1 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.

1.1 Bayes' formula

The basis of Bayesian statistics can be found in the simple Bayes' formula. Let us consider an event space Ω, 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 Ω, 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 π for the posterior distributions, thus obtaining

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

1.2 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 θ the parameter driving the experiment and by X a random variable representing its outcome. Let us consider for simplicity θ as a vector of $\mathbb{R}^{N_p}$, where N_p is the dimension of the parameter space. We will then denote by θ_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 θ, 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 ε. 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 θ such that its true value falls in the interval up to a threshold probability.

In the Bayesian frame the estimation of θ 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 $\mathcal{L}(Y|\theta)$ 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.

1.2.1 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 λ, μ are real parameters and we consider 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 λ which is often referred to as the *drift* and the diffusion coefficient μ 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 λ is known a priori,
- the diffusion coefficient μ 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, $i = 1, \dots, N_d$, such that $t_{N_d} = T$, with a zero-mean additive Gaussian measurement noise ε, i.e., the observations y_i, are given by

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

  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 μ. In a Bayesian frame, this corresponds to providing a distribution π 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 μ 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 μ. Furthermore, if the prior distribution $\mathcal{Q}$ admits a density in closed form, it is possible to evaluate it on any choice of μ. Therefore, it is possible to compute for each value of μ 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.

1.3 Markov chain Monte Carlo methods

Markov chain Monte Carlo methods (MCMC) are a class of techniques used to perform Bayesian analyses [6, 7]. 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 θ 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 θ 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 θ 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 θ 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 θ 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 θ. 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

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 ϑ 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 ϑ, set $\theta_{i+1} = \vartheta$;
  else
    Set $\theta^{(i+1)} = \theta^{(i)}$;
  end
end

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 [7], 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.

1.3.1 Metropolis-Hastings algorithm

In this paragraph we will introduce one of the most successful MCMC methods, 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 distribution in its second argument. Given the current guess $\theta^{(i)}$, MH proposes the new element of the Markov Chain drawing a value ϑ 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_{MH} representing the move made by MH from $\theta^{(i)}$ to $\theta^{(i+1)}$ is given by [9]

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

where δ_x is the Dirac delta centered in x and ρ 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_{MH} is equivalent to stating that the new guess ϑ generated from the proposal distribution is accepted with probability α and rejected with probability $1-\alpha$. Imposing that P_{MH} satisfies the hypotheses that guarantee the convergence of MCMC, 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 to the

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 ϑ, set $\theta_{i+1} = \vartheta$;
  else
    Set $\theta^{(i+1)} = \theta^{(i)}$;
  end
  Compute S_{i+1} as in (8) ;
end

user of a MH algorithm 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 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 Σ 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.

1.3.2 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 ϑ that are included in the Markov chain $\{\theta^{(i)}\}_{i \geq 0}$. In the MH frame, the acceptance ratio depends on 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. If the initial proposal distribution does not provide with acceptable values ϑ, it may be necessary to tune it during the advancement of MH. An algorithm which targets this issue is the robust adaptive Metropolis (RAM) [12].

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

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

where Σ 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 rate is asymptotically equal to a given value α^* [12]. 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

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

Hence, we can compute S_{n+1} as the Cholesky factorization of the right hand side. Let us remark that this update has to be performed at each iteration of RAM, both in case ϑ is accepted and rejected. The sequence $\{\eta_n\}_{n \geq 1}$ can be any sequence decaying to zero with n. In this work, we consider

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

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. In Algorithm 2 we give the pseudo-code for the RAM update. 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.

Let us consider a two-dimensional real random variable X whose distribution has 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. We then consider a real value σ 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) 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 σ the samples we obtain are equally good, with an acceptance ratio near to α^* (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)

.

1.3.3 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 1.2.1 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) [1]. The main idea of the proposed pseudo-marginal algorithms is modifying the target of the algorithm to a

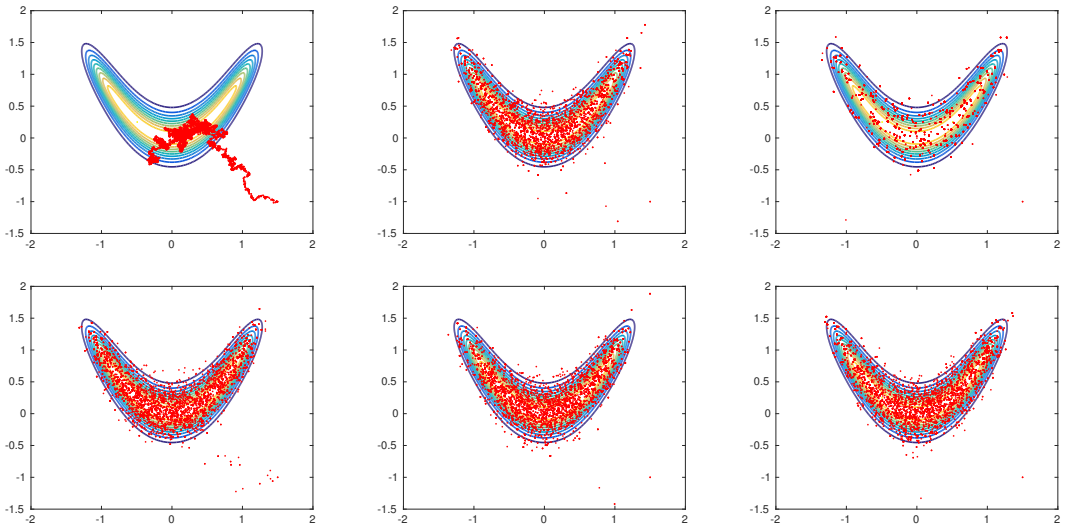

Figure 1: Samples produced by MH and RAM for the distribution (9). The contour lines of the density function are plotted for all the sets of results. In the first row we show the results obtained with MH for a normal update with covariance $\Sigma = \sigma^2 I$ with $\sigma = \{0.01, 0.5, 2.0\}$ from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

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 ξ. The acceptance probability α_M has then the same form of α 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 [9]. 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 θ 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 θ. It has been shown [2, 9] that under appropriate assumptions the following properties are valid

1. the transition kernel P_M given by (11) converges to an invariant distribution π_M with 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 π_M obtained with MCWM converges to the true marginal distribution π if M tends to infinity [9, Theorem 4.1],

3. under stronger assumptions, it is possible to obtain convergence rates of π_M to π with respect to M [9, Theorem 4.2 and Proposition 4.1].

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 ϑ 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 ϑ, set $\theta_{i+1} = \vartheta$;
  else
    Set $\theta^{(i+1)} = \theta^{(i)}$;
  end
end

Let us consider the example provided in Section 1.2.1. 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 [8] that we can apply together with a Monte Carlo approximation to compute an estimator of the likelihood, thus obtaining a value π_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.

1.3.4 How to deal with inadmissible parameter values

Let us consider without loss of generality a one-dimensional real parameter θ 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 ϑ 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 ϑ is generated by $q(\theta^{(i)}, \cdot)$, which is a truncated Gaussian distribution of mean $\theta^{(i)}$ and fixed variance σ. 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 σ. In (12) the function φ is defined as

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

and Φ 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 one. Let us remark that this proposal distribution is not symmetric, therefore α 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 θ. In this case, thanks to the symmetry properties of the function Φ, the acceptance probability α 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.

2 Probabilistic Methods

Several methods have been developed to integrate an Ordinary Differential Equation (ODE) numerically (...). If a deterministic solver is employed, the error introduced by the numerical method is however difficult to quantify (...). In the frame of statistical analyses, quantifying the impact on the uncertainty of the solution due to the numerical approximation is of the utmost importance (...). Therefore, a new class of probabilistic numerical methods have been recently proposed [3] (...).
Let us consider $f\colon \mathbb{R}^d \to \mathbb{R}^d$ and the following ODE

$$\begin{aligned} \frac{\mathrm{d}u(t)}{\mathrm{d}t} &= f(u), \quad t \in (0, T], \\ u(0) &= u_0, \qquad u_0 \in \mathbb{R}^d. \end{aligned} \tag{13}$$

Integrating numerically (13) with a deterministic method on a time discretization $t_k = kh, k = 0, \dots, N, T = Nh$ gives a numerical solution $U_k, k = 0, \dots, N$, defined by

$$\begin{aligned} U_{k+1} &= \Psi(U_k), \quad k = 0, \dots, N-1, \\ U_0 &= u_0, \end{aligned} \tag{14}$$

where Ψ defines one step of a deterministic numerical method to integrate (13).
The idea behind probabilistic methods is adding at each step of the numerical integration a noise component, *i.e.*,

$$\begin{aligned} U_{k+1} &= \Psi(U_k) + \xi_k(h), \\ U_0 &= u_0, \end{aligned} \tag{15}$$

where $\xi_k(h)$ are i.i.d. Gaussian random variables.

2.1 Deterministic methods

We choose the method defined in (14) within the class of the Runge-Kutta methods, since they have been extensively analyzed theoretically and numerically, providing results which can be exploited in the following of this work.
In particular, a method of fourth order is the classic explicit Runge-Kutta method (RK4). It is defined by the following Butcher table

0				
1/2	1/2			
1/2	0	1/2		
1	0	0	1	
	1/6	1/3	1/3	1/6

2.2 Method derivation (M. motivation)

2.3 Strong convergence

In this section we prove a result about strong convergence of the method defined in (15). The following discrete Gronwall lemma is needed in the proof.

Proposition 2.1 (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

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

Two assumptions are necessary to prove the strong convergence result. The first assumption is on the noise model.

Assumption 2.1.

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

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

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

where $p \geq 1$.

Let us remark that if $Q = \sigma I$, with I the identity matrix in $\mathbb{R}^{d\times d}$ and $\sigma > 0$, the method (15) 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 $Z_{k,i} \sim \mathcal{N}(0,1), i = 1,\dots,d$.
An assumption on the numerical method is needed.

Assumption 2.2. *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 Ψ_h has uniform local truncation error of order $q+1$*

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

The following result hold.

Proposition 2.2 (Strong Convergence)**.** *Under assumptions 2.1 and 2.2 it follows that there is $K > 0$ such that*

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

Furthermore

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

This result implies that a reasonable choice in p of Assumption 2.1 is $p = q$.

Proof. Given the method in (15) and writing the exact solution of (13) 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 2.1

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

Developing the square and since Φ_h is Lipschitz continuous with constant $(1+Lh)$ and $\epsilon_k = \mathcal{O}(h^{q+1})$ thanks to Assumption 2.2

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

Then, using Cauchy-Schwarz on the inner product

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

Therefore by Proposition 2.1

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

□

2.4 Weak convergence

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 (13) and the numerical method (15). Using the Lie derivative notation, it is possible to find the differential operators $\mathcal{L}$ and $\mathcal{L}^h$ such that for all $\varphi \in \mathcal{C}^\infty(\mathbb{R}^d, \mathbb{R})$

$$\begin{aligned}\varphi(\Phi_h(u)) &= \left(e^{h\mathcal{L}}\varphi\right)(u),\\ \mathbb{E}\varphi(U_1|U_0 = u) &= \left(e^{h\mathcal{L}^h}\varphi\right)(u).\end{aligned} \tag{16}$$

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

$$\frac{\mathrm{d}\hat{u}}{\mathrm{d}t} = f^h(\hat{u}),$$

and a modified SDE

$$\mathrm{d}\tilde{u} = f^h\tilde{u}\,\mathrm{d}t + \sqrt{h^{2p}Q}\,\mathrm{d}W\,, \tag{17}$$

where p has been introduced in Assumption 2.1. We rewrite the solution of these equations in terms of Lie derivatives as for (16) introducing the differential operators $\hat{\mathcal{L}}$ and $\tilde{\mathcal{L}}$, *i.e.*,

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

Therefore,

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

where $\tilde{\mathcal{L}}^h$ is the *generator* of (17). (... all the passages to get to (28) in [3] ...).

Assumption 2.3. *The function f in (13) is in $\mathcal{C}^\infty$ and all its derivatives are uniformly bounded in $\mathbb{R}^n$. Furthermore, f is such that for all functions φ in $\mathcal{C}^\infty(\mathbb{R}^n, \mathbb{R})$*

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

for some $L > 0$.

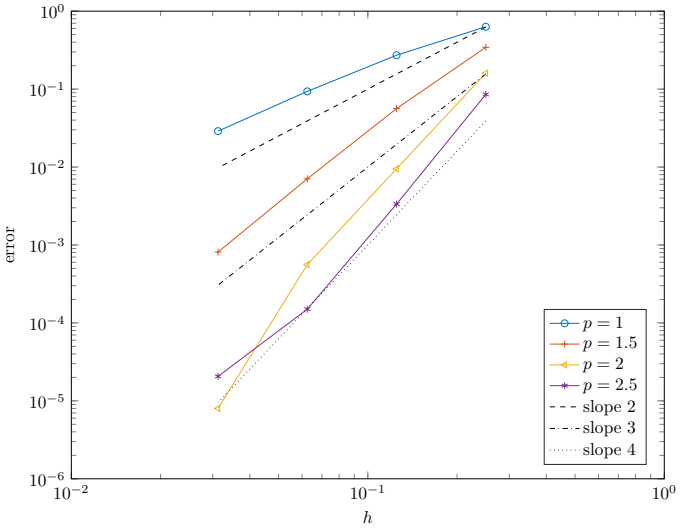

Figure 2: Weak order of convergence of (15) applied to (18).

We can now state the following result about weak convergence.

Proposition 2.3. *Consider the numerical method* (15) *and Assumptions 2.1, 2.2 and 2.3. Then for any function* φ *in* $\mathcal{C}^\infty$ *endowed with the properties of Assumption 2.3,*

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

and

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

with u *and* $\tilde{u}$ *solutions of* (13) *and* (17).

Proof. proof in [3]. □

2.4.1 Numerical verification of weak order

Let us consider for (13) the FitzHugh-Nagumo model, defined by

$$\begin{aligned} \frac{\mathrm{d}x_1}{\mathrm{d}t} &= c\left(x_1 - \frac{x_1^3}{3} + x_2\right), \\ \frac{\mathrm{d}x_2}{\mathrm{d}t} &= -\frac{1}{c}(x_1 - a + bx_2), \end{aligned} \tag{18}$$

where $a, b, c \in \mathbb{R}$. In particular, we choose $a = 0.2, b = 0.2, c = 3$. We provide the system with the initial condition $x_1(0) = -1, x_2(0) = 1$. We integrate numerically this equation with (15), using RK4 as Ψ. Therefore, Assumption 2.2 holds with $q = 4$. Moreover, we consider Q in Assumption 2.1 to be $Q = \sigma I$ with $\sigma = 0.1$. We approximate the solution up to time $T = 10$ with p in Assumption 2.1 equal to $1, 1.5, 2, 2.5$ and h vary in the range $0.25/(2^i), i = 0, \ldots, 3$. We approximate $\mathbb{E}^h\left(\varphi(U_k)\right)$ using a Monte Carlo simulation over 50000 trajectories and compare it with the solution computed on a fine grid to obtain an estimation of the weak error. Results (Figure 2) show that the predicted order $\min\{2p, q\}$ applies in this example. In particular, since $q = 4$, it is possible to notice that no difference in order is detected between the cases $p = 2$ and $p = 2.5$.

2.5 Monte Carlo approximation of probabilistic solvers

Let us consider the numerical method introduced in (15) and the Monte Carlo approximation

$$\hat{Z} = \frac{1}{M}\sum_{i=1}^{M} \varphi\left(U_N^{(i)}\right). \tag{19}$$

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 2.3 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 2.1. *Consider the numerical method* (15) *applied to a one-dimensional ODE with ψ any explicit Runge-Kutta method on s stages and Assumption 2.1. Then the numerical solution U_k at time $t_k = kh$ satisfies*

$$\mathrm{Var}(U_k) \leq C_1 \mathrm{Var}(U_0) + C_2 \sigma Q h^{2p},$$

with C_1, C_2 positive constants.

Proof. Let us consider as the numerical integrator ψ the Explicit Euler method and $p = 1$, coherently with the common choice that p is equal to the order or the deterministic integrator. Then, thanks to Assumption 2.1 we can write the numerical solution U_{k+1} as

$$U_{k+1} = U_k + hf(U_k) + h^{3/2}\sigma Q Z_k,$$

with Z_k a random variable such that $\mathbb{E}[Z_k] = 1$ and independent of U_k Then

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

where we exploited that for any random variables X, Y,

$$\mathrm{Var}(X + Y) \leq 2\,\mathrm{Var}(X) + 2\,\mathrm{Var}(Y). \tag{20}$$

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

$$\begin{aligned}\mathrm{Var}(f(U_k)) &= \mathrm{Var}(f(U_k) - f(\mathbb{E}[U_k])) \\ &\leq \mathbb{E}\left[(f(U_k) - f(\mathbb{E}[U_k]))^2\right] \\ &\leq C_L^2 \mathbb{E}\left[(U_k - \mathbb{E}(U_k))^2\right] \\ &= C_L^2\,\mathrm{Var}(U_k).\end{aligned} \tag{21}$$

Hence, we find

$$\begin{aligned}\mathrm{Var}(U_{k+1}) &\leq 2(1 + C_L^2 h^2)\,\mathrm{Var}(U_k) + \sigma Q h^3 \\ &\leq 2(1 + C_L^2 h^2)^k\,\mathrm{Var}(U_0) + \sigma Q T h^2,\end{aligned}$$

thus the result is proved choosing $C_1 = 2(1 + C_L^2 T^2)$ and $C_2 = T$.
Let us consider now any explicit Runge-Kutta method ψ, and let us rewrite (15) as

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

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

$$\tilde{\psi}(U_k) = \sum_{i=1}^{s} b_i K_i,$$

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

$$\mathrm{Var}(U_{k+1}) \leq 2\,\mathrm{Var}(U_k) + 2h^2\,\mathrm{Var}(\tilde{\psi}(U_k)) + \sigma Q h^{2p+1}.$$

Let us consider the second term. A direct bound, following from a generalization on s terms of (20) is

$$\mathrm{Var}(\tilde{\psi}(U_k)) \leq s \sum_{i=1}^{s} b_i^2 \,\mathrm{Var}(K_i). \tag{22}$$

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

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

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 (22) and get

$$\begin{aligned}
\mathrm{Var}(U_{k+1}) &\leq 2(1 + Cs \textstyle\sum_{i=1}^{s} b_i^2) \,\mathrm{Var}(U_k) + \sigma Q h^{2p+1} \\
&\leq \tilde{C} \,\mathrm{Var}(U_0) + T\sigma Q h^{2p},
\end{aligned}$$

thus giving the desired result with $C_1 = \tilde{C}$ and $C_2 = T$. □

Provided with this result we can now consider the variance of the Monte Carlo estimator $\hat{Z}$ introduced in (19). Since the samples $U_N^{(i)}$ are independent and identically distributed as U_N, we have

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

If the function φ is Lipschitz continuous we can use (21) and Lemma 2.1 and get

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

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

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

Let us remark [3] that a common choice for the noise scale p is the order of the deterministic solver q, therefore the bias and the variance terms in the MSE are both of order $2q$ with respect to h.

2.5.1 Numerical experiment

We consider the FitzHug-Nagumo problem introduced in (18) 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 φ 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 Explicit Euler 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. Results (Figure 3) confirm the result presented in Lemma 2.1, 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.

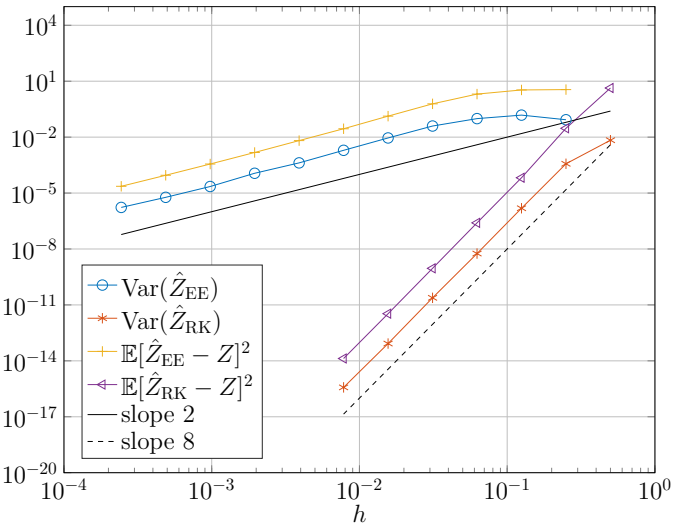

Figure 3: Variance and squared bias of the Monte Carlo estimator $\hat{Z}$ with Explicit Euler and RK4 applied to (18). The two components of the MSE have the same order of convergence with respect to the time step h.

2.6 Multi-level Monte Carlo

In this section, we will explain how to apply the Multi-level Monte Carlo method (MLMC) in this frame. Consider the approximation U_k given by (15) to the solution $u(t)$ of (13). Given φ a function in $\mathcal{C}^\infty$, let us denote by $Z = \mathbb{E}(\varphi(U_N)), Nh = T$ the expectation of the numerical solution at final time. If a standard Monte Carlo method over M realizations of the numerical solution is applied, the only accessible quantity is

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

where the index i is referred to the i-th trajectory. Therefore, the quantity $\hat{Z}$ is an unbiased estimator of Z. Then, the Mean Square Error (MSE) of $\hat{Z}$ is given by

$$\begin{aligned} \mathrm{MSE}(\hat{Z}) &= \mathbb{E}\left(\hat{Z} - \varphi\left(u(T)\right)\right)^2 \\ &= \mathrm{Var}\left(\hat{Z}\right) + \left(\mathbb{E}\left(\hat{Z} - \varphi\left(u(T)\right)\right)\right)^2 \\ &= \frac{C_1}{M} + C_2 h^{2\min\{2p,q\}}, \end{aligned} \tag{23}$$

where we have used Proposition 2.3 with C_1, C_2 positive constants. If we introduce as a measure for the error

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

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

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

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

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

The idea of MLMC is introducing an *hierarchical sampling*, introducing levels $l = 0, \ldots, L$, which have time step $h_l = T/N^l$ with $N_l = 2^l$. For each level, the number of trajectories is variable and

is denoted by M_l. 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

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

The variance is then computable as

$$\mathrm{Var}(\bar{Z}) = \sum_{l=0}^{L} \frac{1}{M_l^2} \sum_{i=1}^{M_l} \mathrm{Var}\left(\varphi_l^{(i)} - \varphi_{l-1}^{(i)} \right) = \sum_{l=0}^{L} \frac{V_l}{M_l}.$$

Thanks to Proposition 2.2 it is possible to estimate V_l.

Lemma 2.2. *If φ 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 recall that for any random variable Y_1, Y_2, it is true that

$$\mathrm{Var}(Y_1 + Y_2) \leq 2\left(\mathrm{Var}(Y_1) + \mathrm{Var}(Y_2) \right).$$

Let us consider now 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 the property of the variance above

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

Then, considering singularly the two terms and denoting by K the Lipschitz constant of φ

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

where the last bound is given by Proposition 2.2. □

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 ε, we get for the finest time step

$$h_L = \mathcal{O}\left(\varepsilon^{1/\min\{2p,q\}}\right). \tag{24}$$

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 (24) we can impose

$$L = \left|\log_2 \varepsilon^{1/q}\right|, \tag{25}$$

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 (24) 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 (15) is tuned so that $p = q$, as in this case neither the strong or the weak order are spoiled by the noise added to the model. Therefore, the first of the three cases presented above is of the highest interest. The plot in Figure **??** shows that in this case MLMC is particularly favorable with respect to Monte Carlo when the integrator has an order q which is small (e.g., $q = 1$).

2.6.1 Numerical example

We consider the Fitzhug-Nagumo problem (18) and we aim to verify the cost of MLMC with respect to standard Monte Carlo for the estimation of the expectation of the solution at final time when applying the numerical method (15). We consider the case $q = p = 1$, using as a deterministic integrator the explicit Euler method. Hence, once a value of accuracy ε is requested, the number of stages L as well as the time steps $h_l, l = 0, \ldots, L$, are imposed using (25) and (24). In order to set up the standard Monte Carlo method, we consider the cost obtained in the MLMC simulation,

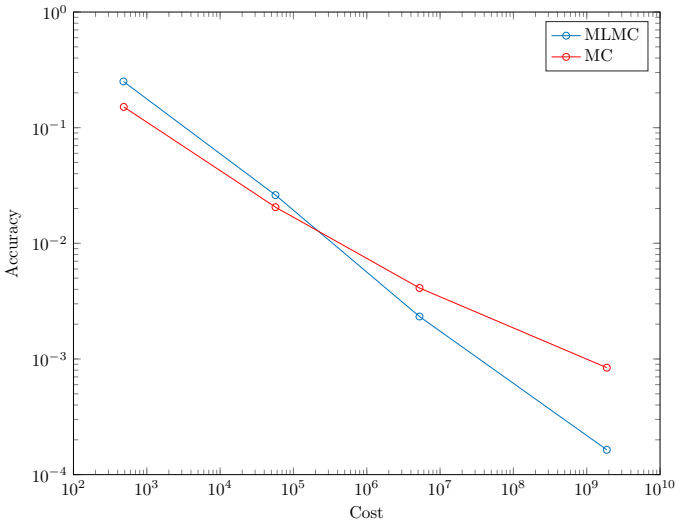

Figure 4: Accuracy of MLMC and standard Monte Carlo for the FitzHug-Nagumo problem with fixed cost.

denote by $\hat{C}$ and impose it to be equal for the standard Monte Carlo. In order to obtain a good balance between the error terms in (23) we impose

$$\begin{aligned} \frac{T}{h}M &= \hat{C}, \\ M &= \lceil h^{-2q} \rceil, \end{aligned}$$

thus obtaining for the time step

$$h = \left(\frac{T}{\hat{C}}\right)^{1/(2q+1)}.$$

In this way, the computational cost for MLMC and standard Monte Carlo are imposed to be artificially equal and the two methods can be compared for their weak error with respect to an accurate solution. We impose for MLMC four values of accuracy $\varepsilon = 0.1, 0.01, 0.001, 0.0001$, and apply the aforementioned technique to compare MLMC and Monte Carlo. Results (Figure 4) show that imposing L and $h_l, l = 0, \dots, L$ as above the obtained accuracy in the same order of magnitude as ε. Furthermore, the obtained accuracy is smaller for MLMC than MC if the cost

3 Bayesian inference of parameters

3.1 Bayesian inference - an introduction

(is it necessary?) Introduction of Bayesian inference, concluding with

$$\pi(\theta|d) \propto \mathcal{Q}(\theta)\mathcal{L}(d|\theta)$$

3.2 Bayesian inference of the parameters of an ODE

Let us consider the following initial value problem. Given u_0 a vector in $\mathbb{R}^d$ and a parameter set θ in $\mathbb{R}^p$

$$\begin{aligned} \frac{\mathrm{d}u}{\mathrm{d}t}(t;\theta) &= f(u(t;\theta)), \\ u(0;\theta) &= u_0. \end{aligned} \tag{26}$$

We consider the case in which θ is not known a priori and the problem of estimating its distribution. We consider the set of data d_i in $\mathbb{R}^d$ with $i = 1, \dots, D$ which represents the observed state of the

system (26) at a set of time $t_i, i = 1, \dots, D$ in which an observational noise ε is present. We assume that ε is normally distributed with zero mean and a given variance Γ, i.e.,

$$d_i = u(t_i) + \varepsilon_i, \quad \varepsilon_i \sim \mathcal{N}(0, \Gamma).$$

If the solution of (26) is computable analytically, then thanks to Bayes theorem we know that once a prior distribution $\mathcal{Q}(\theta)$ is specified, the posterior distribution of θ is given by Bayes' formula and can be expressed as

$$\pi(\theta|d) \propto \mathcal{Q}(\theta)\mathcal{L}(d|u(t;\theta)). \tag{27}$$

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

$$\mathcal{L}(d|u(\theta)) \propto \exp\left(\frac{1}{2}\sum_{i=1}^{D}\left(u(t_i;\theta) - d_i\right)^T \Gamma^{-2}\left(u(t_i;\theta) - d_i\right)\right).$$

In many cases, the analytical solution of (26) is not computable in closed form, therefore one replaces the analytical solution with its numerical approximation, which we denote by $U^{h,0}$. We then replace the analytical solution in the ODE, hoping that it does not spoil the quality of the posterior distribution, i.e.,

$$\mathcal{Q}(\theta)\mathcal{L}(d|u(\theta)) \approx \mathcal{Q}(\theta)\mathcal{L}(d|U^{h,0}(t;\theta)).$$

Another approach consists of considering the probabilistic numerical method (15) and a set of realizations of its solution. Let us denote by $U^{h,\sigma}(t,\xi;\theta)$ the numerical solution in this case, where σ represents the amount of uncertainty that is introduced with the method (see Assumption 2.1 and the following remarks). We then replace the likelihood function in (27) integrating over the random variable ξ, thus obtaining the following Bayes' rule

$$\pi(\theta|d) \propto \mathcal{Q}(\theta)\int \mathcal{L}(d|U^{h,\sigma}(t,\xi;\theta))\mathrm{d}\xi.$$

In [3] the authors claim that using the deterministic numerical solution for the purpose of estimating the parameters leads to unreliable posterior distributions, whilst the approximation of $\pi(\theta|d)$ provided by the probabilistic method takes accordingly into account the error introduced by the numerical solution.
In order to draw from the posterior distribution $\pi(\theta|d)$ one has to perform a Markov Chain Monte Carlo method, which is a class of numerical methods for Bayesian inference, briefly introduced in the following section.

3.3 Markov Chain Monte Carlo methods

The Markov Chain Monte Carlo (MCMC) methods are a useful tool for performing Bayesian inference. The main idea behind these methods is creating a chain of guesses of a parameter θ in order to build an approximation of its posterior distribution.
One of the most popular MCMC methods is the Metropolis-Hastings (MH) algorithm [7], presented in pseudo-code in Algorithm 1. In this algorithm, the new guess ϑ of the parameter θ value is drawn from a proposal function $q(\theta_k, \cdot)$ dependent on the current guess θ_k. Then, the new value ϑ is included in the chain as θ_{k+1} with a probability α dependent on the ratio between the posterior distribution evaluated in ϑ and θ, as in line **??** of Algorithm 1. Otherwise, θ_{k+1} is chosen to be equal to θ_k.
Let us remark that if $q(x, y)$ is a symmetric function, then the expression of the probability α at the k-th step simplifies to

$$\alpha = \min\left\{1, \frac{\pi(\vartheta|d)}{\pi(\theta_k|d)}\right\}.$$

This is the case, for example, of a Gaussian proposal distribution, which is a common choice (ADDREF) in case no a priori restriction is imposed on the range of θ.
Let us consider the problem of finding the distribution of the parameter θ defining an ODE. In

this case, once the new guess ϑ is generated from the proposal distribution, it is necessary to solve numerically (26) in order to determine the value of the likelihood function. In particular, assuming that the proposal distribution q is symmetric, the value of α at the k-th step in this frame reads in case the deterministic solver is adopted

$$\alpha = \min\left\{1, \frac{\mathcal{Q}(\vartheta)\mathcal{L}(d|U^{h,0}(t;\vartheta))}{\mathcal{Q}(\theta_k)\mathcal{L}(d|U^{h,0}(t;\theta_k))}\right\},$$

while for the probabilistic solver one gets

$$\alpha^{h,\sigma} = \min\left\{1, \frac{\mathcal{Q}(\vartheta)\int\mathcal{L}(d|U^{h,\sigma}(t,\xi;\vartheta))\mathrm{d}\xi}{\mathcal{Q}(\theta_k)\int\mathcal{L}(d|U^{h,\sigma}(t,\xi;\theta_k))\mathrm{d}\xi}\right\}. \tag{28}$$

The integrals in (28) are not trivial to compute, therefore a Monte Carlo approach has to be exploited. In particular, considering M realizations $\{\xi_i\}_{i=1}^M$of the random variable ξ, one can approximate the integral of the likelihood as

$$\int\mathcal{L}(d|U^{h,\sigma}(t,\xi;\theta))\mathrm{d}\xi \approx \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}(d|U^{h,\sigma}(t,\xi_i;\theta)). \tag{29}$$

In this way, the probability α is computable and can be used to obtain the distribution of the parameter θ.

3.3.1 Noisy pseudo-marginal MCMC

It is crucial to understand whether the approximation of the integrals in (28) influence the convergence of the posterior distribution to the true distribution of θ. Let us denote by $\pi(\theta|d)$ the real posterior distribution of θ, i.e.,

$$\pi(\theta|d) \propto \mathcal{Q}(\theta)\mathcal{L}(d|u(t,\theta)),$$

where u is the exact solution of the equation. Then, let us denote by $\pi^{h,\sigma}(\theta|d)$ the distribution obtained applying MH with the *transition kernel* (define it) induced by the probability $\alpha^{h,\sigma}$. Finally, let us denote by $\pi_N^{h,\sigma}(\theta|d)$ the distribution obtained approximating the integrals with Monte Carlo sums. We can rewrite the probability under the form of a *pseudo-marginal* Metropolis-Hastings (add reference). If one defines the following weights

$$\begin{aligned} W_{\theta,N} &= \frac{1}{N}\frac{\sum_{i=1}^{N}\mathcal{L}(d|U^{h,\sigma}(t,\xi_i;\theta))}{\int\mathcal{L}(d|U^{h,\sigma}(t,\xi;\theta))\mathrm{d}\xi} \\ &= \frac{1}{N}\sum_{i=1}^{N}\frac{\mathcal{L}(d|U^{h,\sigma}(t,\xi_i;\theta))}{\int\mathcal{L}(d|U^{h,\sigma}(t,\xi;\theta))\mathrm{d}\xi} \\ &= \frac{1}{N}\sum_{i=1}^{N}W_\theta^{(i)}, \end{aligned}$$

then the probability of acceptance can be rewritten as

$$\alpha_N^{h,\sigma} = \min\left\{1, \frac{\pi(\vartheta|d)W_{\vartheta,N}q(\theta_k,\vartheta)}{\pi(\theta_k|d)W_{\theta_k,N}q(\vartheta,\theta_k)}\right\}.$$

Let us remark that the random variables $W_\theta^{(i)}$ are i.i.d. with the property

$$\mathbb{E}(W_\theta^{(i)}) = 1, \quad i = 1,\dots,N,$$

and in the same way $W_{\theta,N}$ has unitary expectation. The probability can be computed in two different ways

1. the weight $W_{\theta_k,N}$ is not recomputed from the last iteration and only $W_{\vartheta,N}$ is drawn,

2. at each iteration both $W_{\theta_k,N}$ and $W_{\vartheta,N}$ are computed.

The second approach defines a noisy pseudo-marginal Metropolis-Hastings algorithm [2, 9, 10], which requires a double computational cost per iteration, as two Monte Carlo simulation have to be carried out for each MCMC iteration. On the other hand, the value of the likelihood at θ_k could be artificially good due to a particularly favorable set of realizations of ξ. Therefore, the ratio of the posteriors could be small, implying that the chain might remain blocked at the same guess of θ for an arbitrarily large number of iterations. In practice, the noisy approach guarantees a fast mixing, so that even with a double cost per-iteration it is computationally faster than the first approach.

We now consider the convergence of the probability distribution obtained with the noisy pseudo-marginal approach to the real distribution. We consider the total variation distance, which is defined as follow [5]

Definition 3.1. *Given two probability measures ν and μ on a measurable space $(\mathcal{X}, \mathcal{B}(\mathcal{X}))$, the total variation distance between P and Q is defined as*

$$\|\nu - \mu\|_{TV} \coloneqq \sup_{A \in \mathcal{B}(\mathcal{X})} |\nu(A) - \mu(A)|$$

Let us remark that the total variation distance between two probability measures is not often practical to compute. The Hellinger distance is more practical, especially in case the distributions are Gaussian. The Hellinger distance is defined as follows [5].

Definition 3.2. *If f, g are densities of the measures μ and ν on a measurable space $(\mathcal{X}, \mathcal{B}(\mathcal{X}))$ with respect to a dominating measure λ,*

$$d_H(\mu,\nu) = \left[\int_{\mathcal{X}} \left(\sqrt{f} - \sqrt{g}\right)^2 \mathrm{d}\lambda\right]^{1/2} = \left[2\left(1 - \int_{\mathcal{X}} \sqrt{fg}\right)\right]^{1/2}.$$

In the Gaussian case, if $\mu = \mathcal{N}(\mu_1, \Sigma_1)$, $\nu = \mathcal{N}(\mu_2, \Sigma_2)$, the Hellinger distance is given by

$$d_H(\mu,\nu)^2 = 1 - \frac{\det(\Sigma_1)^{1/4}\det(\Sigma_2)^{1/4}}{\det\left(\frac{\Sigma_1+\Sigma_2}{2}\right)^{1/2}} \exp\left(-\frac{1}{8}(\mu_1-\mu_2)^T\left(\frac{\Sigma_1+\Sigma_2}{2}\right)^{-1}(\mu_1-\mu_2)\right).$$

The Hellinger distance is equivalent to the total variation distance with the relation [5]

$$\frac{d_H(\mu,\nu)^2}{2} \leq \|\mu - \nu\|_{TV} \leq d_H(\mu,\nu), \tag{30}$$

hence when the total variation distance will not be computable, we will estimate it using the Hellinger distance. We consider now the distance between the true distribution of θ and the distribution obtained with the noisy-pseudomarginal approach. Let us remark that by the triangular inequality

$$\left\|\pi - \pi_N^{h,\sigma}\right\|_{TV} \leq \left\|\pi - \pi^{h,\sigma}\right\|_{TV} + \left\|\pi^{h,\sigma} - \pi_N^{h,\sigma}\right\|_{TV}.$$

Intuitively, the first term in the sum concerns the numerical accuracy of the numerical method, while the second term concerns the quality of the approximation performed in (29). For the first term, if θ is a vector of $\mathbb{R}^g$ we remark that thanks to Theorem 2.3 we have

$$\begin{aligned}
\left\|\pi - \pi^{h,\sigma}\right\|_{TV} &= C \sup_{A \in \mathcal{B}(\mathbb{R}^g)} \int_A \mathcal{Q}(\theta)\left(\mathcal{L}(d|u(\theta)) - \left(\int \mathcal{L}(d|U^{h,\sigma}(\theta,\xi))\mathrm{d}\xi\right)\right)\mathrm{d}\theta \\
&= C \sup_{A \in \mathcal{B}(\mathbb{R}^g)} \int_A \mathcal{Q}(\theta)\left(\mathcal{L}(d|u(\theta)) - \mathbb{E}^{\xi}\left(\mathcal{L}(d|U^{h,\sigma}(\theta,\xi))\right)\right)\mathrm{d}\theta \\
&\leq C h^{\min\{q,2p\}} \sup_{A \in \mathcal{B}(\mathbb{R}^g)} \int_A \mathcal{Q}(\theta)\mathrm{d}\theta \\
&= C h^{\min\{q,2p\}},
\end{aligned}$$

where $\mathbb{E}^{\xi}(\cdot)$ denotes the expectation with respect to the random variable ξ and C is a positive constant independent of h.
For the second term, since the weights $W_{\theta,N}$ are given by arithmetic averages and have unitary expectation, the following result has been shown [9].

Proposition 3.1. *Under appropriate conditions (add them?) there exist* $0 < \delta < 1/6$, $C_\delta > 0$ *and* $N_0 \in N^+$ *such that for all* $N \geq N_0$

$$\left\| \pi^{h,\sigma} - \pi_N^{h,\sigma} \right\|_{TV} \leq C_\delta \frac{\log(N)}{N^{\frac{1}{2}-\delta}}.$$

With the two results above, we can now estimate the convergence with respect to h and N to the true probability distribution in the total variation distance

$$\begin{aligned} \left\| \pi - \pi_N^{h,\sigma} \right\|_{TV} &\leq \left\| \pi - \pi^{h,\sigma} \right\|_{TV} + \left\| \pi^{h,\sigma} - \pi_N^{h,\sigma} \right\|_{TV} \\ &\leq C h^{\min\{q,2p\}} + C_\delta \frac{\log(N)}{N^{\frac{1}{2}-\delta}}. \end{aligned}$$

where C and C_δ are specified above. Hence, defining the error e as

$$e \coloneqq \left\| \pi - \pi_N^{h,\sigma} \right\|_{TV}$$

and imposing it to be equal to $\mathcal{O}(\varepsilon)$ where ε is a desired tolerance we find that the time step h has to satisfy

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

As far as the number of samples N in $W_{\theta,N}$ is concerned, if we define the function $F_\delta \colon \mathbb{R} \to \mathbb{R}$ as

$$F_\delta(N) \coloneqq \frac{\log(N)}{N^{\frac{1}{2}-\delta}},$$

then its inverse function F_δ^{-1} is given by

$$F_\delta^{-1}(x) = \exp\left(\frac{1}{\gamma} W(\gamma x) \right).$$

where $\gamma \coloneqq \delta - 1/2$ and W is the Lambert function. Therefore, in order to balance the two error terms it is necessary to impose

$$N = \mathcal{O}(F_\delta^{-1}(\varepsilon)).$$

Thus, the computational cost per iteration of the noisy Metropolis-Hastings algorithm is given by

$$\begin{aligned} \text{cost} &= \mathcal{O}(h^{-1} N) \\ &= \mathcal{O}\left(\varepsilon^{-1/\min\{q,2p\}} F_\delta^{-1}(\varepsilon) \right) \end{aligned}$$

Let us remark that the computational cost predicted by this formula is rapidly growing for small values of ε, leading to unaffordable computational times when a precise computation is required.

3.3.2 A MLMC approach

Instead of arithmetic averages, use

$$W_{\theta,\text{MLMC}} = \sum_{l=0}^{L} \frac{1}{M_l} \sum_{i=1}^{M_l} \left(W_l^{(i)} - W_{l-1}^{(i)} \right),$$

where (definition of W_l)

3.3.3 Numerical example

We consider the test equation

$$\begin{aligned} \frac{\mathrm{d}u(t)}{\mathrm{d}t} &= \lambda u(t), \\ u(0) &= 1 \end{aligned}$$

with λ a real negative parameter. If λ is big in absolute value, the equation is stiff. In this experiment, we are not interested in stiff equations, therefore we consider $\lambda = -0.5$. The analytical solution of this equation is known and is given by

$$u(t) = \exp(\lambda t), \quad t > 0.$$

We are interested in verifying the order of convergence of the noisy MH presented in section 3.3.1. We consider the prior distribution to be a Gaussian centered in the true value of the parameter $\bar{\lambda} = -0.5$ with unitary variance. Then, we generate data from the analytical solution at $t = 1$ with a normal disturbance, i.e.,

$$d = \exp(\lambda) + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \Gamma),$$

with $\Gamma = 0.001$. In this way, it is possible to generate the true posterior distribution $\pi(\lambda|d)$ through Bayes' rule

$$\pi(\lambda|d) \propto \mathcal{Q}(\lambda)\mathcal{L}(d|\exp(\lambda)),$$

where the prior distribution is given by

$$\mathcal{Q}(\lambda) = (2\pi)^{-1/2} \exp\left(-\frac{1}{2}(\lambda - \bar{\lambda})\right),$$

and the likelihood is given by

$$\mathcal{L}(d|\exp(\lambda)) = (2\pi\Gamma)^{-1/2} \exp\left(-\frac{1}{2\Gamma}(\exp(\lambda) - d)^2\right).$$

Normalizing the product of prior and likelihood, we obtain the true posterior distribution for the parameter λ. We consider now the RAM algorithm for the probabilistic method (15) with explicit Euler as a deterministic solver. We consider $h = \{0.1, 0.05, 0.01\}$ and $N = \{1, 10, 100, 1000, 10000\}$. In this way, we can observe for each value of h the convergence of the posterior distribution $\pi_N^{h,\sigma}$ to the exact distribution. In order to estimate the total variation distance between $\pi_N^{h,\sigma}$ and π we consider the bound given by the Hellinger distance between the normal distributions with the estimate values of mean and variance (30).

3.3.4 A Gaussian filtering approach

An interesting approach for the Bayesian analysis of the parameters of an ODE or of an SDE has been recently proposed in [11]. In this paper, the authors propose a Gaussian filtering approach to solve the differential equation at each step of an MCMC algorithm, avoiding in this way the computationally inefficient Monte Carlo simulation typical of the MCWM approach. The method consists in building at each iteration of the MCMC algorithm a Gaussian approximation of the solution of an SDE, computing the evolution of the mean and variance of a Gaussian distribution with an ODE approach. Let us consider the acceptance probability in a standard MH

$$\alpha = \min\left\{ \frac{\mathcal{Q}(\vartheta)\mathcal{L}(d|\vartheta)}{\mathcal{Q}(\theta_k)\mathcal{L}(d|\theta_k)} \right\}.$$

While the prior distribution is easy to evaluate, the likelihood function is in this case intractable. The pseudo-marginal MCMC approach and its noisy version approximate the likelihood with a Monte Carlo simulation, which can be extremely costly.
Let us consider $f\colon \mathbb{R}^{N_s} \to \mathbb{R}^{N_s}$, $g\colon \mathbb{R}^{N_s} \to \mathbb{R}^{N_s \times N_s}$, W a d-dimensional Wiener process and the following SDE

$$\begin{aligned} \mathrm{d}U(t;\theta) &= f_\theta(U)\mathrm{d}t + g_\theta(U)\mathrm{d}W, \quad 0 < t \leq T, \\ U(0) &= U_0, \end{aligned} \tag{31}$$

where we assume that U_0 is a deterministic initial condition in $\mathbb{R}^{N_s}$ and that the functions f, g depend on a parameter θ of $\mathbb{R}^{N_p}$. Under Assumption 2.1 with $Q = \sigma I$, solving an ODE with the probabilistic method defined in (15) is equivalent to solving numerically (31) for the choice

$$g(U) = G = \sigma h^p I,$$

with I the identity matrix in $\mathbb{R}^{N_s \times N_s}$. Since the method proposed in [11] is applicable to any SDE of the form (31), in the following we formally maintain this more general notation. Let us denote by y_i an observation of the state of (31) at time t_i for $i = 1, \dots, N_d$, and by Y_i the set of all observations up to time t_i, i.e.,

$$Y_i = \{y_1, y_2, \dots, y_{i-1}, y_i\}.$$

Let us furthermore assume that y_i is measured from the solution with the following additive noise model

$$y_i = U(t_i) + \varepsilon_i, \quad \varepsilon_i \sim \mathcal{N}(0, \Sigma_y).$$

The likelihood appearing in the probability α can be therefore written as

$$\begin{aligned}\mathcal{L}(Y_k|\theta) &= \prod_{i=1}^{k} (2\pi \det(\Sigma_y))^{-k/2} \exp\left(-\frac{1}{2}(U(t_i;\theta) - y_i)^T \Sigma_y^{-1} (U(t_i;\theta) - y_i)\right) \\ &= \prod_{i=1}^{k} \mathcal{L}_i(y_i|\theta).\end{aligned}$$

3.4 Numerical example

We consider the ODE defined in (**??**) 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 θ the vector $\bar{\theta} = (0.2, 0.2, 3)$. In order to produce the observations d_i, we consider a reference solution $\bar{u}(t, \bar{\theta})$ produced with a fine time step and its values at $t_i = 1, 2, 3, \dots, 39$ and then we consider

$$d_i = \bar{u}(t_i, \bar{\theta}) + \varepsilon_i, \quad \varepsilon_i \sim \mathcal{N}(0, 10^{-3} I), \quad i = 1, \dots, 39,$$

where I is the identity matrix in $\mathbb{R}^{3\times 3}$. Therefore, we consider a diagonal noise with independent normal components having all variance 10^{-3}. We approximate the posterior distribution $\pi(\theta|d)$ with both the deterministic and the probabilistic solvers using time steps h_i in the range $\{0.1, 0.05, 0.01, 0.005\}$. The proposal function $q(x, y)$ for MH is Gaussian, and the prior distribution $\mathcal{Q}(\theta)$ is lognormal with unitary variance and mean $\bar{\theta}$. We consider 10^6 iterations of the MH algorithm for all time steps and in both the deterministic and the probabilistic case. Results show that

- In the deterministic case (Figure 5) the marginals of the posterior distributions show an extremely small variance and are not nested for different time steps. This means that the estimation of parameters is not reliable as it does not account for the error introduced by the numerical approximation of the ODE.
- In the probabilistic case (Figure 6) the results show bigger variances, with posterior distributions which fully account for the numerical approximation. In fact, one can see that for the smaller values of the time step the marginal distributions are more concentrated, while for the big values (e.g. $h = 0.1, 0.05$) the estimation of θ is clearly unreliable, as it is supposed to be due to numerical integration.

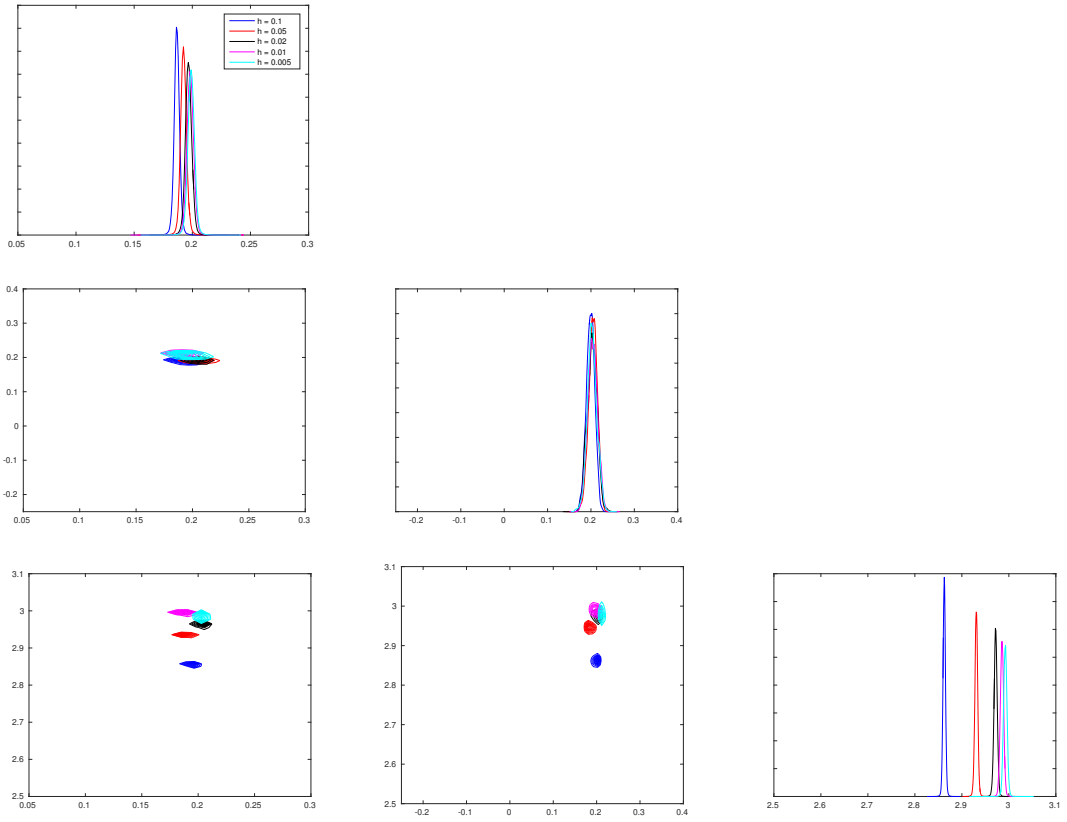

Figure 5: Marginal distributions for θ obtained with the deterministic solver. The posterior distributions are clearly concentrated and mutually singular.

4 Numerical example

Let us consider the Lorenz system, *i.e.*,

$$\begin{aligned}\frac{\mathrm{d}x}{\mathrm{d}t} &= \sigma(y-x),\\ \frac{\mathrm{d}y}{\mathrm{d}t} &= x(\rho-z)-y,\\ \frac{\mathrm{d}z}{\mathrm{d}t} &= xy-\beta z,\end{aligned}$$

where $\sigma = 10, \rho = 28, \beta = 8/3$. We provide the system with initial conditions

$$x(0) = -10, \quad y(0) = -1, \quad z(0) = 40,$$

and integrate the system up to the final time $T = 20$. We choose as deterministic integrator the classic fourth-order Runge-Kutta method, with timestep $h = 0.001$. Results show that the realizations of the numerical solutions obtained with the probabilistic method follow the behaviour of the deterministic method in the first part of the time span and then show a chaotic behaviour, revealing the features of the underlying system. The three components x, y, z of the solution obtained with the deterministic and probabilistic solvers are depicted in Figure 7.

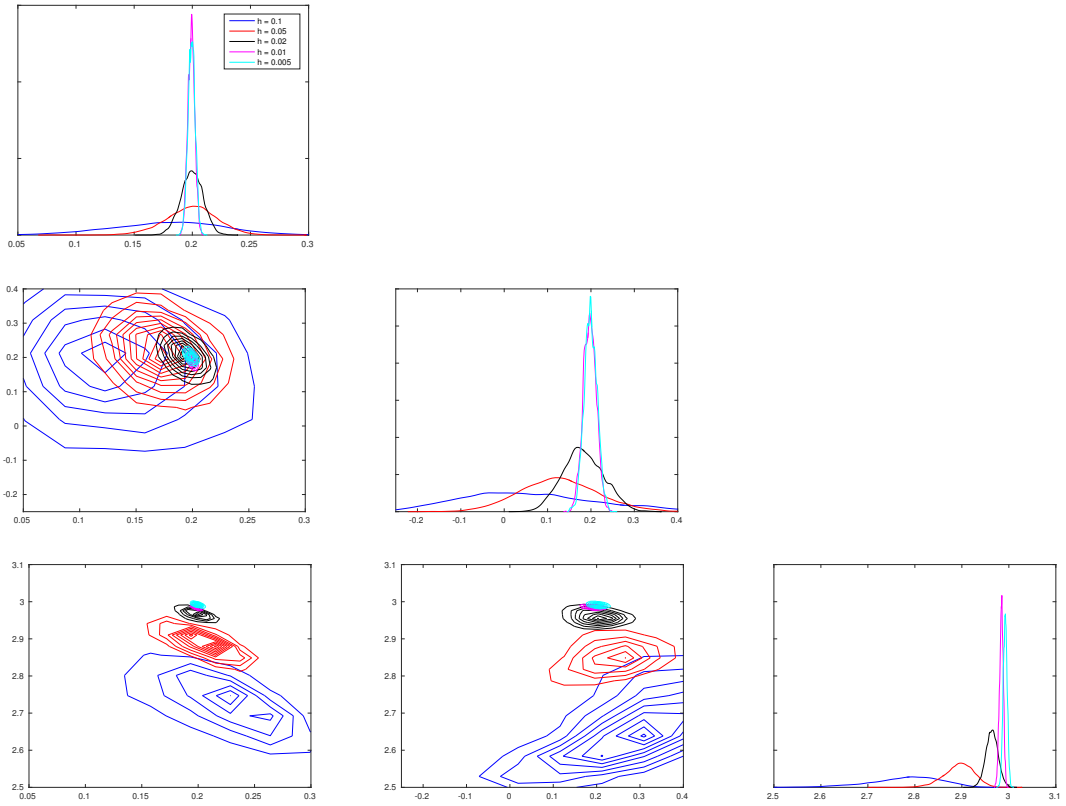

Figure 6: Marginal distributions for θ obtained with the probabilistic solver. The posterior distributions account for the numerical error.

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

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

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

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

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

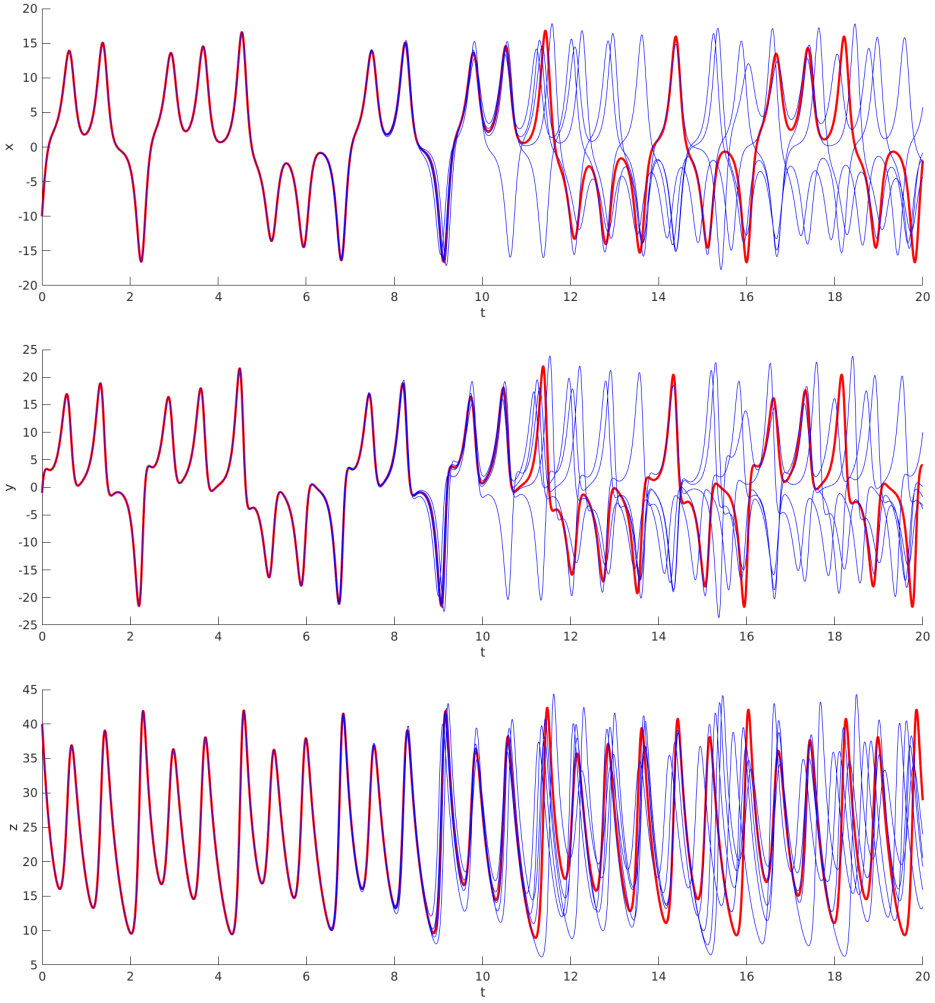

Figure 7: Solution of the Lorenz system obtained with the deterministic solver (red) and realizations of the solution obtained with the probabilistic solver (blue).

[10] P. D. O'Neill, D. J. Balding, N. G. Becker, M. Eerola, and D. Mollison, *Analyses of infectious disease data from household outbreaks by Markov chain Monte Carlo methods*, J. R. Stat. Soc. Ser. C. Appl. Stat., 49 (2000), pp. 517–542.

[11] S. Särkkä, J. Hartikainen, I. S. Mbalawata, and H. Haario, *Posterior inference on parameters of stochastic differential equations via non-linear Gaussian filtering and adaptive MCMC*, Stat. Comput., 25 (2015), pp. 427–437.

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