# Probabilistic Integration for Uncertainty Quantification in Differential Equation Models



Probabilistic Numerics Workshop

Saturday 8th December 2012

Ben Calderhead
Research Fellow
CoMPLEX and Statistical Science
University College London

Joint work with

Prof. Dave Campbell & Oksana Chkrebtii
Department of Statistics
Simon Fraser University

Prof. Mark Girolami
Statistical Science
University College London

An overview of a paper in (final stages of) preparation...

Differential equations are ubiquitous in many areas of Science and Engineering.

Differential equations are ubiquitous in many areas of Science and Engineering.

Particularly useful for modelling natural phenomena in terms of the rates of change of components in a dynamical system.

Differential equations are ubiquitous in many areas of Science and Engineering.

Particularly useful for modelling natural phenomena in terms of the rates of change of components in a dynamical system.

Allow us to describe complex behaviour with few but readily interpretable parameters.

In many cases, particularly in biology, we only have observed and noisy data for the output of the system - we don't know the true input parameter values.
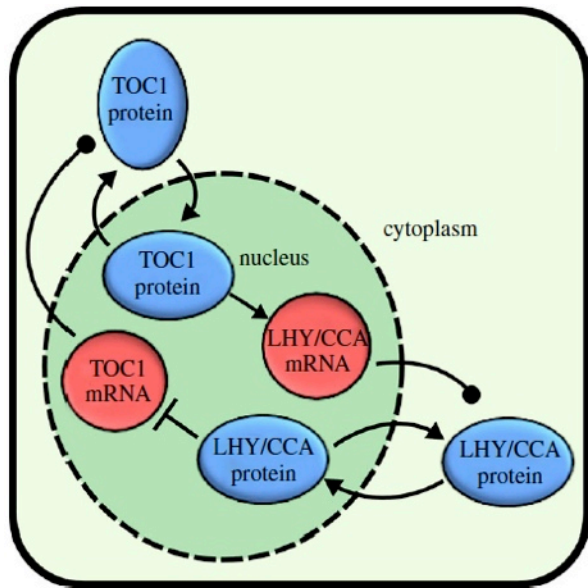
$$\frac{\mathrm{d}[LHY]_m}{\mathrm{d}t} = \frac{n_1[TOC1]_n^a}{g_1^a + [TOC1]_n^a} - \frac{m_1[LHY]_m}{k_1 + [LHY]_m},$$

$$\frac{\mathrm{d}[LHY]_c}{\mathrm{d}t} = p_1[LHY]_m - r_1[LHY]_c$$

$$+ r_2[LHY]_n - \frac{m_2[LHY]_c}{k_2 + [LHY]_c},$$

$$\frac{\mathrm{d}[LHY]_n}{\mathrm{d}t} = r_1[LHY]_c - r_2[LHY]_n - \frac{m_3[LHY]_n}{k_3 + [LHY]_n},$$

$$\frac{\mathrm{d}[TOC1]_m}{\mathrm{d}t} = \frac{n_2\,g_2^b}{g_2^b + [LHY]_n^b} - \frac{m_4[TOC1]_m}{k_4 + [TOC1]_m},$$

$$\frac{\mathrm{d}[TOC1]_c}{\mathrm{d}t} = p_2[TOC1]_m - r_3[TOC1]_c$$

$$+ r_4[TOC1]_n - \frac{m_5[TOC1]_c}{k_5 + [TOC1]_c},$$

$$\text{and } \frac{\mathrm{d}[TOC1]_n}{\mathrm{d}t} = r_3[TOC1]_c - r_4[TOC1]_n$$

$$- \frac{m_6[TOC1]_n}{k_6 + [TOC1]_n}.$$
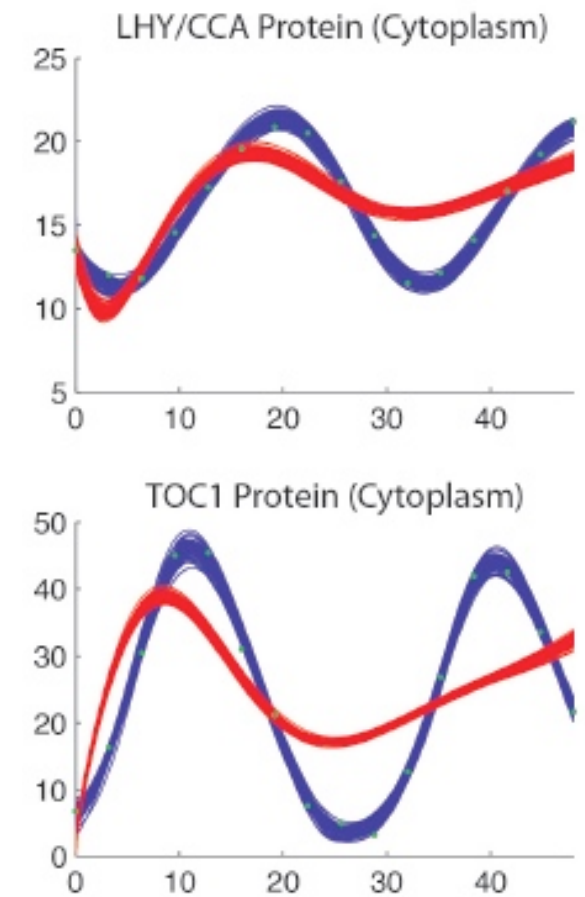
- Often sparse, uncertain data with unobserved species

- Often multiple network topologies consistent with the known biology

We can adopt a **Bayesian approach** to characterise the uncertainty and make inferences about parameters values and underlying structure of the system

- define an error model and prior

- Run your favourite MCMC algorithm,
e.g. Riemannian Manifold MCMC (B.C., PhD Thesis, 2011)

- Calculate marginal likelihoods for model comparison
e.g. thermodynamic integration, Chib's method etc.

- Which parameters should we use for a given model?

- Which model structure is most appropriate to describe the system of interest?
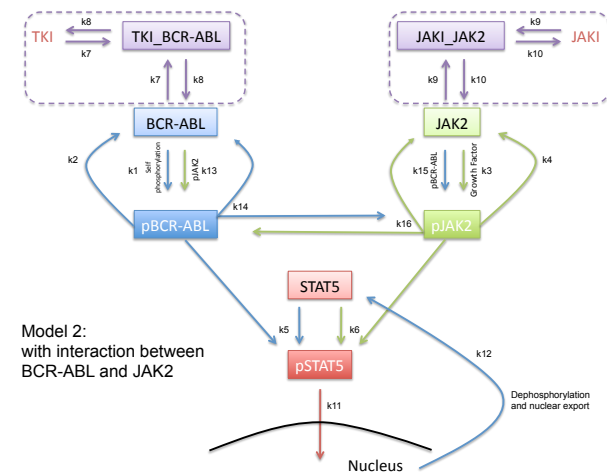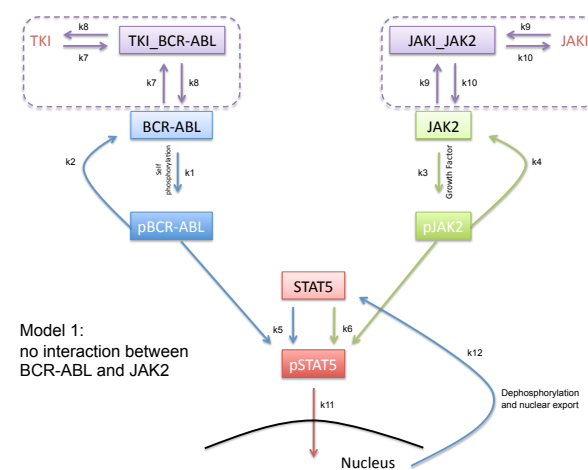


Nonlinear dynamics, correlation structure, identifiability... all create problems for standard MCMC.

In the Bayesian formalism, the posterior distribution characterises the uncertainty in the parameters

$$p(\boldsymbol{\theta}|\mathbf{y}) = \frac{p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathbf{y})} = \frac{p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{\int p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}}$$

We can consider an MCMC approach the "gold standard" as we can calculate quantities to arbitrary precision, given sufficient samples

The challenge: *How do we do this most efficiently?*

Employing a system of ODEs as a statistical model, we can define the log-likelihood as

$$\mathcal{L}(\mathbf{Y}_{\cdot,n}|\mathbf{X}(\boldsymbol{\theta},\mathbf{x}_0)_{\cdot,n},\boldsymbol{\Sigma}_n) \propto \sum_{n=1}^{N} -\frac{1}{2}\left(\mathbf{Y}_{\cdot,n} - \mathbf{X}(\boldsymbol{\theta},\mathbf{x}_0)_{\cdot,n}\right)^T \boldsymbol{\Sigma}_n^{-1}\left(\mathbf{Y}_{\cdot,n} - \mathbf{X}(\boldsymbol{\theta},\mathbf{x}_0)_{\cdot,n}\right)$$

Note the assumption that the solution to the ODEs is exact!

- Majority of differential equations do not have analytical solutions and must be solved numerically

- Deterministic solutions approximately satisfy the model dynamics to within a given error tolerance

- For parameter inference/estimation we make the assumption that the numerical solution is the true solution

- Majority of differential equations do not have analytical solutions and must be solved numerically

- Deterministic solutions approximately satisfy the model dynamics to within a given error tolerance

- For parameter inference/estimation we make the assumption that the numerical solution is the true solution

Is there a way of solving differential equations in a statistical manner that more fully characterises the uncertainty in the numerical solution?

## Main Idea

Develop a sequential Bayesian numerical integrator for solving systems of differential equations using Gaussian processes.

## Main Idea

Develop a sequential Bayesian numerical integrator for solving systems of differential equations using Gaussian processes.

First proposed by John Skilling in 1991 in a conference paper, however only for very simple ODEs and not computationally efficient. There does not appear to have been any further work in this area since then.

**Why solve differential equations probabilistically?**

- Can choose how much computational effort to put into the solution and fully characterise the uncertainty in the approximate solution

- Uncertainty in the solution may be propagated through the rest of a Bayesian analysis

- Could choose function evaluation points in a principled manner based on the variance of the predicted solution at any point

Recap: Numerically solving ordinary differential equations

$$\begin{cases} \dot{x}(t) = f_\theta(t, x(t)), & t \in [a, b], \\ x(a) = x_0, \end{cases}$$

We may solve this approximately using a numerical method.
e.g. Euler's method, Runge-Kutta, many others...

$$x_{n+1} = x_n + \epsilon f_\theta(x_n)$$
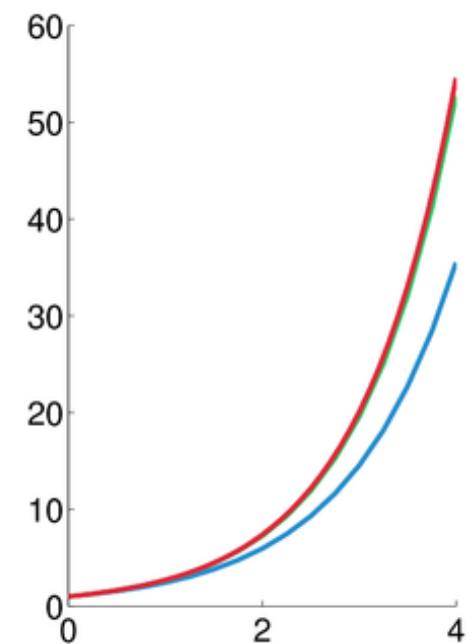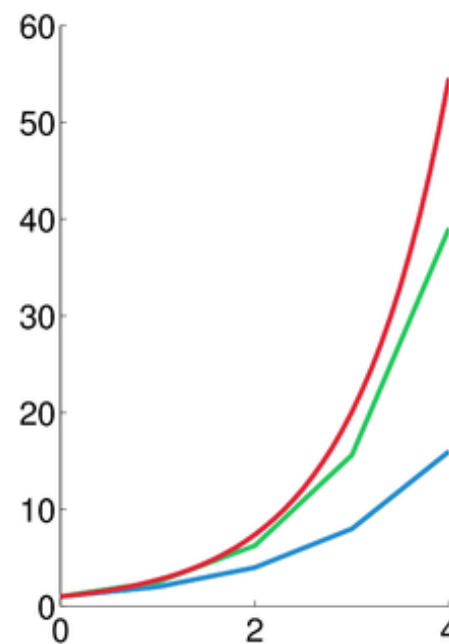


Many sophisticated, high speed solvers are available. After many iterations we obtain a solution that approximates the true solution to with a specified error tolerance.

In this work we shall model the *derivative* of the solution as a **Gaussian process** (using the original notation of Skilling)

$$\dot{x}(t) = \dot{l}(t) + \int_{-\infty}^{\infty} R(z,t)h(z)dz$$

$$= Rh(t)$$

using the representation as a convolution of a Gaussian white noise process, h, and some integrable kernel function, R.

$$x(t) = l(t) + Qh(t)$$

where Q is the integrated kernel function $Q(t,z) = \int_{a}^{t} R(x,z)dx$

and the choice of function $l(t) = \int_{a}^{t} \dot{l}(x)dx$ is a modelling step

to allow us to satisfy boundary conditions.

**Goal**: to obtain the posterior distribution of the solution process given a sequence of vector field evaluations

$$f(\mathbf{s}, \tilde{\mathbf{x}}) = f_\theta(s_1, \tilde{x}(s_1)) : f_\theta(s_N, \tilde{x}(s_N))$$

For given model parameters, we assume that the vector field evaluations are noisy measurements of the solution derivative function at chosen evaluation points.

$$\begin{cases} \dot{x}(t) = f_\theta(t, x(t)), & t \in [a, b], \\ x(a) = x_0, \end{cases}$$

Assuming Gaussian error model, posterior follows in the standard form,

$$\dot{x}(\mathbf{t}) \mid f(\mathbf{s}, \tilde{\mathbf{x}}) \sim \mathcal{N}\left(\dot{\mu}_N(\mathbf{t}), \dot{\Gamma}_N(\mathbf{t})\right)$$

$$x(\mathbf{t}) \mid f(\mathbf{s}, \tilde{\mathbf{x}}) \sim \mathcal{N}\left(\mu_N(\mathbf{t}), \Gamma_N(\mathbf{t})\right)$$

where the means for the standard and integrated GP posteriors follow as,

$$\dot{\mu}_N(\mathbf{t}) = \dot{l}(\mathbf{t}) + RR(\mathbf{t}, \mathbf{s})\left(\Lambda_N + RR(\mathbf{s}, \mathbf{s})\right)^{-1}\left(f(\mathbf{s}, \tilde{\mathbf{x}}) - l(\mathbf{s})\right)$$

$$\mu_N(\mathbf{t}) = l(\mathbf{t}) + QR(\mathbf{t}, \mathbf{s})\left(\Lambda_N + RR(\mathbf{s}, \mathbf{s})\right)^{-1}\left(f(\mathbf{s}, \tilde{\mathbf{x}}) - \dot{l}(\mathbf{s})\right)$$

and the covariances as,

$$\dot{\Gamma}_N(\mathbf{t}) = RR(\mathbf{t}, \mathbf{t}) - RR(\mathbf{t}, \mathbf{s})\left(\Lambda_N + RR(\mathbf{s}, \mathbf{s})\right)^{-1} RR(\mathbf{s}, \mathbf{t})$$

$$\Gamma_N(\mathbf{t}) = QQ(\mathbf{t}, \mathbf{t}) - QR(\mathbf{t}, \mathbf{s})\left(\Lambda_N + RR(\mathbf{s}, \mathbf{s})\right)^{-1} RQ(\mathbf{s}, \mathbf{t})$$

Using the notation:   $AB(t_1, t_2) = \frac{1}{\alpha} \int_{-\infty}^{\infty} A(t_1, z) B(t_2, z) \mathrm{d}z$

$$\begin{cases} \dot{x}(t) = f_\theta(t, x(t)), & t \in [a, b], \\ x(a) = x_0, \end{cases}$$

Solving ordinary differential equations is inherently sequential, and we propose the following scheme:

1. Conditional on initial states and model parameters, obtain the first *derivative* observation by evaluating the vector field at the initial state value.

2. Predict the next *state* value based on the derivative observation.

3. Re-estimate the solution in the state space and use the state prediction to obtain the second *derivative* observation.

In practice, every time we iterate, the number of data points increases by one.  At every iteration we must re-evaluate the predictive GP posterior distributions, which becomes very slow using the standard approach because of required matrix inversion.

An interesting difference with traditional methods is that steps are made not according to a deterministic rule, but rather from a predictive distribution.

In practice, every time we iterate, the number of data points increases by one. At every iteration we must re-evaluate the predictive GP posterior distributions, which becomes very slow using the standard approach because of required matrix inversion.

An interesting difference with traditional methods is that steps are made not according to a deterministic rule, but rather from a predictive distribution.

**To whet your appetite...** in our paper, we

- provide a recursive formulation which avoids matrix inversion around an order of magnitude faster.

- show that our probabilistic solver is a consistent estimator of the true solution for the initial value problem
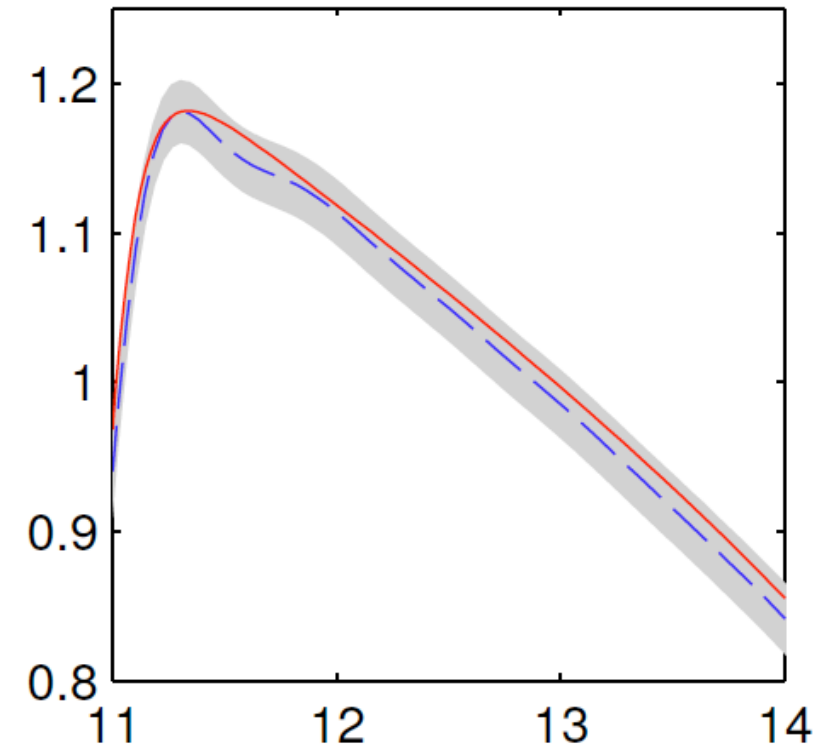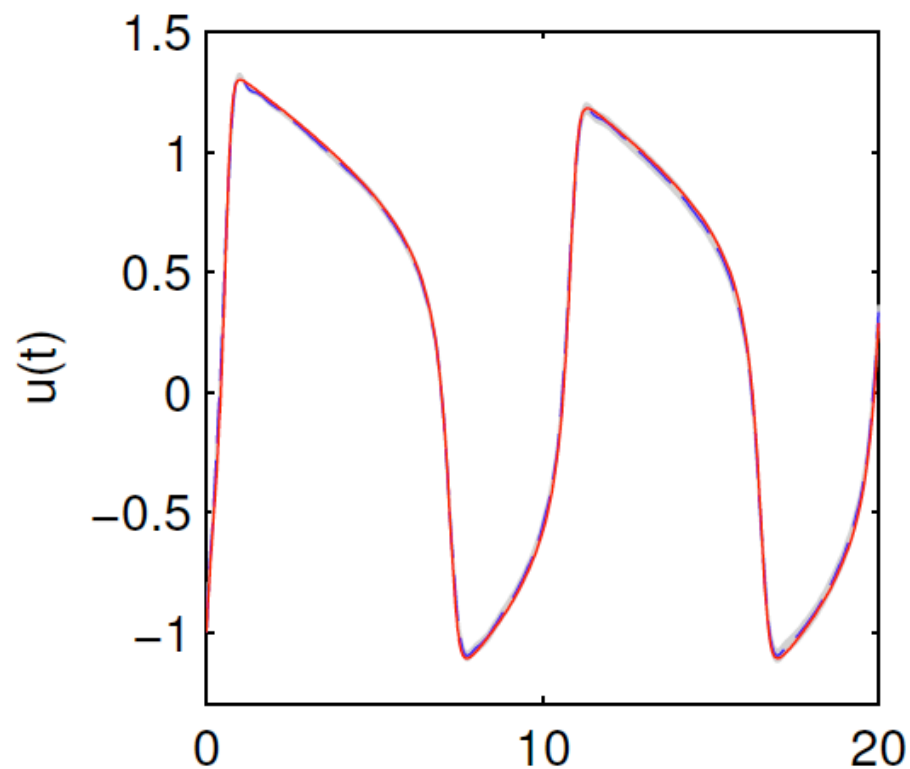
## An Example

FitzHugh-Nagumo system describes the dynamics of action potential generation within a nerve axon.

$$\dot{u}(t) \equiv f^{(U)}(t, u, v) = \theta_3 \left[ u(t) - v(t)^3 + v(t) \right], \quad u(a) = u_0,$$
$$\dot{v}(t) \equiv f^{(V)}(t, u, v) = \frac{1}{\theta_3} \left[ u(t) - \theta_1 + \theta_2 v(t) \right], \quad v(a) = v_0,$$

Interesting test model - oscillatory dynamics, sharp changes in derivatives and multimodality in posterior distribution when doing Bayesian inference.

# An Example



The numerical (solid line) and mean probabilistic (dashed line) solutions for the FitzHugh-Nagumo system for a particular set of parameters. The grey bands show 2 standard deviations around the probabilistic solution mean.

The numerical solution was obtained via the ode45 function in MATLAB with error tolerances of 10^-3 . The probabilistic solution was constructed using 300 solver knots and auxiliary parameters were sampled using MCMC.

## Fully Probabilistic Inference

In this case we simulate observations from the numerical solution of the FitzHugh-Nagumo model with model parameters (0.2, 0.2, 3) and initial conditions (-1, 1). We add some Gaussian noise and employ Bayesian inference to obtain posterior distributions over the ODE model parameters and initial conditions.

# Model parameters and initial conditions
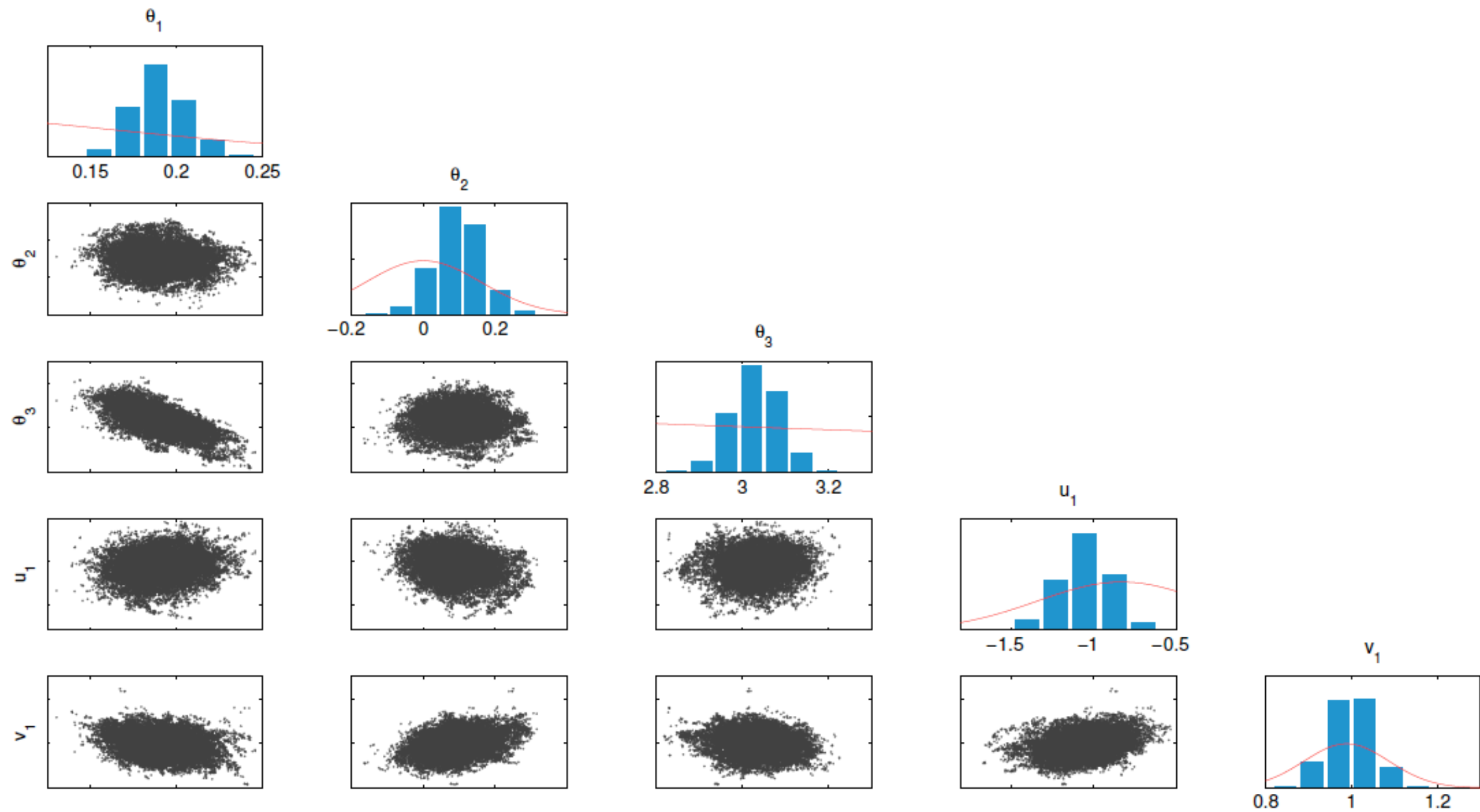
# Auxiliary parameters - GP prior-precision and lengthscale.
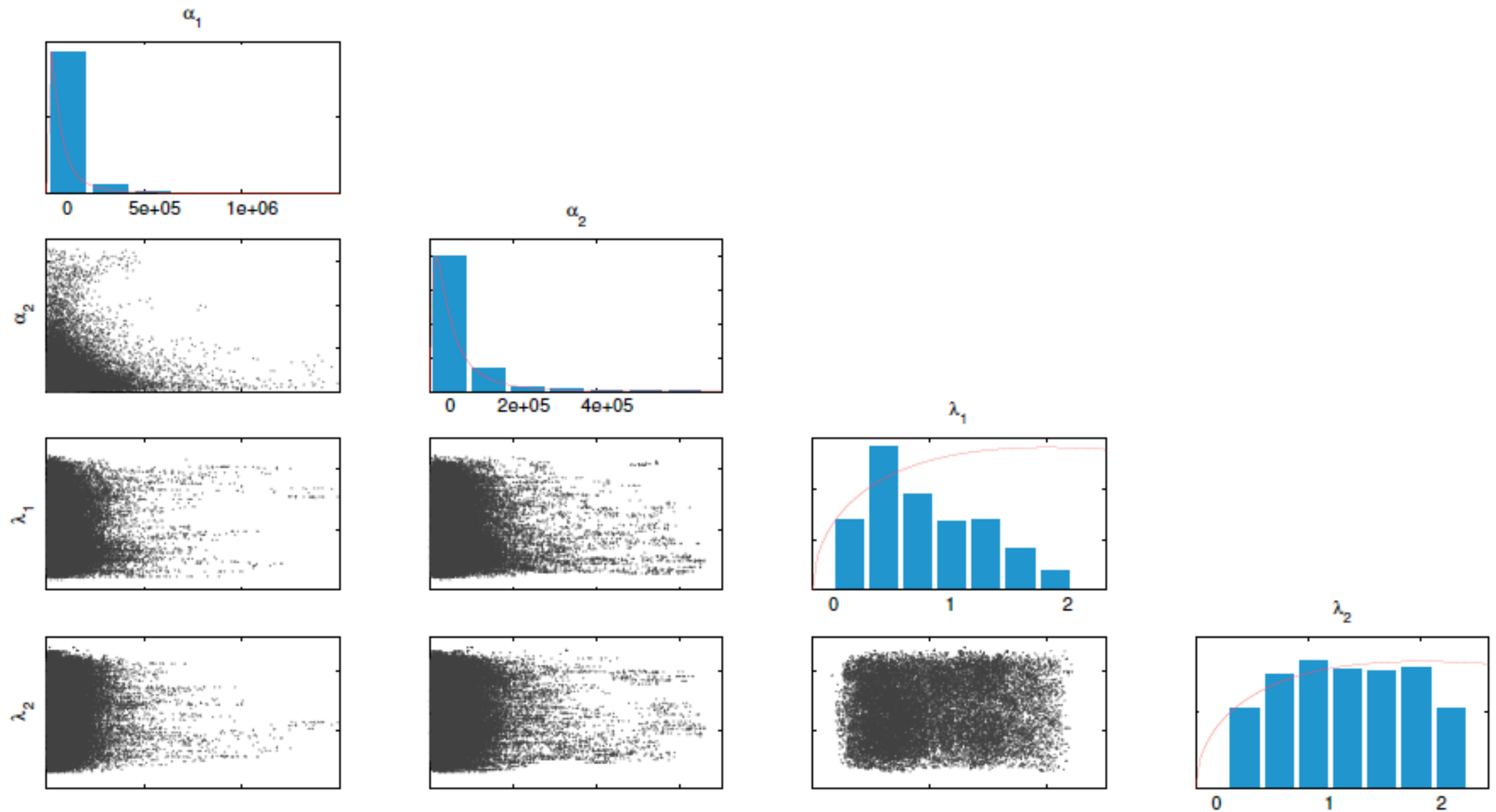
## Extensions

It is relatively straightforward to extend this work to delay differential equations, where we must infer an *initial function* instead of initial values.

The use of sparse covariance functions could speed up inference, as could maximum likelihood estimation of auxiliary hyperparameters.

Investigation of the impact of alternative kernels/covariance functions on the accuracy of the solver.

Adaptive selection of time steps based on predicted variance and required accuracy of the solution.

# CONCLUSIONS

- Rather than using a single deterministic solution, a probabilistic approach provides a family of such solutions with associated probability distribution

- This functional uncertainty in the solution (i.e. solver uncertainty) may be propagated through the rest of the inference framework

- The sequential nature of the GP estimator leads to a recursive implementation that avoids expensive matrix inversions

# ACKNOWLEDGEMENTS

- NIPS Travel Award

- EPSRC "2020 Science" Research Fellowship

- PASCAL2