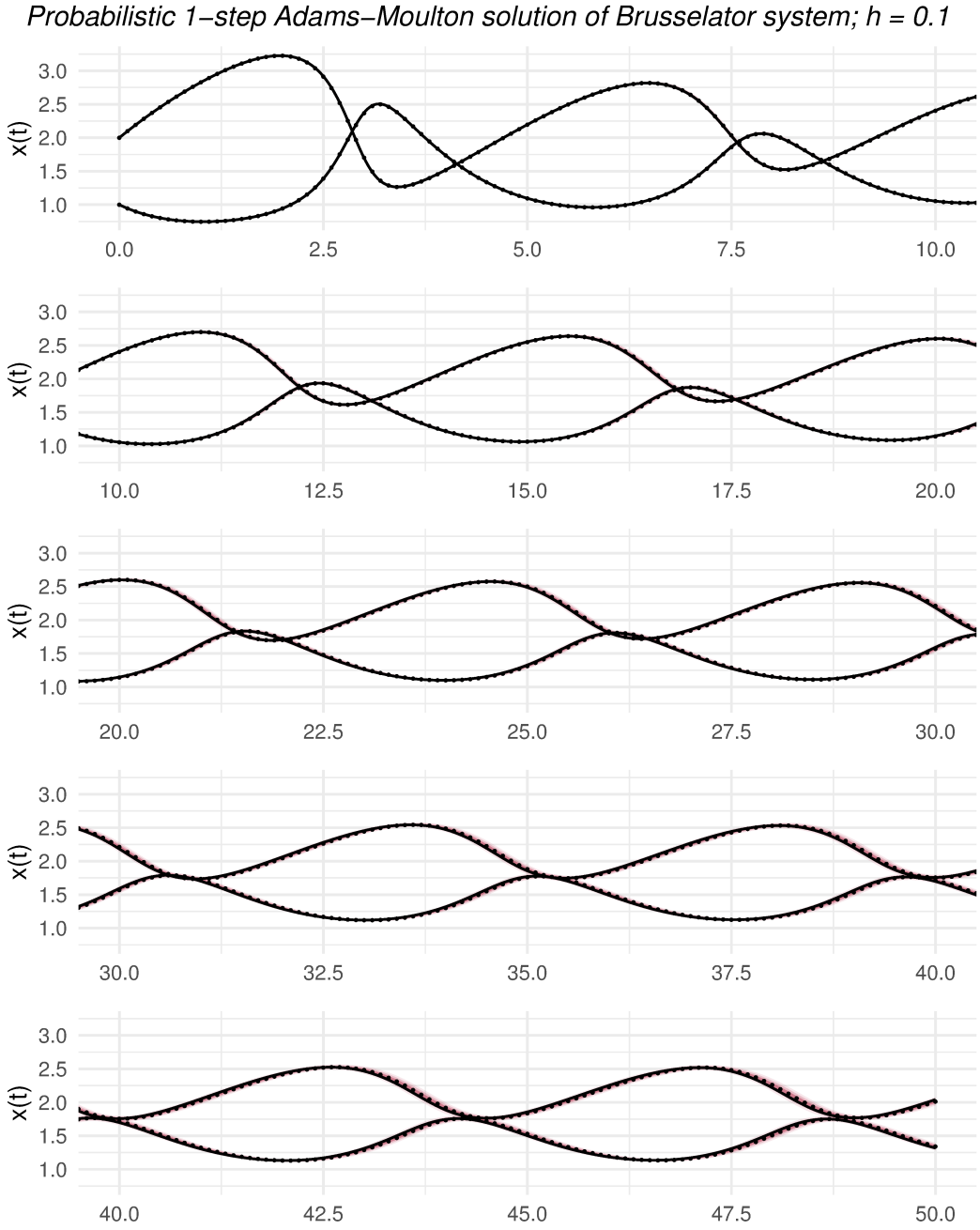


Figure 5.13: Summarised ensemble path plot of the Brusselator system solved in the range $t \in [0, 50]$ using a calibrated probabilistic second-order Adams–Moulton integrator with step-size $h = 0.1$. The coloured bands represent 1σ, 2σ and 3σ intervals, calculated from 100 repetitions of the forward solve. The black dots describe the trajectory of the corresponding classical method, while the solid black line represents a reference solution calculated using a fine-mesh Runge–Kutta solver.

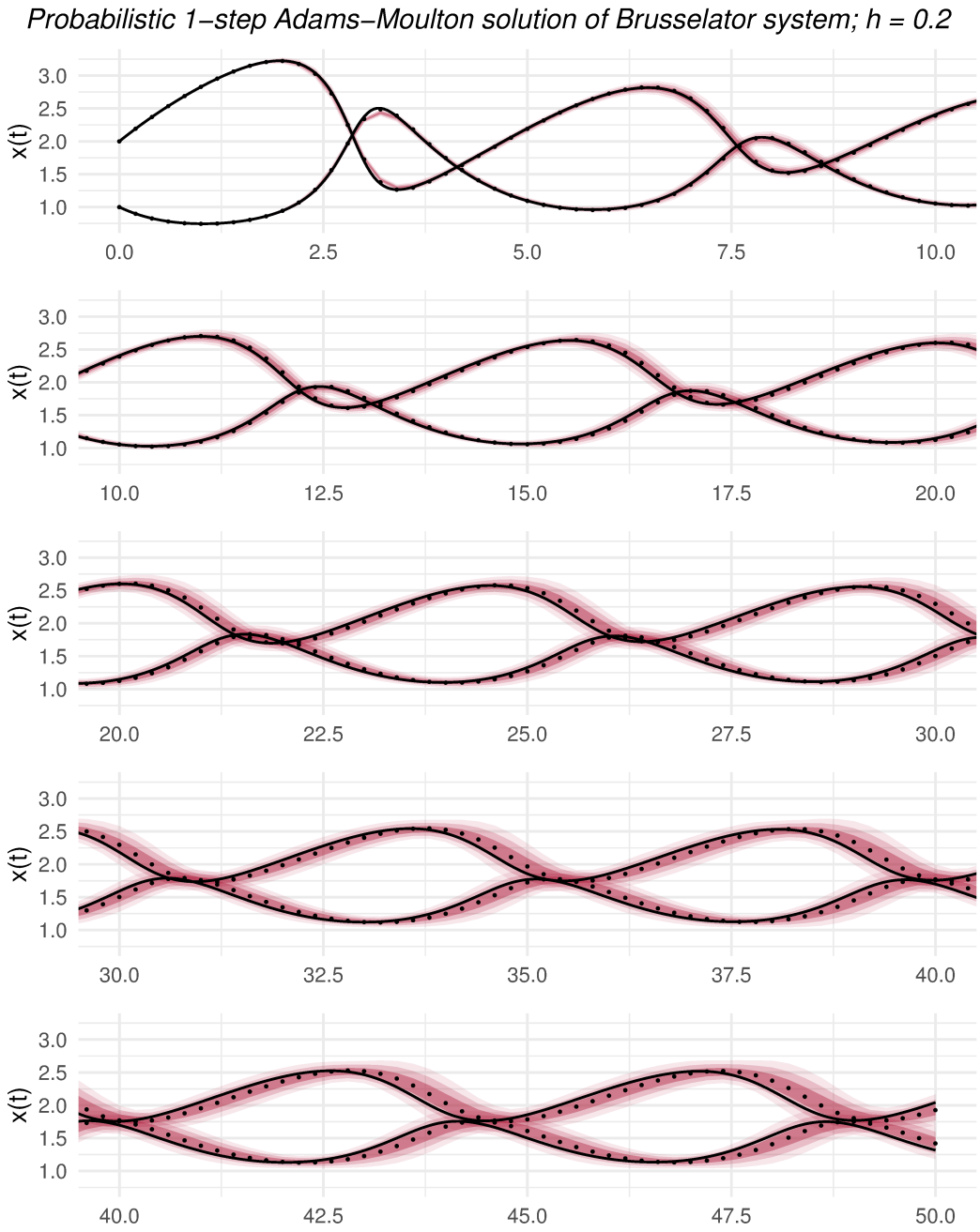


Figure 5.14: Summarised ensemble path plot of the Brusselator system solved in the range $t \in [0, 50]$ using a calibrated probabilistic second-order Adams–Moulton integrator with step-size $h = 0.2$. The coloured bands represent 1σ, 2σ and 3σ intervals, calculated from 100 repetitions of the forward solve. The black dots describe the trajectory of the corresponding classical method, while the solid black line represents a reference solution calculated using a fine-mesh Runge–Kutta solver.

the backward Euler integrator are more symmetrically distributed around zero than for the forward Euler integrator. This could be encouraging evidence of improved model fit arising from the fundamental difference in structure of the two integrators, though caution is required in drawing such a conclusion on the basis of a single simulation.

5.3.2 *Higher-order methods*

We now consider the result of implementing calibrated second- and higher-order probabilistic multistep methods. For these simulations we use the second of our test examples, the Brusselator, and we extend the time interval to $t \in [0, 50]$ to highlight the longer-time performance of the algorithms. We plot the results of several simulations in figures 5.7–5.14.

In these plots, rather than plotting each trajectory as a sequence of dots, we have summarised the integrator output by calculating the sample mean and sample standard deviations of the Monte Carlo ensemble $Z_i^{[1]}, \ldots, Z_i^{[K]}$ for each time index i, and plotting the 1σ, 2σ and 3σ intervals as shaded bands. In each case, the black dots give the trajectory $Z_{0:N}^{\text{classic}}$ of the classical classical integrator, while the thick solid black line gives a reference solution calculated using a fourth-order Runge–Kutta solver with step-size $h = 0.005$.

Figure 5.7 highlights how at the chosen step-size, the classical forward Euler method accumulates significant error well before the end of the interval of interest. From the point at which this occurs forward, there is very little relationship between the paths of the black dots and the solid black lines, demonstrating the manifest inadequacy of this method when applied to this problem.

The probabilistic method instead outputs a distribution from which credible intervals can be calculated—it can be easily verified that the true solution is contained within the 2σ band throughout the interval of integration. In this way, the distribution reported for $x(t)$ at, say, $t = 50$ would place non-negligible measure on an interval containing the true solution. Once again, this is not a quantitative conclusion, but nonetheless this feature is the minimum we should expect of probabilistic integrators.

The same principle is repeated in figures 5.8 and 5.9 for 2-step and 3-step probabilistic Adams–Bashforth methods respectively. In the latter case, the high accuracy of the third-order method means that even over this long interval, the uncertainty in the solution is almost invisible. As a result, we plot in figure 5.10 the same simulation but run at double the step-size, $h = 0.2$. This once again shows the satisfactory performance of the probabilistic method, though it is interesting to note that the artefacts visible around $t = 3$ —a by-product of the quadratic interpolants involved in

third-order Adams-type methods—are not captured by the probabilistic method. Our instinct is that a fundamentally different approach would be required if, in designing a probabilistic algorithm, it was felt essential that these local error phenomena also be accounted for.

Figure 5.11 plots the output of the probabilistic backward Euler method applied to the same problem. It is immediately apparent that, while the true solution is once again contained within the 2σ band as in figure 5.7, the interpretation this time must be different. In this case, the path of the dots representing the solution using the classical backward Euler method is so obviously wrong that, in effect, any estimator based on it would be completely uninformative of the true solution. After approximately $t = 20$, the trajectory remains in what appears to be a stable, static equilibrium, though the true dynamics remain oscillatory. The probabilistic solver correctly captures this—the constant width credible bands in the range $t \in [40, 50]$ accurately reflect the completely uninformative nature of the integrator whose error is being modelled. The conclusion that this output suggests—that the method in question is not informative for the true solution at all—is *itself* ultimately informative for the practitioner.

In order to demonstrate that the probabilistic backward Euler method *can* work in a similar manner to the explicit probabilistic methods of figures 5.7–5.10 when the step-size h is chosen commensurate with the accuracy and stability of the underlying method, we provide a plot in figure 5.12 of this integrator being run with a much smaller step-size of $h = 0.02$. This plot shows that at this reduced step-size, the probabilistic solver does captures the correct dynamics—with meaningful credible bands nevertheless representing the scale of uncertainty in the method's output.

To complete our visual survey of the new methods considered in this thesis, in figure 5.13 we give the output of the second-order probabilistic Adams–Moulton integrator for $h = 0.1$ and, since as in the case of the third-order Adams–Bashforth method the output is hard to discern by eye, we repeat the calculation for $h = 0.2$ in figure 5.14.

5.4 INFERENCE IN THE INVERSE PROBLEM

Having discussed the output of probabilistic algorithms on the solution of a fully-specified forward model, we now consider the effect of their use in the case that the values of the model parameters θ are unknown. We seek to infer these unknown parameters in a Bayesian inverse problem framework. Once again, we first verify and then extend the approach from Conrad et al. [Con16].

Our example in this section is the FitzHugh–Nagumo model from Section 5.1.1, considered in the interval $t \in [0, 10]$. We first generate a synthetic dataset Y of 10 two-dimensional data-points collected at times $t_Y = \{1, 2, \ldots, 10\}$. After calculating

a high-accuracy Runge–Kutta solution, the value of this reference solution at each time ordinate t_{Y_j} is independently corrupted by centred Gaussian noise with variance $\sigma = 2.5 \times 10^{-3} \cdot \mathbb{I}_2$.[42]

We then treat the parameter θ as unknown and assign to each component $\theta^{(\nu)}$ a log-normal prior distribution such that $p(\log \theta^{(\nu)}) \sim \mathcal{N}(\theta^{(\nu)}_{\text{true}}, 1)$, independently for each $\nu \in \{1, 2, 3\}$. This follows the approach in Conrad et al. [Con16]. (An alternative reasoning for how to set the parameter prior for this system is suggested by Campbell [Cam07, §2.3] and stems from an understanding of the range of parameters outside of which the qualitative behaviour of the system changes—it stops exhibiting periodicity—and then spreading the prior probability mass mainly on this range.) In preliminary experiments, we found that prior choice has very limited impact on posterior output and thus parameter inference—indeed even using the improper prior $p(\theta) \propto 1$ throughout was not found to be problematic. These findings are likely due to the fact that the synthetic data are sufficiently informative about the system that the prior term in the model is dominated by the likelihood.

We assume a Gaussian likelihood with variance σ the same as that used to generate the synthetic data, and we also assume that the initial state X_0 is known. We then run an MCMC procedure to infer the posterior distribution of θ. The way in which the MCMC algorithm is designed critically affects its sampling performance—indeed, different choices can even result in a subtly different problem being solved. The next section considers this issue.

5.4.1 *MCMC for randomised integrators*

Recall from Section 1.4.1 that the object of our interest is the posterior distribution $p(\theta|Y)$. Application of Bayes' Theorem gives the proportionality relation

$$p(\theta|Y) \propto p(\theta)L(Y;\theta) \tag{5.4}$$

In the framework we have been considering, the likelihood term $L(Y;\theta)$ depends on the numerical solution Z of an ODE, and the core theme of this thesis has concerned the use of randomised methods in this context, and the treatment of Z as a random variable rather than as a deterministic object.

Randomised methods make a draw ω from a sample space Ω, use this instantiation to calculate the random numerical solution Z, and then execute the random likelihood

[42] For simplicity, we choose to generate the data Y at time ordinates which will coincide with discrete knots of our later simulations. Of course, if this were not the case—as may well be the case in a real-data setting—some interpolation procedure would have to be additionally performed in order to calculate likelihood values we need.

evaluation relative to this approximate forward solution of the system. Specifically, for methods based on stepwise Gaussian perturbations ξ_i—including the probabilistic Adams–Bashforth integrator from Section 3.3 and the Gaussian approximation to the probabilistic Adams–Moulton integrator from Section 4.4.2—the set of *realised* perturbations (a draw from the random variable ξ) is determined by the random seed ω and possibly θ, and Z of course depends on these.[43]

The likelihood function—expressed in a way that explicitly notes these dependencies—should properly be written as $L(Y;\theta,Z(\xi))$, and a complete Bayesian analysis then requires that the corresponding posterior be rewritten as $p(\theta,Z(\xi)|Y)$. Of course, this newly-introduced randomness needs to be marginalised—this results in a modification to the right-hand side of equation (5.4) given by

$$p(\theta)\int_{\Xi} L\big(Y;\theta,Z(\xi|\theta)\big)\,\mathrm{d}\mathbb{P}_{\xi} \tag{5.5}$$

Note that, as in (2.11), we have chosen to write the probability measure over which the expectation is taken as $\mathbb{P}_{\xi|\theta}$ rather than $p(\xi|\theta)$, and the random numerical solution as $Z(\xi|\theta)$. This emphasises firstly that, in general, ξ is allowed to depend on the parameter θ. In the case of the explicit integrator of Chapter 3, ξ consists of a discrete sequence of Gaussian perturbations with fixed scale and as such has a distribution independent of θ. However, in the case of the implicit integrator of Chapter 4, it is clear that the perturbations ξ *do* depend on θ. Secondly, the reason for avoiding the notation $p(\xi|\theta)$ is to highlight the fact that, in the latter case, we do not have access to a pointwise-evaluable density for ξ given θ—instead, we can draw samples from its distribution. Since our overall aim is to draw a set of samples from $p(\theta|Y)$, this is sufficient for our purposes.

It is thus clear that the expression in (5.5) can clearly only be approximated numerically, even pointwise. However, the additional stochasticity now present means that there are several possible approaches to implementing a Monte Carlo algorithm for θ, which we now survey.

Marginalising over the randomised likelihood

This strategy marginalises the introduced stochastic randomness in order to give a random approximation $L^K(Y;\theta)$ to the expected likelihood $\mathbb{E}_{\xi|\theta}L(Y;\theta,Z(\xi))$. This is then simply inserted into the MCMC algorithm for θ as if it were the true (inaccessible)

[43] The unapproximated implicit methods of Section 4.4.3 , which require an internal Monte Carlo simulation to implement, are more complicated to treat. It may theoretically be possible to treat ω as additionally including the random seed driving the inner Monte Carlo calculations at each step, though this would introduce significant additional complexity. Later in this section we consider some further consequences of this issue.

likelihood $L(Y;\theta)$. This is the approach suggested and implemented by Conrad et al. [Con16]. The approximation $L^K(Y;\theta)$ is made by another, inner, Monte Carlo procedure, given by

$$L^K(Y;\theta) = \frac{1}{K}\sum_{k=1}^{K} L\big(Y;\theta,Z(\xi^{[k]})\big) \tag{5.6}$$

At each iteration m of the outer MCMC over θ, this algorithm samples multiple $\xi^{[k]}$ independently from $\mathbb{P}_{\xi|\theta}$,[44] evaluates the random likelihood for each, then forms a Monte Carlo sum to give an approximation to the expected likelihood $\mathbb{E}_{\xi|\theta}L(Y;\theta,Z(\xi))$. This gives samples from (5.5) of the form

$$p(\theta^{[m]})\left[\frac{1}{K}\sum_{k=1}^{K} L\big(Y;\theta^{[m]},Z(\xi^{[k,m]})\big)\right] \tag{5.7}$$

The incorporation of the likelihood approximation into an MCMC algorithm is justified by the principle of psuedo-marginal MCMC [And09]. Strictly speaking, this requires an *unbiased* estimator of the likelihood in order for the outer MCMC algorithm to still correctly target the exact posterior distribution. However, we cannot claim here that the inner Monte Carlo sum is unbiased. If, as in Conrad et al. [Con16, §3.2], we accept that we cannot eliminate all the bias and instead are primarily concerned with undue optimism in the predicted variance, this approach may suffice. Nevertheless, it is clear that this introduces a further type of approximation error to the problem.[45]

In our experiments, we found that for the MCMC over θ to be effective, K—the number of iterations in the inner Monte Carlo sum—is sometimes required to be very large. This seems to be because the estimator for the expected likelihood $L^K(Y;\theta)$ has high enough variance, even for moderate K, that proposed moves θ^* are rarely accepted in the outer simulation targeting $p(\theta|Y)$. In effect, an algorithm of this form operates over the joint space $\Theta \times \Xi$ but, since the $\xi^{[k]}$ are sampled independently, candidate moves of the outer MCMC over θ are invariably distant from the current point in the *joint* space. This hinders mixing.

As a very rough conclusion, we find $K < N/10$ (where $N = \lfloor h^{-1}t_{\text{end}} \rfloor$ is the number of steps in a run of the forward model) makes it impossible for a Metropolis–Hastings algorithm targeting $p(\theta|Y)$ to mix properly. Thus, in our example, solving

[44] In reality, this simply means re-running the randomised integrator multiple times with the set of stepwise perturbations $\xi^{[k]}(\omega)$ generated by different random seeds $\omega^{[k]}$ each time.

[45] If the desire is to actually *quantify* the effect of such biased approximations on posterior inference in this setting, several theoretical results focusing on this type of construction are given in Lie et al. [Lie18]. These results focus on giving bounds on the Hellinger metric between the exact posterior and one calculated using randomised solutions of the forward model.

the FitzHugh–Nagumo for $t \in [0, 10]$ with a relatively coarse step-size of $h = 0.1$ requires a minimum of 10 repetitions of the likelihood evaluation at each step. In fact, for the algorithm to function 'well' rather than simply adequately, we find we require $K \gg N/10$. This introduces a significant additional computational burden.

Fixing ξ for the entire simulation

An alternative approach, intended to avoid the instability caused by the variability of the inner Monte Carlo sum in (5.7), is to pre-generate a finite set of K instantiations $\xi^\dagger \equiv \{\xi^{[1]}, \ldots, \xi^{[K]}\}$ and reuse these at every step of the outer MCMC. In effect, this eliminates ξ as a random variable altogether, and the algorithm targets instead a modified deterministic posterior $p_{\xi^\dagger}(\theta|Y)$. In this scenario, posterior inference over θ will depend on the specific set $\xi^\dagger$, and the effect of this can be expected to be hard to quantify. Furthermore, while the outer MCMC over θ may be induced to mix better in this set-up, the computational expense of running K parallel forward solves at each step is still present.

It is important to note that this idea assumes that a given sample $\omega^{[k]} \in \Omega$ generates a realised value $\xi(\omega^{[k]})$ of ξ which can be meaningfully reused by the algorithm in subsequent steps. If ξ is a set of N independent Gaussian perturbations then this is straightforward, since those same perturbations can simply be added stepwise during a second simulation using a new parameter value. However, as pointed out in footnote 43, for an implicit integrator requiring an inner Monte Carlo simulation at each step—a random procedure itself—it is not obvious how to do this. As a result, this approach is only realistic for implicit integrators under the Gaussian approximation regime of Section 4.4.2.

As an aside, this formulation opens up the possibility of *choosing* $\xi^\dagger$ intentionally to minimise in some way the variance of L^K relative to the value of K. This thought is somewhat reminiscent of quasi-Monte Carlo methods [Caf98], though the complicated relationship between ξ and the quasi-randomised likelihood $L(Y; \theta, Z(\xi))$ means that this is likely to be a highly non-trivial endeavour.

Simultaneous inference over θ and $Z(\xi)$

Setting $K = 1$ in (5.7) results in samples from (5.5) of the form

$$p(\theta^{[m]})L(Y; \theta^{[m]}, Z(\xi^{[m]})) \tag{5.8}$$

This formulation—in terms of a single Monte Carlo loop—circumvents the need to perform multiple likelihood calculations for each candidate parameter θ^*, but unless care is taken in the design of the algorithm, it will suffer from the problem of mixing even more than before, since clearly $1 \leq N/10$ for most realistic simulations.

Considered as in (5.8), the MCMC targets the joint posterior $p(\theta, Z(\xi)|Y)$ directly, though some strategy has to have been assumed for independently sampling each $\xi^{[m]}$ from the measure $\mathbb{P}_{\xi|\theta}$. How then should this joint posterior be sampled from? Even if a marginal density $p(\xi|\theta)$ *is* accessible—as in the case of integrators using stepwise Gaussian perturbations—it makes little sense to try to construct a Markov chain to sample from it. This is because the dimension of Ξ is the number of time-steps N, and when this is large it will be difficult to construct a transition kernel which will output proposals ξ^* likely to be accepted by a Markov chain-type algorithm, due to the well-known 'curse of dimensionality'.

One could instead construct a Markov chain on Θ alone and propose ξ^* independently from $p(\xi|\theta)$ at each iteration. However, the effect on the value of $L(Y;\theta,Z(\xi))$ of an independently-sampled novel ξ^* will be even more pronounced without the likelihood-averaging present in the $K > 1$ setting described above—this construction is simply that one but with $K = 1$. It is therefore clear that proposing (θ^*, ξ^*) simultaneously at each step is unworkable, since a Metropolis–Hastings algorithm has little hope of generating a useful set of samples in this regime.

Our suggested solution is to use a hybrid scheme, in which a candidate parameter θ^* is proposed and accepted or rejected having had its likelihood calculated using the same instantiation of perturbations $\xi^{[m]}$ as in the current iteration k. If accepted as $\theta^{[m+1]}$, a new $\xi^{[m+1]}$ can then be sampled and the likelihood value recalculated ready for the next proposal. The proposal at step $m+1$ is then compared to this new value. Pseudocode for this algorithm is given in Algorithm 2.

This approach requires that $L(Y;\theta,Z(\xi))$ be recalculated exactly once for each time a new parameter value θ^* is accepted. The cost of this strategy is therefore bounded by twice the cost of an MCMC algorithm operating with a classical integrator—the bound being achieved only in the scenario that all proposed moves θ^* are accepted. This is not an unreasonable uplift in total computation, especially when compared to strategies which require many more forward solves per sample.[46]

For the remainder of our simulations we adopt this last-described approach, and find it works well in general. We verify the sampling performance of our algorithm in the next section by supplying some specimen diagnostic plots from the simulations alongside our main statistical output.

[46] Strictly-speaking, this algorithm does not target the exact density $p(\theta, Z(\xi)|Y)$ in the case of the implicit integrator, for which ξ depends on θ. This is because the use of a novel θ^* at each iteration does not keep ξ exactly constant, thus the additive perturbations cannot easily be reused. In order to ensure absolute exactness, the additional computational expense of drawing θ^* independently, followed by $\xi^*|\theta^*$ independently, is therefore unavoidable. For our simulations, we accept this minor deviation from exactness in order to produce a workable algorithm.

MCMC ALGORITHM FOR SAMPLING $p(\theta, Z(\xi)|Y)$

1 INPUT $\theta^{[1]}$
2 $\xi^{[1]} \sim \mathbb{P}_\xi$
3 FOR $1 \leq m \leq M$
4 $\quad \phi^{[m,m]} \leftarrow p(\theta^{[m]})L(Y;\theta^{[m]}, Z(\xi^{[m]}))$
5 $\quad \theta^* \sim q_m(\cdot|\theta^{[m]})$
6 $\quad \phi^{[*,m]} \leftarrow p(\theta^*)L(Y;\theta^*, Z(\xi^{[m]}))$
7 $\quad \alpha^{[m]} \leftarrow \min(1, \phi^{[*,m]}/\phi^{[m,m]})$
8 $\quad r^{[m]} \sim \mathcal{U}[0,1]$
9 $\quad$ IF $r^{[m]} < a^{[m]}$
10 $\quad\quad \theta^{[m+1]} \leftarrow \theta^*$
11 $\quad\quad \xi^{[m+1]} \sim \mathbb{P}_\xi$
13 $\quad$ ELSE
14 $\quad\quad \theta^{[m+1]} \leftarrow \theta^{[m]}$
16 $\quad$ END
15 $\quad m \leftarrow m + 1$
17 END
18 OUTPUT $\theta^{[2]}, \ldots, \theta^{[M]}$

Algorithm 2: Algorithm for drawing M samples from the joint posterior $p(\theta, Z(\xi)|Y)$, exact when $\mathbb{P}_{\xi|\theta} = \mathbb{P}_\xi$. At iteration m, $\xi^{[m]}$ is drawn independently from the measure $\mathbb{P}_\xi$—for methods based on stepwise Gaussian perturbations, this draw is then used in the calculation of $Z(\xi^{[m]})$. By contrast, θ^* is drawn from a symmetric Markov chain transition kernel $q_m(\cdot|\theta^{[m]})$, whose form depends on the value of m as described by equation (5.9). In assessing whether to accept θ^*, the algorithm uses the same sample $\xi^{[m]}$ to calculate the value of ϕ for both current and proposed points. A straightforward modification to the expression for $\alpha^{[m]}$ in line 7 would permit generalisation to non-symmetric $q_m(\cdot|\theta^{[m]})$. For further details see main text.

5.4.2 *Parameter inference for the FitzHugh–Nagumo model*

In this section we present the results of implementing an MCMC algorithm of the type just considered, for the purposes of inferring the parameters of the synthetic-data inverse problem detailed in Section 5.4.

To draw samples from $p(\theta, Z(\xi)|Y)$, we construct an MCMC algorithm using the Adaptive Metropolis–Hastings algorithm of Haario et al. [Haa01], whereby the transition kernel $q_m(\theta^*|\theta^{[m]})$ at step m is a centred Gaussian $\mathcal{N}(0, \Sigma_m)$ with variance equal to[47]

$$\Sigma_m = \begin{cases} 10^{-1}h \cdot \mathbb{I}_D, & \text{for } 0 \leq m \leq M' \\ (2.38)^2 D^{-1} \cdot \left(\mathrm{Cov}\left(\theta^{[0]}, \ldots, \theta^{[m-1]}\right) + 10^{-5} \cdot \mathbb{I}_D\right) & \text{for } M' < m \leq M \end{cases} \tag{5.9}$$

The index M' represents an initialisation horizon before which the algorithm runs as classical (unadaptive) Metropolis–Hastings, and after which adaptation begins. The variable D is simply the dimension of the parameter θ, in this case 3. The scale $(2.38)^2 D^{-1}$ arises from a recommendation on optimal scaling of Metropolis–Hastings algorithms in Roberts et al. [Rob97], while the addition of a small multiple of the identity matrix is recommended by Haario et al. themselves to reduce the likelihood of instability in the algorithm resulting from a near-singular covariance matrix [Haa01]. Note that it is possible to calculate the covariance term recursively—this avoids the expense of having to recalculate it from scratch at each iteration, which will increase with m if performed non-recursively.

We collect 1000 samples by initialising the MCMC at the true parameter values and running $M = 11000$ iterations, setting $m' = 500$, discarding the first $2m' = 1000$ samples as burn-in, then thinning by a factor of 10. We perform this for four different step-sizes $h = \{0.005, 0.01, 0.02, 0.05\}$, first using the classical solver, then a calibrated probabilistic solver, to evaluate the likelihood of the forward model.

Specimen performance diagnostics are given in figure 5.19, which displays the path of the Markov Chain and its autocorrelation sequence for the simulation run using the probabilistic forward Euler method *(top two panes)* and probabilistic backward Euler method *(bottom two panes)*, both with step-size $h = 0.01$. From these, it is apparent that the chains mix well using either algorithm and that successive samples are not closely correlated. Simulation diagnostics for the other values of h are similar.

For the forward Euler and backward Euler methods, the resulting posterior distributions for $(\theta^{(3)} \equiv c, \theta^{(2)} \equiv b)$ are shown in figures 5.15 and 5.16 respectively. In each case, the top pane plots the posteriors obtained when using the classical integrator. For the coarser simulations, the bias these produce is evident—a practitioner reporting a maximum a posteriori or empirical mean estimator and associated credible intervals for $\theta^{(3)} \equiv c$ would unknowingly submit high confidence in an incorrect conclusion.

The bottom panes, which display the same simulations but now with the forward model solved using calibrated probabilistic integrators, demonstrate how these posteriors have broadened to curb this unjustified overconfidence. The bias is largely unaffected, but the variance has increased so that the true solution is now located in a region of non-negligible posterior probability mass. The result for the forward Euler method is similar to that in Conrad et al. [Con16]; the result for the backward Euler

[47] Note that this transition kernel is symmetric, *i.e.* $q_m(\theta'|\theta'') = q_m(\theta''|\theta')$ for any $\theta', \theta'' \in \Theta$ and for all $0 \leq m \leq M$. This feature is assumed in Algorithm 2—the given form of the acceptance probability $\alpha^{[m]}$ requires it—though generalisation to non-symmetric kernels involves only minor modifications.

method is based on the new construction in Chapter 4 and demonstrates similar qualitative behaviour.

These conclusions are also reported in tabular form in tables 5.2 and 5.3. Here, the empirical mean estimator and the empirical standard deviation of each set of 1000 samples is given. We then report the error in the empirical mean estimator (compared to the known true parameter value) and the number of standard deviations that this is equivalent to. Particularly for the parameter c, the effect described in the previous paragraph is clearly visible. For example, for all runs except those with smallest h, the posterior mean estimator calculated from the MCMC simulations using the classical integrators is over two standard deviations from the true value. Loosely speaking, this range represents an approximately 95% credible region (if considered in one dimension).

Similar results are obtainable for the second-order methods, and we present equivalent plots and tables in figures 5.17–5.18 and tables 5.4–5.5. The same essential phenomenon described for the first-order methods is partly visible, though it is also true that these results are less clear-cut in several respects. For the 2-step Adams–Bashforth method (figure 5.17), only the simulation with step-size $h = 0.1$ shows marked bias in the posterior estimate resulting from use of the classical solver. The probabilistic solver redresses this by returning a much broader posterior, as desired, though in this case it is not clear if the degree to which the posterior has widened correctly reflects the aforementioned bias. It is also the case that a broadening effect is observed with smaller step-sizes—most visible for $h = 0.05$—in cases where the classical integrator shows very little bias.

The results for the second-order implicit method are also equivocal. Here, very little difference is apparent between the posteriors resulting from simulations of different mesh-sizes. This output suggests that the spread of the observed posterior distribution ultimately results from the noise in the data Y and that, in the sense we are interested in, the algorithms are all solving the 'exact' problem. In other words, the parameter uncertainty visible here is likely to derive from the underlying variability of the data in the problem, rather than from any shortcoming in the chosen numerical method.

The obvious response to this finding would be to consider situations in which the scale of noise in the data is smaller or larger than here. We undertook several such experiments and found close to singular posteriors in the former case—probably due to the data becoming so much more highly informative with small noise; and experienced problems with stability in the latter case—likely due to the sensitive dependence of the dynamics on the parameter θ, where the noisier data provides the opportunity for sufficient variation that the integrity of the MCMC simulation is compromised.

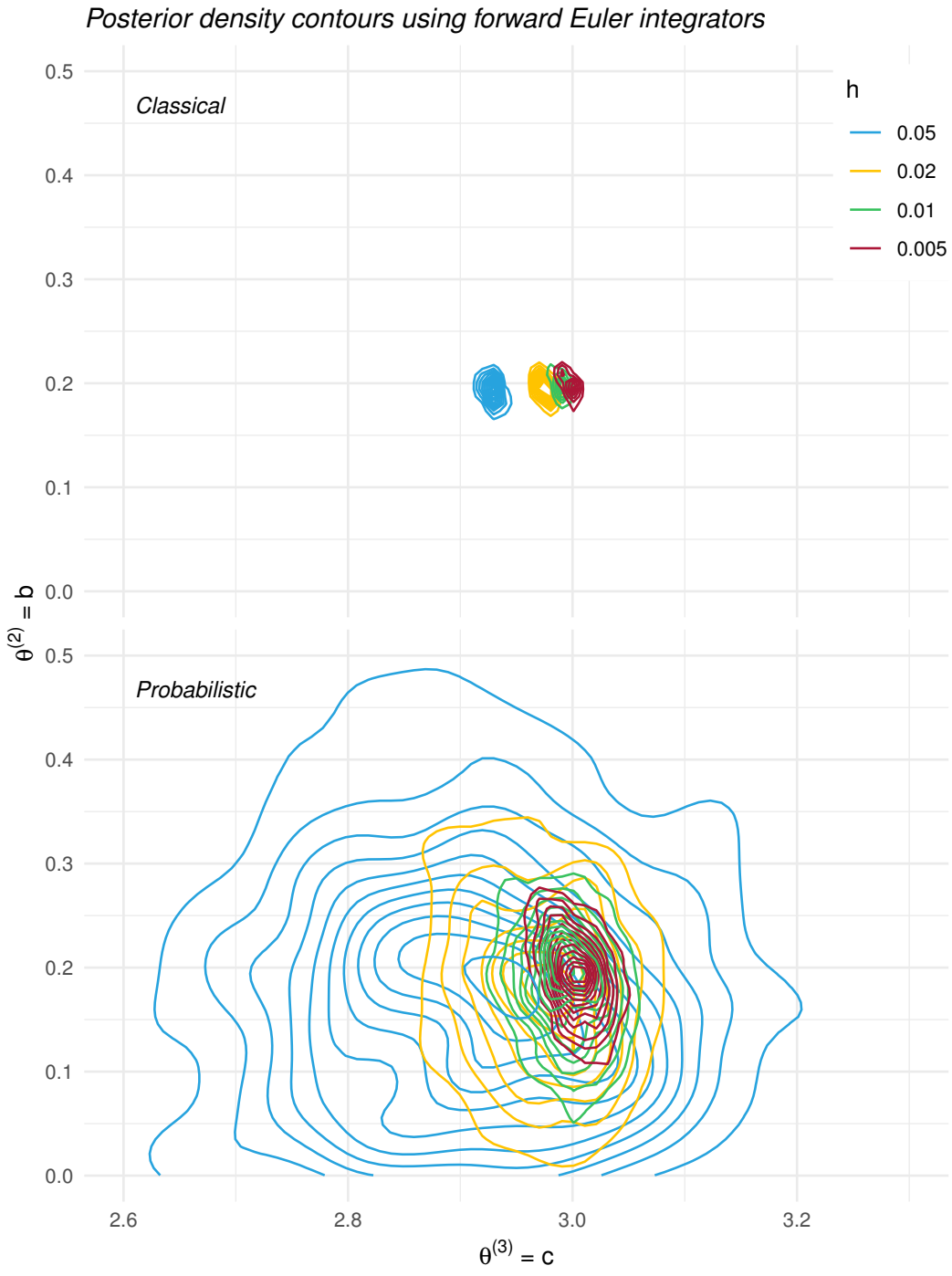


Figure 5.15: Posterior density contours of (b, c) for the FitzHugh–Nagumo inverse problem. The contours are calculated by kernel density estimates based on 1000 samples in each case. The top pane shows posteriors resulting when the forward model is solved by the classical forward Euler method, while the bottom pane shows those solved by the probabilistic forward Euler method. The different colours represent the simulation repeated at different step-sizes h.

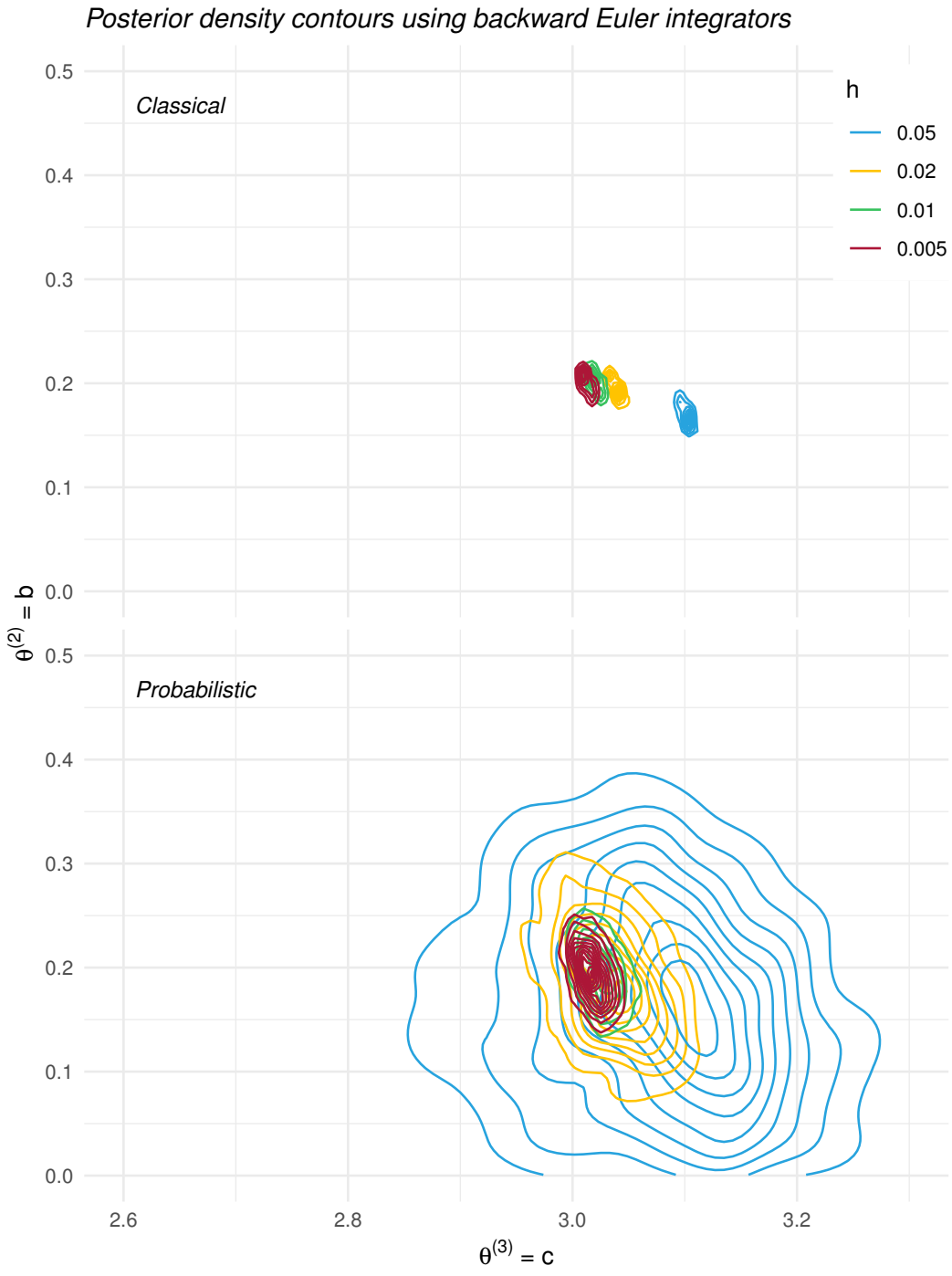


Figure 5.16: Posterior density contours of (b, c) for the FitzHugh–Nagumo inverse problem. The contours are calculated by kernel density estimates based on 1000 samples in each case. The top pane shows posteriors resulting when the forward model is solved by the classical backward Euler method, while the bottom pane shows those solved by the probabilistic backward Euler method. The different colours represent the simulation repeated at different step-sizes h.

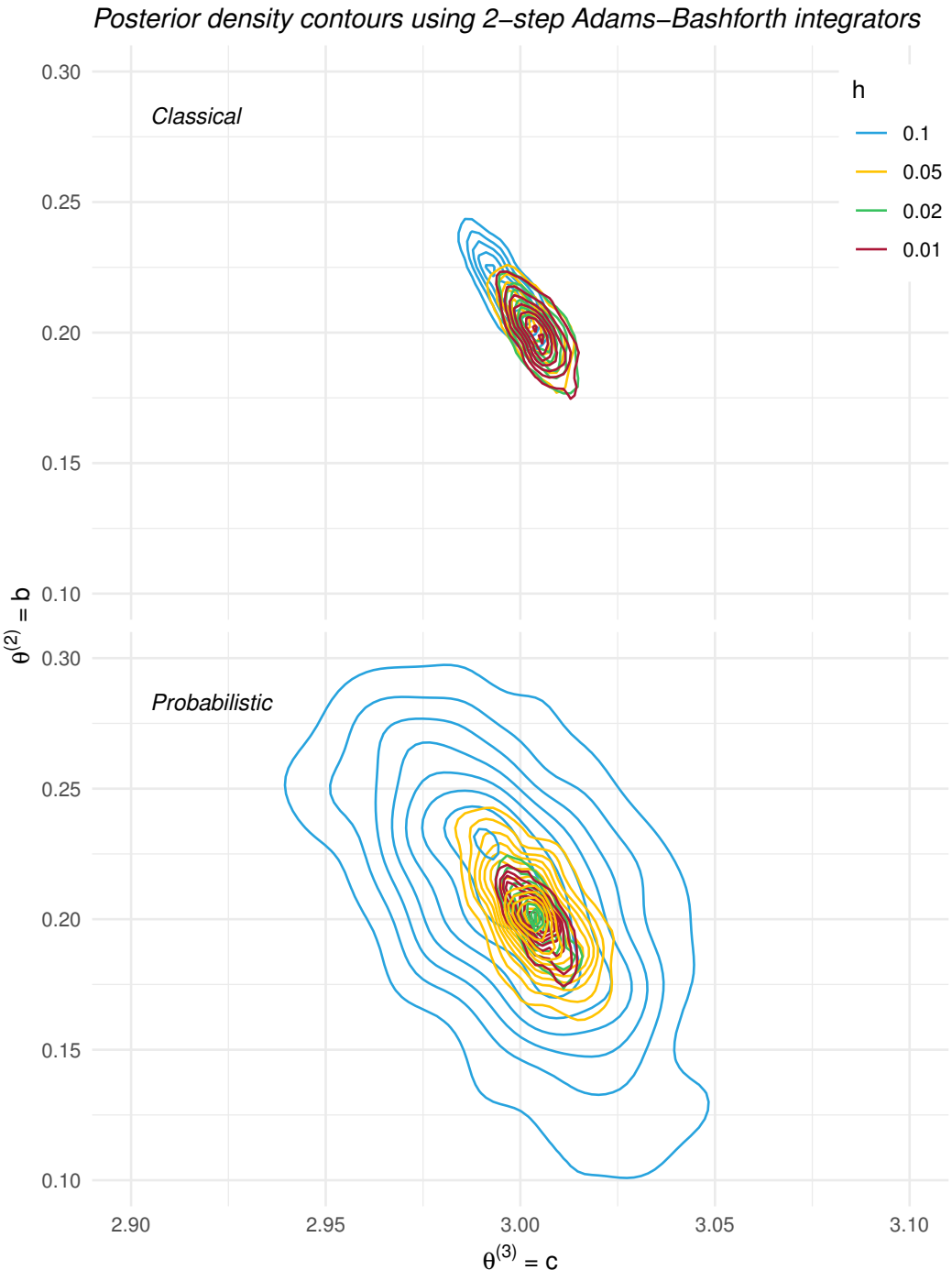


Figure 5.17: Posterior density contours of (b, c) for the FitzHugh–Nagumo inverse problem. The contours are calculated by kernel density estimates based on 1000 samples in each case. The top pane shows posteriors resulting when the forward model is solved by the classical 2-step Adams–Bashforth method, while the bottom pane shows those solved by the probabilistic 2-step Adams–Bashforth method. The different colours represent the simulation repeated at different step-sizes h.

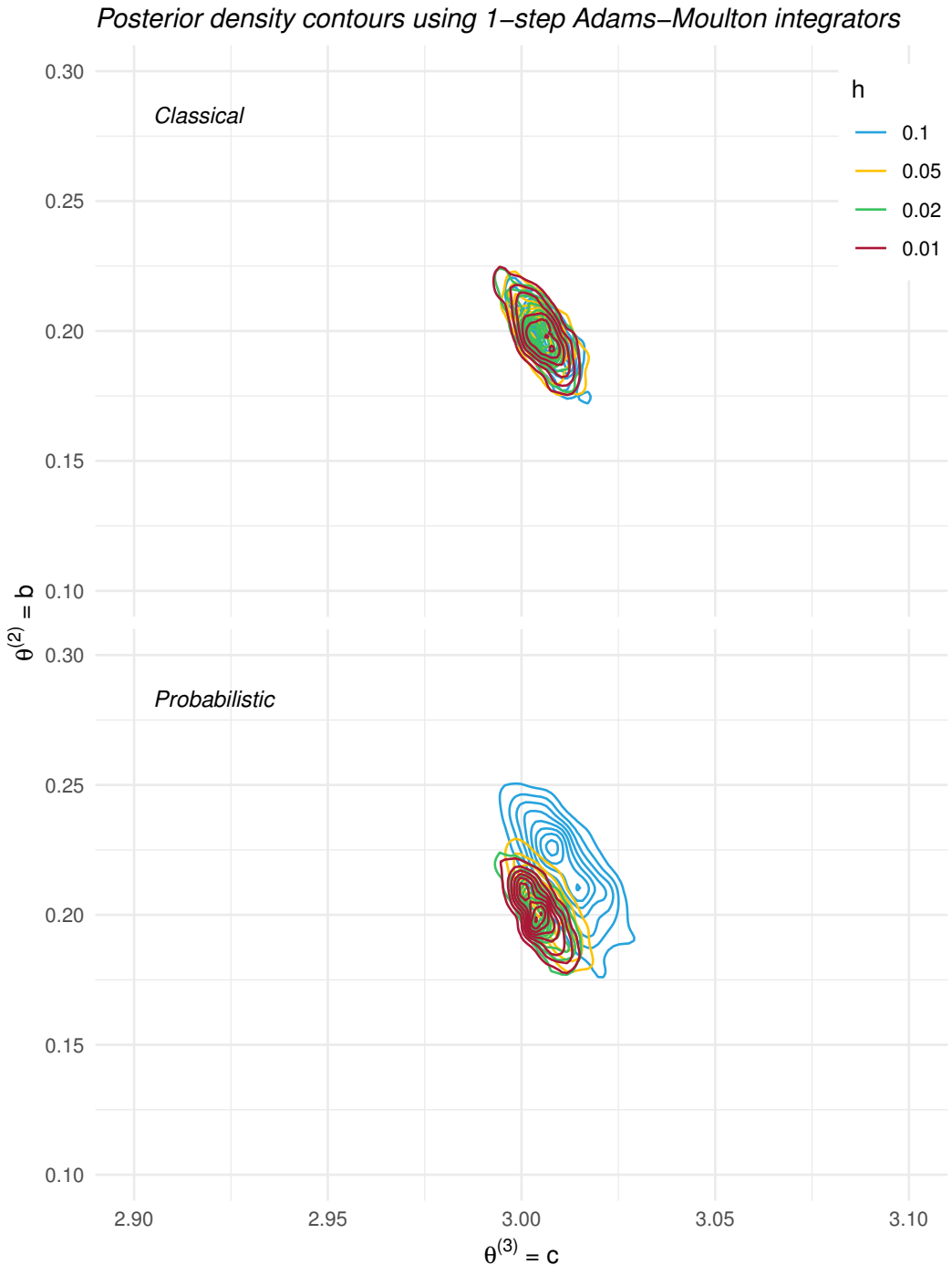


Figure 5.18: Posterior density contours of (b, c) for the FitzHugh–Nagumo inverse problem. The contours are calculated by kernel density estimates based on 1000 samples in each case. The top pane shows posteriors resulting when the forward model is solved by the classical 1-step Adams–Moulton method, while the bottom pane shows those solved by the probabilistic 1-step (second-order) Adams–Moulton method. The different colours represent the simulation repeated at different step-sizes h.

	h		$\widehat{\theta} := \mathbb{E}(\theta\|Y)$	$\sigma := \sqrt{\mathrm{Var}(\theta\|Y)}$	$\widehat{\theta} - \theta$	$\left\|\frac{\widehat{\theta} - \theta}{\sigma}\right\|$
CLASSICAL FORWARD EULER	0.1	a	0.1888	0.0022	-0.0112	5.15
		b	0.1834	0.0120	-0.0166	1.38
		c	2.8522	0.0061	-0.1478	24.26
	0.05	a	0.1940	0.0020	-0.0060	3.02
		b	0.1924	0.0118	-0.0076	0.64
		c	2.9287	0.0057	-0.0713	12.62
	0.02	a	0.1977	0.0018	-0.0023	1.24
		b	0.1962	0.0108	-0.0038	0.35
		c	2.9748	0.0057	-0.0252	4.45
	0.01	a	0.1991	0.0017	-0.0009	0.53
		b	0.1972	0.0109	-0.0028	0.25
		c	2.9899	0.0055	-0.0101	1.84
	0.005	a	0.1997	0.0016	-0.0003	0.17
		b	0.1973	0.0111	-0.0027	0.24
		c	2.9979	0.0053	-0.0021	0.39
PROBABILISTIC FORWARD EULER	0.1	a	0.1731	0.0959	-0.0269	0.28
		b	0.2548	0.1685	0.0548	0.32
		c	2.8444	0.2403	-0.1556	0.65
	0.05	a	0.1876	0.0606	-0.0124	0.20
		b	0.1880	0.1169	-0.0120	0.10
		c	2.9105	0.1370	-0.0895	0.65
	0.02	a	0.1984	0.0223	-0.0016	0.07
		b	0.1802	0.0788	-0.0198	0.25
		c	2.9715	0.0559	-0.0285	0.51
	0.01	a	0.1996	0.0119	-0.0004	0.03
		b	0.1853	0.0522	-0.0147	0.28
		c	2.9935	0.0309	-0.0065	0.21
	0.005	a	0.1995	0.0075	-0.0005	0.06
		b	0.1922	0.0375	-0.0078	0.21
		c	3.0032	0.0202	0.0032	0.16

Table 5.2: Posterior summary for $\theta = (a, b, c)$ in the FitzHugh–Nagumo model. The forward solve was undertaken using the classical *(top pane)* or probabilistic *(bottom pane)* forward Euler method. For each simulation, an MCMC was run giving 1000 sample trajectories $\theta^{[k]}$. The posterior ensemble mean $\widehat{\theta} := \mathbb{E}(\theta|Y)$ and standard deviation $\sigma := \sqrt{\mathrm{Var}(\theta|Y)}$ are given, along with the error $\widehat{\theta} - \theta$ of the mean estimator relative to the true values (0.2,0.2,3). The last column displays the relative magnitude of this error and the ensemble standard deviation.

	h		$\widehat{\theta} := \mathbb{E}(\theta\|Y)$	$\sigma := \sqrt{\mathrm{Var}(\theta\|Y)}$	$\widehat{\theta} - \theta$	$\left\|\frac{\widehat{\theta} - \theta}{\sigma}\right\|$
CLASSICAL BACKWARD EULER	0.05	a	0.2092	0.0015	0.0092	5.98
		b	0.1684	0.0125	-0.0316	2.52
		c	3.1015	0.0052	0.1015	19.69
	0.02	a	0.2035	0.0014	0.0035	2.44
		b	0.1949	0.0108	-0.0051	0.48
		c	3.0392	0.0047	0.0392	8.27
	0.01	a	0.2017	0.0015	0.0017	1.14
		b	0.2004	0.0109	0.0004	0.04
		c	3.0206	0.0052	0.0206	3.96
	0.005	a	0.2010	0.0015	0.0010	0.66
		b	0.2000	0.0109	0.0000	0.00
		c	3.0127	0.0050	0.0127	2.52
PROBABILISTIC BACKWARD EULER	0.05	a	0.2112	0.0357	0.0112	0.31
		b	0.1710	0.0933	-0.0290	0.31
		c	3.0713	0.0978	0.0713	0.73
	0.02	a	0.2068	0.0131	0.0068	0.52
		b	0.1833	0.0563	-0.0167	0.30
		c	3.0356	0.0374	0.0356	0.95
	0.01	a	0.2022	0.0072	0.0022	0.30
		b	0.1930	0.0355	-0.0070	0.20
		c	3.0247	0.0190	0.0247	1.30
	0.005	a	0.2011	0.0044	0.0011	0.25
		b	0.1939	0.0253	-0.0061	0.24
		c	3.0171	0.0129	0.0171	1.33

Table 5.3: Posterior summary for $\theta = (a, b, c)$ in the FitzHugh–Nagumo model. The forward solve was undertaken using the classical *(top pane)* or probabilistic *(bottom pane)* backward Euler method. For each simulation, an MCMC was run giving 1000 sample trajectories $\theta^{[k]}$. The posterior ensemble mean $\widehat{\theta} := \mathbb{E}(\theta|Y)$ and standard deviation $\sigma := \sqrt{\mathrm{Var}(\theta|Y)}$ are given, along with the error $\widehat{\theta} - \theta$ of the mean estimator relative to the true values (0.2,0.2,3). The last column displays the relative magnitude of this error and the ensemble standard deviation.

	h		$\widehat{\theta} := \mathbb{E}(\theta\|Y)$	$\sigma := \sqrt{\mathrm{Var}(\theta\|Y)}$	$\widehat{\theta} - \theta$	$\left\|\frac{\widehat{\theta} - \theta}{\sigma}\right\|$
CLASSICAL 2-STEP ADAMS–BASHFORTH	0.1	a	0.2008	0.0014	0.0008	0.58
		b	0.1965	0.0104	-0.0035	0.33
		c	3.0072	0.0048	0.0072	1.51
	0.05	a	0.2005	0.0015	0.0005	0.30
		b	0.1982	0.0114	-0.0018	0.16
		c	3.0056	0.0053	0.0056	1.06
	0.02	a	0.2004	0.0015	0.0004	0.26
		b	0.1993	0.0108	-0.0007	0.07
		c	3.0049	0.0052	0.0049	0.94
	0.01	a	0.2004	0.0016	0.0004	0.23
		b	0.1985	0.0115	-0.0015	0.13
		c	3.0055	0.0056	0.0055	0.97
PROBABILISTIC 2-STEP ADAMS–BASHFOTH	0.1	a	0.1988	0.0023	-0.0012	0.49
		b	0.2174	0.0177	0.0174	0.99
		c	3.0112	0.0083	0.0112	1.35
	0.05	a	0.1999	0.0016	-0.0001	0.05
		b	0.2035	0.0129	0.0035	0.27
		c	3.0070	0.0059	0.0070	1.18
	0.02	a	0.2002	0.0015	0.0002	0.17
		b	0.2006	0.0112	0.0006	0.05
		c	3.0052	0.0053	0.0052	0.98
	0.01	a	0.2003	0.0015	0.0003	0.21
		b	0.2002	0.0111	0.0002	0.02
		c	3.0047	0.0051	0.0047	0.91

Table 5.4: Posterior summary for $\theta = (a, b, c)$ in the FitzHugh–Nagumo model. The forward solve was undertaken using the classical *(top pane)* or probabilistic *(bottom pane)* 2-step Adams–Bashforth method. For each simulation, an MCMC was run giving 1000 sample trajectories $\theta^{[k]}$. The posterior ensemble mean $\widehat{\theta} := \mathbb{E}(\theta|Y)$ and standard deviation $\sigma := \sqrt{\mathrm{Var}(\theta|Y)}$ are given, along with the error $\widehat{\theta} - \theta$ of the mean estimator relative to the true values (0.2,0.2,3). The last column displays the relative magnitude of this error and the ensemble standard deviation.

	h		$\widehat{\theta} := \mathbb{E}(\theta\|Y)$	$\sigma := \sqrt{\mathrm{Var}(\theta\|Y)}$	$\widehat{\theta} - \theta$	$\left\|\frac{\widehat{\theta} - \theta}{\sigma}\right\|$
CLASSICAL 1-STEP ADAMS–MOULTON	0.1	a	0.2008	0.0014	0.0008	0.58
		b	0.1965	0.0104	-0.0035	0.33
		c	3.0072	0.0048	0.0072	1.51
	0.05	a	0.2005	0.0015	0.0005	0.30
		b	0.1982	0.0114	-0.0018	0.16
		c	3.0056	0.0053	0.0056	1.06
	0.02	a	0.2004	0.0015	0.0004	0.26
		b	0.1993	0.0108	-0.0007	0.07
		c	3.0049	0.0052	0.0049	0.94
	0.01	a	0.2004	0.0016	0.0004	0.23
		b	0.1985	0.0115	-0.0015	0.13
		c	3.0055	0.0056	0.0055	0.97
PROBABILISTIC 1-STEP ADAMS–MOULTON	0.1	a	0.1988	0.0023	-0.0012	0.49
		b	0.2174	0.0177	0.0174	0.99
		c	3.0112	0.0083	0.0112	1.35
	0.05	a	0.1999	0.0016	-0.0001	0.05
		b	0.2035	0.0129	0.0035	0.27
		c	3.0070	0.0059	0.0070	1.18
	0.02	a	0.2002	0.0015	0.0002	0.17
		b	0.2006	0.0112	0.0006	0.05
		c	3.0052	0.0053	0.0052	0.98
	0.01	a	0.2003	0.0015	0.0003	0.21
		b	0.2002	0.0111	0.0002	0.02
		c	3.0047	0.0051	0.0047	0.91

Table 5.5: Posterior summary for $\theta = (a, b, c)$ in the FitzHugh–Nagumo model. The forward solve was undertaken using the classical *(top pane)* or probabilistic *(bottom pane)* 1-step Adams–Moulton method. For each simulation, an MCMC was run giving 1000 sample trajectories $\theta^{[k]}$. The posterior ensemble mean $\widehat{\theta} := \mathbb{E}(\theta|Y)$ and standard deviation $\sigma := \sqrt{\mathrm{Var}(\theta|Y)}$ are given, along with the error $\widehat{\theta} - \theta$ of the mean estimator relative to the true values (0.2,0.2,3). The last column displays the relative magnitude of this error and the ensemble standard deviation.

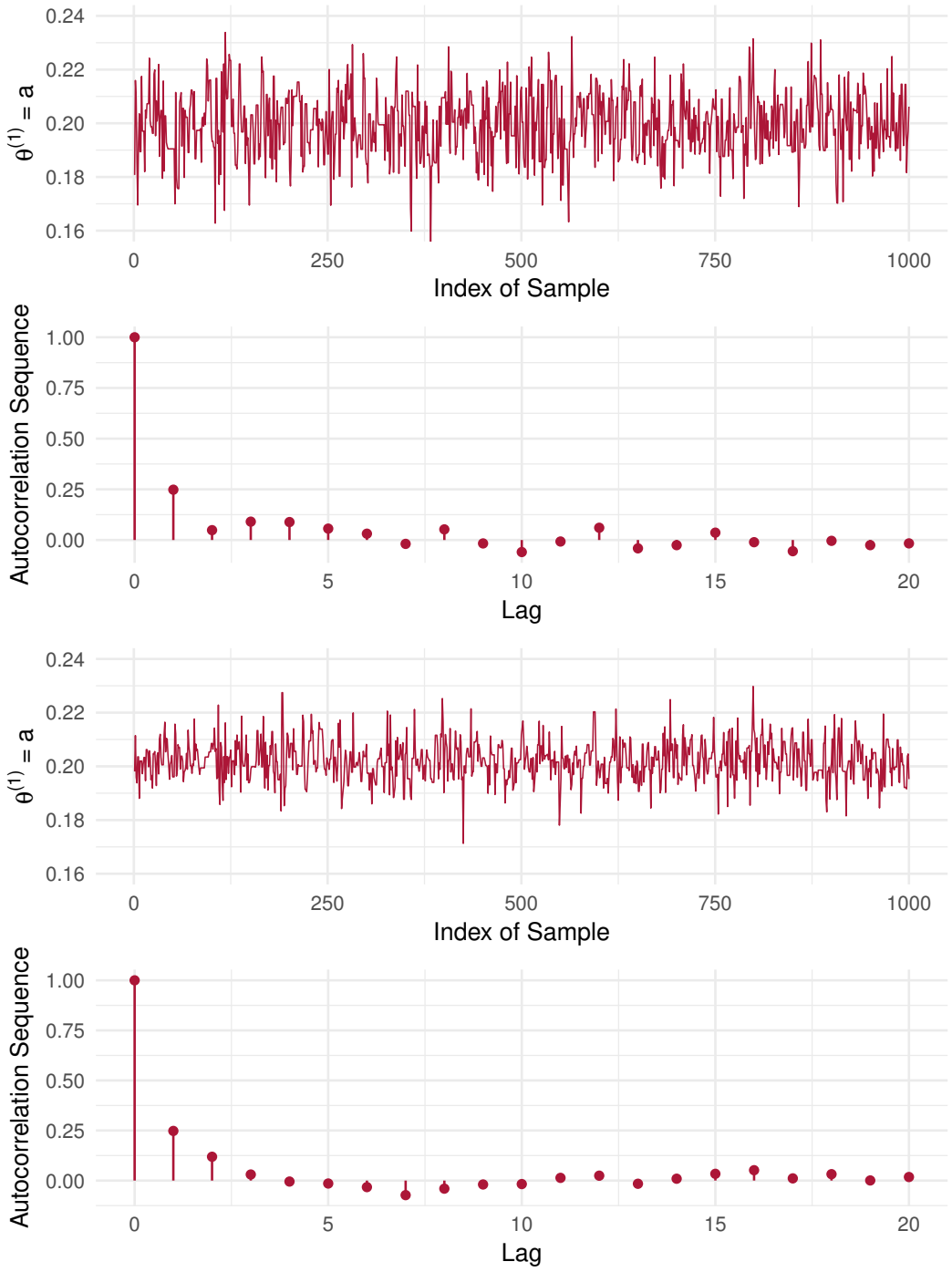


Figure 5.19: Example performance diagnostic plots of the MCMC simulations run to infer the parameters of the FitzHugh–Nagumo model. These plots summarise the sampling for the parameter a and correspond to the case $h = 0.01$ (the green posterior in figures 5.15 and 5.16). The top two panes display respectively the path of the Markov chain and its autocorrelation sequence when the forward model is solved using the probabilistic forward Euler method, and the bottom two panes display the same for the probabilistic backward Euler method.

We also ran simulations for third- and higher-orders methods, and found an exacerbation of these issues. The simplicity of the test problem, combined with lack of stability of those higher-order methods even in the classical case, means that the desired effect becomes ever more difficult to demonstrate. Further investigation is required to identify problems for which these higher-order integrators may be more appropriate, and for which stability issues do not form a practical impediment to their use. We briefly reconsider this issue in Section 6.2.3.

5.5 UNCERTAINTY IN THE FORWARD MODEL

In this section we present an interesting by-product of the parameter inference process undertaken in Section 5.4.2. As discussed there, the result of performing an MCMC simulation on the distribution $p(\theta, Z(\xi)|Y)$ is the acquisition of paired samples $\{\theta^{[m]}, Z^{[m]}\}$ of model parameters and forward model trajectories. Marginalising over the paths Z—in that context, simply disregarding those samples—allowed us to perform posterior inference on θ.

We now explore the opposite scenario. Consider a situation in which we have a dataset Y and a differential equation model of the form (1.11) whose parameters are not known *a priori*. Our object of interest in this scenario may be the value of the solution $x(t_{\text{end}})$ of the initial value problem at time t_{end}. In this case, we can instead marginalise the parameter θ by discarding the samples $\{\theta^{[m]}\}$ and interpret instead the ensemble of trajectories $\{Z^{[m]}\}$ as representing the uncertainty in $x(t)$, in the manner described in Section 2.2.

We demonstrate this by once again returning to the Brusselator model. We generate 10 synthetic data at times $t_Y = 1, 2, \ldots, 10$, each with independent measurement error $\sigma = 2.5 \times 10^{-3} \cdot \mathbb{I}_2$, exactly as we did for the FitzHugh–Nagumo model in Section 5.1.1. We use the parameters $\theta = (1.4, 3)$, though of course after generating the data we proceed by considering them unknown. Once again, we assign log-normal priors $p(\log \theta^{(v)}) \sim \mathcal{N}(\theta^{(v)}_{\text{true}}, 1)$ independently for each component of θ, and assume a Gaussian error model with σ known, as well as known initial value $X_0 = (1, 2)^T$.

We integrate the system over a time window much longer than the range of locations at which we have generated data. For lower-order methods, or where too large a step-size h is chosen, we expect that at some point the integrators will fail, in a similar manner to that seen when integrating the fully-specified forward model in Section 5.4.2. As there, we desire the probabilistic versions of the algorithms to detect this breakdown in some way, indicating to the user that the method selected is inappropriate in the given context. For this example, we will assume that the values of the solution function $x_\theta(t)$ at the three times $t = 10, 30, 50$ are of interest.

In figures 5.20–5.23, we plot the summary statistics of ensembles of 1000 solution trajectories $\{Z^{[m]}\}$ of the forward model, each corresponding to a sample $\theta^{[m]}$ generated by an MCMC algorithm analogous to that explained in Section 5.4.1. The figures plot the trajectories resulting from the use of the classical *(top two panes)* and probabilistic *(bottom two panes)* integrators for respectively the first- and second- order implicit and explicit methods. The step-size throughout is $h = 0.1$, and we display the results for the interval $t \in [0, 50]$.

As in figures 5.7–5.14, the three coloured bands represent 1σ, 2σ and 3σ bands. The dashed blue line describes the ensemble mean—this is *not* the single trajectory of one particular 'mean' sample but instead the result of calculating the mean for each time point t_i separately and plotting the path of these means as a sort of pseudo-trajectory. The solid black line again describes the reference solution $x_\theta(t)$.

For both first-order methods (figures 5.20 and 5.22), the use of a classical solver with this value of h visibly fails to correctly capture the underlying dynamics. For a periodic system such as the Brusselator, there exists a horizon beyond which the global error in an iterative integrator accumulates in such a way that effectively no informative conclusion can be drawn about the value of the solution. This is a well-known phenomenon in numerical integration, and was discussed in Section 4.4.1 in the context of integrating the fully-specified forward model. The first-order algorithms have clearly reached and passed this horizon during the interval shown.

Despite some visible notion of solution uncertainty arising from the variation in the parameter θ, any sensible estimator for $x(50)$ derived from the blue paths in the top panes of figures 5.20 and 5.22 and reported—the ensemble mean, say—will be an obviously poor estimate for the true solution, shown in black. Furthermore, this poor estimate will be reported with high confidence since no indication exists in the output that it may be wrong.

In thinking about this conclusion, it is instructive to consider the sequences of black dots in figures 5.7 and 5.11, which describe the trajectory of the classical method when the parameter value θ was taken to be known. Now with θ unknown, the top panes of figures 5.20 and 5.22 give the range of the solution paths arising from the joint MCMC procedure over θ and Z. This ensemble of paths is equivalent in a sense to the black dots in figures 5.7 and 5.11—the non-zero variance of the ensemble arising from the parameter uncertainty in this setup. However, while these bands do have noticeable non-zero width, they clearly do not correctly track the true underlying dynamics. Thus in this case, the parameter uncertainty has a much smaller effect on the marginal path posterior $p(Z(\xi)|Y)$ than does the error in the numerical method being used to solve the problem.

The probabilistic versions of these simulations, the output of which is shown in the bottom two panes of figures 5.20 and 5.22, confirm the inadequacy of these methods used for this problem. In fact, the output of the probabilistic forward Euler method is further worsened by the uncertainty in θ (compare figure 5.7), resulting in the suggested distribution for $x(50)$ becoming almost totally uninformative. As we argued in Section 4.4.1, this is itself an informative statement—rather than reporting an incorrect estimator with high confidence, the practitioner can instead justifiably report the inadequacy of the chosen solver.

Results for the second-order methods are similar—in both the explicit and implicit cases shown in figures 5.21 and 5.23, parameter uncertainty causes a visible increase in the width of the credible intervals when compared with the plots in figures 5.8 and 5.13.

The analysis in this section can be interpreted in a more quantitative manner by considering the tabulated output in tables 5.6–5.7. These give posterior summaries for the distributions of the random variables $x(10)$, $x(30)$ and $x(50)$, with the figures relating to the first component $x^{(1)}$ of the two-dimensional Brusselator system given in figure 5.6 and those for the second component $x^{(2)}$ in figure 5.7.

In these tables, columns 3–7 provide the equivalent summaries as were given for the parameters a, b, c in tables 5.2–5.5. The column headed $\left|\sigma^{-1}(\widehat{X}_t - X_t)\right|$ gives the relative scale of the error made by reporting the posterior mean as an estimator for the unknown variable. Consider the figures given in this column for the classical forward Euler method. As in the case of the parameter posteriors explored in Section 5.4.2, these suggest that the error of this estimator is many times larger than the ensemble standard deviation that our model takes as the probability distribution of the randomised numerical method.

However, unlike there, this statistic is unhelpful elsewhere in the simulation output. In particular, while the ensemble standard deviation $\sigma := \sqrt{\mathrm{Var}(X_t|Y)}$ typically increases with t and decreases when considering the methods of higher accuracy, the estimator error $\widehat{X}_t - X_t$ does not increase in an unbounded fashion, as it did during posterior inference over the parameters. The reason for this is that the periodic nature of the dynamics means that the effective range of x is constrained to a compact subspace of $\mathbb{R}^2$. Thus, even for numerical methods which have obviously failed to correctly track the dynamics, the effect is an 'out-of-phase' trajectory, rather than one with ever-growing magnitude of error. The expression $\left|\sigma^{-1}(\widehat{X}_t - X_t)\right|$ then becomes a problematic summary statistic since it does not necessarily evaluate to higher values in cases of evident poor integrator performance.

This point is observable in the case of the classical backward Euler method by considering the entries in the penultimate column of table 5.6. Here, while the quality of integration clearly declines as time increases, with the path of the credible band

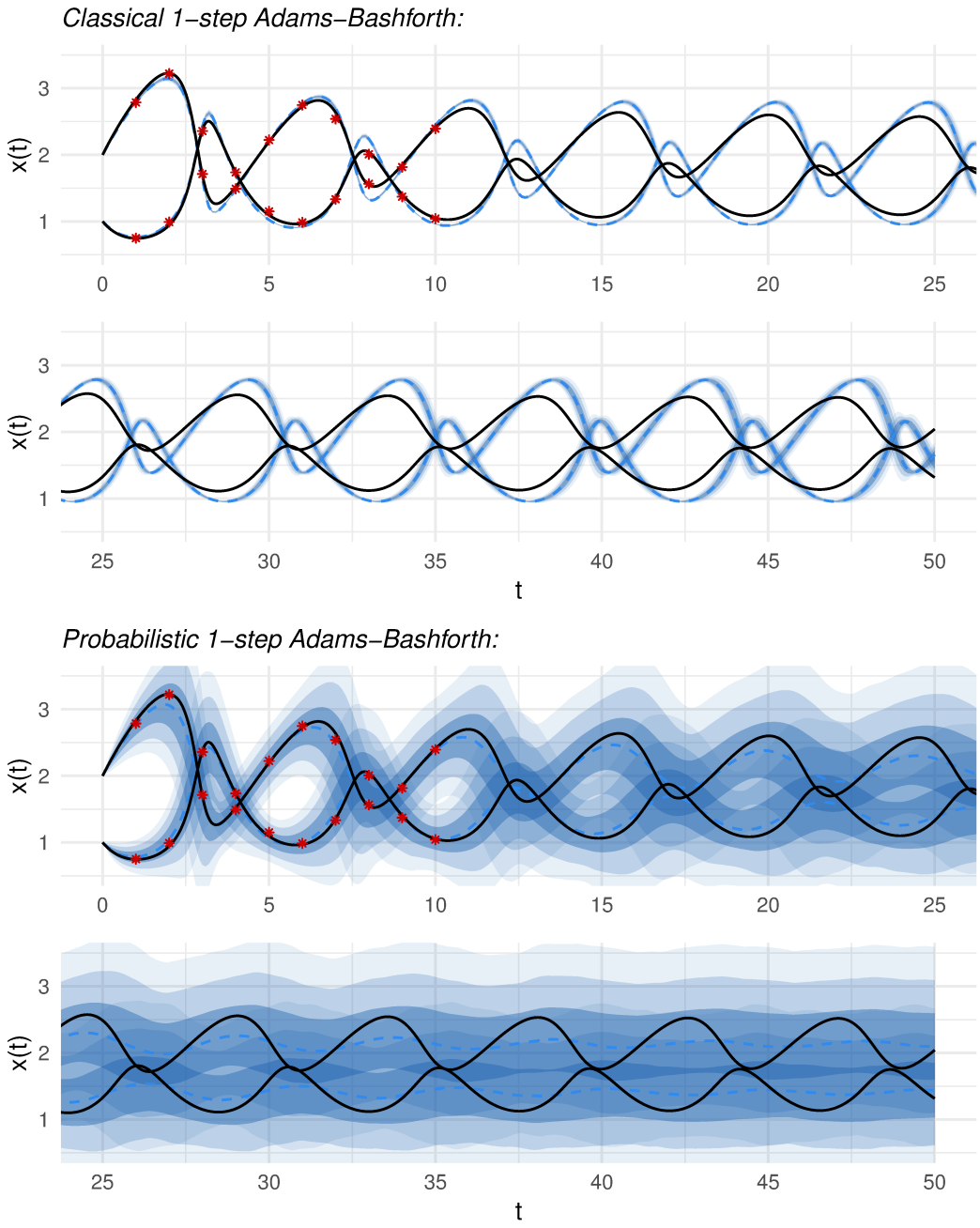


Figure 5.20: Summarised ensemble path plot of the Brusselator system solved in the range $t \in [0, 50]$ using first- and second-order probabilistic Adams–Bashforth integrators for the forward solve, in an unknown-parameter setting. The dashed blue lines give the approximate trajectory of the ensemble mean of 1000 samples, while the coloured bands represent 1σ, 2σ and 3σ intervals. The solid black line gives an reference solution calculated using a fine-mesh Runge–Kutta solver. The 20 red points represent the synthetic data generated for the simulation.

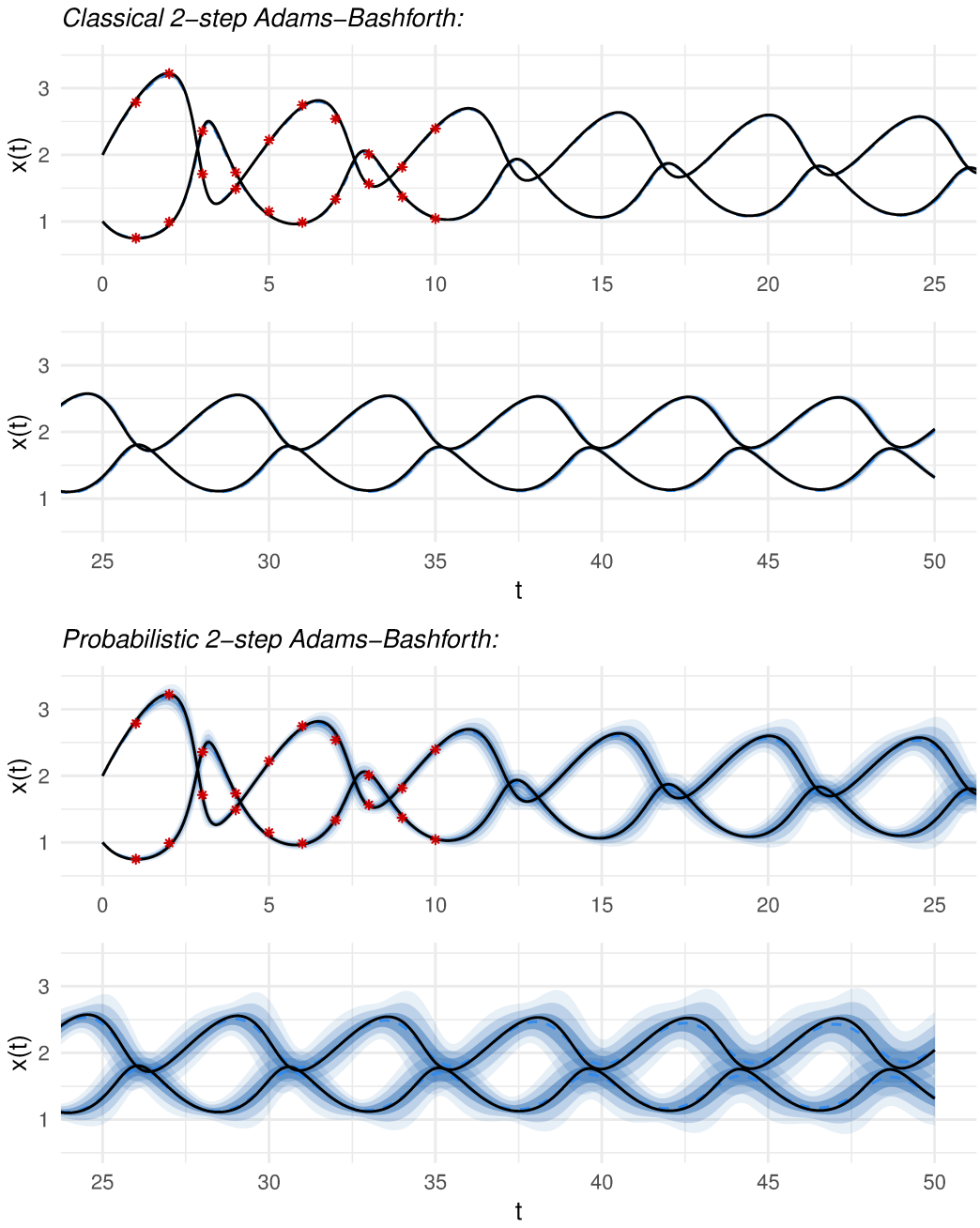


Figure 5.21: Summarised ensemble path plot of the Brusselator system solved in the range $t \in [0, 50]$ using first- and second-order probabilistic Adams–Moulton integrators for the forward solve, in an unknown-parameter setting. The dashed blue lines give the approximate trajectory of the ensemble mean of 1000 samples, while the coloured bands represent 1σ, 2σ and 3σ intervals. The solid black line gives an reference solution calculated using a fine-mesh Runge–Kutta solver. The 20 red points represent the synthetic data generated for the simulation.

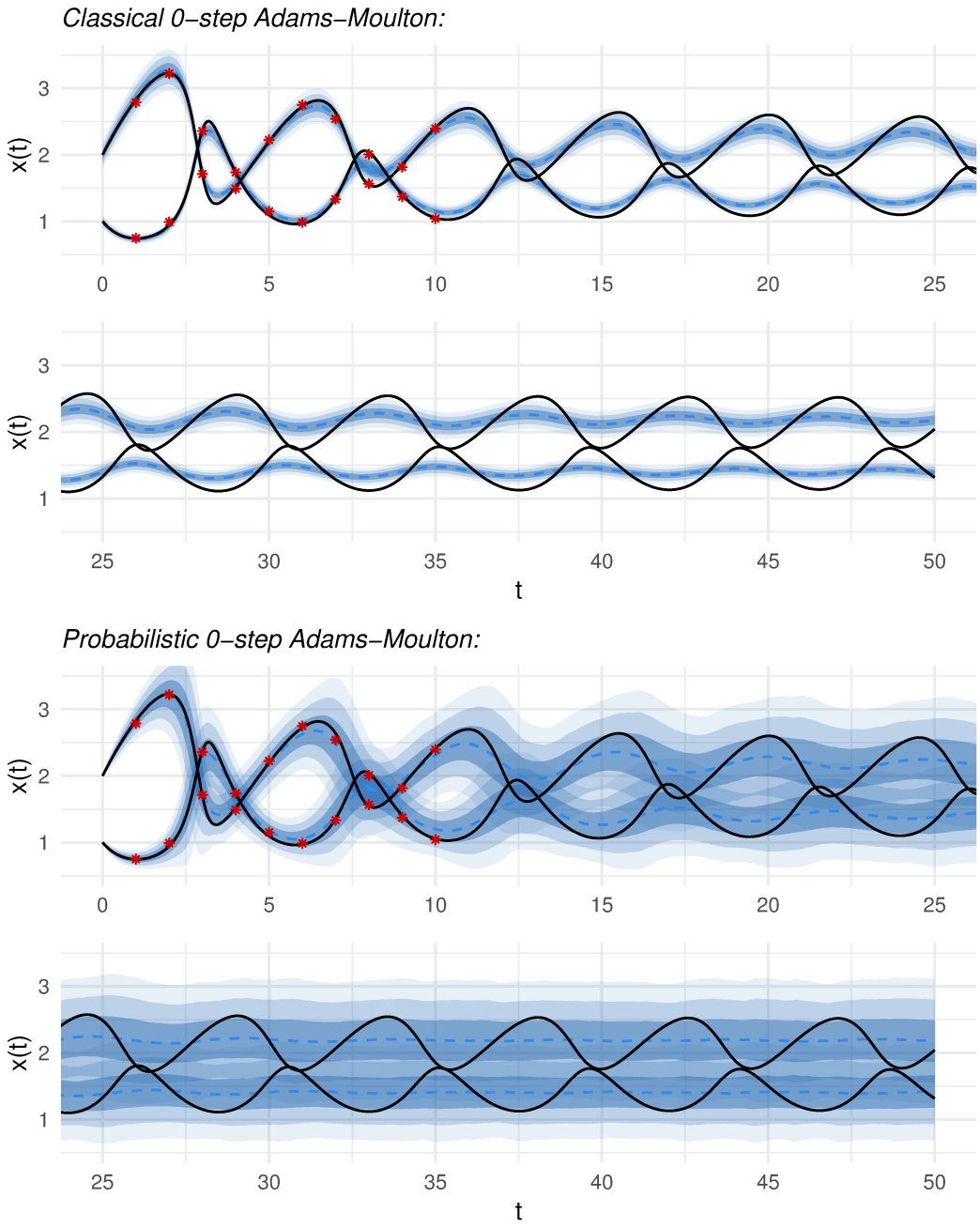


Figure 5.22: Summarised ensemble path plot of the Brusselator system solved in the range $t \in [0, 50]$ using first- and second-order probabilistic Adams–Bashforth integrators for the forward solve, in an unknown-parameter setting. The dashed blue lines give the approximate trajectory of the ensemble mean of 1000 samples, while the coloured bands represent 1σ, 2σ and 3σ intervals. The solid black line gives an reference solution calculated using a fine-mesh Runge–Kutta solver. The 20 red points represent the synthetic data generated for the simulation.

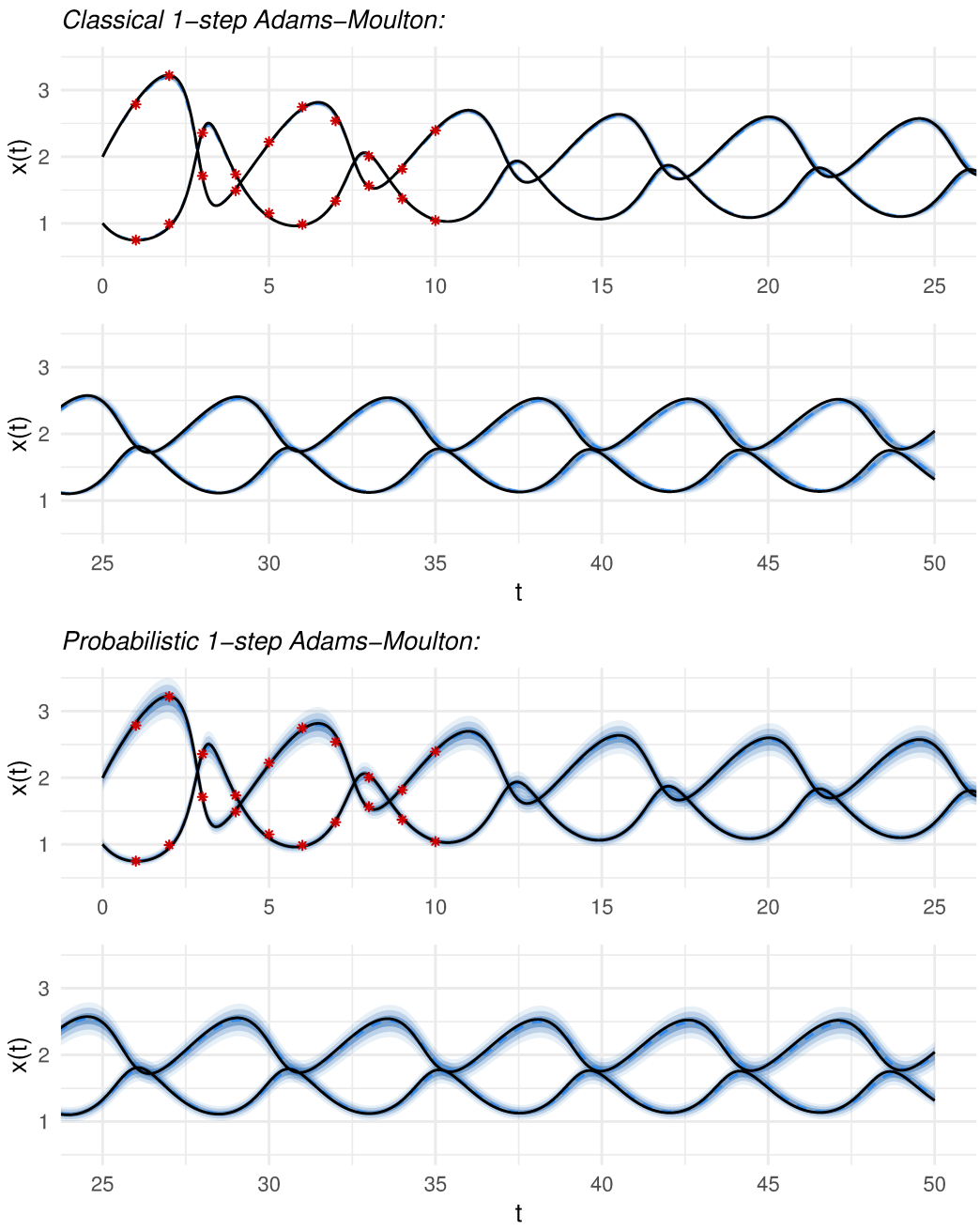


Figure 5.23: Summarised ensemble path plot of the Brusselator system solved in the range $t \in [0, 50]$ using first- and second-order probabilistic Adams–Moulton integrators for the forward solve, in an unknown-parameter setting. The dashed blue lines give the approximate trajectory of the ensemble mean of 1000 samples, while the coloured bands represent 1σ, 2σ and 3σ intervals. The solid black line gives an reference solution calculated using a fine-mesh Runge–Kutta solver. The 20 red points represent the synthetic data generated for the simulation.

			FIRST COMPONENT				
		t	$\widehat{X}_t := \mathbb{E}(X_t \mid Y)$	$\sigma := \sqrt{\mathrm{Var}(X_t \mid Y)}$	$\widehat{X}_t - X_t$	$\left\lvert \frac{\widehat{X}_t - X_t}{\sigma} \right\rvert$	$\frac{\sigma \times 100}{\text{Range}}$
CLASSICAL INTEGRATION OF FORWARD MODEL	AB1	10	0.961	0.002	-0.093	45.92	0.3
		30	1.443	0.055	-0.149	2.725	9.3
		50	1.520	0.071	0.203	2.862	12.0
	AB2	10	1.049	0.001	-0.004	3.260	0.2
		30	1.561	0.016	-0.031	1.979	2.7
		50	1.319	0.013	0.002	0.156	2.3
	AM0	10	1.152	0.037	0.098	2.697	6.2
		30	1.472	0.048	-0.121	2.497	8.2
		50	1.386	0.044	0.069	1.575	7.5
	AM1	10	1.057	0.005	0.003	0.677	0.8
		30	1.531	0.035	-0.062	1.773	5.9
		50	1.376	0.033	0.059	1.798	5.6
PROBABILISTIC INTEGRATION OF FORWARD MODEL	AB1	10	1.062	0.169	0.009	0.051	28.8
		30	1.439	0.417	-0.154	0.369	71.0
		50	1.438	0.408	0.121	0.296	69.4
	AB2	10	1.055	0.024	0.002	0.064	4.0
		30	1.570	0.156	-0.023	0.146	26.5
		50	1.335	0.139	0.018	0.131	23.7
	AM0	10	1.200	0.137	0.146	1.069	23.2
		30	1.409	0.243	-0.183	0.754	41.4
		50	1.421	0.244	0.104	0.427	41.4
	AM1	10	1.057	0.033	0.003	0.086	5.7
		30	1.545	0.052	-0.047	0.909	8.9
		50	1.359	0.046	0.042	0.914	7.8

Reference solution: $x^{(1)}(10,30,50) = (1.0538, 1.5924, 1.3170)$

$$\text{Range} = \max_{t \in [10,50]} x^{(1)}(t) - \min_{t \in [10,50]} x^{(1)}(t) = 0.5877$$

Table 5.6: Posterior summary for the first component $x^{(1)}(t)$ in the Brusselator model simulated with parameters assumed unknown. For each method, an MCMC was run giving 1000 sample trajectories $Z^{[k]}$. Collecting the ensemble of values at times $t = 10, 30, 50$, the table gives their mean, standard deviation, and the error of the mean estimator as compared to a reference solution. The relative scale of this error is given in the second-to-last column. The last column gives a measure of the scale of the sample standard deviation as a proportion of the estimated global range of the true solution, expressed as a percentage.

		SECOND COMPONENT				
	t	$\widehat{X}_t := \mathbb{E}(X_t \mid Y)$	$\sigma := \sqrt{\mathrm{Var}(X_t \mid Y)}$	$\widehat{X}_t - X_t$	$\left\lvert \frac{\widehat{X}_t - X_t}{\sigma} \right\rvert$	$\frac{\sigma \times 100}{\text{Range}}$
CLASSICAL INTEGRATION OF FORWARD MODEL						
AB1	10	2.446	0.009	0.039	4.385	1.2
	30	2.533	0.0578	0.350	6.061	7.6
	50	1.657	0.059	-0.385	6.555	7.7
AB2	10	2.393	0.004	-0.014	3.684	0.5
	30	2.207	0.022	0.025	1.118	2.9
	50	2.016	0.019	-0.026	1.327	2.5
AM0	10	2.361	0.075	-0.046	0.610	9.9
	30	2.161	0.069	-0.022	0.316	9.1
	50	2.177	0.070	0.135	1.938	9.1
AM1	10	2.387	0.010	-0.020	1.982	1.3
	30	2.234	0.046	0.050	1.105	6.0
	50	1.974	0.041	-0.068	1.655	5.4
PROBABILISTIC INTEGRATION OF FORWARD MODEL						
AB1	10	2.378	0.261	-0.029	0.110	34.3
	30	2.177	0.502	-0.006	0.0127	65.9
	50	2.098	0.496	0.057	0.114	65.2
AB2	10	2.395	0.048	-0.012	0.258	6.3
	30	2.160	0.212	-0.022	0.106	27.8
	50	2.054	0.185	0.012	0.066	24.3
AM0	10	2.309	0.213	-0.098	0.459	28.0
	30	2.203	0.298	0.020	0.068	39.2
	50	2.176	0.311	0.134	0.433	40.8
AM1	10	2.388	0.076	-0.019	0.248	9.9
	30	2.213	0.074	0.030	0.405	9.7
	50	1.991	0.067	-0.051	0.766	8.7

Reference solution: $x^{(2)}(10,30,50) = (2.4071, 2.1829, 2.0417)$

$$\text{Range} = \max_{t\in[10,50]} x^{(2)}(t) - \min_{t\in[10,50]} x^{(2)}(t) = 0.7618$$

Table 5.7: Posterior summary for the second component $x^{(2)}(t)$ in the Brusselator model simulated with parameters assumed unknown. For each method, an MCMC was run giving 1000 sample trajectories $Z^{[k]}$. Collecting the ensemble of values at times $t = 10, 30, 50$, the table gives their mean, standard deviation, and the error of the mean estimator as compared to a reference solution. The relative scale of this error is given in the second-to-last column. The last column gives a measure of the scale of the sample standard deviation as a proportion of the estimated global range of the true solution, expressed as a percentage.

bearing less and less resemblance to the true solution, the quantity $\left|\sigma^{-1}(\widehat{X}_t - X_t)\right|$ decreases from $x(10)$ to $x(30)$, and then once again to $x(50)$.

As a result of this observation, we introduce a different statistic $\sigma/\text{Range}(\%)$. This expresses—as a percentage—the scale of the accumulated error represented by the ensemble of randomised trajectories compared to the effective range of the dynamics. The latter quantity is given for the component v by

$$\text{Effective range}^{(v)} = \max_{t\in[10,50]} x^{(v)}(t) - \min_{t\in[10,50]} x^{(v)}(t) \tag{5.10}$$

We calculate this using the exact dynamics, though an effective measure could also be estimated using the output trajectories only. In the situation that the distribution of trajectories covers a large proportion of the effective range of the solution, this statistic indicates that the integrator has failed to correctly track the underlying process. In the case of the backward Euler method just described, table 5.6 shows that both σ and $\sigma/\text{Range}(\%)$ remain close to constant as we pass from $x(30)$ to $x(50)$, allowing the practitioner to conclude that the output is uninformative as an estimator for the true dynamics.

6

DISCUSSION & CONCLUSION

6.1 SUMMARY OF CONTRIBUTION

This thesis examines the principles and the practice of adopting a probabilistic approach to the integration of initial value problems. We have identified and explored in detail a number of different strands of this concept, most of which fall under the umbrella of probabilistic numerical methods, though we also considered approaches which pre-date this relatively recent paradigm.

In Section 2.2 we drew attention to the fundamental difference in the structure of the statistical model adopted by randomising classical numerical methods, as compared to those approaches which assimilate numerical data directly into a functional model. This difference has been somewhat glossed over in previous treatments of the topic—our exposition has sought to highlight it in order to make explicit the merits and shortcomings of each scheme.

In Chapter 3 we introduced a novel extension to the work of Conrad et al. [Con16] which generalises the one-step integrators proposed there to the multistep setting—specifically, we defined a family of probabilistic Adams–Bashforth integrators and proved their convergence rigorously, in the process giving bounds on the scale of the permissible stepwise perturbations compatible with this convergence.

The construction was motivated from a Gaussian process perspective, revealing an interesting resemblance to several earlier works by other authors. However, we also described the algorithm in terms of additive stepwise perturbations of the classical

method, with the latter presentation much more similar to the randomised one-step integrators of Conrad et al. [Con16].

Having discussed several subtle issues which arose after the attempt to extend this construction to implicit multistep methods, we introduced a totally new paradigm to fix the inconsistencies thrown up by previous approaches. The resulting probabilistic Adams–Moulton methods do not advance from step to step with an assumed Gaussian model for the local error. This feature simultaneously gives them the ability to characterise the numerical error in a richer way, but also increases the cost of implementing them.

Finally we considered the thorny issue of calibration of probabilistic methods. We first adopted and then significantly extended the scale-matching idea in Conrad et al. [Con16]—the latter because their method is not directly applicable to our newly-proposed implicit methods without modification. In Chapter 5 this calibration procedure was implemented in the context of two test problems, for both the explicit and implicit integrators. After an exploration of different strategies for stochastic sampling, we then applied the new algorithms in the context of parameter inference in the inverse problem setting.

The simulations in Chapter 5 are intended to provide indicative results and showcase the methods as applied to some simple problems. In particular, we note that the calibration procedure—an essential but hitherto largely overlooked component of any practical implementation of probabilistic numerical algorithms—can be said to work effectively in the tested contexts, though it is certainly not computationally cheap. The potential to apply probabilistic methods to more complex problems—for instance to higher-dimensional dynamical systems, or in settings in which the experimental data Y is less structured—is predicated on the development of methods which are both theoretically sound, as we have demonstrated in this thesis, but also are practical computationally. It is not our intention to avoid the latter issue entirely, but it is not the primary focus of our study, and for this reason we have not specifically highlighted the speed benchmarks of our test simulations.

6.2 FUTURE AVENUES FOR RESEARCH

In the remainder of this chapter, we make some miscellaneous observations relating to methods we have studied, and discuss some open questions which may prove fruitful avenues for future research.

6.2.1 *Extensions to related integrators*

In Section 3.1 we summarised a range of different families of iterative ODE solvers, and made passing reference to the overall framework of 'general linear methods' introduced by Butcher [But06] of which these methods are special cases. Forming randomised versions of many of these methods would be a straightforward extension to the work presented here. Theorem 3, in which we established the conditions for the convergence of randomised linear multistep methods, already covers several additional named methods in widespread use.

For instance, backward differentiation formulae (BDF)—in which a polynomial is constructed to interpolate past *state* values Z_{i-j} (rather than function values F_{i-j} as in Adams-type methods)—are of the specified form, as can be observed by examining equation (3.25) and taking $a_s \neq 0$; $\sum_j a_j = 1$; $b_{-1} \neq 0$; and $b_j = 0$ for $j \neq -1$. These methods are typically only used in their implicit form, since the explicit versions ($b_{-1} = 0$) have poor stability properties [Hai08, §III.1]. This immediately suggests a discrepancy relation—analogous to that defined in equation (4.10)—of the form

$$r(z) = \frac{z - \sum_{j=0}^{s-1} \beta_{j,s}^{BDF} Z_{i-j}}{h\beta_{-1,s}^{BDF}} - f(z, \theta)$$

Straightforward modifications to the proof of Theorem 4 show that the probabilistic backward difference formula is well-defined (the analysis is identical from equation (4.17) onward), and convergent by appeal to Theorem 3. In classical numerical analysis, BDF methods are often chosen for stiff problems, due to their enhanced stability properties over Adams-type methods [Cur52]. It may be that their probabilistic counterparts also find use in this context.[48]

More expansive generalisations would include methods which are simultaneously multistage and multistep. Theorem 2 covers the convergence of multistage one-step methods, though the assumption is made that perturbations ξ_i are only added to the final estimate for Z_{i+1}, after the complete calculation of all intermediate stages. Whether these intermediate stages can also be randomised, and whether this can be done in a multistep setting, are both open questions. (Whether they *should* be randomised is also not incontrovertible.) The lack of theoretical bounds for the error incurred by the intermediate stages of Runge–Kutta integrators may hamper this analysis to a degree.

In the case of multistage, multistep methods, there are also potential implications for the consistency of the statistical model, since the results of the computations at intermediate stages are discarded before advancing to the next iteration. If these

[48] We discuss the issue of stability further in Section 6.2.3.

stages are then recalculated during the next iteration *at the same time ordinates*, there is a potential conflict between the outcomes of these two calculations, since from a statistical viewpoint they could be considered to represent the same random variable.

To see this, consider an explicit 2-step method with one intermediate stage calculated at each iteration. In the process of determining Z_{i+1}, the variables F_i and F_{i+1} are conditioned upon. As in the 2-step Adams–Bashforth method, the values that these variables take are known once we are at time t_i, since they have already been calculated in previous iterations. The intermediate stage also requires the execution of additional model interrogations at fractional times, which we will term $F_{i+1/2}$ and $F_{i-1/2}$. These values are discarded after Z_{i+1} is determined. At the subsequent iteration, however, the same process (whose objective now is to determine Z_{i+2}) requires the re-interrogation of the model at time $t_{i+1/2}$. Since the calculation of this variable depends on different inputs to previously, in general a different outcome would be expected.

The statistical implications of conditioning on a random variable which has previously been discarded recalls the protracted discussion in Macdonald et al. [Mac15] (summarised in Section 4.1.1) about whether these two random variables should be considered as separate entities or, if not, the nature of the conditional relationship between them.

As a result of these and other issues, we anticipate a complete extension to general linear methods to be a potentially fraught endeavour. Despite this, we hope that future research is able to make progress in unraveling these issues, and further generalise the probabilistic paradigm to this wider class of algorithms.

Finally, we briefly note the ubiquity of variable step-size methods in classical numerical analysis and point out that no currently-existing probabilistic solver currently incorporates this arrangement. Recent work by Abdulle & Garegnani [Abd18], which randomises the step-size h in order to produce empirical uncertainty measures—akin to those output by the methods in this thesis—is the closest in character, though in that work the step-size is explicitly randomised rather than *controlled*. Control over step-size—based on feedback from the progress of the algorithm itself—is the approach typically favoured in the implementation of variable step-size methods since, when carefully implemented, significant improvements in accuracy and stability are possible. We suggest that the future integration of this type of algorithm into the probabilistic paradigm would hugely advance the general usefulness of the new class of integrators we have introduced.

6.2.2 *Other extensions*

At various points during the preparation of this thesis, several minor modifications/generalisations suggested themselves, which we were unable to investigate to a level that justified their inclusion in the main body of work. We collect these supplementary thoughts in this section.

Firstly, the transformation $g(u, \eta) = \exp(-u^2/2\eta^2)$ defined in equation (4.7) is required in order to turn the discrepancy $r(z)$ into a valid probability measure. (The fact that it does so is rigorously proved in part *(i)* of Theorem 4.) This particular form of g was chosen primarily because of its functional similarity to a the probability density function of a Gaussian distribution, meaning that the scaling matrix H, while not strictly-speaking a variance, can be thought of as 'almost' a variance. This in turn allowed us to employ the delta method in the design of our calibration procedure.

An obvious question to ask is if other transformations g are possible. There is no reason to believe *a priori* that the model for the step-forward distribution is more reasonable using this particular function g than any other—this would be just as arbitrary as assuming that Gaussian error is universally most appropriate, a notion we have criticised. Could there be any reason to believe that a Laplace-type transformation $g(u, \eta) = \exp(-|u/\eta|)$, or some other form entirely, is appropriate? It is certainly the case that convergence would have to be verified individually in each case, and an entirely different proof strategy would be required in seeking analogues to Theorem 4.

Secondly, we briefly remarked upon different designs of the calibration scheme in Remark 4.3. As described there, our experimentation with these alternative approaches led to problems with stability in some cases. Nevertheless, we feel that a disciplined investigation into this topic would be a fruitful extension to our work, particularly with a view to broadening its applicability to high-dimensional problems or to those with significantly variable length-scales across the range of integration.

A problem likely to be further out of reach is that of establishing rules of thumb for the calibration of probabilistic integrators, so that lengthy preliminary investigations of the type conducted in Section 5.2 are not required in advance of each application. If, for a given probabilistic integrator, some relationship could be deduced between the value of α^* and some other characteristics of the method (such as its step-size, or its error constant), and/or of the problem being solved (whether global, like its Lipschitz constant; or local, such as stepwise error estimates or some derived function of its Jacobian), this could obviate the need for lengthy preparatory runs. A comprehensive investigation into this issue would either require many more test models to be considered, or some theoretical insight about the general behaviour of these methods.

Lastly, we remark that some of the procedures described in this thesis may be amenable to parallelisation, with the goal being a substantial reduction in 'wall-clock' running time of the proposed algorithms. While iterative IVP solvers are inherently sequential, sampling of the non-parametric step-forward distribution need not be—for example if a rejection sampling approach is used (as posited in Section 4.4.1), samples could be generated in parallel. Furthermore, paralellisation of the entire forward solve—using different random seeds ω on each core—would be an effective way of creating an ensemble of (independent) randomised trajectories.

6.2.3 Stability

One recurring issue in the numerical solution of differential equations is that of the degree of stability of the chosen method of integration. The characterisation of stability is in general an extremely subtle issue—several lengthy studies exist considering this specific point [Hac14; Dah58]. A number of stability concepts with different definitions exist (see footnote 26) and these can be used to assess the properties of particular numerical methods and compare their suitability for a particular application. For example, the 'region of absolute stability' [Sül03, §12] of a linear multistep method is defined as a particular connected subset of the complex plane relating to the roots of a polynomial derived from the method's coefficients. However, such notions can only hope to capture some universal characteristic of the method since, by definition, they do not consider the specific application of the method under consideration.

The details of the definitions of these stability measures are not of primary concern for us, but the key point is that in a *practical* sense, whether or not a method is stable in a particular context depends heavily on the function $f(\cdot, \theta)$ defining the dynamics of the ODE being solved, the step-size h, the accuracy of the initialisation procedure, and more. It is intimately related to the concept of stiffness, several diverse definitions of which were described in Section 4.2.

With reference to our work, the introduction of stochasticity to numerical methods of this type is bound to affect these high-level properties. If the primary aim of deploying probabilistic methods is to prevent unjustified certainty in incorrect conclusions, then this issue could be said to have limited consequences—at the very worst, a randomised method suffering from poor stability will output even wider uncertainty bands that it otherwise would. However, if the main aim is to *model* the error of the underlying method probabilistically, a large difference in the qualitative behaviour of the corresponding probabilistic method has the potential to prevent this taking place at all.

In our simulations, we did occasionally come up against this issue. For higher order multistep methods—whose classical counterparts have smaller regions of absolute stability—it was sometimes the case that integrators calibrated by the scale-matching

method described in Section 4.3 did not converge properly when applied to a particular problem, with a particular choice of step-size, for which the unperturbed method performed as intended. It is not clear whether a more complex perturbation procedure (which could take account of the variation in error scale across the integration range or across dimension; see remark 4.3) would remedy this—this would also be an interesting avenue for further research.

A comprehensive, mathematically rigorous, analysis of the stability properties of randomised algorithms of the type discussed here does not yet exist. Some work in this direction has been started, for example by Lie et al. [Lie18]. Our instinct—and the evidence of that work—suggests that establishing results of this type is a highly non-trivial endeavour. Nevertheless we hope that the strong statistical motivation for the methods considered in this thesis, and their position within a growing framework of ever more effective probabilistic algorithms for numerical tasks, will provide the impetus for continuing research in the future.

*

BIBLIOGRAPHY

[Abd18] A. Abdulle and G. Garegnani. Random time step probabilistic methods for uncertainty quantification in chaotic and geometric numerical integration. *arXiv:1801.01340*, 2018.

[Abr65] M. Abramowitz and I. Stegun. *Handbook of Mathematical Functions: With Formulas, Graphs, and Mathematical Tables*. Courier Corporation, 1965.

[Ada83] J. Adams. *With an Explanation of the Method of Integration Employed in Constructing the Tables Which Give the Theoretical Form of Such Drops*. In F. Bashforth. *An Attempt to Test the Theories of Capillary Action by Comparing the Theoretical and Measured Forms of Drops of Fluid*. Cambridge University Press, 1883.

[Adl81] R. Adler. *The Geometry of Random Fields*. SIAM Classics in Applied Mathematics 65, 1981.

[And09] C. Andrieu and G. Roberts. The pseudo-marginal approach for efficient Monte Carlo computations. *Annals of Statistics*, 37(2):697–725, 2009.

[Arn13] A. Arnold, D. Calvetti and E. Somersalo. Linear multistep methods, particle filtering and sequential Monte Carlo. *Inverse problems*, 29(8):085007, 2013.

[Arn92] V. Arnol'd. *Ordinary Differential Equations*. Springer, 3rd edition, 1992.

[Ast11] R. Aster, B. Borchers and C. Thurber. *Parameter Estimation and Inverse Problems*. Academic Press, 2011.

[Atk09] K. Atkinson, W. Han and D. Stewart. *Numerical Solution of Ordinary Differential Equations*. John Wiley & Sons, 2009.

[Bar14] D. Barber and Y. Wang. Gaussian processes for Bayesian estimation in ordinary differential equations. *Proceedings of the 31st International Conference on Machine Learning*, PMLR 32(2):1485–1493, 2014.

[Bea10] N. Beauman. *Boxer, Beetle*. Sceptre, 2010.

[Bes08] A. Beskos, G. Roberts, A. Stuart and J. Voss. MCMC methods for diffusion bridges. *Stochastics and Dynamics*, 8(03):319–350, 2008.

[Bha46] A. Bhattacharyya. On a measure of divergence between two multinomial populations. *Sankhya: The Indian Journal of Statistics*, 7(4):401–406, 1946.

[Bie86] L. Biegler, J. Damiano and G. Blau. Nonlinear parameter estimation: a case study comparison. *AIChE Journal*, 32(1):29–45, 1986.

[Blu02] G. Bluman and S. Anco. *Symmetry and Integration Methods for Differential Equations*. Springer, 2002.

[Bra11] F. Brauer and C. Castillo-Chavez. *Mathematical Models in Population Biology and Epidemiology*. Springer, 2nd edition, 2011.

[Bri04] R. Bringhurst. *The Elements of Typographic Style*. Hartley & Marks, Publishers, 3rd edition, 2004.

[Buc06] E. Buckwar and R. Winkler. Multistep methods for SDEs and their application to problems with small noise. *SIAM Journal of Numerical Analysis*, 44(2):779–803, 2006.

[Bur95] K. Burrage. *Parallel and Sequential Methods for Ordinary Differential Equations*. Clarendon Press, 1995.

[But06] J. Butcher. General Linear Methods. *Acta Numerica*, 15:157–256, 2006.

[But08] J. Butcher. *Numerical Methods for Ordinary Differential Equations*. John Wiley & Sons, 2nd edition, 2008.

[But96] J. Butcher and G. Wanner. Runge–Kutta methods: some historical notes. *Applied Numerical Mathematics*, 22(1):113–151, 1996.

[Caf98] R. Caflisch. Monte Carlo and quasi-Monte Carlo methods. *Acta Numerica*, 7:1–49, 1998.

[Cal09] B. Calderhead, M. Girolami and N. Lawrence. Accelerating Bayesian inference over nonlinear differential equations with Gaussian processes. *Advances in Neural Information Processing Systems*, 21:217–224, 2009.

[Cam07] D. Campbell. Bayesian collocation tempering and generalized profiling for estimation of parameters from differential equation models. PhD thesis. McGill University, 2007.

[Cam12] D. Campbell and R. Steele. Smooth functional tempering for nonlinear differential equation models. *Statistics and Computing*, 22(2):429–443, 2012.

[Cao04] Y. Cao and L. Petzold. A posteriori error estimation and global error control for ordinary differential equations by the adjoint method. *SIAM Journal on Scientific Computing*, 26(2):359–374, 2004.

[Cap16] M. Capistrán, J. Christen and S. Donnet. Bayesian analysis of ODEs: solver optimal accuracy and Bayes factors. *SIAM/ASA Journal on Uncertainty Quantification*, 4(1):829–849, 2016.

[Chk13] O. Chkrebtii. Probabilistic solution of differential equations for Bayesian uncertainty quantification and inference. PhD thesis. Simon Fraser University, 2013.

[Chk16] O. Chkrebtii, D. Campbell, B. Calderhead and M. Girolami. Bayesian solution uncertainty quantification for differential equations. *Bayesian Analysis*, 11(4):1239–1267, 2016.

[Cle57] C. Clenshaw. The numerical solution of linear differential equations in Chebyshev series. *Mathematical Proceedings of the Cambridge Philosophical Society*, 53(1):134–149, 1957.

[Coc17] J. Cockayne, C. Oates, T. Sullivan and M. Girolami. Bayesian probabilistic numerical methods. *arXiv:1702.03673*, 2017.

[Cod55] E. Coddington and N. Levinson. *Theory of Ordinary Differential Equations*. McGraw–Hill, 1955.

[Con16] P. Conrad, M. Girolami, S. Särkkä, A. Stuart and K. Zygalakis. Statistical analysis of differential equations: introducing probability measures on numerical solutions. *Statistics and Computing*, 27(4):1065– 1082, 2016.

[Cot13] S. Cotter, G. Roberts, A. Stuart and D. White. MCMC methods for functions: modifying old algorithms to make them faster. *Statistical Science*, 28(3):424–446, 2013.

[Cox06] D. Cox. *Principles of Statistical Inference*. Cambridge University Press, 2006.

[Cur52] C. Curtiss and J. Hirschfelder. Integration of Stiff Equations. *Proceedings of the National Academy of Sciences of the United States of America*, 38(3):235–243, 1952.

[Dah58] G. Dahlquist. Stability and error bounds in the numerical integration of ordinary differential equations. PhD Thesis. Stockholm University, 1958.

[Deu02] P. Deuflhard and F. Bornemann. *Scientific Computing with Ordinary Differential Equations*. Springer, 2002.

[Dia88] P. Diaconis. Bayesian numerical analysis. *Statistical Decision Theory and Related Topics IV*, 1:163–175, 1988.

[Don07] S. Donnet and A. Samson. Estimation of parameters in incomplete data models defined by dynamical systems. *Journal of Statistical Planning and Inference*, 137(9):2815–2831, 2007.

[Don13] F. Dondelinger, S. Rogers and D. Husmeier. ODE parameter inference using adaptive gradient matching with Gaussian processes. *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics*, 31:216–228, 2013.

[Est00] D. Estep, M. Larson and R. Williams. Estimating the error of numerical solutions of systems of reaction-diffusion equations. *Memoirs of the American Mathematical Society*, (696):1–109, 2000.

[Est95] D. Estep. A posteriori error bounds and global error control for approximation of ordinary differential equations. *SIAM Journal on Numerical Analysis*, 32(1):1–48, 1995.

[Fit61] R. FitzHugh. Impulses and physiological states in theoretical models of nerve membrane. *Biophysical Journal*, 1(6):445–466, 1961.

[Fre80] H. Freedman. *Deterministic Mathematical Models in Population Ecology*. M. Dekker, 1980.

[Gea71] C. Gear. *Numerical Initial Value Problems in Ordinary Differential Equations*. Prentice-Hall, 1971.

[Gea81] C. Gear. Numerical solution of ordinary differential equations: is there anything left to do? *SIAM Review*, 23(1):10–24, 1981.

[Gel08] A. Gelman. Objections to Bayesian statistics. *Bayesian Analysis*, 3(3):445–449, 2008.

[Gel13] A. Gelman, J. Carlin, H. Stern, D. Dunson, A. Vehtari and D. Rubin. *Bayesian Data Analysis*. CRC Press, 3rd edition, 2013.

[Gel96] A. Gelman, F. Bois and J. Jiang. Physiological pharmacokinetic analysis using population modeling and informative prior distributions. *Journal of the American Statistical Association*, 91(436):1400–1412, 1996.

[Gem84] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984.

[Gey92] C. Geyer. Practical Markov chain Monte Carlo. *Statistical Science*, 7(4):473–483, 1992.

[Gir11] M. Girolami and B. Calderhead. Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of the Royal Statistical Society: Series B*, 73(2):123–214, 2011.

[Gra03] T. Graepel. Solving noisy linear operator equations by Gaussian processes: application to ordinary and partial differential equations. *Proceedings of the Twentieth International Conference on Machine Learning*:234–241, 2003.

[Gre12] W. Greene. *Econometric Analysis*. Pearson/Prentice Hall, 7th edition, 2012.

[Gri01] G. Grimmett and D. Stirzaker. *Probability and Random Processes*. Oxford University Press, 3rd edition, 2001.

[Gri10] D. Griffiths and D. Higham. *Numerical Methods for Ordinary Differential Equations: Initial Value Problems*. Springer, 2010.

[Haa01] H. Haario, E. Saksman and J. Tamminen. An adaptive Metropolis algorithm. *Bernoulli*, 7(2):223–242, 2001.

[Hac14] W. Hackbusch. *The Concept of Stability in Numerical Mathematics*. Springer, 2014.

[Had02] J. Hadamard. Sur les problèmes aux dérivés partielles et leur signification physique. *Princeton University Bulletin*, 13:49–52, 1902.

[Hai08] E. Hairer, S. Nørsett and G. Wanner. *Solving Ordinary Differential Equations I: Nonstiff Problems*. Springer, 2nd revised edition, 2008.

[Hai10] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*. Springer, 2nd revised edition, 2010.

[Has70] W. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[Hau15] S. Hauberg, M. Schober, M. Liptrot, P. Hennig and A. Feragen. A random Riemannian metric for probabilistic shortest-path tractography. *MICCAI 2015: Medical Image Computing and Computer-Assisted Intervention*, 1:597–604, 2015.

[Hen14] P. Hennig and S. Hauberg. Probabilistic solutions to differential equations and their application to Riemannian statistics. *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, PMLR 33:347–355, 2014.

[Hen15] P. Hennig, M. Osborne and M. Girolami. Probabilistic numerics and uncertainty in computations. *Proceedings of the Royal Society: Series A*, 471:20150142, 2015.

[Hen62] P. Henrici. *Discrete Variable Methods in Ordinary Differential Equations*. Wiley, 1962.

[Hin02] G. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, 2002.

[Hod52] A. Hodgkin and A. Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The journal of physiology*, 117(4):500–544, 1952.

[Hor12] R. Horn and C. Johnson. *Matrix Analysis*. Cambridge University Press, 2nd edition, 2012.

[Hul63] T. Hull and A. Creemer. Efficiency of predictor-corrector procedures. *Journal of the ACM*, 10(3):291–301, 1963.

[Hul66] T. Hull and J. Swenson. Tests of probabilistic models for propagation of roundoff errors. *Communications of the ACM*, 9(2):108–113, 1966.

[Ise09] A. Iserles. *A First Course in the Numerical Analysis of Differential Equations*. Cambridge University Press, 2nd edition, 2009.

[Jac09] Z. Jackiewicz. *General Linear Methods for Ordinary Differential Equations*. John Wiley & Sons, 2009.

[Jay03] E. Jaynes. *Probability Theory: The Logic of Science*. Cambridge University Press, 2003.

[Jen12] A. Jensen, S. Ditlevsen, M. Kessler and O. Papaspiliopoulos. Markov chain Monte Carlo approach to parameter estimation in the FitzHugh–Nagumo model. *Physical Review E*, 86(4):041114, 2012.

[Kai06] J. Kaipio and E. Somersalo. *Statistical and Computational Inverse Problems*. Springer, 2006.

[Kal60] R. Kalman. A new approach to linear filtering and prediction problems. *Journal of basic engineering*, 82(1):35–45, 1960.

[Kar91] I. Karatzas, S. Shreve, S. Shreve and S. Shreve. *Brownian Motion and Stochastic Calculus*. Springer, 1991.

[Ken01] M. Kennedy and A. O'Hagan. Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B*, 63(3):425–464, 2001.

[Ker16] H. Kersting and P. Hennig. Active uncertainty calibration in Bayesian ODE solvers. *Proceedings of the 32nd Conference on Uncertainty in Artificial Intelligence*:309–318, 2016.

[Ker18] H. Kersting, T. Sullivan and P. Hennig. Convergence rates of Gaussian ODE filters. *arXiv:1807.09737*, 2018.

[Kol09] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

[Lam91] J. Lambert. *Numerical Methods for Ordinary Differential Systems: The Initial Value Problem*. John Wiley and Sons, 1991.

[Lar72] F. Larkin. Gaussian measure in Hilbert space and applications in numerical analysis. *Rocky Mountain Journal of Mathematics*, 2(3):379–422, 1972.

[Law15] K. Law, A. Stuart and K. Zygalakis. *Data Assimilation: A Mathematical Introduction*. Springer, 2015.

[Lef71] R. Lefever and G. Nicolis. Chemical instabilities and sustained oscillations. *Journal of Theoretical Biology*, 30(2):267–284, 1971.

[LeV07] R. LeVeque. *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems*. SIAM, 2007.

[Lie17] H. Lie, A. Stuart and T. Sullivan. Strong convergence rates of probabilistic integrators for ordinary differential equations. *arXiv:1703.03680*, 2017.

[Lie18] H. Lie, T. Sullivan and A. Teckentrup. Random forward models and log-likelihoods in Bayesian inverse problems. *SIAM/ASA Journal on Uncertainty Quantification*, 6(4):1600–1629, 2018.

[Liu01] J. Liu. *Monte Carlo Strategies in Scientific Computing*. Springer, 2001.

[Mac15] B. Macdonald, C. Higham and D. Husmeier. Controversy in mechanistic modelling with Gaussian processes. *Proceedings of The 32nd International Conference on Machine Learning*, PMLR 37:1539–1547, 2015.

[Mag17] E. Magnani, H. Kersting, M. Schober and P. Hennig. Bayesian filtering for ODEs with bounded derivatives. *arXiv:1709.08471*, 2017.

[Mar00] T. Marlin. *Process Control: Designing Processes and Control Systems for Dynamic Performance*. McGraw–Hill, 2nd edition, 2000.

[Met53] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller and E. Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953.

[Mou26] F. Moulton. *New Methods in Exterior Ballistics*. University of Chicago Press, 1926.

[Nag62] J. Nagumo, S. Arimoto and S. Yoshizawa. An active pulse transmission line simulating nerve axon. *Proceedings of the IRE*, 50(10):2061–2070, 1962.

[Oeh92] G. Oehlert. A Note on the Delta Method. *The American Statistician*, 46(1):27–29, 1992.

[OHa92] A. O'Hagan. Some Bayesian numerical analysis. *Bayesian Statistics 4: Proceedings of the Fourth Valencia International Meeting*, 4:345–363, 1992.

[Øks98] B. Øksendal. *Stochastic Differential Equations: An Introduction with Applications*. Springer–Verlag, 5th edition, 1998.

[Pal09] R. Palais and R. Palais. *Differential Equations, Mechanics, and Computation*. American Mathematical Society, 2009.

[Pea85] J. Pearl. Bayesian networks: a model of self-activated memory for evidential reasoning. *Proceedings of the 7th Conference of the Cognitive Science Society*:329–334, 1985.

[Pil11] N. Pillai, A. Stuart and A. Thiery. Optimal proposal design for random walk type Metropolis algorithms with Gaussian random field priors. *arXiv:1108.1494*, 2011.

[Pre07] W. Press, S. Teukolsky, W. Vetterling and B. Flannery. *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, 3rd edition, 2007.

[Qua00] A. Quarteroni, R. Sacco and F. Saleri. *Numerical Mathematics*. Springer, 2000.

[Ram07] J. Ramsay, G. Hooker, D. Campbell and J. Cao. Parameter estimation for differential equations: a generalized smoothing approach. *Journal of the Royal Statistical Society: Series B*, 69(5):741–796, 2007.

[Ran16] S. Ranciati, C. Viroli and E. Wit. Bayesian smooth-and-match strategy for ordinary differential equations models that are linear in the parameters. *arXiv:1604.02318*, 2016.

[Ras06] C. Rasmussen and C. Williams. *Gaussian Processes for Machine Learning*. University Press Group Limited, 2006.

[Rob02] G. Roberts and O. Stramer. Langevin diffusions and Metropolis–Hastings algorithms. *Methodology And Computing In Applied Probability*, 4(4):337–357, 2002.

[Rob04] C. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer, New York, 2nd edition, 2004.

[Rob07] C. Robert. *The Bayesian Choice: From Decision-Theoretic Foundations to Computational Implementation*. Springer, 2nd edition, 2007.

[Rob56] H. Robbins. An Empirical Bayes Approach to Statistics. *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability*, 1:157–163, 1956.

[Rob97] G. Roberts, A. Gelman and W. Gilks. Weak convergence and optimal scaling of random walk Metropolis algorithms. *The Annals of Applied Probability*, 7(1):110–120, 1997.

[Roc12] C. Rocsoreanu, A. Georgescu and N. Giurgiteanu. *The FitzHugh-Nagumo Model: Bifurcation and Dynamics*. Springer, 2012.

[Sär06] S. Särkkä. Recursive Bayesian inference on stochastic differential equations. PhD Thesis. Helsinki University of Technology, 2006.

[Sär13] S. Särkkä. *Bayesian Filtering and Smoothing*. Cambridge University Press, 2013.

[Sch14] M. Schober, D. Duvenaud and P. Hennig. Probabilistic ODE Solvers with Runge-Kutta means. *Advances in Neural Information Processing Systems*, 27:739–747, 2014.

[Sch17] J. Schott. *Matrix Analysis for Statistics*. John Wiley & Sons, 3rd edition, 2017.

[Sch18] M. Schober, S. Särkkä and P. Hennig. A probabilistic model for the numerical solution of initial value problems. *Statistics and Computing*:1–24, 2018.

[Sha94] L. Shampine. *Numerical Solution of Ordinary Differential Equations*. CRC Press, 1994.

[Sis18] S. Sisson, Y. Fan and M. Beaumont. *Handbook of Approximate Bayesian Computation*. CRC Press, 2018.

[Ski91] J. Skilling. Bayesian solution of ordinary differential equations. In C. Smith, G. Erickson and P. Neudorfer, editors, *Maximum Entropy and Bayesian Methods*, pp. 23–37. Springer, 1991.

[Smi13] R. Smith. *Uncertainty Quantification: Theory, Implementation, and Applications*. SIAM, 2013.

[Sol03] E. Solak, R. Murray-Smith, E. Solak, W. Leithead, C. Rasmussen and D. Leith. Derivative observations in Gaussian Process models of dynamic systems. *Advances in Neural Information Processing Systems*, 16:1057–1064, 2003.

[Stu10] A. Stuart. Inverse problems: a Bayesian perspective. *Acta Numerica*, 19:451–559, 2010.

[Stu98] A. Stuart and A. Humphries. *Dynamical Systems and Numerical Analysis*. Cambridge University Press, 1998.

[Sül03] E. Süli and D. Mayers. *An Introduction to Numerical Analysis*. Cambridge University Press, 2003.

[Sul15] T. Sullivan. *Introduction to Uncertainty Quantification*. Springer, 2015.

[Tan07] C. Tannert, H. Elvers and B. Jandrig. The ethics of uncertainty. In the light of possible dangers, research becomes a moral duty. *EMBO Reports*, 8(10):892–896, 2007.

[Tar05] A. Tarantola. *Inverse Problem Theory and Methods for Model Parameter Estimation*. SIAM, 2005.

[Tey14] O. Teymur. Efficient MCMC sampling using Gaussian approximations. Masters thesis. Imperial College London, 2014.

[Tey16] O. Teymur, K. Zygalakis and B. Calderhead. Probabilistic linear multistep methods. *Advances in Neural Information Processing Systems*, 29:4314–4321, 2016.

[Tey18] O. Teymur, H. Lie, T. Sullivan and B. Calderhead. Implicit probabilistic integrators for ODEs. *Advances in Neural Information Processing Systems*, 2018. To appear.

[Tik95] A. Tikhonov, A. Goncharsky, V. Stepanov and A. Yagola. *Numerical Methods for the Solution of Ill-Posed Problems*. Springer, 1995.

[Tro18] F. Tronarp, H. Kersting, S. Särkkä and P. Hennig. Probabilistic solutions to ordinary differential equations as non-linear Bayesian filtering: a new perspective. *arXiv:1810.03440*, 2018.

[Tuf01] E. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, 2nd edition, 2001.

[Ugg08] E. Uggedal. Social navigation on the social web: unobtrusive prototyping of activity streams in established spaces. Masters Thesis. University of Oslo, 2008.

[Var82] J. Varah. A spline least squares method for numerical parameter estimation in differential equations. *SIAM Journal on Scientific and Statistical Computing*, 3(1):28–46, 1982.

[Von51] J. Von Neumann. Various techniques used in connection with random digits. *Journal of Research of the National Bureau of Standards*. Applied Math Series 3:36–38, 1951.

[Wan18] J. Wang, J. Cockayne and C. Oates. On the Bayesian solution of differential equations. *arXiv:1805.07109*, 2018.

[Xue10] H. Xue, H. Miao and H. Wu. Sieve estimation of constant and time-varying coefficients in nonlinear ordinary differential equation models by considering both numerical error and measurement error. *The annals of statistics*, 38(4):2351–2387, 2010.

[You05] G. Young and R. Smith. *Essentials of Statistical Inference*. Cambridge University Press, 2005.

The visual presentation of this thesis was inspired in part by the typographic recommendations of Robert Bringhurst [Bri04] and the graphical display guidelines of Edward Tufte [Tuf01]. The LaTeX style file is a heavily-modified version of a template by Eivind Uggedal [Ugg08].