Uncertainty quantification for complex physical models

Michael Goldstein
Durham University

Complex physical models

Most large and complex physical systems are studied by mathematical models, implemented as high dimensional computer simulators. Some examples are:

Complex physical models

Most large and complex physical systems are studied by mathematical models, implemented as high dimensional computer simulators. Some examples are:

Oil reservoirs An oil reservoir simulator is used to manage assets associated with the reservoir, in order to develop efficient production schedules, etc.

Natural Hazards Floods, volcanoes, and so forth, are all studied by large computer simulators.

Climate change Large scale climate simulators are constructed to assess likely effects of human intervention upon future climate behaviour.

Galaxy formation The study of the development of the Universe is carried out by using a Galaxy formation simulator.

Disease modelling Agent based models are used to study interventions to control infectious diseases.

Energy planning Simulators of future energy demand and provision are key components of planning for energy investment.

Complex physical models

Most large and complex physical systems are studied by mathematical models, implemented as high dimensional computer simulators. Some examples are:

Oil reservoirs An oil reservoir simulator is used to manage assets associated with the reservoir, in order to develop efficient production schedules, etc.

Natural Hazards Floods, volcanoes, and so forth, are all studied by large computer simulators.

Climate change Large scale climate simulators are constructed to assess likely effects of human intervention upon future climate behaviour.

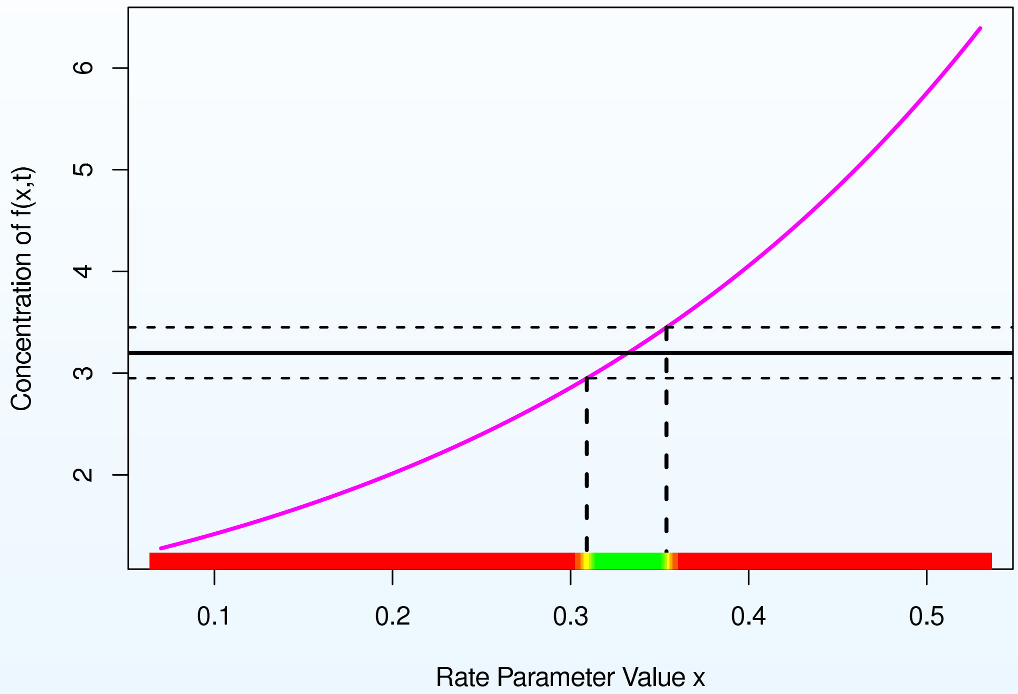
Galaxy formation The study of the development of the Universe is carried out by using a Galaxy formation simulator.

Disease modelling Agent based models are used to study interventions to control infectious diseases.

Energy planning Simulators of future energy demand and provision are key components of planning for energy investment.

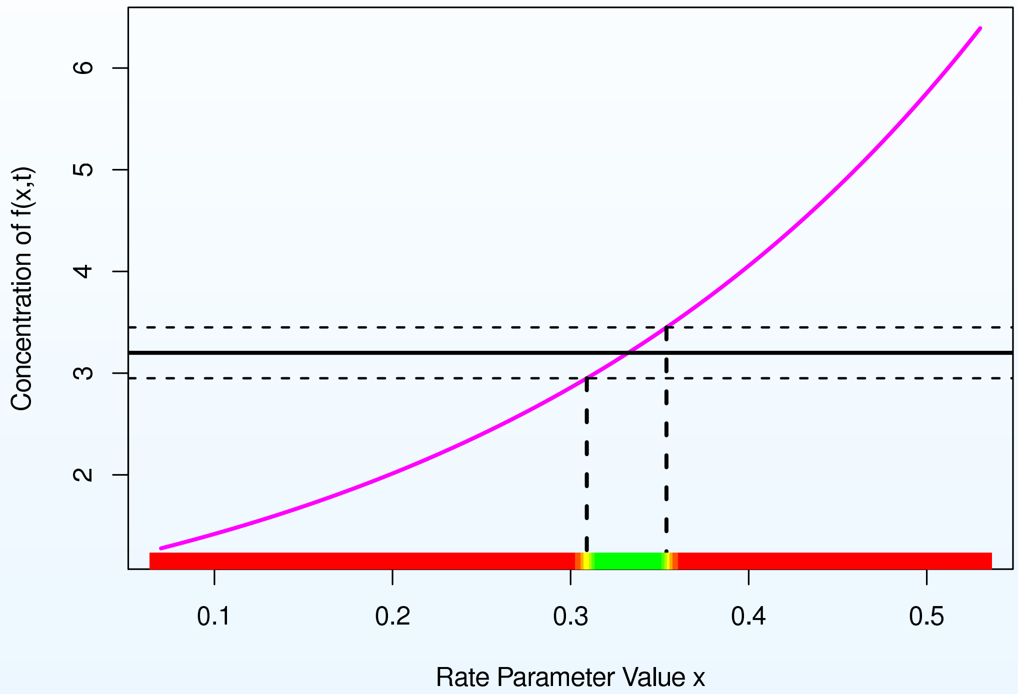
The science in each of these applications is completely different. However, the underlying methodology for handling uncertainty is the same.

Simple example



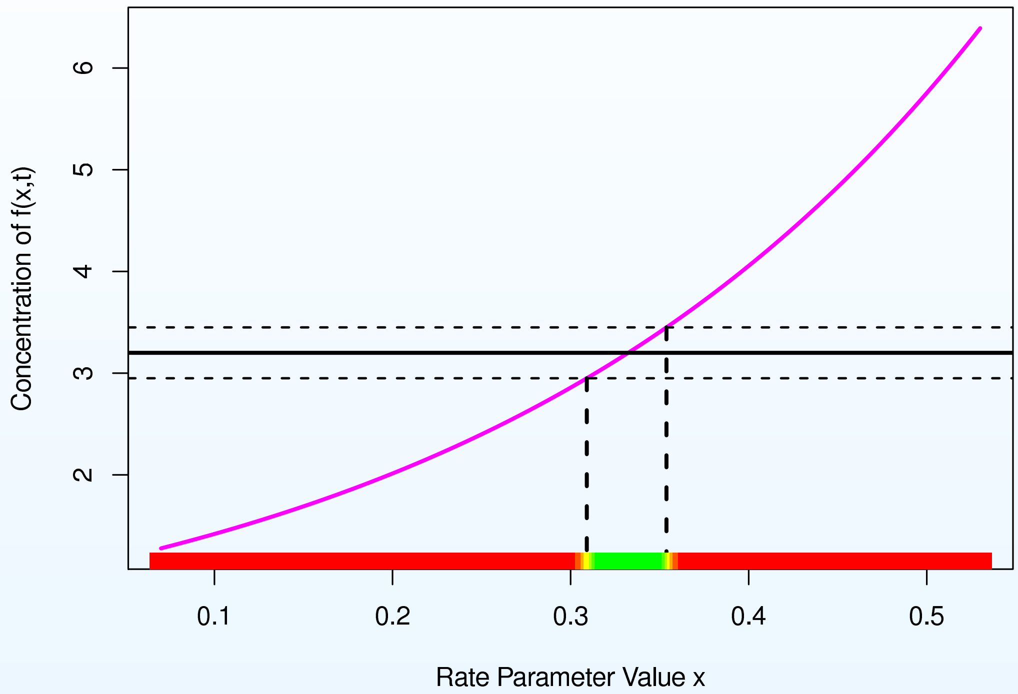
One input, x, one output $f(x)$.

Simple example



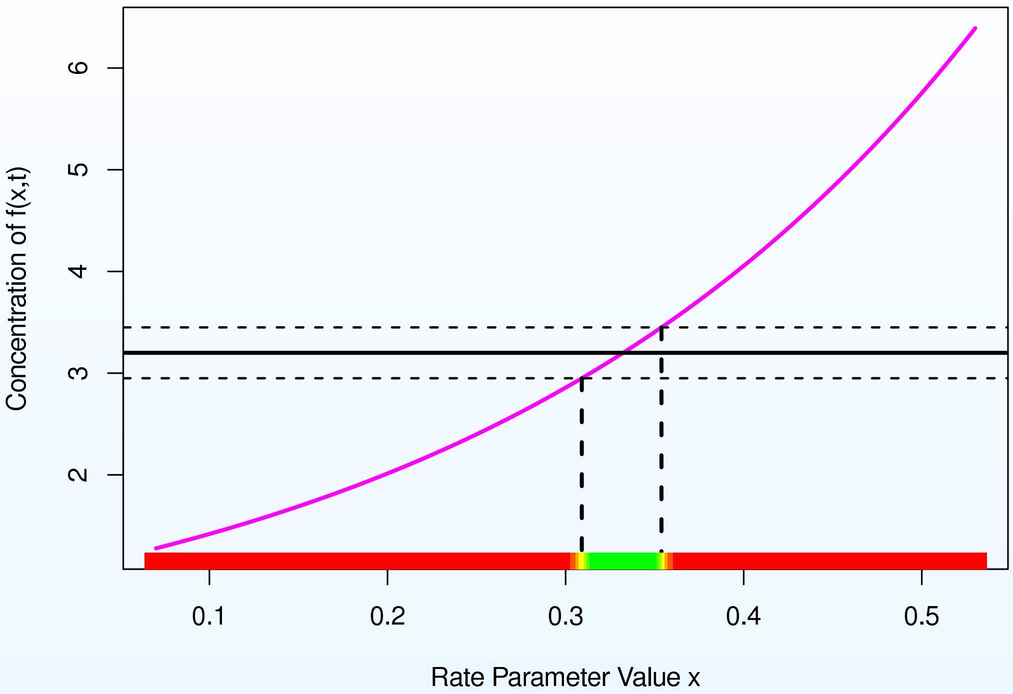
One input, x, one output $f(x)$. Function shown as pink line.

Simple example



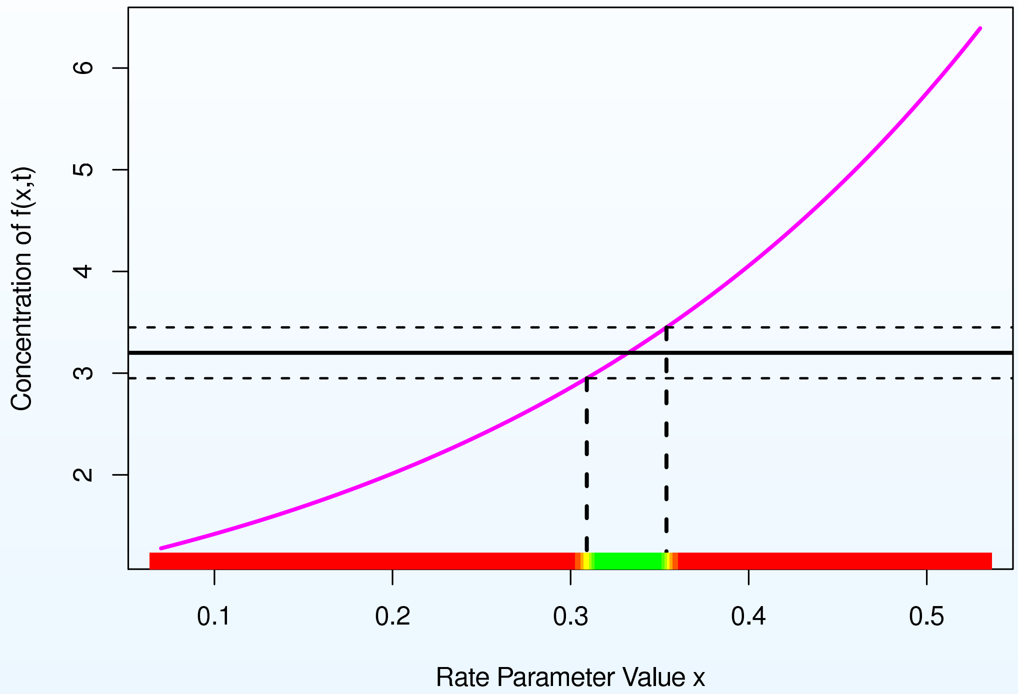
One input, x, one output $f(x)$. Function shown as pink line.
Observation of value of $f(x)$ (black line)

Simple example



One input, x, one output $f(x)$. Function shown as pink line.
Observation of value of $f(x)$ (black line) made with error (dotted lines).

Simple example

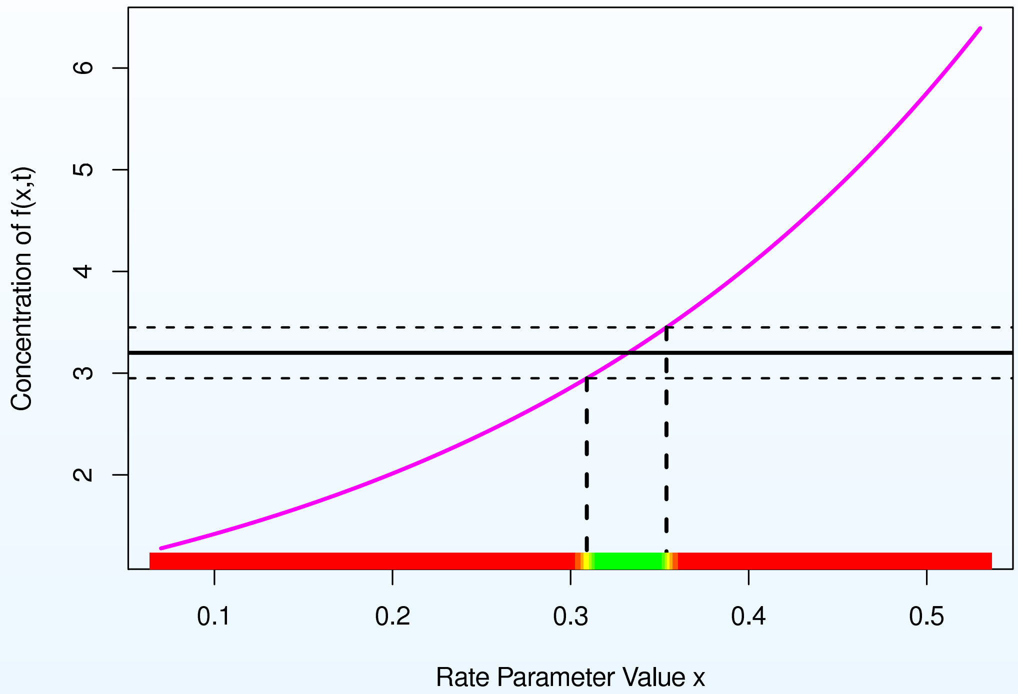


One input, x, one output $f(x)$. Function shown as pink line.

Observation of value of $f(x)$ (black line) made with error (dotted lines).

Uncertainty in the measurement of $f(x)$ leads to uncertainty in the inferred values of x.

Simple example



One input, x, one output $f(x)$. Function shown as pink line.

Observation of value of $f(x)$ (black line) made with error (dotted lines).

Uncertainty in the measurement of $f(x)$ leads to uncertainty in the inferred values of x.

Hence we see a range (green/yellow) of possible values of x consistent with the measurements, with all the implausible values of x in red.

Three problems with this approach

(i) We can't draw the curve on the graph

(because our model is expensive to evaluate for any choice of inputs).

Three problems with this approach

(i) We can’t draw the curve on the graph

(because our model is expensive to evaluate for any choice of inputs).

(ii) Even if we could draw the curve, we wouldn’t be able to do the inversion

(because the model has many inputs and many outputs, linked by complex curves)

Three problems with this approach

(i) We can't draw the curve on the graph

(because our model is expensive to evaluate for any choice of inputs).

(ii) Even if we could draw the curve, we wouldn't be able to do the inversion

(because the model has many inputs and many outputs, linked by complex curves)

(iii) Even if we could do the inversion, this would only tell us about future performance of the model, not the real world

(because the model is not the same as the world)

Three problems with this approach

(i) We can't draw the curve on the graph

(because our model is expensive to evaluate for any choice of inputs).

(ii) Even if we could draw the curve, we wouldn't be able to do the inversion

(because the model has many inputs and many outputs, linked by complex curves)

(iii) Even if we could do the inversion, this would only tell us about future performance of the model, not the real world

(because the model is not the same as the world)

Different physical models vary in many aspects, but the approaches for addressing these problems are very similar

(which is why there is a common underlying methodology).

General Sources of Uncertainty

(i) parametric uncertainty (each model requires a, typically high dimensional, parametric specification)

(ii) measurement uncertainty (as the model is calibrated against system data all of which is measured with error),

(iii) condition uncertainty (uncertainty as to boundary conditions, initial conditions, and forcing functions),

(iv) functional uncertainty (model evaluations take a long time, so the function is unknown almost everywhere),

(v) solution uncertainty (as the system equations can only be solved to some necessary level of approximation),

(vi) stochastic uncertainty (either the model is stochastic, or it should be),

(vii) structural uncertainty (model only approximates the physical system).

(viii) multi-model uncertainty(typically, several models for the physical system)

(ix) Decision uncertainty (mismatch between model and real world decisions)

The Bayesian approach

In this approach, all probabilities are the subjective judgements of individuals so, not **the** probability that we will achieve the project target, but Anne's probability or Bob's probability (which may differ) for this outcome.

The Bayesian approach

In this approach, all probabilities are the subjective judgements of individuals so, not **the** probability that we will achieve the project target, but Anne's probability or Bob's probability (which may differ) for this outcome.

The Bayesian approach unifies and synthesises all of the different sources of uncertainty into an overall judgement of uncertainty. It has many excellent tools to help Anne and Bob to create careful, well founded and clearly documented uncertainty judgements, and, if they do differ, to explore the underlying reasons for such disagreemnt and to suggest possible resolutions.

The Bayesian approach

In this approach, all probabilities are the subjective judgements of individuals so, not **the** probability that we will achieve the project target, but Anne's probability or Bob's probability (which may differ) for this outcome.

The Bayesian approach unifies and synthesises all of the different sources of uncertainty into an overall judgement of uncertainty. It has many excellent tools to help Anne and Bob to create careful, well founded and clearly documented uncertainty judgements, and, if they do differ, to explore the underlying reasons for such disagreemnt and to suggest possible resolutions.

The Bayesian approach for studying uncertainty in computer models is well established and successful. It feeds naturally into decision analysis for identifying and validating good choices for system optimisation.

The Bayesian approach

In this approach, all probabilities are the subjective judgements of individuals so, not **the** probability that we will achieve the project target, but Anne's probability or Bob's probability (which may differ) for this outcome.

The Bayesian approach unifies and synthesises all of the different sources of uncertainty into an overall judgement of uncertainty. It has many excellent tools to help Anne and Bob to create careful, well founded and clearly documented uncertainty judgements, and, if they do differ, to explore the underlying reasons for such disagreemnt and to suggest possible resolutions.

The Bayesian approach for studying uncertainty in computer models is well established and successful. It feeds naturally into decision analysis for identifying and validating good choices for system optimisation.

There are good pragmatic methods for simplifying and streamlining the uncertainty analysis for large and complex problems.

The Bayesian approach

In this approach, all probabilities are the subjective judgements of individuals so, not **the** probability that we will achieve the project target, but Anne's probability or Bob's probability (which may differ) for this outcome.

The Bayesian approach unifies and synthesises all of the different sources of uncertainty into an overall judgement of uncertainty. It has many excellent tools to help Anne and Bob to create careful, well founded and clearly documented uncertainty judgements, and, if they do differ, to explore the underlying reasons for such disagreemnt and to suggest possible resolutions.

The Bayesian approach for studying uncertainty in computer models is well established and successful. It feeds naturally into decision analysis for identifying and validating good choices for system optimisation.

There are good pragmatic methods for simplifying and streamlining the uncertainty analysis for large and complex problems.

For my preferred flavour of Bayes see **M. Goldstein, D.A. Wooff** (2007) *"Bayes Linear Statistics: Theory and Methods"* Wiley

Function emulation

Uncertainty analysis, for high dimensional problems, is particularly challenging if $f(x)$ is expensive, in time and computational resources, to evaluate for any choice of x.

Function emulation

Uncertainty analysis, for high dimensional problems, is particularly challenging if $f(x)$ is expensive, in time and computational resources, to evaluate for any choice of x.

In such cases, f must be treated as uncertain for all input choices except the small subset for which an actual evaluation has been made.

Function emulation

Uncertainty analysis, for high dimensional problems, is particularly challenging if $f(x)$ is expensive, in time and computational resources, to evaluate for any choice of x.

In such cases, f must be treated as uncertain for all input choices except the small subset for which an actual evaluation has been made.

Therefore, we must construct a description of the uncertainty about the value of $f(x)$ for each x. Such a representation is often termed an emulator of the simulator.

Function emulation

Uncertainty analysis, for high dimensional problems, is particularly challenging if $f(x)$ is expensive, in time and computational resources, to evaluate for any choice of x.

In such cases, f must be treated as uncertain for all input choices except the small subset for which an actual evaluation has been made.

Therefore, we must construct a description of the uncertainty about the value of $f(x)$ for each x. Such a representation is often termed an emulator of the simulator.

The emulator both contains
(i) an approximation to the simulator and

Function emulation

Uncertainty analysis, for high dimensional problems, is particularly challenging if $f(x)$ is expensive, in time and computational resources, to evaluate for any choice of x.

In such cases, f must be treated as uncertain for all input choices except the small subset for which an actual evaluation has been made.

Therefore, we must construct a description of the uncertainty about the value of $f(x)$ for each x. Such a representation is often termed an emulator of the simulator.

The emulator both contains
(i) an approximation to the simulator and

(ii) an assessment of the likely magnitude of the error of the approximation.

Function emulation

Uncertainty analysis, for high dimensional problems, is particularly challenging if $f(x)$ is expensive, in time and computational resources, to evaluate for any choice of x.

In such cases, f must be treated as uncertain for all input choices except the small subset for which an actual evaluation has been made.

Therefore, we must construct a description of the uncertainty about the value of $f(x)$ for each x. Such a representation is often termed an emulator of the simulator.

The emulator both contains
(i) an approximation to the simulator and

(ii) an assessment of the likely magnitude of the error of the approximation.

Unlike the original simulator, the emulator is fast to evaluate for any choice of inputs. This allows us to explore model behaviour for all physically meaningful input specifications.

Form of the emulator

We may represent beliefs about component f_i of f, using an emulator:

$$f_i(x) = \sum_j \beta_{ij} g_{ij}(x) + u_i(x)$$

Form of the emulator

We may represent beliefs about component f_i of f, using an emulator:

$$f_i(x) = \sum_j \beta_{ij} g_{ij}(x) + u_i(x)$$

Global Variation

$\{\beta_{ij}\}$ are unknown scalars,

g_{ij} are known deterministic functions of x, (for example, polynomials)

Form of the emulator

We may represent beliefs about component f_i of f, using an emulator:

$$f_i(x) = \sum_j \beta_{ij} g_{ij}(x) + u_i(x)$$

Global Variation

$\{\beta_{ij}\}$ are unknown scalars,

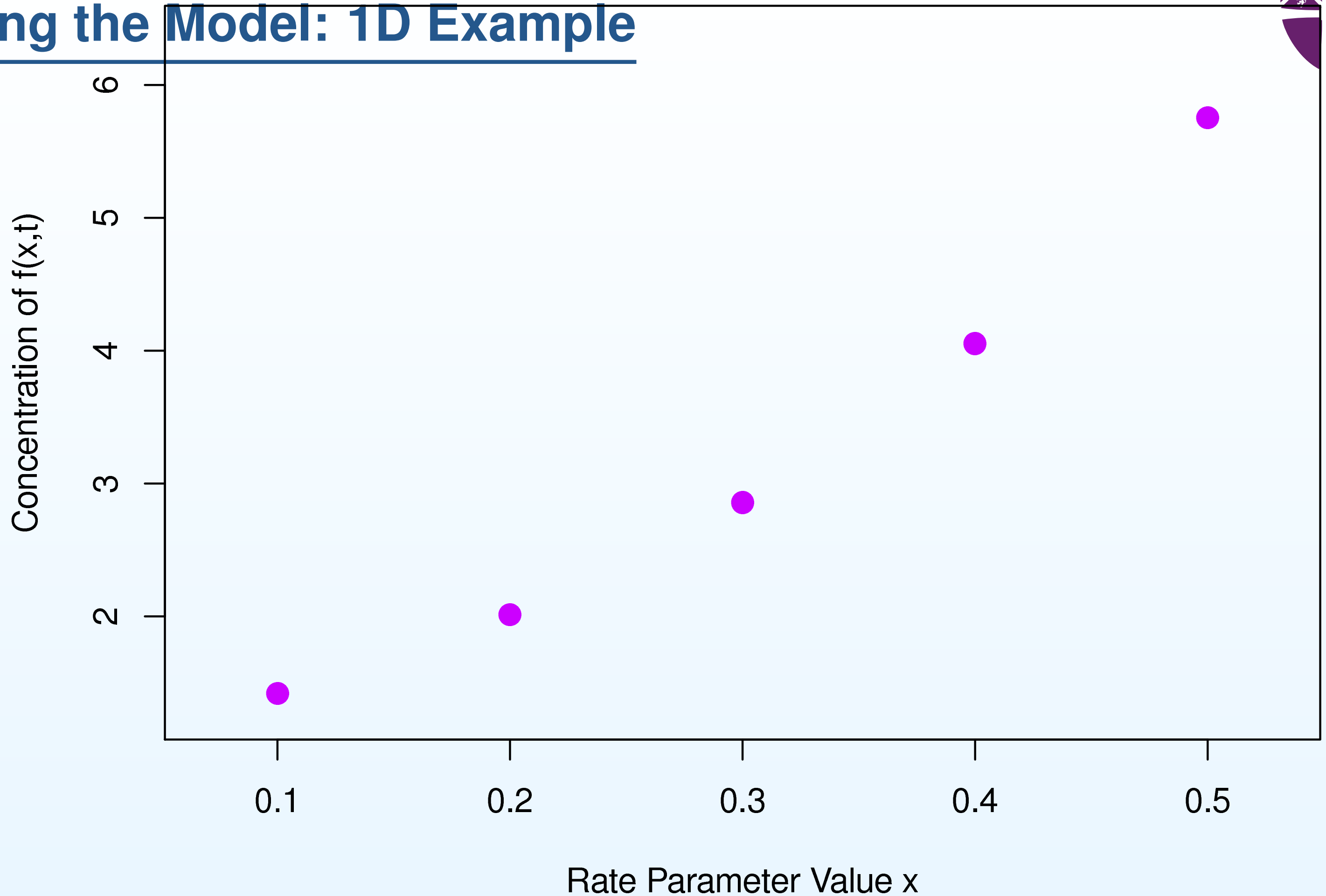
g_{ij} are known deterministic functions of x, (for example, polynomials)

Local Variation

$u_i(x)$ is a second order stationary stochastic process, with (for example) correlation function

$$\mathrm{Corr}(u_i(x), u_i(x')) = \exp(-(\tfrac{\|x-x'\|}{\theta_i})^2)$$

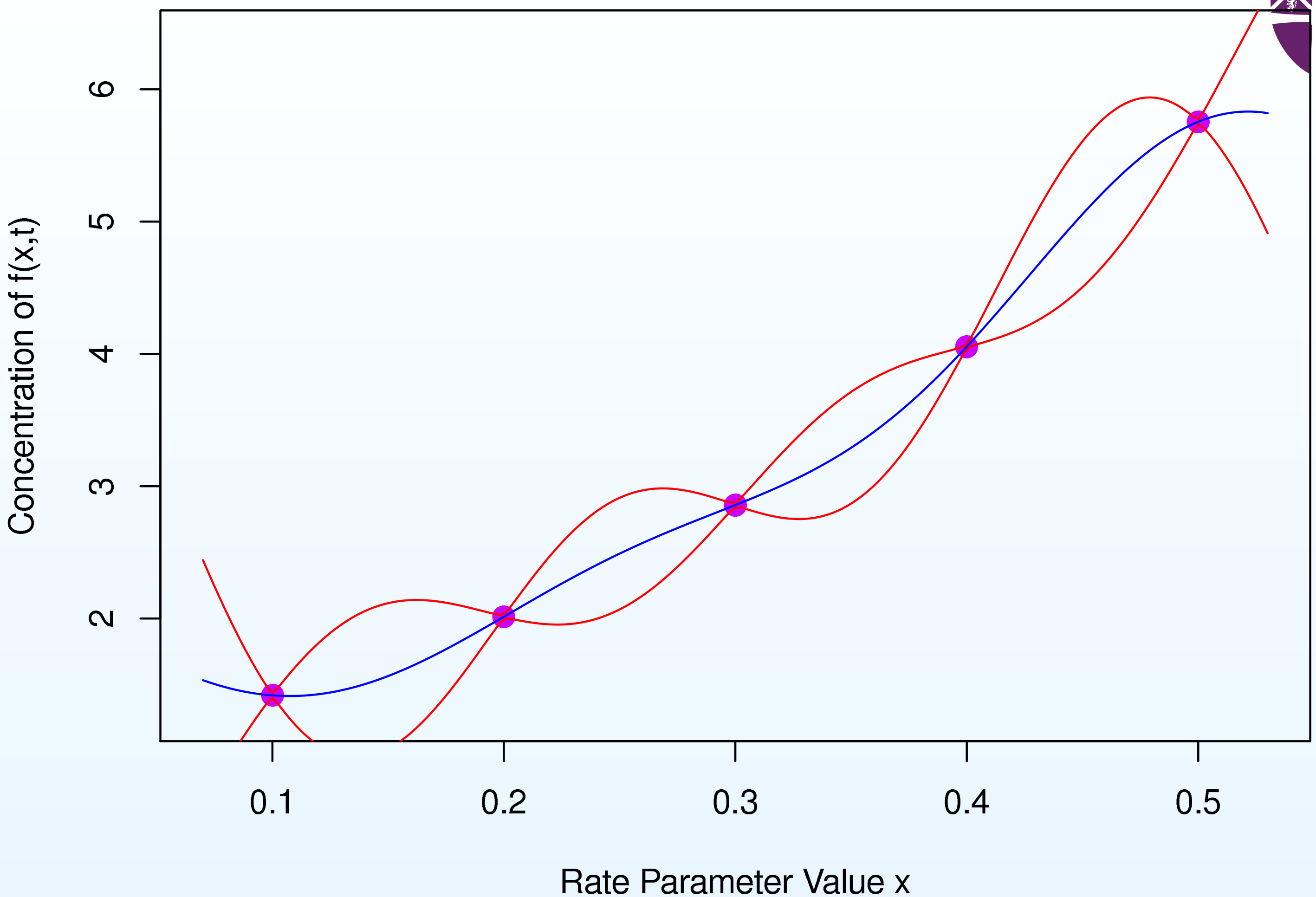
Emulating the Model: 1D Example



Suppose that we do not have the analytic solution of $f(x)$.

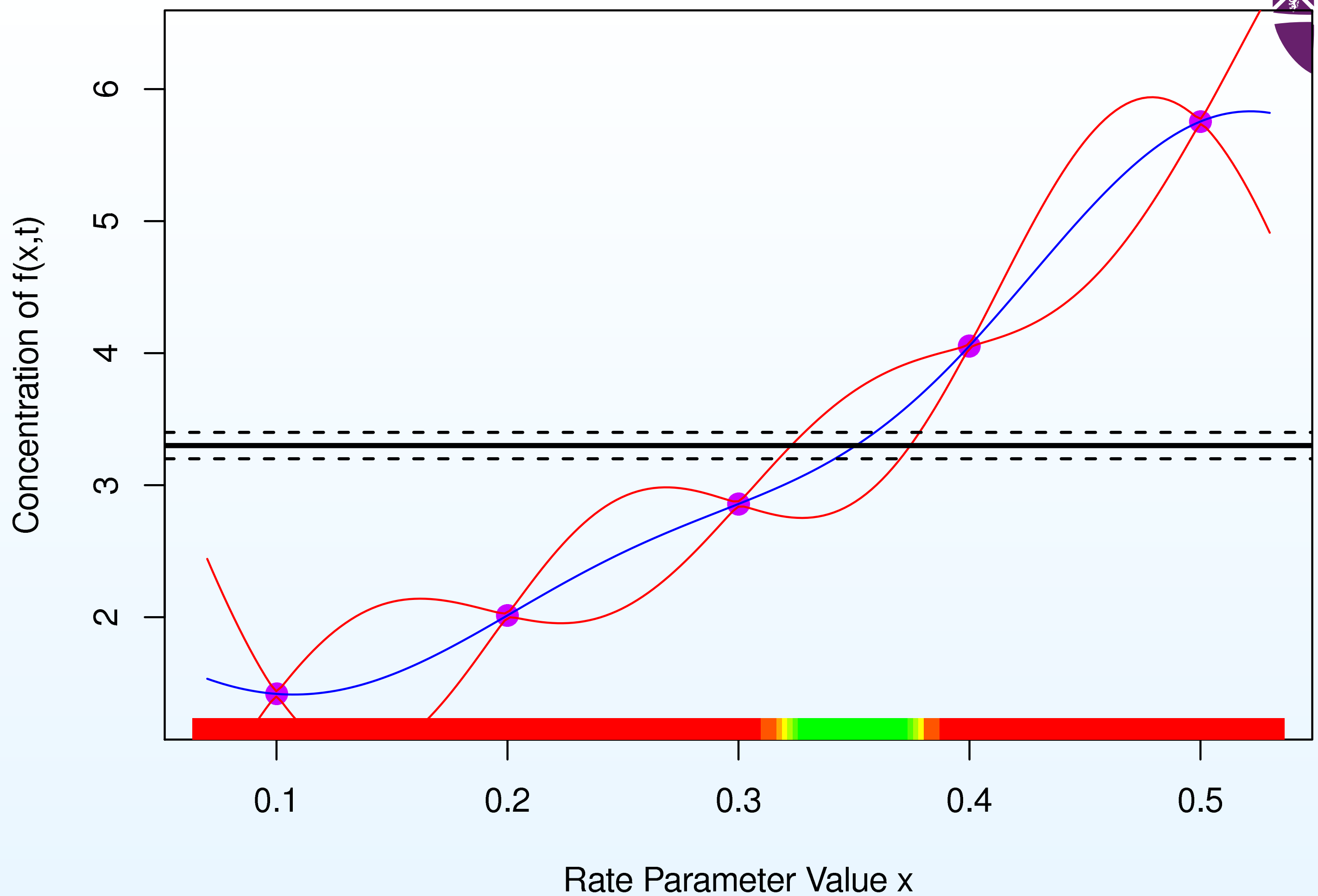
Instead we only have five evaluations of the model.

Therefore we emulate the function.



The emulator can be used to represent our beliefs about the behaviour of the model at untested values of x, and is fast to evaluate.

It gives both the expected value of $f(x)$ (the blue line) along with a credible interval for $f(x)$ (the red lines) representing the uncertainty about the model's behaviour.



Comparing the emulator to the observed measurement we again identify the set of x values currently consistent with this data.

The uncertainty on x now includes uncertainty coming from the emulator.

Emulation methods

We fit the emulators using our favourite statistical tools - generalised least squares, maximum likelihood, Bayes - with a generous helping of expert judgement.

Emulation methods

We fit the emulators using our favourite statistical tools - generalised least squares, maximum likelihood, Bayes - with a generous helping of expert judgement.

We use efficient space filling designs to generate the set of simulator evaluations to carry out in order to fit the emulators.

Emulation methods

We fit the emulators using our favourite statistical tools - generalised least squares, maximum likelihood, Bayes - with a generous helping of expert judgement.

We use efficient space filling designs to generate the set of simulator evaluations to carry out in order to fit the emulators.

Make an assessment of the scientific plausibility of each emulator. Use careful diagnostics to test the validity of our emulators, for example, assessing the reliability of the emulator for predicting the simulator at new evaluations - paying particular attention to extrapolation.

Emulation methods

We fit the emulators using our favourite statistical tools - generalised least squares, maximum likelihood, Bayes - with a generous helping of expert judgement.

We use efficient space filling designs to generate the set of simulator evaluations to carry out in order to fit the emulators.

Make an assessment of the scientific plausibility of each emulator. Use careful diagnostics to test the validity of our emulators, for example, assessing the reliability of the emulator for predicting the simulator at new evaluations - paying particular attention to extrapolation.

If the simulator is very slow to evaluate, then we may create a fast approximation to the simulator to support building the emulator.

Fitting simulators to data

Simulator calibration aims to identify the best choices of input parameters x^*, based on matching data z to the corresponding simulator outputs $f_h(x)$.
However

Fitting simulators to data

Simulator calibration aims to identify the best choices of input parameters x^*, based on matching data z to the corresponding simulator outputs $f_h(x)$. However

(i) we may not believe in a unique true input value for the simulator;

(because the parameters are not real things - they only exist inside the model - and different choices may be good for fitting different outputs)

Fitting simulators to data

Simulator calibration aims to identify the best choices of input parameters x^*, based on matching data z to the corresponding simulator outputs $f_h(x)$. However

(i) we may not believe in a unique true input value for the simulator;

(because the parameters are not real things - they only exist inside the model - and different choices may be good for fitting different outputs)

(ii) we may be unsure whether there are any good choices of input parameters

(because there may be serious problems with our simulator)

Fitting simulators to data

Simulator calibration aims to identify the best choices of input parameters x^*, based on matching data z to the corresponding simulator outputs $f_h(x)$. However

(i) we may not believe in a unique true input value for the simulator;

(because the parameters are not real things - they only exist inside the model - and different choices may be good for fitting different outputs)

(ii) we may be unsure whether there are any good choices of input parameters

(because there may be serious problems with our simulator)

(iii) full probabilistic calibration analysis may be very difficult/non-robust for complex simulators.

(because the likelihood surface is complicated and multi-modal, and the Bayes answer often depends on features of the prior distribution which are hard to specify meaningfully)

History matching

A conceptually simple procedure is “history matching”.

This means finding the collection, $C(z)$, of all input choices x for which the simulator output is consistent with the observed data, z, taking into account all of the uncertainties in the problem.

History matching

A conceptually simple procedure is “history matching”.

This means finding the collection, $C(z)$, of all input choices x for which the simulator output is consistent with the observed data, z, taking into account all of the uncertainties in the problem.

If the data is informative for the parameter space, then $C(z)$ will typically form a tiny percentage of the original parameter space.

[$C(z)$ might be empty - suggesting problems with the simulator or data.]

History matching

A conceptually simple procedure is "history matching".

This means finding the collection, $C(z)$, of all input choices x for which the simulator output is consistent with the observed data, z, taking into account all of the uncertainties in the problem.

If the data is informative for the parameter space, then $C(z)$ will typically form a tiny percentage of the original parameter space.

[$C(z)$ might be empty - suggesting problems with the simulator or data.]

We use an 'implausibility measure' $I(x)$ based on a probabilistic metric such as

$$I(x) = \frac{(z - \mathrm{E}(f_h(x)))^2}{\mathrm{Var}(z - \mathrm{E}(f_h(x)))}$$

(where the variance in the denominator is the sum of variance terms e.g. measurement error, emulator error, discrepancy error and so forth.)

History matching

A conceptually simple procedure is “history matching”.

This means finding the collection, $C(z)$, of all input choices x for which the simulator output is consistent with the observed data, z, taking into account all of the uncertainties in the problem.

If the data is informative for the parameter space, then $C(z)$ will typically form a tiny percentage of the original parameter space.

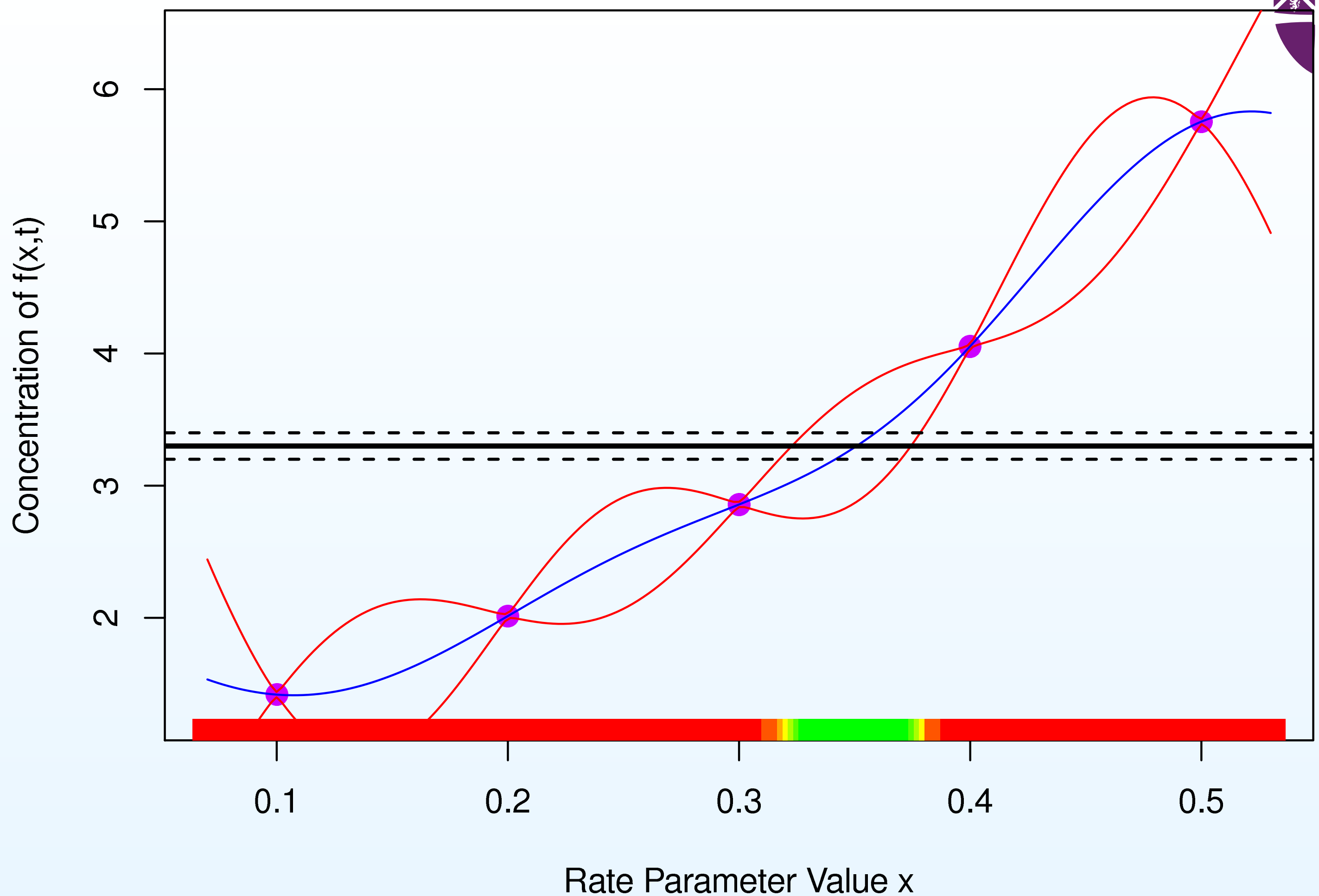
[$C(z)$ might be empty - suggesting problems with the simulator or data.]

We use an ‘implausibility measure’ $I(x)$ based on a probabilistic metric such as

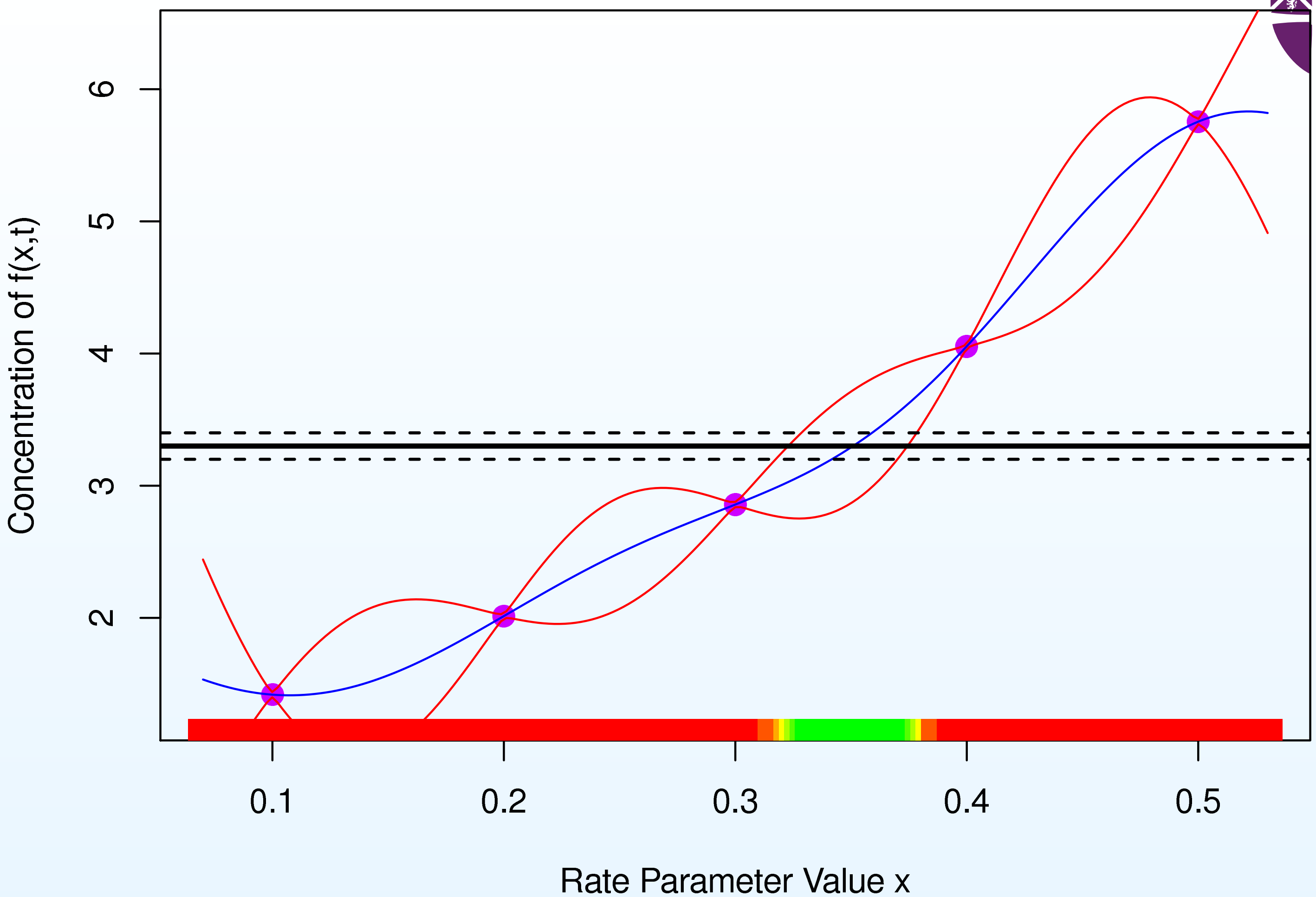
$$I(x) = \frac{(z - \mathrm{E}(f_h(x)))^2}{\mathrm{Var}(z - \mathrm{E}(f_h(x)))}$$

(where the variance in the denominator is the sum of variance terms e.g. measurement error, emulator error, discrepancy error and so forth.)

History matching is an iterative global search procedure.

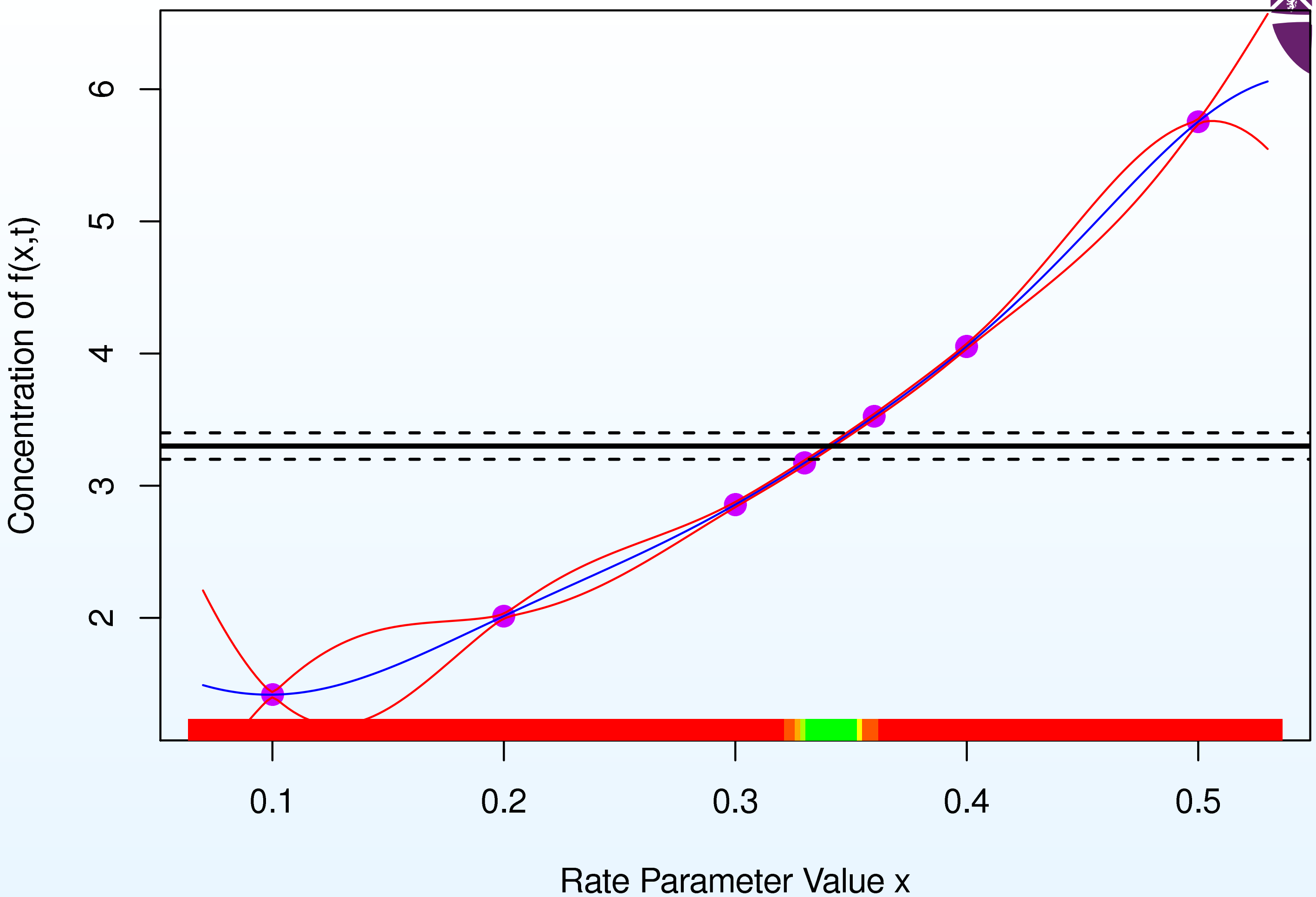


Comparing the emulator to the observed measurement we have identified the set of x values (the green values) which match the observed history, when we take into account all of the uncertainties (here, measurement and emulator error).

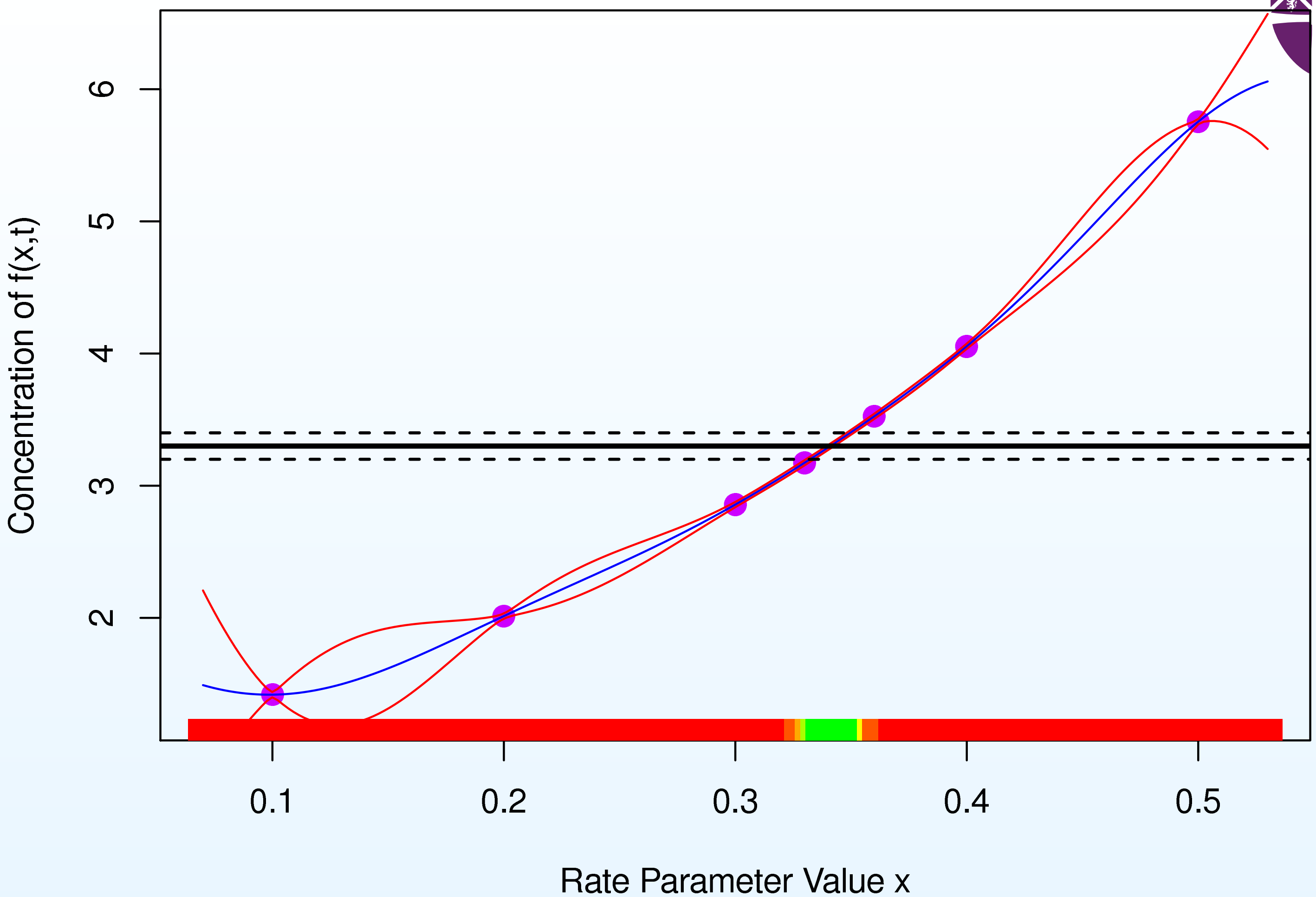


We now remove all of the implausible x values (the red values).

We perform a 2nd iteration or wave of evaluations in the green region to improve emulator accuracy.



Now the emulator is more accurate than the observation, and we can identify the set of all x values of interest.



Now the emulator is more accurate than the observation, and we can identify the set of all x values of interest.

Note that we only need accurate emulation of the simulator in the region close to the output match.

(And the first few waves of history matching typically involve only those outputs which are relatively straightforward to emulate.)

Example: HIV modelling

This case study was based on a research project that explored HIV transmission in Uganda.

Example: HIV modelling

This case study was based on a research project that explored HIV transmission in Uganda.

The simulator used, Mukwano, is a dynamic, stochastic, individual based computer model that simulates the life histories of hypothetical individuals (births, deaths, sexual partnership formation and dissolution and HIV transmission, modelled using time-dependent rates).

Example: HIV modelling

This case study was based on a research project that explored HIV transmission in Uganda.

The simulator used, Mukwano, is a dynamic, stochastic, individual based computer model that simulates the life histories of hypothetical individuals (births, deaths, sexual partnership formation and dissolution and HIV transmission, modelled using time-dependent rates).

Each individual is represented by a number of characteristics, such as gender, date of birth, HIV status, level of sexual activity, concurrency level.

Example: HIV modelling

This case study was based on a research project that explored HIV transmission in Uganda.

The simulator used, Mukwano, is a dynamic, stochastic, individual based computer model that simulates the life histories of hypothetical individuals (births, deaths, sexual partnership formation and dissolution and HIV transmission, modelled using time-dependent rates).

Each individual is represented by a number of characteristics, such as gender, date of birth, HIV status, level of sexual activity, concurrency level.

The behavioural inputs take different values in each of three calendar time periods. This enables sexual behaviour to vary over time.

Example: HIV modelling

This case study was based on a research project that explored HIV transmission in Uganda.

The simulator used, Mukwano, is a dynamic, stochastic, individual based computer model that simulates the life histories of hypothetical individuals (births, deaths, sexual partnership formation and dissolution and HIV transmission, modelled using time-dependent rates).

Each individual is represented by a number of characteristics, such as gender, date of birth, HIV status, level of sexual activity, concurrency level.

The behavioural inputs take different values in each of three calendar time periods. This enables sexual behaviour to vary over time.

Twenty behavioural and two epidemiologic inputs were varied for this study.

Example: empirical data

The empirical 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.

Example: empirical data

The empirical 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.

Every year, demographic information on the cohort is updated, the population is tested for HIV, and a behavioural questionnaire is conducted.

Example: empirical data

The empirical 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.

Every year, demographic information on the cohort is updated, the population is tested for HIV, and a behavioural questionnaire is conducted.

In this study, there are 18 simulator outputs with calibration targets and limits for what constitutes an acceptable match.

Example: empirical data

The empirical 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.

Every year, demographic information on the cohort is updated, the population is tested for HIV, and a behavioural questionnaire is conducted.

In this study, there are 18 simulator outputs with calibration targets and limits for what constitutes an acceptable match.

These include male and female population sizes,

male and female HIV prevalences at three time points.

outputs that check that the behavioural features of the model matched the empirical data.

Example: empirical data

The empirical 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.

Every year, demographic information on the cohort is updated, the population is tested for HIV, and a behavioural questionnaire is conducted.

In this study, there are 18 simulator outputs with calibration targets and limits for what constitutes an acceptable match.

These include male and female population sizes,

male and female HIV prevalences at three time points.

outputs that check that the behavioural features of the model matched the empirical data.

The run time for a single simulation for the study varies between 10 minutes and 3 hours.

Example: references

Full details of example are in the paper:

Ioannis Andrianakis , Ian R. Vernon, Nicky McCreesh, Trevelyan J. McKinley, Jeremy E. Oakley, Rebecca N. Nsubuga, Michael Goldstein, Richard G. White

(2015) Bayesian History Matching of Complex Infectious Disease Models Using Emulation: A Tutorial and a Case Study on HIV in Uganda,
PLOS Computational Biology.

Example: references

Full details of example are in the paper:

Ioannis Andrianakis , Ian R. Vernon, Nicky McCreesh, Trevelyan J. McKinley, Jeremy E. Oakley, Rebecca N. Nsubuga, Michael Goldstein, Richard G. White

(2015) Bayesian History Matching of Complex Infectious Disease Models Using Emulation: A Tutorial and a Case Study on HIV in Uganda,
PLOS Computational Biology.

More careful and detailed treatment in

Ioannis Andrianakis , Ian R. Vernon, Nicky McCreesh, Trevelyan J. McKinley, Jeremy E. Oakley, Rebecca N. Nsubuga, Michael Goldstein, Richard G. White

(2017) Efficient history matching of a high dimensional individual based HIV transmission model”
in SIAM/ASA Journal on Uncertainty Quantification.

which applies a development of the same ideas to a much larger version of the model (96 inputs, 50 outputs).

Example: HIV emulation and history matching

Emulation 220 point maximin Latin hypercube (100 replications at each point to average out stochasticity) 20 point Latin hypercube for validation set.
Logit transforms for outputs in [0,1].

Example: HIV emulation and history matching

Emulation 220 point maximin Latin hypercube (100 replications at each point to average out stochasticity) 20 point Latin hypercube for validation set.

Logit transforms for outputs in [0,1].

Global fit used a cubic polynomial in the inputs.

Example: HIV emulation and history matching

Emulation 220 point maximin Latin hypercube (100 replications at each point to average out stochasticity) 20 point Latin hypercube for validation set.

Logit transforms for outputs in [0,1].

Global fit used a cubic polynomial in the inputs.

All emulators validated in first wave except for 2 (HIV prevalence, male and female, 2007) which were emulated in later waves.

Example: HIV emulation and history matching

Emulation 220 point maximin Latin hypercube (100 replications at each point to average out stochasticity) 20 point Latin hypercube for validation set.

Logit transforms for outputs in [0,1].

Global fit used a cubic polynomial in the inputs.

All emulators validated in first wave except for 2 (HIV prevalence, male and female, 2007) which were emulated in later waves.

History matching 5.5×10^8 points were drawn from a 22 dimensional uniform distribution in [0,1] and the implausibility was evaluated for each one of them.

Initial implausibility criterion was the maximum implausibility

Example: HIV emulation and history matching

Emulation 220 point maximin Latin hypercube (100 replications at each point to average out stochasticity) 20 point Latin hypercube for validation set.
Logit transforms for outputs in [0,1].
Global fit used a cubic polynomial in the inputs.
All emulators validated in first wave except for 2 (HIV prevalence, male and female, 2007) which were emulated in later waves.
History matching 5.5×10^8 points were drawn from a 22 dimensional uniform distribution in [0,1] and the implausibility was evaluated for each one of them.
Initial implausibility criterion was the maximum implausibility
Evaluation was done on a 240 node cluster, completed in less than 5 minutes.

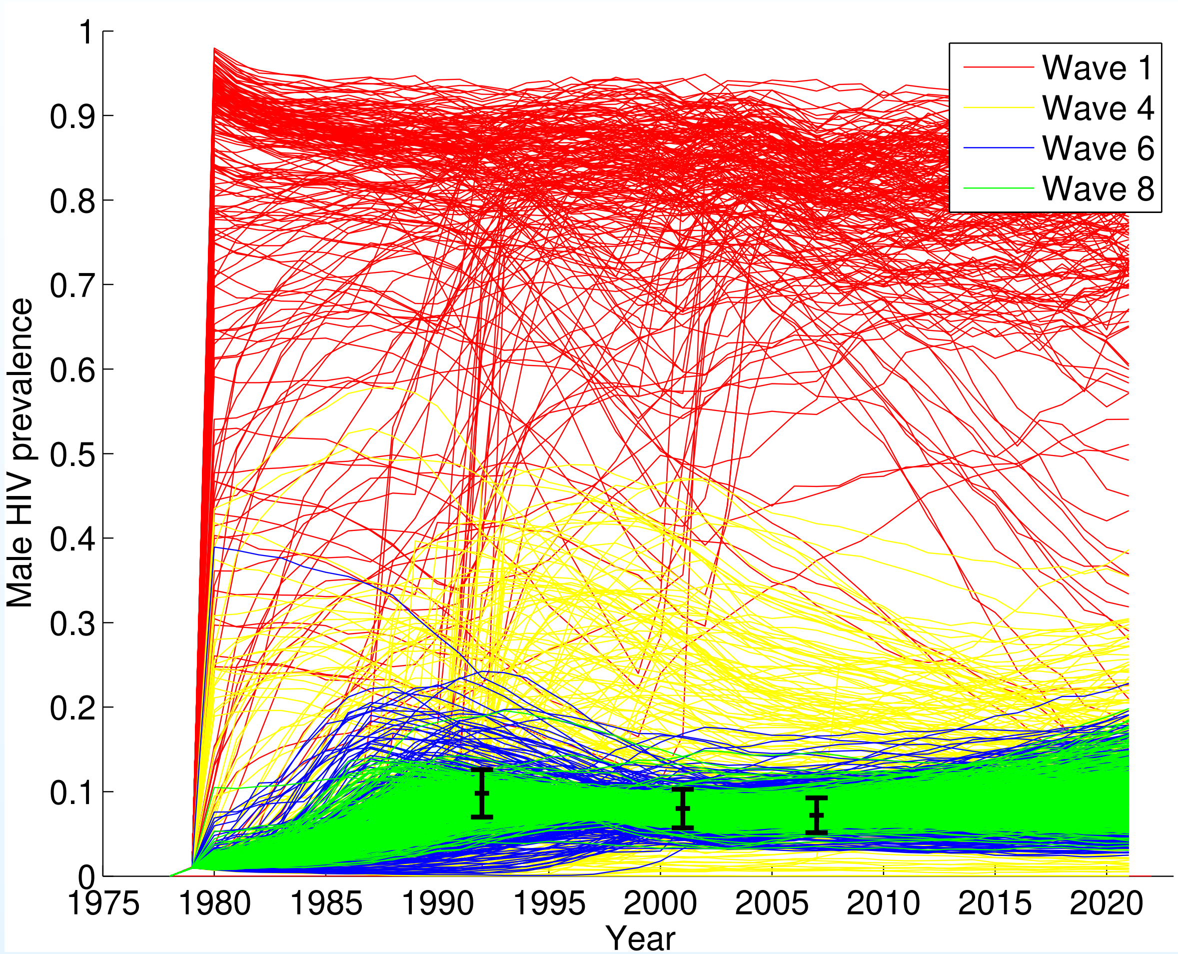
Example: HIV emulation and history matching

Emulation 220 point maximin Latin hypercube (100 replications at each point to average out stochasticity) 20 point Latin hypercube for validation set.
Logit transforms for outputs in [0,1].
Global fit used a cubic polynomial in the inputs.
All emulators validated in first wave except for 2 (HIV prevalence, male and female, 2007) which were emulated in later waves.
History matching 5.5×10^8 points were drawn from a 22 dimensional uniform distribution in [0,1] and the implausibility was evaluated for each one of them.
Initial implausibility criterion was the maximum implausibility
Evaluation was done on a 240 node cluster, completed in less than 5 minutes.
Only 21644 passed the implausibility test, implying that the volume of non-implausible space at this wave is of size 4×10^{-5} of the original input space.

Example: HIV emulation and history matching

Emulation 220 point maximin Latin hypercube (100 replications at each point to average out stochasticity) 20 point Latin hypercube for validation set.
Logit transforms for outputs in [0,1].
Global fit used a cubic polynomial in the inputs.
All emulators validated in first wave except for 2 (HIV prevalence, male and female, 2007) which were emulated in later waves.
History matching 5.5×10^8 points were drawn from a 22 dimensional uniform distribution in [0,1] and the implausibility was evaluated for each one of them.
Initial implausibility criterion was the maximum implausibility
Evaluation was done on a 240 node cluster, completed in less than 5 minutes.
Only 21644 passed the implausibility test, implying that the volume of non-implausible space at this wave is of size 4×10^{-5} of the original input space.
This process was repeated through 10 waves.

History matching for the case study



After 10 waves, we have reduced the space to about 10^{-11} of original space. Around 65% of the simulator evaluations in the final space give runs with acceptable matches to the historical data.

The HMER package

HMER (history matching and emulation in R) is a system developed under a collaboration, with Wellcome Trust funding, between

The HMER package

HMER (history matching and emulation in R) is a system developed under a collaboration, with Wellcome Trust funding, between

Andrew Iskauskas, Michael Goldstein, Ian Vernon (Durham)

The HMER package

HMER (history matching and emulation in R) is a system developed under a collaboration, with Wellcome Trust funding, between

Andrew Iskauskas, Michael Goldstein, Ian Vernon (Durham)

Nicky McCreesh, Danny Scarponi, Richard White (LSHTM)

The HMER package

HMER (history matching and emulation in R) is a system developed under a collaboration, with Wellcome Trust funding, between

Andrew Iskauskas, Michael Goldstein, Ian Vernon (Durham)

Nicky McCreesh, Danny Scarponi, Richard White (LSHTM)

TJ McKinley (Exeter)

The HMER package

HMER (history matching and emulation in R) is a system developed under a collaboration, with Wellcome Trust funding, between

Andrew Iskauskas, Michael Goldstein, Ian Vernon (Durham)

Nicky McCreesh, Danny Scarponi, Richard White (LSHTM)

TJ McKinley (Exeter)

building on a previous collaboration funded by MRC.

The HMER package

HMER (history matching and emulation in R) is a system developed under a collaboration, with Wellcome Trust funding, between

Andrew Iskauskas, Michael Goldstein, Ian Vernon (Durham)

Nicky McCreesh, Danny Scarponi, Richard White (LSHTM)

TJ McKinley (Exeter)

building on a previous collaboration funded by MRC.

The package is customised for epidemic models, but the underlying methodology is fully general.

The HMER package

The package is available from CRAN.

You can find detailed documentation at

https://github.com/andy-iskauskas/hmer

The HMER package

The package is available from CRAN.

You can find detailed documentation at

https://github.com/andy-iskauskas/hmer

The project web-page, which has lots of support material is at

https://hmer-package.github.io/website/

The HMER package

The package is available from CRAN.

You can find detailed documentation at

https://github.com/andy-iskauskas/hmer

The project web-page, which has lots of support material is at

https://hmer-package.github.io/website/

The programme has been extensively tested.

The HMER package

The package is available from CRAN.

You can find detailed documentation at

https://github.com/andy-iskauskas/hmer

The project web-page, which has lots of support material is at

https://hmer-package.github.io/website/

The programme has been extensively tested.

For example history matching a complex deterministic model for the country-level implementation of tuberculosis vaccines to 114 countries, fitting to 9–13 target measures, by varying 19–22 input parameters.

The HMER package

The package is available from CRAN.

You can find detailed documentation at

https://github.com/andy-iskauskas/hmer

The project web-page, which has lots of support material is at

https://hmer-package.github.io/website/

The programme has been extensively tested.

For example history matching a complex deterministic model for the country-level implementation of tuberculosis vaccines to 114 countries, fitting to 9–13 target measures, by varying 19–22 input parameters.

105 countries were successfully matched (i.e. producing many parameter choices which match history)

The remaining 9 countries revealed evidence of model or data misspecification.

Limitations of physical models

A physical model is a description of the way in which
system properties (the inputs to the model)
affect system behaviour (the output of the model).

Limitations of physical models

A physical model is a description of the way in which system properties (the inputs to the model) affect system behaviour (the output of the model).

This description involves two basic types of simplification.

Limitations of physical models

A physical model is a description of the way in which

system properties (the inputs to the model)

affect system behaviour (the output of the model).

This description involves two basic types of simplification.

(i) we approximate the properties of the system (as these properties are too complicated to describe fully and anyway we don't know them)

Limitations of physical models

A physical model is a description of the way in which system properties (the inputs to the model) affect system behaviour (the output of the model).

This description involves two basic types of simplification.

(i) we approximate the properties of the system (as these properties are too complicated to describe fully and anyway we don’t know them)

(ii) we approximate the rules for finding system behaviour given system properties (because of necessary mathematical and numerical simplifications, and because we do not fully understand the relationships which govern the process).

Limitations of physical models

A physical model is a description of the way in which
system properties (the inputs to the model)
affect system behaviour (the output of the model).

This description involves two basic types of simplification.

(i) we approximate the properties of the system (as these properties are too complicated to describe fully and anyway we don’t know them)

(ii) we approximate the rules for finding system behaviour given system properties (because of necessary mathematical and numerical simplifications, and because we do not fully understand the relationships which govern the process).

Neither of these approximations invalidates the modelling process.

Limitations of physical models

A physical model is a description of the way in which system properties (the inputs to the model) affect system behaviour (the output of the model).

This description involves two basic types of simplification.

(i) we approximate the properties of the system (as these properties are too complicated to describe fully and anyway we don't know them)

(ii) we approximate the rules for finding system behaviour given system properties (because of necessary mathematical and numerical simplifications, and because we do not fully understand the relationships which govern the process).

Neither of these approximations invalidates the modelling process.

Problems only arise when we forget these simplifications and confuse the analysis of the model with the corresponding analysis for the physical system itself.

Internal discrepancy

Structural uncertainty assessessessment should form a central part of the problem analysis. We may distinguish two types of model discrepancy.

Internal discrepancy

Structural uncertainty assessessment should form a central part of the problem analysis. We may distinguish two types of model discrepancy.

(i) **Internal discrepancy**
Any aspect of discrepancy we can assess by direct experiments on the computer simulator.

For example,

Internal discrepancy

Structural uncertainty assessessment should form a central part of the problem analysis. We may distinguish two types of model discrepancy.

(i) **Internal discrepancy**
Any aspect of discrepancy we can assess by direct experiments on the computer simulator.

For example,

we may vary parameters/forcing functions held fixed in the standard analysis,

we may add random noise to the state vector which the model propagates,

we may allow parameters to vary over time/space.

Internal discrepancy

Structural uncertainty assessessment should form a central part of the problem analysis. We may distinguish two types of model discrepancy.

(i) **Internal discrepancy**
Any aspect of discrepancy we can assess by direct experiments on the computer simulator.

For example,

we may vary parameters/forcing functions held fixed in the standard analysis,

we may add random noise to the state vector which the model propagates,

we may allow parameters to vary over time/space.

We assess internal discrepancy by
(i) carrying out detailed experiments to determine discrepancy variances and correlations for certain input choices,
(ii) using emulation to extend these assessments over the input space.

External discrepancy

(ii) **External discrepancy**

This arises from the inherent limitations of the modelling process embodied in the simulator.

It is determined by a combination of expert judgements and statistical estimation.

External discrepancy

(ii) **External discrepancy**

This arises from the inherent limitations of the modelling process embodied in the simulator.

It is determined by a combination of expert judgements and statistical estimation.

The simplest way to incorporate external discrepancy is to add an extra component of uncertainty to the simulator outputs. For example we may add, say, 10% additional error to account for structural discrepancy.

External discrepancy

(ii) **External discrepancy**

This arises from the inherent limitations of the modelling process embodied in the simulator.

It is determined by a combination of expert judgements and statistical estimation.

The simplest way to incorporate external discrepancy is to add an extra component of uncertainty to the simulator outputs. For example we may add, say, 10% additional error to account for structural discrepancy.

Better is to consider what we know about the limitations of the model, and build a probabilistic representation of additional features of the relationship between system properties and behaviour.

Sometimes, this is called **reification**,

(from reify - to treat an abstract concept as if it was real).

External discrepancy

(ii) **External discrepancy**

This arises from the inherent limitations of the modelling process embodied in the simulator.

It is determined by a combination of expert judgements and statistical estimation.

The simplest way to incorporate external discrepancy is to add an extra component of uncertainty to the simulator outputs. For example we may add, say, 10% additional error to account for structural discrepancy.

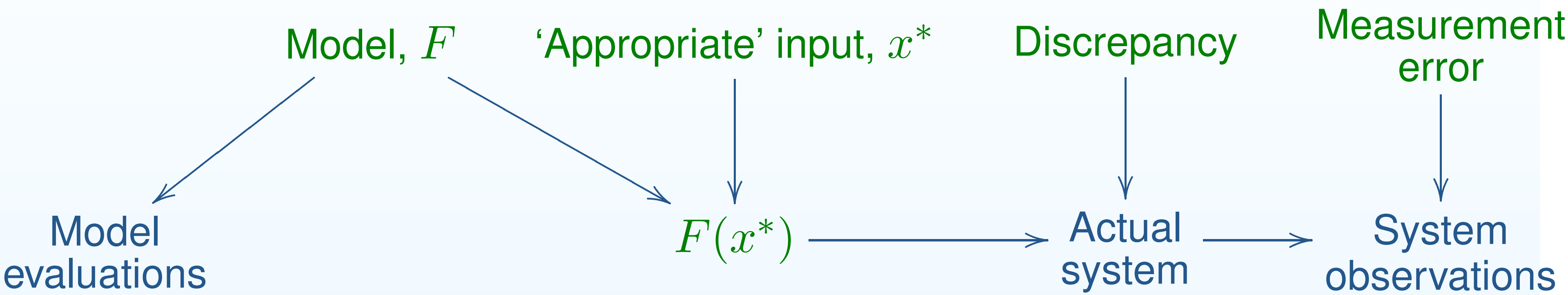
Better is to consider what we know about the limitations of the model, and build a probabilistic representation of additional features of the relationship between system properties and behaviour.

Sometimes, this is called **reification**,

(from reify - to treat an abstract concept as if it was real).

We cannot evaluate the reified simulator, but we can emulate it.

Relating the model and the system by reification

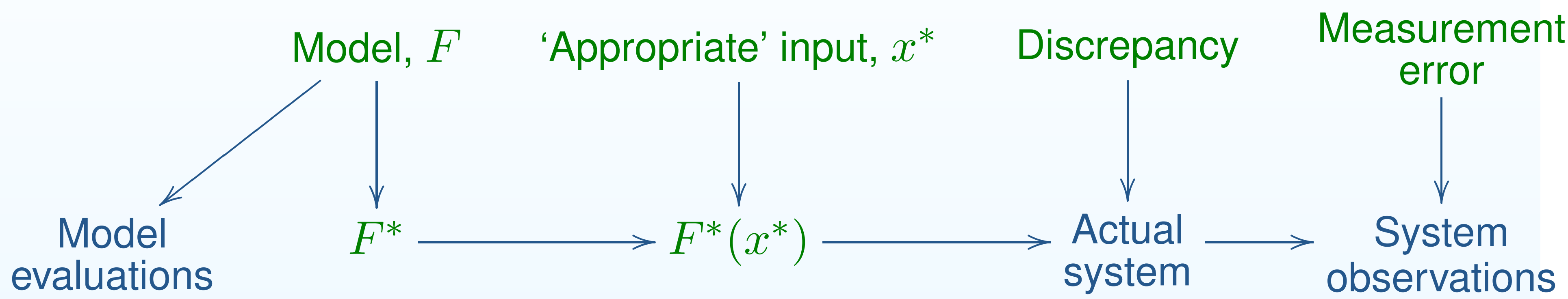


1. We start with a collection of model evaluations, and some observations on actual system and a preliminary uncertainty map.

2. Our model F is informative for y because F is informative for reified model F^*.

3. Emulating F^* allows us to calibrate/forecast/intervene based on the uncertainty representation, rather than the model.

Relating the model and the system by reification



1. We start with a collection of model evaluations, and some observations on actual system and a preliminary uncertainty map.
2. Our model F is informative for y because F is informative for reified model F^*.
3. Emulating F^* allows us to calibrate/forecast/intervene based on the uncertainty representation, rather than the model.

Relating the model and the system by reification

$$
\begin{tikzcd}
 & \text{Model, } F \arrow[dl] \arrow[d] & \text{`Appropriate' input, } x^* \arrow[d] & \text{Discrepancy} \arrow[d] & \text{Measurement error} \arrow[d] \\
\text{Model evaluations} & F^* \arrow[r] & F^*(x^*) \arrow[r] & \text{Actual system} \arrow[r] & \text{System observations}
\end{tikzcd}
$$

1. We start with a collection of model evaluations, and some observations on actual system and a preliminary uncertainty map.

2. Our model F is informative for y because F is informative for reified model F^*.

3. Emulating F^* allows us to calibrate/forecast/intervene based on the uncertainty representation, rather than the model.

Adding structural discrepancy to forecasts

We represent the physical system as the combination of the simulator forecast and the structural discrepancy.

Adding structural discrepancy to forecasts

We represent the physical system as the combination of the simulator forecast and the structural discrepancy.

From this we can derive the joint distribution of the past and the future and therefore make inferences about the future, given the past and our control choices. There are straightforward ways to do this.

Adding structural discrepancy to forecasts

We represent the physical system as the combination of the simulator forecast and the structural discrepancy.

From this we can derive the joint distribution of the past and the future and therefore make inferences about the future, given the past and our control choices. There are straightforward ways to do this.

Careful discrepancy assessment will

(i) correct our overconfidence in our projections
(by adding appropriate levels of additional uncertainty)

Adding structural discrepancy to forecasts

We represent the physical system as the combination of the simulator forecast and the structural discrepancy.

From this we can derive the joint distribution of the past and the future and therefore make inferences about the future, given the past and our control choices. There are straightforward ways to do this.

Careful discrepancy assessment will

(i) correct our overconfidence in our projections
(by adding appropriate levels of additional uncertainty)

(ii) increase our forecast accuracy
(by correcting for systematic biases in our simulator).

Adding structural discrepancy to forecasts

We represent the physical system as the combination of the simulator forecast and the structural discrepancy.

From this we can derive the joint distribution of the past and the future and therefore make inferences about the future, given the past and our control choices. There are straightforward ways to do this.

Careful discrepancy assessment will

(i) correct our overconfidence in our projections
(by adding appropriate levels of additional uncertainty)

(ii) increase our forecast accuracy
(by correcting for systematic biases in our simulator).

(iii) help us to make reliable control choices for future outcomes.
(by recognising the real world risks of our various control choices).

Concluding comments

There is a general Bayes, and Bayes linear, methodology for performing detailed uncertainty analyses for complex real world phenomena which are modelled by computer simulators.

Concluding comments

There is a general Bayes, and Bayes linear, methodology for performing detailed uncertainty analyses for complex real world phenomena which are modelled by computer simulators.

Key features of this methodology are

Concluding comments

There is a general Bayes, and Bayes linear, methodology for performing detailed uncertainty analyses for complex real world phenomena which are modelled by computer simulators.

Key features of this methodology are

(i) **simulator emulation**, to allow us to explore the full range of outputs of the simulator

Concluding comments

There is a general Bayes, and Bayes linear, methodology for performing detailed uncertainty analyses for complex real world phenomena which are modelled by computer simulators.

Key features of this methodology are

(i) **simulator emulation**, to allow us to explore the full range of outputs of the simulator

(ii) **structural discrepancy modelling**, to make reliable uncertainty statements about the real world

Concluding comments

There is a general Bayes, and Bayes linear, methodology for performing detailed uncertainty analyses for complex real world phenomena which are modelled by computer simulators.

Key features of this methodology are

(i) **simulator emulation**, to allow us to explore the full range of outputs of the simulator

(ii) **structural discrepancy modelling**, to make reliable uncertainty statements about the real world

(iii) **history matching** to identify all input choices consistent with historical data

Concluding comments

There is a general Bayes, and Bayes linear, methodology for performing detailed uncertainty analyses for complex real world phenomena which are modelled by computer simulators.

Key features of this methodology are

(i) **simulator emulation**, to allow us to explore the full range of outputs of the simulator

(ii) **structural discrepancy modelling**, to make reliable uncertainty statements about the real world

(iii) **history matching** to identify all input choices consistent with historical data

(iv) **system forecasting and optimisation** to identify real world decisions which are appropriate and robust across all conditions consistent with historical outcomes (using emulation and decision analysis)

Concluding comments

There is a general Bayes, and Bayes linear, methodology for performing detailed uncertainty analyses for complex real world phenomena which are modelled by computer simulators.

Key features of this methodology are

(i) **simulator emulation**, to allow us to explore the full range of outputs of the simulator

(ii) **structural discrepancy modelling**, to make reliable uncertainty statements about the real world

(iii) **history matching** to identify all input choices consistent with historical data

(iv) **system forecasting and optimisation** to identify real world decisions which are appropriate and robust across all conditions consistent with historical outcomes (using emulation and decision analysis)

Simulators should be constructed in ways which support these tasks.

References

J. Cumming, M. Goldstein Bayes Linear Uncertainty Analysis for Oil Reservoirs Based on Multiscale Computer Experiments (2009), in the Handbook of Applied Bayesian Analysis,eds A. O'Hagan, M. West, OUP
[Overview, with example showing treatment of multi-level design.]

M. Goldstein and N. Huntley (2017) Bayes linear emulation, history matching and forecasting for complex computer simulators, in The Handbook of Uncertainty Quantification, Ghanem, Higdon, Owhad (eds), Springer
[Overview, with example showing treatment of internal discrepancy.]

M. Goldstein and J.C.Rougier (2009). Reified Bayesian modelling and inference for physical systems (with discussion), JSPI, 139, 1221-1239
[Detailed treatment of external discrepancy.]

I. Vernon, M. Goldstein, R. Bower (2010), Galaxy Formation: a Bayesian Uncertainty Analysis (with discussion) , Bayesian Analysis, 5(4): 619–670.
[Very detailed case study for whole universe.]

M. Goldstein Subjective Bayesian analysis: principles and practice (2006) Bayesian Analysis, 1, 403-420 (and 'Rejoinder to discussion': 465-472)
[Why you should be a subjective Bayesian!]