

Inference in a Partial Differential Equations Model of Pulmonary Arterial and Venous Blood Circulation using Statistical Emulation

Umberto Noè⁽¹⁾, Weiwei Chen⁽¹⁾, Maurizio Filippone⁽²⁾, Nicholas Hill⁽¹⁾,
and Dirk Husmeier⁽¹⁾

(1) SoftMech and University of Glasgow
School of Mathematics and Statistics, 15 University Gardens, Glasgow, UK.
u.no.e.1@research.gla.ac.uk

(2) Eurecom
Campus SophiaTech, 450 Route des Chappes, Biot, France.

Keywords: Statistical Inference, Gaussian Processes, Emulation, Simulator, Global Optimization, Nonlinear Differential Equations, Pulmonary Blood Circulation.

Abstract. The present article addresses the problem of inference in a multiscale computational model of pulmonary arterial and venous blood circulation. The model we consider is a computationally expensive simulator of physiological processes which, given specific parameter values, solves a system of nonlinear partial differential equations and returns predicted pressure and flow values at different locations in the arterial and venous blood vessels. The standard approach in parameter calibration for computer codes is to emulate the simulator using a Gaussian Process (GP) prior. In the present work, we take a different approach and emulate the objective function itself, i.e. the residual sum of squares between the simulations and the observed data. We demonstrate that this modified emulator achieves a reduction in the computational costs of inference by two orders of magnitude (from 8 hours to 3 minutes CPU time).

1 Introduction

The present work is motivated by a partial differential equations (PDE) model of the pressure and flow wave propagation in the pulmonary arterial and venous system under normal physiological and pathological conditions. This model is an extension of previous studies, which considered only the arterial system or part of the venous system [3]. The PDEs depend on various bio-physical parameters, related e.g. to blood vessel geometry or fluid dynamics. These parameters can typically not be measured *in vivo* and hence need to be inferred indirectly from the observed blood flow and pressure distributions. In principal, this is straightforward. Under the assumption of a suitable noise model, the solutions of the PDEs define the likelihood of the data, and the parameters can then be inferred in a maximum likelihood sense. However, a closed-form solution of the maximum likelihood equations is typically not available, which calls for an iterative optimization procedure. Since a closed-form solution of the PDEs is not available either, each optimization step requires a numerical solution of the PDEs. This is numerically expensive, especially given that the likelihood function is typically multi-modal, and the optimization problem is NP-hard. In the present work, our goal is to reduce the computational costs of inference with the concept of *emulation*. This is to be distinguished from the explicit numerical solution of the PDEs, which we henceforth refer to as *simulation* (or *simulator* when referring to a specific solution).

2 Model

In our model of the pulmonary circulation, seven large arteries and four large veins are modelled explicitly, while the smaller vessels are represented by structured trees

(Figure 1). A magnetic resonance imaging (MRI) based measurement of the right ventricular output provides the inlet flow for the system.

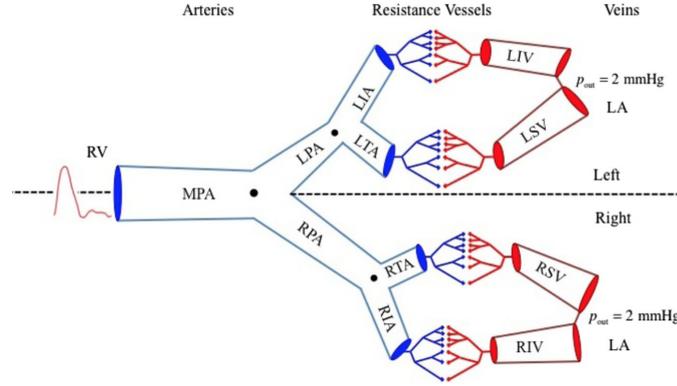


Figure 1: Schematic of the pulmonary circulation consisting of large arteries, arterioles, venules and large veins from Qureshi et al. [5]. Seven large arteries are considered in this model, i.e. the main pulmonary artery (MPA), the left (LPA) and right (RPA) pulmonary arteries, the left interlobular artery (LIA), the left trunk artery (LTA), the right interlobular artery (RIA), and the right trunk artery (RTA). The four terminal arteries LIA, LTA, RIA, and RTA are connected to four large veins, i.e. the left inferior vein (LIV), left superior vein (LSV), right inferior vein (RIV), and right superior vein (RSV), via structured trees of resistance vessels.

The large arteries and veins are modelled as tapered elastic tubes, and the geometries are based on measurements of proximal and distal radii and vessel lengths [5]. The cross-sectional area averaged blood flow and pressure are predicted from a non-linear model based on the incompressible Navier-Stokes equations for a Newtonian fluid [3].

The small arteries and veins are modelled as structured trees at each end of the terminal large arteries and veins to mimic the dynamics in the vascular beds [5]. With a given parent vessel r_p , the radii of daughter vessels are scaled linearly with radius $r_{d_1} = \alpha r_p$ and $r_{d_2} = \beta r_p$, where α and β are the scaling factors. The vessels bifurcate until the radii of each terminal vessel is smaller than a given minimum radius r_{\min} . The radius relation over the bifurcations is

$$r_p^\xi = r_{d_1}^\xi + r_{d_2}^\xi, \quad 2.33 \leq \xi \leq 3.0, \quad (1)$$

where the exponent $\xi = 2.33$ corresponds to laminar flow, $\xi = 3.0$ corresponds to turbulent flow [3], p represents parent vessel, and d_1 and d_2 represent daughter vessels. With the definition of the area ratio $\eta = (r_{d_1}^2 + r_{d_2}^2)/r_p^2$ and the asymmetry ratio $\gamma = (r_{d_2}/r_{d_1})^2$ it is possible to determine the scaling factors α and β according to the relations $\alpha = (1 + \gamma^{\xi/2})^{-1/\xi}$ and $\beta = \alpha\sqrt{\gamma}$. The parameters, ξ , γ , r_{\min} and a given root radius r_0 , determine the size and density of the structured tree. The cross-sectional area averaged blood flow and pressure in these small arteries and veins are computed from the linearized incompressible axisymmetric Navier-Stokes equations [5].

The system of nonlinear partial differential equations is available from Qureshi et al. [5], and its numerical solution, which depends on various bio-physical parameters, will henceforth be referred to as the ‘simulator’. Particular interest lies in the estimation of the parameter ξ , because low values are indicative of the clinically relevant problem of vascular rarefaction, as in pulmonary hypertension. Its estimation is performed in the numerically stable range $2.65 \leq \xi \leq 3$, keeping the other parameters of the model fixed to biologically meaningful values from the literature, and by focusing on the pressure change over time and location in the MPA.

Computational inference of the bio-physical parameters entails repeated forward simulations for different parameter configurations. In this model a forward simulation takes around 12 seconds of CPU time. Given the multi-modality of the objective

function, a standard global optimization algorithm requires a large number of forward simulations, which comes at substantial computational costs even for the inference of just a single parameter. In the following section, we discuss a faster method based on the concepts of statistical emulation and Bayesian optimization.

3 Method

To perform efficient optimization of a computationally expensive objective function, we let a statistical emulator guide the optimization process. The emulator, $f(\cdot)$, is based on a Gaussian Process (GP) prior with kernel function k_ψ , which depends on the hyperparameters ψ . These hyperparameters are inferred in a maximum a posteriori sense. Unlike standard emulation, we aim to directly emulate the objective function $y(\cdot) : \mathbb{X} \rightarrow \mathbb{R}$, where \mathbb{X} is the bio-physical parameter space. We use the following hierarchical Bayesian nonparametric regression model:

$$\begin{aligned} \mathbf{y} \mid \mathbf{f}, \log \sigma &\sim N(\mathbf{f}, \sigma^2) \\ f(\mathbf{x}) \mid \phi, \psi &\sim GP(m_\phi(\mathbf{x}), k_\psi(\mathbf{x}, \mathbf{x}')) \\ \theta = \{\phi, \psi, \log \sigma\} &\sim P(\phi)P(\psi)P(\log \sigma), \end{aligned}$$

where $\mathbf{x}, \mathbf{x}' \in \mathbb{X}$, $\mathbf{y} = [y(\mathbf{x}_1), \dots, y(\mathbf{x}_n)]^\top$, $\mathbf{f} = [f(\mathbf{x}_1), \dots, f(\mathbf{x}_n)]^\top$ and we assume throughout the analysis that $m(\mathbf{x}) = 0$, $\forall \mathbf{x} \in \mathbb{X}$, as a consequence of data standardization. The inputs have been scaled to the unit interval $[0, 1]^d$ throughout the analysis, where d is the dimension of the input (parameter) space. The noise standard deviation was fixed to a small value of $\sigma = 10^{-3}$ to improve the conditioning of the covariance matrix. For the log kernel lengthscale, $\log \ell$, we assume a univariate interval prior with linear decay in log domain such that the prior probability mass is concentrated in the interval $[\log 0.05, \log 0.80]$. This concentration avoids length scales close to zero, which would effectively overfit the data, and lengthscales close to 1, which would lead to over-smoothing. These settings also improve the condition number of the matrix.

The starting point of the optimization should be a good initial guess of the objective function, which is obtained by conditioning the GP on the objective function evaluated at a set of design points in the input (i.e. parameter) space. In the present work, we follow Jones et al. [1] and use a space filling Latin hypercube design, with the number of initial input points set to $10 \times d$.

To minimize the evaluation-costly objective function we use a sequential strategy proposed in [1], called Efficient Global Optimization (EGO), which selects iteratively the point with the highest expected improvement over the incumbent minimum. Let the random variable *improvement* be $I(\mathbf{x}) = \max\{y_{\min} - f(\mathbf{x}), 0\}$, where

- $f(\mathbf{x}) \sim N(m(\mathbf{x}), s^2(\mathbf{x}))$ is the marginal GP at the point of interest \mathbf{x} ;
- $y_{\min} = y(\mathbf{x}_{\min})$ is the best function value known so far;
- $I(\mathbf{x}) > 0$ if \mathbf{x} has a lower function value than the incumbent solution;
- $I(\mathbf{x}) = 0$ otherwise.

The expected improvement (EI) is the expected value of the random variable $I(\mathbf{x})$ and has the analytical formula [2]:

$$\text{EI}(\mathbf{x}) = (y_{\min} - m(\mathbf{x}))\Phi\left(\frac{y_{\min} - m(\mathbf{x})}{s(\mathbf{x})}\right) + s(\mathbf{x})\phi\left(\frac{y_{\min} - m(\mathbf{x})}{s(\mathbf{x})}\right).$$

The problem of directly optimizing the objective function derived from the computationally expensive simulator, $y(\cdot)$, is now shifted to the optimization of the computationally cheap EI acquisition function, or minimizing $-\log \text{EI}(\mathbf{x})$. Even if this function is multimodal, it can be efficiently optimized using standard state-of-the-art global optimization solvers, like the Dividing Rectangles algorithm [4], as the computational costs for obtaining EI are negligible to those required for computing y . Once the minimum

\mathbf{x}_* of $-\log \text{EI}(\mathbf{x})$ has been found, we compute the expensive objective function at the new input \mathbf{x}_* and obtain the output $y_* = y(\mathbf{x}_*)$. The new data are added to the training dataset and the GP hyperparameters are updated. The process continues iteratively until convergence.

4 Illustration

We now illustrate the EGO algorithm on the ‘Sasena’ [7] function, shown in Figure 2 (a), where at each iteration we sample the objective at the input that minimizes $-\log \text{EI}$:

$$y(\mathbf{x}) = 2 + 0.01(x_2 - x_1^2)^2 + (1 - x_1)^2 + 2(2 - x_2)^2 + 7 \sin(0.5x_1) \sin(0.7x_1x_2)$$

$$0 \leq x_1 \leq 5, 0 \leq x_2 \leq 5.$$

At the first iteration the function is evaluated at the LH design (white dots). The GP is

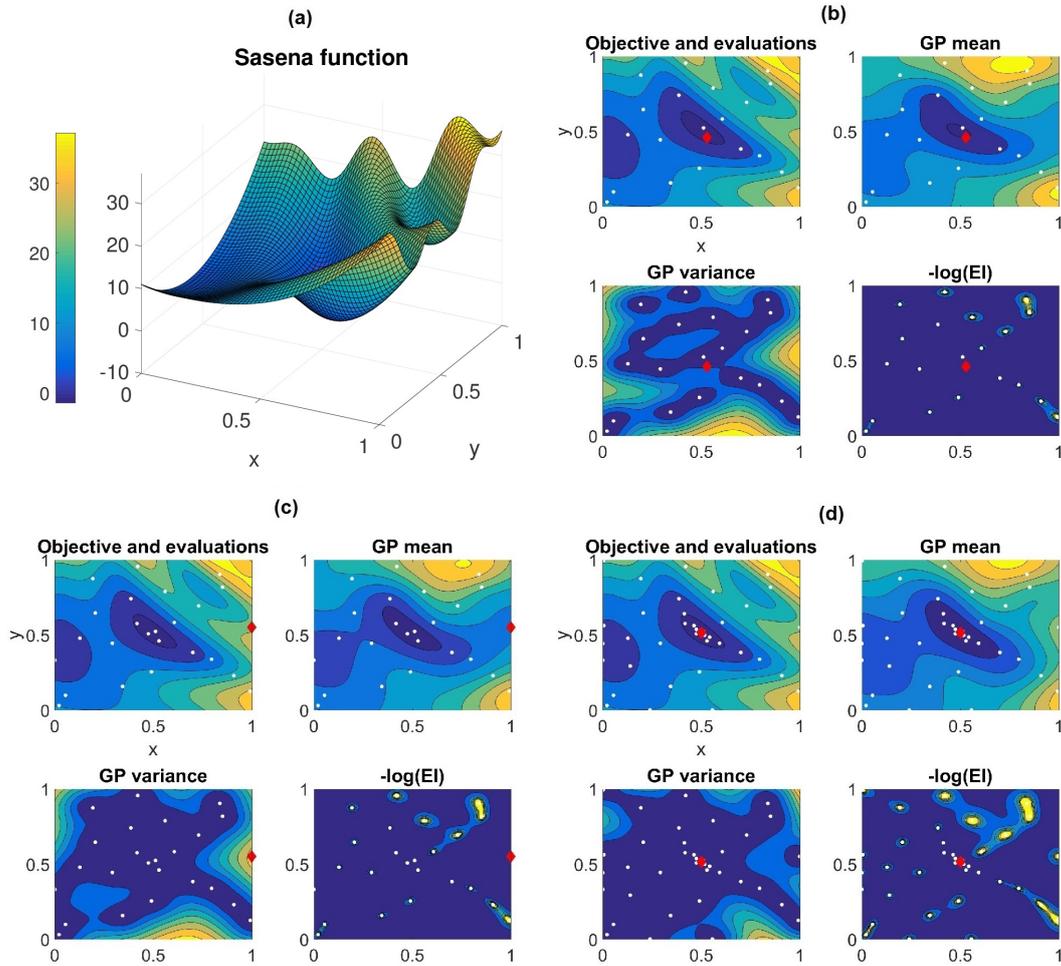


Figure 2: EGO algorithm with Matérn $\nu = 5/2$ kernel. (a) Sasena function, scaled to $[0, 1]^2$, to be minimized. (b) EGO 1st iteration and training inputs (white dots). (c) 5th iteration. (d) 25th and last iteration. The red diamond represents the next evaluation point and it is the minimizer of $-\log \text{EI}$.

trained on the data. We can see that at the beginning, with our initial design and chosen number of design points, we already have a fairly good picture of our objective function. Then iteratively the acquisition function is optimized, the new sampled points are added to the dataset and the GP is trained again. From this illustration, we see that the EI function is higher when the predicted mean is smaller than the incumbent minimum, Figure 2 (b) and (d), and is also higher when the variance is high, Figure 2 (c).

5 Simulation

In order to assess the proposed inference scheme for the pulmonary model, we simulated noise-free pressure data \mathbf{p}_0 (18432-dimensional vectors, from 18 locations measured at

Table 1: Performance of different kernels for RSS minimization.

Kernel	n ^r of iterations	≈ time (min)	ξ_{\min}	$-\sum_{i=1}^n \log P(r_i \xi_i, \mathcal{D}_{-i})$
Neural Network	17	3.5	2.7649	18.8147
Squared Exponential	30	6.1	2.7645	13.7417
Matérn $\nu = 5/2$	43	8.7	2.7602	13.4098

1024 time points) from the PDE model with $\xi_0 = 2.76$. We then created a space filling Latin Hypercube design for ξ in $[2.65, 3]$, and carried out a forward simulation for each element of the design, obtaining a set of simulations $\{\xi_i, \mathbf{p}_i, i = 1, \dots, n\}$. As objective function we chose the residual sum of squares (RSS) between the simulations and the observations, $\tilde{r}_i = \|\mathbf{p}_i - \mathbf{p}_0\|^2$, and we fitted a GP to the training data $\mathcal{D} = \{\xi, \mathbf{r}\}$, where the vector \mathbf{r} is the standardized RSS vector $\tilde{\mathbf{r}}$. We then applied the emulation-based approach and the EGO algorithm described in the previous sections, with input $\mathbf{x}_i = \xi_i$ and output $y_i = r_i$.

6 Results

Table 1 shows results of the emulation-based global minimization of the expensive RSS for three different kernel functions: Neural Network (NN), Squared Exponential (SE) and Matérn with $\nu = 5/2$; see equations 4.29, 4.9 and 4.17 in [6] respectively. The second column reports the number of iterations to convergence, and the third column shows the corresponding CPU time. The convergence criterion we used was 5 consecutive steps having $\|\xi_i - \xi_{i-1}\| < 10^{-6}$ or 5 consecutive function evaluations leading to differences in their values below $|r_i - r_{i-1}| < 10^{-6}$. We also imposed a maximum number of iterations of 1000, and we stopped iterating when the improvement over the incumbent minimum was less than 10^{-12} . The fourth column of Table 1 reports the estimated minimum across three different kernels, and shows that the inferred parameter is consistently close to the true value of $\xi_0 = 2.76$. The fifth column of Table 1 shows the negative sum of log leave-one-out predictive probabilities, where \mathcal{D}_{-i} denotes the training dataset with the i^{th} sample removed. The Matérn kernel has the best out-of-sample predictive performance compared to the other covariance functions. The table shows the trade-off between the number of iterations, the computational time, the accuracy in the estimation of the parameter, and the predictive power. Figure 3 (a) shows the 1st, and (b) the 7th and last iteration of the EGO algorithm. Figure 3 (c) shows model validation diagnostics, following the procedure described in [1]. While the specific illustration is given for the NN kernel, all three covariance functions passed the validation in terms of (c, left) actual function value vs cross-validated predictions, (c, middle) standardized leave-one-out cross validated residuals $(r_i - m_{-i}(\xi_i))/s_{-i}(\xi_i)$ being in $[-3, 3]$, and (c, right) standardized cross-validated residuals vs theoretical values from a sample of n independent standard normal variables.

7 Conclusion

Our aim was to perform inference in a computationally expensive and novel model of the double-sided arterial and venous pulmonary blood circulation. The parameter of interest, ξ , governs the vessel parent-to-daughter radius relation (1), with low values indicating vascular problems of clinical interest. As ξ increases, the number of vessels in the structured tree will also increase; similarly, as it decreases, the number of vessels will also decrease, simulating the vascular rarefaction clinical condition.

In previous studies with state-of-the-art non-emulation-based global optimization algorithms, like genetic algorithms or the method proposed in [8], we found that the

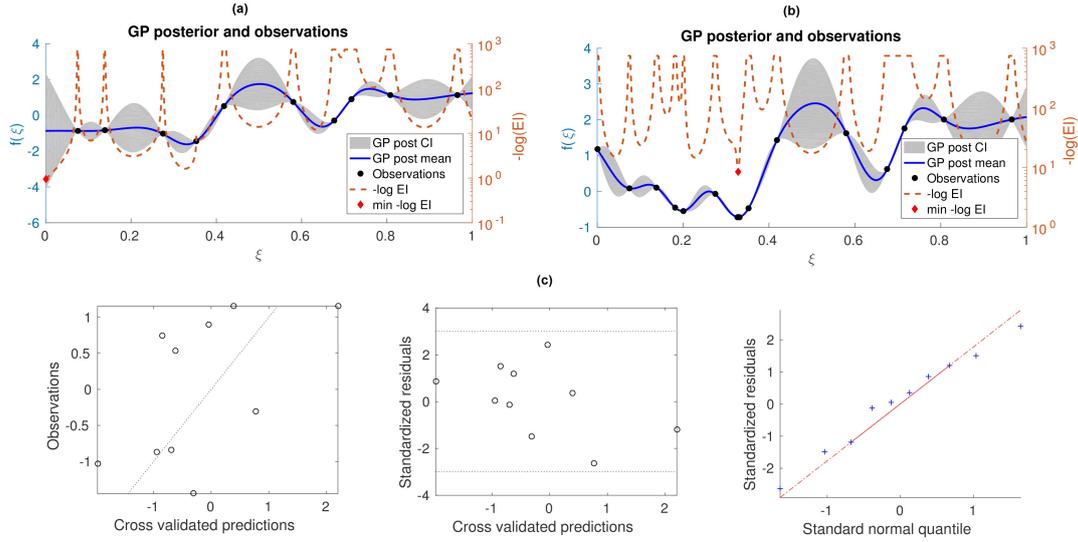


Figure 3: EGO algorithm with NN kernel. (a) 1st iteration. (b) 17th and last iteration. (c) Model validation.

number of required function evaluations was in the order of 2×10^3 . Given that the computational costs of a single forward simulation are about 12 seconds, the total computational costs are in the order of 8 hours. The results shown in Figure 3 show that the proposed emulation-based approach achieves a substantial reduction in the number of forward simulations, to values as low as just 17 iterations, which corresponds to a total reduction of the computational complexity by two orders of magnitude, to just 3 minutes.

In our future work, we plan to apply the proposed emulation scheme to the Holzapfel-Ogden soft tissue mechanics model of the heart [9], where the computational costs of a single forward simulation are in the order of an hour, and the overall computational savings promise to be even more substantial.

Acknowledgments

UN is supported by a scholarship from the Biometrika Trust. SoftMech is a research centre on multi-scale modelling in soft-tissue mechanics, funded by EPSRC (grant no. EP/N014642/1).

References

- [1] D.R. Jones, M. Schonlau and W.J. Welch. “Efficient Global Optimization of Expensive Black-Box Functions”. *Journal of Global Optimization*, vol. 13, pp. 455–492, 1998.
- [2] J. Mockus, V. Tiesis and A. Zilinskas. “The application of bayesian methods for seeking the extremum”. In *Towards Global Optimization*, vol. 2, pp. 117–129, 1978.
- [3] M.S. Olufsen. “Structured tree outflow condition for blood flow in larger systemic arteries”. *American journal of physiology. Heart and circulatory physiology*, 276:H257–H268, 1999.
- [4] C.D. Perttunen, D.R. Jones and B.E. Stuckman. “Lipschitzian optimization without the lipschitz constant”. *Journal of Optimization Theory and Application*, 79(1):157–181, 1993.
- [5] M.U. Qureshi, G.D.A. Vaughan, C. Sainsbury, et al. “Numerical simulation of blood flow and pressure drop in the pulmonary arterial and venous circulation”. *Biomechanics and modeling in mechanobiology*, 13(5): 1137–1154, 2014.
- [6] C.E. Rasmussen and C.K.I. Williams. “Gaussian Processes for Machine Learning”. The MIT Press, 2005.
- [7] M.J. Sasena. “Optimization of Computer Simulations via Smoothing Splines and Kriging Meta-models”. MSc Thesis, University of Michigan, 1998.
- [8] Z. Ugray et al. “Scatter Search and Local NLP Solvers: A Multistart Framework for Global Optimization”. *INFORMS Journal on Computing*, vol. 19, no. 3, pp. 328–340, 2007.
- [9] H.M. Wang et al.. “Structure-based finite strain modelling of the human left ventricle in diastole”. *International Journal for Numerical Methods in Biomedical Engineering*, 29:83–103, 2013.