Uncertainty quantification for complex simulators using emulation

Richard Wilkinson

School of Maths and Statistics
University of Sheffield

British Antarctic Survey
October 2016

Rohrlich (1991): Computer simulation is

> *'a key milestone somewhat comparable to the milestone that started the empirical approach (Galileo) and the deterministic mathematical approach to dynamics (Newton and Laplace)'*

Challenges for statistics:
How do we make inferences about the world from a simulation of it?

- how do we relate simulators to reality? (model error)
- how do we estimate tunable parameters? (calibration)
- how do we deal with computational constraints? (stat. comp.)
- how do we make uncertainty statements about the world that combine models, data and their corresponding errors? (UQ)

There is an inherent a lack of quantitative information on the uncertainty surrounding a simulation - unlike in physical experiments.

Outline

- Uncertainty quantification (UQ)
- Emulators
- High dimensional problems
- Calibration

Representation of uncertainty

Probability can be used to represent uncertainty.

- Given minimal assumptions, probability can be shown to be the only rational way to represent uncertainty.

Representation of uncertainty

Probability can be used to represent uncertainty.

- Given minimal assumptions, probability can be shown to be the only rational way to represent uncertainty.
- Probability is subjective probability - distributions represent degrees of belief of individuals.

Representation of uncertainty

Probability can be used to represent uncertainty.

- Given minimal assumptions, probability can be shown to be the only rational way to represent uncertainty.
- Probability is subjective probability - distributions represent degrees of belief of individuals.
- All uncertainty quantities θ can be given distributions $\pi(\theta)$ that represent our uncertainty about the value
  - unknown functions will be described by probability distributions across a class of unknown functions

Uncertainty quantification motivation

1. Uncertainty analysis
   - What is the distribition of $f(\theta)$ given uncertainty about θ
2. Sensitivity analysis
   - Assign uncertainty in our predictions ($\mathbb{V}\text{ar}f(\theta)$) to various uncertainties in the inputs ($\mathbb{V}\text{ar}(\theta)$)?
   - Useful for understanding the simulator response, designing future measurements (what should we measure to most decrease our uncertainty?)
3. Estimating model discrepancy
4. Parameter estimation/calibration
   - Estimating boundary conditions and unknowns
5. Calibrated prediction
   - Suppose we wish to predict $\mathcal{D}_f$ given $\mathcal{D}_p$, taking account of parametric uncertainty:

$$\pi(\mathcal{D}_f \mid \mathcal{D}_p) = \int \pi(\mathcal{D}_f \mid \theta)\pi(\theta \mid \mathcal{D}_p)\mathrm{d}\theta$$

   - Includes hindcasting/reconstructions...

Code uncertainty

For complex simulators, run times might be long, ruling out brute-force approaches such as Monte Carlo methods.

Code uncertainty

For complex simulators, run times might be long, ruling out brute-force approaches such as Monte Carlo methods.

- All inference must be done using a finite ensemble of model runs

$$\mathcal{D}_{sim} = \{(\theta_i, \eta(\theta_i))\}_{i=1,\ldots,N}$$

- If θ is not in the ensemble, then we are uncertainty about the value of $\eta(\theta)$.

Code uncertainty

For complex simulators, run times might be long, ruling out brute-force approaches such as Monte Carlo methods.

- All inference must be done using a finite ensemble of model runs

$$\mathcal{D}_{sim} = \{(\theta_i, \eta(\theta_i))\}_{i=1,\ldots,N}$$

- If θ is not in the ensemble, then we are uncertainty about the value of $\eta(\theta)$.

If θ is multidimensional, then even short run times can rule out brute force approaches

- $\dim(\theta) \in \mathbb{R}^{10}$ then 1000 simulator runs is only enough for one point in each corner of the design space.

Meta-modelling

Idea: If the simulator is expensive, build a cheap model of it and use this in any analysis.

'a model of the model'

We call this a meta-model/surrogate/*emulator*/reduced-order model of the simulator f.

- Try to find $\eta(\theta)$ such that

$$\eta(\theta) \approx f(\theta) \quad \forall \quad \theta \in I \subset \mathbb{R}$$

Meta-modelling

Idea: If the simulator is expensive, build a cheap model of it and use this in any analysis.

'a model of the model'

We call this a meta-model/surrogate/*emulator*/reduced-order model of the simulator f.

- Try to find $\eta(\theta)$ such that

$$\eta(\theta) \approx f(\theta) \quad \forall \quad \theta \in I \subset \mathbb{R}$$

An emulator should come with an assessment of its accuracy

- rather than just predicting $\eta(\theta)$ it should predict $\pi(\eta(\theta)|\mathcal{D}_{sim})$ - our uncertainty about the simulator value given the ensemble $\mathcal{D}_{sim}$.

Gaussian Process Emulators

Gaussian processes (GPs) provide a flexible nonparametric family of functions:

$$\eta(\cdot) \sim GP(m(\cdot), \sigma^2 c(\cdot, \cdot))$$

where $m(\cdot)$ is a prior mean function, and $c(\cdot, \cdot)$ is the prior covariance function (semi-definite).

Gaussian Process Emulators

Gaussian processes (GPs) provide a flexible nonparametric family of functions:

$$\eta(\cdot) \sim GP(m(\cdot), \sigma^2 c(\cdot, \cdot))$$

where $m(\cdot)$ is a prior mean function, and $c(\cdot, \cdot)$ is the prior covariance function (semi-definite).

Definition If $f(\cdot) \sim GP(m(\cdot), c(\cdot, \cdot))$ then for any collection of inputs $x_1, \ldots, x_n$ the vector

$$(f(x_1), \ldots, f(x_n))^T \sim MVN(m(\mathbf{x}), \sigma^2 \mathbf{\Sigma})$$

where $\Sigma_{ij} = c(x_i, x_j)$.

Family includes cubic splines and Brownian motion.

Meta-modelling

Gaussian Process Emulators

Gaussian processes are invariant under Bayesian updating.

If we observe the ensemble of model runs $\mathcal{D}_{\text{sim}}$, then update our prior belief about η in light of the ensemble of model runs:

$$\eta(\cdot)|\mathcal{D}_{sim} \sim GP(m^*(\cdot), \sigma^2 c^*(\cdot,\cdot))$$

where m^* and c^* are the updated mean and covariance functions.

Meta-modelling

Gaussian Process Emulators

Gaussian processes are invariant under Bayesian updating.

If we observe the ensemble of model runs $\mathcal{D}_{\text{sim}}$, then update our prior belief about η in light of the ensemble of model runs:

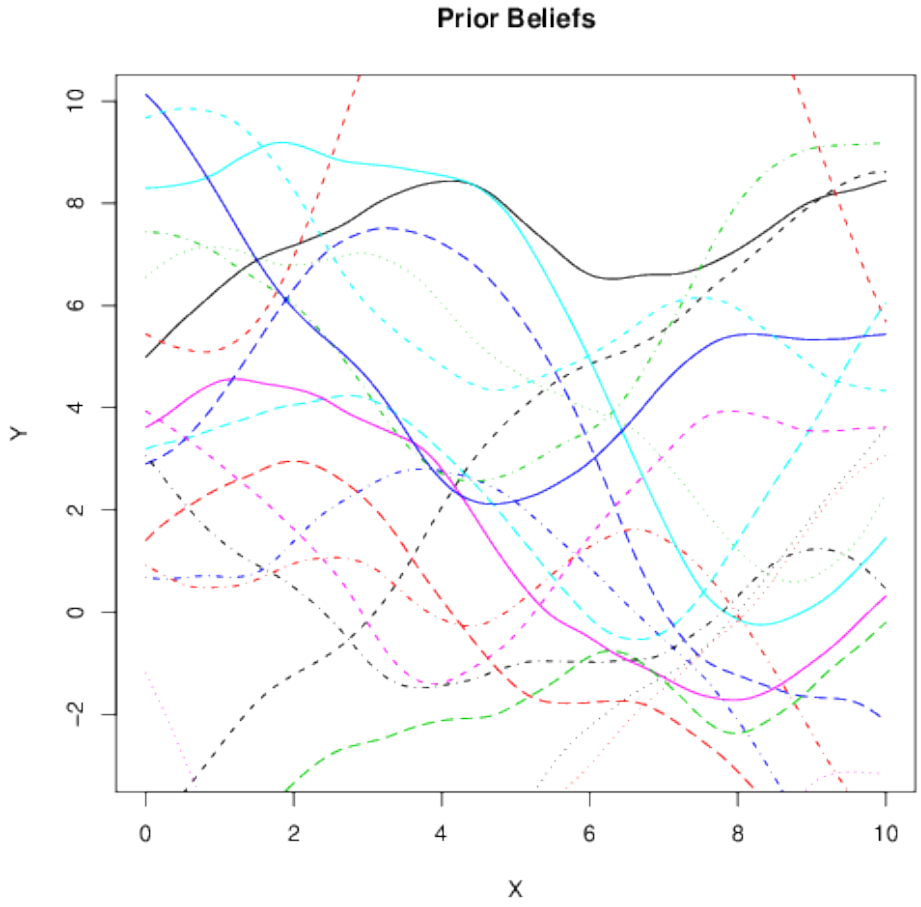
$$\eta(\cdot)|\mathcal{D}_{sim} \sim GP(m^*(\cdot), \sigma^2 c^*(\cdot,\cdot))$$

where m^* and c^* are the updated mean and covariance functions.

We can also condition on any linear transformation of η (eg knowledge of the derivative, integral), symmetry, monotonicity,....

Gaussian Process prior for unknown functions

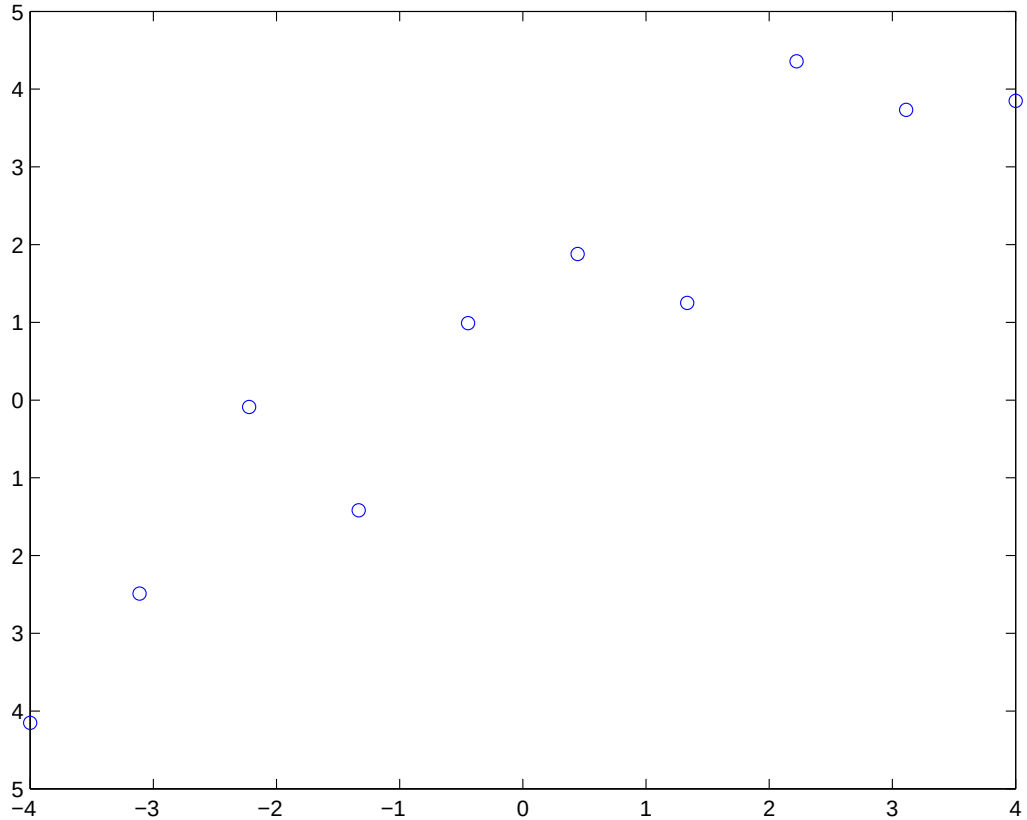
Prior belief about f



GPs can be understood as prior distributions over functions. Their properties, such as the smoothness and differentiability are controlled by the choice of mean and covariance functions, and the hyper-parameters.

Gaussian Process prior for unknown functions

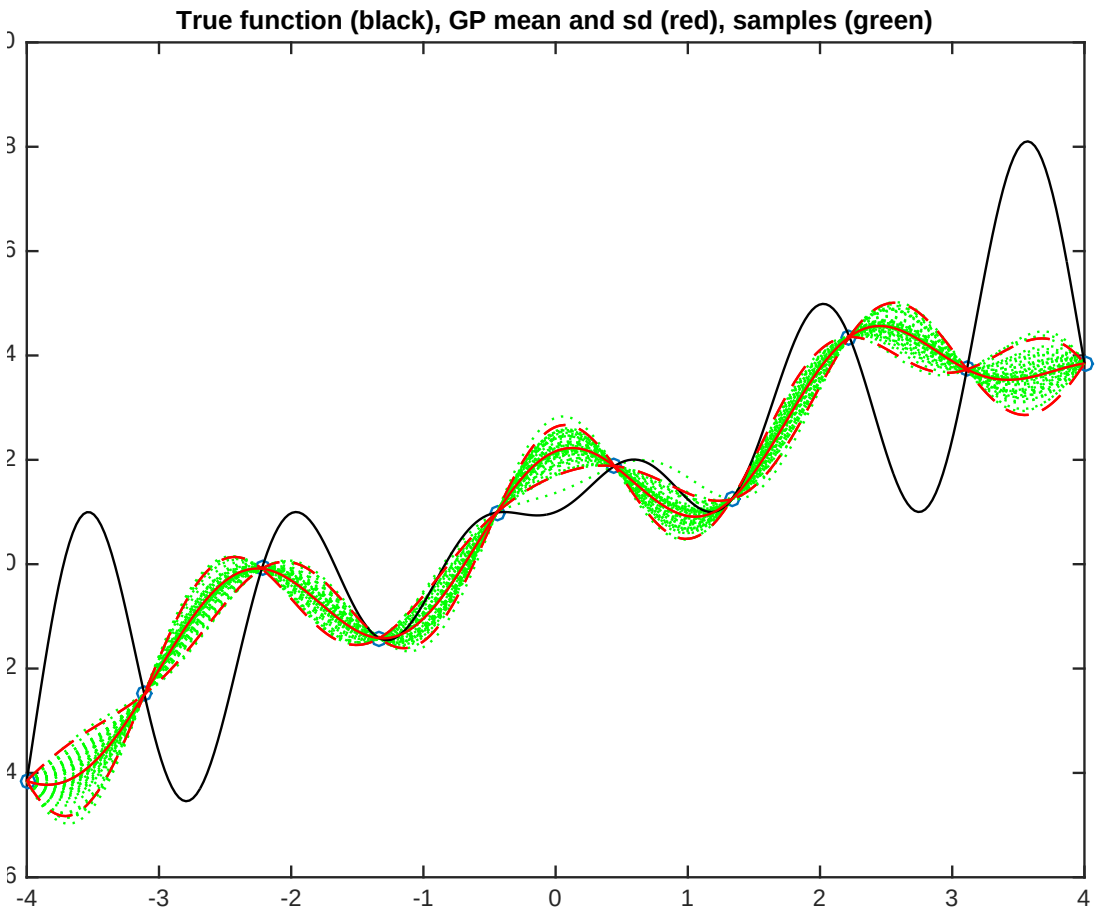
$y = f(x) = 1 + x + x\sin(4x)$ - 10 data points



Once we observe the data $D = \{(x_i, y_i)\}$, we can update our prior belief about the unknown function $f(x)$

Gaussian Process emulation - posterior beliefs about $f(\cdot)$

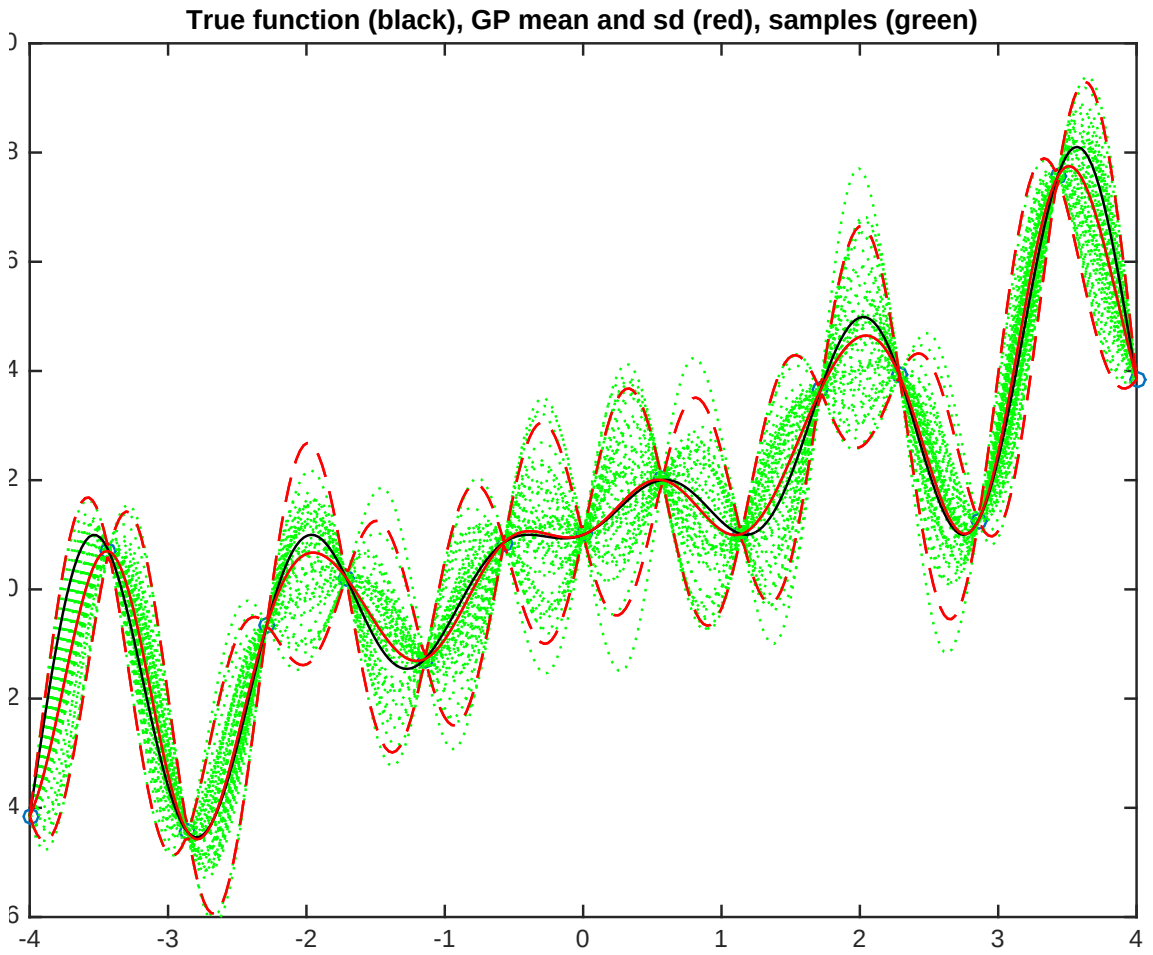
$y = 1 + x + x\sin(4x)$ - 10 data points



Perverse example: we can spot errors using cross-validation $\rightarrow$ More data required.

Gaussian Process emulation

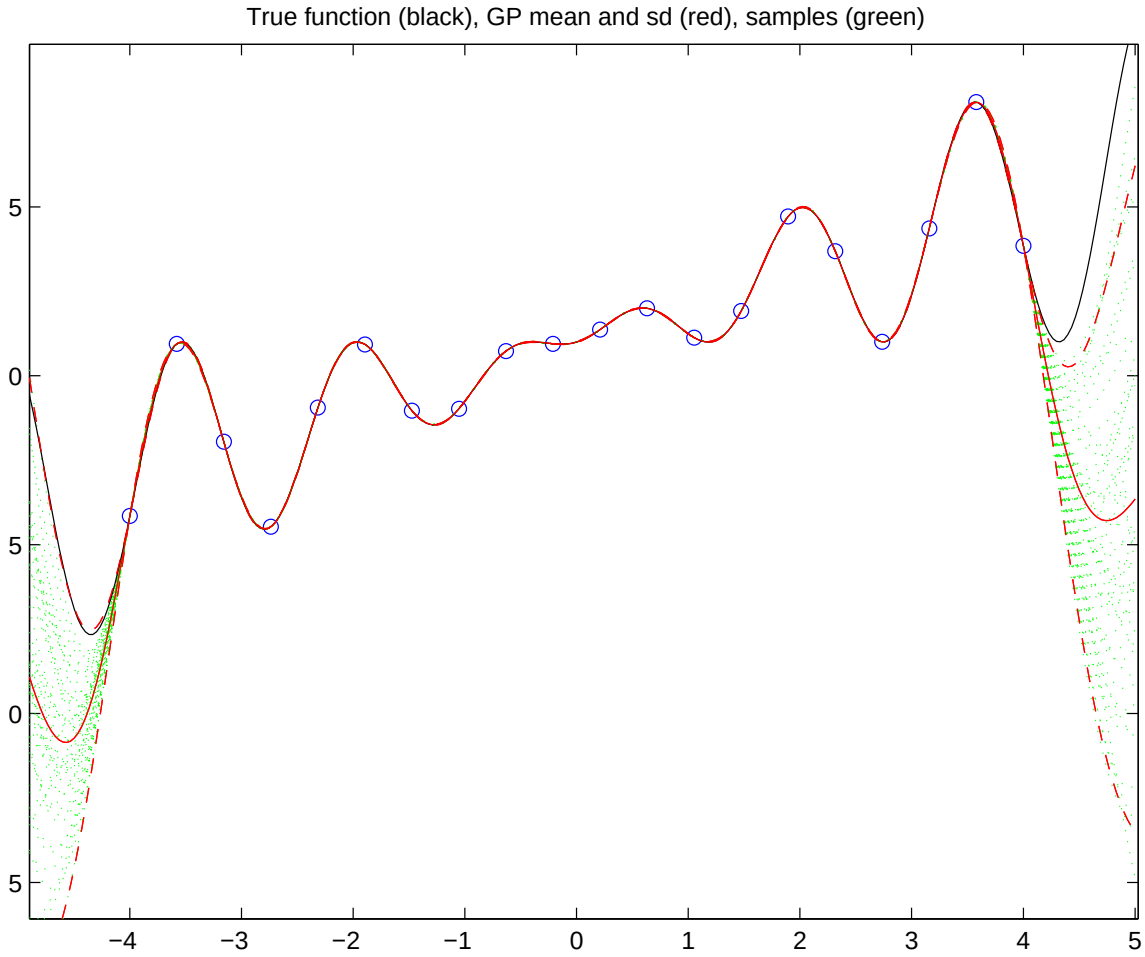
$y = 1 + x + x\sin(4x)$ - 15 data points



The covariance function is key. There are a small number of common choices, e.g., squared exponential (RBF/Gaussian), Matern, neural-net

Gaussian Process emulation

$y = 1 + x + x\sin(4x)$ - 20 data points



We can add, multiply and transform any covariance function to obtain a new valid covariance function.

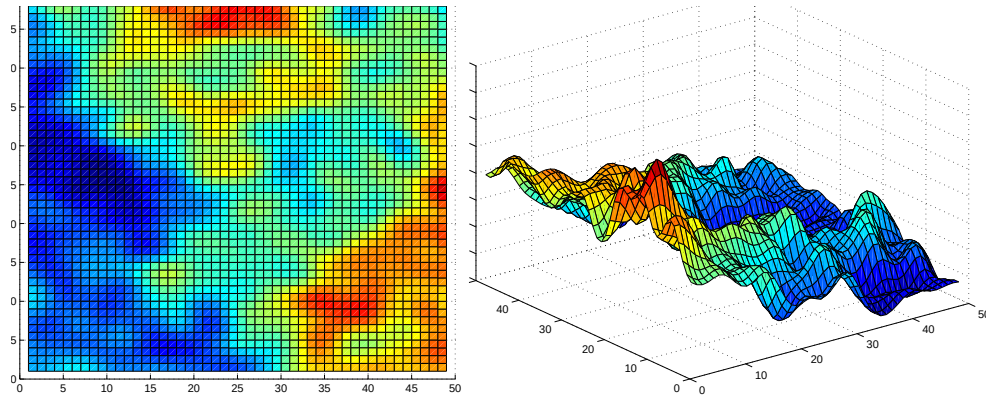
High dimensional problems

Carbon capture and storage

Knowledge of the physical problem is encoded in a simulator f

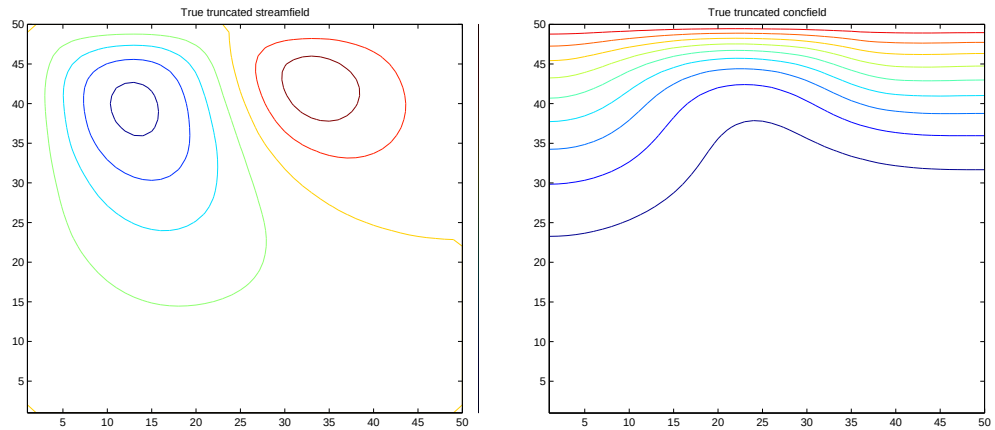
Inputs:

Permeability field, K
(2d field)



$\downarrow f(K)$

$f(K)$



Outputs:

Stream func. (2d field),
concentration (2d field),
surface flux (1d scalar),
$\vdots$

Surface Flux$= 6.43, \ldots$

Uncertainty quantification (UQ) for CCS

The simulator maps from permeability field K to outputs such as the surface flux $\mathcal{S}$. Let $f(K)$ denote this mapping

$$f : K \rightarrow \mathcal{S}$$

For most problems the permeability K is unknown.

Uncertainty quantification (UQ) for CCS

The simulator maps from permeability field K to outputs such as the surface flux $\mathcal{S}$. Let $f(K)$ denote this mapping

$$f : K \to \mathcal{S}$$

For most problems the permeability K is unknown.

If we assume a distribution for $K \sim \pi(K)$, we can quantify our uncertainty about $\mathcal{S} = f(K)$.

- e.g., by finding the cumulative distribution function (CDF) of $\mathcal{S}$:

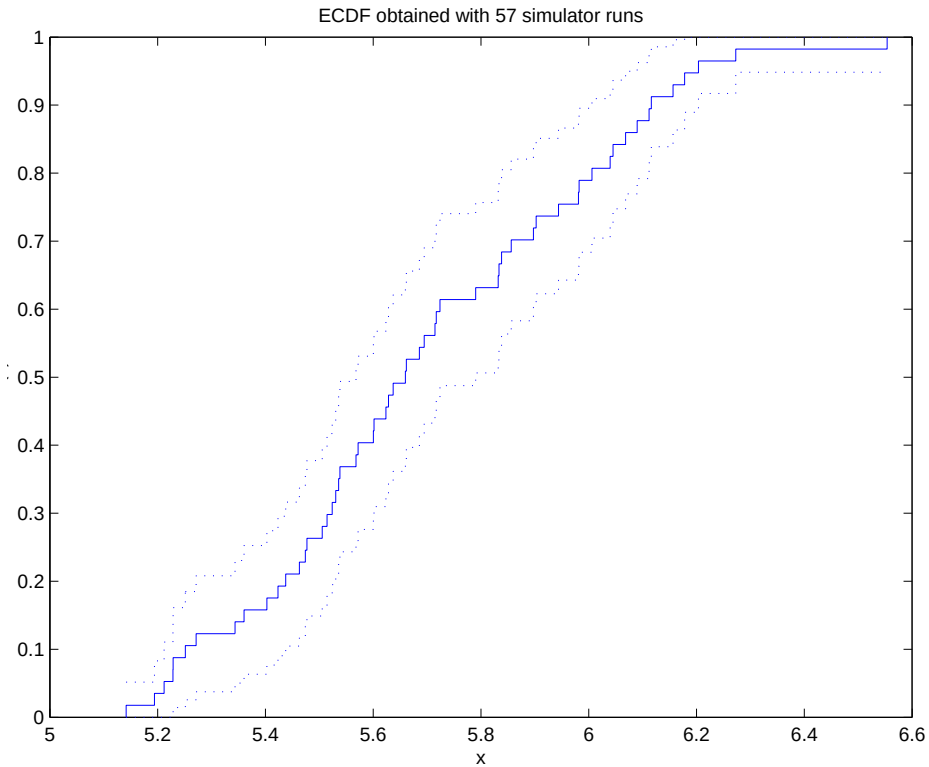
$$F(s) = \mathbb{P}(f(K) \leq s)$$

UQ for complex computer models

Gold standard approach: Monte Carlo simulation

- Draw $K_1, \ldots, K_N \sim \pi(K)$, and evaluate the simulator at each giving fluxes
  $s_1 = f(K_1), \ldots, s_N = f(K_N)$
- Estimate the empirical CDF

$$\widehat{F}(s) = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}_{s_i \leq s}$$

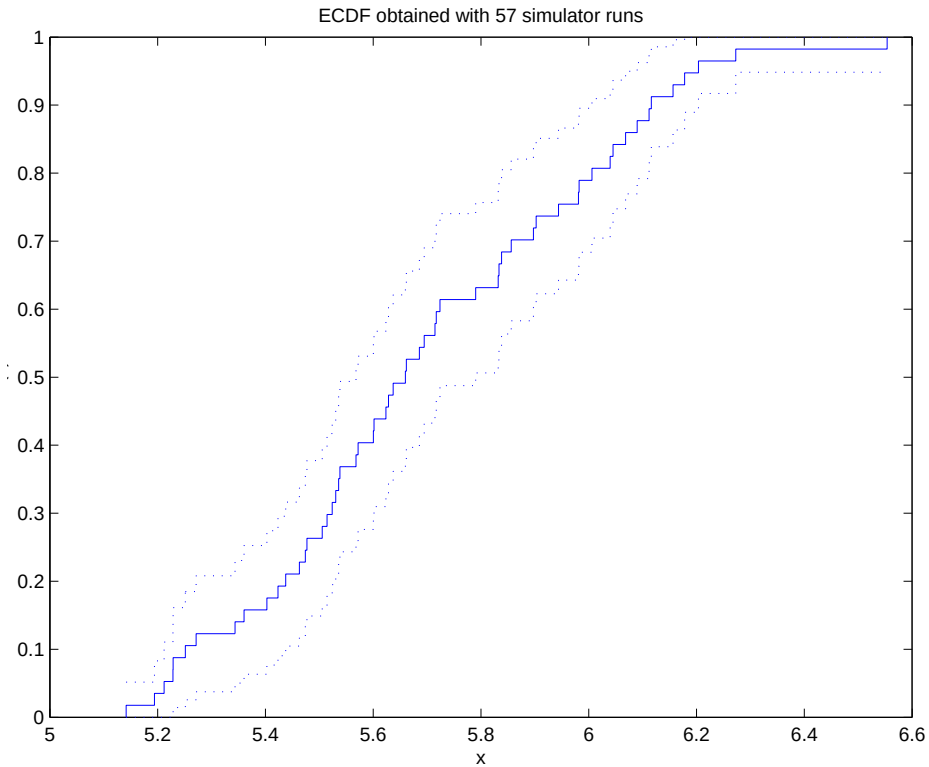


UQ for complex computer models

Gold standard approach: Monte Carlo simulation

- Draw $K_1, \ldots, K_N \sim \pi(K)$, and evaluate the simulator at each giving fluxes
  $s_1 = f(K_1), \ldots, s_N = f(K_N)$
- Estimate the empirical CDF

$$\widehat{F}(s) = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}_{s_i \leq s}$$



Note that $N = 10^3$ is not large if we want quantiles in the tail of the distribution

However the cost of the simulator means we are limited to $\sim$100 evaluations.

Multivariate Emulation

Wilkinson 2010

How can we deal with multivariate ouput?

- Build independent or separable multivariate emulators,
- Linear model of coregionalization?

Multivariate Emulation

Wilkinson 2010

How can we deal with multivariate ouput?

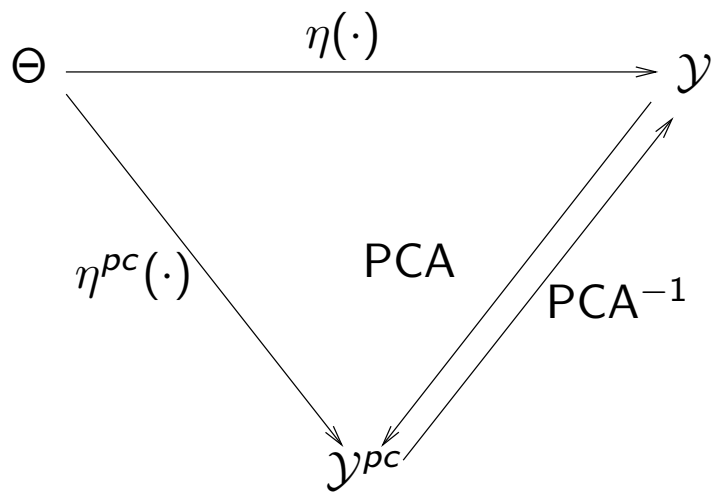
- Build independent or separable multivariate emulators,
- Linear model of coregionalization?

Instead, if the outputs are highly correlated we can reduce the dimension of the data by projecting the data into some lower dimensional space $\mathcal{Y}^{pc}$, i.e., assume

$$y = Wy^{pc} + e$$

where $dim(y) >> dim(y^{pc})$

Emulate from Θ to the reduced dimensional output space $\mathcal{Y}^{pc}$



Principal Component Emulation (EOF)

1. Centre and scale the data $\mathcal{D}_{\text{sim}}$ (scaling prior specification simpler).
2. Find the singular value decomposition of $\mathcal{D}_{\text{sim}}$.

$$\mathcal{D}_{\text{sim}} = U \Gamma V^*.$$

   Γ contains the singular values (eigenvalues), and V the principal components (eigenvectors).

3. Decide on the dimension of the principal subspace, n^* say, and throw away all but the n^* leading principal components. An orthonormal basis for the principal subspace is given by the first n^* columns of V, denoted V_1. Let V_2 be the matrix of discarded columns.
4. Project $\mathcal{D}_{\text{sim}}$ onto the principal subspace to find $\mathcal{D}_{\text{sim}}^{pc} = \mathcal{D}_{\text{sim}} V_1$

PCA emulation

We then emulate the reduced dimension model

$$\eta_{pc}(\cdot) = (\eta_{pc}^1(\cdot), \ldots, \eta_{pc}^{n^*}(\cdot)).$$

- Each component η_{pc}^i will be uncorrelated (in the ensemble) but not necessarily independent. We use independent Gaussian processes for each component.
- The output can be reconstructed (accounting for reconstruction error) by modelling the discarded components as Gaussian noise with variance equal to the corresponding eigenvalue:

$$\eta(\theta) = V_1 \eta_{pc}(\theta) + V_2 \text{diag}(\Lambda)$$

where $\Lambda_i \sim N(0, \Gamma_{ii})$ ($\Gamma_{ii} = i^{th}$ eigenvalue).

Comments

- This approach (PCA emulation) requires that the outputs are highly correlated.
- We are assuming that the output $\mathcal{D}_{\text{sim}}$ is really a linear combination of a smaller number of variables,

$$\eta(\theta) = \mathbf{v}_1 \eta^1_{pc}(\theta) + \ldots + \mathbf{v}_{n^*} \eta^{n^*}_{pc}(\theta)$$

  which may be a reasonable assumption in many situations, eg, temporal spatial fields.
- Although PCA is a linear method, the method can be used on highly non-linear models as we are still using non-linear Gaussian processes to map from Θ to $\mathcal{Y}^{pc}$ – the linear assumption applies only to the dimension reduction (and can be generalised).
- This method accounts for the reconstruction error from reducing the dimension of the data.

Emulating simulators with high dimensional input

For the CCS simulator, the input is a permeability field which only needs to be known at a finite but large number of locations,

- e.g. if we use a 100×100 grid in the solver, K contains 10^4 entries
- Impossible to directly model $f : \mathbb{R}^{10,000} \to \mathbb{R}$

Emulating simulators with high dimensional input

For the CCS simulator, the input is a permeability field which only needs to be known at a finite but large number of locations,

- e.g. if we use a 100×100 grid in the solver, K contains 10^4 entries
- Impossible to directly model $f : \mathbb{R}^{10,000} \to \mathbb{R}$

We can use the same idea to reduce the dimension of the inputs. However, because we know the distribution of K, it is more efficient to use the Karhunen-Loève (KL) expansion of K (rather than learn it empirically as in PCA)

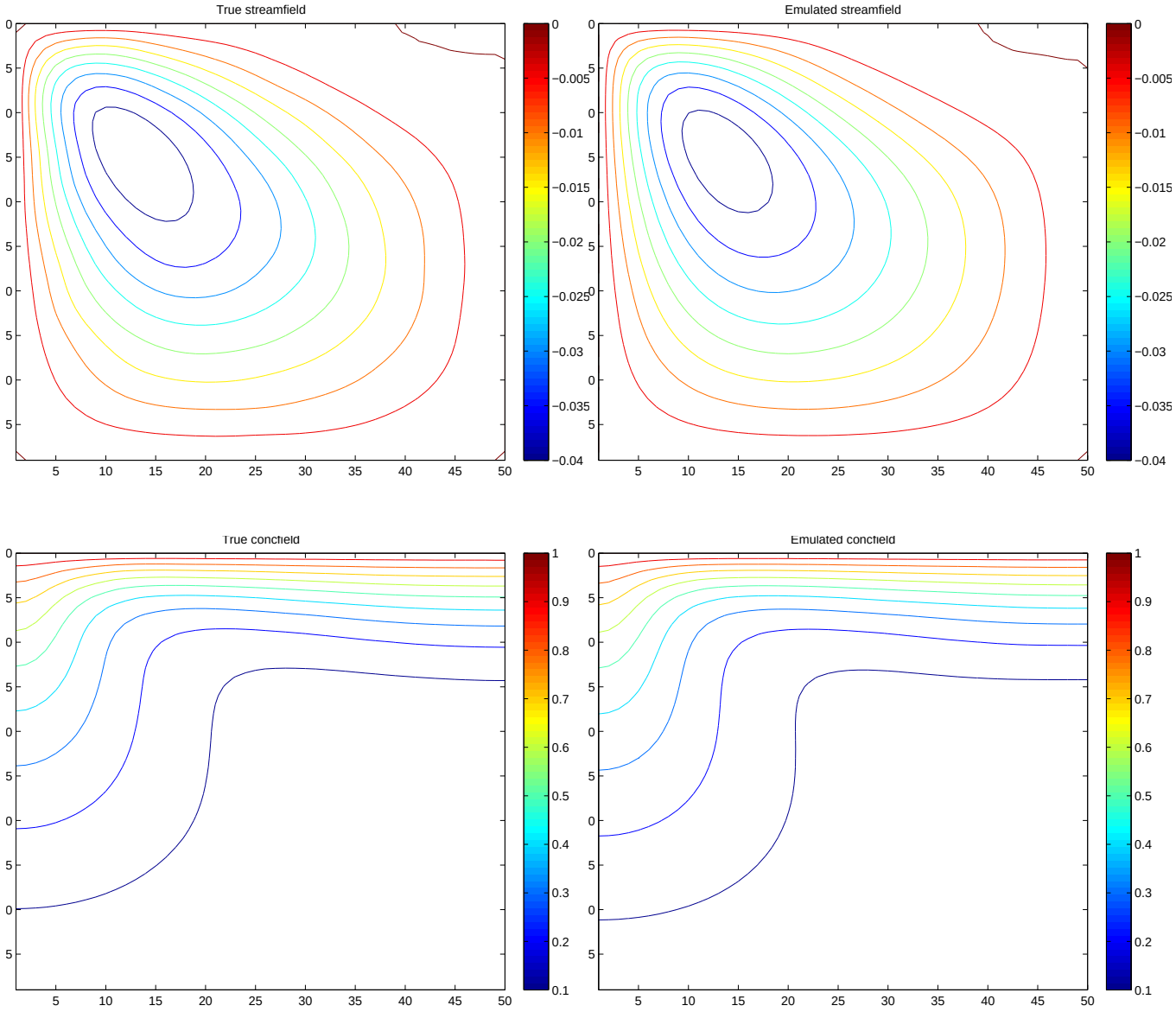
- $K = \exp(Z)$ where $Z \sim GP(m, C)$
- Z can be represented as

$$Z(\cdot) = \sum_{i=1}^{\infty} \lambda_i \xi_i \phi_i(\cdot)$$

where λ_i and ϕ_i are the eigenvalues and eigenfunctions of the covariance function of Z and $\xi_i \sim N(0, 1)$.

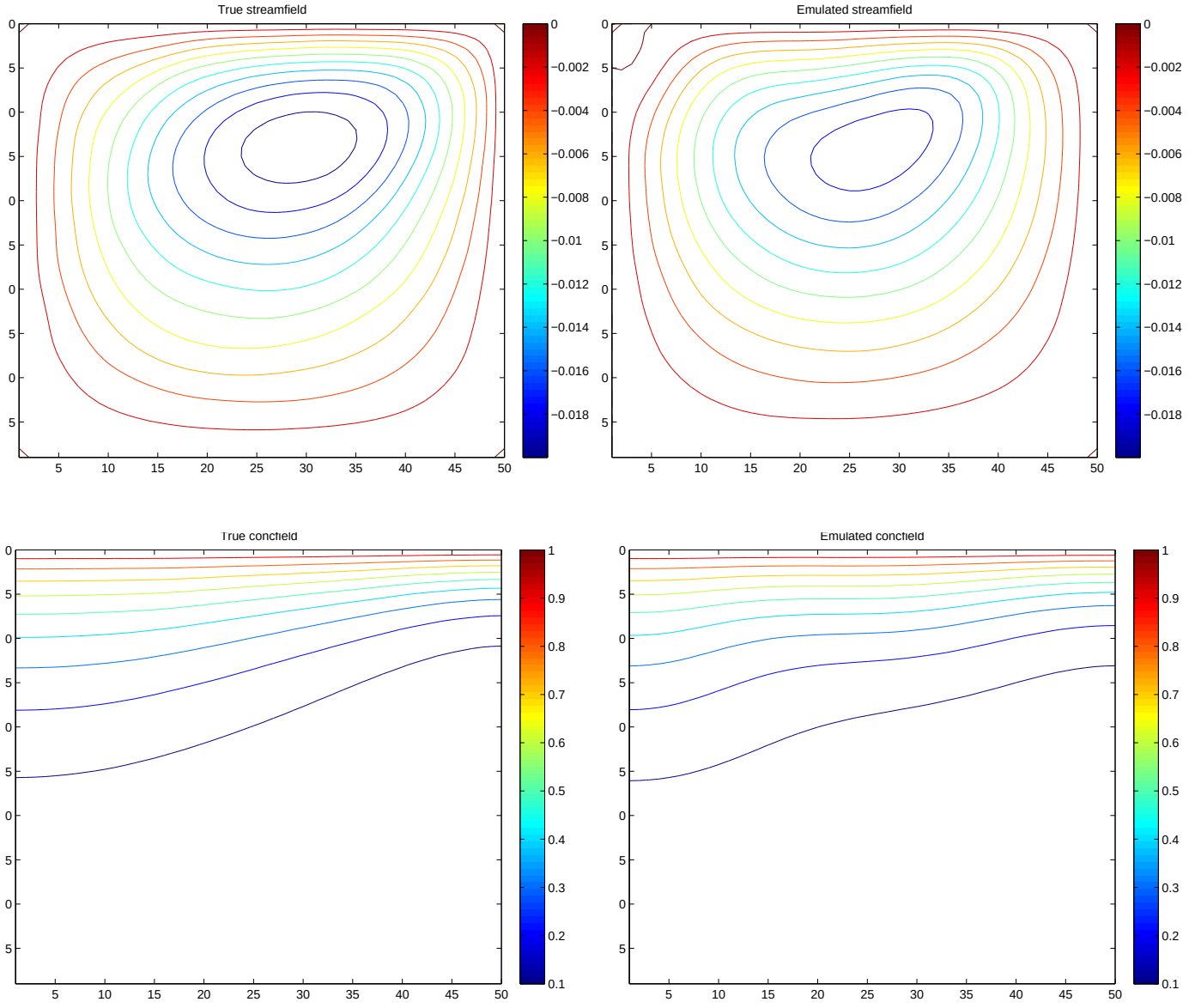
Emulating the stream function and concentration fields

Left=true, right = emulated, 118 training runs, held out test set.



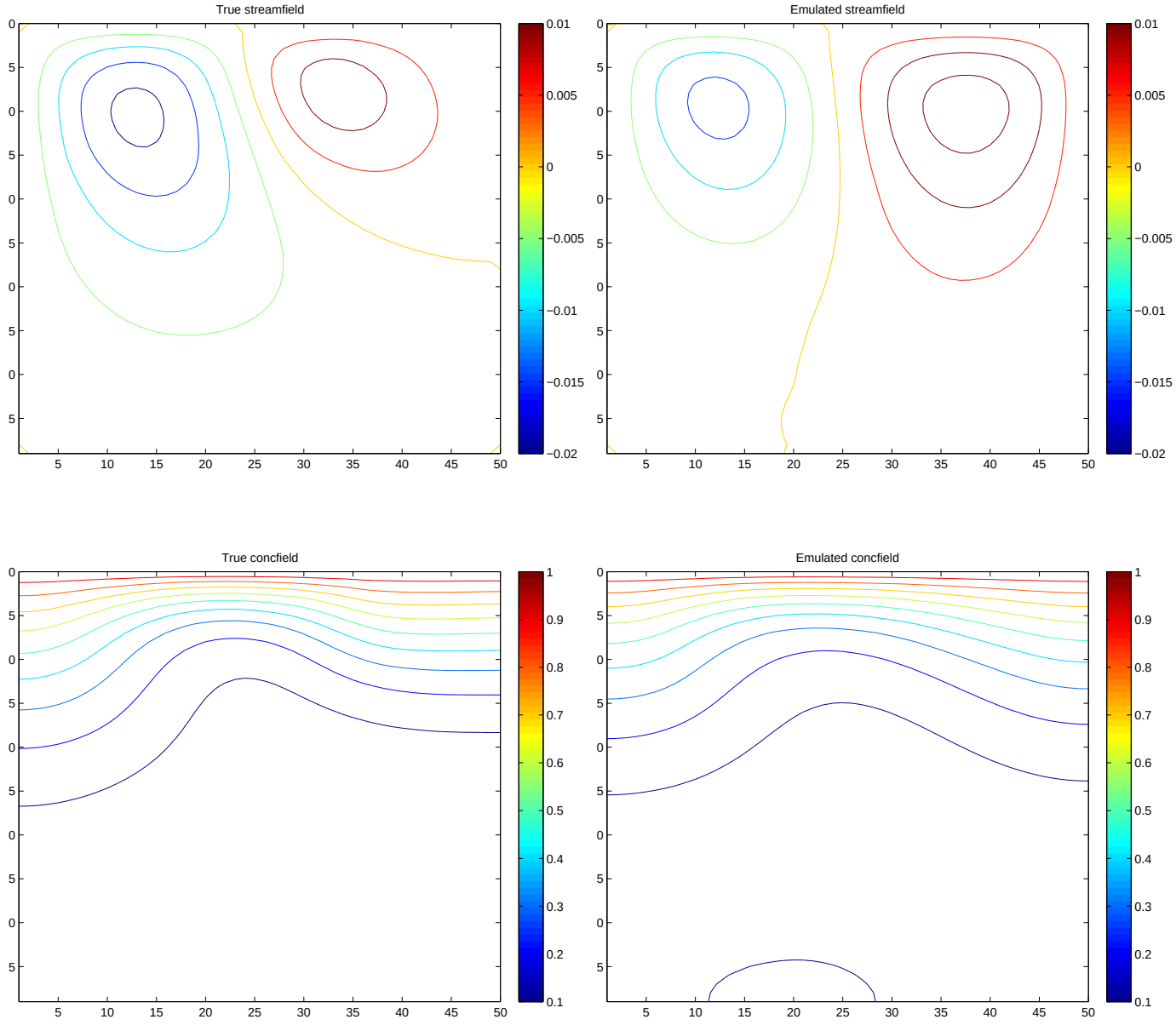
Emulating the stream function and concentration fields

Left=true, right = emulated, 118 training runs, held out test set.



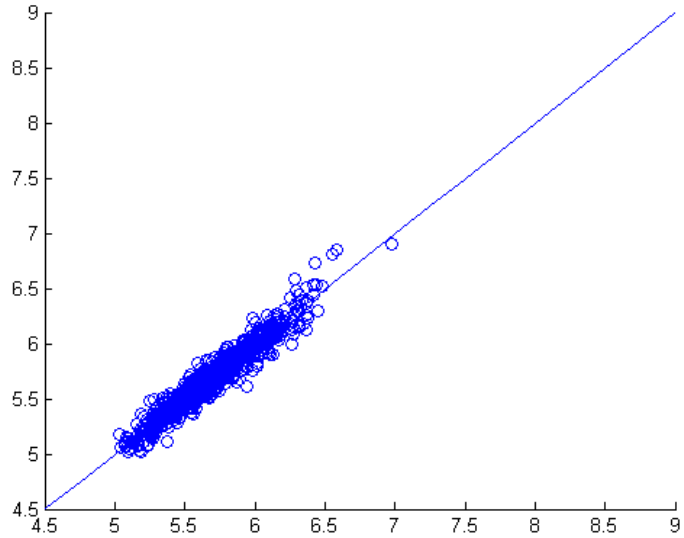
Emulating the stream function and concentration fields

Left=true, right = emulated, 118 training runs, held out test set.

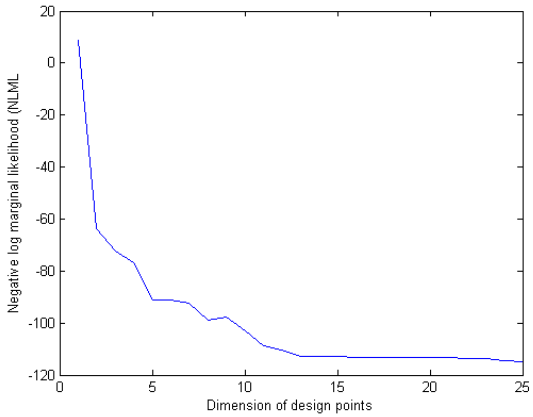


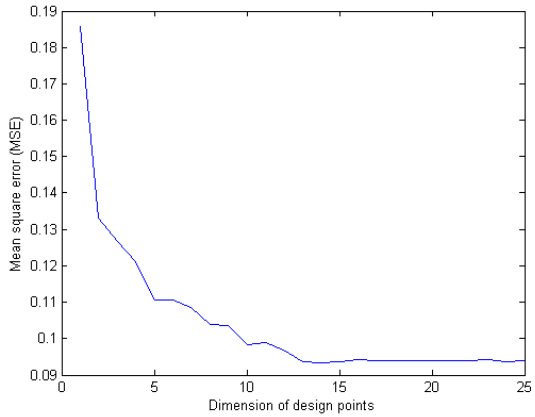
Predictive performance vs n = no. of KL components

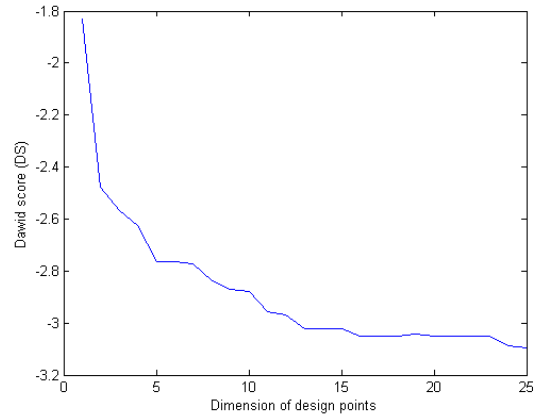
We can assess the accuracy of the emulator by examining the prediction error on a held-out test set. Plotting predicted vs true value indicates the accuracy the GP emulator.



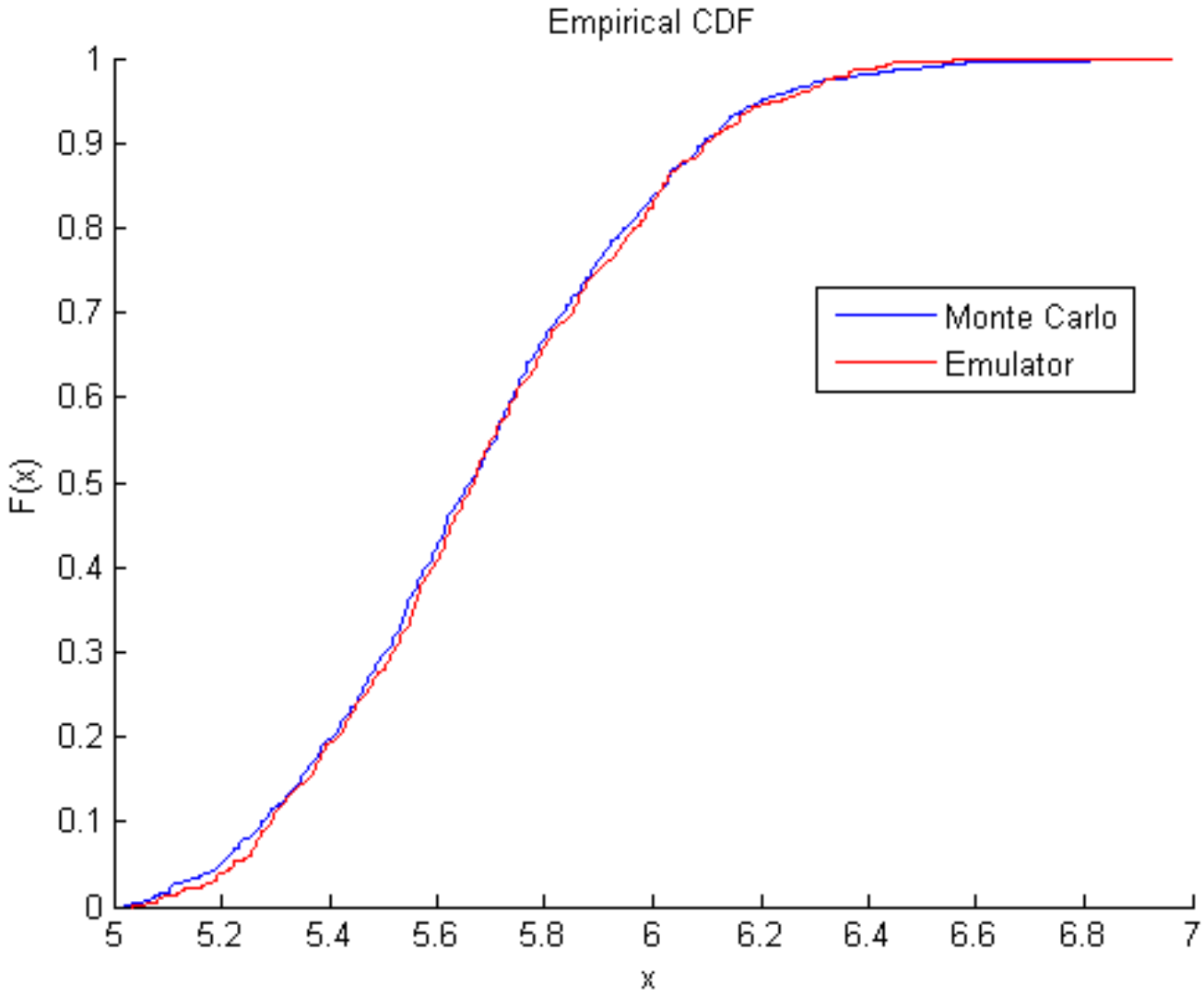
We can also choose the number of KL components to retain using numerical scores







CCS simulator results - 20 simulator training runs



Blue line = CDF from using 10^3 Monte Carlo samples from the simulator
Red line = CDF obtained using emulator (trained with 20 simulator runs, rational quadratic covariance function)

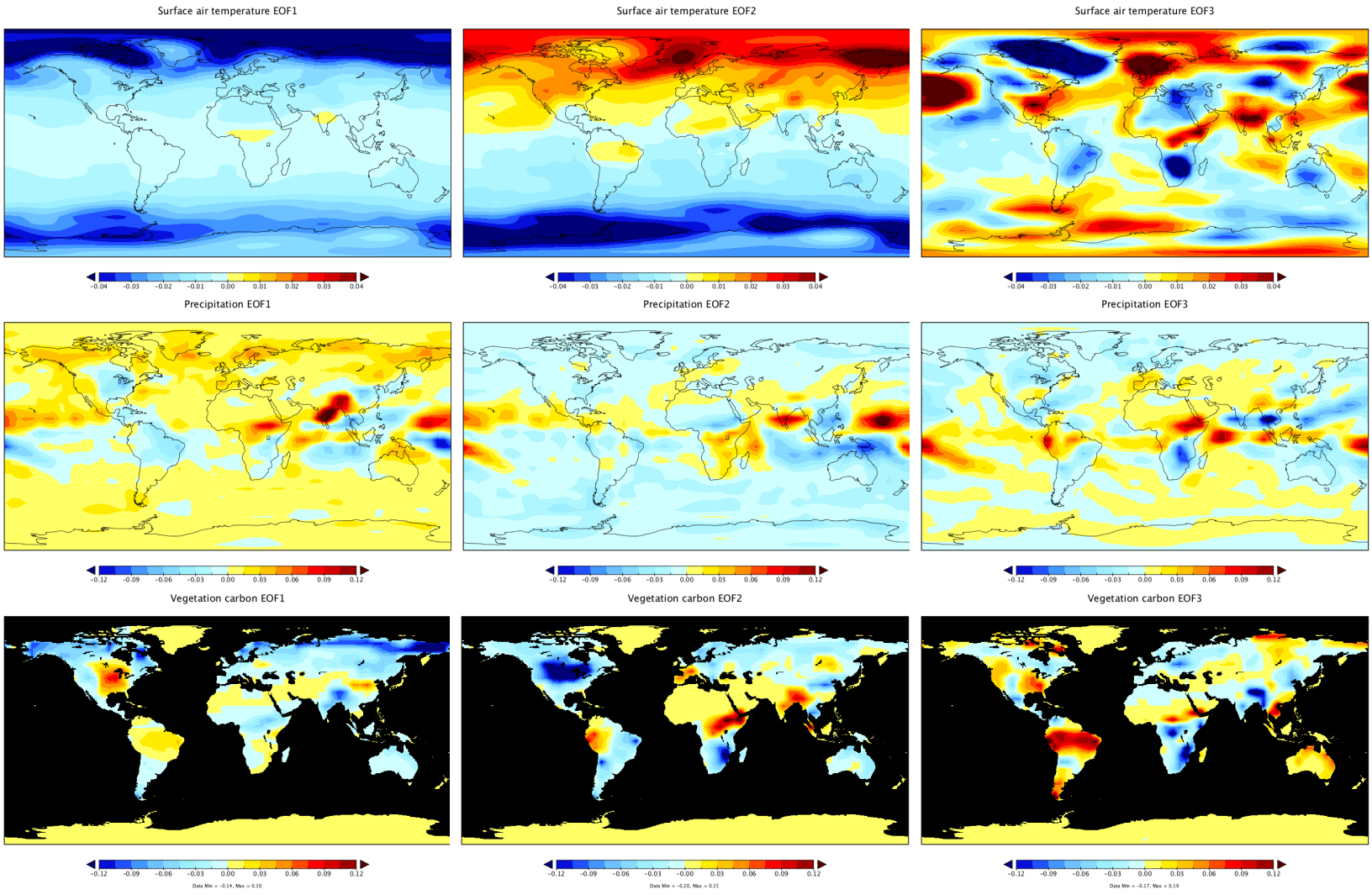
PLASIM-ENTS

Holden, Edwards, Garthwaite, Wilkinson 2015

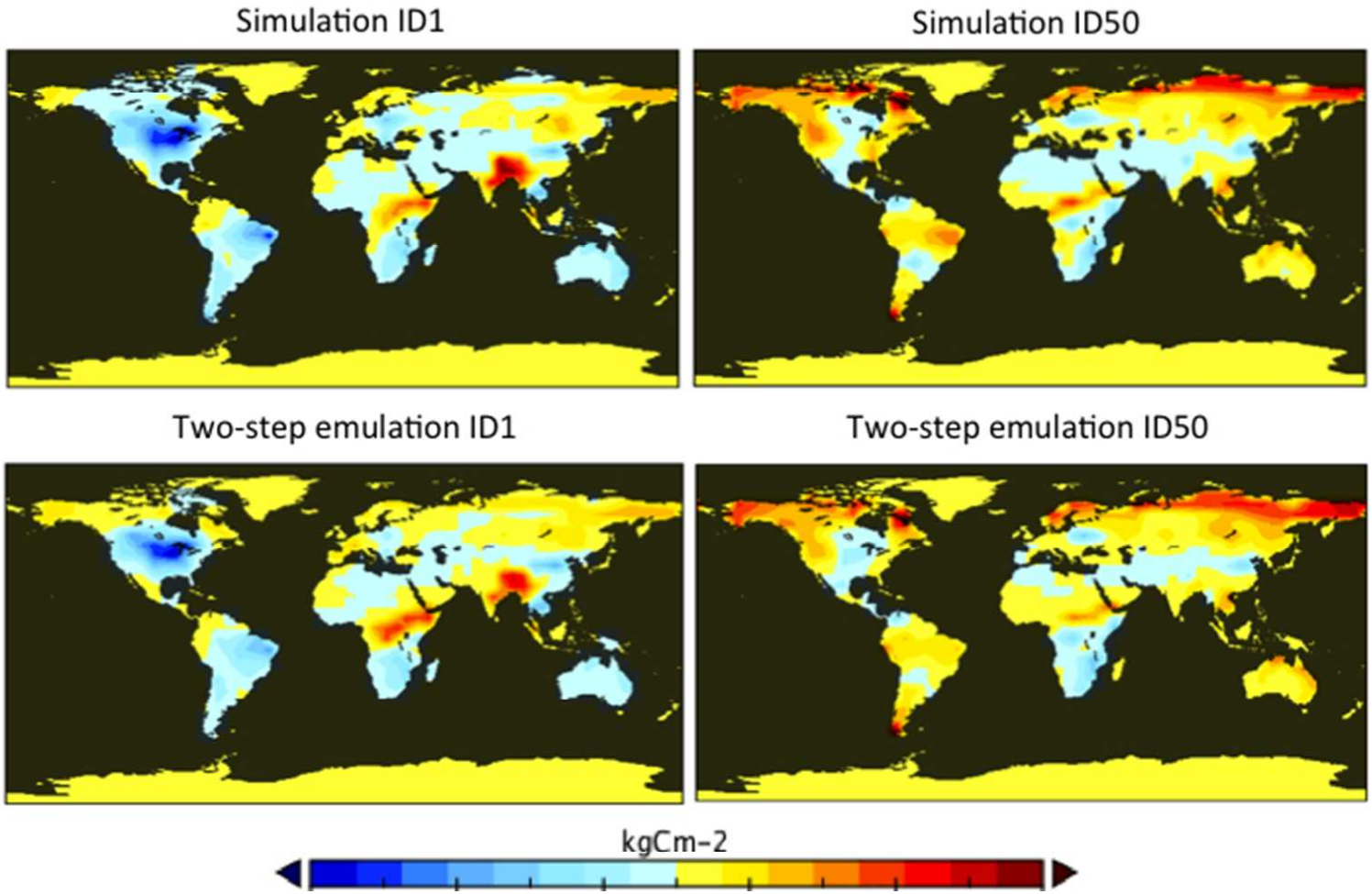
- Planet Simulator coupled to the terrestrial carbon model ENTS
- Inputs are eccentricity, obliquity, precession describing Earth's orbit around the sun.
- Model climate (annual average surface temperature and rainfall) and vegetation (annual average vegetation carbon density) spatial fields (on a 64×32) grid.

We used an ensemble of 50 simulations

Principal components



Leave-one-out cross validation of the emulator



We can then use the PC-emulator to do sensitivity analysis.

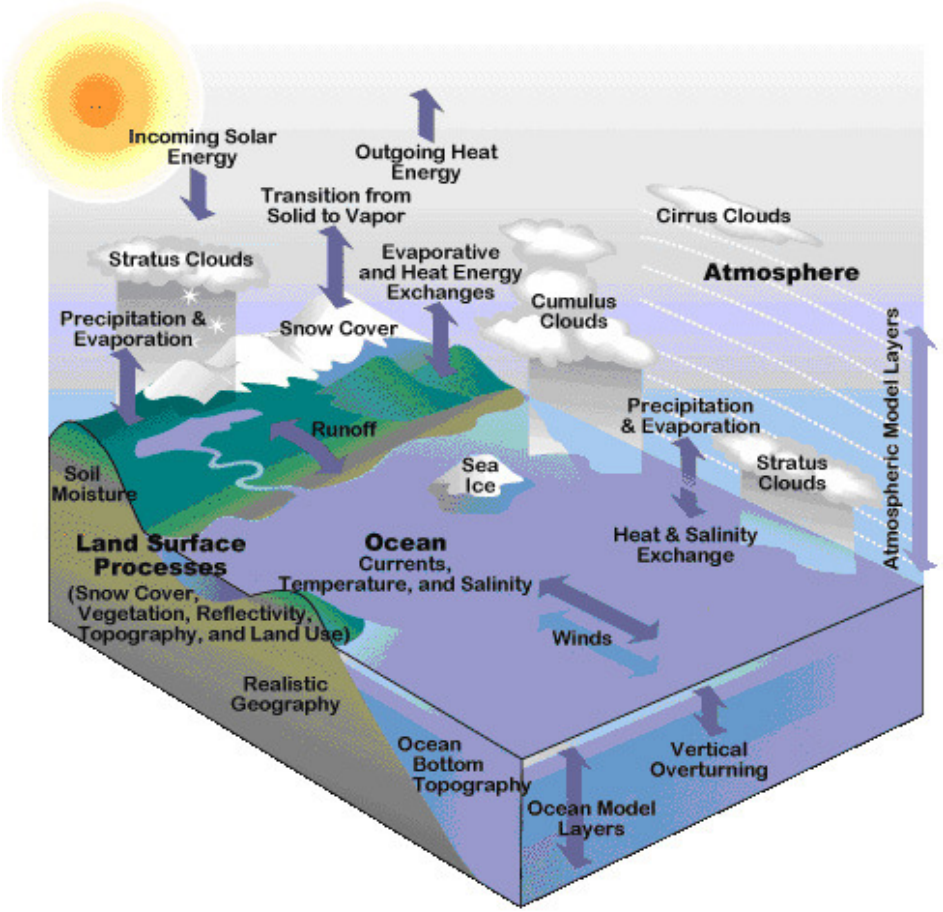
Calibration

The inverse problem

Most models are forwards models, i.e., specify parameters θ and i.c.s and the model $\eta(\cdot)$ generates output $\mathcal{D}$. Often, we are interested in the inverse-problem, i.e., observe data, want to estimate parameter values.

Different terminology:

- Calibration
- Data assimilation
- Parameter estimation
- Inverse-problem
- Bayesian inference



Calibration requires

- Statistical model relating simulator to the data
  - All models are wrong etc.... but how do we characterize the error?
  - The aim of inference should influence your approach. Is the aim parameter estimation or calibrated prediction?
- Inferential framework
  - Maximum likelihood or some other optimization approach
  - Bayesian
  - History matching
- Computational inference scheme
  - MCMC, emulators, etc

Bayesian calibration

Kennedy and O'Hagan 2001

The calibration framework used is:

$$\mathcal{D} = \eta(\theta) + \delta + \epsilon$$

- All quantities are multidimensional (e.g. time-series, spatial fields)
- ϵ represents measurement error
  - Often $\epsilon \sim N(0, \Sigma)$
- δ represents simulator discrepancy
  - Standard approach is to assume $\delta(\cdot) \sim GP(m(\cdot), c(\cdot, \cdot))$

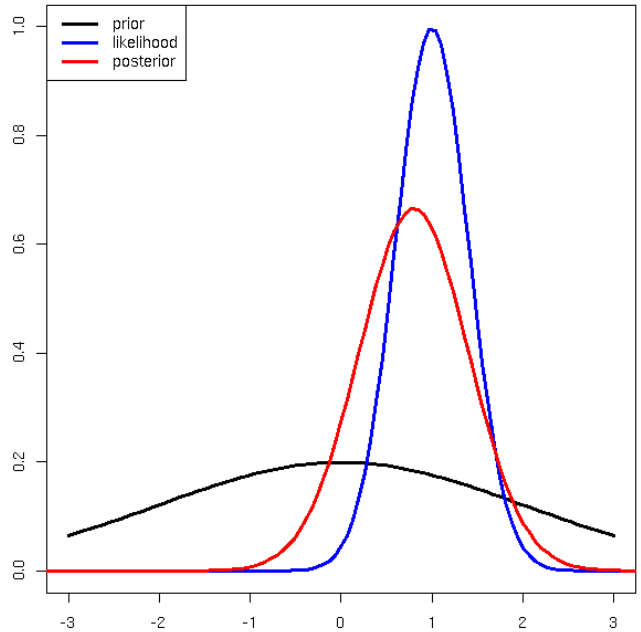
Statistical specification is complete once all unknowns (θ, m, c, Σ etc) have prior distributions specified.

Bayesian calibration

Kennedy and O'Hagan 2001

We then aim to find the posterior distribution

$$\pi(\theta \mid \mathcal{D}) = \frac{\pi(\theta)\pi(\mathcal{D} \mid \theta)}{\pi(\mathcal{D})}$$

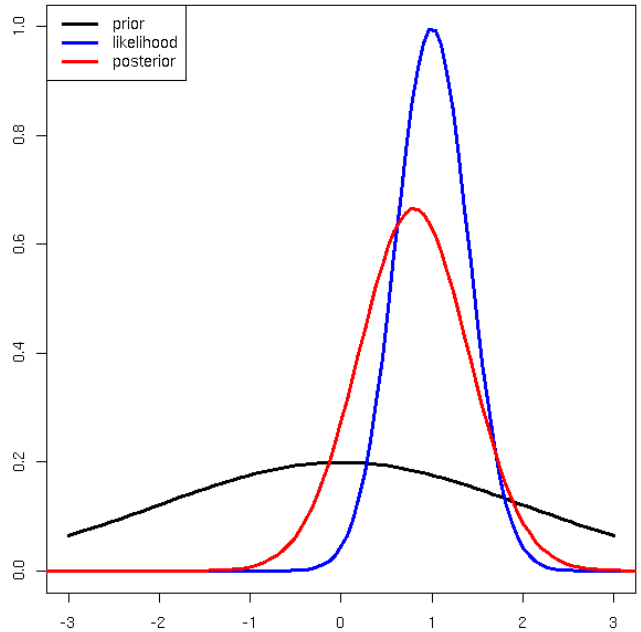


Bayesian calibration

Kennedy and O'Hagan 2001

We then aim to find the posterior distribution

$$\pi(\theta \mid \mathcal{D}) = \frac{\pi(\theta)\pi(\mathcal{D} \mid \theta)}{\pi(\mathcal{D})}$$



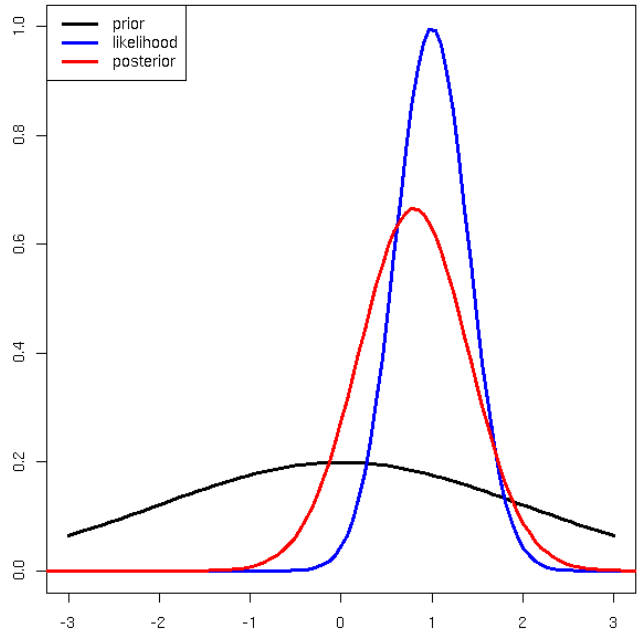
- Strength of the approach is that it combines all available sources of information
- Weakness is that it can be hard to specify priors and a statistical model for the discrepancy
- Also the are identifiability issues between θ and δ (Brynjarsdóttir and O'Hagan 2014) (prediction or parameter estimation?)

Bayesian calibration

Kennedy and O'Hagan 2001

We then aim to find the posterior distribution

$$\pi(\theta \mid \mathcal{D}) = \frac{\pi(\theta)\pi(\mathcal{D} \mid \theta)}{\pi(\mathcal{D})}$$



- Strength of the approach is that it combines all available sources of information
- Weakness is that it can be hard to specify priors and a statistical model for the discrepancy
- Also the are identifiability issues between θ and δ (Brynjarsdóttir and O'Hagan 2014) (prediction or parameter estimation?)

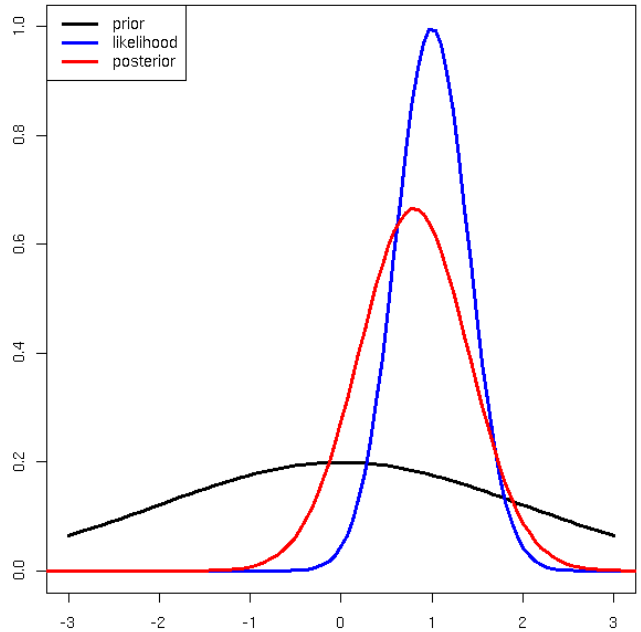
Bayesian calibration

Kennedy and O'Hagan 2001

We then aim to find the posterior distribution

$$\pi(\theta \mid \mathcal{D}) = \frac{\pi(\theta)\pi(\mathcal{D} \mid \theta)}{\pi(\mathcal{D})}$$

For all but the simplest problems, this calculation is hard!



- Strength of the approach is that it combines all available sources of information
- Weakness is that it can be hard to specify priors and a statistical model for the discrepancy
- Also the are identifiability issues between θ and δ (Brynjarsdóttir and O'Hagan 2014) (prediction or parameter estimation?)

History matching

Craig *et al.* 1997

An alternative to Bayesian calibration is history matching.

Instead of a posterior distribution over parameter space, we aim to classify parameter space as implausible or (non-im)plausible.

$$I(\theta)^2 = \frac{(\mathbb{E}\eta(\theta) - D)^2}{Var(\delta) + Var(\epsilon) + Var(\eta)}$$

Declare θ implausible if $I(\theta) > 3$

- Advantages are that we only need specify variances for δ rather than a complete statistical model
- Conservative approach - aim is to rule out parameter values that are clearly bad, rather than find a posterior over good values
- Disadvantage is that it ignores a lot of information and doesn't allow probabilistic predictions.

Emulating ice sheet models

Chang, Haran, Applegate, Pollard 2016

- PSU3D-ICE model used to simulate long-term evolution of the West Antarctic Ice Sheet
  - Parametric uncertainty is important source of uncertainty in future projections of WAIS volume change
- They use modern and palaeo data to calibrate the simulator
  - Time series of grounding line positions (location of the transition from grounded ice to ice shelf) along the central flowline in the Amundsen Sea Embayment (ASE) since the LGM (RAISED consortium 2014)
  - Modern binary spatial pattern of presence and absence of grounded ice in the ASE (Bedmap2 dataset).
- Aim to calibrate 4 simulator parameters
  - sub-ice-shelf oceanic melt factor (OCFAC)
  - calving factor (CALV)
  - basal sliding coefficient (CRH)
  - asthenospheric relaxation e-folding time (TAU)

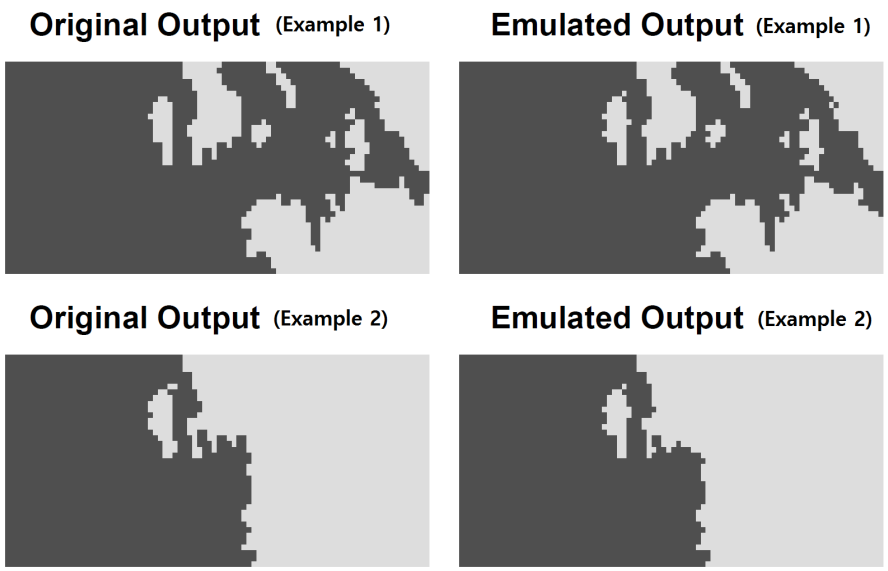
Emulating ice sheet models

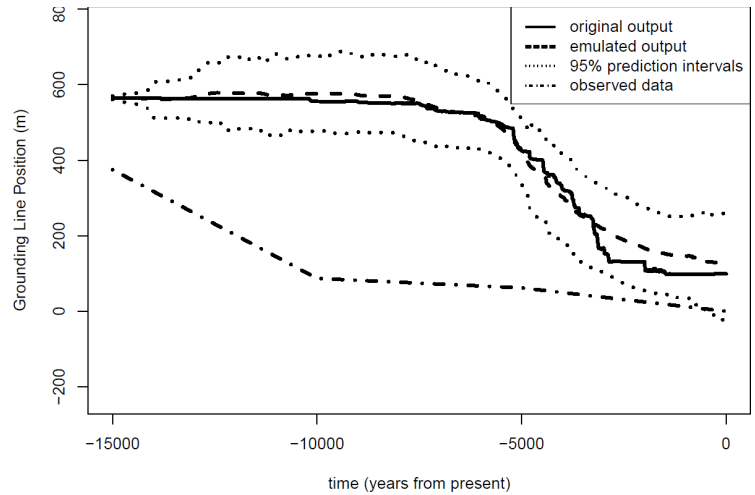
Chang, Haran, Applegate, Pollard 2016

The used an ensemble of 461 simulator evaluations

- Each time series has 1500 points
- Spatial map information at 3182 grid points

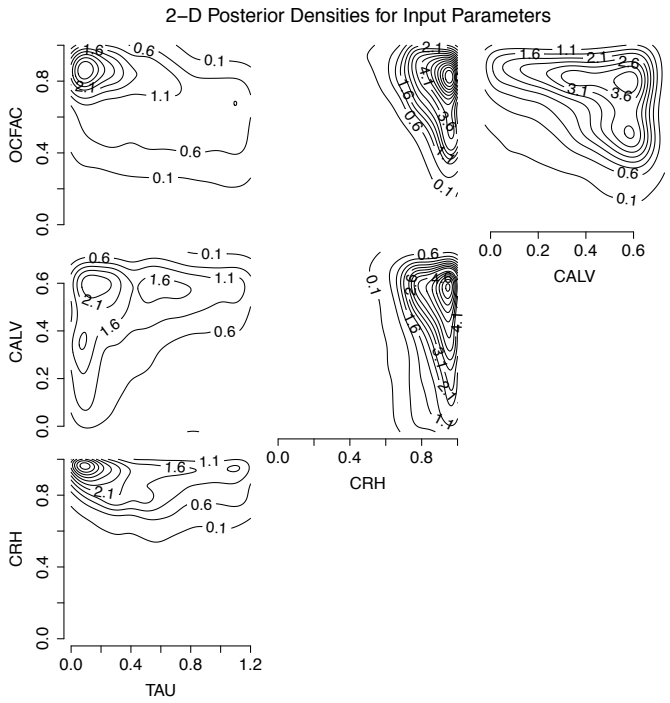
They use a PCA-emulator (with logit likelihood) and probabilistic calibration.



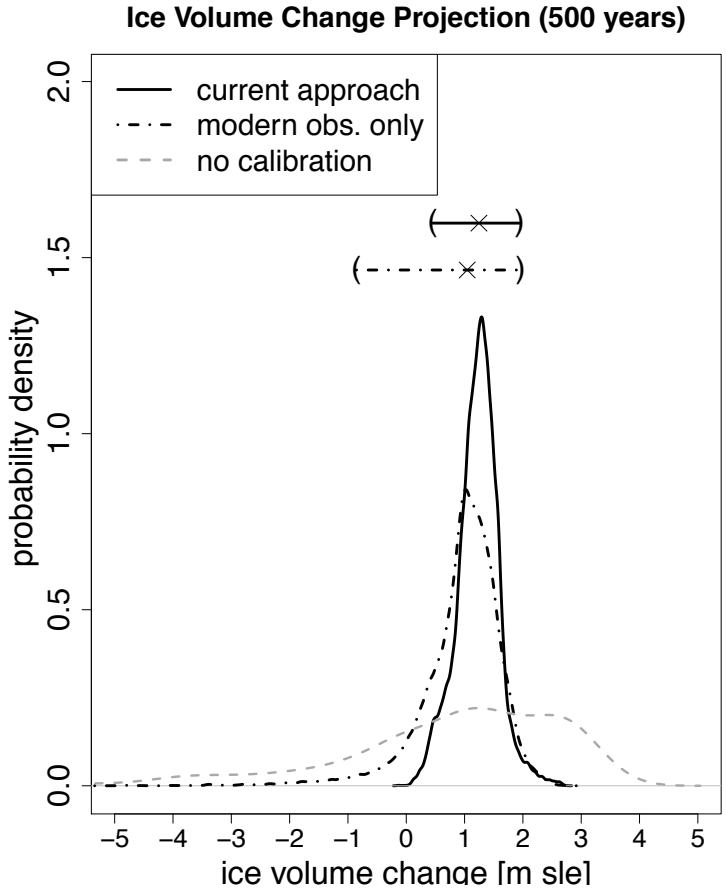


Emulating ice sheet models

Chang, Haran, Applegate, Pollard 2016

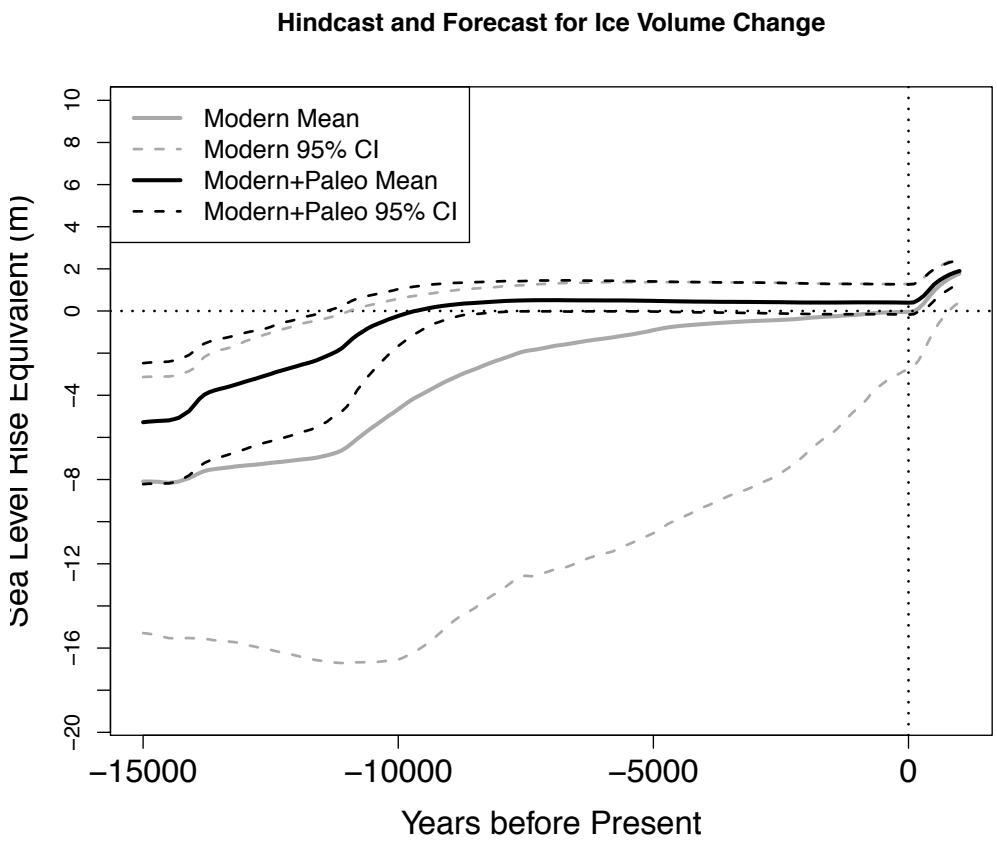


(b) Modern Binary Patterns and Past Grounding Line Positio



Emulating ice sheet models

Chang, Haran, Applegate, Pollard 2016



Tuning NEMO with history matching

WIlliamson, Blaker, Sinha 2016

- Tune the NEMO ORCA2 global ocean model run at 2deg resolution.
- Vary 20 ocean parameters
- Initial ensemble of 400 simulations
- Tune the model to metrics derived from the 1960-1990 climatological mean depth profiles of global mean temperature and salinity computed from the EN3 climatology

WIlliamson, Blaker, Sinha 2016

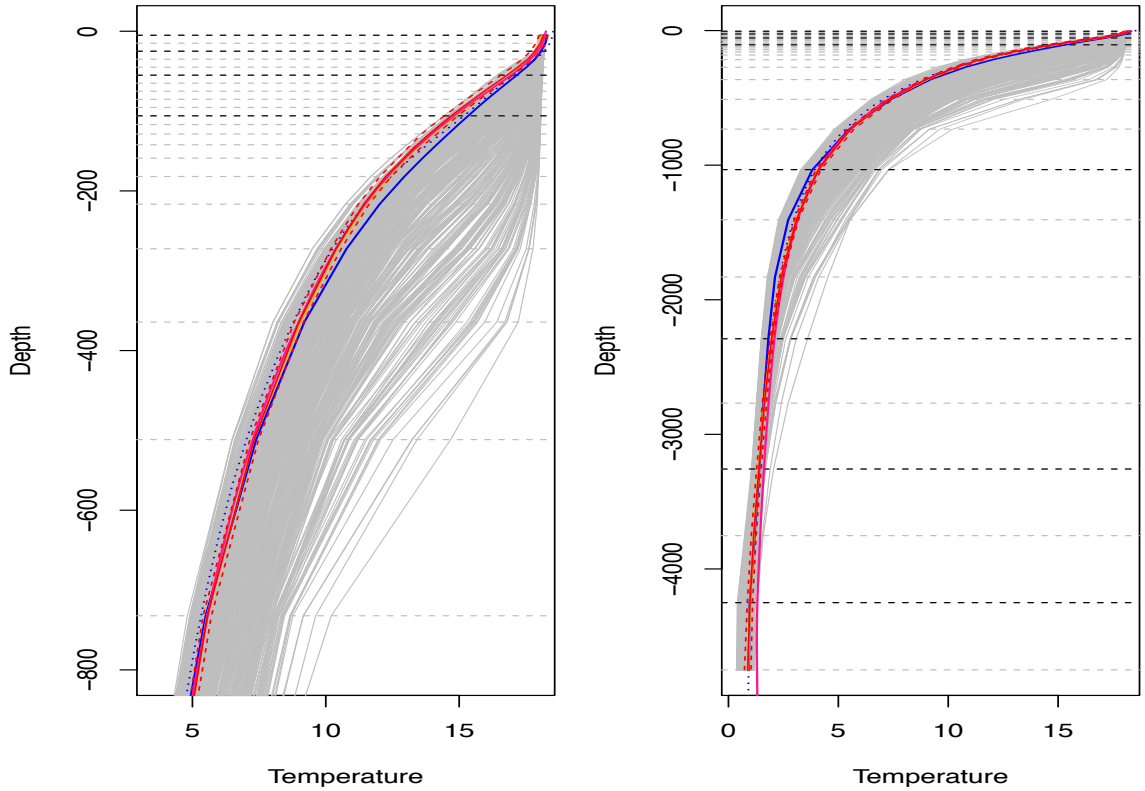


Figure 1. Global mean potential temperature as a function of depth from: EN3 (red, with error bounds indicated by red dashed lines); all wave 1 ensemble members (grey); standard ORCA2 (dark blue); World Ocean Atlas (pink, Locarnini et al. (2013)); the initial state (gold); and GO5 averaged over years 1996-2005 (Megann et al. (2014)) (blue dotted). The left panel shows a vertical zoom of the top 800 m, whilst the right panel shows the full depth.

WIlliamson, Blaker, Sinha 2016

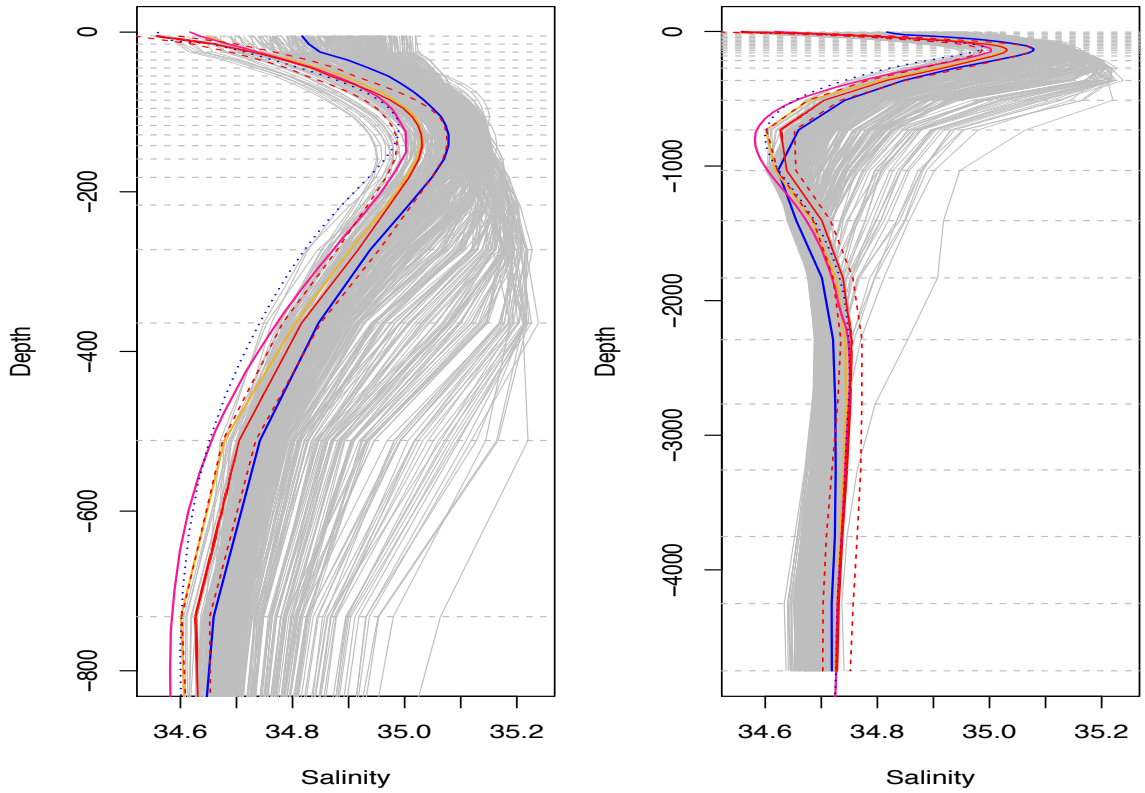


Figure 2. Global mean salinity as a function of depth from: EN3 (red, with error bounds indicated by red dashed lines); all wave 1 ensemble members (grey); standard ORCA2 (dark blue); World Ocean Atlas (pink, Zweng et al. (2013)); the initial state (gold); and GO5 averaged over years 1996-2005 (Megann et al. (2014)) (blue dotted). The left panel shows a vertical zoom of the top 800 m, whilst the right panel shows the full depth.

WIlliamson, Blaker, Sinha 2016

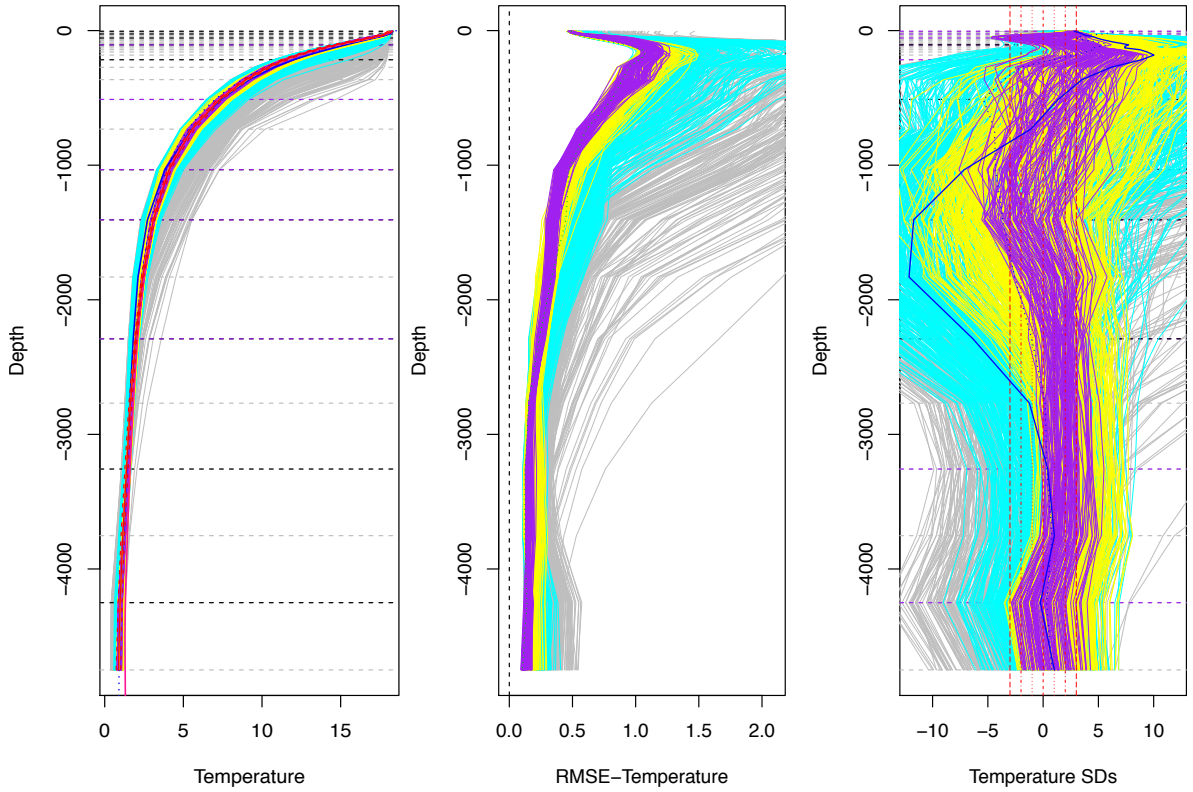


Figure 6. Global mean depth profiles of temperature (left), temperature RMSE (centre), and temperature depth profiles cast as departure from the EN3 global mean profile in units of σ. Colours represent W3 NROY (purple), W2 NROY (yellow), W1 NROY (cyan) and W1 RO (grey). The standard configuration is shown in solid blue, GO5 (ORCA025) as dotted blue, the initial conditions (magenta), WOA (gold).

WIlliamson, Blaker, Sinha 2016

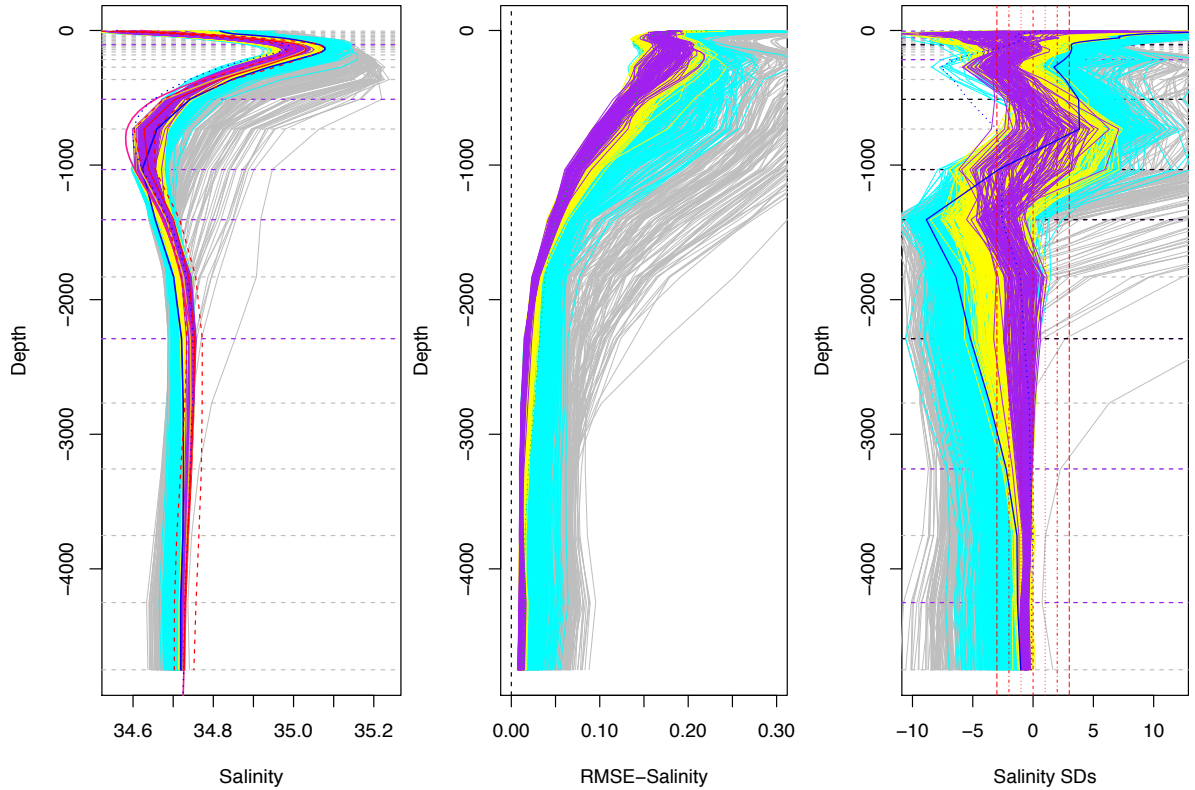


Figure 7. Global mean depth profiles of salinity (left), salinity RMSE (centre), and salinity depth profiles cast as departure from the EN3 global mean profile in units of σ. Colours represent W3 NROY (purple), W2 NROY (yellow), W1 NROY (cyan) and W1 RO (grey). The standard configuration is shown in solid blue, GO5 (ORCA025) as dotted blue, the initial conditions (magenta), WOA (gold).

Conclusions

- For complex expensive simulators Gaussian process emulators can enable UQ approaches that would otherwise be impossible.
- For highly correlated multivariate output, principal component emulation can work well and is computationally cheap and easy to implement.
- A large number of output dimensions can be reduced to a smaller number of principal component scores which can then be emulated, accounting for any error in the compression.

Conclusions

- For complex expensive simulators Gaussian process emulators can enable UQ approaches that would otherwise be impossible.
- For highly correlated multivariate output, principal component emulation can work well and is computationally cheap and easy to implement.
- A large number of output dimensions can be reduced to a smaller number of principal component scores which can then be emulated, accounting for any error in the compression.

Thank you for listening!