

# **MCMC with Delayed Acceptance using a Surrogate Model with an Application to Cardiovascular Fluid Dynamics**

**L. Mihaela Paun<sup>1</sup>, Mitchel Colebank<sup>2</sup>, M. Umar Qureshi<sup>2</sup>, Mette Olufsen<sup>2</sup>, Nicholas Hill<sup>1</sup>, Dirk Husmeier<sup>1</sup>**

<sup>1</sup>School of Mathematics and Statistics, University of Glasgow  
Glasgow, UK, G12 8QQ

[l.paun.1@research.gla.ac.uk](mailto:l.paun.1@research.gla.ac.uk); [nicholas.hill@glasgow.ac.uk](mailto:nicholas.hill@glasgow.ac.uk); [dirk.husmeier@glasgow.ac.uk](mailto:dirk.husmeier@glasgow.ac.uk)

<sup>2</sup>Department of Mathematics, NC State University  
Raleigh, NC 27695, USA

[mjcoleba@ncsu.edu](mailto:mjcoleba@ncsu.edu); [muquresh@ncsu.edu](mailto:muquresh@ncsu.edu); [msolufse@ncsu.edu](mailto:msolufse@ncsu.edu)

**Abstract** - Parameter estimation and uncertainty quantification in physiological modelling is a vital step towards personalised medicine. Current state-of-the-art in this research area performs parameter optimisation, with very limited uncertainty quantification. This paper demonstrates the advantage of novel sampling methods, applied on a complex biological PDE system of the pulmonary circulation. The aim is to find an efficient and accurate method for the inference and uncertainty quantification of unknown parameters, relevant for disease diagnosis (pulmonary hypertension) from limited and noisy blood pressure data. The data likelihood is expensive to evaluate as it requires solving numerically a system of PDEs. Therefore, having a model that best trades off accuracy and computational efficiency is of uppermost importance. In this study, we employ fast Bayesian methods, namely MCMC algorithms coupled with emulation using Gaussian Processes, to achieve a computational speed-up. We compare the Delayed Rejection Adaptive Metropolis algorithm in a History Matching framework to the delayed acceptance Adaptive Metropolis algorithm. The first algorithm draws samples from the approximate posterior distribution, while the latter is guaranteed to generate samples from the exact posterior distribution asymptotically. In this paper we propose and derive the n-steps ahead delayed acceptance Metropolis-Hastings algorithm, which is a generalisation of the classical 1-step ahead delayed acceptance Metropolis-Hastings. We show the superiority of the n-steps ahead algorithm over the 1-step ahead method.

**Keywords:** Emulation, Gaussian Processes, MCMC, Delayed Acceptance, History Matching, PDEs, Cardiovascular modelling

## **Nomenclature**

daMH: delayed acceptance Metropolis-Hastings; DRAM: Delayed Rejection Adaptive Metropolis; daAM: delayed acceptance Adaptive Metropolis; GP: Gaussian Process; ESS: effective sample size; MPSRF: Multivariate Potential Scale Reduction Factor; PDE: partial differential equation.

## **1. Introduction**

An important aspect in cardiovascular modelling is parameter inference and uncertainty quantification. This is because of the increasing complexity of the mathematical models which accommodate multiscale interconnected components and large numbers of blood vessels. The result is a high number of unknown parameters, typically highly correlated, that cannot be measured directly and need to be learnt from the limited amount of noisy data available. This, together with the high computational costs required for solving the system of equations (e.g. partial differential equations (PDEs)), for which there are no closed-form solutions, makes the problem challenging in terms of finding sensible parameter values in a reasonable time frame, suitable for real-time treatment planning. Our study improves on the current state-of-the-art (optimisation) for parameter inference [1]; we employ sampling-based Bayesian methods (MCMC) [2] to provide uncertainty quantification for the parameters of interest. However, traditional MCMC algorithms are computationally slow, as sampling from the posterior distribution requires repeatedly obtaining the data likelihood, which is calculated by solving the PDEs numerically. This motivates us to couple MCMC with emulation [3]. To this end, we build a surrogate model (known as the ‘emulator’) for the (unnormalised) log posterior distribution. We do this in a delayed acceptance framework [4, 5]. Only if a proposal is accepted under the surrogate model do we then use the simulator (i.e. evaluate the PDEs) for the Metropolis-Hastings acceptance/rejection step. The algorithm is exact in the sense of converging to the true posterior distribution. Our study introduces the n-steps ahead delayed acceptance Metropolis-Hastings (daMH) algorithm, and we show its superiority over the classical 1-step ahead daMH. We compare these two algorithms to the History Matching approach (with Delayed Rejection

Adaptive Metropolis), which converges to the approximate posterior distribution. We thus learn a number of parameters that are relevant for the diagnosis and treatment of cardiovascular diseases, such as coronary artery disease, heart failure, and pulmonary hypertension, which is the focus of this study.

## 2. MCMC with emulation using Gaussian Processes

The random walk Metropolis algorithm was first introduced by Metropolis in [6]. Hastings generalised the algorithm, by allowing for an asymmetric proposal distribution in a Metropolis-Hastings (MH) algorithm [7]. Various methodological advancements, like Delayed Rejection Adaptive Metropolis (DRAM) [8], have increased the computational efficiency of MH. DRAM automatically adapts the covariance matrix of the proposal based on past chain samples (AM step), and the algorithm makes a new attempt to move to a new point upon rejection of the proposed point (DR step). These MCMC algorithms can be combined with emulation using Gaussian Processes (GPs) [9, 3], as follows:

- *Initial design stage.* Starting from a space filling design in parameter space (e.g. Sobol sequence), integrate the PDEs numerically for each parameter vector to get the true log likelihood. Use these points to build a GP emulator for the log likelihood.
- *Exploratory phase.* Gather information about the target distribution by running MCMC on the emulated surrogate log posterior of the PDE parameters; the proposed point is subject to a MH accept/reject step, for which the simulator is called. The emulator is sequentially re-trained (i.e. optimum covariance hyperparameters are found by maximisation of the log marginal likelihood of the hyperparameters) as new points are accepted, which are iteratively added as further training points.
- *Sampling phase.* Use the emulator constructed in the exploratory phase to draw samples from the target distribution using MCMC. At this stage, the emulator is no longer updated.

## 3. Delayed acceptance Metropolis-Hastings

### 3.1. Review of classical delayed acceptance MCMC

Consider a probabilistic model  $p(\mathbf{y}|\boldsymbol{\theta})$ , where  $\mathbf{y}$  are the data and  $\boldsymbol{\theta}$  are the model parameters with prior probability  $p(\boldsymbol{\theta})$ . Let  $\tilde{p}(\mathbf{y}|\boldsymbol{\theta})$  denote a computationally cheap surrogate model. The corresponding posterior probabilities are given by  $p(\boldsymbol{\theta}|\mathbf{y}) \propto p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})$  and  $\tilde{p}(\boldsymbol{\theta}|\mathbf{y}) \propto \tilde{p}(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})$ . The delayed acceptance MCMC algorithm, proposed in [4, 5], is based on a two-stage scheme: we first evaluate the acceptance step using the emulator (eq 1) and only upon acceptance do we call the simulator (eq 2) to decide whether to accept or reject the proposed point. This has the potential to achieve computational savings.

1. Given the current parameter vector  $\boldsymbol{\theta}$ , draw new parameters  $\boldsymbol{\theta}^*$  from the proposal distribution  $q(\boldsymbol{\theta}^*|\boldsymbol{\theta})$ , and accept the move with the following acceptance probability:

$$\alpha_1(\boldsymbol{\theta}^*|\boldsymbol{\theta}) = 1 \wedge \frac{\tilde{p}(\boldsymbol{\theta}^*|\mathbf{y})q(\boldsymbol{\theta}|\boldsymbol{\theta}^*)}{\tilde{p}(\boldsymbol{\theta}|\mathbf{y})q(\boldsymbol{\theta}^*|\boldsymbol{\theta})}, \quad (1)$$

where we use the shorthand notation  $a \wedge b = \min\{a, b\}$ .

2. If the proposed parameter vector  $\boldsymbol{\theta}^*$  gets accepted at the first stage, accept it with the following second stage acceptance probability:

$$\alpha_2(\boldsymbol{\theta}^*|\boldsymbol{\theta}) = 1 \wedge \frac{p(\boldsymbol{\theta}^*|\mathbf{y}) \tilde{p}(\boldsymbol{\theta}|\mathbf{y})}{p(\boldsymbol{\theta}|\mathbf{y}) \tilde{p}(\boldsymbol{\theta}^*|\mathbf{y})}. \quad (2)$$

The algorithm preserves detailed balance with respect to the posterior distribution  $p(\boldsymbol{\theta}|\mathbf{y})$ .

PROOF

$$q(\boldsymbol{\theta}^*|\boldsymbol{\theta})\alpha_1(\boldsymbol{\theta}^*|\boldsymbol{\theta})\alpha_2(\boldsymbol{\theta}^*|\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathbf{y}) = q(\boldsymbol{\theta}^*|\boldsymbol{\theta})\alpha_1(\boldsymbol{\theta}^*|\boldsymbol{\theta})\alpha_2(\boldsymbol{\theta}^*|\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathbf{y})\frac{\tilde{p}(\boldsymbol{\theta}|\mathbf{y})}{\tilde{p}(\boldsymbol{\theta}^*|\mathbf{y})} \quad (3)$$

$$= \left[ \tilde{p}(\boldsymbol{\theta}|\mathbf{y})q(\boldsymbol{\theta}^*|\boldsymbol{\theta}) \wedge \tilde{p}(\boldsymbol{\theta}^*|\mathbf{y})q(\boldsymbol{\theta}|\boldsymbol{\theta}^*) \right] \left[ 1 \wedge \frac{p(\boldsymbol{\theta}^*|\mathbf{y})}{p(\boldsymbol{\theta}|\mathbf{y})} \frac{\tilde{p}(\boldsymbol{\theta}|\mathbf{y})}{\tilde{p}(\boldsymbol{\theta}^*|\mathbf{y})} \right] \frac{p(\boldsymbol{\theta}|\mathbf{y})}{\tilde{p}(\boldsymbol{\theta}|\mathbf{y})} \quad (4)$$

$$= \left[ \tilde{p}(\boldsymbol{\theta}|\mathbf{y})q(\boldsymbol{\theta}^*|\boldsymbol{\theta}) \wedge \tilde{p}(\boldsymbol{\theta}^*|\mathbf{y})q(\boldsymbol{\theta}|\boldsymbol{\theta}^*) \right] \left[ \frac{p(\boldsymbol{\theta}|\mathbf{y})}{\tilde{p}(\boldsymbol{\theta}|\mathbf{y})} \wedge \frac{p(\boldsymbol{\theta}^*|\mathbf{y})}{\tilde{p}(\boldsymbol{\theta}^*|\mathbf{y})} \right] \quad (5)$$

This term is invariant with respect to a variable exchange  $\boldsymbol{\theta} \leftrightarrow \boldsymbol{\theta}^*$ , hence detailed balance holds:

$$q(\boldsymbol{\theta}^*|\boldsymbol{\theta})\alpha_1(\boldsymbol{\theta}^*|\boldsymbol{\theta})\alpha_2(\boldsymbol{\theta}^*|\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathbf{y}) = q(\boldsymbol{\theta}|\boldsymbol{\theta}^*)\alpha_1(\boldsymbol{\theta}|\boldsymbol{\theta}^*)\alpha_2(\boldsymbol{\theta}|\boldsymbol{\theta}^*)p(\boldsymbol{\theta}^*|\mathbf{y}). \quad (6)$$

### 3.2. n-steps ahead Delayed acceptance Metropolis-Hastings

The n-steps ahead delayed acceptance MH (daMH) algorithm consists of the following two stages:

1. For  $i$  from 2 to  $n$  : Given the current parameter vector  $\boldsymbol{\theta}_{i-1}$ , draw new parameters  $\boldsymbol{\theta}_i$  from the proposal distribution  $q(\boldsymbol{\theta}_i|\boldsymbol{\theta}_{i-1})$ , and accept the move with the following acceptance probability:

$$\alpha_1(\boldsymbol{\theta}_i|\boldsymbol{\theta}_{i-1}) = 1 \wedge \frac{\tilde{p}(\boldsymbol{\theta}_i|\mathbf{y})q(\boldsymbol{\theta}_{i-1}|\boldsymbol{\theta}_i)}{\tilde{p}(\boldsymbol{\theta}_{i-1}|\mathbf{y})q(\boldsymbol{\theta}_i|\boldsymbol{\theta}_{i-1})}. \quad (7)$$

2. The final proposed parameter vector  $\boldsymbol{\theta}_n$  is subject to a second stage acceptance probability, as follows:

$$\alpha_2(\boldsymbol{\theta}_n|\boldsymbol{\theta}_1) = 1 \wedge \frac{p(\boldsymbol{\theta}_n|\mathbf{y})}{p(\boldsymbol{\theta}_1|\mathbf{y})} \frac{\tilde{p}(\boldsymbol{\theta}_1|\mathbf{y})}{\tilde{p}(\boldsymbol{\theta}_n|\mathbf{y})}. \quad (8)$$

This algorithm preserves detailed balance with respect to  $p(\boldsymbol{\theta}|\mathbf{y})$ .

PROOF

First, we prove by induction that detailed balance holds for the n-steps ahead daMH. We assume detailed balance is satisfied for  $n - 1$  steps ahead daMH – see eq (9) below. Now we prove detailed balance for  $n$  steps ahead daMH.

$$T(\boldsymbol{\theta}_{n-1}|\boldsymbol{\theta}_1)p(\boldsymbol{\theta}_1|\mathbf{y}) = T(\boldsymbol{\theta}_1|\boldsymbol{\theta}_{n-1})p(\boldsymbol{\theta}_{n-1}|\mathbf{y}), \quad (9)$$

where  $T(\boldsymbol{\theta}_{n-1}|\boldsymbol{\theta}_1)$  denotes the  $n - 1$  steps ahead transition probability ( $\boldsymbol{\theta}_1 \rightarrow \boldsymbol{\theta}_{n-1}$ ), which we can take to be:

$$T(\boldsymbol{\theta}_{n-1}|\boldsymbol{\theta}_1) = q(\boldsymbol{\theta}_{n-1}|\boldsymbol{\theta}_1)\alpha(\boldsymbol{\theta}_{n-1}|\boldsymbol{\theta}_1). \quad (10)$$

Detailed balance is satisfied when transitioning from the  $(n - 1)^{\text{th}}$  point to the  $n^{\text{th}}$  point, i.e.

$$T(\boldsymbol{\theta}_n|\boldsymbol{\theta}_{n-1})p(\boldsymbol{\theta}_{n-1}|\mathbf{y}) = T(\boldsymbol{\theta}_{n-1}|\boldsymbol{\theta}_n)p(\boldsymbol{\theta}_n|\mathbf{y}) \Rightarrow T(\boldsymbol{\theta}_n|\boldsymbol{\theta}_{n-1}) = \frac{T(\boldsymbol{\theta}_{n-1}|\boldsymbol{\theta}_n)p(\boldsymbol{\theta}_n|\mathbf{y})}{p(\boldsymbol{\theta}_{n-1}|\mathbf{y})}. \quad (11)$$

Hence, by multiplying (9) and (11), we get:

$$T(\boldsymbol{\theta}_{n-1}|\boldsymbol{\theta}_1)T(\boldsymbol{\theta}_n|\boldsymbol{\theta}_{n-1})p(\boldsymbol{\theta}_1|\mathbf{y}) = T(\boldsymbol{\theta}_1|\boldsymbol{\theta}_{n-1})T(\boldsymbol{\theta}_{n-1}|\boldsymbol{\theta}_n)p(\boldsymbol{\theta}_n|\mathbf{y}), \quad (12)$$

which, by marginalising over  $\boldsymbol{\theta}_{n-1}$ , yields

$$T(\boldsymbol{\theta}_n|\boldsymbol{\theta}_1)p(\boldsymbol{\theta}_1|\mathbf{y}) = T(\boldsymbol{\theta}_1|\boldsymbol{\theta}_n)p(\boldsymbol{\theta}_n|\mathbf{y}). \quad (13)$$

Hence, detailed balance holds in n-steps ahead MH.

Next, we show that detailed balance in daMH for  $n = 3$  holds. The ratio of the second stage acceptance probabilities is

$$\frac{\alpha_2(\theta_n|\theta_1)}{\alpha_2(\theta_1|\theta_n)} = \frac{p(\theta_n|\mathbf{y}) \tilde{p}(\theta_1|\mathbf{y})}{p(\theta_1|\mathbf{y}) \tilde{p}(\theta_n|\mathbf{y})} \Leftrightarrow \alpha_2(\theta_n|\theta_1) = \alpha_2(\theta_1|\theta_n) \frac{p(\theta_n|\mathbf{y}) \tilde{p}(\theta_1|\mathbf{y})}{p(\theta_1|\mathbf{y}) \tilde{p}(\theta_n|\mathbf{y})}. \quad (14)$$

For  $n = 3$ :

$$p(\theta_1|\mathbf{y})q(\theta_2|\theta_1) \left[ 1 \wedge \frac{\tilde{p}(\theta_2|\mathbf{y})q(\theta_1|\theta_2)}{\tilde{p}(\theta_1|\mathbf{y})q(\theta_2|\theta_1)} \right] q(\theta_3|\theta_2) \left[ 1 \wedge \frac{\tilde{p}(\theta_3|\mathbf{y})q(\theta_2|\theta_3)}{\tilde{p}(\theta_2|\mathbf{y})q(\theta_3|\theta_2)} \right] \alpha_2(\theta_3|\theta_1) \quad (15)$$

$$= p(\theta_1|\mathbf{y})q(\theta_2|\theta_1) \left[ 1 \wedge \frac{\tilde{p}(\theta_2|\mathbf{y})q(\theta_1|\theta_2)}{\tilde{p}(\theta_1|\mathbf{y})q(\theta_2|\theta_1)} \right] q(\theta_3|\theta_2) \left[ 1 \wedge \frac{\tilde{p}(\theta_3|\mathbf{y})q(\theta_2|\theta_3)}{\tilde{p}(\theta_2|\mathbf{y})q(\theta_3|\theta_2)} \right] \alpha_2(\theta_1|\theta_3) \frac{p(\theta_3|\mathbf{y}) \tilde{p}(\theta_1|\mathbf{y})}{p(\theta_1|\mathbf{y}) \tilde{p}(\theta_3|\mathbf{y})} \quad (16)$$

$$= \frac{p(\theta_3|\mathbf{y})}{\tilde{p}(\theta_3|\mathbf{y})} \left[ q(\theta_2|\theta_3) \tilde{p}(\theta_3|\mathbf{y}) \wedge q(\theta_3|\theta_2) \tilde{p}(\theta_2|\mathbf{y}) \right] \frac{1}{\tilde{p}(\theta_2|\mathbf{y})} \left[ q(\theta_1|\theta_2) \tilde{p}(\theta_2|\mathbf{y}) \wedge q(\theta_2|\theta_1) \tilde{p}(\theta_1|\mathbf{y}) \right] \alpha_2(\theta_1|\theta_3) \quad (17)$$

$$= p(\theta_3|\mathbf{y})q(\theta_2|\theta_3) \left[ 1 \wedge \frac{\tilde{p}(\theta_2|\mathbf{y})q(\theta_3|\theta_2)}{\tilde{p}(\theta_3|\mathbf{y})q(\theta_2|\theta_3)} \right] q(\theta_1|\theta_2) \left[ 1 \wedge \frac{\tilde{p}(\theta_1|\mathbf{y})q(\theta_2|\theta_1)}{\tilde{p}(\theta_2|\mathbf{y})q(\theta_1|\theta_2)} \right] \alpha_2(\theta_1|\theta_3). \quad (18)$$

This expression is invariant with respect to a path reversal  $\theta_1 \rightarrow \theta_2 \rightarrow \theta_3 \Leftrightarrow \theta_3 \rightarrow \theta_2 \rightarrow \theta_1$ . Hence, detailed balance holds for n-steps ahead daMH when  $n = 3$ . This can be generalised by mathematical induction.

#### 4. History Matching with DRAM

The History Matching [10] scheme zooms into the high log likelihood regions by introducing "waves" of interpolants that increasingly focus on the mode of the log likelihood. In each "wave", a Gaussian Process (GP) model is built for the log likelihood, which rules out regions of space as implausible. In the first "wave", the GP model is built on the entire parameter space. In subsequent 'waves', a subset of the space is discarded based on an implausibility criterion:  $\mathbb{E}(f(\tilde{\theta})|D) + 3\sqrt{\text{var}(f(\tilde{\theta})|D)} < \max(\log p(\mathbf{y}|\theta)) - T$ , where  $T$  is a threshold value chosen so that  $\max(\log p(\mathbf{y}|\theta)) - \log p(\mathbf{y}|\theta) > T$  ensures we obtain enough points in that particular "wave" that satisfy the criterion;  $\max(\log p(\mathbf{y}|\theta))$  is the maximum (true) log likelihood;  $\mathbb{E}(f(\tilde{\theta})|D)$  and  $\sqrt{\text{var}(f(\tilde{\theta})|D)}$  are the GP posterior predictive mean and standard deviation of the log likelihood function  $f(\cdot)$  for a new parameter vector  $\tilde{\theta}$ , given the training points  $D = \{(\theta_i, f(\theta_i))\}_{i=1}^n$ . The left hand side of the equation above ensures that the predicted log likelihood value is less than the exact log likelihood value with probability less than 0.003% – this prevents the emulator from discarding parameter values that the simulator would fail to discard. We continue to zoom into regions of high likelihood in subsequent waves, while ensuring a good trade-off between accuracy and computational efficiency, i.e. by using a reasonable number of points to provide an accurate GP at the cost of fewer PDE evaluations, which are required to obtain  $p(\mathbf{y}|\theta)$ . The next step is to run MCMC on the surface of the emulated posterior distribution; we chose DRAM [8] in our work. Note that as opposed to daMH, there is no guarantee of convergence to the exact posterior distribution in the asymptotic limit, i.e. the resulting samples will only come from an approximate posterior distribution.

#### 5. Application – Pulmonary circulation

Our 21-vessel network model of pulmonary arteries [1], expressed as a system of PDEs, predicts one-dimensional (1D) arterial flow and pressure in the lungs. The fluid dynamics model, derived from the incompressible axisymmetric Navier–Stokes equations for a Newtonian fluid, is coupled with a constitutive equation predicting the elasticity of the blood vessel walls:

$$\frac{\partial A}{\partial t} + \frac{\partial q}{\partial x} = 0, \quad \frac{\partial q}{\partial t} + \frac{\partial}{\partial x} \frac{q^2}{A} + \frac{A}{\rho} \frac{\partial p}{\partial x} = -\frac{2\pi\mu r}{\delta} \frac{q}{A}, \quad p = p_0 + \frac{4}{3}s \left( 1 - \sqrt{\frac{A_0}{A}} \right), \quad (19)$$

where  $x$  (cm) is the length along the centerline of the vessel and  $t$  (s) is time,  $p$  (mmHg) is the blood pressure,  $q$  (ml/s) is the blood flow rate,  $A$  (cm<sup>2</sup>) is the vessel cross-sectional area,  $A_0$  is the reference cross-sectional area,  $s$  (mmHg) is the arterial stiffness (assumed constant across the network),  $\rho = 1.055$  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. To provide the boundary conditions, we impose flow at the inlet of the main pulmonary artery (MPA), as well as conservation of flow ( $q_p = q_{d_1} + q_{d_2}$ ) and continuity of total pressure ( $p_p = p_{d_1} = p_{d_2}$ ) at the vessel junctions ( $p$ : parent,  $d_1$  and  $d_2$ : daughters), and we attach 3-element Windkessel models (two resistors  $R_1$  and  $R_2$ , and a capacitor,  $C$ ) to the end of every terminal artery, i.e.

$$Z(\omega) = R_1 + \frac{R_2}{1 + i\omega CR_2} \implies q(L, t) = \frac{1}{T} \int_0^T p(L, t - \tau) Z(\tau) d\tau, \quad (20)$$

where  $Z(\omega)$  is the impedance vector,  $\omega$  is the angular frequency,  $T$  is the length of the cardiac cycle,  $R_1$  and  $R_2$  (mmHg s/ml) are the proximal and distal peripheral resistances, and  $C$  (ml/mmHg) is the peripheral capacitance. The system of equations (19) are solved numerically using the Richtmyer two step Lax-Wendroff scheme. The Windkessel elements ( $R_1, R_2, C$ ) vary across the different terminal arteries. Their nominal values are calculated first ( $R_{01}, R_{02}, C_0$  in eq (21)) [11], then a scaling factor for these estimates is introduced ( $r_1, r_2, c$ ), and it is kept constant across all terminal arteries:

$$R_1 = (1 - 0.5r_1)R_{01}, \quad R_2 = (1 - 0.5r_2)R_{02} \quad \text{and} \quad C = (1 - 0.5c)C_0. \quad (21)$$

The parameter vector to be estimated is 4-dimensional:  $\theta = \{s, r_1, r_2, c\}$ . The data used in our study [1] are a time series of blood flow and pressure from a proximal location in the MPA in a healthy mouse lung; this was obtained by averaging blood pressure and flow measurements taken across 20 heart beats at that location. We use the measured blood flow as inflow in the MPA and the parameter estimation is based on the invasively measured pressure (left-hand panel of Figure 2) [12].

## 6. Simulations

In our code we use Matlab and we implement the GP models using the GPstuff toolbox [13], and the DRAM and AM algorithms make use of the MCMC toolbox [8] <sup>1</sup>. The pressure waveforms are simulated by numerically integrating the PDEs using the C++ software developed by Olufsen et al. [1]. A numerical integration of the PDEs for a given parameter vector requires about 20s (CPU time) on our hardware. <sup>2</sup> We assume the following:

- Likelihood based on the simulator:  $y_i | \theta \sim \mathcal{N}(m_i(\theta), \sigma^2)$ ,  $i = 1, \dots, n$ , where  $n$ : number of time points, i.e.

$$p(\mathbf{y} | \theta, \sigma^2) = \left( \frac{1}{\sqrt{2\pi\sigma^2}} \right)^n \exp \left( -\frac{\sum_{i=1}^n (y_i - m_i(\theta))^2}{2\sigma^2} \right), \quad (22)$$

where  $\mathbf{m}(\theta) = (m_1(\theta), \dots, m_n(\theta))$  is the vector of predictions from the PDEs and  $\mathbf{y} = (y_1, \dots, y_n)$  is the measured data.

- Likelihood predicted by the emulator:  $\tilde{p}(\mathbf{y} | \theta, \sigma^2)$  is the GP posterior predictive mean for the particular  $\theta$ . We emulate the log likelihood, not the signal.
- Priors:  $\theta_j \sim \text{Beta}(1, 1)$ ,  $l_j \leq \theta_j \leq u_j$ , i.e. Beta(1, 1) distribution with support within physiologically sensible ranges  $[l_j, u_j]$  ( $j = 1 \dots d$ , with  $d$ : parameter dimensionality).

Also,  $\sigma^2 \sim \text{Inv-Gamma}(a, b)$ , with  $a = 0.5n_s$  and  $b = 0.5n_s\gamma_s^2$ , where  $\gamma_s^2 = \frac{\min(\sum_{i=1}^n (y_i - m_i(\theta))^2)}{n - d}$  is the prior value for  $\sigma^2$  and  $n_s = 1$  is the prior accuracy for  $\gamma_s^2$  [11]:  $p(\sigma^2) = \frac{b^a}{\Gamma(a)} (\sigma^2)^{-a-1} \exp\left(-\frac{b}{\sigma^2}\right)$ .

- Posterior distribution:  $p(\theta, \sigma^2 | \mathbf{y}) \propto p(\mathbf{y} | \theta, \sigma^2) p(\theta) p(\sigma^2)$  and  $\tilde{p}(\theta, \sigma^2 | \mathbf{y}) \propto \tilde{p}(\mathbf{y} | \theta, \sigma^2) p(\theta) p(\sigma^2)$ .

<sup>1</sup>MCMC toolbox available from <http://helios.fmi.fi/~lainema/dram/>

<sup>2</sup>RedHat Enterprise Linux 6 machine

At every MCMC iteration, we sample  $\theta$  using MCMC (e.g. emulation AM algorithm), and conditional on  $\theta$ ,  $\sigma^2$  is sampled in a Gibbs step from  $p(\sigma^2|\theta, \mathbf{y})$ , which due to the conjugacy of the prior is available in closed form:

$$p(\sigma^2|\theta, \mathbf{y}) = \text{Inv-Gamma} \left( \frac{n}{2} + a, 0.5 \sum_{i=1}^n (y_i - m_i(\theta))^2 + b \right). \quad (23)$$

We keep  $\sigma^2$  fixed in the burnin phase and infer it in the sampling phase.

We check for convergence to the correct posterior distribution using the effective sample size (ESS) [14] and the Brooks Multivariate Potential Scale Reduction Factor (MPSRF) [15]. We choose 3 'waves' for the History Matching approach, as the GP diagnostics [16] show good agreement between the emulator predictions and the simulator outputs after the third wave. Subsequently, we run a number of DRAM iterations for the burnin phase until  $\text{MPSRF} \leq 1.1$ . Next, we draw 1000 samples to constitute the sampling phase. For the emulation MCMC algorithms, we draw 600 parameter configurations from the Sobol sequence, which represents the initial design stage of the method. For the exploratory phase, we run the MH algorithm with initial parameter values taken to be the points in the initial design stage which had the highest log likelihood value. The samples drawn in the exploratory phase are used to refine the GP emulator. For the sampling phase we draw 1000 samples with the following two algorithms: the 1-step ahead daAM, and the n-steps ahead daAM. A squared exponential covariance kernel [9] is used when building the GP emulator. The initial design is constructed over a compact domain. To avoid potential numerical instabilities, parameters are mapped into the unit interval for building the GP emulator and then mapped back into the original domain for the PDE simulator. In the sampling phase, we run 10 Markov chains in parallel; each chain has a different initial value for the parameters (chosen randomly from the samples visited in the exploratory phase to increase the likelihood that we do not start in a low posterior distribution region) and is started from different random number generator seeds. We record the minimum ESS across all parameters, and the minimum ESS normalised by the elapsed time for the simulation. We record the elapsed time using the function `timeit` in `Matlab`, which times the programme a number of times and returns the median elapsed time.

## 7. Results

The right-hand panel of Figure 1 shows kernel density estimation plots for the marginal posterior distributions of the PDE parameters  $\theta = (s, r_1, r_2, c)$ , obtained with the three algorithms compared in our study: 1-step ahead daAM, n-steps ahead daAM and History Matching. The distributions mostly overlap, and since the recorded MPSRF is  $\leq 1.1$  for all three methods, this indicates that the algorithms provide posterior samples from approximately the same distribution. This is reinforced by the QQ plots in the left-hand panel of Figure 1, which show the posterior samples from every algorithm against the posterior samples for every other algorithm. The next step is to obtain the posterior mean values for the parameters and insert them into the mathematical solver. We obtain a simulated blood pressure waveform (left-hand panel of Figure 2), which reveals good agreement between the measured blood pressure data and our predicted data, except for a peak offset, potentially caused by the model specifying the elastic behaviour of the blood vessels. We next focus on critically assessing the performance of the three algorithms tested. The right-hand panel of Figure 2 shows the minimum ESS across all 4 parameters; we notice that our algorithm, n-steps ahead daAM, with  $n = 10$  (chosen to ensure an acceptance probability  $> 80\%$ ), clearly outperforms the classical 1-step ahead daAM, and the latter is only slightly better than the approximate method, History Matching with DRAM. The right-hand panel of Figure 2 also shows the minimum ESS normalised by the median elapsed time. The History Matching with DRAM approach has the advantage of being very fast as it does not evaluate the expensive PDEs in the sampling procedure, whereas both 1-step ahead daAM and n-steps ahead daAM rely on numerically integrating the PDEs, and hence come at a higher computational cost than History Matching. Nevertheless, we observe that our algorithm, the n-steps ahead daAM is better than the 1-step version of it; they both have the advantage of drawing samples from the exact target distribution, compared to the History Matching approach, which generates samples from the approximate target distribution. The QQ plots suggest that in our problem, the GP model for the posterior distribution in the History Matching method are an accurate representation of the true posterior distribution. The mean acceptance probabilities across all 10 runs are: History Matching with DRAM (51%), 1-step ahead daAM (35%), n-steps ahead daAM (90%).

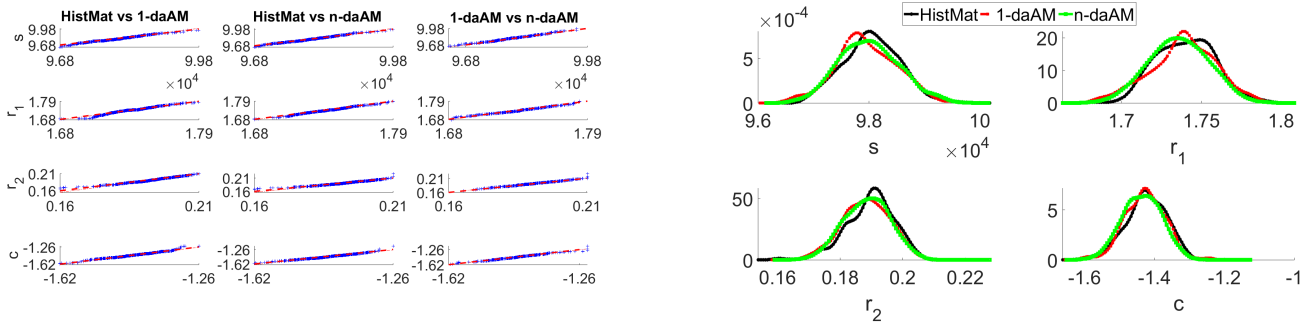


Fig. 1: Left panel: QQ plots, where posterior distributions (based on 1000 samples) from every emulation MCMC algorithm (History Matching with DRAM, 1-step ahead delayed acceptance Adaptive Metropolis (daAM), n-steps ahead daAM) are plotted against each other for every parameter ( $s$ : stiffness,  $r_1, r_2, c$ : Windkessel scaling parameters). Right panel: Kernel density estimate plots for the marginal posterior probability densities (based on 1000 samples) of the 4 parameters.

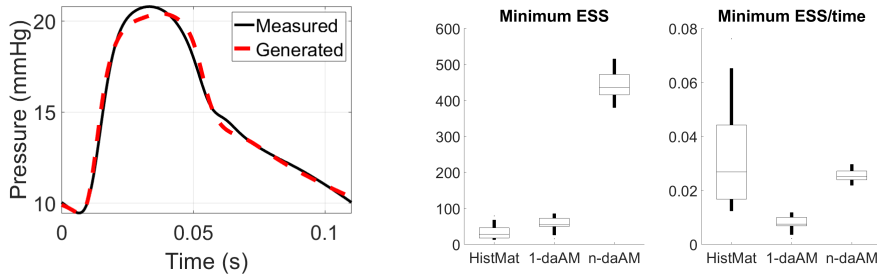


Fig. 2: Left panel: Measured blood pressure time series (black solid line) compared with simulated pressure time series (red long dashed line) obtained by using the posterior mean parameter values in the numerical integration of the PDEs. The posterior mean values were obtained from the simulation with n-steps ahead delayed acceptance Adaptive Metropolis (daAM). Right panel: Distribution of minimum effective sample size (ESS) across parameters, and min(ESS) normalised by median elapsed time for all algorithms tested. Distribution obtained based on 10 chains, with 1000 chain samples (post burnin), drawn using the emulation MCMC algorithms: History Matching with DRAM, 1-step ahead daAM, n-steps ahead daAM.

## 8. Conclusion

This study looks at fast MCMC algorithms combined with emulation of the (unnormalised) posterior distribution using GPs. We have demonstrated the accuracy and efficiency of this approach for parameter inference and uncertainty quantification in computationally expensive mathematical models, which has the potential to become a valuable tool in the decision support system for personalised medicine. We have introduced and derived the n-steps ahead daMH, and have shown that it outperforms the classical 1-step ahead daMH in terms of efficiency. We have compared these algorithms to the History Matching method with GPs, which provides comparable performance, as quantified by the effective sample size normalised by the simulation elapsed time. We would expect the History Matching approach to degrade in accuracy if more parameters were used, making it more difficult to ensure a good coverage of the parameter space, and hence a less accurate emulator. In contrast, as daMH is mathematically guaranteed to converge to the true posterior distribution, a suboptimal emulator will not affect the accuracy, rather the computational efficiency will drop. Future work will include learning the correlation of the measurement noise, which is currently assumed to be independent and identically distributed, an assumption which might not be valid [1].

## Acknowledgements

This work is part of the research programme of the Centre for Multiscale Soft Tissue Mechanics with Application to Heart & Cancer (SoftMech), funded by the Engineering and Physical Sciences Research Council (EPSRC) of the UK, grant reference number EP/N014642/1. Dirk Husmeier is supported by a grant from the Royal Society of Edinburgh, award number 62335. Olufsen and Qureshi are supported by the National Science Foundation (NSF) grant reference number NSF-DMS 1615820, Olufsen and Colebank are supported by NSF-DMS 1246991, and Colebank is additionally supported by the American Heart

Foundation (AHA) grant number 19PRE34380459. The data were made available by N. Chesler, Department of Biomedical Engineering, University of Wisconsin, Madison.

## References

- [1] M. Qureshi, M. Colebank, L. Paun, N. Chesler, M. Haider, N. Hill, D. Husmeier, and M. Olufsen, “A computational study of pulmonary hemodynamics in healthy and hypoxic mice,” *Biomechanics and Modelling in Mechanobiology*, vol. 18, no. 1, pp. 219–243, 2018.
- [2] W. Gilks, S. Richardson, and D. Spiegelhalter, *Markov Chain Monte Carlo in Practice*. Chapman & Hall/CRC Interdisciplinary Statistics, Taylor & Francis, 1995.
- [3] C. Rasmussen, “Gaussian Processes to Speed up Hybrid Monte Carlo for Expensive Bayesian Integrals,” *Bayesian Statistics 7*, vol. 7, pp. 651–659, 01 2003.
- [4] J. Christen and C. Fox, “Markov Chain Monte Carlo using an approximation,” *Journal of Computational and Graphical Statistics*, vol. 14, no. 4, pp. 795–810, 2005.
- [5] C. Sherlock, A. Golightly, and D. Henderson, “Adaptive, Delayed-Acceptance MCMC for Targets With Expensive Likelihoods,” *Journal of Computational and Graphical Statistics*, vol. 26, no. 2, pp. 434–444, 2017.
- [6] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, “Equation of state calculations by fast computing machines,” *The Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, 1953.
- [7] W. K. Hastings, “Monte Carlo sampling methods using Markov chains and their applications,” *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970.
- [8] H. Haario, M. Laine, A. Mira, and E. Saksman, “DRAM: Efficient adaptive MCMC,” *Statistics and Computing*, vol. 16, no. 4, pp. 339–354, 2006.
- [9] C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.
- [10] R. Wilkinson, “Accelerating ABC methods using Gaussian processes,” in *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics* (S. Kaski and J. Corander, eds.), vol. 33 of *Proceedings of Machine Learning Research*, (Reykjavik, Iceland), pp. 1015–1023, PMLR, 2014.
- [11] L. Paun, M. Qureshi, M. Colebank, N. Hill, M. Olufsen, M. Haider, and D. Husmeier, “MCMC methods for inference in a mathematical model of pulmonary circulation,” *Statistica Neerlandica*, vol. 72, no. 3, pp. 306–338, 2018.
- [12] D. Tabima, A. Roldan-Alzate, Z. Wang, T. Hacker, R. Molther, and N. Chesler, “Persistent vascular collagen accumulation alters hemodynamic recovery from chronic hypoxia,” *J. Biomech*, vol. 45, no. 5, pp. 799–804, 2012.
- [13] J. Vanhatalo, J. Riihimäki, J. Hartikainen, P. Jylänki, V. Tolvanen, and A. Vehtari, “GPstuff: Bayesian Modeling with Gaussian Processes,” *J. Mach. Learn. Res.*, vol. 14, pp. 1175–1179, Apr. 2013.
- [14] R. Kass, B. Carlin, A. Gelman, and R. Neal, “Markov Chain Monte Carlo in Practice: A Roundtable Discussion,” *The American Statistician*, vol. 52, no. 2, pp. 93–100, 1998.
- [15] S. Brooks and A. Gelman, “General methods for monitoring convergence of iterative simulations,” *Journal of Computational and Graphical Statistics*, vol. 7, no. 4, pp. 434–455, 1998.
- [16] L. Bastos and A. O’Hagan, “Diagnostics for Gaussian Process Emulators,” *Technometrics*, vol. 51, no. 4, pp. 425–438, 2009.