Statistics and Computing (2022) 32:1
https://doi.org/10.1007/s11222-021-10060-4

Emulation-accelerated Hamiltonian Monte Carlo algorithms for parameter estimation and uncertainty quantification in differential equation models

L. Mihaela Paun[1] · **Dirk Husmeier**[1]

Received: 24 February 2021 / Accepted: 21 October 2021 / Published online: 23 November 2021

Abstract
We propose to accelerate Hamiltonian and Lagrangian Monte Carlo algorithms by coupling them with Gaussian processes for emulation of the log unnormalised posterior distribution. We provide proofs of detailed balance with respect to the exact posterior distribution for these algorithms, and validate the correctness of the samplers' implementation by Geweke consistency tests. We implement these algorithms in a delayed acceptance (DA) framework, and investigate whether the DA scheme can offer computational gains over the standard algorithms. A comparative evaluation study is carried out to assess the performance of the methods on a series of models described by differential equations, including a real-world application of a 1D fluid-dynamics model of the pulmonary blood circulation. The aim is to identify the algorithm which gives the best trade-off between accuracy and computational efficiency, to be used in nonlinear DE models, which are computationally onerous due to repeated numerical integrations in a Bayesian analysis. Results showed no advantage of the DA scheme over the standard algorithms with respect to several efficiency measures based on the effective sample size for most methods and DE models considered. These gradient-driven algorithms register a high acceptance rate, thus the number of expensive forward model evaluations is not significantly reduced by the first emulator-based stage of DA. Additionally, the Lagrangian Dynamical Monte Carlo and Riemann Manifold Hamiltonian Monte Carlo tended to register the highest efficiency (in terms of effective sample size normalised by the number of forward model evaluations), followed by the Hamiltonian Monte Carlo, and the No U-turn sampler tended to be the least efficient.

Keywords Parameter estimation · Uncertainty quantification · MCMC · Emulation · Gaussian processes · Differential equations

1 Introduction

Parameter estimation and uncertainty quantification (UQ) in systems of nonlinear ordinary and partial differential equations (ODEs/PDEs) is a topical research area with the emergence of complex mathematical models expressed via ODEs or PDEs. Such models are heavily used throughout all science and engineering fields to understand the underlying mechanisms behind a process (e.g. biological systems (Wilkinson 2007), or physiology (Mirams et al. 2016)). Statistical inference allows estimation of the unknown model parameters from the data in a robust and coherent manner within a Bayesian or frequentist framework. This is however a challenging task to accomplish since nonlinear ODE/PDE models that faithfully capture real-world processes of interest are analytically intractable and can only be solved using numerical integration. While this may not be a problem if the numerical integration is performed only a few times, it quickly becomes a major hindrance if incorporated within an adaptive parameter estimation procedure requiring thousands of ODE/PDE evaluations, incurring high computational costs. In addition, non-identifiable parameters due to model formulation or insufficient amount of data, and any strong parameter correlations further complicate the statistical analysis. Bayesian methods can be employed to provide posterior probability distributions over parameters; Markov

✉ L. Mihaela Paun
l.paun.1@research.gla.ac.uk

Dirk Husmeier
dirk.husmeier@glasgow.ac.uk

[1] School of Mathematics and Statistics, University of Glasgow, Glasgow G12 8QQ, UK

Chain Monte Carlo (MCMC) algorithms are generally used to sample from the posterior distribution.

Finding efficient algorithms that return high effective sample sizes (ESS) in a reasonable time frame is challenging, especially if there are strong correlations between the parameters, which would retard the convergence of standard MCMC algorithms, such as Metropolis–Hastings (M–H) (Turner et al. 2013). The implication of using standard MCMC algorithms on such problems is that a small step size is needed to obtain a reasonable acceptance rate, which in turn means that low ESS (or high auto-correlation) is obtained. This problem can be alleviated by using more advanced MCMC algorithms, such as the HMC algorithm (Neal 2011). HMC introduces an auxiliary variable and makes use of the gradient of the posterior distribution for more informed moves in parameter space. However, while it has been shown on numerous occasions that the HMC algorithm outperforms random-walk algorithms in terms of efficiency (e.g. in Ch. 5 in Brooks et al. 2011 or in Sengupta et al. 2016), it has rarely been applied to nonlinear ODE or PDE models, with four noticeable exceptions Kramer et al. (2014), Lê et al. (2016), Bui-Thanh and Girolami (2014), Sengupta et al. (2016).

The major drawback of applying HMC to ODE/PDE models over the M–H algorithm is related to the large number of model evaluations before a proposal is made. In HMC, trajectories are simulated, and throughout each trajectory, the ODE/PDEs are evaluated multiple times (for calculating the likelihood and its gradient) until a proposal is made, unlike the M–H algorithm, which requires one single ODE/PDE evaluation for a proposal to be made. To reduce the computational burden, several approaches have been proposed in the literature. In a special class of ODE models (steady state data models), Kramer et al. (2014) make use of a special property of steady state data to obtain output sensitivities (i.e. derivatives of the model output with respect to the unknown parameters) required in the Hamiltonian equations, through analytical calculations. However, the approach in Kramer et al. (2014) cannot be applied to dynamic time data ODE/PDE models for which the output sensitivities can only be obtained via numerical integration. Sengupta et al. (2016) compare the performance in terms of computational speed and accuracy of three methods for calculating the likelihood gradients: finite differences,[1] forward sensitivities[2] and the adjoint method,[3] and the latter was shown to be superior (specific details of these methods can also be found in Sengupta et al. (2014)). Similarly, Bui-Thanh and Girolami (2014) apply the adjoint method in a PDE system to compute the first-, second- and third-order derivatives of the likelihood as part of the Riemann Manifold HMC (RMHMC) algorithm. The approaches taken in Sengupta et al. (2016) and Bui-Thanh and Girolami (2014) can nevertheless still be too computationally expensive for large class problems. The study in Chen et al. (2014) introduces stochastic-gradient HMC, which sub-samples the data and uses stochastic gradients as a replacement to full-data gradients in the Hamiltonian equations. Data sub-sampling introduces noise, which gets propagated to the Hamiltonian differential equations, thus the convergence to the posterior distribution depends on stochastic differential equations, which may reduce exploration efficiency and accuracy (Betancourt 2015).

Another approach proposed to speed up HMC involves the replacement of the expensive likelihood (or posterior distribution) with computationally cheaper surrogate models (Rasmussen 2003; Zhang et al. 2017), i.e. HMC is coupled with statistical emulation of the unnormalised posterior distribution. For this method, two routes can be taken: drawing samples from the approximate posterior distribution defined by the surrogate model ('emulator'), or drawing samples from a distribution that asymptotically converges to the true posterior distribution ('simulator'). The first route, taken in multiple studies (Peirlinck et al. 2019; Schiavazzi et al. 2016; Paun et al. 2019; Costabal et al. 2019; Bliznyuk et al. 2008; Dietzel and Reichert 2014; Fielding et al. 2011; Wilkinson 2014), samples from the surrogate posterior distribution, resulting in substantial gains in computational efficiency (since the expensive model is no longer used), however accuracy is sacrificed.

Methods that correct for the bias and ensure asymptotic exactness use the surrogate for the proposal only (see studies in Rasmussen 2003; Zhang et al. 2017; Paun and Husmeier 2020), or incrementally refine the surrogate model as MCMC proceeds (Zhang et al. 2017). Other studies employ similar approaches guaranteeing asymptotic convergence to the posterior distribution (Wu and Li 2016; Conrad et al. 2016, 2018; Gong and Duan 2017), e.g. the study in Conrad et al. (2016) uses forward simulations from the expensive model to continually refine a local approximation of the log unnormalised posterior, however the algorithm depends on various heuristic parameters, which critically affect the computational efficiency and may be difficult to tune in practice. The delayed acceptance (DA) scheme (Christen and Fox 2005; Golightly et al. 2015) is another exact method, and it does not depend on any heuristically set terms. This method, employed in Christen and Fox (2005), Golightly et al. (2015), Sherlock et al. (2017), Higdon et al. (2011), Cui et al. (2011), Quiroz et al. (2018), Banterle et al. (2019), Paun and Husmeier (2020), is a two-stage acceptance procedure, with two separate acceptance/rejection decisions. The first decision is a computationally fast pre-filter step based on the surrogate model, which upon rejection of a proposed new parameter

[1] $\frac{df(x)}{dx} = \frac{f(x+h)-f(x)}{h}$, where $h > 0$ is a small constant

[2] The ODE system is augmented to include the gradient of the states with respect to the parameters (called the sensitivity derivative equation)

[3] Avoids solving the sensitivity derivative equation by the use of a linear adjoint equation, which is easier to numerically integrate

avoids carrying out the computationally expensive second step based on the original model.

In the current study, we employ HMC in combination with statistical emulation using Gaussian processes (GPs) of the log unnormalised posterior distribution within the GP-HMC algorithm, following ideas in Rasmussen (2003) and Lan et al. (2016). Throughout the trajectory, HMC runs at low computational costs in the emulated space, and the Metropolis–Hastings acceptance/rejection step at the end of the trajectory is based on the ratio of the true posterior distributions. This requires one single numerical integration of the ODEs/PDEs throughout one HMC trajectory segment to obtain one parameter proposal, which substantially reduces the computational complexity. The algorithm is exact in the sense of converging to the true posterior distribution asymptotically, assuming no discretisation errors are introduced from the numerical integration of the ODEs/PDEs.[4]

Our methodological contributions are as follows. Firstly, we generalise the GP-HMC algorithm (Rasmussen 2003) to algorithms which advance HMC: No U-turn sampler (NUTS) (Hoffman and Gelman 2014), RMHMC (Girolami and Calderhead 2011), and Lagrangian Dynamical Monte Carlo (LDMC) (Lan et al. 2015). This includes novel detailed balance proofs with respect to the correct posterior distribution, and we show how these subsume the proof of the GP-HMC algorithm as a limiting case. Secondly, we extend these algorithms by including DA, with formal detailed balance proofs, to investigate if the DA scheme brings any computational gains over the standard algorithms. Thirdly, we assess the computational efficiency of these emulation algorithms on a series of complex nonlinear ODE/PDE models.

The present study extends our previous work (Paun and Husmeier 2020) by providing a theoretical framework for the emulation algorithms and by investigating the efficiency of the DA scheme in the context of the emulation HMC algorithms. Moreover, we extend the simulation study to provide a broad benchmark set of ODE/PDE models that are representative of the complexity of typical real-world applications, allowing us to carry out a sound method evaluation.

2 Background

2.1 Bayesian inference in ODEs/PDEs

By denoting the solution (output) from the ODEs/PDEs as $\mathbf{m}(\boldsymbol{\theta})$, which is a function of the unknown parameters $\boldsymbol{\theta}$, we assume Normal data likelihood:

$$p(\mathbf{y}|\boldsymbol{\theta},\sigma^2) = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^n \exp\left(-\frac{\sum_{i=1}^n (y_i - m_i(\boldsymbol{\theta}))^2}{2\sigma^2}\right) = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^n \exp\left(-\frac{\mathcal{S}(\boldsymbol{\theta})}{2\sigma^2}\right), \tag{1}$$

where $\mathbf{m}(\boldsymbol{\theta}) = (m_1(\boldsymbol{\theta}), \ldots, m_n(\boldsymbol{\theta}))$ is the vector of predictions from the ODEs/PDEs, $\mathbf{y} = (y_1, \ldots, y_n)$ is the vector of measurements, n is the number of data points, $\mathcal{S}(\boldsymbol{\theta})$ is the residual sum-of-squares (RSS) value corresponding to the parameter vector $\boldsymbol{\theta}$, and σ^2 is the noise variance.

We employ Bayesian methods to infer $\boldsymbol{\theta}$, which is done via exploration of the posterior distribution: $p(\boldsymbol{\theta}|\mathbf{y}) = \frac{p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{\int_{\boldsymbol{\theta}} p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}}$, where $p(\boldsymbol{\theta})$ is the prior distribution. The priors for the ODE/PDE model parameters are given in Sect. 4.

The integral in the denominator is intractable for the complex DE models considered, thus the posterior distribution is approximated using numerical schemes based on MCMC sampling: $p(\boldsymbol{\theta}|\mathbf{y}) \propto p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})$.

To accelerate the sampling procedure, we couple MCMC with emulation using GPs. We emulate RSS, and the emulated likelihood is defined as:

$$\tilde{p}(\mathbf{y}|\boldsymbol{\theta},\sigma^2) = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^n \exp\left(-\frac{\tilde{\mathcal{S}}(\boldsymbol{\theta})}{2\sigma^2}\right), \tag{2}$$

where $\tilde{\mathcal{S}}(\boldsymbol{\theta})$ is the RSS value predicted by the emulator for the particular $\boldsymbol{\theta}$. The emulated posterior distribution becomes: $\tilde{p}(\boldsymbol{\theta}|\mathbf{y}) \propto \tilde{p}(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})$.

In Sect. 2.2 we briefly review the HMC-type algorithms utilised and in Sect. 2.3 we show how HMC can be coupled with GPs as part of the GP-HMC algorithm (Rasmussen 2003). The methodological contribution of our paper is described in Sect. 3.

2.2 Gradient-based MCMC algorithms

2.2.1 HMC

HMC (Neal 2011) introduces an auxiliary variable, the 'momentum' with a mass matrix $\mathbf{M}$ (identity matrix), and simulates Hamiltonian dynamics by using gradient information from the log target density. An HMC trajectory is simulated by numerically integrating the Hamiltonian dynamics with the leapfrog integrator (Neal 2011) using a small step size, $\epsilon > 0$ and a number of leapfrog steps, L. However, the numerical integration induces an error, and the bias is corrected by a M–H accept/reject step. More details can be found in the Supplement, Sect. 2.

[4] Investigation of the discretisation errors is beyond the scope of this paper.

2.2.2 RMHMC

Riemann Manifold HMC (RMHMC) (Girolami and Calderhead 2011) exploits the Riemannian geometry of the parameter space, and sets $\mathbf{M}$ based on the curvature of the (approximate) target distribution at every step throughout the trajectory. $\mathbf{M}$ is the metric tensor of the Riemann space and is calculated based on the expected Fisher information matrix plus the negative Hessian log prior:

$$\begin{aligned} M_{i,j} &= \mathbb{E}_{\mathbf{y}|\boldsymbol{\theta}}\left[-\frac{\partial^2 \log p(\mathbf{y},\boldsymbol{\theta})}{\partial\theta_i \partial\theta_j}\right] \\ &= \mathbb{E}_{\mathbf{y}|\boldsymbol{\theta}}\left[-\frac{\partial^2 \log p(\mathbf{y}|\boldsymbol{\theta})}{\partial\theta_i \partial\theta_j}\right] + \left(-\frac{\partial^2 \log p(\boldsymbol{\theta})}{\partial\theta_i \partial\theta_j}\right). \end{aligned} \tag{3}$$

An implicit integrator, the generalised leapfrog scheme, is used for the numerical integration of the Hamiltonian dynamics, since reversibility with the leapfrog integrator is no longer satisfied.

2.2.3 LDMC

LDMC (Lan et al. 2015) simulates Lagrangian dynamics (instead of Hamiltonian dynamics), thus the 'velocity' variables replace the 'momentum' variables. This enables the use of an explicit geometric integrator, which substantially improves the computational efficiency of the costly RMHMC implicit integrator. The consequence is that the volume in phase space is no longer preserved, and to ensure detailed balance (Lan et al. 2015), the Jacobian transformation is needed to adjust the M–H acceptance probability. In LDMC, similarly to RMHMC, $\mathbf{M}$ is adjusted to the curvature of the (approximate) posterior distribution at every step throughout the trajectory.

2.2.4 NUTS

The No U-turn sampler (NUTS) (Hoffman and Gelman 2014) chooses L recursively by moving in Euclidean parameter space (i.e. $\mathbf{M}$ equals the identity matrix) until the leapfrog trajectory starts to double back and retrace its steps. This is achieved via a tree doubling process building a binary tree, whose leaf nodes correspond to the position-momentum states. The points collected along the trajectory are sampled in a way that preserves detailed balance (Hoffman and Gelman 2014). NUTS uses stochastic optimisation (the primal-dual averaging algorithm) to tune ϵ in the burn-in phase.

2.2.5 Bayesian optimisation for parameter tuning

In our work we employ Bayesian optimisation (Wang et al. 2013; Mockus et al. 1978) to automatically tune the step size ϵ and number of steps L in HMC, RMHMC and LDMC. ϵ and L are optimised by maximising an objective function, which, following Wang et al. (2013), we take to be the expected squared jumping distance (ESJD) normalised by the number of leapfrog steps, $\frac{\mathbb{E}^{\epsilon,L}_{p(.)}||\boldsymbol{\theta}^{(i+1)} - \boldsymbol{\theta}^{(i)}||^2}{\sqrt{L}}$. The idea of emulation and Bayesian optimisation (Shahriari et al. 2016) is adopted, with optimisation of the Upper Confidence Bound acquisition function (Wang et al. 2013).

2.3 HMC coupled with emulation using GPs

GP-HMC (Rasmussen 2003) combines the HMC algorithm with GPs (briefly reviewed in the Supplement, Sect. 1), used for statistical emulation of the log unnormalised posterior distribution (Kennedy and O'Hagan 2001; Conti et al. 2009; Conti and O'Hagan 2010a). The need for using the GP-HMC algorithm over the plain HMC algorithm is illustrated in Algorithm 1, which comparatively provides the total number of model (ODE/PDE) evaluations incurred by the GP-HMC and the plain HMC algorithms and shows the differences between these two algorithms. Algorithm 1 is just a conceptual outline; the reader is referred to Algorithms S.1a, S.1b, S.1c and S.1d in the Supplement for a detailed pseudocode.

Algorithm 1 Conceptual outline for Hamiltonian Monte Carlo (HMC) in the left side column vs HMC coupled with emulation using Gaussian processes (GP-HMC) algorithm in the right side column. The total number of model (ODE/PDE) evaluations required for running each algorithm is: HMC – $\mathbf{SL(d+1)}$ vs GP-HMC – $\mathbf{S}$

Define a d-dimensional vector $\boldsymbol{\theta}$ with θ_k the k^{th} element, $k = 1,\ldots,d$; S: number of HMC samples; L: number of HMC trajectory steps; $p(\mathbf{y}\|\boldsymbol{\theta})$: simulator data likelihood (eq (1))	Define a d-dimensional vector $\boldsymbol{\theta}$ with θ_k the k^{th} element, $k = 1,\ldots,d$; S: number of HMC samples; L: number of HMC trajectory steps; $\tilde{p}(\mathbf{y}\|\boldsymbol{\theta})$: emulator data likelihood (eq (2))
for i = 1 : S **do**	**for** i = 1 : S **do**
for j = 1 : L **do**	**for** j = 1 : L **do**
Solve PDEs to get $\log p(\mathbf{y}\|\boldsymbol{\theta}^j)$ and $\frac{\partial \log p(\mathbf{y}\|\boldsymbol{\theta}^j)}{\partial\theta_k^j}$	Use GPs to predict $\log \tilde{p}(\mathbf{y}\|\boldsymbol{\theta}^j)$ and $\frac{\partial \log \tilde{p}(\mathbf{y}\|\boldsymbol{\theta}^j)}{\partial\theta_k^j}$
end for	**end for**
Propose $\boldsymbol{\theta}^L$	Propose $\boldsymbol{\theta}^L$
Solve ODEs/PDEs to get $\log p(\mathbf{y}\|\boldsymbol{\theta}^L)$ and accept/reject in a M-H step	Solve ODEs/PDEs to get $\log p(\mathbf{y}\|\boldsymbol{\theta}^L)$ and accept/reject in a M-H step
end for	**end for**

Algorithm 1 illustrates that HMC running in the original log likelihood space requires many more expensive model evaluations compared to HMC running in the emulated log likelihood space. This is because the former requires the numerical integration of the differential equations at every leapfrog step throughout the Hamiltonian dynamics, while the latter only requires one single numerical integration at the end of the leapfrog trajectory per HMC sample. An HMC trajectory typically has in the order of $L = 100$–1000 steps, which if carried out in the original space would require in the order of $100 \times (d+1)$ to $1000 \times (d+1)$ (d: number of parameters) model evaluations per HMC sample, thus a reduction in the computational complexity by about two to three orders of magnitude is obtained if the emulator provides a faithful representation of the log likelihood. The GP-HMC algorithm is guaranteed to satisfy detailed balance with respect to the correct posterior distribution, and in Sect. 4 of the Supplement we give the proof.

The GP-HMC algorithm proceeds as follows:

- *Initial design stage.* Choose parameter values from a space filling design in parameter space (e.g. a Latin hypercube McKay et al. 1979 or a Sobol sequence Bratley and Fox 1988), and perform an expensive model evaluation for each parameter vector to obtain the true log likelihood value. These training points are used to create a GP emulator.
- *Exploratory phase.* 'Learn' the form of the posterior distribution by running HMC in the space of the emulated posterior distribution, with the proposed point at the end of the HMC trajectory being subject to an M–H step based on the true posterior distribution (see Sect. 4 of the Supplement for more details on this). If the proposed parameter vector is accepted, add it together with the corresponding log likelihood value to the list of existing training points for the emulator, which is subsequently refined by maximisation of the log marginal likelihood with respect to the covariance hyperparameters—see Eq. 2 or 20 in the Supplement. Training points collected in the initial design stage are gradually removed as they tend to come from low posterior density regions and can bias the inference results. Points in low probability density regions could encourage overfitting by driving the GP lengthscale to small values due to large changes in output space happening over small distances in input space. Following Rasmussen (2003), we set the emulated 'potential energy' of the HMC algorithm (see Sect. 2.2) to

$$\tilde{E}(\boldsymbol{\theta}^*) = \frac{\mathbb{E}(f(\boldsymbol{\theta}^*)|D) - \sqrt{\mathrm{var}(f(\boldsymbol{\theta}^*)|D)}}{2\sigma^2} + \frac{n}{2}\log(2\pi\sigma^2) - \log p(\boldsymbol{\theta}^*). \quad (4)$$

Here $f(.)$ is the emulated RSS function, $\mathbb{E}(f(\boldsymbol{\theta}^*)|D)$ is the GP posterior predictive mean given the training points D (see Eq. 4 or 12 in the Supplement) and $\sqrt{\mathrm{var}(f(\boldsymbol{\theta}^*)|D)}$ is the GP posterior predictive standard deviation (see Eq. 5 or 13 in the Supplement) for RSS at unseen parameter configurations $\boldsymbol{\theta}^*$ conditional on the training points D. This drives the exploration into regions of high posterior probability (low value of $\mathbb{E}(.)$) or high uncertainty (large value of $\sqrt{\mathrm{var}(.)}$). If $\sqrt{\mathrm{var}(.)}$>3 along the HMC trajectory, the algorithm steps into a region of high uncertainty, where the GP needs to be further trained, thus the simulation is stopped prematurely before reaching the end of the trajectory and an expensive model evaluation is performed at this point.

- *Sampling phase.* Draw parameter samples by running HMC in the space of the emulated posterior distribution defined by the emulator constructed in the exploratory phase (the emulator is no longer refined at this stage). Proposed points are accepted/rejected in a M–H step according to the simulator (see Sect. 4 of the Supplement for more details). If the rejection rate is too high (e.g. above 35%), this indicates that the emulated posterior distribution is not an accurate enough representation of the original posterior distribution[5] and an extension of the exploratory phase is needed. We set the emulated 'potential energy' in the HMC algorithm to Rasmussen (2003):

$$\tilde{E}(\boldsymbol{\theta}^*) = \frac{\mathbb{E}(f(\boldsymbol{\theta}^*)|D)}{2\sigma^2} + \frac{n}{2}\log(2\pi\sigma^2) - \log p(\boldsymbol{\theta}^*), \quad (5)$$

where Eq. (5) gives the unnormalised log posterior probability.

We note that Eq. (4) is a modified version of (5) used in the exploratory phase only to aid exploration by subtracting a term dependent on the posterior predictive variance. The effect of this term is to shift the trade-off between exploration and exploitation towards exploration. A wider exploration during the exploratory phase is advisable to facilitate a better global coverage of the configuration space. This is a standard trick widely used in Bayesian optimization—see e.g. Shahriari et al. (2016).

Delayed acceptance In several studies (Golightly et al. 2015; Sherlock et al. 2017) it has been hypothesized that the delayed

[5] The accuracy of the emulator can be checked by diagnostics (Bastos and O'Hagan 2009).

Table 1 Table of abbreviations for the methods used in this study

Abbreviation	Explanation
HMC	Hamiltonian Monte Carlo
RMHMC	Riemann Manifold HMC
LDMC	Lagrangian dynamical Monte Carlo
NUTS	No U-turn sampler
GP-*x*	Combination of algorithm *x* (listed above) and Gaussian processes
DA-GP-*x*	GP-x algorithm (*x* listed above) coupled with delayed acceptance
noDA-GP-*x*	Standard (without delayed acceptance) GP-*x* algorithm (*x* listed above)
DA-GP-hybrid-*x*-*y*	Hybrid algorithm between DA-GP-*x* and DA-GP-*y* (*x*, *y* listed above)

acceptance scheme could potentially speed up simulations. This scheme is a two-stage acceptance procedure, with two M–H acceptance/rejection steps. The first step is a computationally fast pre-filter step, in which proposed values are accepted/rejected based on the emulator (first Eq. in (6)). Detailed balance with respect to the true posterior distribution is ensured through the second M–H step, which invokes the original posterior distribution for the acceptance/rejection of the those samples accepted in the first step (second Eq. in (6)).

$$\begin{aligned} \alpha_1(\boldsymbol{\theta}^*|\boldsymbol{\theta}) &= 1 \wedge \frac{\tilde{p}(\boldsymbol{\theta}^*|\mathbf{y})q(\boldsymbol{\theta}|\boldsymbol{\theta}^*)}{\tilde{p}(\boldsymbol{\theta}|\mathbf{y})q(\boldsymbol{\theta}^*|\boldsymbol{\theta})}, \\ \alpha_2(\boldsymbol{\theta}^*|\boldsymbol{\theta}) &= 1 \wedge \frac{p(\boldsymbol{\theta}^*|\mathbf{y})}{p(\boldsymbol{\theta}|\mathbf{y})}\frac{\tilde{p}(\boldsymbol{\theta}|\mathbf{y})}{\tilde{p}(\boldsymbol{\theta}^*|\mathbf{y})}, \end{aligned} \tag{6}$$

where $q(.)$ is the proposal density and $a \wedge b = \min\{a, b\}$. If a proposal move is rejected based on the emulator in the first step, the computationally expensive second step will never have to be carried out.

3 Methodological contribution

Our methodological contributions are as follows. Firstly, we generalise Rasmussen's GP-HMC algorithm (Rasmussen 2003) to more modern HMC variants, namely GP-NUTS, GP-RMHMC and GP-LDMC (see abbreviations in Table 1), include formal detailed balance proofs, and show how these subsume the proof of the GP-HMC algorithm as a limiting case. Secondly, we extend GP-HMC and its variations by including DA, with formal detailed balance proofs for the following algorithms: DA-GP-HMC, DA-GP-NUTS, DA-GP-RMHMC and DA-GP-LDMC (see abbreviations in Table 1). Thirdly, we assess the computational efficiency and accuracy of all these emulation methods in simulation studies.

While HMC coupled with emulation has been proposed in the literature before (Rasmussen 2003; Lan et al. 2016), our study is the first to provide, collate and unify detailed balance proofs in a theoretical framework for all algorithms listed in Table 1. Section 4 in the Supplement offers detailed balance proofs for HMC, RMHMC and LDMC, and we show how the former two are limiting cases of the latter. The RMHMC proofs (Sects. 4.3 and 4.4 in the Supplement) are a special case of the LDMC proofs (Sects. 4.5 and 4.6 in the Supplement), with the proposal probability ratio set to 1 (rather than the Jacobian of the variable transformation). The HMC proofs (Sects. 4.1 and 4.2 in the Supplement), are a more specialised case even still, with a constant mass matrix. We mention that Lan et al. (2015) prove detailed balance for the vanilla LDMC algorithm, and our contribution is to extend this proof to LDMC with emulation (Sects. 4.5 and 4.6 in the Supplement). Moreover, we extend the DA proof in Sherlock et al. (2017) from random walk M–H to HMC. These proofs set the basis for the more complex proof of NUTS coupled with emulation (with and without DA, in Sects. 4.7 and 4.8 of the Supplement), which we regard as our main theoretical contribution.

Throughout this work, ergodicity of the system defined by the various MCMC samplers is assumed to hold. Ergodicity could be proven by adapting the proofs in Livingstone et al. (2019), Banterle et al. (2019), Sherlock et al. (2017) and Conrad et al. (2016) to the generalised scenario where proposal moves are based on an emulated log likelihood, but this is beyond the scope of the present paper.

3.1 Brief discussion on efficiency for all algorithms proposed

The noDA-GP-NUTS algorithm (see abbreviations in Table 1) requires evaluation of the differential equations several times (equal to the number of tree doublings, i.e. tree height) along the trajectory before making a proposal; see the proof in Sect. 4.7 in the Supplement, in particular Eq. (37), which shows that the simulator potential function $E(\boldsymbol{\theta})$ needs evaluating for every tree doubling. In contrast, DA-GP-NUTS only evaluates the ODEs/PDEs once at the end of the trajectory, for the final proposed point (as do all the other algorithms investigated: noDA-GP-HMC, DA-

GP-HMC, noDA-GP-RMHMC, DA-GP-RMHMC, noDA-GP-LDMC, DA-GP-LDMC—see abbreviations in Table 1). This implies that the noDA-GP-NUTS requires a much larger number of forward evaluations (roughly one order of magnitude larger) than the DA-GP-NUTS. For this reason, we did not implement this algorithm. We employ all the other algorithms: noDA-GP-HMC (which is the standard GP-HMC), DA-GP-HMC, DA-GP-NUTS, noDA-GP-RMHMC, DA-GP-RMHMC, noDA-GP-LDMC, DA-GP-LDMC (Table 1) on a number of ODE/PDE problems, presented in Sect. 4.

4 ODE/PDE test examples

This section describes the test examples on which we applied the emulation methods under consideration.

4.1 Sinusoidal example

We introduce a linear ODE model, which is a sinusoidal toy problem, defined via the expression

$$\frac{d^2 f}{dt^2} + B^2 f = 0 \quad \Leftrightarrow \quad \frac{df}{dt} = z, \quad \frac{dz}{dt} + B^2 f = 0. \tag{7}$$

This ODE model has an analytical solution: $f(t) = A\sin(B(t + C))$, where A: amplitude, $\frac{2\pi}{B}$: period, C: phase shift are the unknown parameters, which are estimated from the data. We only considered one period, $t \in [0, 2\pi]$, to ensure unimodality of the likelihood. We simulated data at 50 equally spaced time points in the range $[0, 2\pi]$. We set the true parameter values as: $A = 3$, $B = 1$ and $C = 0.05$. We added iid additive Gaussian noise to the clean data generated using Eq. (7), and we set the variance of the noise σ^2 to 0.12 (signal-to-noise, SNR, set to 80). An illustration of the data is given in panel (a) of Fig. 1.

The task in the assessment of the proposed sampling schemes was to estimate the parameters A, B, C. The noise variance was kept fixed at its true value, however this simplification was relaxed for the next examples. We placed a Gaussian prior distribution on the log of the parameters to ensure positivity ($\log A \sim \mathcal{N}(\log 4, 0.02)$, $\log B \sim \mathcal{N}(\log 1, 0.01)$, $\log C \sim \mathcal{N}(\log 0.05, 0.05)$). These priors were chosen to encourage unimodality of the posterior distribution.

4.2 FitzHugh–Nagumo

The FitzHugh–Nagumo model, developed by Fitzhugh (1961) and Nagumo et al. (1962) to model the behaviour of spike potentials in the giant axon of neurons, is defined as a nonlinear ODE model:

$$\frac{dV}{dt} = \gamma\left(V - \frac{V^3}{3} + R\right), \quad \frac{dR}{dt} = \frac{1}{\gamma}(V - \alpha + \beta R). \tag{8}$$

The system describes the reciprocal dependency between the voltage V across an axon membrane (characterising the self-excitation of the axon membrane) and the recovery R, acting as outwards currents (providing a feedback response). This example is a widely used benchmark problem, with the model having been used as a mathematical representation for cardiac dynamics (Göktepe and Kuhl 2009) and neuro-degenerative diseases (Brüggemeier et al. 2014). This example deviates from the regular sinusoidal oscillations of the first toy example, with Eq. (8) defining a highly nonlinear likelihood surface (Ramsay et al. 2007) as the three parameters α, β, γ are varied. The FitzHugh–Nagumo model has proven to be a very challenging problem for alternative inference methods, e.g. based on gradient matching (Campbell and Steele 2012), thus making it an ideal test bed for our inference procedure.

We followed Campbell and Steele (2012) and generated data from the model with the following parameter values: $\alpha = 0.2, \beta = 0.2, \gamma = 3$, and initial values $(V_0, R_0) = (-1, 1)$. We used 100 time points equally spaced in the interval $[0, 20]$ ms. As in Campbell and Steele (2012), we added iid additive Gaussian noise to the data, with the following variances: $\sigma_V^2 = 0.25, \sigma_R^2 = 0.16$ ($\text{SNR}_V = 9$, $\text{SNR}_R = 3$). A depiction of the data can be seen in panel (b) of Fig. 1.

In our estimation procedure we learnt five parameters: $\alpha, \beta, \gamma, \sigma_V^2, \sigma_R^2$ from data from the two 'species' V and R. Following the study in Chapter 8 of Lawrence et al. (2010), we set a Gamma(2,1) prior for parameters α, β, γ, while for the noise variances: σ_V^2, σ_R^2 we chose a weakly informative Inverse-Gamma(0.001, 0.001) prior.

4.3 Biochemical signalling pathway

The biochemical signalling pathway model, characterised by the nonlinear ODE system in Eq. (9), uses the Michaelis–Menten kinetic law to describe the activation of a protein R into its active form R_{pp} in the presence of an enzyme S, followed by the degradation of the enzyme into its inactive form, D (see panel (a) in Fig. 2 for a graphical representation).

$$\begin{aligned} \frac{dS}{dt} &= -k_1 S, \quad \frac{dD}{dt} = k_1 S, \\ \frac{dR}{dt} &= -\frac{V_1 R S}{K_{m_1} + R} + \frac{V_2 R_{pp}}{K_{m_2} + R_{pp}}, \\ \frac{dR_{pp}}{dt} &= \frac{V_1 R S}{K_{m_1} + R} - \frac{V_2 R_{pp}}{K_{m_2} + R_{pp}}. \end{aligned} \tag{9}$$

Cell signalling has been used in cancer modelling (Martin 2004) and modelling of neuro-degenerative diseases (Kim and Choi 2010). The system of eqns in (9) depends on five

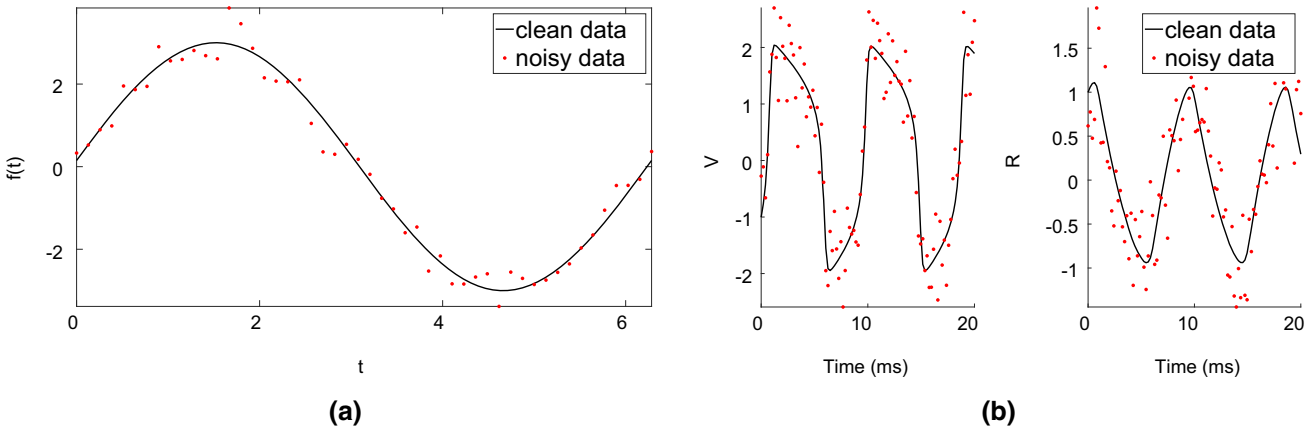

Fig. 1 Panel **a** An example of noise-free data generated from the sinusoidal model using Eq. (7) (continuous black line). To this we added iid additive Gaussian noise with variance 0.12 (red dots), and the noisy data were used in the inference procedure. Panel **b** An example of noise-free data generated from the FitzHugh–Nagumo model using Eq. (8) (continuous black line). To these we added iid additive Gaussian noise with variance 0.25—left signal and 0.16—right signal (red dots). Data from the two signals were used for the inference procedure. (Color figure online)

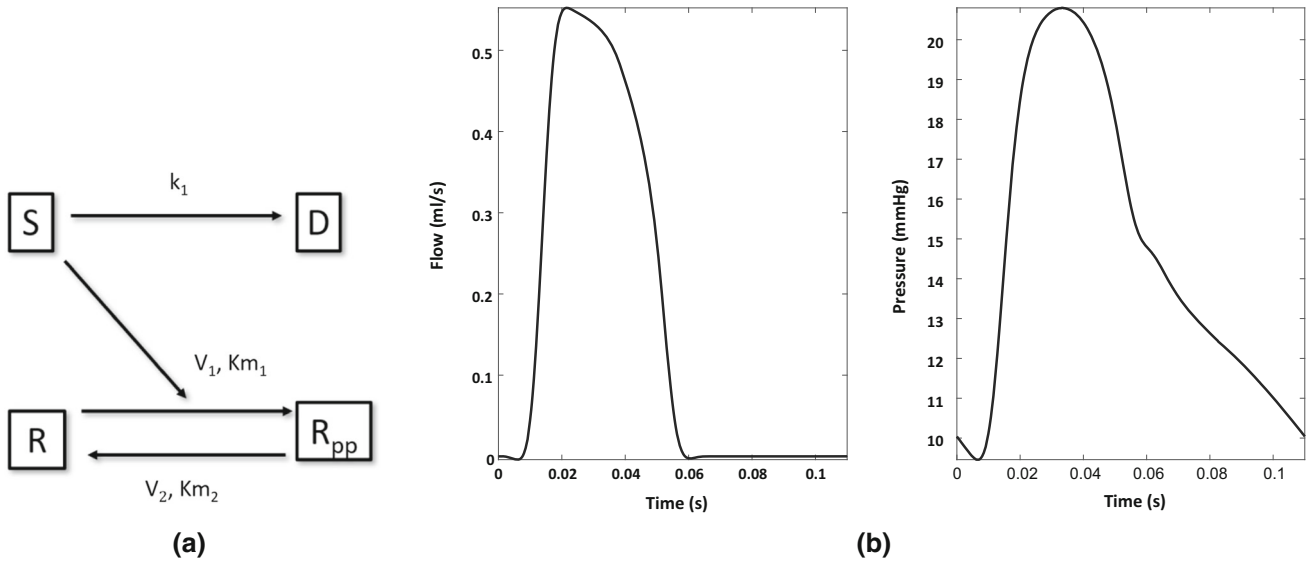

Fig. 2 Panel **a**: Graphical representation of the protein signalling pathway in Eq. (9). The model uses the Michaelis–Menten kinetic law to describe the activation of a protein R into its active form R_{pp} in the presence of an enzyme S, followed by the degradation of the enzyme into its inactive form D. Figure adapted from Chapter 8 in Lawrence et al. (2010). Panel **b**: Measured pulmonary blood flow (left) and pressure (right) in the main pulmonary artery of a healthy mouse. The flow data were used as inlet boundary conditions in the PDEs (10), while the pressure data constituted the main signal used for inference

kinetic parameters: k_1, V_1, K_{m_1}, V_2, K_{m_2}, which control how fast the biochemical processes involving the five 'species' (S, D, R, R_{pp}) take place.

We followed the study in Chapter 8 of Lawrence et al. (2010) and generated data from the model with the following parameter values: $k_1 = 0.05$, $V_1 = 0.2$, $K_{m_1} = 0.1$, $V_2 = 0.1$, $K_{m_2} = 0.1$, and initial values $(S_0, D_0, R_0, R_{pp_0}) = (1, 0, 1, 0)$. We used 20 data points within the interval [0, 100] s, measured at time points {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 18, 20, 25, 30, 40, 60, 80, 100}. We added iid additive Gaussian noise to the data, with the variance of 0.0004 (Lawrence et al. 2010) for all 'species' ($\text{SNR}_S = \text{SNR}_D = 270$, $\text{SNR}_R = \text{SNR}_{R_{pp}} = 130$). A depiction of the data can be seen in Fig. 3.

We estimated five parameters: k_1, V_1, K_{m_1}, V_2, K_{m_2} from data from the four 'species': S, D, R, R_{pp}. Following Lawrence et al. (2010), we set a Gamma(1,1) prior for parameters k_1, V_1, K_{m_1}, V_2, K_{m_2}, while for the noise variances: $\sigma_S^2, \sigma_D^2, \sigma_R^2, \sigma_{R_{pp}}^2$ we chose a weakly informative Inverse-Gamma(0.001, 0.001) prior.

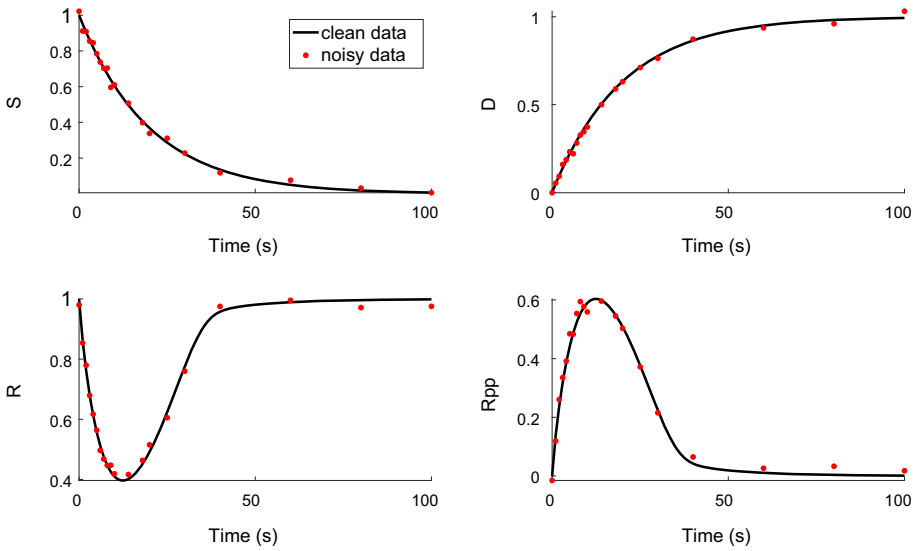

Fig. 3 An example of noise-free data generated from the biochemical signalling pathway model using Eq. (9) (continuous black line). To these we added iid additive Gaussian noise with variance 0.0004 (Lawrence et al. 2010) for all four signals (red dots). Data from all four signals were used for the inference procedure. (Color figure online)

4.4 Real-world application: fluid-dynamics model of the pulmonary blood circulation

One dimensional (1D) fluid-dynamics models are used to model the pulmonary blood circulation (Qureshi et al. 2017, 2018), and enable the non-intrusive diagnostication of pulmonary hypertension (high blood pressure in the pulmonary system, i.e. the blood vessel network connected to the right ventricle of the heart). If left untreated, pulmonary hypertension may lead to coronary artery disease, stroke and heart failure. Currently, the diagnosis process requires knowledge of the pulmonary blood pressure, which can only be obtained invasively for patients using right-heart catheterisation (unlike for the systemic circulation, i.e. the blood vessel network connected to the left ventricle of the heart, for which a sphygmomanometer may be used non-invasively). The ultimate goal is to combine magnetic resonance imaging (MRI) with mathematical modelling and statistical inference to develop a non-invasive alternative. The 1D fluid-dynamics model (Qureshi et al. 2017, 2018) is described by a set of nonlinear PDEs:

$$\frac{\partial A}{\partial t}+\frac{\partial q}{\partial x}=0,\quad \frac{\partial q}{\partial t}+\frac{\partial}{\partial x}\frac{q^2}{A}+\frac{A}{\rho}\frac{\partial p}{\partial x}=-\frac{2\pi\mu r}{\delta}\frac{q}{A},$$
$$p=p_0+\frac{4}{3}s\left(1-\sqrt{\frac{A_0}{A}}\right), \tag{10}$$

where x (cm) and t (s) are the longitudinal spatial and temporal coordinates, p (mmHg) is the blood pressure, q (ml/s) is the blood flow rate, A (cm^2) is the cross-sectional area. $A(x,t)=\pi r(x,t)^2$, where $r(x,t)$ (cm) is the vessel radius. Also, A_0 is the unstressed vessel cross-sectional area, p_0 (mmHg) is the transmural pressure, s (mmHg) is the arterial network stiffness (kept constant across the network of vessels), $\rho=1.055$ g/ml is the blood density, $\mu=0.049$ g/(cm s) is the viscosity and $\delta=\sqrt{\mu T/2\pi\rho}$ cm is the boundary-layer thickness of the velocity profile. The boundary conditions are specified as follows: As inlet boundary conditions, an inlet flow at the main pulmonary artery (MPA) was prescribed, which was measured with ultrasound—see the left panel in Fig. 2b. At the vessel junctions, conservation of flow ($q_{\mathrm{p}}=q_{\mathrm{d}_1}+q_{\mathrm{d}_2}$) and continuity of pressure ($p_{\mathrm{p}}=p_{\mathrm{d}_1}=p_{\mathrm{d}_2}$) were ensured; here p represents the parent vessel, and d_1 and d_2 represent the daughter vessels. As outflow boundary conditions, three-element Windkessel models (two resistors R_1 and R_2, and a capacitor, C) were attached at every terminal artery, and are of the form $Z(\omega)=R_1+\frac{R_2}{1+i\omega CR_2}\implies q(L,t)=\frac{1}{T}\int_0^T p(L,t-\tau)Z(\tau)d\tau$, where $Z(\omega)$ is the impedance, ω is the angular frequency, T is the length of the cardiac cycle, R_1, R_2 (mmHg s/ml) are the two resistances, and C (ml/mmHg) is the capacitance.

The three Windkessel elements (R_1, R_2, C) vary across the different terminal arteries. The nominal values for every terminal vessel j were calculated first ($R_{01}^j, R_{02}^j, C_0^j$ in Eq. (11)), see Qureshi et al. (2018) for details, then global scaling factors r_1, r_2, c for these estimates were introduced, and they were kept constant across all terminal arteries:

$$R_1^j=(1-0.5r_1)R_{01}^j,\quad R_2^j=(1-0.5r_2)R_{02}^j,$$
$$C^j=(1-0.5c)C_0^j. \tag{11}$$

We used measured blood pressure (a time series consisting of 512 temporal pressure points) in the MPA to infer the various biophysical parameters: s, r_1, r_2, c. An example of the pulmonary blood pressure time series in a healthy mouse is given in the right panel of Fig. 2b.

The parameters lie within biologically plausible ranges, as established by Qureshi et al. (2018): $s\in[7\times10^4, 5\times$

$10^5]$, $r_1 \in [-0.5, 1.92]$, $r_2 \in [-0.5, 1.0]$, $c \in [-2.5, 1.5]$). We set a rescaled Beta(1, 1) prior for the parameters to enforce that they lie within the prescribed ranges, while for the noise variance σ^2 we chose a weakly informative Inverse-Gamma(0.001, 0.001) prior.

5 Simulations

5.1 Software

Our statistical methods were implemented in `Matlab` (Mathworks, Natick, MA) and simulations were run on a RedHat Enterprise Linux 6 machine with Intel(R) Xeon(R) CPU E5-2680 v2 2.80 GHz and 32 GB RAM.

We constructed the GP models using the `GPstuff` toolbox (Vanhatalo et al. 2013) and the MCMC convergence diagnostics (multivariate potential scale reduction factor, MPSRF Brooks and Gelman 1998 and ESS Kass et al. 1998) made use of functions from the MCMC toolbox (Laine 2007). To run the No U-turn sampler, Riemann Manifold HMC and Lagrangian Dynamical MC, we used the Matlab implementations developed by the authors of the papers where these algorithms were proposed (Hoffman and Gelman 2014; Girolami and Calderhead 2011; Lan et al. 2015).

For the numerical integration of the ODEs, we used the *ode15s* Matlab solver for the FitzHugh–Nagumo model, and the *ode23* Matlab solver for the biochemical signalling pathway model. The PDEs of the 1D fluid-dynamics model were numerically integrated using a two step Lax–Wendroff scheme (Lax and Wendroff 1960) implemented in `C++` by Qureshi et al. (2018); Olufsen et al. (2000). We note that for the two synthetic examples (FitzHugh–Nagumo and biochemical pathway), we have used the same numerical integrator for data generation and for inference. Given that we do not allow for potential signal distortion caused by the numerical integration, our inference results are therefore in principle over-optimistic. However, since our dynamical systems have well-behaved (as opposed to chaotic) attractors (a periodic limit cycle for the FitzHugh–Nagumo example and a stable fixed point for the biochemical pathway example), the effect of the numerical integration is practically negligible.

Our software implementation and data sets are available at https://github.com/LMihaelaP/Hamiltonian-Monte-Carlo-with-emulation.git.

5.2 Method implementation details

5.2.1 GP kernel

For the GP emulator of RSS, we used a squared exponential kernel (see Sect. 4.2 in Rasmussen and Williams 2005), which is infinitely differentiable,[6] allowing to analytically compute[7] first-order derivatives of the GP posterior predictive mean and variance (Eqs. 4, 5, or 12, 13 in the Supplement) needed in HMC, and second- and third-order derivatives of the GP posterior predictive mean, needed in RMHMC and LDMC. We note that differentiation is a linear operator, so the derivative of a GP is again a GP for differentiable kernels; see Sect. 9.4 in Rasmussen and Williams (2005).

5.2.2 GP mean

For the sinusoidal, FitzHugh–Nagumo and pulmonary models we used a zero mean GP prior, while for the biochemical pathway model we used a second-order polynomial for the mean basis functions (for reasons discussed in Sect. 6.3), i.e. in 5D: $\mathbf{h}(\mathbf{x}) = [1 \quad x_1 \quad \dots \quad x_5 \quad x_1^2 \dots x_5^2 \quad x_1x_2 \dots x_4x_5]^{\mathrm{T}}$. Further relevant implementation details are offered in the Supplement, Sect. 5.1.

5.2.3 Multiple GPs

For problems with multiple 'species' (biochemical pathway and FitzHugh–Nagumo examples), we emulated the RSS for every 'species' independently, thus Eq. (2) becomes $\tilde{p}(\mathbf{y}|\boldsymbol{\theta}, \boldsymbol{\sigma^2}) = \prod_{j=1}^{J} \left(\left(\frac{1}{\sqrt{2\pi\sigma_j^2}} \right)^{n_j} \exp\left(-\frac{\tilde{\mathcal{S}}_j(\boldsymbol{\theta})}{2\sigma_j^2} \right) \right)$, where J is the total number of 'species'.

5.2.4 Data sets

For all three synthetic examples (sinusoidal, FitzHugh–Nagumo and biochemical pathway) we generated 10 synthetic data sets with different noise instantiations. For the pulmonary example we had one single measured data set.

5.2.5 GP-HMC phases

Initial design stage At the initial stage of the GP-HMC algorithm (Rasmussen 2003), the number of training points can be determined by quantifying the efficiency of the MCMC sampler in the beginning of the exploratory phase (the acceptance rate of the sampler run using the initial emulator should not be too low, e.g. below 20%). The Supplement, Sect. 5.3 offers more implementation details.

Exploratory phase In the exploratory phase of GP-HMC (Rasmussen 2003) we tried to ensure that a minimum num-

[6] For example, the Matèrn 3/2 or 5/2 kernels cannot be used for the RMHMC and LDMC algorithms, as they are once or twice differentiable, respectively.

[7] We checked the analytical derivative against the numerical derivative; a difference in Euclidean space below a small threshold, e.g. 0.1%, was taken as an indication that the two were in agreement.

ber of training points is stored (to boost computational efficiency), while preserving a high enough emulator accuracy (as quantified by GP diagnostics (Bastos and O'Hagan 2009)), and a high acceptance rate in the sampling phase (>65% (Neal 2011)). Initially $100 \times d$ (d: parameter dimensionality) training points can be used, and if the acceptance rate in the sampling phase is sub-optimal (<65%), the exploratory phase can be extended. Generally this number depends on the parameter dimensionality and the complexity of the posterior distribution. For example, we found the rule of thumb presented in a Bayesian optimisation context (Jones et al. 1998), $10 \times d$, inadequate, providing a sub-optimal emulator in an MCMC context.

The noise variance was kept fixed in the first part of the exploratory phase to enable learning the parameters while avoiding changes in curvature of the log likelihood induced by varying the noise variance (last term of Eq. (1)). An informed initial guess can be calculated based on the RSS value obtained from fitting a nonlinear regression model to the data. The acceptance rate in the exploratory phase of the algorithm can then act as an objective metric to assess the appropriateness of the σ^2 value. If the acceptance rate is too low, e.g. <10%, σ^2 is too large, making the likelihood in eqns (1) and (2) peaked and causing a large discrepancy between the emulated and original likelihood. If the acceptance rate is too large, e.g. >90%, σ^2 should be reduced to allow the sampler to focus on the area of interest and avoid flattening out the likelihood landscape. The σ^2 value can be sequentially altered by, e.g. 10%, and the exploratory phase re-run with a monitoring of the acceptance rate. We note here that we allow σ^2 to depend on the algorithmic implementation in the exploratory phase only, when the emulator is not a faithful representation of the simulator. In the sampling phase of the algorithm, σ^2 is drawn from the conditional posterior distribution, conditional on the data and dependent on the model. Further implementation details from the exploratory phase of the algorithm are given in the Supplement, Sect. 5.3.

Sampling phase In the sampling phase of GP-HMC (Rasmussen 2003) we allowed for 100 samples as burn-in phase (chosen to ensure that MPSRF ≤ 1.1 (Brooks and Gelman 1998)), and 2000 samples were subsequently drawn and used for inference. Besides sampling the model parameters, we also sampled the noise variance in a Gibbs step. Every algorithm (DA-GP-HMC, noDA-GP-HMC, DA-GP-NUTS, DA-GP-RMHMC, noDA-GP-RMHMC, DA-GP-LDMC, noDA-GP-LDMC) was run 10 times from different random number generator seeds and different starting values for the parameters, selected from the points collected in the exploratory phase, to make it less likely that we start in a low probability region. For each of the 10 simulations, we recorded ESS for each parameter:

$$\mathrm{ESS}^i = \frac{N}{1 + 2\sum_{l=1}^{\infty} \rho_l^i}, \quad i = 1, \ldots, d \tag{12}$$

where d: parameter dimensionality, and using Eq. (12), we calculated the minimum, median and maximum ESS across parameters as:

$$\begin{aligned} \mathrm{MinESS} &= \min_i \left(\mathrm{ESS}^i\right), \\ \mathrm{MedianESS} &= \mathrm{median}_i \left(\mathrm{ESS}^i\right), \\ \mathrm{MaxESS} &= \max_i \left(\mathrm{ESS}^i\right). \end{aligned} \tag{13}$$

Details about the implementation of Bayesian Optimisation for parameter tuning of the various algorithms can be found in Sect. 5.4 of the Supplement, and details about required parameter transformations are in Sect. 5.5 of the Supplement.

5.3 Setting the mass matrix for RMHMC/LDMC

For second-order algorithms (RMHMC, LDMC) obtaining the metric tensor is necessary. This is not trivial when emulating the objective function. Equation (3) shows what the mass matrix can be set to to ensure that it is a proper metric tensor, i.e. positive definite, and distances $\Delta\boldsymbol{\theta}^{\mathrm{T}}\mathbf{M}(\boldsymbol{\theta})\Delta\boldsymbol{\theta}$ between probability distribution functions $p(\mathbf{y}, \boldsymbol{\theta})$ and $p(\mathbf{y}, \boldsymbol{\theta} + \Delta\boldsymbol{\theta})$ on the Riemann manifold do not depend on the parameterisation of the statistical model, i.e. they are invariant with respect to nonlinear parameter transformations (see Sect. 3.2 in Calderhead 2012 and Sect. 5.4.2 in Murphy 2012 for more details). Emulating the objective function (RSS) instead of the model output results in information loss (Davies et al. 2019), and hence inability of calculating the expectation in Eq. (3). More precisely, for the model defined in Eq. (1), we may take the log to obtain

$$\log p(\mathbf{y}|\boldsymbol{\theta}, \sigma^2) = \left(-\frac{n}{2}\log(2\pi\sigma^2)\right) - \frac{\sum_{i=1}^{n}(y_i - m_i(\boldsymbol{\theta}))^2}{2\sigma^2}, \tag{14}$$

and the corresponding first- and second-order derivatives are:

$$\frac{\partial \log p(\mathbf{y}|\boldsymbol{\theta}, \sigma^2)}{\theta_j} = \frac{1}{\sigma^2}\sum_{i=1}^{n}\left((y_i - m_i(\boldsymbol{\theta}))\frac{\partial m_i(\boldsymbol{\theta})}{\partial \theta_j}\right), \tag{15}$$

$$\begin{aligned} \frac{\partial^2 \log p(\mathbf{y}|\boldsymbol{\theta}, \sigma^2)}{\theta_k \theta_j} = \frac{1}{\sigma^2}\sum_{i=1}^{n}\Bigg(&-\left(\frac{\partial m_i(\boldsymbol{\theta})}{\partial \theta_k}\frac{\partial m_i(\boldsymbol{\theta})}{\partial \theta_j}\right) \\ &+ (y_i - m_i(\boldsymbol{\theta}))\frac{\partial^2 m_i(\boldsymbol{\theta})}{\partial \theta_k \partial \theta_j}\Bigg). \end{aligned} \tag{16}$$

By noting that $\mathbb{E}(\mathbf{y}|\boldsymbol{\theta}) = \mathbf{m}(\boldsymbol{\theta})$ and taking the expectation with respect to $\mathbf{y}|\boldsymbol{\theta}$ of the negative term in (16):

$$\mathbb{E}_{\mathbf{y}|\boldsymbol{\theta}}\left(-\frac{\partial^2 \log p(\mathbf{y}|\boldsymbol{\theta},\sigma^2)}{\theta_k\theta_j}\right) = \frac{1}{\sigma^2}\sum_{i=1}^{n}\left(\frac{\partial m_i(\boldsymbol{\theta})}{\partial\theta_k}\frac{\partial m_i(\boldsymbol{\theta})}{\partial\theta_j}\right). \tag{17}$$

Given that we emulate the expression $\sum_{i=1}^{n}(y_i - m_i(\boldsymbol{\theta}))^2$, we do not know the individual terms inside the sum, hence taking the expectation of the expression in Eq. (16) is impossible. Instead, we can set $\mathbf{M}$ to be the observed Fisher Information matrix (the matrix of negative second-order derivatives of the log likelihood), plus the negative Hessian matrix of the log prior, i.e.

$$\begin{aligned} M_{i,j} &= -\frac{\partial^2 \log p(\mathbf{y},\boldsymbol{\theta})}{\partial\theta_i\partial\theta_j} \\ &= -\frac{\partial^2 \log p(\mathbf{y}|\boldsymbol{\theta})}{\partial\theta_i\partial\theta_j} + \left(-\frac{\partial^2 \log p(\boldsymbol{\theta})}{\partial\theta_i\partial\theta_j}\right). \end{aligned} \tag{18}$$

However, this matrix is not guaranteed to be positive definite. In that case, we set $\mathbf{M}$ to be

$$\begin{aligned} M_{i,j} &= \lambda\left(-\frac{\partial^2 \log p(\mathbf{y},\boldsymbol{\theta})}{\partial\theta_i\partial\theta_j}\right) \\ &\quad + (1-\lambda)\left(\frac{\partial \log p(\mathbf{y},\boldsymbol{\theta})}{\partial\theta_i}\frac{\partial \log p(\mathbf{y},\boldsymbol{\theta})}{\partial\theta_j}\right), \end{aligned} \tag{19}$$

where $\lambda = 0$ leads to the empirical Fisher information matrix, which is semi-positive definite by construction. Thus the form in Eq. (19) ensures that the mass matrix is at least semi-positive definite for a small enough $\lambda \in [0, 1]$. This form is motivated by the fact that the two terms are identical in expectation over $\mathbf{y}|\boldsymbol{\theta}$ for a constant prior, and may lead to a lower condition number of the mass matrix (hence numerical stability) than setting λ exactly equal to 0.

We found that setting the metric tensor as per Eq. (19) does not work for all parameter values tried by the sampler. In some cases, parameter values throughout the trajectory tend to take extreme, unrealistic values, subsequently leading to numerical instabilities in the ODE solver, which is called for the end trajectory value. In that case, the approach taken is akin to a Quasi-Newton method in optimisation (Broyden 1972), i.e. if at any point throughout the trajectory, the mass matrix is numerically unstable, the simulation within the trajectory is stopped prematurely before reaching the end. A new simulation is started from the beginning of the trajectory and the HMC algorithm is used instead of RMHMC or LDMC for that particular iteration. The resulting posterior samples will have been drawn using a hybrid version of HMC and RMHMC or LDMC, and the proposed algorithm is called hybrid-HMC-RMHMC or hybrid-HMC-LDMC. This algorithm naturally satisfies detailed balance since each sample is drawn using a valid sampler (either HMC or RMHMC/LDMC).

5.4 Sampler consistency checks

To check the mathematical and coding correctness of the samplers derived, we implemented the Geweke consistency test (Geweke 2004), reviewed in the Supplement, Sect. 3.

6 Numerical results

For all test examples and emulation methods implemented in our study we checked for MCMC convergence using MPSRF, and an MPSRF ≤ 1.1 was recorded for all simulations, thus there was no evidence of lack of convergence.

6.1 Sinusoidal example

6.1.1 Geweke consistency test: mathematical and coding correctness of the sampler

We checked the mathematical and coding correctness of the samplers' implementation using the Geweke consistency test (Geweke 2004). This was done for the linear ODE sinusoidal example to ensure no high computational costs attributed to a large number of ODE numerical integrations were incurred. In Fig. 1 of the Supplement we show QQ plots, i.e. quantiles of the prior distribution against quantiles of the ensemble of posterior distributions for all parameters (A, B, C) for three of the algorithms: DA-GP-HMC (top panel), noDA-GP-HMC (middle panel), DA-GP-NUTS (bottom panel), see Table 1 for abbreviation explanations. The points lie on the equality line, and we take this as evidence that the implementation of the three samplers is correct. We extrapolate this conclusion to the other samplers: (no)DA-GP-RMHMC and (no)DA-GP-LDMC since by looking at the proofs in Sect. 4 of the Supplement, we notice that they are a straightforward extension of (no)DA-GP-HMC.

6.1.2 DA versus noDA

Next, we investigated whether the DA scheme offers any gains in computational efficiency. To quantify efficiency, for each of the 10 data sets, we calculated the median ESS across all parameters, as defined in Eq. (13), for all DA Hamiltonian/Lagrangian algorithms against their noDA alternative (with the exception of NUTS, as justified in Sect. 3.1). Using hypothesis testing, for the DA and noDA schemes, we tested for mean equality of the median ESS values normalised by the total number of MCMC samples N, and by the number

Table 2 Accuracy in parameter and functional space for the sinusoidal example For each of the emulation methods compared we show the mean and standard deviation of the parameter posterior medians for 10 data sets, calculated using Eq. (22). The true parameter values are also shown. R^2 (Eq. (23)), as a measure of fit to the data is also displayed.

Legend: HMC—Hamiltonian Monte Carlo, NUTS: No U-turn sampler, RMHMC—Riemann Manifold Hamiltonian Monte Carlo, LDMC—Lagrangian Dynamical Monte Carlo, DA—delayed acceptance, GP: Gaussian process

Algorithm	A	B	C	R^2
DA-GP-HMC	3.0337 (0.0659)	1.0013 (0.0052)	0.0507 (0.0036)	0.9725 (0.0040)
DA-GP-NUTS	3.0348 (0.0645)	1.0012 (0.0052)	0.0508 (0.0032)	0.9725 (0.0040)
DA-GP-RMHMC	3.0358 (0.0656)	1.0011 (0.0053)	0.0505 (0.0035)	0.9725 (0.0040)
DA-GP-LDMC	3.0348 (0.0650)	1.0014 (0.0052)	0.0506 (0.0035)	0.9725 (0.0040)
True value	3	1	0.05	–

of forward (ODE) evaluations:

$$H_0 : \mu_{\text{ESS}}^{\text{noDA}} = \mu_{\text{ESS}}^{\text{DA}} \quad \text{versus} \quad H_1 : \mu_{\text{ESS}}^{\text{noDA}} \neq \mu_{\text{ESS}}^{\text{DA}}. \tag{20}$$

In Eq. (20), $\mu_{\text{ESS}}^{\text{noDA}}$ is the mean ESS for the noDA method in the sample of 10 medians from the 10 data sets, i.e.

$$\mu_{\text{ESS}}^{\text{noDA}} = \frac{1}{K}\sum_{k=1}^{K}[ESS]_k^{\text{noDA}}, \quad K = 10, \tag{21}$$

where $[ESS]_k^{\text{noDA}}$ is the median ESS across all parameters (Eq. (13)) for the k^{th} data set corresponding to the noDA method. In contrast, $\mu_{\text{ESS}}^{\text{DA}}$ is the mean ESS for the DA method in the sample of 10 medians from 10 data sets. The distribution of the 10 ESS values for the DA and noDA methods is shown in Fig. 2(a) of the Supplement. The test revealed a p-value > 0.05, hence we cannot reject the null hypothesis. Given that our next goal was to compare the different algorithms, we made the arbitrary choice of going ahead with the DA algorithms.

6.1.3 Accuracy

We first compare the performance of the DA-GP-HMC, DA-GP-NUTS, DA-GP-RMHMC and DA-GP-LDMC algorithms (see Table 1 for abbreviation explanations) in terms of accuracy in both parameter and functional space.

Parameter space Table 2 illustrates the mean of all ODE parameter posterior medians (i.e. the mean of the K posterior medians from the K data sets), and the corresponding standard deviation, i.e.

$$\begin{aligned}\bar{\boldsymbol{\theta}} &= \frac{1}{K}\sum_{k=1}^{K}[\boldsymbol{\theta}]_k,\\ \sigma(\boldsymbol{\theta}) &= \sqrt{\frac{1}{K-1}\sum_{k=1}^{K}([\boldsymbol{\theta}]_k - \bar{\boldsymbol{\theta}})^2},\end{aligned} \tag{22}$$

where $[\boldsymbol{\theta}]_k$ is the parameter posterior median vector for the k^{th} data set. More specifically, for each of the K data sets, we find the parameter posterior median. We then take the mean over all K posterior medians for every parameter, which is compared to the true parameter value, as a measure of accuracy and to test for unbiasedness. For a large enough number of data sets, K, this mean should be close to the true parameter value.

Setting $K = 10$, we observe that all four algorithms register a very similar performance in terms of accuracy, and the true value lies within the interval given by mean $\pm$ 2 std.

Functional space Next we examined the performance of the algorithms in functional space, quantified using R^2, which indicates how good the fit between the data and the signal generated with the posterior median, is. R^2 is defined in terms of RSS and the total sum-of-squares SS_{total}.

We can define R^2, RSS and SS_{total} for each of the 10 data sets, i.e. for the k^{th} data set, $k = 1, \ldots K, \quad K = 10$:

$$\begin{aligned}R_k^2 &= 1 - \frac{\text{RSS}_k}{\text{SS}_k^{\text{total}}},\\ \text{RSS}_k &= \sum_{i=1}^{n}(y_{i,k} - m_i([\boldsymbol{\theta}]_k))^2 = \sum_{i=1}^{n}\epsilon_{i,k}^2,\\ \text{SS}_k^{\text{total}} &= \sum_{i=1}^{n}(y_{i,k} - \bar{\mathbf{y}}_k)^2, \quad \bar{\mathbf{y}}_k = \frac{1}{n}\sum_{i=1}^{n}y_{i,k},\end{aligned} \tag{23}$$

where n is the number of data points in a data set. Thus, R^2 lies within [0,1], and the higher R^2, the better the fit.

Table 2 shows the mean and standard deviation of R^2 over the 10 data sets. We notice that R^2 is very high, with a mean of $\sim$ 0.97 for all methods, suggesting that all algorithms perform equally well in terms of predicting a signal which is very similar to the data.

Parameter UQ We also quantified the uncertainty in the parameter estimates for all methods. In Fig. 3 of the Supplement we illustrate the marginal posterior densities for the parameters A, B, C obtained with 1D kernel density estimation from the posterior samples drawn with the four

emulation methods (a randomly selected data set out of the 10 data sets was used). We also superimpose results obtained using a long-run MCMC algorithm (direct MCMC, the Adaptive Metropolis (AM) algorithm Haario et al. 2001), that draws samples directly from the asymptotically exact posterior distribution. HMC running directly on the original posterior distribution would incur too many forward evaluations (see Sect. 2.3 for a discussion on this), hence in our work we opt for a random-walk algorithm (AM). To obtain the marginal posterior densities, we used the kernel smoothing function estimate for univariate data with the optimal bandwidth for normal densities (Bowman and Azzalini 1997). We note that the marginal posterior densities are comparable, backing up previous findings that all four emulation methods perform very similarly, and their performance is close to that of the long-run MCMC sampler.

6.1.4 Efficiency

The next step in the analysis was to compare the different methods in terms of efficiency, which we quantified using min, median and max ESS (see Eq. (13)) normalised by the total number of MCMC samples N and by the number of forward (model) evaluations.[8,9] Results based on the two efficiency measures are presented in Fig. 4, showing the distribution of these quantities over 10 data sets. When inspecting ESS/N (left panel), we observe great variability between the distributions for the minESS/N, medianESS/N and maxESS/N for the first-order methods (HMC and NUTS), in contrast with the higher-order methods (RMHMC and LDMC). In terms of minESS/N, RMHMC and LDMC are clearly superior to HMC and NUTS, while in terms of medianESS/N all methods are more comparable, and similarly in terms of maxESS/N, with the HMC algorithm having moderate advantage. The same pattern is observed when inspecting ESS normalised by the number of model evaluations (right panel), which is expected, considering the high acceptance rate ($>95\%$ across all methods), i.e. the number of model evaluations is very close to N.

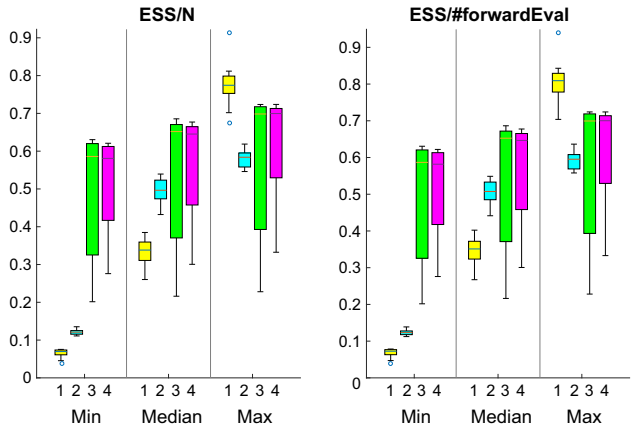

Fig. 4 Efficiency in terms of ESS of the algorithms compared (1: DA-GP-HMC, 2: DA-GP-NUTS, 3: DA-GP-RMHMC, 4: DA-GP-LDMC) for the sinusoidal example. Min, median, max ESS were calculated using Eq. (13). We show the distribution over 10 data sets. Legend: HMC—Hamiltonian Monte Carlo, NUTS: No U-turn sampler, RMHMC—Riemann Manifold Hamiltonian Monte Carlo, LDMC—Lagrangian Dynamical Monte Carlo, DA—delayed acceptance, GP: Gaussian process, ESS: effective sample size, N—number of total MCMC samples, #forwardEval—number of forward simulations from the ODE model. See Table 1 for further explanations of the abbreviations

6.2 FitzHugh–Nagumo

6.2.1 Hybrid algorithms for non-positive definite negative Hessian matrix

For the FitzHugh–Nagumo model we encountered difficulties with running the higher-order methods (RMHMC and LDMC) due to the negative Hessian matrix of the log unnormalised posterior not being positive definite, as illustrated in Fig. 5, where we display the regions in 2D parameter space for which the negative Hessian matrix is not positive definite. As outlined in Sect. 5.3, for this example we replaced the RMHMC or LDMC methods by the emulation hybrid-HMC-RMHMC or hybrid-HMC-LDMC, and for the RMHMC/LDMC component of the method, we used Eq. (19) to set the metric tensor. The emulation hybrid-HMC-LDMC algorithm registered a percentage of 25% (average over the 10 data sets) of HMC-drawn samples, and 75% of LDMC-drawn samples out of the total number of MCMC samples. Also, the emulation hybrid-HMC-RMHMC algorithm registered a percentage of 34% (average over the 10 data sets) of HMC-drawn samples, and 66% of RMHMC-drawn samples out of the total number of MCMC samples. For the FitzHugh–Nagumo model, results in subsequent sections are shown for the hybrid algorithms.

[8] Given that the number of parameters for the sinusoidal example is three, the min, median and max ESS correspond to the ESS for each of the three parameters. For comparability with the other mathematical models, we report the results in terms of the former three measures.

[9] The number of model evaluations is representative of the sampling phase, and excludes the number of samples used for the initial and exploratory phases. The latter samples are the same across methods, rendering their inclusion unnecessary for method comparison.

6.2.2 DA versus noDA

The distribution of the ESS values for the DA and noDA methods is shown in Fig. 2(b) of the Supplement. The hypothesis test investigating the effect of using the DA scheme revealed a p-value > 0.05 (see Eq. (20)) for all efficiency measures, thus, there is no difference in efficiency between the DA and noDA algorithms. For consistency with the sinusoidal example, we took forward the DA algorithms.

6.2.3 Accuracy

Parameter space Table 3 illustrates the mean and standard deviation of the posterior medians over 10 data sets (see Eq. (22)) for the ODE parameters drawn with all emulation methods, and of the noise variances, sampled with Gibbs sampling.

All four emulation algorithms perform very similarly in terms of accuracy, with the true values for the ODE parameters and the two noise variances being contained within the interval given by mean $\pm$ 2 std.

Functional space Table 3 gives the the mean and standard deviation of R^2 (Eq. (23)) over 10 data sets for both 'species' obtained with every emulation algorithm. The methods perform very similarly, and one of the signals appears to be better learnt (mean $R^2 \sim 0.9$) than the other (mean $R^2 \sim 0.7$).

Parameter UQ Regarding UQ, all the methods are on a par, as evident in Fig. 4 in the Supplement, showing the marginal posterior densities for the ODE parameters and noise variances for one randomly selected data set out of the 10 data sets, and their performance is close to that of a long-run MCMC sampler (direct MCMC) drawing samples from the asymptotically exact posterior distribution.

6.2.4 Efficiency

Results based on the min, median and max ESS (see Eq. (13) normalised by the total number of MCMC samples N and by the number of forward (model) evaluations are presented in Fig. 6, which shows the distribution of these quantities over 10 data sets. When inspecting ESS/N (left panel), we again note a much larger variability between the distributions for the minESS/N, medianESS/N and maxESS/N for the first-order methods (HMC and NUTS), unlike the higher-order methods (hybrid-HMC-RMHMC and hybrid-HMC-LDMC). Another observation is that NUTS tends to perform systematically worst. In terms of minESS/N, HMC, hybrid-HMC-RMHMC and hybrid-HMC-LDMC are comparable, while in terms of medianESS/N and maxESS/N, HMC seems to have an advantage. The same observations can be made when inspecting ESS normalised by the number of model evaluations (right panel), which is expected,

Table 3 Accuracy in parameter and functional space for the FitzHugh–Nagumo example For each of the emulation methods compared we show the mean and standard deviation of the parameter posterior medians for 10 data sets, calculated using Eq. (22) (note that the noise variances were sampled using Gibbs sampling). The true parameter values are also shown. R^2 (Eq. (23)), as a measure of fit to the data is also displayed for each of the two 'species'. Legend: HMC—Hamiltonian Monte Carlo, NUTS: No U-turn sampler, RMHMC—Riemann Manifold Hamiltonian Monte Carlo, LDMC—Lagrangian Dynamical Monte Carlo, DA - delayed acceptance, GP: Gaussian process. See Table 1 for further explanations of the abbreviations

Algorithm	α	β	γ	σ_V^2 and σ_R^2	R^2
DA-GP-HMC	0.1840 (0.0282)	0.2507 (0.0991)	2.9821 (0.0465)	0.2587 (0.0355) and 0.1677 (0.0183)	0.8960 (0.0111) and 0.7353 (0.0348)
DA-GP-NUTS	0.1837 (0.0285)	0.2494 (0.1010)	2.9829 (0.0468)	0.2588 (0.0356) and 0.1679 (0.0173)	0.8963 (0.0110) and 0.7354 (0.0348)
DA-GP-hybrid-RMHMC-HMC	0.1844 (0.0288)	0.2610 (0.1006)	2.9834 (0.0466)	0.2592 (0.0351) and 0.1680 (0.0178)	0.8964 (0.0111) and 0.7353 (0.0350)
DA-GP-hybrid-LDMC-HMC	0.1842 (0.0279)	0.2629 (0.1030)	2.9825 (0.0471)	0.2596 (0.0356) and 0.1680 (0.0181)	0.8964 (0.0111) and 0.7353 (0.0349)
True value	0.2	0.2	3	0.25 and 0.16	–

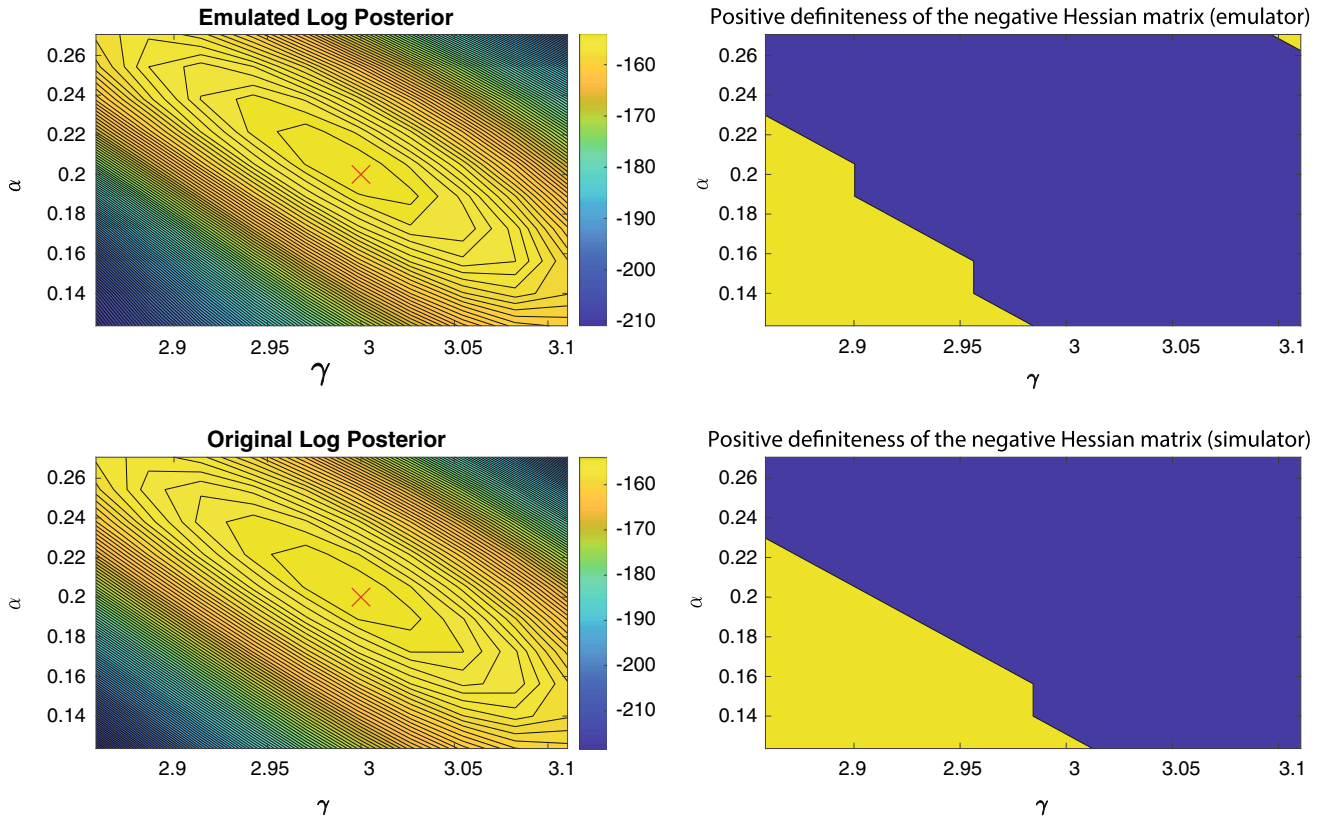

Fig. 5 Displaying the positive definiteness of the negative Hessian matrix for the emulated (top) and original (bottom) log unnormalised posterior distribution of two of the parameters (the third parameter is kept fixed at its true value) for the FitzHugh–Nagumo model. In right column plots, blue stands for positive definite and yellow for non-positive definite matrix. The red cross in the landscape of the log unnormalised posterior distribution marks the true parameter value. (Color figure online)

considering the high acceptance rate (>80% across all methods).

6.3 Biochemical pathway

6.3.1 A zero mean GP versus a quadratic mean GP

For the biochemical pathway example, we implemented both a zero mean GP model, as well as a quadratic mean function GP model for the RSS. The latter potentially places high prior density values near the mode, where a parabola-shaped form of the log unnormalised posterior distribution is expected. The quadratic mean function fits this parabola, thus regions far away from the mode are suppressed. Our findings are that the zero mean model encouraged adding 'extreme' RSS values (high relative to the low RSS region) to the list of training points as a consequence of stepping into a region of high uncertainty of the emulator. Therefore, the zero mean GP emulator is not a faithful representation of the simulator, which leads to poor performance in the sampling phase, as shown in Table 4. Table 4 displays quantitative metrics, showing that the quadratic mean GP leads to better perfor-

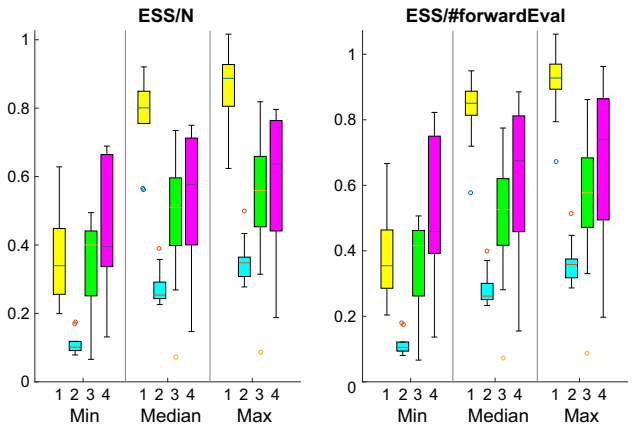

Fig. 6 Efficiency in terms of ESS of the algorithms compared (1: DA-GP-HMC, 2: DA-GP-NUTS, 3: DA-GP-hybrid-HMC-RMHMC, 4: DA-GP-hybrid-HMC-LDMC) for the FitzHugh–Nagumo example. Min, median, max ESS are calculated using Eq. (13). We show the distribution over 10 data sets. Legend: HMC—Hamiltonian Monte Carlo, NUTS: No U-turn sampler, RMHMC—Riemann Manifold Hamiltonian Monte Carlo, LDMC—Lagrangian Dynamical Monte Carlo, DA—delayed acceptance, GP: Gaussian process, ESS: effective sample size, N—number of total MCMC samples, #forwardEval—number of forward simulations from the ODE model

Table 4 Hamiltonian Monte Carlo (HMC) results for two GP models (zero mean vs. quadratic mean GP) of the log unnormalised posterior for the biochemical signalling pathway model in Eq. (9) The acceptance rate and effective sample size (ESS) normalised by the number of HMC iterations N is the median over 10 algorithm initialisations for one data set

GP model	Acceptance rate (%)	ESS/N
Zero-mean GP	57	(0.24, 0.04, 0.02, 0.03, 0.06)
Quadratic mean GP	77	(0.55, 0.48, 0.24, 0.32, 0.42)

mance (in terms of acceptance rate and ESS) compared to the zero mean GP model.

Additionally, in Fig. 8 of the Supplement we show parameter posterior samples collected in the sampling phase for one randomly selected data set out of the 10 data sets. The top two figures show posterior samples generated from the zero mean GP log unnormalised posterior and two different chain initialisations, while the bottom two figures show samples drawn from the quadratic mean GP log unnormalised posterior. The chain mixing based on the latter GP model is better than the former, although for both models, periods of chain stagnations are registered, an issue which we discuss in Sect. 7.6. Consequently, in the next sections, results produced with a quadratic mean function GP prior model are presented.

6.3.2 Hybrid algorithms for non-positive definite negative Hessian matrix

Applying the higher-order methods (RMHMC and LDMC) posed difficulties due to the negative Hessian matrix of the log unnormalised posterior not always being positive definite, as illustrated for a 2D parameter space in Fig. 7 of the Supplement. Similarly to the FitzHugh–Nagumo model, we implemented the hybrid algorithms as described in Sect. 5.3. Our findings revealed the following: for some data sets, the hybrid algorithms returned samples drawn mostly with the HMC algorithm, i.e. the percentage of HMC-drawn samples was 85–90%, while 10–15% of the samples were drawn with RMHMC or LDMC, the reason being a high condition number ($> 10^{15}$) of the mass matrix set using Eq. (19). For other data sets, the Bayesian optimisation employed for RMHMC or LDMC set the tuning parameters to very low values (higher values would encourage the sampler to step into regions where the matrix has high condition number), leading to the sampler making tiny, ineffective moves, resulting in low ESS. These issues illustrate that the second-order methods (RMHMC, LDMC) combined with emulation of the log unnormalised posterior distribution effectively reduce to the standard first-order HMC with emulation method due to the sub-optimality of the mass matrix. However, it is important to emphasize that our proposed hybrid algorithm provides a safety net that enables the overall algorithm to finish successfully by dynamically switching between first- and second-order methods.

In light of these findings, the second-order methods were excluded from the comparison, and subsequent results for the biochemical pathway example are presented for HMC and NUTS only.

6.3.3 DA versus noDA

The hypothesis testing comparing the mean of the distribution over 10 data sets of the normalised minimum, median and maximum ESS (Eq. (13)) between the DA and noDA scheme revealed a p-value > 0.05 for all measures, except MaxESS/number of forward evaluations, for which DA seems to have a slight advantage. The distribution of these ESS values for the DA and noDA methods is shown in Fig. 2(c) of the Supplement. For the purpose of comparing the Hamiltonian Monte Carlo algorithms and to ensure consistency with previous examples, we proceeded with the DA algorithms.

6.3.4 Accuracy

Parameter space Table 5 illustrates the mean and standard deviation of the posterior medians over 10 data sets (see Eq. (22)) for the ODE parameters drawn with all emulation methods, and of the noise variances, sampled with Gibbs sampling.

The methods perform very similarly in terms of accuracy, with the inferred values for the ODE parameters and the 'species' noise variances being contained within the interval given by mean $\pm$ 2 std.

Functional space Table 5 shows the mean and standard deviation of R^2 (Eq. (23)) over 10 data sets of R^2 for every 'species' obtained with every emulation method. A similar performance of the methods is registered, with all methods giving a very large R^2 (mean of ~ 0.99).

Table 5 Accuracy in parameter and functional space for the biochemical pathway example

Algorithm	k_1	V_1	K_{m_1}	V_2	K_{m_2}	σ_S^2 and σ_D^2 and σ_R^2 and $\sigma_{R_{pp}}^2$	R^2
DA-GP-HMC	0.0506 (0.0008)	0.2021 (0.0142)	0.1166 (0.0516)	0.0996 (0.0049)	0.1142 (0.0233)	0.0005 (0.0001) and 0.0005 (0.0001) and 0.0006 (0.0001) and 0.0005 (0.0001)	0.9961 (0.0012) and 0.9963 (0.0008) and 0.9915 (0.0027) and 0.9931 (0.0028)
DA-GP-NUTS	0.0506 (0.0008)	0.2021 (0.0146)	0.1167 (0.0518)	0.0997 (0.0052)	0.1150 (0.0268)	0.0005 (0.0001) and 0.0005 (0.0001) and 0.0006 (0.0001) and 0.0005 (0.0001)	0.9961 (0.0012) and 0.9963 (0.0008) and 0.9914 (0.0029) and 0.9930 (0.0028)
True value	0.05	0.2	0.1	0.1	0.1	0.0004 and 0.0004 and 0.0004 and 0.0004	–

For each of the emulation methods compared we show the mean and standard deviation of the parameter posterior medians for 10 data sets, calculated using Eq. (22) (note that the noise variances were sampled using Gibbs sampling). The true parameter values are also shown. R^2 (Eq. (23)), as a measure of fit to the data is also displayed for each of the four 'species'. Legend: HMC—Hamiltonian Monte Carlo, NUTS: No U-turn sampler, RMHMC—Riemann Manifold Hamiltonian Monte Carlo, LDMC—Lagrangian Dynamical Monte Carlo, DA—delayed acceptance, GP: Gaussian process. See Table 1 for further explanations of the abbreviations

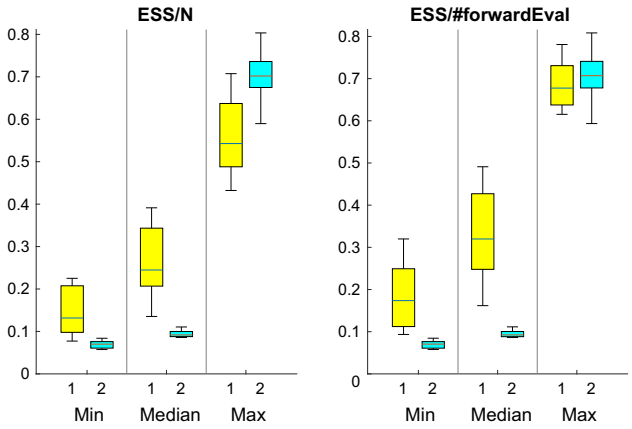

Fig. 7 Efficiency in terms of ESS of the algorithms compared (1: DA-GP-HMC, 2: DA-GP-NUTS) for the biochemical pathway example. Min, median, max ESS were calculated using Eq. (13). We show the distribution over 10 data sets. Legend: HMC—Hamiltonian Monte Carlo, NUTS: No U-turn sampler, DA—delayed acceptance, GP: Gaussian process, ESS: effective sample size, N—number of total MCMC samples, #forwardEval—number of forward simulations from the ODE model. See Table 1 for further explanations of the abbreviations

Parameter UQ The two methods give overlapping marginal posterior densities for the parameters, see Fig. 5 in the Supplement.

6.3.5 Efficiency

Method comparison Figure 7 shows the distribution of min, median and max ESS (see Eq. (13)) normalised by the total number of MCMC samples N and by the number of forward (model) evaluations over 10 data sets for HMC and NUTS. In terms of normalised MinESS and MedianESS, the HMC algorithm has a superior performance to NUTS, while a fair degree of similarity in the performance with respect to normalised MaxESS between the two algorithms is recorded.
Chain stagnation The simulation results in Figure 8 of the Supplement show fairly long periods of rejections, after which the sampler recovers, with good mixing exhibited. This issue points to a mismatch between the emulator and the simulator; a more thorough discussion on this is given in Sect. 7.6.

6.4 Real-world application: 1D fluid-dynamics model of the pulmonary blood circulation

6.4.1 DA versus noDA

For the particular data set available, we test for the equality of mean normalised ESS of the distribution *over all four parameters* between the DA and noDA schemes for all four emulation algorithms. The hypothesis test reveals a p-value > 0.05 for all algorithms. These findings suggest no difference

in efficiency between the DA and noDA schemes. Similarly to the previous examples, we took forward the DA algorithms.

6.4.2 Accuracy

Table 6 shows the posterior median and 95% credible interval of the PDE parameters obtained from the posterior samples generated with all the emulation methods (DA-GP-HMC, DA-GP-NUTS, DA-GP-RMHMC, DA-GP-LDMC, see Table 1 for abbreviation explanations) and with a long-run MCMC sampler, and of the noise variance, sampled with Gibbs sampling. These results are for one single measured (real) data set, for which the ground truth parameter values are unknown. In addition, in Fig. 6 of the Supplement we plot the marginal posterior densities of the parameters. In the absence of a gold standard, to test whether the emulation approach gives any bias in the results, we also present results obtained with a long-run MCMC sampler.

Table 6 and Fig. 6 of the Supplement suggest that all methods provide very similar results. The overlapping densities for the different algorithms indicate that the methods provide samples from approximately the same distribution. In the absence of a proper gold standard, the agreement between the predicted posterior probability distributions across all emulation algorithms and the long-run (direct) MCMC, can be taken as a proxy for accuracy. This statement is backed up by a very high R^2 value of 0.99 (see Table 6) registered by all methods, indicating a very good fit to the measured data.

6.4.3 Efficiency

Figure 8 displays the distribution of ESS (Eq. (12)) normalised by the total number of MCMC samples N and by the number of forward (model) evaluations *over all four parameters* for the single data set analysed. RMHMC appears superior to all other algorithms when analysing ESS/N (left panel) or ESS/#forwardEval (right panel). Additionally, NUTS systematically performs more poorly than the other algorithms. LDMC is clearly superior to HMC when looking at the minimum or median ESS/N or ESS/#forwardEval, and HMC is better when looking at the maximum ESS/N or ESS/#forwardEval. The distribution of these quantities is much more variable for HMC and NUTS than for RMHMC and LDMC.

7 Discussion

We have contributed to the research field of accelerating Hamiltonian/Lagrangian Monte Carlo algorithms by coupling them with Gaussian processes for emulation of the log unnormalised posterior distribution. We have provided proofs of detailed balance with respect to the asymp-

Table 6 Accuracy in parameter and functional space for the statistical inference performed on the fluid-dynamics pulmonary application We show the parameter posterior medians and 95% credible interval obtained with all emulation methods (the noise variance was sampled using Gibbs sampling), as well as R^2, calculated using Eq. (23). Results obtained with the Adaptive Metropolis (Haario et al. 2001) algorithm drawing samples approximately from the exact posterior distribution (direct MCMC) are also shown. Legend: HMC—Hamiltonian Monte Carlo, NUTS: No U-turn sampler, RMHMC—Riemann Manifold Hamiltonian Monte Carlo, LDMC—Lagrangian Dynamical Monte Carlo, DA—delayed acceptance, GP: Gaussian process. See Table 1 for further explanations of the abbreviations

Algorithm	s	r_1	r_2	c	σ^2	R^2
DA-GP-HMC	98,043 (97,004, 99,159)	1.7371 (1.6997, 1.7765)	0.1892 (0.1740, 0.2035)	− 1.4339 (− 1.5495, − 1.3264)	0.1256 (0.1119, 0.1420)	0.99
DA-GP-NUTS	98,021 (96,998, 99,136)	1.7364 (1.6996, 1.7748)	0.1891 (0.1743, 0.2026)	− 1.4322 (− 1.5409, − 1.3259)	0.1256 (0.1116, 0.1425)	0.99
DA-GP-RMHMC	98,038 (97,040, 99,052)	1.7374 (1.7023, 1.7716)	0.1889 (0.1728, 0.2035)	− 1.4340 (− 1.5505, − 1.3260)	0.1255 (0.1109, 0.1432)	0.99
DA-GP-LDMC	98,024 (96,967, 99,132)	1.7366 (1.7006, 1.7731)	0.1893 (0.1734, 0.2037)	− 1.4294 (− 1.5591, − 1.3280)	0.1258 (0.1122, 0.1429)	0.99
Direct MCMC	98,025 (96,997, 99,068)	1.7359 (1.7003, 1.7743)	0.1891 (0.1743, 0.2038)	− 1.4358 (− 1.5493, − 1.3222)	0.1257 (0.1114, 0.1425)	0.99

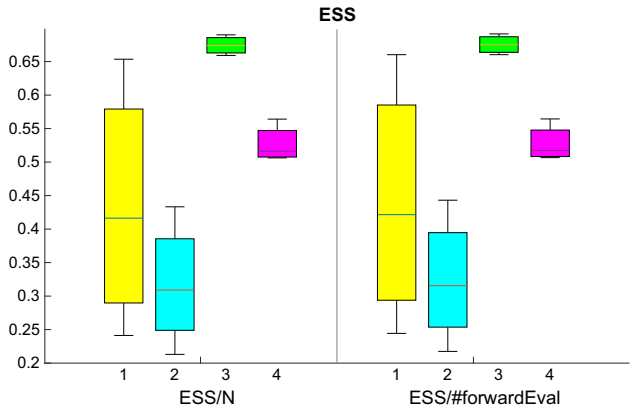

Fig. 8 Efficiency in terms of ESS of the algorithms compared (1: DA-GP-HMC, 2: DA-GP-NUTS, 3: DA-GP-RMHMC, 4: DA-GP-LDMC) for the fluid-dynamics pulmonary example. ESS was calculated using Eq. (12). We show the distribution of ESS over the four parameters for one single data set. Legend: HMC—Hamiltonian Monte Carlo, NUTS: No U-turn sampler, RMHMC—Riemann Manifold Hamiltonian Monte Carlo, LDMC—Lagrangian Dynamical Monte Carlo, DA—delayed acceptance, GP: Gaussian process, ESS: effective sample size, N—number of total MCMC samples, #forwardEval—number of forward simulations from the PDE model. See Table 1 for further explanations of the abbreviations

totically exact posterior distribution for these algorithms, and validated the mathematical and coding correctness of the samplers' implementation by Geweke consistency tests (Fig. 1 in the Supplement). Moreover, we have carried out a comparative evaluation study to assess the performance of the methods on a series of ODE/PDE models (sinusoidal, FitzHugh–Nagumo, biochemical pathway and fluid-dynamics pulmonary model). We have aimed to identify the most computationally efficient and accurate parameter inference and UQ tools to be applied to nonlinear ODE or PDE models. Typically, these models incur onerous computational costs caused by repeated numerical integrations as part of an iterative sampling scheme. In addition, we have investigated whether the delayed acceptance scheme used in conjunction with these emulation algorithms can further offer computational gains over the standard algorithms.

7.1 A discussion on the algorithms compared

We have compared the following algorithms: noDA-GP-HMC (i.e. standard GP-HMC), DA-GP-HMC, DA-GP-NUTS, noDA-GP-RMHMC, DA-GP-RMHMC, noDA-GP-LDMC and DA-GP-LDMC (see Table 1 for explanations of the abbreviations). While the standard GP-HMC was originally proposed in Rasmussen (2003), all the other algorithms are our contribution. The noDA-GP-NUTS algorithm was not implemented in practice due to requiring a number of expensive model evaluations roughly one order of magnitude higher than DA-GP-NUTS, see the proof in Sect. 4.7 of the Supplement.

7.1.1 Hybrid algorithms for non-positive definite negative Hessian matrix

As discussed in Sect. 5.3, due to emulating the log unnormalised posterior instead of the signals (i.e. the solutions of the ODEs/PDEs), we could not use the expected Fisher information matrix when setting the mass matrix in RMHMC or LDMC. Instead we used the observed Fisher information matrix. The resulting negative Hessian matrix of the log unnormalised posterior distribution (which is the sum of the observed Fisher information matrix and the matrix of negative second-order derivatives of the log prior) is not guaranteed to be positive definite. This was an issue for the FitzHugh–Nagumo and biochemical pathway models (see Fig. 5 in the main paper and Fig. 7 in the Supplement), but not for the sinusoidal and pulmonary models. For the former models, we adopted a form for the mass matrix based on a combination of the observed and empirical Fisher information matrix (Eq. (19)), ensuring at least positive semi-definiteness. The downside was that the matrix can have a high condition number ($> 10^{15}$). To overcome this, we took a hybrid approach: if at any point throughout the trajectory the matrix was numerically unstable, the simulation within the trajectory was stopped prematurely, and HMC was run instead of RMHMC/LDMC for that particular iteration. The FitzHugh–Nagumo model in particular benefited from this hybrid approach, as efficiency gain over the HMC algorithm was achieved, with roughly two thirds of samples being RMHMC-drawn and three quarters of samples being LDMC-drawn (while the rest of one third and one quarter were HMC-drawn samples). In contrast, the biochemical pathway model registered no efficiency gain over HMC, as most samples were drawn with the latter.

7.2 Emulation of the model output

To avoid the loss of information inherent in emulating the log unnormalised posterior distribution (Davies et al. 2019) (see Sect. 5.3 for specific equations and details), the multivariate signal could be emulated instead, using e.g. ensembles of single-output emulators (MS) (Conti and O'Hagan 2010b), or multivariate-output Gaussian processes (MO) (Álvarez et al. 2010; Moreno-Muñoz et al. 2018). However, emulating a high-dimensional output brings computational challenges. The computational costs for emulating a high-dimensional complex output will increase significantly when compared to emulating a one-dimensional function (Wilkinson 2014; Davies et al. 2019). General methods based on marginalisation over covariance between outputs with an uninformative prior, as done in Conti et al. (2009) and Conti and O'Hagan

(2010a) does not take into account relevant information about the data (e.g. periodicity of the time series). Additionally, allowing for the particular dependence of a time series (e.g. correlations, periodicity) requires a thorough exploration to identify an appropriate emulation strategy, e.g. using ensembles of independent GPs or multivariate-output GPs, sums or products of different kernel functions, or different forms for the mean function.

In light of the findings of this study, a new research direction could be that of finding a trade-off between high computational complexity (due to emulating a multivariate output) and potential efficiency gains of a second-order Hamiltonian/Lagrangian scheme guaranteeing a positive definite Fisher information matrix.

7.3 Advantage of delayed acceptance

There is no evidence that the DA scheme brings any computational gains when coupled with Hamiltonian/Lagrangian Monte Carlo algorithms. The efficiency of DA-GP-HMC, noDA-GP-HMC, DA-GP-RMHMC, noDA-GP-RMHMC, DA-GP-LDMC, noDA-GP-LDMC (see Table 1 for abbreviation explanations), as measured in terms of ESS normalised by the total number of MCMC samples and by the number of model (forward) evaluations, is comparable between the DA and noDA algorithms, a conclusion drawn based on a formal hypothesis test testing for equal sample means of the normalised ESS.

While an MCMC with DA approach has been taken in previous studies in the literature (Christen and Fox 2005; Golightly et al. 2015; Sherlock et al. 2017; Higdon et al. 2011; Cui et al. 2011; Quiroz et al. 2018; Banterle et al. 2019), to our best knowledge, our current study is the only one to complement our previous study (Paun and Husmeier 2020), and use DA in conjunction with Hamiltonian/Lagrangian Monte Carlo algorithms. Other studies have compared standard random-walk MCMC algorithms to their DA version. For example, Golightly et al. (2015) showed that the DA scheme can lead to improvements in computational efficiency in a particle MCMC algorithm applied to stochastic kinetic models. Additionally, Banterle et al. (2019) and Quiroz et al. (2018) showed that DA brings computational advantages when applied to M–H algorithms on large data sets, for which data sub-sampling is employed. However, these MCMC algorithms are based on a random-walk, which is known to have a lower acceptance rate (and efficiency) than the gradient-driven Hamiltonian Monte Carlo algorithms (see Ch. 5 in Brooks et al. 2011 or Sengupta et al. 2016). Therefore, the former algorithms provide more potential for improvement with the DA scheme than the latter, i.e. if a rejection is more likely, a higher number of computationally expensive model evaluations are avoided by the first stage employing the emulator.[10]

7.4 Accuracy

The accuracy in parameter and functional space proved to be very similar between the different methods for all ODE/PDE models considered, see Tables 2, 3 and 5 and 6. In addition, for the toy examples, we showed that the algorithms were able to learn the true parameter values that generated the data (Tables 2, 3 and 5). The marginal posterior densities constructed from the MCMC posterior samples showed overlapping densities, indicating that the uncertainty quantification was on a par for all methods (Figs. 3, 4, 5 and 6 in the Supplement).

7.5 Efficiency

ESS normalised by the total number of MCMC samples In terms of ESS/N (left panel in Figs. 4, 6, 7, 8), the performance of DA-GP-NUTS was generally inferior to that of the other algorithms (DA-GP-HMC, DA-GP-RMHMC, DA-GP-LDMC, see Table 1 for abbreviation explanations). A possible explanation is that for DA-GP-NUTS the tuning of the step size and number of steps is performed in the emulated log unnormalised posterior entirely, based on samples accepted at the emulator stage, due to the construction of the algorithm (see the proof in Sect. 4.7 of the Supplement). In contrast, for the other three algorithms, the tuning is performed based on samples accepted at the simulator stage, thus the simulator plays a role in the choice of optimum tuning parameters, positively impacting efficiency.

In terms of minESS/N, generally the RMHMC and LDMC algorithms perform better than HMC, while in terms of medianESS/N or maxESS/N no clear pattern is observed (sometimes RMHMC and LDMC are better, other times HMC is preferred). We also note a much larger discrepancy between minESS/N and maxESS/N for HMC and NUTS than RMHMC and LDMC, for which ESS/N varies much less across parameters. This is a consequence of the latter two algorithms using a mass matrix set via the curvature of the log unnormalised posterior, while HMC and NUTS use an identity matrix as the mass matrix, and the optimum step size is restricted by the lowest marginal variance. Thus, a first-order algorithm like HMC or NUTS can be more inefficient (e.g. in terms of minESS/N) for problems with large discrepancies between the lowest and largest marginal variance. Generally, RMHMC and LDMC perform similarly, an exception is the pulmonary

[10] Provided the emulator is an accurate representation of the simulator.

model, which registers better performance for RMHMC than LDMC.

ESS normalised by the number of forward evaluations In terms of ESS/#forwardEval (right panel in Figs. 4, 6, 7, 8), a similar pattern as for ESS/N is observed, which is expected given the high acceptance rate (>80%), meaning that the number of model evaluations is close to the total number of MCMC samples. This finding also helps explain why we found no advantage of the DA scheme: generally most proposals are accepted at the emulator (first) stage, and are subsequently subject to the accept/reject decision at the simulator (second) stage.

The above interpretations for the FitzHugh–Nagumo model apply to the emulation hybrid-RMHMC-HMC/hybrid-LDMC-HMC instead of the standard emulation RMHMC/LDMC.

7.6 Limitations and future improvements for the biochemical pathway example

The biochemical pathway example was a hard problem to emulate due to the high correlations manifested through long ridges in the log unnormalised posterior landscape (see Fig. 7 in the Supplement). While a quadratic mean GP prior improved the acceptance rate, mixing and efficiency (see Table 4 of the main paper and Fig. 8 of the Supplement), a larger number of training points would have been needed for an optimal coverage of the parameter space. However, this would have resulted in high CPU times[11] due to different operations performed repeatedly (to compute the GP predictive mean and up to its third-order derivatives) involving the high-dimensional covariance matrix.

The consequence was a mismatch between the emulator and the simulator in the tails of the target distribution: proposals were accepted at the emulator stage, but rejected at the simulator stage. The result was occasional chain stagnations, i.e. 'stickiness', see Fig. 8 of the Supplement. The use of a larger number of training points in conjunction with sparse GPs (Titsias 2009; Hensman et al. 2015), which optimally select a lower number of points retaining the maximum information at reduced costs could overcome the issues presented and constitutes future work. This strategy could potentially be coupled with continuous refinement of the emulator when the sampler steps into a region of high uncertainty, similar to the study in Conrad et al. (2016), to avoid deciding when to stop the exploratory phase, during which the emulator is refined. It is worth mentioning that the 'stickiness' problem is a notorious issue in pseudo-marginal MCMC (Drovandi et al. 2018; Murray and Graham 2016), in which the estimator of the target distribution is inconsistent with the true target distribution in the tails. This is a similar issue in nature to that encountered with emulation MCMC, thus, an interesting future direction would be an interdisciplinary cross-breeding between emulation MCMC and pseudo-marginal MCMC.

[11] Care has to be taken to ensure that the computational times in approaches employing the emulator remain lower than when the simulator solely is used, else the entire purpose of emulation is defied.

8 Conclusions

We have provided theoretical and empirical investigations into Hamiltonian/Lagrangian Monte Carlo algorithms coupled with Gaussian processes for emulation of the log unnormalised posterior distribution. We have proved that these emulation algorithms satisfy detailed balance with respect to the exact posterior distribution. Additionally, we have investigated whether the delayed acceptance scheme is computationally advantageous over the standard algorithms. We have carried out an empirical efficiency assessment of the emulation methods on a series of ODE/PDE models, including toy problems and a real-world application of computational fluid-dynamics of the pulmonary blood circulation model. We have aimed to identify the most computationally efficient and accurate parameter inference and UQ tools to be applied to nonlinear ODE or PDE models, which typically incur onerous computational costs due to the need for repeated numerical integrations. Results showed no advantage of the delayed acceptance scheme over the standard algorithms with respect to efficiency measures based on the effective sample size. Additionally, our methods estimated the true parameter values well, with all methods performing similarly across the ODE/PDE models considered. The Lagrangian Dynamical Monte Carlo and Riemann Manifold Hamiltonian Monte Carlo tended to register the highest efficiency (in terms of effective sample size normalised by the number of forward model evaluations), followed by the Hamiltonian Monte Carlo, and the No U-turn sampler tended to be the least efficient.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s11222-021-10060-4.

Acknowledgements This work is part of the research programme of the Centre for Multiscale Soft Tissue Mechanics with Application to Heart and Cancer (SofTMech), funded by the Engineering and Physical Sciences Research Council (EPSRC) of the UK, grant reference number EP/N014642/1. This work is also part of the research programme of the SofTMech Statistical Emulation and Translation Hub, funded by EPSRC, Grant reference number EP/T017899/1. We thank Mette Olufsen and Mitchel Colebank for providing the code for the fluid-dynamics pulmonary model. Last, but not least, I would like to thank my daughter, Sofia, for waiting just enough time to allow me to submit this paper before coming to this world.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

Álvarez, M., Luengo, D., Titsias, M., Lawrence, N.D.: Efficient multioutput Gaussian processes through variational inducing kernels. In: Teh, Y.W., Titterington, M. (eds) Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, PMLR, Chia Laguna Resort, Sardinia, Italy, Proceedings of Machine Learning Research, vol. 9, pp. 25–32 (2010)

Banterle, M., Grazian, C., Lee, A., Robert, C.P.: Accelerating Metropolis–Hastings algorithms by delayed acceptance. Found. Data Sci. **1**(2), 103–128 (2019)

Bastos, L.S., O'Hagan, A.: Diagnostics for Gaussian process emulators. Technometrics **51**(4), 425–438 (2009)

Betancourt M (2015) The fundamental incompatibility of scalable Hamiltonian Monte Carlo and Naive data subsampling. In: Bach F, Blei D (eds) Proceedings of the 32nd International Conference on Machine Learning, PMLR, Lille, France, Proceedings of Machine Learning Research, vol. 37, pp 533–540

Bliznyuk, N., Ruppert, D., Shoemaker, C., Regis, R., Wild, S., Mugunthan, P.: Bayesian calibration and uncertainty analysis for computationally expensive models using optimization and radial basis function approximation. J. Comput. Graph. Stat. **17**, 270–294 (2008)

Bowman, A.W., Azzalini, A.: Applied Smoothing Techniques for Data Analysis. Oxford University Press Inc, New York (1997)

Bratley, P., Fox, B.: Algorithm 659: implementing Sobol's quasirandom sequence generator. ACM Trans. Math. Softw. **14**(1), 88–100 (1988)

Brooks, S., Gelman, A.: General methods for monitoring convergence of iterative simulations. J. Comput. Graph. Stat. **7**(4), 434–455 (1998)

Brooks, S., Gelman, A., Jones, G.L., Meng, X.L. (eds.): Handbook of Markov Chain Monte Carlo. Handbooks of Modern Statistical Methods, Chapman and Hall (2011)

Broyden, C.G.: Quasi-Newton Methods. Academic Press, London (1972)

Brüggemeier, B., Schusterreiter, C., Pavlou, H., Jenkins, N., Lynch, S., Bianchi, A., Cai, X.: Improving the utility of drosophila melanogaster for neurodegenerative disease research by modelling courtship behaviour patterns. In: Report Summarising the Outcomes from the UK NC3R's and POEM's Meeting (2014)

Bui-Thanh, T., Girolami, M.: Solving large-scale PDE-constrained Bayesian inverse problems with Riemann Manifold Hamiltonian Monte Carlo. Inverse Prob. **30**(11), 114014 (2014)

Calderhead, B.: Differential Geometric MCMC Methods and Applications. PhD Thesis, University of Glasgow (2012)

Campbell, D., Steele, R.J.: Smooth functional tempering for nonlinear differential equation models. Stat. Comput. **22**, 429–443 (2012)

Chen, T., Fox, E., Guestrin, C.: Stochastic Gradient Hamiltonian Monte Carlo. In: 31st International Conference on Machine Learning, ICML vol. 5 (2014)

Christen, J., Fox, C.: Markov Chain Monte Carlo using an approximation. J. Comput. Graph. Stat. **14**(4), 795–810 (2005)

Conrad, P.R., Marzouk, Y.M., Pillai, N.S., Smith, A.: Accelerating asymptotically exact MCMC for computationally intensive models via local approximations. J. Am. Stat. Assoc. **111**(516), 1591–1607 (2016)

Conrad, P.R., Davis, A.D., Marzouk, Y.M., Pillai, N.S., Smith, A.: Parallel local approximation MCMC for expensive models. SIAM/ASA J. Uncertain. Quantif. **6**(1), 339–373 (2018)

Conti, S., O'Hagan, A.: Bayesian emulation of complex multi-output and dynamic computer models. J. Stat. Plan. Inference **140**(3), 640–651 (2010)

Conti, S., O'Hagan, A.: Bayesian emulation of complex multi-output and dynamic computer models. J. Stat. Plan. Inference **140**(3), 640–651 (2010)

Conti, S., Gosling, J.P., Oakley, J.E., O'Hagan, A.: Gaussian process emulation of dynamic computer codes. Biometrika **96**(3), 663–676 (2009)

Costabal, F.S., Matsuno, K., Yao, J., Perdikaris, P., Kuhl, E.: Machine learning in drug development: characterizing the effect of 30 drugs on the QT interval using Gaussian process regression, sensitivity analysis, and uncertainty quantification. Comput. Methods Appl. Mech. Eng. **348**, 313–333 (2019)

Cui, T., Fox, C., O'Sullivan, M.: Bayesian calibration of a large-scale geothermal reservoir model by a new adaptive delayed acceptance Metropolis Hastings algorithm. Water Resour. Res. **47**(10), W10521 (2011)

Davies, V., Noé, U., Lazarus, A., Gao, H., Macdonald, B., Berry, C., Luo, X., Husmeier, D.: Fast parameter inference in a biomechanical model of the left ventricle by using statistical emulation. J. R. Stat. Soc. Ser. C: Appl. Stat. **68**(5), 1555–1576 (2019)

Dietzel, A., Reichert, P.: Bayesian inference of a lake water quality model by emulating its posterior density. Water Resour. Res. **50**(10), 7626–7647 (2014)

Drovandi, C.C., Moores, M.T., Boys, R.J.: Accelerating pseudo-marginal MCMC using Gaussian processes. Comput. Stat. Data Anal. **118**, 1–17 (2018)

Fielding, M., Nott, D., Liong, S.Y.: Efficient MCMC schemes for computationally expensive posterior distributions. Technometrics **53**, 16–28 (2011)

Fitzhugh, R.: Impulses and physiological states in theoretical models of nerve membrane. Biophys. J . **1**(6), 445–466 (1961)

Geweke, J.: Getting it right: joint distribution tests of posterior simulators. J. Am. Stat. Assoc. **99**(467), 799–804 (2004)

Girolami, M., Calderhead, B.: Riemann Manifold Langevin and Hamiltonian Monte Carlo methods. J. R. Stat. Soc. Ser. B (Stat. Methodol.) **73**(2), 123–214 (2011)

Golightly, A., Henderson, D.A., Sherlock, C.: Delayed acceptance particle MCMC for exact inference in stochastic kinetic models. Stat. Comput. **25**(5), 1039–1055 (2015)

Gong, W., Duan, Q.: An adaptive surrogate modeling-based sampling strategy for parameter optimization and distribution estimation (ASMO-PODE). Environ. Modell. Softw. **95**, 61–75 (2017)

Göktepe, S., Kuhl, E.: Computational modeling of cardiac electrophysiology: a novel finite element approach. Int. J. Numer. Methods Eng. **79**(2), 156–178 (2009)

Haario, H., Saksman, E., Tamminen, J.: An adaptive Metropolis algorithm. Bernoulli **7**(2), 223–242 (2001)

Hensman, J., Matthews, A.G., Filippone, M., Ghahramani, Z. MCMC for variationally sparse Gaussian processes. In: Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., Garnett, R. (eds) Advances in Neural Information Processing Systems. Curran Associates Inc., vol. 28 (2015)

Higdon, D., Reese, C., Moulton, J., Vrugt, J., Fox, C.: Posterior exploration for computationally intensive forward models, pp. 401–418 (2011)

Hoffman, M., Gelman, A.: The no-U-turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. J. Mach. Learn. Res. **15**(1), 1593–1623 (2014)

Jones, D., Schonlau, M., Welch, W.: Efficient global optimization of expensive black-box functions. J. Glob. Optim. **13**, 455–492 (1998)

Kass, R., Carlin, B., Gelman, A., Neal, R.: Markov Chain Monte Carlo in practice: a roundtable discussion. Am. Stat. **52**(2), 93–100 (1998)

Kennedy, M.C., O'Hagan, A.: Bayesian calibration of computer models. J. R. Stat. Soc. Ser. B (Stat. Methodol.) **63**(3), 425–464 (2001)

Kim, E.K., Choi, E.J.: Pathological roles of MAPK signaling pathways in human diseases. Biochim. Biophysi. Acta (BBA) Mol. Basis Disease **1802**(4), 396–405 (2010)

Kramer, A., Calderhead, B., Radde, N.: Hamiltonian Monte Carlo methods for efficient parameter estimation in steady state dynamical systems. BMC Bioinform. **15**(1), 253 (2014)

Laine, M.: MCMC Toolbox for Matlab. http://helios.fmi.fi/~lainema/dram/ (2007)

Lan, S., Stathopoulos, V., Shahbaba, B., Girolami, M.: Markov Chain Monte Carlo from Lagrangian dynamics. J. Comput. Graph. Stat. **24**(2), 357–378 (2015)

Lan, S., Bui-Thanh, T., Christie, M., Girolami, M.: Emulation of higher-order tensors in manifold Monte Carlo methods for Bayesian inverse problems. J. Comput. Phys. **308**, 81–101 (2016)

Lawrence, N.D., Girolami, M., Rattray, M., Sanguinetti, G.: Learning and Inference in Computational Systems Biology. MIT Press, Cambridge (2010)

Lax, P., Wendroff, B.: Systems of conservation laws. Commun. Pure Appl. Math. **13**(2), 217–237 (1960)

Livingstone, S., Betancourt, M., Byrne, S., Girolami, M.: On the geometric ergodicity of Hamiltonian Monte Carlo. Bernoulli **25**(4A), 3109–3138 (2019)

Lê, M., Delingette, H., Kalpathy-Cramer, J., Gerstner, E.R., Batchelor, T., Unkelbach, J., Ayache, N.: MRI based Bayesian personalization of a tumor growth model. IEEE Trans. Med. Imaging **35**(10), 2329–2339 (2016)

Martin, G.: Cell signaling and cancer. Cancer Cell **4**(3), 167–174 (2004)

McKay, M.D., Beckman, R.J., Conover, W.J.: Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics **21**(2), 239–245 (1979)

Mirams, G.R., Pathmanathan, P., Gray, R.A., Challenor, P., Clayton, R.H.: Uncertainty and variability in computational and mathematical models of cardiac physiology. J. Physiol. **594**(23), 6833–6847 (2016)

Mockus, J., Tiesis, V., Zilinskas, A.: The application of Bayesian methods for seeking the extremum. Towards Glob. Optim. **2**, 117–129 (1978)

Moreno-Muñoz, P., Artés, A., Álvarez, M.: Heterogeneous multi-output Gaussian process prediction. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (eds) Advances in Neural Information Processing Systems 31, pp. 6711–6720. Curran Associates, Inc. (2018)

Murphy, K.P.: Machine Learning: A Probabilistic Approach. MIT Press, Cambridge (2012)

Murray, I., Graham, M.: Pseudo-marginal slice sampling. In: Gretton, A., Robert, C.C. (eds) Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, Cadiz, Spain, JMLR: W&CP, vol. 51, pp. 911–919 (2016)

Nagumo, J., Arimoto, S., Yoshizawa, S.: An active pulse transmission line simulating nerve axon. Proc. IRE **50**(10), 2061–2070 (1962)

Neal, R.: MCMC using Hamiltonian dynamics. Handb. Markov Chain Monte Carlo **2**, 113–162 (2011)

Olufsen, M., Peskin, C.S., Kim, W., Pedersen, E.M., Nadim, A., Larsen, J.: Numerical simulation and experimental validation of blood flow in arteries with structured-tree outflow conditions. Ann. Biomed. Eng. **28**, 1281–1299 (2000)

Paun, L.M., Husmeier, D.: Markov chain Monte Carlo with Gaussian processes for fast parameter estimation and uncertainty quantification in a 1D fluid-dynamics model of the pulmonary circulation. Int. J. Numer. Methods Biomed. Eng. p. e3421 (2020)

Paun, L.M., Colebank, M., Umar Qureshi, M., Olufsen, M., Hill, N., Husmeier, D.: MCMC with Delayed Acceptance using a Surrogate Model with an Application to Cardiovascular Fluid Dynamics. In: Proceedings of the International Conference on Statistics: Theory and Applications (ICSTA'19) (2019)

Peirlinck, M., Sahli Costabal, F., Sack, K.L., Choy, J.S., Kassab, G.S., Guccione, J.M., De Beule, M., Segers, P., Kuhl, E.: Using machine learning to characterize heart failure across the scales. Biomech. Model. Mechanobiol. **18**(6), 1987–2001 (2019)

Quiroz, M., Tran, M.N., Villani, M., Kohn, R.: Speeding up MCMC by delayed acceptance and data subsampling. J. Comput. Graph. Stat. **27**(1), 12–22 (2018)

Qureshi, M., Haider, M., Chesler, N., Olufsen, M.: Simulating effects of hypoxia on pulmonary haemodynamics in mice. In: Proceedings of the 5th International Conference on Computational and Mathematical Biomedical Engineering (CMBE 2017), vol 1, pp. 271–274. Zeta Computational Resources Ltd. (2017)

Qureshi, M., Colebank, M., Paun, L., Chesler, N., Haider, M., Hill, N., Husmeier, D., Olufsen, M.: A computational study of pulmonary hemodynamics in healthy and hypoxic mice. Biomech. Modell. Mechanobiol. **18**(1), 219–243 (2018)

Ramsay, J.O., Hooker, G., Campbell, D., Cao, J.: Parameter estimation for differential equations: a generalized smoothing approach. J. R. Stat. Soc. B **69**(5), 741–796 (2007)

Rasmussen, C.: Gaussian processes to speed up hybrid Monte Carlo for expensive Bayesian integrals. Bayesian Stat. **7**(7), 651–659 (2003)

Rasmussen, C., Williams, C.: Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning). The MIT Press, Cambridge (2005)

Schiavazzi, D., Arbia, G., Baker, C., Hlavacek, A., Hsia, T., Marsden, A., Vignon-Clementel, I.: Uncertainty quantification in virtual surgery hemodynamics predictions for single ventricle palliation. Int. J. Numer. Methods Biomed. Eng. **32**(3), e02737 (2016)

Sengupta, B., Friston, K., Penny, W.: Efficient gradient computation for dynamical models. Neuroimage **98**, 521–527 (2014)

Sengupta, B., Friston, K., Penny, W.: Gradient-based MCMC samplers for dynamic causal modelling. Neuroimage **125**, 1107–1118 (2016)

Shahriari, B., Swersky, K., Wang, Z., Adams, R.P., de Freitas, N.: Taking the human out of the loop: a review of Bayesian optimization. Proc. IEEE **104**, 148–175 (2016)

Sherlock, C., Golightly, A., Henderson, D.: Adaptive, delayed-acceptance MCMC for targets with expensive likelihoods. J. Comput. Graph. Stat. **26**(2), 434–444 (2017)

Titsias M (2009) Variational learning of inducing variables in sparse Gaussian processes. In: van Dyk, D., Welling, M. (eds) Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics, PMLR, Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA, Proceedings of Machine Learning Research, vol. 5, pp. 567–574

Turner, B., Sederberg, P., Brown, S., Steyvers, M.: A method for efficiently sampling from distributions with correlated dimensions. Psychol. Methods **18**(3), 368–384 (2013)

Vanhatalo, J., Riihimäki, J., Hartikainen, J., Jylänki, P., Tolvanen, V., Vehtari, A.: GPstuff: Bayesian modeling with Gaussian processes. J. Mach. Learn. Res. **14**(1), 1175–1179 (2013)

Wang, Z., Mohamed, S., de Freitas, N.: Adaptive Hamiltonian and Riemann Manifold Monte Carlo samplers. In: Proceedings of the 30th International Conference on International Conference on Machine Learning—Vol. 28, JMLR.org, ICML'13, pp. III–1462–III–1470 (2013)

Wilkinson, D.J.: Bayesian methods in bioinformatics and computational systems biology. Brief. Bioinform. **8**(2), 109–116 (2007)

Wilkinson, R.: Accelerating ABC methods using Gaussian processes. In: Kaski, S., Corander, J. (eds) Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics, PMLR, Reykjavik, Iceland, Proceedings of Machine Learning Research, vol. 33, pp. 1015–1023 (2014)

Wu, K., Li, J.: A surrogate accelerated multicanonical Monte Carlo method for uncertainty quantification. J. Comput. Phys. **321**, 1098–1109 (2016)

Zhang, C., Shahbaba, B., Zhao, H.: Hamiltonian Monte Carlo acceleration using surrogate functions with random bases. Stat. Comput. **27**(6), 1473–1490 (2017)