Inference for Complex Numerical Models

Peter Challenor

The Alan Turing Institute

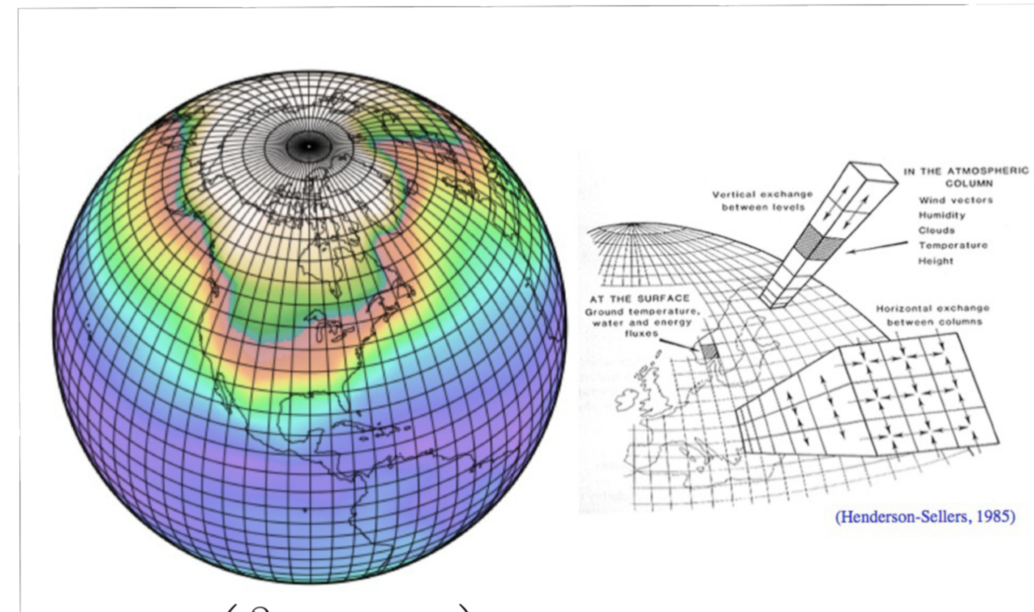
University of Exeter


COVID-19
Coronavirus
Vaccine

Complex Numerical Models

- Solve thousands of equations on very large computers

- Take many hours to run



$$\rho\left(\frac{\partial v}{\partial t}+v.\nabla v\right)=-\nabla p+\mu\nabla^2 v+f$$

- We cannot afford many runs

- But we want do inference

Complex Numerical Models

Real world Problem	**Mathematical Model PDEs**	**Discretised model FE/FD**	**Computer code**	**Emulator**

‘Black Box’ Model

- What is a black box model?
- We cannot change the model code
- (Non-intrusive methods)
- Work on propriety models or commercial codes

Inference

- Uncertainty quantification
- Sensitivity Analysis
- Uncertainty Analysis
- Inverse Modelling (calibration)

Two Levels of Inference

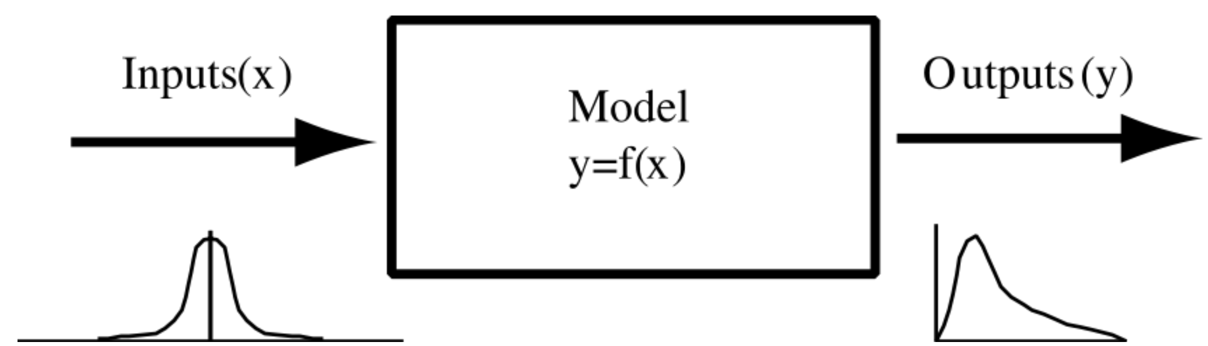
- Inference to build the emulator
- Inference to relate the numerical model to the real world (calibration, tuning, inverse modelling)

Building emulators
(Modelling Models)

- Use a Gaussian process (shallow learning)
- Include mean term; low order polynomials
- Could just use polynomials (lightweight emulators)
- Deep learning

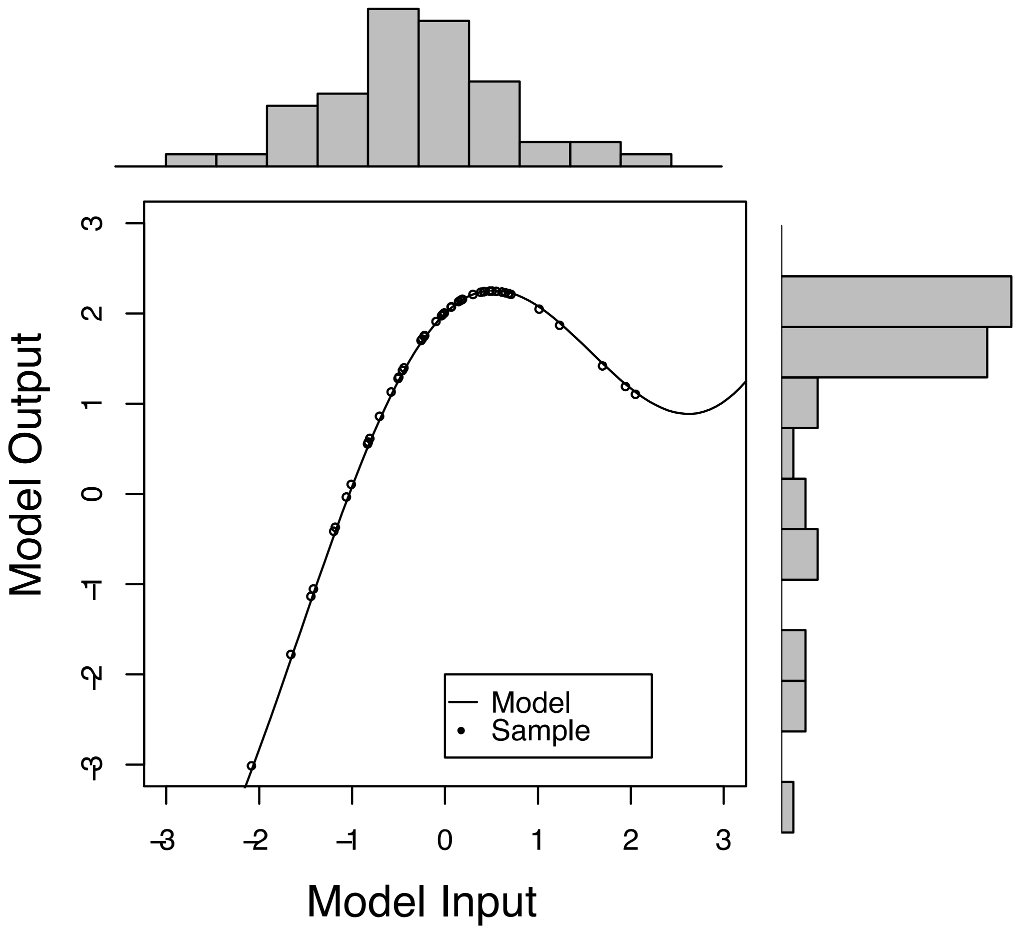
Where does the uncertainty come from?

- Even for deterministic models
  - The model inputs are uncertain
  - The model structure is uncertain
- Some models are themselves stochastic (COVID)



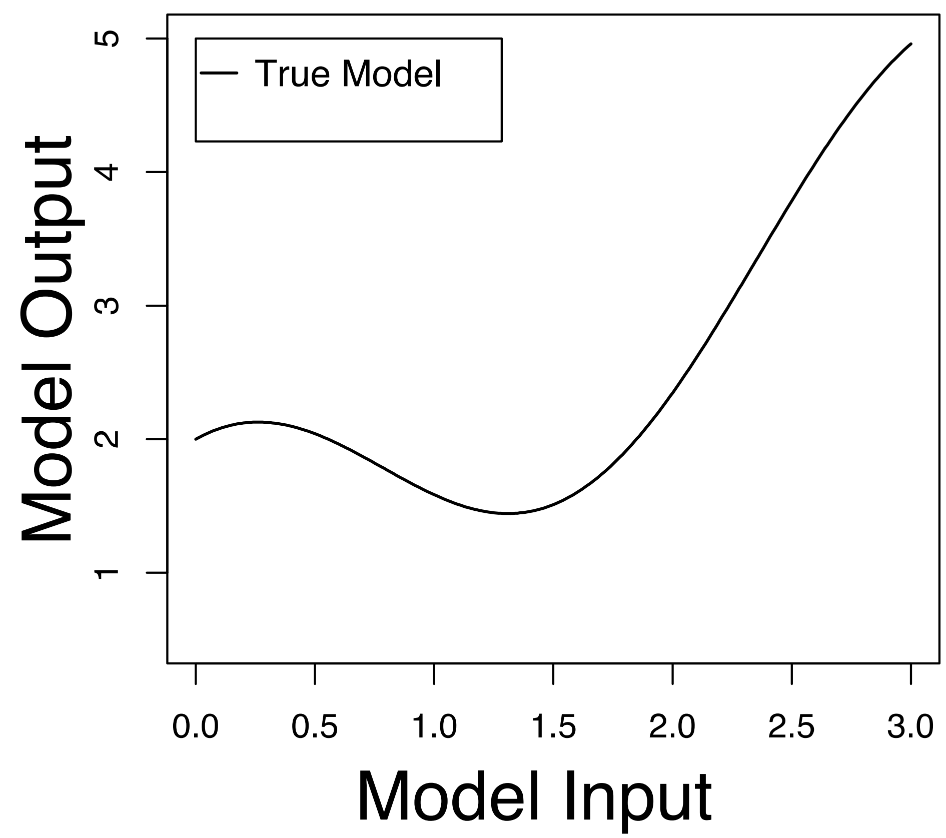
Monte Carlo

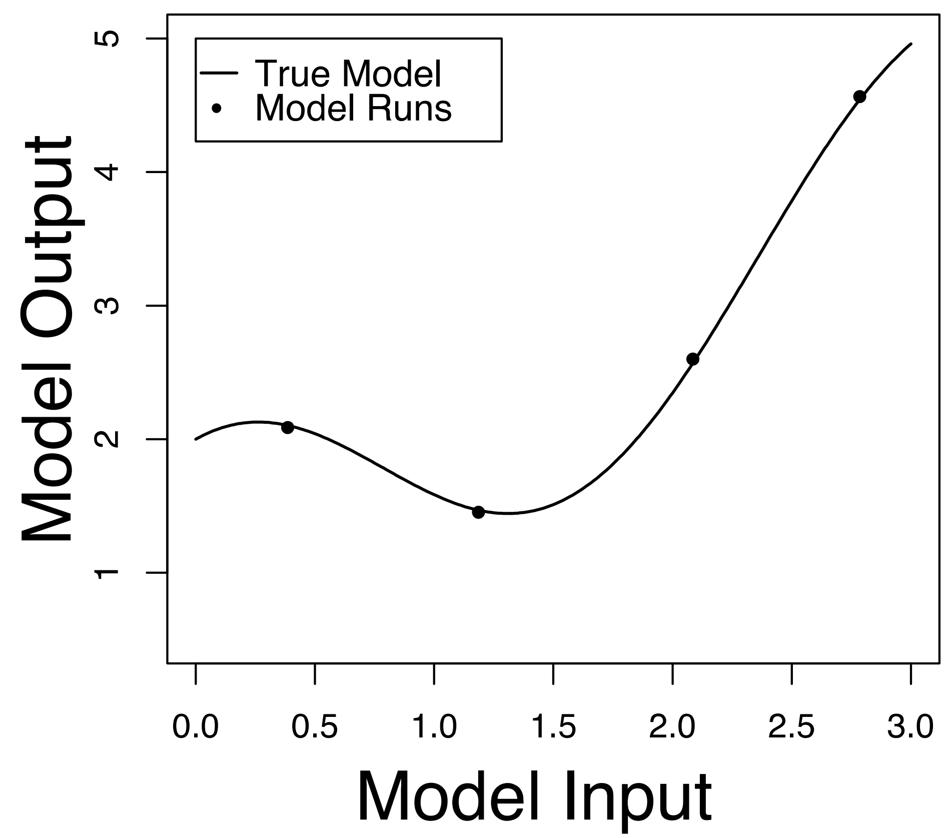
- Classical method
- Requires many thousands of runs
- We cannot afford that

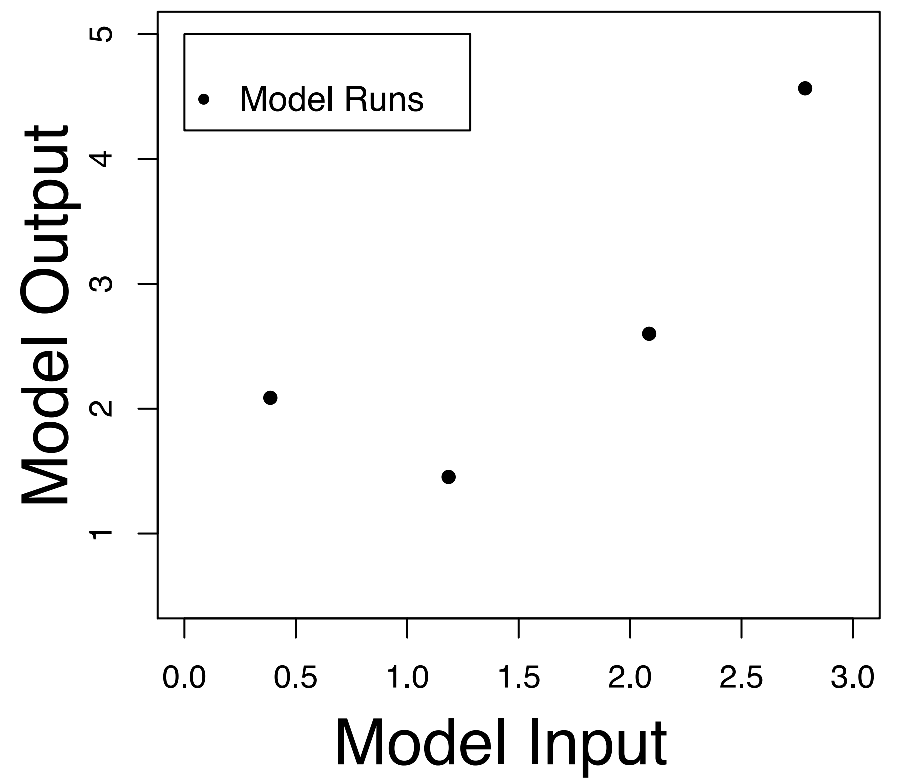


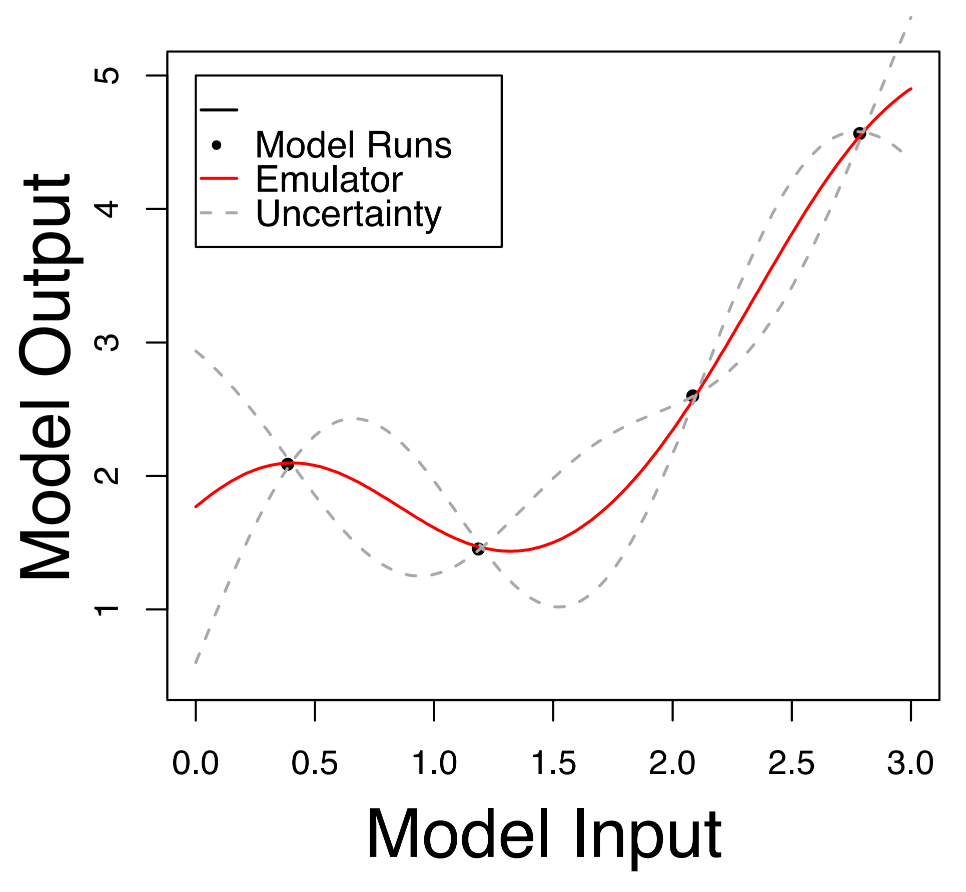
Emulators

- Emulators are *surrogate* models with the addition of a measure of uncertainty.
- We use Gaussian processes
- Emulators are *fast* to evaluate, less than a second on a laptop vs many hours on an HPC

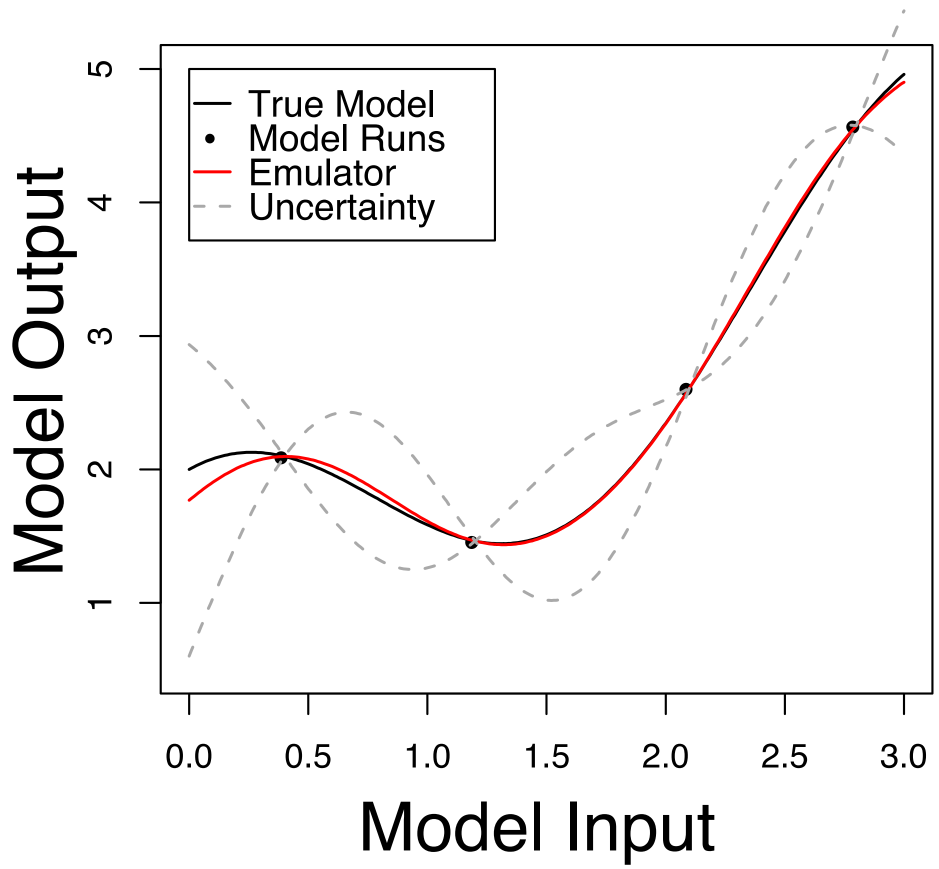

True Model
Model Output
Model Input
0.0
0.5
1.0
1.5
2.0
2.5
3.0
1
2
3
4
5


True Model
Model Runs
Model Output
Model Input
0.0
0.5
1.0
1.5
2.0
2.5
3.0
1
2
3
4
5


Model Runs
Model Output
Model Input
0.0
0.5
1.0
1.5
2.0
2.5
3.0
1
2
3
4
5



Model Runs
Emulator
Uncertainty
Model Output
Model Input
0.0
0.5
1.0
1.5
2.0
2.5
3.0
1
2
3
4
5

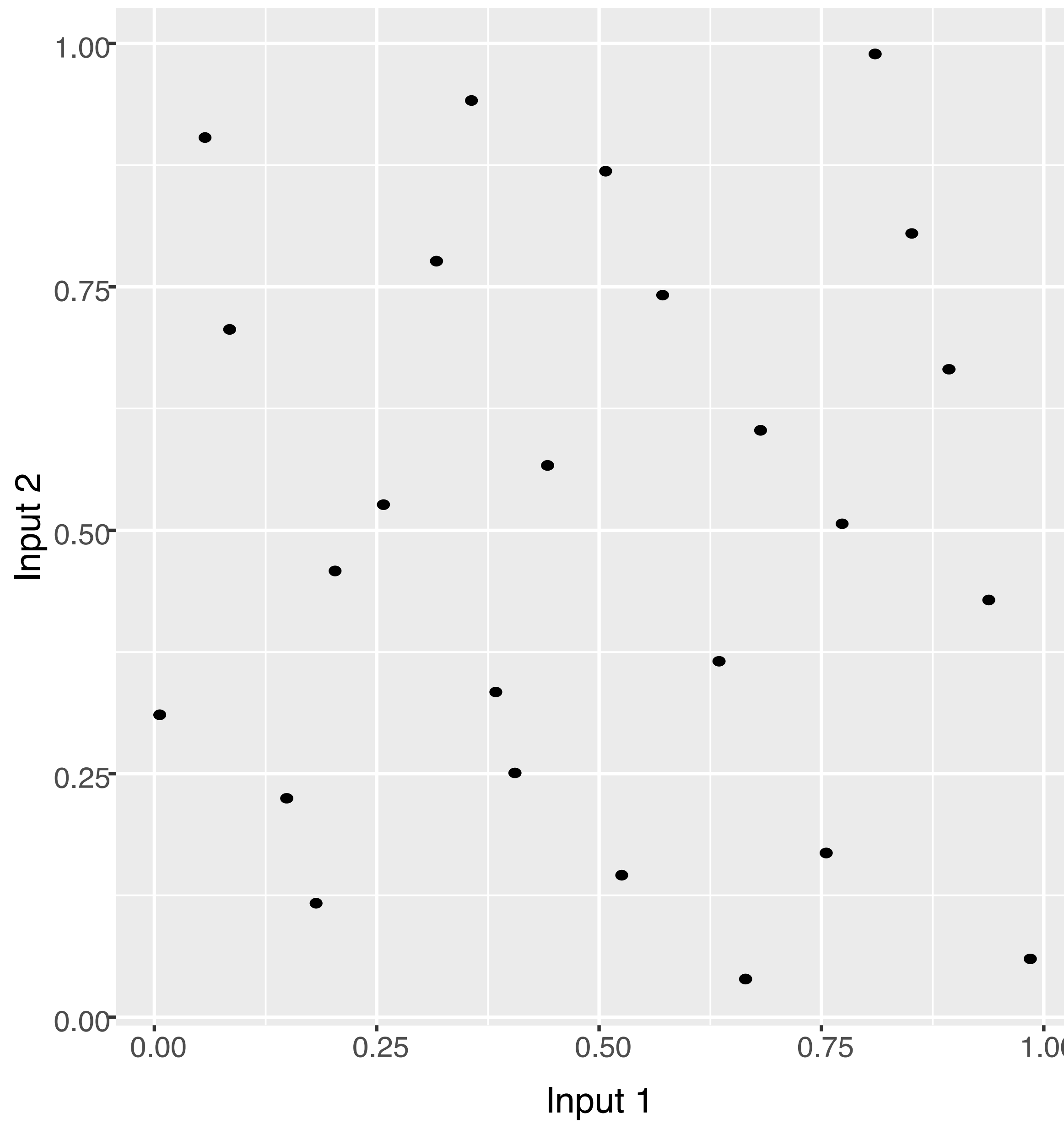

True Model
Model Runs
Emulator
Uncertainty
Model Output
Model Input
0.0
0.5
1.0
1.5
2.0
2.5
3.0
1
2
3
4
5

Uncertainty Quantification

- Prediction
- Sensitivity Analysis
- Uncertainty Analysis
- Model Calibration

Design

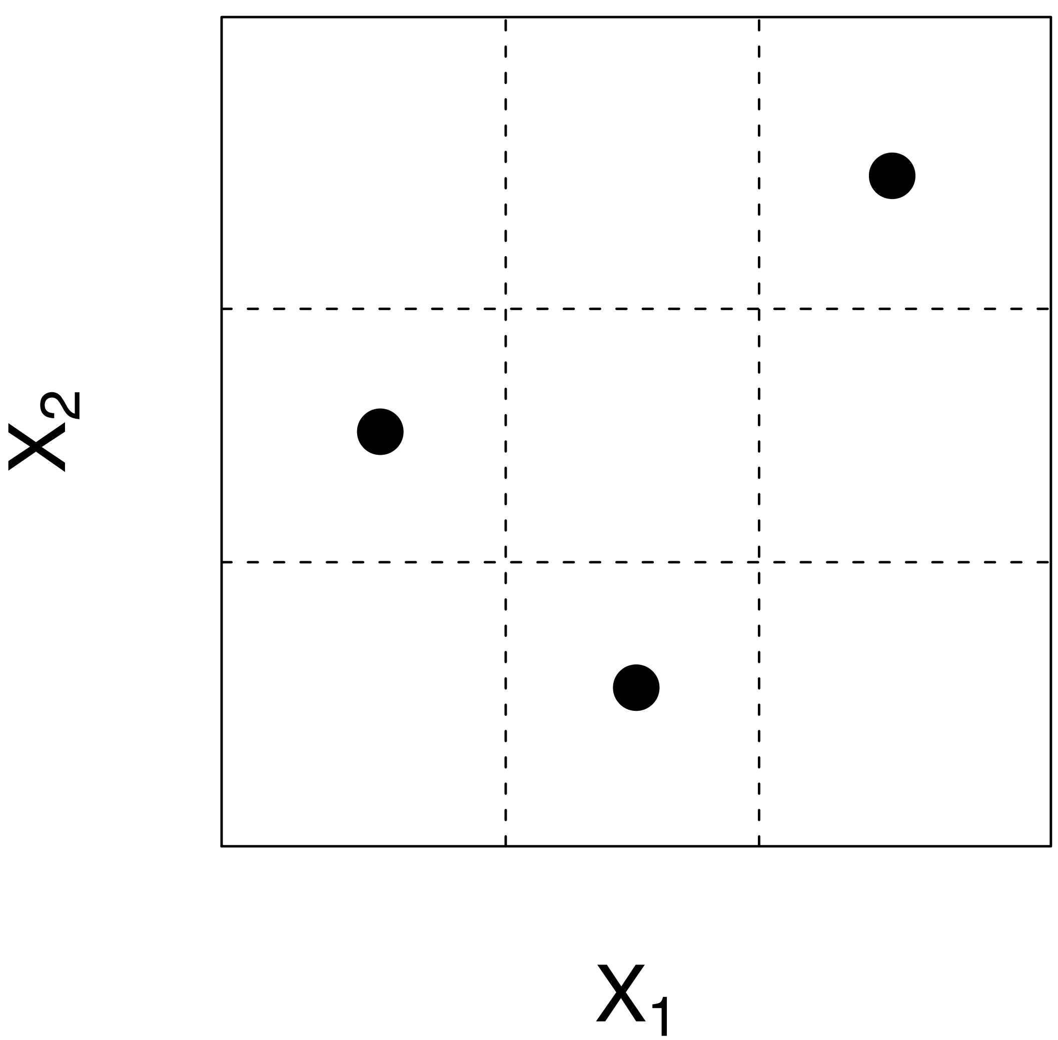
- Where do we do the model runs
- Space filling
- Sequential design

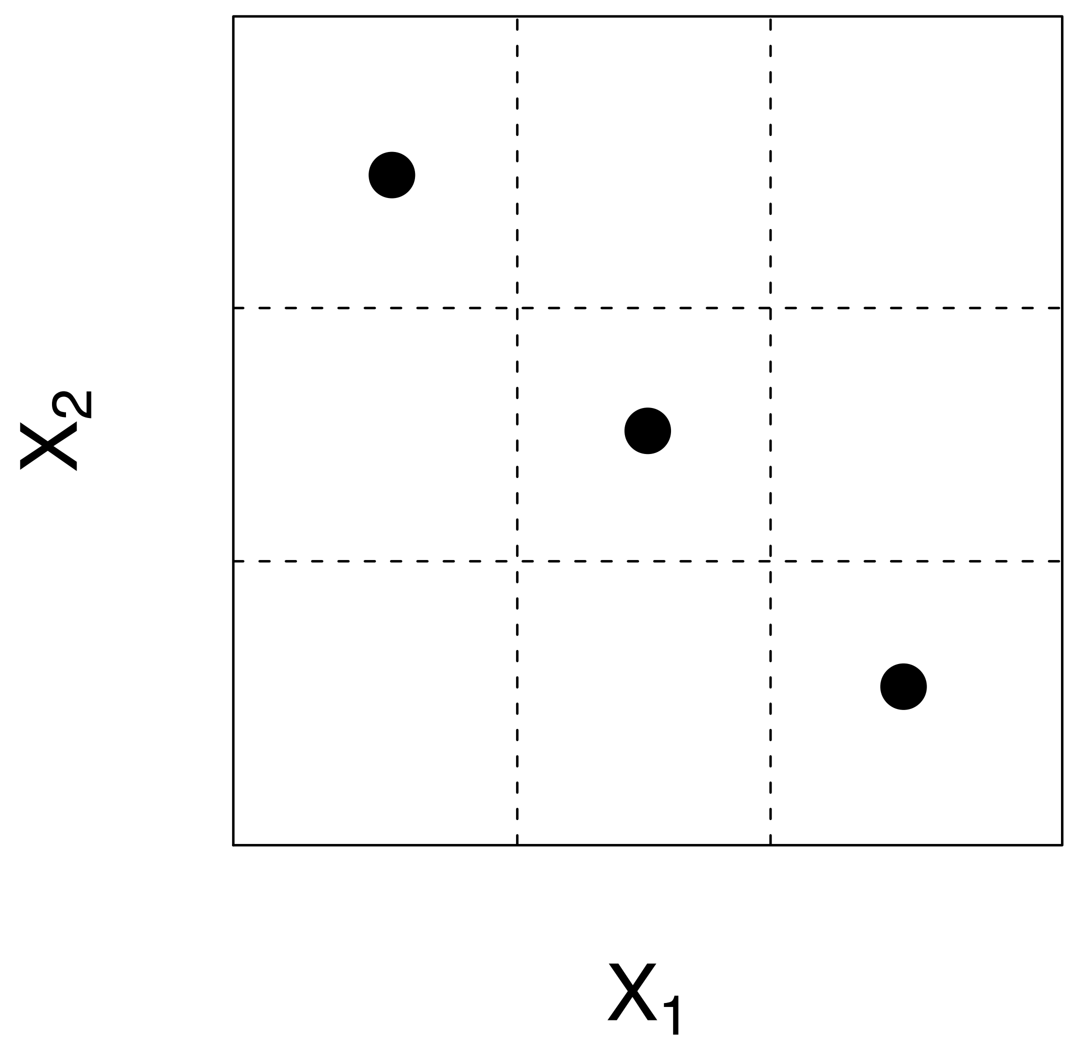


One Shot Designs

- Latin Hypercubes
- Maximin Latin hypercubes
- Low discrepancy sequences

The Latin Hypercube

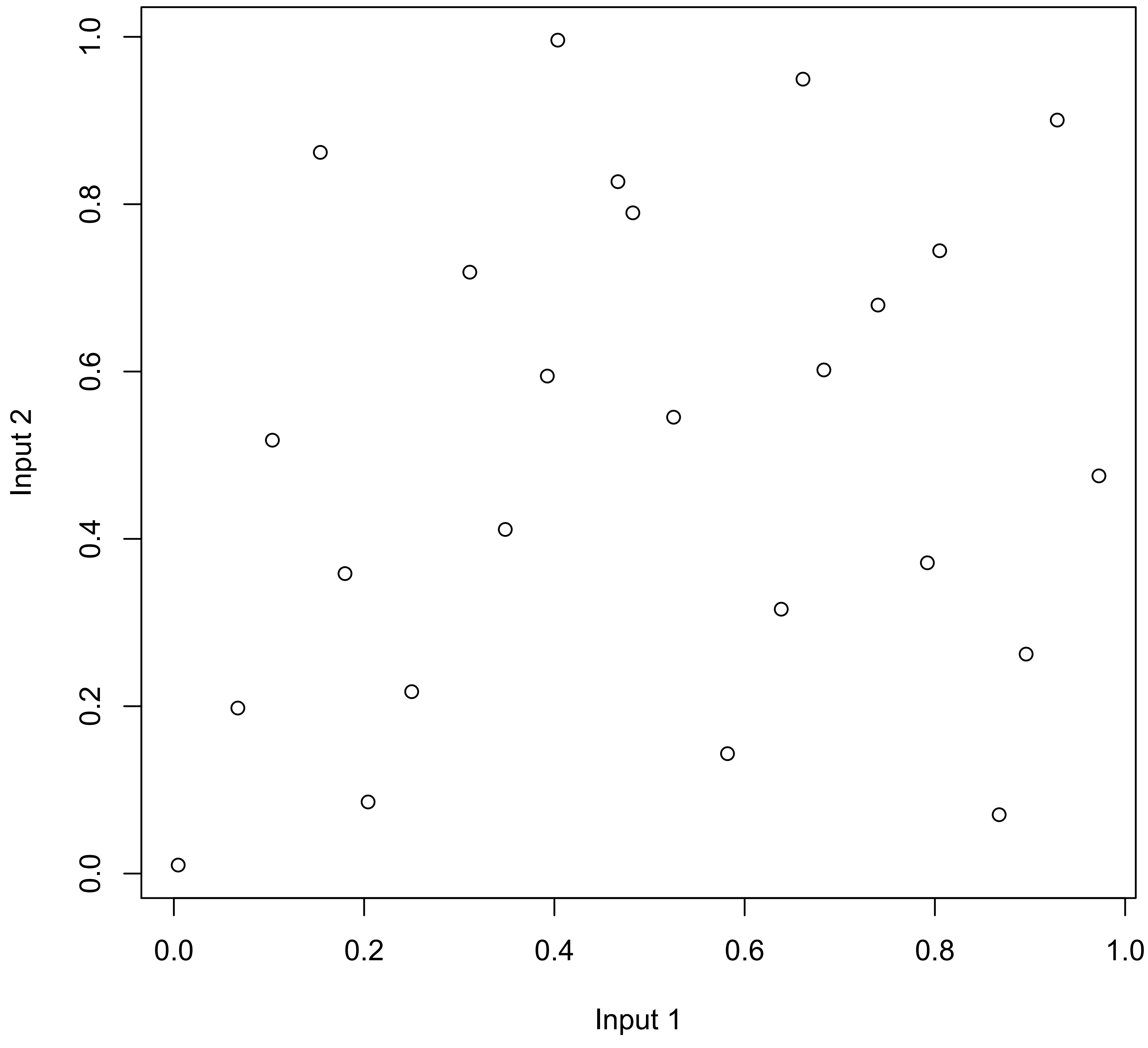




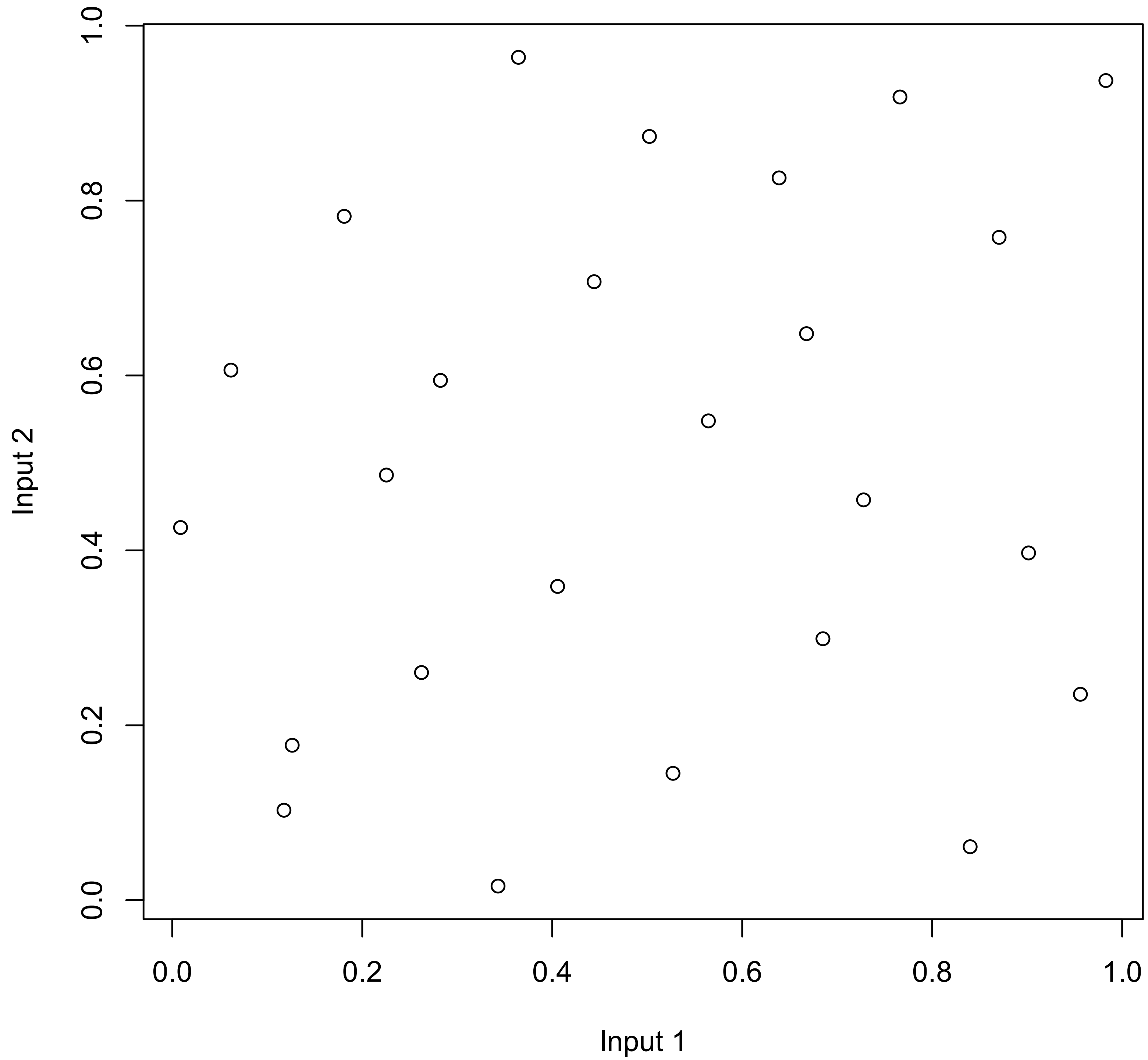
The Latin Hypercube

- We don’t have an algorithm for the optimal Latin hypercube
- What is a good Latin hypercube?
- Maximin
- Orthogonal designs

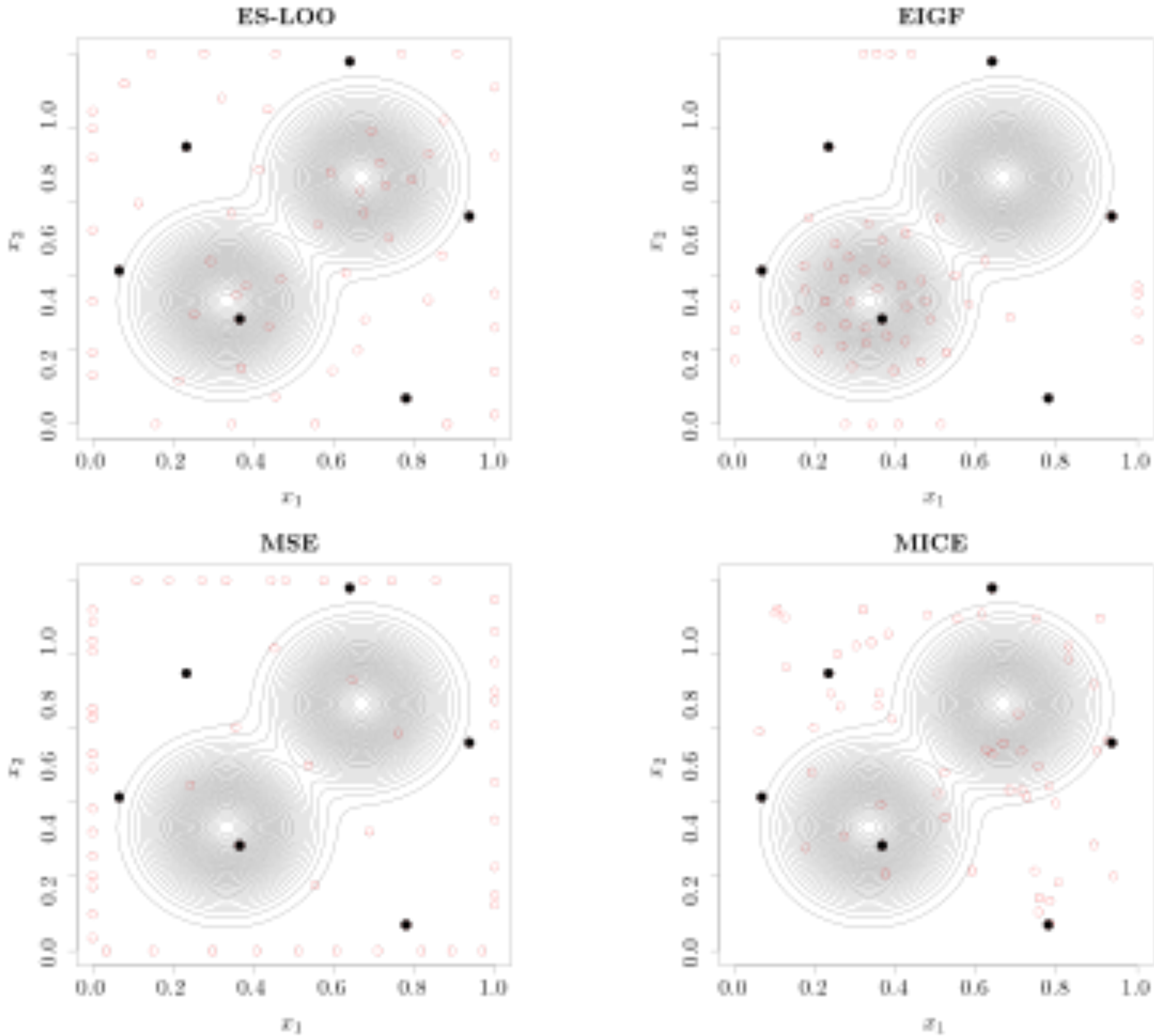
A Latin Hypercube

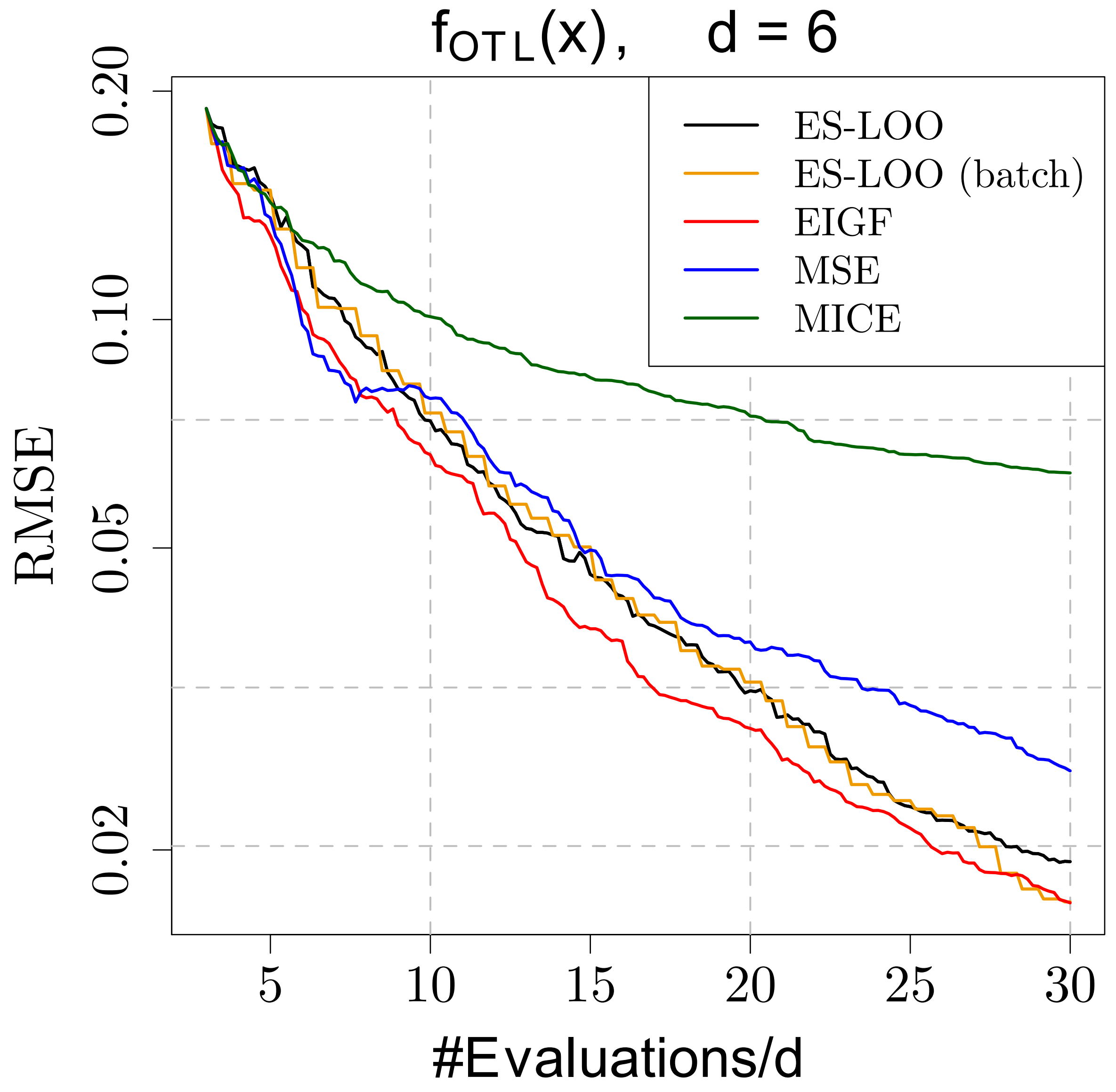


A maximin LHC

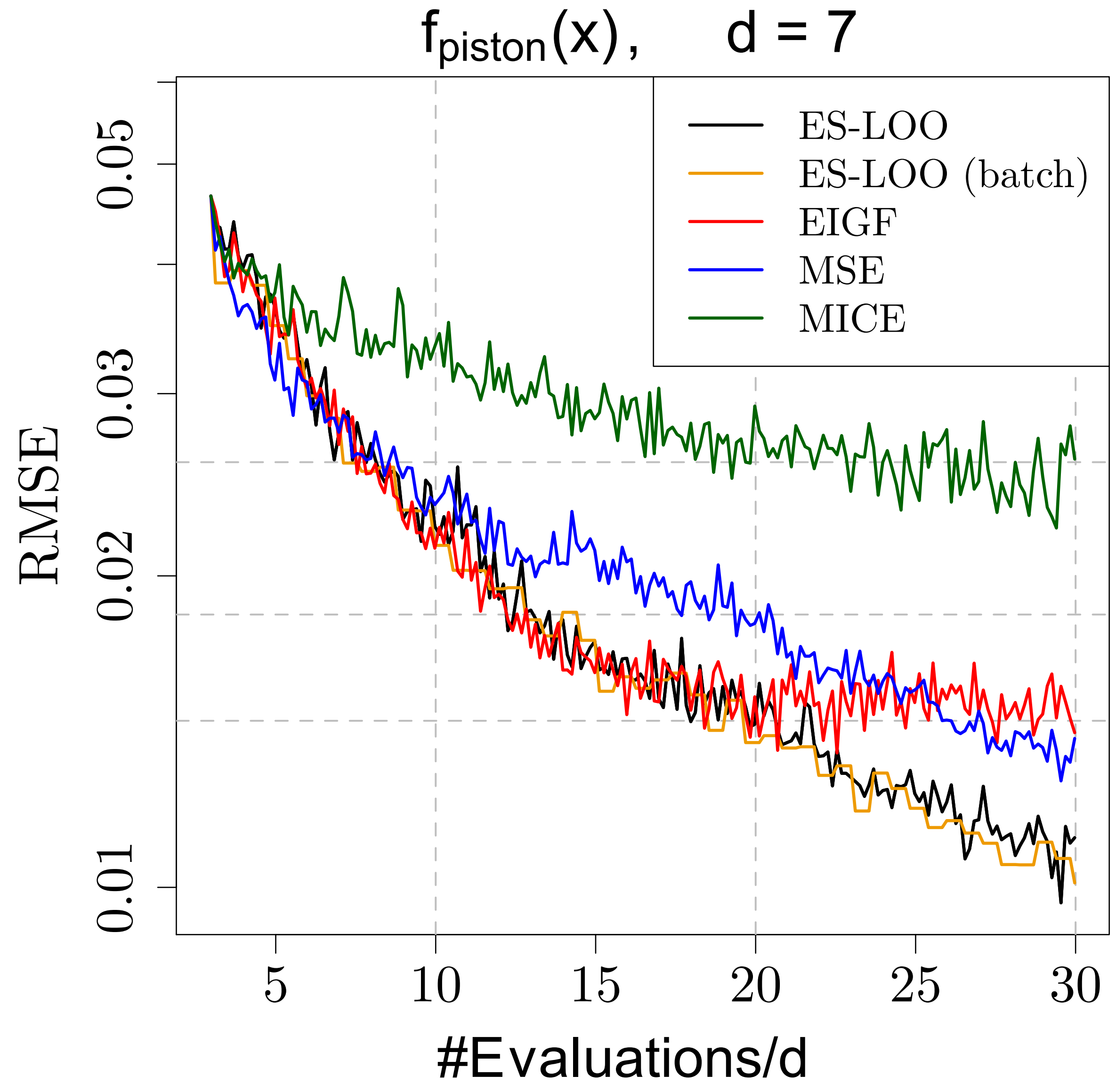


Sequential Designs





f_OTL(x), d = 6
RMSE
0.20
0.10
0.05
0.02
5
10
15
20
25
30
#Evaluations/d
ES-LOO
ES-LOO (batch)
EIGF
MSE
MICE



f_piston(x), d = 7
RMSE
0.05
0.03
0.02
0.01
5
10
15
20
25
30
#Evaluations/d
ES-LOO
ES-LOO (batch)
EIGF
MSE
MICE

Estimation for Emulation

- Either MLE or Bayes
- Non-linear parameters (length scales, nuggets) either plug in or full Bayes

The role of Nuggets

- A nugget is a white noise term added to the GP model
- In geo-statistics used for instrumental noise
- But we have deterministic models
- Stochastic models different
- Nuggets included for numerical reasons

- Likelihood surface complex
- Other maxima/posteriors
-

Relationship between models and the real world

- Models are designed to inform us about the real world
- They are not the same as the real world
- The real world is not a set of equations
- The discretised equations are not the continuum equations
- The code is not the discretised equations

All models are wrong, but some are useful

George Box

Model Discrepancy

- It is important to take model discrepancy into account
- Least squares or Bayesian calibration will give the wrong answer
- And the uncertainty will go to zero as you increase the amount of dat

Kennedy and O’Hagan (2001)

- Kennedy and O’Hagan came up with an ingenious solution
- Model the difference between the model and reality as the sum of two GPs
- One is the emulator of the model and the other is the discrepancy

Identifiability

- This fine for prediction (we know the sum of the GPs)
- But suffers from identifiability problems
- Strong priors
- Constrain the discrepancy or the emulator

An Alternative

- Don't try to find the 'best' set of inputs (x)
- Find inputs (x) that are *implausible* given the data (y)
- This is a lot easier
- No optimisation
- No sampling posterior

History Matching

- Set up a measure of the distance between the data and the model prediction

$$Imp = \sqrt{\frac{E(y - f(x))^2}{V(y - f(x))}}$$

- If this distance is too far. That value of $\boldsymbol{x}$ is implausible

- We can expand the variance term to give

$$Imp = \sqrt{\frac{(y - E(f(x)))^2}{V_y + V_{f(x)}}}$$

- Where V_y is the variance of y
- and $V_{f(x)}$ is the variance of $f(x)$
- For Imp >3 we say that the inputs (x) are implausible (Pukelsheim (1994))

- but could be expensive to run in which case we can only compute Imp in a small number of places

- Replace $f(x)$ with our emulator $f^*(x)$

- Expanding the variance as before gives

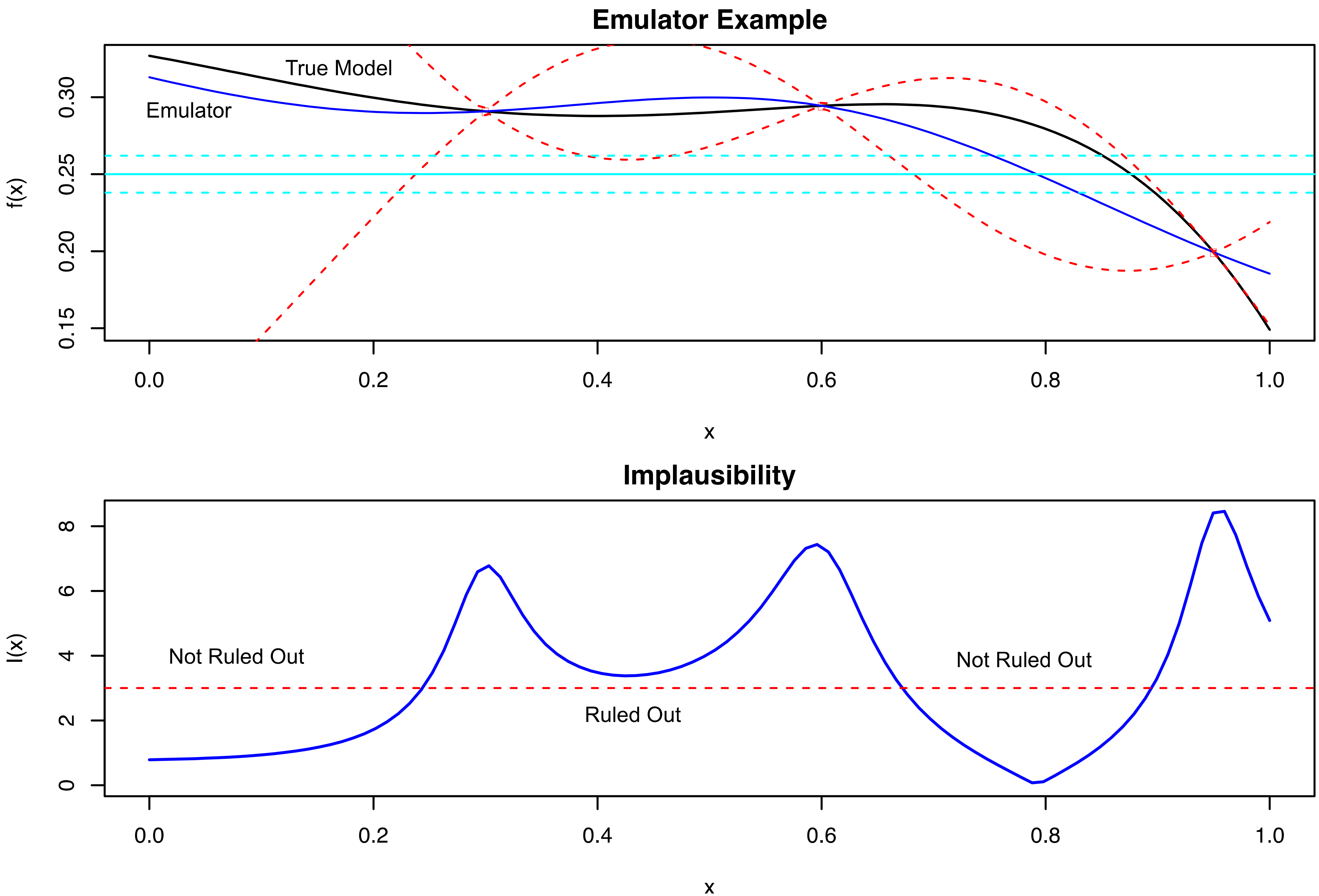
$$Imp = \sqrt{\frac{y - E(f(x))^2}{V_y + V_{emul} + V_{disc}}}$$

- V_y is the variance of the data y
- V_{emul} is the emulator variance
- V_{disc} is the model discrepancy

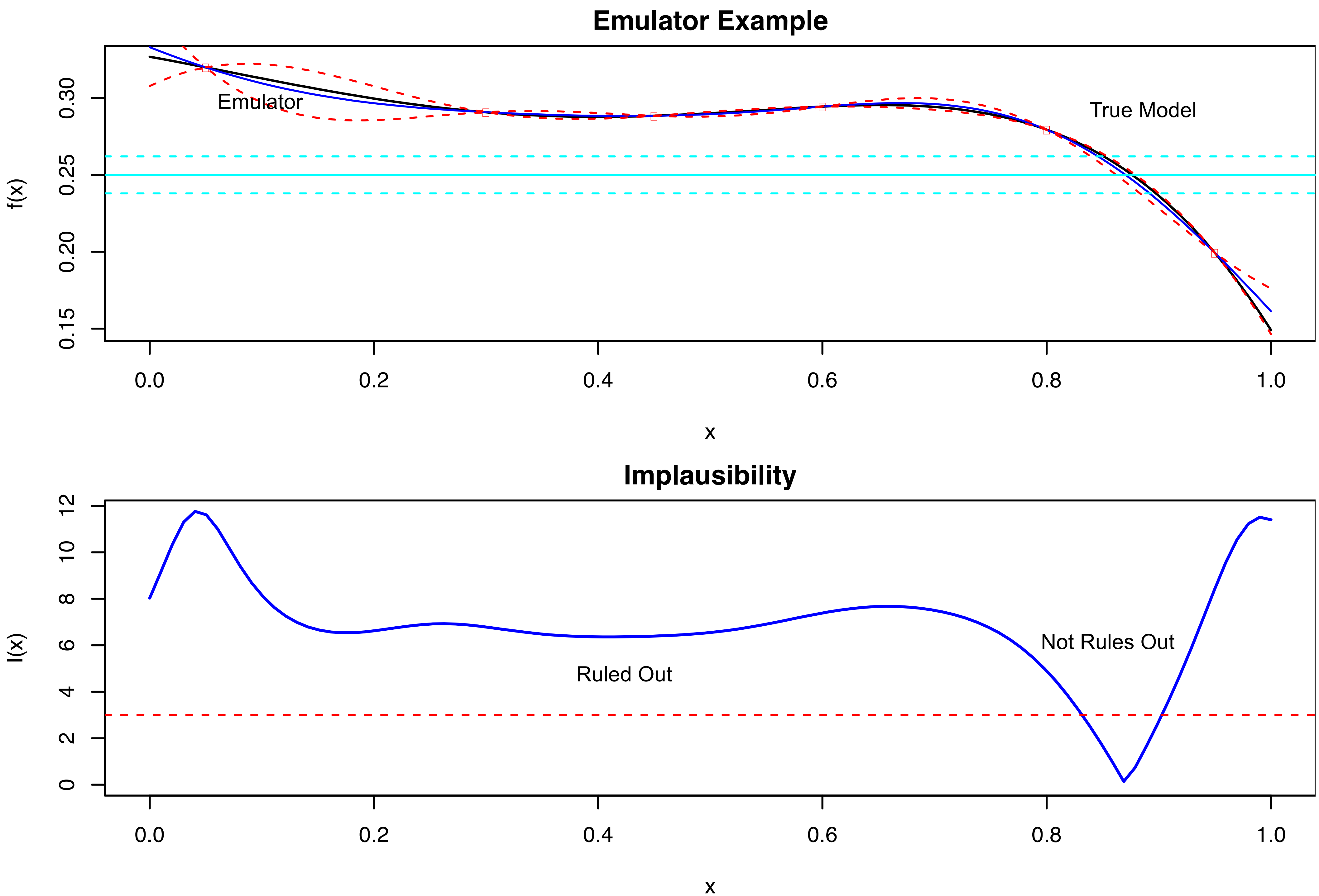
Procedure

- Collect data
- Run designed experiment
- Build emulator
- Perform history matching
- All points with `Imp` <3 deemed *not implausible*
- If we have many metrics take `max(Imp)`
- These constitute the *Not Ruled Out Yet* (NROY) space

- Design additional experiment within NROY space (wave 2)
- Rebuild emulator
- History match
- Repeat until NROY is either small enough or does not shrink
- At which point we may need more data

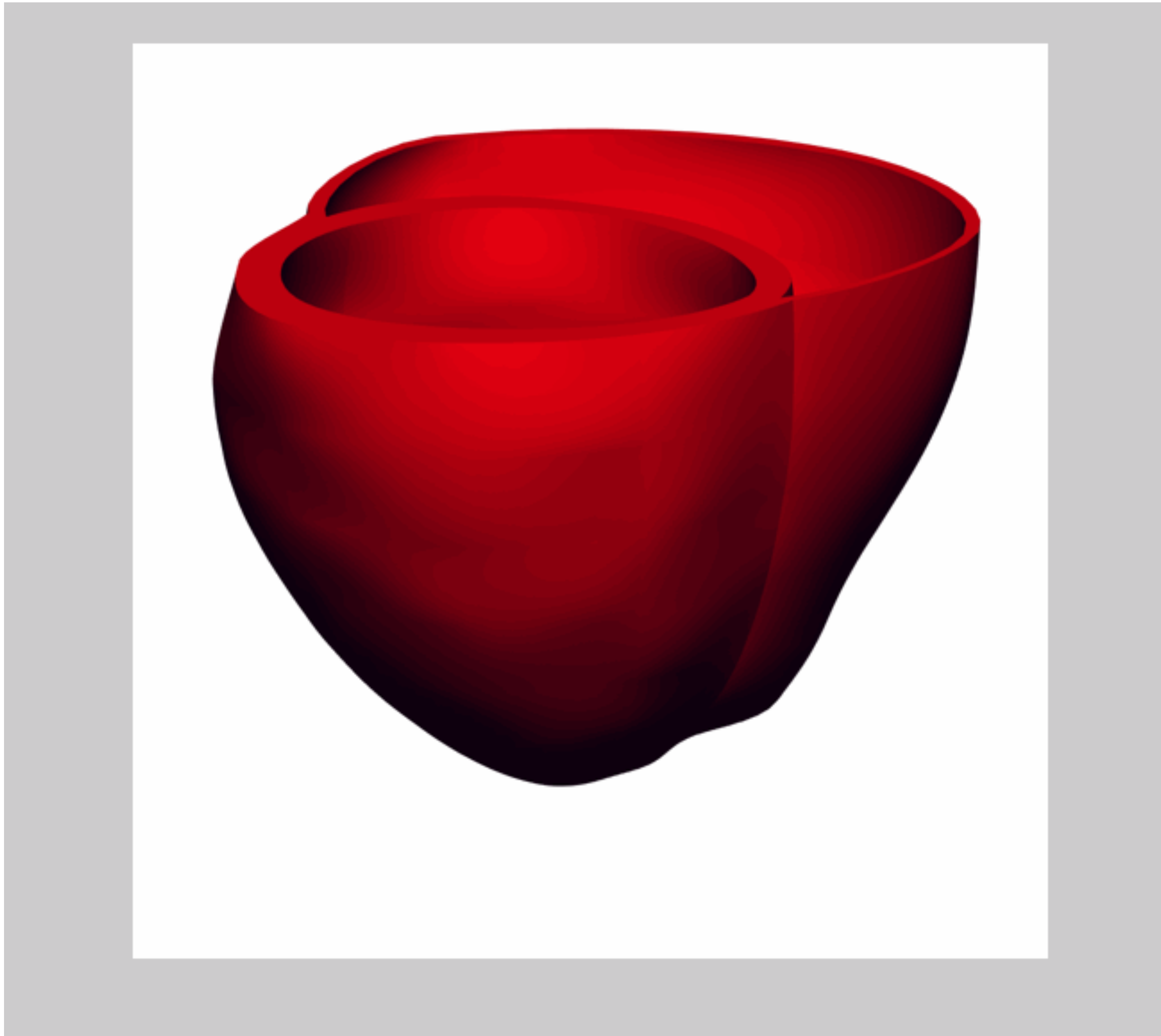


Emulator Example
True Model
Emulator
f(x)
0.15
0.20
0.25
0.30
0.0
0.2
0.4
0.6
0.8
1.0
x
Implausibility
Not Ruled Out
Ruled Out
Not Ruled Out
I(x)
0
2
4
6
8
0.0
0.2
0.4
0.6
0.8
1.0
x



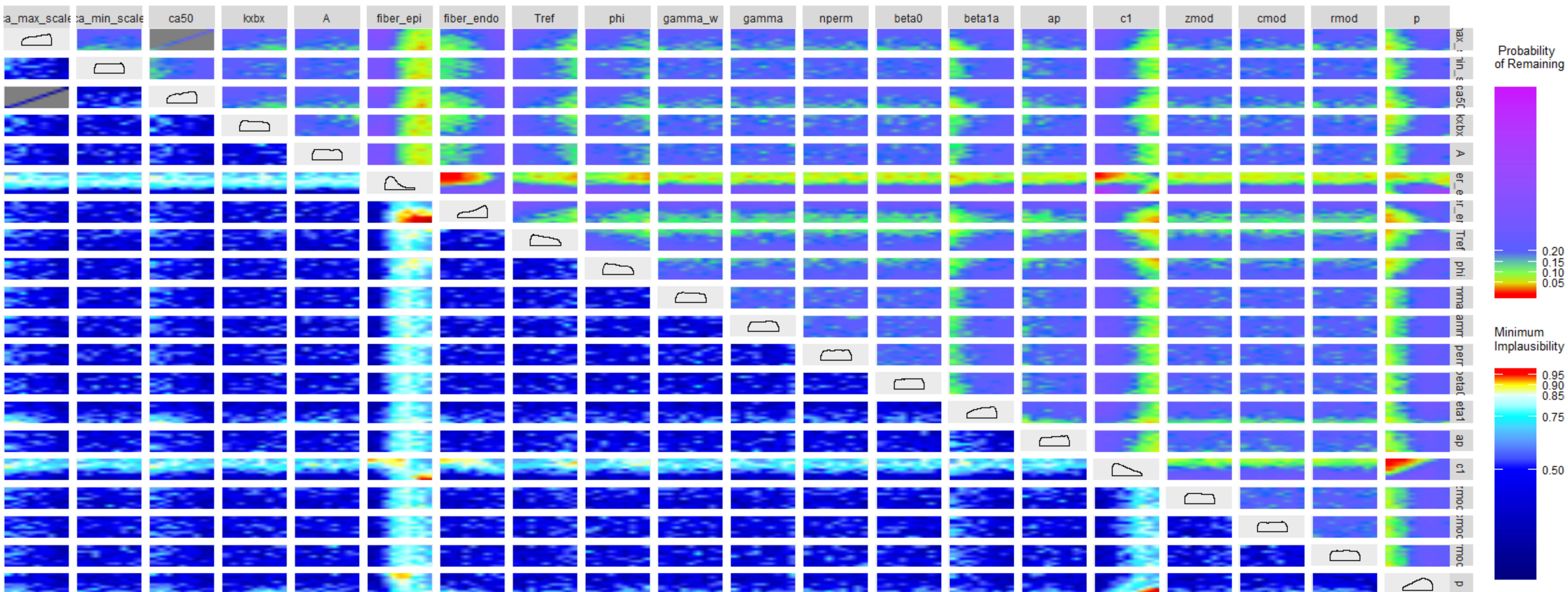
Emulator Example
f(x)
0.15
0.20
0.25
0.30
Emulator
True Model
0.0
0.2
0.4
0.6
0.8
1.0
x
Implausibility
I(x)
0
2
4
6
8
10
12
Ruled Out
Not Rules Out
0.0
0.2
0.4
0.6
0.8
1.0
x

A Cardiac Model

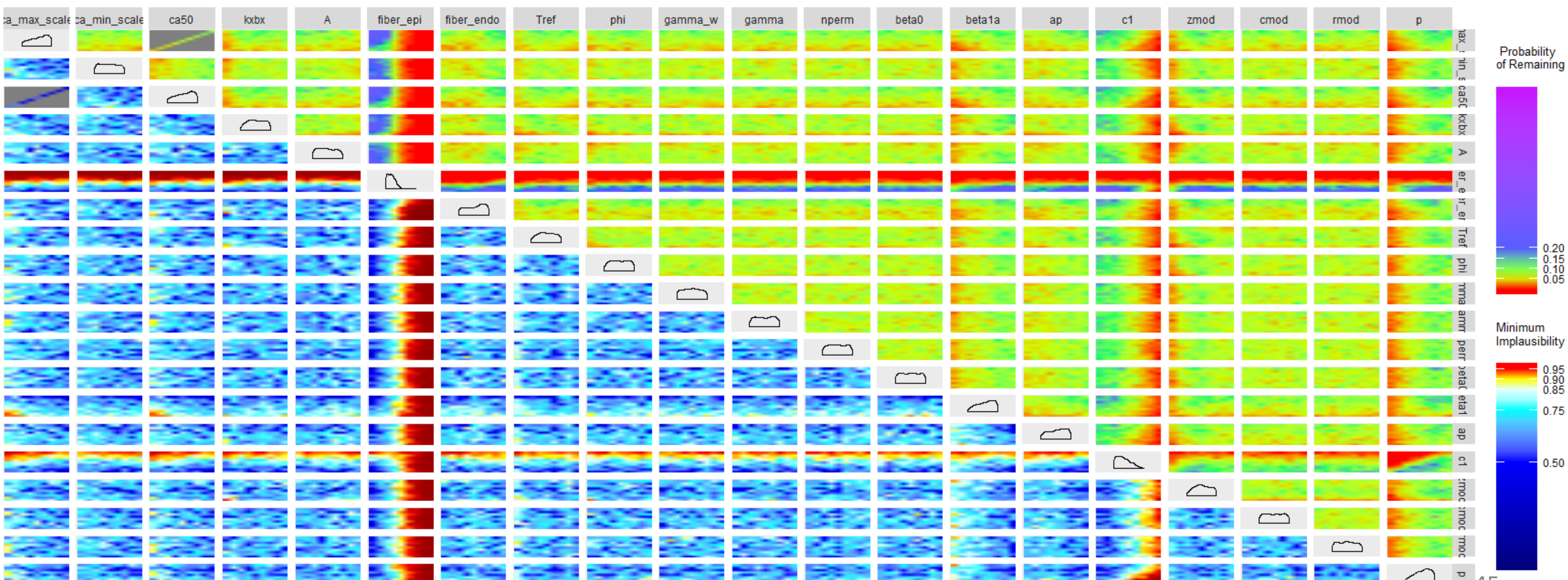


Thanks to Steve Neiderer, KCL/St Thomas

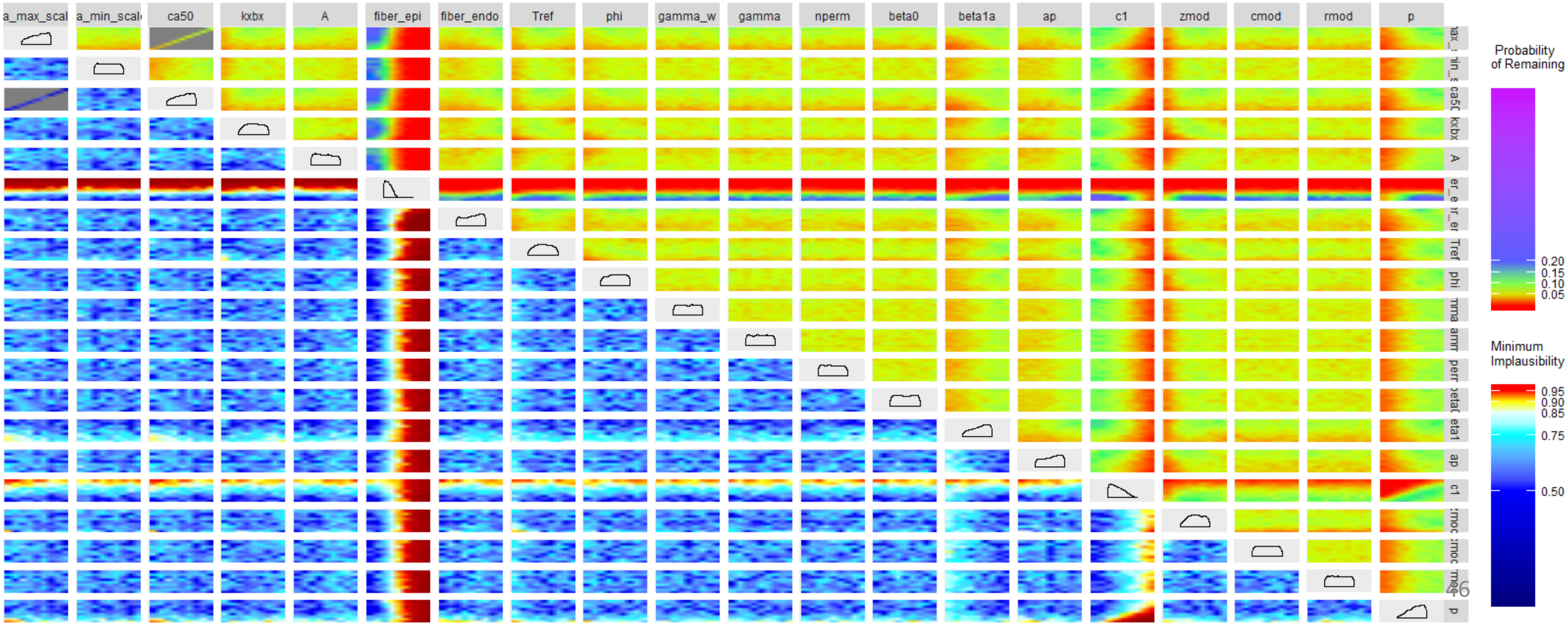
Wave 1: 25% of the parameter space remains

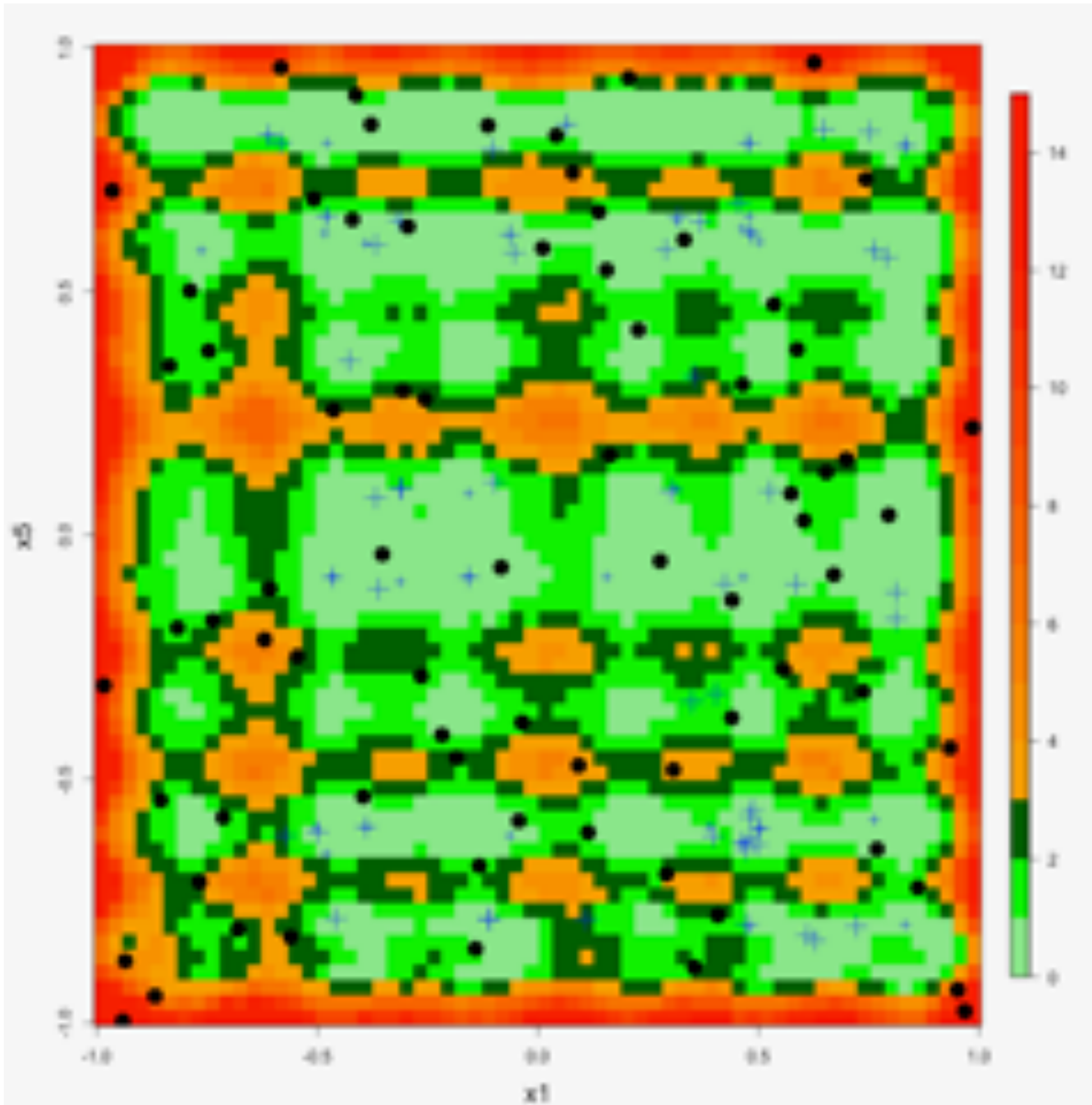


Wave 2: 6% of the parameter space remains



Wave 3: 5% of the parameter space remains





x5
x1
-1.0
-0.5
0.0
0.5
1.0
0
2
4
6
8
10
12
14

Research Topics

- Geometry of NROY
- Stochastic models
- Deep learning emulators
- Hierarchical models - Exascale computing
- Dynamical Emulators
- Interaction between physical and computer exits