Introduction to Uncertainty Quantification and Gaussian Processes

Jeremy Oakley

School of Mathematics and Statistics, University of Sheffield

The University Of Sheffield.

jeremy-oakley.staff.shef.ac.uk

Uncertainty Quantification (UQ)

- Statistical problems associated with use of complex mathematical/mechanistic/'computer' models
- An active area of research in the Statistics community since the 1980s
  - Design and Analysis of Computer Experiments
  - Bayesian Analysis of Computer Code Outputs
- More recent interest from the Applied Maths community
- Many analyses involve the use of Gaussian processes

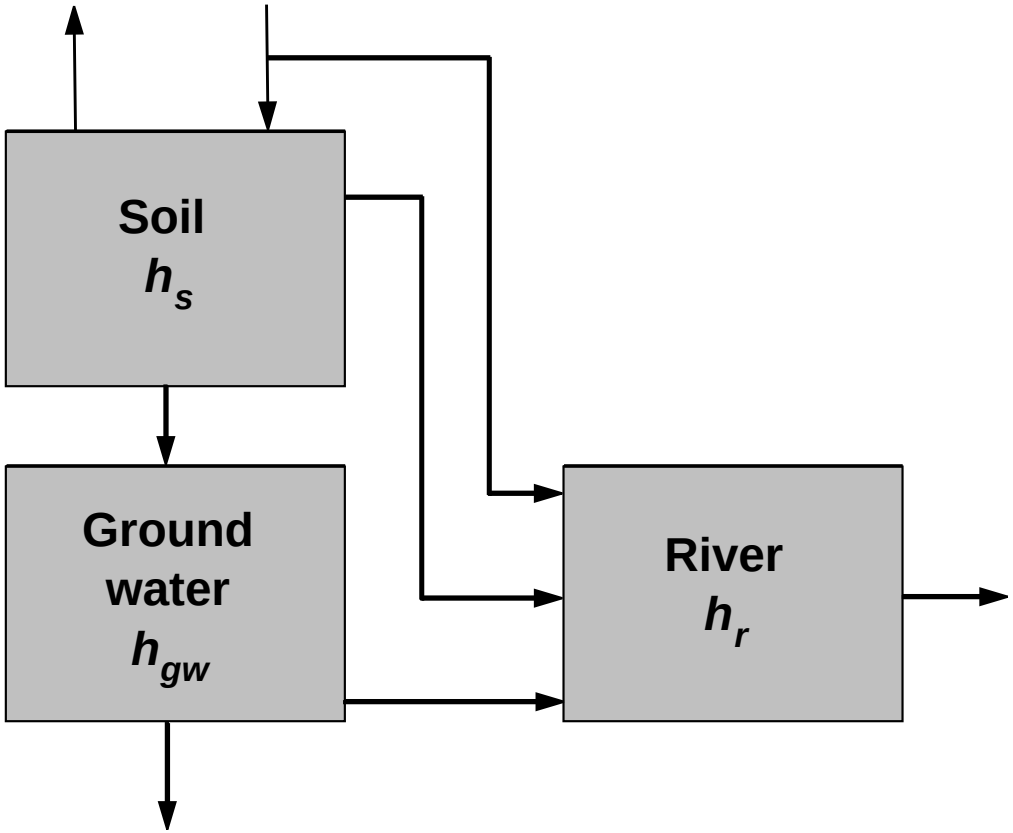
Computer models

- Computer model: function f with inputs x and outputs y

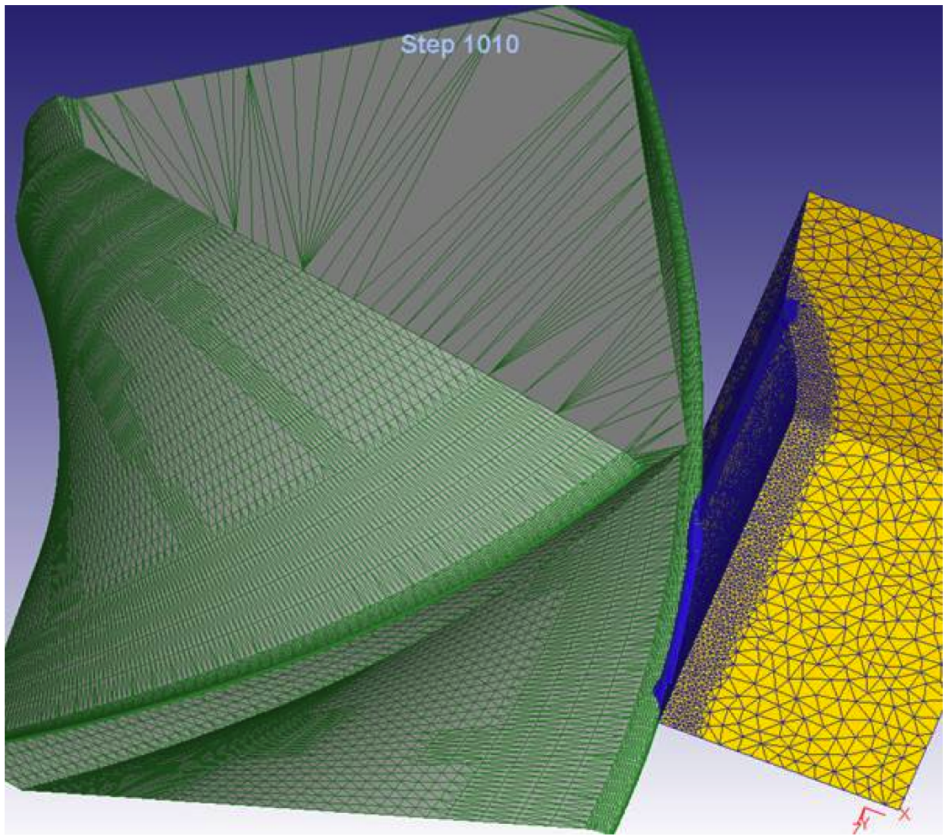
$$y = f(x).$$

- Refer to computer model as a *simulator*
- f usually not available in closed form.
- f constructed from modeller's understanding of the process.
  - There may be no physical input-output data.
- f may be deterministic.
- Computer experiment: evaluating f at difference choices of x
  - A 'simulator run': evaluating f at a single choice of x.

Example: rainfall run-off simulator

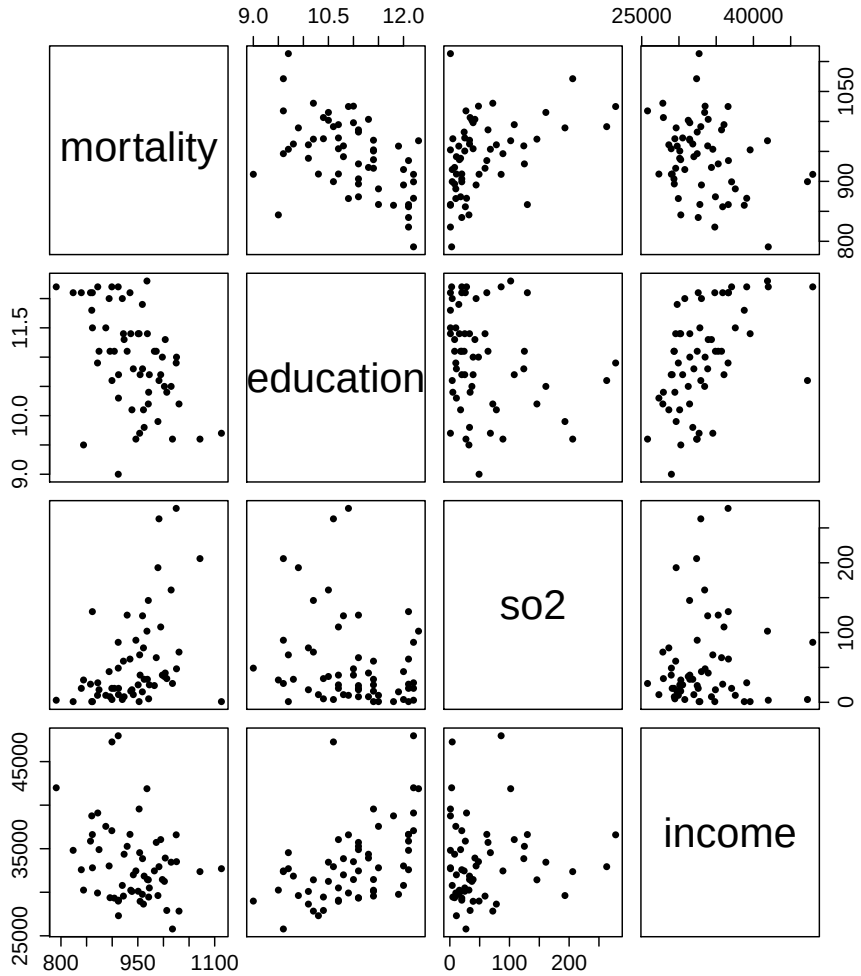


Example: finite element modelling



- simulator of machining of a metal
- Inputs: tool parameters, cutting parameters and material properties
- Outputs: forces and temperatures at various locations

Not concerned with (purely) statistical models



$$\text{mortality}_i = \beta_0 + \beta_1 \times \log(\text{so2})_i + \beta_2 \times \text{income} + \beta_3 \times \text{education} + \varepsilon_i$$

Uncertainty in simulator inputs

- Simulator may be set up to accept 'controllable' inputs only.
- But there may be other parameters/coefficients/variables 'hard-wired' within the simulator.
- We define the input x to include these other numerical values used to calculate the outputs.
- Suppose that there is a true input value, X, with at least some elements of X uncertain.
- What is our uncertainty about $Y = f(X)$?
- We quantify uncertainty about X with a probability distribution p_X
- Then need to obtain the distribution p_Y.
- Can propagate uncertainty using Monte Carlo: sample $X_1, \ldots, X_N$ from p_X and evaluate $f(X_1), \ldots, f(X_N)$
- What do we do if f is computationally expensive?(Gaussian processes!)

Probabilistic sensitivity analysis of simulator outputs

- Interested in $Y = f(\mathbf{X})$, where $\mathbf{X}$ is uncertain with (known) distribution $p_{\mathbf{X}}$.
- Sensitivity analysis: which elements in $\mathbf{X} = \{X_1, \ldots, X_d\}$ are most responsible for the uncertainty in $Y = f(\mathbf{X})$?
- Write $\mathbf{X} = (X_i, \mathbf{X}_{-i})$. Consider 'importance' of X_i via

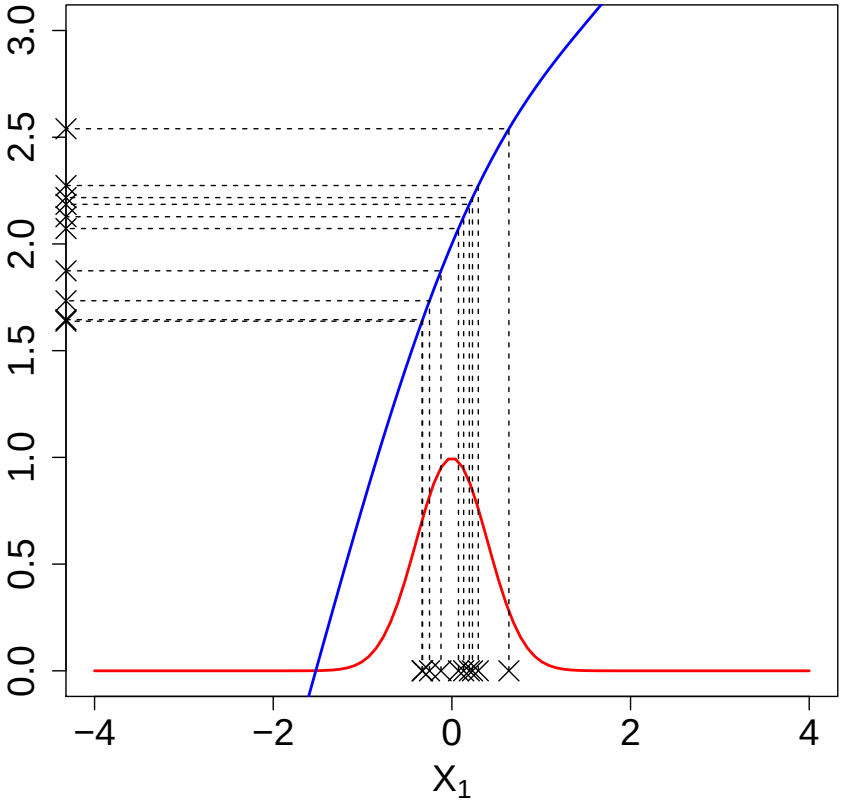
$$Var_{X_i}\{E_{\mathbf{X}_{-i}}(Y|X_i)\}$$

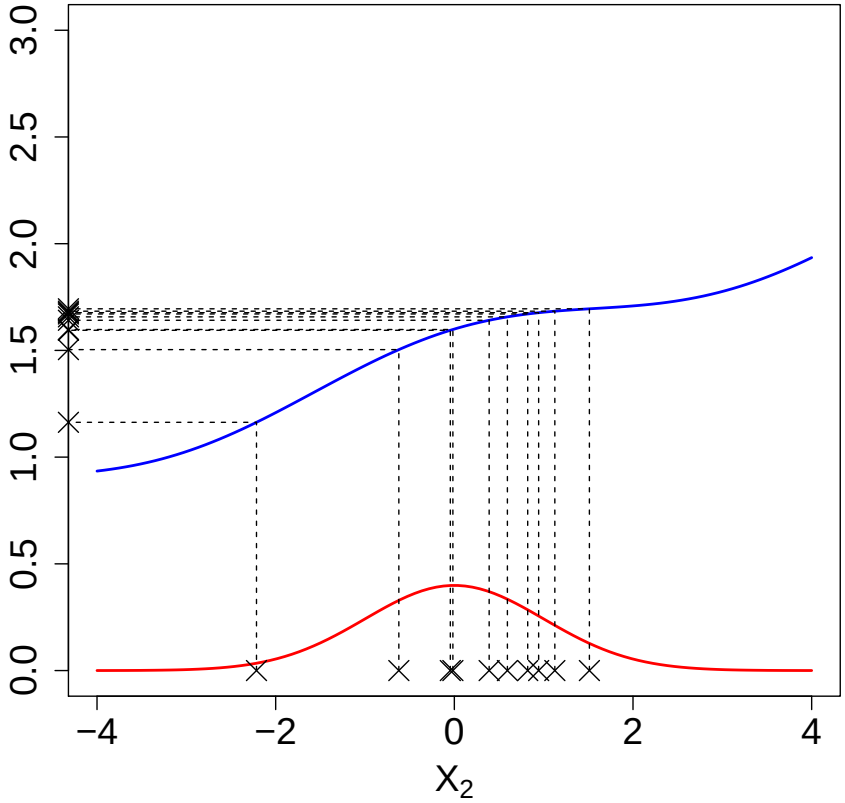
- The expected reduction in variance if value of X_i is learnt, because

$$Var(Y) = Var_{X_i}\{E_{\mathbf{X}_{-i}}(Y|X_i)\} + E_{X_i}\{Var_{\mathbf{X}_{-i}}(Y|X_i)\}$$

- Can speed up computation with Gaussian processes...
- ...but (sometimes) other methods are better!

Variance-based sensitivity analysis





— $p_{X_1}(x_1)$

— $E(Y|X_1 = x_1)$

— $p_{X_2}(x_2)$

— $E(Y|X_2 = x_2)$

Need to think carefully about input distributions

Consider

$$f(x) = \exp(-x),$$

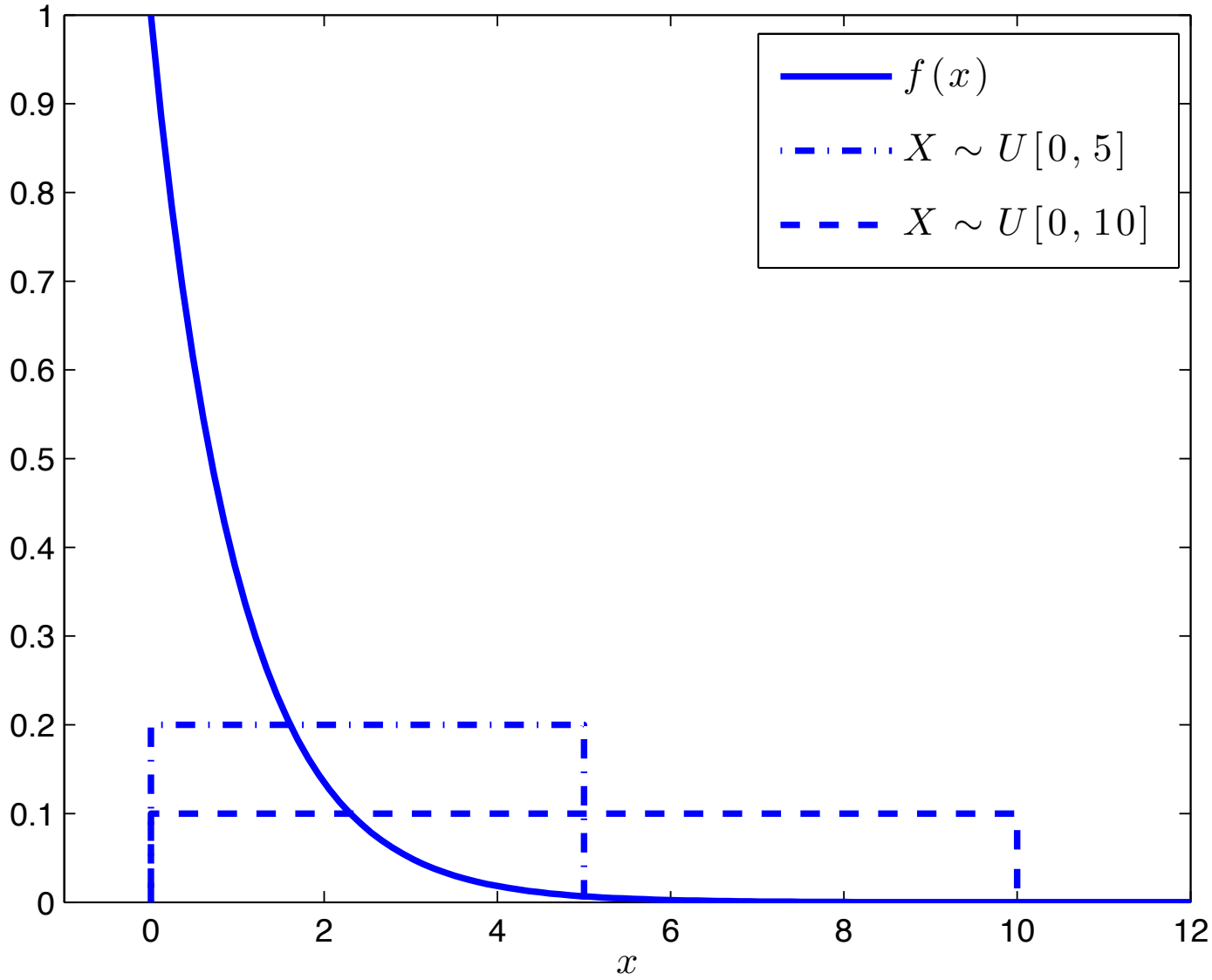
with $Y = f(X)$ and

$$X \sim U[0, b].$$

In this case we have

$$Var(Y) = \frac{b - 2 + 4\exp(-b) - (b+2)\exp(-2b)}{2b^2},$$

Increasing b increases the variance of X but *decreases* the variance of Y


1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
2
4
6
8
10
12
x
$f(x)$
$X \sim U[0, 5]$
$X \sim U[0, 10]$

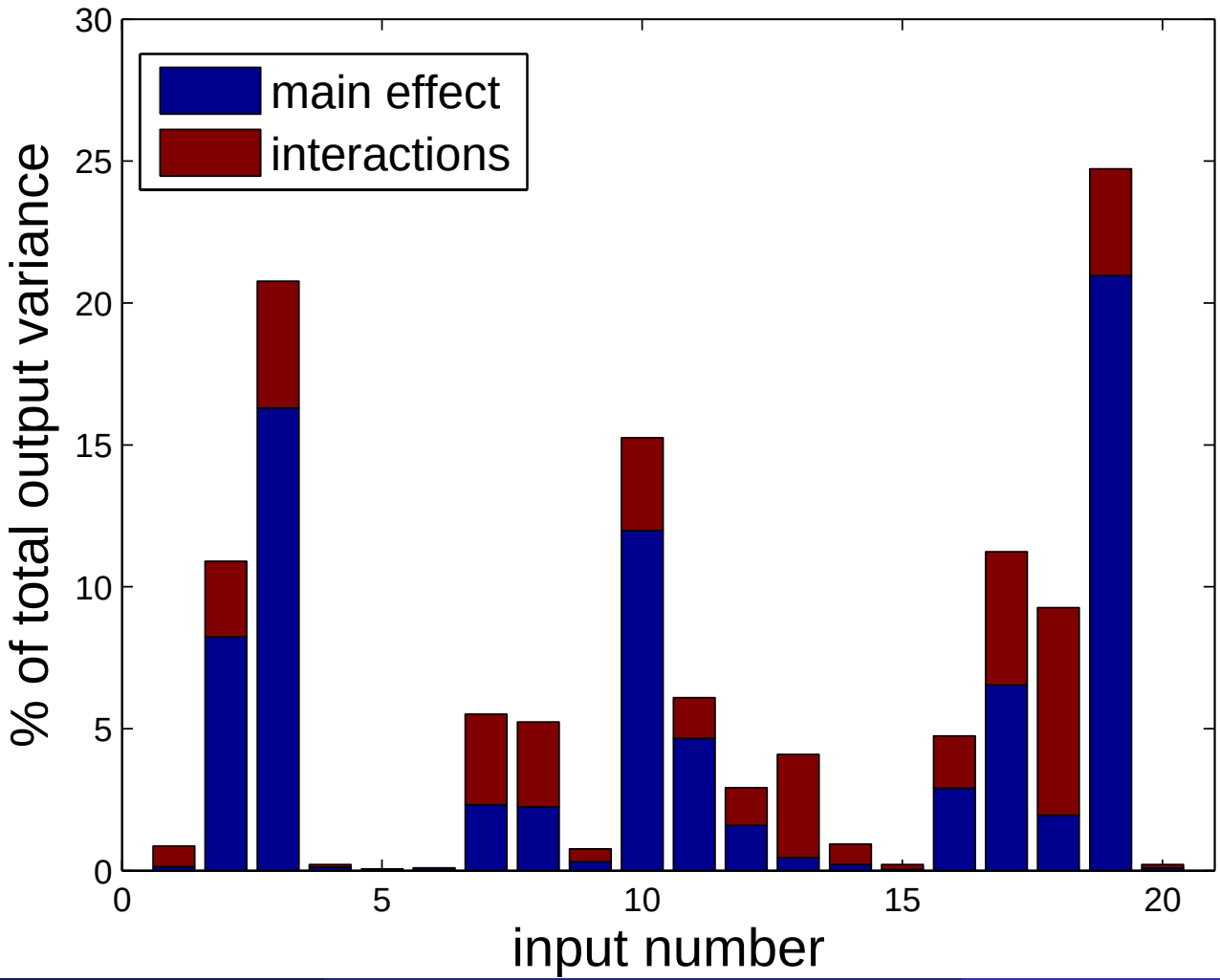
Example[1]: modelling Rotavirus

- Simulator developed by GlaxoSmithKline. Predicts incidence of rotavirus in a population before and after a vaccine is administered to a proportion of the infant population
- Deterministic compartmental model, 672 compartments (16 disease stages $\times$ 42 age classes)
- Inputs include transmission rates between age groups, reduction in risk following each infection
- Outputs: time series of rotavirus incidence for six age groups following vaccination programme
- GSK analysis investigated sensitivity of output to 9 inputs, using 8200 simulator runs
- We consider sensitivity of output to 20 inputs, using 340 simulator runs

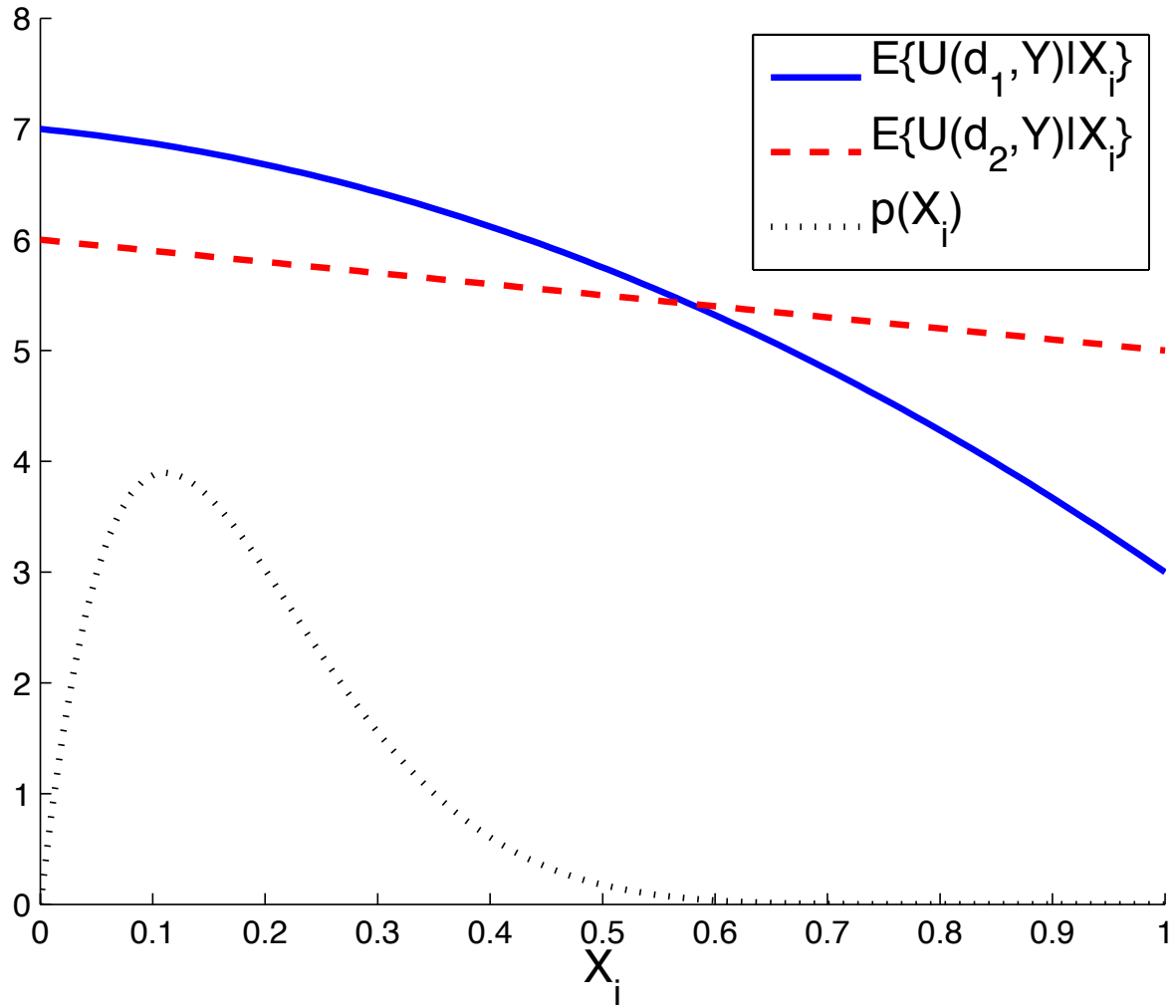
[1] MUCM case study: analysis by John Paul Gosling, Hugo Maruri-Aguilar, Alexis Boukouvalas

Variance based sensitivity analysis

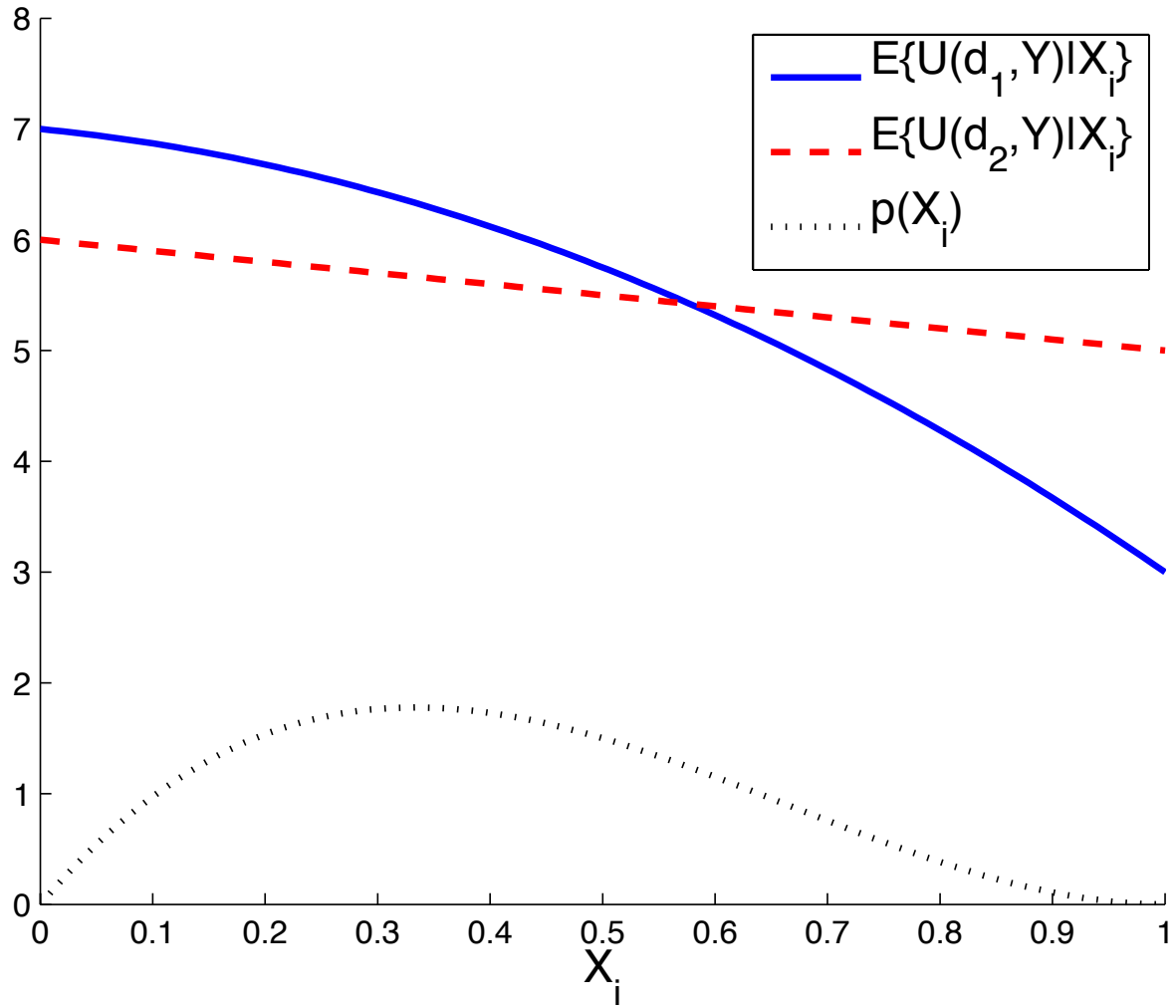
Analysis for an individual output: no. of infections in 2-3 age group after 2 years



Decision-theoretic sensitivity analysis



Decision-theoretic sensitivity analysis



A quick plug

- Thursday morning: workshop on sensitivity analysis
- Theory, computation, brief intro to eliciting input distributions
- Practical using R
- But no Gaussian processes!

Calibration/ inverse problems

Example problem from Kennedy and O'Hagan (2001)

- A Gaussian plume deposition simulator $f(x_{cont}, x_{calib})$ predicts deposition of radionuclides at a location x_{cont} following release of unknown concentration X_{calib} from point source
- Measurements of the true deposition $z(x_{cont})$ at a limited number of locations x_{cont} available.
- Aim: to predict deposition at other locations using both data and simulator.
- What value of x_{calib} do we use?
- And what happens if the simulator is wrong?

Calibrating an imperfect simulator

- Wish to estimate $X_{calib} = g$: acceleration due to Earth's gravity
- I drop a tennis ball from my office window at height x_{cont}, and time its descent to the ground
- Estimate g via

$$t = \sqrt{2x_{cont}g}$$

- Will have error in measurements, so take replicates
- The more measurements I take, the more certain I become about the *wrong value*

density

The calibration model

- Kennedy and O'Hagan (2001)

$$z(x_{cont,i}) = \rho f(x_{cont,i}, X_{calib}) + \delta(x_{cont,i}) + \varepsilon_i$$

  $\delta(x_{cont,i})$ is the **discrepancy** (bias) between simulator output and reality.

- δ modelled as a Gaussian process
- Doesn't always go down well with modellers!

  "I'm horrified! You should be improving your models with better physics!"

- Accounting for simulator discrepancy important, otherwise
  - can become certain about a 'wrong' input value
  - simulator predictions can be spuriously precise

Calibration inputs

$$z(x_{cont,i}) = \rho f(x_{cont,i}, X_{calib}) + \delta(x_{cont,i}) + \varepsilon_i$$

- Two types of input
  - Observable: has true value, independent of the simulator
  - Tuning input: artefact of the simulator. 'Best value' rather than 'true value'
- Kennedy & O'Hagan model good for interpolating physical observations...
- ...but problems if aim is to learn true values of 'observable' calibration inputs or extrapolate, even if allowance for simulator discrepancy δ

Brynjarsdottir, J. and O'Hagan, A. (2014). Learning about physical parameters: The importance of simulator discrepancy. Inverse Problems, 30.

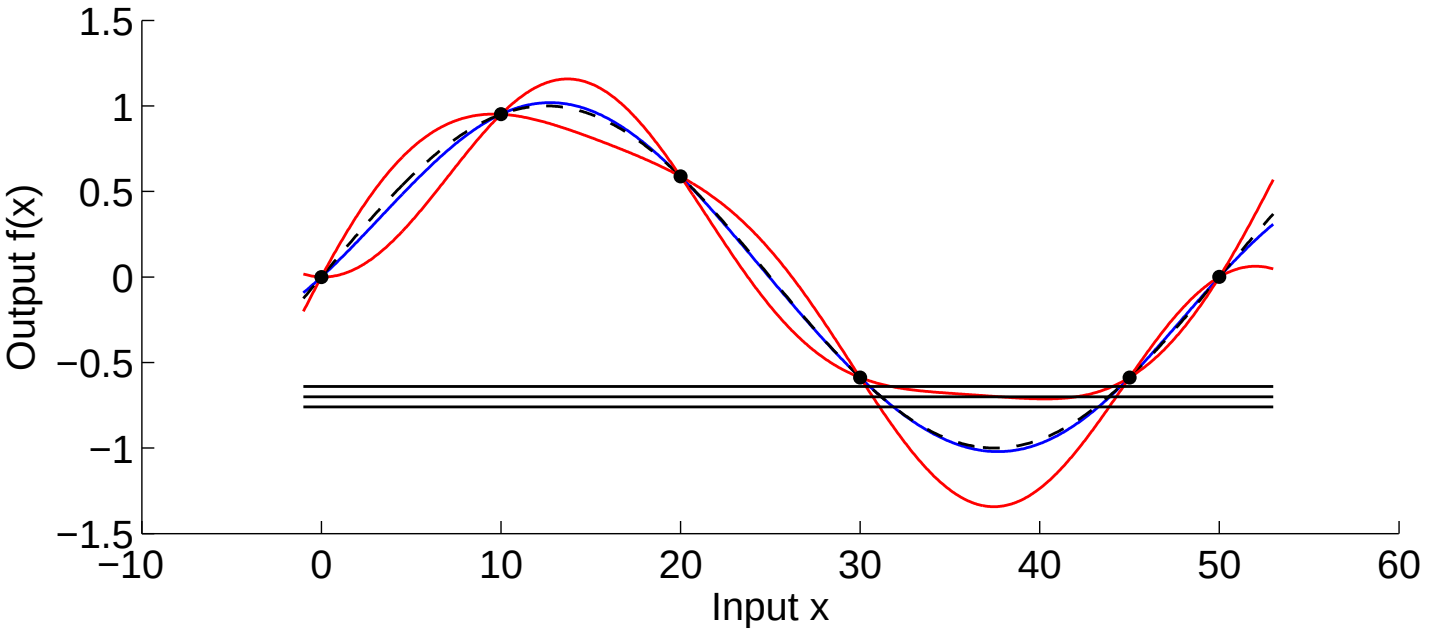
Calibration via history matching

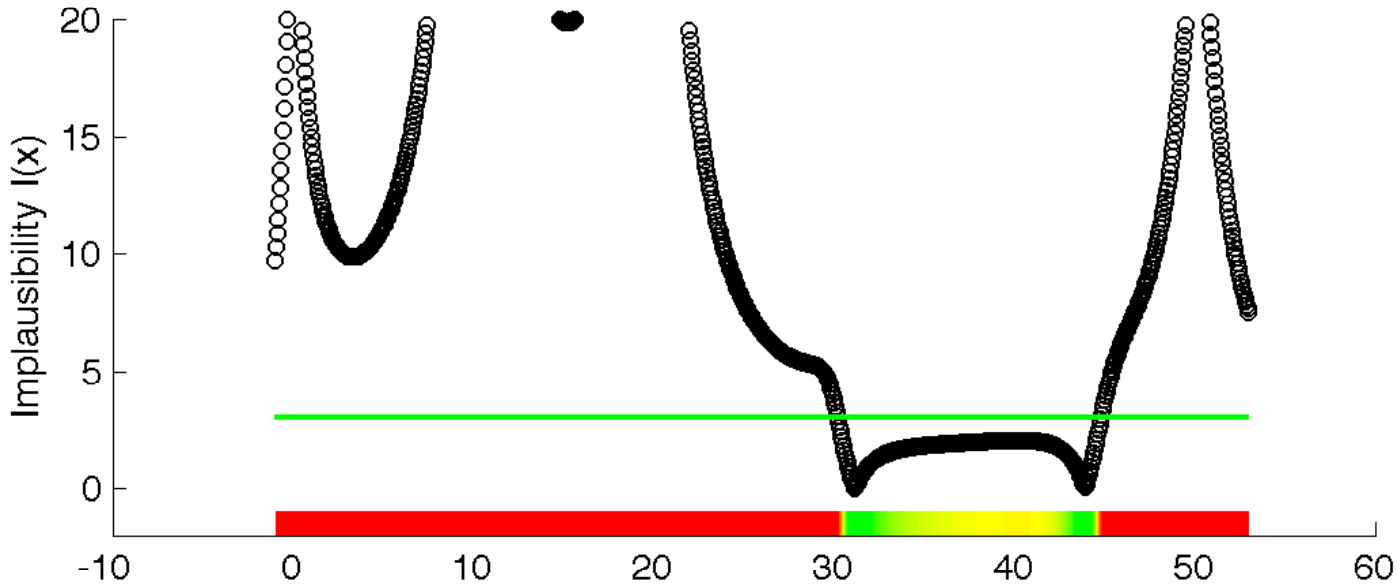
- The calibration problem again: try to find x such that $f(x)$ is 'close' to observation z
- Emphasis now on discarding region of input space where the simulator can't fit the data
- For computationally expensive simulator, will use Gaussian process emulator for f
- Assess the "implausibility" of an input value x via

$$I(x) = \frac{|z - E\{f(x)\}|}{[Var\{f(x)\} + Var(\varepsilon) + Var(\delta)]^{1/2}}.$$

- For multiple outputs, can consider maximum implausibility for each output

History matching: toy example



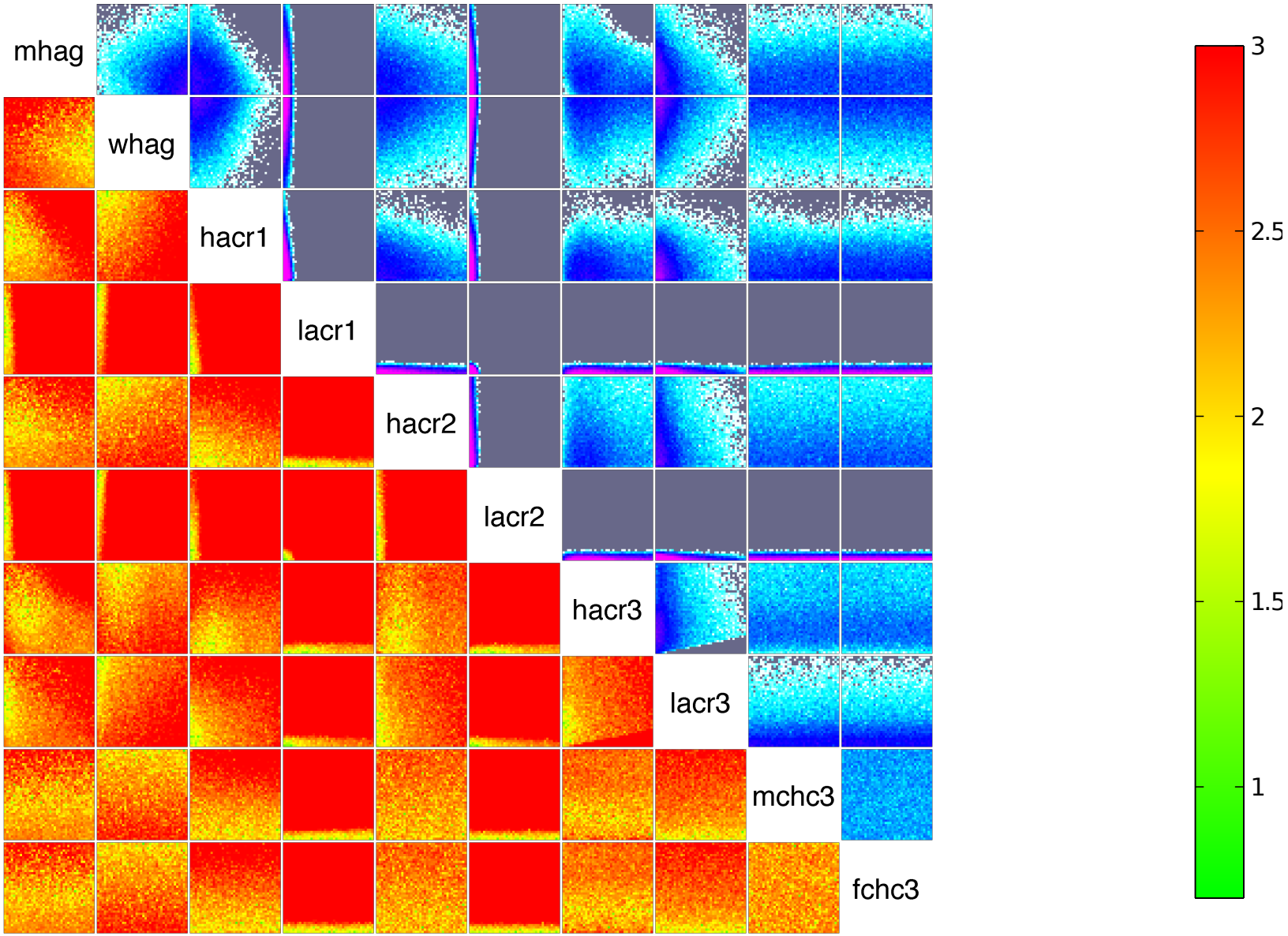


Case study[2]: HIV in Uganda

- "Mukwano": a dynamic, stochastic, individual based model that simulates sexual partnerships and HIV transmission
- Births, deaths, partnership formation and dissolution and HIV transmission were modelled using time-dependent rates
- 22 inputs, e.g. proportions of men and women in "high sexual activity" groups, transition probability of HIV per sex act during primary stage of infection
- Calibration data were collected from a rural general population cohort in South-West Uganda. The cohort was established in 1989 and currently consists of the residents of 25 villages
- History matching iterated through 10 waves, 200-500 simulator runs per wave

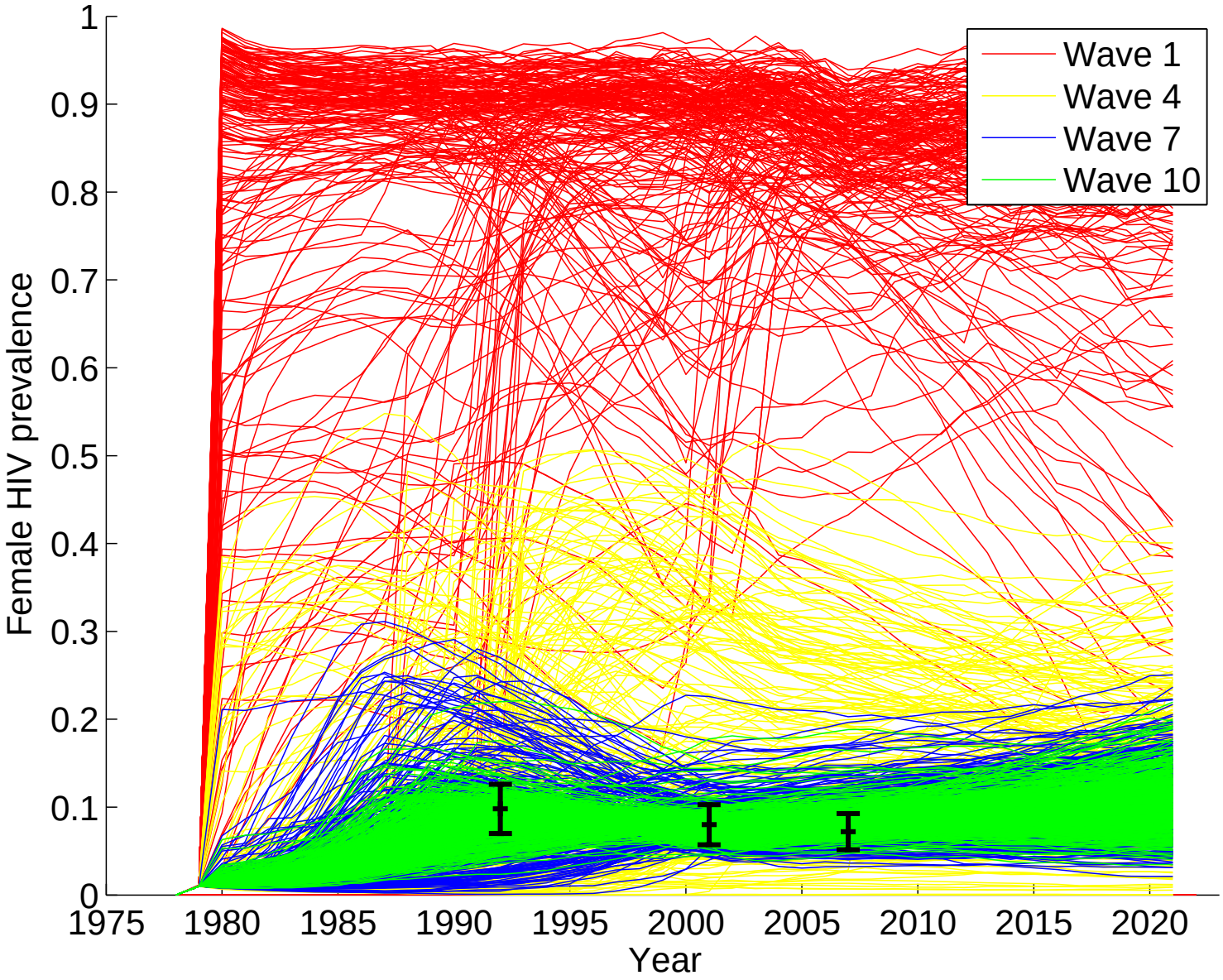
[2] I. Andrianakis, I. Vernon, N. McCreesh, T.J. McKinley, J.O., R. Nsubuga, M. Goldstein and R.G. White

Wave 1


mhag
whag
hacr1
lacr1
hacr2
lacr2
hacr3
lacr3
mchc3
fchc3
3
2.5
2
1.5
1

Wave 4

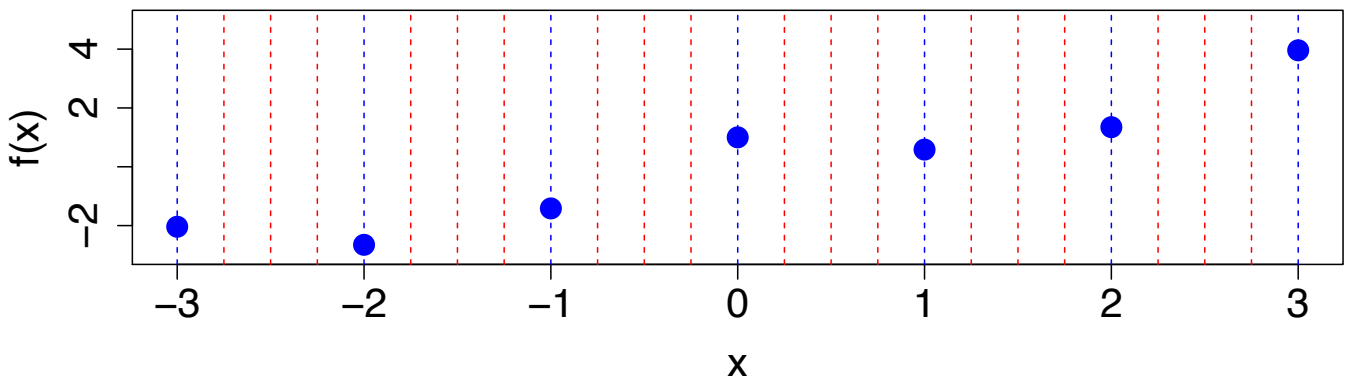
Simulator runs after history matching



Part 2: Gaussian processes for computer models

- Lots of interest in the UQ community, dating back to Sacks et al. (1989)
- GPs use to "emulate" computationally expensive simulators $y = f(x)$.
- The simulators f are often deterministic
- Emphasis on 'small data'
  - Motivation for using a GP is that we cannot obtain many simulator runs

Emulators

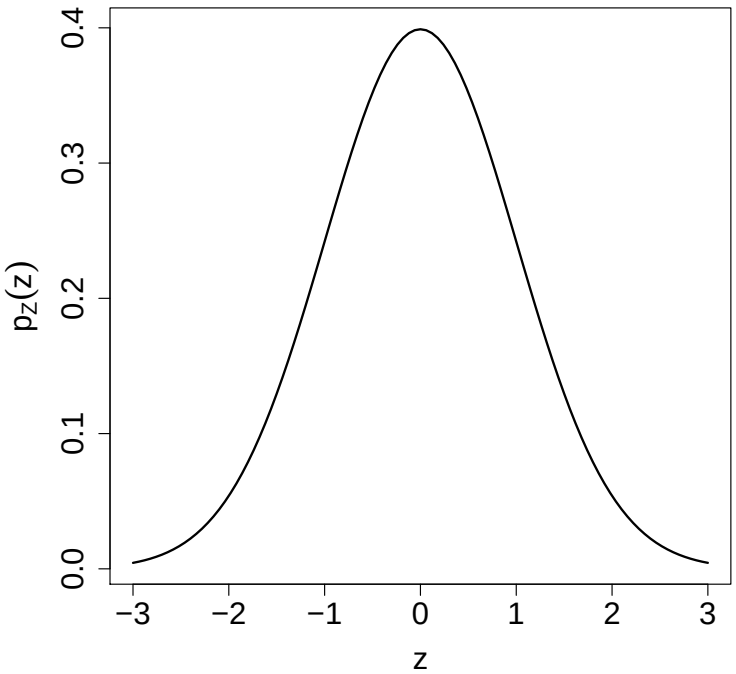


- Want $f(x_1), \ldots, f(x_N)$, but only know $f(x_1), \ldots, f(x_n)$, for $n << N$.
- Could estimate f given $f(x_1), \ldots, f(x_n)$
  - but can we quantify uncertainty in the estimate?
- A statistical inference problem:
  - Treat f as an *uncertain* function
  - Derive a probability distribution for f given $f(x_1), \ldots, f(x_n)$ (an "emulator")
  - Distribution represents a subjective judgement; there is no 'true' distribution for f.
- Popular choice of distribution for f: Gaussian process

Probability distributions for functions

Univariate normal (Gaussian) distribution $Z \sim \mathcal{N}(m, v)$

$$p_Z(z) = \frac{1}{\sqrt{2\pi v}} \exp\left\{-\frac{1}{2v}(z-m)^2\right\}$$

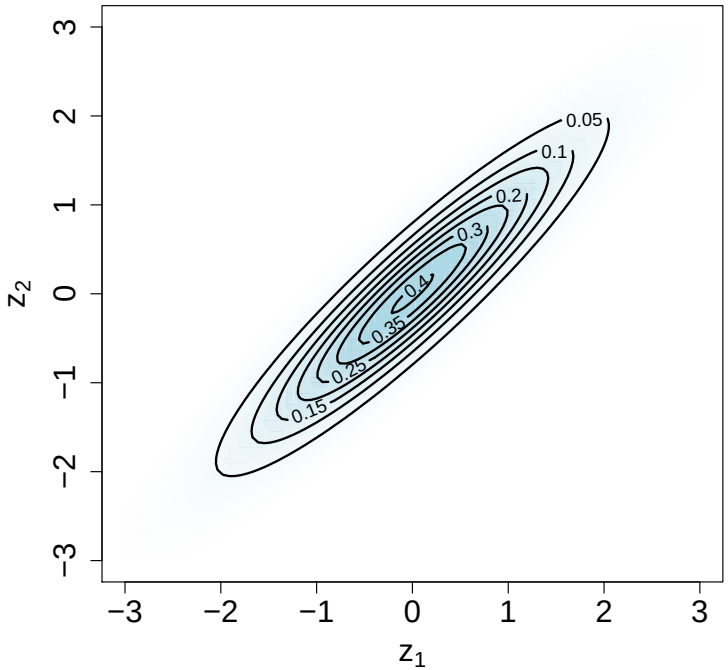


$$Z \sim \mathcal{N}(m = 0, v = 1).$$

Probability distributions for functions

Multivariate normal distribution $\boldsymbol{Z} \sim \mathcal{N}_d(\boldsymbol{m}, V)$

$$p_{\boldsymbol{Z}}(\boldsymbol{z}) = \frac{|V|^{-\frac{1}{2}}}{(2\pi)^{\frac{d}{2}}} \exp\left\{-\frac{1}{2}(\boldsymbol{z}-\boldsymbol{m})^T V^{-1}(\boldsymbol{z}-\boldsymbol{m})\right\}$$

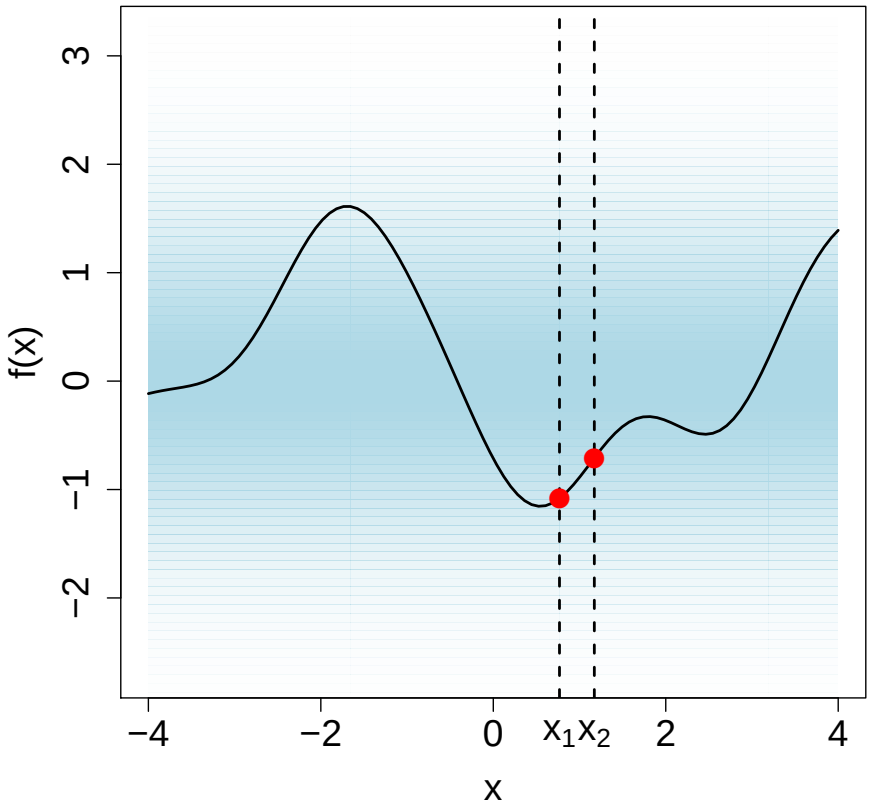


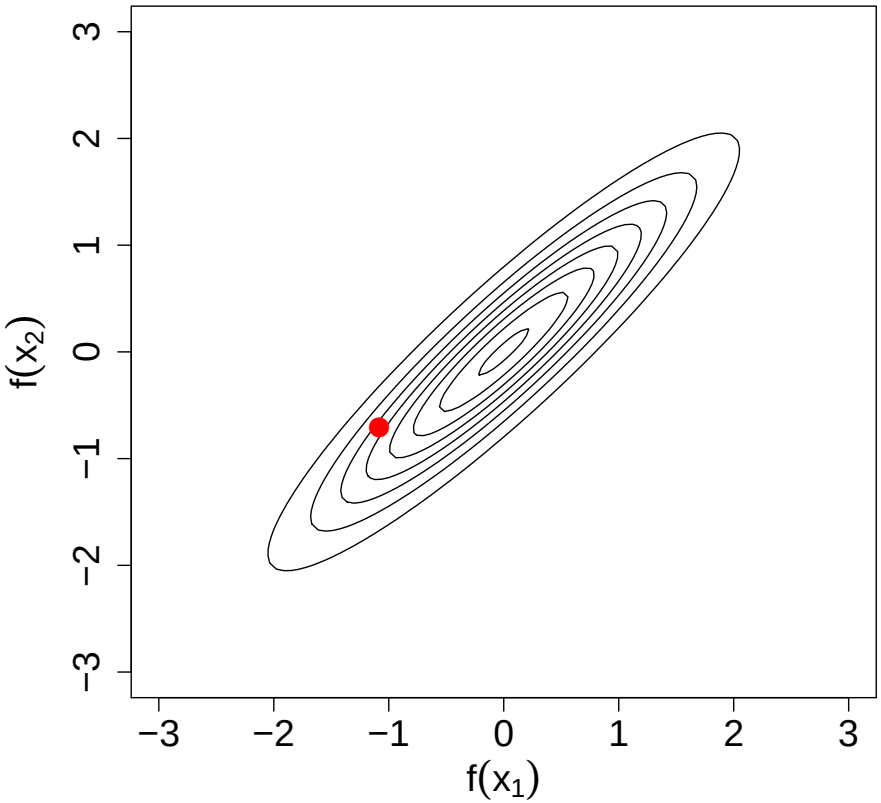
$$\begin{pmatrix} Z_1 \\ Z_2 \end{pmatrix} \sim \mathcal{N}_2\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & 0.9 \\ 0.9 & 1 \end{pmatrix}\right)$$

Probability distributions for functions

Gaussian process

Infinite set of random variables (e.g. $f(x_1), f(x_2), \ldots$) with the property that any finite subset have a multivariate normal distribution





Modelling a function as a Gaussian process

Argument alert!

- This treatment of GPs will *appear* to be different!
  - "The mean function"
- It isn't really!
- Bear with me...

Modelling a function as a Gaussian process

We write

$$f(x) = m(x) + Z(x)$$

- $m(.)$: the mean function - a parametric function of x, e.g.

$$m(x) = \sum_{i=0}^{p} \beta_i x^i,$$

  β_i uncertain (but can be integrated out of joint posterior analytically).
- $Z(.)$ a zero mean Gaussian process
- Gaussian assumption is for the *deviation* of $f(x)$ from $m(x)$.

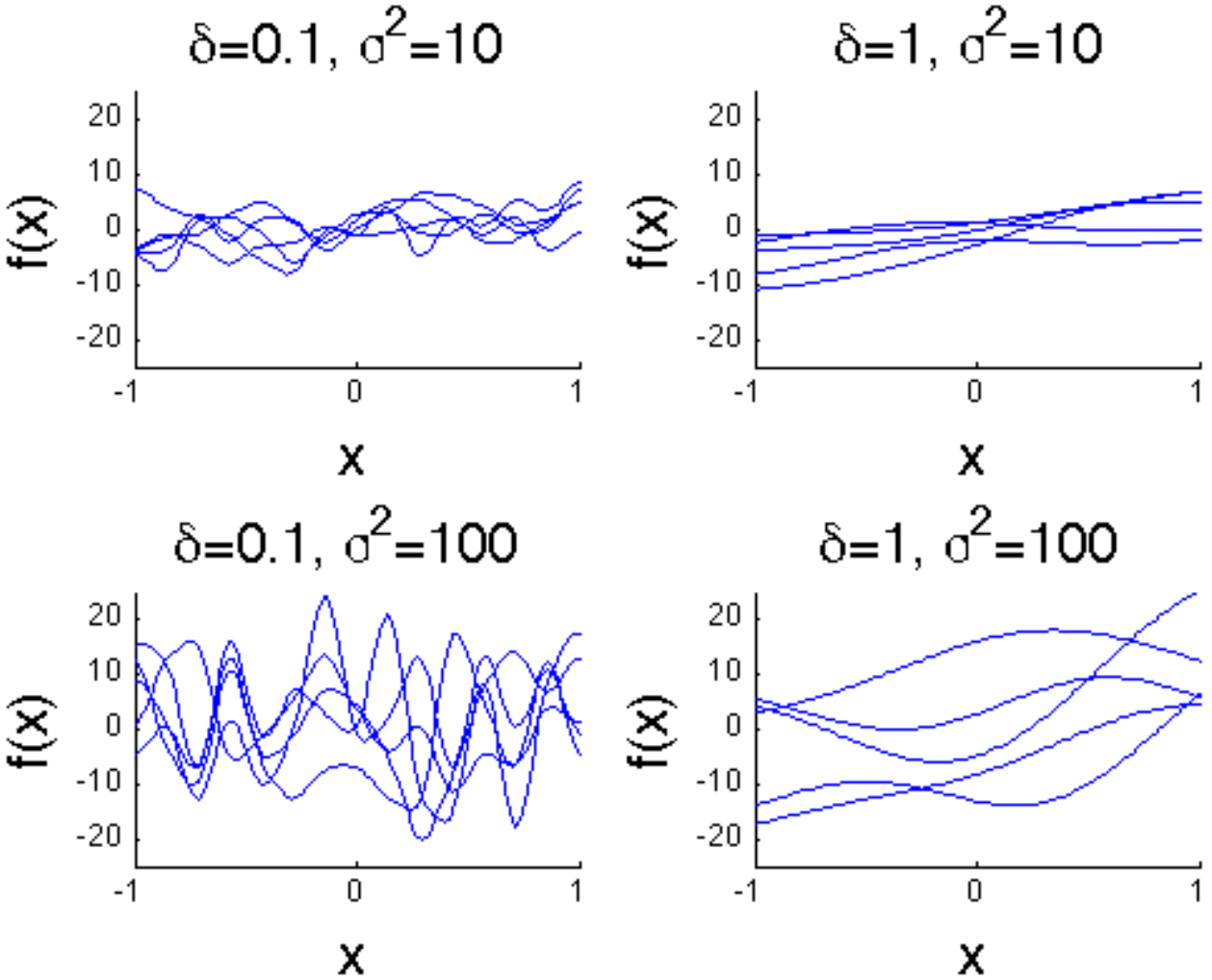
$$f(x) = m(x) + Z(x)$$

- Specify $Z(.)$ by its covariance function, eg

$$Cov\{f(x_i), f(x_j)\} = \sigma^2 \exp\left\{-\left(\frac{x_i - x_j}{\delta}\right)^2\right\}$$

- σ^2 determines how far the $f(x)$ deviates from $m(x)$.
- δ describes how 'wiggly' the function looks

Covariance function parameters



Equivalent ways to parameterise a GP

- Consider, for scalar input x,

$$f(x) = x\beta + Z(x)$$

with $\beta|\sigma^2 \sim N(0, v\sigma^2)$ (with v chosen), and $Z(x)$ a zero mean GP as before.

- Then

$$E[f(x)] = 0$$

and

$$Cov(f(x_1), f(x_2)|\sigma^2, \theta) = v\sigma^2 x_1 x_2 + \sigma^2 c_\theta(x_1, x_2) \qquad (1)$$

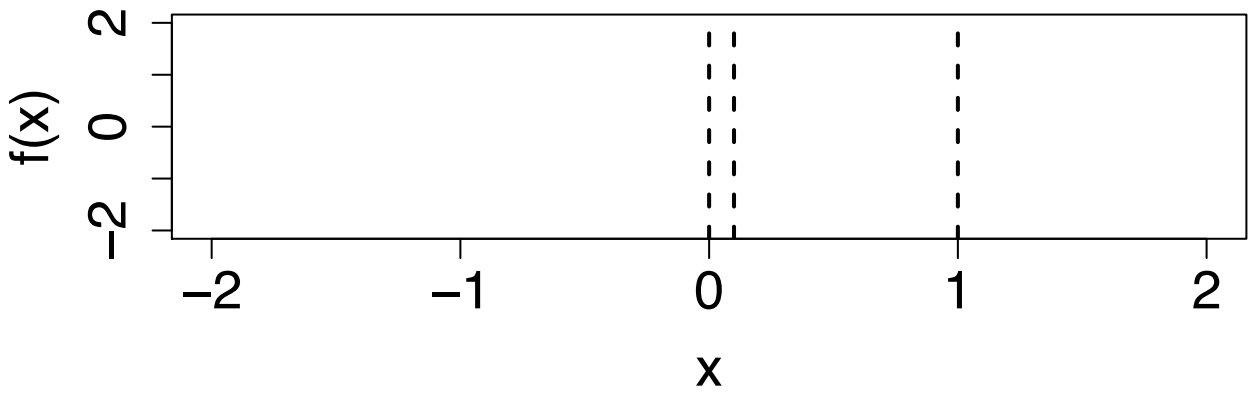
- If we *specify* a zero mean GP with covariance function (1), for fixed v, model is equivalent to the hierarchical GP with the mean function $x\beta$.

- For the most common modelling choices, GPs with "mean functions" can be re-written as zero-mean GPs with a modified covariance kernel
- Which way you write the GP isn't important...
- ...but choice of mean function/covariance kernel is!

Example

Suppose we have

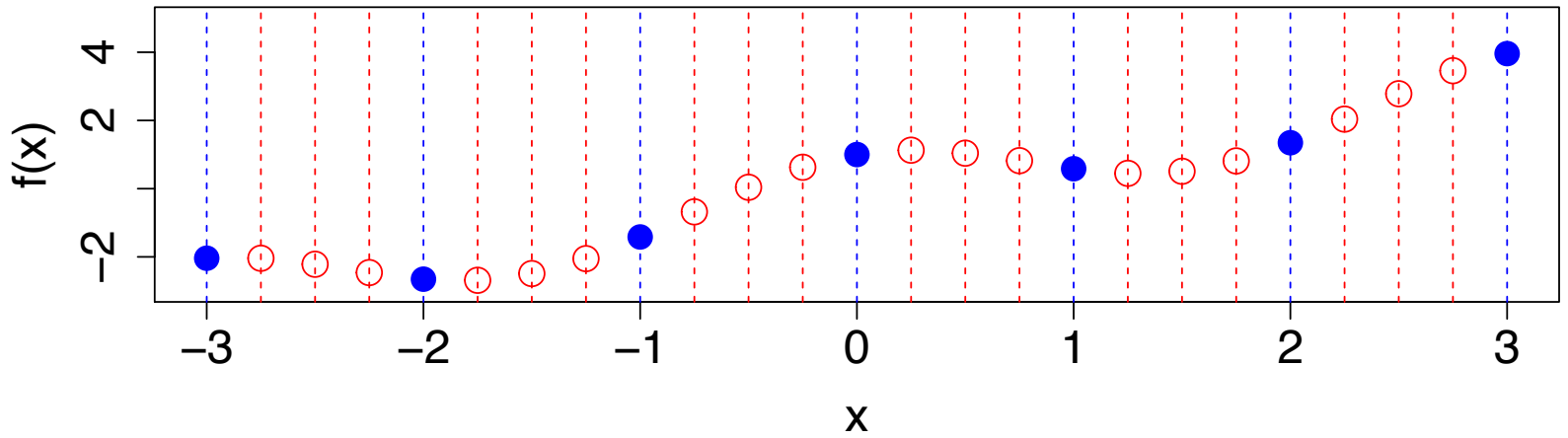
$$\begin{aligned} E\{f(x)\} = m(x) &= 1 + x, \\ Cov\{f(x_i), f(x_j)\} &= \exp\left\{-\left(\frac{x_i - x_j}{0.5}\right)^2\right\}. \end{aligned}$$



Then

$$\begin{pmatrix} f(0) \\ f(0.1) \\ f(1) \end{pmatrix} \sim \mathcal{N}_3 \left(\begin{pmatrix} 1 \\ 1.1 \\ 2 \end{pmatrix}, \begin{pmatrix} 1 & 0.96 & 0.02 \\ 0.96 & 1 & 0.04 \\ 0.02 & 0.04 & 1 \end{pmatrix} \right)$$

Why we like Gaussian processes for modelling functions

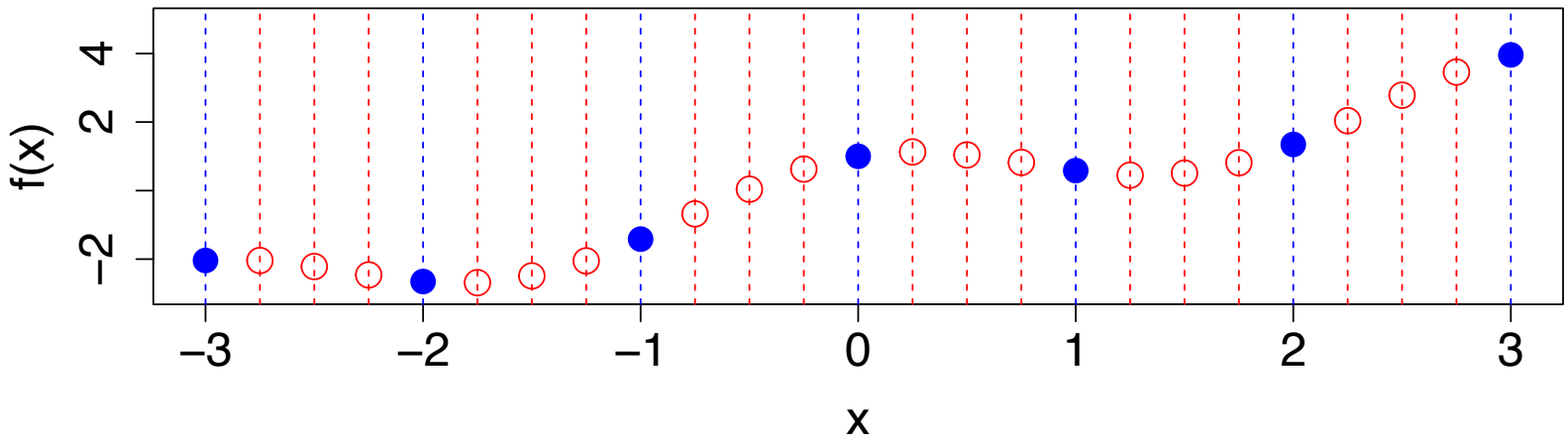


- We have observed $f(x_1), \ldots, f(x_n)$
- We want to know $f(x_{n+1}), \ldots, f(x_N)$
- Define

$$\boldsymbol{y} = (f(x_1), \ldots, f(x_n), f(x_{n+1}), \ldots, f(x_N))^T = (\boldsymbol{y}_1^T, \boldsymbol{y}_2^T)^T.$$

Represent uncertainty about $f(.)$ using a GP, so

$$\boldsymbol{y} \sim \mathcal{N}_N(\boldsymbol{m}, V).$$



We have

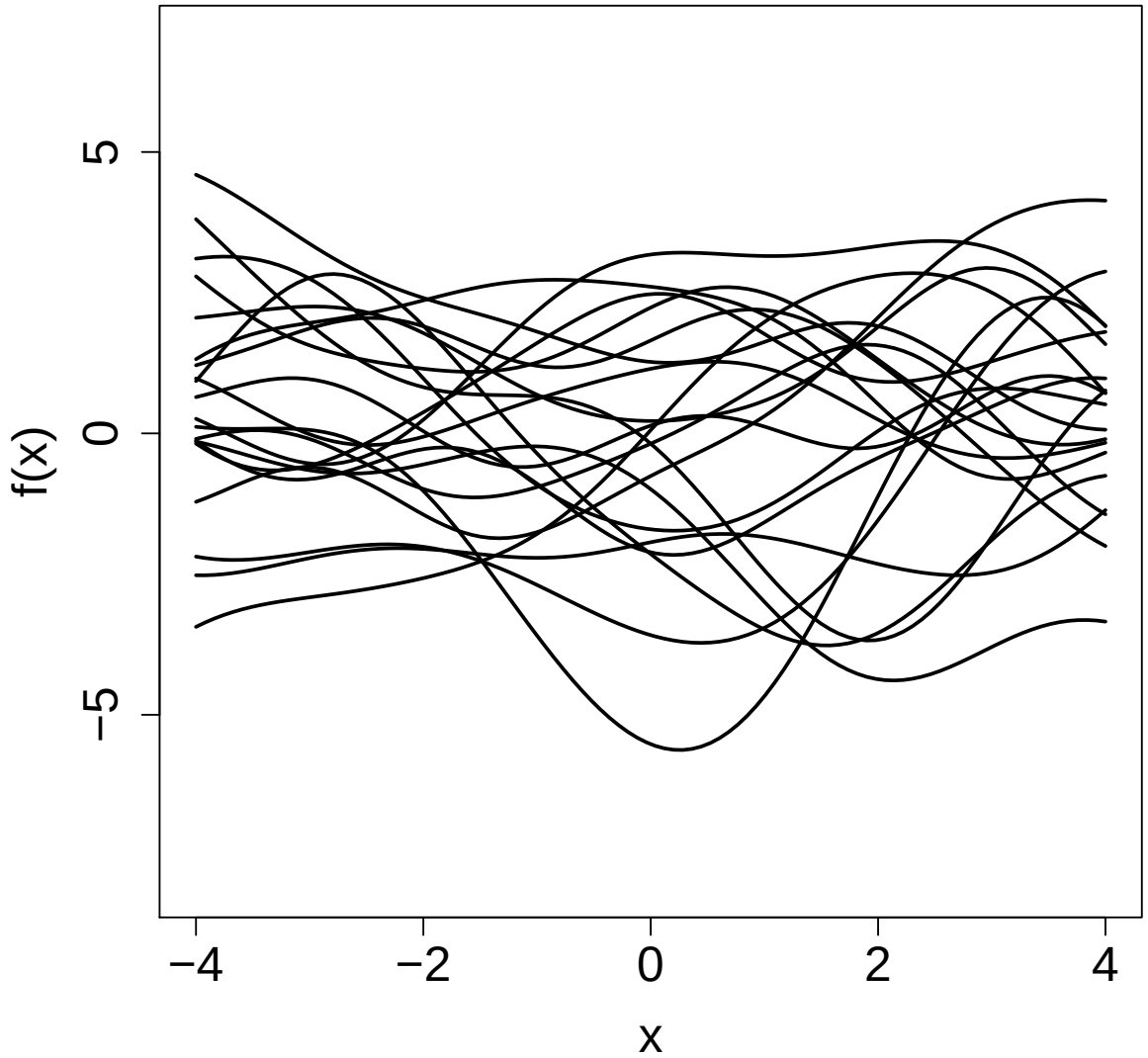
$$\boldsymbol{y} = \begin{pmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \end{pmatrix} \sim \mathcal{N}_N \left(\begin{pmatrix} \boldsymbol{m}_1 \\ \boldsymbol{m}_2 \end{pmatrix}, \begin{pmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{pmatrix} \right)$$

Then

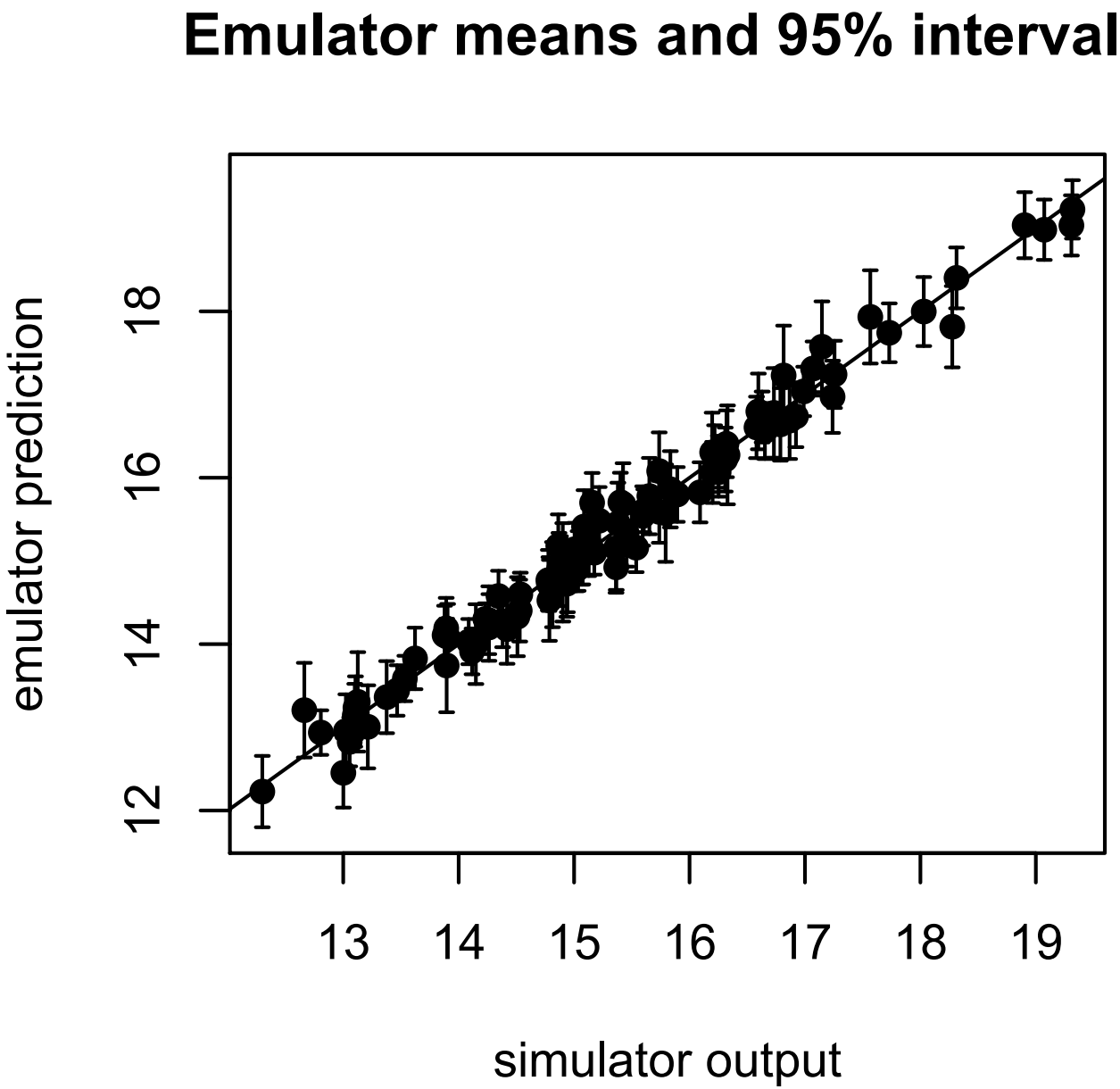
$$\begin{aligned} \boldsymbol{y}_2 | \boldsymbol{y}_1 &\sim \mathcal{N}_{N-n}(\boldsymbol{m}_2^*, V_{22}^*), \\ \boldsymbol{m}_2^* &= \boldsymbol{m}_2 + V_{21} V_{11}^{-1} (\boldsymbol{y}_1 - \boldsymbol{m}_1), \\ V_{22}^* &= V_{22} - V_{21} V_{11}^{-1} V_{12} \end{aligned}$$

'Easy' to predict (and quantify uncertainty about) $\boldsymbol{y}_2$ having observed $\boldsymbol{y}_1$.

Updating a Gaussian process emulator



Example: 18 input climate simulator, 255 simulator runs



Emulators: the benefits

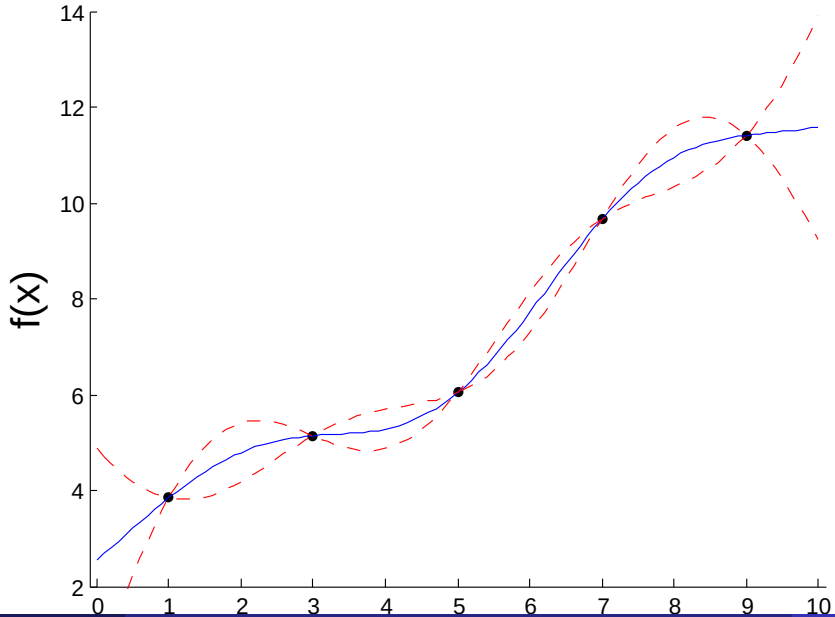
- simulator user runs the simulator as often as possible to get $f(x_1), \ldots, f(x_n)$
- n will be small if f is computationally expensive
- Emulator allows us to predict $f(x_{n+1}), f(x_{n+2}), \ldots$ at any other inputs, and quantify joint uncertainty in predictions, *almost instantaneously*
- Makes analyses requiring many simulator runs feasible, even if n relatively small
- Can derive other useful quantities (almost) for free:
  - We have an uncertain, true input X with probability distribution $p_X(x)$
  - Define $Y = f(X)$. Want to know

$$M := E(Y|f) = \int_{\mathcal{X}} f(x) p_X(x) dx$$

  - With Gaussian process emulator, M has a normal distribution, can derive expressions for mean and variance of M

The emulator does not *replace* the simulator

- In a computer experiment, may want to know $f(x_1), \ldots, f(x_N)$, but can only observe $f(x_1), \ldots, f(x_n)$, with $n < N$.
- As part of the analysis, we work with $p\{f(x_{n+1}), \ldots, f(x_N) | f(x_1), \ldots, f(x_n)\}$, which we get from the emulator.
- If the simulator has given us the value of $f(x)$, the emulator will give us the *same* value

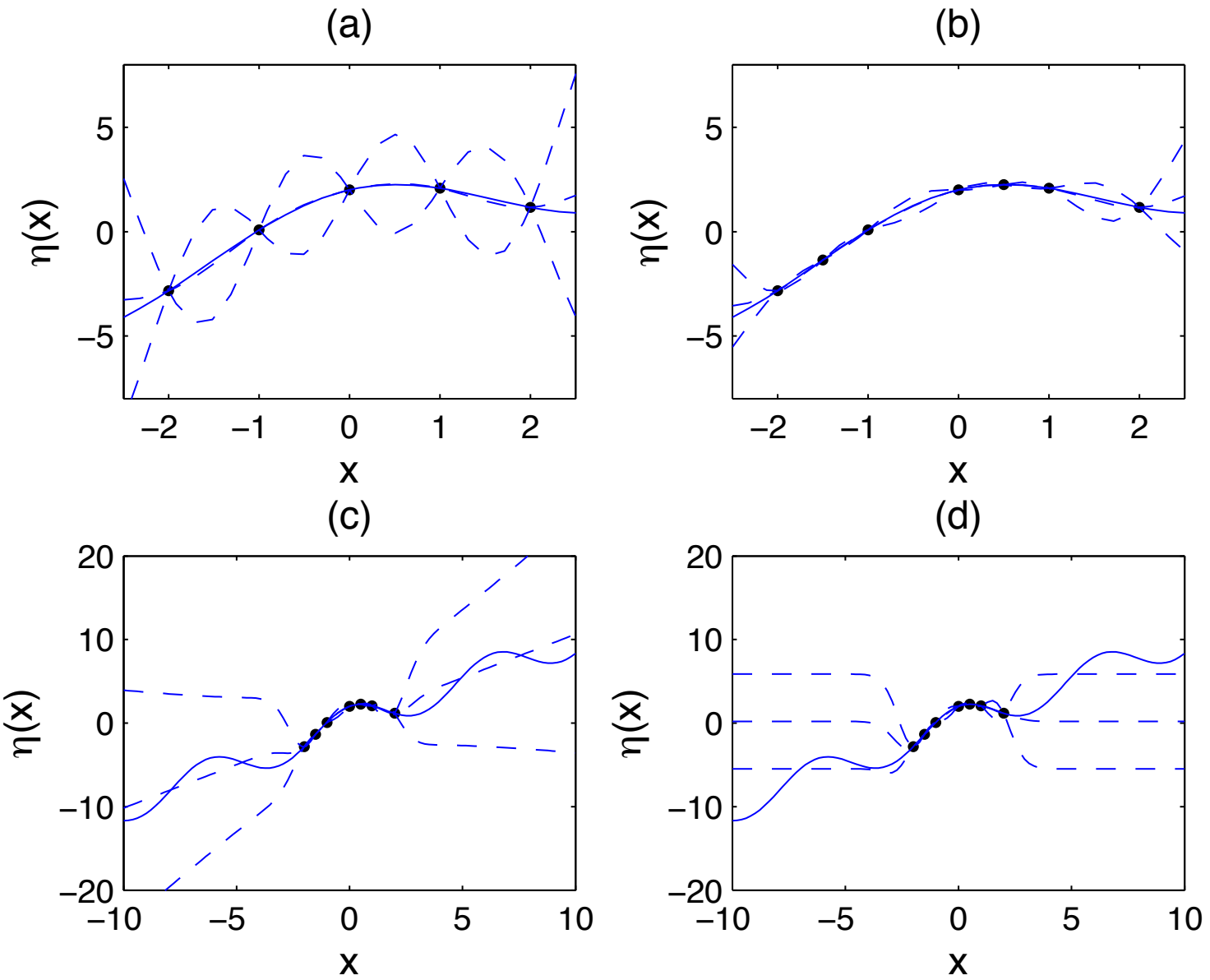


Choice of mean function

(or things to think about when choosing your covariance kernel)

$$f(x) = m(x) + Z(x)$$

- Can be important if training data set small; gaps between points can be large.
- If extrapolating, emulator predictions will 'revert' to $m(x)$


(a)
(b)
(c)
(d)
η(x)
x
5
0
−5
20
10
−10
−20
−2
−1
1
2
−10
−5
5
10

Choice of mean function

$$f(x) = m(x) + Z(x)$$

- Usually choose simple linear form. If $x = (x_1, \ldots, x_d)$,

$$m(x) = \beta_0 + \beta_1 x_1 + \ldots + \beta_d x_d.$$

- Can treat $\beta_0, \ldots, \beta_d$ as uncertain, and integrate out of posterior
- Some claim $m(x) = \beta_0$ constant form works better...
- ...others claim better to include higher order polynomial terms (quadratics, interactions)
- For multi level simulator case, fast simulator can be used as a prior mean for the slow simulator
- Using a second GP (with noise) for the mean can help deal with nonstationarity (Ba and Joseph, 2012)

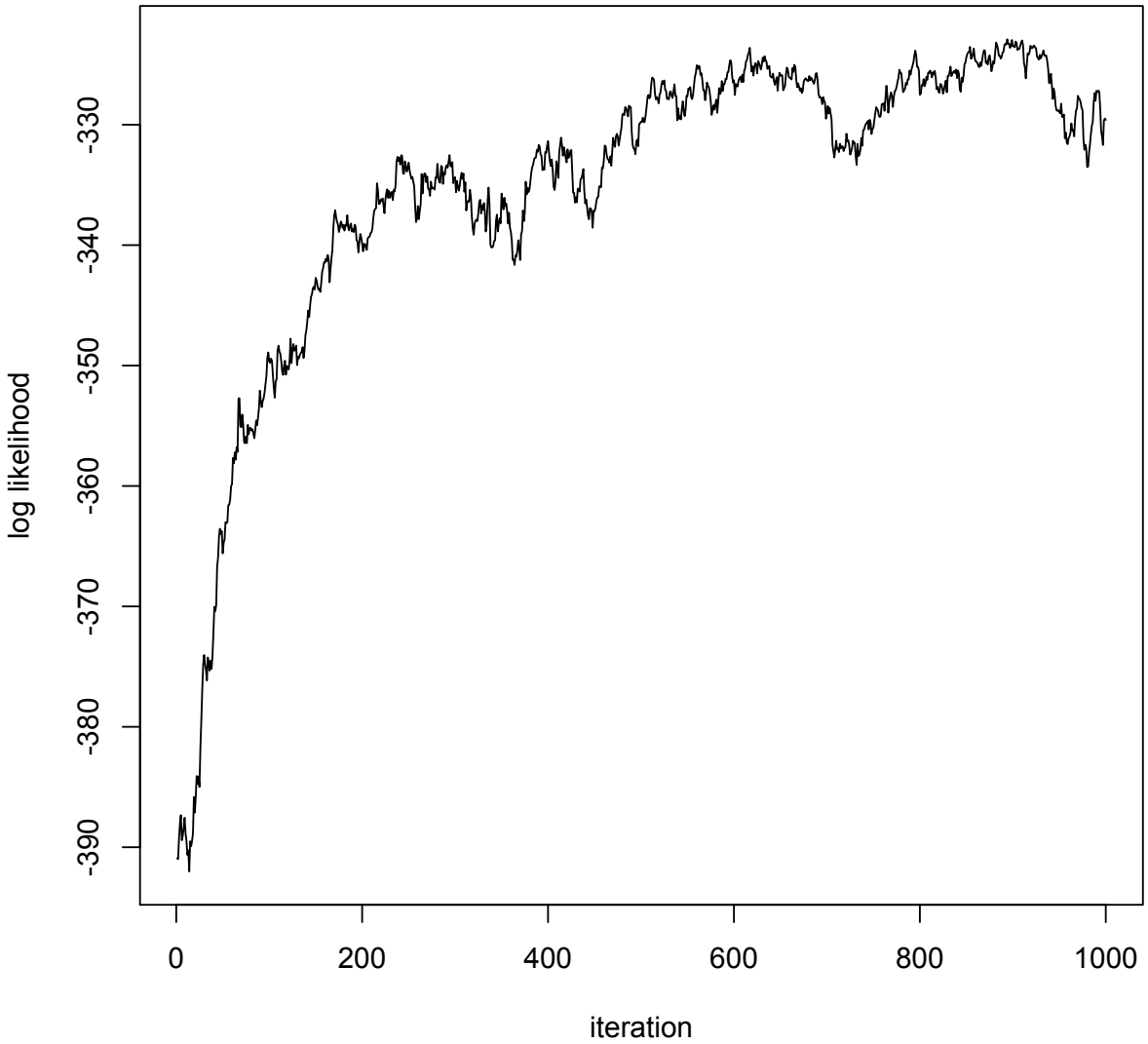
Estimating the correlation function parameters

For vector input $x = (x_1, \ldots, x_d)$.

$$c(x, x') = \exp\left\{-\sum_{i=1}^{d}\left(\frac{x_i - x_i'}{\delta_i}\right)^2\right\}$$

- Can integrate out σ^2 analytically, but not δ_i
- Maximum likelihood probably most popular: the main computational burden
- Some authors do 'full Bayes' using MCMC
- Others fix the correlation parameters, and include more polynomial terms in the mean
- Importance sampling:
  Nagy, B., Loeppky, J. L. and Welch, W. J. (2007). Fast Bayesian inference for Gaussian process models.

Initial Gibbs sampler can be helpful for starting an optimiser. Example: 18 input climate simulator



Summary

- UQ: lots of interesting statistical problems in the use of deterministic computer models
  - Propagating uncertainty
  - Sensitivity analysis for identifying 'important' inputs
  - Calibration/history matching/inverse problems
- Modelling 'simulator discrepancy' perhaps the most important challenge
- Gaussian processes popular for dealing with computationally expensive models
- Often used with small datasets - diagnostics important