Bayesian Uncertainty Quantification for Differential Equations

Mark Girolami

Department of Statistics
University of Warwick

MLSS, May 2014

Enabling Quantification *of*

EQUIP

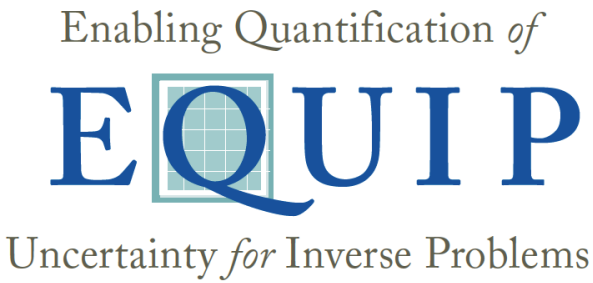
Uncertainty *for* Inverse Problems

Joint Work

http://web.warwick.ac.uk/PODES/

Programme of Research

- Andrew Stuart - PI - University of Warwick
- Gareth Roberts - University of Warwick
- Mike Christie - Petroleum Institute, Heriot Watt University
- Mark Girolami - University College London



`http://warwick.ac.uk/equip/`

Outline

- Motivation for Uncertainty Quantification

Outline

- Motivation for Uncertainty Quantification
- Probabilistic model of infinite and finite dimensional mismatch

Outline

- Motivation for Uncertainty Quantification
- Probabilistic model of infinite and finite dimensional mismatch
- Full Bayesian posterior measure for model uncertainty

Outline

- Motivation for Uncertainty Quantification
- Probabilistic model of infinite and finite dimensional mismatch
- Full Bayesian posterior measure for model uncertainty
- Probabilistic integration and construction of posterior measure

Outline

- Motivation for Uncertainty Quantification
- Probabilistic model of infinite and finite dimensional mismatch
- Full Bayesian posterior measure for model uncertainty
- Probabilistic integration and construction of posterior measure
- Posterior sampling of differential equation solutions

Outline

- Motivation for Uncertainty Quantification
- Probabilistic model of infinite and finite dimensional mismatch
- Full Bayesian posterior measure for model uncertainty
- Probabilistic integration and construction of posterior measure
- Posterior sampling of differential equation solutions
- Chaotic ODE and PDE systems

Outline

- Motivation for Uncertainty Quantification
- Probabilistic model of infinite and finite dimensional mismatch
- Full Bayesian posterior measure for model uncertainty
- Probabilistic integration and construction of posterior measure
- Posterior sampling of differential equation solutions
- Chaotic ODE and PDE systems
- Illustration on protein dynamics

Outline

- Motivation for Uncertainty Quantification
- Probabilistic model of infinite and finite dimensional mismatch
- Full Bayesian posterior measure for model uncertainty
- Probabilistic integration and construction of posterior measure
- Posterior sampling of differential equation solutions
- Chaotic ODE and PDE systems
- Illustration on protein dynamics
- Conclusions and Discussion

Rev Thomas Bayes - Probability Inversion

System Modelling with Differential Equations

- In systems of differential equations, the derivatives with respect to spatial variables, $x \in \mathcal{D} \subset \mathbb{R}^d$, and temporal variables, $t \in [a, b] \subset \mathbb{R}^+$, are related to the implicitly defined states, $\mathrm{u}(x, t) \in \mathbb{R}^P$, which are hence often analytically intractable.

System Modelling with Differential Equations

- In systems of differential equations, the derivatives with respect to spatial variables, $x \in \mathcal{D} \subset \mathbb{R}^d$, and temporal variables, $t \in [a, b] \subset \mathbb{R}^+$, are related to the implicitly defined states, $\mathrm{u}(x, t) \in \mathbb{R}^P$, which are hence often analytically intractable.
- The Initial Value Problem (IVP) models the system states with fixed initial condition $\mathrm{u}^*(a)$, evolving according to the ODE as follows,

$$\left\{ \begin{array}{ll} \mathrm{u}_t(t) = f\big(t, \mathrm{u}(t), \theta\big), & t \in [a, b], \\ \mathrm{u}(a) = \mathrm{u}^*(a). & \end{array} \right\} \tag{1}$$

  The existence of a solution is guaranteed under mild conditions

System Modelling with Differential Equations

- The Mixed Boundary Value Problem (MBVP) may constrain different states at different time points. Typically these constraints are imposed at the ends of the time domain giving the general form for two state mixed boundary value problems,

$$\begin{cases} \left(\mathrm{u}_t(t), \mathrm{v}_t(t)\right) = f\left(t, \left(\mathrm{u}(t), \mathrm{v}(t)\right), \theta\right), & t \in [a, b], \\ g\left(\mathrm{v}(a), \mathrm{u}(b)\right) = 0, \end{cases} \tag{2}$$

  which can be straightforwardly generalised to higher dimensions and extrapolated beyond the final time point b. Whereas a unique IVP solution exists under relatively mild conditions, imposing mixed boundary constraints can result in multiple solutions

System Modelling with Differential Equations

- PDEs are slightly more complex, as an illustrative example the parabolic diffusion equation, modelling the heat diffusion over time along a single spatial dimension by,

$$\begin{cases} u_t(x,t) &= \kappa u_{xx}(x,t), & t \in [0, 0.25],\ x \in [0,1] \\ u(x,t) &= \sin(x\pi), & t = 0,\ x \in [0,1], \\ u(x,t) &= 0, & t \in [0, 0.25],\ x = 0, 1. \end{cases} \tag{3}$$

Quantifying Uncertainty in Differential Equation Models

- Consider data, $y(\mathbf{t})$, observed at discrete time points $\mathbf{t} = [t_1, t_2, \ldots, t_T]$ and a set of model parameters θ.

Quantifying Uncertainty in Differential Equation Models

- Consider data, $y(\mathbf{t})$, observed at discrete time points $\mathbf{t} = [t_1, t_2, \ldots, t_T]$ and a set of model parameters θ.
- Using the exact solution of a mathematical model represented by $\mathrm{u}^*(\mathbf{t}, \theta)$, a simplified observation model based on some measurement error structure $\epsilon(\mathbf{t})$ is,

$$y(\mathbf{t}) = \mathrm{u}^*(\mathbf{t}, \theta) + \epsilon(\mathbf{t}). \tag{4}$$

Quantifying Uncertainty in Differential Equation Models

- Consider data, $y(\mathbf{t})$, observed at discrete time points $\mathbf{t} = [t_1, t_2, \ldots, t_T]$ and a set of model parameters θ.
- Using the exact solution of a mathematical model represented by $\mathrm{u}^*(\mathbf{t}, \theta)$, a simplified observation model based on some measurement error structure $\epsilon(\mathbf{t})$ is,

$$y(\mathbf{t}) = \mathrm{u}^*(\mathbf{t}, \theta) + \epsilon(\mathbf{t}). \tag{4}$$

- Following Kennedy and O'Hagan by defining $\delta(\mathbf{t})$ as a random function drawn from a Gaussian Process (GP), the observational model becomes,

$$y(\mathbf{t}) = \mathrm{u}^*(\mathbf{t}, \theta) + \delta(\mathbf{t}) + \epsilon(\mathbf{t}). \tag{5}$$

Quantifying Uncertainty in Differential Equation Models

- Consider data, $y(\mathbf{t})$, observed at discrete time points $\mathbf{t} = [t_1, t_2, \ldots, t_T]$ and a set of model parameters θ.
- Using the exact solution of a mathematical model represented by $\mathrm{u}^*(\mathbf{t}, \theta)$, a simplified observation model based on some measurement error structure $\epsilon(\mathbf{t})$ is,

$$y(\mathbf{t}) = \mathrm{u}^*(\mathbf{t}, \theta) + \epsilon(\mathbf{t}). \tag{4}$$

- Following Kennedy and O'Hagan by defining $\delta(\mathbf{t})$ as a random function drawn from a Gaussian Process (GP), the observational model becomes,

$$y(\mathbf{t}) = \mathrm{u}^*(\mathbf{t}, \theta) + \delta(\mathbf{t}) + \epsilon(\mathbf{t}). \tag{5}$$

- Due to the lack of an analytical solution for most nonlinear differential equations, the likelihood $p\big(y(\mathbf{t}) \mid \mathrm{u}^*(\mathbf{t}, \theta), \theta\big)$ cannot be obtained in closed form.

Quantifying Uncertainty in Differential Equation Models

- This issue is dealt with throughout the statistics literature by replacing the exact likelihood with a surrogate, $p\left(y(\mathbf{t}) \mid \hat{u}^N(\mathbf{t}, \theta), \theta\right)$, based on an N-dimensional approximate solution, $\hat{u}^N(\mathbf{t}, \theta)$, obtained using numerical integration methods

Quantifying Uncertainty in Differential Equation Models

- This issue is dealt with throughout the statistics literature by replacing the exact likelihood with a surrogate, $p\left(y(\mathbf{t}) \mid \hat{u}^N(\mathbf{t}, \theta), \theta\right)$, based on an N-dimensional approximate solution, $\hat{u}^N(\mathbf{t}, \theta)$, obtained using numerical integration methods
- The seemingly innocuous assumption of negligible numerical integration error can subsequently lead to serious statistical bias and misleading inferences for certain classes of differential equation models

Quantifying Uncertainty in Differential Equation Models

- This issue is dealt with throughout the statistics literature by replacing the exact likelihood with a surrogate, $p\big(y(\mathbf{t}) \mid \hat{\mathrm{u}}^N(\mathbf{t}, \theta), \theta\big)$, based on an N-dimensional approximate solution, $\hat{\mathrm{u}}^N(\mathbf{t}, \theta)$, obtained using numerical integration methods
- The seemingly innocuous assumption of negligible numerical integration error can subsequently lead to serious statistical bias and misleading inferences for certain classes of differential equation models
- We may represent this additional uncertainty using the term $\zeta(\mathbf{t}, \theta) = \mathrm{u}^*(\mathbf{t}, \theta) - \hat{\mathrm{u}}^N(\mathbf{t}, \theta)$, such that,

$$y(\mathbf{t}) = \hat{\mathrm{u}}^N(\mathbf{t}, \theta) + \zeta(\mathbf{t}, \theta) + \delta(\mathbf{t}) + \epsilon(\mathbf{t}). \tag{6}$$

Quantifying Uncertainty in Differential Equation Models

- This issue is dealt with throughout the statistics literature by replacing the exact likelihood with a surrogate, $p\big(y(\mathbf{t}) \mid \hat{\mathrm{u}}^N(\mathbf{t}, \theta), \theta\big)$, based on an N-dimensional approximate solution, $\hat{\mathrm{u}}^N(\mathbf{t}, \theta)$, obtained using numerical integration methods
- The seemingly innocuous assumption of negligible numerical integration error can subsequently lead to serious statistical bias and misleading inferences for certain classes of differential equation models
- We may represent this additional uncertainty using the term $\zeta(\mathbf{t}, \theta) = \mathrm{u}^*(\mathbf{t}, \theta) - \hat{\mathrm{u}}^N(\mathbf{t}, \theta)$, such that,

$$y(\mathbf{t}) = \hat{\mathrm{u}}^N(\mathbf{t}, \theta) + \zeta(\mathbf{t}, \theta) + \delta(\mathbf{t}) + \epsilon(\mathbf{t}). \tag{6}$$

- We "open the black box" by explicitly modelling the solution and associated discretisation uncertainty, $\hat{\mathrm{u}}^N(\mathbf{t}, \theta) + \zeta(\mathbf{t}, \theta)$. This allows the Kennedy and O'Hagan framework to be further enriched by incorporating detailed knowledge of the mathematical model being employed

Quantifying Uncertainty in Differential Equation Models

- We model uncertainty in a finite dimensional representation of the infinite dimensional solution through a probability statement on a space of suitably smooth functions.

Quantifying Uncertainty in Differential Equation Models

- We model uncertainty in a finite dimensional representation of the infinite dimensional solution through a probability statement on a space of suitably smooth functions.
- Restricting to Hilbert spaces for modelling our knowledge of $\mathrm{u}^*(\mathbf{t}, \theta)$, we define a Gaussian prior measure on the function space. We then directly model our knowledge about the solution via the stochastic process $\mathrm{u}(\mathbf{t}, \theta)$, thus replacing (6) with $y(\mathbf{t}) = \mathrm{u}(\mathbf{t}, \theta) + \delta(\mathbf{t}) + \epsilon(\mathbf{t})$.

Quantifying Uncertainty in Differential Equation Models

- We model uncertainty in a finite dimensional representation of the infinite dimensional solution through a probability statement on a space of suitably smooth functions.
- Restricting to Hilbert spaces for modelling our knowledge of $\mathrm{u}^*(\mathbf{t}, \theta)$, we define a Gaussian prior measure on the function space. We then directly model our knowledge about the solution via the stochastic process $\mathrm{u}(\mathbf{t}, \theta)$, thus replacing (6) with $y(\mathbf{t}) = \mathrm{u}(\mathbf{t}, \theta) + \delta(\mathbf{t}) + \epsilon(\mathbf{t})$.
- Focus attention on the joint posterior measure over differential equation model states, parameters, and associated auxiliary parameters, Ψ, of our probabilistic model of uncertainty,

$$p\big(\theta, \mathrm{u}(\mathbf{t}, \theta), \Psi \mid y(\mathbf{t}), N\big) \propto \underbrace{p\big(y(\mathbf{t}) \mid \mathrm{u}(\mathbf{t}, \theta), \theta\big)}_{\text{Likelihood}} \times \underbrace{p\big(\mathrm{u}(\mathbf{t}, \theta) \mid \theta, \Psi, N\big)}_{\text{Model}} \times \underbrace{p\big(\theta, \Psi\big)}_{\text{Prior}}.$$

Gaussian Prior Measure for States and Derivatives

- Denote by R_λ a deterministic, square integrable kernel function with length-scale $\lambda \in (0, \infty)$

Gaussian Prior Measure for States and Derivatives

- Denote by R_λ a deterministic, square integrable kernel function with length-scale $\lambda \in (0, \infty)$
- Its integrated version is $Q_\lambda(t_1, t_2) = \int_a^{t_1} R_\lambda(s, t_2) \mathrm{d}s$.

Gaussian Prior Measure for States and Derivatives

- Denote by R_λ a deterministic, square integrable kernel function with length-scale $\lambda \in (0, \infty)$
- Its integrated version is $Q_\lambda(t_1, t_2) = \int_a^{t_1} R_\lambda(s, t_2) \mathrm{d}s$.
- Model for the derivative has covariance operator $\mathrm{cov}(u_t(t_1), u_t(t_2)) = \alpha^{-1} \int_{\mathbb{R}} R_\lambda(t_1, s) R_\lambda(t_2, s) \mathrm{d}s := RR(t_1, t_2)$, where α is a prior precision parameter.

Gaussian Prior Measure for States and Derivatives

- Denote by R_λ a deterministic, square integrable kernel function with length-scale $\lambda \in (0, \infty)$
- Its integrated version is $\mathrm{Q}_\lambda(t_1, t_2) = \int_a^{t_1} \mathrm{R}_\lambda(s, t_2)\mathrm{d}s$.
- Model for the derivative has covariance operator $\mathrm{cov}(\mathrm{u}_t(t_1), \mathrm{u}_t(t_2)) = \alpha^{-1} \int_{\mathbb{R}} \mathrm{R}_\lambda(t_1, s)\mathrm{R}_\lambda(t_2, s)\mathrm{d}s := \mathrm{RR}(t_1, t_2)$, where α is a prior precision parameter.
- Covariance on the state is its integrated version, $\mathrm{cov}(\mathrm{u}(t_1), \mathrm{u}(t_2)) = \alpha^{-1} \int_{\mathbb{R}} \mathrm{Q}_\lambda(t_1, s)\mathrm{Q}_\lambda(t_2, s)\mathrm{d}s := \mathrm{QQ}(t_1, t_2)$

Gaussian Prior Measure for States and Derivatives

- Denote by R_λ a deterministic, square integrable kernel function with length-scale $\lambda \in (0, \infty)$
- Its integrated version is $\mathrm{Q}_\lambda(t_1, t_2) = \int_a^{t_1} \mathrm{R}_\lambda(s, t_2)\mathrm{d}s$.
- Model for the derivative has covariance operator $\mathrm{cov}(\mathrm{u}_t(t_1), \mathrm{u}_t(t_2)) = \alpha^{-1} \int_{\mathbb{R}} \mathrm{R}_\lambda(t_1, s)\mathrm{R}_\lambda(t_2, s)\mathrm{d}s := \mathrm{RR}(t_1, t_2)$, where α is a prior precision parameter.
- Covariance on the state is its integrated version, $\mathrm{cov}(\mathrm{u}(t_1), \mathrm{u}(t_2)) = \alpha^{-1} \int_{\mathbb{R}} \mathrm{Q}_\lambda(t_1, s)\mathrm{Q}_\lambda(t_2, s)\mathrm{d}s := \mathrm{QQ}(t_1, t_2)$
- The cross covariance terms are defined in a similar manner and denoted as $\mathrm{RQ}(t_1, t_2)$ and $\mathrm{QR}(t_1, t_2)$ respectively. We assume a joint Gaussian prior measure on the state and its derivative,

Gaussian Prior Measure for States and Derivatives

- Denote by R_λ a deterministic, square integrable kernel function with length-scale $\lambda \in (0, \infty)$
- Its integrated version is $\mathrm{Q}_\lambda(t_1, t_2) = \int_a^{t_1} \mathrm{R}_\lambda(s, t_2)\mathrm{d}s$.
- Model for the derivative has covariance operator $\mathrm{cov}(\mathrm{u}_t(t_1), \mathrm{u}_t(t_2)) = \alpha^{-1} \int_{\mathbb{R}} \mathrm{R}_\lambda(t_1, s)\mathrm{R}_\lambda(t_2, s)\mathrm{d}s := \mathrm{RR}(t_1, t_2)$, where α is a prior precision parameter.
- Covariance on the state is its integrated version, $\mathrm{cov}(\mathrm{u}(t_1), \mathrm{u}(t_2)) = \alpha^{-1} \int_{\mathbb{R}} \mathrm{Q}_\lambda(t_1, s)\mathrm{Q}_\lambda(t_2, s)\mathrm{d}s := \mathrm{QQ}(t_1, t_2)$
- The cross covariance terms are defined in a similar manner and denoted as $\mathrm{RQ}(t_1, t_2)$ and $\mathrm{QR}(t_1, t_2)$ respectively. We assume a joint Gaussian prior measure on the state and its derivative,

$$\begin{bmatrix} \mathrm{u}_t \\ \mathrm{u} \end{bmatrix} \sim \mathcal{GP}\left(\begin{bmatrix} \mathrm{m}_t \\ \mathrm{m} \end{bmatrix}, \begin{bmatrix} \mathrm{RR}(t_1, t_2) & \mathrm{QR}^\dagger(t_1, t_2) \\ \mathrm{QR}(t_1, t_2) & \mathrm{QQ}(t_1, t_2) \end{bmatrix} \right). \tag{7}$$

Gaussian Prior Measure for States and Derivatives

- One modelling choice for incorporating the spatial component of the PDE into the covariance is to adopt a product structure. The spatial covariance structures are defined similarly to the temporal form in the ODE example as,

$$
\begin{aligned}
\mathrm{cov}(\mathrm{u}_{xx}(x_1,t_1),\mathrm{u}_{xx}(x_2,t_2)) &= \beta^{-1}\int_{\mathbb{R}} \mathrm{R}_\mu(x_1,z)\mathrm{R}_\mu(x_2,z)\mathrm{d}z\ \mathrm{QQ}(t_1,t_2)\\
&:= \mathrm{RR}(x_1,x_2)\mathrm{QQ}(t_1,t_2),\\
\mathrm{cov}(\mathrm{u}(x_1,t_1),\mathrm{u}(x_2,t_2)) &= \beta^{-1}\int_{\mathbb{R}} \mathrm{S}_\mu(x_1,z)\mathrm{S}_\mu(x_2,z)\mathrm{d}z\ \mathrm{QQ}(t_1,t_2)\\
&:= \mathrm{SS}(x_1,x_2)\mathrm{QQ}(t_1,t_2),
\end{aligned}
$$

where β is a spatial prior precision parameter.

Gaussian Prior Measure for States and Derivatives

- One modelling choice for incorporating the spatial component of the PDE into the covariance is to adopt a product structure. The spatial covariance structures are defined similarly to the temporal form in the ODE example as,

$$\begin{aligned}
\mathrm{cov}(\mathrm{u}_{xx}(x_1,t_1),\mathrm{u}_{xx}(x_2,t_2)) &= \beta^{-1}\int_{\mathbb{R}} \mathrm{R}_\mu(x_1,z)\mathrm{R}_\mu(x_2,z)\mathrm{d}z\ \mathrm{QQ}(t_1,t_2)\\
&:= \mathrm{RR}(x_1,x_2)\mathrm{QQ}(t_1,t_2),\\
\mathrm{cov}(\mathrm{u}(x_1,t_1),\mathrm{u}(x_2,t_2)) &= \beta^{-1}\int_{\mathbb{R}} \mathrm{S}_\mu(x_1,z)\mathrm{S}_\mu(x_2,z)\mathrm{d}z\ \mathrm{QQ}(t_1,t_2)\\
&:= \mathrm{SS}(x_1,x_2)\mathrm{QQ}(t_1,t_2),
\end{aligned}$$

where β is a spatial prior precision parameter.

The prior construction follows by defining a product structure for space and time,

$$\mathcal{GP}\left(\begin{bmatrix}\mathrm{m}_{xx}\\ \mathrm{m}_t\\ \mathrm{m}\end{bmatrix}, \begin{bmatrix}\mathrm{RR}(x_1,x_2)\mathrm{QQ}(t_1,t_2) & \mathrm{SR}^\dagger(x_1,x_2)\mathrm{QR}(t_1,t_2) & \mathrm{SR}^\dagger(x_1,x_2)\mathrm{QQ}(t_1,t_2)\\ \mathrm{SR}(x_1,x_2)\mathrm{QR}^\dagger(t_1,t_2) & \mathrm{SS}(x_1,x_2)\mathrm{RR}(t_1,t_2) & \mathrm{SS}(x_1,x_2)\mathrm{QR}^\dagger(t_1,t_2)\\ \mathrm{SR}(x_1,x_2)\mathrm{QQ}(t_1,t_2) & \mathrm{SS}(x_1,x_2)\mathrm{QR}(t_1,t_2) & \mathrm{SS}(x_1,x_2)\mathrm{QQ}(t_1,t_2)\end{bmatrix}\right)$$

Probabilistic Solution as Latent Function Estimation

- Consider a discretisation grid of N time points $\mathbf{s} := [s_1, \ldots, s_N]$ on the interval $[a, b]$ such that $a = s_1 \leq \cdots \leq s_N = b$.

Probabilistic Solution as Latent Function Estimation

- Consider a discretisation grid of N time points $\mathbf{s} := [s_1, \ldots, s_N]$ on the interval $[a, b]$ such that $a = s_1 \leq \cdots \leq s_N = b$.
- Begin by fixing the known initial value, $\mathrm{u}(s_1) := \mathrm{u}^*(a)$, and computing the exact derivative $\mathrm{f}_1 := f(s_1, \mathrm{u}(s_1), \theta)$ at s_1, via the deterministic ODE.

Probabilistic Solution as Latent Function Estimation

- Consider a discretisation grid of N time points $\mathbf{s} := [s_1, \ldots, s_N]$ on the interval $[a, b]$ such that $a = s_1 \leq \cdots \leq s_N = b$.
- Begin by fixing the known initial value, $\mathrm{u}(s_1) := \mathrm{u}^*(a)$, and computing the exact derivative $\mathrm{f}_1 := f(s_1, \mathrm{u}(s_1), \theta)$ at s_1, via the deterministic ODE.
- Update joint GP prior given the computed exact derivative f_1, obtaining the conditional predictive distribution for the state at the subsequent grid location s_2,

$$p(\mathrm{u}(s_2) \mid \mathrm{f}_1, \Psi) = \mathcal{N}\big(\mathrm{u}(s_2) \mid \mathrm{m}(s_2), \mathrm{C}(s_2, s_2)\big), \tag{8}$$

Probabilistic Solution as Latent Function Estimation

- Consider a discretisation grid of N time points $\mathbf{s} := [s_1, \ldots, s_N]$ on the interval $[a, b]$ such that $a = s_1 \leq \cdots \leq s_N = b$.
- Begin by fixing the known initial value, $\mathrm{u}(s_1) := \mathrm{u}^*(a)$, and computing the exact derivative $\mathrm{f}_1 := f(s_1, \mathrm{u}(s_1), \theta)$ at s_1, via the deterministic ODE.
- Update joint GP prior given the computed exact derivative f_1, obtaining the conditional predictive distribution for the state at the subsequent grid location s_2,

$$p(\mathrm{u}(s_2) \mid \mathrm{f}_1, \Psi) = \mathcal{N}\big(\mathrm{u}(s_2) \mid \mathrm{m}(s_2), \mathrm{C}(s_2, s_2)\big), \tag{8}$$

with,

$$\begin{aligned} \mathrm{m}(s_2) &= \mathrm{QR}(s_2, s_1)\mathrm{RR}(s_1, s_1)^{-1}\mathrm{f}_1, \\ \mathrm{C}(s_2, s_2) &= \mathrm{QQ}(s_2, s_2) - \mathrm{QR}(s_2, s_1)\mathrm{RR}(s_1, s_1)^{-1}\mathrm{QR}(s_2, s_1). \end{aligned}$$

Probabilistic Solution as Latent Function Estimation

- Now sample a realisation, $\mathrm{u}(s_2)$, of the predictive process, and again link our prior to the deterministic ODE model by computing $\mathrm{f}_2 := f(s_2, \mathrm{u}(s_2), \theta)$.

Probabilistic Solution as Latent Function Estimation

- Now sample a realisation, $\mathrm{u}(s_2)$, of the predictive process, and again link our prior to the deterministic ODE model by computing $\mathrm{f}_2 := f(s_2, \mathrm{u}(s_2), \theta)$.
- In contrast to the first time point s_1, we can no longer guarantee that the realisation at the second time point, $\mathrm{u}(s_2)$, and its derivative, $\mathrm{u}_t(s_2)$, exactly satisfy the ODE model, i.e. that $\mathrm{u}_t(s_2) = \mathrm{f}_2$.

Probabilistic Solution as Latent Function Estimation

- Now sample a realisation, $\mathrm{u}(s_2)$, of the predictive process, and again link our prior to the deterministic ODE model by computing $\mathrm{f}_2 := f(s_2, \mathrm{u}(s_2), \theta)$.
- In contrast to the first time point s_1, we can no longer guarantee that the realisation at the second time point, $\mathrm{u}(s_2)$, and its derivative, $\mathrm{u}_t(s_2)$, exactly satisfy the ODE model, i.e. that $\mathrm{u}_t(s_2) = \mathrm{f}_2$.
- Therefore, at time s_2 we explicitly model the mismatch between the ODE evaluation f_2 and the process derivative, $\mathrm{u}_t(s_2)$, as,

$$p\big(\mathrm{u}_t(s_2) \mid \mathrm{f}_2, \Psi\big) = \mathcal{N}\big(\mathrm{u}_t(s_2) \mid \mathrm{f}_2, \mathrm{C}_t(s_2, s_2)\big),$$

Probabilistic Solution as Latent Function Estimation

- Now sample a realisation, $\mathrm{u}(s_2)$, of the predictive process, and again link our prior to the deterministic ODE model by computing $\mathrm{f}_2 := f(s_2, \mathrm{u}(s_2), \theta)$.
- In contrast to the first time point s_1, we can no longer guarantee that the realisation at the second time point, $\mathrm{u}(s_2)$, and its derivative, $\mathrm{u}_t(s_2)$, exactly satisfy the ODE model, i.e. that $\mathrm{u}_t(s_2) = \mathrm{f}_2$.
- Therefore, at time s_2 we explicitly model the mismatch between the ODE evaluation f_2 and the process derivative, $\mathrm{u}_t(s_2)$, as,

$$p\big(\mathrm{u}_t(s_2) \mid \mathrm{f}_2, \Psi\big) = \mathcal{N}\big(\mathrm{u}_t(s_2) \mid \mathrm{f}_2, \mathrm{C}_t(s_2, s_2)\big),$$

- where the magnitude of the mismatch may be described by the variance, $\mathrm{C}_t(s_2, s_2)$, of the predictive posterior over the derivative given by,

$$\mathrm{C}_t(s_2, s_2) = \mathrm{RR}(s_2, s_2) - \mathrm{RR}(s_2, s_1)\mathrm{RR}(s_1, s_1)^{-1}\mathrm{RR}(s_1, s_2).$$

Probabilistic Solution as Latent Function Estimation

- Condition on the augmented vector $\mathbf{f}_{1:2} := [\mathrm{f}_1, f(s_2, \mathrm{u}(s_2), \theta)]$. We therefore define the matrix $\Lambda_{2\times 2} := \mathrm{diag}\{0, \mathrm{C}_t(s_2, s_2)\}$ to describe the mismatch between the process derivative and the ODE function evaluations at s_1 and s_2.

Probabilistic Solution as Latent Function Estimation

- Condition on the augmented vector $\mathbf{f}_{1:2} := [\mathrm{f}_1, f(s_2, \mathrm{u}(s_2), \theta)]$. We therefore define the matrix $\Lambda_{2\times 2} := \mathrm{diag}\{0, \mathrm{C}_t(s_2, s_2)\}$ to describe the mismatch between the process derivative and the ODE function evaluations at s_1 and s_2. The new predictive posterior,

$$p(\mathrm{u}(s_3) \mid \mathbf{f}_{1:2}, \Psi) = \mathcal{N}\big(\mathrm{u}(s_3) \mid \mathrm{m}(s_3), \mathrm{C}(s_3, s_3)\big),$$

  has mean and covariance,

Probabilistic Solution as Latent Function Estimation

- Condition on the augmented vector $\mathbf{f}_{1:2} := [\mathrm{f}_1, f(s_2, \mathrm{u}(s_2), \theta)]$. We therefore define the matrix $\Lambda_{2\times 2} := \mathrm{diag}\{0, \mathrm{C}_t(s_2, s_2)\}$ to describe the mismatch between the process derivative and the ODE function evaluations at s_1 and s_2. The new predictive posterior,

$$p(\mathrm{u}(s_3) \mid \mathbf{f}_{1:2}, \Psi) = \mathcal{N}\big(\mathrm{u}(s_3) \mid \mathrm{m}(s_3), \mathrm{C}(s_3, s_3)\big),$$

has mean and covariance,

$$\mathrm{m}(s_3) = \mathrm{QR}(s_3, \mathbf{s}_{1:2})\big(\mathrm{RR}(\mathbf{s}_{1:2}, \mathbf{s}_{1:2}) + \Lambda_{2\times 2}\big)^{-1}\mathbf{f}_{1:2},$$

$$\mathrm{C}(s_3, s_3) = \mathrm{QQ}(s_3, s_3) - \mathrm{QR}(s_3, \mathbf{s}_{1:2})\big(\mathrm{RR}(\mathbf{s}_{1:2}, \mathbf{s}_{1:2}) + \Lambda_{2\times 2}\big)^{-1}\mathrm{QR}(s_3, \mathbf{s}_{1:2})^\top.$$

Probabilistic Solution as Latent Function Estimation

- Condition on the augmented vector $\mathbf{f}_{1:2} := [\mathrm{f}_1, f(s_2, \mathrm{u}(s_2), \theta)]$. We therefore define the matrix $\Lambda_{2\times 2} := \mathrm{diag}\{0, \mathrm{C}_t(s_2, s_2)\}$ to describe the mismatch between the process derivative and the ODE function evaluations at s_1 and s_2. The new predictive posterior,

$$p(\mathrm{u}(s_3) \mid \mathbf{f}_{1:2}, \Psi) = \mathcal{N}\big(\mathrm{u}(s_3) \mid \mathrm{m}(s_3), \mathrm{C}(s_3, s_3)\big),$$

has mean and covariance,

$$\mathrm{m}(s_3) = \mathrm{QR}(s_3, \mathbf{s}_{1:2})\big(\mathrm{RR}(\mathbf{s}_{1:2}, \mathbf{s}_{1:2}) + \Lambda_{2\times 2}\big)^{-1}\mathbf{f}_{1:2},$$
$$\mathrm{C}(s_3, s_3) = \mathrm{QQ}(s_3, s_3) - \mathrm{QR}(s_3, \mathbf{s}_{1:2})\big(\mathrm{RR}(\mathbf{s}_{1:2}, \mathbf{s}_{1:2}) + \Lambda_{2\times 2}\big)^{-1}\mathrm{QR}(s_3, \mathbf{s}_{1:2})^\top.$$

As in the previous two steps, we sample the state realisation $\mathrm{u}(s_3)$ from the above predictive posterior distribution. We then apply the deterministic transformation f, to obtain $\mathrm{f}_3 := f(s_3, \mathrm{u}(s_3), \theta)$, whose mismatch with the realised derivative $\mathrm{u}_t(s_3)$ is again modelled as,

$$p\big(\mathrm{u}_t(s_3) \mid \mathrm{f}_3, \Psi\big) = \mathcal{N}\big(\mathrm{u}_t(s_3) \mid \mathrm{f}_3, \mathrm{C}_t(s_3, s_3)\big).$$

Overall Sampling Scheme

Algorithm 1 Sample from the joint posterior distribution of u and $\mathbf{f}_{1:N}$ for an ODE initial value problem given θ, Ψ, N

1: At time $s_1 := a$, initialise the derivative $\mathrm{f}_1 := f(s_1, \mathrm{u}(s_1), \theta)$ for initial state $\mathrm{u}(s_1) := \mathrm{u}^*(a)$, and define associated model-derivative mismatch, $\Lambda_{1\times 1} := 0$;

2: **for** $n = 1 : N - 1$ **do**

3: Define the predictive state mean and variance,

$$\mathrm{m}(s_{n+1}) = \mathrm{QR}(s_{n+1}, \mathbf{s}_{1:n})(\mathrm{RR}(\mathbf{s}_{1:n}, \mathbf{s}_{1:n}) + \Lambda_{n\times n})^{-1}\mathbf{f}_{1:n},$$
$$\mathrm{C}(s_{n+1}, s_{n+1}) = \mathrm{QQ}(s_{n+1}, s_{n+1}) - \mathrm{QR}(s_{n+1}, \mathbf{s}_{1:n})(\mathrm{RR}(\mathbf{s}_{1:n}, \mathbf{s}_{1:n}) + \Lambda_{n\times n})^{-1}\mathrm{QR}(s_{n+1}, \mathbf{s}_{1:n})^\top;$$

4: Sample step-ahead realisation $\mathrm{u}(s_{n+1})$ from the predictive distribution of the state,

$$p(\mathrm{u}(s_{n+1}) \mid \mathbf{f}_{1:n}, \Psi) = \mathcal{N}(\mathrm{u}(s_{n+1}) \mid \mathrm{m}(s_{n+1}), \mathrm{C}(s_{n+1}, s_{n+1}));$$

5: Evaluate the ODE model $\mathrm{f}_{n+1} := f(s_{n+1}, \mathrm{u}(s_{n+1}), \theta)$ for realisation $\mathrm{u}(s_{n+1})$ at the subsequent grid point, s_{n+1}, and augment the vector $\mathbf{f}_{1:n+1} := [\mathbf{f}_{1:n}, \mathrm{f}_{n+1}]$;

6: Define the predictive derivative variance,

$$\mathrm{C}_t(s_{n+1}, s_{n+1}) = \mathrm{RR}(s_{n+1}, s_{n+1}) - \mathrm{RR}(s_{n+1}, \mathbf{s}_{1:n})(\mathrm{RR}(\mathbf{s}_{1:n}, \mathbf{s}_{1:n}) + \Lambda_{n\times n})^{-1}\mathrm{RR}(\mathbf{s}_{1:n}, s_{n+1})^\top,$$

and augment the matrix $\Lambda_{(n+1)\times(n+1)} := \mathrm{diag}\{\Lambda_{n\times n}, \mathrm{C}_t(s_{n+1}, s_{n+1})\}$;

7: **end for**

8: Define,

$$\mathrm{m}(\cdot) = \mathrm{QR}(\cdot, \mathbf{s}_{1:N})(\mathrm{RR}(\mathbf{s}_{1:N}, \mathbf{s}_{1:N}) + \Lambda_{N\times N})^{-1}\mathbf{f}_{1:N},$$
$$\mathrm{C}(\cdot, \cdot) = \mathrm{QQ}(\cdot, \cdot) - \mathrm{QR}(\cdot, \mathbf{s}_{1:N})(\mathrm{RR}(\mathbf{s}_{1:N}, \mathbf{s}_{1:N}) + \Lambda_{N\times N})^{-1}\mathrm{QR}(\cdot, \mathbf{s}_{1:N})^\top;$$

Return both $\mathrm{u} \sim \mathcal{GP}(\mathrm{m}, \mathrm{C})$ and $\mathbf{f}_{1:N}$.

Consistency of Sampling Scheme

- The probabilistic IVP solution and its derivative at the n'th ($1 \leq n \leq N$) iteration of Algorithm 1 are Gaussian with mean and covariance that can be expressed recursively as,

$$\mathrm{m}^{n}(t_1) = \mathrm{m}^{n-1}(t_1) + (\mathrm{f}_n - \mathrm{m}_t^{n-1}(s_n)) \int_a^{t_1} \frac{\mathrm{C}_t^{n-1}(s, s_n)}{\Lambda_{n\times n}^{(n,n)} + \mathrm{C}_t^{n-1}(s_n, s_n)} \mathrm{d}s,$$

$$\mathrm{m}_t^{n}(t_1) = \mathrm{m}_t^{n-1}(t_1) + (\mathrm{f}_n - \mathrm{m}_t^{n-1}(s_n)) \frac{\mathrm{C}_t^{n-1}(t_1, s_n)}{\Lambda_{n\times n}^{(n,n)} + \mathrm{C}_t^{n-1}(s_n, s_n)},$$

$$\mathrm{C}^{n}(t_1, t_2) = \mathrm{C}^{n-1}(t_1, t_2) - \int_a^{t_1}\int_a^{t_2} \frac{\mathrm{C}_t^{n-1}(z, s_n)\,\mathrm{C}_t^{n-1}(s_n, s)}{\Lambda_{n\times n}^{(n,n)} + \mathrm{C}_t^{n-1}(s_n, s_n)} \mathrm{d}s\mathrm{d}z,$$

$$\mathrm{C}_t^{n}(t_1, t_2) = \mathrm{C}_t^{n-1}(t_1, t_2) - \frac{\mathrm{C}_t^{n-1}(t_1, s_n)\,\mathrm{C}_t^{n-1}(s_n, t_2)}{\Lambda_{n\times n}^{(n,n)} + \mathrm{C}_t^{n-1}(s_n, s_n)},$$

where m^0 and m_t^0 are the prior means and C^0 and C_t^0 the prior covariances of the state and derivatives.

Consistency of Sampling Scheme

- For a given $t \in [a, b]$ we find n such that $t \in [s_n, s_{n+1}]$, and bound the expected absolute difference between the nth probabilistic solution and the exact solution as follows,

$$
\begin{aligned}
\beta_n(t) &:= \mathrm{E}\left\{ \left| \mathrm{u}(t) - \mathrm{u}^*(t) \right| \;\middle|\; \mathbf{f}_{1:n}, \theta, \Psi \right\} \\
&= \mathrm{E}\left[\mathrm{u}(t) - \mathrm{u}^*(t) \mid \mathbf{f}_{1:n}, \theta, \Psi \right] \left\{ 1 - 2\Phi \left(- \frac{\mathrm{E}\left[\mathrm{u}(t) - \mathrm{u}^*(t) \mid \mathbf{f}_{1:n}, \theta, \Psi \right]}{\sqrt{\mathrm{C}^n(t,t)}} \right) \right\} \\
&\qquad + \sqrt{\frac{2}{\pi} \mathrm{C}^n(t,t)} \; \exp \left\{ - \frac{\left(\mathrm{E}\left[\mathrm{u}(t) - \mathrm{u}^*(t) \mid \mathbf{f}_{1:n}, \theta, \Psi \right] \right)^2}{2\mathrm{C}^n(t,t)} \right\}, \\
&\leq \left| \mathrm{E}\left\{ \mathrm{u}(t) - \mathrm{u}^*(t) \mid \mathbf{f}_{1:n}, \theta, \Psi \right\} \right| + \sqrt{2\mathrm{C}^1(t,t)} \qquad (9)
\end{aligned}
$$

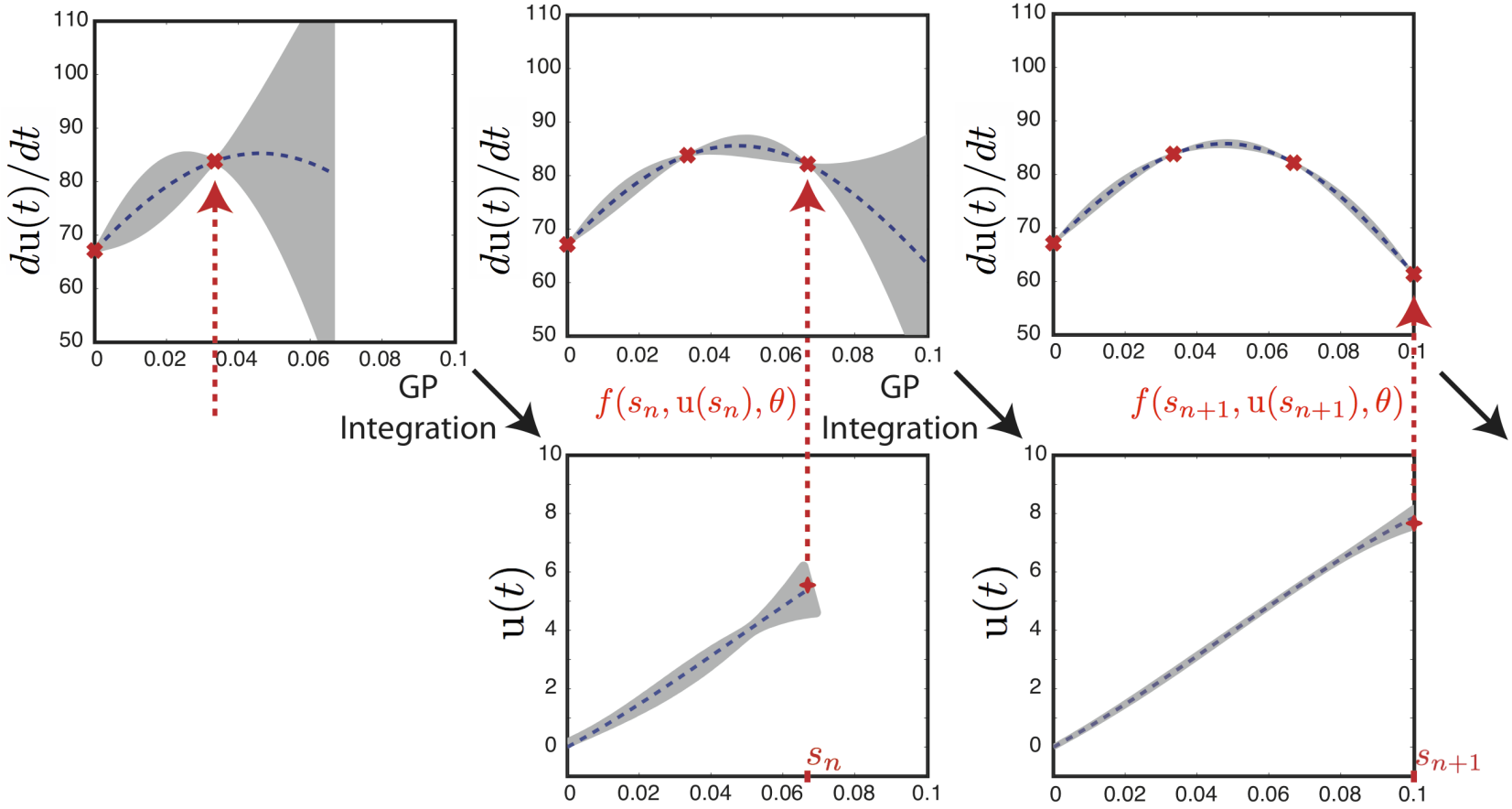


Figure: Illustration of method for generating a sample from the joint distribution of derivative observations and possible trajectories with density $p(\mathrm{u}(\mathbf{t}), \mathbf{f}_{1:N} \mid \theta, \Psi)$. Given two derivative model realisations (red points), we obtain a posterior distribution over the derivative space (top left) and over the state space (bottom middle). A sample is then drawn from the predictive posterior over the states at the next time point s_n (bottom middle), and a model realisation is obtained by mapping $\mathrm{u}(s_n)$ to the derivative space via the function f (top middle, rightmost red point). Given these three model evaluations, this procedure may be repeated (bottom right, top right).

Posterior Sampling for Inverse Problem

Algorithm 2 Draw K samples from the posterior distribution with density $p\big(\theta, \mathrm{u}(\mathbf{t}) \mid y(\mathbf{t}), \Psi\big)$

1: Initialise θ and conditionally sample a realisation of the state $\mathrm{u}(\mathbf{t})$;
2: **for** $k = 1 : K$ **do**
3: Propose $\theta' \sim q(\theta' \mid \theta)$, where q is a proposal density;
4: Sample a probabilistic realisation of the state $\mathrm{u}'(\mathbf{t})$ conditioned on θ' via Algorithm 1;
5: Compute:

$$\rho = \frac{q(\theta' \mid \theta)}{q(\theta \mid \theta')} \, \frac{p(\theta')}{p(\theta)} \, \frac{p\big(y(\mathbf{t}) \mid \mathcal{G}\big(\mathrm{u}'(\mathbf{t}), \theta'\big), \Sigma\big)}{p\big(y(\mathbf{t}) \mid \mathcal{G}\big(\mathrm{u}(\mathbf{t}), \theta\big), \Sigma\big)};$$

6: **if** $\min\{1, \rho\} > \mathrm{U}[0, 1]$ **then**
7: Update $\theta = \theta'$;
8: Update $\mathrm{u}(\mathbf{t}) = \mathrm{u}'(\mathbf{t})$;
9: **end if**
10: Return $\theta, \mathrm{u}(\mathbf{t})$.
11: **end for**

Heat Equation PDE

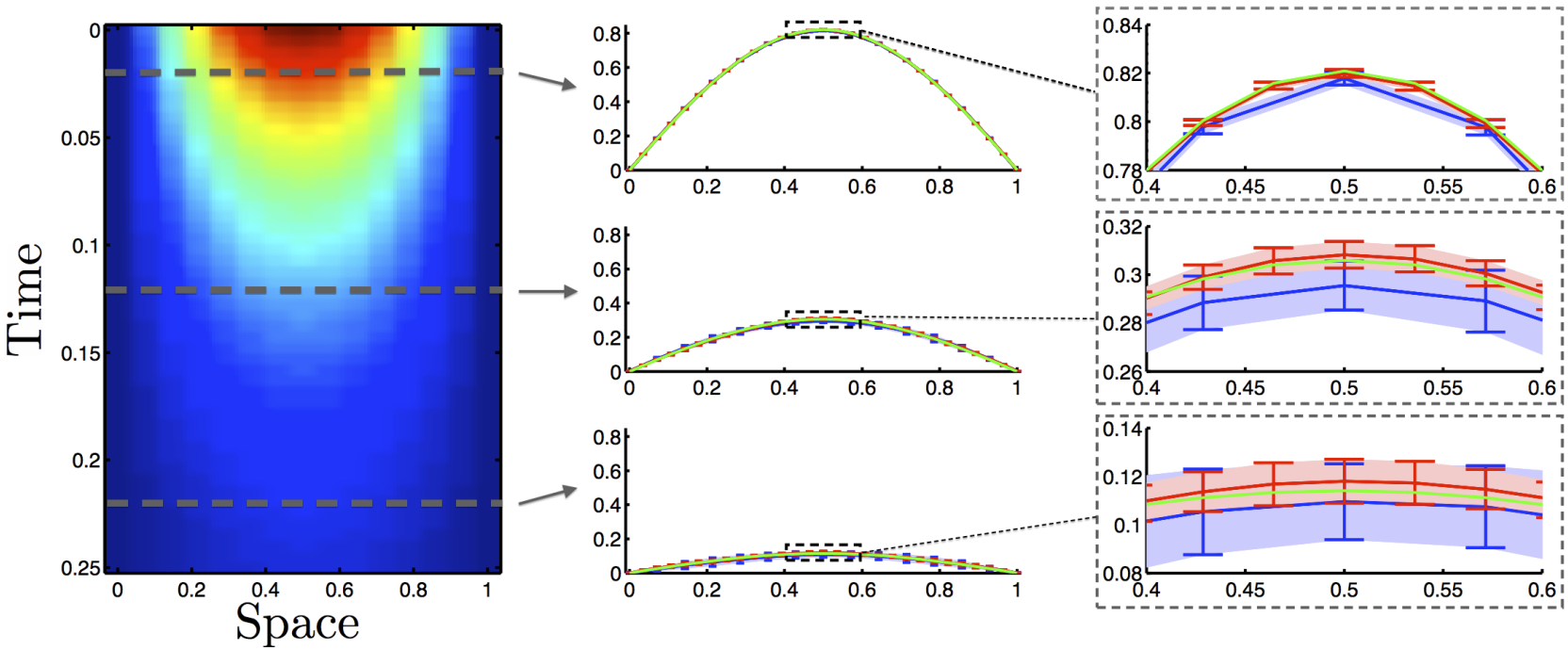


Figure: We illustrate the probabilistic output of the solution to the heat equation PDE, with $\kappa = 1$, integrated between $t = 0$ and $t = 0.25$ using two grid sizes; the coarser mesh (shown in blue) consists of 15 spatial discretisation points and 50 time discretisation points, the finer mesh consists of 29 spatial discretisation points and 100 time discretisation points. We show the spatial posterior predictions at three time points; $t = 0.02$ (top), $t = 0.12$ (middle) and $t = 0.22$ (bottom). The exact solution at each time point is represented by the green line. The error bars show the mean and 2 standard deviations for each of the probabilistic solutions calculated using 50 simulations.

Heat Equation PDE

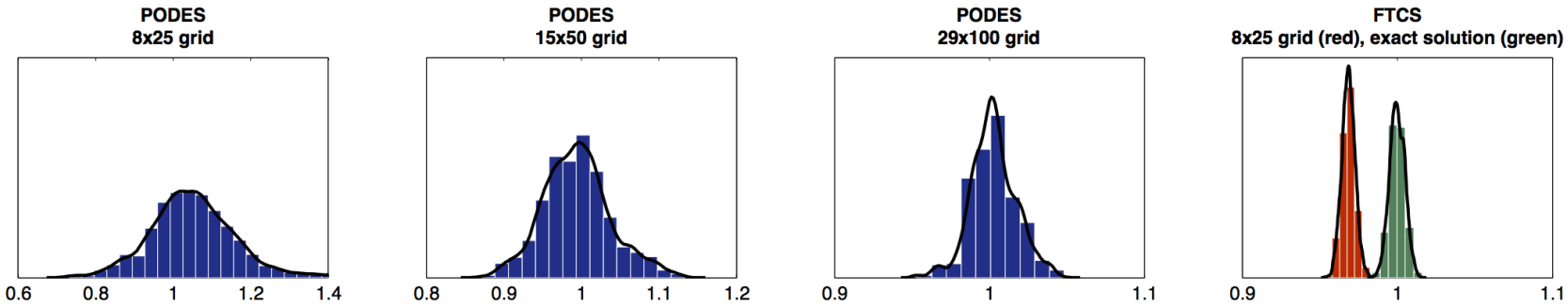


Figure: We illustrate the inverse problem by performing inference over the parameter κ in the heat equation, integrated between $t = 0$ and $t = 0.25$. We generate data over a grid of 8 spatial discretisation points and 25 time discretisation points by using the exact solution with $\kappa = 1$, then adding noise with standard deviation of 0.005. We firstly use the probabilistic differential equation solver (PODES) using three grid sizes; a coarse mesh consisting of 8 spatial discretisation points and 25 time discretisation points (far left), a finer mesh consisting of 15 spatial discretisation points and 50 time discretisation points (second from left), and a further finer mesh consisting of 29 spatial discretisation points and 100 time discretisation points (second from right). Note the change in scale as the posterior variance decreases with increasing resolution of the discretisation. As an illustrative comparison, we show the posterior distributions using a deterministic forward in time, centred in space (FTCS) integration scheme (far right). If the discretisation is not fine enough, we obtain an overconfident biased posterior that assigns negligible probability mass to the true value of κ. In contrast, use of the exact solution produces a perfectly unbiased posterior, as expected.

Lorenz Dynamic System

- Sampling yields consistent inference for Lipschitz-continuous equations

Lorenz Dynamic System

- Sampling yields consistent inference for Lipschitz-continuous equations
- Consider sampling solutions for Lorenz system

$$\begin{aligned}
\dot{u} &= -\sigma u + \sigma v \\
\dot{v} &= -ru - v - uw \\
\dot{w} &= uv - bw, \\
(u(a), v(a), w(a)) &= (u_0, v_0, w_0).
\end{aligned}$$

Lorenz Dynamic System

- Sampling yields consistent inference for Lipschitz-continuous equations
- Consider sampling solutions for Lorenz system

$$\begin{aligned} \dot{u} &= -\sigma u + \sigma v \\ \dot{v} &= -ru - v - uw \\ \dot{w} &= uv - bw, \\ (u(a), v(a), w(a)) &= (u_0, v_0, w_0). \end{aligned}$$

- Setting $(\sigma, r, b) = (10, 8/3, 28)$, initial states $(-11, -5, 38)$ neighbouring trajectories diverge exponentially fast

Lorenz Dynamic System

- Sampling yields consistent inference for Lipschitz-continuous equations
- Consider sampling solutions for Lorenz system

$$\begin{aligned} \dot{u} &= -\sigma u + \sigma v \\ \dot{v} &= -ru - v - uw \\ \dot{w} &= uv - bw, \\ (u(a), v(a), w(a)) &= (u_0, v_0, w_0). \end{aligned}$$

- Setting $(\sigma, r, b) = (10, 8/3, 28)$, initial states $(-11, -5, 38)$ neighbouring trajectories diverge exponentially fast
- Probabilistic solution using 3000 equally spaced solver knots on the interval $(0, 10)$. Squared exponential covariance function, length scale twice step size, prior precision $1e^{-3}$, reflecting prior knowledge system exhibits chaotic dynamics.

Lorenz Dynamic System

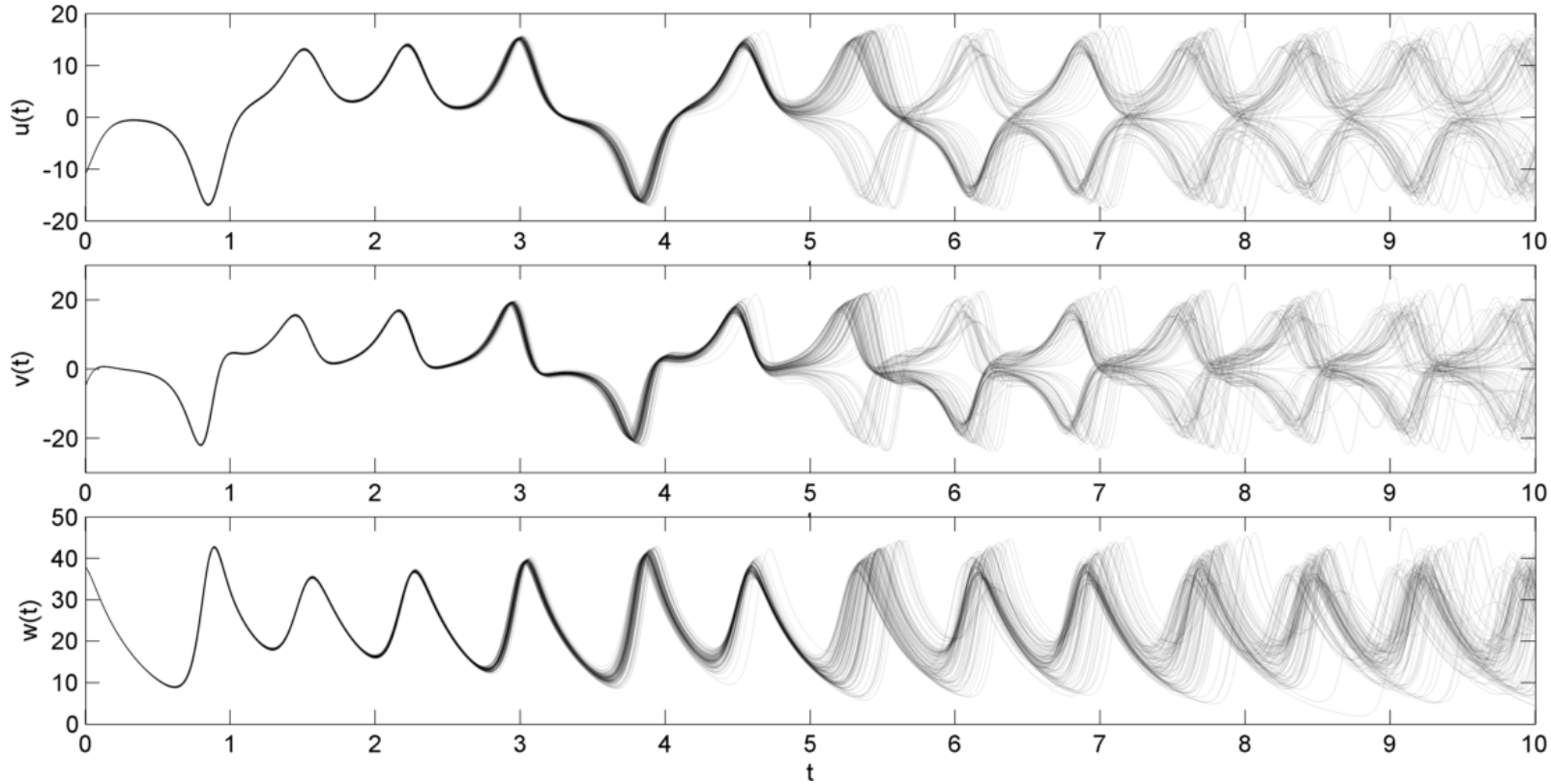


Figure: One hundred sample solutions drawn from posterior Gaussian measure for the Lorenz system under a fixed initial state, and fixed step size. Probabilisitic description of the chaotic behaviour resulting from the finite mesh approximation.

Example: Navier-Stokes Equations (2-d domain)

The Navier-Stokes (NS) PDEs model the time evolution of P components of the velocity, $\mathbf{u} : \mathscr{D} \to \mathcal{R}^P$, of an incompressible fluid on a spatial domain $\mathscr{X}$. The NS boundary value problem on the spatio-temporal domain $\mathscr{D} = \mathscr{X} \times \mathcal{T}$ is given by:

$$\begin{cases} \frac{\partial}{\partial t}\mathbf{u} - \theta_1 \, \Delta\mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{u} &= \mathrm{f} - \nabla \mathrm{p}, & (x,t) \in \mathscr{D}, \\ \nabla \cdot \mathbf{u} &= 0, & (x,t) \in \mathscr{D}, \\ \int \mathbf{u}^{(j)} \, \mathrm{d}x &= 0, & (x,t) \in \mathscr{D}, \ j = 1,2, \\ \mathbf{u} &= \mathbf{u}_B, & (x,t) \in \mathscr{X} \times \{0\}. \end{cases} \tag{10}$$

Example: Navier-Stokes Equations (2-d domain)

The Navier-Stokes (NS) PDEs model the time evolution of P components of the velocity, $\mathbf{u} : \mathscr{D} \to \mathcal{R}^P$, of an incompressible fluid on a spatial domain $\mathcal{X}$. The NS boundary value problem on the spatio-temporal domain $\mathscr{D} = \mathcal{X} \times \mathcal{T}$ is given by:

$$\begin{cases} \frac{\partial}{\partial t}\mathbf{u} - \theta_1 \Delta\mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{u} &= \mathrm{f} - \nabla \mathrm{p}, & (x,t) \in \mathscr{D}, \\ \nabla \cdot \mathbf{u} &= 0, & (x,t) \in \mathscr{D}, \\ \int \mathbf{u}^{(j)}\, \mathrm{d}x &= 0, & (x,t) \in \mathscr{D},\ j = 1,2, \\ \mathbf{u} &= \mathbf{u}_B, & (x,t) \in \mathcal{X} \times \{0\}. \end{cases} \tag{10}$$

This model is parameterized by the viscosity of the fluid, $\theta_1 > 0$; the pressure function $\mathrm{p} : \mathscr{D} \to \mathbb{R}$; and the external time-homogeneous forcing function $\mathrm{f} : \mathcal{X} \to \mathbb{R}$. We consider the NS equations over a 2-dimensional torus shaped domain, $\mathcal{X} = [0, 2\pi) \times [0, 2\pi)$, expressed in spherical coordinates. We further assume periodic boundary conditions, and viscosity $\theta_1 = 1 \times 10^{-3}$ in the turbulent regime. For simplicity, we consider the unforced NS equations ($\mathrm{f} = 0$).

Example: Navier-Stokes Equations (2-d domain)

Often, the quantity of interest is the vorticity, or local spinning motion of the incompressible fluid, which we define as,

$$\varpi(x, t, \theta) = -\nabla \times \mathbf{u}(x, t, \theta),$$

where clockwise rotation corresponds to positive vorticity. Vorticity can also help to better visualize the solution of the NS system by summarizing the two components of velocity by a one-dimensional function.

Example: Navier-Stokes Equations (2-d domain)

Often, the quantity of interest is the vorticity, or local spinning motion of the incompressible fluid, which we define as,

$$\varpi(x, t, \theta) = -\nabla \times \mathbf{u}(x, t, \theta),$$

where clockwise rotation corresponds to positive vorticity. Vorticity can also help to better visualize the solution of the NS system by summarizing the two components of velocity by a one-dimensional function.

The Navier-Stokes model was reduced to a set of 64×64 stiff coupled ODEs with associated constraints through a pseudo-spectral projection in Fourier space.

Example: Navier-Stokes Equations (2-d domain)

Often, the quantity of interest is the vorticity, or local spinning motion of the incompressible fluid, which we define as,

$$\varpi(x, t, \theta) = -\nabla \times \mathbf{u}(x, t, \theta),$$

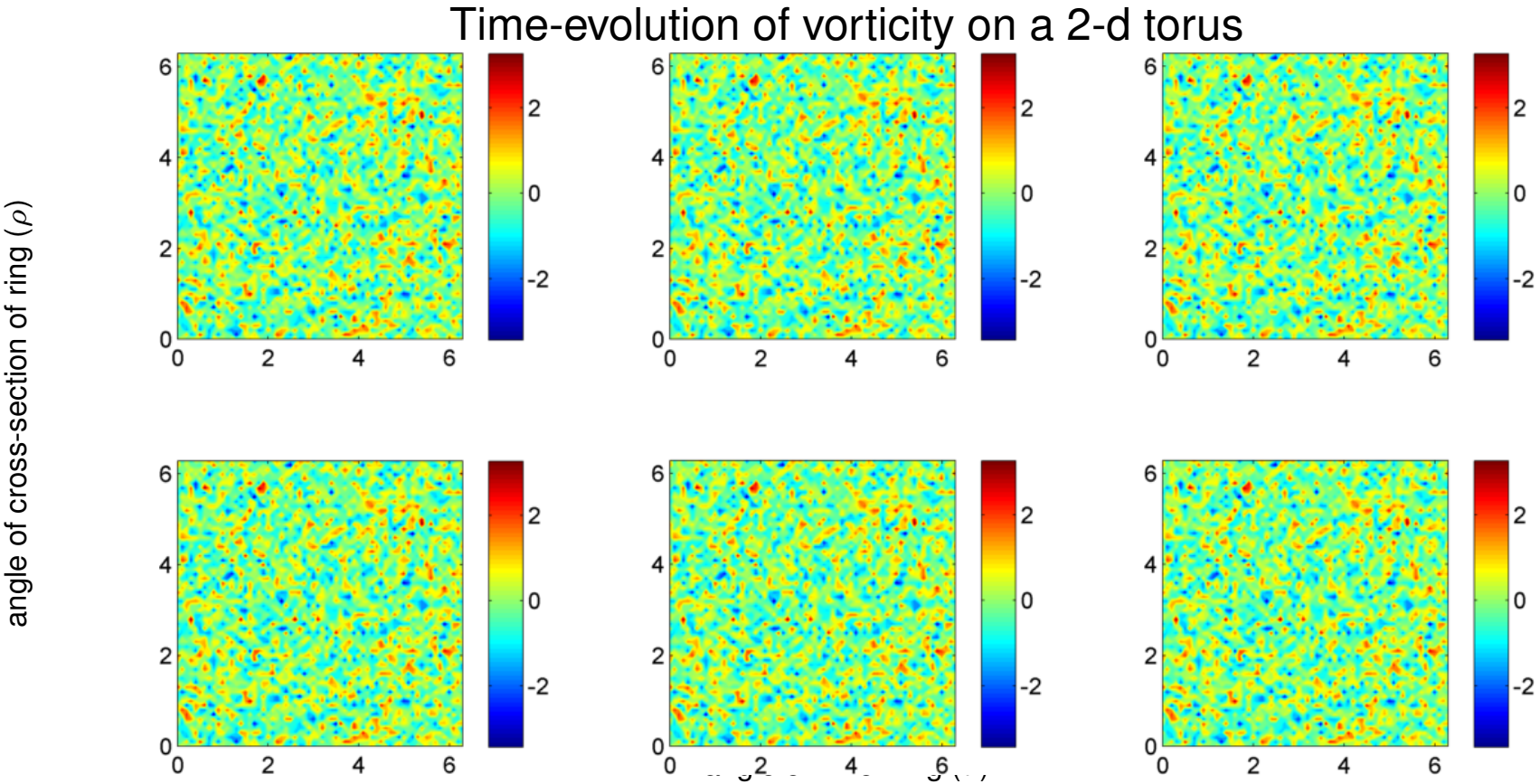
where clockwise rotation corresponds to positive vorticity. Vorticity can also help to better visualize the solution of the NS system by summarizing the two components of velocity by a one-dimensional function.

The Navier-Stokes model was reduced to a set of 64×64 stiff coupled ODEs with associated constraints through a pseudo-spectral projection in Fourier space.

The spatial discretization grid was equally spaced. The initial velocity field is generated from a bivariate normal distribution at each of the mesh points.

Example: Navier-Stokes Equations (2-d domain)

The probabilistic solution accounts temporal discretization uncertainty



Kuramoto-Sivashinsky model of reaction-diffusion

- Kuramoto-Sivashinsky model of reaction-diffusion systems

Kuramoto-Sivashinsky model of reaction-diffusion

- Kuramoto-Sivashinsky model of reaction-diffusion systems

$$\frac{\partial}{\partial t}u = -u\frac{\partial}{\partial x}u - \frac{\partial^2}{\partial x^2}u - \frac{\partial^4}{\partial x^4}u,$$

Kuramoto-Sivashinsky model of reaction-diffusion

- Kuramoto-Sivashinsky model of reaction-diffusion systems

$$\frac{\partial}{\partial t}u = -u\frac{\partial}{\partial x}u - \frac{\partial^2}{\partial x^2}u - \frac{\partial^4}{\partial x^4}u,$$

- Domain $x \in [0, 32\pi], t \in [0, 150]$

Kuramoto-Sivashinsky model of reaction-diffusion

- Kuramoto-Sivashinsky model of reaction-diffusion systems

$$\frac{\partial}{\partial t}u = -u\frac{\partial}{\partial x}u - \frac{\partial^2}{\partial x^2}u - \frac{\partial^4}{\partial x^4}u,$$

- Domain $x \in [0, 32\pi], t \in [0, 150]$
- Initial function $u(0, x) = \cos(x/16)\left\{1 + \sin(x/16)\right\}$

Kuramoto-Sivashinsky model of reaction-diffusion

- Kuramoto-Sivashinsky model of reaction-diffusion systems

$$\frac{\partial}{\partial t}u = -u\frac{\partial}{\partial x}u - \frac{\partial^2}{\partial x^2}u - \frac{\partial^4}{\partial x^4}u,$$

- Domain $x \in [0, 32\pi], t \in [0, 150]$
- Initial function $u(0, x) = \cos(x/16)\left\{1 + \sin(x/16)\right\}$
- Discretize spatial domain, obtaining a high-dimensional (128 dimensions) system of stiff ODEs

Kuramoto-Sivashinsky model of reaction-diffusion

- Kuramoto-Sivashinsky model of reaction-diffusion systems

$$\frac{\partial}{\partial t}u = -u\frac{\partial}{\partial x}u - \frac{\partial^2}{\partial x^2}u - \frac{\partial^4}{\partial x^4}u,$$

- Domain $x \in [0, 32\pi], t \in [0, 150]$
- Initial function $u(0, x) = \cos(x/16)\left\{1 + \sin(x/16)\right\}$
- Discretize spatial domain, obtaining a high-dimensional (128 dimensions) system of stiff ODEs
- Use the integrating factor method to transform the system to one of purely nonlinear ODEs

Kuramoto-Sivashinsky model of reaction-diffusion

- Kuramoto-Sivashinsky model of reaction-diffusion systems

$$\frac{\partial}{\partial t}u = -u\frac{\partial}{\partial x}u - \frac{\partial^2}{\partial x^2}u - \frac{\partial^4}{\partial x^4}u,$$

- Domain $x \in [0, 32\pi], t \in [0, 150]$
- Initial function $u(0, x) = \cos(x/16)\left\{1 + \sin(x/16)\right\}$
- Discretize spatial domain, obtaining a high-dimensional (128 dimensions) system of stiff ODEs
- Use the integrating factor method to transform the system to one of purely nonlinear ODEs
- Probabilistic IVP solutions sampled using 2K uniform solver knots

Kuramoto-Sivashinsky model of reaction-diffusion

- Kuramoto-Sivashinsky model of reaction-diffusion systems

$$\frac{\partial}{\partial t}u = -u\frac{\partial}{\partial x}u - \frac{\partial^2}{\partial x^2}u - \frac{\partial^4}{\partial x^4}u,$$

- Domain $x \in [0, 32\pi], t \in [0, 150]$
- Initial function $u(0, x) = \cos(x/16)\left\{1 + \sin(x/16)\right\}$
- Discretize spatial domain, obtaining a high-dimensional (128 dimensions) system of stiff ODEs
- Use the integrating factor method to transform the system to one of purely nonlinear ODEs
- Probabilistic IVP solutions sampled using 2K uniform solver knots
- Fifteen solution samples illustrate uncertainty over domain propagates through system resulting in noticeably distinct dynamics, not captured by deterministic numerical solvers.

Kuramoto-Sivashinsky model of reaction-diffusion

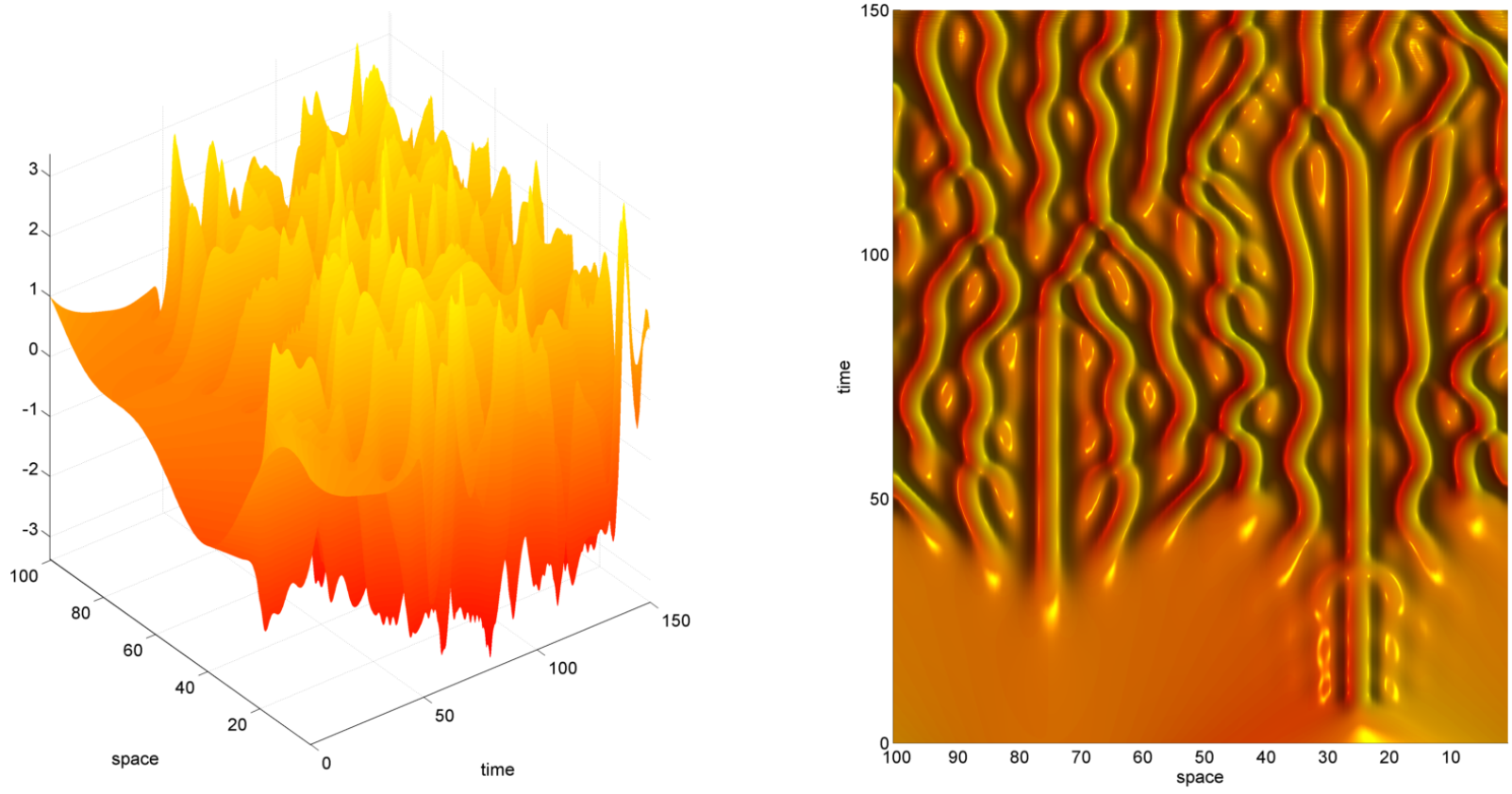


Figure: Side view and top view of a probabilistic solution realization of the Kuramoto-Sivashinsky PDE with initial function $u(0, x) = \cos(x/16)\{1 + \sin(x/16)\}$ and domain $x \in [0, 32\pi]$, $t \in [0, 150]$.

Kuramoto-Sivashinsky model of reaction-diffusion

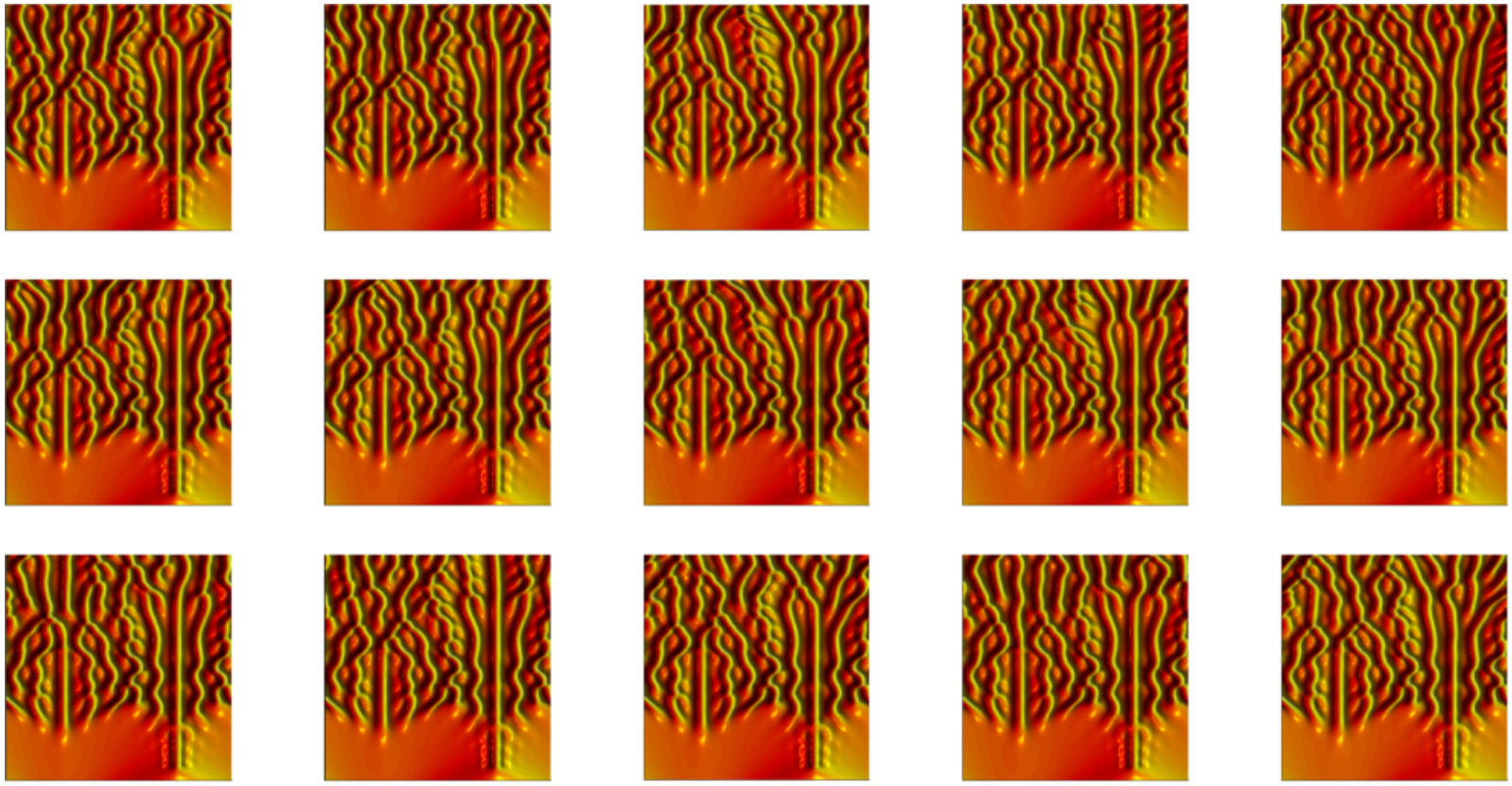


Figure: Fifteen realizations of the probabilistic solution of the Kuramoto-Sivashinsky PDE using a fixed initial function. The solution is known to exhibit temporal chaos. Deterministic numerical solutions only capture one type of behaviour given a fixed initial function, which can lead to bias when used in conjunction with data-based inference.

Full Bayesian Uncertainty Quantification

- Exploit Riemann Manifold MCMC Langevin or HMC simulation

Full Bayesian Uncertainty Quantification

- Exploit Riemann Manifold MCMC Langevin or HMC simulation
- Note that FI for $p\left(x(\mathbf{t})\middle|\mathbf{f}_{1:N}, \boldsymbol{\theta}, \mathbf{x}_0, \Psi\right) = \mathcal{N}_T\left(m_N(\mathbf{t}), \mathcal{C}_N(\mathbf{t}, \mathbf{t})\right)$ follows as

Full Bayesian Uncertainty Quantification

- Exploit Riemann Manifold MCMC Langevin or HMC simulation
- Note that FI for $p\left(x(\mathbf{t})\middle|\mathbf{f}_{1:N},\boldsymbol{\theta},\mathbf{x}_0,\Psi\right)=\mathcal{N}_T\left(m_N(\mathbf{t}),\mathcal{C}_N(\mathbf{t},\mathbf{t})\right)$ follows as

$$\frac{\partial m_N(\mathbf{t})}{\partial\theta_i}\mathcal{C}_N^{-1}(\mathbf{t},\mathbf{t})\frac{\partial m_N(\mathbf{t})}{\partial\theta_j}$$

Full Bayesian Uncertainty Quantification

- Exploit Riemann Manifold MCMC Langevin or HMC simulation
- Note that FI for $p\big(x(\mathbf{t})\big|\mathbf{f}_{1:N},\boldsymbol{\theta},\mathbf{x}_0,\Psi\big)=\mathcal{N}_T\big(m_N(\mathbf{t}),\mathcal{C}_N(\mathbf{t},\mathbf{t})\big)$ follows as

$$\frac{\partial m_N(\mathbf{t})}{\partial\theta_i}\mathcal{C}_N^{-1}(\mathbf{t},\mathbf{t})\frac{\partial m_N(\mathbf{t})}{\partial\theta_j}$$

$$\frac{\partial m_N(\mathbf{t})}{\partial\theta_i}=\mathrm{QR}(\mathbf{t},\mathbf{s})\big(\Lambda_n+\mathrm{RR}(\mathbf{s},\mathbf{s})\big)^{-1}\frac{\partial\mathbf{f}_{1:N}(\mathbf{t})}{\partial\theta_i}$$

Full Bayesian Uncertainty Quantification

- Exploit Riemann Manifold MCMC Langevin or HMC simulation
- Note that FI for $p\big(x(\mathbf{t})\big|\mathbf{f}_{1:N}, \boldsymbol{\theta}, \mathbf{x}_0, \Psi\big) = \mathcal{N}_T\big(m_N(\mathbf{t}), \mathcal{C}_N(\mathbf{t},\mathbf{t})\big)$ follows as

$$\frac{\partial m_N(\mathbf{t})}{\partial \theta_i} \mathcal{C}_N^{-1}(\mathbf{t},\mathbf{t}) \frac{\partial m_N(\mathbf{t})}{\partial \theta_j}$$

$$\frac{\partial m_N(\mathbf{t})}{\partial \theta_i} = \mathrm{QR}(\mathbf{t},\mathbf{s})\big(\Lambda_n + \mathrm{RR}(\mathbf{s},\mathbf{s})\big)^{-1} \frac{\partial \mathbf{f}_{1:N}(\mathbf{t})}{\partial \theta_i}$$

- This representation only requires pre-computed solution of linear systems for metric tensor of MMALA (and simplified version) and RMHMC

Full Bayesian Uncertainty Quantification

- Exploit Riemann Manifold MCMC Langevin or HMC simulation
- Note that FI for $p\left(x(\mathbf{t})\middle|\mathbf{f}_{1:N},\boldsymbol{\theta},\mathbf{x}_0,\Psi\right)=\mathcal{N}_T\left(m_N(\mathbf{t}),\mathcal{C}_N(\mathbf{t},\mathbf{t})\right)$ follows as

$$\frac{\partial m_N(\mathbf{t})}{\partial\theta_i}\mathcal{C}_N^{-1}(\mathbf{t},\mathbf{t})\frac{\partial m_N(\mathbf{t})}{\partial\theta_j}$$

$$\frac{\partial m_N(\mathbf{t})}{\partial\theta_i}=\mathrm{QR}(\mathbf{t},\mathbf{s})\left(\Lambda_n+\mathrm{RR}(\mathbf{s},\mathbf{s})\right)^{-1}\frac{\partial\mathbf{f}_{1:N}(\mathbf{t})}{\partial\theta_i}$$

- This representation only requires pre-computed solution of linear systems for metric tensor of MMALA (and simplified version) and RMHMC
- Highly efficient proposal mechanism for very high-dimensional scenarios - see Girolami and Calderhead, 2011 for details.

Inference for model of cellular signal transduction

- States rates of change described by delay differential equation system

Inference for model of cellular signal transduction

- States rates of change described by delay differential equation system

$$
\begin{aligned}
\dot{u}^{(1)}(t) &= -k_1 u^{(1)}(t)\ \mathrm{Epo}R_A(t) + 2k_4 u^{(4)}(t-\tau) \\
\dot{u}^{(2)}(t) &= \quad k_1 u^{(1)}(t)\ \mathrm{Epo}R_A(t) - k_2 {u^{(2)}}^2(t) \\
\dot{u}^{(3)}(t) &= -k_3 u^{(3)}(t) + 0.5 k_2 {u^{(2)}}^2(t) \\
\dot{u}^{(4)}(t) &= \quad k_3 u^{(3)}(t) - k_4 u^{(4)}(t-\tau)
\end{aligned}
$$

$u^{(2)}(0) = u^{(3)}(0) = u^{(4)}(0) = 0$ known and $u^{(1)}$ unknown.

Inference for model of cellular signal transduction

- States rates of change described by delay differential equation system

$$
\begin{aligned}
\dot{u}^{(1)}(t) &= -k_1 u^{(1)}(t)\ \mathrm{Epo}R_A(t) + 2k_4 u^{(4)}(t-\tau) \\
\dot{u}^{(2)}(t) &= \quad k_1 u^{(1)}(t)\ \mathrm{Epo}R_A(t) - k_2 {u^{(2)}}^2(t) \\
\dot{u}^{(3)}(t) &= -k_3 u^{(3)}(t) + 0.5 k_2 {u^{(2)}}^2(t) \\
\dot{u}^{(4)}(t) &= \quad k_3 u^{(3)}(t) - k_4 u^{(4)}(t-\tau)
\end{aligned}
$$

  $u^{(2)}(0) = u^{(3)}(0) = u^{(4)}(0) = 0$ known and $u^{(1)}$ unknown.

- States observed indirectly via $\mathcal{G} : \mathbb{R}^3 \times \Theta \to \mathbb{R}^4$

Inference for model of cellular signal transduction

- States rates of change described by delay differential equation system

$$\begin{aligned}
\dot{u}^{(1)}(t) &= -k_1 u^{(1)}(t)\ \mathrm{Epo}R_A(t) + 2k_4 u^{(4)}(t-\tau) \\
\dot{u}^{(2)}(t) &= \quad k_1 u^{(1)}(t)\ \mathrm{Epo}R_A(t) - k_2 {u^{(2)}}^2(t) \\
\dot{u}^{(3)}(t) &= -k_3 u^{(3)}(t) + 0.5 k_2 {u^{(2)}}^2(t) \\
\dot{u}^{(4)}(t) &= \quad k_3 u^{(3)}(t) - k_4 u^{(4)}(t-\tau)
\end{aligned}$$

$u^{(2)}(0) = u^{(3)}(0) = u^{(4)}(0) = 0$ known and $u^{(1)}$ unknown.

- States observed indirectly via $\mathcal{G} : \mathbb{R}^3 \times \Theta \to \mathbb{R}^4$

$$\begin{aligned}
\mathcal{G}_1(\mathbf{u}, \mathbf{k}) &= k_5 \left(u^{(1)} + 2u^{(3)} \right) \\
\mathcal{G}_2(\mathbf{u}, \mathbf{k}) &= k_6 \left(u^{(1)} + u^{(2)} + 2u^{(3)} \right) \\
\mathcal{G}_3(\mathbf{u}, \mathbf{k}) &= u^{(1)} \\
\mathcal{G}_4(\mathbf{u}, \mathbf{k}) &= u^{(3)} / \left(u^{(2)} + u^{(3)} \right)
\end{aligned}$$

Inference for model of cellular signal transduction

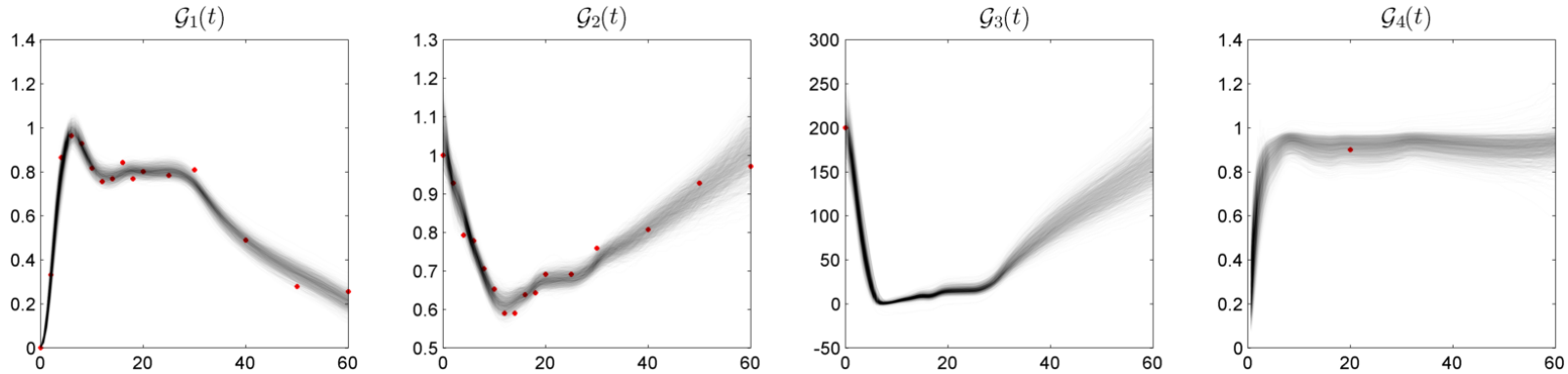


Figure: Experimental data and sample paths of the observation processes obtained by transforming a sample from marginal posterior state distribution by observation function

Inference for model of cellular signal transduction

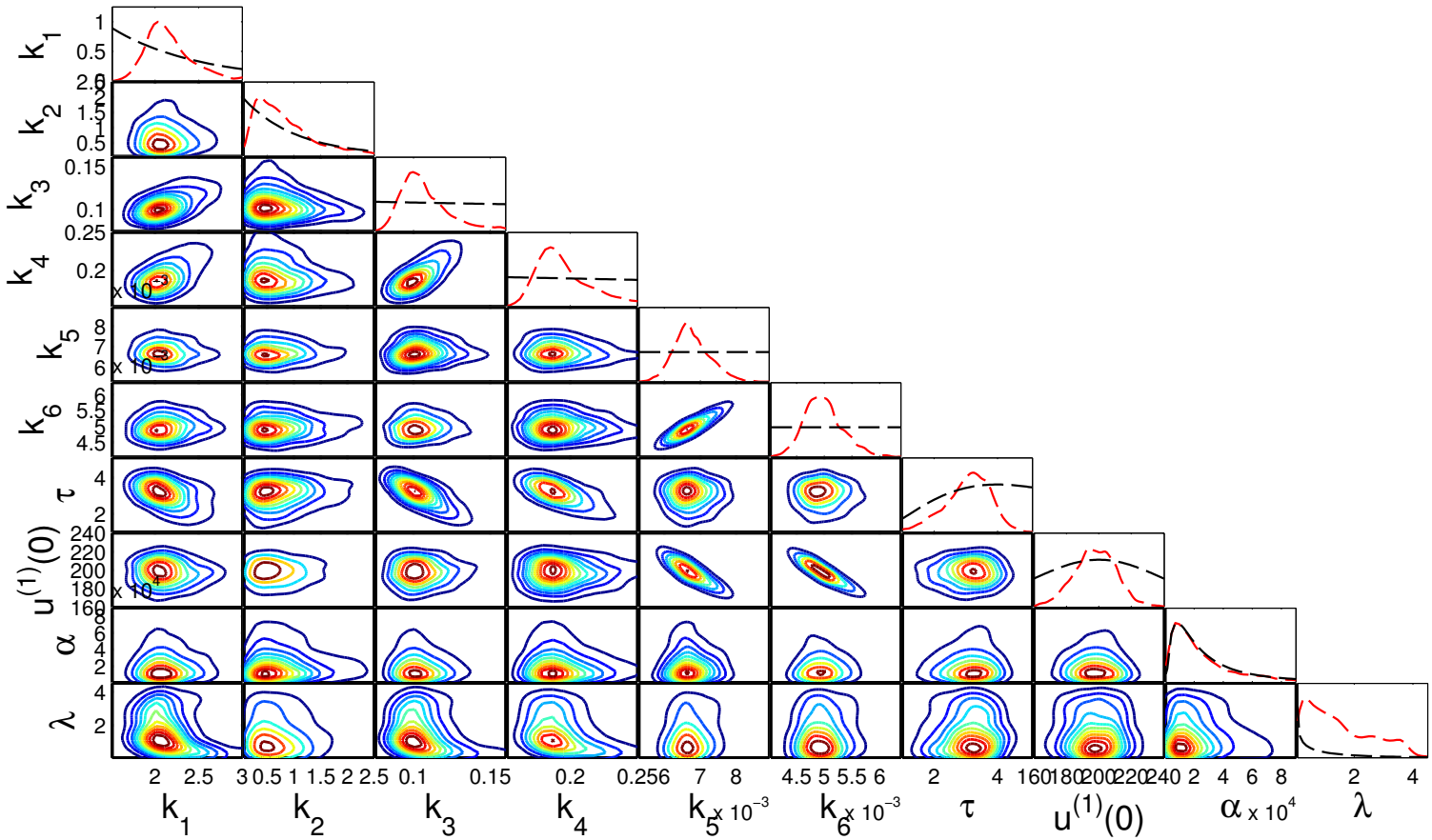


Figure: Marginal parameter posterior based on sample of size 100K generated by a parallel tempering algorithm utilizing seven chains, with the first 10K samples removed.

Conclusions and Discussion

- Addressed issue of model and discrete representation mismatch

Conclusions and Discussion

- Addressed issue of model and discrete representation mismatch
- Exploit probabilistic construction defines measure over infinite dim solution

Conclusions and Discussion

- Addressed issue of model and discrete representation mismatch
- Exploit probabilistic construction defines measure over infinite dim solution
- Inflation of uncertainty in discrete solutions can be assessed - mesh refinement

Conclusions and Discussion

- Addressed issue of model and discrete representation mismatch
- Exploit probabilistic construction defines measure over infinite dim solution
- Inflation of uncertainty in discrete solutions can be assessed - mesh refinement
- A Monte Carlo solution scheme for differential equations follows

Conclusions and Discussion

- Addressed issue of model and discrete representation mismatch
- Exploit probabilistic construction defines measure over infinite dim solution
- Inflation of uncertainty in discrete solutions can be assessed - mesh refinement
- A Monte Carlo solution scheme for differential equations follows
- Statistical machinery of experiment design to adaptively discretise

Conclusions and Discussion

- Addressed issue of model and discrete representation mismatch
- Exploit probabilistic construction defines measure over infinite dim solution
- Inflation of uncertainty in discrete solutions can be assessed - mesh refinement
- A Monte Carlo solution scheme for differential equations follows
- Statistical machinery of experiment design to adaptively discretise
- Consistent estimation of actual solution of differential equations

Conclusions and Discussion

- Addressed issue of model and discrete representation mismatch
- Exploit probabilistic construction defines measure over infinite dim solution
- Inflation of uncertainty in discrete solutions can be assessed - mesh refinement
- A Monte Carlo solution scheme for differential equations follows
- Statistical machinery of experiment design to adaptively discretise
- Consistent estimation of actual solution of differential equations
- Bayesian UQ follows via MCMC where probabilistic solver defines proposals

Conclusions and Discussion

- Addressed issue of model and discrete representation mismatch
- Exploit probabilistic construction defines measure over infinite dim solution
- Inflation of uncertainty in discrete solutions can be assessed - mesh refinement
- A Monte Carlo solution scheme for differential equations follows
- Statistical machinery of experiment design to adaptively discretise
- Consistent estimation of actual solution of differential equations
- Bayesian UQ follows via MCMC where probabilistic solver defines proposals
- Discretisation, Solution and UQ unified in common inferential framework

Conclusions and Discussion

- Addressed issue of model and discrete representation mismatch
- Exploit probabilistic construction defines measure over infinite dim solution
- Inflation of uncertainty in discrete solutions can be assessed - mesh refinement
- A Monte Carlo solution scheme for differential equations follows
- Statistical machinery of experiment design to adaptively discretise
- Consistent estimation of actual solution of differential equations
- Bayesian UQ follows via MCMC where probabilistic solver defines proposals
- Discretisation, Solution and UQ unified in common inferential framework
- Model based geometry available to exploit advanced MCMC simulation schemes

Acknowledgements

- Girolami funded by EPSRC Established Career Fellowship and Royal Society Wolfson Research Merit Award

Acknowledgements

- Girolami funded by EPSRC Established Career Fellowship and Royal Society Wolfson Research Merit Award
- EQUIP Programme Grant Funded by EPSRC
- Forthcoming paper

  Bayesian Uncertainty Quantification for Differential Equations
  arXiv:1306.2365