École polytechnique fédérale de Lausanne

Chair of Computational Mathematics and Numerical Analysis

Master project

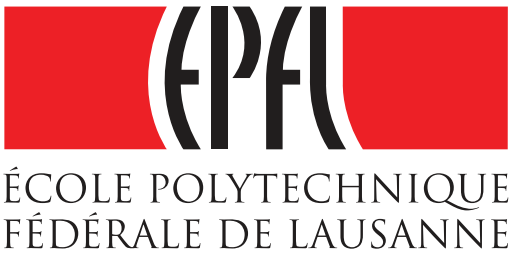
Master in Computational Science and Engineering

# Probabilistic solvers for Ordinary Differential Equations and Bayesian inference of parametrized models

*Author:*
Giacomo Garegnani

*Supervisor:*
Professor Assyr Abdulle

Lausanne, Autumn 2016



## Abstract

In this project we focus on the probabilistic interpretation of the numerical solution of ordinary differential equations. In particular, our main concern is investigating the properties of a probabilistic integrator which is a natural expansion of traditional deterministic methods. While strong and weak convergence of the obtained numerical solution have been studied, literature lacks a rigorous investigation of the properties of Monte Carlo estimators given by this novel numerical scheme. In this work, we prove results of convergence of Monte Carlo estimators in the mean square sense when the deterministic component of the probabilistic integrator is given by a Runge-Kutta method. Finally, we consider Bayesian inference inverse problems involving differential equations, studying the applications of the probabilistic integrator in Markov chain Monte Carlo algorithms.

## Acknowledgments

First and foremost, I would like to thank Professor Assyr Abdulle for his supervision and support during my Master Project, and all the members of ANMC for welcoming me in the group and giving me advice on my work. Moreover, I would like to thank my parents and family for their unconditional support during the last two years. Finally, I would like to thank all my friends at EPFL, in particular Edoardo, Filippo, Francesco, Luca and Philippe, who contributed to making these years in Lausanne an unforgettable experience.

Lausanne, January 2017

# Contents

# 1 Introduction

In recent years, probabilistic numerics has been a growingly relevant field within numerical mathematics. Many classic problems of numerical analysis, such as the approximation of Ordinary and Partial Differential Equations, have been recently reinterpreted from a probabilistic standpoint. In this work, we consider methods to perform Bayesian inference in the frame of parametric Ordinary Differential Equations (ODE's), as well as techniques used to give a probabilistic interpretation of the numerical solution of a differential equation.

Firstly, we present a survey of some of the existing techniques to perform Bayesian inference. In particular, we focus on the Markov Chain Monte Carlo methods (MCMC), a class of algorithms whose variants allow to perform inference in a wide range of different contexts.

Secondly, we consider a probabilistic numerical method for ODE's which has been recently introduced in [3]. In this work, we analyze its properties, presenting results of convergence showed in [3], as well as considering the behavior of Monte Carlo estimators produced by the realizations of the solution. In particular, we show how the Monte Carlo approximation of the probability measure introduced by the probabilistic solver collapses to the punctual true solution of the ODE with respect to the time step chosen for the numerical integration. This aspect has only been partially analyzed before, and clarifying this convergence property theoretically and with numerical experiments is the main contribution of this work.

Finally, we show how it is possible to integrate the probabilistic solver of ODE's in a MCMC algorithm. In [3], the authors show that using the probabilistic numerical method the estimation of the posterior distribution reflects better the uncertainty introduced by the numerical error. On the other hand, it is unclear whether the chosen number of trajectories influences the approximation of the posterior. Following the results on Monte Carlo estimations, we derive convergence rates for the posterior distribution, as well as for the Monte Carlo approximation of the parameter values obtained with MCMC.

The structure of this work is the following. In Section 2 we introduce basic notions of Bayesian inference and focus on the implementation of MCMC algorithms, taking into account some of the issues that could arise from the considered inferential problem. In Section 3, we consider the probabilistic solver of ODE's and present its properties. Finally, we show in Section 4 how to integrate the probabilistic solver in an inferential context, thus drawing our final considerations in the conclusion.

# 2 Bayesian statistics and Markov chain Monte Carlo

In the following section we will briefly discuss the main features of a Bayesian statistical approach, comparing it with the classical inferential statistics. Then, we will present a class of techniques used to practically perform Bayesian inference, the Markov chain Monte Carlo (commonly denoted with the acronym MCMC), discussing some of the possible implementations and properties of these methods.

## 2.1 Bayes' formula

The basis of Bayesian statistics can be found in the simple Bayes' formula. Let us consider an event space $\Omega$, a sigma-algebra $\mathcal{A}$, a probability measure $P$ and the probability space $(\Omega, \mathcal{A}, P)$. If $A$ and $B$ are two events in $\Omega$, the probability of the intersection of $A$ and $B$ is given by

$$\begin{aligned} P(A, B) &= P(A|B)P(B) \\ &= P(B|A)P(A). \end{aligned}$$

The equivalence between the two formulations leads to Bayes' formula

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}. \tag{1}$$

In Bayes' formula, $P(A|B)$ is a probability distribution, therefore its integral has to be equal to one. Therefore, one can rewrite Bayes' rule disregarding the value of $P(B)$ as

$$P(A|B) \propto P(A)P(B|A).$$

The exact value of the probability distribution can be therefore obtained by normalization as

$$P(A|B) = \frac{P(A)P(B|A)}{\int_\Omega P(A)P(B|A)}.$$

The quantities appearing in (1) are commonly referred to as

- posterior distribution $P(A|B)$,
- prior distribution $P(A)$,
- likelihood $P(B|A)$.

The probability distribution of $A$ is often the object of Bayesian inference and the event $B$ is an observable quantity related to $A$. Then the likelihood $P(B|A)$ is not a probability distribution but the likelihood the observations of $B$ have with respect to $A$. Therefore, in order to avoid misinterpretations, we will denote in the following the likelihood by $\mathcal{L}(B|A)$. Moreover, we will adopt in the following sections the notation $\mathcal{Q}$ for the prior and $\pi$ for the posterior distributions, thus obtaining

$$\pi(A|B) \propto \mathcal{Q}(A)\mathcal{L}(B|A).$$

## 2.2 Parametrized models

Bayes' formula opens a new perspective to statistical modeling with respect to the classical inferential standards. In particular, parametrized models are particularly suited to a Bayesian approach. Let us consider a parametrized model for predicting the outcome of an experiment. Let us denote by $\theta$ the parameter driving the experiment and by $X$ a random variable representing its outcome. Let us consider for simplicity $\theta$ as a vector of $\mathbb{R}^{N_p}$, where $N_p$ is the dimension of the parameter space. We will then denote by $\theta_i$ the $i$-th component of the parameter, with $i = 1, \dots, N_p$. For instance, we could consider the toss of a coin and estimate its probability to fall on one of the two side as $\theta$, or more complicated physical models influenced by an intractable source of noise.

In the classical statistic approach we would state an hypothetical distribution for $X$ depending on the parameter (e.g., $X \sim \mathcal{N}(\theta_1, \theta_2)$). Then, let us suppose that a set of observation $\mathcal{Y}_i = \{y_0, y_1, \dots, y_i\}$, $i = 1, \dots, N_d$, of the outcome of the experiment is available. We can consider these observations to be produced by a random variable $Y$ representing a quantity connected to the random variable $X$ by a law, that we denote by $f$, and biased by a measurement error, that we denote by $\varepsilon$. For instance, we could consider the following additive observational model

$$Y \sim f(X) + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \Gamma). \tag{2}$$

Given this background, there are many techniques available in the classical approach to compute an estimator $\hat{\theta}$ of the true parameter and to state a measure of the uncertainty the statistical model has on the estimator. In particular, one can compute analytically quantities such as the mean square error (MSE) of the estimator or give confidence intervals on $\theta$ such that its true value falls in the interval up to a threshold probability.

In the Bayesian frame the estimation of $\theta$ follows a completely different philosophy. The outcome of the Bayesian inference is neither a value of the parameter nor a set of values in which it is likely to be included, but it is a *probability distribution*. Maintaining the notation introduced above, Bayes' formula in this frame reads

$$\pi(\theta|\mathcal{Y}_i) \propto \mathcal{Q}(\theta)\mathcal{L}(\mathcal{Y}_i|\theta).$$

Let us analyze separately the two terms of this equation.

- Given a set of observations $y_i$, $i = 1, \dots, N_d$, and an observational model the likelihood $\mathcal{L}(\mathcal{Y}|\theta)$ can be evaluated. For example, in the Gaussian case introduced in (2) analytical formulas for the likelihood are available.
- The prior distribution $\mathcal{Q}(\theta)$ has to be established before the observation are obtained. This is a crucial part of the process of Bayesian inference, since in practice if the prior distribution is wrong or inadmissible, the obtained posterior may be negatively affected by this choice.

The two approaches give both equally valid results but in a completely different spirit. While in classical statistics the model driving an experiment is predetermined and its parameters are computed using observations, in the Bayesian frame the object of study is the model behind the parameter itself, which is revealed by the observations.

### 2.2.1 An example: parametrized differential equations

In this paragraph we present a simple example that is useful to understand Bayesian inference of parameters in general and the scope of this work in particular. Let us consider the probability space $(\Omega, \mathcal{F}, P)$, a one-dimensional standard Wiener process $\{W(t)\}_{t\geq 0}$ and a filtration $\{\mathcal{F}(t)\}_{t\geq 0}$ such that $W(t)$ is $\mathcal{F}(t)$ measurable. Moreover, let us consider the following one-dimensional stochastic differential equation (SDE)

$$\begin{aligned} dX(t) &= \lambda X(t)\mathrm{d}t + \mu X(t)dW(t), \quad 0 < t < T, \\ X(0) &= X_0, \qquad\qquad\qquad\qquad X_0 \in \mathbb{R}, \end{aligned} \tag{3}$$

where $\lambda, \mu$ are real parameters and $X_0$ is a random variable. It is known that under the hypotheses of Itô calculus the solution of (3) is given by

$$X(t) = X_0 \exp\left(\left(\lambda - \frac{1}{2}\mu^2\right)t + \mu W(t)\right),$$

which is a stochastic process often referred to as *geometric Brownian motion*. This equation and its solution have extensively been studied in numerous applications. For example, it used as a simple financial tool in order to model option or stock pricing, with the parameter $\lambda$ which is often referred to as the *drift* and the diffusion coefficient $\mu$ as the *volatility*. Given the model described by (3), we may be interested in inferring the value of one, or more, of its parameters.

Let us consider the following assumptions

- $X_0$ is a known real value,
- the drift coefficient $\lambda$ is known a priori,
- the diffusion coefficient $\mu$ is unknown but a prior distribution $\mathcal{Q}(\mu)$ has been stated,
- the value of the solution $X(t)$ is observable at a set of times $t_i$, $i = 1, \ldots, N_d$, such that $t_{N_d} = T$, with a zero-mean additive Gaussian measurement noise $\varepsilon$, i.e., the observations $y_i$, are given by

  $$y_i = x(t_i) + \varepsilon_i, \quad \varepsilon_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2), \quad \sigma \in \mathbb{R}, \quad i = 1, \ldots, N_d,$$

  where we denote by $x(t_i)$ a realization of $X$ evaluated at time $t_i$.

Let us denote as $\mathcal{Y}_i$ the set of all the observations $y_i$ up to time $t_i$. We are interested in estimating the value of the parameter $\mu$. In a Bayesian frame, this corresponds to providing a distribution $\pi$ conditional to the observation following Bayes' rule, i.e.,

$$\pi(\mu|\mathcal{Y}_i) \propto \mathcal{Q}(\mu)\mathcal{L}(\mathcal{Y}_i|\mu).$$

In this simple frame, the knowledge of the analytical form of the solution and of the measurement error gives us an exact notion of the model connecting the parameter and the observations. Hence, for each choice of the value of $\mu$ it is possible to evaluate the likelihood function $\mathcal{L}$ as follows

$$\mathcal{L}(\mathcal{Y}_{N_d}|\mu) = (2\pi\sigma^2)^{-N_d/2} \prod_{k=1}^{N_d} \mathbb{E}\left[\exp\left(-\frac{\sigma^2}{2}(X(t_k) - y_k)^2\right)\right],$$

where we omitted the implicit dependence of the process $X$ on $\mu$. Furthermore, if the prior distribution $\mathcal{Q}$ admits a density in closed form, it is possible to evaluate it on any choice of $\mu$. Therefore, it is possible to compute for each value of $\mu$ the value of the posterior distribution associated with the available set of measurements.

In this simple example the analytical form of any of the quantities of Bayes' formula and the small dimension of the parameter space imply that with a low effort it is possible to determine the value of the posterior distribution. In general this is not true, and as we will show in the next sections fine Monte Carlo techniques have been proposed to generate samples from any distribution.

## 2.3 Markov chain Monte Carlo methods

Markov chain Monte Carlo methods (MCMC) are a class of techniques used to perform Bayesian analyses [9, 14]. In the following we will present the main idea behind the method as well as some examples of their implementation.

Let us consider a model which has a random variable $X$ as its outcome parametrized by a parameter $\theta$ and a set of observations $\mathcal{Y}_i = \{y_1, y_2, \dots, y_i\}$, $i = 1, \dots, N_d$, providing information regarding $X$. Then, thanks to Bayes' rule, we can construct the posterior distribution of $\theta$ by Bayes' rule

$$\pi(\theta|\mathcal{Y}_i) \propto \mathcal{Q}(\theta)\mathcal{L}(\mathcal{Y}_i|\theta).$$

As in the previous paragraphs, let us assume that $\theta$ is a real-valued parameter of dimension $N_p$. If the parameter space has a high dimension, it is computationally expensive exploring all the possible values in order to build the posterior distribution, especially if evaluating the model connecting $\theta$ and the random variable $X$ is non-trivial. If we are interested in knowing the expectation of some measurable function $g\colon \mathbb{R}^{N_p} \to \mathbb{R}$ of $\theta$ we can proceed by the following Monte Carlo evaluation

$$\mathbb{E}\left[g(\theta)\right] = \int_{\mathbb{R}^{N_p}} g(\theta)\pi(\mathrm{d}\theta|\mathcal{Y}_i) \approx \frac{1}{N}\sum_{k=1}^{N} g(\theta^{(k)}), \tag{4}$$

where $\theta^{(k)}$, $k = 1, \dots, N$, is a set of realizations of $\theta$. While the equality in the equation follows from the definition of expectation, there is no guarantee that the Monte Carlo estimator will be a good approximation of the expectation regardless of the samples. MCMC techniques consist in generating samples such that the Monte Carlo approximation is valid without exploring the whole parameter space, which would lead to an unaffordable computational time on any modern computer. As the name of the methods suggests, given an initial guess $\theta^{(0)}$, MCMC builds a discrete Markov chain $\{\theta^{(i)}\}_{i\geq 0}$ such that the Monte Carlo approximation in (4) is valid. Formally, this is achieved considering a *transition kernel* $P$ which given the current element of the chain $\theta^{(i)}$ produces the next guess $\theta^{(i+1)}$. Under a set of assumptions on $P$ [14], we have the theoretical guarantee that the samples $\theta^{(i)}$ are drawn from the same *stationary distribution* for $i$ large enough. We can build many transition kernels having this property, and any valid choice of $P$ leads to a different MCMC method. In the following, we will present the widely-used *Metropolis-Hastings* algorithm, as well as two of its variants that were necessary for our work.

**Algorithm 1:** Metropolis-Hastings.

**Data**: $\theta^{(0)} \in \mathbb{R}^{N_p}, N \in \mathbb{N}_0$.

Compute $\pi(\theta_0)$ ;
**for** $i = 0, \dots, N$ **do**
  Draw $\vartheta$ from $q(\theta^{(i)}, \cdot)$ ;
  Compute the acceptance probability $\alpha(\theta^{(i)}, \vartheta)$ as in (5) ;
  Draw $u$ from $\mathcal{U}(0, 1)$ ;
  **if** $\alpha > u$ **then**
    Accept $\vartheta$, set $\theta_{i+1} = \vartheta$ ;
  **else**
    Set $\theta^{(i+1)} = \theta^{(i)}$;
  **end**
**end**

### 2.3.1 Metropolis-Hastings algorithm

In this paragraph we will introduce one of the most successful MCMC algorithms, the Metropolis-Hastings method (MH). In MH, the samples forming the Markov chain are generated following a *proposal distribution* $q$: $\mathbb{R}^{N_p} \times \mathbb{R}^{N_p} \to \mathbb{R}^+$ which satisfies the condition

$$\int_{\mathbb{R}^{N_p}} q(x, y)\mathrm{d}y = 1,$$

thus $q$ is a probability density function in its second argument. Given the current guess $\theta^{(i)}$, MH proposes the new element of the Markov Chain drawing a value $\vartheta$ from $q(\theta^{(i)}, \cdot)$. The new guess is not automatically accepted as the new element $\theta^{(i+1)}$ of the Markov chain, but it is accepted with a probability, that we denote by $\alpha(\theta^{(i)}, \theta^{(i+1)})$. Formally, the transition kernel $P_{\mathrm{MH}}$ representing the move made by MH from $\theta^{(i)}$ to $\theta^{(i+1)}$ is given by [17]

$$P_{\mathrm{MH}}(\theta^{(i)}, \theta^{(i+1)}) = \alpha(\theta^{(i)}, \theta^{(i+1)})q(\theta^{(i)}, \theta^{(i+1)}) + \delta_{\theta^{(i)}}(\theta^{(i+1)})\rho(\theta^{(i)}),$$

where $\delta_x$ is the Dirac delta centered in $x$ and $\rho$ is defined as

$$\rho(\theta^{(i)}) \coloneqq 1 - \int_{\mathbb{R}^{N_p}} \alpha(\theta^{(i)}, x)q(\theta^{(i)}, x)\mathrm{d}x.$$

In words, the expression of the transition kernel $P_{\mathrm{MH}}$ is equivalent to stating that the new guess $\vartheta$ generated from the proposal distribution is accepted with probability $\alpha$ and rejected with probability $1-\alpha$. Imposing that $P_{\mathrm{MH}}$ satisfies the hypotheses that guarantee the convergence of MCMC [14], we can get the expression of the acceptance probability in closed form as

$$\alpha(\theta^{(i)}, \vartheta) = \min\left\{\frac{\pi(\vartheta)q(\vartheta, \theta^{(i)})}{\pi(\theta^{(i)})q(\theta^{(i)}, \vartheta)}, 1\right\}. \tag{5}$$

As it is possible to remark from its pseudo-code, given in Algorithm 1, MH is extremely simple to implement on a computer in any programming language. In fact, the only choice left in practice is the proposal distribution $q(x, y)$. Unfortunately, this choice could impact negatively the behavior of MH, slowing dramatically its convergence towards the stationary distribution of the Markov chain. Let us first remark that if the proposal distribution is

**Algorithm 2:** Robust adaptive Metropolis.

**Data**: $\theta^{(0)} \in \mathbb{R}^{N_p}$, $N \in \mathbb{N}_0$, $S_0 \in \mathbb{R}^{N_p \times N_p}$, $\alpha^* \in (0,1)$.

Compute $\pi(\theta_0)$ ;
**for** $i = 0, \dots, N$ **do**
  Draw $z$ from $Z \sim \mathcal{N}(0, I)$ ;
  $\vartheta = \theta^{(i)} + S_i z$ ;
  Compute the acceptance probability $\alpha(\theta^{(i)}, \vartheta)$ as in (5) ;
  Draw $u$ from $\mathcal{U}(0,1)$ ;
  **if** $\alpha > u$ **then**
    Accept $\vartheta$, set $\theta_{i+1} = \vartheta$ ;
  **else**
    Set $\theta^{(i+1)} = \theta^{(i)}$;
  **end**
  Compute $S_{i+1}$ as in (8) ;
**end**

a symmetric function in its two arguments, i.e., $q(x,y) = q(y,x)$, the expression of the acceptance probability simplifies to

$$\alpha(\theta^{(i)}, \vartheta) = \min\left\{\frac{\pi(\vartheta)}{\pi(\theta^{(i)})}, 1\right\}. \tag{6}$$

For example, a Gaussian proposal distribution centered in $\theta^{(i)}$ with covariance matrix $\Sigma$ in $\mathbb{R}^{N_p \times N_p}$ is a common choice for $q(x,y)$. In this case, the proposal distribution is given up to a normalization constant by

$$q(x,y) \propto \exp\left(-\frac{1}{2}(x-y)^T \Sigma^{-1}(x-y)\right). \tag{7}$$

In this work, we mainly used a Gaussian proposal distribution, therefore the acceptance probability will be of the form (6).

Two main issues have to be taken into account before moving on to the practical applications of MH we considered for this work.

1. What is a good choice for the proposal function $q(x, \cdot)$?
2. How can we modify MH in case it is not possible, or not practical, to evaluate the posterior distribution $\pi(\theta)$?

In the following paragraphs we will present two approaches to modify MH targeting these two questions.

### 2.3.2 An adaptive approach

In the frame of MH algorithms, it is important to have a control on the *acceptance ratio*, i.e., the ratio of new proposed values $\vartheta$ that are included in the Markov chain $\{\theta^{(i)}\}_{i \geq 0}$. In the MH frame, the acceptance ratio is strongly related to the chosen proposal distribution, as if the new guess produced via the proposal distribution have a low probability of being

accepted, a low value of acceptance ratio will result from the algorithm. On the other hand, if the acceptance ratio is too high, the posterior distribution could be explored too slowly. Often, the optimal acceptance ratio for MH are considered to be 0.234 or 0.44 [20]. If the initial proposal distribution does not provide with acceptable values $\vartheta$, it may be necessary to tune it during the advancement of MH. An algorithm which targets this issue is the robust adaptive Metropolis (RAM) [20].

Let us consider the case of a Gaussian proposal distribution $q(x, y)$ as in (7). At the $n$-th step of MH the new guess $\vartheta$ of the parameter is given by

$$\vartheta = \theta^{(n)} + z, \quad Z \sim \mathcal{N}(0, \Sigma),$$

where $\Sigma$ is the covariance matrix. It is possible to build a sequence of matrices such that the convergence properties of MH are not spoiled and the acceptance ratio is asymptotically equal to a given value $\alpha^*$ [20]. This is obtained through the following update

$$\vartheta = \theta_k + S_n z_n, \quad Z_n \sim \mathcal{N}(0, I),$$

with $S_n$ a lower triangular positive definite matrix and $I$ the identity matrix. Given an initial choice $S_0$, the matrix $S_n$ is updated at each iteration with a lower triangular matrix $S_{n+1}$ satisfying

$$S_{n+1} S_{n+1}^T = S_n \left( I + \eta_n \left( \alpha(\theta^{(n)}, \vartheta) - \alpha^* \right) \frac{z_n z_n^T}{z_n^T z_n} \right) S_n^T. \tag{8}$$

Let us remark that if $S_n$ is symmetric definite positive, then the right hand side of the equality above is symmetric definite positive. Hence, we can compute $S_{n+1}$ with a Cholesky factorization. Let us remark that this update has to be performed at each iteration of RAM, both in case $\vartheta$ is accepted and rejected. The sequence $\{\eta_n\}_{n \geq 1}$ can be any sequence decaying to zero with $n$. In [20], the author suggests the choice

$$\eta_n = n^{-\gamma}, \quad 0.5 < \gamma \leq 1,$$

which guarantees convergence of RAM to the invariant distribution. Often the computational cost needed for the evaluation of the posterior distribution is high with respect to the dimension $N_p$ of the parameter space. Therefore, performing a Cholesky factorization at each iteration, which has a complexity of $\mathcal{O}(N_p^3)$, does not spoil the performances of RAM with respect to a standard MH. In Algorithm 2 we give the pseudo-code for the RAM update.

We now present a numerical example showing the optimal behavior of RAM. Let us consider a two-dimensional real random variable $X$ whose distribution admits the following density

$$\pi(X) \propto \exp\big(-10(X_1^2 - X_2)^2 - (X_1 - 0.25)^4\big), \tag{9}$$

where we denoted by $X_i$, $i = 1, 2$ the two components of $X$ and we omitted the normalization constant. This distribution is widely used to test MCMC algorithms [14]. We then consider a real value $\sigma$ in the set $\{0.01, 0.5, 2.0\}$ and target the distribution defined by (9) either using a standard MH with the proposal distribution given by a zero-centered normal distribution with covariance $\Sigma = \sigma^2 I$, or using RAM with the same choice of covariance structure as an initial guess and $\alpha^* = 0.4$. We run $N = 5000$ iterations of both algorithms and register all the guesses they produce as well as the final acceptance ratio. Results

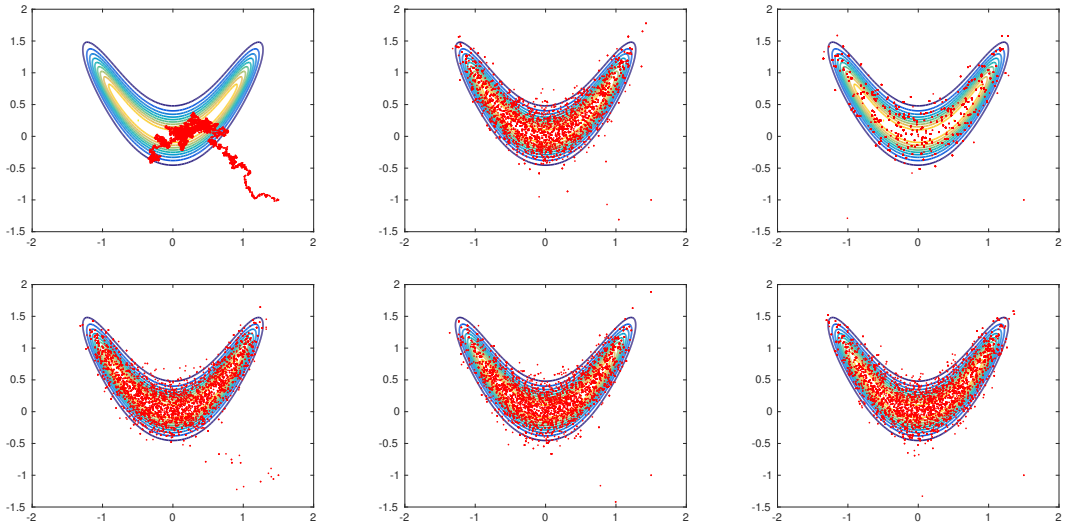


**Figure 1:** Samples produced by MH and RAM for the distribution (9). The contour lines of the density function are plotted for all the sets of results. In the first row we show the results obtained with MH for a normal update with covariance $\Sigma = \sigma^2 I$ with $\sigma = \{0.01, 0.5, 2.0\}$ from left to right. In the second row we show the results obtained with RAM with the same values of $\Sigma$ as an initial guess of the covariance structure.

(Figure 1) show that for the $\sigma = 0.01$ and $\sigma = 2.0$ standard MH fails to properly describe the posterior distribution, either accepting too many guesses and partially describing the posterior, or refusing almost all guesses therefore obtaining an insufficient number of samples. On the other hand, RAM adapts the step and for any choice of $\sigma$ the samples we obtain are equally good, with an acceptance ratio near to $\alpha^*$ (Table 1).

| MCMC | $\sigma = 0.01$ | $\sigma = 0.5$ | $\sigma = 2.0$ |
|---|---|---|---|
| MH | 0.96 | 0.35 | 0.06 |
| RAM | 0.43 | 0.40 | 0.38 |

**Table 1:** Acceptance ratios for MH and RAM with posterior distribution (9)

.

### 2.3.3 Pseudo-marginal Metropolis-Hastings

In this paragraph we discuss the second issue presented above. Let us consider the case in which it is not possible to evaluate the posterior distribution $\pi(\theta)$, or it is too computational expensive. For instance, in the example we provided in Section 2.2.1 the analytical solution of the SDE is computable. If we have a general equation which does not admit a closed-form solution, it is not possible to evaluate the likelihood function. Therefore, the standard MH algorithm and its adaptive version RAM are not applicable.

An algorithm that has been proposed to overcome this issue is the so-called *pseudo-marginal* MCMC [4], which is also known as particle Markov chain Monte Carlo (PMCMC)

**Algorithm 3:** Monte Carlo within Metropolis.

**Data**: $\theta^{(0)} \in \mathbb{R}^{N_p}$, $N \in \mathbb{N}_0$.

Compute $\pi(\theta_0)$ ;
**for** $i = 0, \dots, N$ **do**
  Draw $\vartheta$ from $q(\theta^{(i)}, \cdot)$ ;
  Compute the estimators $\pi_M(\theta^{(i)}, \xi)$ and $\pi_M(\vartheta, \xi)$ as in (10) ;
  Compute the acceptance probability $\alpha_M(\theta^{(i)}, \vartheta)$ as in (11);
  Draw $u$ from $\mathcal{U}(0, 1)$ ;
  **if** $\alpha > u$ **then**
    Accept $\vartheta$, set $\theta_{i+1} = \vartheta$ ;
  **else**
    Set $\theta^{(i+1)} = \theta^{(i)}$;
  **end**
**end**

[1]. The main idea of the proposed pseudo-marginal algorithms is modifying the target of the algorithm to a distribution $\pi(\theta, \xi)$ that admits $\pi(\theta)$ as a marginal distribution and that is easier than $\pi(\theta)$ to evaluate. Then, we can compute an unbiased Monte Carlo approximation $\pi_M(\theta)$ of the marginal distribution as

$$\pi_M(\theta) = \frac{1}{M} \sum_{i=1}^{M} \pi(\theta, \xi^{(i)}), \tag{10}$$

where the values $\xi^{(i)}$ are realizations of the random variable $\xi$. The acceptance probability $\alpha_M$ has then the same form of $\alpha$ in the standard MH, with $\pi_M(\theta)$ instead of the true marginal distribution, i.e.,

$$\alpha_M(\theta^{(i)}, \vartheta) = \min \left\{ \frac{\pi_M(\vartheta) q(\vartheta, \theta^{(i)})}{\pi_M(\theta^{(i)}) q(\theta^{(i)}, \vartheta)}, 1 \right\}. \tag{11}$$

The pseudo-code of the resulting algorithm is shown in Algorithm 3. Let us remark that if the estimator $\pi_M(\theta^{(i)})$ at the $i$-th iteration of MCMC is computed at each iteration and not recycled from the previous iterations, the resulting algorithm is often referred to as Monte Carlo within Metropolis (MCWM) [2] or noisy pseudo-marginal Metropolis [17]. Even though recomputing the estimator may be computationally expensive, the resulting Markov chain has an higher acceptance ratio, i.e., it explores the relevant values of the parameter $\theta$ faster, therefore defining better the posterior distribution. The main issue that has been addressed by the research on this kind of pseudo-marginal algorithms is whether the invariant distribution of the Markov chain converges to the marginal posterior distribution of the random variable $\theta$. It has been shown [2, 17] that under appropriate assumptions the following properties are valid

1. the transition kernel $P_M$ given by (11) converges to an invariant distribution $\pi_M$ with respect to the number of iterations $N$ of MCMC if the number of Monte Carlo draws $M$ is large enough [2, Theorem 9],

2. the invariant distribution $\pi_M$ obtained with MCWM converges to the true marginal distribution $\pi$ if $M$ tends to infinity [17, Theorem 4.1],

3. under stronger assumptions, it is possible to obtain convergence rates of $\pi_M$ to $\pi$ with respect to $M$ [17, Theorem 4.2 and Proposition 4.1].

Let us consider the example provided in Section 2.2.1. If we choose an SDE which does not admit a closed-form solution, it is impossible to evaluate the posterior distribution, as the likelihood function does not admit an analytical expression. On the other hand, there exists a large variety of numerical methods [16] that we can apply together with a Monte Carlo approximation to compute an estimator of the likelihood, thus obtaining a value $\pi_M$ as in (10). Hence, while it is impossible in this case to get the exact value of the posterior distribution, we can approximate it through an auxiliary simulation. Therefore, it is possible to apply a MCWM algorithm and obtain an approximation of $\pi(\theta)$ in this case as well.

### 2.3.4 How to deal with inadmissible parameter values

Let us consider without loss of generality a one-dimensional real parameter $\theta$ that can assume values only on a subset of $\mathbb{R}$. For instance, let us consider as the parameter space the interval $I = [a, b]$. If a Gaussian proposal function $q(x, y)$ is adopted in the implementation of MH, the unboundedness of the support of the proposal distribution results in a new guess $\vartheta$ which takes values outside $I$ with a non-zero probability. In this case, we choose to adopt as proposal function a *truncated Gaussian distribution*. The new guess $\vartheta$ is generated by $q(\theta^{(i)}, \cdot)$, which is a truncated Gaussian distribution of mean $\theta^{(i)}$ and fixed variance $\sigma$. The analytical expression of $q$ in this case is given by

$$q(x, y; a, b, \sigma) = \frac{1}{\sigma} \frac{\varphi\left((y - x)/\sigma\right)}{\Phi\left((b - x)/\sigma\right) - \Phi\left((a - x)/\sigma\right)}, \tag{12}$$

where we explicitly added the dependence on $a$, $b$ and $\sigma$. In (12) the function $\varphi$ is defined as

$$\varphi(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right),$$

and $\Phi$ is the standard Gaussian cumulative distribution function, where we assume that if $b = \infty$ then $\Phi((b - x)/\sigma)$ equals one, and if $a = -\infty$ then $\Phi((a - x)/\sigma)$ equals zero. Let us remark that this proposal distribution is not symmetric, therefore $\alpha$ in MH has to take into account the ratio between the proposal distribution evaluated in the old and the new guesses of the parameter. Hence, in this case the acceptance probability is given by

$$\alpha(\theta^{(i)}, \vartheta) = \min\left\{ \frac{\pi(\vartheta)\left(\Phi\left((b - \theta^{(i)})/\sigma\right) - \Phi\left((a - \theta^{(i)})/\sigma\right)\right)}{\pi(\theta^{(i)})\left(\Phi\left((b - \vartheta)/\sigma\right) - \Phi\left((a - \vartheta)/\sigma\right)\right)}, 1 \right\}.$$

Let us consider the example of a non-negative random variable $\theta$. In this case, thanks to the symmetry properties of the function $\Phi$, the acceptance probability $\alpha$ simplifies to

$$\alpha(\theta^{(i)}, \vartheta) = \min\left\{ \frac{\pi(\vartheta)\Phi\left(\theta^{(i)}/\sigma\right)}{\pi(\theta^{(i)})\Phi\left(\vartheta/\sigma\right)}, 1 \right\}.$$

As far as the practical implementation is concerned, modern programming languages often provide with generators of pseudo-random Gaussian numbers. In order to obtain a truncated Gaussian distribution, a practical procedure could be generating random numbers until a number in the acceptable range is generated.

**Algorithm 4:** Monitoring convergence.

**Data**: $K \in \mathbb{N}$; $\theta^{(0),i} \in \mathbb{R}^{N_p}, i = 1, \dots, K$; $N_0 \in \mathbb{N}$; $\bar{\rho} > 1$.

count = 0 ;
**while** $\exists\, i \in \{1, 2, \dots, N_p\} : \rho_i < \bar{\rho}$ **do**
  Initialize $K$ parallel MCMC with starting value $\theta^{(\text{count}),i}$ for $i = 1, \dots, K$ ;
  Execute $N_0$ iterations of each MCMC algorithm ;
  Get the chains $\Theta_i$ for each MCMC and assemble the mixed chain $\Theta_{\text{mix}}$ ;
  **foreach** *component $j$ of $\theta$ in $\mathbb{R}^{N_p}$* **do**
    **foreach** *chain $\Theta_i$* **do**
      Compute the variance $V_i^j$ of the component $j$ in $\Theta_i$ ;
    **end**
    Compute the mean within chains $V_{\text{mean}}^j$ averaging the variances $V_i^j$ ;
    Compute the variance $V_{\text{mix}}^j$ of the component $j$ in $\Theta_{\text{mix}}$ ;
    Compute the potential scale reduction factor as in (13) ;
  **end**
  count = count + $N_0$ ;
**end**

### 2.3.5 Monitoring convergence

It is unclear from the discussion of all the variants of MCMC discussed above how to choose an optimal number of samples $N$. In other words, no convergence criteria for the Markov chain have been presented. An interesting approach in order to monitor the convergence of MCMC consists in mixing several Markov chains [6]. The main idea consists in starting $K$ parallel MCMC with different initial guesses and check the properties of the single chains with respect to the chain resulting from the mixing of all the single chains. Let us denote by $\Theta_i$, with $i = 1, \dots, K$ the single chains, and by $\Theta_{\text{mix}}$ the mixed chain, i.e.

$$\Theta_{\text{mix}} = \bigcup_{i=1}^{K} \Theta_i.$$

Let us consider the estimation of the distribution of a random variable $\theta$ with values in $\mathbb{R}^{N_p}$ with components $\theta_1, \dots, \theta_{N_p}$. Let us moreover assume that all the MCMC algorithms have reached the iteration $N_0$. Then, we compute separately the population variance within each chain for each component $j$ of the random variable and denote it by $V_i^j$, for $i = 1, \dots, K$ and $j = 1, \dots, N_p$, i.e.

$$V_i^j = \frac{1}{N_0} \sum_{k=0}^{N_0} \left( \theta_j - \frac{1}{N_0} \sum_{h=0}^{N_0} \theta_j \right)^2.$$

Then, we average these variances and denote the result as $V_{\text{mean}}^j$. Finally, we compute the population variance for each component of $\theta$ of the mixed chain $\Theta_{\text{mix}}$ and we denote it $V_{\text{mix}}^j$. We now define the *potential scale reduction factor* $\rho_j$ of the $j$-th component of $\theta$ as [6]

$$\rho_j \coloneqq \sqrt{V_{\text{mix}}^j / V_{\text{mean}}^j}. \tag{13}$$

Let us remark that $\rho_j$ is greater than one for any sample, and a value close to one implies that all the single chains have in averaged mixed as well as the mixed chain. We therefore check whether all the $\rho_j$ are smaller than a certain threshold $\bar{\rho}$ (e.g., $\bar{\rho} = 1.05$) every $N_0$ iterations, and stop the MCMC algorithms if the condition is verified (Algorithm 4), thus keeping as the output of the mixed chain $\Theta_{\text{mix}}$.

# 3 Probabilistic Methods for Ordinary Differential Equations

The numerical integration of Ordinary Differential Equations (ODE's) is a topic of numerical mathematics which has been deeply studied in the last decades. The main focus of researchers was providing and analyzing methods endowed with properties such as convergence, conservation of invariants and stability. One class of methods which has proved to be particularly successful are the Runge-Kutta methods, which can alternatively provide some of the properties above. It is well-known that these methods are based on the choice of a discretization grid identified by a time step, that we denote by $h$, where the numerical solution will be computed. The introduction of a temporal discretization results in an error, which will be proportional to the time step $h$ at a power $q$. The quantity $q$ is referred to as the *order* of the numerical method, and gives an indication of the convergence properties of the numerical solution towards the exact solution of the ODE.
Recently, researchers have been interested in providing probabilistic numerical solutions to ODE's [3], thus introducing probabilistic measures which describe the numerical solution instead of punctual solutions. In this way it is possible to provide a quantification of the numerical error introduced at final time, which will depend both on the chosen discretization step and on the properties of the considered equation.
Let us consider $f\colon \mathbb{R}^d \to \mathbb{R}^d$ and the following ODE

$$\begin{aligned} u'(t) &= f(u), \quad t \in (0,T], \\ u(0) &= u_0, \qquad u_0 \in \mathbb{R}^d. \end{aligned} \tag{14}$$

Given a time step $h > 0$, we now consider a discretization of the time interval $[0,T]$ defined as $t_k = kh$, with $k = 0,\dots,N$, and with $T = Nh$. Let us now consider the numerical solution numerical solution $U_k$ at time $t_k$ given by a Runge-Kutta method. We can write one step of the time integration as

$$\begin{aligned} U_{k+1} &= \Psi_h(U_k), \quad k = 0,\dots,N-1, \\ U_0 &= u_0, \end{aligned} \tag{15}$$

where $\Psi$ defines the flow map of the Runge-Kutta method. In the following section, we will briefly introduce Runge-Kutta methods and some of their properties, thus deriving the notation above. The probabilistic method we considered in this work, first introduced in [3], considers independent identically distributed (i.i.d.) random variables $\xi_k$, which will explicitly depend on the discretization step $h$, and add them at each step of the numerical integration, i.e.,

$$\begin{aligned} U_{k+1} &= \Psi_h(U_k) + \xi_k(h), \\ U_0 &= u_0. \end{aligned} \tag{16}$$

Hence, the probabilistic solution will depend on the realizations of the random variables $\xi_k$, thus providing a family of numerical solutions instead of a punctual solution as in deterministic numerical analysis theory.
In the following section, we will briefly present Runge-Kutta methods we considered further in this work, summarizing some properties defining these methods. Then, we will present a motivating example for the probabilistic solver we introduced above. Finally, we will present some properties of the method together with numerical experiments empirically confirming our theoretical investigations.

## 3.1 Deterministic methods

The deterministic numerical methods we consider in this work belong to the class of Runge-Kutta methods. The properties of Runge-Kutta methods are extensively treated, for example, in [10, 11]. In the following, we recall the definition of the method as well as some basic properties.
Let us give the definition of this class of numerical methods.

**Definition 3.1.** *Let us consider* (14)*, $s \in \mathbb{N}^*$ and real numbers $b_i$, for $i = 1, \dots, s$, and $a_{ij}$, for $i, j = 1, \dots, s$. Moreover, let us consider a time step $h > 0$ and a time discretization $t_k = kh$ for $k = 0, \dots, N$, with $t_N = T$. An $s$-stage Runge-Kutta method is given by*

$$\begin{aligned} U_0 &= u_0, \\ K_i &= f(U_k + h \textstyle\sum_{j=1}^{s} a_{ij} K_j), \quad i = 1, \dots, s, \\ U_{k+1} &= U_k + h \textstyle\sum_{i=1}^{s} b_i K_i, \qquad k = 0, \dots, N-1, \end{aligned}$$

*where $K_i$ are vectors of $\mathbb{R}^d$ for all $i = 1, \dots, s$.*

*Remark* 3.1. Runge-Kutta methods are usually defined for non-autonomous systems taking into account evaluation points $c_i$ for $i = 1, \dots, s$. In our case, we focus on autonomous systems of the form (14), therefore we do not include the evaluation points $c_i$ in the definition of Runge-Kutta methods. Let us remark that for the consistency of the numerical method it is required that $c_i = \sum_{j=1}^{s} a_{ij}$, therefore the coefficients $c_i$ are uniquely defined given the coefficients $a_{ij}$. Moreover, any non-autonomous system can be transformed in autonomous form via a straightforward change of variables.

Runge-Kutta methods are completely defined by their coefficients $a_{ij}, b_i$. Therefore, these coefficients together with the evaluation points $c_i$, are usually organized graphically in a table called *Butcher tableau*, i.e.,

$$\begin{array}{c|ccc} c_1 & a_{11} & \dots & a_{1s} \\ \vdots & \vdots & & \vdots \\ c_s & a_{s1} & \dots & a_{ss} \\ \hline & b_1 & \dots & b_s \end{array}$$

This graphical representation allows to present in a compact form the numerical method. The family of Runge-Kutta methods is divided in two subsets, explicit and implicit methods.

**Definition 3.2.** *A Runge-Kutta method is explicit if and only if $a_{ij} = 0$ for all $i \leq j \leq s$, otherwise it is implicit.*

The main difference from an implementation point of view between explicit and implicit methods is that implicit methods require the solution of a nonlinear system of equations at each time step, while explicit methods are completely defined thanks to a recursive process. Therefore, implicit methods require a higher computational cost than explicit methods, as fixed point or Newton iterations have to be employed at each time step to compute the solution. On the other side, implicit methods are endowed with favorable properties which are not achievable by explicit methods. In this work, we will mainly consider explicit methods, as Monte Carlo simulations will be required and the computational cost given by implicit methods would not be affordable. In particular, we will

consider for our examples mainly three numerical schemes, the Explicit Euler (EE), the explicit midpoint (MP) and the classic Runge-Kutta (RK4) method, which are defined by the following Butcher tableaux

$$\text{(EE)}\ \begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array} \qquad \text{(MP)}\ \begin{array}{c|cc} 0 & 0 & 0 \\ 1/2 & 1/2 & 0 \\ \hline & 0 & 1 \end{array} \qquad \text{(RK4)}\ \begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ \hline & 1/6 & 1/3 & 1/3 & 1/6 \end{array}$$

As we can remark from their tableaux, these methods are all explicit, with one, two and four stages respectively.
One relevant property of Runge-Kutta methods in the frame of this work is the order of convergence, defined as follows.

**Definition 3.3.** *Let us consider a sufficiently smooth differential equation* (14) *and the numerical solution $U_1$ given by one step of the Runge-Kutta method defined in Definition 3.1. If there exist $q > 0$ and a positive constant $C$ independent of $h$ such that*

$$\|u(h) - U_1\| \leq Ch^{q+1},$$

*then the method has local order $q$.*

It is possible to verify [10] that EE has local order 1, MP has local order 2 and RK4 has local order 4.
The notation in (15) is derived from the notion of exact and numerical flow map of the differential equation (14). Let us consider the function $\Phi\colon \mathbb{R}^d \to \mathbb{R}^d$ such that the solution $u$ is given by

$$u(t) = \Phi_t(u_0),$$

i.e., the flow of the differential equation $\Phi$ maps the initial condition into the solution at time $t$. In the same way, we can write the solution at a time $t_2$ given the solution at time $t_1 < t_2$ as

$$u(t_2) = \Phi_{t_2 - t_1}(u(t_1)).$$

The numerical solution given by a Runge-Kutta method can be as well written in terms of a flow map. If we consider a time step $h$ and the solution $U_k$ at the time step $t_k$, then we can write in a compact form the numerical solution at time $t_{k+1}$ as

$$U_{k+1} = \Psi_h(U_k),$$

where the function $\Psi\colon \mathbb{R}^d \to \mathbb{R}^d$ is the numerical flow map of the Runge-Kutta method. Let us remark that the function $\Psi$ is uniquely determined by the coefficients $a_{ij}, b_i$ introduced above.
Stability is one of the main concerns when applying Runge-Kutta methods to a stable ODE [11]. In particular, when integrating numerically a *stiff* equation the stability of the numerical method is fundamental. There are no precise definitions of the stiffness of a differential equation. In general, we can say that a stable equation (14) is stiff when $\max|\mathrm{Re}\{\lambda_i\}|$ is large, where $\lambda_i$, with $i = 1, \ldots, d$, are the eigenvalues of the Jacobian of the function $f$ defining the ODE evaluated at some value $\hat{u}$. Often, the value $L = \max|\mathrm{Re}\{\lambda_i\}|$ is referred to as the *stiffness index* of the differential equation. For example,

when parabolic Partial Differential Equations (PDE's) are discretized in space with the method of lines or with the Finite Elements Methods, the temporal ODE's arising from the discretization are often stiff. In order to introduce the basic stability definitions for Runge-Kutta methods, we have to consider the following test equation

$$\begin{aligned} u(t) &= \lambda u, \quad \lambda < 0,\ t > 0, \\ u(0) &= u_0. \end{aligned} \tag{17}$$

Applying a Runge-Kutta method to the test equation allows to deduce its stability properties for any sufficiently smooth equation (14) thanks to a linearization procedure. The following result is needed for this introduction about stability.

**Proposition 3.1.** *Consider a s-stage Runge-Kutta method defined by coefficients $b_i$, $a_{ij}$ applied to the test problem* (17)*. Then, it is possible to write one step of the method as*

$$U_1 = R(h\lambda)U_0,$$

*where $R(z)$ is a rational function given by*

$$R(z) = 1 + b^T z(I - zA)^{-1}\mathbb{1},$$

*where $I$ is the identity matrix in $\mathbb{R}^{s\times s}$ and*

$$\begin{aligned} b &= \begin{pmatrix} b_1 & b_2 & \dots & b_s \end{pmatrix}^T, \\ A &= (a_{ij})_{i,j=1}^{s}, \\ \mathbb{1} &= \begin{pmatrix} 1 & 1 & \dots & 1 \end{pmatrix}^T \in \mathbb{R}^s. \end{aligned}$$

The function $R(z)$ above defines the main stability concepts of the numerical integrator, and therefore is called the *stability function* of the Runge-Kutta method. In particular, the numerical approximation of the solution of (17) at time $t_k = kh$ is given by

$$U_k = R(h\lambda)^k U_0.$$

Therefore, if we require that the numerical solution remains bounded, we have to impose

$$|R(h\lambda)| \leq 1.$$

This defines the stability domain of the Runge-Kutta method.

**Definition 3.4.** *The stability domain $S$ of any Runge-Kutta method is*

$$S := \{z \in \mathbb{C}\colon |R(z)| \leq 1\}.$$

Studying the domain $S$ we can the stability properties of the numerical method, such as the $A$-stability.

**Definition 3.5.** *A numerical method is called A-stable if*

$$S \supset \mathbb{C}^-, \quad \mathbb{C}^- := \{z \in \mathbb{C}\colon \operatorname{Re}(z) < 0\}.$$

Let us remark that numerical methods which are $A$-stable do not introduce any restriction on the time step when applied to the test equation. Moreover, it is possible to show that explicit Runge-Kutta methods are never $A$-stable. This is due to the fact that for explicit methods $R(z)$ is a polynomial, therefore it diverges when $z \to -\infty$. Hence, in order to ensure $A$-stability, it is necessary to consider implicit methods. Let us consider for example EE. Its stability function is given by

$$R_{\mathrm{EE}}(z) = 1 + z.$$

Since $z = h\lambda$ and in order to ensure stability, the time step $h$ has to satisfy

$$h \leq \frac{1}{2|\lambda|}. \tag{18}$$

If we consider a problem where $\lambda$ is large in absolute value, integrating the ODE with EE implies a high computational cost. All explicit numerical method have condition on the time step similar to (18). On the other hand, as we have explained above, in order to compute the numerical solution with an implicit method it is necessary to solve a nonlinear system at each iteration of the time integration. Hence, for an equal time step $h$, the computational cost for implicit methods is considerably higher than for explicit methods, and it is not possible to predict a priori the number of function evaluations needed to achieve a fixed tolerance. In order to overcome this issue, stabilized explicit methods based on Chebyshev polynomials have been studied. In this work, we consider the Runge-Kutta-Chebyshev (RKC) method [19]. Given a number of stages $s$ in $\mathbb{N}^*$, with $s \geq 2$, one step of RKC applied to (14) is defined by the following recursion

$$\begin{aligned}
g_0 &= U_0, \\
g_1 &= U_0 + \frac{h}{s^2} f(U_0), \\
g_i &= \frac{2h}{s^2} f(g_{i-1}) + 2g_{i-1} - g_{i-2}, \quad i = 2, \ldots, s, \\
U_1 &= g_s.
\end{aligned}$$

Thanks to the properties of Chebyshev polynomials, we have that the stability polynomial $R_s(z)$ of a $s$-stage RKC method is such that

$$|R_s(z)| \leq 1 \iff z \in [-2s^2, 0].$$

Let us remark that the size of the stability domain on the real negative axis increases quadratically with the number of stages of the method, which corresponds to the number of function evaluations required to perform one temporal step. Hence, if $\lambda < 0$ and given an arbitrary time step $h$, RKC is stable provided that the number of stages satisfies

$$s \geq \max\left\{2, \sqrt{\frac{1}{2}h\lambda}\right\}.$$

Finally, it is possible to show that RKC has order one. For all the reasons above, it is clear that RKC methods are well suited for approximating the solution of stiff equations with large real negative eigenvalues, as, e.g., discretized parabolic PDE's.
The basic definitions and properties introduced above are sufficient for understanding the analysis of the probabilistic numerical method introduced in (16). In the following, we will motivate the numerical method as well as study some of its features.

## 3.2 Motivation of probabilistic methods

Chaotic ODE's are a class of equations whose solution depends strongly on the initial condition. In particular, a small perturbation to the state of the system can lead to a completely different solution. Therefore, when we integrate numerically a chaotic equation, the deviation from the exact solution due to the discretization error can lead to providing numerical solutions which are far from the exact solution. Hence, in this case probabilistic methods can be applied to provide probability distributions instead of potentially wrong punctual solutions.
One of the most famous examples of chaotic ODE's is the Lorenz system. This ODE is defined by

$$\begin{aligned} x' &= \sigma(y - x), & x(0) &= -10, \\ y' &= x(\rho - z) - y, & y(0) &= -1, \\ z' &= xy - \beta z, & z(0) &= 40. \end{aligned} \tag{19}$$

Edward Lorenz discovered in 1963 that for certain values of the parameters $\rho$, $\sigma$ and $\beta$ this equation has a chaotic behavior, i.e., small variations of the initial conditions result in completely different solutions of the system. A sample trajectory of the solution in the state space is depicted in Figure 2.
Let us consider $\sigma = 10$, $\rho = 28$, $\beta = 8/3$. For this value, the solution has indeed a chaotic behavior. We solve the equation with a time step $h = 0.01$ using the probabilistic solver implemented with the Gauss collocation method on two stages, an implicit Runge-Kutta scheme of fourth order defined by the following Butcher tableau

| | | |
|---|---|---|
| $1/2 - \sqrt{3}/6$ | $1/4$ | $1/4 - \sqrt{3}/6$ |
| $1/2 + \sqrt{3}/6$ | $1/4 + \sqrt{3}/6$ | $1/4$ |
| | $1/2$ | $1/2$ |

In Figure 3 we show 10 realizations of the numerical solution given by the probabilistic solver and the numerical solution without any added perturbation from initial time $t = 0$ to final time $T = 20$. We can remark that all the trajectories given by the probabilistic solver coincide with the punctual solution up to a time $\bar{t}$, approximately equal to 12, where the random perturbation force the numerical solution on another trajectory.

## 3.3 Method properties

The properties of the probabilistic method introduced in (16) have been partially investigated in [3], where a result on strong and weak convergence of the numerical solution is derived. In the following, we will briefly introduce this analysis and show a numerical example that confirms the theoretical results.
A property which conversely has not been investigated is the convergence of a Monte Carlo approximation of the numerical solution. Let us consider the distribution $\mathcal{Q}_h$ given by the probabilistic solver, and the distribution $\delta_u$ representing the Dirac delta concentrated on the true solution. In [3], and as we will show in the following, the convergence of $\mathcal{Q}_h$ to $\delta_u$ is theoretically described. As pointed out in [15], we can only access a distribution $\mathcal{Q}_h^M$ through $M$ repeated samples of the probabilistic solution, which will approximate the distribution $\mathcal{Q}_h$. As it is schematically described in [15], the two following convergences have to be analyzed

$$\mathcal{Q}_h^M \overset{M\to\infty}{\to} \mathcal{Q}_h \overset{h\to 0}{\to} \delta_u.$$

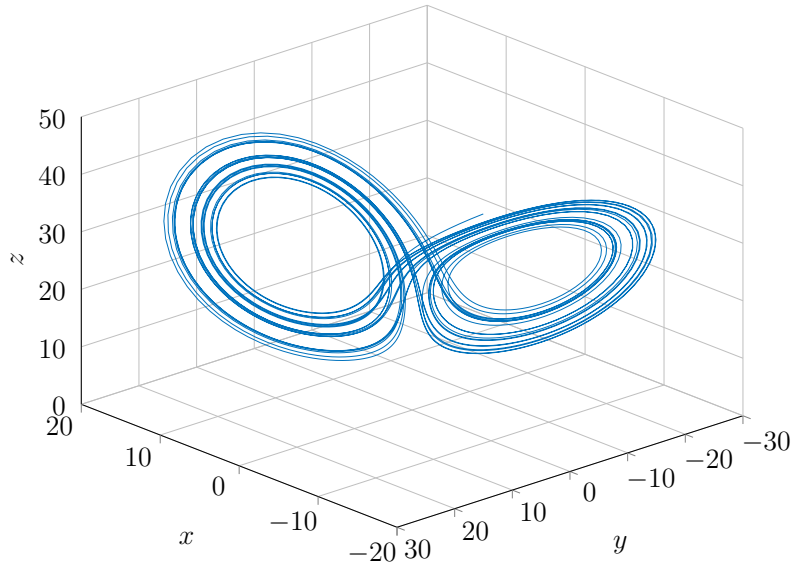



**Figure 2:** Trajectory of the solution of the Lorenz system in the state space.

In [3], only the second convergence is treated. In the following sections, we will show that in case the initial condition is deterministic, the convergence of $\mathcal{Q}_h^M$ to $\delta_u$ is independent of the number of samples $M$, and therefore the computational cost due to the approximation of $\mathcal{Q}_h$ through repeated sampling is negligible.
Finally, we will discuss the case where this favorable convergence property is not valid together with an application of multi-level techniques, and we will conclude with a brief analysis of the stability of this numerical method.

### 3.3.1 Strong convergence

The first property we treat is the strong convergence of the numerical solution. All the results presented can be found in details in [3]. Let us first recall the definition of the strong order of convergence of a numerical method.

**Definition 3.6.** *Given the equation* (14)*, the probabilistic method* (16) *is said to have a strong order of convergence $r$ if there exists a constant $C$ such that*

$$\sup_{t_k=kh} \mathbb{E}|U_k - u(t_k)| \leq Ch^r,$$

*for $h$ small enough.*

In order to prove a result of strong convergence for the method defined in (16), we need to state two assumptions on the random variables $\xi_k$ as well as on the deterministic part of the probabilistic integrator.

**Assumption 3.1.** *The random variables $\xi_k(t)$ satisfy*

$$\mathbb{E}|\xi_k(t)\xi_k(t)^T|_F^2 \leq Kt^{2p+1}.$$

*Furthermore, there exists a matrix $Q$ independent of $h$ such that*

$$\mathbb{E}[\xi_k(h)\xi_h(h)^T] = Qh^{2p+1},$$

*where $p \geq 1$.*

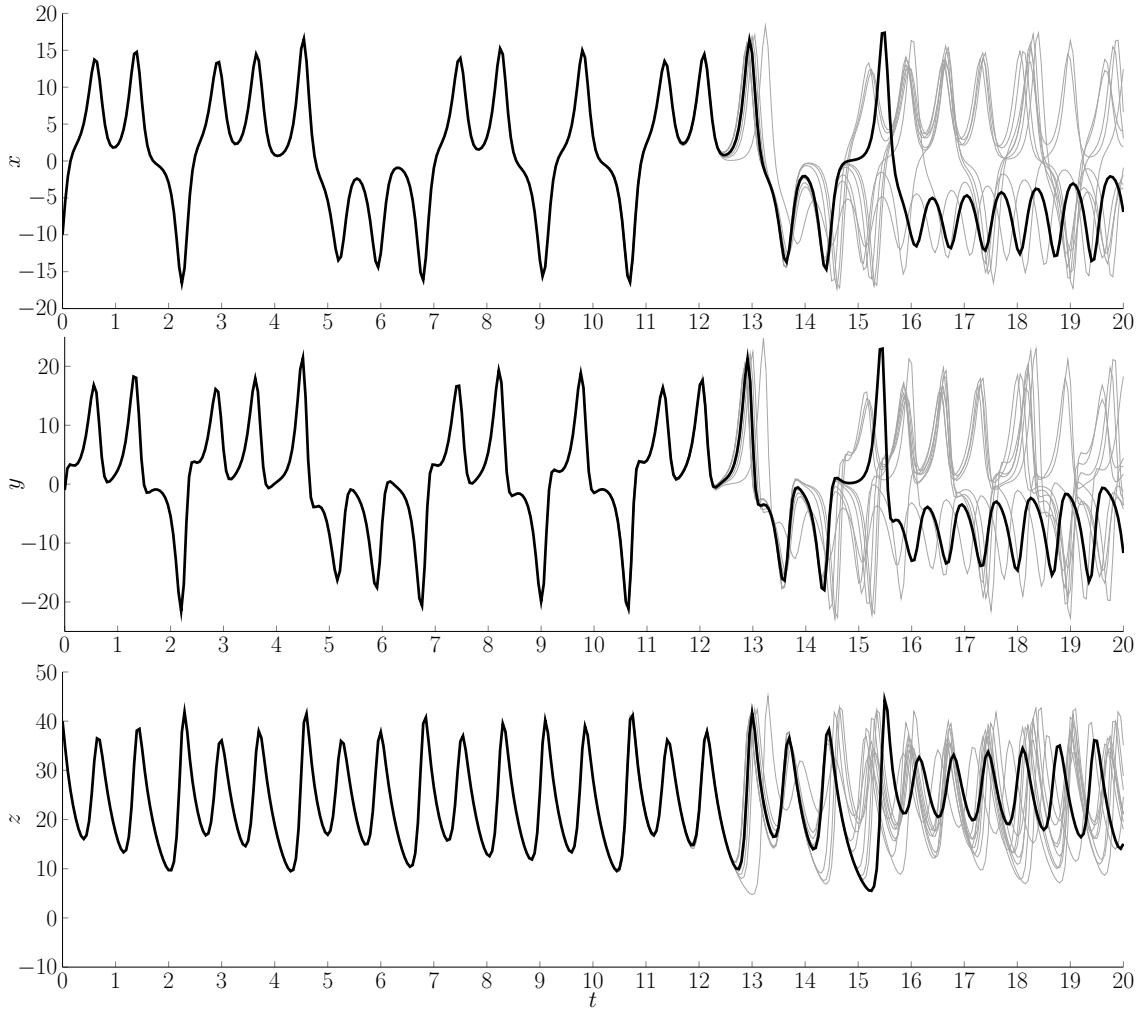



**Figure 3:** Solution of the Lorenz system obtained with the deterministic solver (thick black) and realizations of the solution obtained with the probabilistic solver (light gray).

Let us remark that if $Q = \sigma I$, with $I$ the identity matrix in $\mathbb{R}^{d\times d}$ and $\sigma > 0$, the method (16) can be simulated by

$$U_{k+1} = \Psi_h(U_k) + \sqrt{\sigma} h^{p+\frac{1}{2}} Z_k,$$

where $Z_k$ is a Gaussian random vector with independent entries $Z_{k,i} \sim \mathcal{N}(0,1), i = 1,\dots,d$.
A further assumption on the deterministic component $\Psi$ of the numerical method is needed.

**Assumption 3.2.** *The function $f$ and a sufficient number of its derivatives are bounded uniformly in $\mathbb{R}^n$ in order to ensure that $f$ is globally Lipschitz and that the numerical flow map $\Psi_h$ has uniform local truncation error of order $q+1$, i.e.,*

$$\sup_{u\in\mathbb{R}^n} |\Psi_t(u) - \Phi_t(u)| \le K t^{q+1}.$$

Moreover, the following discrete Gronwall lemma has to be exploited to complete the proof.

**Lemma 3.1** (Discrete Gronwall Lemma)**.** *Let $y_n$ be a nonnegative sequence and $C_1, C_2$ positive constants. If*

$$y_n \leq C_1 + C_2 \sum_{k=0}^{n-1} y_k,$$

*then*

$$y_n \leq C_1 \exp(nC_2).$$

Under the two assumptions above we can now state the following result of strong convergence.

**Proposition 3.2** (Strong Convergence)**.** *Under assumptions 3.1 and 3.2 it follows that there is $K > 0$ such that*

$$\sup_{0<kh<T} \mathbb{E}|u_k - U_K|^2 \leq K h^{2\min\{p,q\}}. \tag{20}$$

*Furthermore*

$$\sup_{0\leq t\leq T} \mathbb{E}|u(t) - U(t)| \leq K h^{\min\{p,q\}}.$$

Let us report for a matter of completeness the proof of the first inequality. The second result regards the convergence of a continuous interpolation of the numerical solution on grid points, where the interpolation has to be intended as a stochastic process depending on a continuous random variable $\xi(t)$.

*Proof.* Given the method in (16) and writing the exact solution of (14) as

$$u_{k+1} = \Phi_h(u_k),$$

one can compute the truncation error $\epsilon_k = \Psi_h(U_k) - \Phi_h(U_k)$, so that

$$U_{k+1} = \Phi_h(U_k) + \epsilon_k + \xi_k(h).$$

Therefore

$$\begin{aligned} e_{k+1} &= u_k - U_k \\ &= \Phi_h(u_k) - \Phi_h(u_k - e_k) - \epsilon_k - \xi_k(h). \end{aligned}$$

Taking the expectation and under Assumption 3.1

$$\mathbb{E}|e_{k+1}|^2 = \mathbb{E}|\Phi_h(u_k) - \Phi_h(u_k - e_k) - \epsilon_k|^2 + \mathcal{O}(h^{2p+1}).$$

Developing the square and since $\Phi_h$ is Lipschitz continuous with constant $(1 + Lh)$ and $\epsilon_k = \mathcal{O}(h^{q+1})$ thanks to Assumption 3.2

$$\begin{aligned} \mathbb{E}|e_{k+1}|^2 \leq & (1 + Lh)^2 \mathbb{E}|e_k|^2 + \mathbb{E}\left|\left(h^{\frac{1}{2}}(\Phi_h(u_k) - \Phi_h(u_k - e_k), h^{-\frac{1}{2}}\epsilon_k\right)\right| \\ & + \mathcal{O}\left(h^{2q+2}\right) + \mathcal{O}\left(h^{2p+1}\right). \end{aligned}$$

Then, using Cauchy-Schwarz on the inner product

$$\begin{aligned}\mathbb{E}|e_{k+1}|^2 &\leq (1+\mathcal{O}(h))\,\mathbb{E}|e_k|^2 + \mathcal{O}\left(h^{2q+1}\right) + \mathcal{O}\left(h^{2p+1}\right)\\ &\leq C_1 h\mathbb{E}|e_k|^2 + \mathbb{E}|e_k|^2 + \mathcal{O}\left(h^{2q+1}\right) + \mathcal{O}\left(h^{2p+1}\right)\\ &\leq C_1 h\sum_{i=0}^{k}\mathbb{E}|e_i|^2 + \mathcal{O}\left(h^{-1}\right)\left(\mathcal{O}\left(h^{2q+1}\right) + \mathcal{O}\left(h^{2p+1}\right)\right)\\ &\leq C_1 h\sum_{i=0}^{k}\mathbb{E}|e_i|^2 + \mathcal{O}\left(h^{2q}\right) + \mathcal{O}\left(h^{2p}\right).\end{aligned}$$

Therefore by Proposition 3.1

$$\begin{aligned}\mathbb{E}|e_k|^2 &\leq C_2 h^{2\min\{p,q\}}\exp(C_1 kh)\\ &\leq C_2 h^{2\min\{p,q\}}\exp(C_1 T)\\ &\leq Ch^{2\min\{p,q\}},\end{aligned}$$

which concludes the proof of (20). □

*Remark* 3.2. Let us remark that the result Theorem 3.2 indicates that a good choice for the noise scale, i.e., the integer $p$, is given by the order of convergence of the deterministic method $q$. In practice, this means that the stochastic noise introduced during time integration has the same order of magnitude of the numerical error with respect to the step of the time discretization $h$.

### 3.3.2 Weak convergence

Another relevant feature of stochastic numerical methods is their weak convergence. The following analysis has been carried out in [3], and we report the main steps in this work. Let us recall the definition of weak order of convergence

**Definition 3.7.** *Given the equation* (14)*, the probabilistic method* (16) *is said to have a weak order of convergence $r$ if there exists a constant $C$ such that, for any function $\varphi$ sufficiently smooth*

$$\sup_{t_k=kh}|\mathbb{E}[\varphi(U_k)] - \varphi(u(t_k))| \leq Ch^r,$$

*for $h$ small enough.*

For the numerical method introduced in (16), a result of weak convergence can be proved using a technique of *backward error analysis*. The main idea behind this technique is finding a *modified equation* that the numerical method solves exactly or with a higher accuracy than the original equation.
Let us consider (14) and the numerical method (16). Using the Lie derivative notation, it is possible to find the differential operators $\mathcal{L}$ and $\mathcal{L}^h$ such that for all $\varphi \in \mathcal{C}^\infty(\mathbb{R}^d, \mathbb{R})$

$$\begin{aligned}\varphi(\Phi_h(u)) &= \left(e^{h\mathcal{L}}\varphi\right)(u),\\ \mathbb{E}[\varphi(U_1|U_0=u)] &= \left(e^{h\mathcal{L}^h}\varphi\right)(u).\end{aligned} \tag{21}$$

In particular, $\mathcal{L} = f \cdot \nabla$ and the explicit definition of $\mathcal{L}^h$ is not needed in this scope. We now introduced a modified ODE

$$\hat{u}' = f^h(\hat{u}),$$

and a modified SDE

$$\mathrm{d}\tilde{u} = f^h \tilde{u}\,\mathrm{d}t + \sqrt{h^{2p}Q}\,\mathrm{d}W\,, \tag{22}$$

where $p$ has been introduced in Assumption 3.1. We rewrite the solution of these equations in terms of Lie derivatives as for (21) introducing the differential operators $\hat{\mathcal{L}}$ and $\tilde{\mathcal{L}}$, *i.e.*,

$$\varphi\left(\hat{u}(h)|\hat{u}(0) = u\right) = \left(e^{h\hat{\mathcal{L}}^h}\varphi\right)(u),$$
$$\varphi\left(\tilde{u}(h)|\tilde{u}(0) = u\right) = \left(e^{h\tilde{\mathcal{L}}^h}\varphi\right)(u).$$

Therefore,

$$\hat{\mathcal{L}}^h = f^h \cdot \nabla,$$
$$\tilde{\mathcal{L}}^h = f^h \cdot \nabla + \frac{1}{2}h^{2p}Q : \nabla^2,$$

where the scalar product $A : B$ in $\mathbb{R}^{d\times d}$ is defined as $A : B = \mathrm{tr}\left(A^T B\right)$. The operator $\tilde{\mathcal{L}}^h$ is the *generator* of (22). Introducing the modified equations above, it is possible to study the weak error between the solution of (14) and the numerical solution given by the probabilistic integrator [3]. In particular, this is achievable considering the approximation given by the probabilistic integrator of the solution of the modified SDE (22). Let us introduce for completeness the following assumption, needed to prove the result of weak convergence.

**Assumption 3.3.** *The function $f$ in* (14) *is in $\mathcal{C}^\infty$ and all its derivatives are uniformly bounded in $\mathbb{R}^n$. Furthermore, $f$ is such that for all functions $\varphi$ in $\mathcal{C}^\infty(\mathbb{R}^n, \mathbb{R})$*

$$\sup_{u\in\mathbb{R}^n}\left|e^{h\mathcal{L}}\varphi(u)\right| \leq (1 + Lh)\sup_{u\in\mathbb{R}^n}|\varphi(u)|\,,$$
$$\sup_{u\in\mathbb{R}^n}\left|e^{h\mathcal{L}^h}\varphi(u)\right| \leq (1 + Lh)\sup_{u\in\mathbb{R}^n}|\varphi(u)|\,,$$

*for some $L > 0$.*

We can now state the following result of weak convergence.

**Proposition 3.3.** *Consider the numerical method* (16) *and Assumptions 3.1, 3.2 and 3.3. Then for any function $\varphi$ in $\mathcal{C}^\infty$ endowed with the properties of Assumption 3.3,*

$$|\varphi(u(T)) - \mathbb{E}[\varphi(U_k)]| \leq Kh^{\min\{2p,q\}}, \quad kh = T,$$

*and*

$$|\mathbb{E}[\varphi(\tilde{u}(T))] - \mathbb{E}[\varphi(U_k)]| \leq Kh^{2p+1}, \quad kh = T,$$

*with $u$ and $\tilde{u}$ solutions of* (14) *and* (22).

The proof of this result can be found in [3]. In the following, we will verify numerically the correctness of this result.

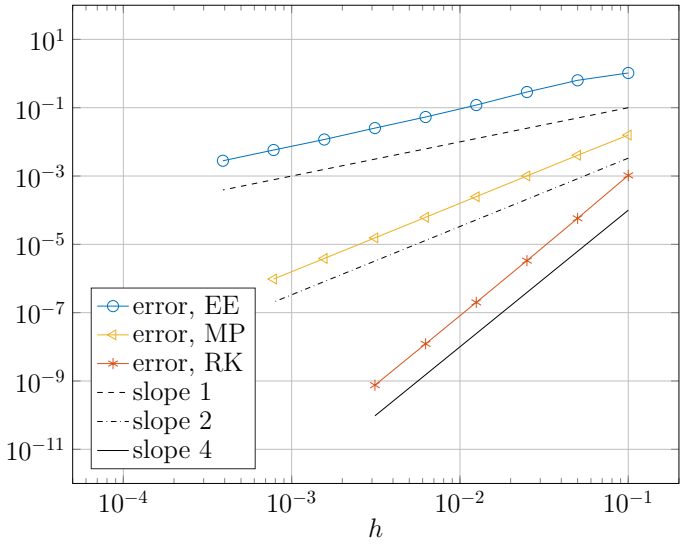



**Figure 4:** Weak order of convergence of (16) applied to (23).

### 3.3.3 Numerical experiment - Weak convergence

Let us consider the FitzHug-Nagumo model, defined by the following ODE

$$\begin{aligned} x' &= c\left(x - \frac{x^3}{3} + y\right), \quad x(0) = -1, \\ y' &= -\frac{1}{c}(x - a + by), \quad y(0) = 1, \end{aligned} \tag{23}$$

where $a, b, c$ are real parameters with values $a = 0.2$, $b = 0.2$, $c = 3$. We integrate numerically the equation with (16), using EE, MP and RK4 as deterministic method up to final time $T = 1$. Therefore, Assumption 3.2 holds with $q = 1$, $q = 2$ and $q = 4$ respectively. We consider that Assumption 3.1 is verified with $Q = \sigma I$, where $\sigma = 0.5$ and $I$ is the identity matrix. We then consider for each deterministic solver the value of $p$ to be equal to $q$. Therefore, the weak order of convergence of (16) is equal to the order of the Runge-Kutta integrator. We let $h$ vary in order to verify the order of convergence and consider 500000 repetitions of the trajectory in order to approximate the expectation of the solution and obtain smooth error curves. Furthermore, we consider as function $\varphi$ in Theorem 3.3 to be the Euclidean norm of the solution. In order to compute the error, we use a reference solution computed with RK4 with time step $h = 10^{-7}$, so that the error of the reference solution is negligible. Results (Figure 4) show that the weak order of convergence of the three considered methods respect the theoretical order predicted in Theorem 3.3.

### 3.3.4 Monte Carlo approximation

Let us consider the numerical method introduced in (16) and the Monte Carlo approximation

$$\hat{Z} = \frac{1}{M}\sum_{i=1}^{M} \varphi\left(U_N^{(i)}\right). \tag{24}$$

The mean square error (MSE) of $\hat{Z}$ is given by

$$\begin{aligned}\mathrm{MSE}(\hat{Z}) &= \mathbb{E}\left[\left(\hat{Z} - \varphi\left(u(T)\right)\right)^2\right] \\ &= \mathrm{Var}(\hat{Z}) + \mathbb{E}\left[\hat{Z} - \varphi\left(u(T)\right)\right]^2 \\ &\leq \mathrm{Var}(\hat{Z}) + Ch^{2\min\{2p,q\}},\end{aligned}$$

where the second term is bounded thanks to proposition 3.3 with $C > 0$. In the standard theory of SDE's, the first term is bounded by $CM^{-1}$, where $C$ is a positive constant. In the probabilistic solver we consider in this work, it is possible to bound the first term with a function of the time step $h$. Intuitively, this favorable property comes from the fact that the noise scale is of the same order of magnitude of the time step.

**Lemma 3.2.** *Consider the numerical method* (16) *applied to a one-dimensional ODE with* $\Psi$ *any explicit Runge-Kutta method on* $s$ *stages and Assumption 3.1. Then the numerical solution* $U_k$ *at time* $t_k = kh$ *satisfies*

$$\mathrm{Var}(U_k) \leq C_1 \,\mathrm{Var}(U_0) + C_2 Q h^{2p}, \quad k = 1, \ldots, N,$$

*with* $C_1$, $C_2$ *positive constants.*

*Proof.* Let us consider as the numerical integrator $\Psi$ the Explicit Euler method and $p = 1$, coherently with the common choice that $p$ is equal to the order or the deterministic integrator. Then, we can write the numerical solution $U_{k+1}$ as

$$U_{k+1} = U_k + hf(U_k) + \xi_k(h),$$

Then, thanks to Assumption 3.1, we can compute the variance of $U_{k+1}$ as

$$\begin{aligned}\mathrm{Var}(U_{k+1}) &= \mathrm{Var}(U_k + hf(U_k)) + h^3 Q \\ &\leq 2\,\mathrm{Var}(U_k) + 2h^2\,\mathrm{Var}(f(U_k)) + h^3 Q,\end{aligned} \tag{25}$$

where $Q > 0$ and we exploited that for any random variables $X$, $Y$,

$$\mathrm{Var}(X + Y) \leq 2\,\mathrm{Var}(X) + 2\,\mathrm{Var}(Y). \tag{26}$$

Since $f$ is Lipschitz continuous with constant $C_L$, we can bound the second term in the sum above as

$$\begin{aligned}\mathrm{Var}(f(U_k)) &= \mathrm{Var}(f(U_k) - f(\mathbb{E}\left[U_k\right])) \\ &\leq \mathbb{E}\left[(f(U_k) - f(\mathbb{E}\left[U_k\right]))^2\right] \\ &\leq C_L^2 \mathbb{E}\left[(U_k - \mathbb{E}(U_k))^2\right] \\ &= C_L^2\,\mathrm{Var}(U_k).\end{aligned} \tag{27}$$

Hence, we find

$$\begin{aligned}\frac{1}{2}\operatorname{Var}(U_{k+1}) &\le (1+C_L^2h^2)\operatorname{Var}(U_k)+\frac{1}{2}Qh^3\\ &\le (1+C_L^2h^2)^2\operatorname{Var}(U_{k-1})+\frac{1}{2}Qh^3(1+C_L^2h^2)+\frac{1}{2}Qh^3\\ &\le (1+C_L^2h^2)^k\operatorname{Var}(U_0)+\frac{1}{2}Qh^3\sum_{i=0}^{k-1}(1+C_L^2h^2)^i\\ &\le \exp\big(C_L^2T^2\big)\operatorname{Var}(U_0)+\frac{1}{2}Qh^3\sum_{i=0}^{k-1}(1+C_L^2h^2)^i\\ &= C_1\operatorname{Var}(U_0)+\frac{1}{2}Qh^3\sum_{i=0}^{k-1}(1+C_L^2h^2)^i.\end{aligned}$$

We then bound the second term as follows

$$\begin{aligned}Qh^3\sum_{i=0}^{k-1}(1+C_L^2h^2)^i &= \frac{(1+C_L^2h^2)^k-1}{h^2C_L^2}Qh^3\\ &\le \frac{\exp\big(kC_L^2h^2\big)-1}{C_L^2}Qh\\ &\le \frac{\exp\big(TC_L^2h\big)-1}{C_L^2}Qh.\end{aligned}\tag{28}$$

We then exploit the Taylor expansion of the exponential function and write

$$\begin{aligned}Qh^3\sum_{i=0}^{k-1}(1+C_L^2h^2)^i &\le \frac{Qh}{C_L^2}\sum_{i=1}^{\infty}\frac{(TC_L^2h)^i}{i!}\\ &\le \left(\frac{1}{C_L^2T}\sum_{i=1}^{\infty}\frac{(T^2C_L^2)^i}{i!}\right)Qh^2\\ &= \frac{\exp\big(T^2C_L^2\big)-1}{C_L^2T}Qh^2\\ &= C_2Qh^2.\end{aligned}$$

Thus, the result is proved for Explicit Euler. Let us consider now any explicit Runge-Kutta method $\Psi$, and let us rewrite (16) as

$$U_{k+1} = U_k + h\tilde{\Psi}(U_k) + \xi_k(h),$$

where $\tilde{\Psi}(x) \coloneqq h^{-1}(\Psi(x)-x)$ is given by

$$\tilde{\Psi}(U_k) = \sum_{i=1}^{s} b_iK_i,$$

and $K_i$, $i=1,\dots,s$, are the stages of the Runge-Kutta method. Then, proceeding as above

$$\operatorname{Var}(U_{k+1}) \le 2\operatorname{Var}(U_k) + 2h^2\operatorname{Var}(\tilde{\Psi}(U_k)) + Qh^{2p+1}.$$

Let us consider the second term. A direct bound, following from a generalization on $s$ terms of (26) is

$$\mathrm{Var}(\tilde{\Psi}(U_k)) \leq s \sum_{i=1}^{s} b_i^2 \,\mathrm{Var}(K_i). \tag{29}$$

Hence, we can consider the variance of each stage singularly. Since we are only considering explicit Runge-Kutta method, it is possible to estimate the single variances recursively

$$\begin{aligned}
\mathrm{Var}(K_1) &= \mathrm{Var}(f(U_k)) \leq C_L^2 \,\mathrm{Var}(U_k), \\
\mathrm{Var}(K_2) &= \mathrm{Var}(f(U_k + ha_{21}K_1)) \leq C_L^2 \,\mathrm{Var}(U_k + ha_{21}K_1) \\
&\leq 2C_L^2(\mathrm{Var}(U_k) + h^2 a_{21}^2 \,\mathrm{Var}(K_1)) \\
&\leq 2C_L^2(1 + T^2 a_{21}^2 C_L^2) \,\mathrm{Var}(U_k) \leq C \,\mathrm{Var}(U_k) \\
\implies \mathrm{Var}(K_i) &\leq \mathrm{Var}(f(U_k + h\textstyle\sum_{j=1}^{i-1} a_{ij}K_j)) \leq C \,\mathrm{Var}(U_k),\ \forall i = 2, \dots, s,
\end{aligned}$$

where $C$ is a positive varying from one line to another depending on $C_L$, $T$ and the coefficients of the Runge-Kutta method. We then substitute in (29) and get

$$\mathrm{Var}(U_{k+1}) \leq 2(1 + Csh^2 \textstyle\sum_{i=1}^{s} b_i^2) \,\mathrm{Var}(U_k) + Qh^{2p+1}.$$

Finally, we can proceed as explained above in detail in the case of Explicit Euler and recur over $k$ to obtain the desired bound. □

We can show a similar result for any implicit Runge-Kutta method. In this case, a restriction on the time step will be required in order to bound the variance of the numerical solution. We will discuss how this restriction is connected with the well-posedeness of the numerical method.

**Lemma 3.3.** *Consider the numerical method* (16) *applied to a one-dimensional ODE with* $\Psi$ *any explicit or implicit Runge-Kutta method on s stages and Assumption 3.1. Then, if* $h$ *is small enough, the numerical solution* $U_k$ *at time* $t_k = kh$ *satisfies*

$$\mathrm{Var}(U_k) \leq C_1 \,\mathrm{Var}(U_0) + C_2 Q h^{2p}, \quad k = 1, \dots, N,$$

*with* $C_1$, $C_2$ *positive constants.*

*Proof.* Let us consider as $\Psi$ the Implicit Euler method and $p = 1$. Then, we can write one step of the probabilistic method as

$$U_k = U_{k-1} + hf(U_k) + \xi_k(h).$$

Applying (25) and (27) we get thanks to Assumption 3.1

$$\mathrm{Var}(U_k) \leq 2 \,\mathrm{Var}(U_{k-1}) + 2h^2 C_L^2 \,\mathrm{Var}(U_k) + Qh^3.$$

Hence, defining the coefficient $\beta$ as

$$\beta = \frac{1}{1 - 2h^2 C_L^2},$$

and if the time step $h$ is bounded by

$$h < \frac{1}{\sqrt{2} C_L},$$

then $\beta > 0$ and we can deduce

$$\begin{aligned}\frac{1}{2}\operatorname{Var}(U_k) &\leq \beta(\operatorname{Var}(U_{k-1}) + \frac{1}{2}Qh^3) \\ &\leq \beta^k \operatorname{Var}(U_0) + \frac{1}{2}\left(\sum_{i=1}^{k}\beta^i\right)Qh^3 \\ &\leq \beta^N \operatorname{Var}(U_0) + \frac{1}{2}T\beta^N Qh^2,\end{aligned}$$

which proves the result for the Implicit Euler method. For any implicit or explicit Runge-Kutta method we can write one step of the probabilistic method as

$$U_k = U_{k-1} + h\sum_{i=1}^{s} b_i K_i + \xi_k(h).$$

Then thanks to (26), (27) and Assumption 3.1 we obtain

$$\operatorname{Var}(U_k) \leq 2\operatorname{Var}(U_{k-1}) + 2h^2 \operatorname{Var}(\textstyle\sum_{i=1}^{s} b_i K_i) + Qh^{2p+1}. \tag{30}$$

Let us consider the second term in the bound above. Thanks to the generalization on $s$ terms of (26) we get

$$\operatorname{Var}(\textstyle\sum_{i=1}^{s} b_i K_i) \leq s\sum_{i=1}^{s} b_i^2 \operatorname{Var}(K_i).$$

Considering now the variance of all single stages of the Runge-Kutta scheme, we can exploit (27) and (26) to get

$$\begin{aligned}\operatorname{Var}(K_i) &= \operatorname{Var}(f(U_{k-1} + h\textstyle\sum_{j=1}^{s} a_{ij}K_j)) \\ &\leq C_L^2 \operatorname{Var}(U_{k-1} + h\textstyle\sum_{j=1}^{s} a_{ij}K_j) \\ &\leq 2C_L^2 \operatorname{Var}(U_{k-1}) + 2C_L^2 h^2 \operatorname{Var}(\textstyle\sum_{j=1}^{s} a_{ij}K_j) \\ &\leq 2C_L^2 \operatorname{Var}(U_{k-1}) + 2C_L^2 h^2 s \max_{i,j=1,\dots,s} a_{ij}^2 \textstyle\sum_{j=1}^{s} \operatorname{Var}(K_j).\end{aligned}$$

Let us define the constant $\alpha > 0$ as

$$\alpha = 2C_L^2 h^2 s \max_{i,j=1,\dots,s} a_{ij}^2.$$

Then, if the time step $h$ satisfies

$$h < \frac{1}{C_L}\left(\frac{1}{2s\max_{i,j=1,\dots,s} a_{ij}^2}\right)^{1/2},$$

we have that $1-\alpha$ is positive and therefore we can bound the variance of the $i$-th Runge-Kutta stage as

$$\operatorname{Var}(K_i) \leq \frac{2C_L^2}{1-\alpha}\operatorname{Var}(U_{k-1}) + \frac{\alpha}{1-\alpha}\textstyle\sum_{j=1, j\neq i}^{s} \operatorname{Var}(K_j).$$

If for each $i$ we consider a numbering of the Runge-Kutta stages such that $i = s$, we can rewrite the inequality above as

$$\operatorname{Var}(K_s) \leq \frac{2C_L^2}{1-\alpha}\operatorname{Var}(U_{k-1}) + \frac{\alpha}{1-\alpha}\textstyle\sum_{j=1}^{s-1} \operatorname{Var}(K_j).$$

Therefore we can apply the discrete Gronwall inequality (Proposition 3.1) and get

$$\mathrm{Var}(K_i) \leq \frac{2C_L^2}{1-\alpha}\,\mathrm{Var}(U_{k-1})\exp\!\left(\frac{\alpha(s-1)}{1-\alpha}\right).$$

Substituting this inequality in (30) we get

$$\begin{aligned}\frac{1}{2}\,\mathrm{Var}(U_k) &\leq \left(1 + h^2 s \frac{2C_L^2}{1-\alpha}\exp\!\left(\frac{\alpha(s-1)}{1-\alpha}\right)\textstyle\sum_{i=1}^{s} b_i^2\right)\mathrm{Var}(U_{k-1}) + \frac{1}{2}Qh^{2p+1} \\ &\leq \left(1 + h^2 s^2 \frac{2C_L^2}{1-\alpha}\exp\!\left(\frac{\alpha(s-1)}{1-\alpha}\right)\max_{i=1,\dots,s} b_i^2\right)\mathrm{Var}(U_{k-1}) + \frac{1}{2}Qh^{2p+1}.\end{aligned}$$

If we define the constant $\hat{C} > 0$ as

$$\hat{C} \coloneqq s^2 \frac{2C_L^2}{1-\alpha}\exp\!\left(\frac{\alpha(s-1)}{1-\alpha}\right)\max_{i=1,\dots,s} b_i^2,$$

we get

$$\frac{1}{2}\,\mathrm{Var}(U_k) \leq (1+\hat{C}h^2)^k\,\mathrm{Var}(U_0) + \frac{1}{2}Qh^{2p+1}\textstyle\sum_{i=0}^{k-1}(1+\hat{C}h^2)^i.$$

For the second term we proceed as in (28) and get for a constant $\tilde{C} > 0$

$$\begin{aligned}\frac{1}{2}\,\mathrm{Var}(U_k) &\leq (1+\hat{C}h^2)^k\,\mathrm{Var}(U_0) + \tilde{C}Qh^{2p} \\ &\leq \exp\!\left(\hat{C}T^2\right)\mathrm{Var}(U_0) + \tilde{C}Qh^{2p},\end{aligned}$$

thus obtaining the desired result. □

*Remark* 3.3. Let us remark that in the limit for $h$ going to zero, the coefficient $\alpha$ defined for the Implicit Euler method tends to one. Therefore, asymptotically the variance of the numerical solution is bounded independently of the Lipschitz constant defining the ODE. Conversely, for any explicit Runge-Kutta method the constant depends on $C_L$ for any value of $h$.

*Remark* 3.4. The requirement on the time step $h$ of Lemma 3.3 is reasonable because it is required by the numerical method for its well-posedness. Let us denote by $F$ the function defining one step of the probabilistic Implicit Euler, i.e.,

$$F(X) = U_{k-1} + hf(X) + \xi_k(h).$$

In order to apply Banach's fixed point theorem and therefore admit the existence of a fixed point $U_k$, $F$ has to be a contraption. Therefore, evaluating $F$ on two points $X$ and $Y$, we get

$$|F(X) - F(Y)| = h|f(X) - f(Y)| \leq hC_L|X - Y|.$$

Hence, we have to impose $h < 1/C_L$, which is only sightly less restrictive than the bound on $h$ required by Lemma 3.3.

We can now consider the MSE of the estimator $\hat{Z}$ introduced in (24).

**Proposition 3.4.** *Under the assumptions of Lemma 3.2 and if $\varphi$ is Lipschitz continuous with constant $C_L$, the following bound for the MSE of $\hat{Z}$ is valid*

$$\mathrm{MSE}(\hat{Z}) \leq C_1 h^{2\min\{2p,q\}} + \frac{C_2}{M}(\mathrm{Var}(U_0) + h^{2p}).$$

*Proof.* The samples $U_N^{(i)}$ are independent and identically distributed as $U_N$, hence

$$\begin{aligned}\mathrm{Var}(\hat{Z}) &= \mathrm{Var}\left(\frac{1}{M}\sum_{i=1}^{M}\varphi\left(U_N^{(i)}\right)\right)\\ &= \frac{1}{M^2}\sum_{i=1}^{M}\mathrm{Var}\left(\varphi\left(U_N\right)\right)\\ &= \frac{1}{M}\mathrm{Var}\left(\varphi\left(U_N\right)\right).\end{aligned}$$

Since the function $\varphi$ is Lipschitz continuous we can use (27) and Lemma 3.2 and get

$$\mathrm{Var}(\hat{Z}) \leq \frac{C}{M}\mathrm{Var}(U_N) \leq \frac{C}{M}(\mathrm{Var}(U_0) + h^{2p})$$

thus obtaining the following bound for the MSE of $\hat{Z}$

$$\mathrm{MSE}(\hat{Z}) \leq C_1 h^{2\min\{2p,q\}} + \frac{C_2}{M}(\mathrm{Var}(U_0) + h^{2p})$$

□

*Remark* 3.5. Let us remark that in case the initial condition $U_0$ is a known deterministic value, i.e., $\mathrm{Var}(U_0)$ is equal to zero, and the noise scale $p$ is chosen equal to the order of the Runge-Kutta integrator $q$, the bound of the MSE can be rewritten simply as

$$\mathrm{MSE}(\hat{Z}) \leq C_1 h^{2q} + C_2\frac{h^{2q}}{M}.$$

*Remark* 3.6. In [3] the authors argue that a reasonable choice for the noise scale $p$ is the order of the deterministic solver $q$, thus for a deterministic initial condition the result above is valid. This result is extremely favourable from the point of view of computational cost. Let us assume that the a tolerance $\varepsilon$ and that the numerical error is measured by means of the square root of the MSE. Then, in order to attain this tolerance we have to impose

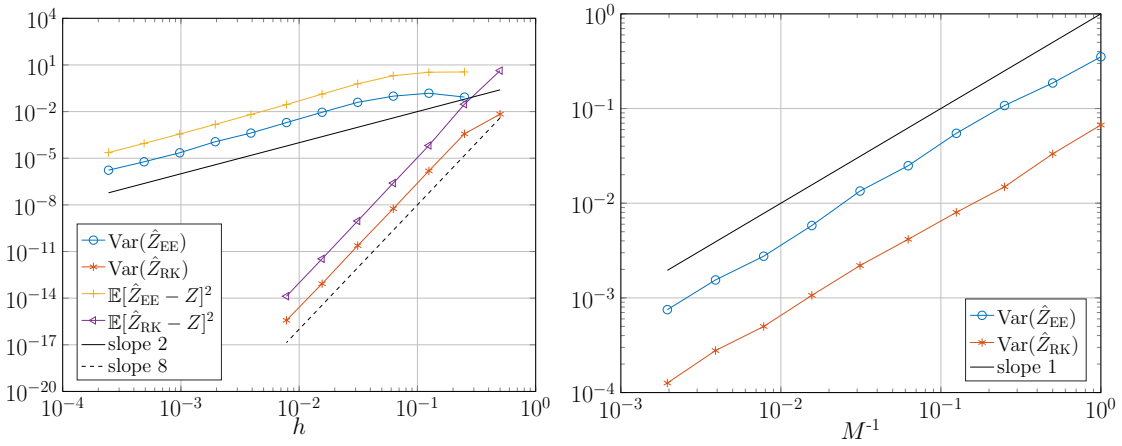
$$h = \mathcal{O}(\varepsilon^{1/q}),$$

without any condition on the number of trajectories $M$, which we can consider to be $\mathcal{O}(1)$. Hence, the computational cost is

$$\mathrm{cost} = \frac{MT}{h} = \mathcal{O}(\varepsilon^{-1/q}),$$

where we considered the final time $T$ to be $\mathcal{O}(1)$.

*Remark* 3.7. In this section we considered only the one-dimensional case ($f\colon \mathbb{R} \to \mathbb{R}$) for a matter of clarity in the notation. However, all the results above are equally valid in the multi-dimensional case without any further assumption.




**(a)** Variation of the time step.

**(b)** Variation of the number of trajectories.

**Figure 5:** Variance and squared bias of the Monte Carlo estimator $\hat{Z}$ with Explicit Euler and RK4 applied to (23). The two components of the MSE have the same order of convergence with respect to the time step $h$. Conversely, the order of convergence with respect to the number of trajectories $M$ with fixed $h$ of the variance of $\hat{Z}$ is equal to one for both methods

### 3.3.5 Numerical experiment - Monte Carlo

We consider the FitzHug-Nagumo problem introduced in (23) with the same initial conditions and parameter values and integrate it up to the final time $T = 10$ with the probabilistic integrator. We choose the function $\varphi$ to be given by $\varphi(X) = X^T X$ and generate a reference solution $Z$ with RK4 computed on a fine time step. We choose as deterministic integrator EE and RK4 and the noise scale $p$ equal to $q$, i.e., one and four respectively. We choose $M = 10$ and the time step $h = 0.5/2^i$ with $i = 0, 1, \ldots, 11$. Then we compute 300 times the estimator $\hat{Z}$ for all the values of the time step, thus estimating its variance and bias. Numerical results (Figure 5a) confirm the theoretical bound presented in Lemma 3.2, as the order of convergence of the variance of $\hat{Z}$ to zero is of order 2 and 8 with respect to $h$ for Explicit Euler and RK4 respectively independently of $M$. We perform another experiment fixing the value of $h$ to 0.5 and varying the number of trajectories in the values $M = 2^i$ with $i = 0, 1, \ldots, 9$. As in the first experiment, we compute 300 times $\hat{Z}$ in order to estimate its variance. Results (Figure 5b) show that the variance has an order equal to 1 for both the methods with respect to $M^{-1}$, thus confirming the theoretical result.

### 3.3.6 Multi-level Monte Carlo

Let us consider the case in which the initial condition is not deterministic, and therefore the variance of $U_0$ is not equal to zero. In this case, we can bound the MSE of the Monte Carlo estimator introduced in (24) as

$$\mathrm{MSE}(\hat{Z}) \leq C_1 h^{2\min\{2p,q\}} + C_2 h^{2p} + C_3 \,\mathrm{Var}(U_0) M^{-1}.$$

We can rewrite the inequality above considering three cases depending on the value of $p$

$$\mathrm{MSE}(\hat{Z}) \leq \begin{cases} C_1 h^{4p} + C_2 h^{2p} + C_3 \,\mathrm{Var}(U_0) M^{-1}, & \text{if } 2p < q, \\ C_1 h^{2q} + C_2 h^{2p} + C_3 \,\mathrm{Var}(U_0) M^{-1}, & \text{if } q \leq 2p < 2q, \\ C_1 h^{2q} + C_2 h^{2p} + C_3 \,\mathrm{Var}(U_0) M^{-1}, & \text{if } 2p \geq 2q, \end{cases}$$

Disregarding the higher order terms in $h$, we can write

$$\mathrm{MSE}(\hat{Z}) \leq \begin{cases} C_1 h^{2p} + C_3 \,\mathrm{Var}(U_0) M^{-1}, & \text{if } p < q, \\ C_1 h^{2q} + C_3 \,\mathrm{Var}(U_0) M^{-1}, & \text{if } p \geq q. \end{cases}$$

Hence, we can write in a compact form

$$\mathrm{MSE}(\hat{Z}) \leq C_1 h^{2\min\{p,q\}} + C_3 \,\mathrm{Var}(U_0) M^{-1}.$$

Let us introduce as a measure for the error, denoted by $e$, the square root of the MSE, i.e.,

$$e = \sqrt{\mathrm{MSE}(\hat{Z})}.$$

Then, in order to have $e = \mathcal{O}(\varepsilon)$, with $\varepsilon$ fixed, one has to set

$$h = \mathcal{O}(\varepsilon^{1/\min\{p,q\}}), \quad M = \mathcal{O}(\varepsilon^{-2}).$$

If we measure the cost as the product between the number of time steps and the number of trajectories, we find easily that in this case

$$\text{cost} = M \frac{T}{h} = \mathcal{O}\left(\varepsilon^{-2-1/\min\{p,q\}}\right),$$

where we assumed that the final time $T$ is $\mathcal{O}(1)$. If the required accuracy $\varepsilon$ is small, the computational cost needed to obtain an acceptable approximation is extremely big. Hence, in this case we can exploit multi-level techniques as the multi-level Monte Carlo (MLMC) [8]. The idea of MLMC is introducing a *hierarchical sampling*, introducing levels $l = 0, \ldots, L$, which have time step $h_l = T/N^l$ with $N_l = K^l$ for some integer $K$. In the following we will consider for simplicity $K = 2$, even though it is possible to choose optimal $K$ for the cost minimization. For each level, the number of trajectories is variable and is denoted by $M_l$. In the following, we will establish the number of trajectories per level minimizing the computational cost needed to obtain a Monte Carlo estimator. The estimator of $Z$ is then constructed as

$$\bar{Z} = \sum_{l=0}^{L} \frac{1}{M_l} \sum_{i=1}^{M_l} \left(\varphi_l^{(i)} - \varphi_{l-1}^{(i)}\right), \quad \varphi_l^{(i)} = \varphi\left(U_{N_l}^{(i)}\right).$$

The values $\varphi_l^{(i)}$ are constructed under two assumptions

1. $\varphi_l^{(i)}$ and $\varphi_{l-1}^{(i)}$, with $\varphi_{-1} \coloneqq 0$, are constructed using the same Brownian path,
2. $\varphi_l^{(i)}, \varphi_{l-1}^{(i)}$ and $\varphi_l^{(j)}, \varphi_{l-1}^{(j)}$ are independent for $i \neq j$.

The internal sum in $\bar{Z}$ is a telescopic sum, hence

$$\mathbb{E}(\varphi_L) = \mathbb{E}(\bar{Z}).$$

Then we can compute the MSE of $\bar{Z}$ as

$$\begin{aligned}
\mathrm{MSE}(\bar{Z}) &= \mathbb{E}\left(\bar{Z} - \varphi\left(u(T)\right)\right)^2 \\
&= \mathrm{Var}\left(\bar{Z}\right) + \left(\mathbb{E}\left(\bar{Z} - \varphi\left(u(T)\right)\right)\right)^2 \\
&= \mathrm{Var}\left(\bar{Z}\right) + \left(\mathbb{E}\left(\varphi\left(U_{N_L}\right) - \varphi\left(u(T)\right)\right)\right)^2 \\
&= \mathrm{Var}\left(\bar{Z}\right) + \mathcal{O}\left(h_L^{2\min\{2p,q\}}\right),
\end{aligned}$$

where we considered Proposition 3.3 about the weak order of the method. Exploiting the independence of the sample paths, we can compute the variance as

$$\begin{aligned}
\mathrm{Var}(\bar{Z}) &= \sum_{l=0}^{L} \frac{1}{M_l^2} \sum_{i=1}^{M_l} \mathrm{Var}(\varphi_l^{(i)} - \varphi_{l-1}^{(i)}) \\
&= \sum_{l=0}^{L} \frac{M_l}{M_l^2} \mathrm{Var}(\varphi_l - \varphi_{l-1}) = \sum_{l=0}^{L} \frac{V_l}{M_l}, \quad V_l \coloneqq \mathrm{Var}(\varphi_l - \varphi_{l-1}).
\end{aligned}$$

Thanks to Proposition 3.2 it is possible to estimate $V_l$.

**Lemma 3.4.** *If $\varphi$ is Lipschitz continuous then*

$$V_l \leq C h_l^{2\min\{p,q\}},$$

*with $C > 0$ is a constant independent of $h_l$.*

*Proof.* Let us consider the case $l = 0$. In this case

$$V_0 = \varphi_0 - \varphi_{-1} = \mathcal{O}(1),$$

as $h_0 = T$. For $l \geq 1$, thanks to (26)

$$\begin{aligned}
\mathrm{Var}(\varphi_l - \varphi_{l-1}) &= \mathrm{Var}\left(\varphi_l - \varphi\left(u(T)\right) + \varphi\left(u(T)\right) - \varphi_{l-1}\right) \\
&\leq 2\left(\mathrm{Var}\left(\varphi_l - \varphi\left(u(T)\right)\right) + \mathrm{Var}\left(\varphi_{l-1} - \varphi\left(u(T)\right)\right)\right).
\end{aligned}$$

Then, considering singularly the two terms and denoting by $K$ the Lipschitz constant of $\varphi$

$$\begin{aligned}
\mathrm{Var}\left(\varphi_l - \varphi\left(u(T)\right)\right) &\leq \mathbb{E}\left(\varphi_l - \varphi\left(u(T)\right)\right)^2 = \mathbb{E}\left(\varphi(U_{N_l}) - \varphi\left(u(T)\right)\right)^2 \\
&\leq K^2 \mathbb{E}\left(U_{N_l} - u(T)\right)^2 \\
&\leq K^2 \mathbb{E}\left|U_{N_l} - u(T)\right|^2 \leq C h_l^{2\min\{p,q\}},
\end{aligned}$$

where the last bound is given by Proposition 3.2. □

Therefore, the MSE is given by

$$\mathrm{MSE}(\bar{Z}) = C_1 h_L^{2\min\{2p,q\}} + C_2 \sum_{l=0}^{L} \frac{h_l^{2\min\{p,q\}}}{M_l}.$$

We would like those two terms to balance, therefore we choose $M_l$ as

$$M_l = \frac{h_l^{2\min\{p,q\}} L}{h_L^{2\min\{2p,q\}}},$$

as in this way

$$\mathrm{MSE}(\bar{Z}) = C_1 h_L^{2\min\{2p,q\}} + C_2 \frac{L+1}{L} h_L^{2\min\{2p,q\}} = \mathcal{O}\left(h_L^{2\min\{2p,q\}}\right).$$

Hence, if we use as a measure of the error

$$e = \sqrt{MSE\left(\bar{Z}\right)},$$

and imposing $e = \mathcal{O}(\varepsilon)$ for a fixed $\varepsilon$, we get for the finest time step

$$h_L = \mathcal{O}\left(\varepsilon^{1/\min\{2p,q\}}\right). \tag{31}$$

Let us compute the cost with this choice of the parameters. Defining the cost as the product of the number of time steps and the number of trajectories, we find

$$\mathrm{cost} = \sum_{l=0}^{L} N_l M_l = \sum_{l=0}^{L} \frac{T}{h_l} \frac{h_l^{2\min\{p,q\}} L}{h_L^{2\min\{2p,q\}}}.$$

For a matter of clarity in the computation, we consider three different cases.

**Case 1:** $q \leq p$

In this case, $\min\{p,q\} = q$ and $\min\{2p,q\} = q$. Therefore

$$\begin{aligned}
\mathrm{cost} &= \sum_{l=0}^{L} \frac{T}{h_l} \frac{h_l^{2q} L}{h_L^{2q}} = \frac{TL}{h_L} \sum_{l=0}^{L} \left(\frac{h_l}{h_L}\right)^{2q-1} \\
&= \frac{TL}{h_L} \sum_{l=0}^{L} 2^{(L-l)(2q-1)} = \frac{TL}{h_L} 2^{L(2q-1)} \sum_{l=0}^{L} 2^{-l(2q-1)} \\
&\leq L 2^{2qL} \frac{1}{1-2^{1-2q}} \leq 2L2^{2qL} = \mathcal{O}\left(L h_L^{-2q}\right),
\end{aligned}$$

where we have assumed $q \geq 1$ so that the geometric series converges. Hence, in order to satisfy $e = \varepsilon$ considering that $h_L = T/2^L$ and (31) we can impose

$$L = \left|\log_2 \varepsilon^{1/q}\right|,$$

and therefore the cost can be expressed as

$$\mathrm{cost} = \mathcal{O}\left(\left|\log_2 \varepsilon^{1/q}\right| \varepsilon^{-2}\right).$$

**Case 2:** $q \geq 2p$

In this case, $\min\{p, q\} = p$ and $\min\{2p, q\} = 2p$. Therefore

$$\begin{aligned}\text{cost} &= \sum_{l=0}^{L} \frac{T}{h_l} \frac{h_l^{2p} L}{h_L^{4p}} = \frac{TL}{h_L^{2p+1}} \sum_{l=0}^{L} \left(\frac{h_l}{h_L}\right)^{2p-1} \\ &= \frac{TL}{h_L^{2p+1}} \sum_{l=0}^{L} 2^{(L-l)(2p-1)} = \frac{TL}{h_L^{2p+1}} 2^{L(2p-1)} \sum_{l=0}^{L} 2^{-l(2p-1)} \\ &\leq \frac{L 2^{2pL}}{h_L^{2p}} \frac{1}{1 - 2^{1-2q}} = \mathcal{O}\left(L h_L^{-4p}\right),\end{aligned}$$

Hence, in view of (31) we impose as before

$$L = \left|\log_2 \varepsilon^{1/2p}\right|,$$

therefore the final expression of the cost is

$$\text{cost} = \mathcal{O}\left(\left|\log_2 \varepsilon^{1/2p}\right| \varepsilon^{-2}\right).$$

**Case 3:** $p < q \leq 2p$

In this case, $\min\{p, q\} = p$ and $\min\{2p, q\} = q$. Therefore

$$\begin{aligned}\text{cost} &= \sum_{l=0}^{L} \frac{T}{h_l} \frac{h_l^{2p} L}{h_L^{2q}} = \frac{TL}{h_L^{2q-2p+1}} \sum_{l=0}^{L} \left(\frac{h_l}{h_L}\right)^{2p-1} \\ &= \frac{TL}{h_L^{2q-2p+1}} \sum_{l=0}^{L} 2^{(L-l)(2p-1)} = \frac{TL}{h_L^{2q-2p+1}} 2^{L(2p-1)} \sum_{l=0}^{L} 2^{-l(2p-1)} \\ &\leq \frac{L 2^{2pL}}{h_L^{2q-2p}} \frac{1}{1 - 2^{1-2q}} = \mathcal{O}\left(L h_L^{2p-2q-2p}\right) = \mathcal{O}\left(L h_L^{-2q}\right).\end{aligned}$$

Hence the number of levels is given by

$$L = \left|\log_2 \varepsilon^{1/q}\right|,$$

and the computational cost is given by

$$\text{cost} = \mathcal{O}\left(\left|\log_2 \varepsilon^{1/q}\right| \varepsilon^{-2}\right).$$

Let us remark that in practice the method (16) is tuned so that $p = q$ in order to introduce a noise of the same magnitude of the numerical error. Therefore, the first of the three cases presented above is the most interesting for practical application of the method.

### 3.3.7 Stability analysis

In the previous sections we analyzed the behavior of the probabilistic integrator for ODE's in terms of its convergence with respect to the time step. Moreover, we analyzed the

convergence of a Monte Carlo approximation of the probabilistic solution towards the exact solution. Another key feature of numerical methods is their stability. We recall that the one step of the numerical method can be written, under Assumption 3.1, as

$$U_{k+1} = \Psi(U_k) + Q^{1/2} h^{p+1/2} Z_k,$$

with $Z_k$, $k = 0, \dots, N$ i.i.d. zero-mean normal random variables with unitary variance. We can write therefore the numerical method as

$$U_{k+1} = \Psi(U_k) + Q^{1/2} h^{p} \Delta W_k,$$

where the random variables $\Delta W_k$, $k = 0, \dots, N$, are standard Wiener increments. This is the stochastic Runge-Kutta method applied to the SDE

$$\mathrm{d}U(t) = f(U(t))\mathrm{d}t + Q^{1/2} h^{p} \mathrm{d}W(t).$$

This is an SDE with *additive noise*, i.e., the noise component is independent on the solution $U$. The following result concerns the $A$-stability of the stochastic Runge-Kutta methods applied to this class of equations [12, Theorem 4.1.]

**Proposition 3.5.** *The stochastic Runge-Kutta method applied to an SDE with additive noise is A-stable if and only if so is its determinist component.*

This stability results is a direct consequence of the independence of the noise component on the value of the solution itself. Therefore, if the approximation of the drift term is stable, so will be the solution of the full SDE. The $A$-stability of Runge-Kutta has been extensively analyzed [11], therefore the choice of an $A$-stable method $\Psi$ is applicable in the frame of probabilistic solvers without any restriction to obtain a family of stable probabilistic numerical solutions.

# 4 Bayesian inference of the parameters of an ODE

We now consider the probabilistic solver introduced in the previous section and study its behavior when used in the context of Bayesian inference. Let us consider $\theta$ a parameter in $\mathbb{R}^{N_p}$, a function $f_\theta \colon \mathbb{R}^d \to \mathbb{R}^d$ depending on the parameter, an element $u_0$ of $\mathbb{R}^d$ and the following ODE

$$\begin{aligned} u'_\theta(t) &= f_\theta(u_\theta(t)), \\ u_\theta(0) &= u_0, \end{aligned} \tag{32}$$

where we write explicitly the dependence of the solution $u$ on the parameter. In the following, we consider $\theta$ to be unknown a priori, and we denote by $\bar{\theta}$ its true value. Moreover, let us consider a set of observed data $\mathcal{Y}_i$ defined as

$$\mathcal{Y}_i = \{y_1, y_2, \dots, y_i\}, \quad i = 1, \dots, D,$$

where $D$ is the total number of observations. We consider the data to be a Gaussian linear function of the exact solution of (32) computed at the true value of $\theta$ at a discrete set of times $t_i$, i.e.

$$y_i = u_{\bar{\theta}}(t_i) + \varepsilon_i, \quad \varepsilon_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \Gamma), \quad i = 1, \dots, D.$$

If the solution of (32) is computable analytically, then thanks to Bayes theorem we know that once a prior distribution $\mathcal{Q}(\theta)$ is specified, the posterior distribution of $\theta$ is given by Bayes' formula and can be expressed as

$$\pi(\theta|\mathcal{Y}) \propto \mathcal{Q}(\theta)\mathcal{L}(\mathcal{Y}|u_\theta(t)). \tag{33}$$

Under the hypothesis that the observational error is normally distributed, the likelihood function is easy to compute and is given by

$$\mathcal{L}(\mathcal{Y}|u_\theta(t)) \propto \exp\left(-\frac{1}{2}\sum_{i=1}^{D} \left(u_\theta(t_i) - y_i\right)^T \Gamma^{-1} \left(u_\theta(t_i) - y_i\right)\right).$$

In most of applications, the exact solution of (32) is not computable in closed form, and therefore Bayes rule cannot be applied directly as in (33). Therefore, a MCMC technique has to be applied to obtain an estimation of the parameter.

## 4.1 Approximation of the likelihood

An unbiased estimator of the likelihood has to be obtained at each step of MH in order to compute the acceptance probability. In particular, we approximate the likelihood using the probabilistic solver with time step $h$, thus obtaining a value $\mathcal{L}_h$ such that

$$\mathcal{L}(\mathcal{Y}|\theta) \approx \mathcal{L}_h(\mathcal{Y}|\theta).$$

The value $\mathcal{L}_h$ is not directly computable, therefore we compute a Monte Carlo estimator over with $M$ realizations of the numerical solution and get the value $\mathcal{L}_h^M$ such that

$$\mathcal{L}_h^M(\mathcal{Y}|\theta) \approx \mathcal{L}_h(\mathcal{Y}|\theta).$$

We then use this value for computing the acceptance probability in the frame of a MCWM algorithm (see Section 2.3.3) to perform Bayesian inference on the value of the parameter

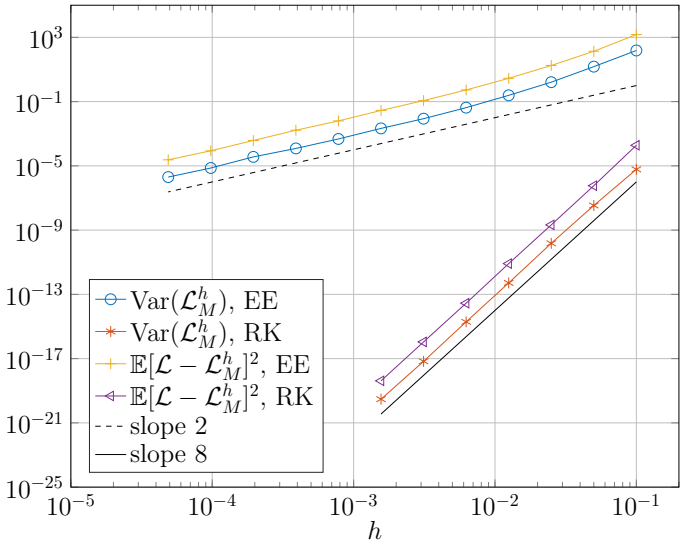



**Figure 6:** Approximation of the likelihood at a fixed value $\bar{\theta}$.

$\theta$. A question which often arises in literature [1, 4, 18] is how many samples $M$ it would be advisable to choose in order to consider the obtained posterior distribution a good approximation of the true posterior. For each value of $\theta$, we can apply Proposition 3.4, thus obtaining

$$\mathrm{MSE}(\mathcal{L}_h^M(\mathcal{Y}|\theta)) \leq C_1 h^{2\min\{2p,q\}} + \frac{C_2}{M} h^{2p}. \tag{34}$$

Therefore, at each step of the MCMC algorithm the approximation of the likelihood function depends uniquely on the time step $h$, on the order $q$ of the Runge-Kutta method employed to implement (16) and on the noise scale $p$ of Assumption 3.1.

### 4.1.1 Numerical experiment - Likelihood

We consider the FitzHug-Nagumo problem defined in (23) and the parameters fixed to the true value $\bar{\theta} = (0.2, 0.2, 3.0)$. We generate ten equispaced observations from initial time $t = 0$ to final time $T = 1$ adding a zero-mean Gaussian perturbation with variance equal to $10^{-2}$ to the two components of a numerical solution computed with a small time step. Then we generate a reference solution using the same small time step without noise in order to have an approximation of $\mathcal{L}(\mathcal{Y}|\bar{\theta})$ with negligible error. We compute 300 realizations of $\mathcal{L}_h^M(\mathcal{Y}|\bar{\theta})$ using EE or RK4 as a deterministic integrator and the noise scale $p = q$ with time steps $h$ in the set $h = 0.1 \cdot 2^{-i}$ with $i = 0, \ldots, 11$, for Euler Forward and $h = 0.1 \cdot 2^{-i}$ for $i = 0, \ldots, 6$, for RK4. Moreover, we consider $M = 10$ trajectories for all time steps, since the convergence rates with respect to $h$ should be observed independently of $M$. Hence, we can estimate the bias of $\mathcal{L}_h^M(\mathcal{Y}|\bar{\theta})$ with respect to the true value of the likelihood and its variance. Results (Figure 6) show that the bound (34) is verified in practice. Therefore, we can treat the likelihood function as any other functional $\varphi$ of the solution and apply the theoretical results presented in Section 3.3.4.

## 4.2 Numerical experiment - Posterior distributions

Let us consider the two-dimensional FitzHug-Nagumo ODE defined in (23) and the problem of determining the values of the parameters $\theta = (a, b, c)^T$ in $\mathbb{R}^3$. We consider as the true value of $\theta$ the vector $\bar{\theta} = (0.2, 0.2, 3)$. We produce a set of synthetic observations $\mathcal{Y}_{10}$ from a numerical solution $\tilde{u}$ computed using $\bar{\theta}$ and a small time step at times $t_i = 1, 2, ..., 10$, with an additive independent Gaussian noise, i.e.,

$$y_i = \tilde{u}_{\bar{\theta}}(t_i) + \varepsilon_i, \quad \varepsilon_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 10^{-2} I), \quad i = 1, \dots, 10,$$

where $I$ is the identity matrix in $\mathbb{R}^{2\times 2}$. Therefore, we consider a diagonal noise with independent normal components having all variance $10^{-2}$. We approximate the posterior distribution $\pi(\theta|\mathcal{Y})$ with both the deterministic and the probabilistic solvers using time step $h = 0.1$. We use the RAM algorithm for the deterministic case and the RAM algorithm applied to MCWM for the probabilistic integrator. In both cases, the proposal distribution $q(x, y)$ is a Gaussian with variance adapted by RAM, and the prior distribution $\mathcal{Q}(\theta)$ is normal with unitary variance and mean $\bar{\theta}$. We consider 50000 iterations of MCMC in both the deterministic and the probabilistic case, with the first 10% of guesses considered as a burn-in. Results (Figure 7) show that the posterior distribution obtained using the deterministic solver is concentrated and biased with respect to the true value of the parameter. On the other side, the probabilistic solver provides with a posterior distribution having a wider support which contains the true value of the parameter. This confirms the claim that the probabilistic solver allows to identify the source of error given by the numerical integration, while in the deterministic case this uncertainty does not result from the obtained distributions.

## 4.3 Convergence of the posterior distribution

We wish to study the convergence of the posterior distribution obtained using the probabilistic method with respect to the true posterior distribution. In the following, we consider only the distance between the target distributions given by an exact algorithm and the target distribution of the MCMC with approximated likelihood, disregarding the error due to the MCMC approximation. This assumption is rather strong, as the convergence of MCMC can be slow depending on the initial condition, the prior distribution and the proposal distribution.
In order to study the convergence, it is necessary to introduce a notion of distance between two probability measures. A standard measure is the *total variation distance*, defined in the following.

**Definition 4.1.** *Given two probability measures $\nu$ and $\mu$ on a measurable space $(\mathcal{X}, \mathcal{B}(\mathcal{X}))$, the total variation distance between $\nu$ and $\mu$ is defined as*

$$d_{\mathrm{TV}}(\nu, \mu) \coloneqq \sup_{A \in \mathcal{B}(\mathcal{X})} |\nu(A) - \mu(A)| .$$

*Moreover, if $\nu$ and $\mu$ admit densities $f$ and $g$ respectively with respect to a dominating measure $\lambda$, then the total variation distance can be expressed as*

$$d_{\mathrm{TV}}(\nu, \mu) \coloneqq \frac{1}{2} \int_{\mathcal{X}} (f(x) - g(x)) \mathrm{d}\lambda(x).$$

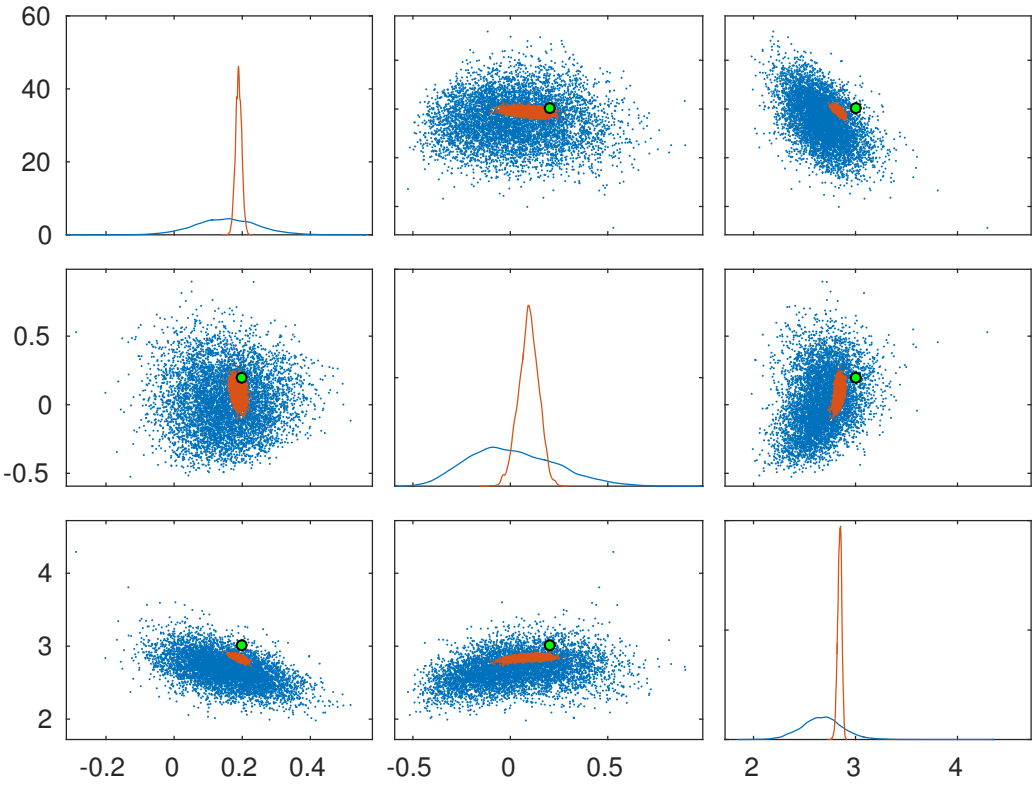



**Figure 7:** Posterior distribution for the parameter $\theta$ defining the FitzHug-Nagumo model. The posterior distributions given by the probabilistic and the deterministic solvers are displayed in blue and red respectively. The true value of the parameters is displayed in thick green dots.

Other notions of distance can be employed when the total variation distance is not practical to compute, such as the Hellinger distance, which is defined as follows [7].

**Definition 4.2.** *If $f, g$ are densities of the measures $\mu$ and $\nu$ on $(\mathbb{R}^n, \mathcal{B}(\mathbb{R}^n))$ with respect to the Lebesgue measure, the Hellinger distance between $\nu$ and $\mu$ is defined as*

$$d^2_{\text{Hell}}(\mu,\nu) \coloneqq \frac{1}{2}\int_{\mathbb{R}^n}\left(\sqrt{f(x)}-\sqrt{g(x)}\right)^2 \mathrm{d}x$$

The Hellinger distance allows us to estimate the total variation distance as the following inequalities hold [7]

$$\frac{d^2_{\text{Hell}}(\mu,\nu)}{2} \leq d_{\text{TV}}(\mu,\nu) \leq d_{\text{Hell}}(\mu,\nu). \tag{35}$$

Let us denote by $\pi_h^M$ the target distribution of the MCWM algorithm employing the probabilistic solver with time step $h$ and $M$ realizations of the numerical solution, and by $\pi$ the exact posterior. In the following, we will denote with $\pi$ and $\pi_h^M$ with a slight abuse of notation both the probability measures and their density functions. We can compute the second moment of the Hellinger distance with respect to the random variables $\xi$ as

$$\begin{aligned}
2\mathbb{E}^\xi\left[d^2_{\text{Hell}}(\pi(\theta|\mathcal{Y}),\pi_h^M(\theta|\mathcal{Y}))\right] &= \mathbb{E}^\xi\left[\int\left(\sqrt{\pi(\theta|\mathcal{Y})}-\sqrt{\pi_h^M(\theta|\mathcal{Y})}\right)^2\mathrm{d}\theta\right] \\
&= \mathbb{E}^\xi\left[\int\left(\sqrt{\mathcal{Q}(\theta)\mathcal{L}(\mathcal{Y}|\theta)}-\sqrt{\mathcal{Q}(\theta)\mathcal{L}_h^M(\mathcal{Y}|\theta)}\right)^2\mathrm{d}\theta\right] \\
&= \int\mathbb{E}^\xi\left[\left(\sqrt{\mathcal{L}(\mathcal{Y}|\theta)}-\sqrt{\mathcal{L}_h^M(\mathcal{Y}|\theta)}\right)^2\right]\mathrm{d}\mathcal{Q}(\theta) \\
&= \int\text{MSE}\left(\sqrt{\mathcal{L}_h^M(\mathcal{Y}|\theta)}\right)\mathrm{d}\mathcal{Q}(\theta) \\
&\leq \int C(\theta)h^{2q}\mathrm{d}\mathcal{Q}(\theta) \\
&= h^{2q}\int C(\theta)\mathrm{d}\mathcal{Q}(\theta) \\
&\leq h^{2q}\sup_{\theta\in\mathbb{R}^{N_p}}C(\theta)\int\mathrm{d}\mathcal{Q}(\theta) \\
&= h^{2q}\sup_{\theta\in\mathbb{R}^{N_p}}C(\theta),
\end{aligned}$$

where $C(\theta)$ is the constant appearing in Proposition 3.4. Let us remark that the constant depends on the Lipschitz constant of the function defining the ODE, which depends non-trivially on $\theta$. Finally, defining

$$\tilde{C} \coloneqq \sqrt{\frac{1}{2}\sup_{\theta\in\mathbb{R}^{N_p}}C(\theta)},$$

we get the following bound on the second moment of the Hellinger distance between the approximated posterior and the posterior obtained with the exact solution

$$\mathbb{E}^\xi\left[d^2_{\text{Hell}}(\pi(\theta|\mathcal{Y}),\pi_h^M(\theta|\mathcal{Y}))\right] \leq \tilde{C}^2h^{2q}.$$

Then, thanks to Jensen's inequality

$$\begin{aligned}\mathbb{E}^{\xi}\left[d_{\mathrm{Hell}}(\pi(\theta|\mathcal{Y}),\pi_h^M(\theta|\mathcal{Y}))\right] &\leq \mathbb{E}^{\xi}\left[d^2_{\mathrm{Hell}}(\pi(\theta|\mathcal{Y}),\pi_h^M(\theta|\mathcal{Y}))\right]^{1/2}\\ &\leq \tilde{C}h^q.\end{aligned}$$

Let us remark that thanks to (35), this bound is equally true for the total variation distance.

## 4.4 Convergence of the Monte Carlo approximation

We are now interested in the convergence of the Monte Carlo approximation of the expectation of the parameter $\theta$ given by MCMC. Let us consider a function $g\colon \mathbb{R}^{N_p} \to \mathbb{R}$ such that $g \in L^{\infty}(\mathbb{R}^{N_p})$. Then, we wish to bound the distance between the expectation of $g(\theta)$ computed with respect to the true measure and to the measure targeted by MCMC implemented with the probabilistc solver and time step $h$. Thanks to the previous result on the total variation distance we get

$$\begin{aligned}\mathbb{E}^{\xi}\left|\mathbb{E}^{\pi}\left[g(\theta)\right] - \mathbb{E}^{\pi_h^M}\left[g(\theta)\right]\right| &= \mathbb{E}^{\xi}\left|\int g(\theta)(\pi(\theta|\mathcal{Y}) - \pi_h^M(\theta|\mathcal{Y}))\mathrm{d}\theta\right|\\ &\leq \|g\|_{\infty}\mathbb{E}^{\xi}\left[\int \left|\pi(\theta|\mathcal{Y}) - \pi_h^M(\theta|\mathcal{Y})\right|\mathrm{d}\theta\right]\\ &= 2\|g\|_{\infty}\mathbb{E}^{\xi}\left[d_{\mathrm{TV}}(\pi,\pi_h^M)\right]\\ &\leq 2\|g\|_{\infty}Ch^q.\end{aligned}$$

We can now compute the variance of the expectation of $g(\theta)$ computed MCMC and the probabilistic integrator as

$$\begin{aligned}\mathrm{Var}^{\xi}(\mathbb{E}^{\pi_h^M}\left[g(\theta)\right]) &= \mathbb{E}^{\xi}\left[\left(\mathbb{E}^{\pi_h^M}\left[g(\theta)\right] - \mathbb{E}^{\xi}\left[\mathbb{E}^{\pi_h^M}\left[g(\theta)\right]\right]\right)^2\right]\\ &= \mathbb{E}^{\xi}\left[\left(\int g(\theta)\mathcal{L}_h^M(\mathcal{Y}|\theta)\mathrm{d}\mathcal{Q}(\theta) - \mathbb{E}^{\xi}\left[\int g(\theta)\mathcal{L}_h^M(\mathcal{Y}|\theta)\mathrm{d}\mathcal{Q}(\theta)\right]\right)^2\right]\\ &= \mathbb{E}^{\xi}\left[\left(\int g(\theta)\left(\mathcal{L}_h^M(\mathcal{Y}|\theta) - \mathbb{E}^{\xi}\left[\mathcal{L}_h^M(\mathcal{Y}|\theta)\right]\right)\mathrm{d}\mathcal{Q}(\theta)\right)^2\right]\\ &\leq \mathbb{E}^{\xi}\left[\int g^2(\theta)\left(\mathcal{L}_h^M(\mathcal{Y}|\theta) - \mathbb{E}^{\xi}\left[\mathcal{L}_h^M(\mathcal{Y}|\theta)\right]\right)^2\mathrm{d}\mathcal{Q}(\theta)\right]\\ &= \int g^2(\theta)\mathbb{E}^{\xi}\left[\left(\mathcal{L}_h^M(\mathcal{Y}|\theta) - \mathbb{E}^{\xi}\left[\mathcal{L}_h^M(\mathcal{Y}|\theta)\right]\right)^2\right]\mathrm{d}\mathcal{Q}(\theta)\\ &= \int g^2(\theta)\,\mathrm{Var}^{\xi}\left(\mathcal{L}_h^M(\mathcal{Y}|\theta)\right)\mathrm{d}\mathcal{Q}(\theta)\\ &\leq \left\|g^2\right\|_{\infty}Ch^{2q},\end{aligned}$$

where we applied Jensen's inequality and Proposition 3.4. Hence, the MSE of this estimation is bounded quadratically with respect to the order of the employed Runge-Kutta method, i.e.,

$$\begin{aligned}\mathrm{MSE}(\mathbb{E}^{\pi_h^M}\left[g(\theta)\right]) &= \mathbb{E}^{\xi}\left[\mathbb{E}^{\pi_h^M}\left[g(\theta)\right] - \mathbb{E}^{\pi}\left[g(\theta)\right]\right]^2 + \mathrm{Var}^{\xi}(\mathbb{E}^{\pi_h^M}\left[g(\theta)\right])\\ &\leq Ch^{2q}.\end{aligned}$$

Let us remark that the convergence rates given by these results can be verified only if the posterior distribution $\pi_h^M$ is known. Otherwise, the expectations are approximated with a Monte Carlo sum and the statistical error has to be taken into account. In the next section, we will consider the Monte Carlo approximation given by MCMC samples and present a method to estimate the statistical error.

## 4.5 Considerations on the MCMC approximation

All the results of convergence presented above regard the approximation of the true posterior distribution with the posterior distribution approximated with time step $h$. These estimations are true if we have access to these distributions. In practice, if the posterior distribution is $\pi(\cdot)$, we use a MCMC method to get the Monte Carlo approximation

$$\mathbb{E}^\pi[g(\theta)] \approx \frac{1}{N}\sum_{i=1}^{N} g(\theta^{(i)}),$$

where $\theta^{(i)}$, $i = 1, \dots, N$, are the samples given by MCMC. Let us denote by $\hat{g}(\theta)$ the Monte Carlo estimator, i.e.,

$$\hat{g}(\theta) = \frac{1}{N}\sum_{i=1}^{N} g(\theta^{(i)}).$$

It is meaningful to study the relation between $\hat{g}(\theta)$ and $\mathbb{E}^\pi[g(\theta)]$ with respect to the number of samples $N$. The main issue is given by the Markov chain property of the samples. In standard Monte Carlo, the samples are independent and therefore estimating their mean and variance is straightforward, as well as studying the asymptotic properties with the central limit theorem. A central limit theorem is available for Markov chains under regularity assumptions on the Markov kernel [13], i.e., with the notation introduced above

$$\sqrt{N}(\hat{g}(\theta) - \mathbb{E}^\pi[g(\theta)]) \xrightarrow{d} \mathcal{N}(0, \sigma_g^2), \tag{36}$$

where the convergence is in the distributional sense and we can define the variance of the Monte Carlo estimator as

$$\sigma_g^2 \coloneqq \mathrm{Var}^\pi(g(\theta^{(0)})) + 2\sum_{i=1}^{\infty} \mathrm{Cov}^\pi(g(\theta^{(0)}), g(\theta^{(i)})) < \infty.$$

In general, it is not possible to compute analytically $\sigma_g$, but many techniques have been proposed in order to estimate it from the Markov chain itself. A common method [5] is to compute the *batch means* of the Markov chain and use them to estimate $\sigma_g^2$.
Let us suppose that there exist two integers $a_N, b_N$ such that $N = a_N b_N$. Then we define the following $a_N$ non-overlapping means of subsequences contained in the Markov chain of length $b_N$

$$\bar{g}_k(\theta) = \frac{1}{b_N}\sum_{i=1}^{b_N} g(\theta^{(kb_N+i)}), \quad k = 0, \dots, a_N - 1.$$

Finally, we can build an estimator of $\sigma_g^2$ computing the population variance of the $\bar{g}_k$ with respect to $\hat{g}$, i.e.,

$$\hat{\sigma}^2_{\mathrm{BM}} = \frac{b_N}{a_N - 1}\sum_{k=0}^{a_N-1} (\bar{g}_k - \hat{g})^2.$$

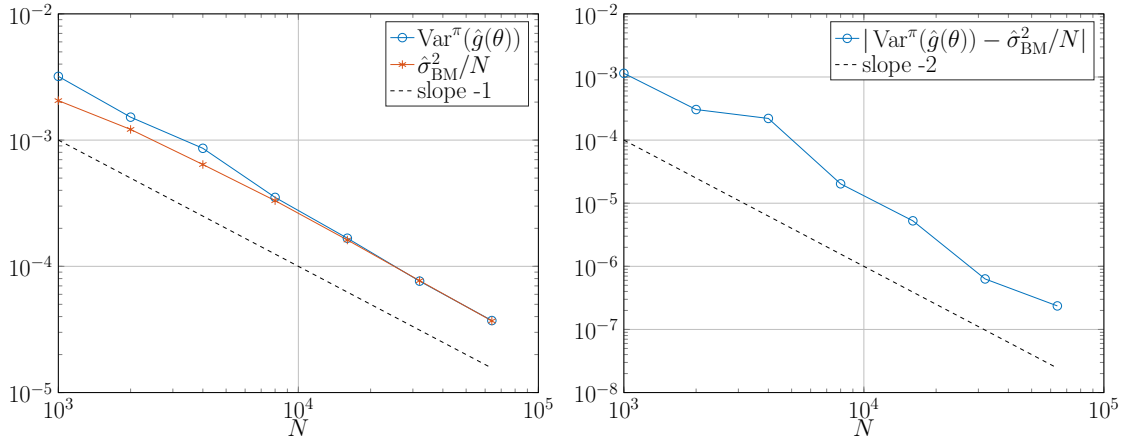



**Figure 8:** Variance of the Monte Carlo approximation given by MCMC with respect to the number of samples $N$. The convergence to zero is linear with respect to the number of samples. The estimator given by the batch means method converges to the true value of the variance with a quadratic order.

Under regularity conditions on the posterior distribution $\pi$ and the transition kernel $P$ defining the Markov chain, and if $a_N$ and $b_N$ are growing functions of $N$ such that $a_N \to \infty$, $b_N \to \infty$ if $N \to \infty$, with $b_N/N \to 0$, it is possible to show [5] that this estimator is *mean-square consistent*, i.e.,

$$\mathrm{MSE}(\hat{\sigma}^2_{\mathrm{BM}}) \overset{N\to\infty}{\to} 0.$$

In [5], the authors suggest to choose $b_N$ to be the square or cubic root of $N$ for accelerating the convergence of the MSE to zero, i.e.,

$$b_N = \lfloor N^{1/3} \rfloor, \quad \text{or,} \quad b_N = \lfloor N^{1/2} \rfloor.$$

We can deduce from the limit in (36) that the standard deviation of the MCMC estimator decreases asymptotically linearly with respect to $N^{1/2}$. Therefore, in order to observe the convergence rates with respect to $h$ presented in Section 4.4, the number of samples of the drawn in the MCMC algorithm will have to be considerably high.

### 4.5.1 Numerical experiment - Batch means estimator

In this experiment we wish to verify numerically the validity of the batch mean estimator presented above. Therefore, we consider the FitzHug-Nagumo model (23) with ten observations $\mathcal{Y}_{10}$ equispaced in the time span $0 \leq t \leq 10$. We run MCMC algorithms with approximated likelihood with $h = 0.1$ and growing number of iterations $N = 10^3 \cdot 2^i$, with $i = 0, 1, \ldots, 6$, and equal starting guesses. For each value of $N$, we perform 400 repetitions of the Markov chain and compute the population variance of the estimator $\hat{g}(\theta)$ as a function of $N$ in order to obtain an estimation of $\sigma_g^2$, where the function $g$ we consider in this experiment is $g(\theta) = \theta^T\theta$. We then compute the batch mean estimator $\hat{\sigma}_{\mathrm{BM}}$ using $b = \lfloor N^{1/2} \rfloor$ for each repetition and average the results in order to obtain a function of the number of samples $N$. Results (Figure 8) show that the variance $\sigma_g^2$ converges as predicted linearly with $N$. Moreover, the estimation provided by $\hat{\sigma}_{\mathrm{BM}}$ is coherent with the theoretical results. We remark that the batch mean estimator seems to converge to

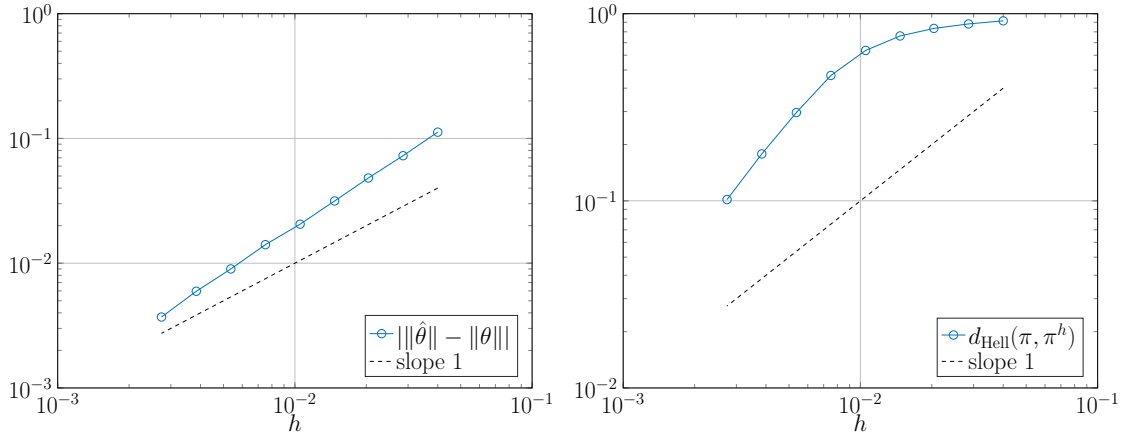



**Figure 9:** Convergence of the parameter to its stationary value and of the Hellinger distance of the probability distributions.

the true value of the sample variance with a quadratic order with respect to the number of samples. Further theoretical and literature investigations will be necessary to study this behavior. Hence, we can conclude that the batch means method provides a consistent estimation of the variance, without running several repetitions of the chains to verify their variance.

## 4.6 Numerical experiment - Convergence of the posterior distribution

We consider the FitzHug-Nagumo model (23) and produce observations $\mathcal{Y}$ at times $t_i = i$ for $i = 1, \dots, 10$ from a reference solution with additive noise with variance $10^{-2}$. We produce a reference posterior distribution using the result of a MCMC algorithm obtained with a small time step $h$. We then vary $h$ in order to observe the convergence of the posterior distribution towards the reference, as well as the convergence of the Monte Carlo estimation. We consider 100000 iterations of RAM applied to MCWM. As discussed above, the number of trajectories $M$ used to approximate the numerical likelihood does not have an influence on the convergence rate of the posterior distribution to the true posterior. Therefore, we just fix $M$ to be equal to one. We consider as the function $g$ of the parameter the Euclidean norm, therefore we consider the approximation

$$\|\theta\| \approx 10^{-6} \sum_{i=1}^{10^6} \left\| \theta^{(i)} \right\|.$$

Results (Figure 9) show the convergence obtained averaging 10 realizations of the entire MCMC chain used in order to simulate the expectation with respect to the random variable $\sigma$.

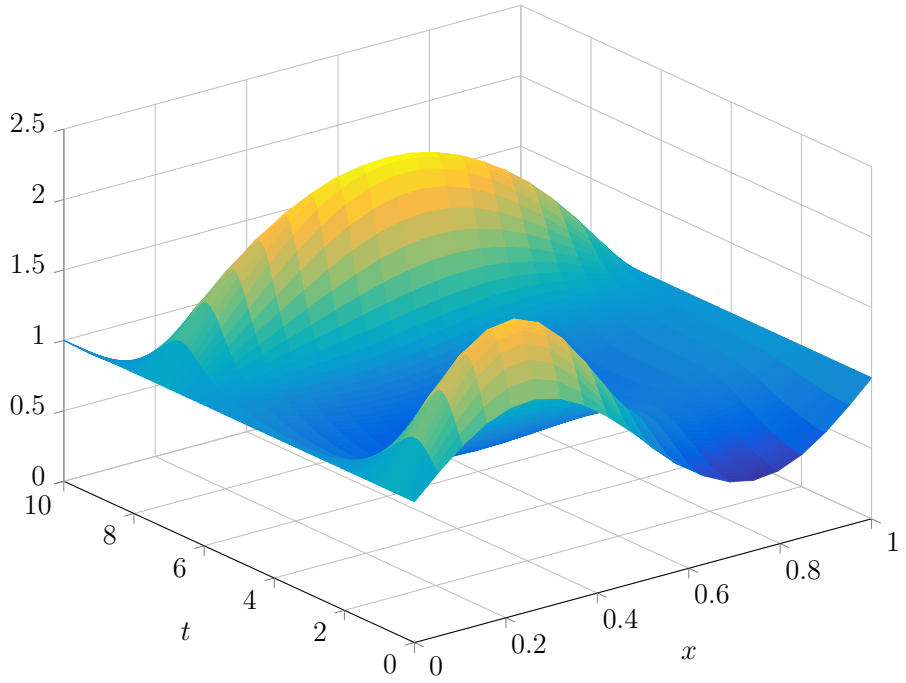



**Figure 10:** Solution of the Brusselator problem, $u$ species.

### 4.7 Stiff ODE's - The Brusselator problem

Let us consider the following parabolic PDE

$$\begin{aligned}
\frac{\partial u}{\partial t} &= 1 + u^2 v + \alpha \frac{\partial^2 u}{\partial x^2}, \quad u = u(x,t) \\
\frac{\partial v}{\partial t} &= 3u - u^2 v + \alpha \frac{\partial^2 v}{\partial x^2}, \quad v = v(x,t), \quad x \in \Omega = (0,1), \quad t \geq 0 \\
u(0,t) &= u(1,t) = 1, \\
v(0,t) &= v(1,t) = 3, \\
u(x,0) &= 1 + \sin(2\pi x), \\
v(x,0) &= 3,
\end{aligned}$$

where $\alpha$ is a positive parameter. The equation is the Brusselator problem [11], modeling the quantity of two substances $u$ and $v$ in a chemical reaction. Let us consider a spatial discretization of the domain $\Omega$ on equispaced points $x_i$, where $i = 0, \ldots, N+1$, with distance $\Delta x = 1/(N+1)$. Then, the PDE above can be transformed to a system of ODE's with the method of lines, which yields for the internal points

$$\begin{aligned}
u_i &= 1 + u_i^2 v_i - 4u_i + \frac{\alpha}{\Delta x^2}(u_{i-1} - 2u_i + u_{i+1}), \\
v_i &= 3u_i - u_i^2 v_i + \frac{\alpha}{\Delta x^2}(v_{i-1} - 2v_i + v_{i+1}), \qquad i = 1, \ldots, N.
\end{aligned} \tag{37}$$

The boundary conditions are then retrieved imposing

$$\begin{aligned} u_0(t) &= u_{N+1}(t) = 1, \\ v_0(t) &= v_{N+1}(t) = 3, \\ u_i(0) &= 1 + 0.5\sin(2\pi x_i), \quad i = 1, \dots, N, \\ v_i(0) &= 3, \qquad\qquad\qquad\quad i = 1, \dots, N. \end{aligned}$$

The solution $u(x,t)$ obtained solving numerically (37) for time $0 \leq t \leq 10$ is displayed in Figure 10. Let us consider $\alpha$ as an unknown parameter and the problem of inferring its value with the MCMC techniques explained in the previous sections. Moreover, let us consider as admissible values for $\alpha$ the interval $I_\alpha = [0, \alpha_{\max}]$. Therefore, we apply the technique explained in Section 2.3.4, with a truncated Gaussian distribution as the proposal distribution of MH. The ODE (37) is stiff for large values of $N$, and therefore using an explicit as the deterministic component in (16) like, for example, EE or RK4, yields restrictions on the time step. In particular, it is possible to prove by linearization and considering the eigenvalues of the discrete Laplacian that the stiffness index $\lambda$ of (37) is given by

$$\lambda = 4\alpha(N+1)^2.$$

Therefore, if we consider the time step restriction for EE and since $\alpha$ is bounded we have

$$h < \frac{1}{8\alpha(N+1)^2} \leq \frac{1}{8\alpha_{\max}(N+1)^2} =: \bar{h}$$

Hence, if we use EE in (16) in MCMC we have to approximate the likelihood with time step $\bar{h}$, which yields a computational cost per iteration of given by

$$\text{cost}_{\text{EE}} = \frac{T}{h} = 8T\alpha_{\max}(N+1)^2,$$

where we measure the cost in terms of number of evaluations of the function $f$ defining the ODE. Since in most of MCMC applications the number of iterations required to have an accurate Monte Carlo estimation is in the order of magnitude of $\mathcal{O}(10^4)$ to $\mathcal{O}(10^6)$, the computational cost required to perform the whole algorithm is extremely high. In order to have a lower computational cost, we can use a stabilized method like RKC. Let us recall that given a time step $h$, RKC is stable if the number of stages satisfies

$$s \geq \max\left\{2, \left\lceil \sqrt{\frac{1}{2}h\lambda} \right\rceil \right\}.$$

Therefore, we can ensure the stability of the method for each of the admissible values $\alpha$ in $I_\alpha$ setting the number of stages $s$ to of RKC to be equal to $\bar{s}$ defined as

$$\bar{s} = \max\left\{2, \left\lceil \sqrt{2h\alpha_{\max}(N+1)^2} \right\rceil \right\}. \tag{38}$$

In this case, the number of function evaluations per iteration of MCMC is given by

$$\text{cost}_{\text{RKC}} = \frac{T}{h}\bar{s} = \frac{T}{h}\max\left\{2, \left\lceil \sqrt{2h\alpha_{\max}(N+1)^2} \right\rceil \right\}. \tag{39}$$

Let us remark that in practice we can keep the time step fixed and adapt the number of stages with respect to the current guess of the parameter $\alpha$ in MCMC. Therefore, the

| Method | EE | RKC | | | |
|---|---|---|---|---|---|
| $h$ | $4.98 \cdot 10^{-7}$ | $10^{-1}$ | $10^{-3}$ | $10^{-5}$ | $10^{-7}$ |
| $s$ | - | 225 | 23 | 3 | 2 |
| cost | $2 \cdot 10^7$ | $2.25 \cdot 10^4$ | $2.3 \cdot 10^5$ | $3 \cdot 10^6$ | $2 \cdot 10^8$ |

**Table 2:** Theoretical number of function evaluations required by EE and RKC per iteration of MCMC.

| $h$ | 0.2 | 0.1 | 0.05 | 0.025 | 0.0125 |
|---|---|---|---|---|---|
| $\hat{\alpha}$ | 0.02509 | 0.020448 | 0.01996 | 0.02055 | 0.01990 |
| mean $s$ | 11.89 | 7.87 | 6.00 | 4.92 | 4.00 |
| mean cost | $0.59 \cdot 10^3$ | $0.79 \cdot 10^3$ | $1.20 \cdot 10^3$ | $1.97 \cdot 10^3$ | $3.20 \cdot 10^3$ |
| $\bar{s}$ (38) | 64 | 46 | 32 | 23 | 16 |
| max cost (39) | $3.2 \cdot 10^3$ | $4.6 \cdot 10^3$ | $6.4 \cdot 10^3$ | $9.2 \cdot 10^3$ | $12.8 \cdot 10^3$ |

**Table 3:** Summary of numerical results for the Brusselator problem.

expression (39) is an upper bound of the actual number of function evaluations required by RKC. Let us consider the spatial discretization to be defined by $N = 500$, $\alpha_{\max} = 1$. In this case, the value of $\bar{h}$ for EE is $4.9801 \cdot 10^{-7}$. We can now consider the time step of RKC to assume values $10^{-1}, 10^{-3}, 10^{-5}, 10^{-7}$ and compare the computational cost for the two methods. In Table 2 we show the computational cost required by the two methods, as well as the number of stages $s$ required by RKC in order to ensure stability with respect to the time step $h$. It is possible to remark that in this application RKC requires a computational cost up to four orders of magnitude smaller than EE.

#### 4.7.1 Numerical experiment - Brusselator

We consider the Brusselator problem with $N = 100$ and the true value of $\alpha$ equal to 0.02 and generate synthetic observations $\mathcal{Y}_{10}$ at times $t_i = 1, 2, \dots 10$ employing RK4 with a fine time step with observational variance 0.01. We consider the interval of admissible value for the parameters to be $I_\alpha = [0, 1]$ and perform 50000 iterations of the MCWM algorithm with RAM. At each iteration, we consider $M = 1$ trajectory of the probabilistic solver with RKC as determinstic component and time step $h$ variable in the set $\{0.2, 0.1, 0.05, 0.025, 0.0125\}$. The starting value for all chains is $\alpha^{(0)} = 0.5$. We adapt the number of stages in the MCMC algorithm with (38), so that the numerical method is stable. In Table 3 we show the Monte Carlo estimation $\hat{\alpha}$ of the parameter for all values of $h$, together with the mean number of stages required to perform the integration, as well as the mean computational cost per iteration. It is possible to remark that the mean number of stages required required by RKC is sensibly lower than the theoretical bound (38), as well as the computational cost. This is due to the fact that the upper bound $\alpha_{\max}$ of the interval $I_\alpha$ is never reached by the MCMC algorithm and the accepted values of the Markov chain are close to the true value of $\alpha$.

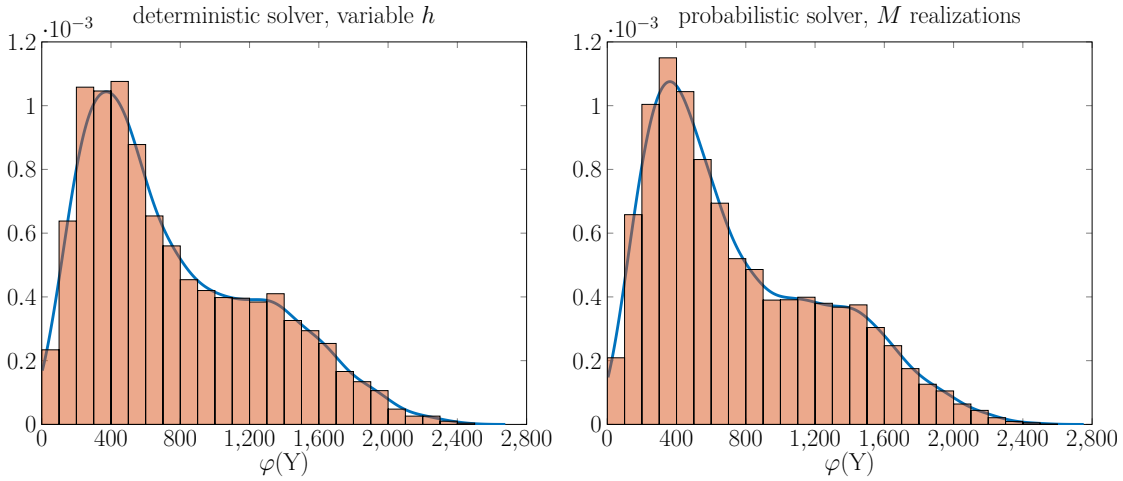



**Figure 11:** Distribution of the value of $\varphi(U_N)$ obtained with variable time step $h$ and the deterministic solver, and with $M$ realizations of the probabilistic solver fixed $h$. The histogram represents the normalized occurrences of the value of $\varphi(\cdot)$ applied to the numerical solution, while the thick blue line represents the fitted empirical density function.

## 5 Conclusions and future work

In this work we considered the probabilistic method (16) that has been recently introduced to solve ODE's. We reported the properties of weak and strong convergence, and analyzed the impact of the variation of time step and number of trajectories on Monte Carlo approximations. Moreover, we considered Bayesian inverse problems involving ODE's and showed how to effectively implement MCMC algorithms using (16), investigating theoretically and with numerical experiments how posterior distributions and Monte Carlo approximations are influenced by an approximation of the likelihood function. Finally, we showed how these methods apply in case of stiff problems arising from the spatial discretization of parabolic PDE's, performing a theoretical and practical cost analysis involving the Brusselator problem.

We believe future research will have to regard the motivation behind probabilistic solver, i.e., a classification of the ODE's for which it is relevant to perform a probabilistic rather than deterministic analysis. In particular, we believe chaotic systems of differential equations should be considered with the utmost attention, as they present non-trivial distributions of the long-term numerical solution. Let us consider for example the Lorenz system (19), with initial condition and parameter values given in section 3.2 chosen so that the system has a chaotic behavior. As discussed above, fixing a time step $h$ in this case and integrating up to final time could result in a solution which is not punctually correct due to the sensibility of the system on perturbations.

Given a time step $h$, let us denote by $U_k(h)$ the numerical solution at time $t_k = kh$ obtained using a deterministic Runge-Kutta method, and by $U_k^{\xi}(h)$ the solution obtained with (16) at the same time, where we write explicitly the dependence on the time step. Moreover, let us consider a maximal time step $h_{\max}$, which could be given for example by a stability limit time step for the chosen Runge-Kutta method, and a minimal time step $h_{\min}$. By varying the time step $h$ in the range $[h_{\min}, h_{\max}]$ and integrating with the

deterministic Runge-Kutta method the equation up to a fixed final time $T$ we obtain a distribution of numerical solutions $U_N(h)$, with $N = Th^{-1}$, which is non-trivial thanks to the chaotic behavior of the differential equation. On the other side, we can integrate the equation fixing the time step to a value $\bar{h}$ and sampling $M$ times the probabilistic solver, thus obtaining a distribution of numerical solutions $U_N^{\xi^{(i)}}(\bar{h})$, with $N = T\bar{h}^{-1}$ and $i = 1, \dots, M$. An interesting research topic will be analyzing whether the two obtained distributions describe the same behavior of the ODE.

We perform a first numerical experiment by choosing $5 \cdot 10^3$ values of the time step for the deterministic MP in the range $[1.68 \cdot 10^{-4}, 2.5 \cdot 10^{-2}]$, as well as $10^4$ realizations of the probabilistic solver implemented with MP and time step $10^{-3}$. We then integrate the Lorenz equation up to final time $T = 200$ with both methods, thus considering the distribution of the functional $\varphi(U) = U^T U$ applied to the numerical solution. Results (Figure 11) show that the two methods lead to empirical distributions with almost identical shape, thus sustaining our hypothesis that the probabilistic solver could be employed to quantify the uncertainty due to the chaotic behavior of the ODE.

## References


[1] C. Andrieu, A. Doucet, and R. Holenstein, *Particle Markov chain Monte Carlo methods*, J. R. Stat. Soc. Ser. B. Stat. Methodol., (2010), pp. 269 – 342.

[2] C. Andrieu and G. O. Roberts, *The pseudo-marginal approach for efficient Monte Carlo computations*, Ann. Statist., 37 (2009), pp. 697–725.

[3] P. R. Conrad, M. Girolami, S. Särkkä, A. Stuart, and K. Zygalakis, *Statistical analysis of differential equations: introducing probability measures on numerical solutions*, Stat. Comput., (2016).

[4] A. Doucet, M. K. Pitt, G. Deligiannidis, and R. Kohn, *Efficient implementation of Markov chain Monte Carlo when using an unbiased likelihood estimator*, Biometrika, (2015), pp. 1 – 19.

[5] J. M. Flegal and G. L. Jones, *Batch means and spectral variance estimators in Markov chain Monte Carlo*, Ann. Statist., 38 (2010), pp. 1034–1070.

[6] A. Gelman and K. Shirley, *Inference from simulations and monitoring convergence*, in Handbook of Markov chain Monte Carlo, S. Brooks, A. Gelman, G. J. Jones, and X.-L. Meng, eds., CRC press, 2011, ch. 6, pp. 163–174.

[7] A. L. Gibbs and F. E. Su, *On choosing and bounding probability metrics*, Int. Stat. Rev., 70 (2002), pp. 419–435.

[8] M. B. Giles, *Multilevel Monte Carlo path simulation*, Operations Research, 56 (2008), pp. 607–617.

[9] W. R. Gilks, *Markov chain Monte Carlo*, Encyclopedia of Biostatistics, 4 (2005).

[10] E. Hairer, C. Lubich, and G. Wanner, *Geometric numerical integration: structure-preserving algorithms for ordinary differential equations*, Springer-Verlag, Berlin and New York, 2002.

[11] E. Hairer and G. Wanner, *Solving ordinary differential equations II. Stiff and differential-algebraic problems*, Springer-Verlag, Berlin and Heidelberg, 1996.

[12] D. B. Hernandez and R. Spigler, *A-stability of Runge-Kutta methods for systems with additive noise*, BIT, 32 (1992), pp. 620–633.

[13] G. L. Jones, *On the Markov chain central limit theorem*, Probab. Surveys, 1 (2004), pp. 299–320.

[14] J. Kaipio and E. Somersalo, *Statistical and Computational Inverse Problems*, Applied Mathematical Sciences, 160, Springer, 2005.

[15] H. Kersting and P. Hennig, *Active uncertainty calibration in Bayesian ODE solvers*, in Proceedings of the 32nd Conference on Uncertainty in Artificial Intelligence (UAI 2016), AUAI Press, 2016, pp. 309–318.

[16] P. E. Kloeden, E. Platen, and H. Schurz, *Numerical solution of SDE through computer experiments*, Universitext, Springer-Verlag, Berlin, 1994.

[17] F. J. Medina-Aguayo, A. Lee, and G. O. Roberts, *Stability of noisy Metropolis–Hastings*, Stat. Comput., 26 (2016), pp. 1187–1211.

[18] M. K. Pitt, R. dos Santos Silva, P. Giordani, and R. Kohn, *On some properties of Markov chain Monte Carlo simulation methods based on the particle filter*, J. Econometrics, 171 (2012), pp. 134–151.

[19] P. van der Houwen and J. Kok, *Numerical solution of a minimax problem*, tech. rep., Mathematical Centre, Amsterdam, 1971. Report TW 124/71.

[20] M. Vihola, *Robust adaptive Metropolis algorithm with coerced acceptance rate*, Stat. Comput., 22 (2012), pp. 997–1008.