Multiindex and Multilevel Markov Chain Monte Carlo in MUQ2

Peter Bastian, Ole Klein, Robert Scheichl, *Linus Seelinger*

Heidelberg University

February 27, 2019

Outline

- Reminder: Inverse Problems
- Markov Chain Monte Carlo

- Multilevel/index MCMC
- Implementation
- Application example

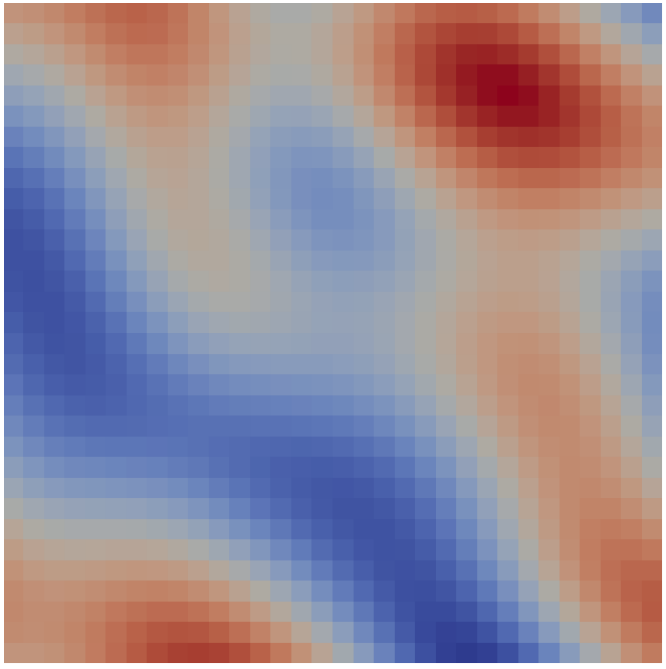
- Parallelization

Section 1

Inverse Problems

Exemplary Inverse Problem

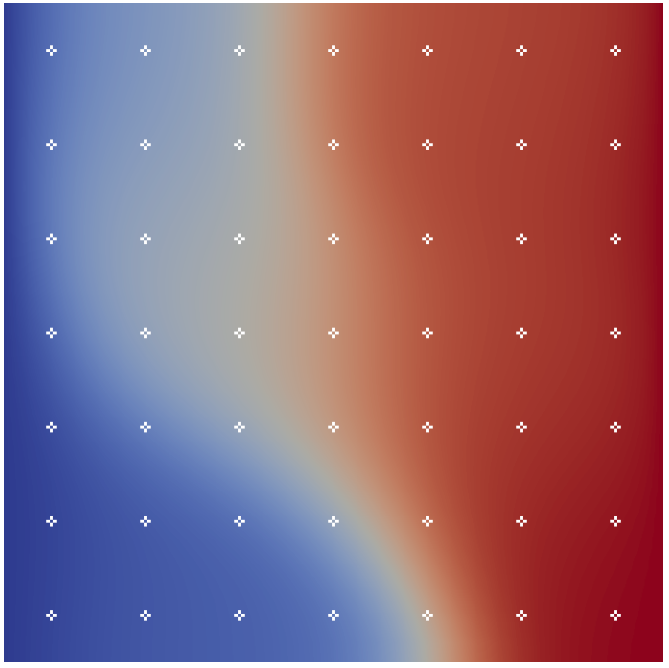
Parameters



$\longrightarrow$
PDE - easy

$\longleftarrow$
hm...

Measurements



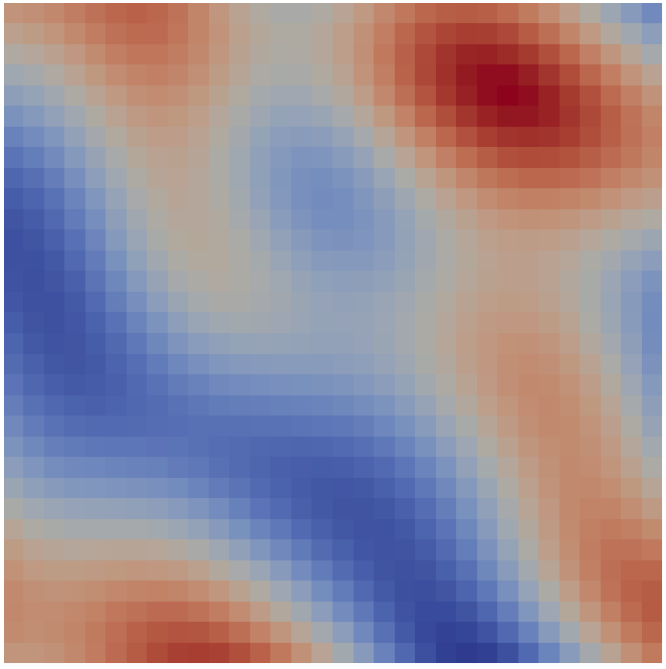
$\rightarrow$ Bayes:

$$f_{P|Z} \propto f_P \cdot f_{Z|P}$$

... how to sample?

Exemplary Inverse Problem

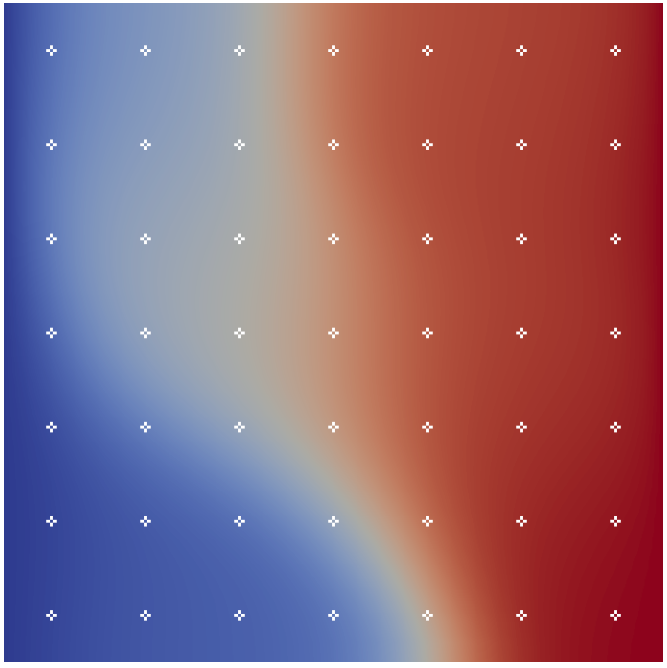
Parameters



$\rightarrow$
PDE - easy

$\leftarrow$
hm...

Measurements



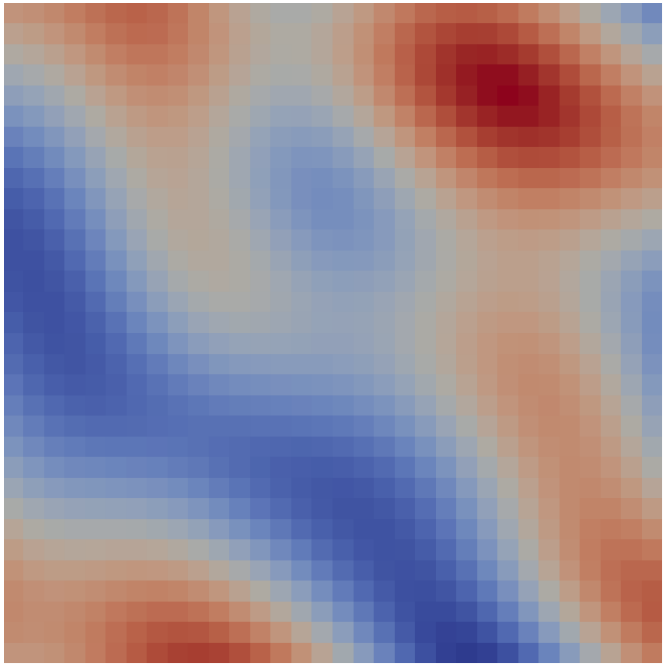
$\rightarrow$ Bayes:

$$f_{P|Z} \propto f_P \cdot f_{Z|P}$$

... how to sample?

Exemplary Inverse Problem

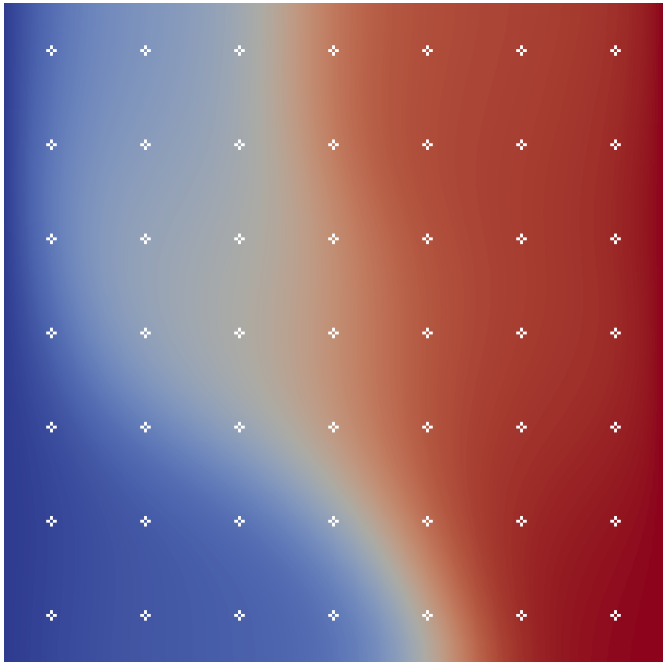
Parameters



$\rightarrow$
PDE - easy

$\leftarrow$
hm...

Measurements



$\rightarrow$ Bayes:

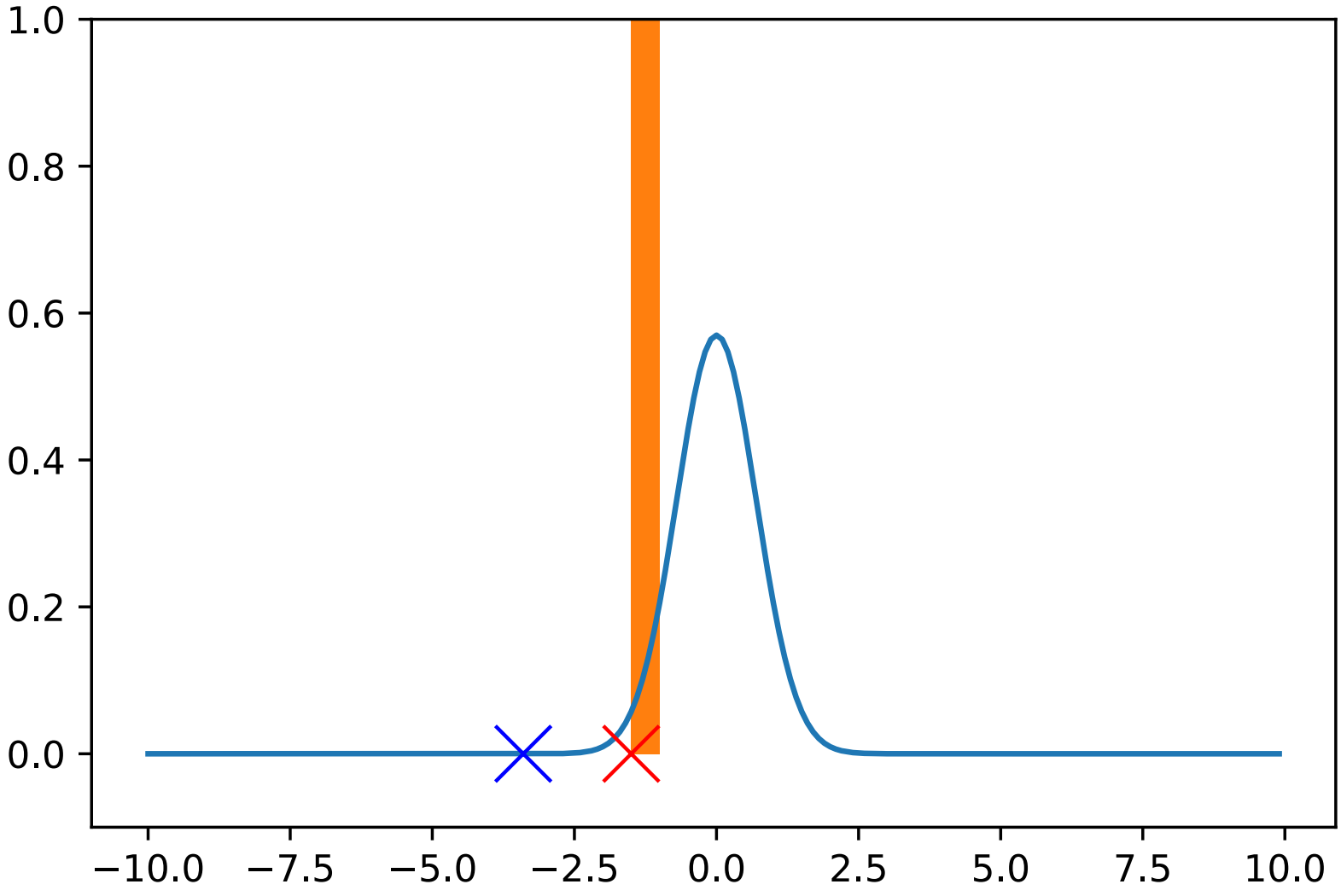
$$f_{P|Z} \propto f_P \cdot f_{Z|P}$$

... how to sample?

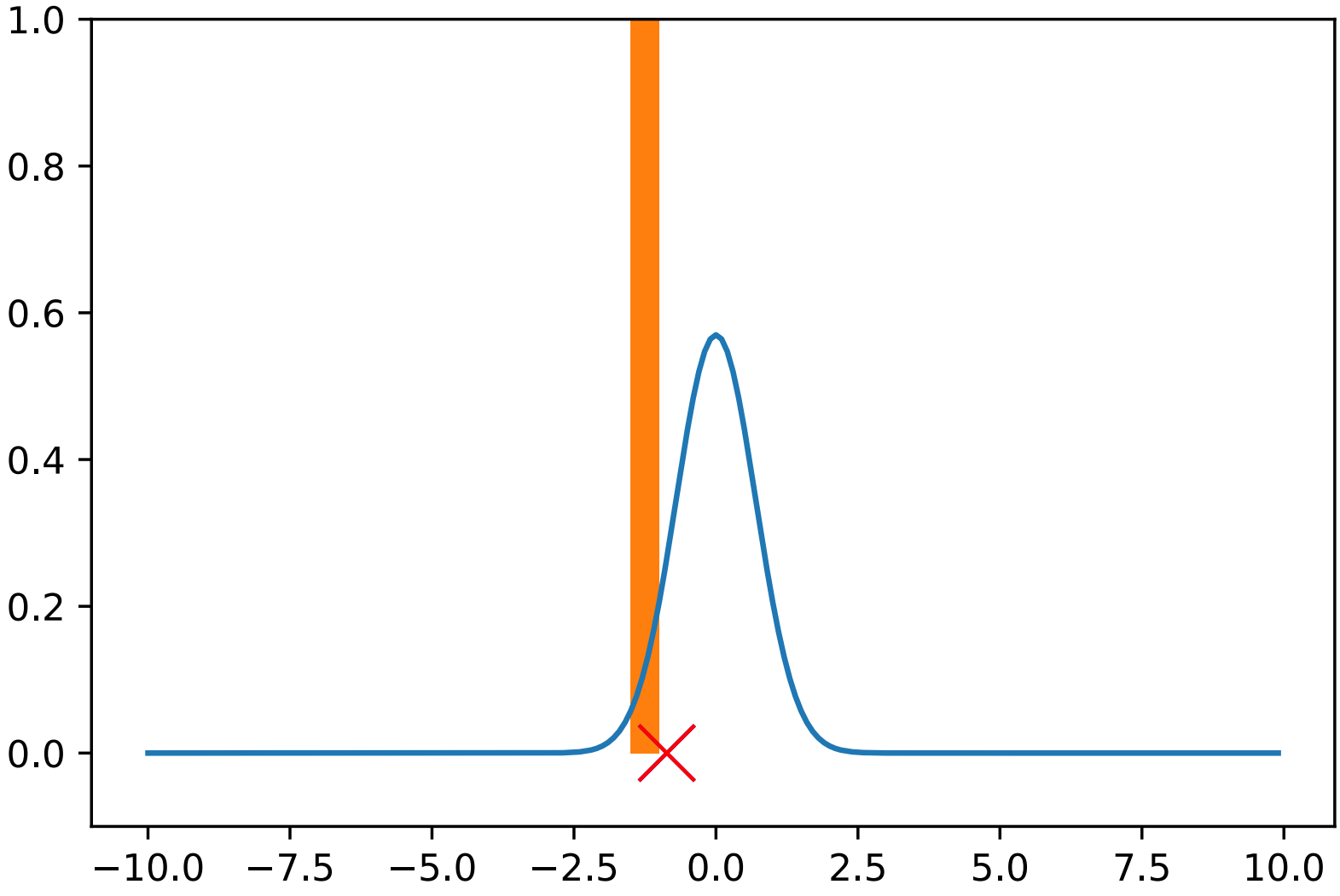
Section 2

Markov Chain Monte Carlo

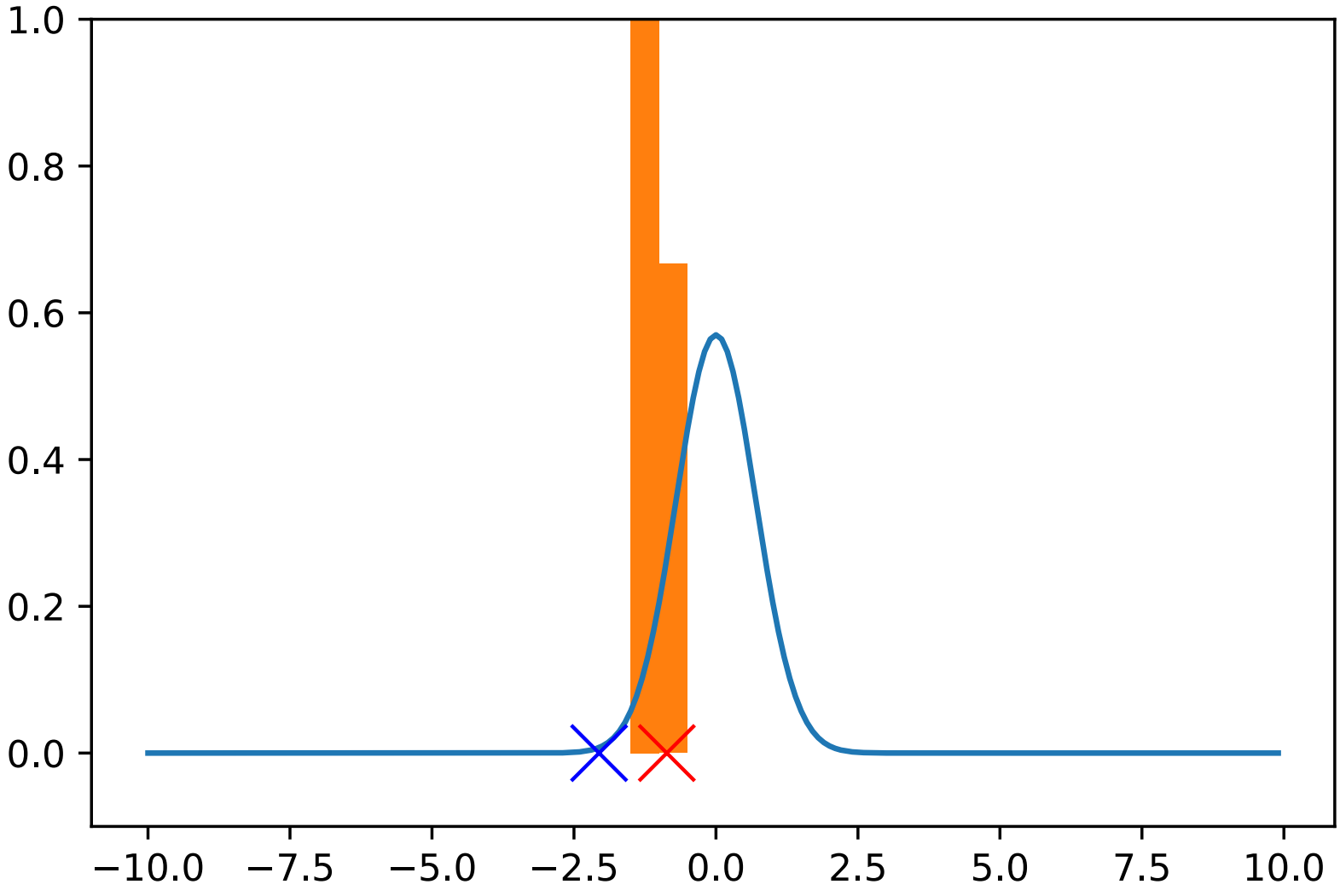
MCMC Example (Sample 1)



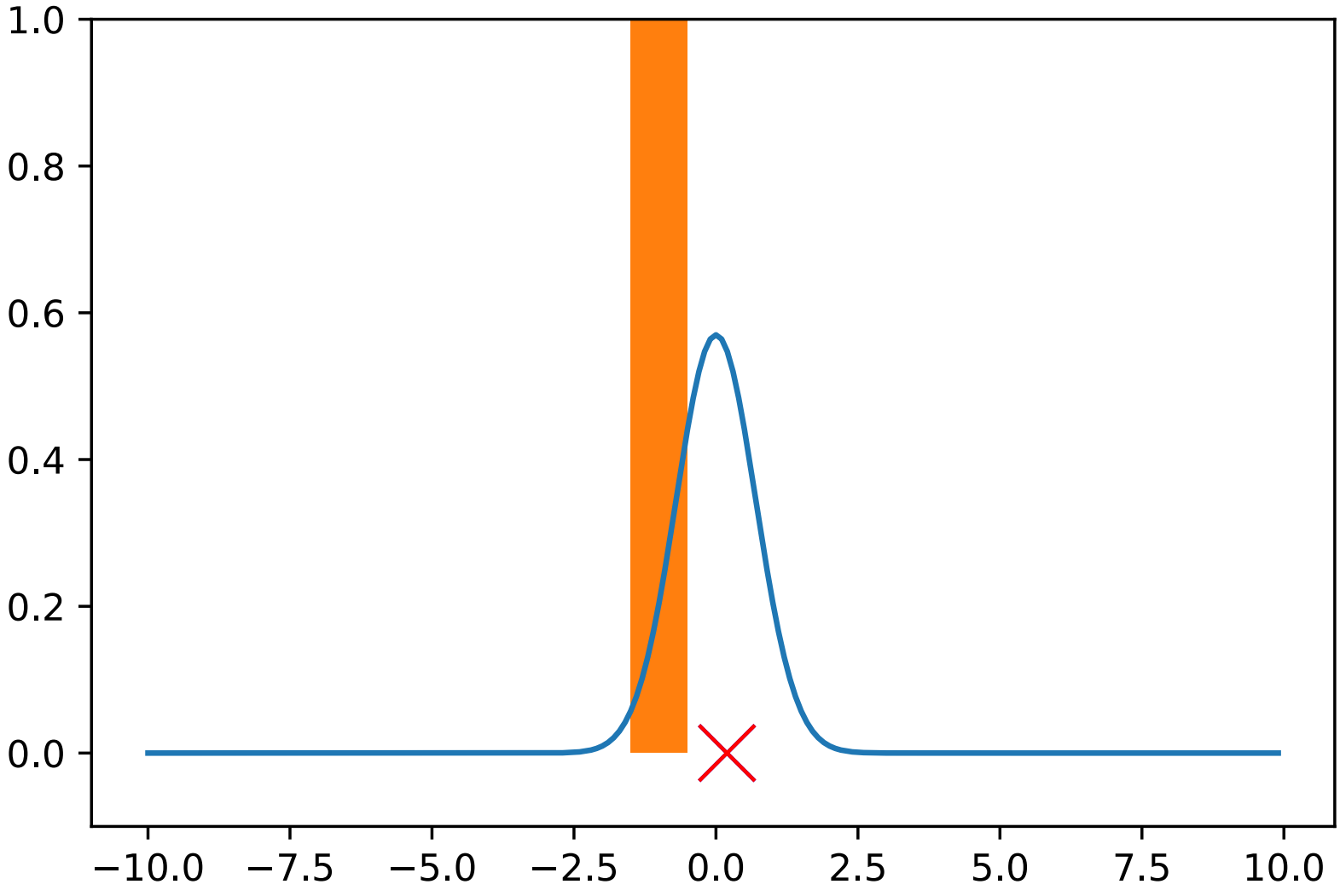
MCMC Example (Sample 2)



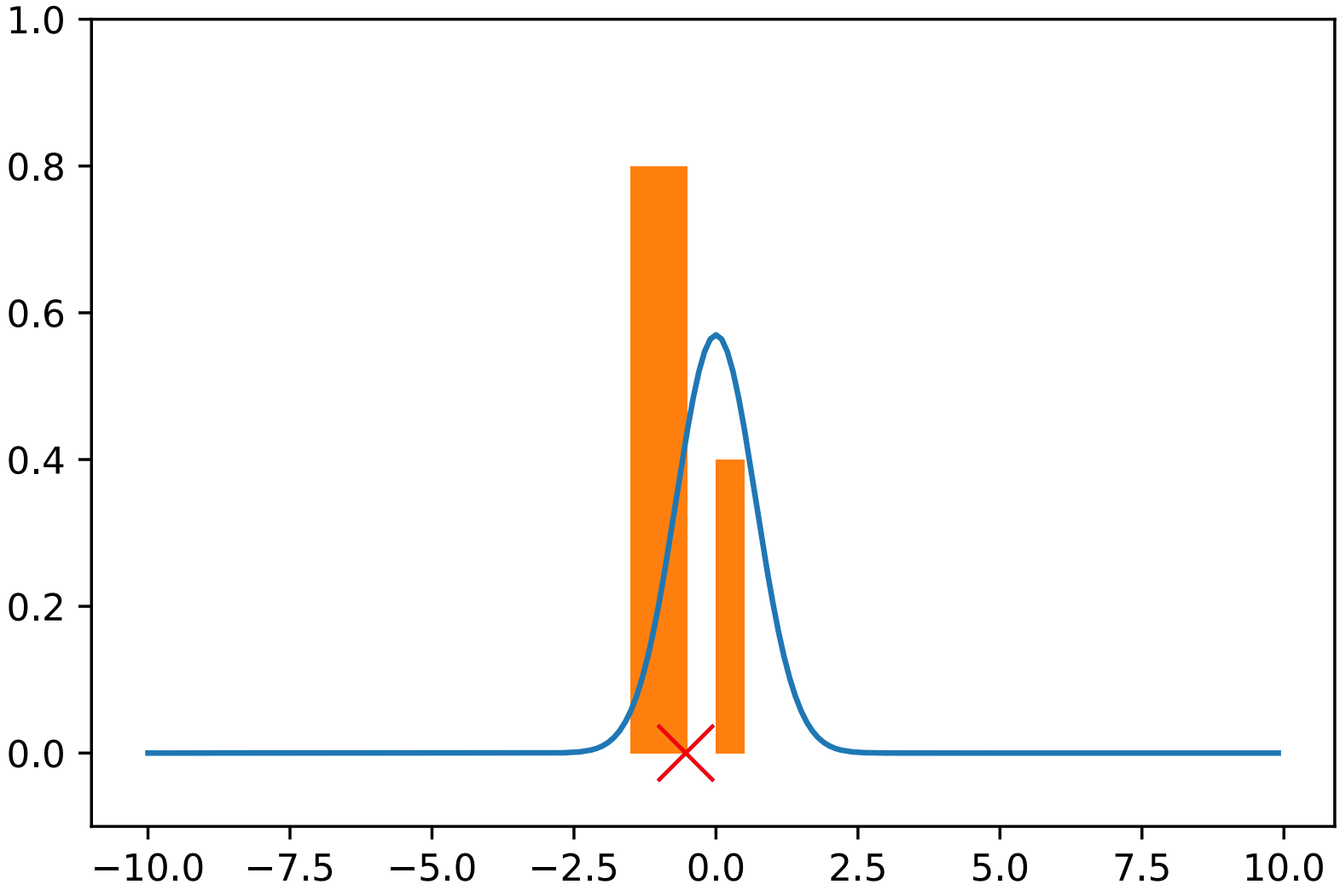
MCMC Example (Sample 3)



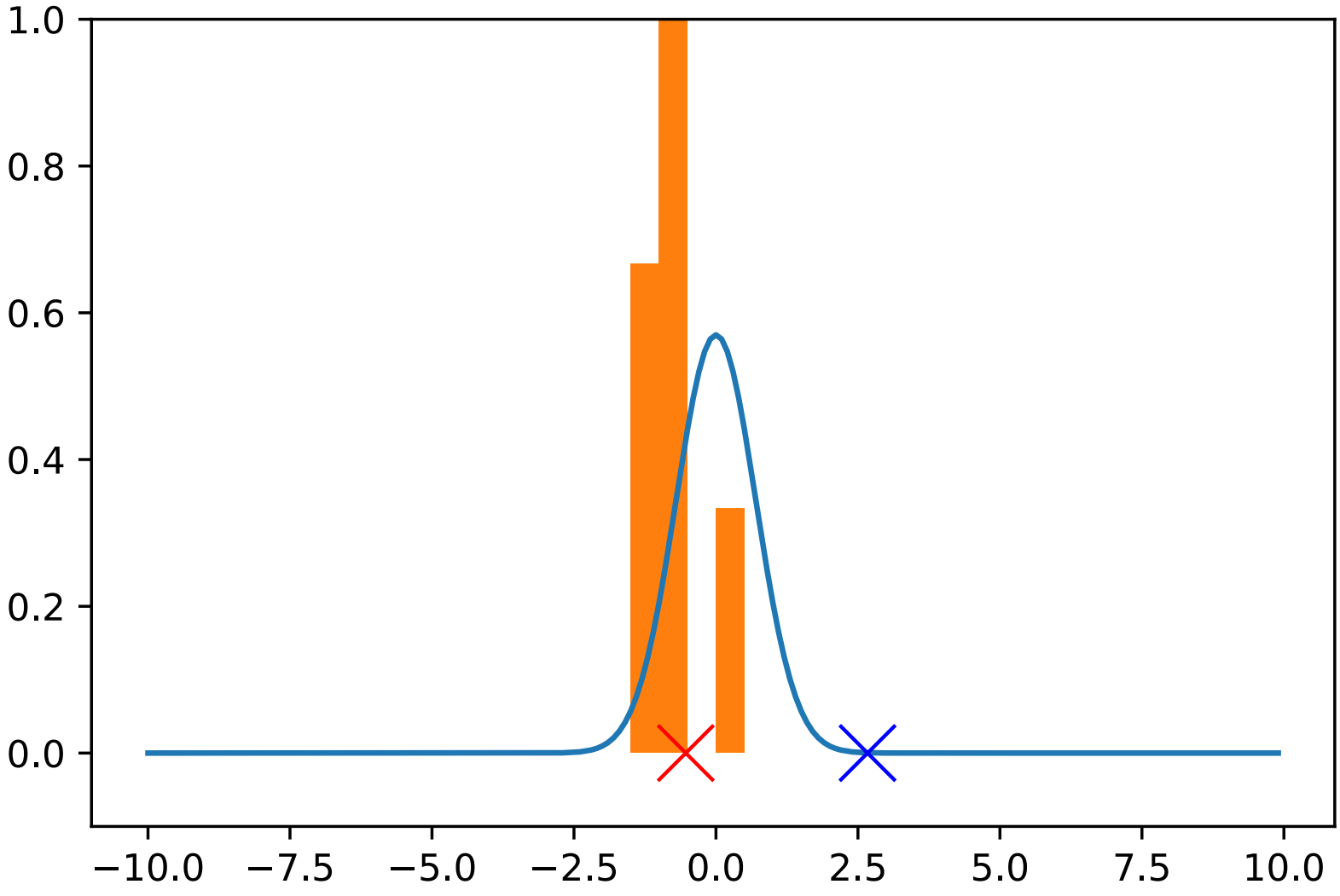
MCMC Example (Sample 4)



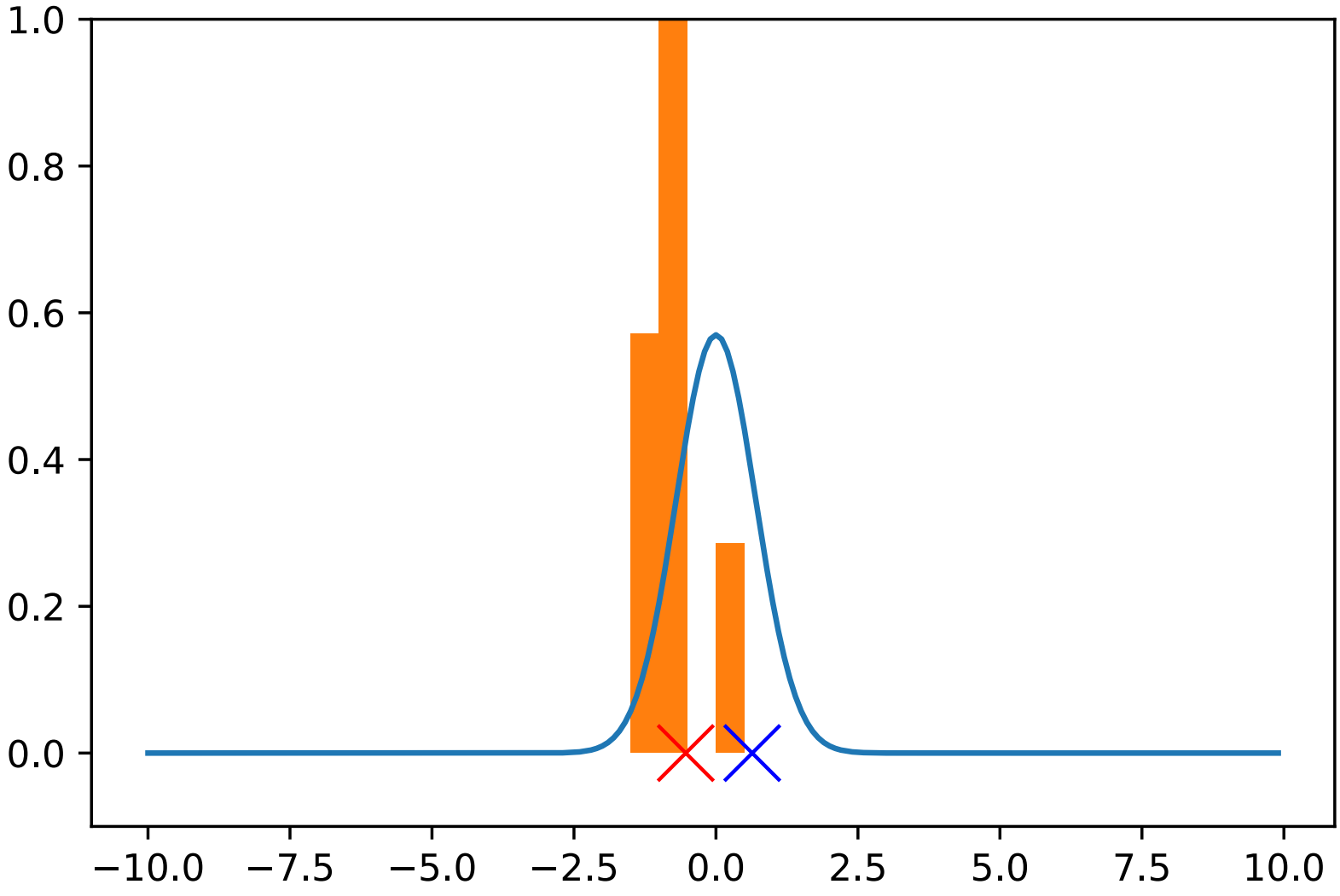
MCMC Example (Sample 5)



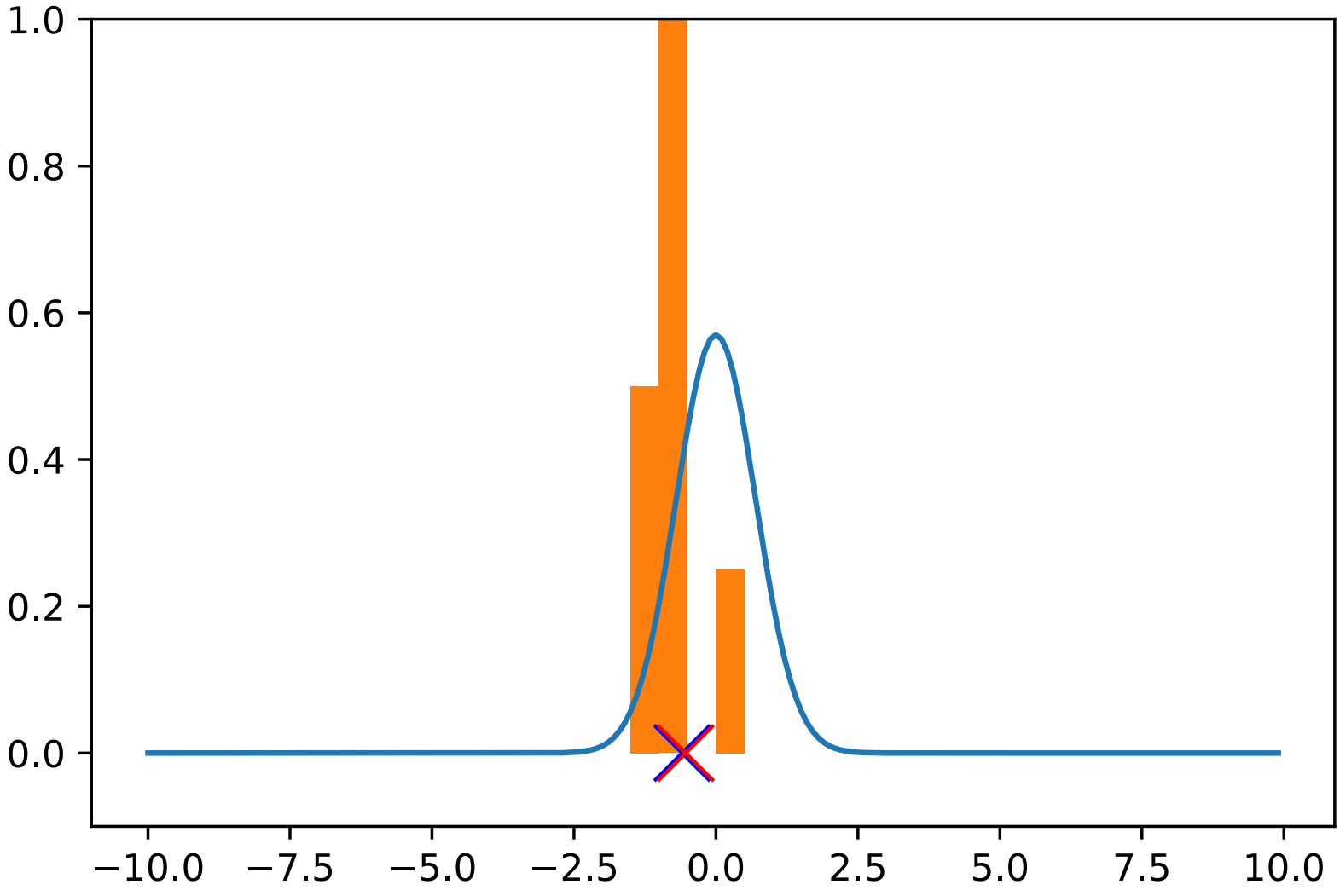
MCMC Example (Sample 6)



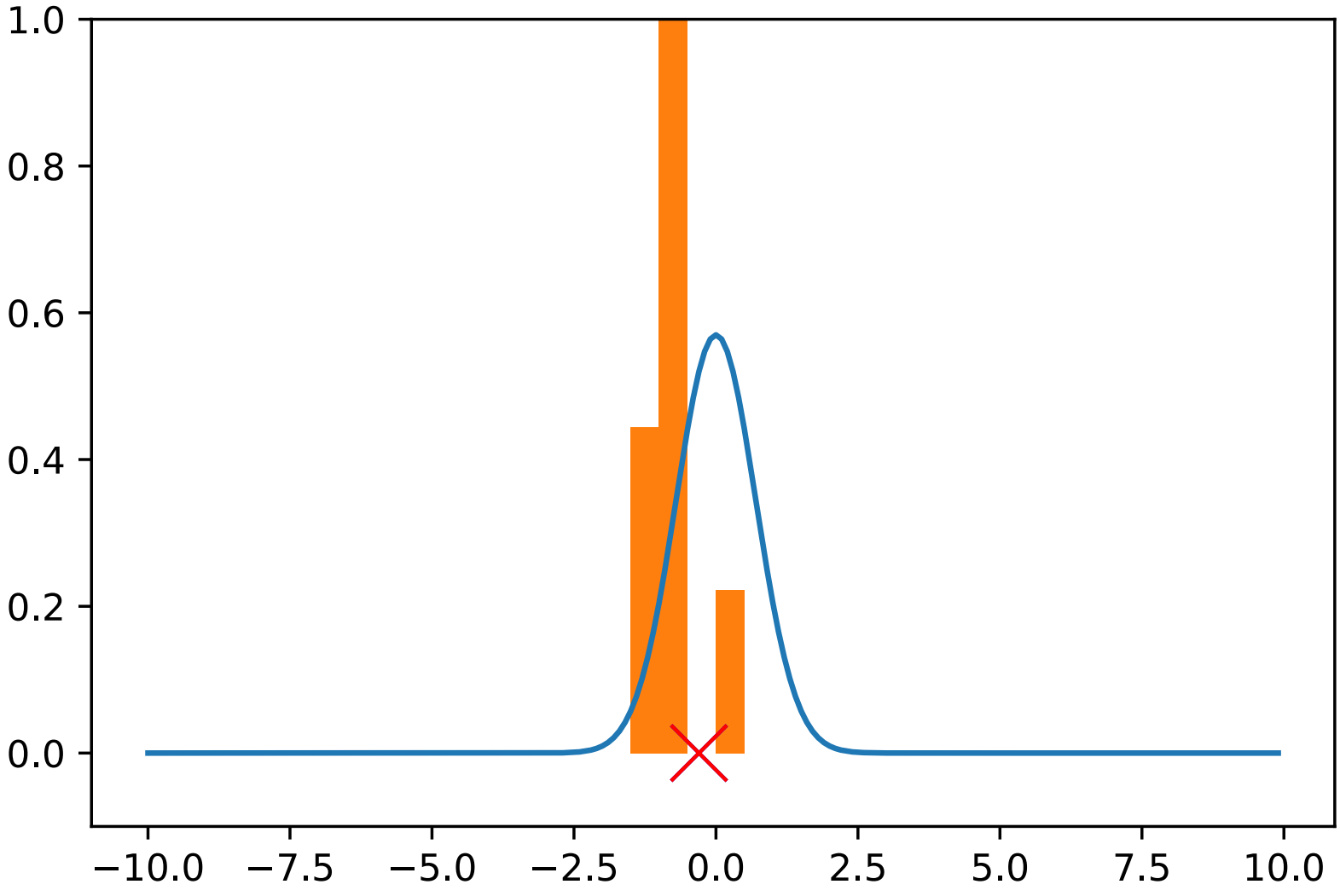
MCMC Example (Sample 7)



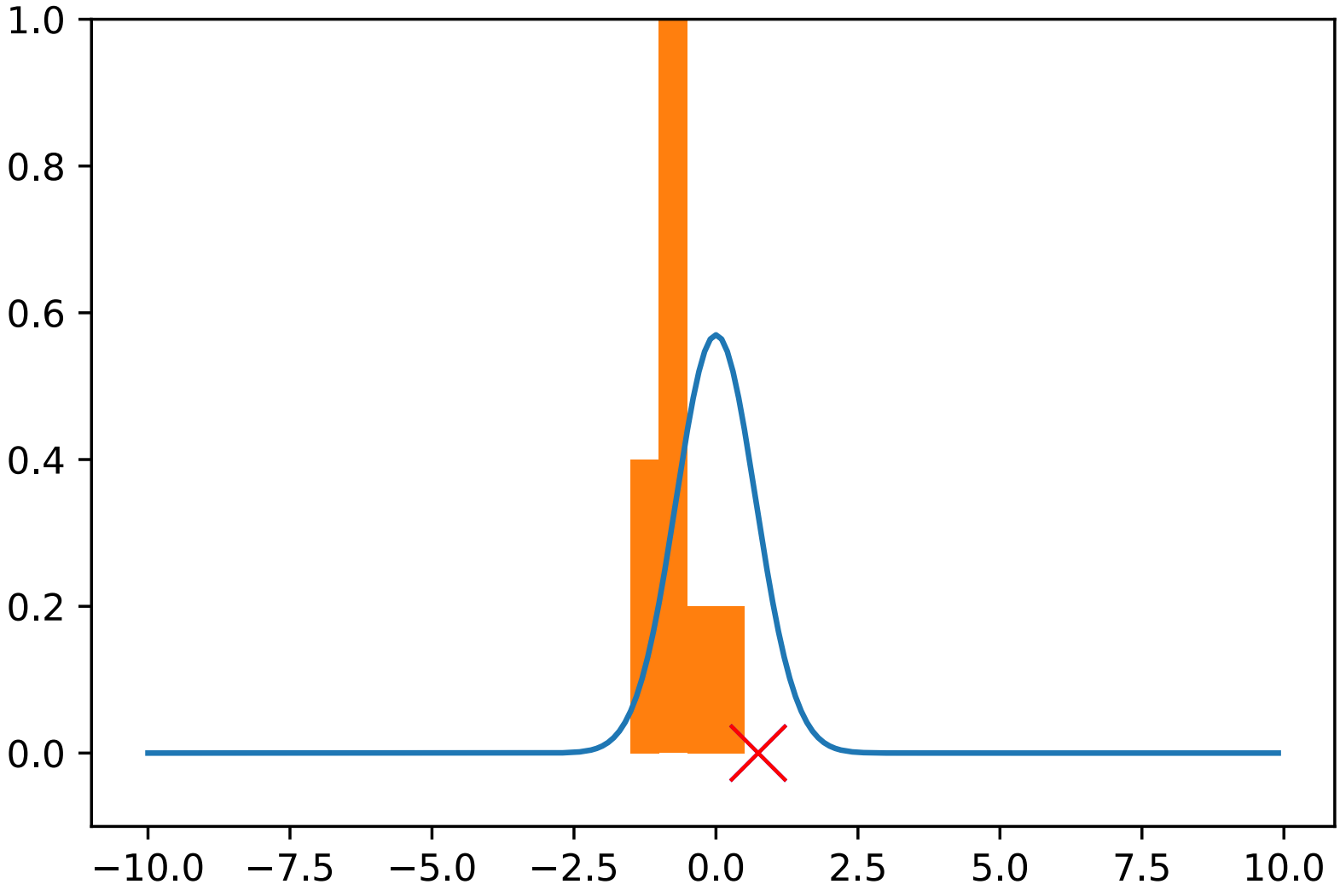
MCMC Example (Sample 8)



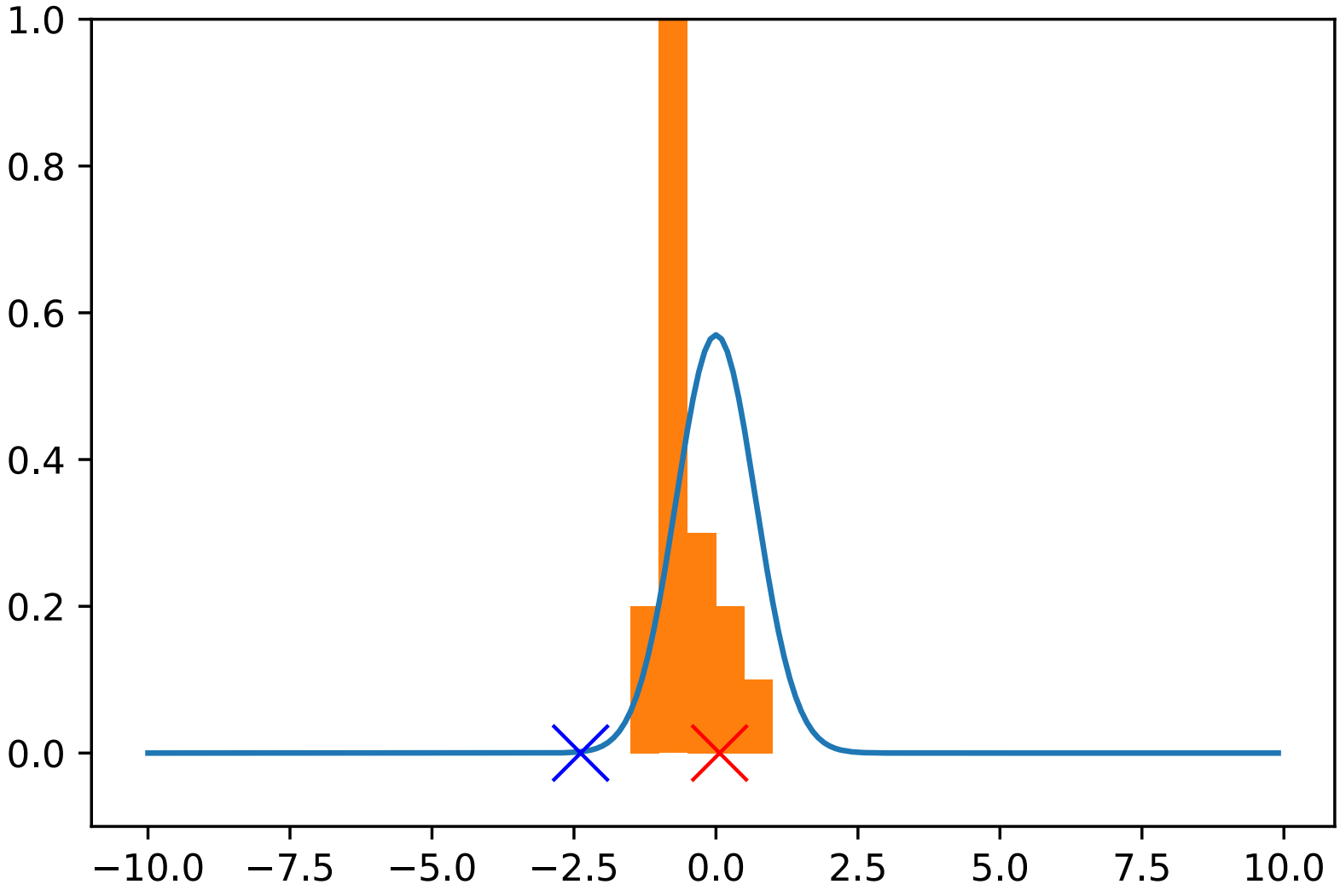
MCMC Example (Sample 9)



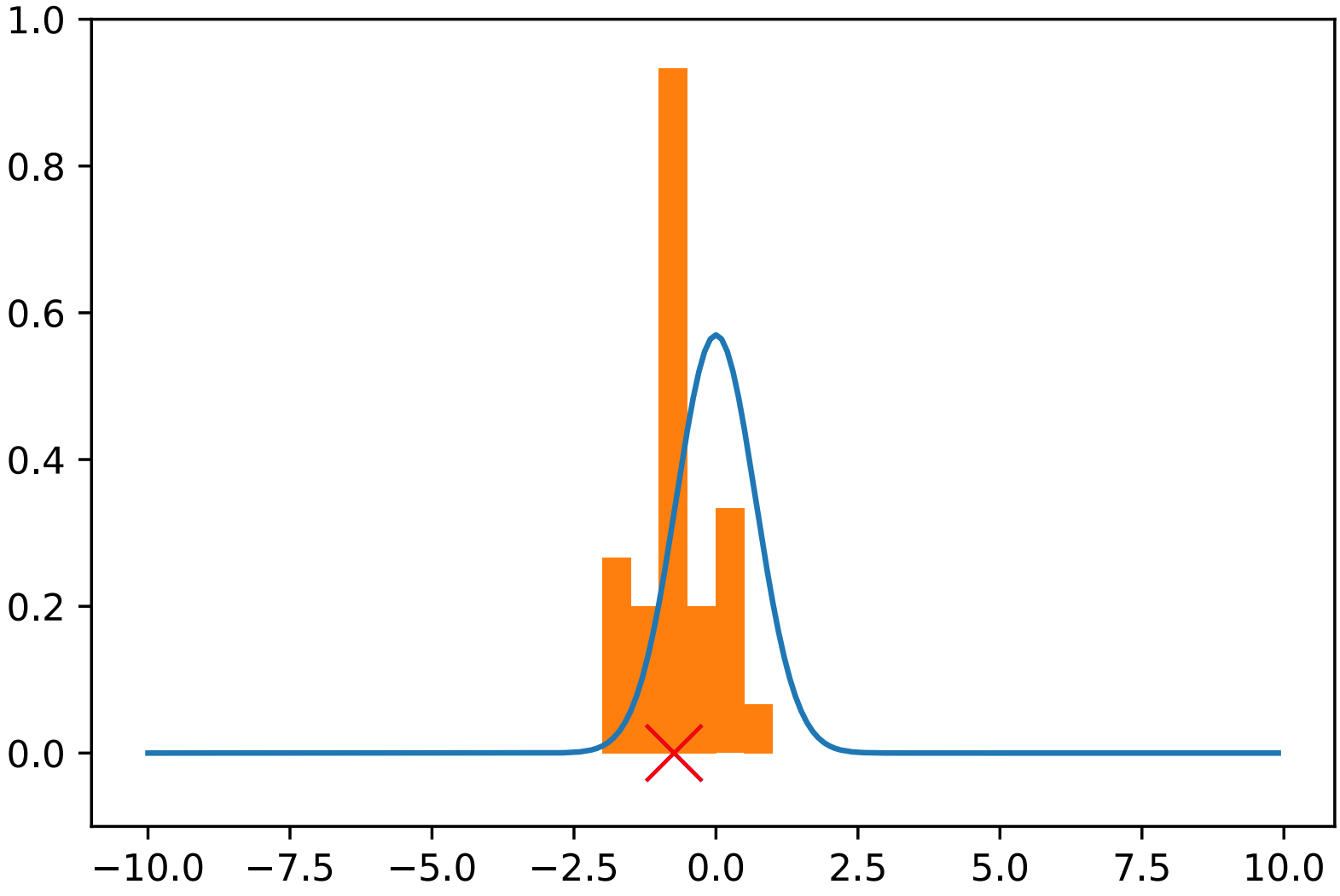
MCMC Example (Sample 20)



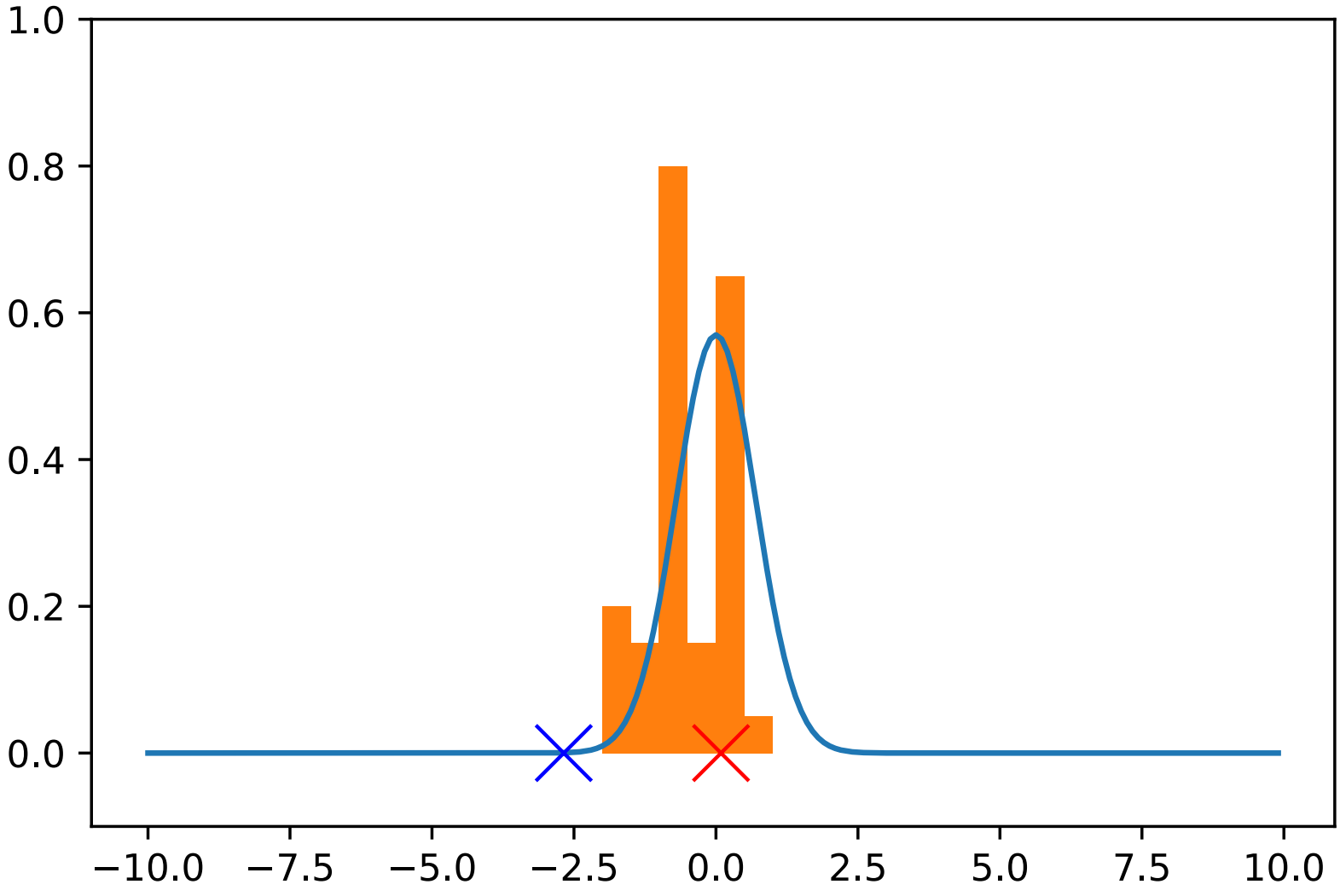
MCMC Example (Sample 30)



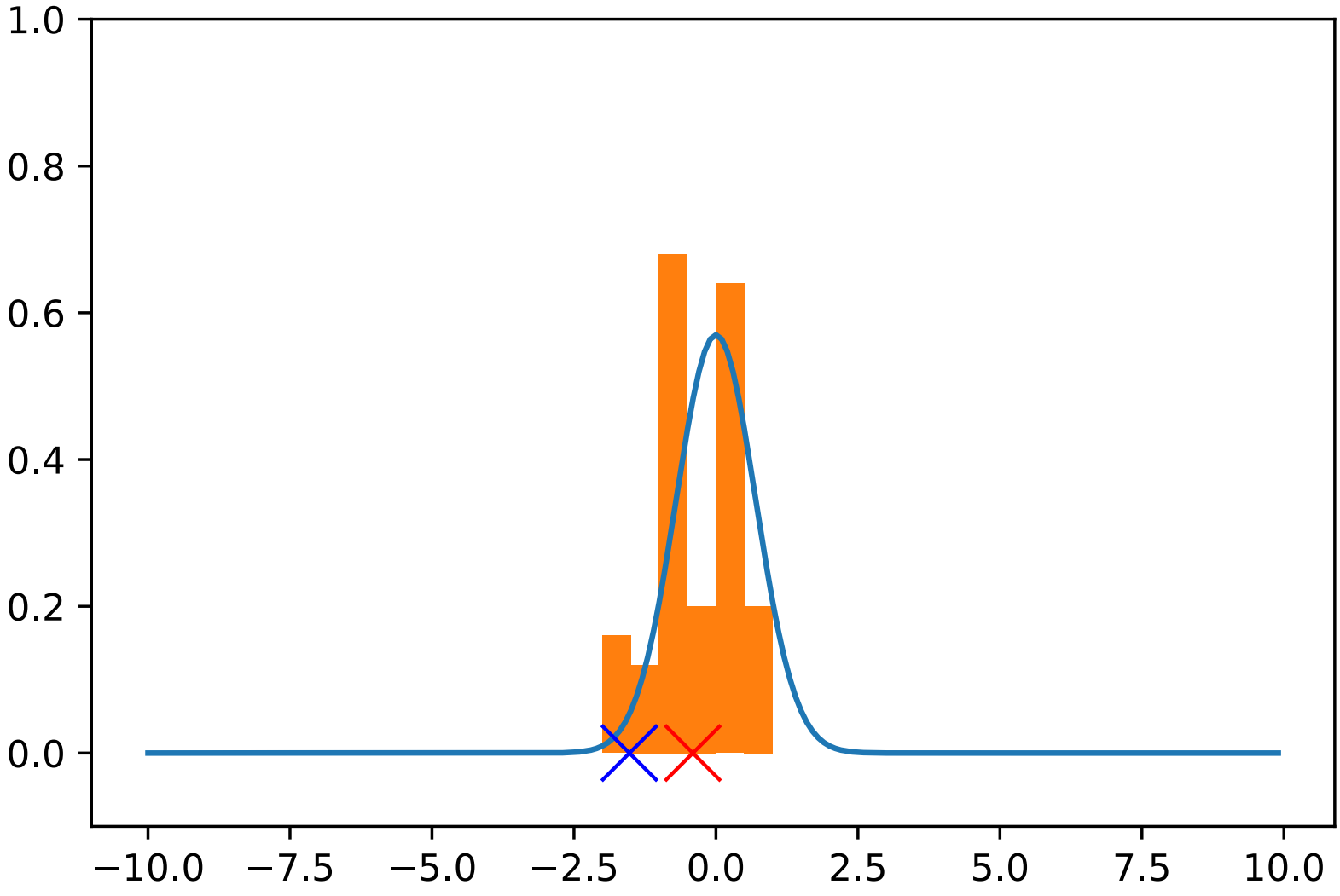
MCMC Example (Sample 40)



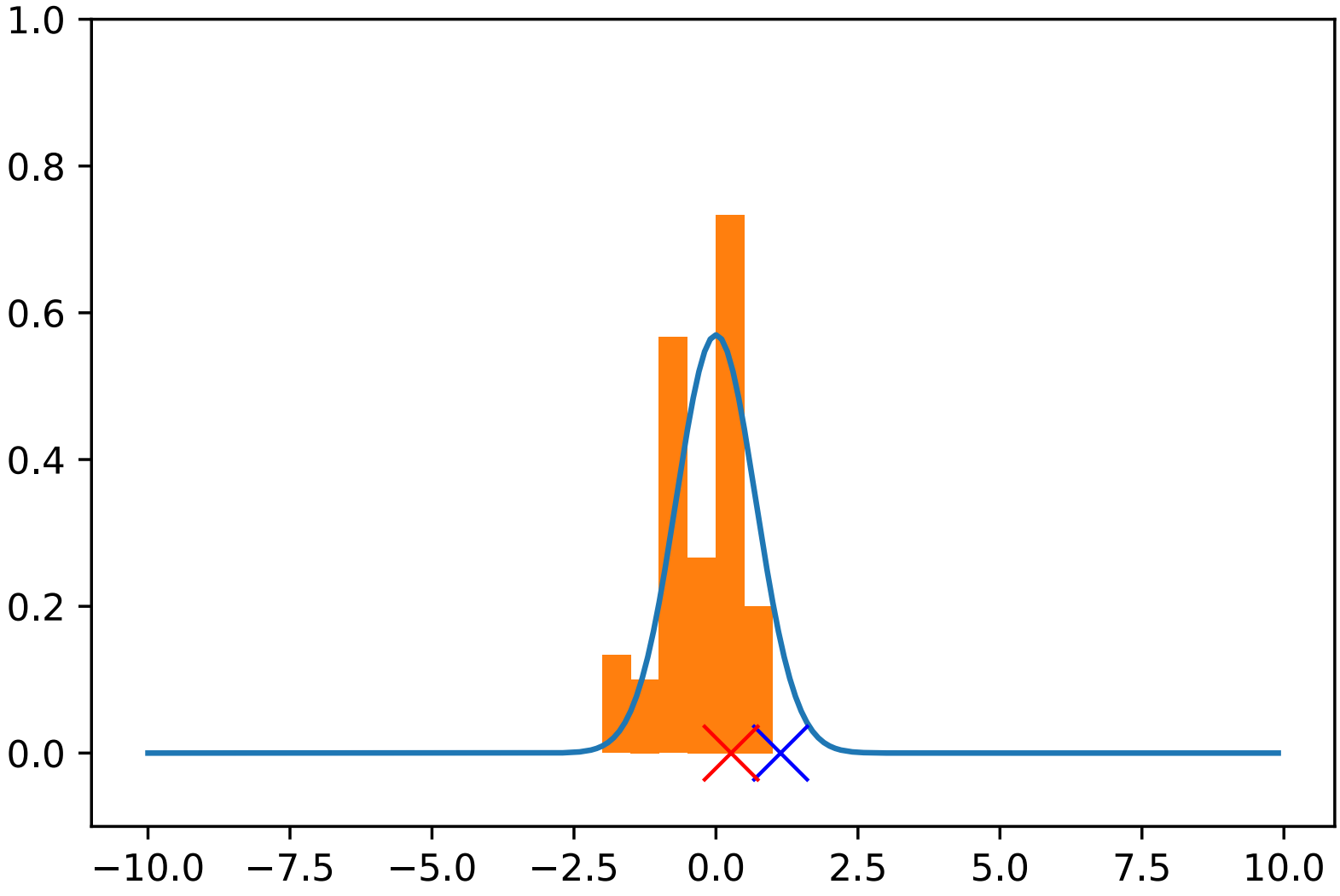
MCMC Example (Sample 50)



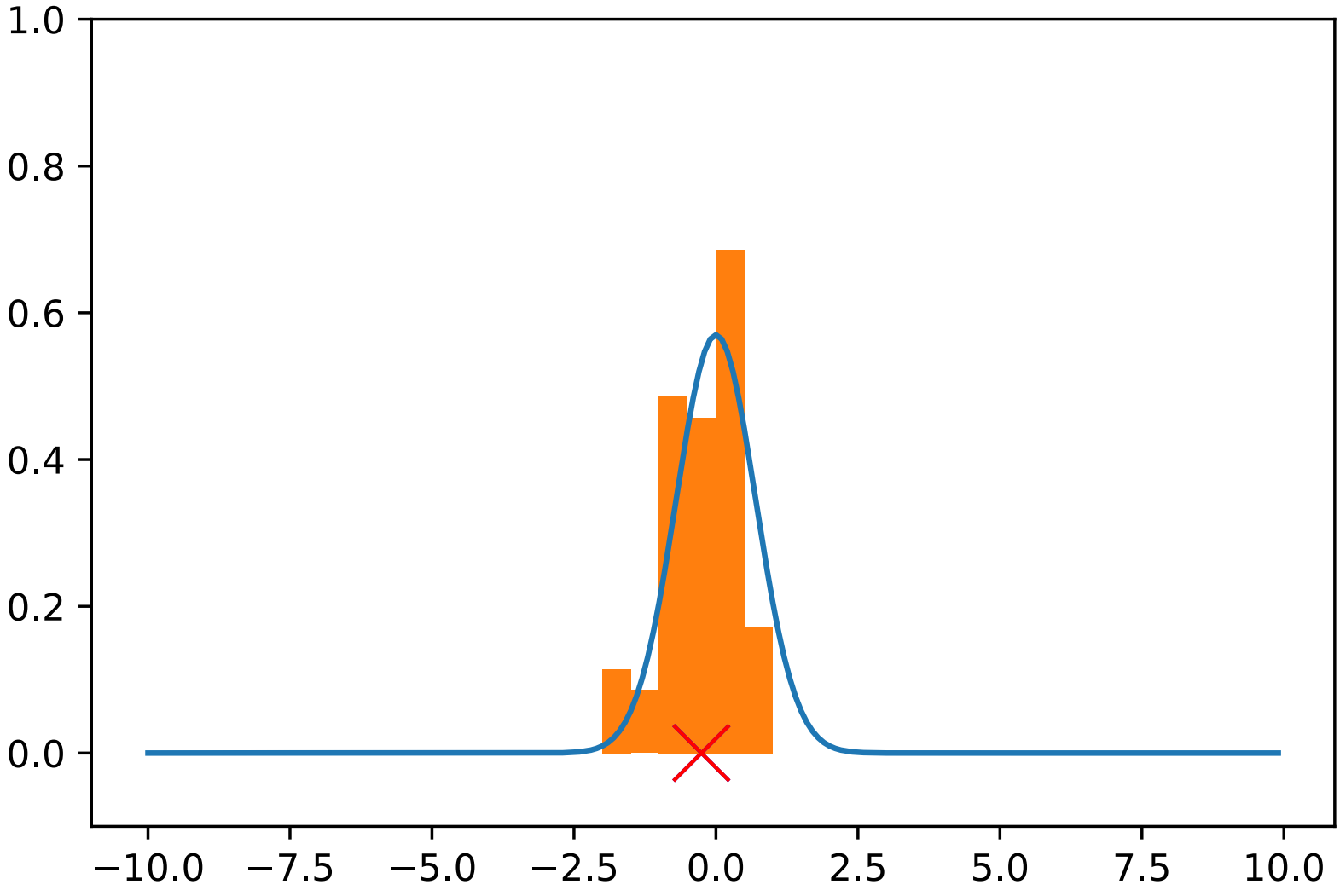
MCMC Example (Sample 60)



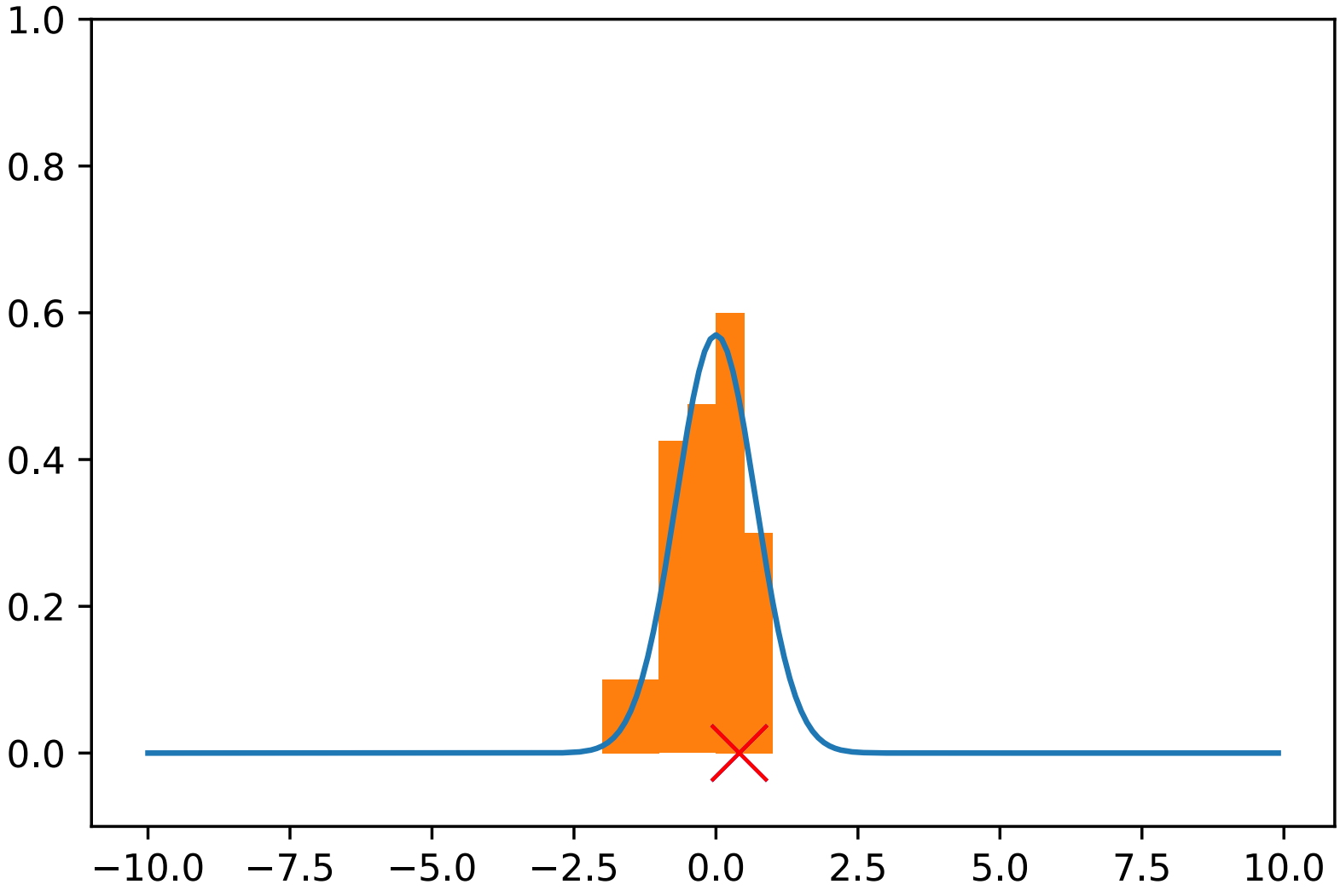
MCMC Example (Sample 70)



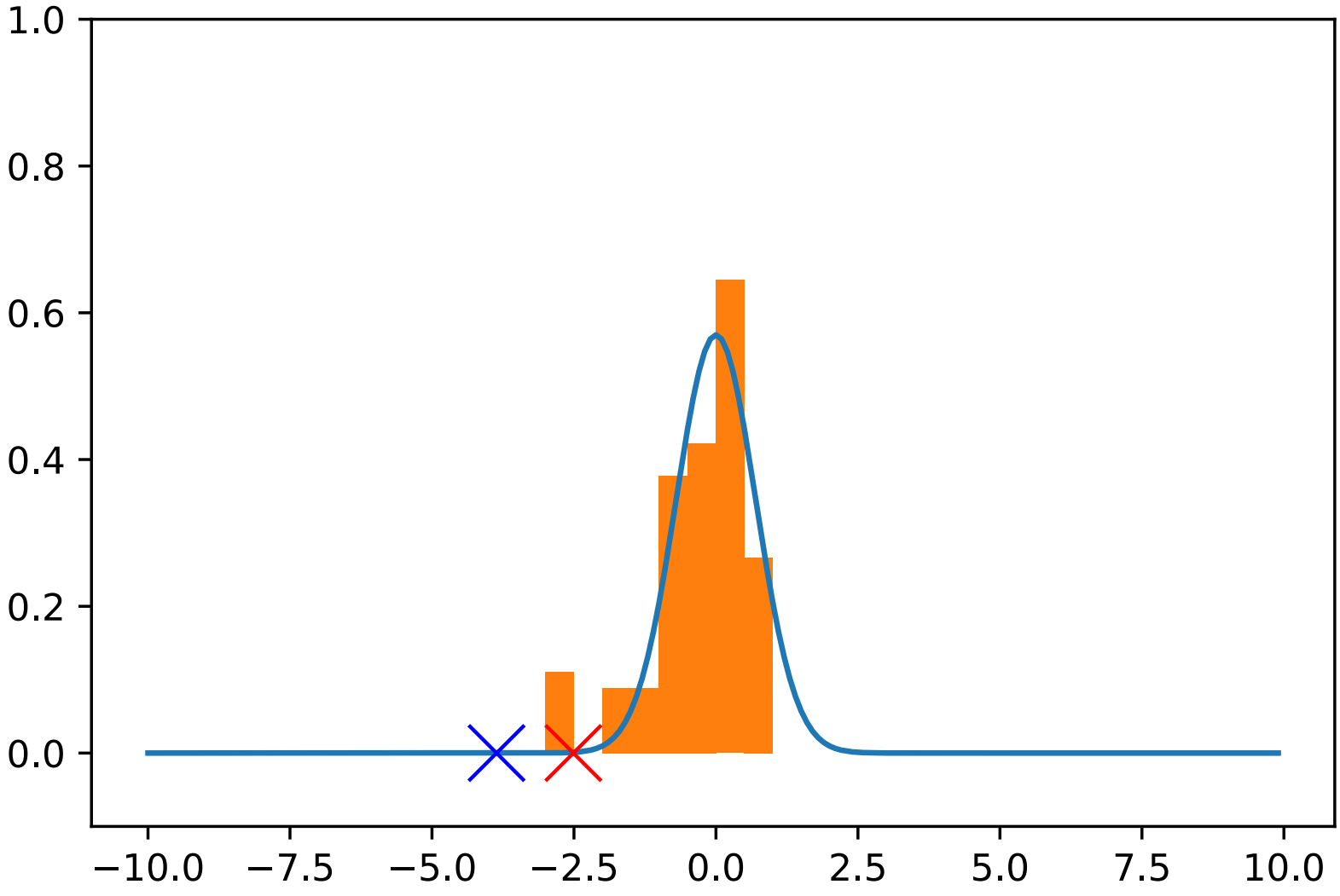
MCMC Example (Sample 80)



MCMC Example (Sample 90)



MCMC Example (Sample 100)



MCMC algorithm

Choose θ^0. For $n \geqslant 0$:

- Given θ^n, generate a proposal θ' from a given proposal distribution $q(\theta'|\theta^n)$
- Accept θ' as a sample with probability

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

  i.e. $\theta^{n+1} = \theta'$ with probability α and $\theta^{n+1} = \theta^n$ otherwise

$\rightarrow$ Sequence of samples from distribution π!

MCMC algorithm

Choose θ^0. For $n \geqslant 0$:

- Given θ^n, generate a proposal θ' from a given proposal distribution $q(\theta'|\theta^n)$
- Accept θ' as a sample with probability

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

  i.e. $\theta^{n+1} = \theta'$ with probability α and $\theta^{n+1} = \theta^n$ otherwise

$\rightarrow$ Sequence of samples from distribution π!

MCMC algorithm

Choose θ^0. For $n \geqslant 0$:

- Given θ^n, generate a proposal θ' from a given proposal distribution $q(\theta'|\theta^n)$
- Accept θ' as a sample with probability

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

  i.e. $\theta^{n+1} = \theta'$ with probability α and $\theta^{n+1} = \theta^n$ otherwise

$\rightarrow$ Sequence of samples from distribution π!

MCMC algorithm

Choose θ^0. For $n \geqslant 0$:

- Given θ^n, generate a proposal θ' from a given proposal distribution $q(\theta'|\theta^n)$
- Accept θ' as a sample with probability

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

  i.e. $\theta^{n+1} = \theta'$ with probability α and $\theta^{n+1} = \theta^n$ otherwise

$\rightarrow$ Sequence of samples from distribution π!

Limits of MCMC

Issues with MCMC:

- Many MCMC steps necessary, each requiring full solution of PDE
- Low acceptance rates for high dimensions $\rightarrow$ waste of samples

$\rightarrow$ Remedy: Multilevel MCMC!

Limits of MCMC

Issues with MCMC:

- Many MCMC steps necessary, each requiring full solution of PDE
- Low acceptance rates for high dimensions $\rightarrow$ waste of samples

$\rightarrow$ Remedy: Multilevel MCMC!

Section 3

Multilevel Markov Chain Monte Carlo

Basic Idea

A380 image: Roger Green

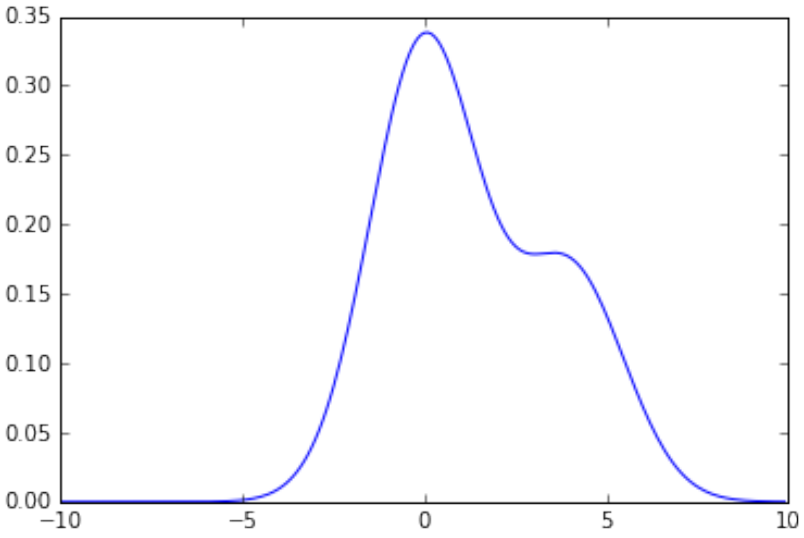
Basic Idea

$\rightarrow$

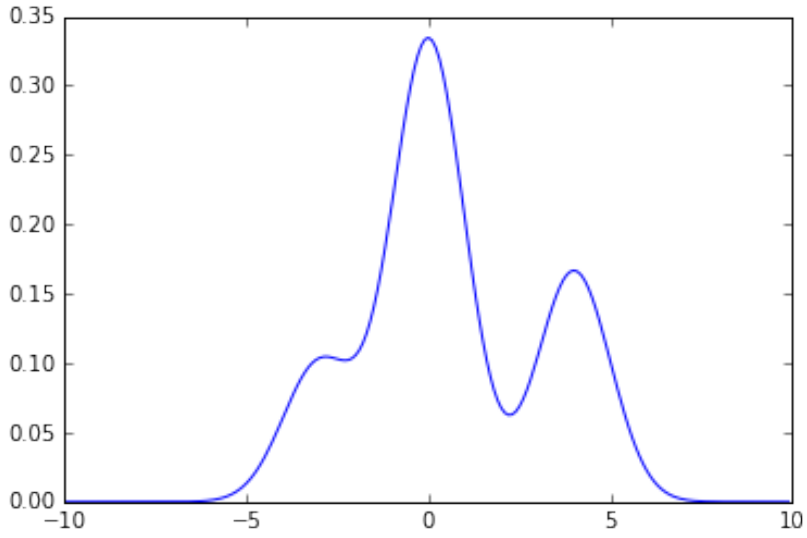
A380 image: Roger Green

Basic Idea

- Construct approximating densities based on cheaper models
  $\rightarrow$ e.g. hierarchy of increasingly fine Finite Element models
- Most samples on coarsest model
- Coarse samples as proposals for fine ones $\rightarrow$ high acceptance rates



$\rightarrow$



Quantity of Interest

- Exploit telescoping sum

$$\mathbb{E}_{\nu^L}[Q_L] = \underbrace{\mathbb{E}_{\nu^0}[Q_0]}_{\text{Rough approximation}} + \sum_{l=1}^{L} \underbrace{(\mathbb{E}_{\nu^l}[Q_l] - \mathbb{E}_{\nu^{l-1}}[Q_{l-1}])}_{\text{Corrections}}$$

- Finer samples only to 'correct' coarse approximation
- Draw proposals for fine levels from coarser ones
  $\rightarrow$ high acceptance rates

Algorithm

On level 0: Regular MCMC chain $\{\theta_i^0\}_{\in N_0}$

On level $k = 1, \ldots, l-1$:

- Given θ_k^j, generate proposal $\theta_k' = \left\{ \begin{array}{c} \theta_{k,C}' \\ \theta_{k,F} \end{array} \right\}$ s.t.
  - $\theta_{k,C}'$ from coarser chain with subsampling and
  - $\theta_{k,F}'$ from fine-level proposal density $q_k(\theta_{k,F}' | \theta_{k,F}^j)$
- Accept θ_k' as θ_k^{j+1} with probability

$$\alpha(\theta_k', \theta_k) = \min \left\{ 1, \frac{\pi_k(\theta_k') q_k(\theta_k^j | \theta_k')}{\pi_k(\theta_k^j) q_k(\theta_k' | \theta_k^j)} \cdot \frac{\pi_{k-1}(\theta_{k,C}^j) q_{k-1}(\theta_{k,C}' | \theta_{k,C}^j)}{\pi_{k-1}(\theta_{k,C}') q_{k-1}(\theta_{k,C}^j | \theta_{k,C}')} \right\},$$

else $\theta_k^{j+1} = \theta_k^j$

Algorithm

On level 0: Regular MCMC chain $\{\theta_i^0\}_{\in N_0}$

On level $k = 1, \ldots, l-1$:

- Given θ_k^j, generate proposal $\theta_k' = \left\{ \begin{array}{c} \theta_{k,C}' \\ \theta_{k,F} \end{array} \right\}$ s.t.
  - $\theta_{k,C}'$ from coarser chain with subsampling and
  - $\theta_{k,F}'$ from fine-level proposal density $q_k(\theta_{k,F}'|\theta_{k,F}^j)$
- Accept θ_k' as θ_k^{j+1} with probability

$$\alpha(\theta_k', \theta_k) = \min\left\{1, \frac{\pi_k(\theta_k')q_k(\theta_k^j|\theta_k')}{\pi_k(\theta_k^j)q_k(\theta_k'|\theta_k^j)} \cdot \frac{\pi_{k-1}(\theta_{k,C}^j)q_{k-1}(\theta_{k,C}'|\theta_{k,C}^j)}{\pi_{k-1}(\theta_{k,C}')q_{k-1}(\theta_{k,C}^j|\theta_{k,C}')}\right\},$$

else $\theta_k^{j+1} = \theta_k^j$

Algorithm

On level 0: Regular MCMC chain $\{\theta_i^0\}_{\in N_0}$

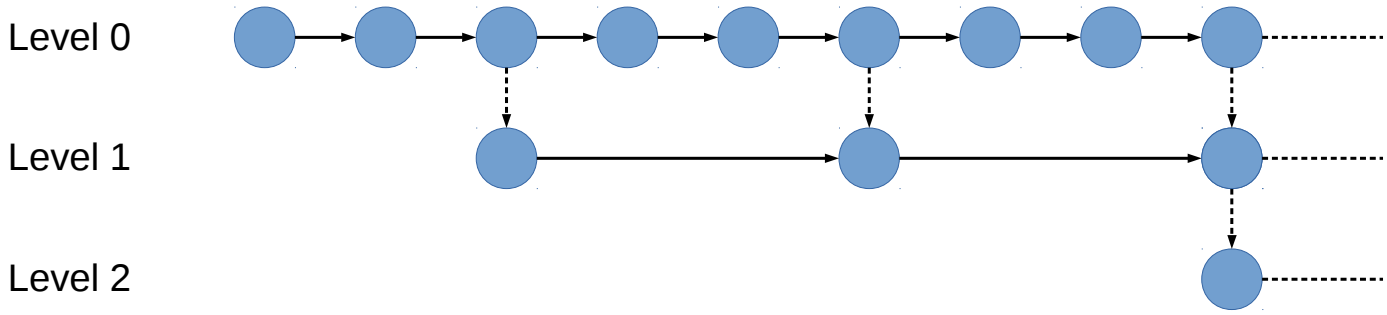
On level $k = 1, \ldots, l-1$:

- Given θ_k^j, generate proposal $\theta_k' = \left\{ \begin{array}{c} \theta_{k,C}' \\ \theta_{k,F} \end{array} \right\}$ s.t.
  - $\theta_{k,C}'$ from coarser chain with subsampling and
  - $\theta_{k,F}'$ from fine-level proposal density $q_k(\theta_{k,F}'|\theta_{k,F}^j)$
- Accept θ_k' as θ_k^{j+1} with probability

$$\alpha(\theta_k', \theta_k) = \min \left\{ 1, \frac{\pi_k(\theta_k')q_k(\theta_k^j|\theta_k')}{\pi_k(\theta_k^j)q_k(\theta_k'|\theta_k^j)} \cdot \frac{\pi_{k-1}(\theta_{k,C}^j)q_{k-1}(\theta_{k,C}'|\theta_{k,C}^j)}{\pi_{k-1}(\theta_{k,C}')q_{k-1}(\theta_{k,C}^j|\theta_{k,C}')} \right\},$$

else $\theta_k^{j+1} = \theta_k^j$

Sampling structure



- Note: Multiindex MCMC analogous, allows for multiple ways of coarsening

Section 4

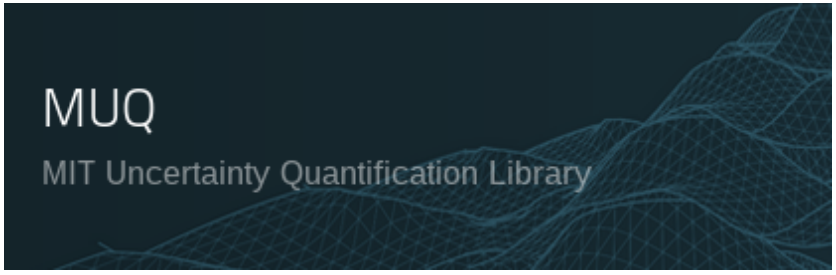
Implementation

DUNE

- Modular C++ framework for PDEs
- Extreme parallel scalability for modern HPC environments
- https://dune-project.org

MUQ

- C++ toolbox for UQ
- Supports hierarchical models, various MCMC methods etc.
- Multilevel/multiindex MCMC infrastructure newly developed!
- http://muq.mit.edu



MIMCMC in MUQ: Internal architecture

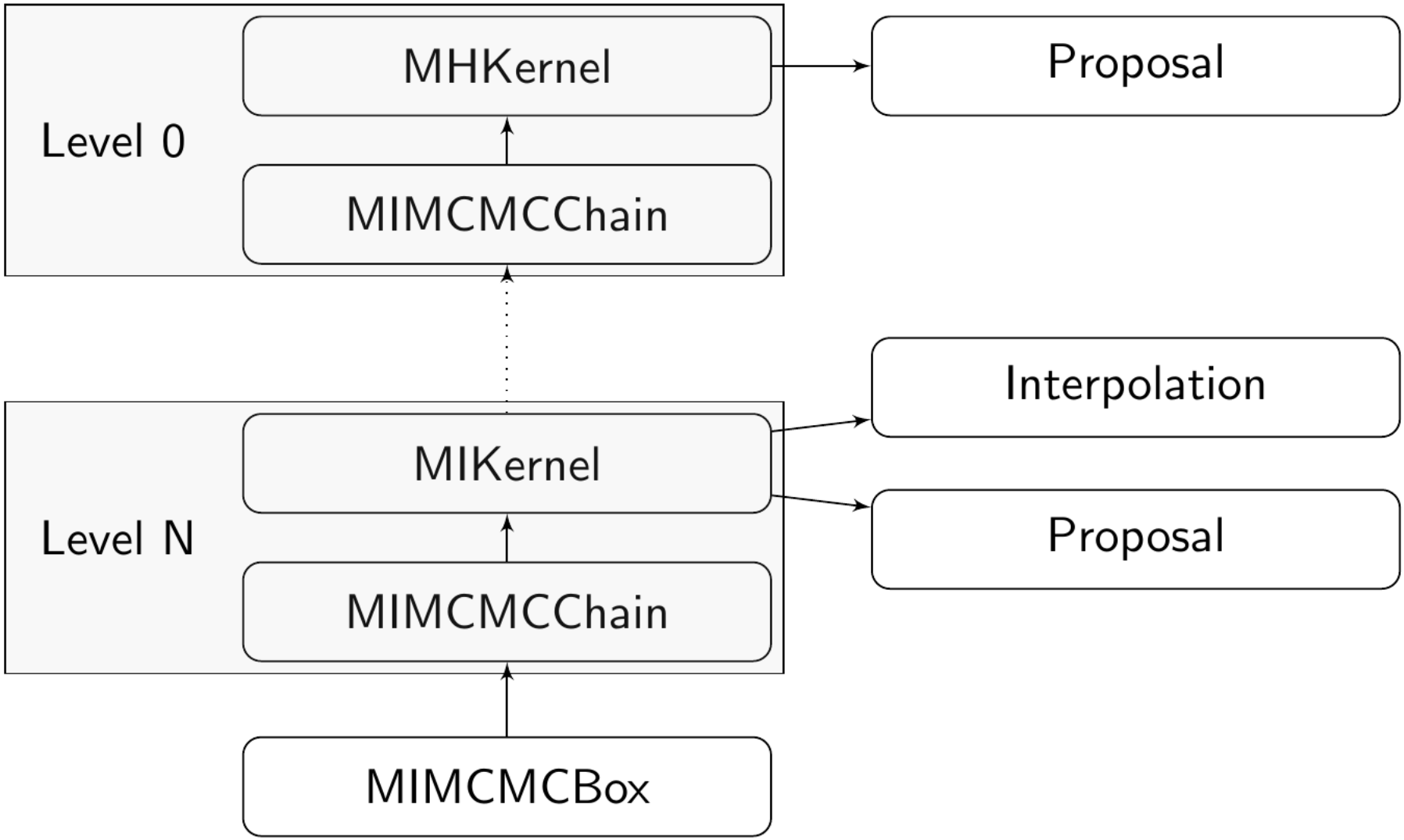


Figure: Multilevel MCMC structure

Using MIMCMC in MUQ: Sampling Problem

```
class MySamplingProblem : public AbstractSamplingProblem {
public:

  MySamplingProblem(std::shared_ptr<MultiIndex> index, std::shared_ptr<parcer::
      Communicator> comm, std::shared_ptr<Eigen::VectorXd> measurements_external_in)
  {
    // Set up model (e.g. PDE solver)
  }

  virtual double LogDensity(unsigned int const t, std::shared_ptr<SamplingState> state,
      AbstractSamplingProblem::SampleType type) override {
    // return your density;
  }

  virtual std::shared_ptr<SamplingState> QOI() override {
    // return your quantity of interest;
  }

};
```

Using MIMCMC in MUQ: Multiindex structure

```
class  MyMIComponentFactory {

  virtual std::shared_ptr<MCMCProposal> Proposal (std::shared_ptr<MultiIndex> index,
                  std::shared_ptr<AbstractSamplingProblem> samplingProblem) override {
    // Which MCMC proposal to use
  }

  virtual std::shared_ptr<MultiIndex> FinestIndex() override {
    // Your finest model index
  }

  virtual std::shared_ptr<MCMCProposal> CoarseProposal (...) override {
    // How to draw coarse proposals
  }

  virtual std::shared_ptr<AbstractSamplingProblem> SamplingProblem (std::shared_ptr<
      MultiIndex> index) override {
    // Your sampling problem
  }

  virtual std::shared_ptr<MIInterpolation> Interpolation (std::shared_ptr<MultiIndex>
      index) override {
    // How to interpolate coarse samples on fine level
  }

  virtual Eigen::VectorXd StartingPoint (std::shared_ptr<MultiIndex> index) override {
    // Where to start chains
  }

};
```

Using MIMCMC in MUQ: Run it

```
// Set parameters
pt.put("NumSamples", 1000);

// Multiindex run
MIMCMC mimcmc(pt, componentFactory);
mimcmc.Run();

// Single-level run
SLMCMC slmcmc (pt, componentFactory);
slmcmc.Run();
```

Section 5

Application

Darcy with random conductivity field

- Simple Darcy model

$$-\nabla \cdot (k(x,\omega)\nabla p(x,\omega)) = 0 \qquad \forall x \in D$$

- Conductivity as parameter to be estimated
- Grid of measurements

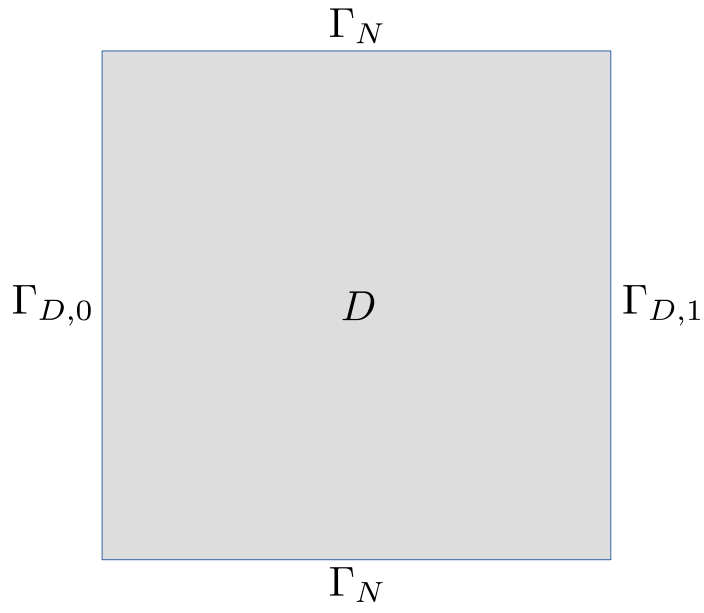
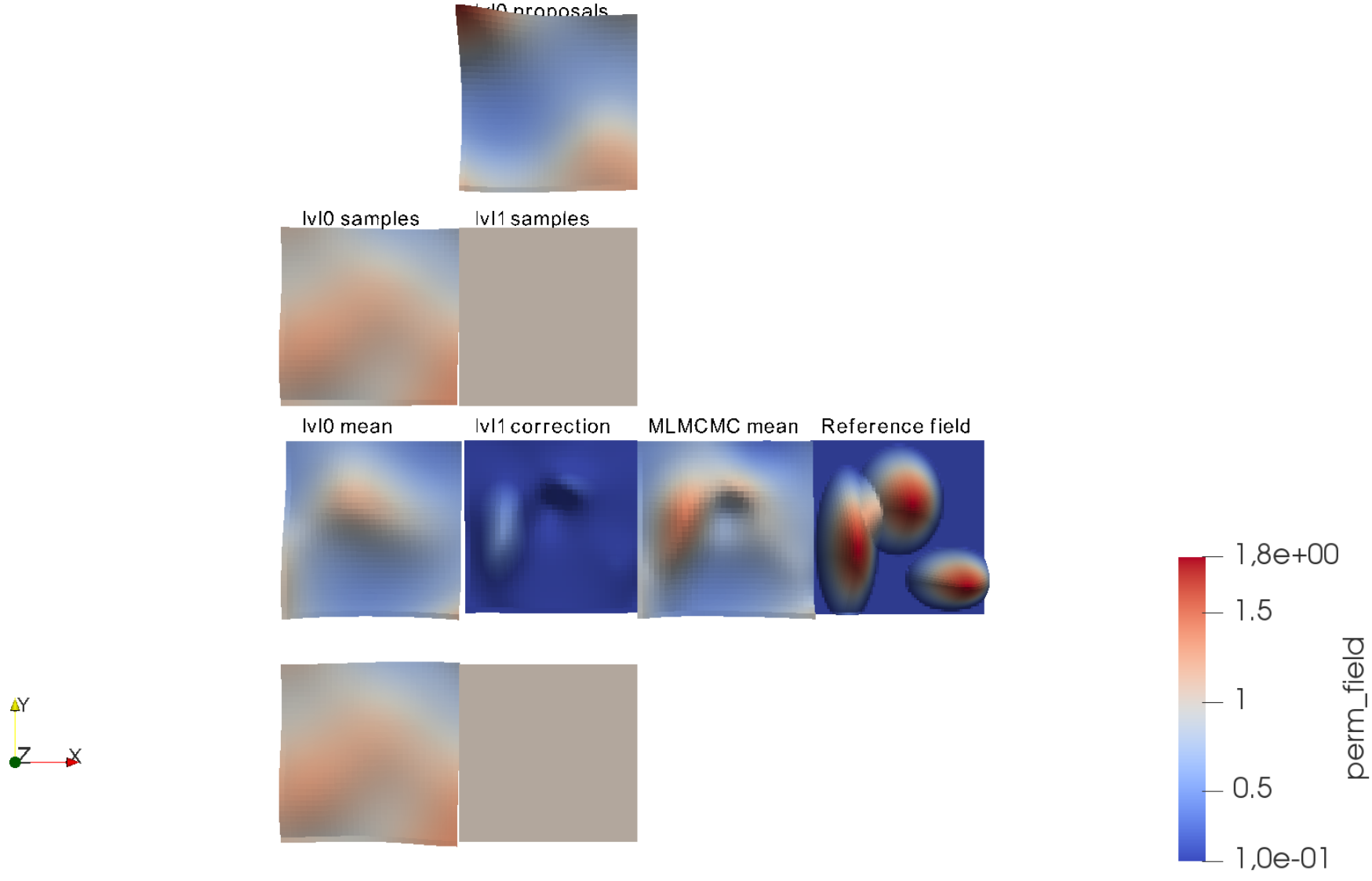


Figure: Domain and boundary condition setup

MLMCMC example

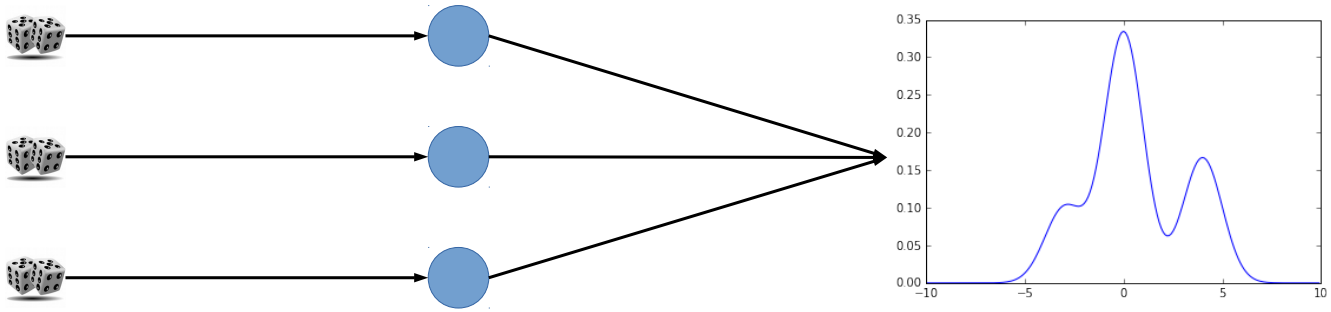


Section 6

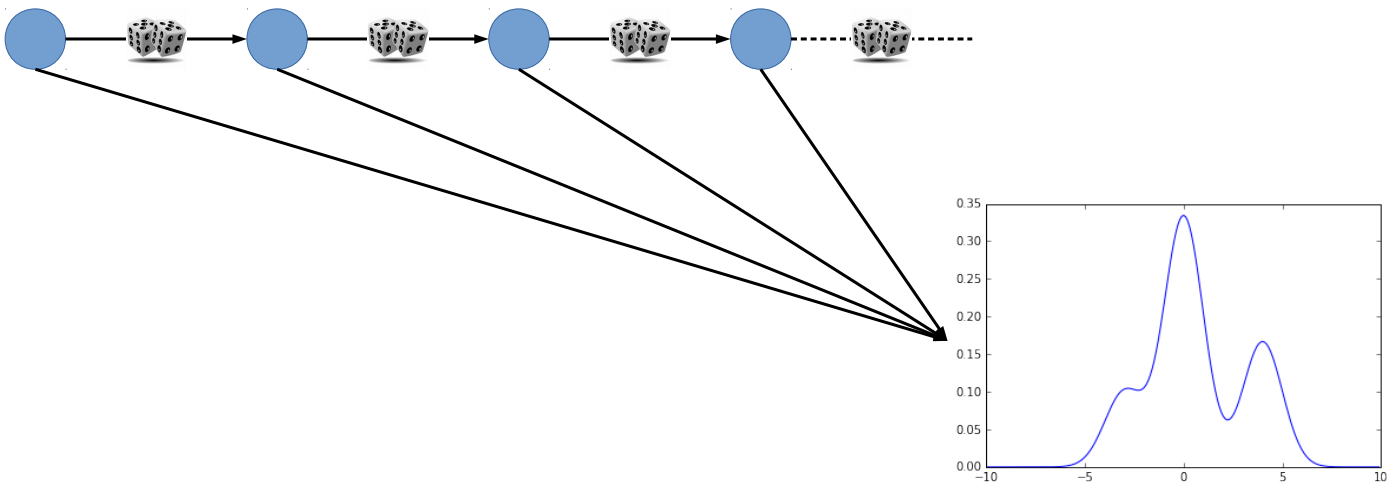
Parallelization

Why not trivial?

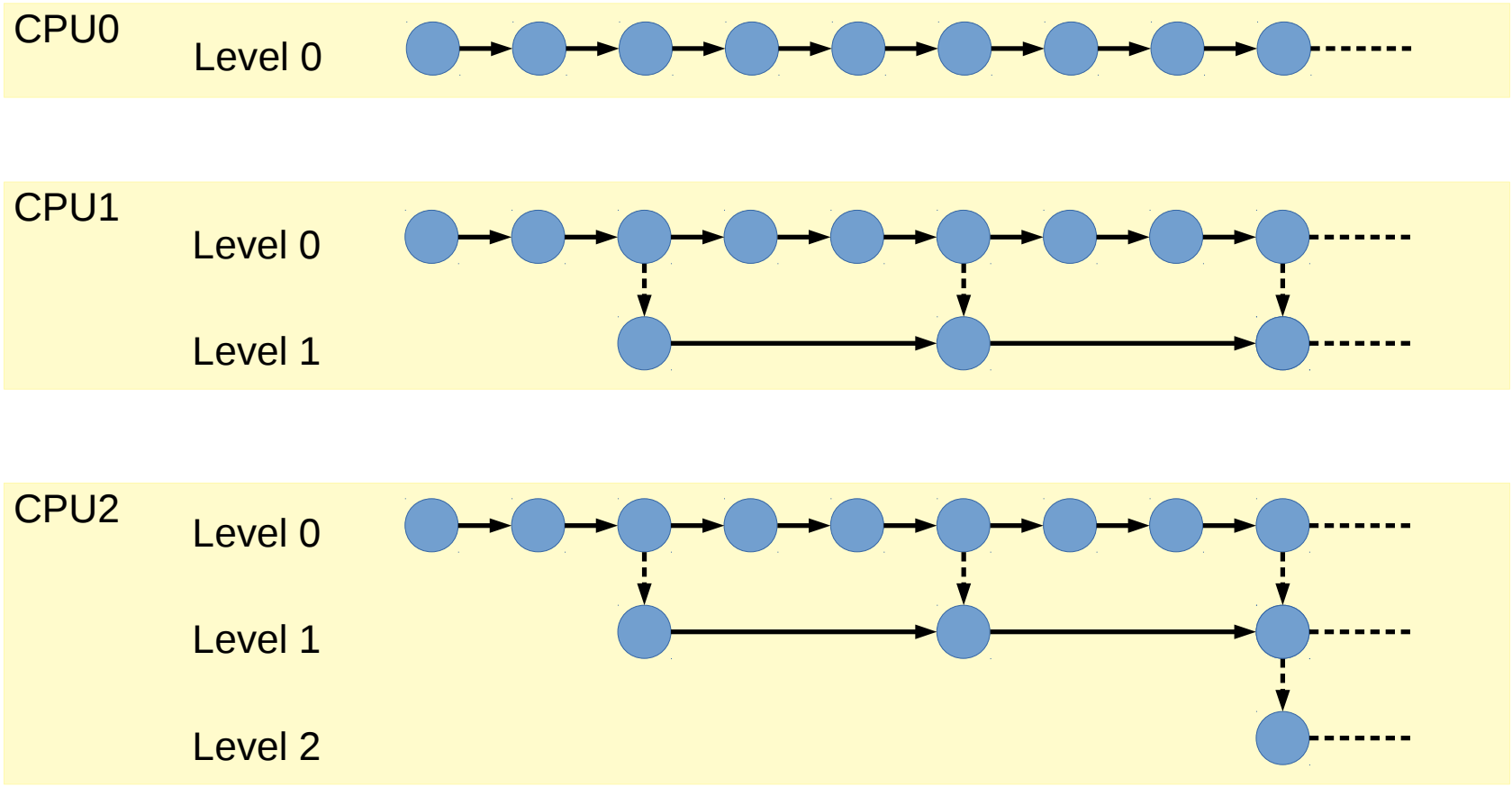
Monte Carlo



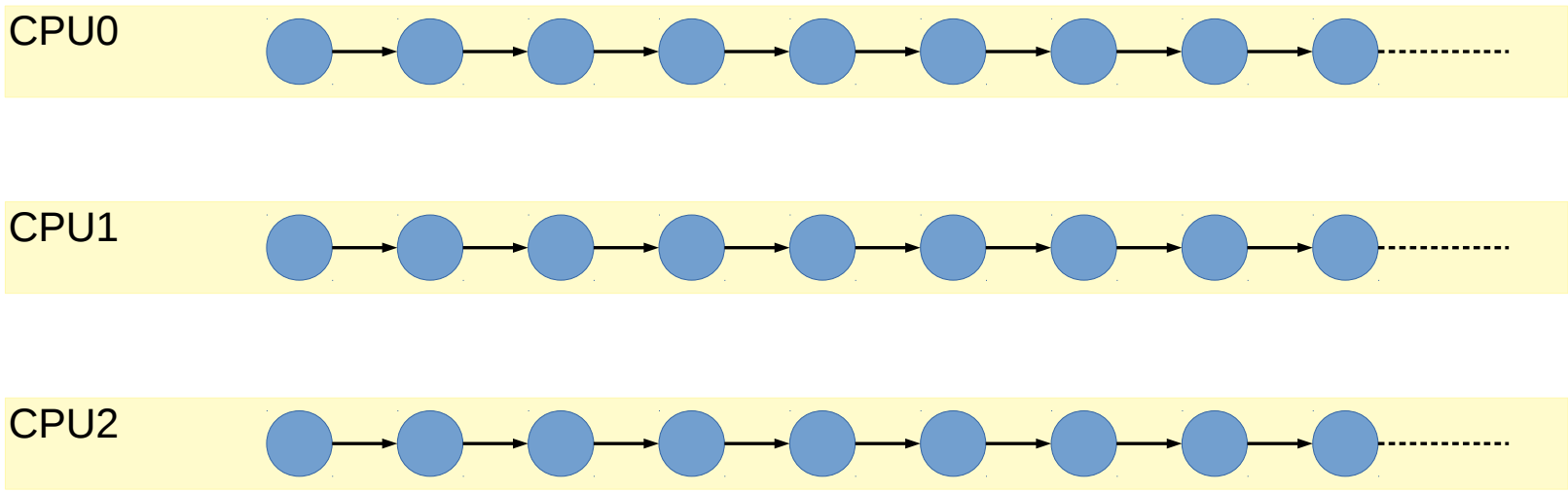
Markov Chain Monte Carlo



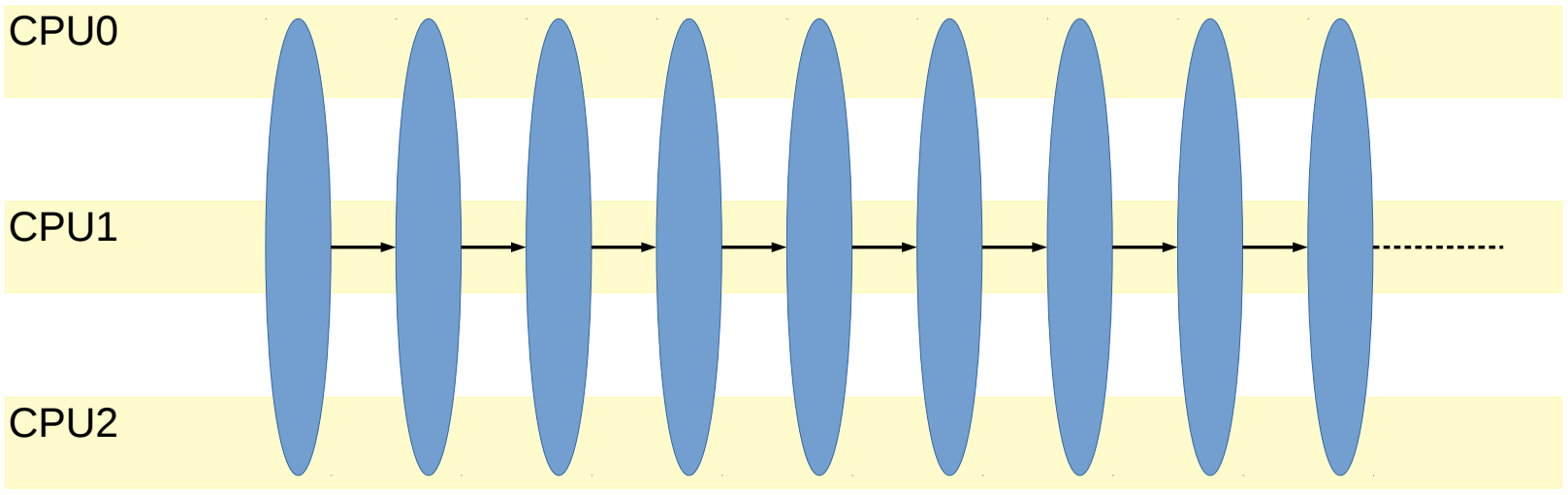
Parallelization: Levels



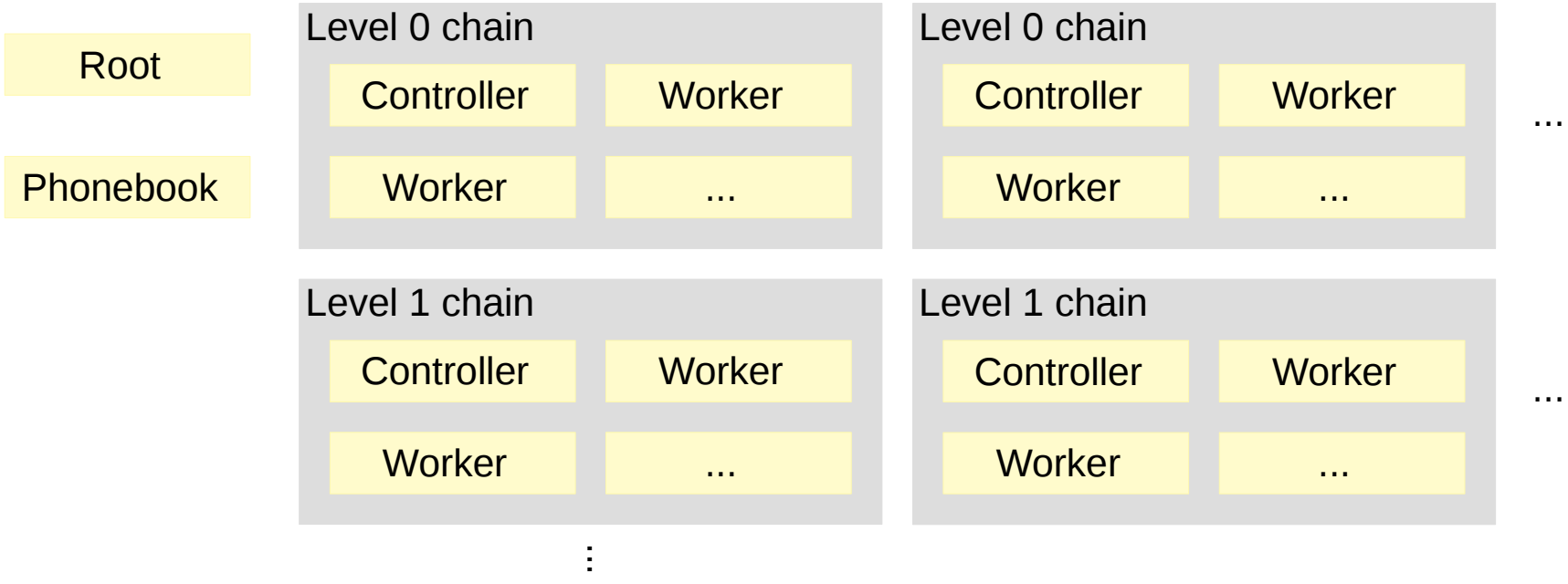
Parallelization: Chains



Parallelization: Models



Parallelization: CPU layout



- Fully dynamic assignment of cores
- Flexible request-driven approach
  $\rightarrow$ no assumptions on high-level structure!

Initial Parallel Testing

- Same model, 60000 samples on coarse, 10000 on fine level
- Run on 80 cores (4 nodes)
- Static load balancing, 87,95% load

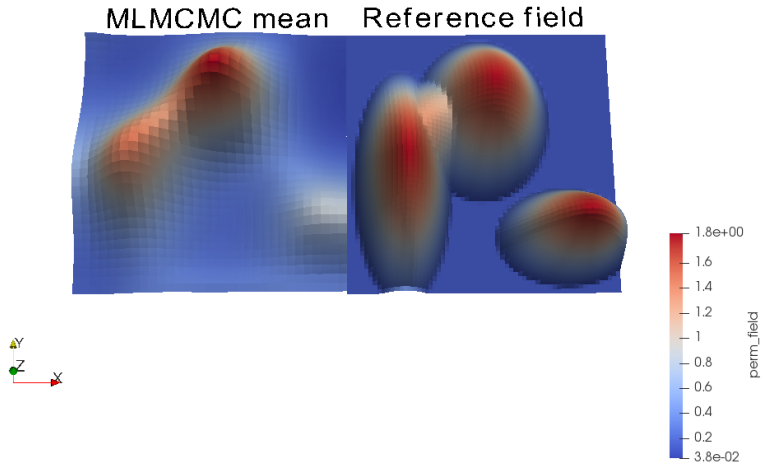


Figure: Parallel MLMCMC result

Future Goals

- Exa-scale HPC scalability
- Interface with ExaHype, run on 500k core SuperMUC
- Connect MLMCMC and Model Order Reduction

Figure: SuperMUC

Summary

- Methods offer excellent potential for HPC-grade inference
- General-purpose MLMCMC/MIMCMC implementation, available now in MUQ2!
- Soon: Abstracted parallelization with minimal user intervention


T J Dodwell, C Ketelsen, R Scheichl and A L Teckentrup. *A Hierarchical Multilevel Markov Chain Monte Carlo Algorithm with Applications to Uncertainty Quantification in Subsurface Flow*. SIAM/ASA Uncertainty Quantification, 2015.


Ajay Jasra, Kengo Kamatani, Kody J. H. Law and Yan Zhou *A Multi-Index Markov Chain Monte Carlo Method*. International Journal for Uncertainty Quantification, 2018.