Bayesian parameter estimation in predictive engineering

Damon McDougall

Institute for Computational Engineering and Sciences, UT Austin

14th August 2014

Motivation

Understand physical phenomena

Observations of phenomena

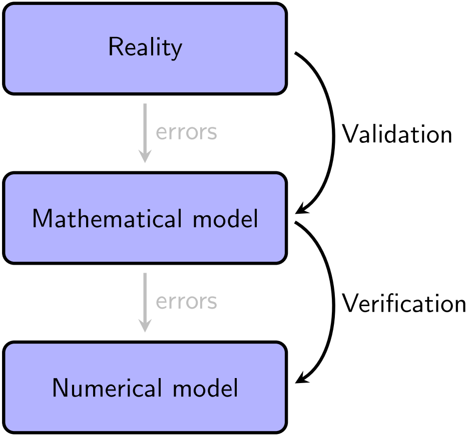
Mathematical model of phenomena (includes some parameters that characterise behaviour)

Numerical model approximating mathematical model

Find parameters in a situation of interest

Use the parameters to do something cool

Understanding errors



Setup

Model (usually a PDE): $\mathcal{G}(u, \theta)$ where u is the initial condition and θ are model paramaters.

u: perhaps an initial condition

θ: perhaps some interesting model parameters (diffusion, convection speed, permeability field, material properties)

Observations:

$$\begin{aligned} & y_{j,k} = u(x_j, t_k) + \eta_{j,k}, \quad \eta_{j,k} \overset{\text{i.i.d}}{\sim} \mathcal{N}(0, \sigma^2) \\ \rightsquigarrow \quad & y = \mathcal{G}(\theta) + \eta, \quad \eta \sim \mathcal{N}(0, \sigma^2 I) \end{aligned}$$

Want:

$$\mathbb{P}(\theta | y) \propto \mathbb{P}(y|\theta)\mathbb{P}(\theta)$$

Why?

Do we need Bayes' theorem?

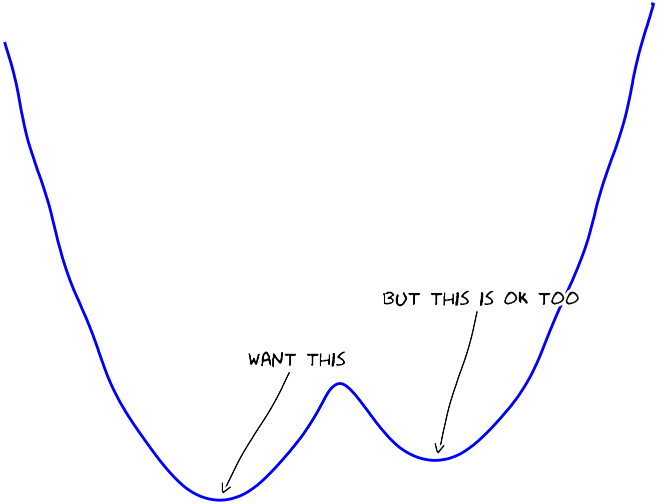
Is Bayes' theorem really necessary? We could minimise

$$J(\theta) = \frac{1}{2\sigma^2}\|\mathcal{G}(\theta) - y\|^2 + \frac{1}{2\lambda^2}\|\theta\|^2$$

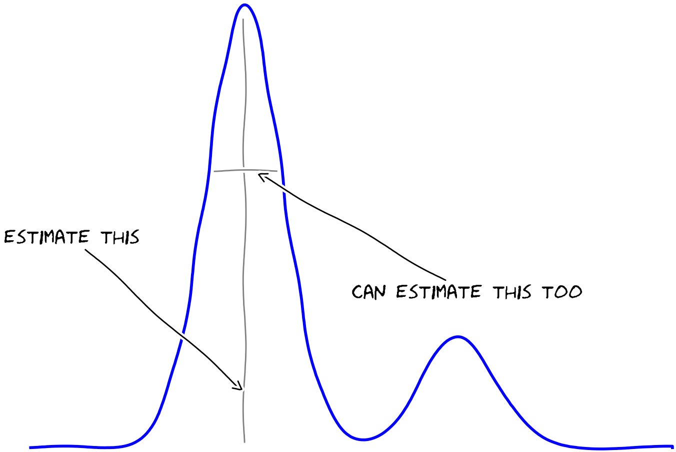
to get

$$\theta^* = \text{argmin}_\theta \, J(\theta)$$

Do we need Bayes' theorem?



Do we need Bayes' theorem?



Do we need Bayes' theorem?

Bayesian methods involve estimating *uncertainty* (as well as mean). They're equivalent.

Deterministic optimisation:

$$J(\theta) = \underbrace{\frac{1}{2\sigma^2}\|\mathcal{G}(\theta) - y\|^2}_{\textit{misfit}} + \underbrace{\frac{1}{2\lambda^2}\|\theta\|^2}_{\textit{regularisation}}$$

Bayesian framework:

$$\begin{aligned}\exp(-J(\theta)) &= \underbrace{\exp\left(-\frac{1}{2\sigma^2}\|\mathcal{G}(\theta) - y\|^2\right)}_{\textit{likelihood}}\underbrace{\exp\left(-\frac{1}{2\lambda^2}\|\theta\|^2\right)}_{\textit{prior}} \\ &= \mathbb{P}(y|\theta)\mathbb{P}(\theta) \\ &\propto \mathbb{P}(\theta|y)\end{aligned}$$

Method for solving Bayesian inverse problems

- Kalman filtering/smoothing methods
  - Kalman filter (Kalman)
  - Ensemble Kalman filter (Evensen)
- Variational methods
  - 3D VAR (Lorenc)
  - 4D VAR (Courtier, Talagrand, Lawless)
- Particle methods
  - Particle filter (Doucet)
- Sampling methods
  - Markov chain Monte Carlo (Metropolis, Hastings)

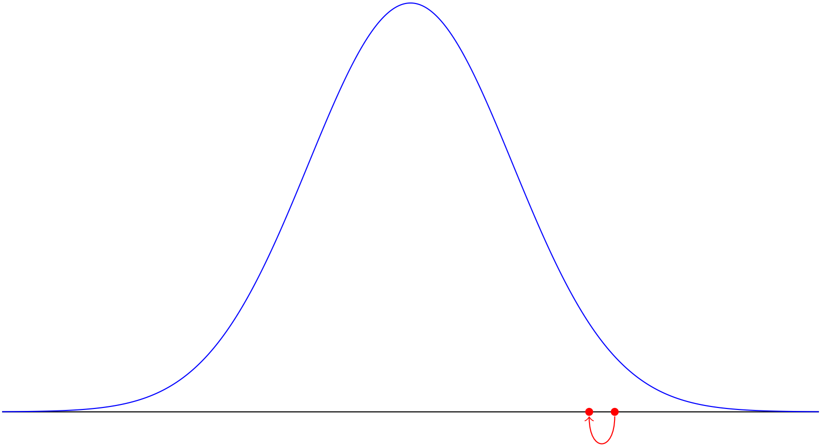
This list is not exhaustive. The body of work is prodigious.

QUESO

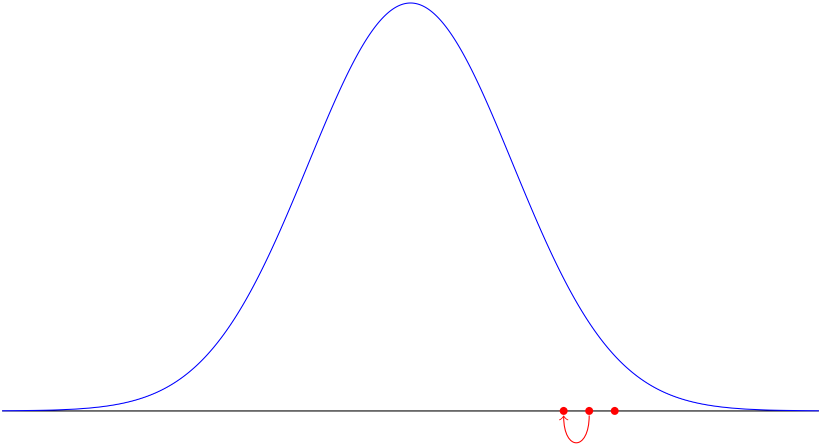
Nutshell: `QUESO` gives samples from $\mathbb{P}(\theta|y)$ (called MCMC)

- Library for Quantifying Uncertainty in Estimation, Simulation and Optimisation
- Born in 2008 as part of PECOS PSAAP programme
- Provides robust and scalable sampling algorithms for UQ in computational models
- Open source
- `C++`
- `MPI` for communication
- Parallel chains, each chain can house several processes
- Dependencies are `MPI`, `Boost` and `GSL`. Other optional features exist
- https://github.com/libqueso/queso

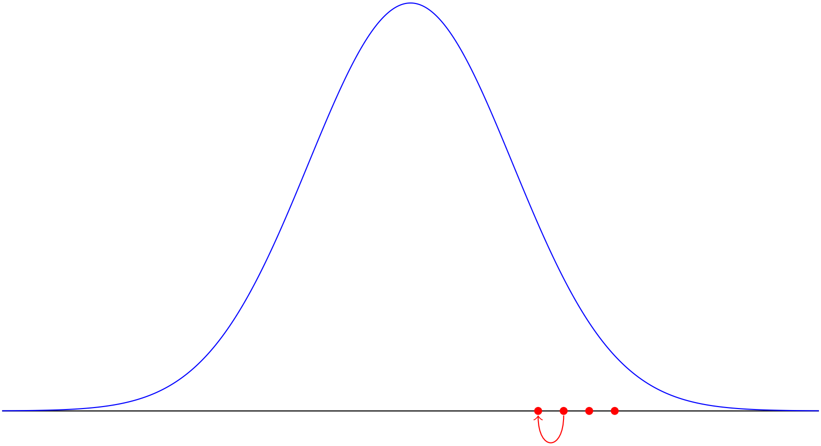
What does MCMC look like?



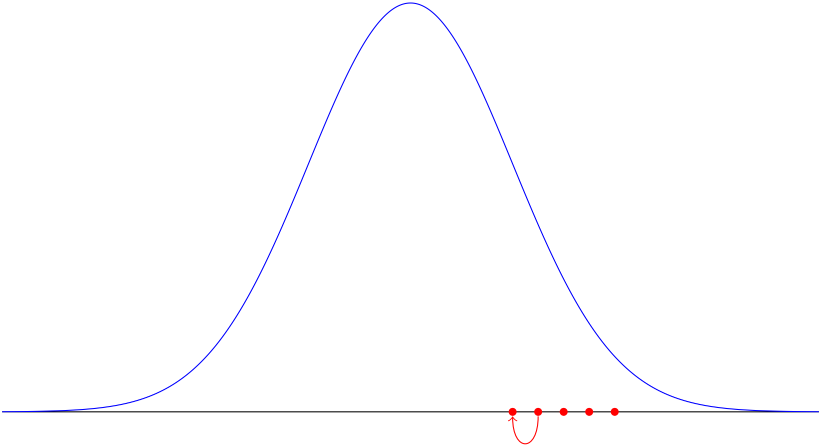
What does MCMC look like?



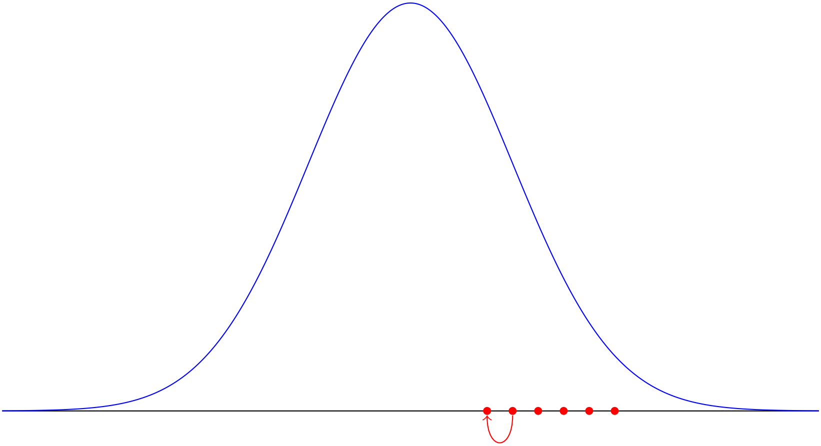
What does MCMC look like?



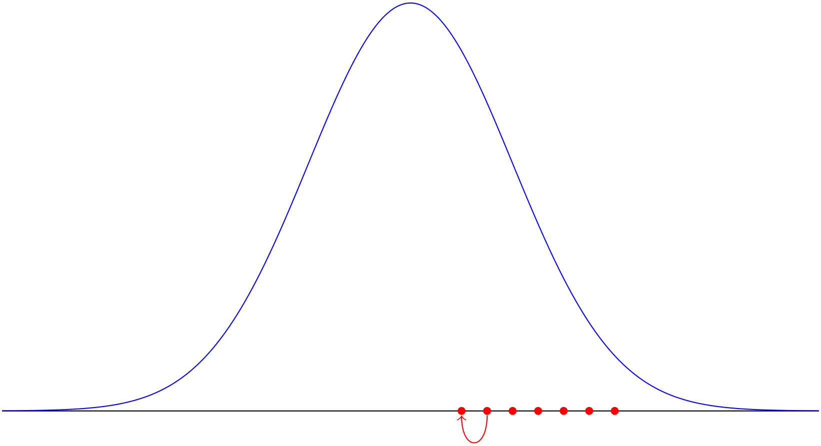
What does MCMC look like?



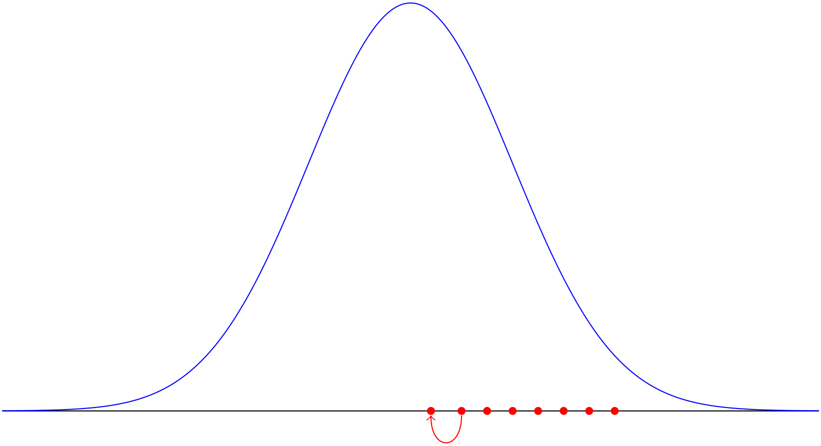
What does MCMC look like?



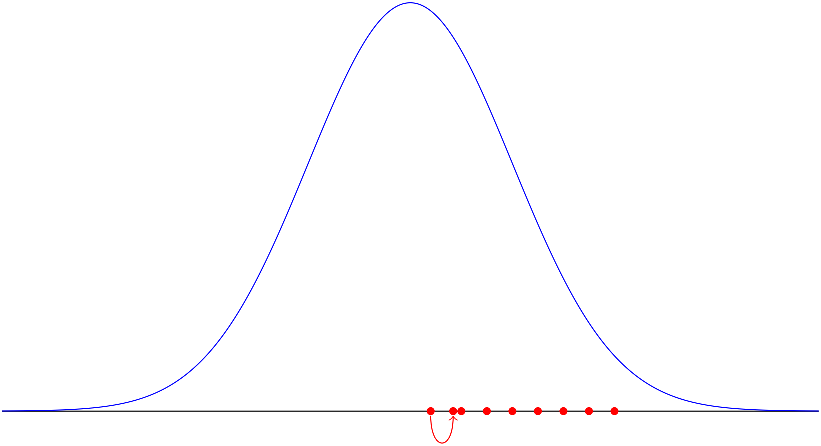
What does MCMC look like?



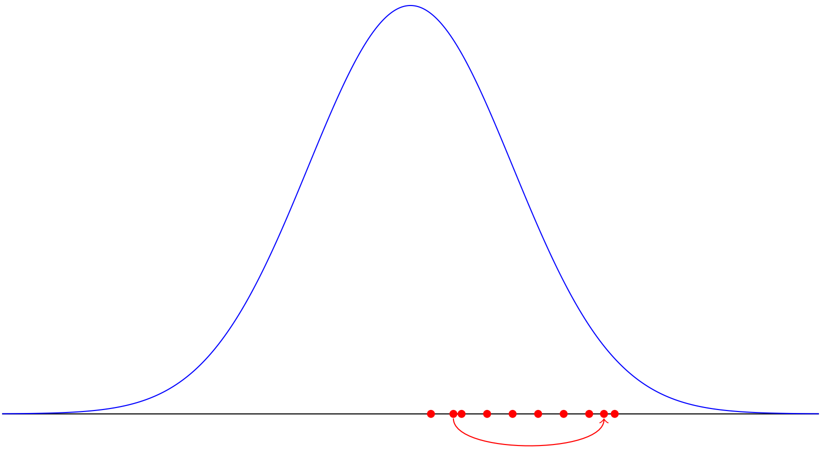
What does MCMC look like?



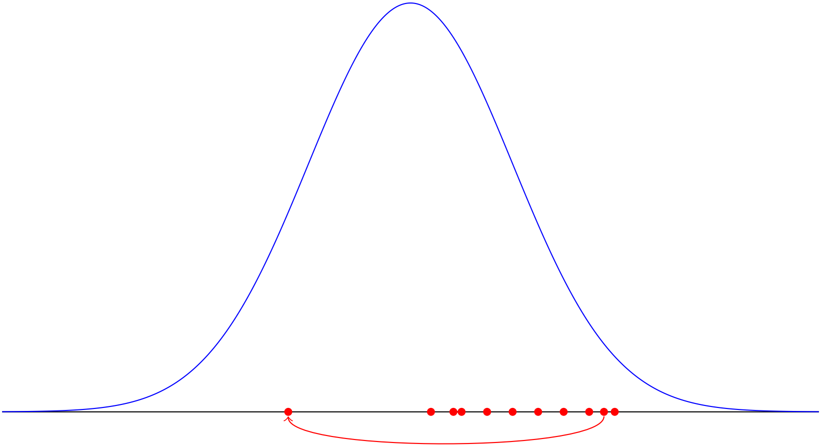
What does MCMC look like?



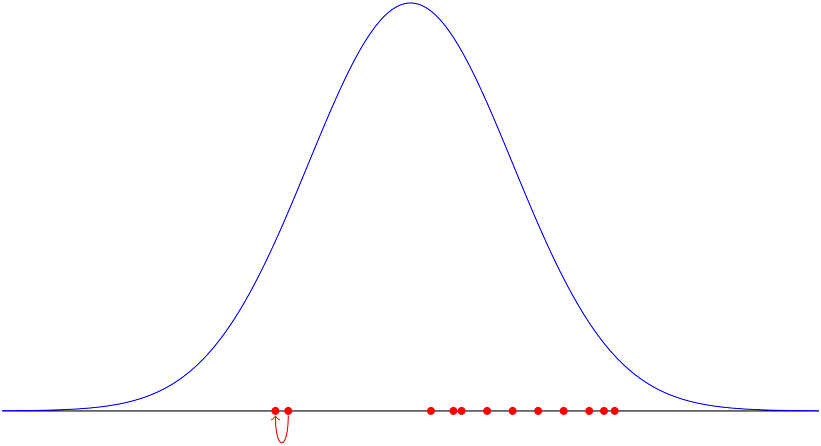
What does MCMC look like?



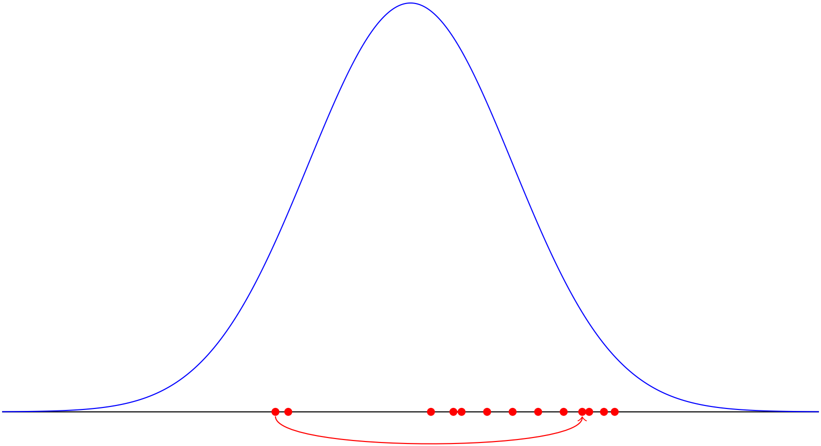
What does MCMC look like?



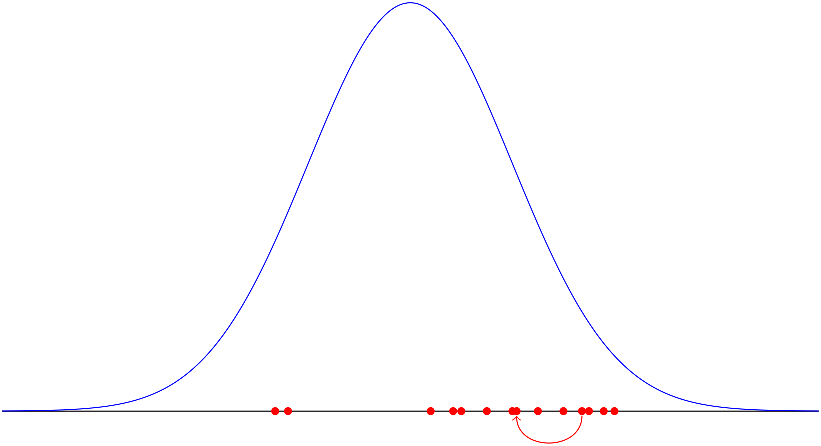
What does MCMC look like?



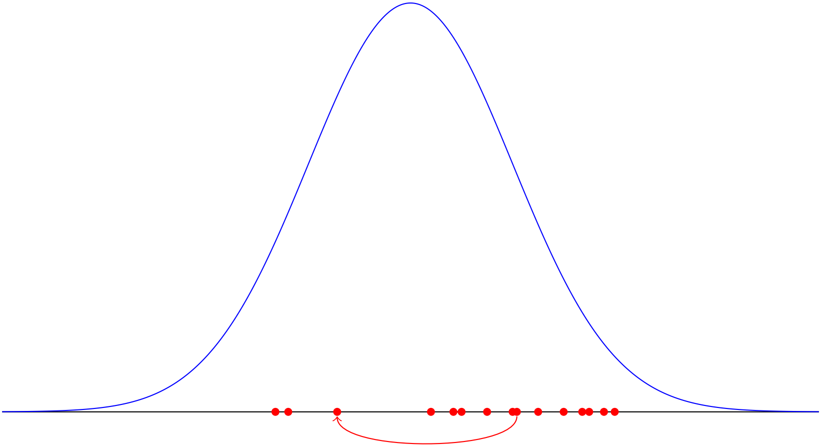
What does MCMC look like?



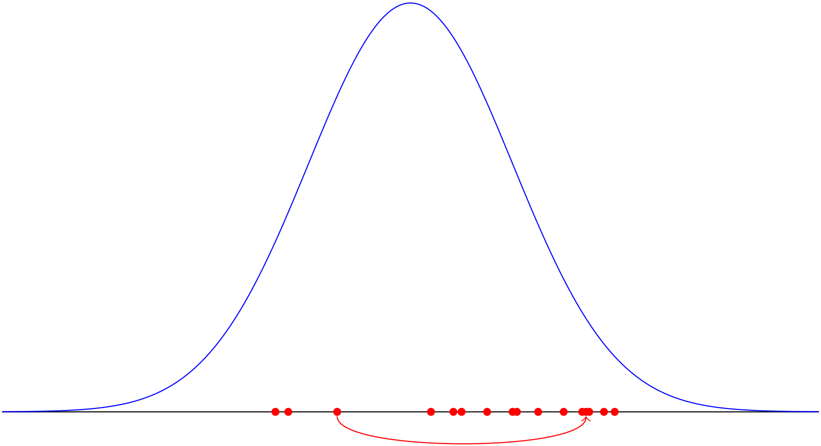
What does MCMC look like?



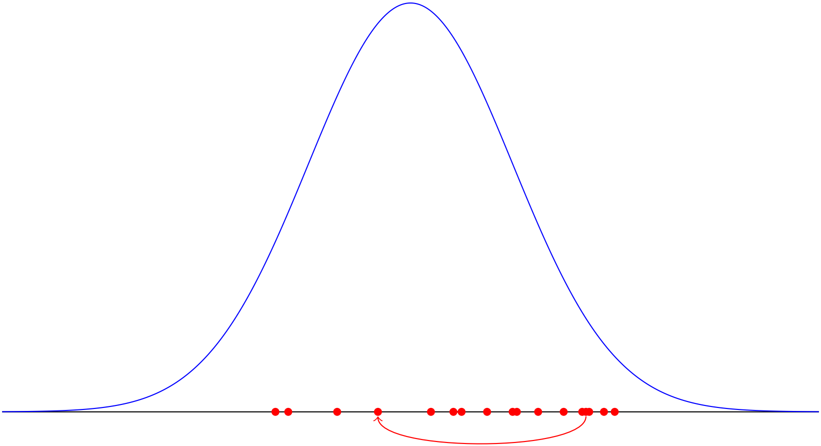
What does MCMC look like?



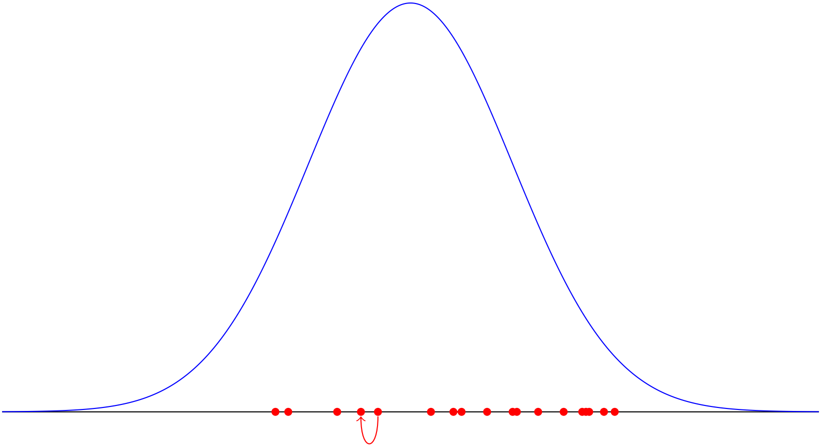
What does MCMC look like?



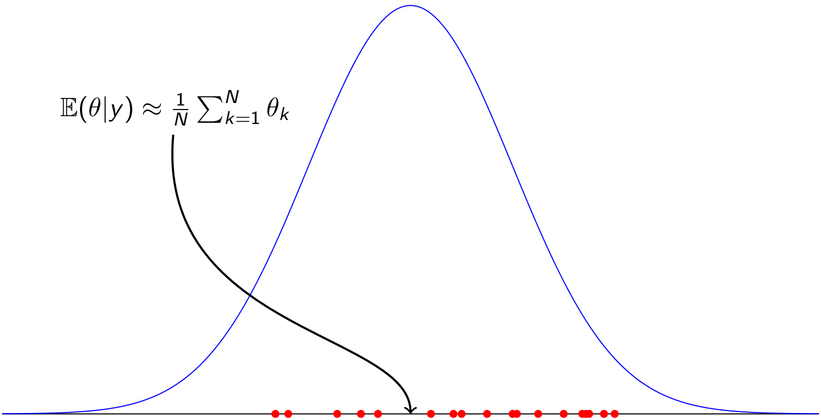
What does MCMC look like?



What does MCMC look like?



What does MCMC look like?



How to do MCMC? Sampling $\mathbb{P}(\theta|y)$

- Idea: Construct $\{\theta_k\}_{k=1}^{\infty}$ cleverly such that $\{\theta_k\}_{k=1}^{\infty} \overset{\text{i.i.d}}{\sim} \mathbb{P}(\theta|y)$

    1. Let θ_j be the 'current' state in the sequence and construct a *proposal*, z

How to do MCMC? Sampling $\mathbb{P}(\theta|y)$

- Idea: Construct $\{\theta_k\}_{k=1}^{\infty}$ cleverly such that $\{\theta_k\}_{k=1}^{\infty} \overset{\text{i.i.d}}{\sim} \mathbb{P}(\theta|y)$

  1. Let θ_j be the 'current' state in the sequence and construct a *proposal*, z

  2. Define $\Phi(\cdot) := \frac{1}{2\sigma^2} \left\| \mathcal{G}(\cdot) - y \right\|^2$

How to do MCMC? Sampling $\mathbb{P}(\theta|y)$

- Idea: Construct $\{\theta_k\}_{k=1}^{\infty}$ cleverly such that $\{\theta_k\}_{k=1}^{\infty} \overset{\text{i.i.d}}{\sim} \mathbb{P}(\theta|y)$

  1. Let θ_j be the 'current' state in the sequence and construct a *proposal*, z

  2. Define $\Phi(\cdot) := \frac{1}{2\sigma^2} \left\| \mathcal{G}(\cdot) - y \right\|^2$
  3. Compute $\alpha(\theta_j, z) = 1 \wedge \exp(\Phi(\theta_j) - \Phi(z))$

How to do MCMC? Sampling $\mathbb{P}(\theta|y)$

- Idea: Construct $\{\theta_k\}_{k=1}^{\infty}$ cleverly such that $\{\theta_k\}_{k=1}^{\infty} \overset{\text{i.i.d}}{\sim} \mathbb{P}(\theta|y)$

  1. Let θ_j be the 'current' state in the sequence and construct a *proposal*, z

  2. Define $\Phi(\cdot) := \frac{1}{2\sigma^2} \|\mathcal{G}(\cdot) - y\|^2$

  3. Compute $\alpha(\theta_j, z) = 1 \wedge \exp(\Phi(\theta_j) - \Phi(z))$

  4. Let

$$\theta_{j+1} = \begin{cases} \theta & \text{with probability } \alpha(\theta_j, z) \\ \theta_j & \text{with probability } 1 - \alpha(\theta_j, z) \end{cases}$$

How to do MCMC? Sampling $\mathbb{P}(\theta|y)$

- Idea: Construct $\{\theta_k\}_{k=1}^{\infty}$ cleverly such that $\{\theta_k\}_{k=1}^{\infty} \overset{\text{i.i.d}}{\sim} \mathbb{P}(\theta|y)$

1. Let θ_j be the 'current' state in the sequence. Make a draw $\xi \sim \mathbb{P}(\theta)$ and construct a *proposal*, z

$$z = (1-\beta^2)^{\frac{1}{2}}\theta_j + \beta\xi, \quad \text{some } \beta \in (0,1)$$

2. Define $\Phi(\cdot) := \frac{1}{2\sigma^2}\|\mathcal{G}(\cdot) - y\|^2$
3. Compute $\alpha(\theta_j, z) = 1 \wedge \exp(\Phi(\theta_j) - \Phi(z))$
4. Let

$$\theta_{j+1} = \begin{cases} \theta & \text{with probability } \alpha(\theta_j, z) \\ \theta_j & \text{with probability } 1 - \alpha(\theta_j, z) \end{cases}$$

How to do MCMC? Sampling $\mathbb{P}(\theta|y)$

- Idea: Construct $\{\theta_k\}_{k=1}^{\infty}$ cleverly such that $\{\theta_k\}_{k=1}^{\infty} \overset{\text{i.i.d}}{\sim} \mathbb{P}(\theta|y)$

  1. Let θ_j be the 'current' state in the sequence. Make a draw $\xi \sim \mathbb{P}(\theta)$ and construct a *proposal*, z

  $$z = (1-\beta^2)^{\frac{1}{2}}\theta_j + \beta\xi, \quad \text{some } \beta \in (0,1)$$

  2. Define $\Phi(\cdot) := \frac{1}{2\sigma^2}\left\|\mathcal{G}(\cdot) - y\right\|^2$
  3. Compute $\alpha(\theta_j, z) = 1 \wedge \exp(\Phi(\theta_j) - \Phi(z))$
  4. Let

  $$\theta_{j+1} = \begin{cases} \theta & \text{with probability } \alpha(\theta_j, z) \\ \theta_j & \text{with probability } 1 - \alpha(\theta_j, z) \end{cases}$$

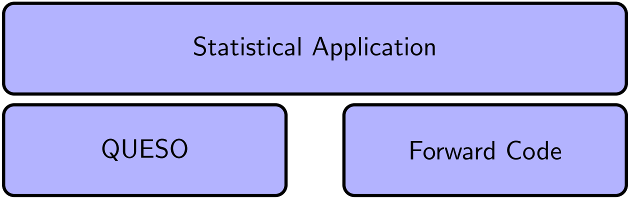
- Take θ_1 to be a draw from $\mathbb{P}(\theta)$

Why use QUESO?

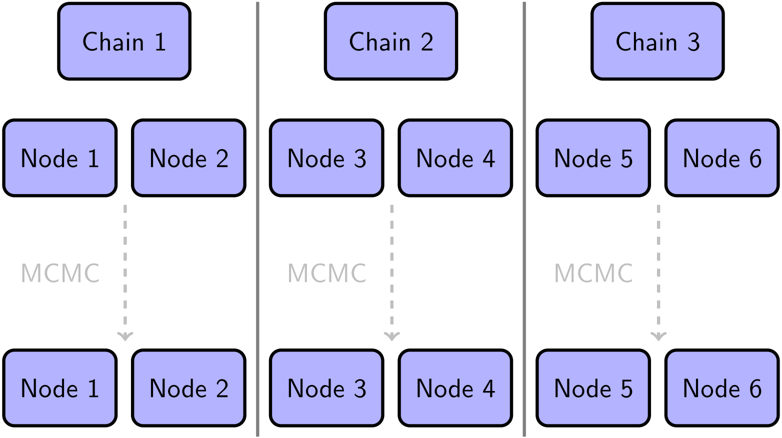
Other solutions are available, e.g. R, PyMC, emcee, MICA, Stan.

QUESO solves the same problem, but:

- Has been designed to be used with large forward problems
- Has been used successfully with 5000+ cores
- Leverages parallel MCMC algorithms
- Supports for finite **and** infinite dimensional problems



Why use QUESO?



Example 1: convection-diffusion

We are given a convection-diffusion model

$$(uc)_x - (\nu c_x)_x = s, \quad x \in [0,1],$$
$$c(0) = c(1) = 0.$$

Functions of x are: u, c and s.

Constants are: ν (viscosity).

The unkown is c, typically concentration.

The underlying convection velocity is u.

The forward problem: Given u and s, find c.

Example 1: convection-diffusion

We are also given observations

$$\text{model} \begin{cases} (uc)_x - (\nu c_x)_x = s, & x \in [0,1], \\ c(0) = c(1) = 0. \end{cases}$$

$$\text{observations} \begin{cases} y_j = c(x_j) + \eta_j, & \eta_j \overset{\text{i.i.d}}{\sim} \mathcal{N}(0, \sigma^2), \\ \rightsquigarrow y = \mathcal{G}(u) + \eta, & \eta \sim \mathcal{N}(0, \sigma^2 I). \end{cases}$$

The observations are of c. We wish to learn about u.

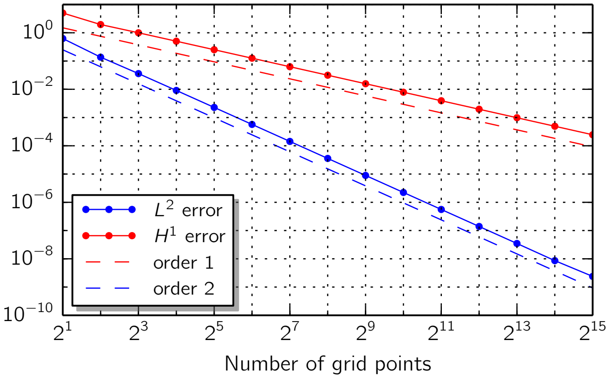
We will use Bayes's theorem:

$$\mathbb{P}(u|y) \propto \mathbb{P}(y|u)\mathbb{P}(u)$$

True $u = 1 - \cos(2\pi x)$
True $s = 2\pi(1 - \cos(2\pi x))\cos(2\pi x) + 2\pi \sin^2(2\pi x) + 4\pi^2 \nu \sin(2\pi x)$

Example 1: convection-diffusion

How do we know we are solving the right PDE ($\mathcal{G}$) to begin with?



Note: Use the `MASA` [1] library to **verify** your forward problem.

[1] Malaya et al., MASA: a library for verification using manufactured and analytical solutions, Engineering with Computers (2012)

Example 1: convection-diffusion

Recap Bayes's theorem,

$$\mathbb{P}(u|y) \propto \mathbb{P}(y|u)\mathbb{P}(u).$$

Remember, we don't know u but have observations and model:

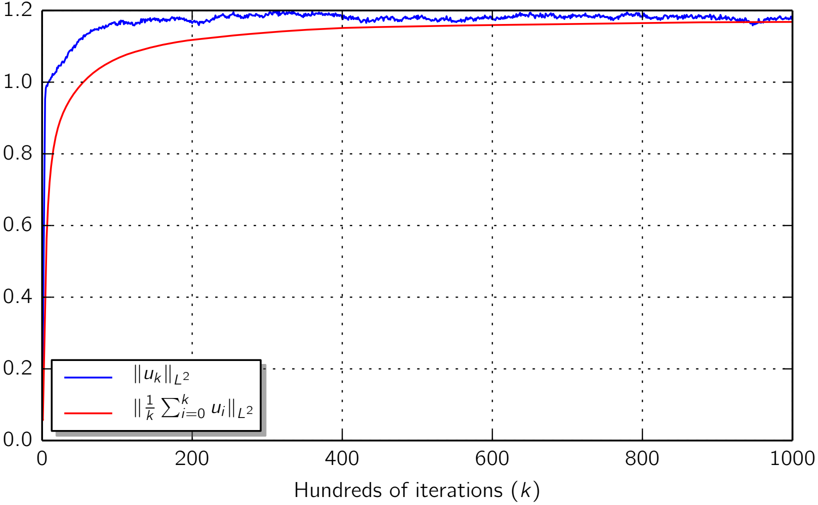
$$y = \mathcal{G}(u) + \eta, \quad \eta \sim \mathcal{N}(0, \sigma^2 I).$$

We also need a prior on u

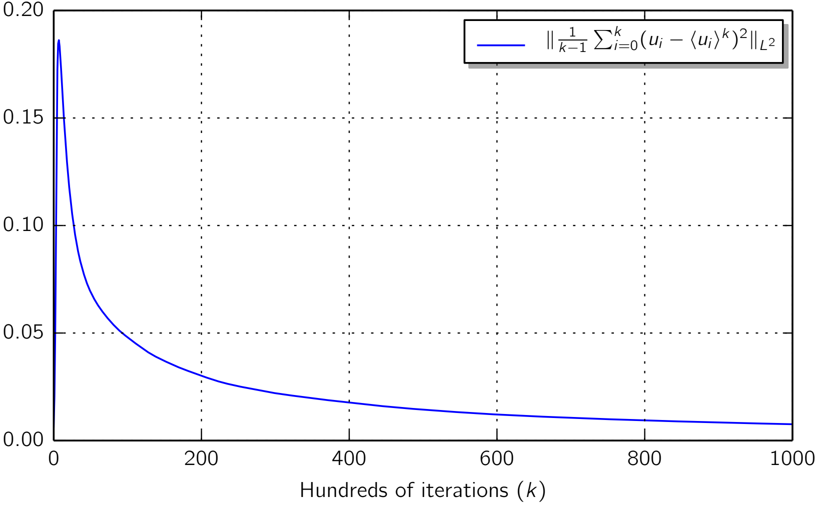
$$\mathbb{P}(u) = \mathcal{N}(0, (-\Delta)^{-\alpha}).$$

Aim is to get information from the posterior.

Example 1: convection-diffusion



Example 1: convection-diffusion

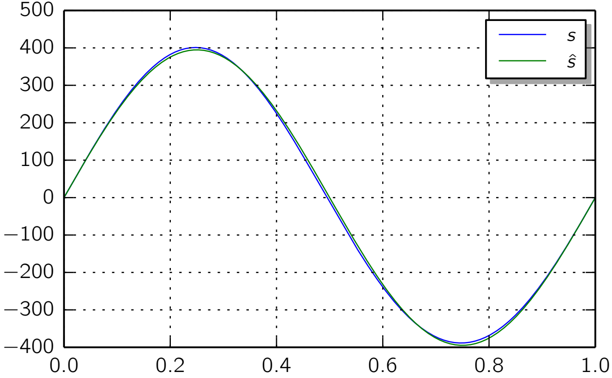


Example 1: convection-diffusion

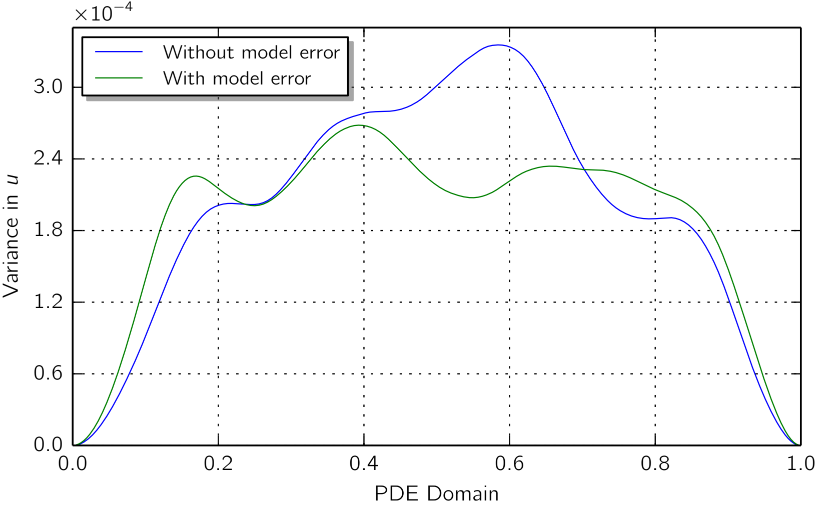
Suppose we got the source term wrong:

$$s = 2\pi(1 - \cos(2\pi x))\cos(2\pi x) + 2\pi \sin^2(2\pi x) + 4\pi^2\nu \sin(2\pi x)$$

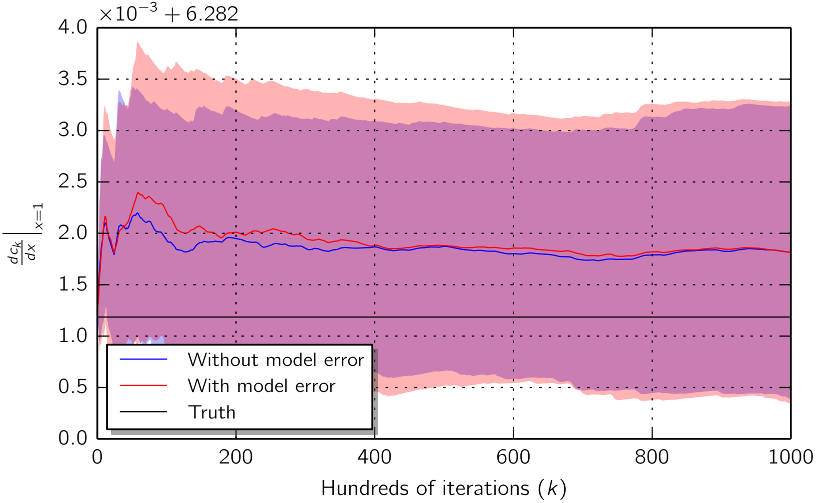
$$\hat{s} = 4\pi^2\nu \sin(2\pi x)$$



Example 1: convection-diffusion



Example 1: convection-diffusion



Example 2: Teleseismic earthquake model

$\mathcal{G}$ computes teleseismic earthquake wave phases P and SH

θ are rupture constraint parameters

$$\theta = (\text{slip magnitude}, \text{slip direction}, \text{start time}, \text{rise time}) \in \mathbb{R}^4$$

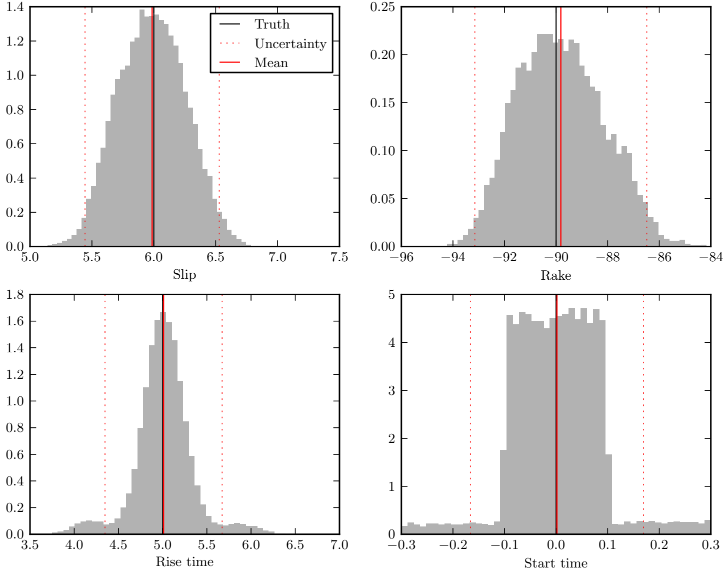
Posterior $\mathbb{P}(\theta|y)$ is a density on a four dimensional space

Observations y are of the produced waveform, with noise

$$\begin{aligned} & y_k = \mathcal{G}_k(\theta) + \eta_k, \quad \eta_k \overset{\text{i.i.d}}{\sim} \mathcal{N}(0, \sigma^2) \\ \rightsquigarrow \quad & y = \mathcal{G}(\theta) + \eta, \quad \eta \sim \mathcal{N}(0, \sigma^2 I) \end{aligned}$$

Prior $\mathbb{P}(\theta)$ is a uniform distribution on $\mathbb{R}^4$ (improper)

Example 2: Kikuchi and Kanamori model



Summary

- Regularised optimisation $\Leftrightarrow$ Bayesian inversion
  - $\therefore$ Bayesian inversion is not scary
- Uncertainty quantification is crucial; prediction
- Wealth of methods; pick your poison
  - My go-to is MCMC, but a different method may suit you better
- Predictive validation
  - The role of experiments and their effect on prediction
  - There is a framework for this (Moser, Oliver, Terejanu, Simmons)
- I'll be at SIAM CSE

Questions?