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?

Probabilistic Numerical Methods

The goal:

$$\text{Numerical Task} \implies \text{Finite Computation} \implies \underline{\text{Distribution}} \text{ on Output}$$

Optimisation

Optimisation

$$\boldsymbol{x}^* = \arg\max f(\boldsymbol{x})$$

- Well-defined:
  - $f \in C^\alpha(\mathcal{X})$ for some $\alpha \geq 0$ and $\boldsymbol{x} \in \mathcal{X}$ a compact subset of $\mathbb{R}^d$.
- Well-posed:
  - Allowed n evaluations of $f(\cdot)$ at inputs which you can select.
  - Aim to minimise $\|\hat{\boldsymbol{x}}^* - \boldsymbol{x}^*\|_2$.

Optimisation

$$\boldsymbol{x}^* = \arg\max f(\boldsymbol{x})$$

- Well-defined:
  - $f \in C^\alpha(\mathcal{X})$ for some $\alpha \geq 0$ and $\boldsymbol{x} \in \mathcal{X}$ a compact subset of $\mathbb{R}^d$.
- Well-posed:
  - Allowed n evaluations of $f(\cdot)$ at inputs which you can select.
  - Aim to minimise $\|\hat{\boldsymbol{x}}^* - \boldsymbol{x}^*\|_2$.

Optimisation

$$\boldsymbol{x}^* = \arg\max f(\boldsymbol{x})$$

- Well-defined:
  - $f \in C^\alpha(\mathcal{X})$ for some $\alpha \geq 0$ and $\boldsymbol{x} \in \mathcal{X}$ a compact subset of $\mathbb{R}^d$.
- Well-posed:
  - Allowed n evaluations of $f(\cdot)$ at inputs which you can select.
  - Aim to minimise $\|\hat{\boldsymbol{x}}^* - \boldsymbol{x}^*\|_2$.

Optimisation

$$\boldsymbol{x}^* = \arg\max f(\boldsymbol{x})$$

- Well-defined:
  - $f \in C^\alpha(\mathcal{X})$ for some $\alpha \geq 0$ and $\boldsymbol{x} \in \mathcal{X}$ a compact subset of $\mathbb{R}^d$.
- Well-posed:
  - Allowed n evaluations of $f(\cdot)$ at inputs which you can select.
  - Aim to minimise $\|\hat{\boldsymbol{x}}^* - \boldsymbol{x}^*\|_2$.

Optimisation

$$\boldsymbol{x}^* = \arg\max f(\boldsymbol{x})$$

- Well-defined:
  - $f \in C^\alpha(\mathcal{X})$ for some $\alpha \geq 0$ and $\boldsymbol{x} \in \mathcal{X}$ a compact subset of $\mathbb{R}^d$.
- Well-posed:
  - Allowed n evaluations of $f(\cdot)$ at inputs which you can select.
  - Aim to minimise $\|\hat{\boldsymbol{x}}^* - \boldsymbol{x}^*\|_2$.

Optimisation

$$\boldsymbol{x}^* = \arg\max f(\boldsymbol{x})$$

- Well-defined:
  - $f \in C^\alpha(\mathcal{X})$ for some $\alpha \geq 0$ and $\boldsymbol{x} \in \mathcal{X}$ a compact subset of $\mathbb{R}^d$.
- Well-posed:
  - Allowed n evaluations of $f(\cdot)$ at inputs which you can select.
  - Aim to minimise $\|\hat{\boldsymbol{x}}^* - \boldsymbol{x}^*\|_2$.

Optimisation

$$\boldsymbol{x}^* = \arg\max f(\boldsymbol{x})$$

Two distinct requirements:

- A method to select the function evaluation locations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$.
  - Uniform grid over $\mathcal{X}$?
  - Adaptive selection, e.g. gradient ascent with estimated gradients?
- An estimator $\{(\boldsymbol{x}_i, f(\boldsymbol{x}_i))\}_{i=1}^n \mapsto \hat{\boldsymbol{x}}^*$.
  - The empirical maximum $\hat{\boldsymbol{x}}^* = \boldsymbol{x}_{i^*}$ where $i^* = \arg\max_{i=1,\ldots,n} f(\boldsymbol{x}_i)$?
  - Something better?
- Key idea: Estimator uncertainty quantification!

Optimisation

$$\boldsymbol{x}^* = \arg\max f(\boldsymbol{x})$$

Two distinct requirements:

- A method to select the function evaluation locations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$.
  - Uniform grid over $\mathcal{X}$?
  - Adaptive selection, e.g. gradient ascent with estimated gradients?
- An estimator $\{(\boldsymbol{x}_i, f(\boldsymbol{x}_i))\}_{i=1}^n \mapsto \hat{\boldsymbol{x}}^*$.
  - The empirical maximum $\hat{\boldsymbol{x}}^* = \boldsymbol{x}_{i^*}$ where $i^* = \arg\max_{i=1,\ldots,n} f(\boldsymbol{x}_i)$?
  - Something better?
- Key idea: Estimator uncertainty quantification!

Optimisation

$$\boldsymbol{x}^* = \arg\max f(\boldsymbol{x})$$

Two distinct requirements:

- A method to select the function evaluation locations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$.
  - Uniform grid over $\mathcal{X}$?
  - Adaptive selection, e.g. gradient ascent with estimated gradients?
- An estimator $\{(\boldsymbol{x}_i, f(\boldsymbol{x}_i))\}_{i=1}^n \mapsto \hat{\boldsymbol{x}}^*$.
  - The empirical maximum $\hat{\boldsymbol{x}}^* = \boldsymbol{x}_{i^*}$ where $i^* = \arg\max_{i=1,\ldots,n} f(\boldsymbol{x}_i)$?
  - 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!

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!

Optimisation

$$\boldsymbol{x}^* = \arg\max f(\boldsymbol{x})$$

Two distinct requirements:

- A method to select the function evaluation locations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$.
  - Uniform grid over $\mathcal{X}$?
  - Adaptive selection, e.g. gradient ascent with estimated gradients?
- An estimator $\{(\boldsymbol{x}_i, f(\boldsymbol{x}_i))\}_{i=1}^n \mapsto \hat{\boldsymbol{x}}^*$.
  - The empirical maximum $\hat{\boldsymbol{x}}^* = \boldsymbol{x}_{i^*}$ where $i^* = \arg\max_{i=1,\ldots,n} f(\boldsymbol{x}_i)$?
  - Something better?
- Key idea: Estimator uncertainty quantification!

Optimisation

$$\boldsymbol{x}^* = \arg\max f(\boldsymbol{x})$$

Two distinct requirements:

- A method to select the function evaluation locations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$.
  - Uniform grid over $\mathcal{X}$?
  - Adaptive selection, e.g. gradient ascent with estimated gradients?
- An estimator $\{(\boldsymbol{x}_i, f(\boldsymbol{x}_i))\}_{i=1}^n \mapsto \hat{\boldsymbol{x}}^*$.
  - The empirical maximum $\hat{\boldsymbol{x}}^* = \boldsymbol{x}_{i^*}$ where $i^* = \arg\max_{i=1,\ldots,n} f(\boldsymbol{x}_i)$?
  - Something better?
- Key idea: Estimator uncertainty quantification!

Optimisation

$$\boldsymbol{x}^* = \arg\max f(\boldsymbol{x})$$

Two distinct requirements:

- A method to select the function evaluation locations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$.
  - Uniform grid over $\mathcal{X}$?
  - Adaptive selection, e.g. gradient ascent with estimated gradients?
- An estimator $\{(\boldsymbol{x}_i, f(\boldsymbol{x}_i))\}_{i=1}^n \mapsto \hat{\boldsymbol{x}}^*$.
  - The empirical maximum $\hat{\boldsymbol{x}}^* = \boldsymbol{x}_{i^*}$ where $i^* = \arg\max_{i=1,\ldots,n} f(\boldsymbol{x}_i)$?
  - Something better?
- Key idea: Estimator uncertainty quantification!

Probabilistic Optimisation

Start with the data that have been collected:

$$\mathcal{D} = \{(\boldsymbol{x}_i, f(\boldsymbol{x}_i))\}_{i=1}^n$$

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

$$f(\boldsymbol{x}) = \beta_1\phi_1(\boldsymbol{x}) + \cdots + \beta_m\phi_m(\boldsymbol{x})$$

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

- Prior $p(\beta_1, \ldots, \beta_m)$
- Likelihood $\prod_{i=1}^n \delta(f(\boldsymbol{x}_i) - \beta_1\phi_1(\boldsymbol{x}_i) - \cdots - \beta_m\phi_m(\boldsymbol{x}_i))$
- Posterior $p(\beta_1, \ldots, \beta_n|\mathcal{D})$
- Posterior marginal $p(\boldsymbol{x}^*|\mathcal{D})$

Probabilistic Optimisation

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 $p(x^* | \mathcal{D})$

Probabilistic Optimisation

Calculations for the conjugate set-up:

Probabilistic Optimisation

Calculations for the conjugate set-up:

- Let: $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_m)$

Probabilistic Optimisation

Calculations for the conjugate set-up:

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

Probabilistic Optimisation

Calculations for the conjugate set-up:

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

Probabilistic Optimisation

Calculations for the conjugate set-up:

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

Probabilistic Optimisation

Calculations for the conjugate set-up:

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

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

  where $\boldsymbol{f} = (f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_n))$ and $[\boldsymbol{\Phi}]_{ij} = \phi_j(\boldsymbol{x}_i)$.

Probabilistic Optimisation

Calculations for the conjugate set-up:

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

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

  where $\boldsymbol{f} = (f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_n))$ and $[\boldsymbol{\Phi}]_{ij} = \phi_j(\boldsymbol{x}_i)$.
- Posterior marginal: $\boldsymbol{x}^*|\mathcal{D} \sim$?

Probabilistic Optimisation

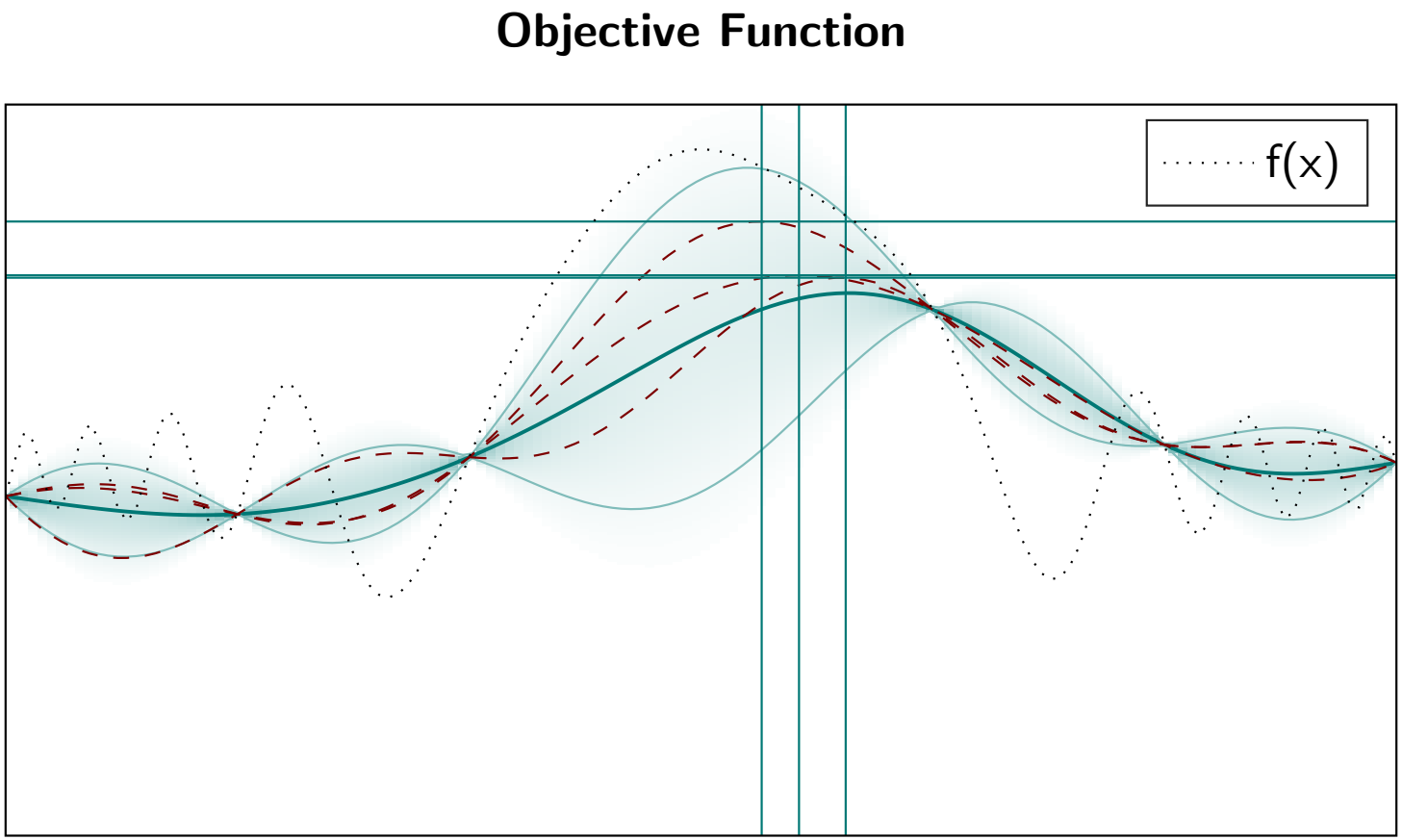
Calculations for the conjugate set-up:

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

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

  where $\boldsymbol{f} = (f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_n))$ and $[\boldsymbol{\Phi}]_{ij} = \phi_j(\boldsymbol{x}_i)$.
- Posterior marginal: $\boldsymbol{x}^*|\mathcal{D} \sim$?
  - Draw $\boldsymbol{\beta}$ from $\boldsymbol{\beta}|\mathcal{D}$
  - Evaluate $\boldsymbol{x}^* = \arg\max \beta_1 \phi_1(\boldsymbol{x}) + \cdots + \beta_m \phi_m(\boldsymbol{x})$
  - Repeat.

Compute $\boldsymbol{x}^* = \arg\max f(\boldsymbol{x})$:



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 $p(x^*|\mathcal{D})$ 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.

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.

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 $p(\boldsymbol{x}^*|\mathcal{D})$ 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 $p(\boldsymbol{x}^* | \mathcal{D})$ 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.

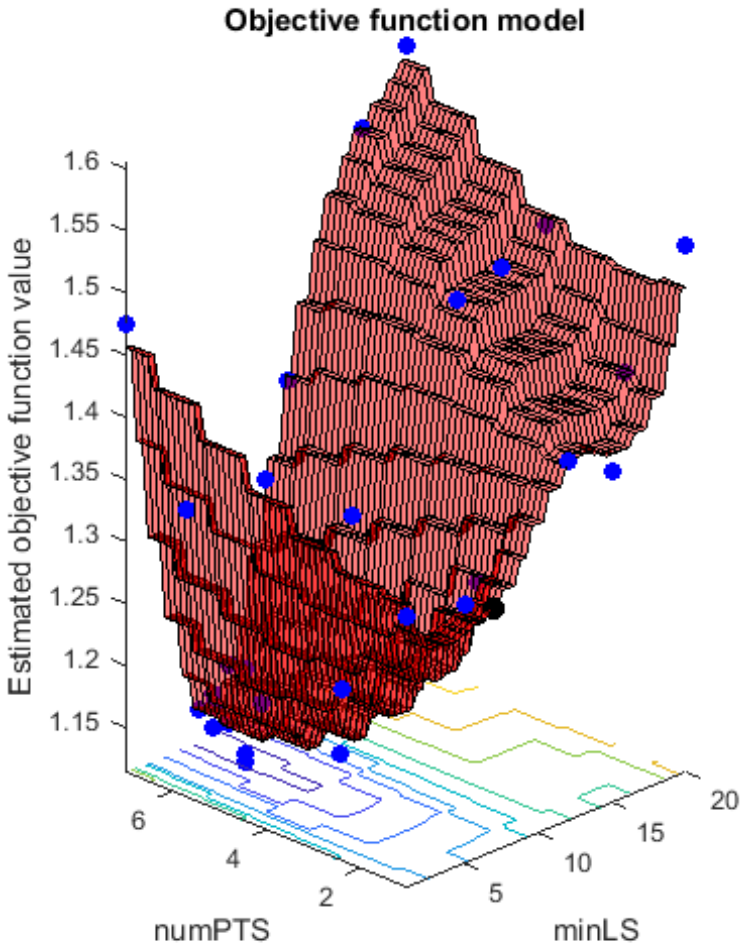
Some Discussion Points

- 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 $p(\boldsymbol{x}^* | \mathcal{D})$ 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 $p(\boldsymbol{x}^* | \mathcal{D})$ 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



`https://uk.mathworks.com/help/stats/tune-random-forest-using-quantile-error-and-bayesian-optimization.html`
(but the MATLAB function doesn't provide uncertainty quantification!)

Integration

Integration

$$I = \int f(\boldsymbol{x})\pi(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$$

- Well-defined:
  - $f \in L_2(\pi)$?
  - $f \in C^\alpha(\mathcal{X})$ for some $\alpha \geq 0$ and $\boldsymbol{x} \in \mathcal{X}$ 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

$$I = \int f(\boldsymbol{x})\pi(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$$

- Well-defined:
  - $f \in L_2(\pi)$?
  - $f \in C^\alpha(\mathcal{X})$ for some $\alpha \geq 0$ and $\boldsymbol{x} \in \mathcal{X}$ 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

$$I = \int f(\boldsymbol{x})\pi(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$$

- Well-defined:
  - $f \in L_2(\pi)$?
  - $f \in C^\alpha(\mathcal{X})$ for some $\alpha \geq 0$ and $\boldsymbol{x} \in \mathcal{X}$ 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

$$I = \int f(\boldsymbol{x})\pi(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$$

- Well-defined:
  - $f \in L_2(\pi)$?
  - $f \in C^\alpha(\mathcal{X})$ for some $\alpha \geq 0$ and $\boldsymbol{x} \in \mathcal{X}$ 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

$$I = \int f(\boldsymbol{x})\pi(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$$

- Well-defined:
  - $f \in L_2(\pi)$?
  - $f \in C^\alpha(\mathcal{X})$ for some $\alpha \geq 0$ and $\boldsymbol{x} \in \mathcal{X}$ 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

$$I = \int f(\boldsymbol{x})\pi(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$$

- Well-defined:
  - $f \in L_2(\pi)$?
  - $f \in C^\alpha(\mathcal{X})$ for some $\alpha \geq 0$ and $\boldsymbol{x} \in \mathcal{X}$ 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

$$I = \int f(\boldsymbol{x})\pi(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$$

- Well-defined:
  - $f \in L_2(\pi)$?
  - $f \in C^\alpha(\mathcal{X})$ for some $\alpha \geq 0$ and $\boldsymbol{x} \in \mathcal{X}$ 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

$$I = \int f(\boldsymbol{x})\pi(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$$

Two distinct requirements:

- A method to select the integrand evaluation locations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$.
  - Uniform grid over $\mathcal{X}$?
  - Adaptive selection, e.g. based on local error indicators?
- An estimator $\{(\boldsymbol{x}_i, f(\boldsymbol{x}_i))\}_{i=1}^n \mapsto \hat{I}$.
  - The arithmetic mean $\hat{I} = \frac{1}{n}\sum_{i=1}^n f(\boldsymbol{x}_i)$?
  - Something better?
- Key idea (again): Estimator uncertainty quantification!

Integration

$$I = \int f(\boldsymbol{x})\pi(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$$

Two distinct requirements:

- A method to select the integrand evaluation locations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$.
  - Uniform grid over $\mathcal{X}$?
  - Adaptive selection, e.g. based on local error indicators?
- An estimator $\{(\boldsymbol{x}_i, f(\boldsymbol{x}_i))\}_{i=1}^n \mapsto \hat{I}$.
  - The arithmetic mean $\hat{I} = \frac{1}{n}\sum_{i=1}^n f(\boldsymbol{x}_i)$?
  - Something better?
- Key idea (again): Estimator uncertainty quantification!

Integration

$$I = \int f(\boldsymbol{x})\pi(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$$

Two distinct requirements:

- A method to select the integrand evaluation locations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$.
  - Uniform grid over $\mathcal{X}$?
  - Adaptive selection, e.g. based on local error indicators?
- An estimator $\{(\boldsymbol{x}_i, f(\boldsymbol{x}_i))\}_{i=1}^n \mapsto \hat{I}$.
  - The arithmetic mean $\hat{I} = \frac{1}{n}\sum_{i=1}^n f(\boldsymbol{x}_i)$?
  - Something better?
- Key idea (again): Estimator uncertainty quantification!

Integration

$$I = \int f(\boldsymbol{x})\pi(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$$

Two distinct requirements:

- A method to select the integrand evaluation locations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$.
  - Uniform grid over $\mathcal{X}$?
  - Adaptive selection, e.g. based on local error indicators?
- An estimator $\{(\boldsymbol{x}_i, f(\boldsymbol{x}_i))\}_{i=1}^n \mapsto \hat{I}$.
  - The arithmetic mean $\hat{I} = \frac{1}{n}\sum_{i=1}^n f(\boldsymbol{x}_i)$?
  - Something better?
- Key idea (again): Estimator uncertainty quantification!

Integration

$$I = \int f(\boldsymbol{x})\pi(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$$

Two distinct requirements:

- A method to select the integrand evaluation locations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$.
  - Uniform grid over $\mathcal{X}$?
  - Adaptive selection, e.g. based on local error indicators?
- An estimator $\{(\boldsymbol{x}_i, f(\boldsymbol{x}_i))\}_{i=1}^n \mapsto \hat{I}$.
  - The arithmetic mean $\hat{I} = \frac{1}{n}\sum_{i=1}^n f(\boldsymbol{x}_i)$?
  - Something better?
- Key idea (again): Estimator uncertainty quantification!

Integration

$$I = \int f(\boldsymbol{x})\pi(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$$

Two distinct requirements:

- A method to select the integrand evaluation locations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$.
  - Uniform grid over $\mathcal{X}$?
  - Adaptive selection, e.g. based on local error indicators?
- An estimator $\{(\boldsymbol{x}_i, f(\boldsymbol{x}_i))\}_{i=1}^n \mapsto \hat{I}$.
  - The arithmetic mean $\hat{I} = \frac{1}{n}\sum_{i=1}^n f(\boldsymbol{x}_i)$?
  - Something better?
- Key idea (again): Estimator uncertainty quantification!

Integration

$$I = \int f(\boldsymbol{x})\pi(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$$

Two distinct requirements:

- A method to select the integrand evaluation locations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$.
  - Uniform grid over $\mathcal{X}$?
  - Adaptive selection, e.g. based on local error indicators?
- An estimator $\{(\boldsymbol{x}_i, f(\boldsymbol{x}_i))\}_{i=1}^n \mapsto \hat{I}$.
  - The arithmetic mean $\hat{I} = \frac{1}{n}\sum_{i=1}^n f(\boldsymbol{x}_i)$?
  - Something better?
- Key idea (again): Estimator uncertainty quantification!

Integration

$$I = \int f(\boldsymbol{x})\pi(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$$

Two distinct requirements:

- A method to select the integrand evaluation locations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$.
  - Uniform grid over $\mathcal{X}$?
  - Adaptive selection, e.g. based on local error indicators?
- An estimator $\{(\boldsymbol{x}_i, f(\boldsymbol{x}_i))\}_{i=1}^n \mapsto \hat{I}$.
  - The arithmetic mean $\hat{I} = \frac{1}{n}\sum_{i=1}^n f(\boldsymbol{x}_i)$?
  - Something better?
- Key idea (again): Estimator uncertainty quantification!

Probabilistic Integration

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 $p(I | \mathcal{D})$

Probabilistic Integration

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 $p(I|\mathcal{D})$

Probabilistic Integration

Calculations for the conjugate set-up:

Probabilistic Integration

Calculations for the conjugate set-up:

- Let: $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_m)$

Probabilistic Integration

Calculations for the conjugate set-up:

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

Probabilistic Integration

Calculations for the conjugate set-up:

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

Probabilistic Integration

Calculations for the conjugate set-up:

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

Probabilistic Integration

Calculations for the conjugate set-up:

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

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

  where $\boldsymbol{f} = (f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_n))$ and $[\boldsymbol{\Phi}]_{ij} = \phi_j(\boldsymbol{x}_i)$.

Probabilistic Integration

Calculations for the conjugate set-up:

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

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

  where $\boldsymbol{f} = (f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_n))$ and $[\boldsymbol{\Phi}]_{ij} = \phi_j(\boldsymbol{x}_i)$.
- Posterior marginal:

Calculations for the conjugate set-up:

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

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

  where $\boldsymbol{f} = (f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_n))$ and $[\boldsymbol{\Phi}]_{ij} = \phi_j(\boldsymbol{x}_i)$.

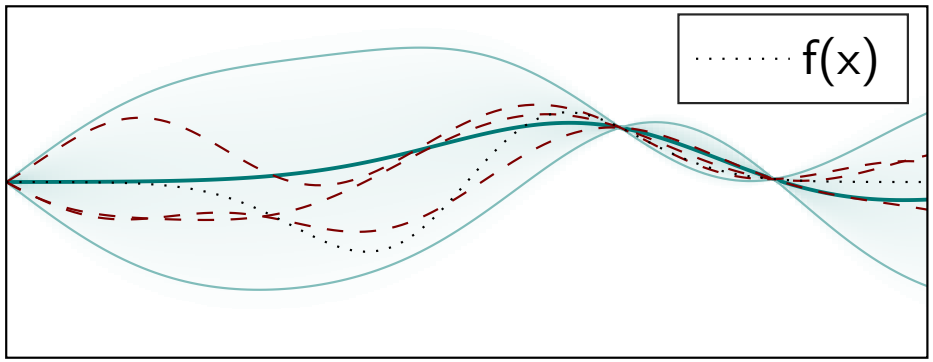
- Posterior marginal:

$$\begin{aligned} I|\mathcal{D} \quad \sim \quad & \text{Student-T}\left(\boldsymbol{\Psi}^\top (\boldsymbol{I} + \boldsymbol{\Phi}^\top \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^\top \boldsymbol{f}, \right. \\ & \left. \frac{1}{n}(\boldsymbol{f}^\top \boldsymbol{\Phi}^\top (\boldsymbol{I} + \boldsymbol{\Phi}\boldsymbol{\Phi}^\top)^{-1} \boldsymbol{\Phi}\boldsymbol{f}) \boldsymbol{\Psi}^\top (\boldsymbol{I} + \boldsymbol{\Phi}^\top \boldsymbol{\Phi})^{-1} \boldsymbol{\Psi}, n\right) \end{aligned}$$

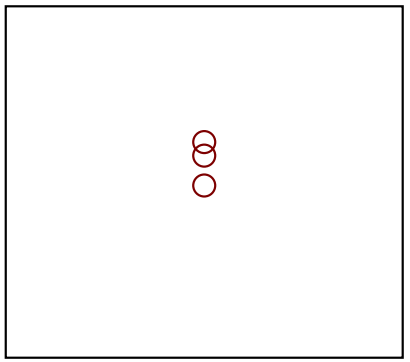
  where $[\boldsymbol{\Psi}]_i = \int \phi_i(\boldsymbol{x}) \pi(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}$.

Compute $\int f(\boldsymbol{x})\pi(\boldsymbol{x})d\boldsymbol{x}$:

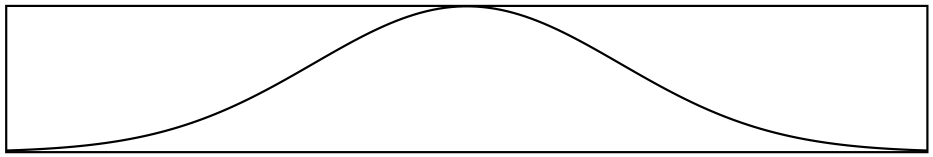
Posterior Samples



Corresponding Integrals



Reference Measure



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 $\{\boldsymbol{x}_i\}_{i=1}^n$?
  - Recall the posterior scale was determined by

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

  - Select $\{\boldsymbol{x}_i\}_{i=1}^n$ to minimise

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

  - Select $\boldsymbol{x}_n$ to minimise $(*)$ based on $\{\boldsymbol{x}_i\}_{i=1}^{n-1}$ 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 $\{\boldsymbol{x}_i\}_{i=1}^n$?
  - Recall the posterior scale was determined by

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

  - Select $\{\boldsymbol{x}_i\}_{i=1}^n$ to minimise

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

  - Select $\boldsymbol{x}_n$ to minimise $(*)$ based on $\{\boldsymbol{x}_i\}_{i=1}^{n-1}$ 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 $\{\boldsymbol{x}_i\}_{i=1}^n$?
  - Recall the posterior scale was determined by

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

  - Select $\{\boldsymbol{x}_i\}_{i=1}^n$ to minimise

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

  - Select $\boldsymbol{x}_n$ to minimise $(*)$ based on $\{\boldsymbol{x}_i\}_{i=1}^{n-1}$ 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 $\{\boldsymbol{x}_i\}_{i=1}^n$?
  - Recall the posterior scale was determined by

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

  - Select $\{\boldsymbol{x}_i\}_{i=1}^n$ to minimise

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

  - Select $\boldsymbol{x}_n$ to minimise $(*)$ based on $\{\boldsymbol{x}_i\}_{i=1}^{n-1}$ 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

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

  - 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 $\{\boldsymbol{x}_i\}_{i=1}^n$?
  - Recall the posterior scale was determined by

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

  - Select $\{\boldsymbol{x}_i\}_{i=1}^n$ to minimise

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

  - Select $\boldsymbol{x}_n$ to minimise $(*)$ based on $\{\boldsymbol{x}_i\}_{i=1}^{n-1}$ 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 $\{\boldsymbol{x}_i\}_{i=1}^n$?
  - Recall the posterior scale was determined by

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

  - Select $\{\boldsymbol{x}_i\}_{i=1}^n$ to minimise

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

  - Select $\boldsymbol{x}_n$ to minimise $(*)$ based on $\{\boldsymbol{x}_i\}_{i=1}^{n-1}$ 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 $\{\boldsymbol{x}_i\}_{i=1}^n$?
  - Recall the posterior scale was determined by

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

  - Select $\{\boldsymbol{x}_i\}_{i=1}^n$ to minimise

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

  - Select $\boldsymbol{x}_n$ to minimise $(*)$ based on $\{\boldsymbol{x}_i\}_{i=1}^{n-1}$ 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 $\{\boldsymbol{x}_i\}_{i=1}^n$?
  - Recall the posterior scale was determined by

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

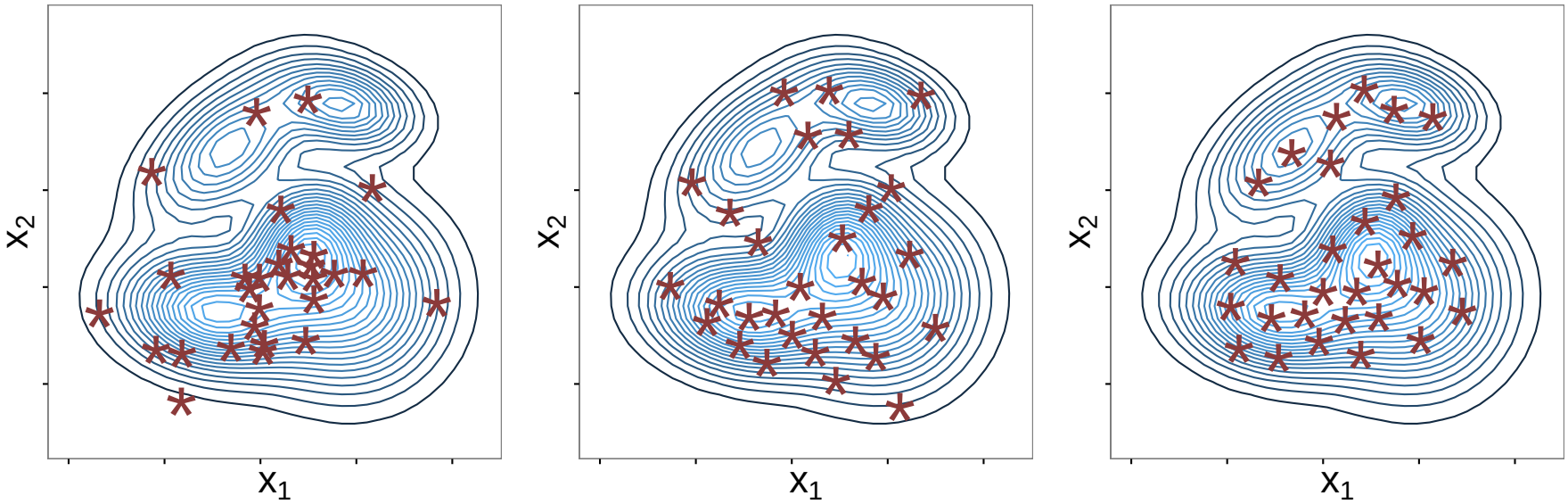
  - Select $\{\boldsymbol{x}_i\}_{i=1}^n$ to minimise

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

  - Select $\boldsymbol{x}_n$ to minimise $(*)$ based on $\{\boldsymbol{x}_i\}_{i=1}^{n-1}$ fixed? (Similar to "sequential Bayesian quadrature".)

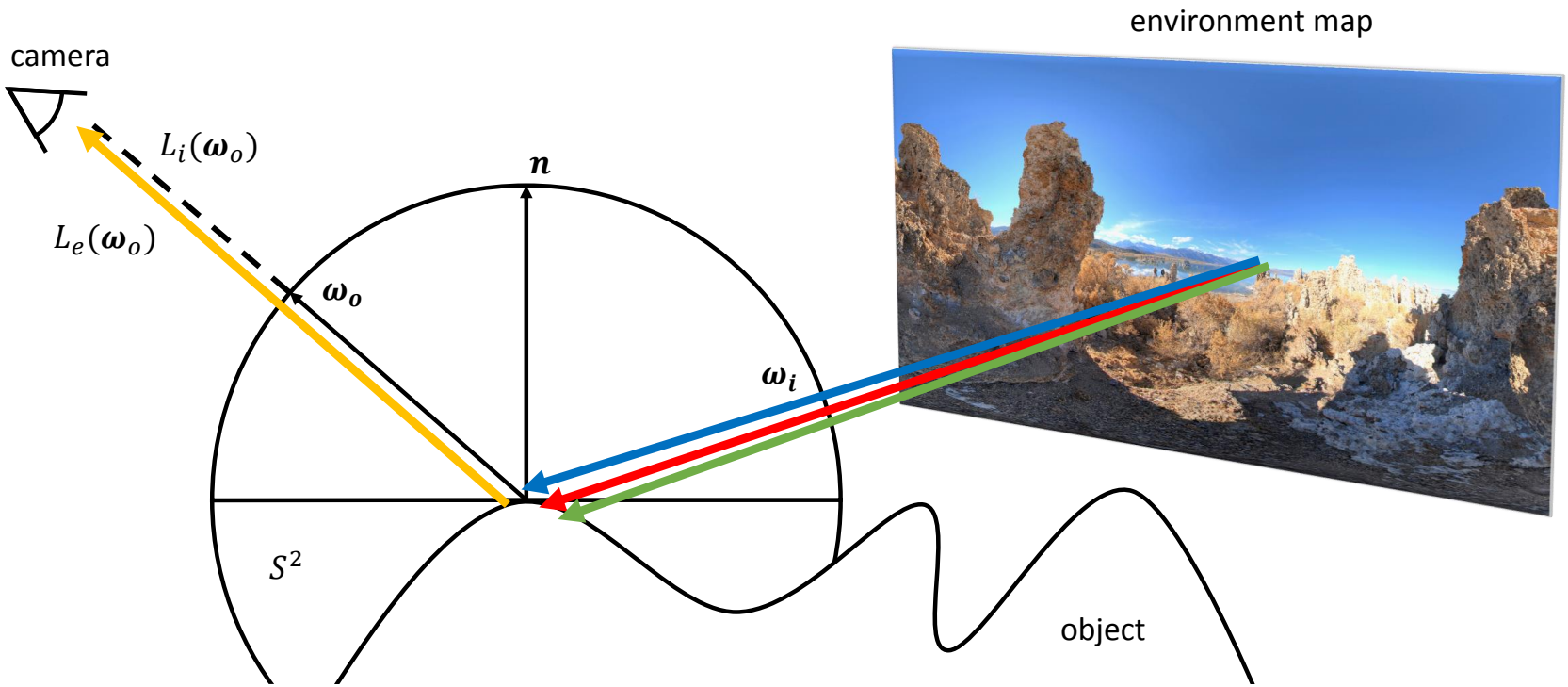
Some Discussion Points

From Briol *et al.*, 2016:



One of these is (a variant on) sequential Bayesian quadrature - which one?

Application



From Briol *et al.*, 2016.

Linear Algebra

Linear Algebra

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$$

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

Linear Algebra

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$$

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

Linear Algebra

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$$

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

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$$

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

Linear Algebra

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$$

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

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$$

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

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$$

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

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$$

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

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$$

Two distinct requirements:

- A method to select the directions $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n$.
  - Random projections?
  - Sequential selection, e.g. gradient descent or conjugate gradient?
- An estimator $\{(\boldsymbol{s}_i^\top \boldsymbol{A}, \boldsymbol{s}_i^\top \boldsymbol{b})\}_{i=1}^n \mapsto \hat{\boldsymbol{x}}$.
  - A minimal $\|\cdot\|_2$ norm vector that satisfies $\boldsymbol{s}_i^\top \boldsymbol{A}\hat{\boldsymbol{x}} = \boldsymbol{s}_i^\top \boldsymbol{b}$ for $i = 1, \ldots, n$?
  - Something better?
- Key idea (again): Estimator uncertainty quantification!

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$$

Two distinct requirements:

- A method to select the directions $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n$.
  - Random projections?
  - Sequential selection, e.g. gradient descent or conjugate gradient?
- An estimator $\{(\boldsymbol{s}_i^\top \boldsymbol{A}, \boldsymbol{s}_i^\top \boldsymbol{b})\}_{i=1}^n \mapsto \hat{\boldsymbol{x}}$.
  - A minimal $\|\cdot\|_2$ norm vector that satisfies $\boldsymbol{s}_i^\top \boldsymbol{A}\hat{\boldsymbol{x}} = \boldsymbol{s}_i^\top \boldsymbol{b}$ for $i = 1, \ldots, n$?
  - Something better?
- Key idea (again): Estimator uncertainty quantification!

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$$

Two distinct requirements:

- A method to select the directions $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n$.
  - Random projections?
  - Sequential selection, e.g. gradient descent or conjugate gradient?
- An estimator $\{(\boldsymbol{s}_i^\top \boldsymbol{A}, \boldsymbol{s}_i^\top \boldsymbol{b})\}_{i=1}^n \mapsto \hat{\boldsymbol{x}}$.
  - A minimal $\|\cdot\|_2$ norm vector that satisfies $\boldsymbol{s}_i^\top \boldsymbol{A}\hat{\boldsymbol{x}} = \boldsymbol{s}_i^\top \boldsymbol{b}$ for $i = 1, \ldots, n$?
  - Something better?
- Key idea (again): Estimator uncertainty quantification!

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$$

Two distinct requirements:

- A method to select the directions $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n$.
  - Random projections?
  - Sequential selection, e.g. gradient descent or conjugate gradient?
- An estimator $\{(\boldsymbol{s}_i^\top \boldsymbol{A}, \boldsymbol{s}_i^\top \boldsymbol{b})\}_{i=1}^n \mapsto \hat{\boldsymbol{x}}$.
  - A minimal $\|\cdot\|_2$ norm vector that satisfies $\boldsymbol{s}_i^\top \boldsymbol{A}\hat{\boldsymbol{x}} = \boldsymbol{s}_i^\top \boldsymbol{b}$ for $i = 1, \ldots, n$?
  - Something better?
- Key idea (again): Estimator uncertainty quantification!

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$$

Two distinct requirements:

- A method to select the directions $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n$.
  - Random projections?
  - Sequential selection, e.g. gradient descent or conjugate gradient?
- An estimator $\{(\boldsymbol{s}_i^\top \boldsymbol{A}, \boldsymbol{s}_i^\top \boldsymbol{b})\}_{i=1}^n \mapsto \hat{\boldsymbol{x}}$.
  - A minimal $\|\cdot\|_2$ norm vector that satisfies $\boldsymbol{s}_i^\top \boldsymbol{A}\hat{\boldsymbol{x}} = \boldsymbol{s}_i^\top \boldsymbol{b}$ for $i = 1, \ldots, n$?
  - Something better?
- Key idea (again): Estimator uncertainty quantification!

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$$

Two distinct requirements:

- A method to select the directions $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n$.
  - Random projections?
  - Sequential selection, e.g. gradient descent or conjugate gradient?
- An estimator $\{(\boldsymbol{s}_i^\top \boldsymbol{A}, \boldsymbol{s}_i^\top \boldsymbol{b})\}_{i=1}^n \mapsto \hat{\boldsymbol{x}}$.
  - A minimal $\|\cdot\|_2$ norm vector that satisfies $\boldsymbol{s}_i^\top \boldsymbol{A}\hat{\boldsymbol{x}} = \boldsymbol{s}_i^\top \boldsymbol{b}$ for $i = 1, \ldots, n$?
  - Something better?
- Key idea (again): Estimator uncertainty quantification!

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$$

Two distinct requirements:

- A method to select the directions $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n$.
  - Random projections?
  - Sequential selection, e.g. gradient descent or conjugate gradient?
- An estimator $\{(\boldsymbol{s}_i^\top \boldsymbol{A}, \boldsymbol{s}_i^\top \boldsymbol{b})\}_{i=1}^n \mapsto \hat{\boldsymbol{x}}$.
  - A minimal $\|\cdot\|_2$ norm vector that satisfies $\boldsymbol{s}_i^\top \boldsymbol{A}\hat{\boldsymbol{x}} = \boldsymbol{s}_i^\top \boldsymbol{b}$ for $i = 1, \ldots, n$?
  - Something better?
- Key idea (again): Estimator uncertainty quantification!

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$$

Two distinct requirements:

- A method to select the directions $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n$.
  - Random projections?
  - Sequential selection, e.g. gradient descent or conjugate gradient?
- An estimator $\{(\boldsymbol{s}_i^\top \boldsymbol{A}, \boldsymbol{s}_i^\top \boldsymbol{b})\}_{i=1}^n \mapsto \hat{\boldsymbol{x}}$.
  - A minimal $\|\cdot\|_2$ norm vector that satisfies $\boldsymbol{s}_i^\top \boldsymbol{A}\hat{\boldsymbol{x}} = \boldsymbol{s}_i^\top \boldsymbol{b}$ for $i = 1, \ldots, n$?
  - Something better?
- Key idea (again): Estimator uncertainty quantification!

Recap: Conjugate Gradient Method

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

Recap: Conjugate Gradient Method

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

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

Recap: Conjugate Gradient Method

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

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

Suppose that we have a conjugate basis $\{\boldsymbol{s}_1, \ldots, \boldsymbol{s}_N\}$ for $\mathbb{R}^N$ (i.e. $\langle \boldsymbol{s}_i, \boldsymbol{s}_j \rangle_{\boldsymbol{A}} = 0$ for all $i \neq j$).

Recap: Conjugate Gradient Method

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

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

Suppose that we have a conjugate basis $\{\boldsymbol{s}_1, \ldots, \boldsymbol{s}_N\}$ for $\mathbb{R}^N$ (i.e. $\langle \boldsymbol{s}_i, \boldsymbol{s}_j \rangle_{\boldsymbol{A}} = 0$ for all $i \neq j$).

Consider the natural sequence of approximations

$$\hat{\boldsymbol{x}}_n \quad = \quad \sum_{i=1}^{n} \alpha_i \boldsymbol{s}_i$$

where each

$$\alpha_i \quad = \quad \frac{\langle \boldsymbol{s}_i, \boldsymbol{x} \rangle_{\boldsymbol{A}}}{\langle \boldsymbol{s}_i, \boldsymbol{s}_i \rangle_{\boldsymbol{A}}} \quad = \quad \frac{\boldsymbol{s}_i^\top \boldsymbol{b}}{\boldsymbol{s}_i^\top \boldsymbol{A} \boldsymbol{s}_i}$$

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

Recap: Conjugate Gradient Method

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

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

Suppose that we have a conjugate basis $\{\boldsymbol{s}_1, \ldots, \boldsymbol{s}_N\}$ for $\mathbb{R}^N$ (i.e. $\langle \boldsymbol{s}_i, \boldsymbol{s}_j \rangle_{\boldsymbol{A}} = 0$ for all $i \neq j$).

Consider the natural sequence of approximations

$$\hat{\boldsymbol{x}}_n \quad = \quad \sum_{i=1}^{n} \alpha_i \boldsymbol{s}_i$$

where each

$$\alpha_i \quad = \quad \frac{\langle \boldsymbol{s}_i, \boldsymbol{x} \rangle_{\boldsymbol{A}}}{\langle \boldsymbol{s}_i, \boldsymbol{s}_i \rangle_{\boldsymbol{A}}} \quad = \quad \frac{\boldsymbol{s}_i^\top \boldsymbol{b}}{\boldsymbol{s}_i^\top \boldsymbol{A} \boldsymbol{s}_i}$$

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 $\{\boldsymbol{s}_1, \ldots, \boldsymbol{s}_N\}$.
- For theory, need to bound $\|\boldsymbol{x} - \hat{\boldsymbol{x}}_n\| = \| \sum_{i=n+1}^{N} \alpha_i \boldsymbol{x}_i \|$ 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

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

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 $\boldsymbol{s}_n$ based on the computation up to iteration $n-1$.

Gradient Descent: Notice that $\boldsymbol{x}$ is a minimum of

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

This suggests to select $\boldsymbol{s}_n = -\nabla f(\hat{\boldsymbol{x}}_{n-1})$ which is equal to $\boldsymbol{r}_{n-1} = \boldsymbol{b} - \boldsymbol{A}\hat{\boldsymbol{x}}_{n-1}$. However, this does not ensure $\{\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n\}$ is a conjugate set.

Conjugate Gradient: A more delicate procedure selects

$$\boldsymbol{s}_n \quad = \quad \boldsymbol{r}_{n-1} - \sum_{i<m} \frac{\boldsymbol{s}_i^\top \boldsymbol{A}\boldsymbol{r}_{n-1}}{\boldsymbol{s}_i^\top \boldsymbol{A}\boldsymbol{s}_i}\boldsymbol{s}_i$$

i.e. gradient descent plus Gram-Schmidt orthogonalisation w.r.t $\langle \cdot, \cdot \rangle_{\boldsymbol{A}}$ to subtract off components in the directions $\{\boldsymbol{s}_1, \ldots, \boldsymbol{s}_{n-1}\}$ 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

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

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

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

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 $\boldsymbol{s}_n$ based on the computation up to iteration $n-1$.

Gradient Descent: Notice that $\boldsymbol{x}$ is a minimum of

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

This suggests to select $\boldsymbol{s}_n = -\nabla f(\hat{\boldsymbol{x}}_{n-1})$ which is equal to $\boldsymbol{r}_{n-1} = \boldsymbol{b} - \boldsymbol{A}\hat{\boldsymbol{x}}_{n-1}$. However, this does not ensure $\{\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n\}$ is a conjugate set.

Conjugate Gradient: A more delicate procedure selects

$$\boldsymbol{s}_n \quad = \quad \boldsymbol{r}_{n-1} - \sum_{i<m} \frac{\boldsymbol{s}_i^\top \boldsymbol{A}\boldsymbol{r}_{n-1}}{\boldsymbol{s}_i^\top \boldsymbol{A}\boldsymbol{s}_i}\boldsymbol{s}_i$$

i.e. gradient descent plus Gram-Schmidt orthogonalisation w.r.t $\langle \cdot, \cdot \rangle_{\boldsymbol{A}}$ to subtract off components in the directions $\{\boldsymbol{s}_1, \ldots, \boldsymbol{s}_{n-1}\}$ already used.

For either method, the computational cost of selecting $\boldsymbol{s}_n$ is $O(N^2)$, so the overall computational overhead added is $O(nN^2)$; the same order as random projections.

Probabilistic Linear Algebra

Start with the data that have been collected:

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

Deploy full Bayesian inference for $\boldsymbol{x}$:

- Prior $p(\boldsymbol{x})$
- Likelihood $\prod_{i=1}^n \delta(\boldsymbol{s}_i^\top \boldsymbol{A}\boldsymbol{x} - \boldsymbol{s}_i^\top \boldsymbol{b})$
- Posterior $p(\boldsymbol{x}|\mathcal{D})$

Probabilistic Linear Algebra

Start with the data that have been collected:

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

Deploy full Bayesian inference for $\boldsymbol{x}$:

- Prior $p(\boldsymbol{x})$
- Likelihood $\prod_{i=1}^n \delta(\boldsymbol{s}_i^\top \boldsymbol{A}\boldsymbol{x} - \boldsymbol{s}_i^\top \boldsymbol{b})$
- Posterior $p(\boldsymbol{x}|\mathcal{D})$

Probabilistic Linear Algebra

Calculations for the conjugate set-up:

Probabilistic Linear Algebra

Calculations for the conjugate set-up:

- Let: $\boldsymbol{S} = [\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n]^\top$

Probabilistic Linear Algebra

Calculations for the conjugate set-up:

- Let: $\boldsymbol{S} = [\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n]^\top$
- Prior:

Probabilistic Linear Algebra

Calculations for the conjugate set-up:

- Let: $\boldsymbol{S} = [\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n]^\top$
- Prior: $\boldsymbol{x}|\lambda \sim \mathsf{N}(\boldsymbol{0}, \lambda \boldsymbol{I})$ and $\lambda \sim p(\lambda) \propto \lambda^{-1}$.

Probabilistic Linear Algebra

Calculations for the conjugate set-up:

- Let: $\boldsymbol{S} = [\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n]^\top$
- Prior: $\boldsymbol{x}|\lambda \sim \mathsf{N}(\boldsymbol{0}, \lambda \boldsymbol{I})$ and $\lambda \sim p(\lambda) \propto \lambda^{-1}$.
- Posterior:

Calculations for the conjugate set-up:

- Let: $\boldsymbol{S} = [\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n]^\top$
- Prior: $\boldsymbol{x}|\lambda \sim \mathsf{N}(\boldsymbol{0}, \lambda \boldsymbol{I})$ and $\lambda \sim p(\lambda) \propto \lambda^{-1}$.
- Posterior:

$$\boldsymbol{x}|\mathcal{D} \quad \sim \quad \mathsf{MVT}\left((\boldsymbol{I} + \boldsymbol{A}^\top \boldsymbol{S}^\top \boldsymbol{S}\boldsymbol{A})^{-1}\boldsymbol{A}^\top \boldsymbol{S}^\top \boldsymbol{S}\boldsymbol{b}, \right.$$
$$\left. \frac{1}{n}(\boldsymbol{b}^\top \boldsymbol{S}^\top \boldsymbol{A}^\top \boldsymbol{S}^\top (\boldsymbol{I} + \boldsymbol{S}\boldsymbol{A}\boldsymbol{A}^\top \boldsymbol{S})^{-1}\boldsymbol{S}\boldsymbol{A}\boldsymbol{S}\boldsymbol{b})(\boldsymbol{I} + \boldsymbol{A}^\top \boldsymbol{S}^\top \boldsymbol{S}\boldsymbol{A})^{-1}, n\right)$$

Calculations for the conjugate set-up:

- Let: $\boldsymbol{S} = [\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n]^\top$
- Prior: $\boldsymbol{x}|\lambda \sim \mathrm{N}(\boldsymbol{0}, \lambda \boldsymbol{I})$ and $\lambda \sim p(\lambda) \propto \lambda^{-1}$.
- Posterior:

$$\boldsymbol{x}|\mathcal{D} \quad \sim \quad \mathrm{MVT}\left((\boldsymbol{I} + \boldsymbol{A}^\top \boldsymbol{S}^\top \boldsymbol{S}\boldsymbol{A})^{-1}\boldsymbol{A}^\top \boldsymbol{S}^\top \boldsymbol{S}\boldsymbol{b}, \right.$$
$$\left. \frac{1}{n}(\boldsymbol{b}^\top \boldsymbol{S}^\top \boldsymbol{A}^\top \boldsymbol{S}^\top (\boldsymbol{I} + \boldsymbol{S}\boldsymbol{A}\boldsymbol{A}^\top \boldsymbol{S})^{-1}\boldsymbol{S}\boldsymbol{A}\boldsymbol{S}\boldsymbol{b})(\boldsymbol{I} + \boldsymbol{A}^\top \boldsymbol{S}^\top \boldsymbol{S}\boldsymbol{A})^{-1}, n\right)$$

- This is for general $\boldsymbol{S}$.

Calculations for the conjugate set-up:

- Let: $\boldsymbol{S} = [\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n]^\top$
- Prior: $\boldsymbol{x}|\lambda \sim \mathrm{N}(\boldsymbol{0}, \lambda \boldsymbol{I})$ and $\lambda \sim p(\lambda) \propto \lambda^{-1}$.
- Posterior:

$$\begin{aligned} \boldsymbol{x}|\mathcal{D} \quad \sim \quad & \mathrm{MVT}\Big((\boldsymbol{I} + \boldsymbol{A}^\top \boldsymbol{S}^\top \boldsymbol{S}\boldsymbol{A})^{-1} \boldsymbol{A}^\top \boldsymbol{S}^\top \boldsymbol{S}\boldsymbol{b}, \\ & \frac{1}{n}(\boldsymbol{b}^\top \boldsymbol{S}^\top \boldsymbol{A}^\top \boldsymbol{S}^\top (\boldsymbol{I} + \boldsymbol{S}\boldsymbol{A}\boldsymbol{A}^\top \boldsymbol{S})^{-1} \boldsymbol{S}\boldsymbol{A}\boldsymbol{S}\boldsymbol{b})(\boldsymbol{I} + \boldsymbol{A}^\top \boldsymbol{S}^\top \boldsymbol{S}\boldsymbol{A})^{-1}, n\Big) \end{aligned}$$

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

Probabilistic Linear Algebra

[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 $\boldsymbol{A}^\top \boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}^\top \boldsymbol{b}$.
- 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 $\boldsymbol{A}^\top \boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}^\top \boldsymbol{b}$.
- 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 $\boldsymbol{A}^\top \boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}^\top \boldsymbol{b}$.
- 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 $\boldsymbol{A}^\top \boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}^\top \boldsymbol{b}$.
- Thus the classical error bounds for CG are inherited.
- The full posterior can be computed in $O(nN^2)$.

Some Discussion Points

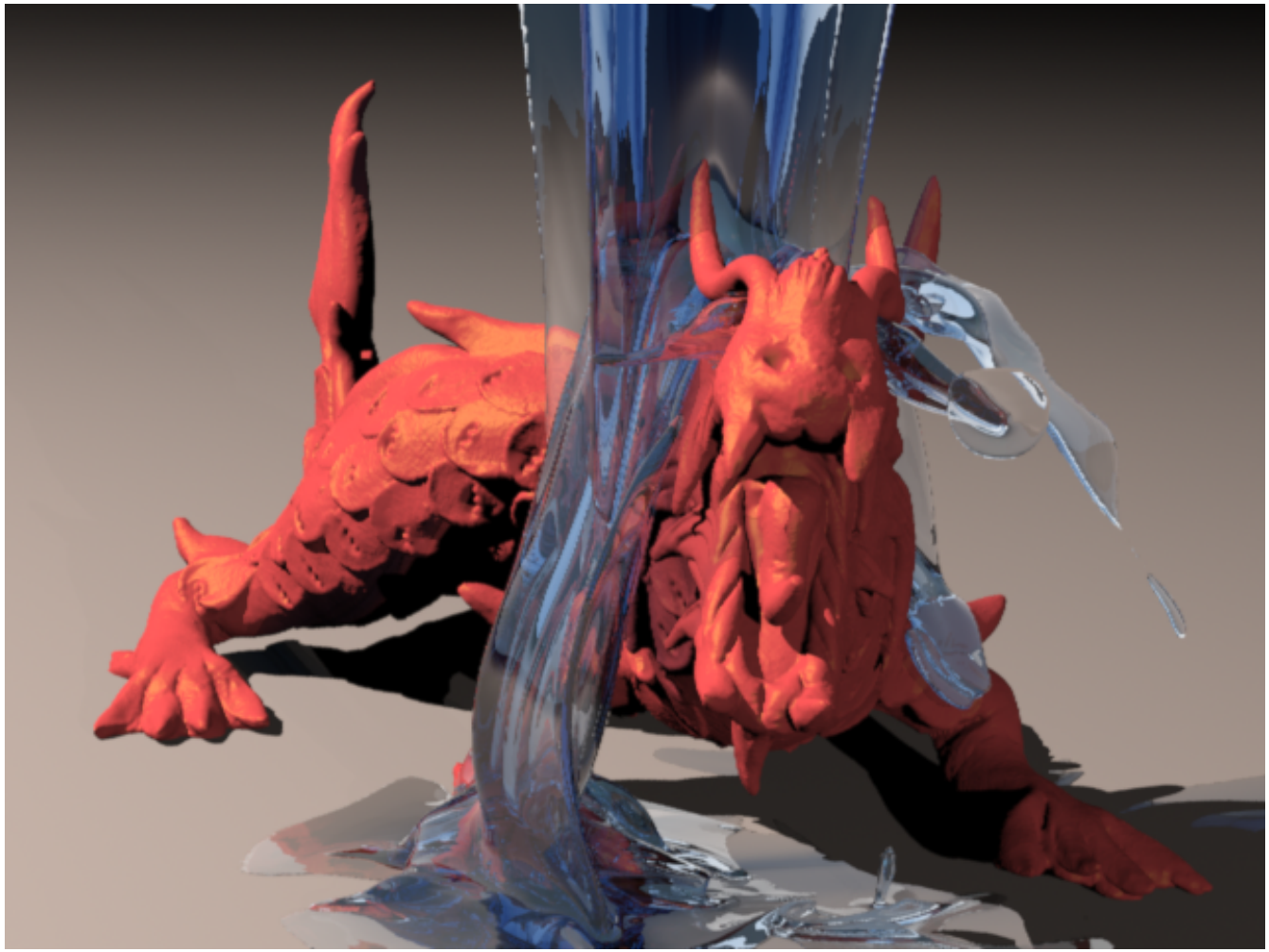
- Approximate linear solvers used extensively in engineering applications.
- Also relevant in statistics, e.g. simulation of spatial random fields.
- It turns out that the posterior mean in our construction coincides with the classical conjugate gradient (CG) method applied to the pre-conditioned system $\boldsymbol{A}^\top \boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}^\top \boldsymbol{b}$.
- 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 $\boldsymbol{A}^\top \boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}^\top \boldsymbol{b}$.
- 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:

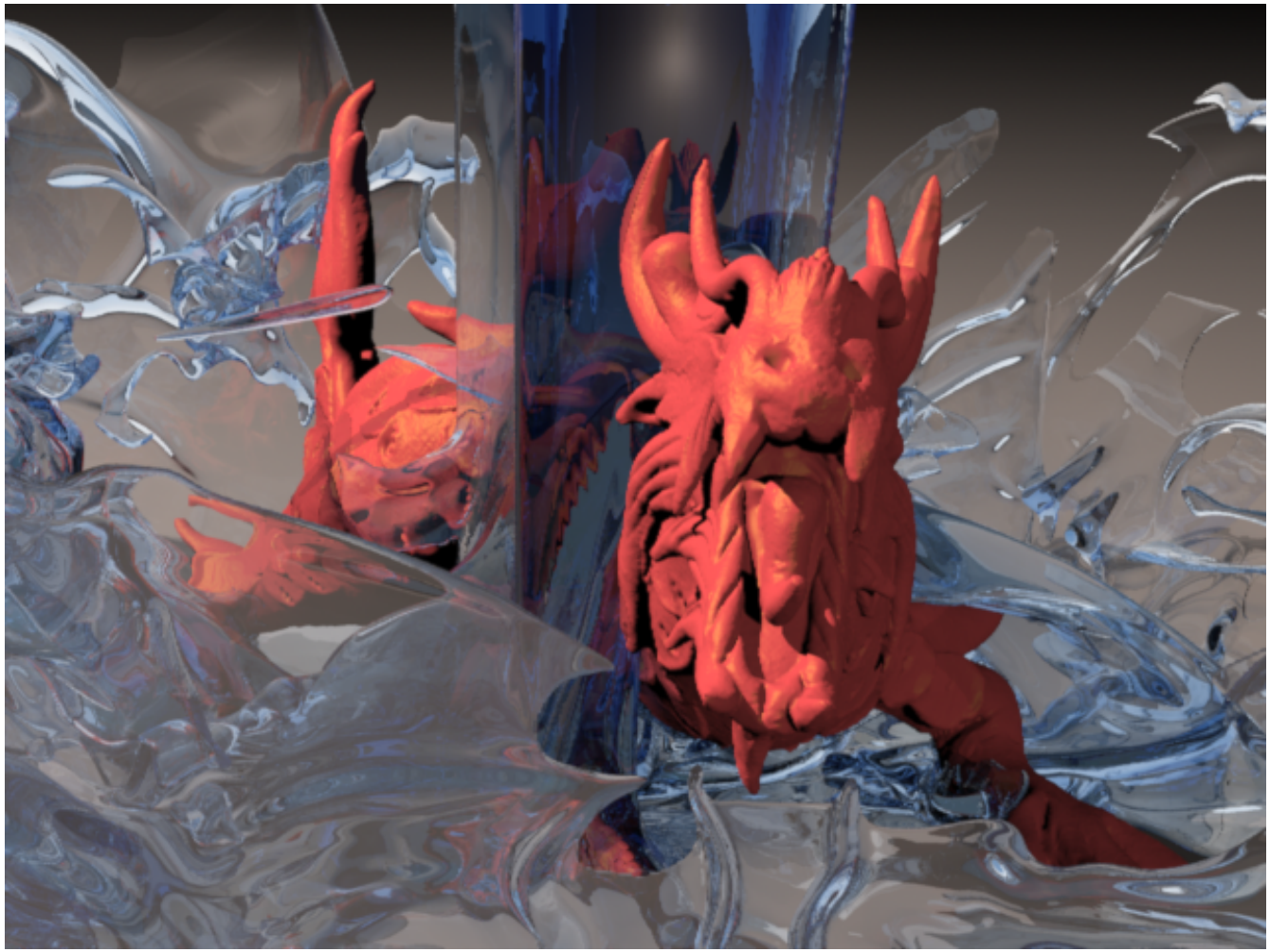


Application

From McAdams *et al.*, SIGGRAPH 2010:

Application

From McAdams *et al.*, SIGGRAPH 2010:



Solution of Differential Equations

Solution of Differential Equations

E.g.:

$$\begin{aligned} \Delta u(\boldsymbol{x}) &= f(\boldsymbol{x}), \quad \boldsymbol{x} \in \mathcal{X} \\ u(\boldsymbol{x}) &= 0, \quad \boldsymbol{x} \in \partial\mathcal{X} \end{aligned}$$

- 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 $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{X} \cup \partial\mathcal{X}$ which you can select.
  - Aim to minimise $\int_{\mathcal{X}} \|\hat{u}(\boldsymbol{x}) - u(\boldsymbol{x})\|_2^2 \mathrm{d}\boldsymbol{x}$.

Solution of Differential Equations

E.g.:

$$\begin{aligned} \Delta u(\boldsymbol{x}) &= f(\boldsymbol{x}), \quad \boldsymbol{x} \in \mathcal{X} \\ u(\boldsymbol{x}) &= 0, \quad \boldsymbol{x} \in \partial\mathcal{X} \end{aligned}$$

- 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 $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{X} \cup \partial\mathcal{X}$ which you can select.
  - Aim to minimise $\int_{\mathcal{X}} \|\hat{u}(\boldsymbol{x}) - u(\boldsymbol{x})\|_2^2 \mathrm{d}\boldsymbol{x}$.

Solution of Differential Equations

E.g.:

$$\begin{aligned} \Delta u(\boldsymbol{x}) &= f(\boldsymbol{x}), \quad \boldsymbol{x} \in \mathcal{X} \\ u(\boldsymbol{x}) &= 0, \quad \boldsymbol{x} \in \partial\mathcal{X} \end{aligned}$$

- 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 $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{X} \cup \partial\mathcal{X}$ which you can select.
  - Aim to minimise $\int_{\mathcal{X}} \|\hat{u}(\boldsymbol{x}) - u(\boldsymbol{x})\|_2^2 \mathrm{d}\boldsymbol{x}$.

Solution of Differential Equations

E.g.:

$$\begin{aligned} \Delta u(\boldsymbol{x}) &= f(\boldsymbol{x}), \quad \boldsymbol{x} \in \mathcal{X} \\ u(\boldsymbol{x}) &= 0, \quad \boldsymbol{x} \in \partial\mathcal{X} \end{aligned}$$

- 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 $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{X} \cup \partial\mathcal{X}$ which you can select.
  - Aim to minimise $\int_{\mathcal{X}} \|\hat{u}(\boldsymbol{x}) - u(\boldsymbol{x})\|_2^2 \mathrm{d}\boldsymbol{x}$.

Solution of Differential Equations

E.g.:

$$\begin{aligned} \Delta u(\boldsymbol{x}) &= f(\boldsymbol{x}), \quad \boldsymbol{x} \in \mathcal{X} \\ u(\boldsymbol{x}) &= 0, \quad \boldsymbol{x} \in \partial\mathcal{X} \end{aligned}$$

- 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 $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{X} \cup \partial\mathcal{X}$ which you can select.
  - Aim to minimise $\int_{\mathcal{X}} \|\hat{u}(\boldsymbol{x}) - u(\boldsymbol{x})\|_2^2 \mathrm{d}\boldsymbol{x}$.

Solution of Differential Equations

E.g.:

$$\begin{aligned} \Delta u(\boldsymbol{x}) &= f(\boldsymbol{x}), \quad \boldsymbol{x} \in \mathcal{X} \\ u(\boldsymbol{x}) &= 0, \quad \boldsymbol{x} \in \partial\mathcal{X} \end{aligned}$$

- 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 $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{X} \cup \partial\mathcal{X}$ which you can select.
  - Aim to minimise $\int_{\mathcal{X}} \|\hat{u}(\boldsymbol{x}) - u(\boldsymbol{x})\|_2^2 \mathrm{d}\boldsymbol{x}$.

Solution of Differential Equations

$$\begin{aligned} \Delta u(\boldsymbol{x}) &= f(\boldsymbol{x}), \quad \boldsymbol{x} \in \mathcal{X} \\ u(\boldsymbol{x}) &= 0, \quad \boldsymbol{x} \in \partial\mathcal{X} \end{aligned}$$

Two distinct requirements:

- A method to select the function evaluation locations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{X} \cup \partial\mathcal{X}$.
  - Corners of a mesh on $\mathcal{X} \cup \partial\mathcal{X}$?
  - An adaptive method?
- An estimator $\{(\boldsymbol{x}_i, f(\boldsymbol{x}_i))\}_{i=1}^n \mapsto \hat{u}(\cdot)$.
  - Linear interpolation of the $f(\boldsymbol{x}_i)$ and then solution of the PDE?
  - Something better?
- Key idea (again): Estimator uncertainty quantification!

Solution of Differential Equations

$$\begin{aligned} \Delta u(\boldsymbol{x}) &= f(\boldsymbol{x}), \quad \boldsymbol{x} \in \mathcal{X} \\ u(\boldsymbol{x}) &= 0, \quad \boldsymbol{x} \in \partial\mathcal{X} \end{aligned}$$

Two distinct requirements:

- A method to select the function evaluation locations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{X} \cup \partial\mathcal{X}$.
  - Corners of a mesh on $\mathcal{X} \cup \partial\mathcal{X}$?
  - An adaptive method?
- An estimator $\{(\boldsymbol{x}_i, f(\boldsymbol{x}_i))\}_{i=1}^n \mapsto \hat{u}(\cdot)$.
  - Linear interpolation of the $f(\boldsymbol{x}_i)$ and then solution of the PDE?
  - Something better?
- Key idea (again): Estimator uncertainty quantification!

Solution of Differential Equations

$$\begin{aligned} \Delta u(\boldsymbol{x}) &= f(\boldsymbol{x}), \quad \boldsymbol{x} \in \mathcal{X} \\ u(\boldsymbol{x}) &= 0, \quad \boldsymbol{x} \in \partial\mathcal{X} \end{aligned}$$

Two distinct requirements:

- A method to select the function evaluation locations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{X} \cup \partial\mathcal{X}$.
  - Corners of a mesh on $\mathcal{X} \cup \partial\mathcal{X}$?
  - An adaptive method?
- An estimator $\{(\boldsymbol{x}_i, f(\boldsymbol{x}_i))\}_{i=1}^n \mapsto \hat{u}(\cdot)$.
  - Linear interpolation of the $f(\boldsymbol{x}_i)$ and then solution of the PDE?
  - Something better?
- Key idea (again): Estimator uncertainty quantification!

Solution of Differential Equations

$$\begin{aligned} \Delta u(\boldsymbol{x}) &= f(\boldsymbol{x}), \quad \boldsymbol{x} \in \mathcal{X} \\ u(\boldsymbol{x}) &= 0, \quad \boldsymbol{x} \in \partial\mathcal{X} \end{aligned}$$

Two distinct requirements:

- A method to select the function evaluation locations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{X} \cup \partial\mathcal{X}$.
  - Corners of a mesh on $\mathcal{X} \cup \partial\mathcal{X}$?
  - An adaptive method?
- An estimator $\{(\boldsymbol{x}_i, f(\boldsymbol{x}_i))\}_{i=1}^n \mapsto \hat{u}(\cdot)$.
  - Linear interpolation of the $f(\boldsymbol{x}_i)$ and then solution of the PDE?
  - Something better?
- Key idea (again): Estimator uncertainty quantification!

Solution of Differential Equations

$$\begin{aligned} \Delta u(\boldsymbol{x}) &= f(\boldsymbol{x}), \quad \boldsymbol{x} \in \mathcal{X} \\ u(\boldsymbol{x}) &= 0, \quad \boldsymbol{x} \in \partial\mathcal{X} \end{aligned}$$

Two distinct requirements:

- A method to select the function evaluation locations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{X} \cup \partial\mathcal{X}$.
  - Corners of a mesh on $\mathcal{X} \cup \partial\mathcal{X}$?
  - An adaptive method?
- An estimator $\{(\boldsymbol{x}_i, f(\boldsymbol{x}_i))\}_{i=1}^n \mapsto \hat{u}(\cdot)$.
  - Linear interpolation of the $f(\boldsymbol{x}_i)$ and then solution of the PDE?
  - Something better?
- Key idea (again): Estimator uncertainty quantification!

Solution of Differential Equations

$$\begin{aligned} \Delta u(\boldsymbol{x}) &= f(\boldsymbol{x}), \quad \boldsymbol{x} \in \mathcal{X} \\ u(\boldsymbol{x}) &= 0, \quad \boldsymbol{x} \in \partial\mathcal{X} \end{aligned}$$

Two distinct requirements:

- A method to select the function evaluation locations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{X} \cup \partial\mathcal{X}$.
  - Corners of a mesh on $\mathcal{X} \cup \partial\mathcal{X}$?
  - An adaptive method?
- An estimator $\{(\boldsymbol{x}_i, f(\boldsymbol{x}_i))\}_{i=1}^n \mapsto \hat{u}(\cdot)$.
  - Linear interpolation of the $f(\boldsymbol{x}_i)$ and then solution of the PDE?
  - Something better?
- Key idea (again): Estimator uncertainty quantification!

Solution of Differential Equations

$$\begin{aligned} \Delta u(\boldsymbol{x}) &= f(\boldsymbol{x}), \quad \boldsymbol{x} \in \mathcal{X} \\ u(\boldsymbol{x}) &= 0, \quad \boldsymbol{x} \in \partial\mathcal{X} \end{aligned}$$

Two distinct requirements:

- A method to select the function evaluation locations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{X} \cup \partial\mathcal{X}$.
  - Corners of a mesh on $\mathcal{X} \cup \partial\mathcal{X}$?
  - An adaptive method?
- An estimator $\{(\boldsymbol{x}_i, f(\boldsymbol{x}_i))\}_{i=1}^n \mapsto \hat{u}(\cdot)$.
  - Linear interpolation of the $f(\boldsymbol{x}_i)$ and then solution of the PDE?
  - Something better?
- Key idea (again): Estimator uncertainty quantification!

Solution of Differential Equations

$$\begin{aligned} \Delta u(\boldsymbol{x}) &= f(\boldsymbol{x}), \quad \boldsymbol{x} \in \mathcal{X} \\ u(\boldsymbol{x}) &= 0, \quad \boldsymbol{x} \in \partial\mathcal{X} \end{aligned}$$

Two distinct requirements:

- A method to select the function evaluation locations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{X} \cup \partial\mathcal{X}$.
  - Corners of a mesh on $\mathcal{X} \cup \partial\mathcal{X}$?
  - An adaptive method?
- An estimator $\{(\boldsymbol{x}_i, f(\boldsymbol{x}_i))\}_{i=1}^n \mapsto \hat{u}(\cdot)$.
  - Linear interpolation of the $f(\boldsymbol{x}_i)$ and then solution of the PDE?
  - Something better?
- Key idea (again): Estimator uncertainty quantification!

Probabilistic Solution of Differential Equations

Start with the data that have been collected:

$$\mathcal{D} = \{(\boldsymbol{x}_i, f(\boldsymbol{x}_i))\}_{i=1}^n$$

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

$$f(\boldsymbol{x}) = \beta_1\phi_1(\boldsymbol{x}) + \cdots + \beta_m\phi_m(\boldsymbol{x})$$

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

- Prior $p(\beta_1, \ldots, \beta_m)$
- Likelihood $\prod_{i=1}^n \delta(f(\boldsymbol{x}_i) - \beta_1\phi_1(\boldsymbol{x}_i) - \cdots - \beta_m\phi_m(\boldsymbol{x}_i))$
- Posterior $p(\beta_1, \ldots, \beta_n|\mathcal{D})$
- Posterior marginal $p(u(\cdot)|\mathcal{D})$

Probabilistic Solution of Differential Equations

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 $p(u(\cdot)|\mathcal{D})$

Probabilistic Solution of Differential Equations

Calculations for the conjugate set-up:

Probabilistic Solution of Differential Equations

Calculations for the conjugate set-up:

- Let: $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_m)$

Probabilistic Solution of Differential Equations

Calculations for the conjugate set-up:

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

Probabilistic Solution of Differential Equations

Calculations for the conjugate set-up:

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

Probabilistic Solution of Differential Equations

Calculations for the conjugate set-up:

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

Calculations for the conjugate set-up:

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

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

  where $\boldsymbol{f} = (f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_n))$ and $[\boldsymbol{\Phi}]_{ij} = \phi_j(\boldsymbol{x}_i)$.

Calculations for the conjugate set-up:

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

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

  where $\boldsymbol{f} = (f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_n))$ and $[\boldsymbol{\Phi}]_{ij} = \phi_j(\boldsymbol{x}_i)$.
- Posterior marginal:

Probabilistic Solution of Differential Equations

Calculations for the conjugate set-up:

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

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

  where $\boldsymbol{f} = (f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_n))$ and $[\boldsymbol{\Phi}]_{ij} = \phi_j(\boldsymbol{x}_i)$.

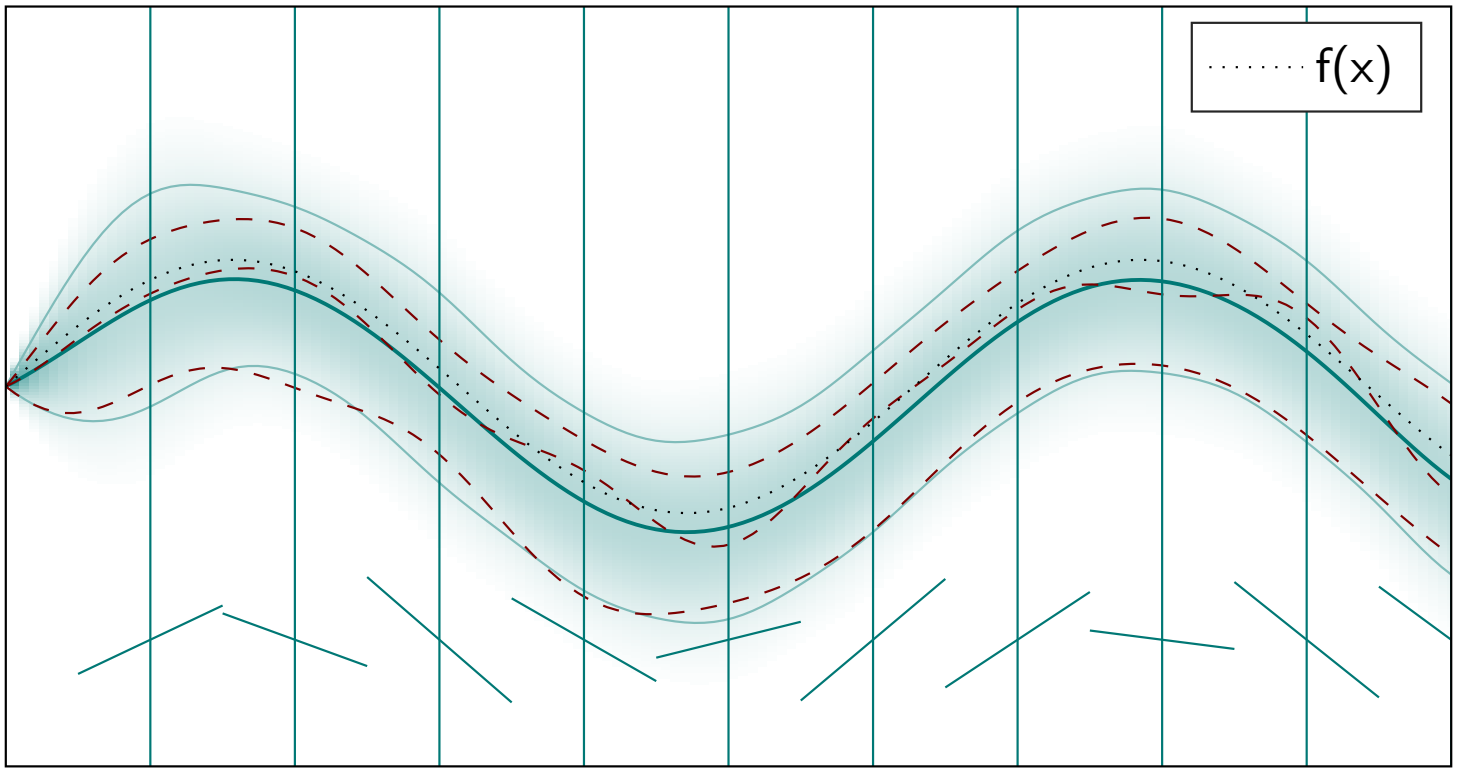
- Posterior marginal:

$$\begin{aligned} u(\boldsymbol{x})|\mathcal{D} \quad \sim \quad & \text{Student-T}\left(\boldsymbol{U}(\boldsymbol{x})^\top (\boldsymbol{I} + \boldsymbol{\Phi}^\top \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^\top \boldsymbol{f}, \right. \\ & \left. \frac{1}{n}(\boldsymbol{f}^\top \boldsymbol{\Phi}^\top (\boldsymbol{I} + \boldsymbol{\Phi}\boldsymbol{\Phi}^\top)^{-1} \boldsymbol{\Phi} \boldsymbol{f}) \boldsymbol{U}(\boldsymbol{x})^\top (\boldsymbol{I} + \boldsymbol{\Phi}^\top \boldsymbol{\Phi})^{-1} \boldsymbol{U}(\boldsymbol{x}), n\right) \end{aligned}$$

  where $[\boldsymbol{U}(\boldsymbol{x})]_i = u_i(\boldsymbol{x})$ 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 ϕ_i if you have the Green's function of the PDE.]

Solve the ODE $\frac{\mathrm{d}u}{\mathrm{d}x} = f(x)$, $u(0) = u_0$ on $x \in [0, T]$:

Solution to an ODE



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 ϵ 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

$$O(h^{\alpha-\rho-\frac{d}{2}}).$$

  - The posterior mass for a ball of radius ϵ 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

$$O(h^{\alpha-\rho-\frac{d}{2}}).$$

  - The posterior mass for a ball of radius ϵ 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

$$O(h^{\alpha-\rho-\frac{d}{2}}).$$

  - The posterior mass for a ball of radius ϵ 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

$$O(h^{\alpha - \rho - \frac{d}{2}}).$$

  - The posterior mass for a ball of radius ϵ centred on the true solution $u(\cdot)$ scales as

$$1 - O\left(\frac{h^{2\alpha - 2\rho - d}}{\epsilon} \right).$$

Some Discussion Points

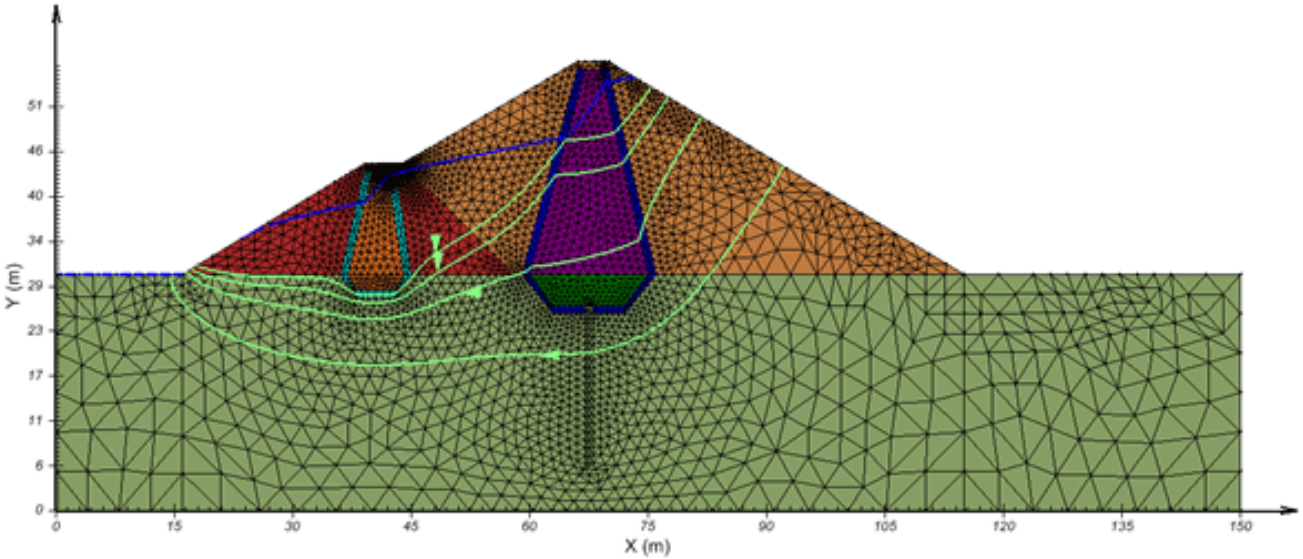
- Posterior mean coincides with a classical "collocation" method.
- Generalises to GPs with the kernel trick.
- Theoretical results (for a method based on GPs) in Cockayne *et al.*, 2016:
  - The posterior mean converges in $\| \cdot \|_\infty$ at a rate

$$O(h^{\alpha - \rho - \frac{d}{2}}).$$

  - The posterior mass for a ball of radius ϵ centred on the true solution $u(\cdot)$ scales as

$$1 - O\left(\frac{h^{2\alpha - 2\rho - d}}{\epsilon} \right).$$

Application



`http://www.svflux.com/subdomains/svflux.com/index.shtml`

Summary

Theoretical Questions

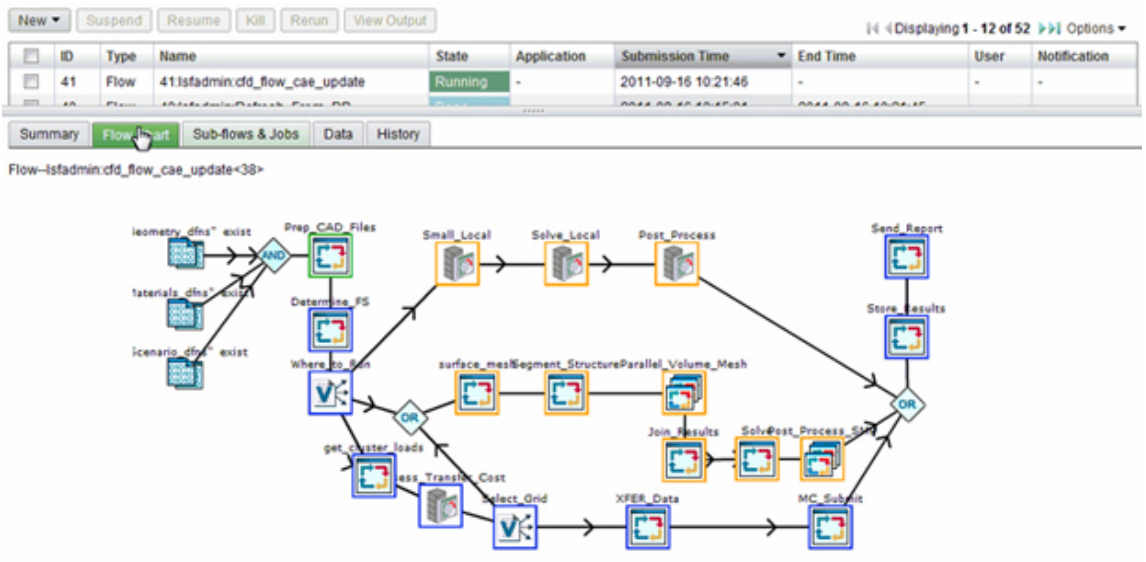
- General theory?
- Beyond linear and Gaussian assumptions?
- Experimental design?

Lots of work to do, but initial results in:

Cockayne J, Oates CJ, Sullivan T, Girolami M
Bayesian Probabilistic Numerical Methods
arXiv:1702.03673 (2017)

Computational Questions

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



[Fig: IBM High Performance Computation]

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.