Large-Scale Bayesian Inference:
Efficient Sampling-Based Approaches

Robert Scheichl

UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

Institute for Applied Mathematics & Interdisciplinary Center for Sceintific Computing

Collaborators: Teckentrup (Edinburgh), Dodwell (Exeter), Ketelsen (Boulder), Seelinger (Heidelberg), Reinarz (Durham), Bader (TUM), Bastian (Heidelberg)

Summer School on *"Hardware-Aware Scientific Computing"*

Heidelberg, Oct 7, 2021

What is Uncertainty Quantification (UQ)?

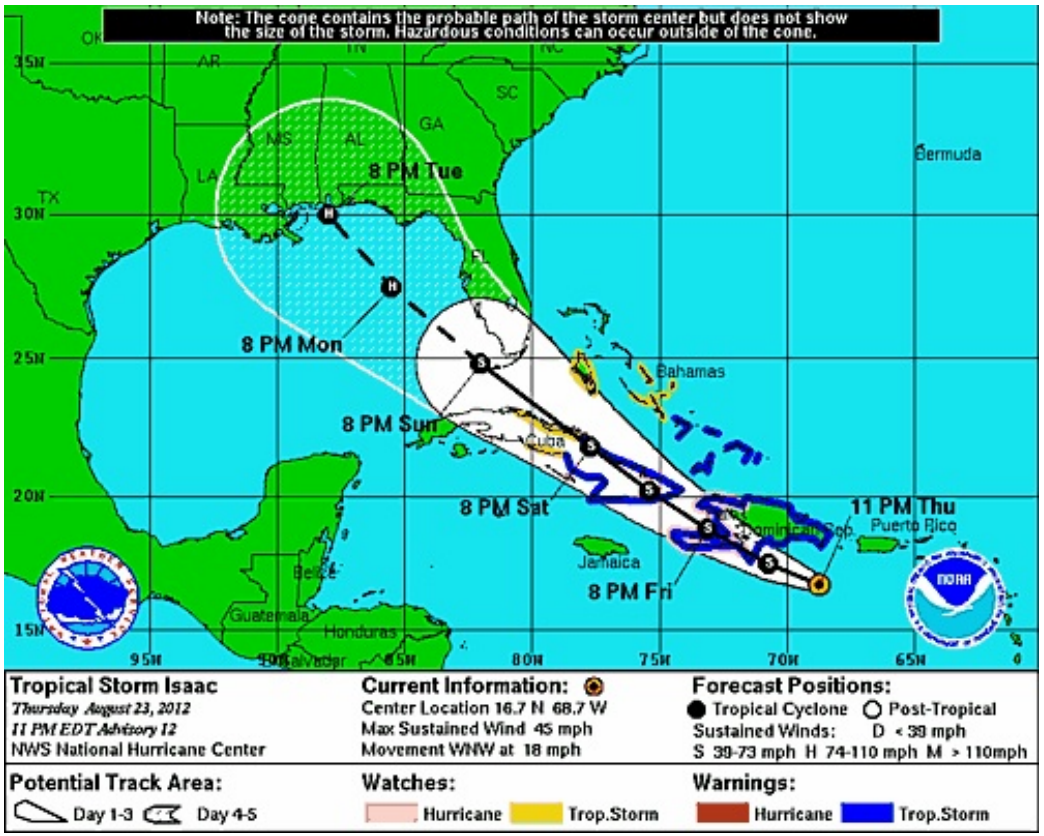
Uncertainty in Modern Life

Many aspects of modern life involve uncertainty:

- **Biology:** health, medicine, pharmaceuticals, gene expression, cancer research
- **Engineering:** automobiles, aircraft, structures, materials
- **Environment:** weather, climate, seismic, subsurface geophysics
- **Physics:** quantum physics, radioactive decay
- **Society:** finance, insurance industry, elections, military

What is Uncertainty Quantification (UQ)?

Examples

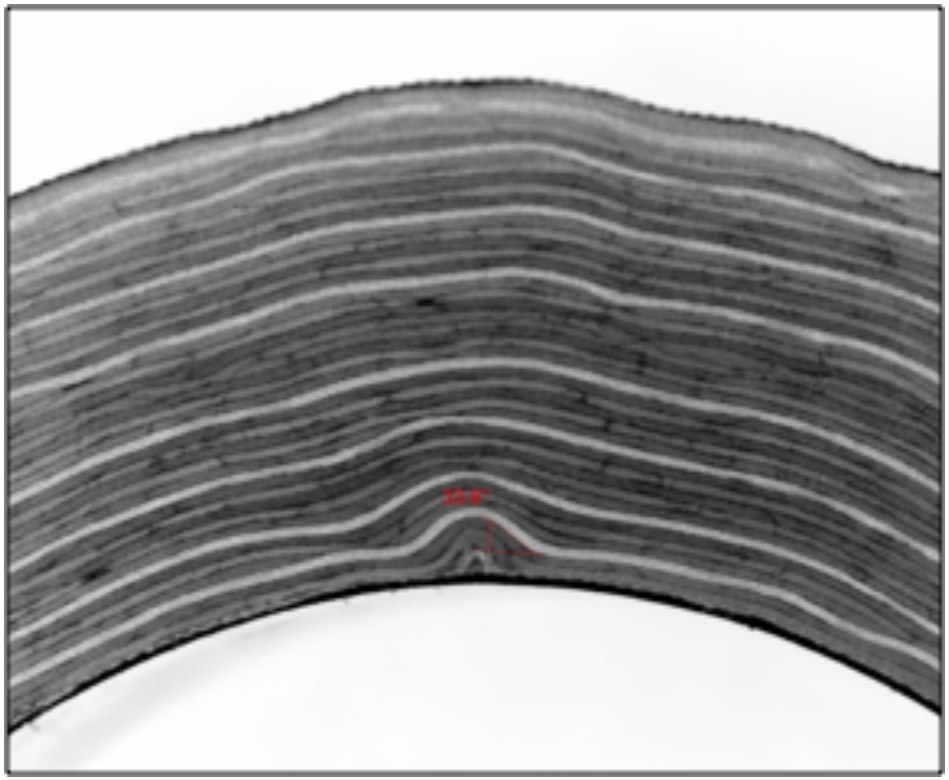


Source: National Hurricane Center, USA

Predicted storm path with uncertainty cones.

What is Uncertainty Quantification (UQ)?

Examples

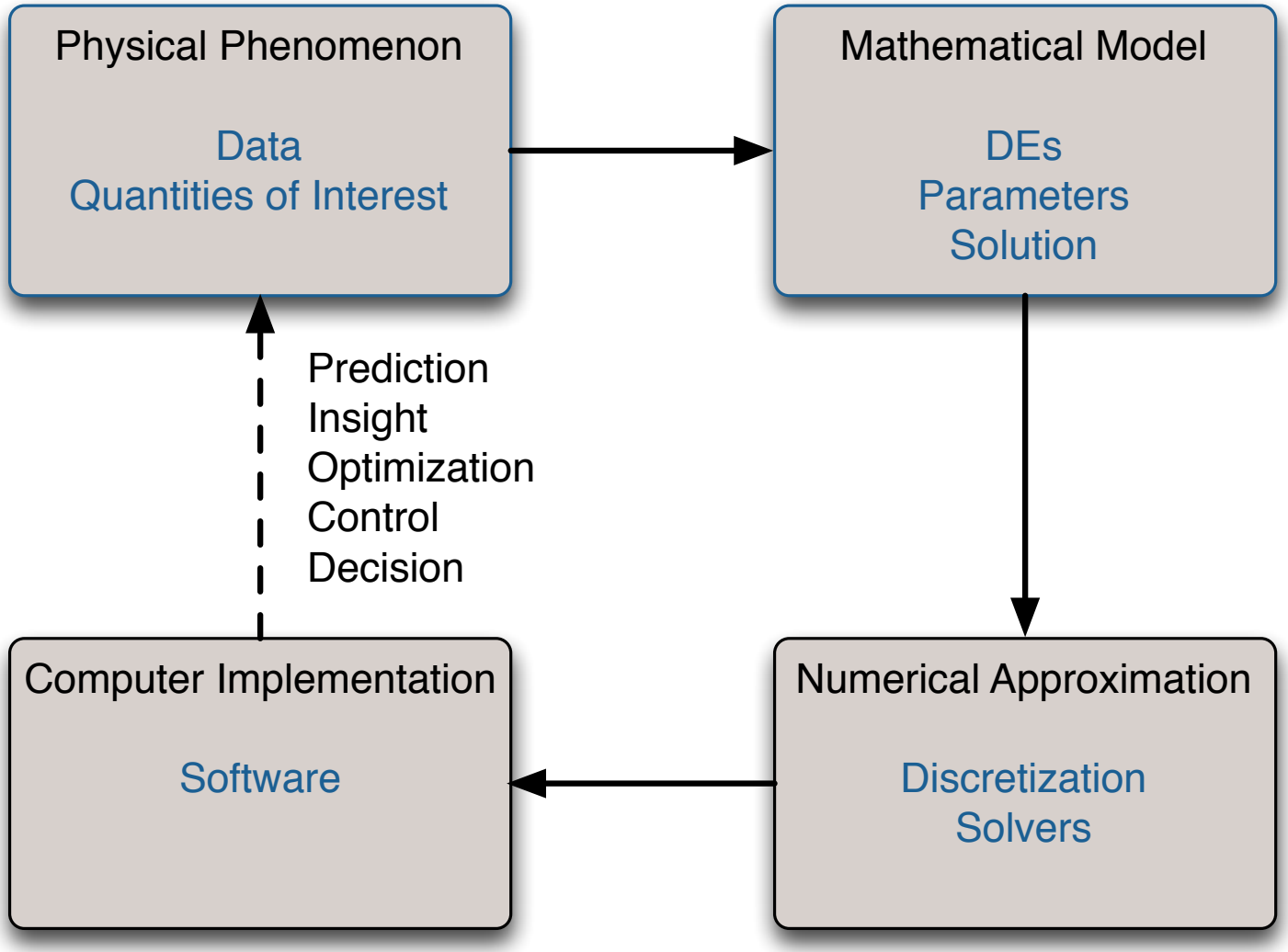


Source: GKN Aerospace

Predicted strength of carbon fibre wing subject to manufacturing defects

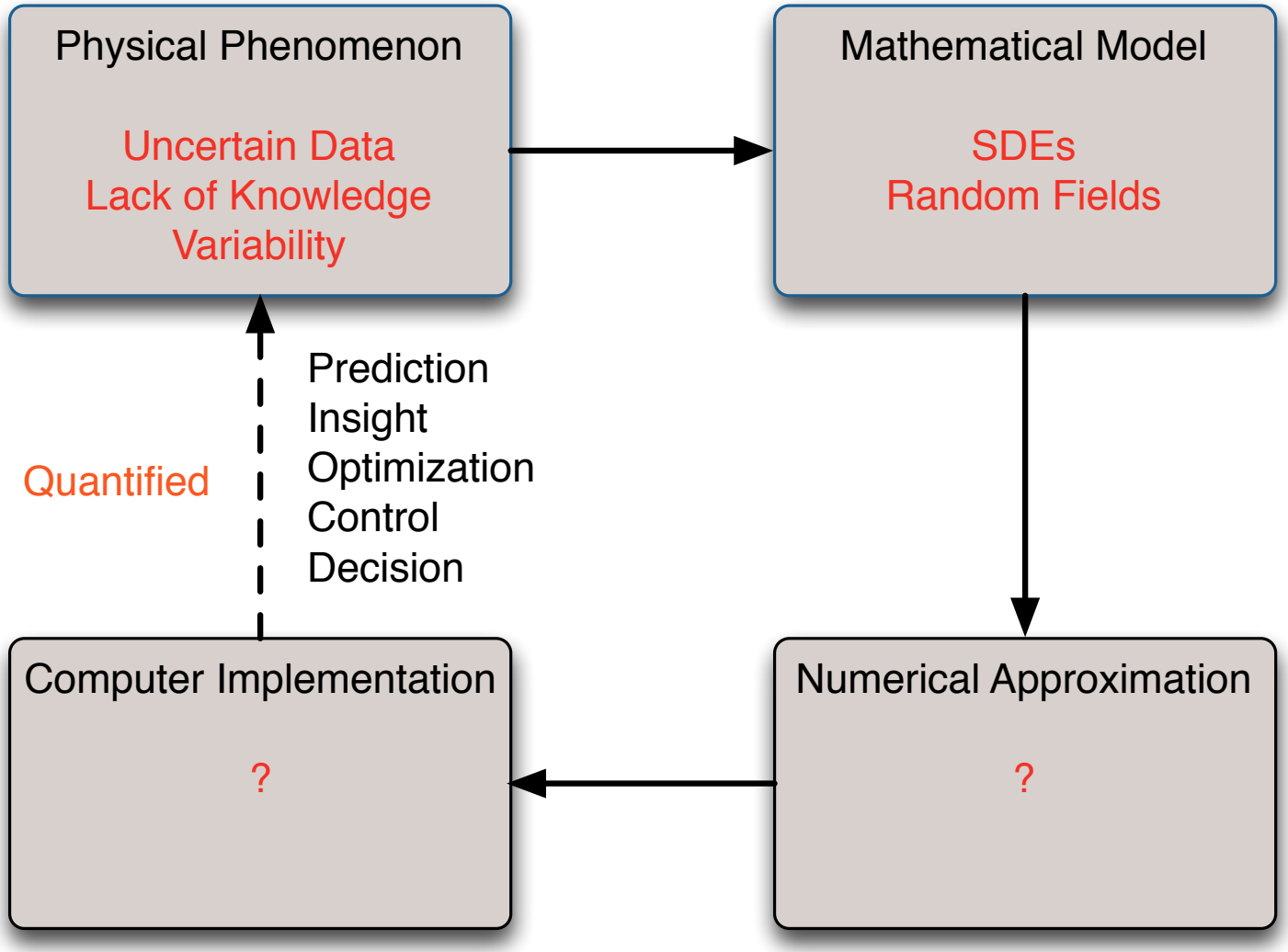
What is Uncertainty Quantification (UQ)?

Uncertainty Quantification and the Scientific Computing Paradigm



What is Uncertainty Quantification (UQ)?

Uncertainty Quantification and the Scientific Computing Paradigm



What is Uncertainty Quantification (UQ)?

The "Fruit Fly" of UQ

The most popular **model problem** in the UQ community is the steady-state diffusion problem with uncertain coefficient function:

$$-\nabla\cdot(k\nabla p) = f \qquad \text{on domain} \quad D \subset \mathbb{R}^d.$$

(an elliptic partial differential equation modelling many processes)

What is Uncertainty Quantification (UQ)?

The "Fruit Fly" of UQ

The most popular **model problem** in the UQ community is the steady-state diffusion problem with uncertain coefficient function:

$$-\nabla\cdot(k\nabla p) = f \qquad \text{on domain} \quad D \subset \mathbb{R}^d.$$

(an elliptic partial differential equation modelling many processes)

Typically only interested in a functional Q of the solution p, known as a quantity of interest (QoI):

$$\text{e.g.} \qquad Q(p) = p(\mathbf{x}_0) \quad \text{or} \quad Q(p) = \frac{1}{|D_0|}\int_{D_0} p(\mathbf{x})\,\mathrm{d}\mathbf{x}.$$

How can uncertainty in the coefficient k be addressed?

What is Uncertainty Quantification (UQ)?

The "Fruit Fly" of UQ

The most popular **model problem** in the UQ community is the steady-state diffusion problem with uncertain coefficient function:

$$-\nabla\cdot(k\nabla p) = f \qquad \text{on domain} \quad D \subset \mathbb{R}^d.$$

(an elliptic partial differential equation modelling many processes)

Typically only interested in a functional Q of the solution p, known as a quantity of interest (QoI):

$$\text{e.g.} \qquad Q(p) = p(\mathbf{x}_0) \quad \text{or} \quad Q(p) = \frac{1}{|D_0|}\int_{D_0} p(\mathbf{x})\,\mathrm{d}\mathbf{x}.$$

How can uncertainty in the coefficient k be addressed?

- Worst case analysis: Calculate *uncertainty interval*

$$\mathscr{I} = \Big[\inf_{\|k-k_0\|<\varepsilon} Q(p(k)), \sup_{\|k-k_0\|<\varepsilon} Q(p(k))\Big].$$

What is Uncertainty Quantification (UQ)?

Probabilistic model

But, in general, some coefficients with $\|k - k_0\| < \varepsilon$ are more likely than others

$$\Rightarrow \quad \textbf{Probabilistic approach}$$

- Introduce probability measure on $S := \{k : \|k - k_0\| < \varepsilon\}$.

What is Uncertainty Quantification (UQ)?

Probabilistic model

But, in general, some coefficients with $\|k - k_0\| < \varepsilon$ are more likely than others

$$\Rightarrow \quad \textbf{Probabilistic approach}$$

- Introduce probability measure on $S := \{k : \|k - k_0\| < \varepsilon\}$.
- Then $Q(p(\cdot))$, as a (measurable) mapping on S, induces a probability measure for the QoI (Uncertainty Propagation)
- **Big issue:** choice of distribution, information too subjective?

What is Uncertainty Quantification (UQ)?

Probabilistic model

But, in general, some coefficients with $\|k - k_0\| < \varepsilon$ are more likely than others

$$\Rightarrow \quad \textbf{Probabilistic approach}$$

- Introduce probability measure on $S := \{k : \|k - k_0\| < \varepsilon\}$.
- Then $Q(p(\cdot))$, as a (measurable) mapping on S, induces a probability measure for the QoI (Uncertainty Propagation)
- **Big issue:** choice of distribution, information too subjective?
- The point of departure for *Bayesian inference* is to choose the distribution based on (output) data (Uncertainty Quantification)
  ("prior" distribution on S becomes less important)

What is Uncertainty Quantification (UQ)?

Probabilistic model

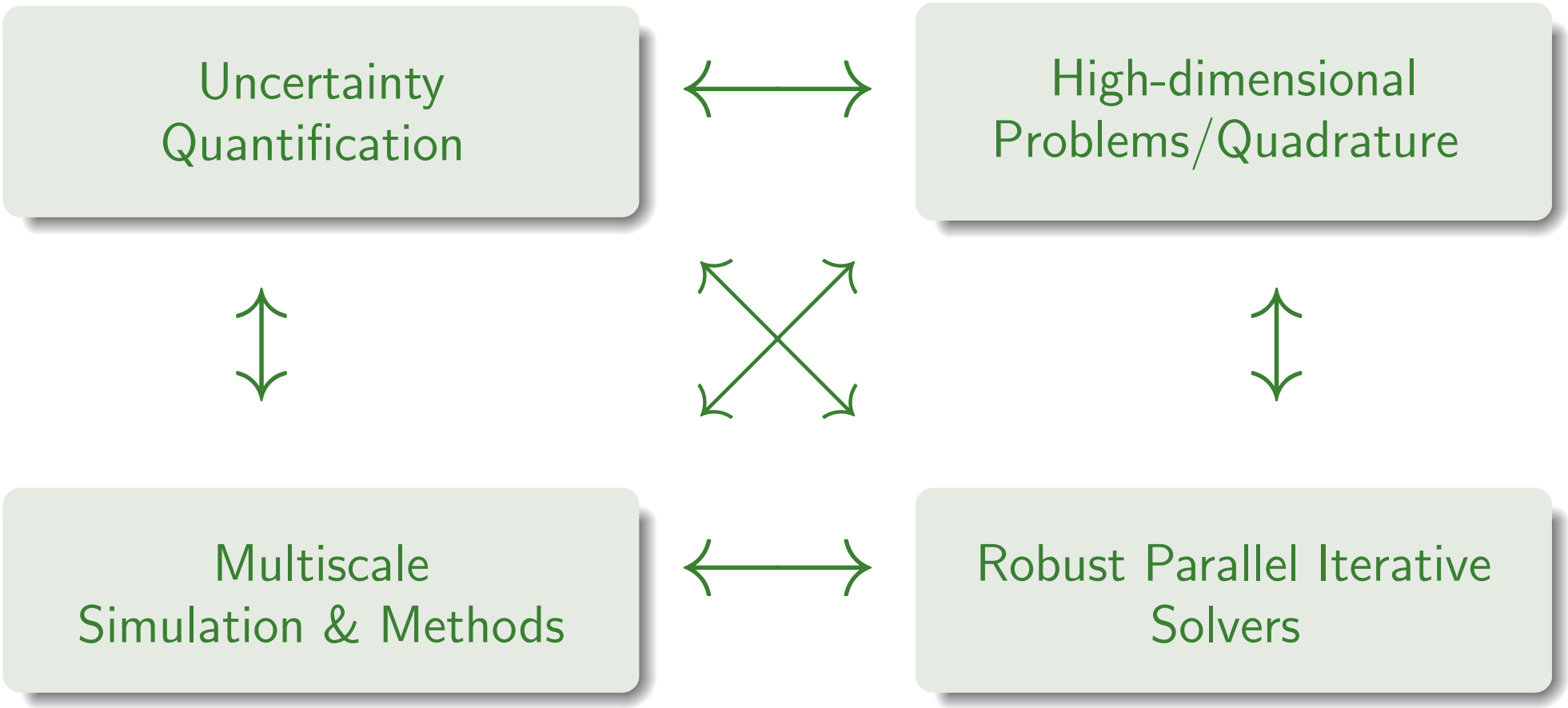
But, in general, some coefficients with $\|k - k_0\| < \varepsilon$ are more likely than others

$$\Rightarrow \quad \textbf{Probabilistic approach}$$

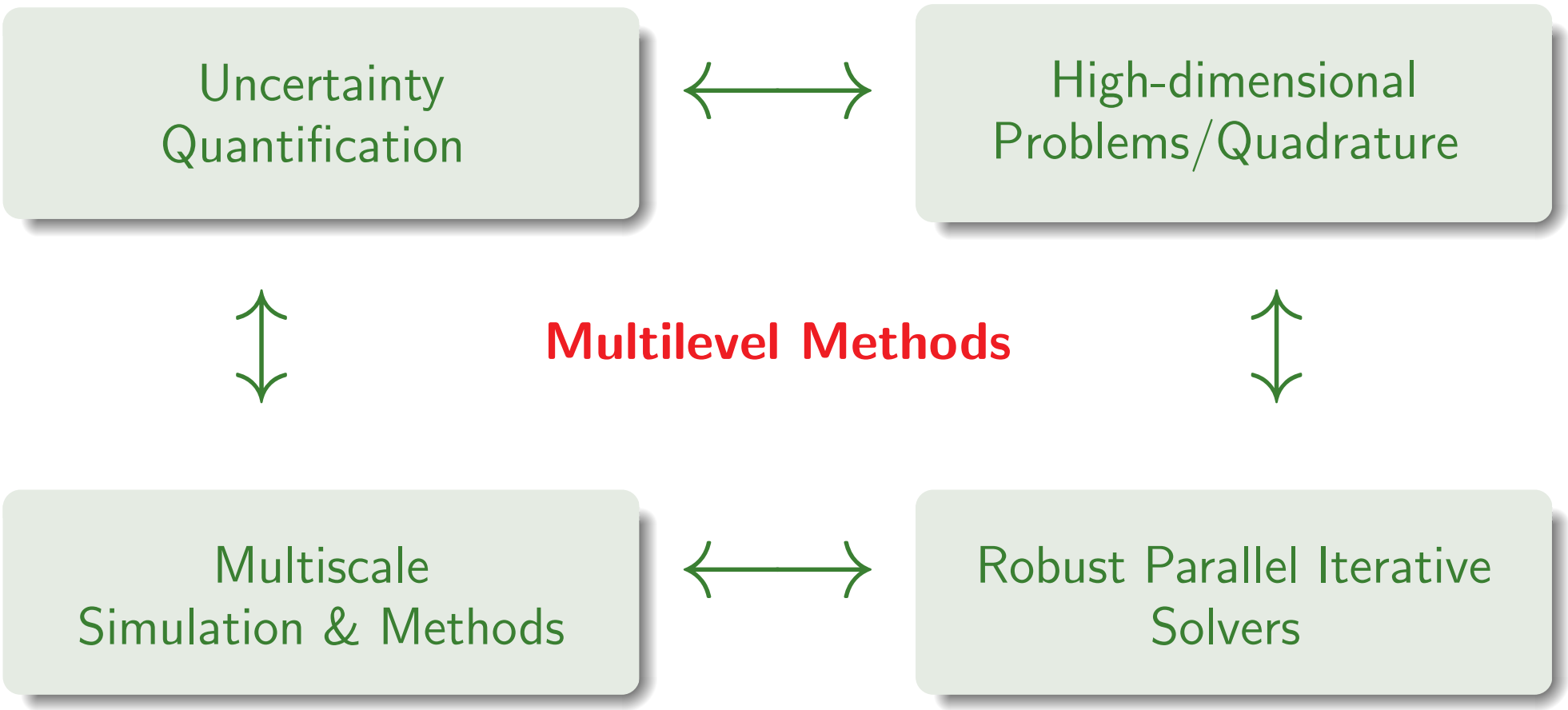
- Introduce probability measure on $S := \{k : \|k - k_0\| < \varepsilon\}$.
- Then $Q(p(\cdot))$, as a (measurable) mapping on S, induces a probability measure for the QoI (Uncertainty Propagation)
- **Big issue:** choice of distribution, information too subjective?
- The point of departure for *Bayesian inference* is to choose the distribution based on (output) data (Uncertainty Quantification)
  ("prior" distribution on S becomes less important)
- **Additional goal:** robust and efficient deterministic solvers and discretisations for multiscale problems

First Lecture!

My Research


Uncertainty Quantification
High-dimensional Problems/Quadrature
Multiscale Simulation & Methods
Robust Parallel Iterative Solvers

My Research

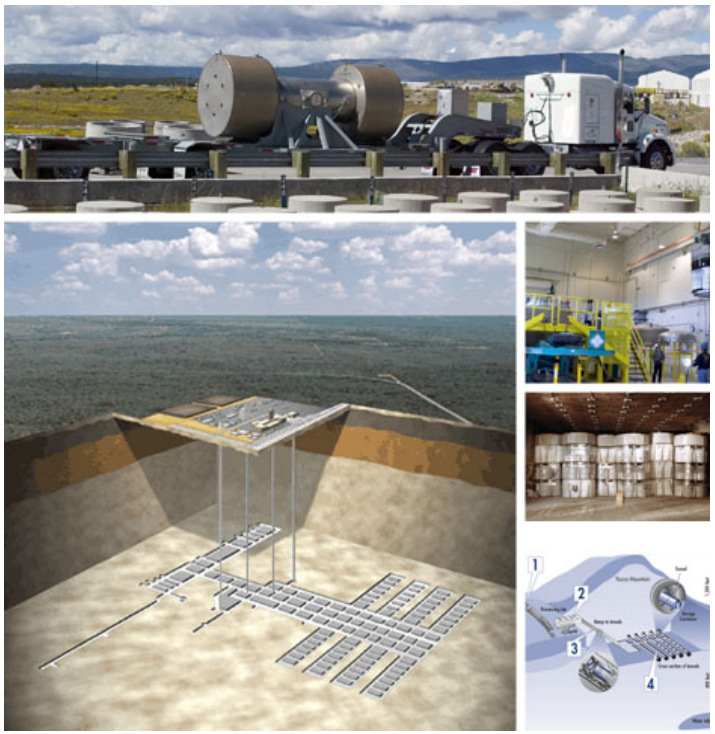


Numerical Analysis / Method Design / Applications

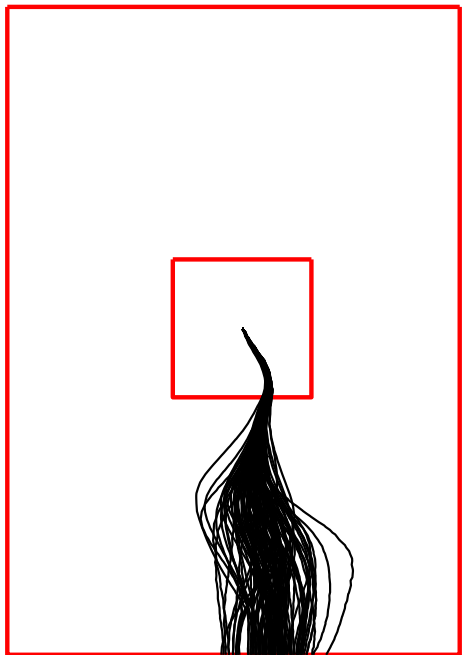
A Case Study: Radioactive Waste Disposal

EPSRC project with UK Nuclear Decommisioning Authority (with Cliffe & Giles)

UQ scenario: accidental release of radionuclides and then transport by groundwater. Quantity of interest: $\log_{10}$ of particle travel time.



WIPP repository, NM



A Case Study: Radioactive Waste Disposal

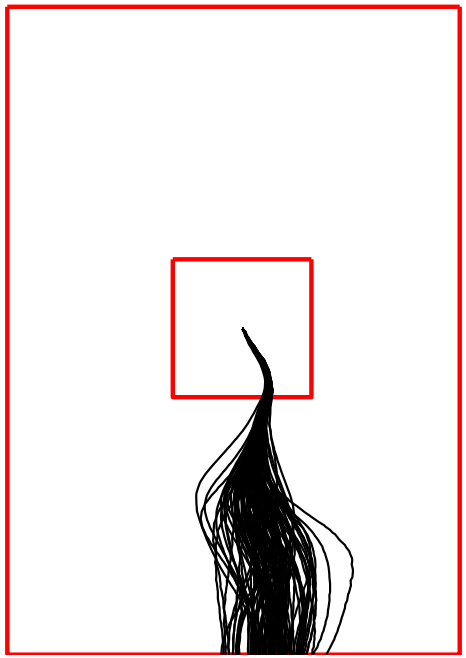
EPSRC project with UK Nuclear Decommisioning Authority (with Cliffe & Giles)

UQ scenario: accidental release of radionuclides and then transport by groundwater. Quantity of interest: $\log_{10}$ of particle travel time.

- Steady-state *Darcy flow* for *pressure* p:

$$-\nabla\cdot\Big(k\nabla p\Big) = 0 \quad \text{on } D, \quad p = p_0 \quad \text{on } \partial D.$$

- *Permeability* k modelled as **random field**.



A Case Study: Radioactive Waste Disposal

EPSRC project with UK Nuclear Decommisioning Authority (with Cliffe & Giles)

UQ scenario: accidental release of radionuclides and then transport by groundwater. Quantity of interest: $\log_{10}$ of particle travel time.

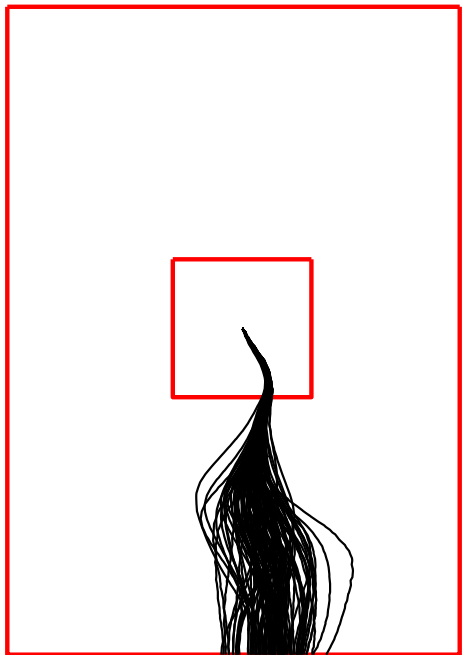
- Steady-state *Darcy flow* for *pressure* p:

$$-\nabla\cdot\Big(k\nabla p\Big) = 0 \quad \text{on } D, \quad p = p_0 \quad \text{on } \partial D.$$

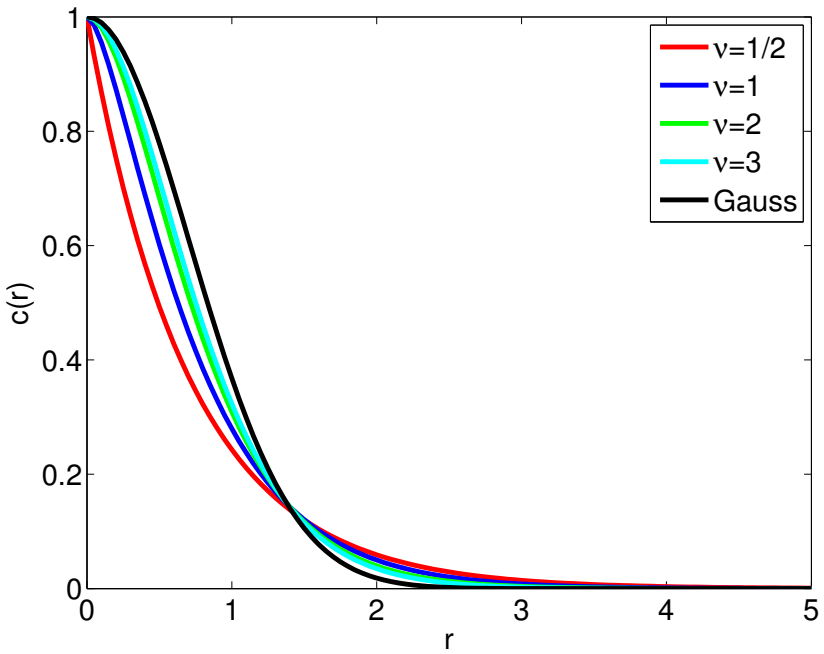
- *Permeability* k modelled as **random field**.
- In hydrology typically **lognormal**:

  $\log k(\mathbf{x},\omega) =: Z(\mathbf{x},\omega)$ **Gaussian** w. *mean* $\overline{Z}(\mathbf{x})$ and stationary and isotropic *covariance*

$$\mathbb{E}\left[\left(Z(\mathbf{x},\cdot)-\overline{Z}(\mathbf{x})\right)\left(Z(\mathbf{y},\cdot)-\overline{Z}(\mathbf{y})\right)\right] = c\left(\|\mathbf{x}-\mathbf{y}\|_2\right)$$



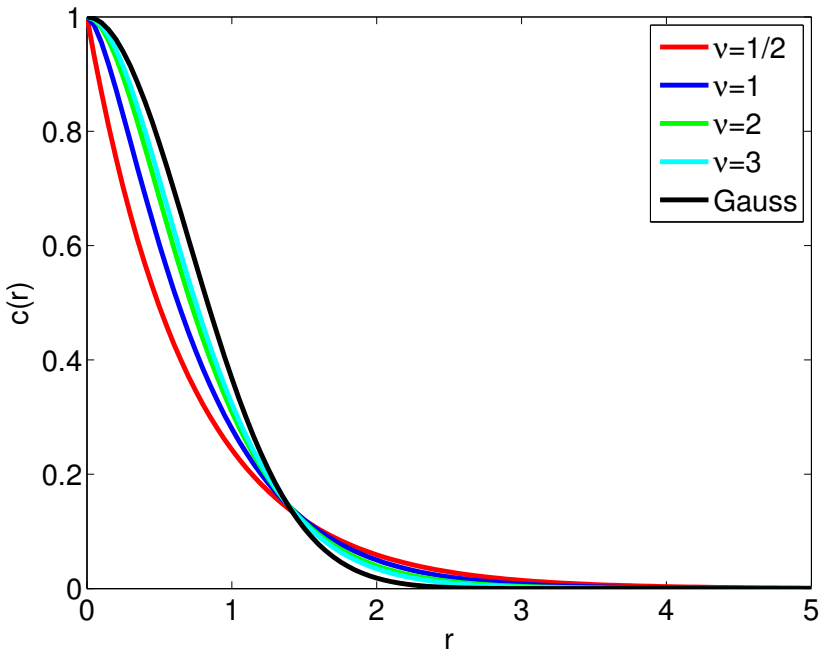
Matérn Family of Covariance Functions



Matérn covariance functions $c(r)$

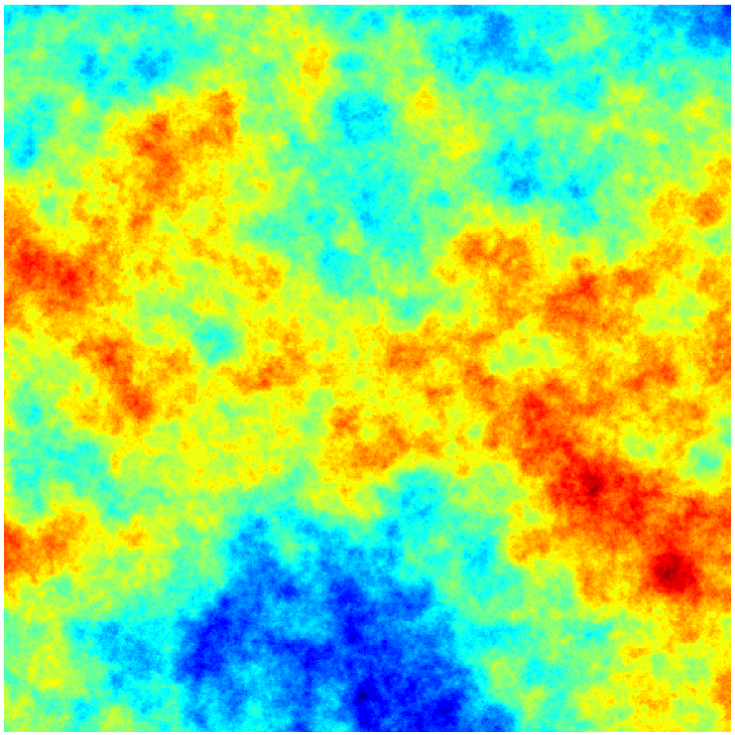
e.g. for $\nu = 0.5$: $c(r) := \sigma^2 \exp\left(-\frac{r}{\lambda}\right)$

Matérn Family of Covariance Functions



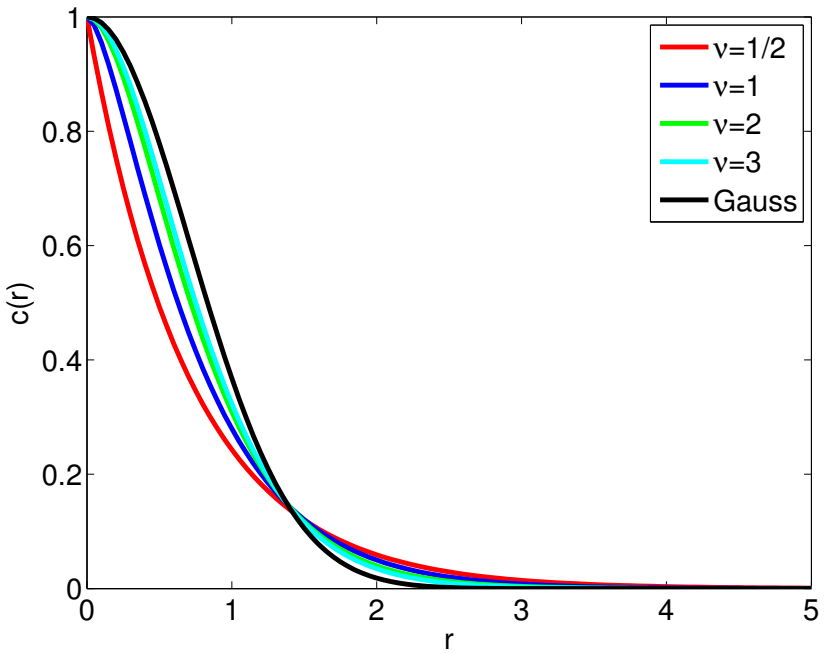
Matérn covariance functions $c(r)$

e.g. for $\nu = 0.5$: $c(r) := \sigma^2 \exp\left(-\frac{r}{\lambda}\right)$



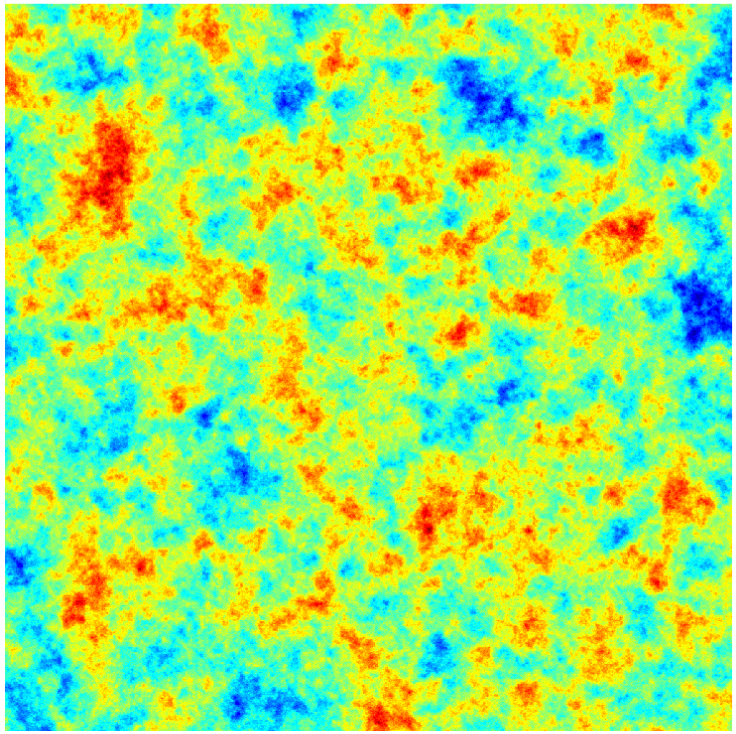
$\nu = 0.5,\ \lambda = 0.5$

Matérn Family of Covariance Functions



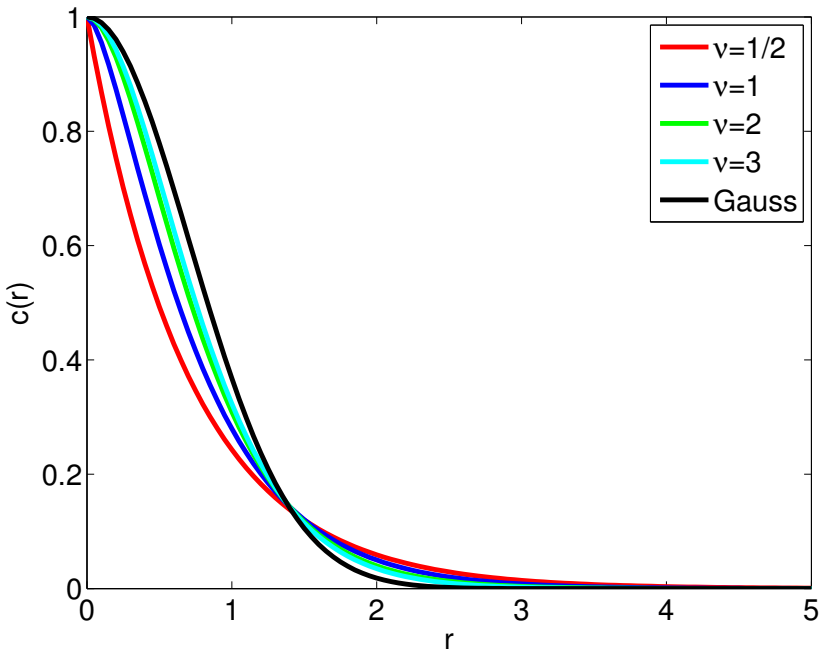
Matérn covariance functions $c(r)$

e.g. for $\nu = 0.5$: $c(r) := \sigma^2 \exp\left(-\frac{r}{\lambda}\right)$



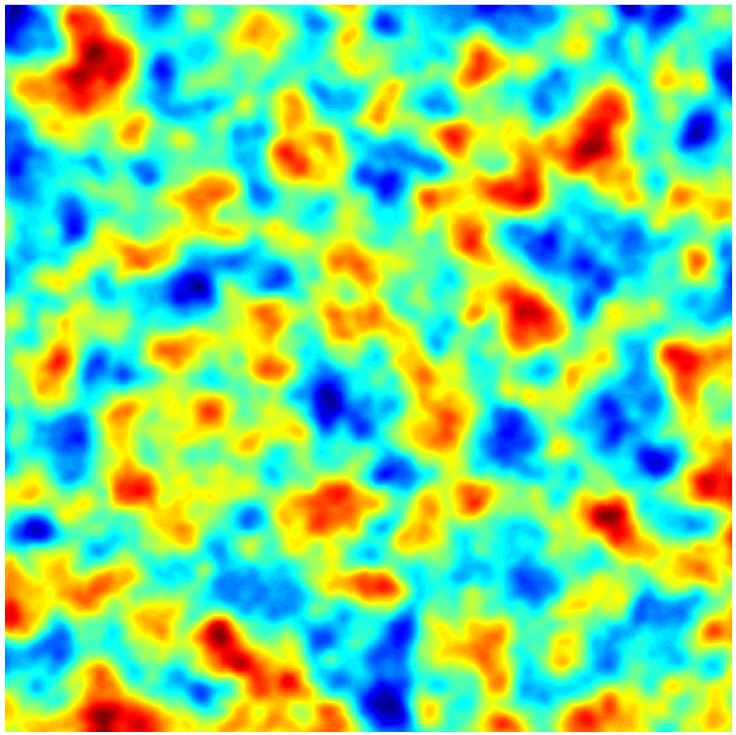
$\nu = 0.5,\ \lambda = 0.05$

Matérn Family of Covariance Functions



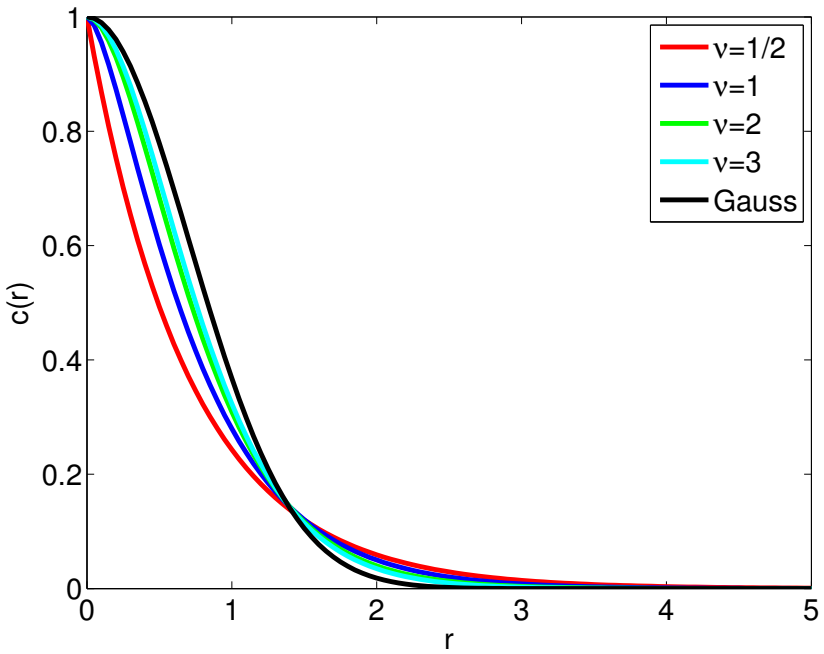
Matérn covariance functions $c(r)$

e.g. for $\nu = 0.5$: $c(r) := \sigma^2 \exp\left(-\frac{r}{\lambda}\right)$



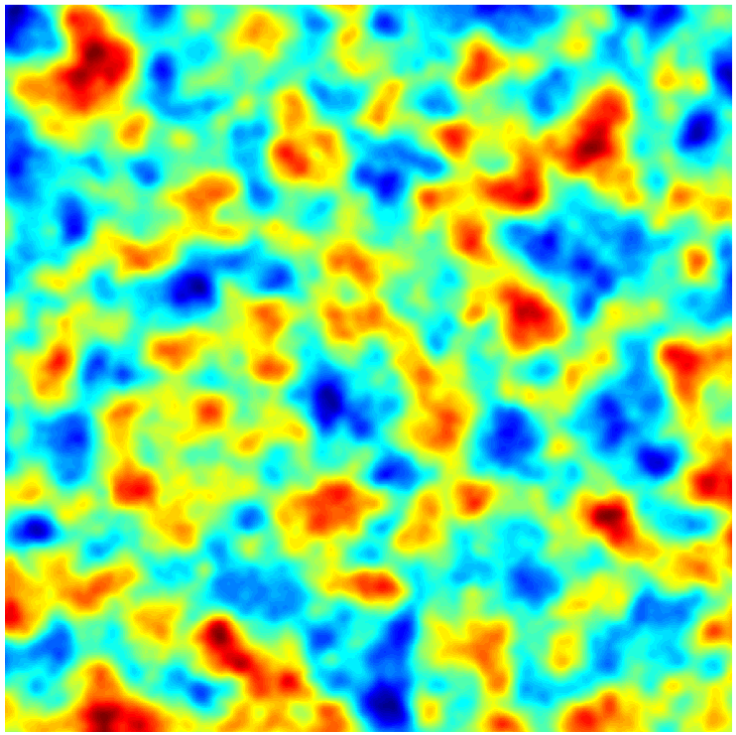
$\nu = 2.5,\ \lambda = 0.05$

Matérn Family of Covariance Functions



Matérn covariance functions $c(r)$

e.g. for $\nu = 0.5$: $c(r) := \sigma^2 \exp\left(-\frac{r}{\lambda}\right)$



$\nu = 2.5,\ \lambda = 0.05$

Smoothness: Realisations $Z(\cdot, \omega) \in C^\eta(D)$ (Hölder), for any $\eta < \nu$,

Typically $\nu < 1$, $\lambda \ll 1 \implies$ **fine** mesh $h \ll 1$

Sampling from $Z = \log k$

Karhunen-Loève expansion ($\mu_m \downarrow 0$ as $m \to \infty$):

$$Z(x,\omega) = \overline{Z}(\mathbf{x}) + \sum_{m=1}^{\infty} \sqrt{\mu_m}\, \phi_m(\mathbf{x})\, Z_m(\omega)$$

$\{\phi_m\}_{m\in\mathbb{N}}$ are the normalised eigenfunctions and $Z_m \sim N(0,1)$ (i.i.d.)

Sampling from $Z = \log k$

In practice: truncated Karhunen-Loève expansion:

$$Z^s(x,\omega) = \overline{Z}(\mathbf{x}) + \sum_{m=1}^{s} \sqrt{\mu_m}\,\phi_m(\mathbf{x})\,Z_m(\omega)$$

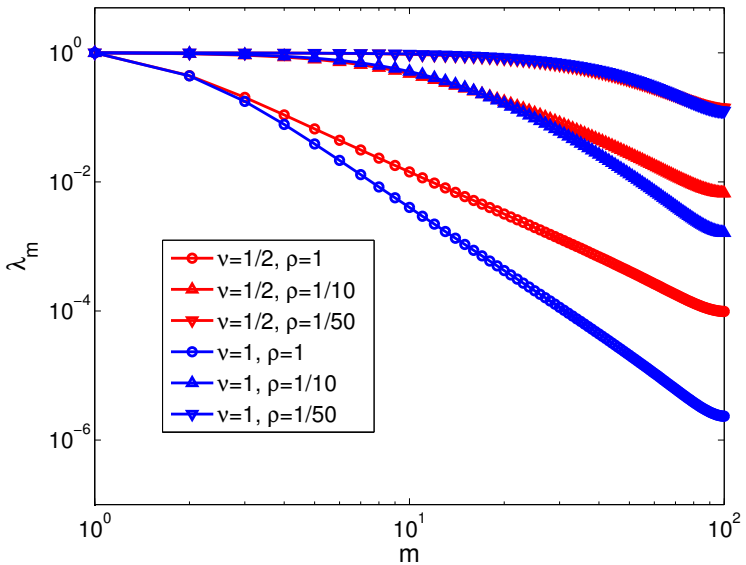
$\{\phi_m\}_{m\leq s}$ are the normalised eigenfunctions and $Z_m \sim N(0,1)$ (i.i.d.)

Sampling from $Z = \log k$

In practice: truncated Karhunen-Loève expansion:

$$Z^s(x,\omega) = \overline{Z}(\mathbf{x}) + \sum_{m=1}^{s} \sqrt{\mu_m}\,\phi_m(\mathbf{x})\,Z_m(\omega)$$

$\{\phi_m\}_{m\le s}$ are the normalised eigenfunctions and $Z_m \sim N(0,1)$ (i.i.d.)



Typically $\lambda \ll 1$ and $\nu < 1$

$\Longrightarrow$

slow KL-eigenvalue decay

$\Longrightarrow$

high stochastic dimension $s \gg 1$

Uncertainty Propagation

The Forward Problem

Uncertainty Propagation – High-dimensional Quadrature

$$Z(\mathbf{x},\omega) \in X \xrightarrow{\text{Model}} p(\mathbf{x},\omega) \in V \xrightarrow{\text{Output}} Q(\omega) \in \mathbb{R}(\text{or } \mathbb{R}^J) \xrightarrow{\text{Statistics}} \mathbb{E}[Q]$$

RF input — RF output — quantity of interest

Uncertainty Propagation – High-dimensional Quadrature

$$Z(\mathbf{x},\omega) \in X \overset{\text{Model}}{\longrightarrow} p(\mathbf{x},\omega) \in V \overset{\text{Output}}{\longrightarrow} Q(\omega) \in \mathbb{R}(\text{or } \mathbb{R}^J) \overset{\text{Statistics}}{\longrightarrow} \mathbb{E}[Q]$$

RF input — RF output — quantity of interest

- In practice, we need to use **finite representations** of Z and p:
  (e.g. truncated KL expansion and FE discretisation of the PDE)

$$\mathbf{Z}(\omega) \in \mathbb{R}^s \overset{\text{Model}(h)}{\longrightarrow} \mathbf{P}(\omega) \in \mathbb{R}^{M_h} \overset{\text{Output}}{\longrightarrow} Q_{h,s}(\omega) \in \mathbb{R}(\text{or } \mathbb{R}^J) \overset{\text{Quadrature}}{\longrightarrow} \widehat{Q}_{h,s}$$

random input — FE state vector — quantity of interest

Uncertainty Propagation – High-dimensional Quadrature

$$Z(\mathbf{x},\omega) \in X \xrightarrow{\text{Model}} p(\mathbf{x},\omega) \in V \xrightarrow{\text{Output}} Q(\omega) \in \mathbb{R}(\text{or } \mathbb{R}^J) \xrightarrow{\text{Statistics}} \mathbb{E}[Q]$$

RF input — RF output — quantity of interest

- In practice, we need to use **finite representations** of Z and p:
  (e.g. truncated KL expansion and FE discretisation of the PDE)

$$\mathbf{Z}(\omega) \in \mathbb{R}^s \xrightarrow{\text{Model}(h)} \mathbf{P}(\omega) \in \mathbb{R}^{M_h} \xrightarrow{\text{Output}} Q_{h,s}(\omega) \in \mathbb{R}(\text{or } \mathbb{R}^J) \xrightarrow{\text{Quadrature}} \widehat{Q}_{h,s}$$

random input — FE state vector — quantity of interest

- Consider only equal weight quadrature rules

$$\widehat{Q}_{h,s} := \frac{1}{N} \sum_{i=1}^{N} Q_{h,s}(\mathbf{Z}^{(i)})$$

- Monte Carlo: $\mathbf{Z}^{(i)}$ i.i.d. random $(\mathrm{N}(\mathbf{0}, I)$ for lognormal diffusion)

Abstract Complexity Analysis

- Depending on how smooth $Q_{h,s}$ is w.r.t. $\mathbf{Z}$, the **quadrature error** is

$$\left\| \widehat{Q}_{h,s} - \mathbb{E}[Q_{h,s}] \right\| = \mathscr{O}(N^{-\eta}), \qquad \text{for some } \eta > 0$$

Abstract Complexity Analysis

- Depending on how smooth $Q_{h,s}$ is w.r.t. $\mathbf{Z}$, the **quadrature error** is

$$\left\| \widehat{Q}_{h,s} - \mathbb{E}[Q_{h,s}] \right\| = \mathscr{O}(N^{-\eta}), \qquad \text{for some } \eta > 0$$

- For **discretisation & truncation error** we assume weak convergence:

$$\left\| \mathbb{E}[Q_{h,s} - Q] \right\| = \mathscr{O}(M_h^{-\alpha}), \qquad \text{for some } \alpha > 0$$

Abstract Complexity Analysis

- Depending on how smooth $Q_{h,s}$ is w.r.t. $\mathbf{Z}$, the **quadrature error** is

$$\left\| \widehat{Q}_{h,s} - \mathbb{E}[Q_{h,s}] \right\| = \mathscr{O}(N^{-\eta}), \qquad \text{for some } \eta > 0$$

- For **discretisation & truncation error** we assume weak convergence:

$$\left\| \mathbb{E}[Q_{h,s} - Q] \right\| = \mathscr{O}(M_h^{-\alpha}), \qquad \text{for some } \alpha > 0$$

- Then the **total error** is:

$$\left\| \widehat{Q}_{h,s} - \mathbb{E}[Q] \right\| \leq \left\| \widehat{Q}_{h,s} - \mathbb{E}[Q_{h,s}] \right\| + \left\| \mathbb{E}[Q_{h,s} - Q] \right\| = \mathscr{O}(M_h^{-\alpha} + N^{-\eta})$$

Abstract Complexity Analysis

- Depending on how smooth $Q_{h,s}$ is w.r.t. $\mathbf{Z}$, the **quadrature error** is

$$\left\| \widehat{Q}_{h,s} - \mathbb{E}[Q_{h,s}] \right\| = \mathscr{O}(N^{-\eta}), \qquad \text{for some } \eta > 0$$

- For **discretisation & truncation error** we assume weak convergence:

$$\left\| \mathbb{E}[Q_{h,s} - Q] \right\| = \mathscr{O}(M_h^{-\alpha}), \qquad \text{for some } \alpha > 0$$

- Then the **total error** is:

$$\left\| \widehat{Q}_{h,s} - \mathbb{E}[Q] \right\| \leq \left\| \widehat{Q}_{h,s} - \mathbb{E}[Q_{h,s}] \right\| + \left\| \mathbb{E}[Q_{h,s} - Q] \right\| = \mathscr{O}(M_h^{-\alpha} + N^{-\eta})$$

- Assuming cost per sample $\text{Cost}(Q_{h,s}^{(i)}) = \mathscr{O}(M_h^{\gamma})$, for some $\gamma \geq 1$.

Abstract Complexity Analysis

- Depending on how smooth $Q_{h,s}$ is w.r.t. $\mathbf{Z}$, the **quadrature error** is

$$\left\| \widehat{Q}_{h,s} - \mathbb{E}[Q_{h,s}] \right\| = \mathscr{O}(N^{-\eta}), \qquad \text{for some } \eta > 0$$

- For **discretisation & truncation error** we assume weak convergence:

$$\left\| \mathbb{E}[Q_{h,s} - Q] \right\| = \mathscr{O}(M_h^{-\alpha}), \qquad \text{for some } \alpha > 0$$

- Then the **total error** is:

$$\left\| \widehat{Q}_{h,s} - \mathbb{E}[Q] \right\| \leq \left\| \widehat{Q}_{h,s} - \mathbb{E}[Q_{h,s}] \right\| + \left\| \mathbb{E}[Q_{h,s} - Q] \right\| = \mathscr{O}(M_h^{-\alpha} + N^{-\eta})$$

- Assuming cost per sample $\text{Cost}(Q_{h,s}^{(i)}) = \mathscr{O}(M_h^{\gamma})$, for some $\gamma \geq 1$.

The ε-Cost (i.e. the cost to achieve error $< \varepsilon$) is

$$\mathscr{C}_\varepsilon(\widehat{Q}_{h,s}) = \mathscr{O}(NM_h^{\gamma}) = \mathscr{O}\left(\varepsilon^{-1/\eta - \gamma/\alpha}\right)$$

Example: Standard Monte Carlo

2D lognormal Darcy (with $\nu = 1/2$), AMG solver, $Q = \| - k \frac{\partial p}{\partial x_1} \|_{L^1(D)}$

- Standard MC: $\eta = 1/2$ Independent of s !

Example: Standard Monte Carlo

2D lognormal Darcy (with $\nu = 1/2$), AMG solver, $Q = \| - k \frac{\partial p}{\partial x_1} \|_{L^1(D)}$

- Standard MC: $\eta = 1/2$ Independent of s !
- Numerically observed cost/sample: $\approx \mathscr{O}(M_h) = \mathscr{O}(h^{-2}) \implies \gamma \approx 1.$
- Numerically observed FE-error: $\approx \mathscr{O}(M_h^{-3/8}) \implies \alpha \approx 3/8.$
  (rigorous analysis also exists)

Example: Standard Monte Carlo

2D lognormal Darcy (with $\nu = 1/2$), AMG solver, $Q = \| - k \frac{\partial p}{\partial x_1} \|_{L^1(D)}$

- Standard MC: $\eta = 1/2$ Independent of s !
- Numerically observed cost/sample: $\approx \mathscr{O}(M_h) = \mathscr{O}(h^{-2}) \implies \gamma \approx 1.$
- Numerically observed FE-error: $\approx \mathscr{O}(M_h^{-3/8}) \implies \alpha \approx 3/8.$
  (rigorous analysis also exists)
- Hence, **total cost** to get error $\mathscr{O}(\varepsilon)$: $\approx \mathscr{O}(\varepsilon^{-14/3})$
  to get error reduction by a factor 2 $\rightarrow$ cost has to grow by a factor 25!

Example: Standard Monte Carlo

2D lognormal Darcy (with $\nu = 1/2$), AMG solver, $Q = \| - k \frac{\partial p}{\partial x_1} \|_{L^1(D)}$

- Standard MC: $\eta = 1/2$ Independent of s !
- Numerically observed cost/sample: $\approx \mathscr{O}(M_h) = \mathscr{O}(h^{-2}) \implies \gamma \approx 1.$
- Numerically observed FE-error: $\approx \mathscr{O}(M_h^{-3/8}) \implies \alpha \approx 3/8.$
  (rigorous analysis also exists)
- Hence, **total cost** to get error $\mathscr{O}(\varepsilon)$: $\approx \mathscr{O}(\varepsilon^{-14/3})$
  to get error reduction by a factor 2 $\rightarrow$ cost has to grow by a factor 25!

Case 1: $\sigma^2 = 1$, $\lambda = 0.3$, $\nu = 0.5$

ε	h^{-1}	N	Cost
0.01	129	1.4×10^4	21 min
0.002	1025	3.5×10^5	30 days

Case 2: $\sigma^2 = 3$, $\lambda = 0.1$, $\nu = 0.5$

ε	h^{-1}	N	Cost
0.01	513	8.5×10^3	4 h
0.002	**Prohibitively large!!**		

(actual numbers & CPU times from 2010 on cluster of 2GHz Intel T7300 procs)

Multilevel Quadrature [Heinrich, 1998], [Giles, 2008], ...

- **Key idea:** "sample" from Q on a hierarchy of levels with different discretization parameters $h_0, \dots, h_L = h$ and $s_0, \dots, s_L = s$, and use

$$Q_{h,s} = \underbrace{Q_0}_{=:Y_0} + \sum_{\ell=1}^{L} \underbrace{(Q_\ell - Q_{\ell-1})}_{=:Y_\ell} = \sum_{\ell=0}^{L} Y_\ell \quad \text{(telescoping sum)}$$

  e.g. uniform mesh refinement $h_\ell = h_{\ell-1}/2$ (write $Q_\ell := Q_{h_\ell, s_\ell}$).

Multilevel Quadrature [Heinrich, 1998], [Giles, 2008], ...

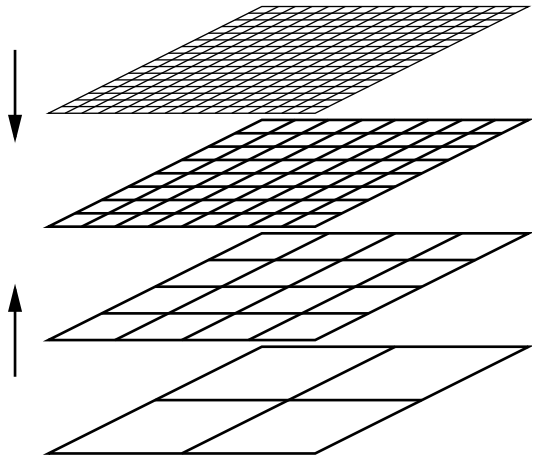
- **Key idea:** "sample" from Q on a hierarchy of levels with different discretization parameters $h_0, \dots, h_L = h$ and $s_0, \dots, s_L = s$, and use

$$Q_{h,s} = \underbrace{Q_0}_{=:Y_0} + \sum_{\ell=1}^{L} \underbrace{(Q_\ell - Q_{\ell-1})}_{=:Y_\ell} = \sum_{\ell=0}^{L} Y_\ell \quad \text{(telescoping sum)}$$

  e.g. uniform mesh refinement $h_\ell = h_{\ell-1}/2$ (write $Q_\ell := Q_{h_\ell, s_\ell}$).

Given a (single-level) quadrature rule $\widehat{Y}_\ell$, define the **multilevel quadrature rule** for $\mathbb{E}[Q]$ by

$$\widehat{Q}^{\mathrm{ML}}_{h,s} := \sum_{\ell=0}^{L} \widehat{Y}_\ell \,.$$



Multilevel Quadrature [Heinrich, 1998], [Giles, 2008], ...

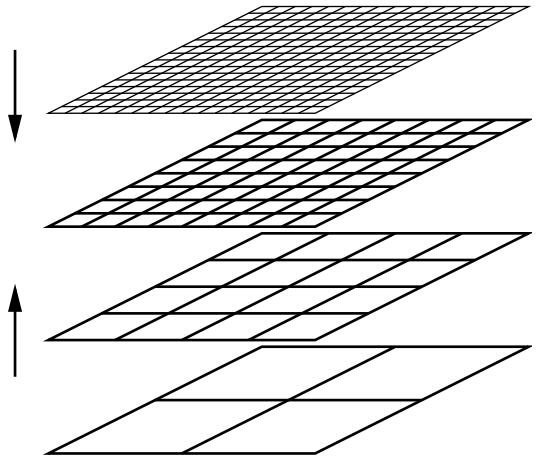
- **Key idea:** "sample" from Q on a hierarchy of levels with different discretization parameters $h_0, \dots, h_L = h$ and $s_0, \dots, s_L = s$, and use

$$Q_{h,s} = \underbrace{Q_0}_{=:Y_0} + \sum_{\ell=1}^{L} \underbrace{(Q_\ell - Q_{\ell-1})}_{=:Y_\ell} = \sum_{\ell=0}^{L} Y_\ell \quad \text{(telescoping sum)}$$

  e.g. uniform mesh refinement $h_\ell = h_{\ell-1}/2$ (write $Q_\ell := Q_{h_\ell, s_\ell}$).

Given a (single-level) quadrature rule $\widehat{Y}_\ell$, define the **multilevel quadrature rule** for $\mathbb{E}[Q]$ by

$$\widehat{Q}^{\mathrm{ML}}_{h,s} := \sum_{\ell=0}^{L} \widehat{Y}_\ell \,.$$



- **e.g. Multilevel Monte Carlo** $Q^{\mathrm{MLMC}}_{h,s} := \sum_{\ell=0}^{L} \left(\frac{1}{N_\ell} \sum_{i=1}^{N_\ell} Y_\ell\left(\mathbf{Z}^{(i)}\right) \right)$

Where does the cost reduction come from?

Multilevel Monte Carlo case

- **Level** 0: $N_0 \approx N^{\mathrm{MC}} \approx \varepsilon^{-2}$, but cost per sample reduced to

$$\mathrm{Cost}(Y_0^{(i)}) = \mathrm{Cost}(Q_0^{(i)}) = \mathscr{O}\left(h_0^{-\gamma}\right) = \mathscr{O}(1)$$

Hence, $\boxed{\mathscr{C}_\varepsilon(\widehat{Y}_0) = \mathscr{O}(\varepsilon^{-2})}$.

Where does the cost reduction come from?

Multilevel Monte Carlo case

- **Level** 0: $N_0 \eqsim N^{\mathrm{MC}} \eqsim \varepsilon^{-2}$, but cost per sample reduced to

$$\mathrm{Cost}(Y_0^{(i)}) = \mathrm{Cost}(Q_0^{(i)}) = \mathscr{O}\left(h_0^{-\gamma}\right) = \mathscr{O}(1)$$

  Hence, $\boxed{\mathscr{C}_\varepsilon(\widehat{Y}_0) = \mathscr{O}(\varepsilon^{-2})}$.

- **Level** L: provided discretisation error converges strongly (pathwise)

$$\|Y_L^{(i)}\| = \|Q_L(\mathbf{Z}^{(i)}) - Q_{L-1}(\mathbf{Z}^{(i)})\| = \mathscr{O}(h_L^\alpha) = \mathscr{O}(\varepsilon)\,.$$

  (Assuming same weak and strong convergence rates – not always the case!)

Where does the cost reduction come from?

Multilevel Monte Carlo case

- **Level** 0: $N_0 \eqsim N^{\mathrm{MC}} \eqsim \varepsilon^{-2}$, but cost per sample reduced to

$$\mathrm{Cost}(Y_0^{(i)}) = \mathrm{Cost}(Q_0^{(i)}) = \mathscr{O}\left(h_0^{-\gamma}\right) = \mathscr{O}(1)$$

Hence, $\boxed{\mathscr{C}_\varepsilon(\widehat{Y}_0) = \mathscr{O}(\varepsilon^{-2})}$.

- **Level** L: provided discretisation error converges strongly (pathwise)

$$\|Y_L^{(i)}\| = \|Q_L(\mathbf{Z}^{(i)}) - Q_{L-1}(\mathbf{Z}^{(i)})\| = \mathscr{O}(h_L^\alpha) = \mathscr{O}(\varepsilon)\,.$$

(Assuming same weak and strong convergence rates – not always the case!)

Then the Law of Large Numbers gives

$$\mathbb{E}\left[\mathbb{E}[Y_L] - \widehat{Y}_L^{\mathrm{MC}}\right]^{1/2} = \mathscr{O}(\varepsilon) N_L^{-1/2}\,.$$

We can choose $N_L = \mathscr{O}(1)$ and $\boxed{\mathscr{C}_\varepsilon(\widehat{Y}_L) = \mathscr{O}(h_L^{-\gamma}) = \mathscr{O}(\varepsilon^{-\gamma/\alpha})}$.

Multilevel Quadrature

Abstract Complexity Theorem (General Case)

Theorem (Cliffe, Giles, RS, Teckentrup, '11 & Teckentrup, Thesis '13)

Suppose there are constants $\alpha, \beta, \gamma, \eta > 0$ *s.t., for all* $\ell = 0, \ldots, L$,

(M1) $\|\mathbb{E}[Q_\ell - Q]\| = \mathcal{O}(M_\ell^{-\alpha})$,

(M2) $\|\widehat{Y}_\ell - \mathbb{E}[Y_\ell]\| = \mathcal{O}(N_\ell^{-\eta} M_\ell^{-\beta/2})$,

(M3) $\mathrm{Cost}(\widehat{Y}_\ell) = \mathcal{O}(N_\ell M_\ell^{\gamma})$.

Then there are L *and* $\{N_\ell\}_{\ell=0}^{L}$ *such that* (in the special case $\beta = 2\alpha$)

$$\mathscr{C}_\varepsilon(\widehat{Q}_{h,s}^{ML}) = \begin{cases} \mathcal{O}(\varepsilon^{-1/\eta}), & \text{if } \alpha > \gamma\eta, \\ \mathcal{O}(\varepsilon^{-1/\eta}\,|\log(\varepsilon)|^{(2\eta+1)/(2\eta)}), & \text{if } \alpha = \gamma\eta, \\ \mathcal{O}(\varepsilon^{-\gamma/\alpha}), & \text{if } \alpha < \gamma\eta. \end{cases}$$

Multilevel Quadrature

Example: Multilevel Monte Carlo for 2D lognormal diffusion and linear FEs

Multilevel Monte Carlo: $\eta = \frac{1}{2}$ (using $\|X\| := \sqrt{\mathbb{E}[X^2]}$, i.e. RMSE)

- Using truncated KLE: $\alpha = \min(1, \nu),\ \gamma = \min(2, 1 + \frac{1}{\nu})$

$$\Rightarrow \quad \mathscr{C}_\varepsilon(\widehat{Q}_{h,s}^{\mathrm{MLMC}}) = \mathscr{O}\left(\varepsilon^{-\max(2,2/\nu)}\right)$$

Multilevel Quadrature

Example: Multilevel Monte Carlo for 2D lognormal diffusion and linear FEs

Multilevel Monte Carlo: $\eta = \frac{1}{2}$ (using $\|X\| := \sqrt{\mathbb{E}[X^2]}$, i.e. RMSE)

- Using truncated KLE: $\alpha = \min(1, \nu),\ \gamma = \min(2, 1 + \frac{1}{\nu})$

$$\Rightarrow \quad \mathscr{C}_\varepsilon(\widehat{Q}_{h,s}^{\mathrm{MLMC}}) = \mathscr{O}\left(\varepsilon^{-\max(2,2/\nu)}\right)$$

Independent of s !

(Can be improved using FFT-based sampling to $\mathscr{O}\left(\varepsilon^{-2}\right)$ for all ν.)

Multilevel Quadrature

Example: Multilevel Monte Carlo for 2D lognormal diffusion and linear FEs

Multilevel Monte Carlo: $\eta = \frac{1}{2}$ (using $\|X\| := \sqrt{\mathbb{E}[X^2]}$, i.e. RMSE)

- Using truncated KLE: $\alpha = \min(1, \nu),\ \gamma = \min(2, 1 + \frac{1}{\nu})$

$$\Rightarrow \quad \mathscr{C}_\varepsilon(\widehat{Q}_{h,s}^{\mathrm{MLMC}}) = \mathscr{O}\left(\varepsilon^{-\max(2,2/\nu)}\right)$$

Independent of s !

(Can be improved using FFT-based sampling to $\mathscr{O}\left(\varepsilon^{-2}\right)$ for all ν.)

Very interesting Numerical Analysis questions to prove (M1) and (M2)
[Charrier, RS, Teckentrup, SINUM '13], [Teckentrup, RS, Giles, Ullmann, NM '13]

as well as to prove (M3)

Multilevel Quadrature

Example: Multilevel Monte Carlo for 2D lognormal diffusion and linear FEs

Multilevel Monte Carlo: $\eta = \frac{1}{2}$ (using $\|X\| := \sqrt{\mathbb{E}[X^2]}$, i.e. RMSE)

- Using truncated KLE: $\alpha = \min(1, \nu),\ \gamma = \min(2, 1 + \frac{1}{\nu})$

$$\Rightarrow \quad \mathscr{C}_\varepsilon(\widehat{Q}^{\mathrm{MLMC}}_{h,s}) = \mathscr{O}\left(\varepsilon^{-\max(2,2/\nu)}\right)$$

Independent of s !

(Can be improved using FFT-based sampling to $\mathscr{O}\left(\varepsilon^{-2}\right)$ for all ν.)

Very interesting Numerical Analysis questions to prove (M1) and (M2)
[Charrier, RS, Teckentrup, SINUM '13], [Teckentrup, RS, Giles, Ullmann, NM '13]

as well as to prove (M3)

Can this be improved?

Multilevel Quasi-Monte Carlo [Kuo et al, 2015], [Kuo et al, 2017]

$$\mathbb{E}[Y_\ell] = \int_{[0,1]^s} Y_\ell\Big(\Phi^{-1}(\boldsymbol{\xi})\Big)\,\mathrm{d}\boldsymbol{\xi} \;\approx\; \frac{1}{N_\ell}\sum_{i=1}^{N_\ell} Y_\ell\Big(\Phi^{-1}\big(\boldsymbol{\xi}^{(i)}\big)\Big) =: \widehat{Y}_\ell^{\mathrm{QMC}}$$

with $\boldsymbol{\Phi} : \mathbb{R}^s \to [0,1]^s$ the cumulative normal distribution function.

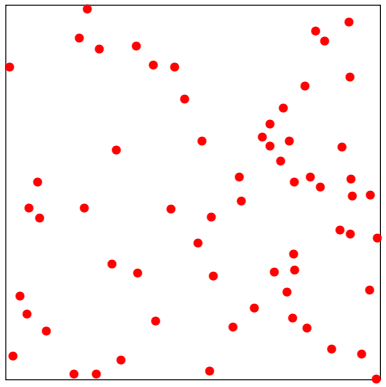
Multilevel Quasi-Monte Carlo [Kuo et al, 2015], [Kuo et al, 2017]

$$\mathbb{E}[Y_\ell] = \int_{[0,1]^s} Y_\ell\Big(\Phi^{-1}(\boldsymbol{\xi})\Big)\,\mathrm{d}\boldsymbol{\xi} \;\approx\; \frac{1}{N_\ell}\sum_{i=1}^{N_\ell} Y_\ell\Big(\Phi^{-1}\big(\boldsymbol{\xi}^{(i)}\big)\Big) =: \widehat{Y}_\ell^{\mathrm{QMC}}$$

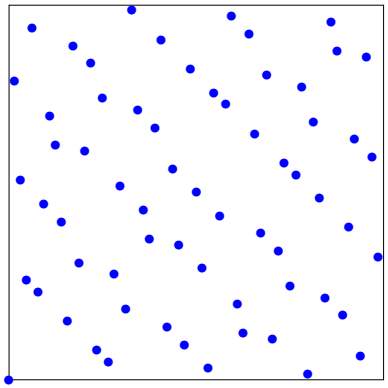
with $\boldsymbol{\Phi} : \mathbb{R}^s \to [0,1]^s$ the cumulative normal distribution function.

Monte Carlo: $\boldsymbol{\xi}^{(i)}$ unif. random
$\mathcal{O}(N^{-1/2})$ convergence
(order of variables irrelevant)

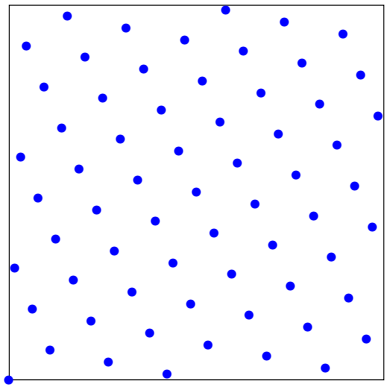
QMC: $\boldsymbol{\xi}^{(i)}$ deterministic
close to $\mathcal{O}(N^{-1})$ convergence
(order of variables **important** !)



64 random points



64 Sobol′ points



64 lattice points

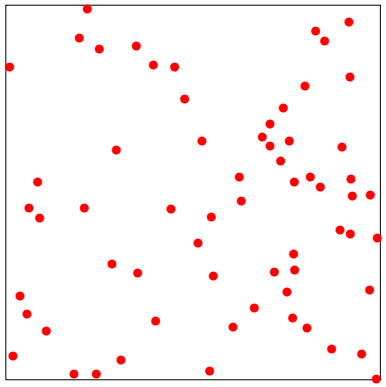
Multilevel Quasi-Monte Carlo [Kuo et al, 2015], [Kuo et al, 2017]

$$\mathbb{E}[Y_\ell] = \int_{[0,1]^s} Y_\ell\Big(\Phi^{-1}(\boldsymbol{\xi})\Big)\,\mathrm{d}\boldsymbol{\xi} \;\approx\; \frac{1}{N_\ell}\sum_{i=1}^{N_\ell} Y_\ell\Big(\Phi^{-1}\big(\boldsymbol{\xi}^{(i)}\big)\Big) =: \widehat{Y}_\ell^{\mathrm{QMC}}$$

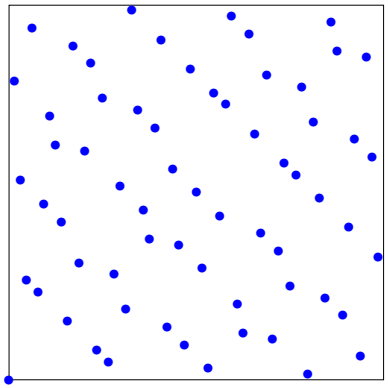
with $\boldsymbol{\Phi} : \mathbb{R}^s \to [0,1]^s$ the cumulative normal distribution function.

Monte Carlo: $\boldsymbol{\xi}^{(i)}$ unif. random
$\mathscr{O}(N^{-1/2})$ convergence
(order of variables irrelevant)

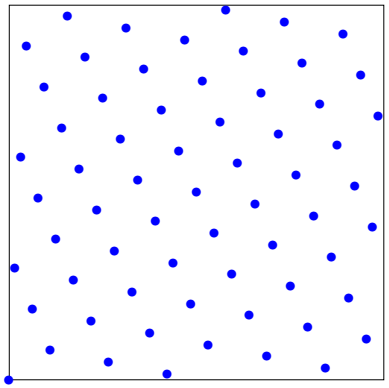
QMC: $\boldsymbol{\xi}^{(i)}$ deterministic
close to $\mathscr{O}(N^{-1})$ convergence
(order of variables **important** !)



64 random points



64 Sobol′ points

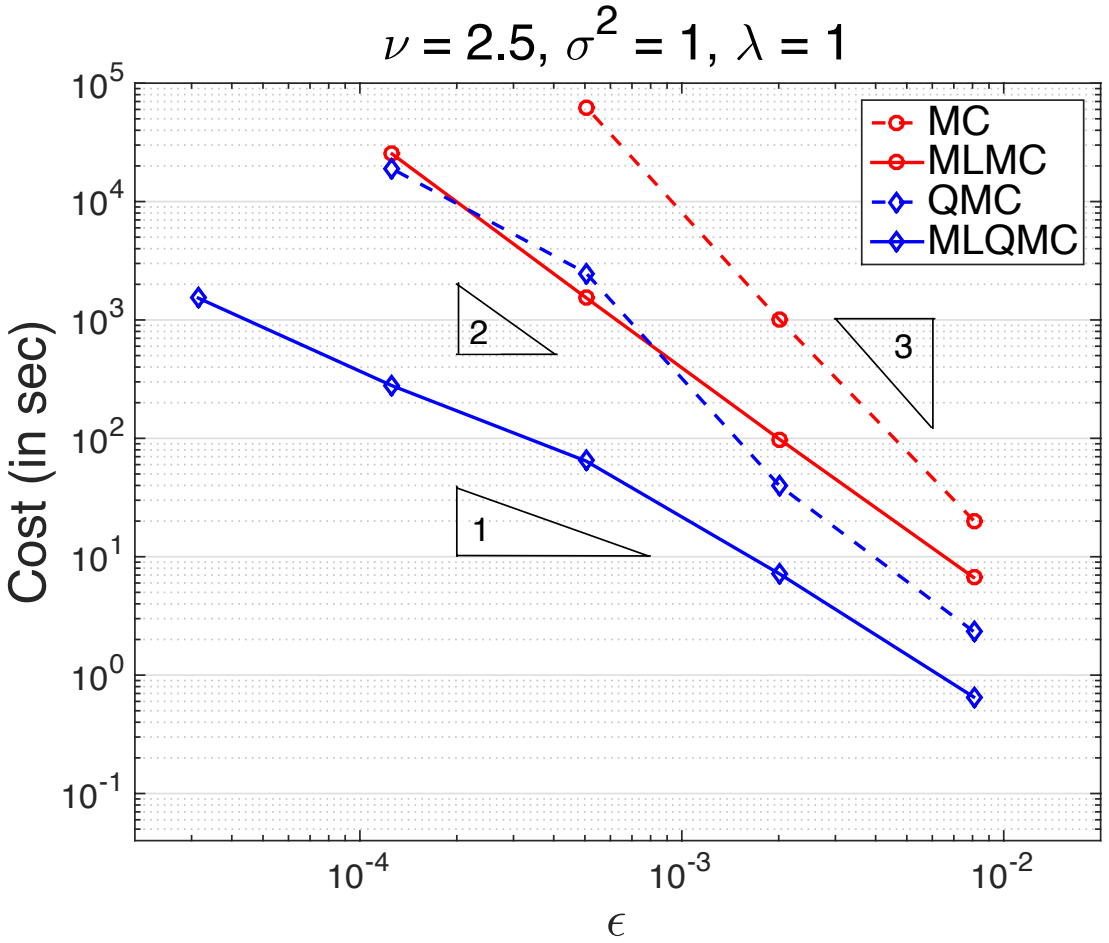


64 lattice points

Theorem. For $\nu > \frac{3d}{2} + 1$ and any $\delta > 0$: $\mathscr{C}_\varepsilon(\widehat{Q}_{h,s}^{\mathrm{MLMC}}) = \mathscr{O}\left(\varepsilon^{-1+\delta}\right)$

Multilevel Quasi-Monte Carlo

Numerical Comparison for 2D lognormal diffusion



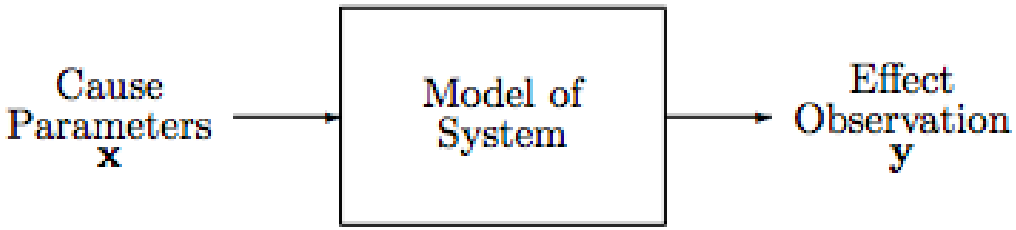
$D = (0,1)^2$; linear FEs; $Q = \frac{1}{|D^*|}\int_{D^*} p\,\mathrm{d}x$; truncated KLE w. $s \sim h^{-2/\nu}$; using a randomised lattice rule with product weights $\gamma_j = 1/j^2$.

Uncertainty Quantification

The Inverse Problem

Uncertainty Quantification – The Inverse Problem

Bayesian interpretation of an inverse problem



The (physical) model gives us $\pi(y|x)$, the conditional probability of observing y given x (**"likelihood"**), e.g. assuming additive Gaussian noise:

$$y = H(x) + \eta$$

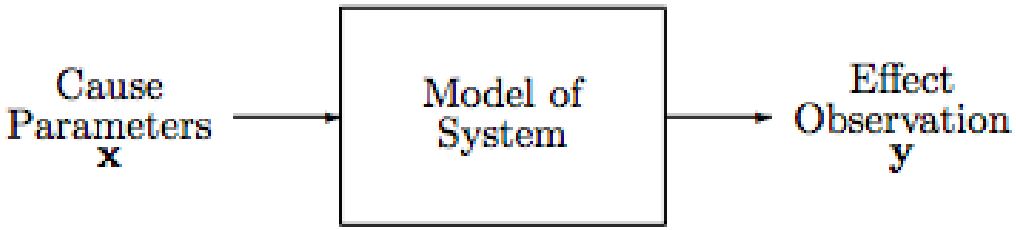
where $H : X \to \mathbb{R}^m$ is the *forward operator* and $\eta \sim \mathrm{N}(\mathbf{0}, \Sigma)$ is the noise.

- So the *maximum likelihood estimate* in the Bayesian interpretation is the solution of the least squares problem

$$\mathrm{argmin}_{x \in X} \|y - H(x)\|^2_{\Sigma^{-1}}$$

Uncertainty Quantification – The Inverse Problem

Bayesian interpretation of an inverse problem



The (physical) model gives us $\pi(y|x)$, the conditional probability of observing y given x (**"likelihood"**), e.g. assuming additive Gaussian noise:

$$y = H(x) + \eta$$

where $H : X \to \mathbb{R}^m$ is the *forward operator* and $\eta \sim \mathrm{N}(\mathbf{0}, \Sigma)$ is the noise.

- So the *maximum likelihood estimate* in the Bayesian interpretation is the solution of the least squares problem

$$\mathrm{argmin}_{x \in X} \|y - H(x)\|^2_{\Sigma^{-1}}$$

But often we are really interested in $\pi(x|y)$, i.e. the conditional probability of possible causes x given the observed data y (**"posterior"** density).

A simple result about conditional probabilities states

$$\pi(x|y) = \frac{\pi(y|x)\pi(x)}{\pi(y)} \qquad \textbf{(Bayes' rule)}$$

where $\pi(x) =$ **prior density** – our knowledge/belief about x

($\pi(y) =$ **marginal** of $\pi(x, y)$ over all possible x, an unimportant scaling factor.)

Uncertainty Quantification – The Inverse Problem

Bayes' rule and MAP estimate

A simple result about conditional probabilities states

$$\pi(x|y) = \frac{\pi(y|x)\pi(x)}{\pi(y)} \qquad \textbf{(Bayes' rule)}$$

where $\pi(x) =$ **prior density** – our knowledge/belief about x

($\pi(y) =$ **marginal** of $\pi(x, y)$ over all possible x, an unimportant scaling factor.)

- The *maximum a posteriori (MAP) estimate* with Gaussian prior $x \sim \mathrm{N}(x_0, R)$ is the solution of the *regularised* least squares problem

$$\mathrm{argmin}_{x \in X} \ \|y - H(x)\|^2_{\Sigma^{-1}} + \|x - x_0\|^2_{R^{-1}}$$

(in addition, we can also get the posterior covariance at the MAP point)

Uncertainty Quantification – The Inverse Problem

Bayes' rule and MAP estimate

A simple result about conditional probabilities states

$$\pi(x|y) = \frac{\pi(y|x)\pi(x)}{\pi(y)} \qquad \textbf{(Bayes' rule)}$$

where $\pi(x) =$ **prior density** – our knowledge/belief about x

($\pi(y) =$ **marginal** of $\pi(x, y)$ over all possible x, an unimportant scaling factor.)

- The *maximum a posteriori (MAP) estimate* with Gaussian prior $x \sim \mathrm{N}(x_0, R)$ is the solution of the *regularised* least squares problem

$$\mathrm{argmin}_{x \in X} \ \|y - H(x)\|^2_{\Sigma^{-1}} + \|x - x_0\|^2_{R^{-1}}$$

(in addition, we can also get the posterior covariance at the MAP point)

Can we do better than finding the MAP estimate and its covariance?

Metropolis-Hastings Markov Chain Monte Carlo

ALGORITHM 1 (Metropolis-Hastings Markov Chain Monte Carlo)

- Choose initial state $x^0 \in X$ (typically "burn in").
- At state x^n generate proposal $x' \in X$ from distribution $q(x'|x^n)$
  (e.g. via a random walk $x' \sim \mathrm{N}(x^n, B)$)
- Accept x' as a sample with probability

$$\alpha(x'|x^n) \;=\; \min\left(1, \frac{\pi(x'|y)\, q(x^n|x')}{\pi(x^n|y)\, q(x'|x^n)}\right)$$

  i.e. $x^{n+1} = x'$ with probability $\alpha(x'|x^n)$; otherwise $x^{n+1} = x^n$.

Metropolis-Hastings Markov Chain Monte Carlo

ALGORITHM 1 (Metropolis-Hastings Markov Chain Monte Carlo)

- Choose initial state $x^0 \in X$ (typically "burn in").
- At state x^n generate proposal $x' \in X$ from distribution $q(x'|x^n)$ (e.g. via a random walk $x' \sim \mathrm{N}(x^n, B)$)
- Accept x' as a sample with probability

$$\alpha(x'|x^n) \;=\; \min\left(1, \frac{\pi(x'|y)\, q(x^n|x')}{\pi(x^n|y)\, q(x'|x^n)}\right)$$

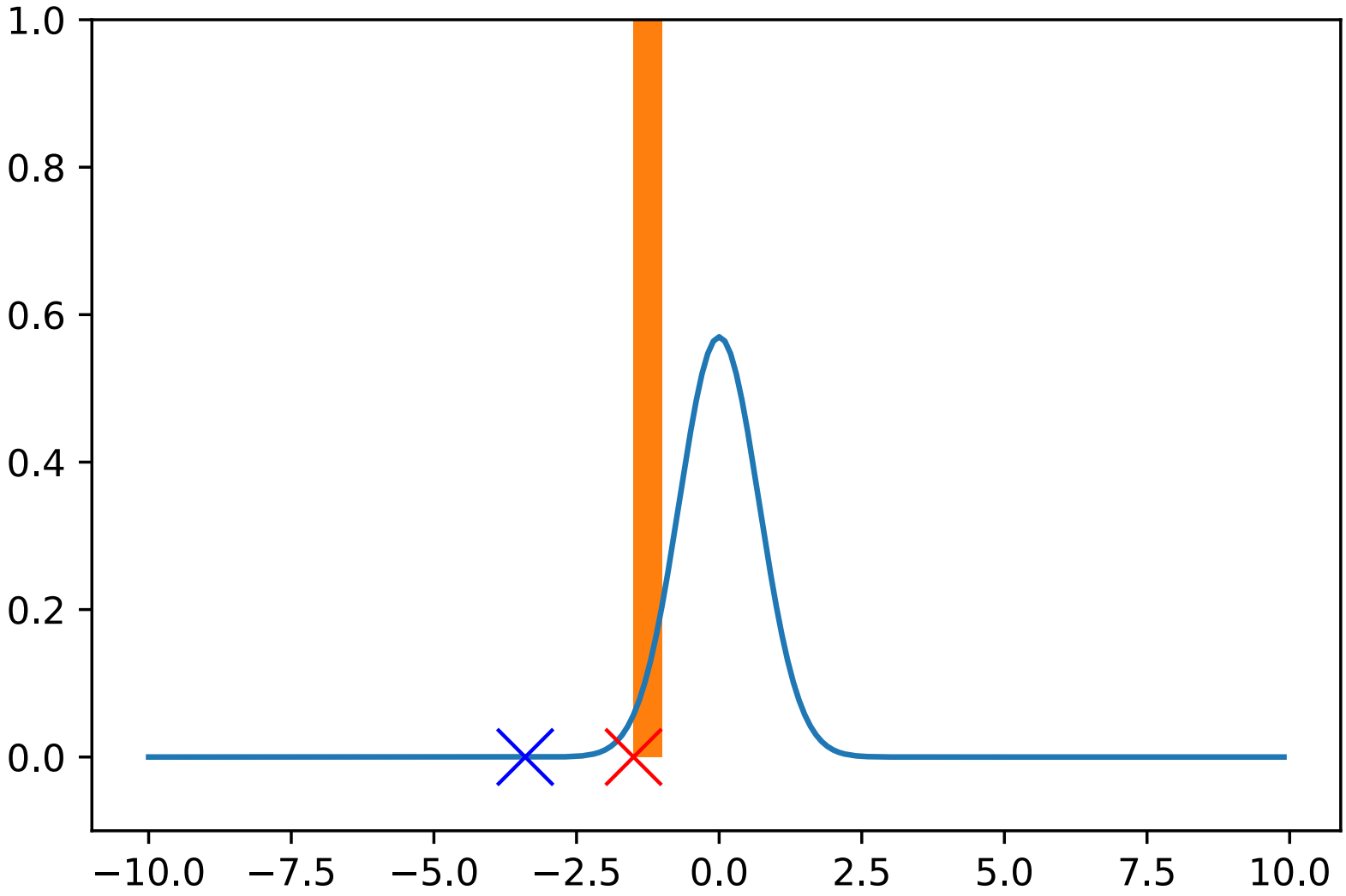
i.e. $x^{n+1} = x'$ with probability $\alpha(x'|x^n)$; otherwise $x^{n+1} = x^n$.

The samples $f(x^n)$ of some output function ("statistic") $f(\cdot)$ can be used for inference as usual (even though not i.i.d.):

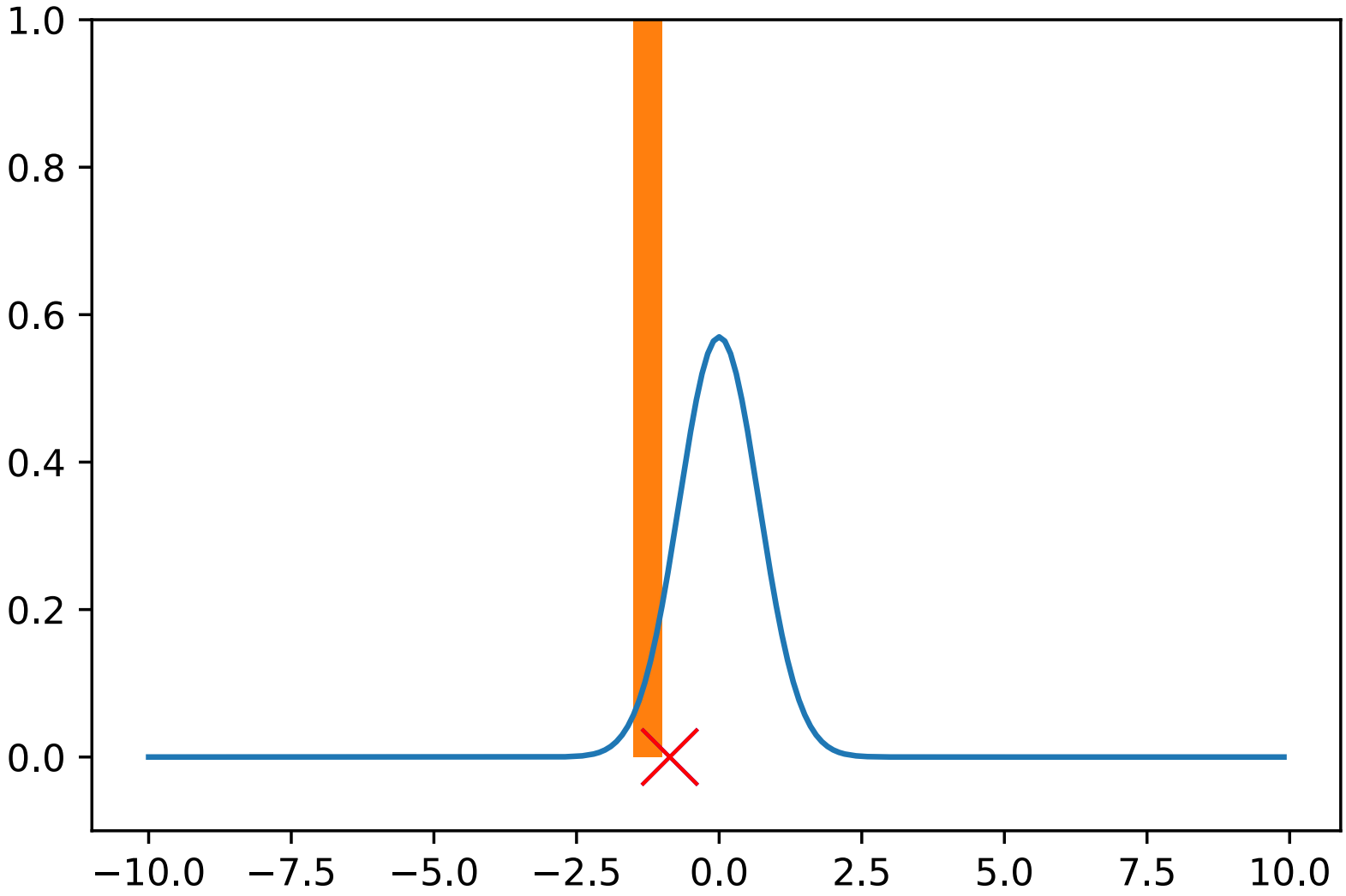
$$\widehat{f}^{\mathrm{MetH}} \;:=\; \frac{1}{N}\sum_{i=1}^{N} f(x^n) \;\approx\; \mathbb{E}_{\pi(x|y)}\left[f(x)\right]$$

BUT asymptotic variance (and thus N) scales w. integr. autocorrelation time **!**

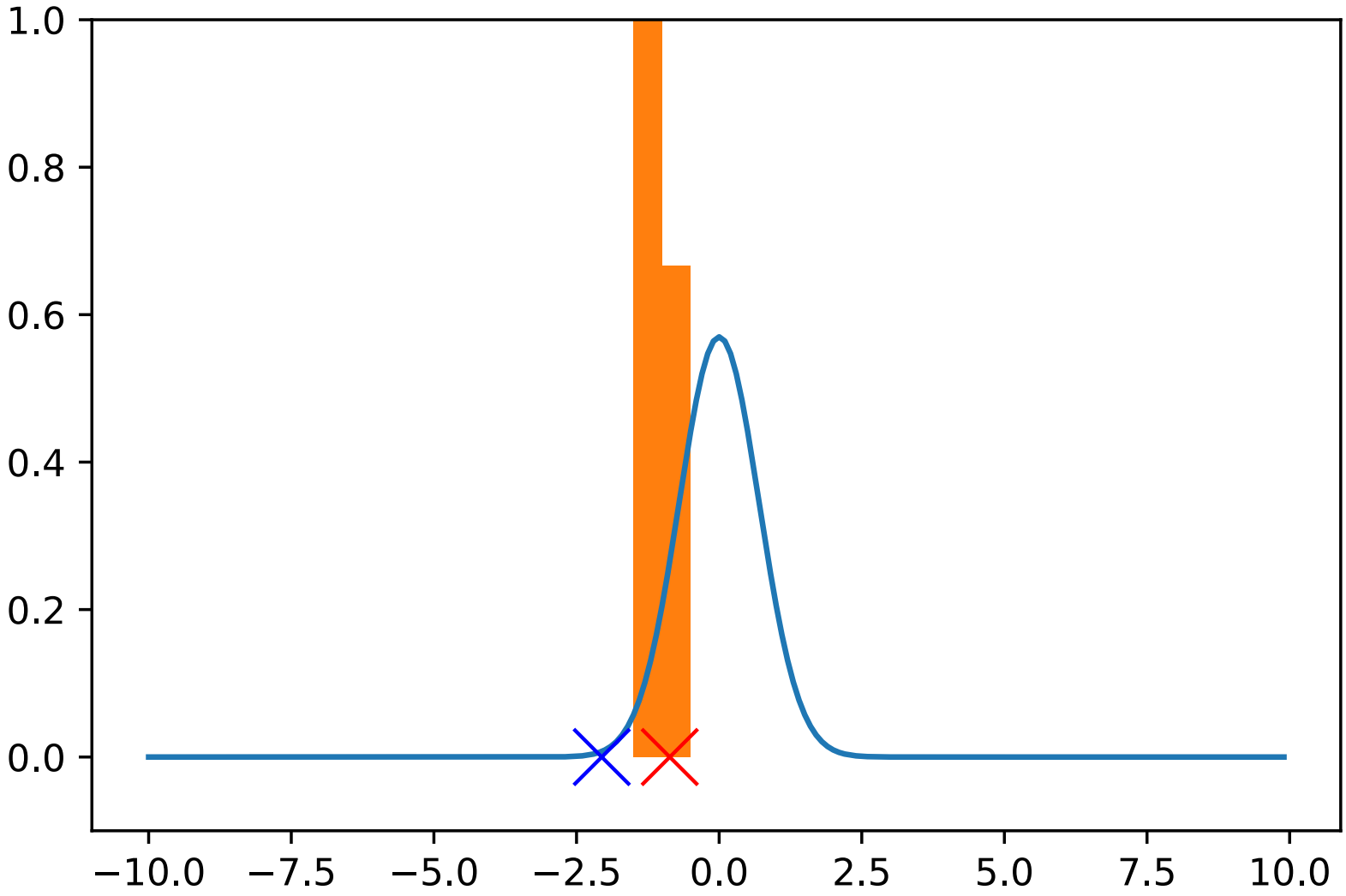
MCMC Example (Sample 1)



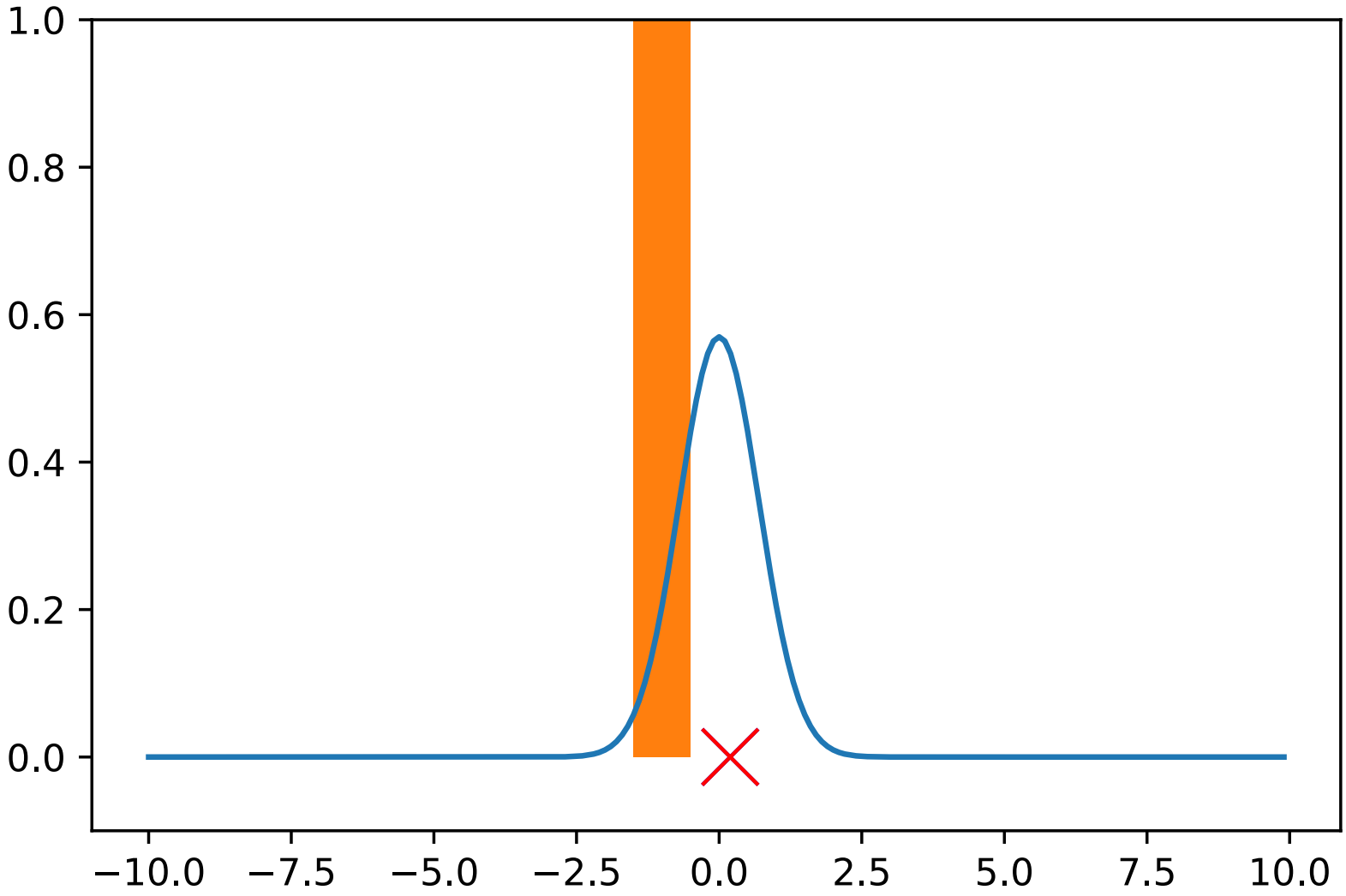
MCMC Example (Sample 2)



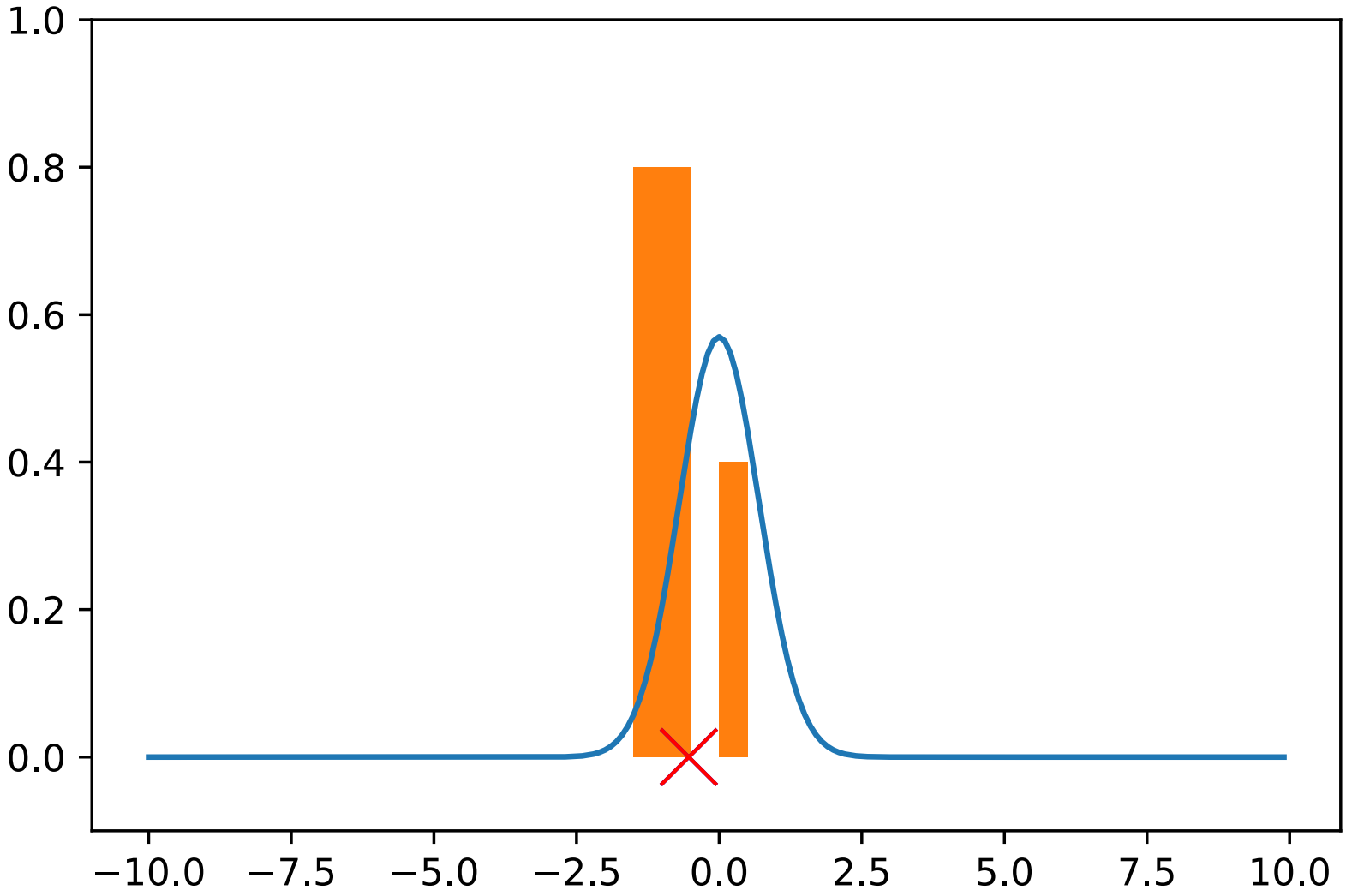
MCMC Example (Sample 3)



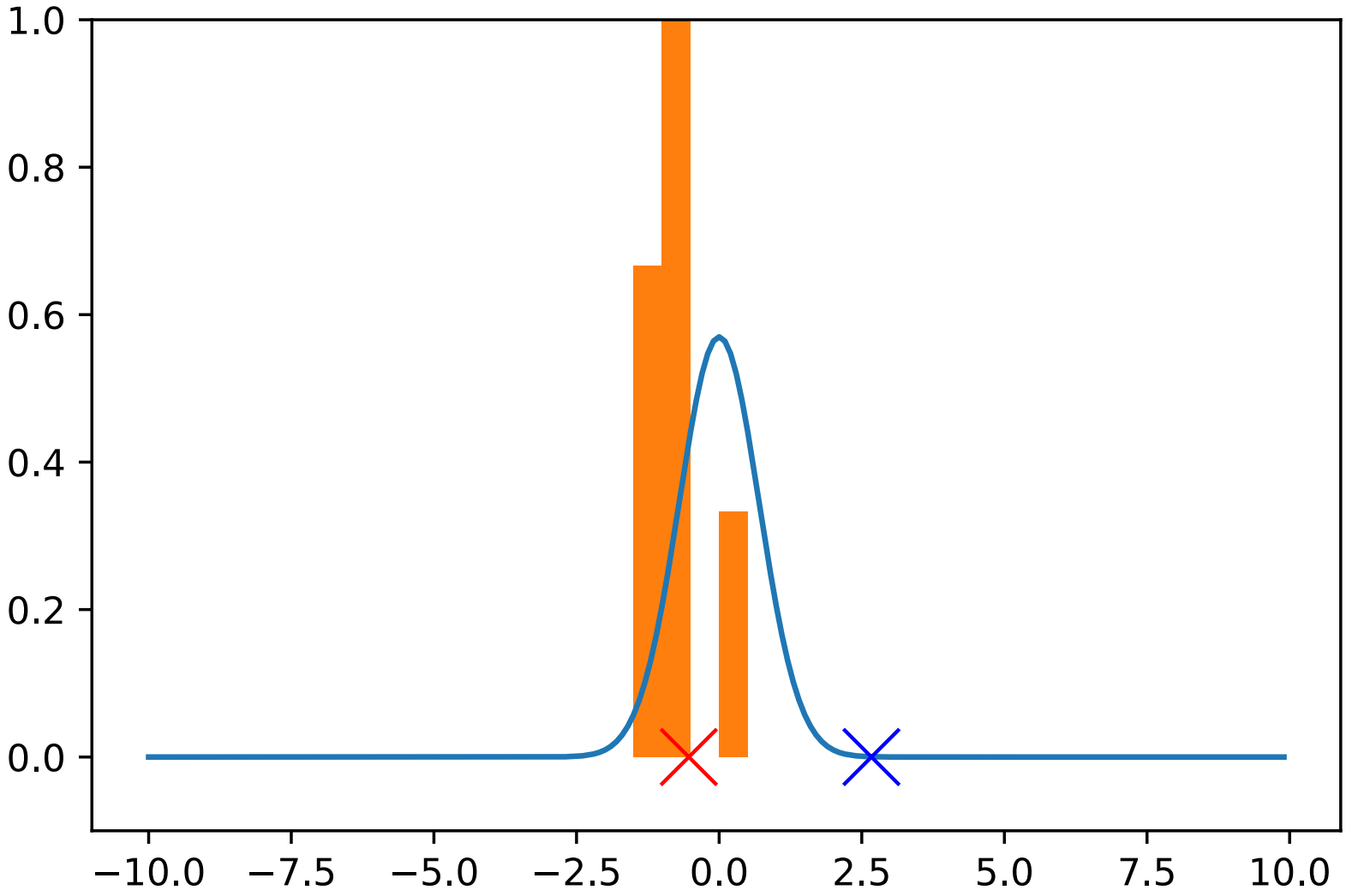
MCMC Example (Sample 4)



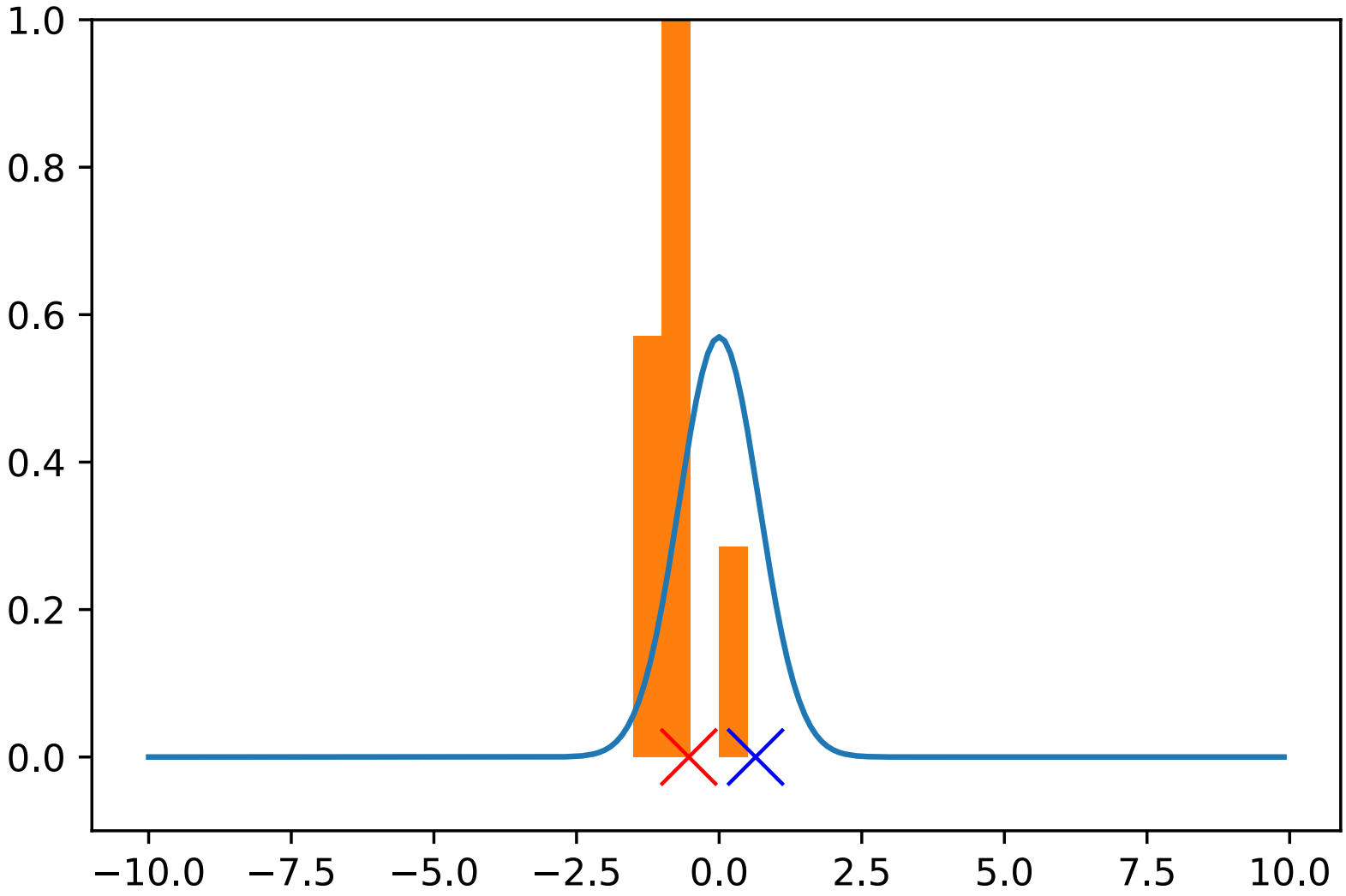
MCMC Example (Sample 5)



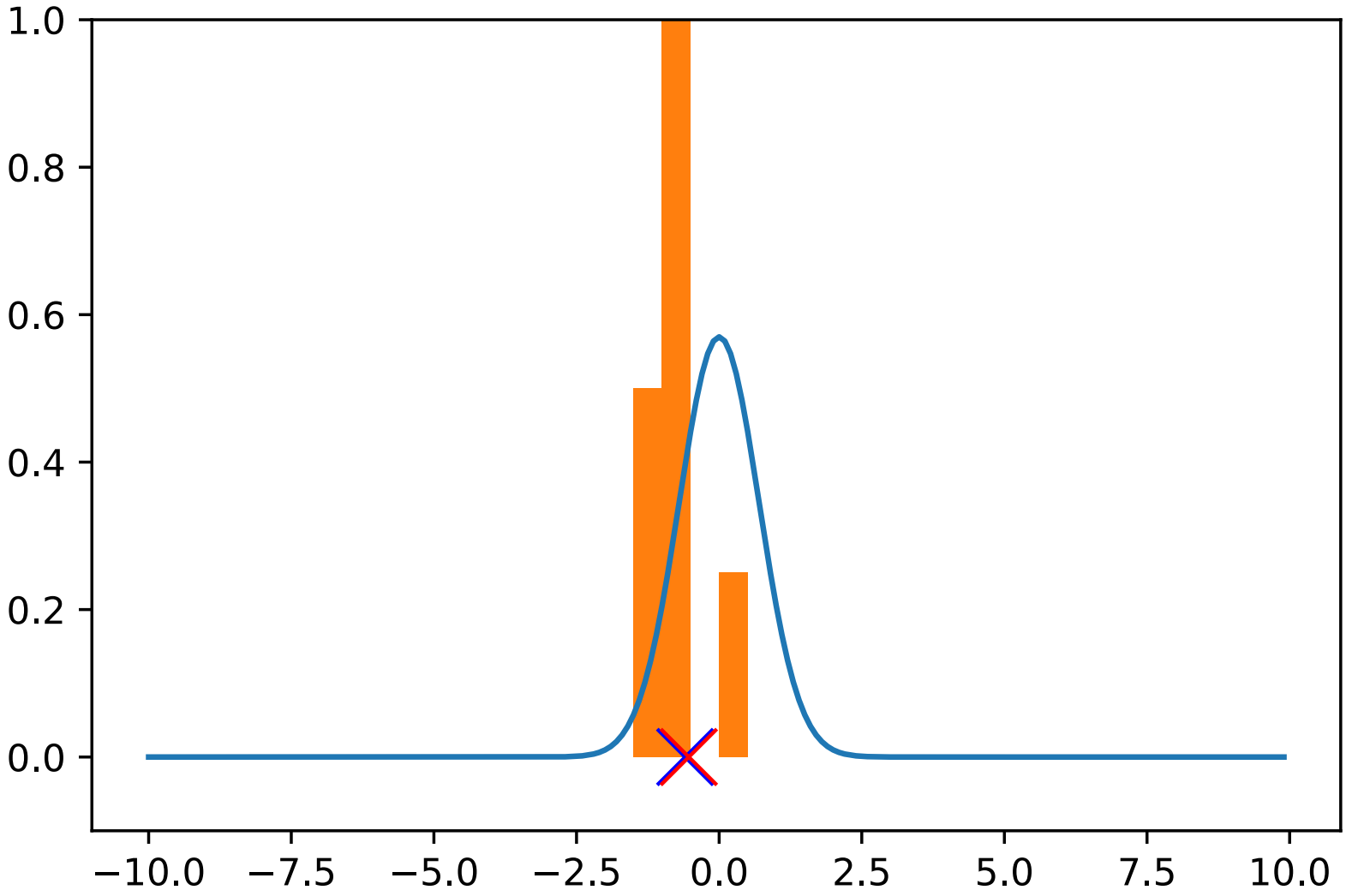
MCMC Example (Sample 6)



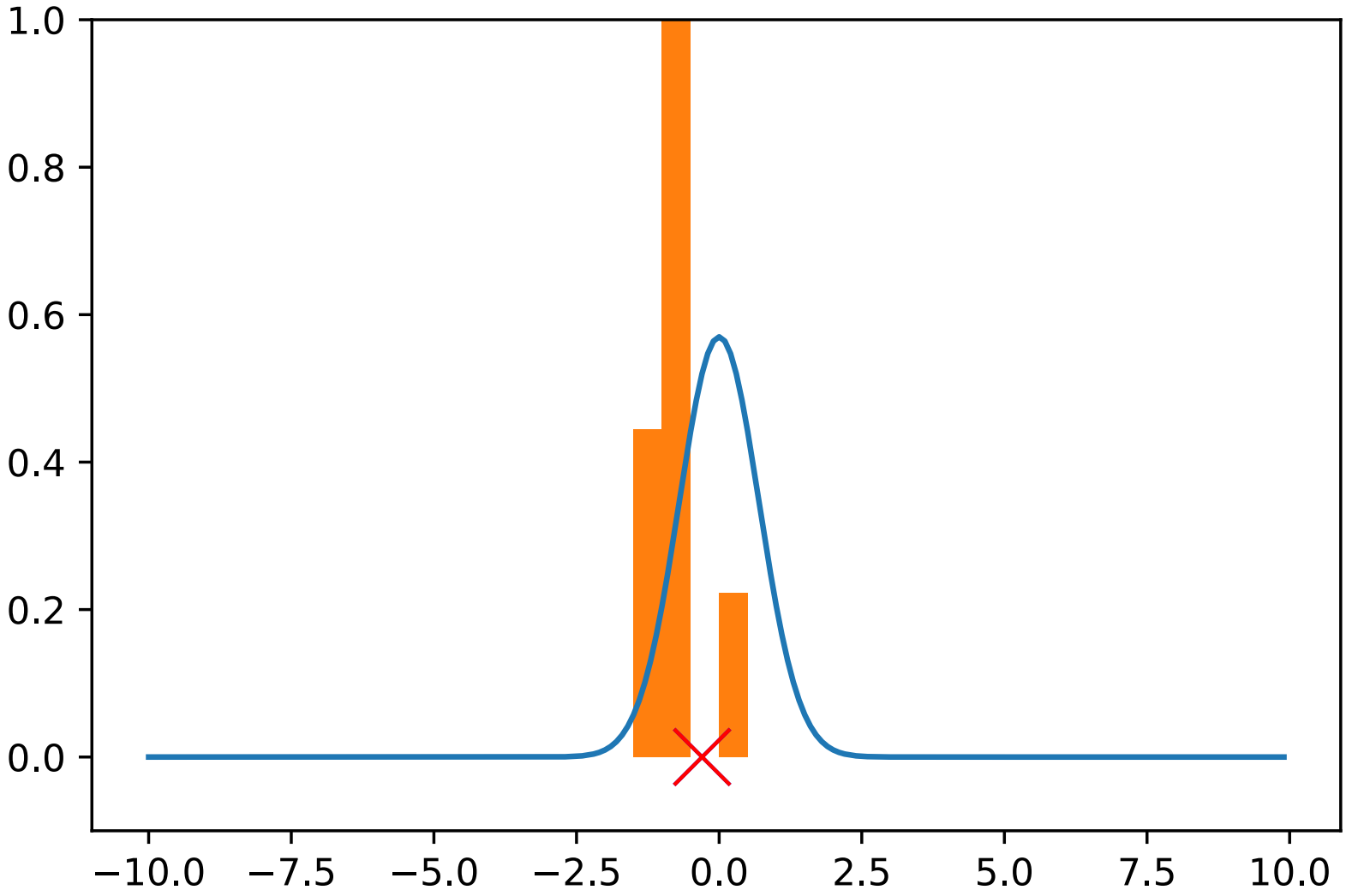
MCMC Example (Sample 7)



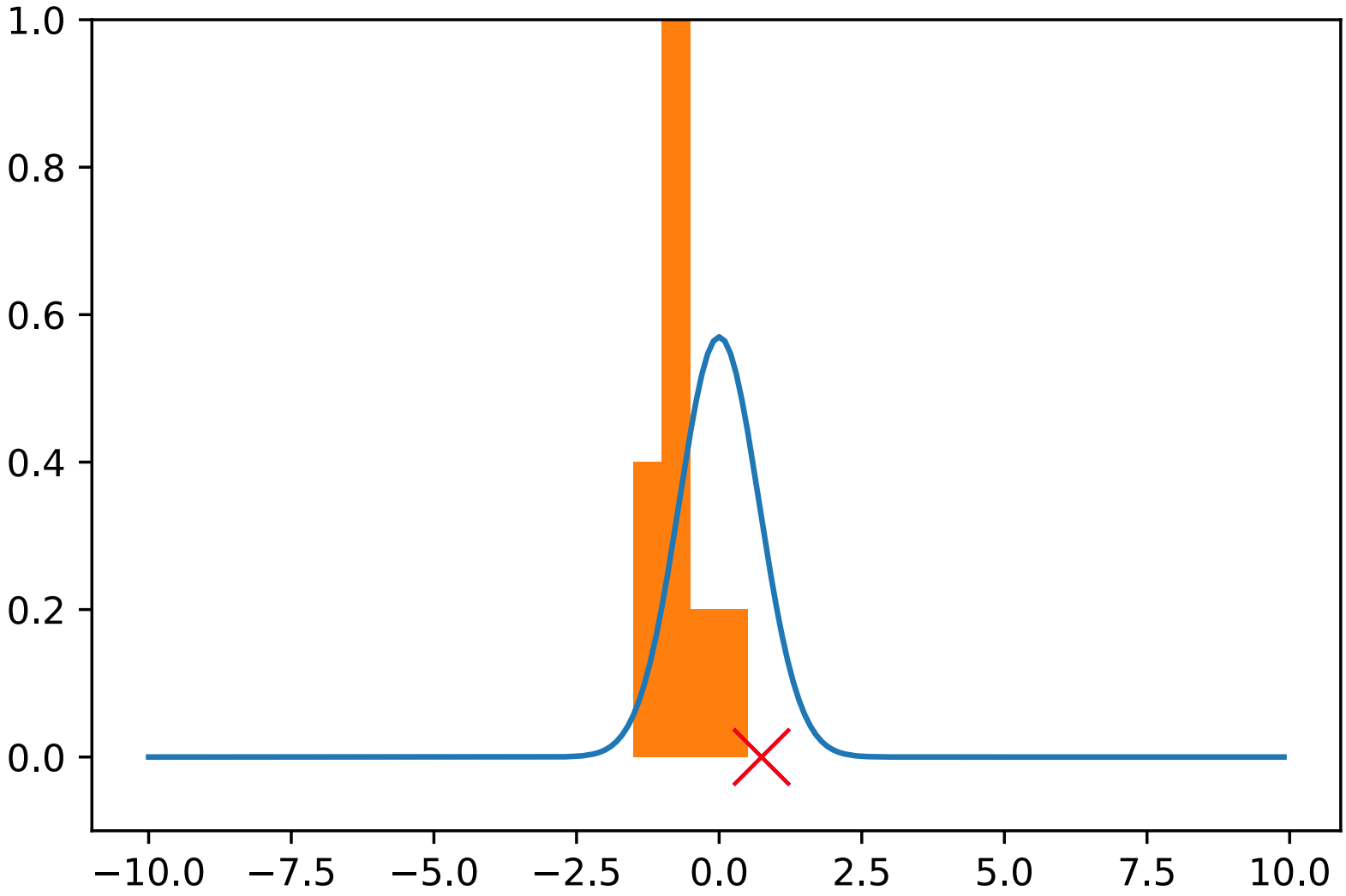
MCMC Example (Sample 8)



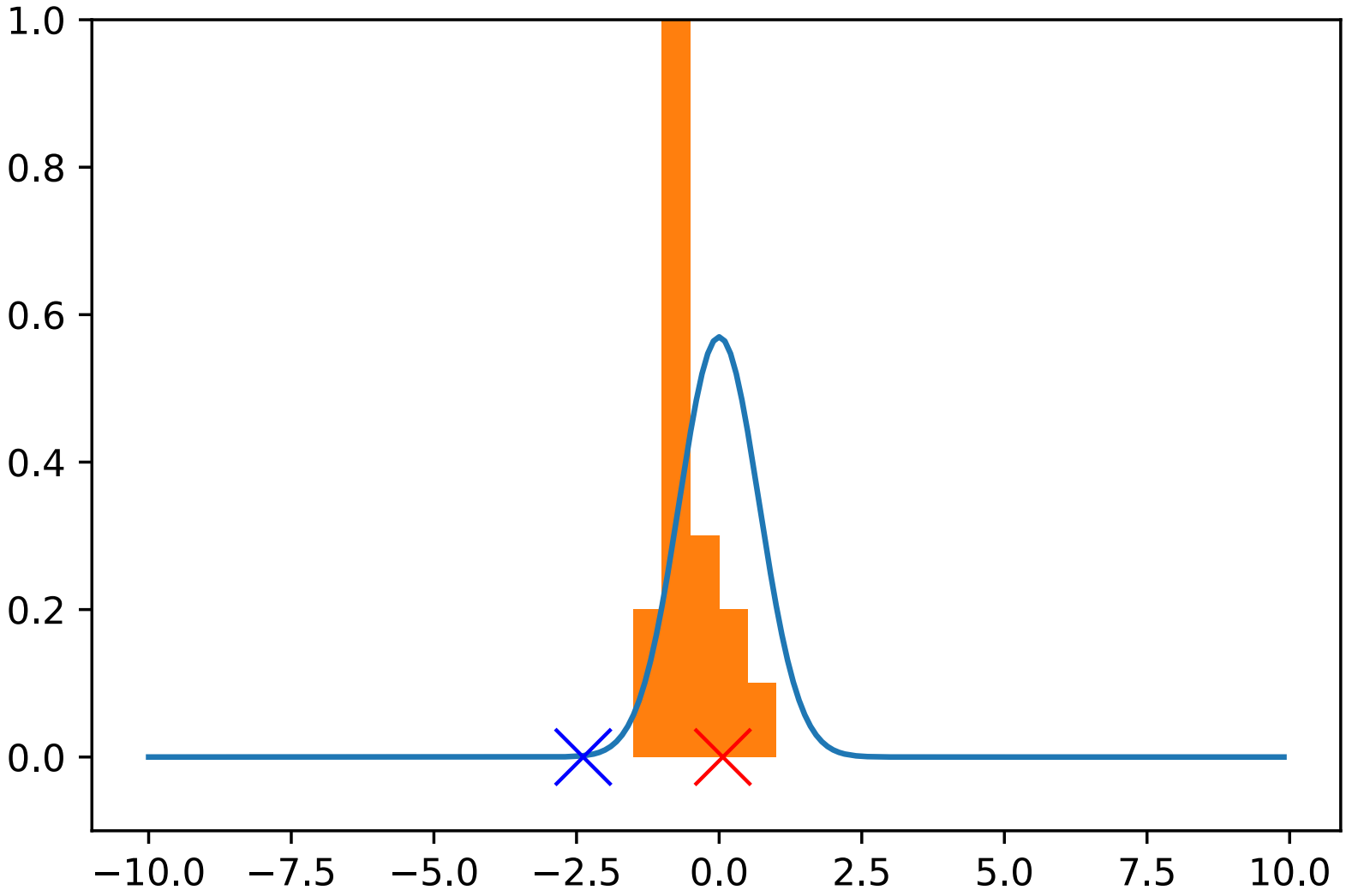
MCMC Example (Sample 9)



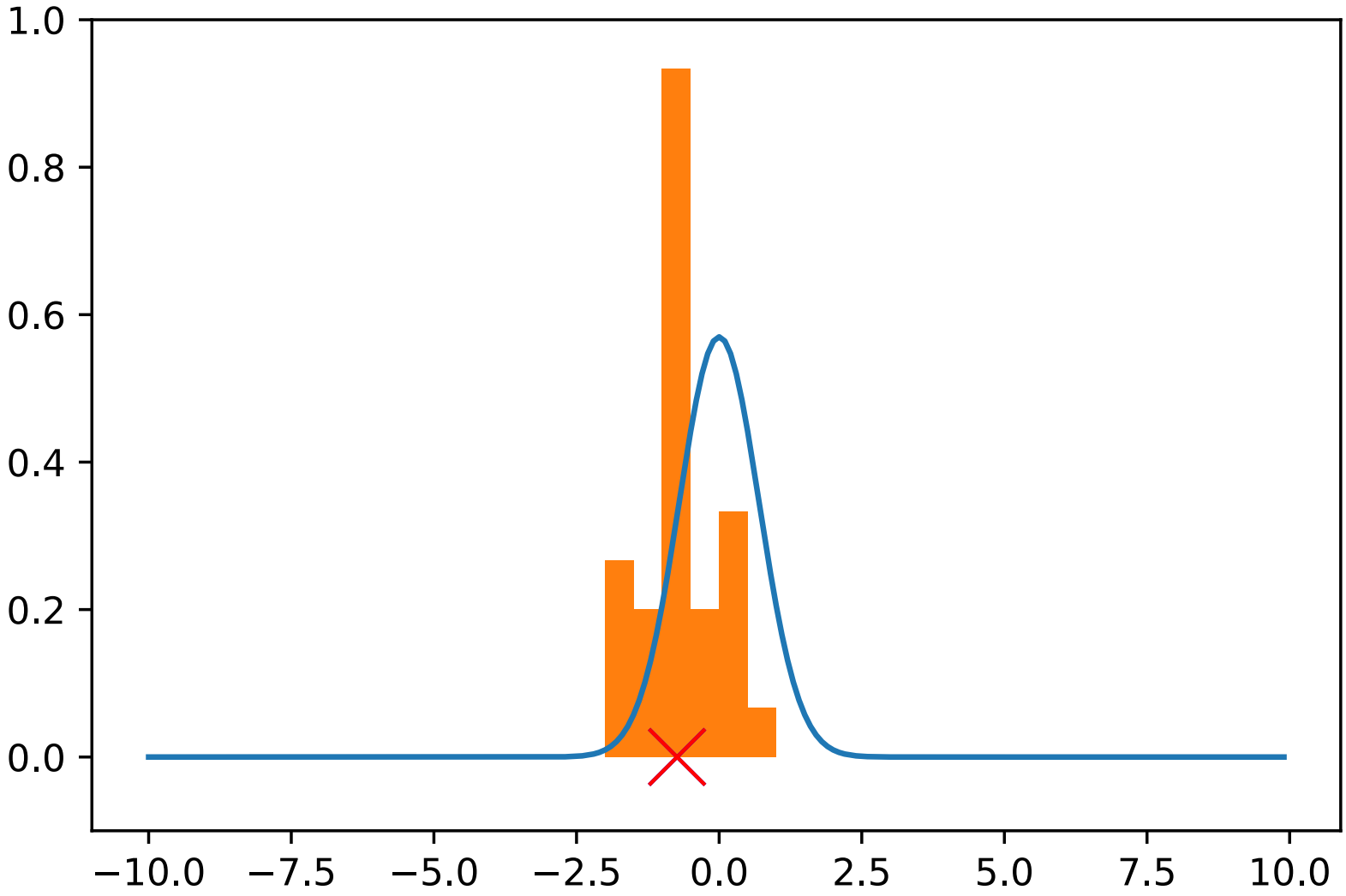
MCMC Example (Sample 20)



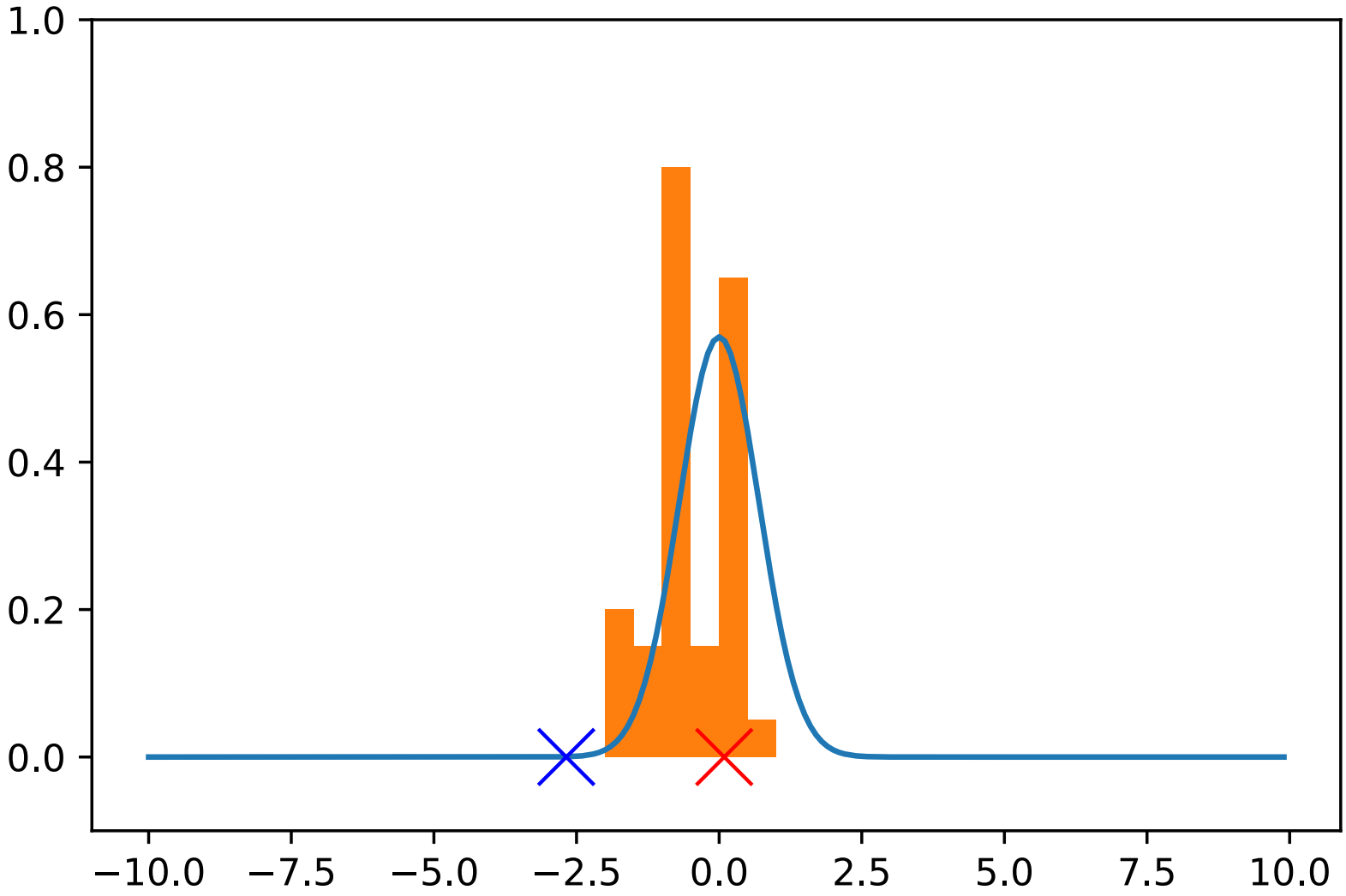
MCMC Example (Sample 30)



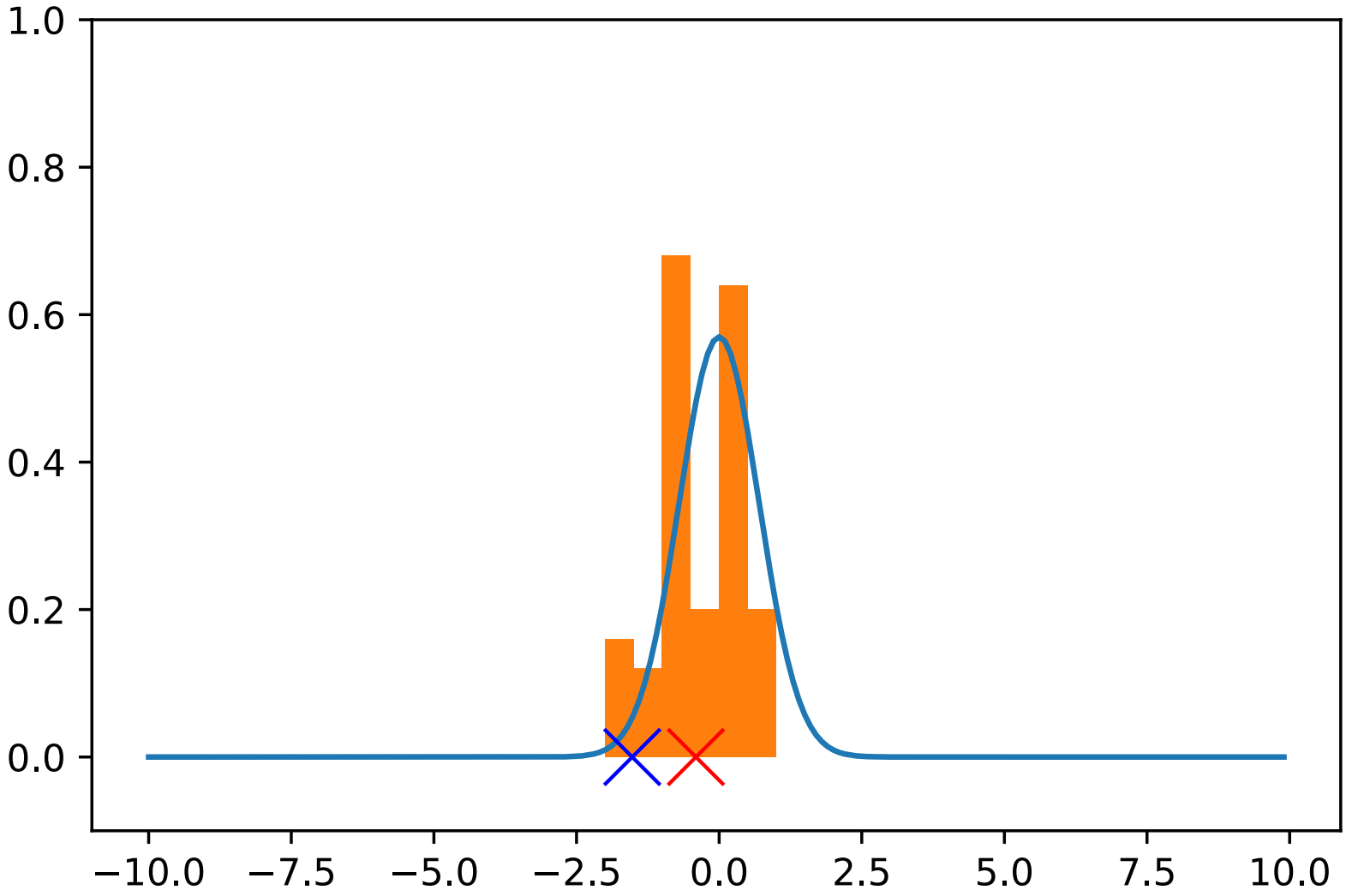
MCMC Example (Sample 40)



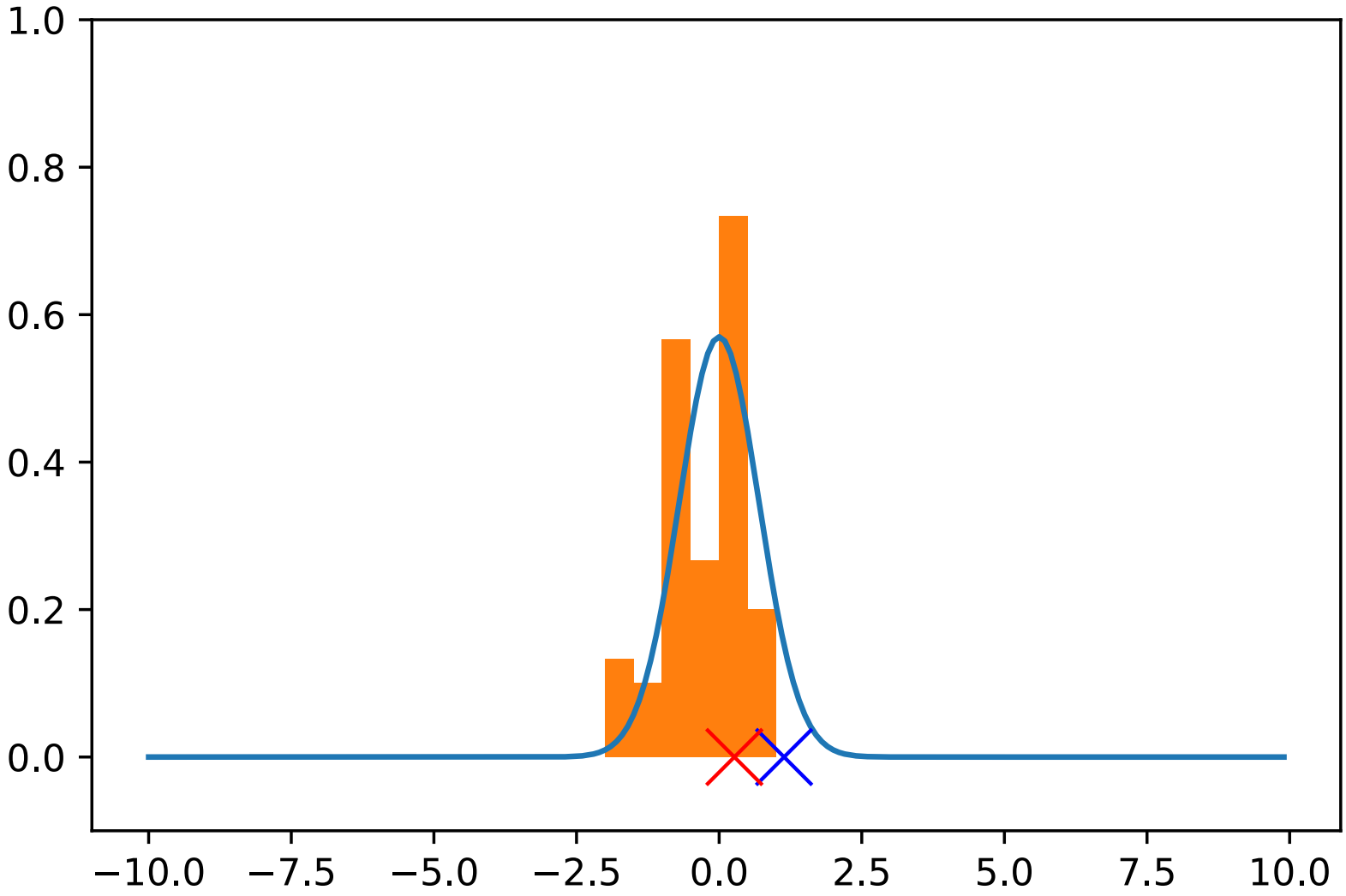
MCMC Example (Sample 50)



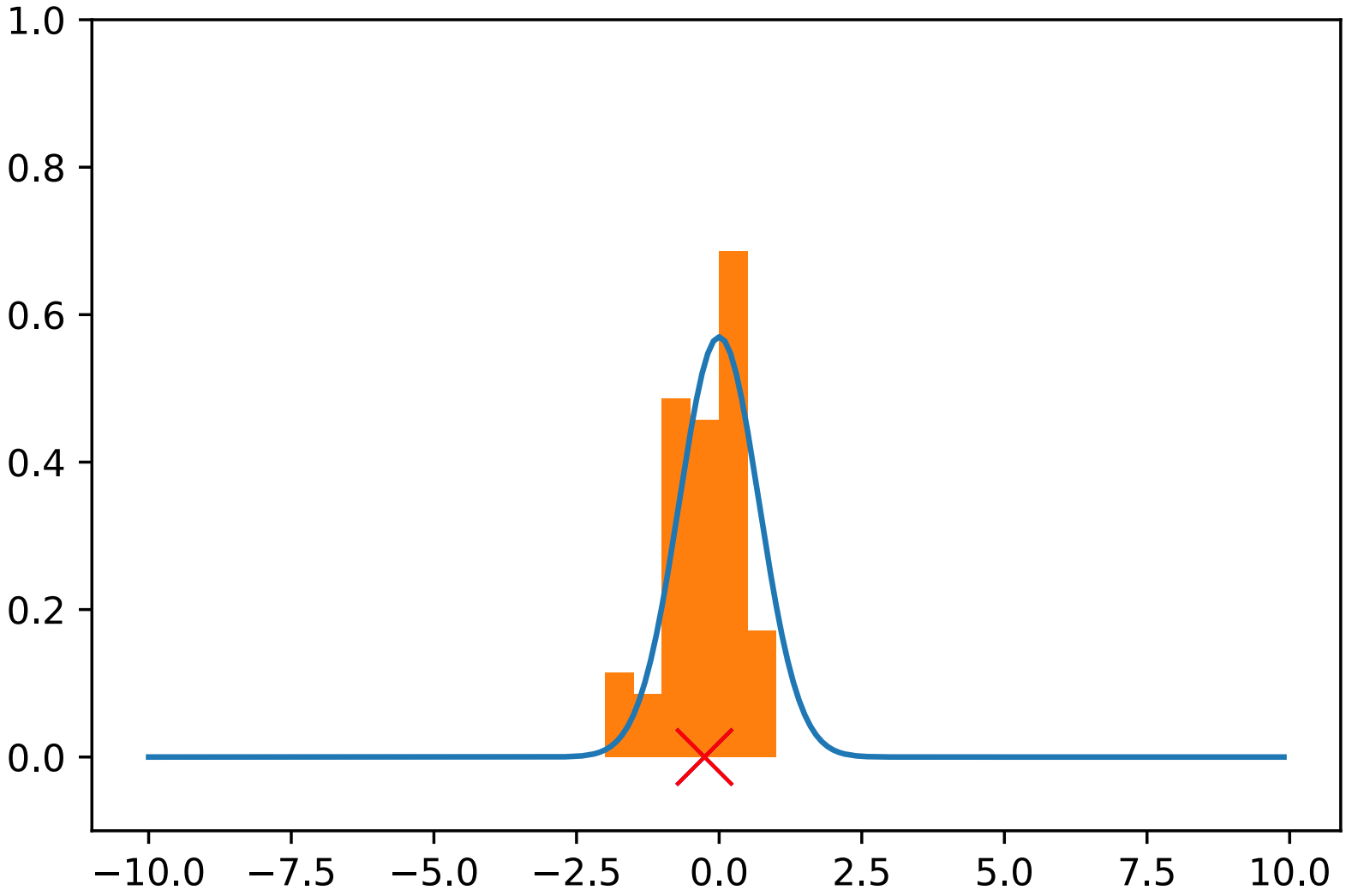
MCMC Example (Sample 60)



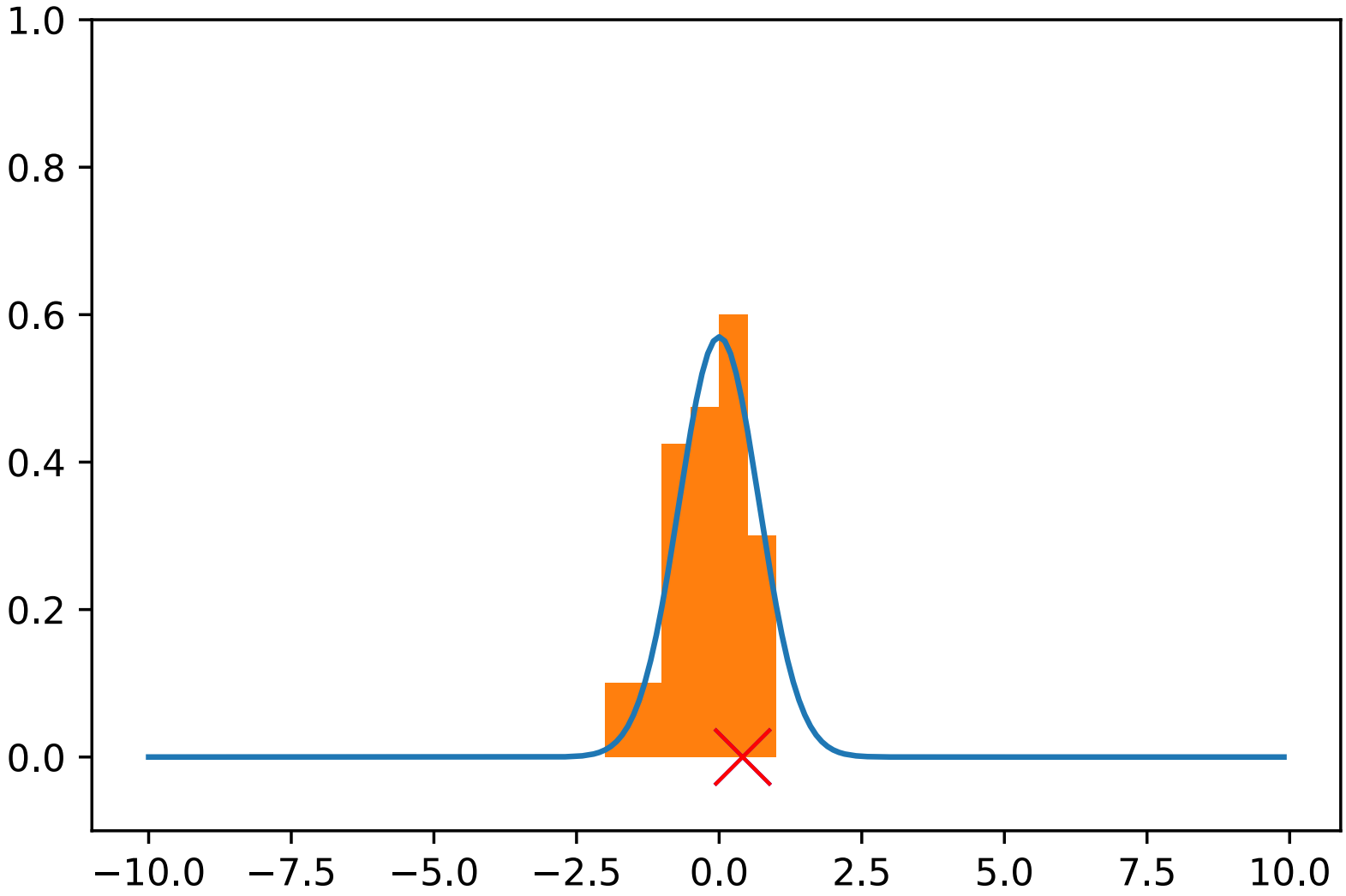
MCMC Example (Sample 70)



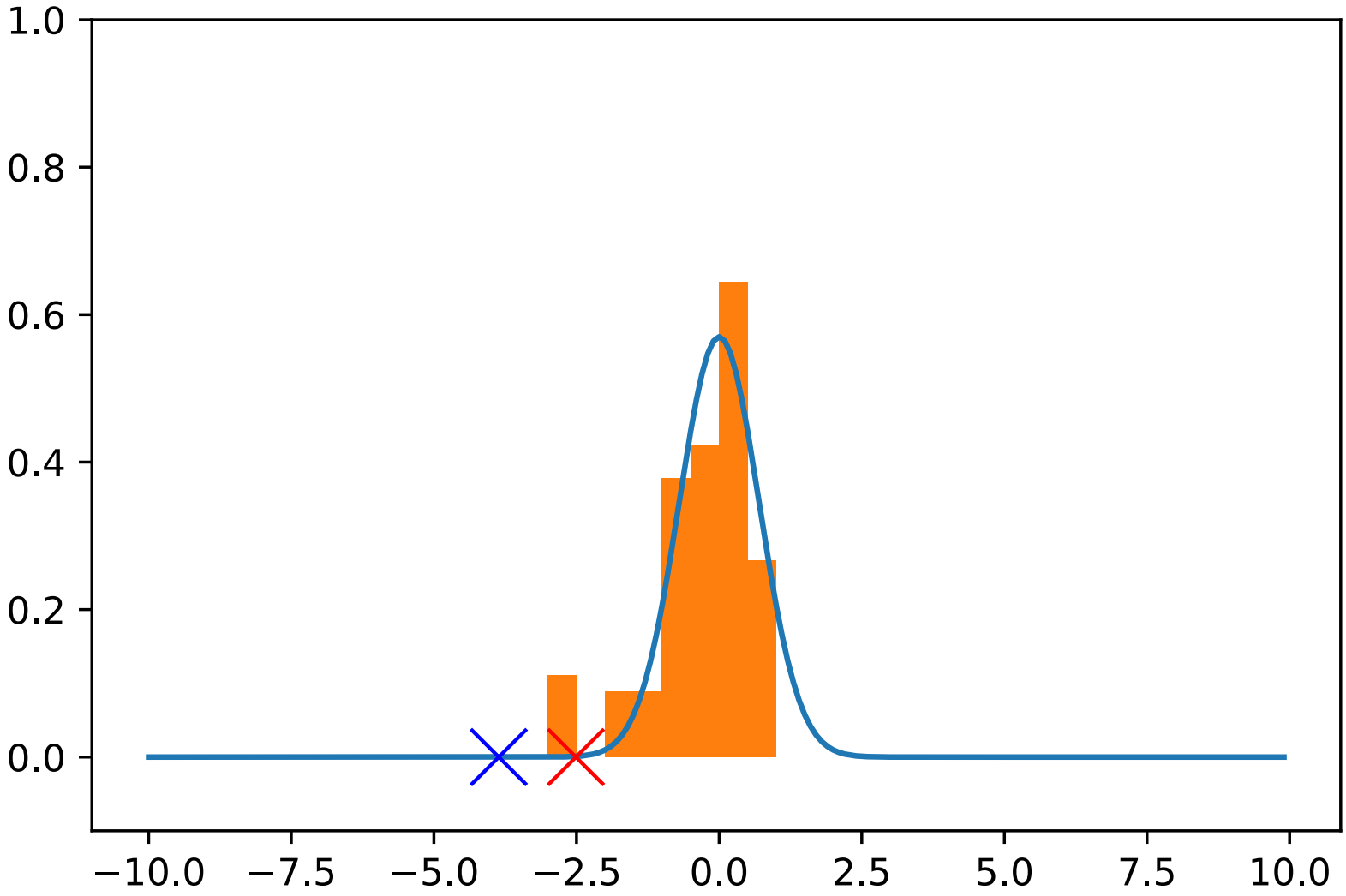
MCMC Example (Sample 80)



MCMC Example (Sample 90)



MCMC Example (Sample 100)



Learning from Sparse and Noisy Data

Particulary important when studying complex physical or biological systems where only **very sparse and noisy data** is available, but good mathematical models exist to describe the system.

Examples:

- Atmospheric, ocean or subsurface flow
- Cardiovascular system or tracer diffusion in brain imaging
- Structural mechanics of composite materials or bones

Learning from Sparse and Noisy Data

Particulary important when studying complex physical or biological systems where only **very sparse and noisy data** is available, but good mathematical models exist to describe the system.

Examples:

- Atmospheric, ocean or subsurface flow
- Cardiovascular system or tracer diffusion in brain imaging
- Structural mechanics of composite materials or bones

Machine Learning and **Neural Networks** alone will **not be sufficient!**

Need to add **mathematical modelling** & **scientific computing** to toolkit:

New Challenges! Sashi's first lecture

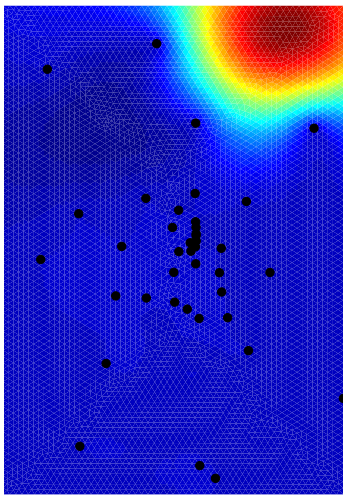
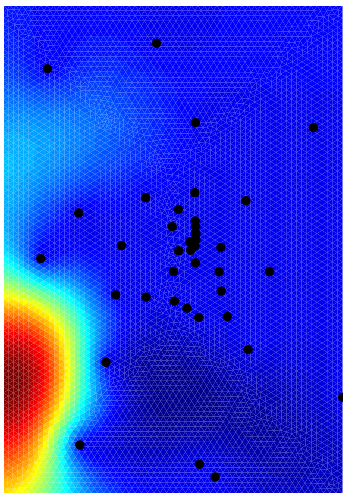
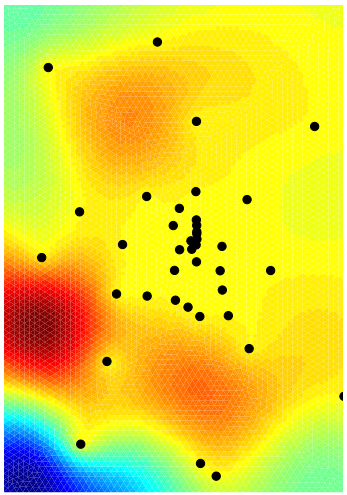
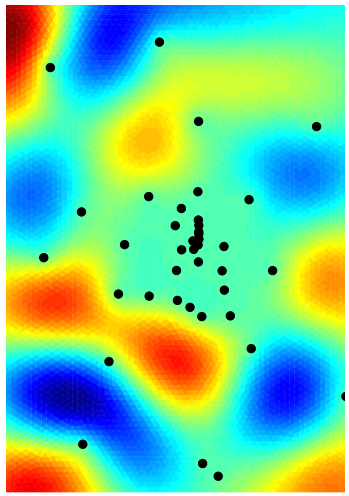
Data for Radioactive Waste Example (WIPP)

Prior Model [Ernst et al, 2014]

$$\log a \approx \sum_{j=1}^{s} \sqrt{\mu_j}\, \phi_j^{\text{cond}}(x)\theta_j(\omega) \text{ with i.i.d. } \theta_j \sim \mathrm{N}(0,1)$$

Karhunen-Loève modes ($j = 1, 2, 9, 16$)
conditioned on 38 permeability observations

(via kriging = Gaussian process regression = low-rank change to covariance operator)

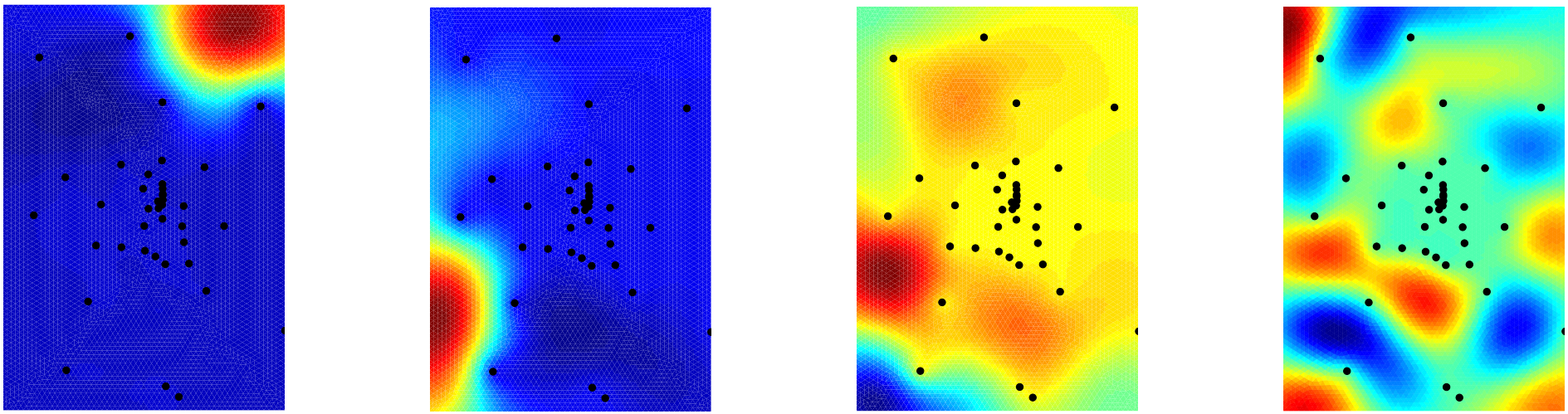
Data for Radioactive Waste Example (WIPP)

Prior Model [Ernst et al, 2014]

$$\log a \approx \sum_{j=1}^{s} \sqrt{\mu_j}\, \phi_j^{\text{cond}}(x)\theta_j(\omega) \text{ with i.i.d. } \theta_j \sim \mathrm{N}(0,1)$$

Karhunen-Loève modes ($j = 1, 2, 9, 16$)
conditioned on 38 permeability observations

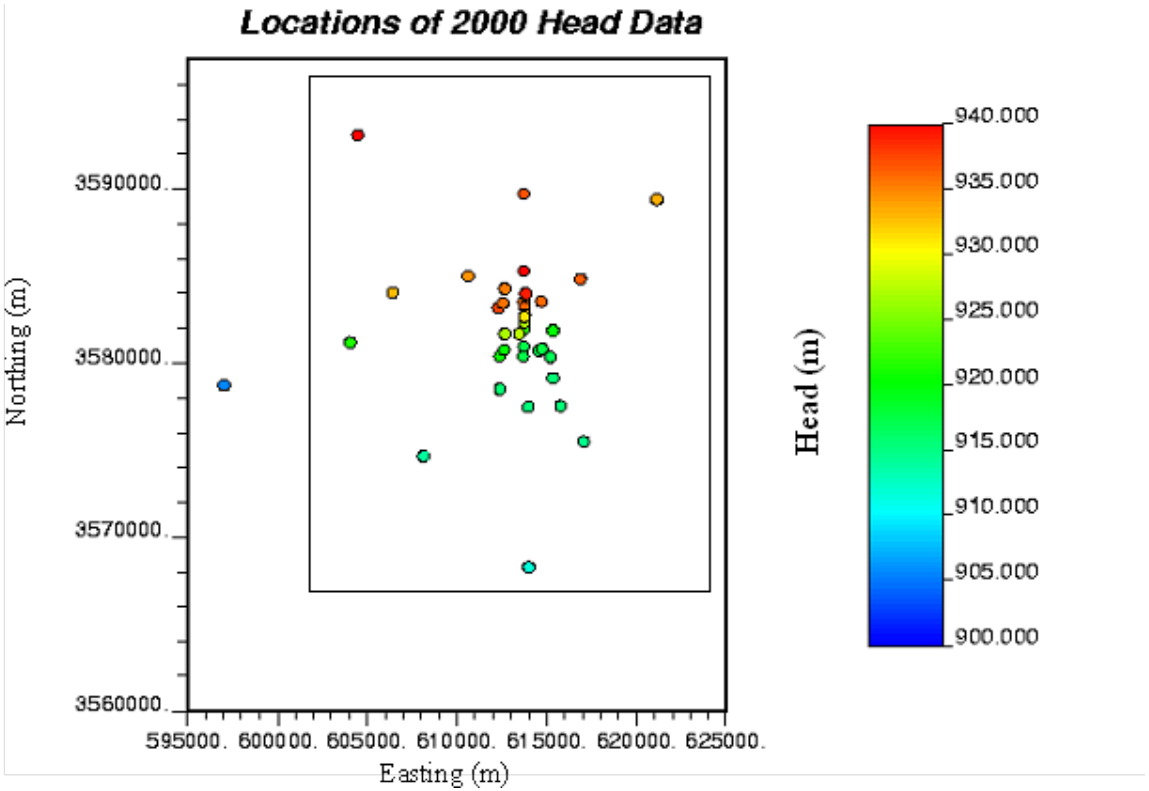
(via kriging = Gaussian process regression = low-rank change to covariance operator)



Prior model: $\pi_{\text{pr},s}(\boldsymbol{\theta})$ is multivariate standard Gaussian density for $\boldsymbol{\theta} \in \mathbb{R}^s$

Data for Radioactive Waste Example (WIPP)

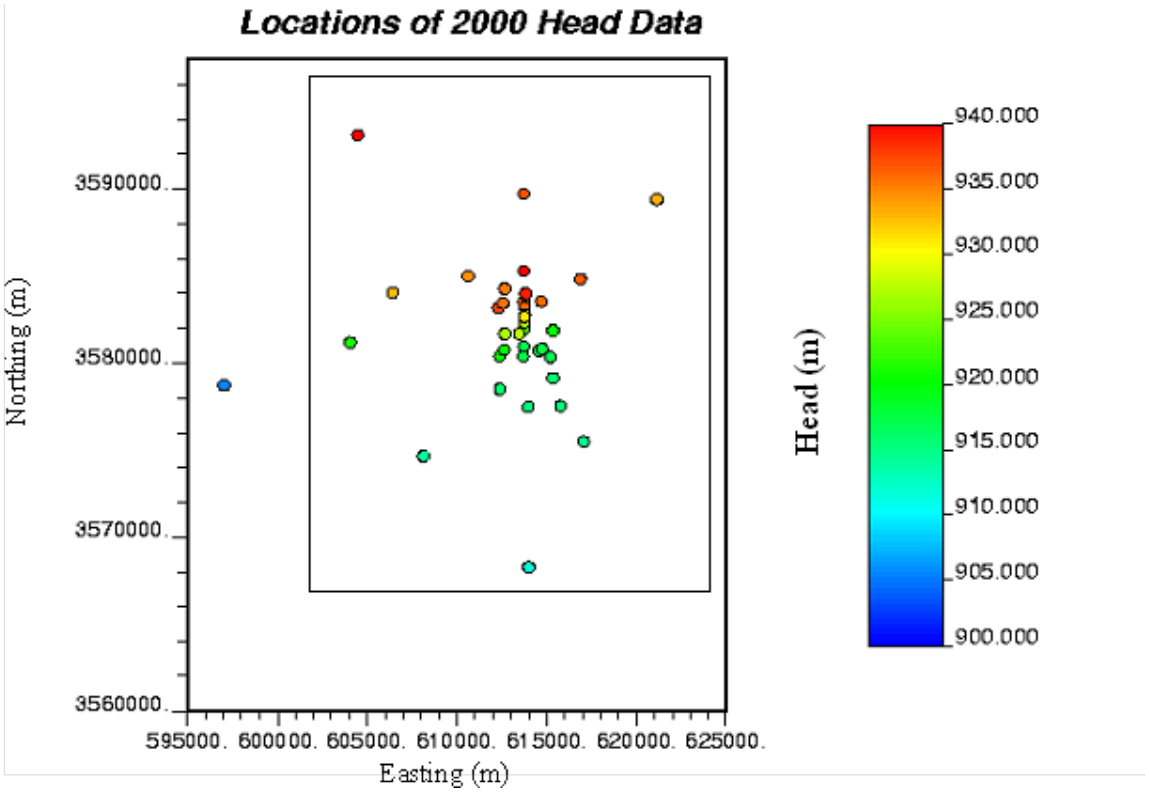
Likelihood Model [Ernst et al, 2014]



- Data y are pressure measurements.
- $F_h(\theta)$ is the model response.

Data for Radioactive Waste Example (WIPP)

Likelihood Model [Ernst et al, 2014]



- Data y are pressure measurements.
- $F_h(\theta)$ is the model response.

Likelihood model: assuming Gaussian errors with covariance Σ

$$\pi_{h,s}(y|\theta) \;\approx\; \exp(-\|y - F_h(\theta)\|^2_{\Sigma^{-1}})$$

Applying MCMC to Radioactive Waste Problem

Posterior through **Bayes' rule**: $\pi_{h,s}(\theta \,|\, y) \;\asymp\; \pi_{h,s}(y|\theta)\, \pi_{\mathrm{pr},s}(\theta)$

Applying MCMC to Radioactive Waste Problem

Posterior through **Bayes' rule**: $\pi_{h,s}(\theta \,|\, y) \;\propto\; \pi_{h,s}(y|\theta)\,\pi_{\mathrm{pr},s}(\theta)$

Use **Metropolis-Hastings MCMC** (Algorithm 1) to produce samples θ^n.
Then use the samples to estimate the posterior expectation of Q:

$$\widehat{Q}_{h,s}^{\mathrm{MetH}} := \frac{1}{N}\sum_{n=1}^{N} Q_{h,s}(\theta^n) \;\approx\; \mathbb{E}_{\pi^\infty}[Q]$$

Applying MCMC to Radioactive Waste Problem

Posterior through **Bayes' rule**: $\pi_{h,s}(\theta \,|\, y) \;\propto\; \pi_{h,s}(y|\theta)\,\pi_{\mathrm{pr},s}(\theta)$

Use **Metropolis-Hastings MCMC** (Algorithm 1) to produce samples θ^n.
Then use the samples to estimate the posterior expectation of Q:

$$\widehat{Q}_{h,s}^{\mathrm{MetH}} := \frac{1}{N}\sum_{n=1}^{N} Q_{h,s}(\theta^n) \;\approx\; \mathbb{E}_{\pi^\infty}[Q]$$

Pros:

- Markov chain $\theta^n \sim \pi_{h,s}$ as $n \to \infty$
  $\Rightarrow$ **"gold standard"** [Stuart et al]
- s-independent, e.g. via pCN sampler
  [Cotter, Dashti, Stuart, 2012]

Applying MCMC to Radioactive Waste Problem

Posterior through **Bayes' rule**: $\pi_{h,s}(\theta \,|\, y) \;\propto\; \pi_{h,s}(y|\theta)\,\pi_{\mathrm{pr},s}(\theta)$

Use **Metropolis-Hastings MCMC** (Algorithm 1) to produce samples θ^n.
Then use the samples to estimate the posterior expectation of Q:

$$\widehat{Q}_{h,s}^{\mathrm{MetH}} := \frac{1}{N}\sum_{n=1}^{N} Q_{h,s}(\theta^n) \;\approx\; \mathbb{E}_{\pi^\infty}[Q]$$

Pros:

- Markov chain $\theta^n \sim \pi_{h,s}$ as $n \to \infty$
  $\Rightarrow$ **"gold standard"** [Stuart et al]
- s-independent, e.g. via pCN sampler
  [Cotter, Dashti, Stuart, 2012]

Cons:

- $\alpha_{h,s}$ very expensive for $h \ll 1$.
- $\alpha_{h,s} < 10\%$ for large s.
- $\mathscr{C}_\varepsilon(\widehat{Q}_{h,s}^{\mathrm{MetH}}) = \mathscr{O}(\varepsilon^{-2-\gamma/\alpha})$,
  but much bigger constant **!**

Applying MCMC to Radioactive Waste Problem

Posterior through **Bayes' rule**: $\pi_{h,s}(\theta \,|\, y) \;\asymp\; \pi_{h,s}(y|\theta)\,\pi_{\mathrm{pr},s}(\theta)$

Use **Metropolis-Hastings MCMC** (Algorithm 1) to produce samples θ^n.
Then use the samples to estimate the posterior expectation of Q:

$$\widehat{Q}_{h,s}^{\mathrm{MetH}} := \frac{1}{N}\sum_{n=1}^{N} Q_{h,s}(\theta^n) \;\approx\; \mathbb{E}_{\pi^\infty}[Q]$$

Pros:

- Markov chain $\theta^n \sim \pi_{h,s}$ as $n \to \infty$
  $\Rightarrow$ **"gold standard"** [Stuart et al]
- s-independent, e.g. via pCN sampler
  [Cotter, Dashti, Stuart, 2012]

Cons:

- $\alpha_{h,s}$ very expensive for $h \ll 1$.
- $\alpha_{h,s} < 10\%$ for large s.
- $\mathscr{C}_\varepsilon(\widehat{Q}_{h,s}^{\mathrm{MetH}}) = \mathscr{O}(\varepsilon^{-2-\gamma/\alpha})$,
  but much bigger constant **!**

Can we again apply the multilevel idea?

Multilevel Markov Chain Monte Carlo

Multilevel Markov Chain Monte Carlo – Idea

[Dodwell, Ketelsen, RS, Teckentrup, JUQ 2015] & [Dodwell et al, SIAM Rev. 2019]

What were the **key ingredients** of "standard" multilevel Monte Carlo?

Multilevel Markov Chain Monte Carlo – Idea

[Dodwell, Ketelsen, RS, Teckentrup, JUQ 2015] & [Dodwell et al, SIAM Rev. 2019]

What were the **key ingredients** of "standard" multilevel Monte Carlo?

- **Telescoping sum:** $\mathbb{E}[Q_L] = \mathbb{E}[Q_0] + \sum_{\ell=1}^{L} \mathbb{E}[Q_\ell - Q_{\ell-1}]$
- Models on coarser levels **much cheaper** to solve ($h_0^{-d} \ll h_L^{-d}$).
- $\mathbb{V}[Q_\ell - Q_{\ell-1}] \overset{\ell\to\infty}{\longrightarrow} \to 0$ as $\Longrightarrow$ much **fewer samples** on finer levels.

Multilevel Markov Chain Monte Carlo – Idea

[Dodwell, Ketelsen, RS, Teckentrup, JUQ 2015] & [Dodwell et al, SIAM Rev. 2019]

What were the **key ingredients** of "standard" multilevel Monte Carlo?

- **Telescoping sum:** $\mathbb{E}[Q_L] = \mathbb{E}[Q_0] + \sum_{\ell=1}^{L} \mathbb{E}[Q_\ell - Q_{\ell-1}]$
- Models on coarser levels **much cheaper** to solve ($h_0^{-d} \ll h_L^{-d}$).
- $\mathbb{V}[Q_\ell - Q_{\ell-1}] \overset{\ell\to\infty}{\longrightarrow} \to 0$ as $\Longrightarrow$ much **fewer samples** on finer levels.

But Important! Now target distribution $\pi_\ell := \pi_{h_\ell, s_\ell}(\cdot \,|\, y)$ **depends on** ℓ:

$$\mathbb{E}_{\pi_L}[Q_L] = \mathbb{E}_{\pi_0}[Q_0] + \sum_\ell \mathbb{E}_{\pi_\ell}[Q_\ell] - \mathbb{E}_{\pi_{\ell-1}}[Q_{\ell-1}]$$

Multilevel Markov Chain Monte Carlo – Idea

[Dodwell, Ketelsen, RS, Teckentrup, JUQ 2015] & [Dodwell et al, SIAM Rev. 2019]

What were the **key ingredients** of "standard" multilevel Monte Carlo?

- **Telescoping sum:** $\mathbb{E}[Q_L] = \mathbb{E}[Q_0] + \sum_{\ell=1}^{L} \mathbb{E}[Q_\ell - Q_{\ell-1}]$
- Models on coarser levels **much cheaper** to solve ($h_0^{-d} \ll h_L^{-d}$).
- $\mathbb{V}[Q_\ell - Q_{\ell-1}] \overset{\ell\to\infty}{\longrightarrow} \to 0$ as $\Longrightarrow$ much **fewer samples** on finer levels.

But Important! Now target distribution $\pi_\ell := \pi_{h_\ell, s_\ell}(\cdot \,|\, y)$ **depends on** ℓ:

$$\mathbb{E}_{\pi_L}[Q_L] = \underbrace{\mathbb{E}_{\pi_0}[Q_0]}_{\text{standard MCMC}} + \sum_\ell \underbrace{\mathbb{E}_{\pi_\ell}[Q_\ell] - \mathbb{E}_{\pi_{\ell-1}}[Q_{\ell-1}]}_{\text{multilevel MCMC (NEW)}}$$

$$\widehat{Q}_{h,s}^{\text{MLMH}} := \frac{1}{N_0} \sum_{n=1}^{N_0} Q_0(\boldsymbol{\Theta}_{0,0}^n) + \sum_{\ell=1}^{L} \frac{1}{N_\ell} \sum_{n=1}^{N_\ell} \left(Q_\ell(\boldsymbol{\Theta}_{\ell,\ell}^n) - Q_{\ell-1}(\boldsymbol{\Theta}_{\ell,\ell-1}^n) \right)$$

with **correlated** Markov chains $\{\boldsymbol{\Theta}_{\ell,\ell-1}^n\}$ and $\{\boldsymbol{\Theta}_{\ell,\ell}^n\}$ (see below).

Multilevel Markov Chain Monte Carlo – Idea

[Dodwell, Ketelsen, RS, Teckentrup, JUQ 2015] & [Dodwell et al, SIAM Rev. 2019]

What were the **key ingredients** of "standard" multilevel Monte Carlo?

- **Telescoping sum:** $\mathbb{E}[Q_L] = \mathbb{E}[Q_0] + \sum_{\ell=1}^{L} \mathbb{E}[Q_\ell - Q_{\ell-1}]$
- Models on coarser levels **much cheaper** to solve ($h_0^{-d} \ll h_L^{-d}$).
- $\mathbb{V}[Q_\ell - Q_{\ell-1}] \overset{\ell\to\infty}{\longrightarrow} \to 0$ as $\Longrightarrow$ much **fewer samples** on finer levels.

But Important! Now target distribution $\pi_\ell := \pi_{h_\ell, s_\ell}(\cdot \,|\, y)$ **depends on** ℓ:

$$\mathbb{E}_{\pi_L}[Q_L] = \underbrace{\mathbb{E}_{\pi_0}[Q_0]}_{\text{standard MCMC}} + \sum_\ell \underbrace{\mathbb{E}_{\pi_\ell}[Q_\ell] - \mathbb{E}_{\pi_{\ell-1}}[Q_{\ell-1}]}_{\text{multilevel MCMC (NEW)}}$$

$$\widehat{Q}_{h,s}^{\text{MLMH}} := \frac{1}{N_0} \sum_{n=1}^{N_0} Q_0(\boldsymbol{\Theta}_{0,0}^n) + \sum_{\ell=1}^{L} \frac{1}{N_\ell} \sum_{n=1}^{N_\ell} \left(Q_\ell(\boldsymbol{\Theta}_{\ell,\ell}^n) - Q_{\ell-1}(\boldsymbol{\Theta}_{\ell,\ell-1}^n) \right)$$

with **correlated** Markov chains $\{\boldsymbol{\Theta}_{\ell,\ell-1}^n\}$ and $\{\boldsymbol{\Theta}_{\ell,\ell}^n\}$ (see below).

For simplicity we describe only the case $s_\ell = s_{\ell-1} = \ldots = s_0$.
(In practice, useful to reduce also number $s_{\ell-1}$ of random parameters on coarser levels.

Multilevel Markov Chain Monte Carlo – Algorithm

Choose **subsampling rates** $t_0 \dots t_L \in \mathbb{N}$ (see below) and set $T_{\ell,k} := \prod_{j=k}^{\ell-1} t_j$.

ALGORITHM 2 (Multilevel Metropolis Hastings MCMC for $Q_\ell - Q_{\ell-1}$)

Given realisations $\boldsymbol{\theta}^n_{\ell,0} \dots \boldsymbol{\theta}^n_{\ell,\ell}$ at state n of Markov chains on levels $0 \le k \le \ell$.

1. $k = 0$: Set $\mathbf{x}_0^0 := \boldsymbol{\theta}^n_{\ell,0}$. Use **Algorithm 1** (standard Metropolis-Hastings) to generate samples $\mathbf{x}_0^i \sim \pi_0$, $i = 1, \dots, T_{\ell,0}$. Set $\boldsymbol{\theta}^{n+1}_{\ell,0} := \mathbf{x}_0^{T_{\ell,0}}$.

Multilevel Markov Chain Monte Carlo – Algorithm

Choose **subsampling rates** $t_0 \dots t_L \in \mathbb{N}$ (see below) and set $T_{\ell,k} := \prod_{j=k}^{\ell-1} t_j$.

ALGORITHM 2 (Multilevel Metropolis Hastings MCMC for $Q_\ell - Q_{\ell-1}$)

Given realisations $\boldsymbol{\theta}^n_{\ell,0} \dots \boldsymbol{\theta}^n_{\ell,\ell}$ at state n of Markov chains on levels $0 \le k \le \ell$.

1. $k = 0$: Set $\mathbf{x}^0_0 := \boldsymbol{\theta}^n_{\ell,0}$. Use **Algorithm 1** (standard Metropolis-Hastings) to generate samples $\mathbf{x}^i_0 \sim \pi_0$, $i = 1, \dots, T_{\ell,0}$. Set $\boldsymbol{\theta}^{n+1}_{\ell,0} := \mathbf{x}_0^{T_{\ell,0}}$.
2. $k > 0$: Set $\mathbf{x}^0_k := \boldsymbol{\theta}^n_{\ell,k}$. Generate samples $\mathbf{x}^i_k \sim \pi_k$, $i = 1 \dots T_{\ell,k}$ as follows:
   (a) Propose $\mathbf{x}'_k = \mathbf{x}_{k-1}^{(i+1)t_{k-1}}$

Subsample to reduce correlation!

Multilevel Markov Chain Monte Carlo – Algorithm

Choose **subsampling rates** $t_0 \dots t_L \in \mathbb{N}$ (see below) and set $T_{\ell,k} := \prod_{j=k}^{\ell-1} t_j$.

ALGORITHM 2 (Multilevel Metropolis Hastings MCMC for $Q_\ell - Q_{\ell-1}$)

Given realisations $\boldsymbol{\theta}_{\ell,0}^n \dots \boldsymbol{\theta}_{\ell,\ell}^n$ at state n of Markov chains on levels $0 \le k \le \ell$.

1. $k = 0$: Set $\mathbf{x}_0^0 := \boldsymbol{\theta}_{\ell,0}^n$. Use **Algorithm 1** (standard Metropolis-Hastings) to generate samples $\mathbf{x}_0^i \sim \pi_0$, $i = 1, \dots, T_{\ell,0}$. Set $\boldsymbol{\theta}_{\ell,0}^{n+1} := \mathbf{x}_0^{T_{\ell,0}}$.

2. $k > 0$: Set $\mathbf{x}_k^0 := \boldsymbol{\theta}_{\ell,k}^n$. Generate samples $\mathbf{x}_k^i \sim \pi_k$, $i = 1 \dots T_{\ell,k}$ as follows:

   (a) Propose $\mathbf{x}_k' = \mathbf{x}_{k-1}^{(i+1)t_{k-1}}$

   Subsample to reduce correlation!

   (b) Accept $\mathbf{x}_k'$ and set $\mathbf{x}_k^{i+1} = \mathbf{x}_k'$ with probability

$$\alpha_k^{\mathrm{ML}}(\mathbf{x}_k' | \mathbf{x}_k^i) = \min\left(1, \frac{\pi_k(\mathbf{x}_k')\, \mathrm{q}_k^{\mathrm{ML}}(\mathbf{x}_k^n \,|\, \mathbf{x}_k')}{\pi^k(\mathbf{x}_k^n)\, \mathrm{q}_k^{\mathrm{ML}}(\mathbf{x}_k' \,|\, \mathbf{x}_k^n)}\right)$$

   Otherwise set $\mathbf{x}_k^{i+1} = \mathbf{x}_k^i$.

Multilevel Markov Chain Monte Carlo – Algorithm

Choose **subsampling rates** $t_0 \dots t_L \in \mathbb{N}$ (see below) and set $T_{\ell,k} := \prod_{j=k}^{\ell-1} t_j$.

ALGORITHM 2 (Multilevel Metropolis Hastings MCMC for $Q_\ell - Q_{\ell-1}$)

Given realisations $\boldsymbol{\theta}_{\ell,0}^n \dots \boldsymbol{\theta}_{\ell,\ell}^n$ at state n of Markov chains on levels $0 \le k \le \ell$.

1. $k = 0$: Set $\mathbf{x}_0^0 := \boldsymbol{\theta}_{\ell,0}^n$. Use **Algorithm 1** (standard Metropolis-Hastings) to generate samples $\mathbf{x}_0^i \sim \pi_0$, $i = 1, \dots, T_{\ell,0}$. Set $\boldsymbol{\theta}_{\ell,0}^{n+1} := \mathbf{x}_0^{T_{\ell,0}}$.

2. $k > 0$: Set $\mathbf{x}_k^0 := \boldsymbol{\theta}_{\ell,k}^n$. Generate samples $\mathbf{x}_k^i \sim \pi_k$, $i = 1 \dots T_{\ell,k}$ as follows:

   (a) Propose $\mathbf{x}_k' = \mathbf{x}_{k-1}^{(i+1)t_{k-1}}$

   Subsample to reduce correlation!

   (b) Accept $\mathbf{x}_k'$ and set $\mathbf{x}_k^{i+1} = \mathbf{x}_k'$ with probability

   $$\alpha_k^{\mathrm{ML}}(\mathbf{x}_k' | \mathbf{x}_k^i) = \min\left(1, \frac{\pi_k(\mathbf{x}_k')\pi_{k-1}(\mathbf{x}_k^n)}{\pi_k(\mathbf{x}_k^n)\pi_{k-1}(\mathbf{x}_k')}\right)$$

   JS Liu, 2001

   Otherwise set $\mathbf{x}_k^{i+1} = \mathbf{x}_k^i$.

   (c) Set $\boldsymbol{\theta}_{\ell,k}^{n+1} := \mathbf{x}_k^{T_{\ell,k}}$ with $T_{\ell,k} := \prod_{j=k}^{\ell-1} t_j$.

Multilevel Markov Chain Monte Carlo – Algorithm

Choose **subsampling rates** $t_0 \dots t_L \in \mathbb{N}$ (see below) and set $T_{\ell,k} := \prod_{j=k}^{\ell-1} t_j$.

ALGORITHM 2 (Multilevel Metropolis Hastings MCMC for $Q_\ell - Q_{\ell-1}$)

Given realisations $\boldsymbol{\theta}^n_{\ell,0} \dots \boldsymbol{\theta}^n_{\ell,\ell}$ at state n of Markov chains on levels $0 \le k \le \ell$.

1. $k = 0$: Set $\mathbf{x}^0_0 := \boldsymbol{\theta}^n_{\ell,0}$. Use **Algorithm 1** (standard Metropolis-Hastings) to generate samples $\mathbf{x}^i_0 \sim \pi_0$, $i = 1, \dots, T_{\ell,0}$. Set $\boldsymbol{\theta}^{n+1}_{\ell,0} := \mathbf{x}_0^{T_{\ell,0}}$.

2. $k > 0$: Set $\mathbf{x}^0_k := \boldsymbol{\theta}^n_{\ell,k}$. Generate samples $\mathbf{x}^i_k \sim \pi_k$, $i = 1 \dots T_{\ell,k}$ as follows:

   (a) Propose $\mathbf{x}'_k = \mathbf{x}^{(i+1)t_{k-1}}_{k-1}$

   Subsample to reduce correlation!

   (b) Accept $\mathbf{x}'_k$ and set $\mathbf{x}^{i+1}_k = \mathbf{x}'_k$ with probability

   $$\alpha^{\mathrm{ML}}_k(\mathbf{x}'_k|\mathbf{x}^i_k) = \min\left(1, \frac{\pi_k(\mathbf{x}'_k)\pi_{k-1}(\mathbf{x}^n_k)}{\pi_k(\mathbf{x}^n_k)\pi_{k-1}(\mathbf{x}'_k)}\right)$$

   JS Liu, 2001

   Otherwise set $\mathbf{x}^{i+1}_k = \mathbf{x}^i_k$.

   (c) Set $\boldsymbol{\theta}^{n+1}_{\ell,k} := \mathbf{x}_k^{T_{\ell,k}}$ with $T_{\ell,k} := \prod_{j=k}^{\ell-1} t_j$.

3. Set $Y^n_\ell := Q_\ell(\boldsymbol{\theta}^n_{\ell,\ell}) - Q_{\ell-1}(\boldsymbol{\theta}^n_{\ell,\ell-1})$.

MLMCMC – Comments

- Each $\{\boldsymbol{\Theta}_{\ell,k}^n\}_{n\geq 1}$, $k = 0, \ldots, \ell$, is a **Markov chain** with $\boldsymbol{\Theta}_{\ell,k}^n \sim \pi_k$ as $n \to \infty$ and $t_\ell \to \infty$.

MLMCMC – Comments

- Each $\{\boldsymbol{\Theta}_{\ell,k}^n\}_{n\geq 1}$, $k = 0, \ldots, \ell$, is a **Markov chain** with $\boldsymbol{\Theta}_{\ell,k}^n \sim \pi_k$ as $n \to \infty$ and $t_\ell \to \infty$.
- Theoretically need $t_\ell \to \infty$ to guarantee **consistency** of multilevel algorithm (no bias between levels)

MLMCMC – Comments

- Each $\{\boldsymbol{\Theta}_{\ell,k}^n\}_{n\geq 1}$, $k = 0, \ldots, \ell$, is a **Markov chain** with $\boldsymbol{\Theta}_{\ell,k}^n \sim \pi_k$ as $n \to \infty$ and $t_\ell \to \infty$.
- Theoretically need $t_\ell \to \infty$ to guarantee **consistency** of multilevel algorithm (no bias between levels)
- In practice, suffices to choose $t_\ell \approx 1 - 2$ times the i.a.c.t.

  (integrated autocorrelation time)
- States may differ between level ℓ and $\ell - 1$:

State $n+1$	Level $\ell - 1$	Level ℓ
accept on level ℓ	$\boldsymbol{\theta}_{\ell,\ell-1}^{n+1}$	$\boldsymbol{\theta}_{\ell,\ell-1}^{n+1}$
reject on level ℓ	$\boldsymbol{\theta}_{\ell,\ell-1}^{n+1}$	$\boldsymbol{\theta}_{\ell,\ell}^{n}$

MLMCMC – Comments

- Each $\{\boldsymbol{\Theta}_{\ell,k}^n\}_{n\geq 1}$, $k = 0, \ldots, \ell$, is a **Markov chain** with $\boldsymbol{\Theta}_{\ell,k}^n \sim \pi_k$ as $n \to \infty$ and $t_\ell \to \infty$.
- Theoretically need $t_\ell \to \infty$ to guarantee **consistency** of multilevel algorithm (no bias between levels)
- In practice, suffices to choose $t_\ell \approx 1 - 2$ times the i.a.c.t.

  (integrated autocorrelation time)
- States may differ between level ℓ and $\ell - 1$:

State $n+1$	Level $\ell - 1$	Level ℓ
accept on level ℓ	$\boldsymbol{\theta}_{\ell,\ell-1}^{n+1}$	$\boldsymbol{\theta}_{\ell,\ell-1}^{n+1}$
reject on level ℓ	$\boldsymbol{\theta}_{\ell,\ell-1}^{n+1}$	$\boldsymbol{\theta}_{\ell,\ell}^{n}$

but not often for larger ℓ since **acceptance probability** $\boldsymbol{\alpha}_\ell^{\mathrm{ML}} \overset{\ell\to\infty}{\longrightarrow} 1$:

MLMCMC – Comments

- Each $\{\boldsymbol{\Theta}_{\ell,k}^n\}_{n\geq 1}$, $k = 0, \ldots, \ell$, is a **Markov chain** with $\boldsymbol{\Theta}_{\ell,k}^n \sim \pi_k$ as $n \to \infty$ and $t_\ell \to \infty$.
- Theoretically need $t_\ell \to \infty$ to guarantee **consistency** of multilevel algorithm (no bias between levels)
- In practice, suffices to choose $t_\ell \approx 1 - 2$ times the i.a.c.t.

  (integrated autocorrelation time)
- States may differ between level ℓ and $\ell - 1$:

State $n+1$	Level $\ell - 1$	Level ℓ
accept on level ℓ	$\boldsymbol{\theta}_{\ell,\ell-1}^{n+1}$	$\boldsymbol{\theta}_{\ell,\ell-1}^{n+1}$
reject on level ℓ	$\boldsymbol{\theta}_{\ell,\ell-1}^{n+1}$	$\boldsymbol{\theta}_{\ell,\ell}^{n}$

but not often for larger ℓ since **acceptance probability** $\boldsymbol{\alpha}_\ell^{\mathrm{ML}} \overset{\ell\to\infty}{\longrightarrow} 1$:

Lemma (Dodwell, Ketelsen, RS, Teckentrup, '15)

$$\mathbb{E}_{\pi_\ell,\pi_\ell}\Big[1 - \boldsymbol{\alpha}_\ell^{ML}(\cdot|\cdot)\Big] = \mathscr{O}\Big(\mathbb{E}_{\pi_{pr}}\Big[\big|F(\cdot) - F_\ell(\cdot)\big|\Big]\Big) = \mathscr{O}(h_\ell^\alpha)$$

Complexity Theorem for Multilevel MCMC (Dodwell et al. '15)

Suppose there are constants $\alpha, \beta, \gamma, \eta > 0$ such that, for all $\ell = 0, \ldots, L$,

M1 $\left|\mathbb{E}_{\pi_\ell}[Q_\ell] - \mathbb{E}_{\pi(\cdot|y)}[Q]\right| = \mathscr{O}(h_\ell^\alpha)$ (discretisation and truncation error)

M2' $\mathbf{Var}_{\text{alg}}[\widehat{Y}_\ell] + \left(\mathbb{E}_{\text{alg}}[\widehat{Y}_\ell] - \mathbb{E}_{\pi_\ell,\pi_{\ell-1}}[\widehat{Y}_\ell]\right)^2 = \mathbf{Var}_{\pi_\ell,\pi_{\ell-1}}[Y_\ell]\,\mathscr{O}(N_\ell^{-1})$ (MCMC-error)

M2 $\mathbf{Var}_{\pi_\ell,\pi_{\ell-1}}[Y_\ell] = \mathscr{O}(h_\ell^\beta)$ (multilevel variance decay)

M3 $\text{Cost}(\widehat{Y}_\ell^{\text{MC}}) = \mathscr{O}(N_\ell\, h_\ell^{-\gamma})$. (cost per sample)

Then there exist L, $\{N_\ell\}_{\ell=0}^L$ s.t. MSE $< \varepsilon^2$ and

$$\mathscr{C}_\varepsilon(\widehat{Q}_{h,s}^{\text{MLMH}}) = \mathscr{O}\left(\varepsilon^{-2-\max\left(0, \frac{\gamma-\beta}{\alpha}\right)}\right) \quad (+ \text{ log-factor when } \beta = \gamma)$$

Complexity Theorem for Multilevel MCMC (Dodwell et al. '15)

Suppose there are constants $\alpha, \beta, \gamma, \eta > 0$ such that, for all $\ell = 0, \ldots, L$,

M1 $\left|\mathbb{E}_{\pi_\ell}[Q_\ell] - \mathbb{E}_{\pi(\cdot|y)}[Q]\right| = \mathscr{O}(h_\ell^\alpha)$ (discretisation and truncation error)

M2' $\mathbf{Var}_{\text{alg}}[\widehat{Y}_\ell] + \left(\mathbb{E}_{\text{alg}}[\widehat{Y}_\ell] - \mathbb{E}_{\pi_\ell,\pi_{\ell-1}}[\widehat{Y}_\ell]\right)^2 = \mathbf{Var}_{\pi_\ell,\pi_{\ell-1}}[Y_\ell]\,\mathscr{O}(N_\ell^{-1})$ (MCMC-error)

M2 $\mathbf{Var}_{\pi_\ell,\pi_{\ell-1}}[Y_\ell] = \mathscr{O}(h_\ell^\beta)$ (multilevel variance decay)

M3 $\text{Cost}(\widehat{Y}_\ell^{\text{MC}}) = \mathscr{O}(N_\ell\, h_\ell^{-\gamma})$. (cost per sample)

Then there exist L, $\{N_\ell\}_{\ell=0}^L$ s.t. $\text{MSE} < \varepsilon^2$ and

$$\mathscr{C}_\varepsilon(\widehat{Q}_{h,s}^{\text{MLMH}}) = \mathscr{O}\left(\varepsilon^{-2-\max\left(0, \frac{\gamma-\beta}{\alpha}\right)}\right) \quad (+ \text{ log-factor when } \beta = \gamma)$$

- Proof of Assumptions **M1** and **M3** similar to i.i.d. case.
- **M2' not** specific to MLMCMC; first steps in [Hairer, Stuart, Vollmer, '11].

Complexity Theorem for Multilevel MCMC (Dodwell et al. '15)

Suppose there are constants $\alpha, \beta, \gamma, \eta > 0$ such that, for all $\ell = 0, \ldots, L$,

M1 $\left|\mathbb{E}_{\pi_\ell}[Q_\ell] - \mathbb{E}_{\pi(\cdot|y)}[Q]\right| = \mathscr{O}(h_\ell^\alpha)$ (discretisation and truncation error)

M2' $\mathbf{Var}_{\text{alg}}[\widehat{Y}_\ell] + \left(\mathbb{E}_{\text{alg}}[\widehat{Y}_\ell] - \mathbb{E}_{\pi_\ell,\pi_{\ell-1}}[\widehat{Y}_\ell]\right)^2 = \mathbf{Var}_{\pi_\ell,\pi_{\ell-1}}[Y_\ell]\,\mathscr{O}(N_\ell^{-1})$ (MCMC-error)

M2 $\mathbf{Var}_{\pi_\ell,\pi_{\ell-1}}[Y_\ell] = \mathscr{O}(h_\ell^\beta)$ (multilevel variance decay)

M3 $\text{Cost}(\widehat{Y}_\ell^{\text{MC}}) = \mathscr{O}(N_\ell\, h_\ell^{-\gamma})$. (cost per sample)

Then there exist L, $\{N_\ell\}_{\ell=0}^L$ s.t. MSE $< \varepsilon^2$ and

$$\mathscr{C}_\varepsilon(\widehat{Q}_{h,s}^{\text{MLMH}}) = \mathscr{O}\left(\varepsilon^{-2-\max\left(0, \frac{\gamma-\beta}{\alpha}\right)}\right) \quad (+ \text{ log-factor when } \beta = \gamma)$$

- Proof of Assumptions **M1** and **M3** similar to i.i.d. case.
- **M2' not** specific to MLMCMC; first steps in [Hairer, Stuart, Vollmer, '11].

Proof of **M2** for lognormal diffusion & linear FEs (Dodwell et al '15)

$$\mathbf{Var}_{\pi_\ell,\pi_{\ell-1}}\left[Q_\ell(\boldsymbol{\Theta}_{\ell,\ell}^n) - Q_{\ell-1}(\boldsymbol{\Theta}_{\ell,\ell-1}^n)\right] = \mathscr{O}(h_\ell^\alpha) \quad (\text{unfortunately } \beta = \alpha \text{ not } 2\alpha)$$

More Comments – Related Literature

- Typically also **increase number of parameters** s_ℓ from level to level and use standard proposal kernel for new parameters (see paper).
- **Subsampling** essential (exact only in limit of infinite subsampling), but small bias for sampling rates of the size of the i.a.c.t.

More Comments – Related Literature

- Typically also **increase number of parameters** s_ℓ from level to level and use standard proposal kernel for new parameters (see paper).
- **Subsampling** essential (exact only in limit of infinite subsampling), but small bias for sampling rates of the size of the i.a.c.t.
- **New** ('multiplicative') version: Current work with Colin Fox (Otago)

More Comments – Related Literature

- Typically also **increase number of parameters** s_ℓ from level to level and use standard proposal kernel for new parameters (see paper).
- **Subsampling** essential (exact only in limit of infinite subsampling), but small bias for sampling rates of the size of the i.a.c.t.
- **New** ('multiplicative') version: Current work with Colin Fox (Otago)
- Algorithm 2 is a special case of a **surrogate transition method** [Liu, Monte Carlo Strategies in Scientific Computing, 2001, §9.4.3]
- and of **delayed acceptance Metropolis-Hastings** [Christen, Fox, '05]

More Comments – Related Literature

- Typically also **increase number of parameters** s_ℓ from level to level and use standard proposal kernel for new parameters (see paper).
- **Subsampling** essential (exact only in limit of infinite subsampling), but small bias for sampling rates of the size of the i.a.c.t.
- **New** ('multiplicative') version: Current work with Colin Fox (Otago)
- Algorithm 2 is a special case of a **surrogate transition method** [Liu, Monte Carlo Strategies in Scientific Computing, 2001, §9.4.3]
- and of **delayed acceptance Metropolis-Hastings** [Christen, Fox, '05]

But crucially exploit also variance reduction & prove rates in MLMCMC

(Current work with C. Fox: adaptive error estimates using the multilevel Markov chains)

More Comments – Related Literature

- Typically also **increase number of parameters** s_ℓ from level to level and use standard proposal kernel for new parameters (see paper).
- **Subsampling** essential (exact only in limit of infinite subsampling), but small bias for sampling rates of the size of the i.a.c.t.
- **New** ('multiplicative') version: Current work with Colin Fox (Otago)
- Algorithm 2 is a special case of a **surrogate transition method** [Liu, Monte Carlo Strategies in Scientific Computing, 2001, §9.4.3]
- and of **delayed acceptance Metropolis-Hastings** [Christen, Fox, '05]

But crucially exploit also variance reduction & prove rates in MLMCMC

(Current work with C. Fox: adaptive error estimates using the multilevel Markov chains)

- Other references on related **mutlilevel MC methods** for Bayesian inverse problems:
  - Hoang, Schwab & Stuart, *Inverse Prob* **29**, 2013
  - Beskos, Jasra, Law & Zhou, *Stoch Proc Appl*, 2017

Numerical Example

Fruit fly (2D lognormal diffusion) on $D = (0,1)^2$ with linear FEs

- **Prior:** Separable exponential covariance with $\sigma^2 = 1$, $\lambda = 0.5$.

  i.e. $\mathbb{E}[Z(x)Z(x')] = \sigma^2 e^{-\frac{|x-x'|}{\lambda} - \frac{|y-y'|}{\lambda}}$

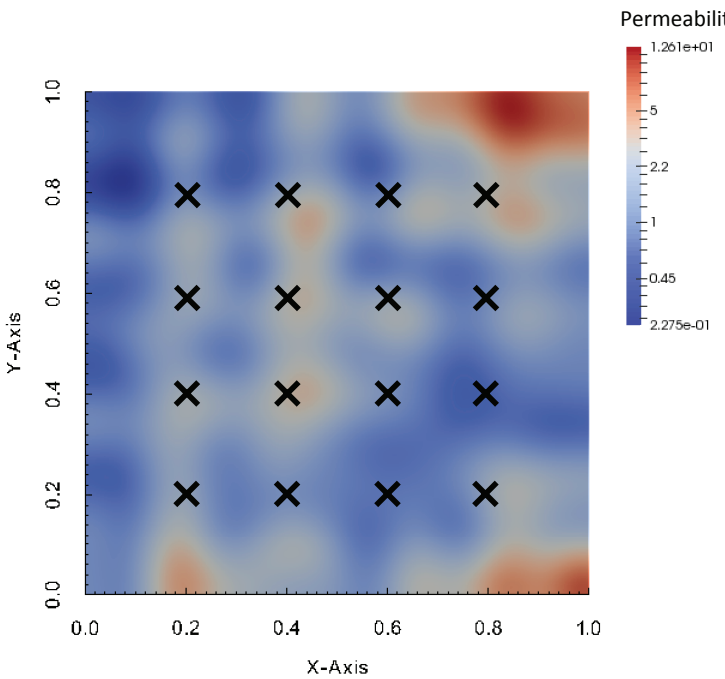
Numerical Example

Fruit fly (2D lognormal diffusion) on $D = (0,1)^2$ with linear FEs

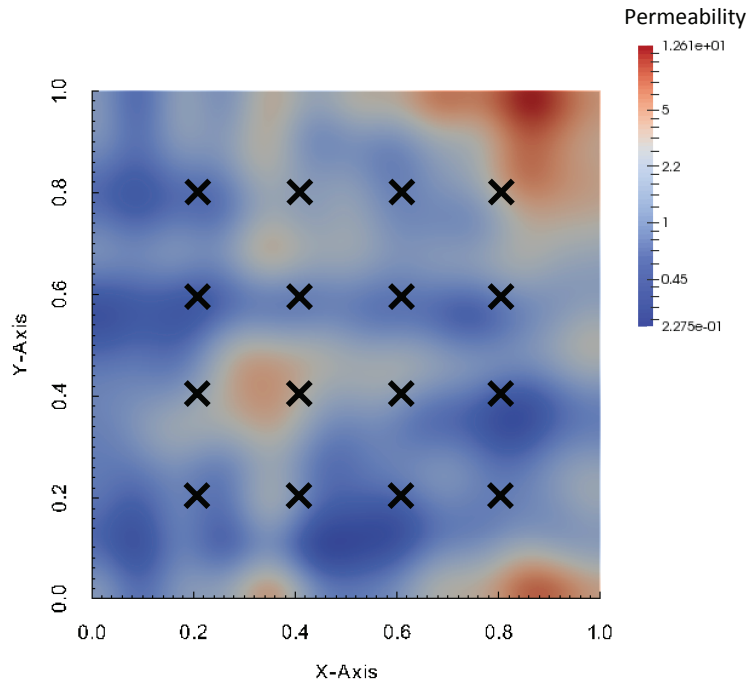
- **Prior:** Separable exponential covariance with $\sigma^2 = 1$, $\lambda = 0.5$.

  i.e. $\mathbb{E}[Z(x)Z(x')] = \sigma^2 e^{-\frac{|x-x'|}{\lambda} - \frac{|y-y'|}{\lambda}}$

- **"Data"** y: Pressure at 16 points $x_j^* \in D$ & error covariance $\Sigma = 10^{-4} I$.



Synthetic Data



Posterior Sample

Numerical Example

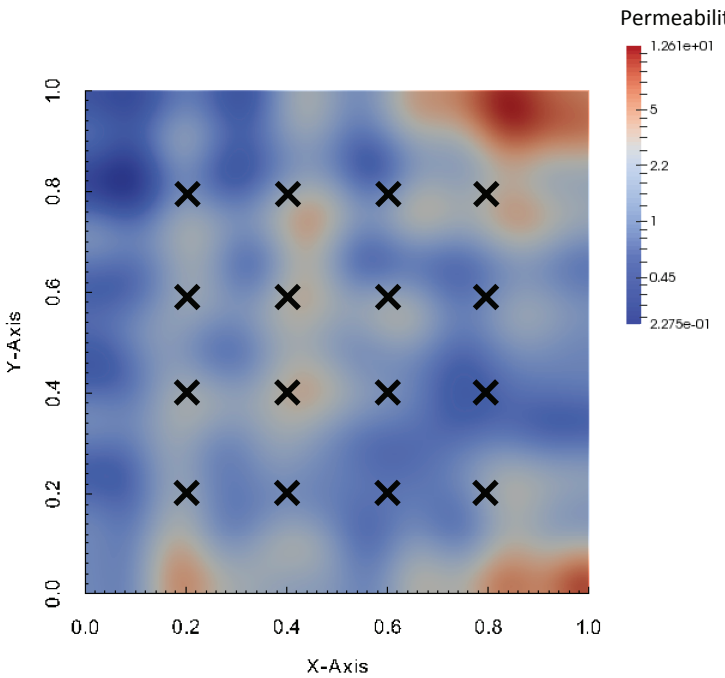
Fruit fly (2D lognormal diffusion) on $D = (0,1)^2$ with linear FEs

- **Prior:** Separable exponential covariance with $\sigma^2 = 1$, $\lambda = 0.5$.

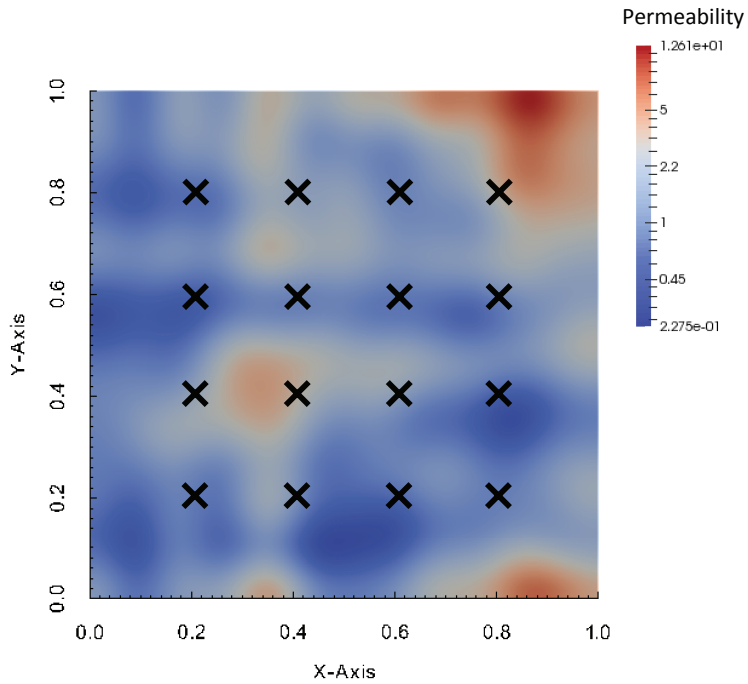
  i.e. $\mathbb{E}[Z(x)Z(x')] = \sigma^2 e^{-\frac{|x-x'|}{\lambda} - \frac{|y-y'|}{\lambda}}$

- **"Data"** y: Pressure at 16 points $x_j^* \in D$ & error covariance $\Sigma = 10^{-4} I$.

- pCN-proposals [Cotter, Dashti, Stuart, 2012]



Synthetic Data

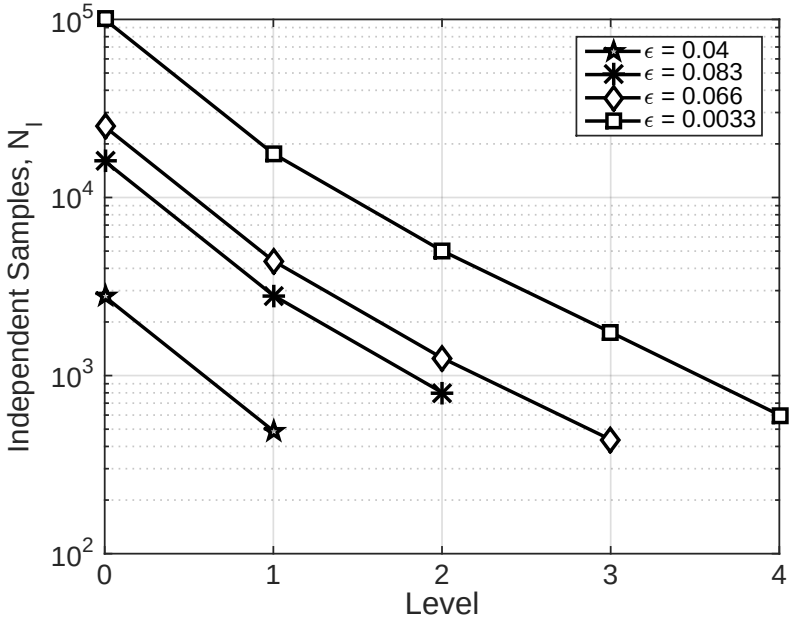


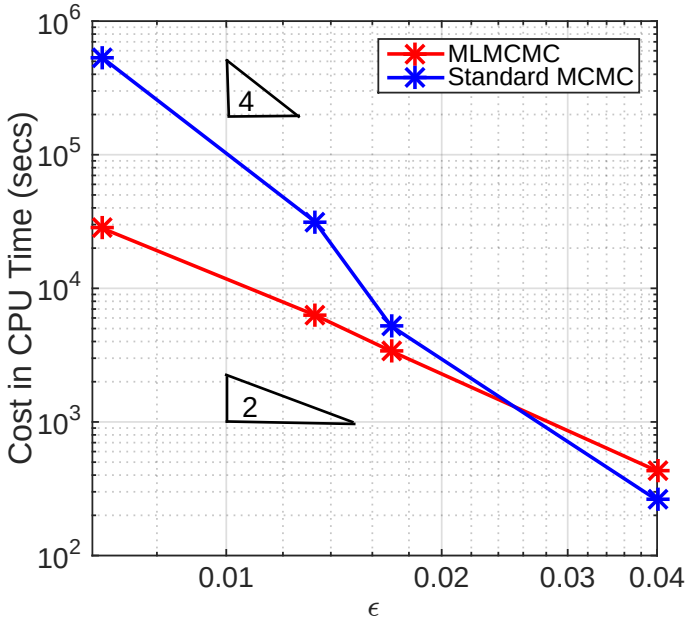
Posterior Sample

Numerical Example

Quantity of interest: $Q = \int_0^1 k \nabla p \, dx_2$; coarsest mesh size: $h_0 = \frac{1}{9}$

- 5-level method with #KL modes increasing from $s_0 = 50$ to $s_4 = 150$

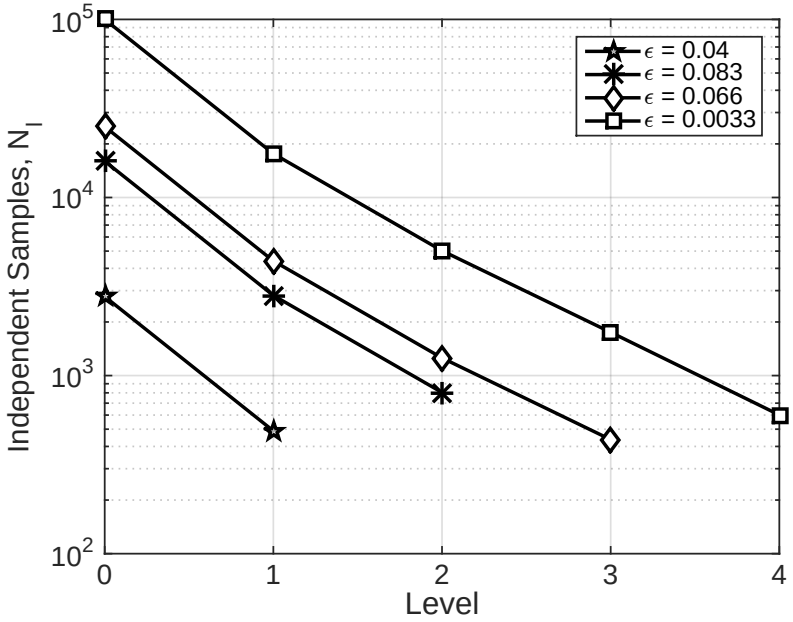


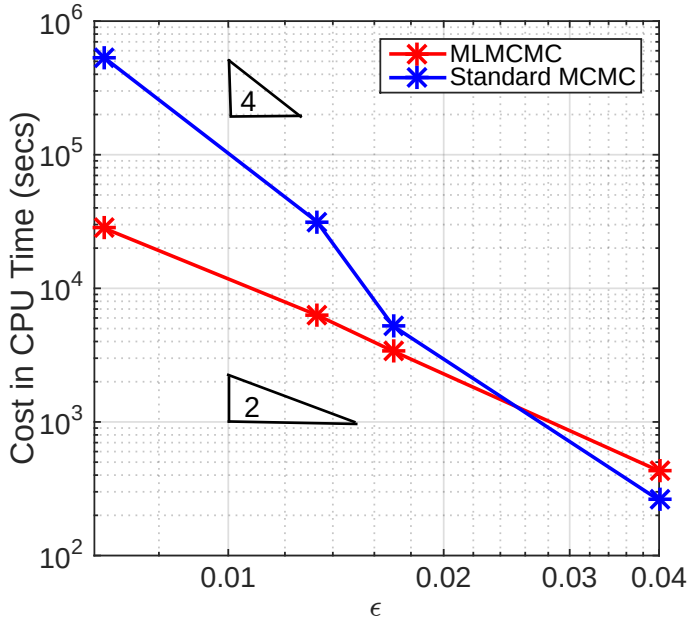


Numerical Example

Quantity of interest: $Q = \int_0^1 k \nabla p \, dx_2$; coarsest mesh size: $h_0 = \frac{1}{9}$

- 5-level method with #KL modes increasing from $s_0 = 50$ to $s_4 = 150$





- #independent samples $= \dfrac{N_\ell}{\tau_\ell}$ (τ_ℓ . . . integrated autocorrelation time)

Level ℓ	0	1	2	3	4
i.a.c. time τ_ℓ	136.23	3.66	2.93	1.46	1.23

Choice of Proposal Distribution

Multilevel DILI (recent preprint with T Cui & G Detommaso)

- **So far:** pCN random walk proposal (uses no gradient/Hessian info)
  [Cotter, Dashti, Stuart, '12]

Choice of Proposal Distribution

Multilevel DILI (recent preprint with T Cui & G Detommaso)

- **So far:** pCN random walk proposal (uses no gradient/Hessian info)
  [Cotter, Dashti, Stuart, '12]
- **Problem:** Dimension independent but very high IACT for $s \to \infty$!
  $\tau_0 \approx 136$ above, i.e. need 136 samples to obtain **one independent** sample!!

Choice of Proposal Distribution

Multilevel DILI (recent preprint with T Cui & G Detommaso)

- **So far:** pCN random walk proposal (uses no gradient/Hessian info)
  [Cotter, Dashti, Stuart, '12]
- **Problem:** Dimension independent but very high IACT for $s \to \infty$!
  $\tau_0 \approx 136$ above, i.e. need 136 samples to obtain **one independent** sample!!
- **However**, can use any other proposal (e.g. MALA, stochastic Newton)

Choice of Proposal Distribution

Multilevel DILI (recent preprint with T Cui & G Detommaso)

- **So far:** pCN random walk proposal (uses no gradient/Hessian info) [Cotter, Dashti, Stuart, '12]
- **Problem:** Dimension independent but very high IACT for $s \to \infty$!
  $\tau_0 \approx 136$ above, i.e. need 136 samples to obtain **one independent** sample!!
- **However**, can use any other proposal (e.g. MALA, stochastic Newton)
- DILI (dimension-independent likelihood-informed) MCMC [Cui, Law, Marzouk '16]
  samples from preconditioned Langevin equation using **low-rank** approximation of data-misfit Hessian at some points (incl. MAP point)

Roland Herzog's talk

Choice of Proposal Distribution

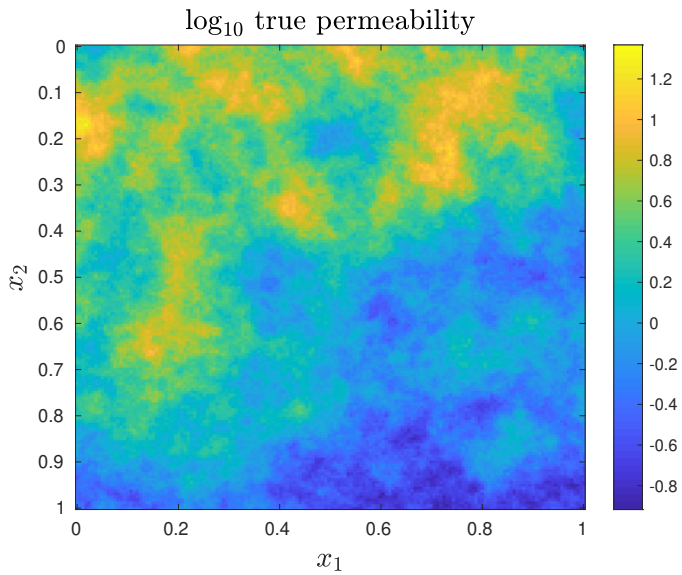
Multilevel DILI (recent preprint with T Cui & G Detommaso)

- **So far:** pCN random walk proposal (uses no gradient/Hessian info) [Cotter, Dashti, Stuart, '12]
- **Problem:** Dimension independent but very high IACT for $s \to \infty$!
  $\tau_0 \approx 136$ above, i.e. need 136 samples to obtain **one independent** sample!!
- **However**, can use any other proposal (e.g. MALA, stochastic Newton)
- DILI (dimension-independent likelihood-informed) MCMC [Cui, Law, Marzouk '16]
  samples from preconditioned Langevin equation using **low-rank** approximation of data-misfit Hessian at some points (incl. MAP point) Roland Herzog's talk
- **New** multilevel construction of DILI (with T Cui and G Detommaso) ...

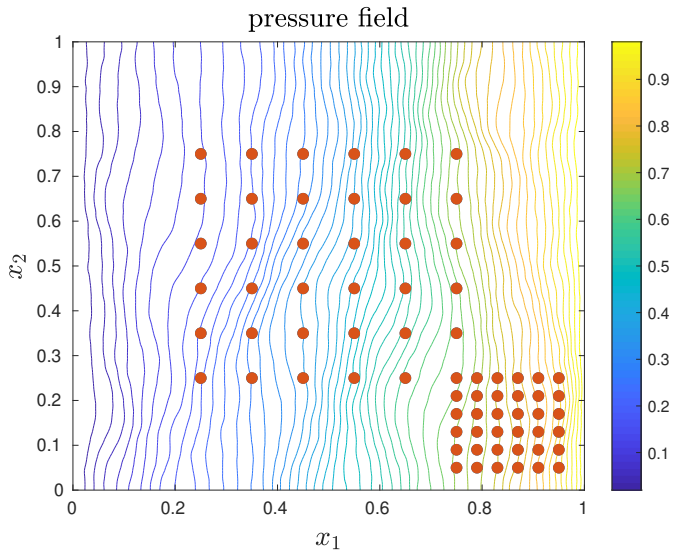
Cui, Detommaso, **RS**, Multilevel dimension-independent likelihood-informed MCMC for large-scale inverse problems, submitted, 2019 [arXiv:1910.12431]

Numerical Test on a Harder Example

True parameter



Pressure



Model:

$$-\nabla \cdot \left(e^{z(x)} \nabla u(s) \right) = 0, \quad x \in [0,1]^2$$

Top/bottom: zero Neumann b.c.; left/right: Dirichlet b.c. zero/one, respectively.

Prior: $z = \log a$ Gaussian w. exponential covariance $k(x, x') = \exp(-5|x - x'|)$

Data: 71 sensors; signal to noise ratio 50.

QoI: $Q^{(\text{flux})}$ = average flux over the left boundary

Numerical Comparison: IACTs & CPU Times

Refined parameters

Level ℓ	0	1	2	3
iact(pCN)	4300	45	48	24
iact(DILI)	34	11	3.6	2.0

$Q_\ell(\boldsymbol{\theta}^n_{\ell,\ell}) - Q_{\ell-1}(\boldsymbol{\theta}^n_{\ell,\ell-1})$

Level ℓ	0	1	2	3
iact(pCN)	4100	4.9	2.8	1.9
iact(DILI)	9.0	4.6	2.4	1.8

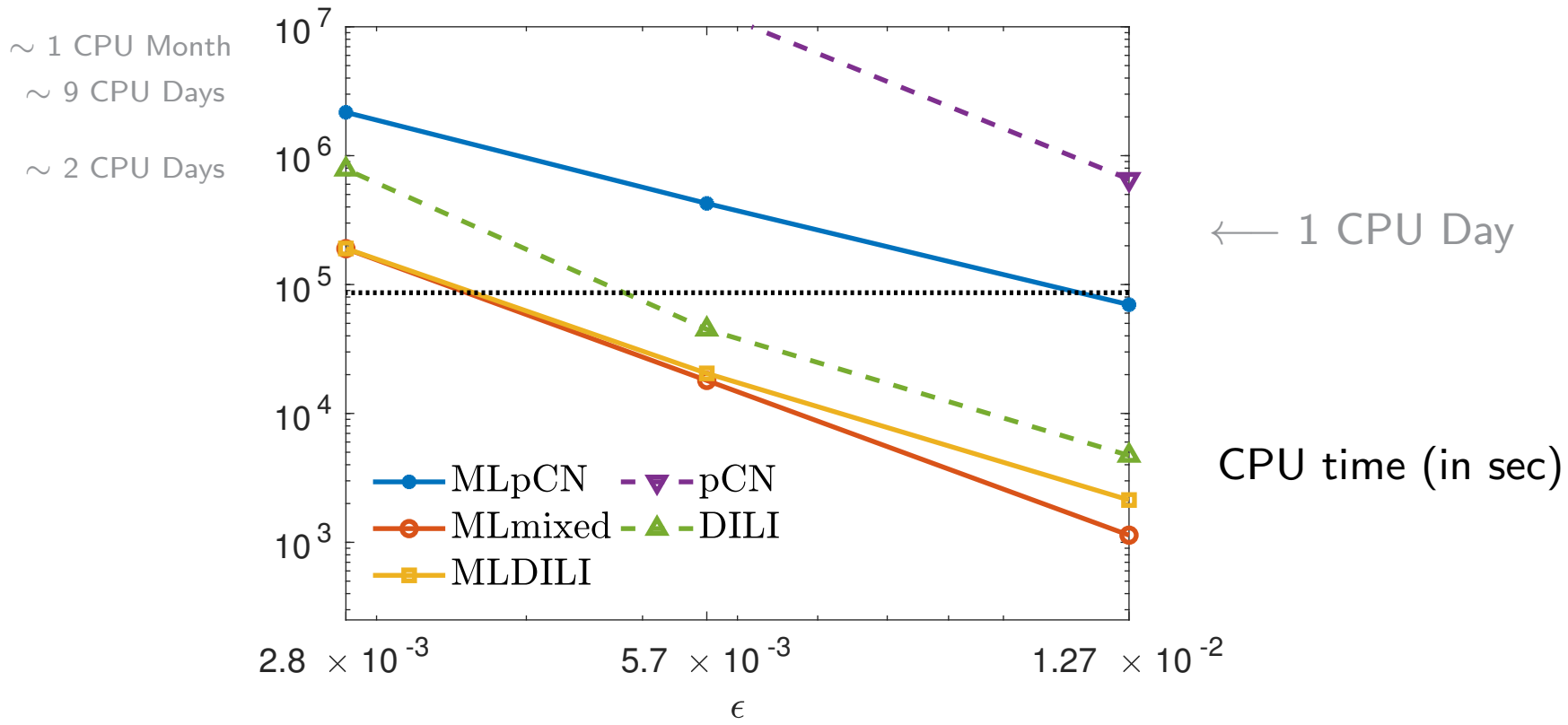
Numerical Comparison: IACTs & CPU Times

Refined parameters

Level ℓ	0	1	2	3
iact(pCN)	4300	45	48	24
iact(DILI)	34	11	3.6	2.0

$Q_\ell(\theta^n_{\ell,\ell}) - Q_{\ell-1}(\theta^n_{\ell,\ell-1})$

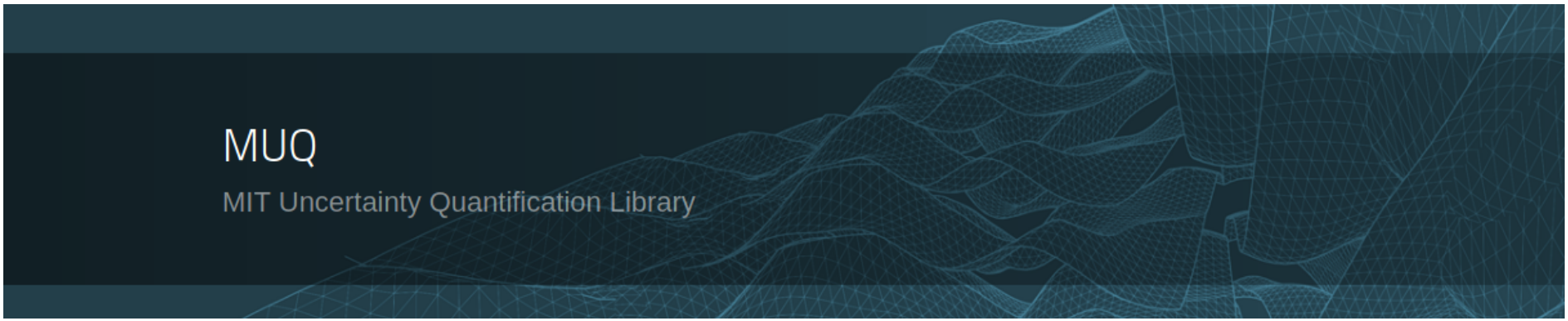
Level ℓ	0	1	2	3
iact(pCN)	4100	4.9	2.8	1.9
iact(DILI)	9.0	4.6	2.4	1.8



Parallel Multilevel MCMC

Software & HPC Experiments

MUQ - MIT Uncertainty Quantification Library



- Modular UQ Library, C++ and Python
- Model-agnostic interfaces
- Numerous UQ algorithms readily available
- `www.mituq.bitbucket.io`

Models in MUQ

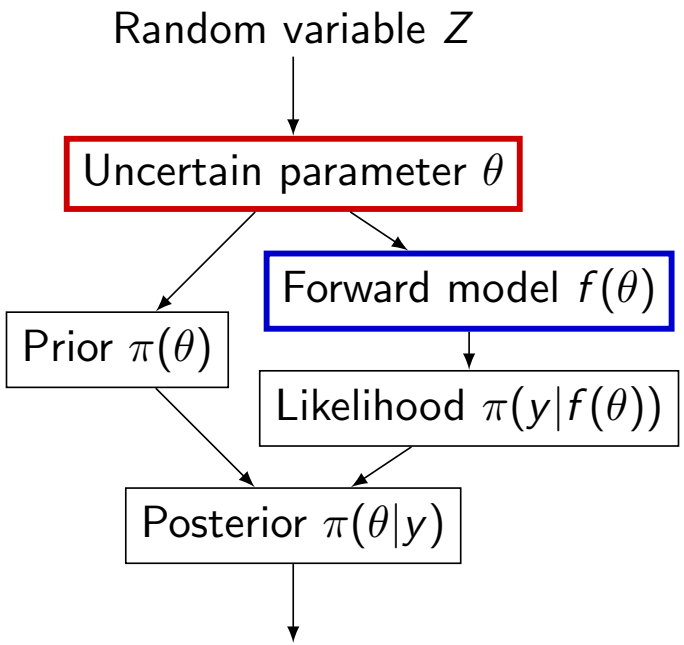


Figure: Model graph example for simple Bayesian problem

- Well-structured, modular construction of advanced models.
- Can couple to external software.
- Easy to switch models / methods!
- For Multilevel (-index): Define set of models.

Modular MCMC Framework

- Proposals: MH, pCN, MALA, DILI, ...
- Kernels: MH, MC, ML/MI, ...
- Chains: Sequential, parallel, ...

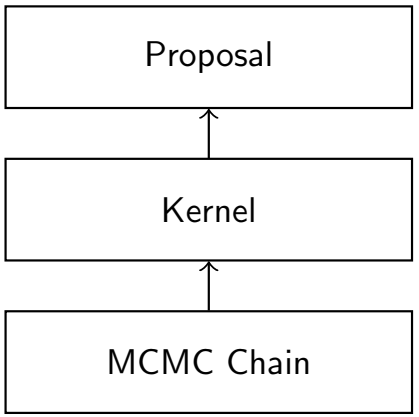


Figure: Modular MCMC architecture

Want a different method? Just switch out one component!

Multilevel (-index) in Modular MCMC Framework

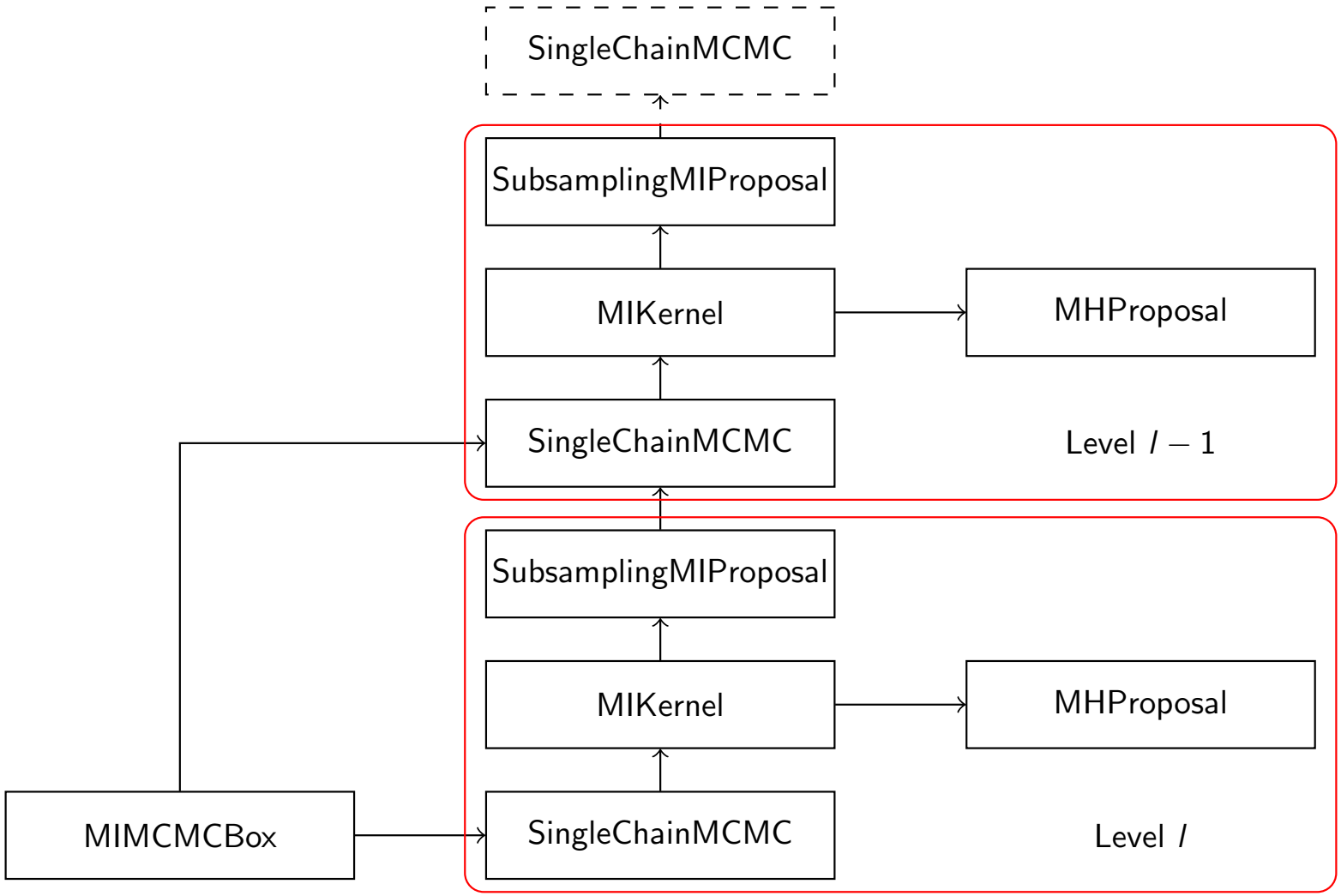
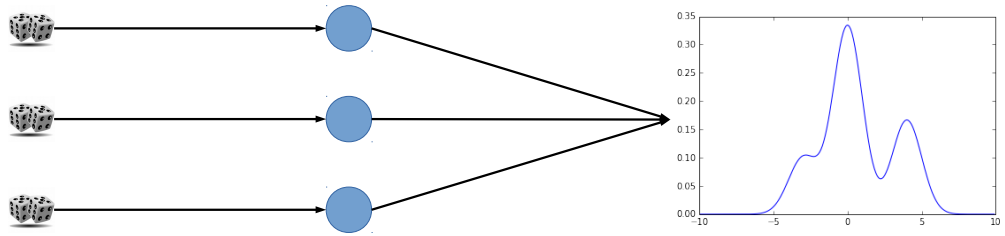


Figure: Sequential architecture for computing $\mathbb{E}_{\nu^l}[Q_l] - \mathbb{E}_{\nu^{l-1}}[Q_{l-1}]$

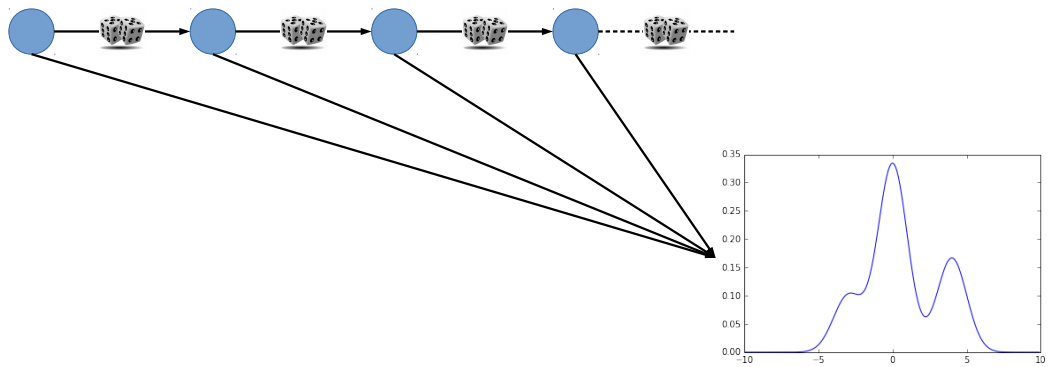
What is the challenge in writing parallel MLMCMC code?

Monte Carlo



Trivially parallel

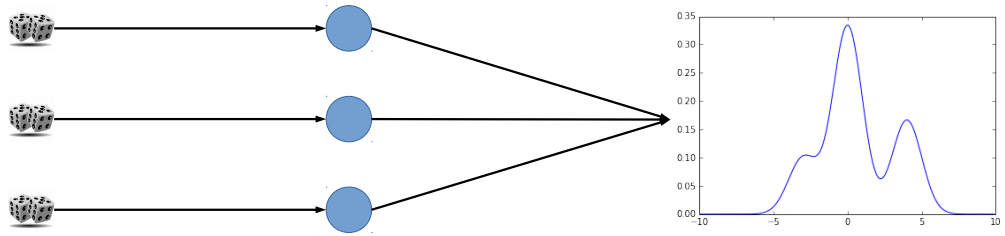
Markov Chain Monte Carlo



Natural data dependencies

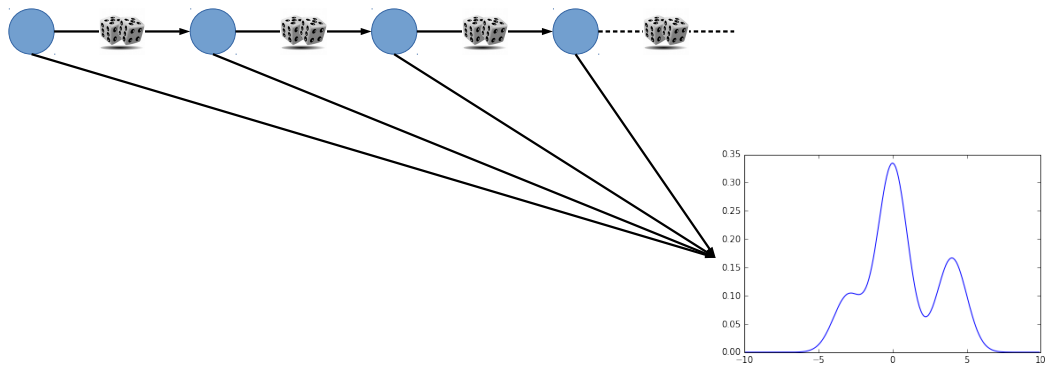
What is the challenge in writing parallel MLMCMC code?

Monte Carlo

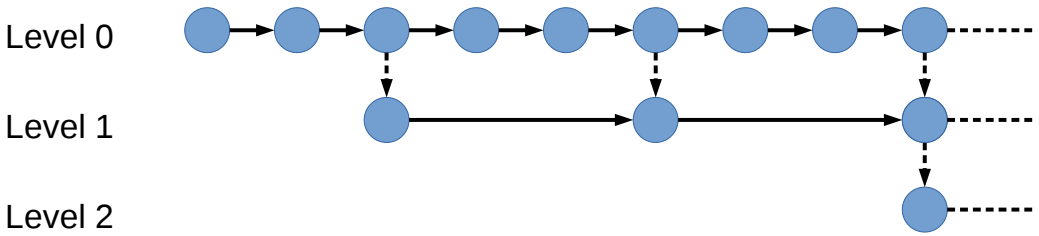


Trivially parallel

Markov Chain Monte Carlo



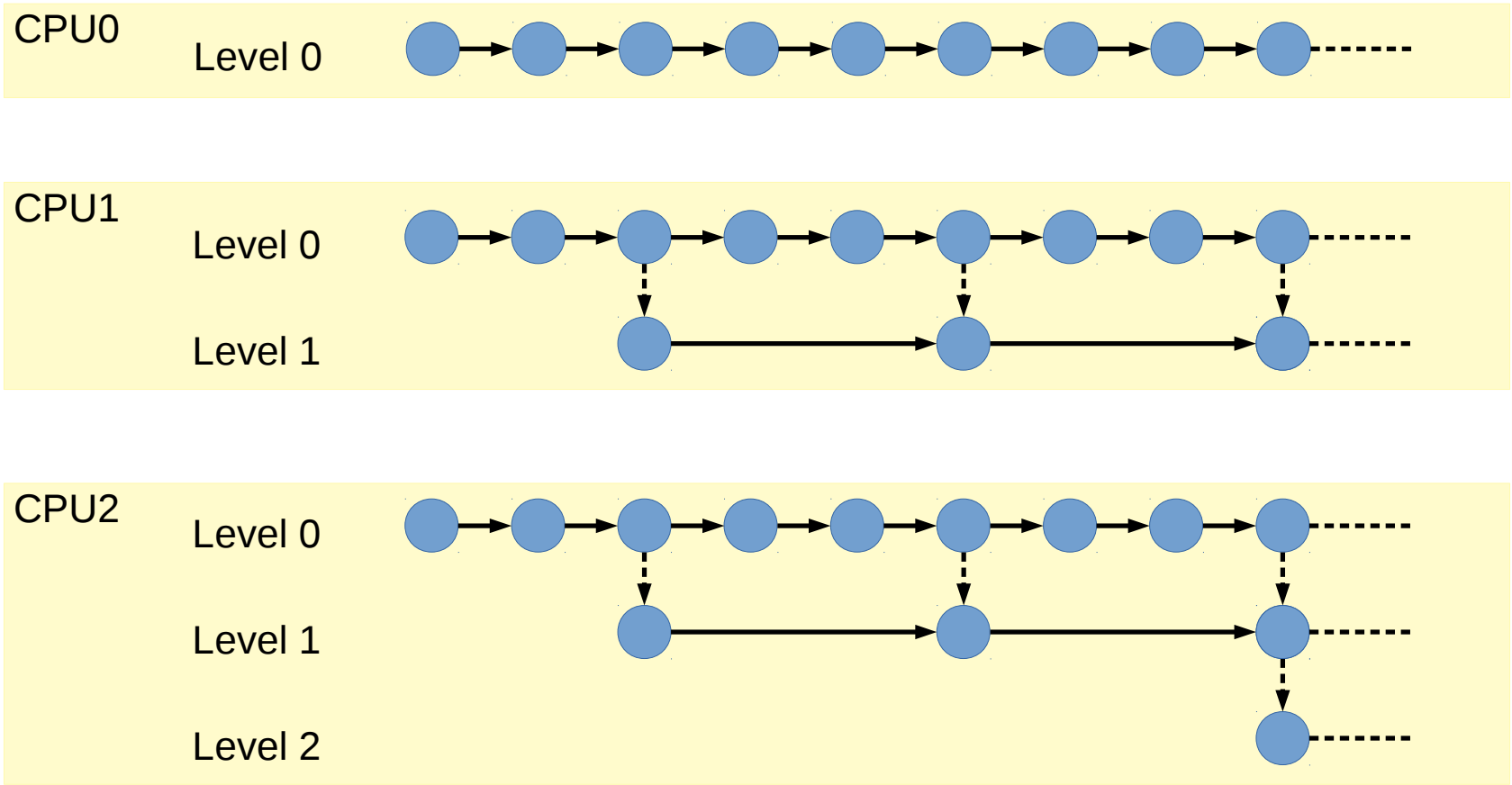
Natural data dependencies



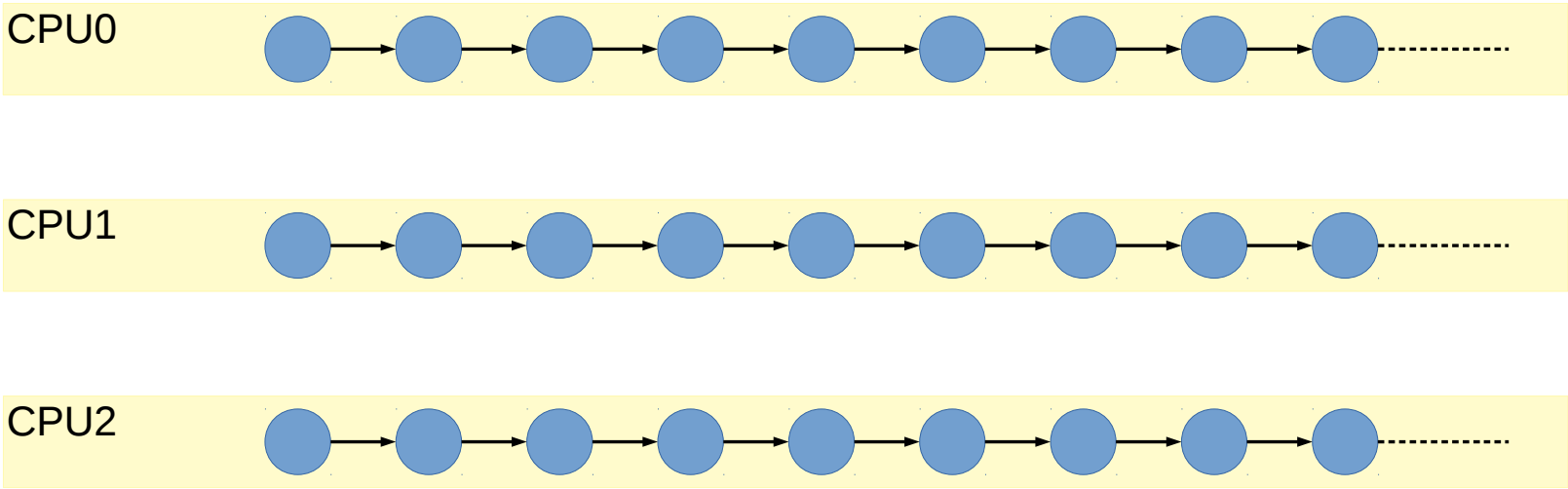
Here **even worse**: Dependencies between levels**!**

MLMCMC Parallelization

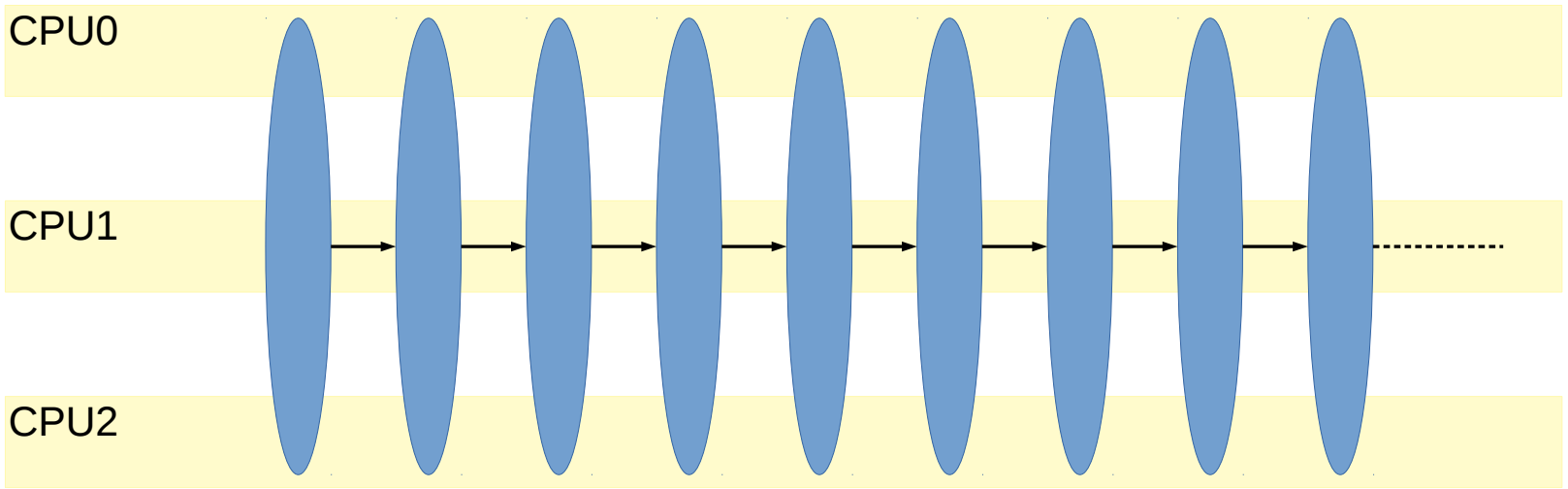
MLMCMC Parallelization: Across Levels



MLMCMC Parallelization: Multiple Chains



MLMCMC Parallelization: Within Model



Parallel Multilevel (-index) MCMC Processor Layout

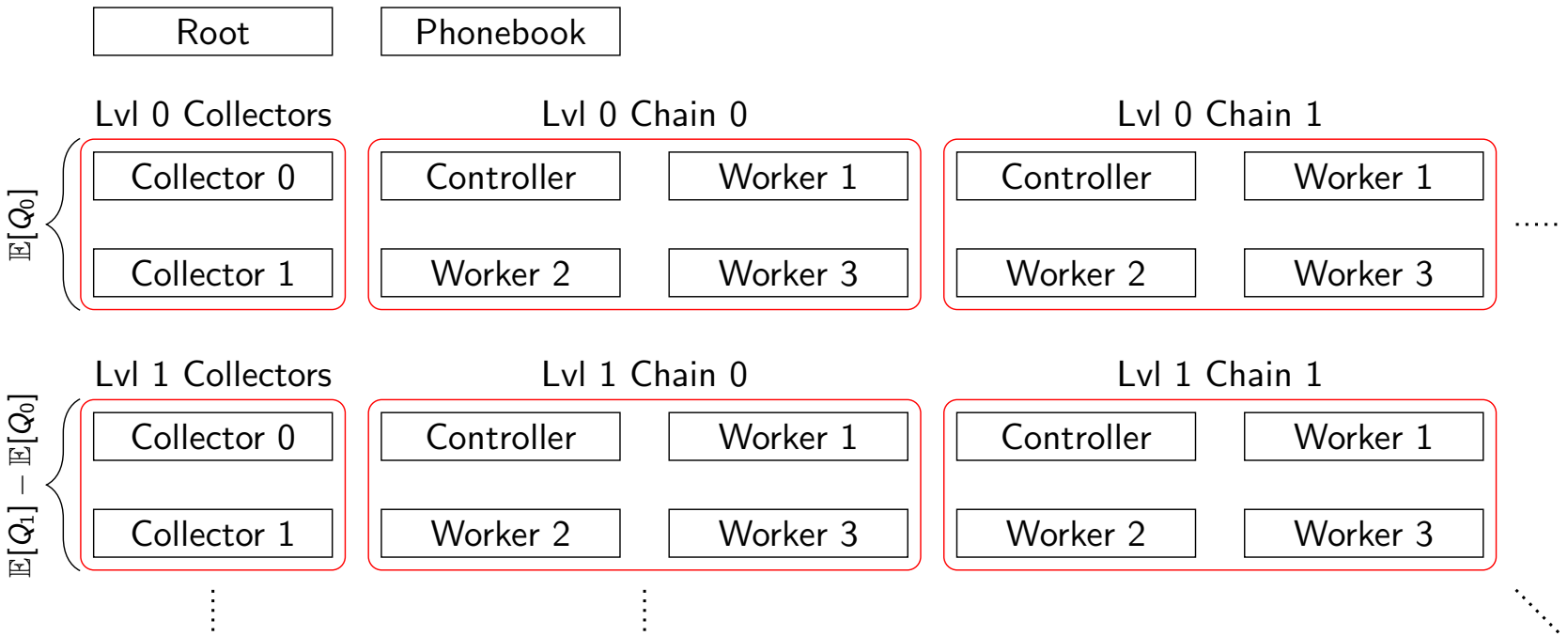
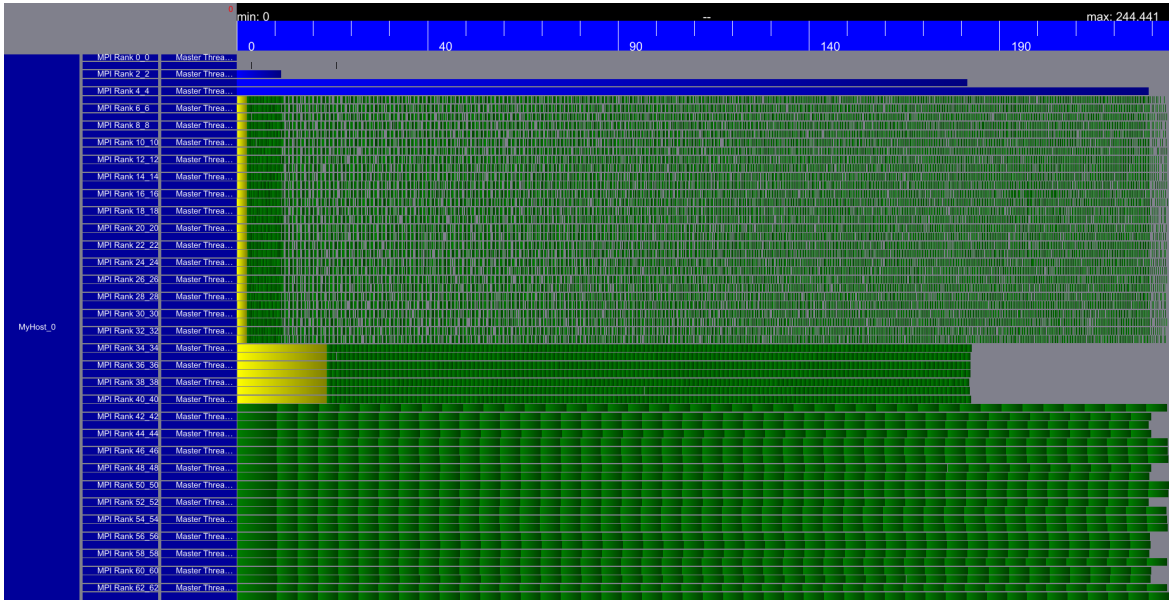


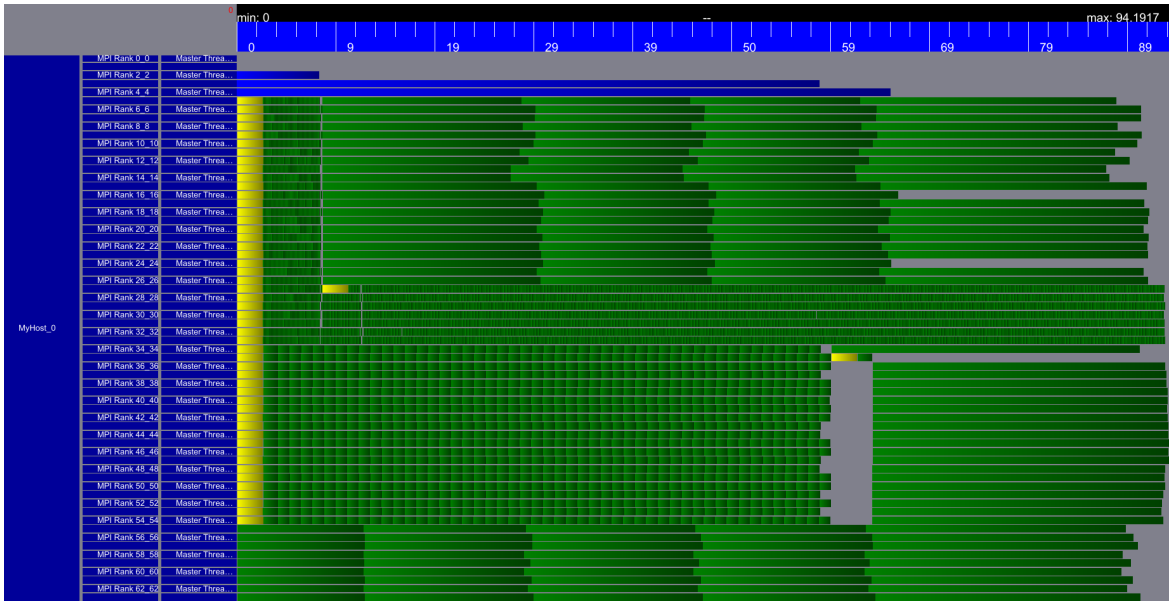
Figure: Parallel layout (each box is a processor / MPI rank)

Tested on 1000 parallel chains, probably more possible
Too complicated? All this happens behind the scenes!

Scheduling Parallel MLMCMC



without scheduling



with scheduling

Parallel Experiments

Poisson problem

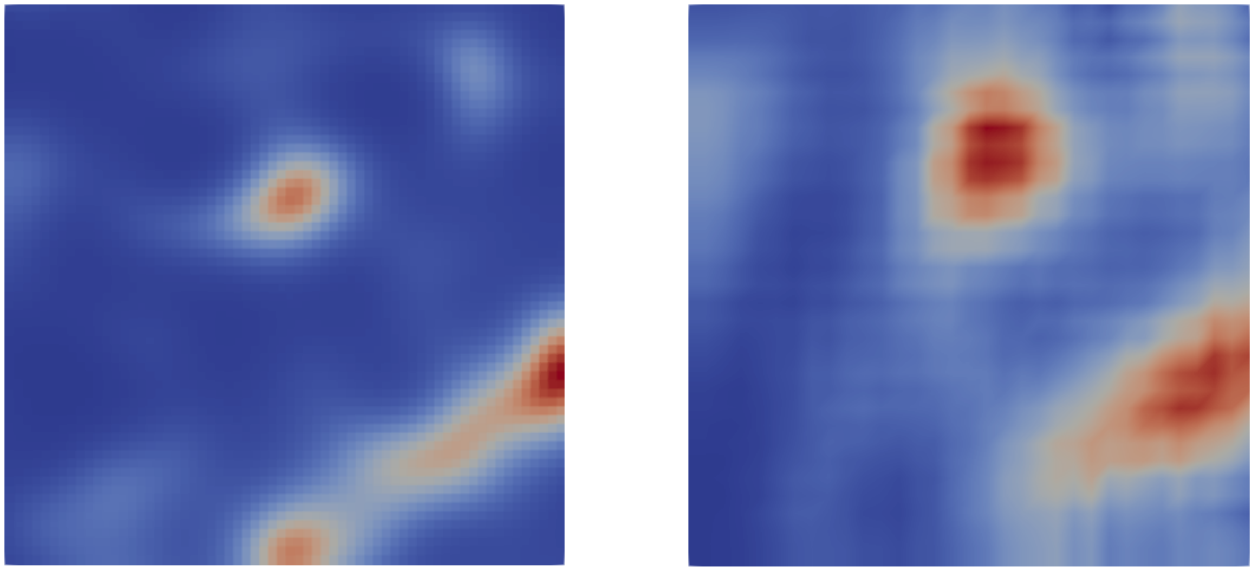


Figure: 'True' field (left). MLMCMC estimator of expected value (right).

level	h_l	DOFs	runtime t_l [ms]	subsamp. ρ_l	i.a.c.t. τ_l	$\mathbb{V}[Q_0]$ or $\mathbb{V}[Q_l - Q_{l-1}]$
0	$\frac{1}{16}$	289	3.35	206	137.3	1.501×10^{-1}
1	$\frac{1}{64}$	4225	45.64	17	11.2	1.121×10^{-3}
2	$\frac{1}{256}$	66049	931.81	0	1.05	4.165×10^{-5}

Strong scaling

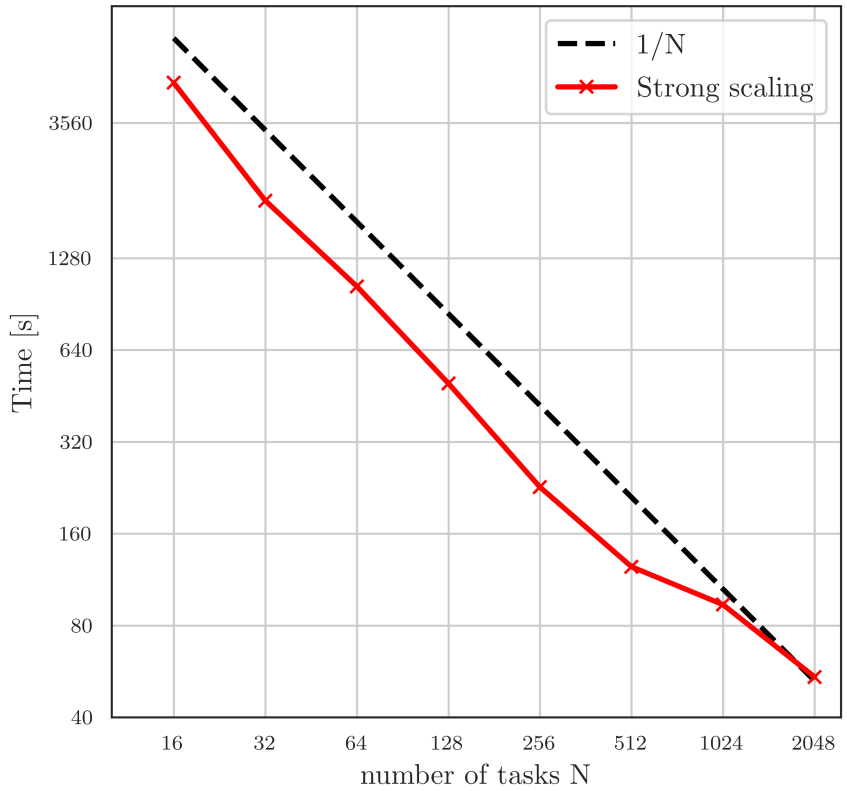


Figure: Scalability of the Poisson model problem for 10^4, 10^3 and 10^2 samples and subsampling rates of 206, 17 and 0 on levels 0, 1 and 2. The model dimensionality remains constant as the number of processors is increased.

Weak scaling

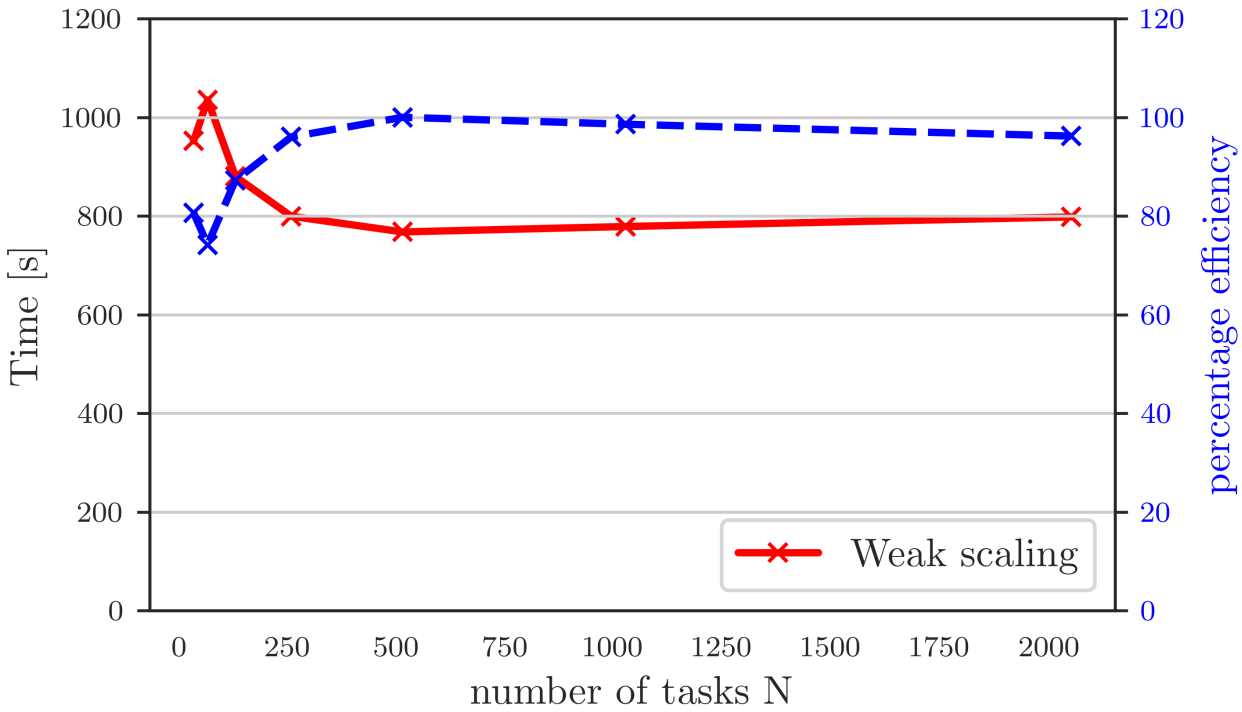
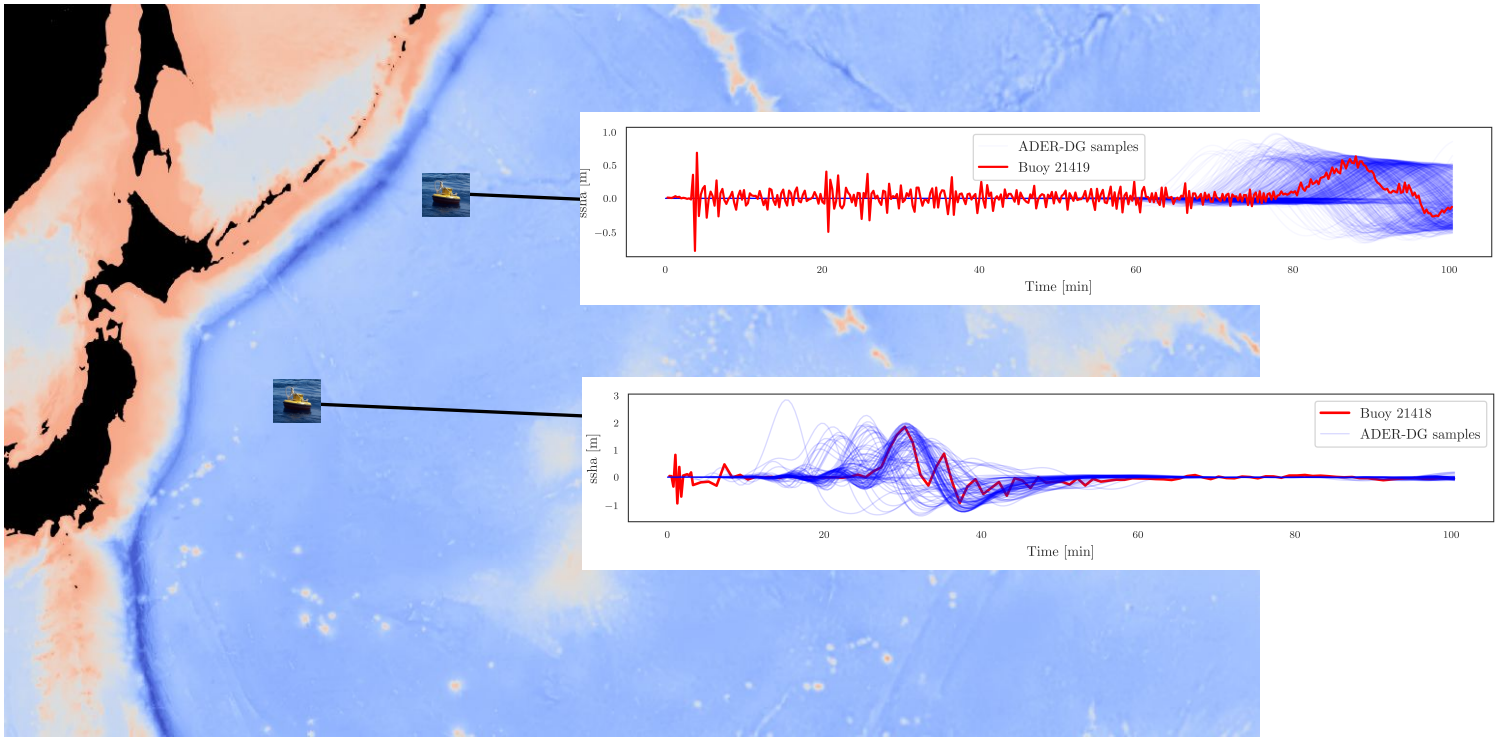


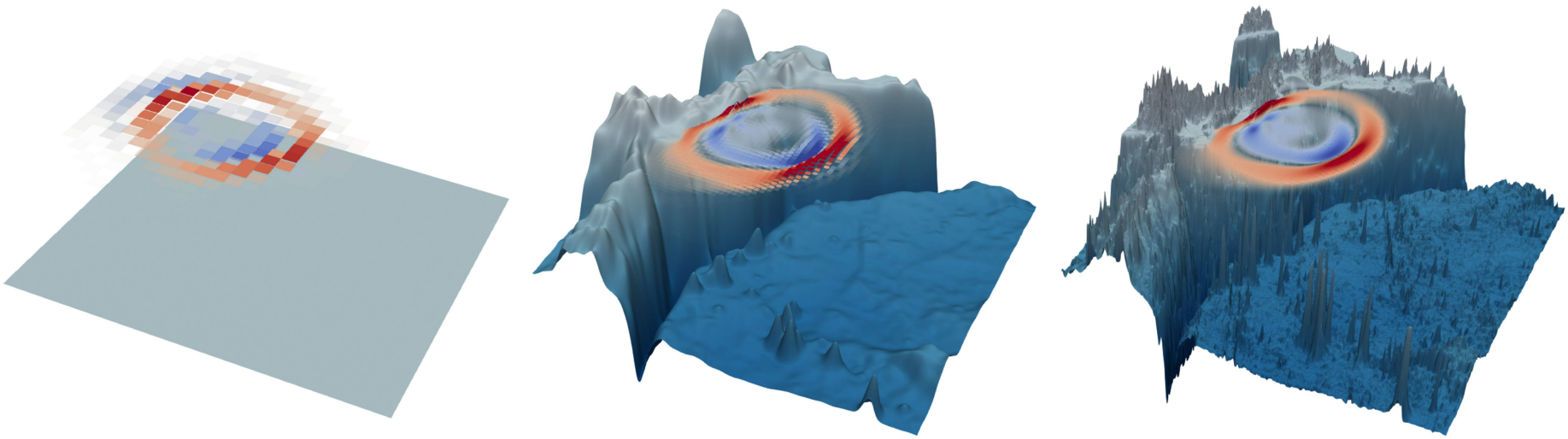
Figure: Weak scalability & parallel efficiency for Poisson problem. At 64 cores 10^4, 10^3 and 10^2 samples are computed on levels 0, 1 and 2, resp. Number of samples increased linearly with number of processors.

Tsunami Application



- Modelling Tohoku event (2011) using Shallow Water Equation and real bathymetry
- Forward model using ExaHyPE PDE engine [Reinarz et al, CPC '20]
- Data: Buoy measurements. Parameter: Tsunami source

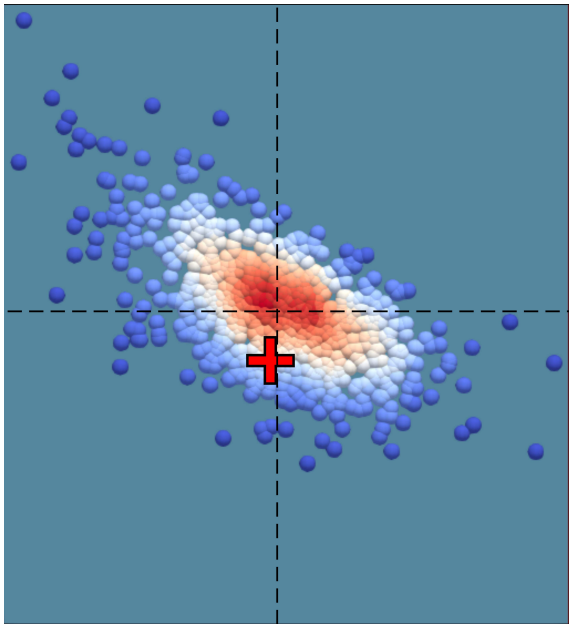
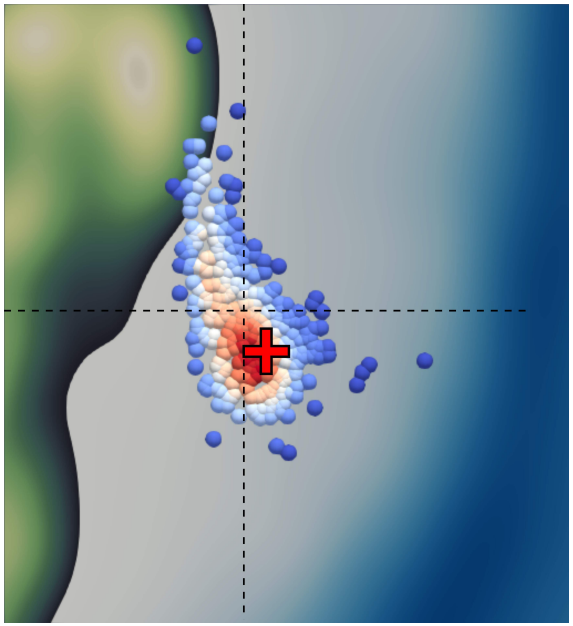
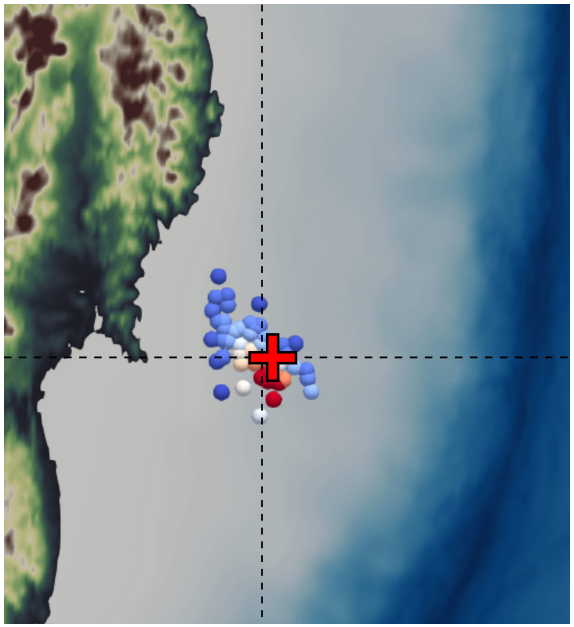
Model hierarchy



Across levels, we adapt

- mesh size
- bathymetry smoothness (specific to hyperbolic solvers!)

Results

level	t_l [s]	subsam. ρ_l	$\mathbb{V}[Q_0]$ or $\mathbb{V}[Q_l - Q_{l-1}]$		$\sum_{k=1}^{l} \mathbb{E}[Q_k - Q_{k-1}]$	$\mathbb{E}[Q_0]+$
0	7.38	25	1984.09	1337.42	3.61	27.96
1	97.3	5	1592.17	1523.18	-12.29	23.39
2	438.1	0	340.56	938.53	-5.46	0.12

Run on 72 nodes, each with 48 cores (3456 in total)

Conclusions

- Introduced **multilevel Monte Carlo** & **model hierarchies** (in general) for UQ and Baysian inference. Highlighted the **huge potential**.
- A vibrant research area with **many open questions**
- A **"no-brainer"** (for me) in practice (if you have a hierarchy)
- I believe, we have only **scratched the surface**, especially in context of **Bayesian inference** & **data assimilation**

Conclusions

- Introduced **multilevel Monte Carlo** & **model hierarchies** (in general) for UQ and Baysian inference. Highlighted the **huge potential**.
- A vibrant research area with **many open questions**
- A **"no-brainer"** (for me) in practice (if you have a hierarchy)
- I believe, we have only **scratched the surface**, especially in context of **Bayesian inference** & **data assimilation**
- Despite data dependencies, Multilevel MCMC ready for HPC
  extra room for parallelism across levels
- Building on modular framework in **MUQ** pays off.
- **Flexibility** regarding model hierarchy.
- **Excellent parallel scalability** and embedding with existing HPC software from applications.

Outlook & Hardware-Awareness

- Application in **other areas** (especially multilevel MCMC):
  (e.g. aerospace composites, geostatistics, imaging, quantum physics,. . .)
- Significant further improvements are possible with using **adaptive, sample-dependent hierarchies**
- Have to yet fully exploit **multi-index** capability.
- Parallel adaptive ML delayed acceptance (w. Dodwell & Lykkegaard)

Outlook & Hardware-Awareness

- Application in **other areas** (especially multilevel MCMC):
  (e.g. aerospace composites, geostatistics, imaging, quantum physics,. . .)
- Significant further improvements are possible with using **adaptive, sample-dependent hierarchies**
- Have to yet fully exploit **multi-index** capability.
- Parallel adaptive ML delayed acceptance (w. Dodwell & Lykkegaard)
- **Hierarchical task-based** programming models to distribute work at run-time. Static load balancing difficult due to adaptivity, error control, . . .
- **Matrix-free** solvers: sum factorisation (high order) & **SIMD** parallelisation across samples (low order)
- **More sophisticated propsals** have other parallelisation challenges, e.g. MAP point calculation = optimisation

Roland Herzog

References & Resources

- J Christen & C Fox, MCMC using an approximation, *J Comp Graph Stat* **14**, 2005
- KA Cliffe, MB Giles, R**RS** & AL Teckentrup, Multilevel Monte Carlo methods and applications to elliptic PDEs with random coefficients Comput Visual Sci **14**, 2011
- VH Hoang, C Schwab & AM Stuart, Complexity analysis of accelerated MCMC methods for Bayesian inversion, *Inverse Prob.* **29**, 2013
- TJ Dodwell, C Ketelsen, **RS** & AL Teckentrup, A hierarchical multilevel Markov chain Monte Carlo algorithm with applications to uncertainty quantification in subsurface flow, *SIAM/ASA J Uncertain Q* **3**, 2015 (also *SIAM Review* **61**, 2019)
- MB Lykkegaard, G Mingas, **RS**, C Fox & TJ Dodwell, Multilevel Delayed Acceptance MCMC with an Adaptive Error Model in PyMC3, *NeurIPS 2020*
- T Cui, G Detommaso & **RS**, Multilevel dimension-independent likelihood-Informed MCMC for large-scale inverse problems, submitted [arXiv:1910.12431]
- L Seelinger, A Reinarz, L Rannabauer, M Bader, P Bastian, **RS**, High Performance UQ with Parallelized Multilevel MCMC, to appear in **SC'21** [arXiv:2107.14552]

References & Resources

- J Christen & C Fox, MCMC using an approximation, *J Comp Graph Stat* **14**, 2005
- KA Cliffe, MB Giles, R**RS** & AL Teckentrup, Multilevel Monte Carlo methods and applications to elliptic PDEs with random coefficients Comput Visual Sci **14**, 2011
- VH Hoang, C Schwab & AM Stuart, Complexity analysis of accelerated MCMC methods for Bayesian inversion, *Inverse Prob.* **29**, 2013
- TJ Dodwell, C Ketelsen, **RS** & AL Teckentrup, A hierarchical multilevel Markov chain Monte Carlo algorithm with applications to uncertainty quantification in subsurface flow, *SIAM/ASA J Uncertain Q* **3**, 2015 (also *SIAM Review* **61**, 2019)
- MB Lykkegaard, G Mingas, **RS**, C Fox & TJ Dodwell, Multilevel Delayed Acceptance MCMC with an Adaptive Error Model in PyMC3, *NeurIPS 2020*
- T Cui, G Detommaso & **RS**, Multilevel dimension-independent likelihood-Informed MCMC for large-scale inverse problems, submitted [arXiv:1910.12431]
- L Seelinger, A Reinarz, L Rannabauer, M Bader, P Bastian, **RS**, High Performance UQ with Parallelized Multilevel MCMC, to appear in **SC'21** [arXiv:2107.14552]

Resources

MUQ

`www.mituq.bitbucket.io`

DUNE

`www.dune-project.org`

ExaHyPE

`www.exahype.eu`