Gaussian process emulation: a tool for computationally expensive modelling

Peter Challenor, Louise Kimpton, James Salter
University of Exeter

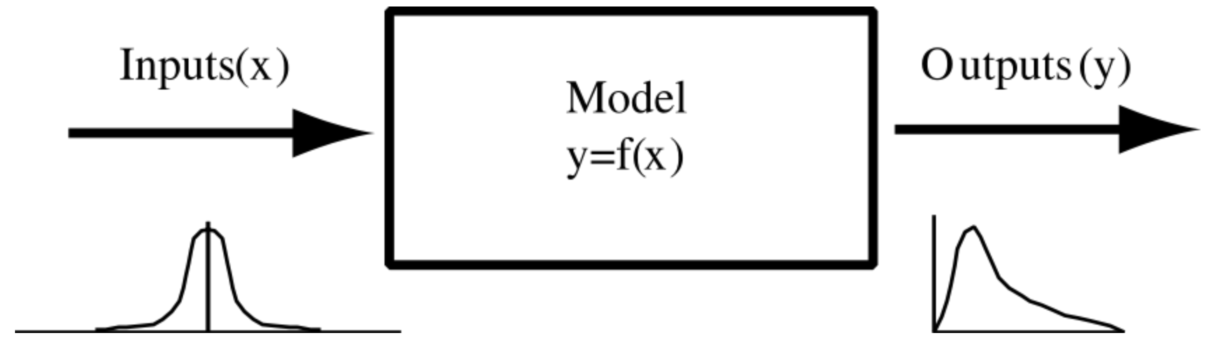
Computational Models

- You have a ‘good’ numerical model
- But it is a little expensive to run
- And we would like to know things that require lots of runs
    - Explore input space
    - Sensitivity Analysis
    - Uncertainty Analysis (Monte Carlo)
    - Inverse modelling (calibration)

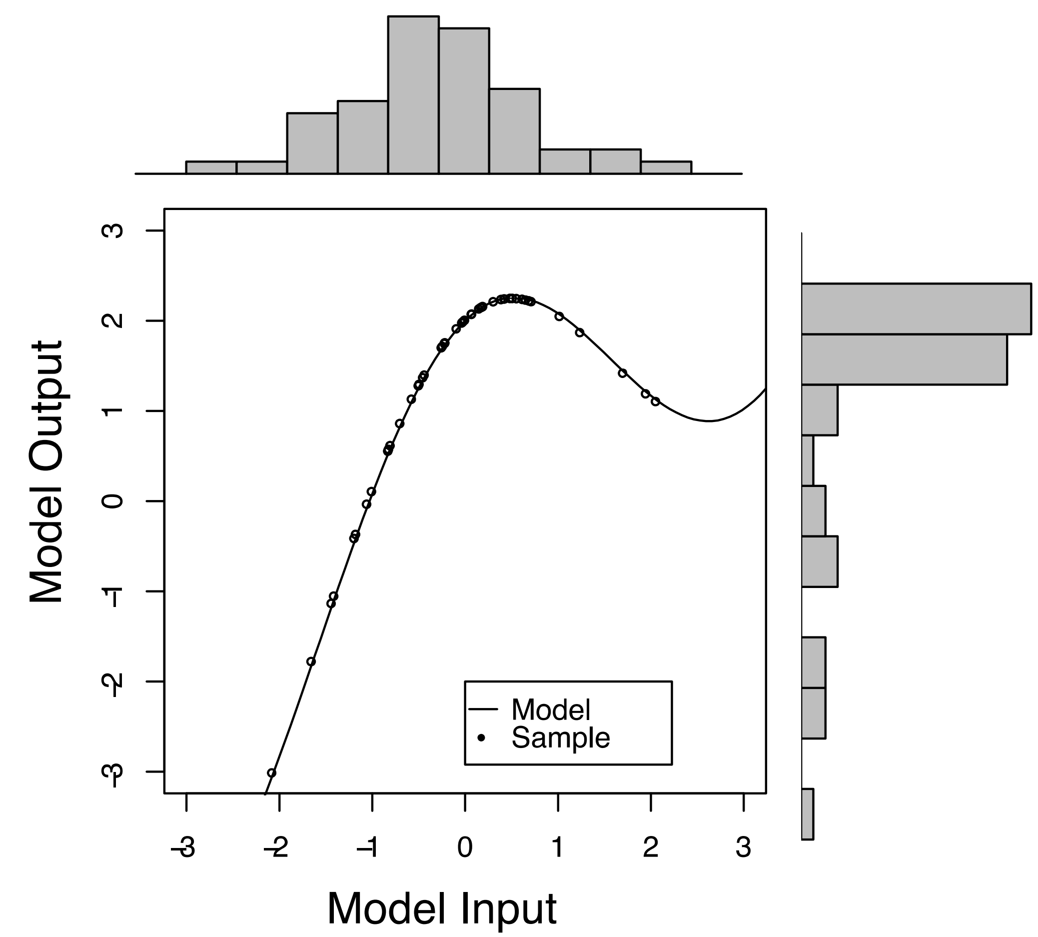
Continuum Models vs ABM

- Continuum models
- ABMs
- Integrals of the model outputs
- Deterministic vs Stochastic Models

Inputs andOutputs



- Model inputs
- Model output
- The outputs work with do not have be model state variables
- Post-processing

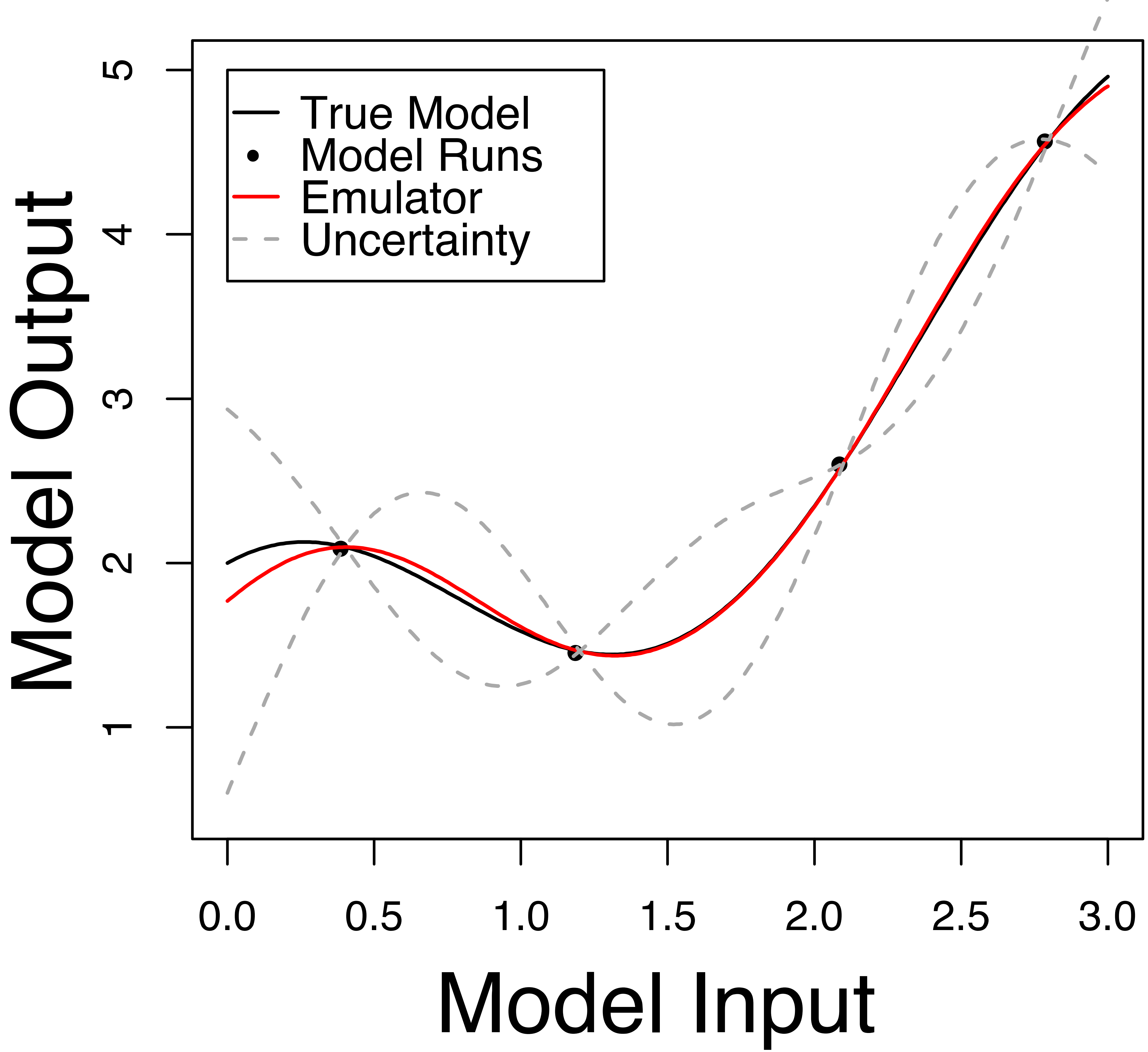


Surrogate Modelling

- One solution to such problems are fast surrogate models
- Possible surrogates include:
    - Polynomials (polynomial chaos)
    - Neural Nets
    - Gaussian processes
    - …
- If the surrogate includes a measure of its own uncertainty we call it an *emulator*

Gaussian Processes

- A Gaussian process is a function defined by mean and covariance functions
- The form of the covariance says how smooth the function is
- And it has a length scale parameter that says how wiggly it is



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

Procdure

1. Set up prior (form of mean and covariance fn's)
2. Design training experiment
3. Run training experiment
4. Estimate model parameters
5. Validate model (LOO or separate experiment)

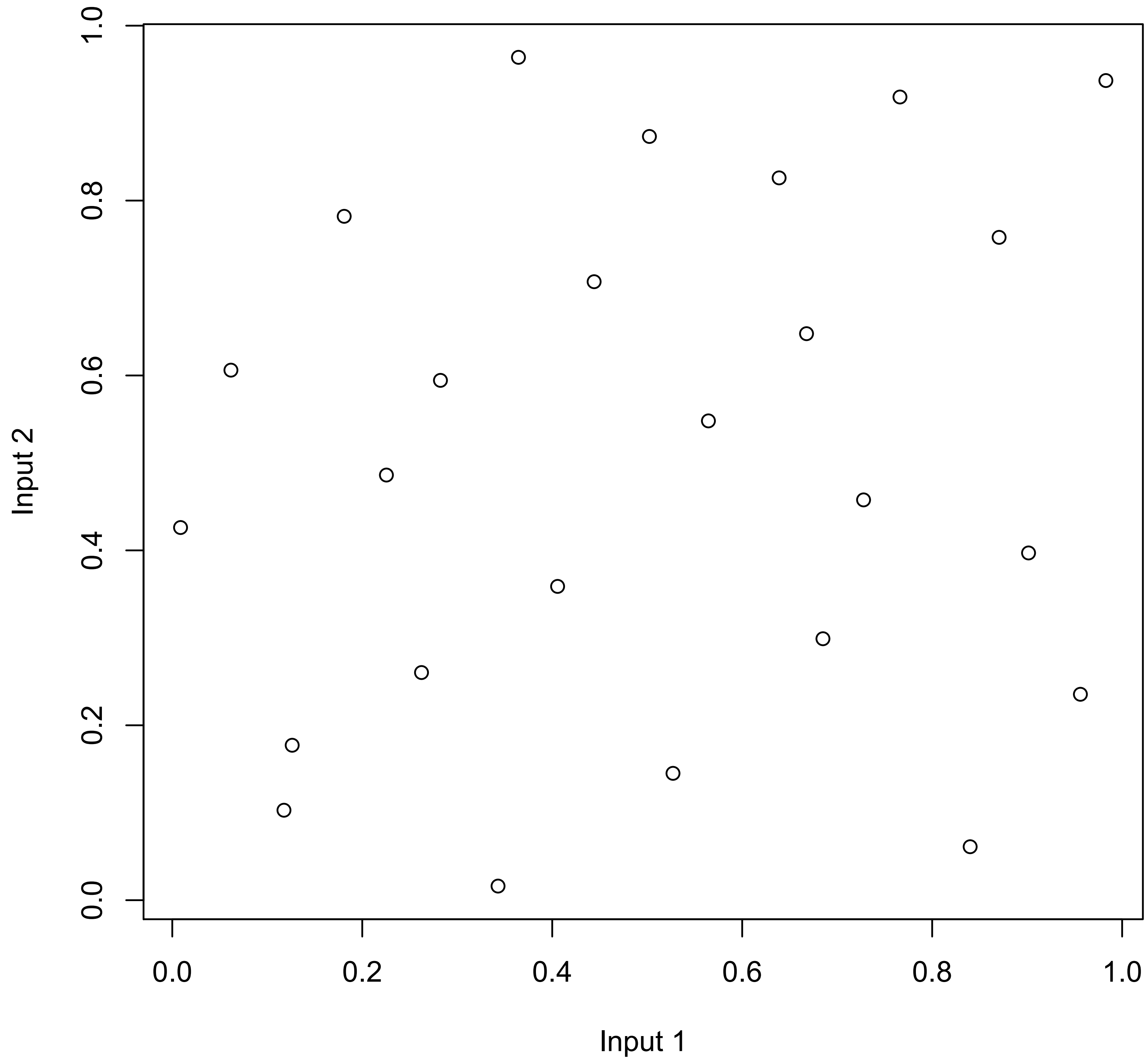
Gaussian process emulators

- Emulators are stochastic functions.
- All realisations interpolate the training data (unless we tell them not to. Add a *nugget*).
- Any set of outputs have Normal distributions with known mean and variance/covariance.
- So we know how well we are doing and can include this additional uncertainty in any calculations.

Design

- Because model runs are expensive
    - Fill space
    - Minimum number of runs
- Latin Hypercubes
- Deterministic models no repetition
- Stochastic models repetition helps estimate the variances

A maximin LHC



What can we use it for

- We now have a fast, approximate version of the model.
- It doesn’t replace the model but can be used, for example, for
    1. Exploration
    2. Sensitivity Analysis
    3. Uncertainty Analysis
    4. Inverse modelling
    5. …

Exploration

- Often want to explore input space
  - May want to find a particular contour or region
  - ‘Needle In Haystack’
  - Use emulator to explore
  - Zoom in, with additional full model runs to reduce uncertainty

Sensitivity Analysis

- How sensitive are your model outputs to the inputs?
- Variance based sensitivity analysis
- Always do two at a time sensitivity to account for non-linearity

Uncertainty Analysis

- The emulator is fast, so Monte Carlo or (Markov Chain Monte Carlo; MCMC) methods are possible.

1. Setup joint distribution for the model inputs
2. Sample from this distribution
3. Propagate through the emulator (adding the emulation error)
4. Gives a sample from the output distribution

Calibration
Inverse Modelling

- Measurements on outputs
- What do they tell us about the inputs?
- Run the model backwards

Classical Inverse Modelling

- Least squares
- Bayesian calibration
- Both assume that there is a ‘best’ solution

Model Discrepancy

- Our models are not perfect
- The data may not lie in the manifold of model solutions
- Classical calibration will get to the closest point
- And then will appear to get more and more certain (but in the wrong place!)
- We need to take discrepancy into account

Kennedy and O’Hagan

- Kennedy and O’Hagan (2001) used two GPs.
- One to emulate the model and to model the discrepancy
- This is a very nice idea
  - But suffers from identifiability issues
  - Soluble with strong priors or additional constraints Brynjarsdottir, J. and O'Hagan, A. (2014)

History Matching

- Instead of trying to find the ‘best’ set of inputs find all the inputs that are implausible given the measurements
- Discard these
- The ‘best’ solution must lie in what is left.

- Take the squared distance between the expectation of the emulator to the data
- Scale it by the sum of three variances
  - The measurement error
  - The emulator variance
  - A discrepancy variance
  - (For stochastic models there is a fourth term)
- And square root it
- If that implausibility measure is greater than 3 the set of inputs is deemed implausible

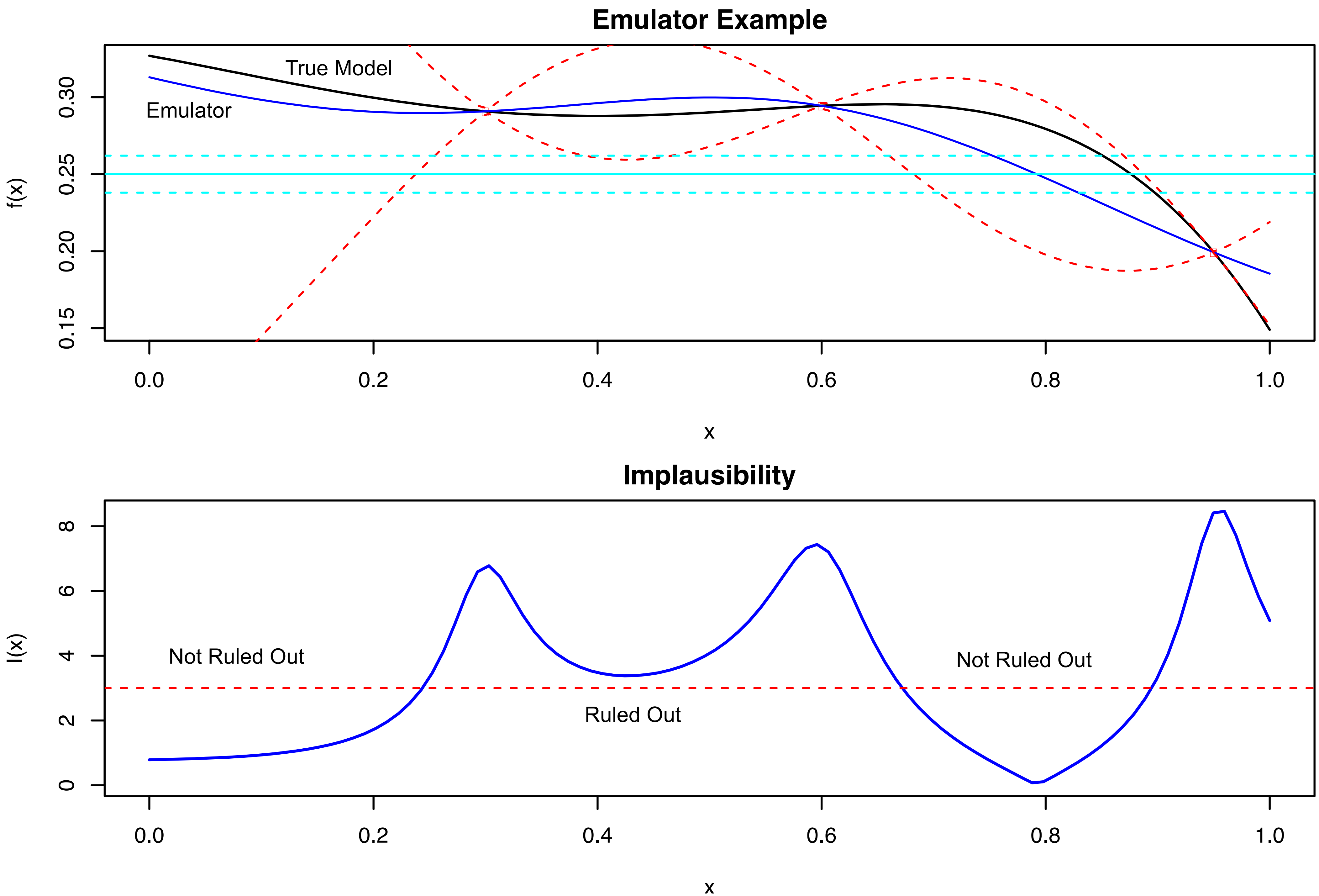
$$Imp = \sqrt{\frac{y - E(f(x))^2}{V_y + V_{emul} + V_{disc}}}$$

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

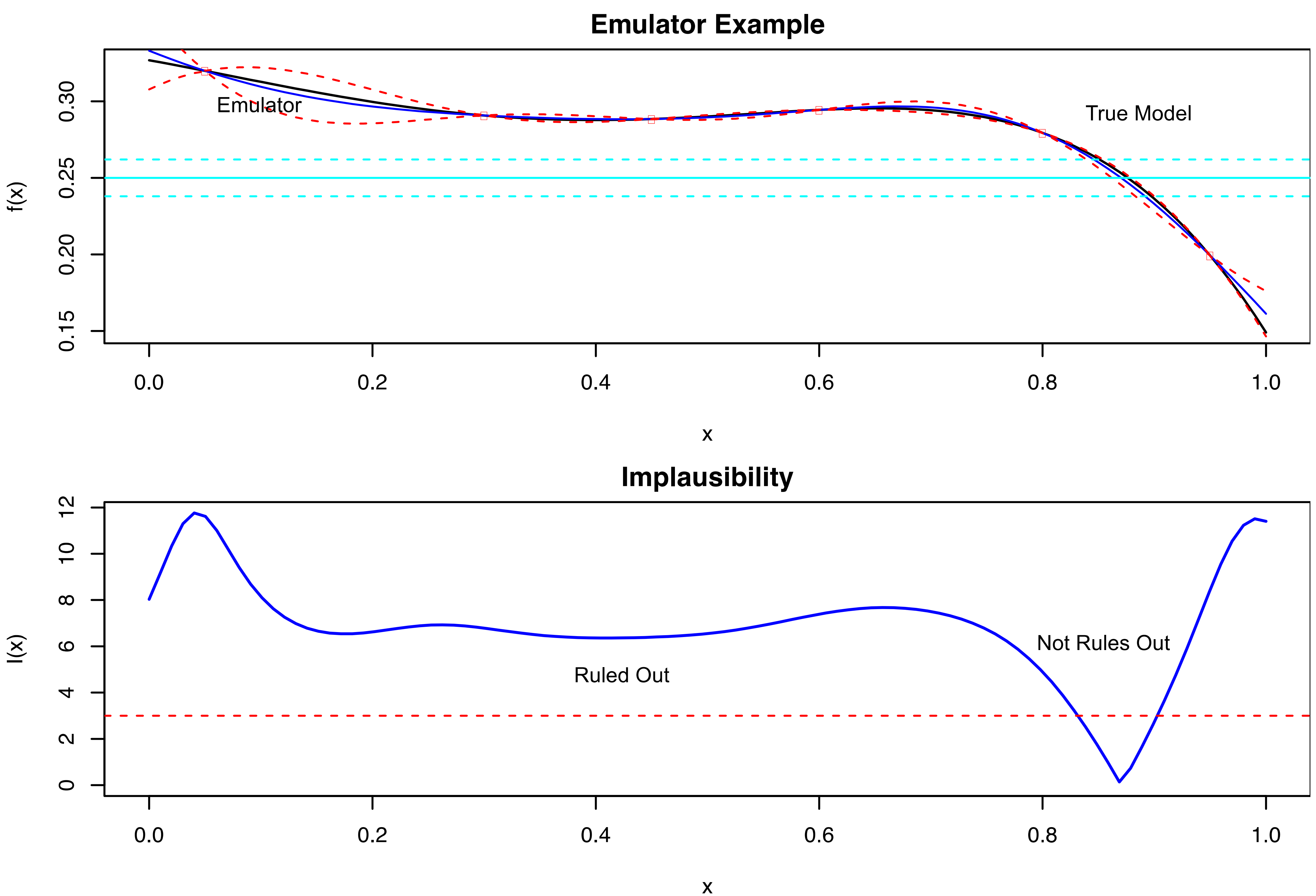
Procedure

- 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

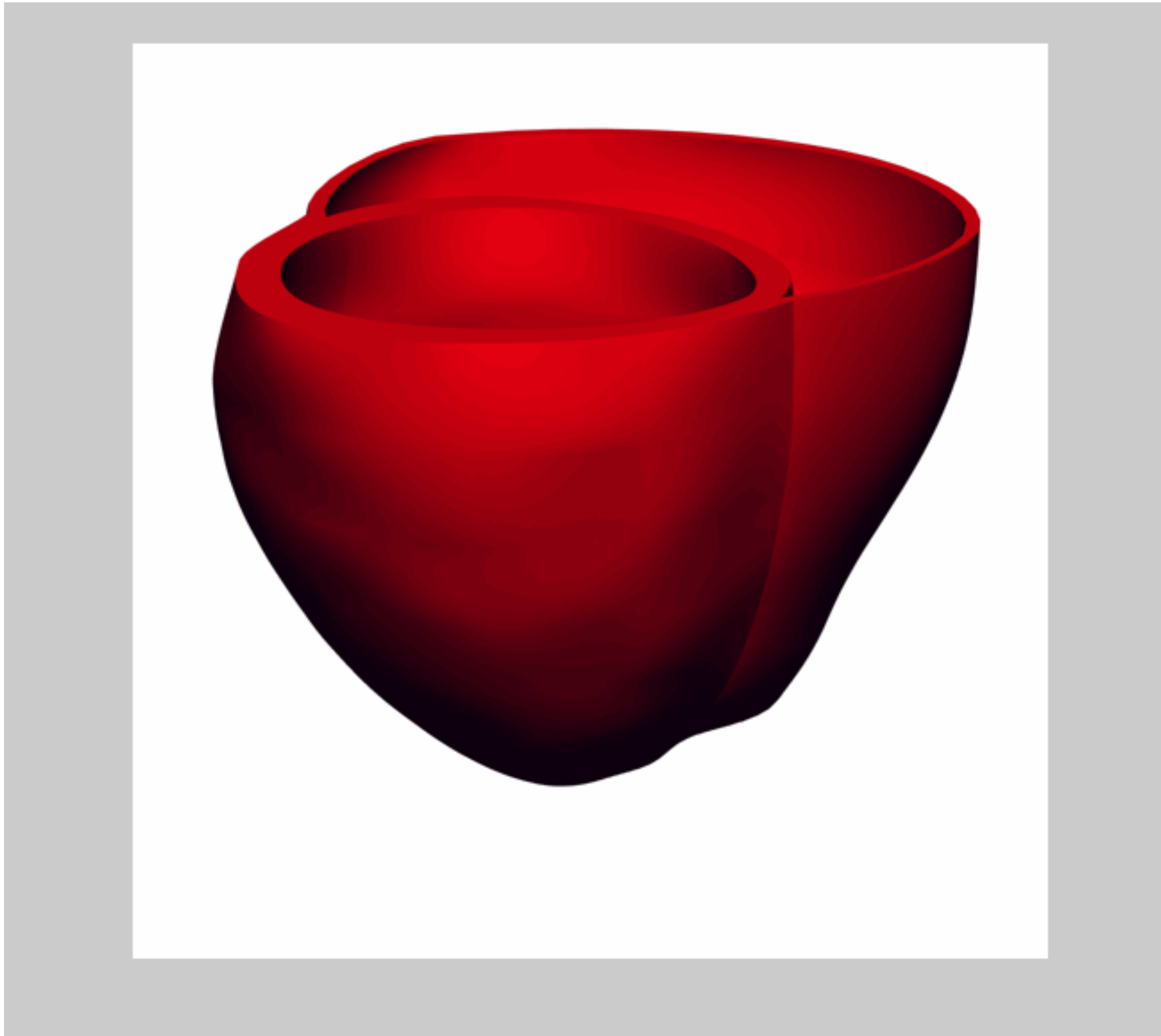


Emulator Example
True Model
Emulator
f(x)
0.15
0.20
0.25
0.30
0.0
0.2
0.4
0.6
0.8
1.0
x
Implausibility
Not Ruled Out
Ruled Out
Not Ruled Out
I(x)
0
2
4
6
8
0.0
0.2
0.4
0.6
0.8
1.0
x



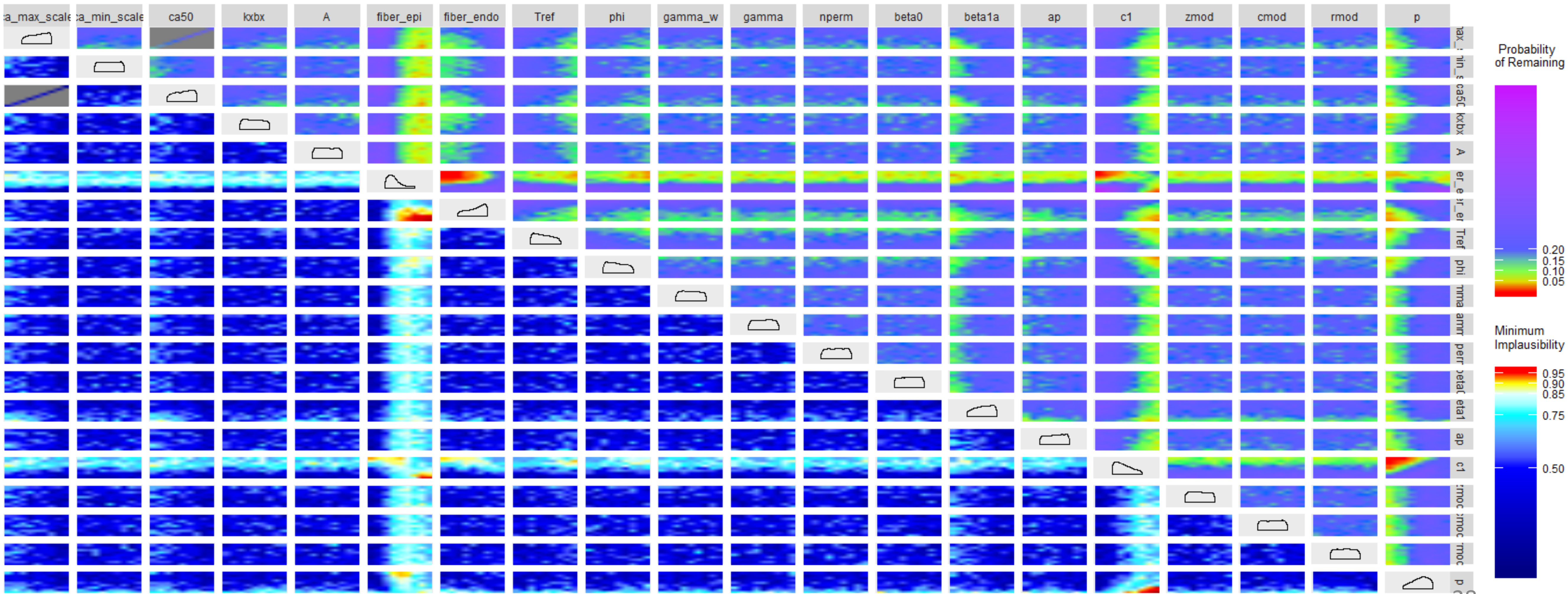
Emulator Example
Emulator
True Model
f(x)
0.15
0.20
0.25
0.30
0.0
0.2
0.4
0.6
0.8
1.0
x
Implausibility
I(x)
0
2
4
6
8
10
12
Ruled Out
Not Rules Out
0.0
0.2
0.4
0.6
0.8
1.0
x

A Cardiac Model

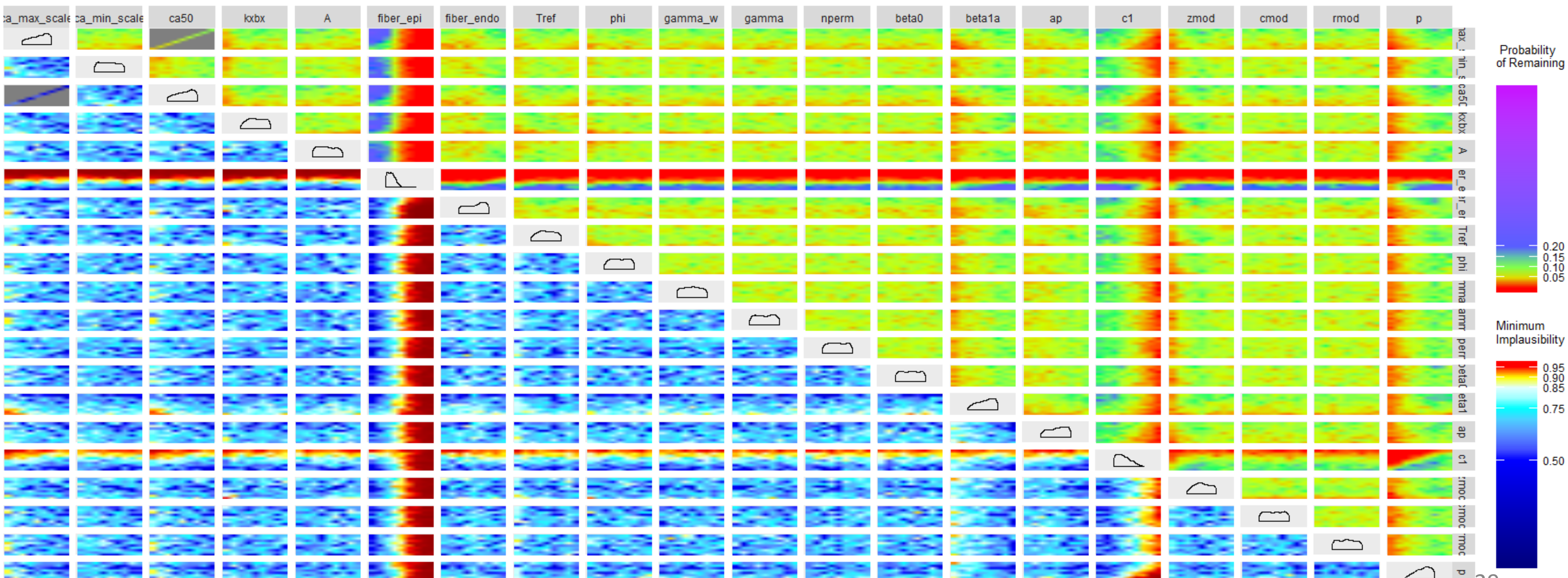


Thanks to Steve Neiderer, KCL/St Thomas

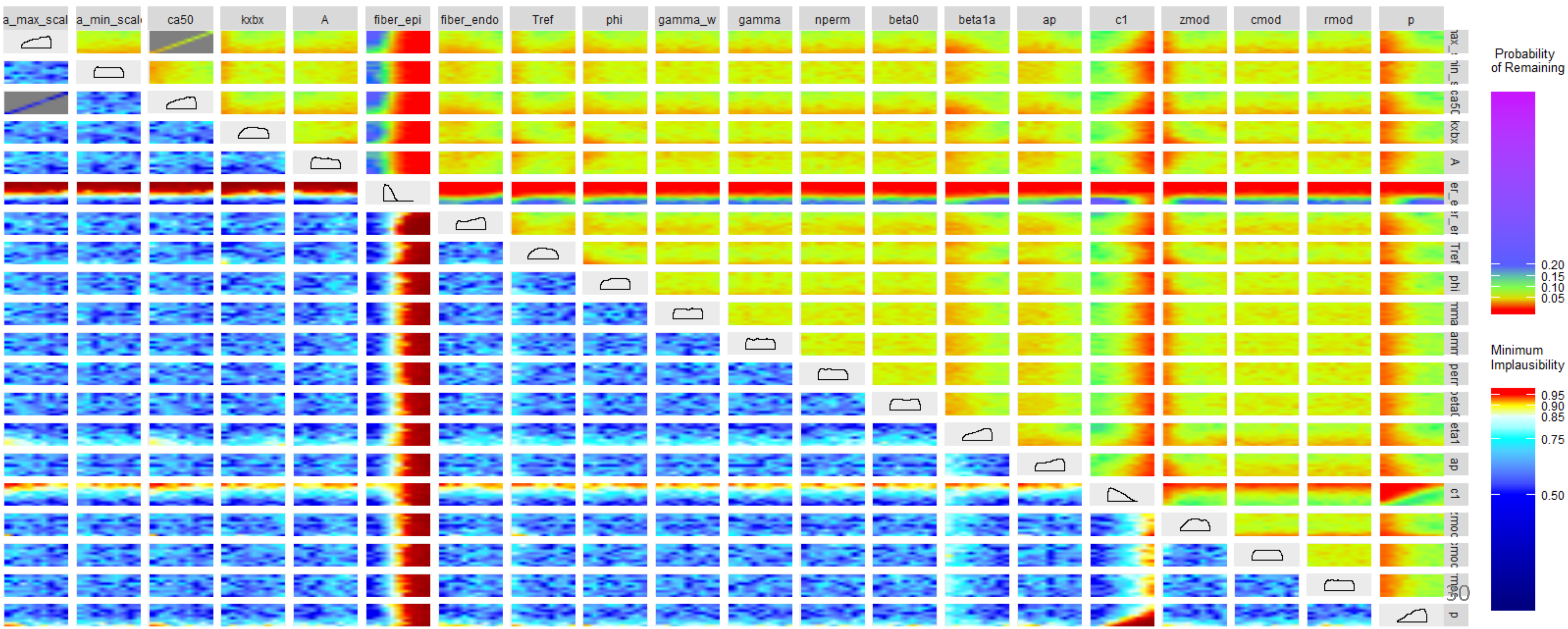
Wave 1: 25% of the parameter space remains



Wave 2: 6% of the parameter space remains



Wave 3: 5% of the parameter space remains



Advanced Topics

- Stochastic models
- Sequential Design
- Hierarchical models/emulators
- Exploiting the unique nature of ABMs