An Introduction to Probabilistic Numerical Methods

Chris. J. Oates
School of Mathematics, Statistics and Physics @ Newcastle University
Programme on Data-Centric Engineering @ Alan Turing Institute

October 2017 @ Turing
Numerical Methods

- optimisation
- integration
- linear algebra
- solution of differential equations
- ...

What is the fuss all about?
The goal:

Numerical Task $\implies$ Finite Computation $\implies$ Distribution on Output
Optimisation
Optimisation

\[ x^* = \arg \max f(x) \]

- Well-defined:
  - \( f \in C^\alpha(\mathcal{X}) \) for some \( \alpha \geq 0 \) and \( x \in \mathcal{X} \) a compact subset of \( \mathbb{R}^d \).

- Well-posed:
  - Allowed \( n \) evaluations of \( f(\cdot) \) at inputs which you can select.
  - Aim to minimise \( \| \hat{x}^* - x^* \|_2 \).
Optimisation

\[ x^* = \arg \max f(x) \]

- **Well-defined:**
  - \( f \in C^\alpha(\mathcal{X}) \) for some \( \alpha \geq 0 \) and \( x \in \mathcal{X} \) a compact subset of \( \mathbb{R}^d \).

- **Well-posed:**
  - Allowed \( n \) evaluations of \( f(\cdot) \) at inputs which you can select.
  - Aim to minimise \( \| \hat{x}^* - x^* \|_2 \).
$x^* = \arg\max f(x)$

- Well-defined:
  - $f \in C^\alpha(\mathcal{X})$ for some $\alpha \geq 0$ and $x \in \mathcal{X}$ a compact subset of $\mathbb{R}^d$.

- Well-posed:
  - Allowed $n$ evaluations of $f(\cdot)$ at inputs which you can select.
  - Aim to minimise $\|\hat{x}^* - x^*\|_2$. 
\[ x^* = \arg \max f(x) \]

- **Well-defined:**

  - \( f \in C^\alpha(X) \) for some \( \alpha \geq 0 \) and \( x \in X \) a compact subset of \( \mathbb{R}^d \).

- **Well-posed:**

  - Allowed \( n \) evaluations of \( f(\cdot) \) at inputs which you can select.
  - Aim to minimise \( \|\hat{x} - x^*\|_2 \).
$x^* = \arg \max f(x)$

- Well-defined:
  - $f \in C^\alpha(\mathcal{X})$ for some $\alpha \geq 0$ and $x \in \mathcal{X}$ a compact subset of $\mathbb{R}^d$.

- Well-posed:
  - Allowed $n$ evaluations of $f(\cdot)$ at inputs which you can select.
  - Aim to minimise $\|\hat{x}^* - x^*\|_2$. 

Chris. J. Oates
Probabilistic Numerical Computation
October 2017 @ Turing
Optimisation

\[ x^* = \arg \max f(x) \]

- Well-defined:
  - \( f \in C^\alpha(X) \) for some \( \alpha \geq 0 \) and \( x \in X \) a compact subset of \( \mathbb{R}^d \).

- Well-posed:
  - Allowed \( n \) evaluations of \( f(\cdot) \) at inputs which you can select.
  - Aim to minimise \( \| \hat{x}^* - x^* \|_2 \).
$$x^* = \arg \max f(x)$$

Two distinct requirements:

- A method to select the function evaluation locations $x_1, \ldots, x_n$.
  - Uniform grid over $X$?
  - Adaptive selection, e.g. gradient ascent with estimated gradients?
- An estimator $\{(x_i, f(x_i))\}_{i=1}^n \mapsto \hat{x}^*$.
  - The empirical maximum $\hat{x}^* = x_{i^*}$ where $i^* = \arg \max_{i=1,\ldots,n} f(x_i)$?
  - Something better?
- Key idea: Estimator uncertainty quantification!
Optimisation

\[ x^* = \text{arg max } f(x) \]

Two distinct requirements:

- A method to select the function evaluation locations \(x_1, \ldots, x_n\).
  - Uniform grid over \(\mathcal{X}\)?
  - Adaptive selection, e.g. gradient ascent with estimated gradients?
- An estimator \(\{(x_i, f(x_i))\}_{i=1}^{n} \mapsto \hat{x}^*\).
  - The empirical maximum \(\hat{x}^* = x_{i^*}\) where \(i^* = \text{arg max}_{i=1,\ldots,n} f(x_i)\)?
  - Something better?

- Key idea: Estimator uncertainty quantification!
\[ x^* = \arg \max f(x) \]

Two distinct requirements:
- A method to select the function evaluation locations \( x_1, \ldots, x_n \).
  - Uniform grid over \( \mathcal{X} \)?
  - Adaptive selection, e.g. gradient ascent with estimated gradients?
- An estimator \( \{(x_i, f(x_i))\}_{i=1}^n \mapsto \hat{x}^* \).
  - The empirical maximum \( \hat{x}^* = x_{i^*} \) where \( i^* = \arg \max_{i=1,\ldots,n} f(x_i) \)?
  - Something better?
- Key idea: Estimator uncertainty quantification!
Optimisation

\[ x^* = \arg \max f(x) \]

Two distinct requirements:

- A method to select the function evaluation locations \( x_1, \ldots, x_n \).
  - Uniform grid over \( \mathcal{X} \)?
  - Adaptive selection, e.g. gradient ascent with estimated gradients?

- An estimator \( \{(x_i, f(x_i))\}_{i=1}^n \mapsto \hat{x}^* \).
  - The empirical maximum \( \hat{x}^* = x_{i^*} \) where \( i^* = \arg \max_{i=1,\ldots,n} f(x_i) \)?
  - Something better?

- Key idea: Estimator uncertainty quantification!
\[ x^* = \arg \max f(x) \]

Two distinct requirements:
- A method to select the function evaluation locations \( x_1, \ldots, x_n \).
  - Uniform grid over \( \mathcal{X} \)?
  - Adaptive selection, e.g. gradient ascent with estimated gradients?
- An estimator \( \{(x_i, f(x_i))\}_{i=1}^n \mapsto \hat{x}^* \).
  - The empirical maximum \( \hat{x}^* = x_{i^*} \) where \( i^* = \arg \max_{i=1,\ldots,n} f(x_i) \)?
  - Something better?

Key idea: Estimator uncertainty quantification!
\[ x^* = \arg \max f(x) \]

Two distinct requirements:

- A method to select the function evaluation locations \( x_1, \ldots, x_n \).
  - Uniform grid over \( \mathcal{X} \)?
  - Adaptive selection, e.g. gradient ascent with estimated gradients?

- An estimator \( \{(x_i, f(x_i))\}_{i=1}^n \mapsto \hat{x}^* \).
  - The empirical maximum \( \hat{x}^* = x_{i^*} \) where \( i^* = \arg \max_{i=1,\ldots,n} f(x_i) \)?
  - Something better?

- Key idea: Estimator uncertainty quantification!

---

Chris. J. Oates  
Probabilistic Numerical Computation  
October 2017 @ Turing  
7 / 42
Optimisation

\[ x^* = \arg \max f(x) \]

Two distinct requirements:

- A method to select the function evaluation locations \( x_1, \ldots, x_n \).
  - Uniform grid over \( X \)?
  - Adaptive selection, e.g. gradient ascent with estimated gradients?
- An estimator \( \{(x_i, f(x_i))\}_{i=1}^n \mapsto \hat{x}^* \).
  - The empirical maximum \( \hat{x}^* = x_{i^*} \) where \( i^* = \arg \max_{i=1,\ldots,n} f(x_i) \)?
  - Something better?
- Key idea: Estimator uncertainty quantification!
\[ x^* = \arg \max f(x) \]

Two distinct requirements:

- A method to select the function evaluation locations \( x_1, \ldots, x_n \).
  - Uniform grid over \( \mathcal{X} \)?
  - Adaptive selection, e.g. gradient ascent with estimated gradients?

- An estimator \( \{(x_i, f(x_i))\}_{i=1}^n \mapsto \hat{x}^* \).
  - The empirical maximum \( \hat{x}^* = x_{i^*} \) where \( i^* = \arg \max_{i=1,\ldots,n} f(x_i) \)?
  - Something better?

- Key idea: Estimator uncertainty quantification!
Start with the data that have been collected:

\[ \mathcal{D} = \{(x_i, f(x_i))\}_{i=1}^n \]

Bayesian linear regression onto a basis \( \{\phi_i\}_{i=1}^m \):

\[ f(x) = \beta_1 \phi_1(x) + \cdots + \beta_m \phi_m(x) \]

with \( n \leq m \in \mathbb{N} \cup \{\infty\} \).

- Prior \( p(\beta_1, \ldots, \beta_m) \)
- Likelihood \( \prod_{i=1}^n \delta(f(x_i) - \beta_1 \phi_1(x_i) - \cdots - \beta_m \phi_m(x_i)) \)
- Posterior \( p(\beta_1, \ldots, \beta_n | \mathcal{D}) \)
- Posterior marginal \( p(x^* | \mathcal{D}) \)
Start with the data that have been collected:

\[ \mathcal{D} = \{(x_i, f(x_i))\}_{i=1}^{n} \]

Bayesian linear regression onto a basis \( \{\phi_i\}_{i=1}^{m} \):

\[ f(x) = \beta_1 \phi_1(x) + \cdots + \beta_m \phi_m(x) \]

with \( n \leq m \in \mathbb{N} \cup \{\infty\} \).

- Prior \( p(\beta_1, \ldots, \beta_m) \)
- Likelihood \( \prod_{i=1}^{n} \delta(f(x_i) - \beta_1 \phi_1(x_i) - \cdots - \beta_m \phi_m(x_i)) \)
- Posterior \( p(\beta_1, \ldots, \beta_n | \mathcal{D}) \)
- Posterior marginal \( p(x^* | \mathcal{D}) \)
Calculations for the conjugate set-up:

Let:

\[ \beta = (\beta_1, \ldots, \beta_m) \]

Prior:

\[ \beta | \lambda \sim \mathcal{N}(0, \lambda I) \quad \text{and} \quad \lambda \sim \text{p}(\lambda) \propto \lambda^{\gamma} \]

Posterior:

\[ \beta | D \sim \text{MVT} \left( (I + \Phi^\top \Phi)^{-1} \Phi^\top f, \frac{1}{n} \right) \]

where \( f = (f(x_1), \ldots, f(x_n)) \) and \( \Phi_{ij} = \phi_j(x_i) \).

Posterior marginal:

\[ x^* | D \sim \,? \]

Draw \( \beta \) from \( \beta | D \)
Evaluate \( x^* = \arg \max \beta_1 \phi_1(x) + \cdots + \beta_m \phi_m(x) \)
Repeat.
Probabilistic Optimisation

Calculations for the conjugate set-up:

- Let: $\beta = (\beta_1, \ldots, \beta_m)$
Calculations for the conjugate set-up:

- Let: $\beta = (\beta_1, \ldots, \beta_m)$
- Prior:

\[\begin{align*}
\beta | \lambda & \sim N(0, \lambda I) \quad \text{and} \\
\lambda & \sim p(\lambda) \propto \lambda^{-1}.
\end{align*}\]
Probabilistic Optimisation

Calculations for the conjugate set-up:

- Let: \( \beta = (\beta_1, \ldots, \beta_m) \)
- Prior: \( \beta|\lambda \sim N(0, \lambda I) \) and \( \lambda \sim p(\lambda) \propto \lambda^{-1} \).

Let: \( \beta = (\beta_1, \ldots, \beta_m) \)

Prior: \( \beta|\lambda \sim N(0, \lambda I) \) and \( \lambda \sim p(\lambda) \propto \lambda^{-1} \).
Calculations for the conjugate set-up:

- Let: $\beta = (\beta_1, \ldots, \beta_m)$
- Prior: $\beta|\lambda \sim N(0, \lambda I)$ and $\lambda \sim p(\lambda) \propto \lambda^{-1}$.
- Posterior:
Calculations for the conjugate set-up:

- Let: $\beta = (\beta_1, \ldots, \beta_m)$
- Prior: $\beta|\lambda \sim N(0, \lambda I)$ and $\lambda \sim p(\lambda) \propto \lambda^{-1}$.
- Posterior:

$$
\beta|\mathcal{D} \sim \text{MVT} \left( (I + \Phi^\top \Phi)^{-1} \Phi^\top f, \frac{1}{n}(f^\top \Phi^\top (I + \Phi \Phi^\top)^{-1} \Phi f)(I + \Phi^\top \Phi)^{-1}, n \right)
$$

where $f = (f(x_1), \ldots, f(x_n))$ and $[\Phi]_{ij} = \phi_j(x_i)$. 

Calculations for the conjugate set-up:

- Let: $\beta = (\beta_1, \ldots, \beta_m)$
- Prior: $\beta | \lambda \sim N(0, \lambda I)$ and $\lambda \sim p(\lambda) \propto \lambda^{-1}$.
- Posterior:

$$
\beta | D \sim \text{MVT} \left( (I + \Phi^T \Phi)^{-1} \Phi^T f, \frac{1}{n} (f^T \Phi^T (I + \Phi \Phi^T)^{-1} \Phi f)(I + \Phi^T \Phi)^{-1}, n \right)
$$

where $f = (f(x_1), \ldots, f(x_n))$ and $[\Phi]_{ij} = \phi_j(x_i)$.
- Posterior marginal: $x^* | D \sim ?$
Calculations for the conjugate set-up:

- Let: $\beta = (\beta_1, \ldots, \beta_m)$
- Prior: $\beta | \lambda \sim N(0, \lambda I)$ and $\lambda \sim p(\lambda) \propto \lambda^{-1}$.
- Posterior:

$$
\beta | D \sim \text{MVT}\left((I + \Phi^\top \Phi)^{-1} \Phi^\top f, \frac{1}{n}(f^\top \Phi^\top (I + \Phi \Phi^\top)^{-1} \Phi f)(I + \Phi^\top \Phi)^{-1}, n\right)
$$

where $f = (f(x_1), \ldots, f(x_n))$ and $[\Phi]_{ij} = \phi_j(x_i)$.

- Posterior marginal: $x^* | D \sim ?$
  - Draw $\beta$ from $\beta | D$
  - Evaluate $x^* = \arg\max \beta_1 \phi_1(x) + \cdots + \beta_m \phi_m(x)$
  - Repeat.
Compute $x^* = \arg \max f(x)$:
Some Discussion Points

- Close connection between statistics and design of numerical optimisation methods.
- Similar to “Bayesian optimisation” (Mockus, 1989).
- Kernel trick maps to Gaussian processes.
- The distributional output $p(x^*|D)$ provides uncertainty quantification.
- Propagation and the Bayesian mantra of Dawid.
- Numerical analysts want to consider order of convergence and constants (of the point estimator).
- Similar considerations relevant to posterior contraction.
Some Discussion Points

- Close connection between statistics and design of numerical optimisation methods.
- Similar to “Bayesian optimisation” (Mockus, 1989).
- Kernel trick maps to Gaussian processes.
- The distributional output \( p(x^*|D) \) provides uncertainty quantification.
- Propagation and the Bayesian mantra of Dawid.
- Numerical analysts want to consider order of convergence and constants (of the point estimator).
- Similar considerations relevant to posterior contraction.
Some Discussion Points

- Close connection between statistics and design of numerical optimisation methods.
- Similar to “Bayesian optimisation” (Mockus, 1989).
  - Kernel trick maps to Gaussian processes.
  - The distributional output $p(x^*|D)$ provides uncertainty quantification.
  - Propagation and the Bayesian mantra of Dawid.
  - Numerical analysts want to consider order of convergence and constants (of the point estimator).
- Similar considerations relevant to posterior contraction.
Some Discussion Points

- Close connection between statistics and design of numerical optimisation methods.
- Similar to “Bayesian optimisation” (Mockus, 1989).
- Kernel trick maps to Gaussian processes.
  - The distributional output $p(x^*|D)$ provides uncertainty quantification.
  - Propagation and the Bayesian mantra of Dawid.
  - Numerical analysts want to consider order of convergence and constants (of the point estimator).
- Similar considerations relevant to posterior contraction.
Some Discussion Points

- Close connection between statistics and design of numerical optimisation methods.
- Similar to “Bayesian optimisation” (Mockus, 1989).
- Kernel trick maps to Gaussian processes.
- The distributional output $p(x^*|D)$ provides uncertainty quantification.
-Propagation and the Bayesian mantra of Dawid.
- Numerical analysts want to consider order of convergence and constants (of the point estimator).
- Similar considerations relevant to posterior contraction.
Some Discussion Points

- Close connection between statistics and design of numerical optimisation methods.
- Similar to “Bayesian optimisation” (Mockus, 1989).
- Kernel trick maps to Gaussian processes.
- The distributional output $p(x^*|D)$ provides uncertainty quantification.
- Propagation and the Bayesian mantra of Dawid.
  - Numerical analysts want to consider order of convergence and constants (of the point estimator).
  - Similar considerations relevant to posterior contraction.
Some Discussion Points

- Close connection between statistics and design of numerical optimisation methods.
- Similar to “Bayesian optimisation” (Mockus, 1989).
- Kernel trick maps to Gaussian processes.
- The distributional output $p(x^*|\mathcal{D})$ provides uncertainty quantification.
- Propagation and the Bayesian mantra of Dawid.
- Numerical analysts want to consider order of convergence and constants (of the point estimator).
- Similar considerations relevant to posterior contraction.
Some Discussion Points

- Close connection between statistics and design of numerical optimisation methods.
- Similar to “Bayesian optimisation” (Mockus, 1989).
- Kernel trick maps to Gaussian processes.
- The distributional output $p(x^*|\mathcal{D})$ provides uncertainty quantification.
- Propagation and the Bayesian mantra of Dawid.
- Numerical analysts want to consider order of convergence and constants (of the point estimator).
- Similar considerations relevant to posterior contraction.
Application

(but the MATLAB function doesn’t provide uncertainty quantification!)
Integration
$$I = \int f(x) \pi(x) \, dx$$

- **Well-defined:**
  - $f \in L_2(\pi)$?
  - $f \in C^\alpha(\mathcal{X})$ for some $\alpha \geq 0$ and $x \in \mathcal{X}$ a compact subset of $\mathbb{R}^d$.

- **Well-posed:**
  - Allowed $n$ evaluations of $f(\cdot)$ at inputs which you can select.
  - Aim to minimise $|\hat{I} - I|$.
Integration

\[ I = \int f(x)\pi(x)dx \]

- **Well-defined:**
  - \( f \in L_2(\pi) \)?
  - \( f \in C^\alpha(\mathcal{X}) \) for some \( \alpha \geq 0 \) and \( x \in \mathcal{X} \) a compact subset of \( \mathbb{R}^d \).

- **Well-posed:**
  - Allowed \( n \) evaluations of \( f(\cdot) \) at inputs which you can select.
  - Aim to minimise \( |\hat{I} - I| \).
Integration

\[ I = \int f(x) \pi(x) \, dx \]

- **Well-defined:**
  - \( f \in L_2(\pi) \)?
  - \( f \in C^\alpha(\mathcal{X}) \) for some \( \alpha \geq 0 \) and \( x \in \mathcal{X} \) a compact subset of \( \mathbb{R}^d \).

- **Well-posed:**
  - Allowed \( n \) evaluations of \( f(\cdot) \) at inputs which you can select.
  - Aim to minimise \( |\hat{I} - I| \).
Integration

\[ I = \int f(x)\pi(x)dx \]

- **Well-defined:**
  - \( f \in L_2(\pi) \)?
  - \( f \in C^\alpha(\mathcal{X}) \) for some \( \alpha \geq 0 \) and \( x \in \mathcal{X} \) a compact subset of \( \mathbb{R}^d \).

- **Well-posed:**
  - Allowed \( n \) evaluations of \( f(\cdot) \) at inputs which you can select.
  - Aim to minimise \( |\hat{I} - I| \).
Integration

\[ I = \int f(x) \pi(x) \, dx \]

- **Well-defined:**
  - \( f \in L_2(\pi) \)?
  - \( f \in C^\alpha(\mathcal{X}) \) for some \( \alpha \geq 0 \) and \( x \in \mathcal{X} \) a compact subset of \( \mathbb{R}^d \).

- **Well-posed:**
  - Allowed \( n \) evaluations of \( f(\cdot) \) at inputs which you can select.
  - Aim to minimise \( |\hat{I} - I| \).
\[ I = \int f(x)\pi(x)\,dx \]

- **Well-defined:**
  - \( f \in L_2(\pi) \)?
  - \( f \in C^\alpha(\mathcal{X}) \) for some \( \alpha \geq 0 \) and \( x \in \mathcal{X} \), a compact subset of \( \mathbb{R}^d \).
- **Well-posed:**
  - Allowed \( n \) evaluations of \( f(\cdot) \) at inputs which you can select.
  - Aim to minimise \( |\hat{I} - I| \).
Integration

\[ I = \int f(x)\pi(x)dx \]

- **Well-defined:**
  - \( f \in L^2(\pi) \)?
  - \( f \in C^\alpha(\mathcal{X}) \) for some \( \alpha \geq 0 \) and \( x \in \mathcal{X} \) a compact subset of \( \mathbb{R}^d \).

- **Well-posed:**
  - Allowed \( n \) evaluations of \( f(\cdot) \) at inputs which you can select.
  - Aim to minimise \( |\hat{I} - I| \).
Integration

\[ I = \int f(x) \pi(x) dx \]

Two distinct requirements:

- A method to select the integrand evaluation locations \( x_1, \ldots, x_n \).
  - Uniform grid over \( \mathcal{X} \)?
  - Adaptive selection, e.g. based on local error indicators?

- An estimator \( \{(x_i, f(x_i))\}_{i=1}^n \mapsto \hat{I} \).
  - The arithmetic mean \( \hat{I} = \frac{1}{n} \sum_{i=1}^n f(x_i) \)?
  - Something better?

- Key idea (again): Estimator uncertainty quantification!
Two distinct requirements:

- A method to select the integrand evaluation locations $x_1, \ldots, x_n$.
  - Uniform grid over $\mathcal{X}$?
  - Adaptive selection, e.g. based on local error indicators?

- An estimator $\{(x_i, f(x_i))\}_{i=1}^n \mapsto \hat{I}$.
  - The arithmetic mean $\hat{I} = \frac{1}{n} \sum_{i=1}^n f(x_i)$?
  - Something better?

- Key idea (again): Estimator uncertainty quantification!
Integration

\[ I = \int f(x)\pi(x)dx \]

Two distinct requirements:

- A method to select the integrand evaluation locations \( x_1, \ldots, x_n \).
  - Uniform grid over \( \mathcal{X} \)?
  - Adaptive selection, e.g. based on local error indicators?

- An \underline{estimator} \( \{(x_i, f(x_i))\}_{i=1}^n \mapsto \hat{I} \).
  - The arithmetic mean \( \hat{I} = \frac{1}{n} \sum_{i=1}^{n} f(x_i) \)?
  - Something better?

- Key idea (again): Estimator uncertainty quantification!
Integration

\[ I = \int f(x) \pi(x) \, dx \]

Two distinct requirements:

- A method to select the integrand evaluation locations \( x_1, \ldots, x_n \).
  - Uniform grid over \( \mathcal{X} \)?
  - Adaptive selection, e.g. based on local error indicators?

- An estimator \( \{(x_i, f(x_i))\}_{i=1}^{n} \mapsto \hat{I} \).
  - The arithmetic mean \( \hat{I} = \frac{1}{n} \sum_{i=1}^{n} f(x_i) \)?
  - Something better?

- Key idea (again): Estimator uncertainty quantification!
Integration

\[ I = \int f(x) \pi(x) \, dx \]

Two distinct requirements:

- A method to select the integrand evaluation locations \( x_1, \ldots, x_n \).
  - Uniform grid over \( X \)?
  - Adaptive selection, e.g. based on local error indicators?

- An estimator \( \{(x_i, f(x_i))\}_{i=1}^n \mapsto \hat{I} \).
  - The arithmetic mean \( \hat{I} = \frac{1}{n} \sum_{i=1}^n f(x_i) \)?
  - Something better?

- Key idea (again): Estimator uncertainty quantification!
Integration

\[ I = \int f(x)\pi(x)dx \]

Two distinct requirements:

- A method to select the integrand evaluation locations \( x_1, \ldots, x_n \).
  - Uniform grid over \( \mathcal{X} \)?
  - Adaptive selection, e.g. based on local error indicators?
- An estimator \( \{(x_i, f(x_i))\}_{i=1}^n \mapsto \hat{I} \).
  - The arithmetic mean \( \hat{I} = \frac{1}{n} \sum_{i=1}^n f(x_i) \)?
  - Something better?
- Key idea (again): Estimator uncertainty quantification!
\[ l = \int f(x)\pi(x)dx \]

Two distinct requirements:

- A method to select the integrand evaluation locations \( x_1, \ldots, x_n \).
  - Uniform grid over \( \mathcal{X} \)?
  - Adaptive selection, e.g. based on local error indicators?

- An estimator \( \{(x_i, f(x_i))\}_{i=1}^n \mapsto \hat{l} \).
  - The arithmetic mean \( \hat{l} = \frac{1}{n} \sum_{i=1}^n f(x_i) \)?
  - Something better?

- Key idea (again): Estimator uncertainty quantification!
Integration

\[ I = \int f(x) \pi(x) \, dx \]

Two distinct requirements:

- A method to select the integrand evaluation locations \( x_1, \ldots, x_n \).
  - Uniform grid over \( \mathcal{X} \)?
  - Adaptive selection, e.g. based on local error indicators?

- An estimator \( \{ (x_i, f(x_i)) \}_{i=1}^{n} \mapsto \hat{I} \).
  - The arithmetic mean \( \hat{I} = \frac{1}{n} \sum_{i=1}^{n} f(x_i) \)?
  - Something better?

- Key idea (again): Estimator uncertainty quantification!
Start with the data that have been collected:

\[ \mathcal{D} = \{(x_i, f(x_i))\}_{i=1}^n \]

Bayesian linear regression onto a basis \( \{\phi_i\}_{i=1}^m \):

\[
    f(x) = \beta_1 \phi_1(x) + \cdots + \beta_m \phi_m(x)
\]

with \( n \leq m \in \mathbb{N} \cup \{\infty\} \).

- Prior \( p(\beta_1, \ldots, \beta_m) \)
- Likelihood \( \prod_{i=1}^n \delta(f(x_i) - \beta_1 \phi_1(x_i) - \cdots - \beta_m \phi_m(x_i)) \)
- Posterior \( p(\beta_1, \ldots, \beta_n|\mathcal{D}) \)
- Posterior marginal \( p(I|\mathcal{D}) \)
Start with the data that have been collected:

\[ \mathcal{D} = \{(x_i, f(x_i))\}_{i=1}^n \]

Bayesian linear regression onto a basis \( \{\phi_i\}_{i=1}^m \):

\[ f(x) = \beta_1 \phi_1(x) + \cdots + \beta_m \phi_m(x) \]

with \( n \leq m \in \mathbb{N} \cup \{ \infty \} \).

- Prior \( p(\beta_1, \ldots, \beta_m) \)
- Likelihood \( \prod_{i=1}^n \delta(f(x_i) - \beta_1 \phi_1(x_i) - \cdots - \beta_m \phi_m(x_i)) \)
- Posterior \( p(\beta_1, \ldots, \beta_n | \mathcal{D}) \)
- Posterior marginal \( p(\mathcal{I} | \mathcal{D}) \)
Calculations for the conjugate set-up:

Let:

\[ \beta = (\beta_1, \ldots, \beta_m) \]

Prior:

\[ \beta | \lambda \sim N(0, \lambda I) \] and \[ \lambda \sim p(\lambda) \propto \lambda^{-1} \].

Posterior:

\[ \beta | D \sim \text{MVT} \left( (I + \Phi^\top \Phi)^{-1} \Phi^\top f, \frac{1}{n}(f^\top \Phi^\top (I + \Phi \Phi^\top)^{-1} \Phi f) (I + \Phi^\top \Phi)^{-1} \right) \]

where \[ f = (f(x_1), \ldots, f(x_n)) \] and \[ \Phi_{ij} = \phi_j(x_i) \].

Posterior marginal:

\[ \lambda | D \sim \text{Student-T} \left( \Psi^\top (I + \Phi^\top \Phi)^{-1} \Phi^\top f, \frac{1}{n}(f^\top \Phi^\top (I + \Phi \Phi^\top)^{-1} \Phi f) \Psi^\top (I + \Phi^\top \Phi)^{-1} \Psi, n \right) \]

where \[ \Psi_i = \int \phi_i(x) \pi(x) \, dx \].

Chris. J. Oates
Probabilistic Numerical Computation
October 2017 @ Turing
Calculations for the conjugate set-up:

Let: $\mathbf{\beta} = (\beta_1, \ldots, \beta_m)$
Calculations for the conjugate set-up:

- Let: \( \mathbf{\beta} = (\beta_1, \ldots, \beta_m) \)
- Prior:

Prior: \( \mathbf{\beta} | \lambda \sim N(0, \lambda I) \) and \( \lambda \sim p(\lambda) \propto \lambda^{-1} \).

Posterior:

\[
\mathbf{\beta} | D \sim \text{MVT}(\left( I + \mathbf{\Phi}^\top \mathbf{\Phi} \right)^{-1} \mathbf{\Phi}^\top \mathbf{f}, \frac{1}{n} \left( \mathbf{f}^\top \mathbf{\Phi}^\top (I + \mathbf{\Phi} \mathbf{\Phi}^\top) \right)^{-1} \mathbf{\Phi}^\top \mathbf{f} \left( I + \mathbf{\Phi}^\top \mathbf{\Phi} \right)^{-1})
\]

where \( \mathbf{f} = (f(x_1), \ldots, f(x_n)) \) and \( \mathbf{\Phi}_{ij} = \phi_j(x_i) \).

Posterior marginal:

\[
\lambda | D \sim \text{Student-T}(\mathbf{\Psi}^\top \left( I + \mathbf{\Phi}^\top \mathbf{\Phi} \right)^{-1} \mathbf{\Phi}^\top \mathbf{f}, \frac{1}{n} \left( \mathbf{f}^\top \mathbf{\Phi}^\top (I + \mathbf{\Phi} \mathbf{\Phi}^\top) \right)^{-1} \mathbf{\Phi}^\top \mathbf{f} \mathbf{\Psi}^\top \left( I + \mathbf{\Phi}^\top \mathbf{\Phi} \right)^{-1} \mathbf{\Psi}, n)
\]

where \( \mathbf{\Psi}_{i} = \int \phi_i(x) \pi(x) \, dx \).
Probabilistic Integration

Calculations for the conjugate set-up:

- Let: $\mathbf{\beta} = (\beta_1, \ldots, \beta_m)$
- Prior: $\mathbf{\beta} | \lambda \sim \mathcal{N}(\mathbf{0}, \lambda \mathbf{I})$ and $\lambda \sim p(\lambda) \propto \lambda^{-1}$.
Calculations for the conjugate set-up:

- Let: $\beta = (\beta_1, \ldots, \beta_m)$
- Prior: $\beta | \lambda \sim N(0, \lambda I)$ and $\lambda \sim p(\lambda) \propto \lambda^{-1}$.
- Posterior:
Calculations for the conjugate set-up:

- Let: \( \beta = (\beta_1, \ldots, \beta_m) \)
- Prior: \( \beta | \lambda \sim \mathcal{N}(0, \lambda I) \) and \( \lambda \sim p(\lambda) \propto \lambda^{-1} \).
- Posterior:

\[
\beta | \mathcal{D} \sim \text{MVT} \left( (I + \Phi^\top \Phi)^{-1} \Phi^\top f, \frac{1}{n} (f^\top \Phi^\top (I + \Phi \Phi^\top)^{-1} \Phi f)(I + \Phi^\top \Phi)^{-1}, n \right)
\]

where \( f = (f(x_1), \ldots, f(x_n)) \) and \( [\Phi]_{ij} = \phi_j(x_i) \).
Calculations for the conjugate set-up:

- Let: $\beta = (\beta_1, \ldots, \beta_m)$
- Prior: $\beta | \lambda \sim N(0, \lambda I)$ and $\lambda \sim p(\lambda) \propto \lambda^{-1}$.
- Posterior:
  
  $$
  \beta | D \sim MVT \left( (I + \Phi^\top \Phi)^{-1} \Phi^\top f, \frac{1}{n} (f^\top \Phi^\top (I + \Phi \Phi^\top)^{-1} \Phi f)(I + \Phi^\top \Phi)^{-1}, n \right)
  $$

  where $f = (f(x_1), \ldots, f(x_n))$ and $[\Phi]_{ij} = \phi_j(x_i)$.
- Posterior marginal:
Calculations for the conjugate set-up:

- Let: $\beta = (\beta_1, \ldots, \beta_m)$
- Prior: $\beta|\lambda \sim N(0, \lambda I)$ and $\lambda \sim p(\lambda) \propto \lambda^{-1}$.
- Posterior:

$$
\beta | \mathcal{D} \sim \text{MVT} \left( (I + \Phi^\top \Phi)^{-1} \Phi^\top f, \frac{1}{n}(f^\top \Phi^\top (I + \Phi \Phi^\top)^{-1} \Phi f)(I + \Phi^\top \Phi)^{-1}, n \right)
$$

where $f = (f(x_1), \ldots, f(x_n))$ and $[\Phi]_{ij} = \phi_j(x_i)$.
- Posterior marginal:

$$
I | \mathcal{D} \sim \text{Student-T} \left( \Psi^\top (I + \Phi^\top \Phi)^{-1} \Phi^\top f, \frac{1}{n}(f^\top \Phi^\top (I + \Phi \Phi^\top)^{-1} \Phi f)\Psi^\top (I + \Phi^\top \Phi)^{-1} \Psi, n \right)
$$

where $[\Psi]_i = \int \phi_i(x) \pi(x) dx$. 
Compute $\int f(x) \pi(x) dx$:
Some Discussion Points

- Similar to “Bayesian quadrature” (O’Hagan, 1991).
- Kernel trick maps to GPs.
- Theoretical results were provided in Briol et al. 2016:
  - Posterior mean $\hat{I}$ satisfies $\hat{I} - I = O_P(n^{-\alpha/d+\epsilon})$.
  - Stronger assumptions on $f$, such as $f \in H^\alpha(0, 1) \otimes \cdots \otimes H^\alpha(0, 1)$, lead to $\hat{I} - I = O_P(n^{-\alpha+\epsilon})$ for an appropriately sparse basis $\{\phi_i\}_{i=1}^n$.
  - Posterior is concentrated on $\hat{I}$, so rates of contraction to $I$ can also be established.
- Posterior mean often coincides with standard quadrature methods (Särkkä et al., 2016).
- How to select the $\{x_i\}_{i=1}^n$?
  - Recall the posterior scale was determined by
    \[
    \frac{1}{n}(f^\top \Phi^\top (I + \Phi \Phi^\top)^{-1} \Phi f) \Psi^\top (I + \Phi^\top \Phi)^{-1} \Psi
    \]
  - Select $\{x_i\}_{i=1}^n$ to minimise
    \[
    (*) \quad \Psi^\top (I + \Phi^\top \Phi)^{-1} \Psi
    \]
  - Select $x_n$ to minimise $(*)$ based on $\{x_i\}_{i=1}^{n-1}$ fixed? (Similar to “sequential Bayesian quadrature”.)
Some Discussion Points

- Similar to “Bayesian quadrature” (O’Hagan, 1991).
- Kernel trick maps to GPs.
- Theoretical results were provided in Briol et al. 2016:
  - Posterior mean $\hat{I}$ satisfies $\hat{I} - I = O_P(n^{-\alpha/d+\epsilon})$.
  - Stronger assumptions on $f$, such as $f \in H^\alpha(0,1) \otimes \cdots \otimes H^\alpha(0,1)$, lead to $\hat{I} - I = O_P(n^{-\alpha+\epsilon})$ for an appropriately sparse basis $\{\phi_i\}_{i=1}^n$.
  - Posterior is concentrated on $\hat{I}$, so rates of contraction to $I$ can also be established.
- Posterior mean often coincides with standard quadrature methods (Särkkä et al., 2016).
- How to select the $\{x_i\}_{i=1}^n$?
  - Recall the posterior scale was determined by
    \[
    \frac{1}{n}(f^\top \Phi^\top (I + \Phi \Phi^\top)^{-1} \Phi f) \Psi^\top (I + \Phi^\top \Phi)^{-1} \Psi
    \]
  - Select $\{x_i\}_{i=1}^n$ to minimise
    \[
    (\ast) \quad \Psi^\top (I + \Phi^\top \Phi)^{-1} \Psi
    \]
  - Select $x_n$ to minimise (\ast) based on $\{x_i\}_{i=1}^{n-1}$ fixed? (Similar to “sequential Bayesian quadrature”.)
Some Discussion Points

- Similar to “Bayesian quadrature” (O’Hagan, 1991).
- Kernel trick maps to GPs.
- Theoretical results were provided in Briol et al. 2016:
  - Posterior mean $\hat{I}$ satisfies $\hat{I} - I = O_P(n^{-\alpha/d+\epsilon})$.
  - Stronger assumptions on $f$, such as $f \in H^\alpha(0, 1) \otimes \cdots \otimes H^\alpha(0, 1)$, lead to $\hat{I} - I = O_P(n^{-\alpha+\epsilon})$ for an appropriately sparse basis $\{\phi_i\}_{i=1}^n$.
  - Posterior is concentrated on $\hat{I}$, so rates of contraction to $I$ can also be established.
- Posterior mean often coincides with standard quadrature methods (Särkkä et al., 2016).
- How to select the $\{x_i\}_{i=1}^n$?
  - Recall the posterior scale was determined by
    \[
    \frac{1}{n}(f^\top \Phi^\top (I + \Phi \Phi^\top)^{-1} \Phi f) \Psi^\top (I + \Phi^\top \Phi)^{-1} \Psi
    \]
  - Select $\{x_i\}_{i=1}^n$ to minimise
    \[\ (*)\ \ \Psi^\top (I + \Phi^\top \Phi)^{-1} \Psi ?\]
  - Select $x_n$ to minimise $(*)$ based on $\{x_i\}_{i=1}^{n-1}$ fixed? (Similar to “sequential Bayesian quadrature”.)
Some Discussion Points

- Similar to “Bayesian quadrature” (O’Hagan, 1991).
- Kernel trick maps to GPs.
- Theoretical results were provided in Briol et al. 2016:
  - Posterior mean \( \hat{I} \) satisfies \( \hat{I} - I = O_P(n^{-\alpha/d+\epsilon}) \).
  - Stronger assumptions on \( f \), such as \( f \in H_\alpha(0, 1) \otimes \cdots \otimes H_\alpha(0, 1) \), lead to \( \hat{I} - I = O_P(n^{-\alpha+\epsilon}) \) for an appropriately sparse basis \( \{\phi_i\}_{i=1}^n \).
  - Posterior is concentrated on \( \hat{I} \), so rates of contraction to \( I \) can also be established.
- Posterior mean often coincides with standard quadrature methods (Särkkä et al., 2016).
- How to select the \( \{x_i\}_{i=1}^n \)?
  - Recall the posterior scale was determined by
    \[
    \frac{1}{n} (f^\top \Phi^\top (I + \Phi \Phi^\top)^{-1} \Phi f) \Psi^\top (I + \Phi^\top \Phi)^{-1} \Psi
    \]
  - Select \( \{x_i\}_{i=1}^n \) to minimise
    \[ (\star) \quad \Psi^\top (I + \Phi^\top \Phi)^{-1} \Psi \ ? \]
  - Select \( x_n \) to minimise \( (\star) \) based on \( \{x_i\}_{i=1}^{n-1} \) fixed? (Similar to “sequential Bayesian quadrature”.)
Some Discussion Points

- Similar to “Bayesian quadrature” (O’Hagan, 1991).
- Kernel trick maps to GPs.
- Theoretical results were provided in Briol et al. 2016:
  - Posterior mean $\hat{I}$ satisfies $\hat{I} - I = O_P(n^{-\alpha/d+\epsilon})$.
  - Stronger assumptions on $f$, such as $f \in H^\alpha(0, 1) \otimes \cdots \otimes H^\alpha(0, 1)$, lead to $\hat{I} - I = O_P(n^{-\alpha+\epsilon})$ for an appropriately sparse basis $\{\phi_i\}_{i=1}^n$.
  - Posterior is concentrated on $\hat{I}$, so rates of contraction to $I$ can also be established.
- Posterior mean often coincides with standard quadrature methods (Särkkä et al., 2016).
- How to select the $\{x_i\}_{i=1}^n$?
  - Recall the posterior scale was determined by
    \[
    \frac{1}{n}(f^\top \Phi^\top (I + \Phi \Phi^\top)^{-1} \Phi f) \Psi^\top (I + \Phi^\top \Phi)^{-1} \Psi
    \]
  - Select $\{x_i\}_{i=1}^n$ to minimise
    \[
    (\star) \quad \Psi^\top (I + \Phi^\top \Phi)^{-1} \Psi ?
    \]
  - Select $x_n$ to minimise (\star) based on $\{x_i\}_{i=1}^{n-1}$ fixed? (Similar to “sequential Bayesian quadrature”.)
Some Discussion Points

- Similar to “Bayesian quadrature” (O’Hagan, 1991).
- Kernel trick maps to GPs.
- Theoretical results were provided in Briol et al. 2016:
  - Posterior mean $\hat{I}$ satisfies $\hat{I} - I = O_P(n^{-\alpha/d+\epsilon})$.
  - Stronger assumptions on $f$, such as $f \in H^\alpha(0, 1) \otimes \cdots \otimes H^\alpha(0, 1)$, lead to $\hat{I} - I = O_P(n^{-\alpha+\epsilon})$ for an appropriately sparse basis $\{\phi_i\}_{i=1}^n$.
  - Posterior is concentrated on $\hat{I}$, so rates of contraction to $I$ can also be established.
- Posterior mean often coincides with standard quadrature methods (Särkkä et al., 2016).
- How to select the $\{x_i\}_{i=1}^n$?
  - Recall the posterior scale was determined by
    \[
    \frac{1}{n}(f^\top \Phi^\top (I + \Phi \Phi^\top)^{-1} \Phi f) \Psi^\top (I + \Phi^\top \Phi)^{-1} \Psi
    \]
  - Select $\{x_i\}_{i=1}^n$ to minimise
    \[
    (*) \quad \Psi^\top (I + \Phi^\top \Phi)^{-1} \Psi \ ?
    \]
  - Select $x_n$ to minimise $(*)$ based on $\{x_i\}_{i=1}^{n-1}$ fixed? (Similar to “sequential Bayesian quadrature”.)
Some Discussion Points

- Similar to “Bayesian quadrature” (O’Hagan, 1991).
- Kernel trick maps to GPs.
- Theoretical results were provided in Briol et al. 2016:
  - Posterior mean \( \hat{I} \) satisfies \( \hat{I} - I = O_P(n^{-\alpha/d+\epsilon}) \).
  - Stronger assumptions on \( f \), such as \( f \in H^\alpha(0,1) \otimes \cdots \otimes H^\alpha(0,1) \), lead to \( \hat{I} - I = O_P(n^{-\alpha+\epsilon}) \) for an appropriately sparse basis \( \{\phi_i\}_{i=1}^n \).
  - Posterior is concentrated on \( \hat{I} \), so rates of contraction to \( I \) can also be established.

- Posterior mean often coincides with standard quadrature methods (Särkkä et al., 2016).

- How to select the \( \{x_i\}_{i=1}^n \)?
  - Recall the posterior scale was determined by
  \[
  \frac{1}{n} (f^\top \Phi^\top (I + \Phi \Phi^\top)^{-1} \Phi f) \Psi^\top (I + \Phi^\top \Phi)^{-1} \Psi
  \]
  - Select \( \{x_i\}_{i=1}^n \) to minimise
  \[
  (*) \quad \Psi^\top (I + \Phi^\top \Phi)^{-1} \Psi \]
  - Select \( x_n \) to minimise \((*)\) based on \( \{x_i\}_{i=1}^{n-1} \) fixed? (Similar to “sequential Bayesian quadrature”.)
Some Discussion Points

- Similar to “Bayesian quadrature” (O’Hagan, 1991).
- Kernel trick maps to GPs.
- Theoretical results were provided in Briol et al. 2016:
  - Posterior mean \( \hat{I} \) satisfies \( \hat{I} - I = O_P(n^{-\alpha/d+\epsilon}) \).
  - Stronger assumptions on \( f \), such as \( f \in H^\alpha(0,1) \otimes \cdots \otimes H^\alpha(0,1) \), lead to \( \hat{I} - I = O_P(n^{-\alpha+\epsilon}) \) for an appropriately sparse basis \( \{\phi_i\}_{i=1}^n \).
  - Posterior is concentrated on \( \hat{I} \), so rates of contraction to \( I \) can also be established.
- Posterior mean often coincides with standard quadrature methods (Särkkä et al., 2016).

  How to select the \( \{x_i\}_{i=1}^n \)?
  - Recall the posterior scale was determined by
    \[
    \frac{1}{n} (f^\top \Phi^\top (I + \Phi \Phi^\top)^{-1} \Phi f) \Psi^\top (I + \Phi^\top \Phi)^{-1} \Psi
    \]
  - Select \( \{x_i\}_{i=1}^n \) to minimise
    \[
    (\star) \quad \Psi^\top (I + \Phi^\top \Phi)^{-1} \Psi
    \]
  - Select \( x_n \) to minimise (\( \star \)) based on \( \{x_i\}_{i=1}^{n-1} \) fixed? (Similar to “sequential Bayesian quadrature”.)
Similar to “Bayesian quadrature” (O’Hagan, 1991).

Kernel trick maps to GPs.

Theoretical results were provided in Briol et al. 2016:

- Posterior mean $\hat{I}$ satisfies $\hat{I} - I = O_P(n^{-\alpha/d + \epsilon})$.
- Stronger assumptions on $f$, such as $f \in H^\alpha(0, 1) \otimes \cdots \otimes H^\alpha(0, 1)$, lead to $\hat{I} - I = O_P(n^{-\alpha + \epsilon})$ for an appropriately sparse basis $\{\phi_i\}_{i=1}^n$.
- Posterior is concentrated on $\hat{I}$, so rates of contraction to $I$ can also be established.

Posterior mean often coincides with standard quadrature methods (Särkkä et al., 2016).

How to select the $\{x_i\}_{i=1}^n$?

- Recall the posterior scale was determined by

$$\frac{1}{n}(f^\top \Phi^\top (I + \Phi \Phi^\top)^{-1} \Phi f) \Psi^\top (I + \Phi^\top \Phi)^{-1} \Psi$$

- Select $\{x_i\}_{i=1}^n$ to minimise

$$(*) \quad \Psi^\top (I + \Phi^\top \Phi)^{-1} \Psi$$

- Select $x_n$ to minimise $(*)$ based on $\{x_i\}_{i=1}^{n-1}$ fixed? (Similar to “sequential Bayesian quadrature”.)
Some Discussion Points

- Similar to “Bayesian quadrature” (O’Hagan, 1991).
- Kernel trick maps to GPs.
- Theoretical results were provided in Briol et al. 2016:
  - Posterior mean \( \hat{I} \) satisfies \( \hat{I} - I = O_P(n^{-\alpha/d+\epsilon}) \).
  - Stronger assumptions on \( f \), such as \( f \in H^\alpha(0,1) \otimes \cdots \otimes H^\alpha(0,1) \), lead to \( \hat{I} - I = O_P(n^{-\alpha+\epsilon}) \) for an appropriately sparse basis \( \{\phi_i\}_{i=1}^n \).
  - Posterior is concentrated on \( \hat{I} \), so rates of contraction to \( I \) can also be established.
- Posterior mean often coincides with standard quadrature methods (Särkkä et al., 2016).
- How to select the \( \{x_i\}_{i=1}^n \)?
  - Recall the posterior scale was determined by
    \[
    \frac{1}{n}(f^\top \Phi^\top (I + \Phi \Phi^\top)^{-1} \Phi f) \Psi^\top (I + \Phi^\top \Phi)^{-1} \Psi
    \]
  - Select \( \{x_i\}_{i=1}^n \) to minimise (\( \ast \)) based on \( \{x_i\}_{i=1}^{n-1} \) fixed? (Similar to “sequential Bayesian quadrature”.)
Some Discussion Points

- Similar to “Bayesian quadrature” (O’Hagan, 1991).
- Kernel trick maps to GPs.
- Theoretical results were provided in Briol et al. 2016:
  - Posterior mean $\hat{I}$ satisfies $\hat{I} - I = O_P(n^{-\alpha/d+\epsilon})$.
  - Stronger assumptions on $f$, such as $f \in H^\alpha(0, 1) \otimes \cdots \otimes H^\alpha(0, 1)$, lead to $\hat{I} - I = O_P(n^{-\alpha+\epsilon})$ for an appropriately sparse basis $\{\phi_i\}_{i=1}^n$.
  - Posterior is concentrated on $\hat{I}$, so rates of contraction to $I$ can also be established.

- Posterior mean often coincides with standard quadrature methods (Särkkä et al., 2016).
- How to select the $\{x_i\}_{i=1}^n$?
  - Recall the posterior scale was determined by
    $$\frac{1}{n}(f^\top \Phi^\top (I + \Phi \Phi^\top)^{-1} \Phi f) \Psi^\top (I + \Phi^\top \Phi)^{-1} \Psi$$
  - Select $\{x_i\}_{i=1}^n$ to minimise
    $$(\star) \quad \Psi^\top (I + \Phi^\top \Phi)^{-1} \Psi$$
  - Select $x_n$ to minimise $\star$ based on $\{x_i\}_{i=1}^{n-1}$ fixed? (Similar to “sequential Bayesian quadrature”.)
Some Discussion Points

- Similar to “Bayesian quadrature” (O’Hagan, 1991).
- Kernel trick maps to GPs.
- Theoretical results were provided in Briol et al. 2016:
  - Posterior mean $\hat{I}$ satisfies $\hat{I} - I = O_P(n^{-\alpha/d+\epsilon})$.
  - Stronger assumptions on $f$, such as $f \in H^{\alpha}(0,1) \otimes \cdots \otimes H^{\alpha}(0,1)$, lead to $\hat{I} - I = O_P(n^{-\alpha+\epsilon})$ for an appropriately sparse basis $\{\phi_i\}_{i=1}^n$.
  - Posterior is concentrated on $\hat{I}$, so rates of contraction to $I$ can also be established.
- Posterior mean often coincides with standard quadrature methods (Särkkä et al., 2016).
- How to select the $\{x_i\}_{i=1}^n$?
  - Recall the posterior scale was determined by
    $$\frac{1}{n}(f^\top \Phi^\top (I + \Phi \Phi^\top)^{-1} \Phi f) \Psi^\top (I + \Phi^\top \Phi)^{-1} \Psi$$
  - Select $\{x_i\}_{i=1}^n$ to minimise
    $$(*) \quad \Psi^\top (I + \Phi^\top \Phi)^{-1} \Psi ?$$
  - Select $x_n$ to minimise $(*)$ based on $\{x_i\}_{i=1}^{n-1}$ fixed? (Similar to “sequential Bayesian quadrature”.)
Some Discussion Points

From Briol et al., 2016:

Figure 2: Illustration of states used for quadrature, based on a Gaussian mixture $\Pi$.

Left: Monte Carlo (MC) sampling from $\Pi$.
Middle: A Sobol sequence, a specific type of Quasi-Monte Carlo (QMC) point sequence, mapped to $\Pi$.
Right: States from an experimental design scheme based on the Frank-Wolfe (FW) algorithm. Estimators based on QMC or FW typically outperform estimators based on MC due to their better coverage of $\Pi$.

A related class of methods is Quasi Monte Carlo (QMC) (Hickernell, 1998). These methods exploit knowledge of the RKHS $H$ to spread the states in an efficient, deterministic way over the domain $X$. QMC also approximates integrals using a quadrature rule $\hat{\Pi}_{\text{QMC}}[f]$ that has uniform weights $w_{\text{QMC}}^i := 1/n$. These methods benefit from an extensive theoretical literature (Dick and Pillichshammer, 2010). The (in some cases) optimal convergence rates as well as sound statistical properties of QMC have recently led to interest within the statistics community (e.g. Hickernell et al., 2005; Gerber and Chopin, 2015; Oates et al., 2016c).

2.4.2 Experimental Design Schemes
An Optimal BQ (OBQ) rule selects states $\{x_i\}_{i=1}^n$ to globally minimise the posterior variance (equivalent to globally minimising the WCE). Sarkka et al. (2016) recently showed that OBQ corresponds to classical quadrature rules (e.g. Gauss-Hermite) for specific choices of covariance function $k$. Indeed, the average case analysis literature (Ritter, 2000) contains upper and lower bounds for the WCE that map directly onto statements about convergence rates for OBQ as $n \to \infty$. However OBQ can generally not be implemented; the problem of finding optimal states is in general NP-hard (Schölkopf and Smola, 2002, Sec. 10.2.3).

A more pragmatic approach to select states is using experimental design methods, such as the greedy algorithm sequentially minimising the posterior variance at each iteration. This rule, called Sequential BQ (SBQ), is straightforward to implement, e.g. using general-purpose numerical optimisation, and is a probabilistic integration method that is often used in practice (Osborne et al., 2012; Gunter et al., 2014). More sophisticated optimisation algorithms have also been used to select states. For example, in the QMC literature Nuyens and Cools (2006) framed the construction of lattice rules as an optimisation problem, in

One of these is (a variant on) sequential Bayesian quadrature - which one?
From Briol et al., 2016.
Linear Algebra
\[ Ax = b \]

- **Well-defined:**
  - \( A \) is a \( N \times N \) symmetric positive definite matrix.

- **Well-posed:**
  - Allowed \( n \ll N \) matrix-vector multiplications.
  - Represented as \( [s_i^\top A]x = s_i^\top b \) for \( i = 1, \ldots, n \).
  - You are allowed to select the directions \( s_1, \ldots, s_n \).
  - Aim to minimise \( \| \hat{x} - x \|_A \) where \( \| z \|_A = \sqrt{z^\top A z} \).
\[ \mathbf{A} \mathbf{x} = \mathbf{b} \]

- **Well-defined:**
  - \( \mathbf{A} \) is a \( N \times N \) symmetric positive definite matrix.

- **Well-posed:**
  - Allowed \( n \ll N \) matrix-vector multiplications.
  - Represented as \( [\mathbf{s}_i^\top \mathbf{A}] \mathbf{x} = \mathbf{s}_i^\top \mathbf{b} \) for \( i = 1, \ldots, n \).
  - You are allowed to select the directions \( \mathbf{s}_1, \ldots, \mathbf{s}_n \).
  - Aim to minimise \( \| \hat{\mathbf{x}} - \mathbf{x} \|_A \) where \( \| \mathbf{z} \|_A = \sqrt{\mathbf{z}^\top \mathbf{A} \mathbf{z}} \).
Well-defined:
- \( A \) is a \( N \times N \) symmetric positive definite matrix.

Well-posed:
- Allowed \( n \ll N \) matrix-vector multiplications.
- Represented as \([s_i^\top A]x = s_i^\top b\) for \( i = 1, \ldots, n \).
- You are allowed to select the directions \( s_1, \ldots, s_n \).
- Aim to minimise \( \|\hat{x} - x\|_A \) where \( \|z\|_A = \sqrt{z^\top A z} \).
\[
Ax = b
\]

- **Well-defined:**
  - \( A \) is a \( N \times N \) symmetric positive definite matrix.

- **Well-posed:**
  - Allowed \( n \ll N \) matrix-vector multiplications.
  - \( A \) \( i \) matrix-vector multiplications.
  - \( \text{Represented as } [s_i^\top A]x = s_i^\top b \text{ for } i = 1, \ldots, n. \)
  - You are allowed to select the directions \( s_1, \ldots, s_n. \)
  - Aim to minimise \( \|\hat{x} - x\|_A \text{ where } \|z\|_A = \sqrt{z^\top Az}. \)
Well-defined:

- $A$ is a $N \times N$ symmetric positive definite matrix.

Well-posed:

- Allowed $n \ll N$ matrix-vector multiplications.
  - Represented as $[s_i^\top A]x = s_i^\top b$ for $i = 1, \ldots, n$.
  - You are allowed to select the directions $s_1, \ldots, s_n$.
  - Aim to minimise $\|\hat{x} - x\|_A$ where $\|z\|_A = \sqrt{z^\top Az}$. 

$$Ax = b$$
\[ Ax = b \]

- **Well-defined:**
  - \( A \) is a \( N \times N \) symmetric positive definite matrix.

- **Well-posed:**
  - Allowed \( n \ll N \) matrix-vector multiplications.
  - Represented as \([s_i^\top A]x = s_i^\top b\) for \( i = 1, \ldots, n \).
  - You are allowed to select the directions \( s_1, \ldots, s_n \).
  - Aim to minimise \( \|\hat{x} - x\|_A \) where \( \|z\|_A = \sqrt{z^\top Az} \).
Well-defined:

- $A$ is a $N \times N$ symmetric positive definite matrix.

Well-posed:

- Allowed $n \ll N$ matrix-vector multiplications.
- Represented as $[s_i^\top A]x = s_i^\top b$ for $i = 1, \ldots, n$.
- You are allowed to select the directions $s_1, \ldots, s_n$.
- Aim to minimise $\|\hat{x} - x\|_A$ where $\|z\|_A = \sqrt{z^\top Az}$.

$$Ax = b$$
Well-defined:  
- \( A \) is a \( N \times N \) symmetric positive definite matrix.

Well-posed:  
- Allowed \( n \ll N \) matrix-vector multiplications.
- Represented as \( [s_i^\top A]x = s_i^\top b \) for \( i = 1, \ldots, n \).
- You are allowed to select the directions \( s_1, \ldots, s_n \).
- Aim to minimise \( \|x - \hat{x}\|_A \) where \( \|z\|_A = \sqrt{z^\top A z} \).
Two distinct requirements:

- A method to select the directions $s_1, \ldots, s_n$.
  - Random projections?
  - Sequential selection, e.g. gradient descent or conjugate gradient?

- An estimator $\{(s_i^T A, s_i^T b)\}_{i=1}^n \mapsto \hat{x}$.
  - A minimal $\| \cdot \|_2$ norm vector that satisfies $s_i^T A \hat{x} = s_i^T b$ for $i = 1, \ldots, n$?
  - Something better?

- Key idea (again): Estimator uncertainty quantification!
Two distinct requirements:

- A method to select the directions $s_1, \ldots, s_n$.
  - Random projections?
  - Sequential selection, e.g. gradient descent or conjugate gradient?
- An estimator $\{(s_i^\top A, s_i^\top b)\}_{i=1}^n \mapsto \hat{x}$.
  - A minimal $\| \cdot \|_2$ norm vector that satisfies $s_i^\top A\hat{x} = s_i^\top b$ for $i = 1, \ldots, n$?
  - Something better?
- Key idea (again): Estimator uncertainty quantification!
Ax = b

Two distinct requirements:

- A method to select the directions $s_1, \ldots, s_n$.
  - Random projections?
  - Sequential selection, e.g. gradient descent or conjugate gradient?

- An estimator $\{(s_i^\top A, s_i^\top b)\}_{i=1}^n \mapsto \hat{x}$.
  - A minimal $\| \cdot \|_2$ norm vector that satisfies $s_i^\top A\hat{x} = s_i^\top b$ for $i = 1, \ldots, n$?
  - Something better?

- Key idea (again): Estimator uncertainty quantification!
Linear Algebra

\[ Ax = b \]

Two distinct requirements:

- A method to select the directions \( s_1, \ldots, s_n \).
  - Random projections?
  - Sequential selection, e.g. gradient descent or conjugate gradient?

- An estimator \( \{ (s_i^\top A, s_i^\top b) \}_{i=1}^n \mapsto \hat{x} \).
  - A minimal \( \| \cdot \|_2 \) norm vector that satisfies \( s_i^\top A\hat{x} = s_i^\top b \) for \( i = 1, \ldots, n \)?
  - Something better?

- Key idea (again): Estimator uncertainty quantification!
Two distinct requirements:

- A method to select the directions $s_1, \ldots, s_n$.
  - Random projections?
  - Sequential selection, e.g. gradient descent or conjugate gradient?

- An estimator $\{(s_i^\top A, s_i^\top b)\}_{i=1}^n \mapsto \hat{x}$.
  - A minimal $\| \cdot \|_2$ norm vector that satisfies $s_i^\top A\hat{x} = s_i^\top b$ for $i = 1, \ldots, n$?
  - Something better?

- Key idea (again): Estimator uncertainty quantification!
Linear Algebra

Ax = b

Two distinct requirements:

- A method to select the directions $s_1, \ldots, s_n$.
  - Random projections?
  - Sequential selection, e.g. gradient descent or conjugate gradient?

- An estimator $\{(s_i^\top A, s_i^\top b)\}_{i=1}^n \mapsto \hat{x}$.
  - A minimal $\| \cdot \|_2$ norm vector that satisfies $s_i^\top A \hat{x} = s_i^\top b$ for $i = 1, \ldots, n$?
  - Something better?

- Key idea (again): Estimator uncertainty quantification!
Two distinct requirements:

- A method to select the directions $s_1, \ldots, s_n$.
  - Random projections?
  - Sequential selection, e.g. gradient descent or conjugate gradient?

- An estimator $\{(s_i^T A, s_i^T b)\}_{i=1}^n \mapsto \hat{x}$.
  - A minimal $\| \cdot \|_2$ norm vector that satisfies $s_i^T A \hat{x} = s_i^T b$ for $i = 1, \ldots, n$?
  - Something better?

- Key idea (again): Estimator uncertainty quantification!
Linear Algebra

Ax = b

Two distinct requirements:
- A method to select the directions $s_1, \ldots, s_n$.
  - Random projections?
  - Sequential selection, e.g. gradient descent or conjugate gradient?

- An estimator $\{(s_i^\top A, s_i^\top b)\}_{i=1}^n \mapsto \hat{x}$.
  - A minimal $\| \cdot \|_2$ norm vector that satisfies $s_i^\top A\hat{x} = s_i^\top b$ for $i = 1, \ldots, n$?
  - Something better?

- Key idea (again): Estimator uncertainty quantification!
Let $\langle \cdot , \cdot \rangle_A$ be the inner-product induced by $\| \cdot \|_A$. (i.e. $\langle z, \tilde{z} \rangle_A = z^\top A \tilde{z}$.)
Recap: Conjugate Gradient Method

Let $\langle \cdot, \cdot \rangle_A$ be the inner-product induced by $\| \cdot \|_A$. (i.e. $\langle z, \tilde{z} \rangle_A = z^\top A \tilde{z}$.)

Call $z, \tilde{z} \in \mathbb{R}^N$ conjugate (w.r.t. $A$) if $\langle z, \tilde{z} \rangle_A = 0$. 

Suppose that we have a conjugate basis \{s_1, \ldots, s_N\} for $\mathbb{R}^N$ (i.e. $\langle s_i, s_j \rangle_A = 0$ for all $i \neq j$).

Consider the natural sequence of approximations

$\hat{x}_n = \sum_{i=1}^n \alpha_i s_i$

where each $\alpha_i = \langle s_i, x \rangle_A / \langle s_i, s_i \rangle_A = s_i^\top b s_i$. The total computational cost is $O(N^2)$.

So what is needed to proceed?

Need a smart choice of \{s_1, \ldots, s_N\}.

For theory, need to bound $\| x - \hat{x}_n \|_{\| \cdot \|}$ in your favourite $\| \cdot \|$.
Recap: Conjugate Gradient Method

Let $\langle \cdot, \cdot \rangle_A$ be the inner-product induced by $\| \cdot \|_A$. (i.e. $\langle z, \tilde{z} \rangle_A = z^\top A \tilde{z}$.)

Call $z, \tilde{z} \in \mathbb{R}^N$ conjugate (w.r.t. $A$) if $\langle z, \tilde{z} \rangle_A = 0$.

Suppose that we have a conjugate basis $\{s_1, \ldots, s_N\}$ for $\mathbb{R}^N$ (i.e. $\langle s_i, s_j \rangle_A = 0$ for all $i \neq j$).
Recap: Conjugate Gradient Method

Let \( \langle \cdot, \cdot \rangle_A \) be the inner-product induced by \( \| \cdot \|_A \). (i.e. \( \langle z, \tilde{z} \rangle_A = z^\top A \tilde{z} \).)

Call \( z, \tilde{z} \in \mathbb{R}^N \) conjugate (w.r.t. \( A \)) if \( \langle z, \tilde{z} \rangle_A = 0 \).

Suppose that we have a conjugate basis \( \{ s_1, \ldots, s_N \} \) for \( \mathbb{R}^N \) (i.e. \( \langle s_i, s_j \rangle_A = 0 \) for all \( i \neq j \)).

Consider the natural sequence of approximations

\[
\hat{x}_n = \sum_{i=1}^{n} \alpha_i s_i
\]

where each

\[
\alpha_i = \frac{\langle s_i, x \rangle_A}{\langle s_i, s_i \rangle_A} = \frac{s_i^\top b}{s_i^\top A s_i}
\]

can be computed in \( O(N^2) \). The total computational cost is \( O(nN^2) \).
Recap: Conjugate Gradient Method

Let $\langle \cdot, \cdot \rangle_A$ be the inner-product induced by $\| \cdot \|_A$. (i.e. $\langle z, \tilde{z} \rangle_A = z^\top A \tilde{z}$.)

Call $z, \tilde{z} \in \mathbb{R}^N$ conjugate (w.r.t. $A$) if $\langle z, \tilde{z} \rangle_A = 0$.

Suppose that we have a conjugate basis $\{s_1, \ldots, s_N\}$ for $\mathbb{R}^N$ (i.e. $\langle s_i, s_j \rangle_A = 0$ for all $i \neq j$).

Consider the natural sequence of approximations

$$\hat{x}_n = \sum_{i=1}^{n} \alpha_i s_i$$

where each

$$\alpha_i = \frac{\langle s_i, x \rangle_A}{\langle s_i, s_i \rangle_A} = \frac{s_i^\top b}{s_i^\top As_i}$$

can be computed in $O(N^2)$. The total computational cost is $O(nN^2)$.

So what is needed to proceed?

- Need a smart choice of $\{s_1, \ldots, s_N\}$.
- For theory, need to bound $\| x - \hat{x}_n \| = \| \sum_{i=n+1}^{N} \alpha_i x_i \|$ in your favourite $\| \cdot \|$.
Recap: Conjugate Gradient Method

Aim is to adaptively select $s_n$ based on the computation up to iteration $n-1$.

Gradient Descent: Notice that $x$ is a minimum of

$$f(x) = \frac{1}{2} x^\top Ax - x^\top b.$$ 

This suggests to select $s_n = -\nabla f(\hat{x}_{n-1})$ which is equal to $r_{n-1} = b - A\hat{x}_{n-1}$. However, this does not ensure $\{s_1, \ldots, s_n\}$ is a conjugate set.

Conjugate Gradient: A more delicate procedure selects

$$s_n = r_{n-1} - \sum_{i<m} \frac{s_i^\top A r_{n-1}}{s_i^\top A s_i} s_i$$

i.e. gradient descent plus Gram-Schmidt orthogonalisation w.r.t $\langle \cdot, \cdot \rangle_A$ to subtract off components in the directions $\{s_1, \ldots, s_{n-1}\}$ already used.
Recap: Conjugate Gradient Method

Aim is to adaptively select $s_n$ based on the computation up to iteration $n - 1$.

**Gradient Descent:** Notice that $x$ is a minimum of

$$f(x) = \frac{1}{2} x^\top Ax - x^\top b.$$  

This suggests to select $s_n = -\nabla f(\hat{x}_{n-1})$ which is equal to $r_{n-1} = b - A\hat{x}_{n-1}$. However, this does not ensure $\{s_1, \ldots, s_n\}$ is a conjugate set.

**Conjugate Gradient:** A more delicate procedure selects

$$s_n = r_{n-1} - \sum_{i<m} s_i^\top A r_{n-1} s_i$$

i.e. gradient descent plus Gram-Schmidt orthogonalisation w.r.t $\langle \cdot, \cdot \rangle_A$ to subtract off components in the directions $\{s_1, \ldots, s_{n-1}\}$ already used.
Aim is to adaptively select \( s_n \) based on the computation up to iteration \( n-1 \).

**Gradient Descent**: Notice that \( x \) is a minimum of

\[
f(x) = \frac{1}{2} x^\top A x - x^\top b.
\]

This suggests to select \( s_n = -\nabla f(\hat{x}_{n-1}) \) which is equal to \( r_{n-1} = b - A\hat{x}_{n-1} \). However, this does not ensure \( \{s_1, \ldots, s_n\} \) is a conjugate set.

**Conjugate Gradient**: A more delicate procedure selects

\[
s_n = r_{n-1} - \sum_{i<m} \frac{s_i^\top A r_{n-1}}{s_i^\top A s_i} s_i
\]

i.e. gradient descent plus Gram-Schmidt orthogonalisation w.r.t \( \langle \cdot, \cdot \rangle_A \) to subtract off components in the directions \( \{s_1, \ldots, s_{n-1}\} \) already used.
Recap: Conjugate Gradient Method

Aim is to adaptively select $s_n$ based on the computation up to iteration $n - 1$.

**Gradient Descent:** Notice that $x$ is a minimum of

$$f(x) = \frac{1}{2} x^\top A x - x^\top b.$$ 

This suggests to select $s_n = -\nabla f(\hat{x}_{n-1})$ which is equal to $r_{n-1} = b - A\hat{x}_{n-1}$. However, this does not ensure $\{s_1, \ldots, s_n\}$ is a conjugate set.

**Conjugate Gradient:** A more delicate procedure selects

$$s_n = r_{n-1} - \sum_{i<m} \frac{s_i^\top A r_{n-1}}{s_i^\top A s_i} s_i$$

i.e. gradient descent plus Gram-Schmidt orthogonalisation w.r.t $\langle \cdot, \cdot \rangle_A$ to subtract off components in the directions $\{s_1, \ldots, s_{n-1}\}$ already used.

For **either** method, the computational cost of selecting $s_n$ is $O(N^2)$, so the overall computational overhead added is $O(nN^2)$; the same order as random projections.
Start with the data that have been collected:

\[ D = \{(s_i^\top A, s_i^\top b)\}_{i=1}^n \]

Deploy full Bayesian inference for \( x \):

- Prior \( p(x) \)
- Likelihood \( \prod_{i=1}^n \delta(s_i^\top Ax - s_i^\top b) \)
- Posterior \( p(x|D) \)
Start with the data that have been collected:

$$\mathcal{D} = \{(s_i^\top A, s_i^\top b)\}_{i=1}^n$$

Deploy full Bayesian inference for $x$:

- Prior $p(x)$
- Likelihood $\prod_{i=1}^n \delta(s_i^\top Ax - s_i^\top b)$
- Posterior $p(x|\mathcal{D})$
Calculations for the conjugate set-up:

Let:

\[ S = [s_1, \ldots, s_n]^\top \]

Prior:

\[ x | \lambda \sim N(0, \lambda I) \text{ and } \lambda \sim p(\lambda) \propto \lambda^{-1}. \]

Posterior:

\[ x | D \sim MVT \left( (I + A^\top S^\top S A)^{-1} A^\top S^\top S b, \frac{1}{n} \left( b^\top S^\top A^\top S^\top (I + S A A^\top S) - 1 S A S b \right) (I + A^\top S^\top S A)^{-1} \right) \]

This is for general \( S \).

For the conjugate gradient method applied to the preconditioned system

\[ A^\top A x = A^\top b \]

we have the orthogonality equation

\[ S A A^\top S^\top = I \]

and the above can be further simplified.
Calculations for the conjugate set-up:

- Let: $\mathbf{S} = [s_1, \ldots, s_n]^\top$

This is for general $\mathbf{S}$. For the conjugate gradient method applied to the pre-conditioned system $\mathbf{A}^\top \mathbf{A} \mathbf{x} = \mathbf{A}^\top \mathbf{b}$ we have the orthogonality equation $\mathbf{S} \mathbf{A} \mathbf{A}^\top \mathbf{S}^\top = \mathbf{I}$ and the above can be further simplified.
Calculations for the conjugate set-up:

- Let: \( \mathbf{S} = [s_1, \ldots, s_n]^\top \)
- Prior:
Calculations for the conjugate set-up:

- Let: $S = [s_1, \ldots, s_n]^\top$
- Prior: $x | \lambda \sim N(0, \lambda I)$ and $\lambda \sim p(\lambda) \propto \lambda^{-1}$. 

For the conjugate gradient method applied to the pre-conditioned system $A^\top Ax = A^\top b$ we have the orthogonality equation $SAA^\top S = I$ and the above can be further simplified.
Calculations for the conjugate set-up:

- Let: \( S = [s_1, \ldots, s_n]^\top \)
- Prior: \( x|\lambda \sim N(\mathbf{0}, \lambda I) \) and \( \lambda \sim p(\lambda) \propto \lambda^{-1} \).
- Posterior:
Calculations for the conjugate set-up:

- Let: \( S = [s_1, \ldots, s_n]^\top \)
- Prior: \( x|\lambda \sim \mathcal{N}(0, \lambda I) \) and \( \lambda \sim p(\lambda) \propto \lambda^{-1} \).
- Posterior:

\[
\begin{align*}
x|D & \sim \text{MVT} \left( (I + A^\top S^\top SA)^{-1} A^\top S^\top Sb, \\
&\quad \frac{1}{n} (b^\top S^\top A^\top S^\top (I + SAA^\top S)^{-1} SASb)(I + A^\top S^\top SA)^{-1}, n \right)
\end{align*}
\]
Calculations for the conjugate set-up:

- Let: \( S = [s_1, \ldots, s_n]^\top \)
- Prior: \( x|\lambda \sim N(0, \lambda I) \) and \( \lambda \sim p(\lambda) \propto \lambda^{-1} \).
- Posterior:

\[
x|D \sim \text{MVT} \left( (I + A^T S^T S A)^{-1} A^T S^T S b, \right.
\]

\[
\frac{1}{n} (b^T S^T A^T S^T (I + S A A^T S)^{-1} S A S b) (I + A^T S^T S A)^{-1}, n \right)
\]

- This is for general \( S \).
Calculations for the conjugate set-up:

- Let: \( S = [s_1, \ldots, s_n]^\top \)
- Prior: \( x | \lambda \sim N(0, \lambda I) \) and \( \lambda \sim p(\lambda) \propto \lambda^{-1} \).
- Posterior:

\[
x | D \sim \text{MVT} \left( (I + A^\top S^\top SA)^{-1} A^\top S^\top Sb, \right.
\]

\[
\frac{1}{n} (b^\top S^\top A^\top S^\top (I + SAA^\top S)^{-1} SASb) (I + A^\top S^\top SA)^{-1}, n \right)
\]

- This is for general \( S \).
- For the conjugate gradient method applied to the pre-conditioned system \( A^\top Ax = A^\top b \) we have the orthogonality equation \( SAA^\top S^\top = I \) and the above can be further simplified.
[no video for this one!]
Some Discussion Points

- Approximate linear solvers used extensively in engineering applications.
- Also relevant in statistics, e.g. simulation of spatial random fields.
- It turns out that the posterior mean in our construction coincides with the classical conjugate gradient (CG) method applied to the pre-conditioned system $A^T A x = A^T b$.
- Thus the classical error bounds for CG are inherited.
- The full posterior can be computed in $O(nN^2)$. 
Some Discussion Points

- Approximate linear solvers used extensively in engineering applications.
- Also relevant in statistics, e.g. simulation of spatial random fields.
- It turns out that the posterior mean in our construction coincides with the classical conjugate gradient (CG) method applied to the pre-conditioned system $A^T Ax = A^T b$.
- Thus the classical error bounds for CG are inherited.
- The full posterior can be computed in $O(nN^2)$. 
Approximate linear solvers used extensively in engineering applications.

Also relevant in statistics, e.g. simulation of spatial random fields.

It turns out that the posterior mean in our construction coincides with the classical conjugate gradient (CG) method applied to the pre-conditioned system $A^\top Ax = A^\top b$.

Thus the classical error bounds for CG are inherited.

The full posterior can be computed in $O(nN^2)$. 
Some Discussion Points

- Approximate linear solvers used extensively in engineering applications.
- Also relevant in statistics, e.g. simulation of spatial random fields.
- It turns out that the posterior mean in our construction coincides with the classical conjugate gradient (CG) method applied to the pre-conditioned system $A^T Ax = A^T b$.
- Thus the classical error bounds for CG are inherited.
- The full posterior can be computed in $O(nN^2)$. 

Approximate linear solvers used extensively in engineering applications.

Also relevant in statistics, e.g. simulation of spatial random fields.

It turns out that the posterior mean in our construction coincides with the classical conjugate gradient (CG) method applied to the pre-conditioned system $A^T A x = A^T b$.

Thus the classical error bounds for CG are inherited.

The full posterior can be computed in $O(nN^2)$. 
Some Discussion Points

- Approximate linear solvers used extensively in engineering applications.
- Also relevant in statistics, e.g. simulation of spatial random fields.
- It turns out that the posterior mean in our construction coincides with the classical conjugate gradient (CG) method applied to the pre-conditioned system
  \[ A^T A x = A^T b. \]
- Thus the classical error bounds for CG are inherited.
- The full posterior can be computed in \( O(nN^2) \).
Application

From McAdams et al., SIGGRAPH 2010:
From McAdams *et al.*, SIGGRAPH 2010:
From McAdams et al., SIGGRAPH 2010:
Solution of Differential Equations
Solution of Differential Equations

E.g.:

\[
\Delta u(x) = f(x), \quad x \in \mathcal{X}
\]
\[
u(x) = 0, \quad x \in \partial \mathcal{X}
\]

- **Well-defined:**
  - \( \mathcal{X} \subset \mathbb{R}^d \) be \( C^{1,1} \)
  - \( f \in L^p(\mathcal{X}), \ p > n/2 \)
  - Cor. 9.18 in Gilbarg and Trudinger ensures \( \exists! \) solution \( u \in W^{2,p}_{\text{loc}}(\mathcal{X}) \cap C^0(\mathcal{X} \cup \partial \mathcal{X}) \)

- **Well-posed:**
  - Allowed \( n \) evaluations of \( f(\cdot) \) at inputs \( x_1, \ldots, x_n \in \mathcal{X} \cup \partial \mathcal{X} \) which you can select.
  - Aim to minimise \( \int_{\mathcal{X}} \| \hat{u}(x) - u(x) \|_2^2 \, dx \).
Solution of Differential Equations

E.g.:

\[ \Delta u(x) = f(x), \quad x \in \mathcal{X} \]
\[ u(x) = 0, \quad x \in \partial \mathcal{X} \]

- **Well-defined:**
  - \( \mathcal{X} \subset \mathbb{R}^d \) be \( C^{1,1} \)
  - \( f \in L^p(\mathcal{X}), \ p > n/2 \)
  - Cor. 9.18 in Gilbarg and Trudinger ensures \( \exists! \) solution \( u \in W^{2,p}_{\text{loc}}(\mathcal{X}) \cap C^0(\mathcal{X} \cup \partial \mathcal{X}) \)

- **Well-posed:**
  - Allowed \( n \) evaluations of \( f(\cdot) \) at inputs \( x_1, \ldots, x_n \in \mathcal{X} \cup \partial \mathcal{X} \) which you can select.
  - Aim to minimise \( \int_{\mathcal{X}} \| \hat{u}(x) - u(x) \|_2^2 \, dx \).
Solution of Differential Equations

E.g.:

\[ \Delta u(x) = f(x), \quad x \in \mathcal{X} \]
\[ u(x) = 0, \quad x \in \partial \mathcal{X} \]

- **Well-defined:**
  - \( \mathcal{X} \subset \mathbb{R}^d \) be \( C^{1,1} \)
  - \( f \in L^p(\mathcal{X}), \ p > n/2 \)
  - Cor. 9.18 in Gilbarg and Trudinger ensures \( \exists ! \) solution \( u \in W^{2,p}_{\text{loc}}(\mathcal{X}) \cap C^0(\mathcal{X} \cup \partial \mathcal{X}) \)

- **Well-posed:**
  - Allowed \( n \) evaluations of \( f(\cdot) \) at inputs \( x_1, \ldots, x_n \in \mathcal{X} \cup \partial \mathcal{X} \) which you can select.
  - Aim to minimise \( \int_{\mathcal{X}} \| \hat{u}(x) - u(x) \|_2^2 \, dx \).
Solution of Differential Equations

E.g.:

\[ \Delta u(x) = f(x), \quad x \in \mathcal{X} \]
\[ u(x) = 0, \quad x \in \partial \mathcal{X} \]

**Well-defined:**
- \( \mathcal{X} \subset \mathbb{R}^d \) be \( C^{1,1} \)
- \( f \in L^p(\mathcal{X}), \) \( p > n/2 \)
- Cor. 9.18 in Gilbarg and Trudinger ensures \( \exists! \) solution \( u \in W^{2,p}_{\text{loc}}(\mathcal{X}) \cap C^0(\mathcal{X} \cup \partial \mathcal{X}) \)

**Well-posed:**
- Allowed \( n \) evaluations of \( f(\cdot) \) at inputs \( x_1, \ldots, x_n \in \mathcal{X} \cup \partial \mathcal{X} \) which you can select.
- Aim to minimise \( \int_{\mathcal{X}} \| \hat{u}(x) - u(x) \|_2^2 \, dx \).
Solution of Differential Equations

E.g.:

\[
\Delta u(x) = f(x), \quad x \in \mathcal{X} \\
u(x) = 0, \quad x \in \partial \mathcal{X}
\]

- **Well-defined:**
  - $\mathcal{X} \subset \mathbb{R}^d$ be $C^{1,1}$
  - $f \in L^p(\mathcal{X}), \ p > n/2$
  - Cor. 9.18 in Gilbarg and Trudinger ensures $\exists!$ solution $u \in W^{2,p}_{\text{loc}}(\mathcal{X}) \cap C^0(\mathcal{X} \cup \partial \mathcal{X})$

- **Well-posed:**
  - Allowed $n$ evaluations of $f(\cdot)$ at inputs $x_1, \ldots, x_n \in \mathcal{X} \cup \partial \mathcal{X}$ which you can select.
  - Aim to minimise $\int_{\mathcal{X}} \| \hat{u}(x) - u(x) \|_2^2 \, dx$. 

Chris. J. Oates
Solution of Differential Equations

E.g.:

\[ \Delta u(x) = f(x), \quad x \in \mathcal{X} \]
\[ u(x) = 0, \quad x \in \partial \mathcal{X} \]

- **Well-defined:**
  - \( \mathcal{X} \subset \mathbb{R}^d \) be \( C^{1,1} \)
  - \( f \in L^p(\mathcal{X}), \quad p > n/2 \)
  - Cor. 9.18 in Gilbarg and Trudinger ensures \( \exists! \) solution \( u \in W^{2,p}_{loc}(\mathcal{X}) \cap C^0(\mathcal{X} \cup \partial \mathcal{X}) \)

- **Well-posed:**
  - Allowed \( n \) evaluations of \( f(\cdot) \) at inputs \( x_1, \ldots, x_n \in \mathcal{X} \cup \partial \mathcal{X} \) which you can select.
  - Aim to minimise \( \int_{\mathcal{X}} \| \hat{u}(x) - u(x) \|^2 \, dx \).
Solution of Differential Equations

\[ \Delta u(x) = f(x), \quad x \in X \]
\[ u(x) = 0, \quad x \in \partial X \]

Two distinct requirements:
- A method to select the function evaluation locations \( x_1, \ldots, x_n \in X \cup \partial X \).
  - Corners of a mesh on \( X \cup \partial X \)?
  - An adaptive method?
- An estimator \( \{(x_i, f(x_i))\}_{i=1}^n \mapsto \hat{u}(\cdot) \).
  - Linear interpolation of the \( f(x_i) \) and then solution of the PDE?
  - Something better?
- Key idea (again): Estimator uncertainty quantification!
\[ \Delta u(x) = f(x), \quad x \in \mathcal{X} \]
\[ u(x) = 0, \quad x \in \partial \mathcal{X} \]

Two distinct requirements:

- A method to select the function evaluation locations \( x_1, \ldots, x_n \in \mathcal{X} \cup \partial \mathcal{X} \).
  - Corners of a mesh on \( \mathcal{X} \cup \partial \mathcal{X} \)?
  - An adaptive method?
- An estimator \( \{(x_i, f(x_i))\}_{i=1}^n \mapsto \hat{u}(\cdot) \).
  - Linear interpolation of the \( f(x_i) \) and then solution of the PDE?
  - Something better?
- Key idea (again): Estimator uncertainty quantification!
\[ \Delta u(x) = f(x), \quad x \in \mathcal{X} \]
\[ u(x) = 0, \quad x \in \partial \mathcal{X} \]

Two distinct requirements:
- A method to select the function evaluation locations \( x_1, \ldots, x_n \in \mathcal{X} \cup \partial \mathcal{X} \).
  - Corners of a mesh on \( \mathcal{X} \cup \partial \mathcal{X} \)?
  - An adaptive method?
- An estimator \( \{(x_i, f(x_i))\}_{i=1}^n \mapsto \hat{u}(\cdot) \).
  - Linear interpolation of the \( f(x_i) \) and then solution of the PDE?
  - Something better?
- Key idea (again): Estimator uncertainty quantification!
Solution of Differential Equations

\[ \Delta u(x) = f(x), \quad x \in X \]
\[ u(x) = 0, \quad x \in \partial X \]

Two distinct requirements:
- A method to select the function evaluation locations \( x_1, \ldots, x_n \in X \cup \partial X \).
  - Corners of a mesh on \( X \cup \partial X \)?
  - An adaptive method?
- An estimator \( \{ (x_i, f(x_i)) \}_{i=1}^n \mapsto \hat{u}(\cdot) \).
  - Linear interpolation of the \( f(x_i) \) and then solution of the PDE?
  - Something better?
- Key idea (again): Estimator uncertainty quantification!
Solution of Differential Equations

\[ \Delta u(x) = f(x), \quad x \in \mathcal{X} \]

\[ u(x) = 0, \quad x \in \partial \mathcal{X} \]

Two distinct requirements:

- A method to select the function evaluation locations \( x_1, \ldots, x_n \in \mathcal{X} \cup \partial \mathcal{X} \).
  - Corners of a mesh on \( \mathcal{X} \cup \partial \mathcal{X} \)?
  - An adaptive method?
- An estimator \( \{(x_i, f(x_i))\}_{i=1}^n \mapsto \hat{u}(\cdot) \).
  - Linear interpolation of the \( f(x_i) \) and then solution of the PDE?
  - Something better?
- Key idea (again): Estimator uncertainty quantification!
\[ \Delta u(x) = f(x), \quad x \in \mathcal{X} \]
\[ u(x) = 0, \quad x \in \partial \mathcal{X} \]

Two distinct requirements:

- A method to select the function evaluation locations \( x_1, \ldots, x_n \in \mathcal{X} \cup \partial \mathcal{X} \).
  - Corners of a mesh on \( \mathcal{X} \cup \partial \mathcal{X} \)?
  - An adaptive method?
- An estimator \( \{(x_i, f(x_i))\}_{i=1}^n \mapsto \hat{u}(\cdot) \).
  - Linear interpolation of the \( f(x_i) \) and then solution of the PDE?
  - Something better?
- Key idea (again): Estimator uncertainty quantification!
Solution of Differential Equations

\[ \Delta u(x) = f(x), \quad x \in \mathcal{X} \]
\[ u(x) = 0, \quad x \in \partial \mathcal{X} \]

Two distinct requirements:
- A method to select the function evaluation locations \( x_1, \ldots, x_n \in \mathcal{X} \cup \partial \mathcal{X} \).
  - Corners of a mesh on \( \mathcal{X} \cup \partial \mathcal{X} \)?
  - An adaptive method?
- An estimator \( \{(x_i, f(x_i))\}_{i=1}^{n} \mapsto \hat{u}(\cdot) \).
  - Linear interpolation of the \( f(x_i) \) and then solution of the PDE?
  - Something better?
- Key idea (again): Estimator uncertainty quantification!
\[ \Delta u(x) = f(x), \quad x \in \mathcal{X} \]
\[ u(x) = 0, \quad x \in \partial \mathcal{X} \]

Two distinct requirements:
- A method to select the function evaluation locations \( x_1, \ldots, x_n \in \mathcal{X} \cup \partial \mathcal{X} \).
  - Corners of a mesh on \( \mathcal{X} \cup \partial \mathcal{X} \)?
  - An adaptive method?
- An estimator \( \{(x_i, f(x_i))\}_{i=1}^n \mapsto \hat{u}(\cdot) \).
  - Linear interpolation of the \( f(x_i) \) and then solution of the PDE?
  - Something better?
- Key idea (again): Estimator uncertainty quantification!
Start with the data that have been collected:

\[ D = \{(x_i, f(x_i))\}_{i=1}^{n} \]

Bayesian linear regression onto a basis \( \{\phi_i\}_{i=1}^{m} \):

\[ f(x) = \beta_1 \phi_1(x) + \cdots + \beta_m \phi_m(x) \]

with \( n \leq m \in \mathbb{N} \cup \{\infty\} \).

- Prior \( p(\beta_1, \ldots, \beta_m) \)
- Likelihood \( \prod_{i=1}^{n} \delta(f(x_i) - \beta_1 \phi_1(x_i) - \cdots - \beta_m \phi_m(x_i)) \)
- Posterior \( p(\beta_1, \ldots, \beta_n | D) \)
- Posterior marginal \( p(u(\cdot) | D) \)
Start with the data that have been collected:

\[ \mathcal{D} = \{(x_i, f(x_i))\}_{i=1}^n \]

Bayesian linear regression onto a basis \( \{\phi_i\}_{i=1}^m \):

\[ f(x) = \beta_1 \phi_1(x) + \cdots + \beta_m \phi_m(x) \]

with \( n \leq m \in \mathbb{N} \cup \{\infty\} \).

- Prior \( p(\beta_1, \ldots, \beta_m) \)
- Likelihood \( \prod_{i=1}^n \delta(f(x_i) - \beta_1 \phi_1(x_i) - \cdots - \beta_m \phi_m(x_i)) \)
- Posterior \( p(\beta_1, \ldots, \beta_n | \mathcal{D}) \)
- Posterior marginal \( p(u(\cdot) | \mathcal{D}) \)
Calculations for the conjugate set-up:

\[
\begin{align*}
\beta &= (\beta_1, \ldots, \beta_m) \\
\text{Prior:} & \quad \beta | \lambda \sim N(0, \lambda I) \quad \text{and} \quad \lambda \sim p(\lambda) \propto \lambda^{-1} \\
\text{Posterior:} & \quad \beta | D \sim \text{MVT} \left( (I + \Phi^\top \Phi)^{-1} \Phi^\top f, \frac{1}{n} \right) \\
\text{Posterior marginal:} & \quad u(x) | D \sim \text{Student-T} \left( U(x) \left( I + \Phi^\top \Phi \right)^{-1} \Phi^\top f, \frac{1}{n} \right) \\
\end{align*}
\]

where \( f = (f(x_1), \ldots, f(x_n)) \) and \( \Phi_{ij} = \phi_j(x_i) \).

\[ U(x) \]

\[ u_i \]

\[ \Delta u = \phi_i \quad \text{on X and} \quad u = 0 \quad \text{on } \partial X. \]  

[N.B. Don’t need to explicitly compute the \( \phi_i \) if you have the Green’s function of the PDE.]
Calculations for the conjugate set-up:

- Let: $\beta = (\beta_1, \ldots, \beta_m)$
Calculations for the conjugate set-up:

- Let: \( \beta = (\beta_1, \ldots, \beta_m) \)
- Prior:

\[
\beta | \lambda \sim N(0, \lambda I) \text{ and } \\
\lambda \sim p(\lambda) \propto \lambda^{-1}.
\]

Posterior:

\[
\beta | D \sim MVT \left( (I + \Phi^\top \Phi)^{-1} \Phi^\top f, \frac{1}{n} \right)
\]

where \( f = (f(x_1), \ldots, f(x_n)) \) and \( \Phi_{ij} = \phi_j(x_i) \).

Posterior marginal:

\[
u(x) | D \sim \text{Student-T} \left( U(x)^\top (I + \Phi^\top \Phi)^{-1} \Phi^\top f, \frac{1}{n} \right)
\]

where \( U(x)_i = \nu_i(x) \) and \( \nu_i \) solves \( \Delta \nu_i = \phi_i \) on \( X \) and \( \nu = 0 \) on \( \partial X \). [N.B. Don't need to explicitly compute the \( \phi_i \) if you have the Green's function of the PDE.]
Calculations for the conjugate set-up:

- Let: \( \beta = (\beta_1, \ldots, \beta_m) \)
- Prior: \( \beta|\lambda \sim N(0, \lambda I) \) and \( \lambda \sim p(\lambda) \propto \lambda^{-1} \).
Calculations for the conjugate set-up:

- Let: $\beta = (\beta_1, \ldots, \beta_m)$
- Prior: $\beta | \lambda \sim N(0, \lambda I)$ and $\lambda \sim p(\lambda) \propto \lambda^{-1}$.
- Posterior:
Calculations for the conjugate set-up:

- Let: \( \beta = (\beta_1, \ldots, \beta_m) \)
- Prior: \( \beta|\lambda \sim N(0, \lambda I) \) and \( \lambda \sim p(\lambda) \propto \lambda^{-1} \).
- Posterior:

\[
\beta|D \sim \text{MVT} \left( \left( I + \Phi^T \Phi \right)^{-1} \Phi^T f, \frac{1}{n} (f^T \Phi^T (I + \Phi \Phi^T)^{-1} \Phi f) (I + \Phi^T \Phi)^{-1}, n \right)
\]

where \( f = (f(x_1), \ldots, f(x_n)) \) and \([\Phi]_{ij} = \phi_j(x_i)\).
Calculations for the conjugate set-up:

- Let: $\beta = (\beta_1, \ldots, \beta_m)$
- Prior: $\beta|\lambda \sim N(0, \lambda I)$ and $\lambda \sim p(\lambda) \propto \lambda^{-1}$.
- Posterior:

$$\beta|\mathcal{D} \sim \text{MVT} \left( (I + \Phi^{\top} \Phi)^{-1} \Phi^{\top} f, \frac{1}{n} (f^{\top} \Phi^{\top} (I + \Phi \Phi^{\top})^{-1} \Phi f)(I + \Phi^{\top} \Phi)^{-1}, n \right)$$

where $f = (f(x_1), \ldots, f(x_n))$ and $[\Phi]_{ij} = \phi_j(x_i)$.
- Posterior marginal:
Calculations for the conjugate set-up:

- Let: $\beta = (\beta_1, \ldots, \beta_m)$
- Prior: $\beta|\lambda \sim N(0, \lambda I)$ and $\lambda \sim p(\lambda) \propto \lambda^{-1}$.
- Posterior:

$$
\beta|\mathcal{D} \sim \text{MVT} \left( (I + \Phi^\top \Phi)^{-1} \Phi^\top f, \frac{1}{n} (f^\top \Phi^\top (I + \Phi \Phi^\top)^{-1} \Phi f)(I + \Phi^\top \Phi)^{-1}, n \right)
$$

where $f = (f(x_1), \ldots, f(x_n))$ and $[\Phi]_{ij} = \phi_j(x_i)$.

- Posterior marginal:

$$
u(x)|\mathcal{D} \sim \text{Student-T} \left( U(x)^\top (I + \Phi^\top \Phi)^{-1} \Phi^\top f, \right.$$

$$\frac{1}{n} (f^\top \Phi^\top (I + \Phi \Phi^\top)^{-1} \Phi f) U(x)^\top (I + \Phi^\top \Phi)^{-1} U(x), n \right)
$$

where $[U(x)]_i = u_i(x)$ and $u_i$ solves $\Delta u = \phi_i$ on $\mathcal{X}$ and $u = 0$ on $\partial \mathcal{X}$. [N.B. Don’t need to explicitly compute the $\phi_i$ if you have the Green’s function of the PDE.]
Solve the ODE \( \frac{du}{dx} = f(x), \ u(0) = u_0 \) on \( x \in [0, T] \):
Some Discussion Points

- Posterior mean coincides with a classical “collocation” method.
- Generalises to GPs with the kernel trick.
- Theoretical results (for a method based on GPs) in Cockayne et al., 2016:
  - The posterior mean converges in $\| \cdot \|_\infty$ at a rate
    \[ O(h^{\alpha - \rho - \frac{d}{2}}). \]
  - The posterior mass for a ball of radius $\epsilon$ centred on the true solution $u(\cdot)$ scales as
    \[ 1 - O\left( \frac{h^{2\alpha - 2\rho - d}}{\epsilon} \right). \]
• Posterior mean coincides with a classical “collocation” method.

• Generalises to GPs with the kernel trick.

• Theoretical results (for a method based on GPs) in Cockayne et al., 2016:
  • The posterior mean converges in $\| \cdot \|_\infty$ at a rate
    $$O(h^{\alpha - \rho - \frac{d}{2}}).$$

  • The posterior mass for a ball of radius $\epsilon$ centred on the true solution $u(\cdot)$ scales as
    $$1 - O\left(\frac{h^{2\alpha - 2\rho - d}}{\epsilon}\right).$$
Some Discussion Points

- Posterior mean coincides with a classical “collocation” method.
- Generalises to GPs with the kernel trick.
- Theoretical results (for a method based on GPs) in Cockayne et al., 2016:
  - The posterior mean converges in $\| \cdot \|_\infty$ at a rate
    
    $$O(h^{\alpha - \rho - d/2}).$$
  
  - The posterior mass for a ball of radius $\epsilon$ centred on the true solution $u(\cdot)$ scales as
    
    $$1 - O\left(\frac{h^{2\alpha - 2\rho - d}}{\epsilon}\right).$$
Some Discussion Points

- Posterior mean coincides with a classical “collocation” method.
- Generalises to GPs with the kernel trick.
- Theoretical results (for a method based on GPs) in Cockayne et al., 2016:
  - The posterior mean converges in $\| \cdot \|_\infty$ at a rate
    \[ O(h^{\alpha-\rho-d/2}). \]
  - The posterior mass for a ball of radius $\epsilon$ centred on the true solution $u(\cdot)$ scales as
    \[ 1 - O\left(\frac{h^{2\alpha-2\rho-d}}{\epsilon}\right). \]
• Posterior mean coincides with a classical “collocation” method.
• Generalises to GPs with the kernel trick.
• Theoretical results (for a method based on GPs) in Cockayne et al., 2016:
  • The posterior mean converges in $\| \cdot \|_\infty$ at a rate
    \[ O(h^{\alpha-\rho-rac{d}{2}}). \]
  • The posterior mass for a ball of radius $\epsilon$ centred on the true solution $u(\cdot)$ scales as
    \[ 1 - O\left(\frac{h^{2\alpha-2\rho-d}}{\epsilon}\right). \]
Posterior mean coincides with a classical “collocation” method.
Generalises to GPs with the kernel trick.
Theoretical results (for a method based on GPs) in Cockayne et al., 2016:
- The posterior mean converges in $\| \cdot \|_\infty$ at a rate
  \[ O(h^{\alpha - \rho - \frac{d}{2}}). \]
- The posterior mass for a ball of radius $\epsilon$ centred on the true solution $u(\cdot)$ scales as
  \[ 1 - O\left( \frac{h^{2\alpha - 2\rho - d}}{\epsilon} \right). \]
Summary
Theoretical Questions

- General theory?
- Beyond linear and Gaussian assumptions?
- Experimental design?

Lots of work to do, but initial results in:

Cockayne J, Oates CJ, Sullivan T, Girolami M
Bayesian Probabilistic Numerical Methods
Computational Questions

- Propagation of uncertainty through a computational workflow?
- Compatibility of multiple probabilistic numerical methods?

The sophistication and scale of modern computer models creates an urgent need to better understand the propagation and accumulation of numerical error within arbitrary - often large - pipelines of computation, so that “numerical risk” to end-users can be controlled.