IDM workshop: emulation and history matching
Part 1: General Principles

Michael Goldstein, Ian Vernon*

*Thanks to MRc, for funding for example in presentation.

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.

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, tsunamis and so forth, are all studied by large computer simulators.

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, tsunamis and so forth, are all studied by large computer simulators.

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, tsunamis and so forth, are all studied by large computer simulators.

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

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

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, tsunamis and so forth, are all studied by large computer simulators.

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

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.

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, tsunamis and so forth, are all studied by large computer simulators.

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

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.

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, tsunamis and so forth, are all studied by large computer simulators.

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

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.

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

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"
to appear 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).

Simple 1D Exponential Growth Example

We are interested in the **concentration of a chemical** evolving in time.
We model this concentration as $f(x,t)$ where x is a rate parameter and t is time.

Simple 1D Exponential Growth Example

We are interested in the **concentration of a chemical** evolving in time.
We model this concentration as $f(x,t)$ where x is a rate parameter and t is time.

We think $f(x,t)$ satisfies the differential equation or model:

$$\frac{df(x,t)}{dt} = xf(x,t) \qquad \Longrightarrow \qquad f(x,t) = f_0 \exp(xt)$$

Simple 1D Exponential Growth Example

We are interested in the **concentration of a chemical** evolving in time.
We model this concentration as $f(x,t)$ where x is a rate parameter and t is time.

We think $f(x,t)$ satisfies the differential equation or model:

$$\frac{df(x,t)}{dt} = xf(x,t) \qquad \Longrightarrow \qquad f(x,t) = f_0 \exp(xt)$$

We suppose the initial conditions are $f_0 = f(x, t=0) = 1$.

Simple 1D Exponential Growth Example

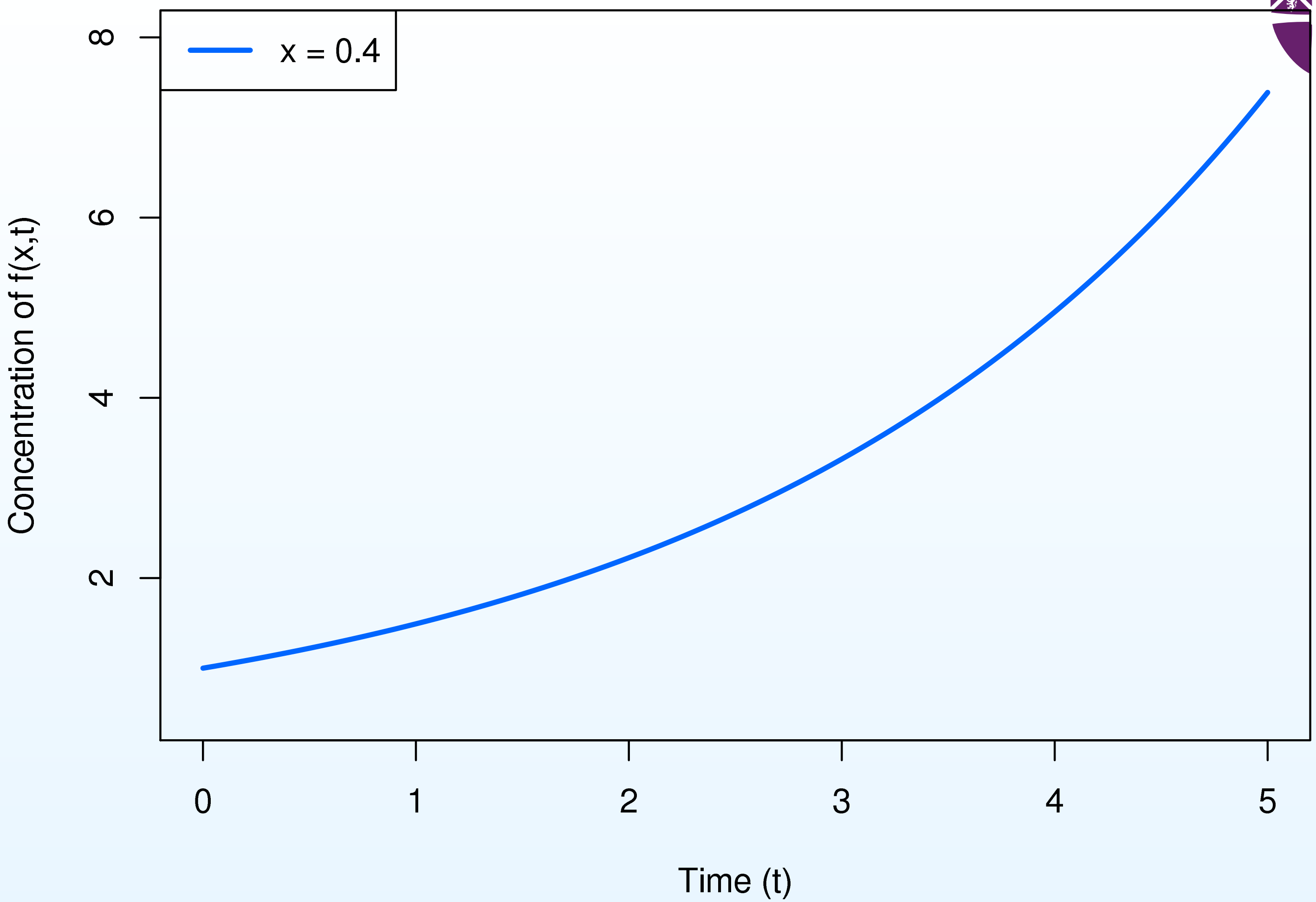
We are interested in the **concentration of a chemical** evolving in time.
We model this concentration as $f(x,t)$ where x is a rate parameter and t is time.

We think $f(x,t)$ satisfies the differential equation or model:

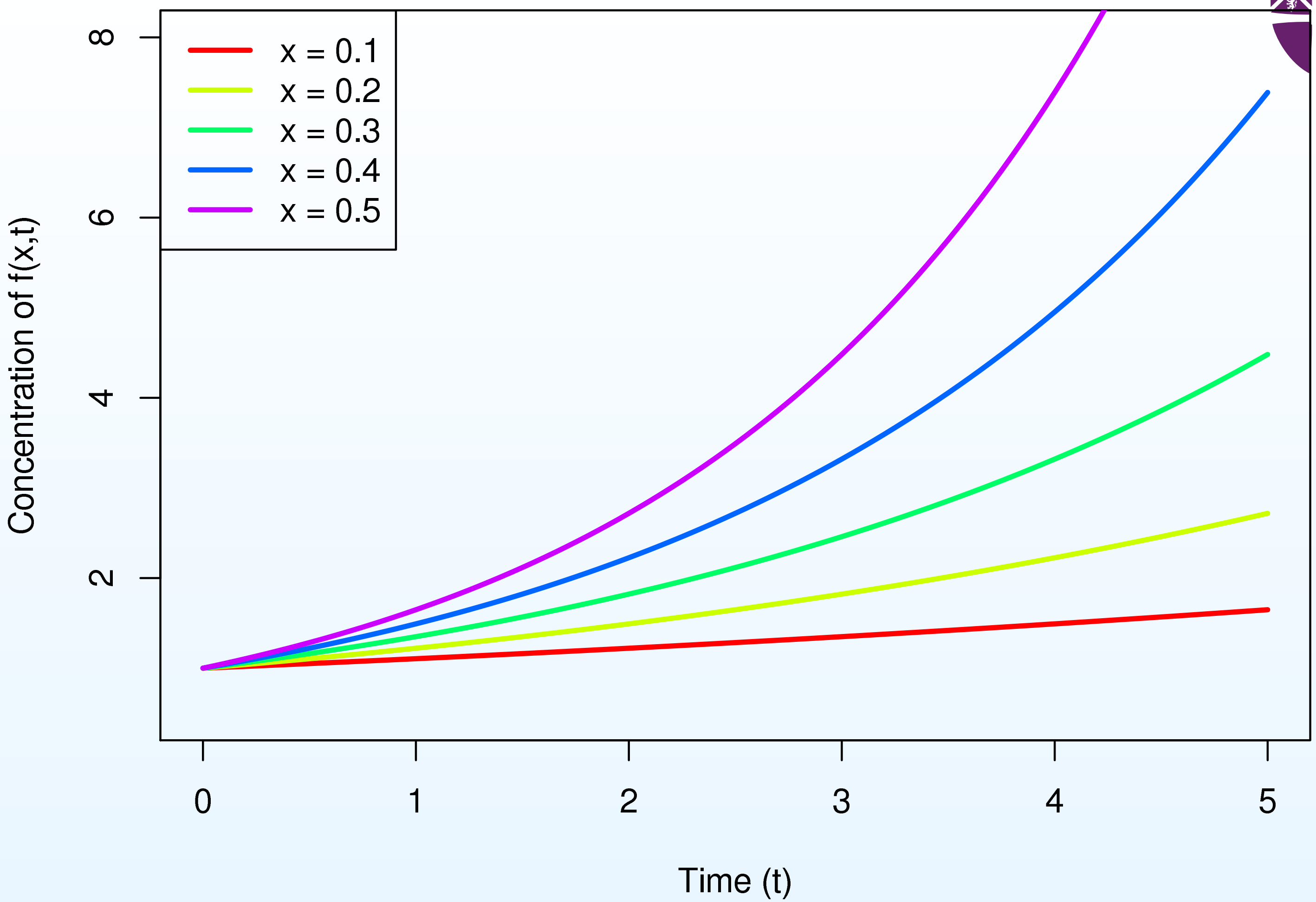
$$\frac{df(x,t)}{dt} = x f(x,t) \qquad \Longrightarrow \qquad f(x,t) = f_0 \exp(xt)$$

We suppose the initial conditions are $f_0 = f(x, t=0) = 1$.

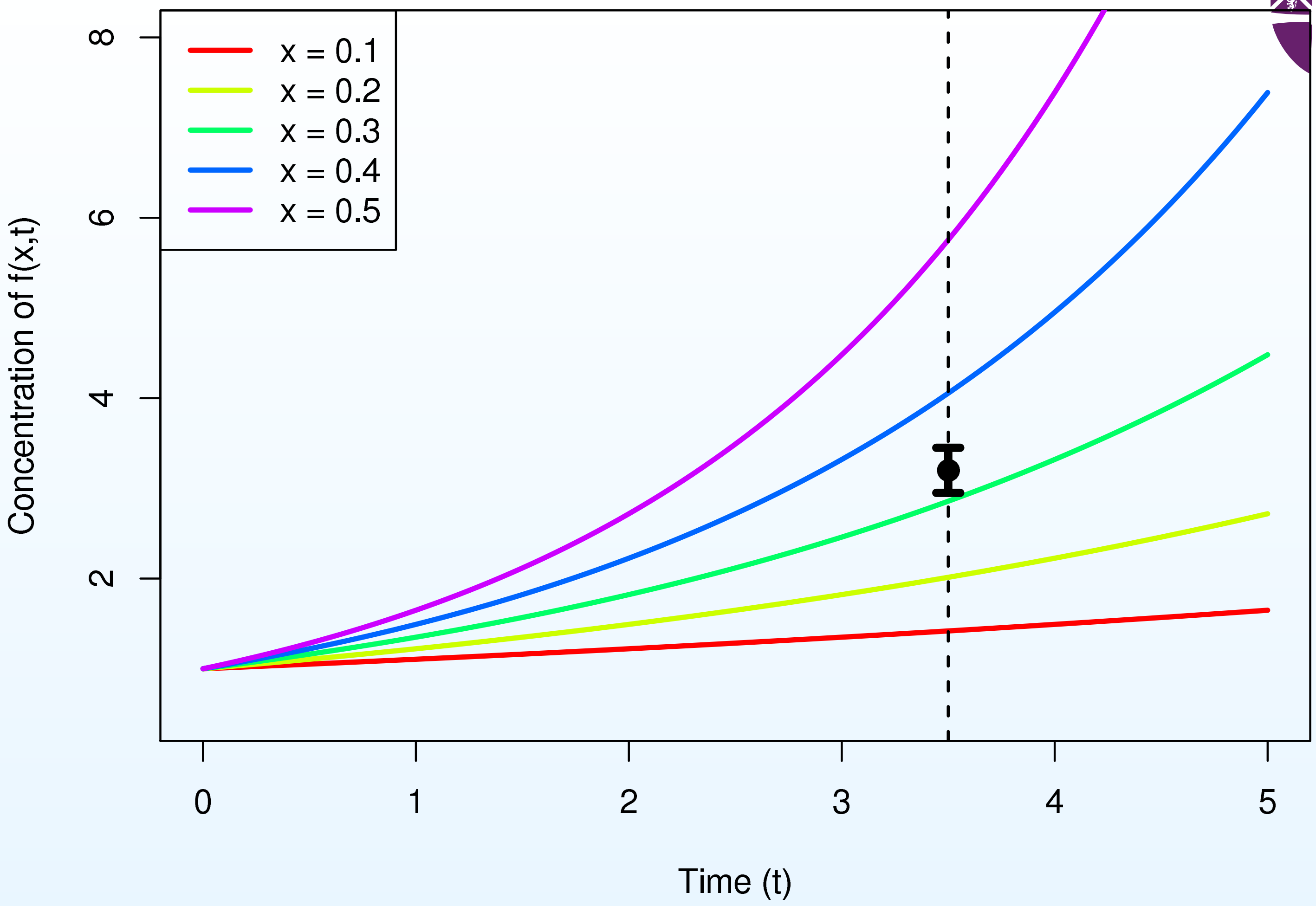
Model features an input parameter x which we want to learn about.



One “model run” with the input parameter $x = 0.4$

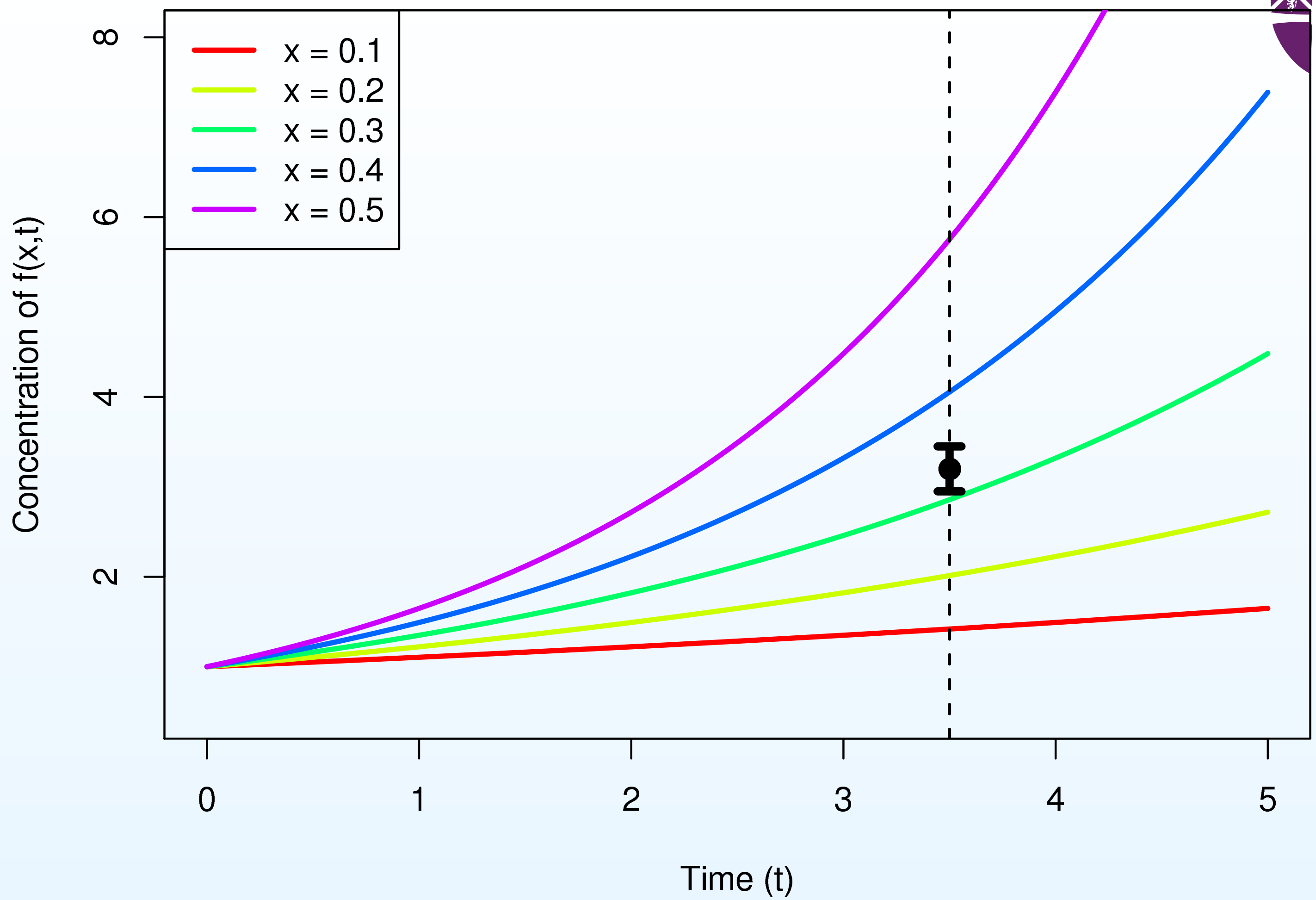


Five model runs with the input parameter varying from $x = 0.1$ to $x = 0.5$



We are going to measure $f(x,t)$ at $t = 3.5$

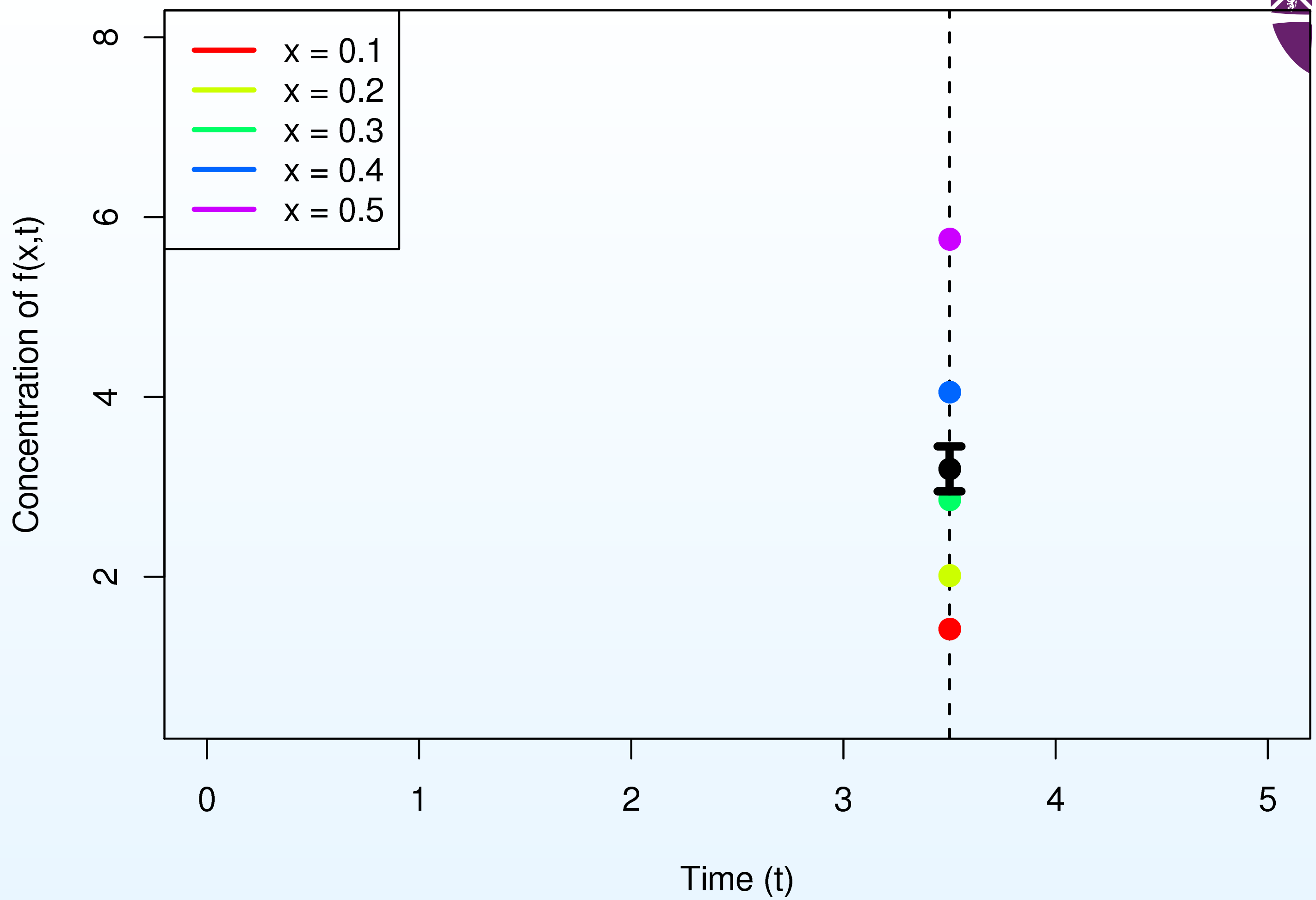
The measurement comes with measurement error.



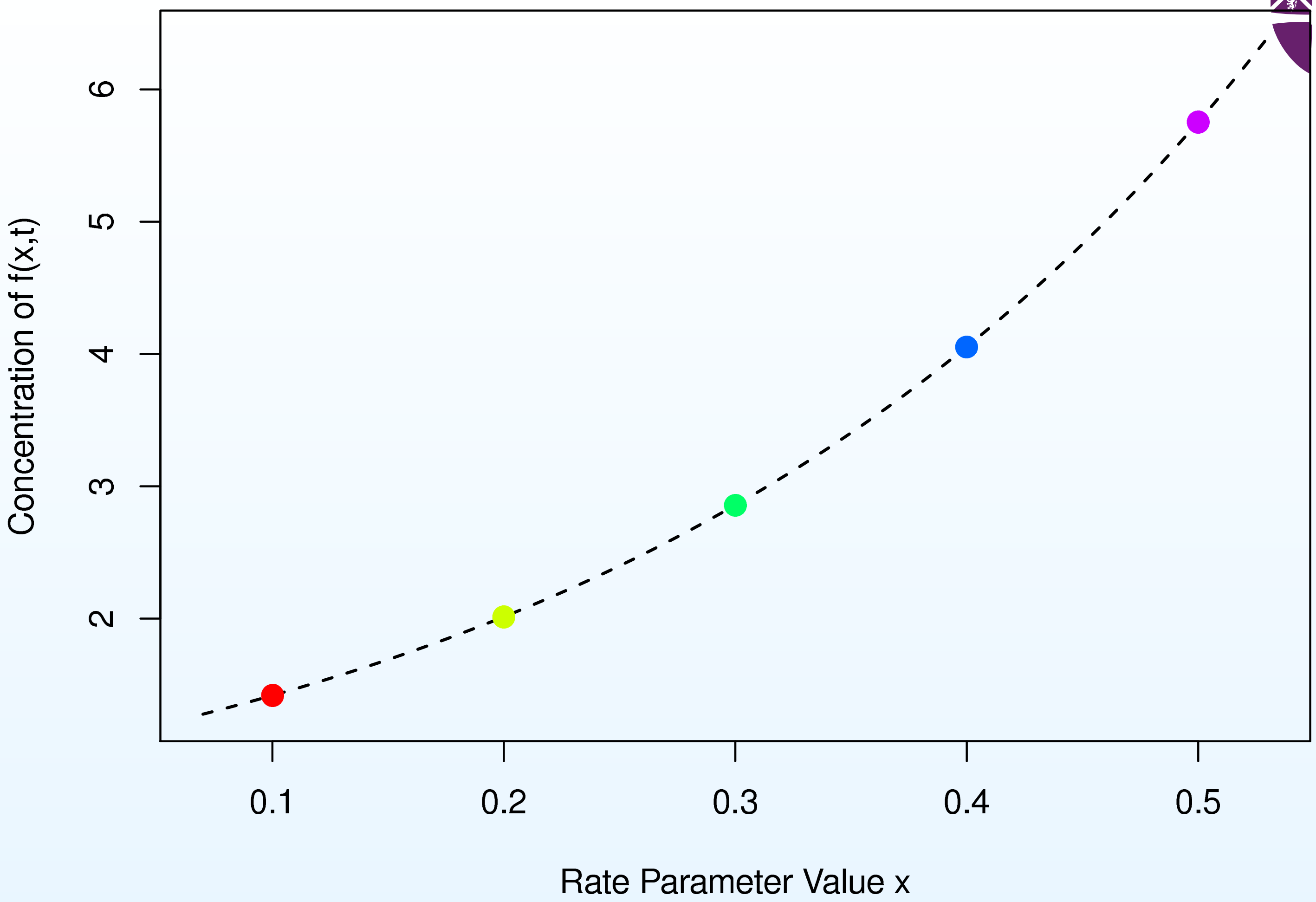
We are going to measure $f(x, t)$ at $t = 3.5$

The measurement comes with measurement error.

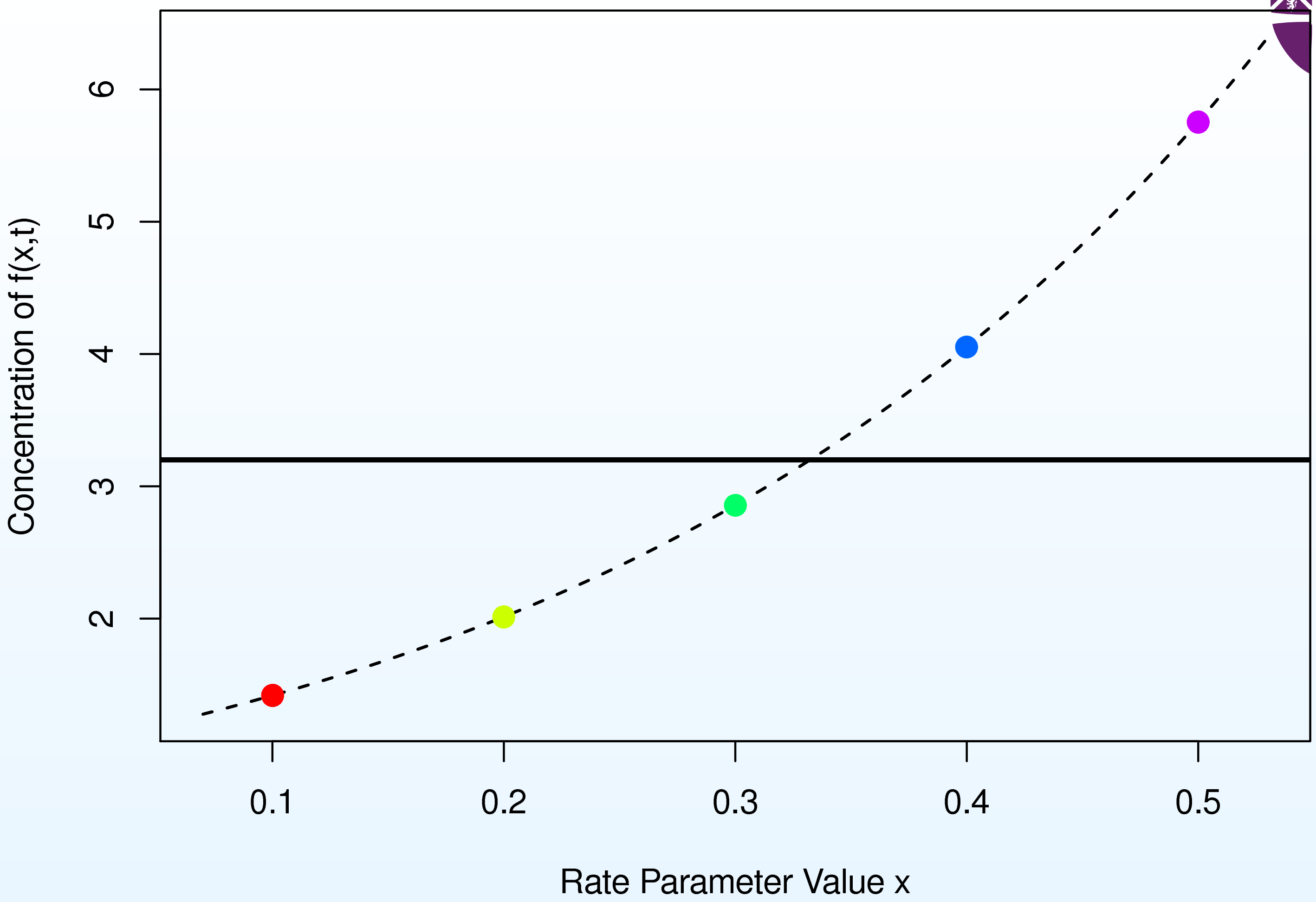
For which values of x is the output $f(x, t = 3.5)$ consistent with the observation?



To answer this, we can now discard other values of $f(x,t)$ and think of $f(x, t = 3.5)$ as a function of x only, that is take $f(x) \equiv f(x, t = 3.5)$

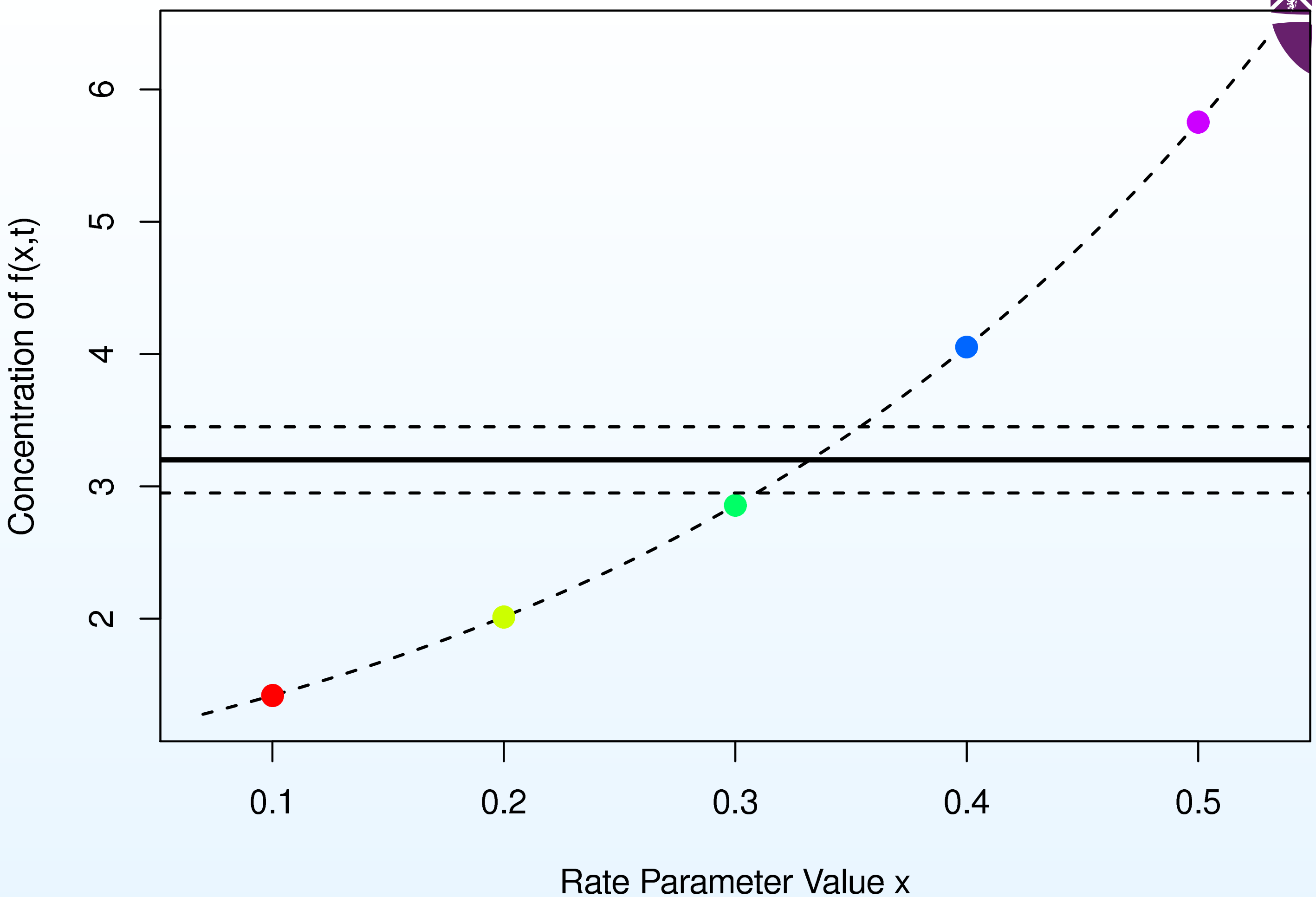


We plot the concentration $f(x)$ as a function of the input parameter x.



We plot the concentration $f(x)$ as a function of the input parameter x.

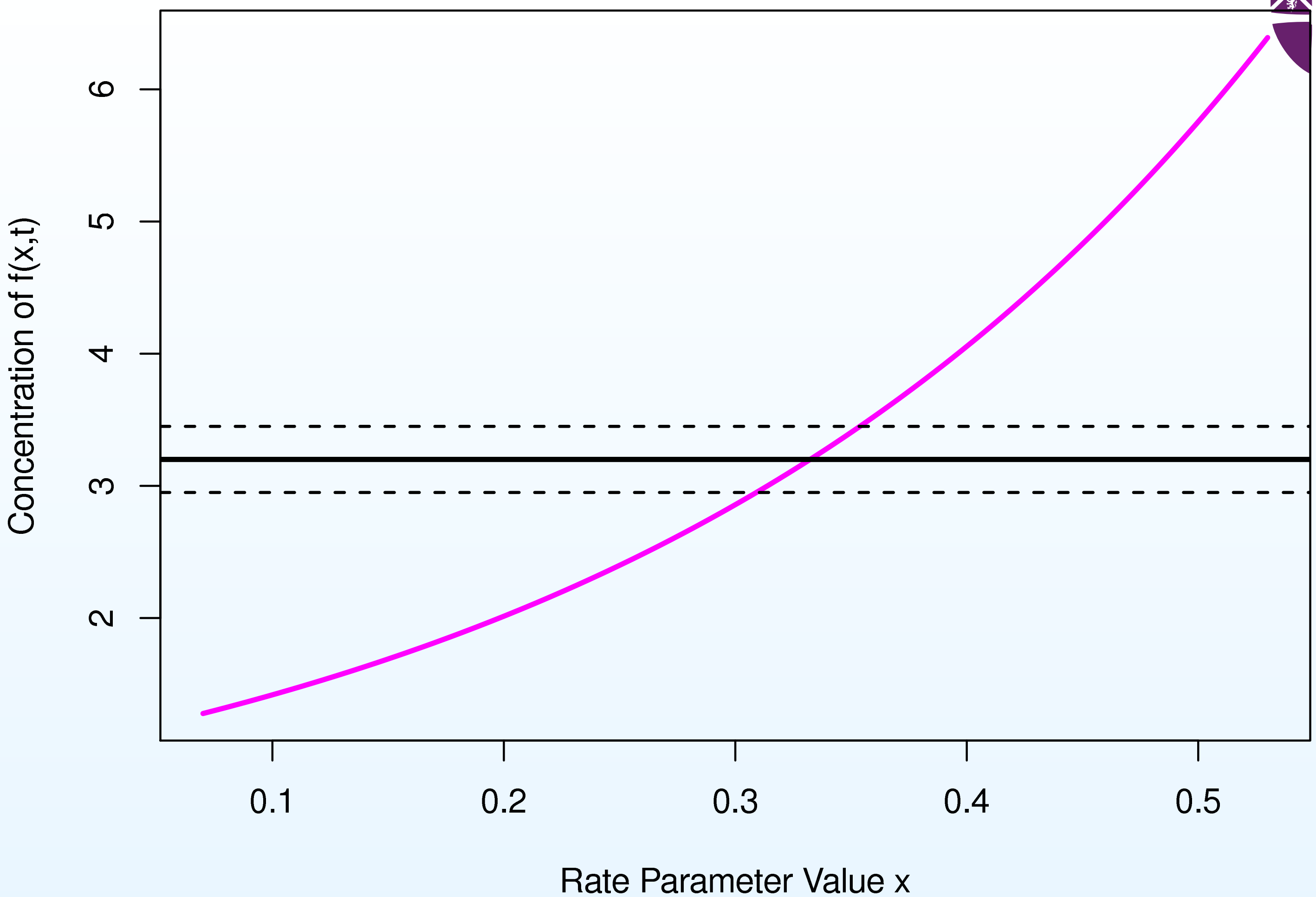
Black horizontal line: the observed measurement of f



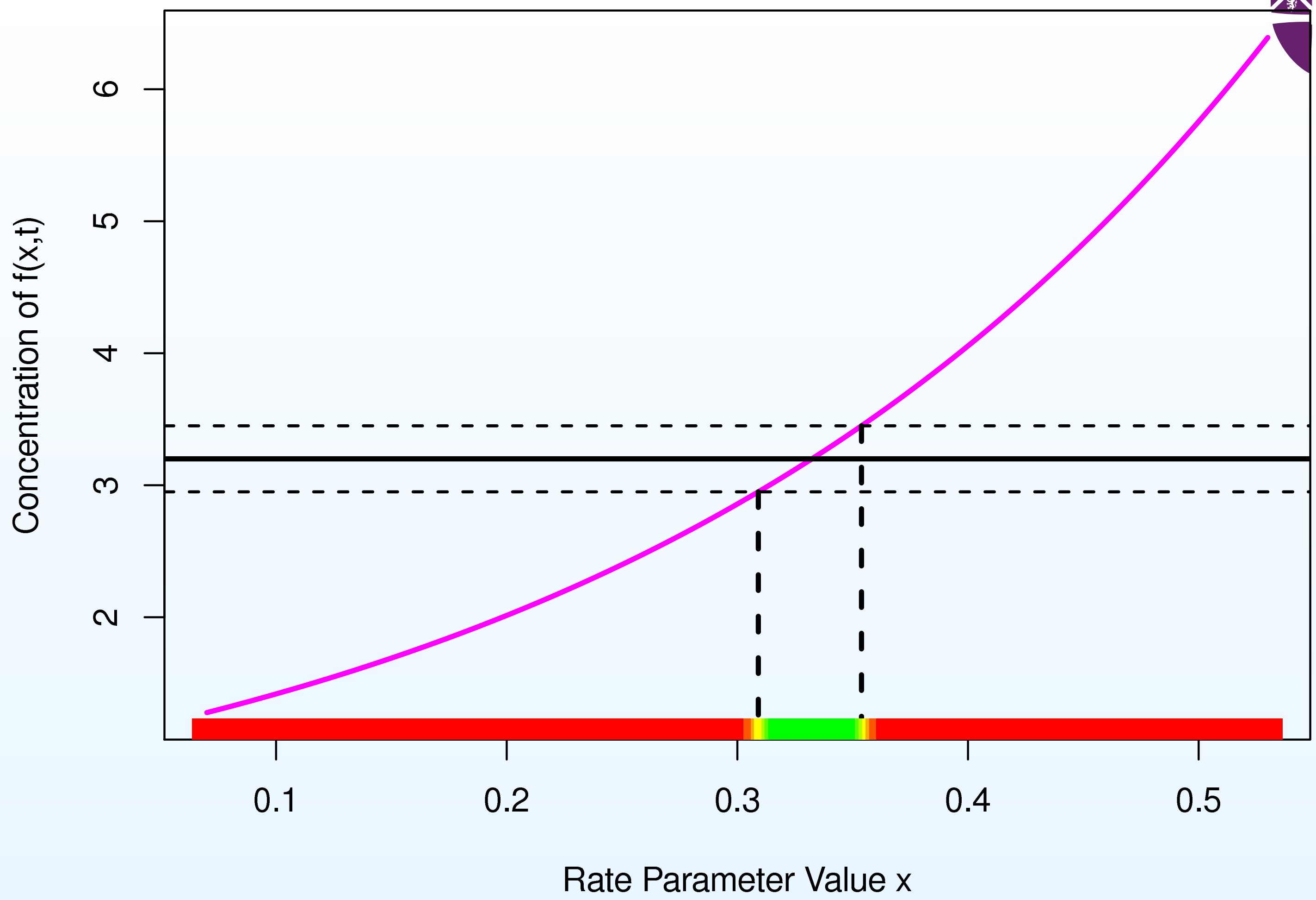
We plot the concentration $f(x)$ as a function of the input parameter x.

Black horizontal line: the observed measurement of f

Dashed horizontal lines: the measurement errors



Uncertainty in the measurement of $f(x, t = 3.5)$ leads to uncertainty in the inferred values of x.



Uncertainty in the measurement of $f(x, t = 3.5)$ 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 the behaviour of the model, not the real real world

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

Three problems with this approach

Durham University

(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 the behaviour of the model, not the real 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 structure of problem

Each simulator can be conceived as a function $f(x)$, where

x: input vector, representing unknown properties of the physical system;

$f(x)$: output vector representing system behaviour.

General structure of problem

Each simulator can be conceived as a function $f(x)$, where

x: input vector, representing unknown properties of the physical system;

$f(x)$: output vector representing system behaviour.

Interest in the “appropriate” choice, x^*, for the system properties x,

General structure of problem

Each simulator can be conceived as a function $f(x)$, where

x: input vector, representing unknown properties of the physical system;

$f(x)$: output vector representing system behaviour.

Interest in the “appropriate” choice, x^*, for the system properties x,

how informative $f(x^*)$ is for actual system behaviour, y.

General structure of problem

Each simulator can be conceived as a function $f(x)$, where

x: input vector, representing unknown properties of the physical system;

$f(x)$: output vector representing system behaviour.

Interest in the “appropriate” choice, x^*, for the system properties x,

how informative $f(x^*)$ is for actual system behaviour, y.

the use of historical observations z, observed with error on a subset y_h of y, corresponding to a sub-vector $f_h(x)$ of the $f(x)$

General structure of problem

Each simulator can be conceived as a function $f(x)$, where

x: input vector, representing unknown properties of the physical system;

$f(x)$: output vector representing system behaviour.

Interest in the "appropriate" choice, x^*, for the system properties x,

how informative $f(x^*)$ is for actual system behaviour, y.

the use of historical observations z, observed with error on a subset y_h of y, corresponding to a sub-vector $f_h(x)$ of the $f(x)$

the best assignment of any decision inputs, d, in the model.

General structure of problem

Each simulator can be conceived as a function $f(x)$, where

x: input vector, representing unknown properties of the physical system;

$f(x)$: output vector representing system behaviour.

Interest in the “appropriate” choice, x^*, for the system properties x,

how informative $f(x^*)$ is for actual system behaviour, y.

the use of historical observations z, observed with error on a subset y_h of y, corresponding to a sub-vector $f_h(x)$ of the $f(x)$

the best assignment of any decision inputs, d, in the model.

In almost all cases
(i) evaluation of $f(x)$ is expensive
(ii) inferring x^* from z is hard
(iii) relating $f(x^*)$ to y is challenging.

The Bayesian approach

The Bayesian approach unifies and synthesises all of the different sources of uncertainty arising in such problems into an overall judgement of uncertainty.

The Bayesian approach

The Bayesian approach unifies and synthesises all of the different sources of uncertainty arising in such problems into an overall judgement of uncertainty.

In this approach, all probablities are the subjective judgements of individuals so, not **the** probability that the disease will spread, but Anne's probability or Bob's probability (which may differ) for this outcome.

The Bayesian approach

The Bayesian approach unifies and synthesises all of the different sources of uncertainty arising in such problems into an overall judgement of uncertainty.

In this approach, all probablities are the subjective judgements of individuals so, not **the** probability that the disease will spread, but Anne's probability or Bob's probability (which may differ) for this outcome.

The Bayesian approach 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

The Bayesian approach unifies and synthesises all of the different sources of uncertainty arising in such problems into an overall judgement of uncertainty.

In this approach, all probablities are the subjective judgements of individuals so, not **the** probability that the disease will spread, but Anne’s probability or Bob’s probability (which may differ) for this outcome.

The Bayesian approach 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, particularly for models of moderate size and complexity.

The Bayesian approach

The Bayesian approach unifies and synthesises all of the different sources of uncertainty arising in such problems into an overall judgement of uncertainty.

In this approach, all probablities are the subjective judgements of individuals so, not **the** probability that the disease will spread, but Anne's probability or Bob's probability (which may differ) for this outcome.

The Bayesian approach 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, particularly for models of moderate size and complexity.

An excellent resource for work in this area is the Managing Uncertainty in Complex Models web-site, www.mucm.ac.uk

The Bayes linear approach

The Bayesian approach can be difficult in large problems because of the extreme level of detail which is required in the specification of beliefs.

The Bayes linear approach

The Bayesian approach can be difficult in large problems because of the extreme level of detail which is required in the specification of beliefs.

In the Bayes linear approach, we combine prior judgements of uncertainty with observational data, using **expectation** rather than **probability** as the primitive.

The Bayes linear approach

The Bayesian approach can be difficult in large problems because of the extreme level of detail which is required in the specification of beliefs.

In the Bayes linear approach, we combine prior judgements of uncertainty with observational data, using **expectation** rather than **probability** as the primitive.

This approach is similar in spirit to a full Bayes analysis, but uses a much simpler approach for prior specification and analysis, and so offers a practical methodology for analysing partially specified beliefs for large problems.

The Bayes linear approach

The Bayesian approach can be difficult in large problems because of the extreme level of detail which is required in the specification of beliefs.

In the Bayes linear approach, we combine prior judgements of uncertainty with observational data, using **expectation** rather than **probability** as the primitive.

This approach is similar in spirit to a full Bayes analysis, but uses a much simpler approach for prior specification and analysis, and so offers a practical methodology for analysing partially specified beliefs for large problems.

Bayes linear adjustment may be viewed as an approximation to a full Bayes analysis or the appropriate analysis given a partial specification.

The Bayes linear approach

The Bayes linear adjusted expectation and variance for vector y given vector z are

$$\mathsf{E}_z[y] = \mathrm{E}(y) + \mathrm{Cov}(y,z)\mathrm{Var}(z)^{-1}(z - \mathrm{E}(z)),$$
$$\mathsf{Var}_z[y] = \mathrm{Var}(y) - \mathrm{Cov}(y,z)\mathrm{Var}(z)^{-1}\mathrm{Cov}(z,y)$$

The Bayes linear approach

The Bayes linear adjusted expectation and variance for vector y given vector z are

$$\mathsf{E}_z[y] = \mathrm{E}(y) + \mathrm{Cov}(y,z)\mathrm{Var}(z)^{-1}(z - \mathrm{E}(z)),$$
$$\mathsf{Var}_z[y] = \mathrm{Var}(y) - \mathrm{Cov}(y,z)\mathrm{Var}(z)^{-1}\mathrm{Cov}(z,y)$$

For a detailed treatment, see
Bayes linear Statistics: Theory and Methods, 2007, (Wiley)
Michael Goldstein and David Wooff

For a quick overview, see
Bayes linear analysis, 2015, Michael Goldstein, in Wiley StatsRef: Statistics Reference Online (7 pages)

And all of our papers in this area contain examples of Bayes linear computations.

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. [For example, large disease transmission models.]

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. [For example, large disease transmission models.]

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. [For example, large disease transmission models.]

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.

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. [For example, large disease transmission models.]

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. [For example, large disease transmission models.]

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. [For example, large disease transmission models.]

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. [For example, large disease transmission models.]

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

Durham University

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,

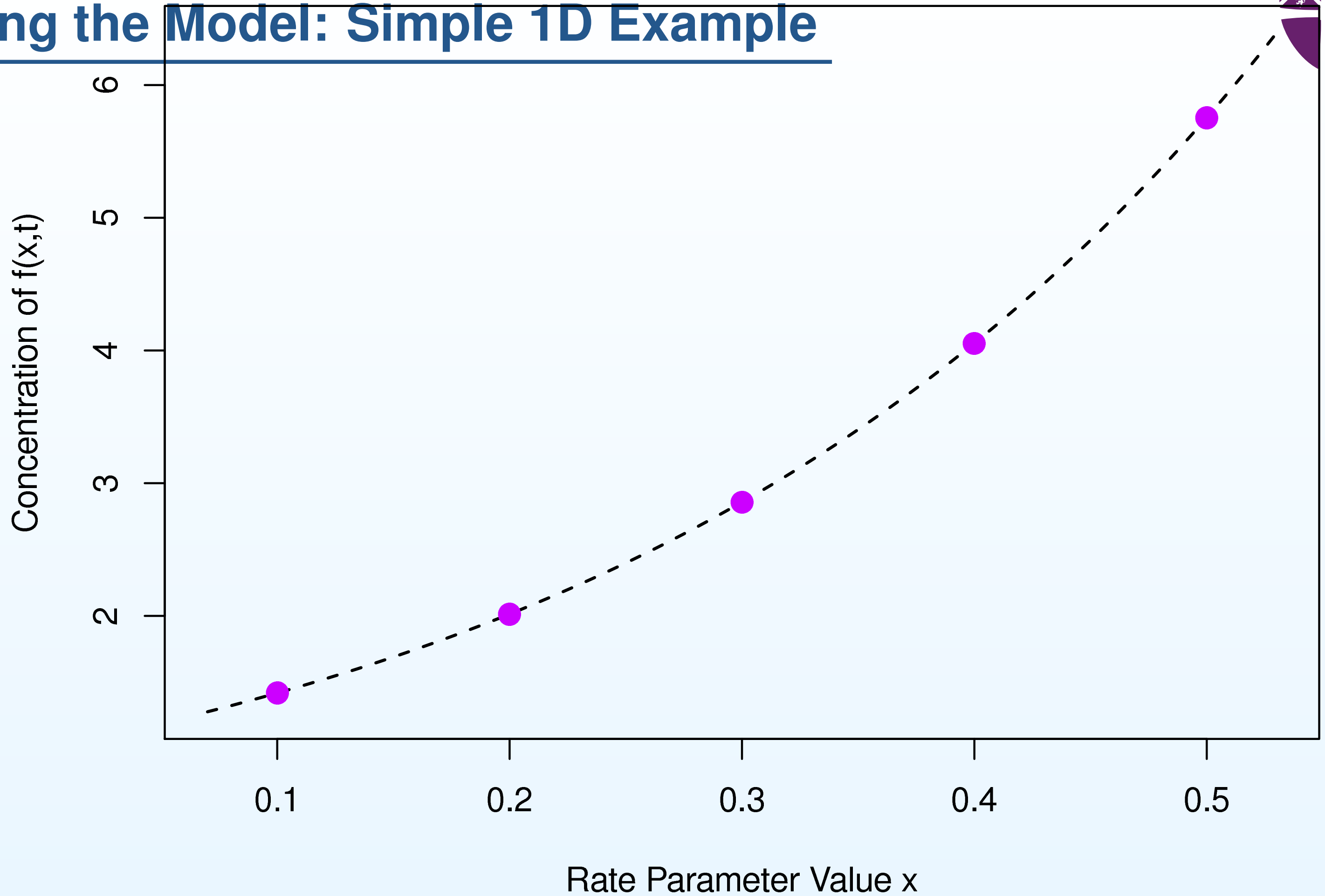
g_{ij} are known deterministic functions of x, (for example, polynomials)

Local Variation

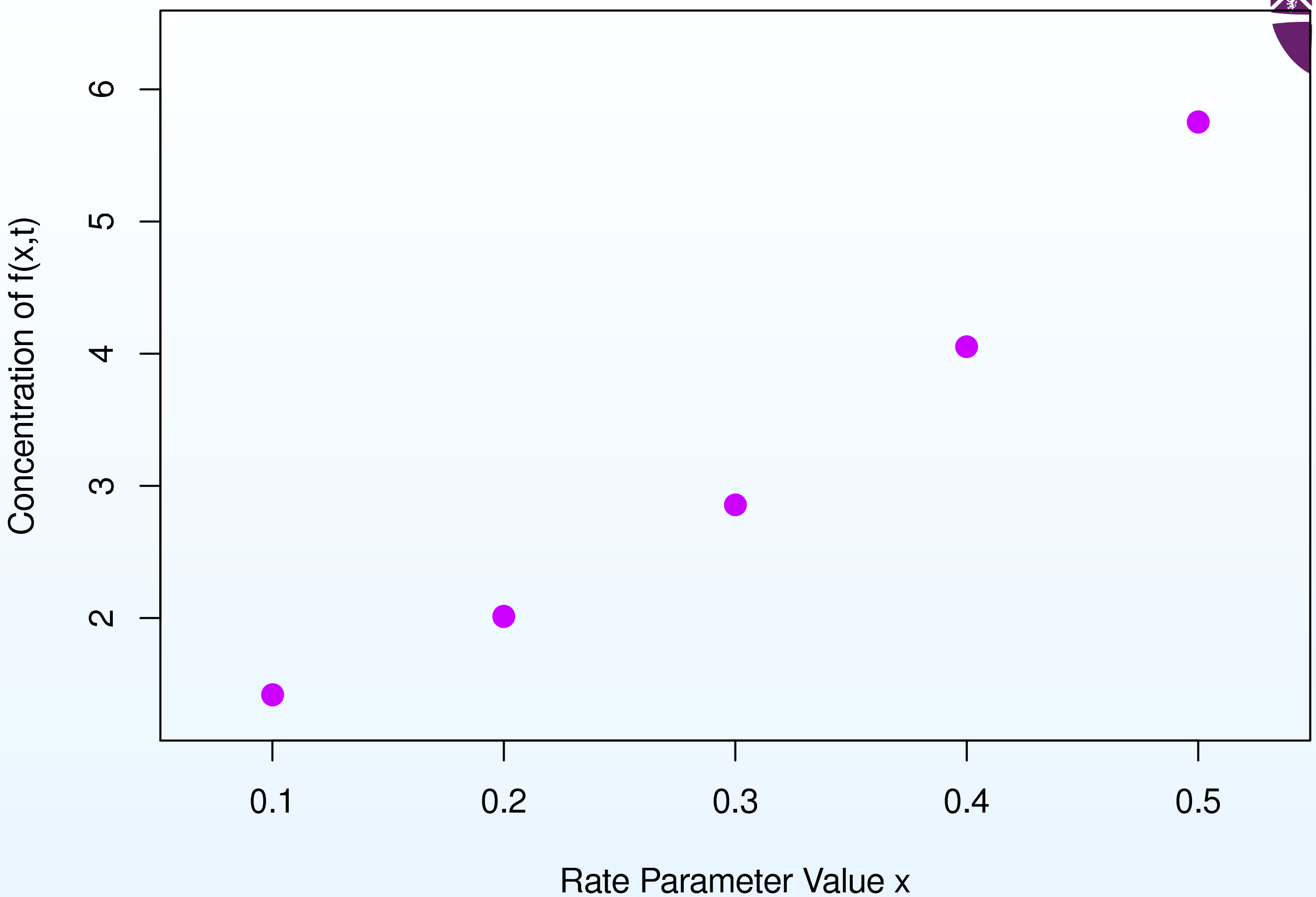
$u_i(x)$ is a second order stationary stochastic process, with (for example) correlation function

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

Emulating the Model: Simple 1D Example

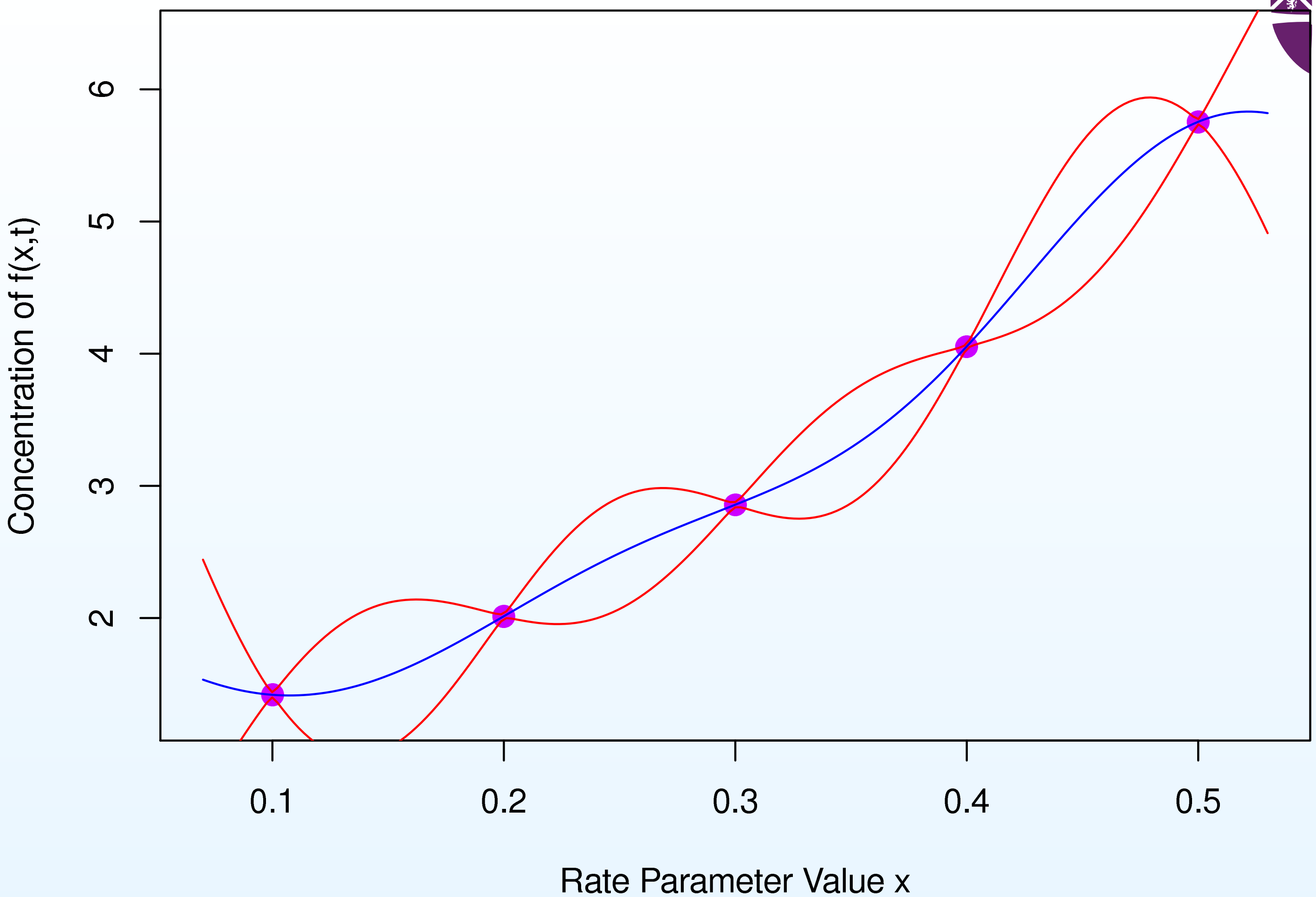


Consider the graph of $f(x)$: in general we do not have the analytic solution of $f(x)$, here given by the dashed line.

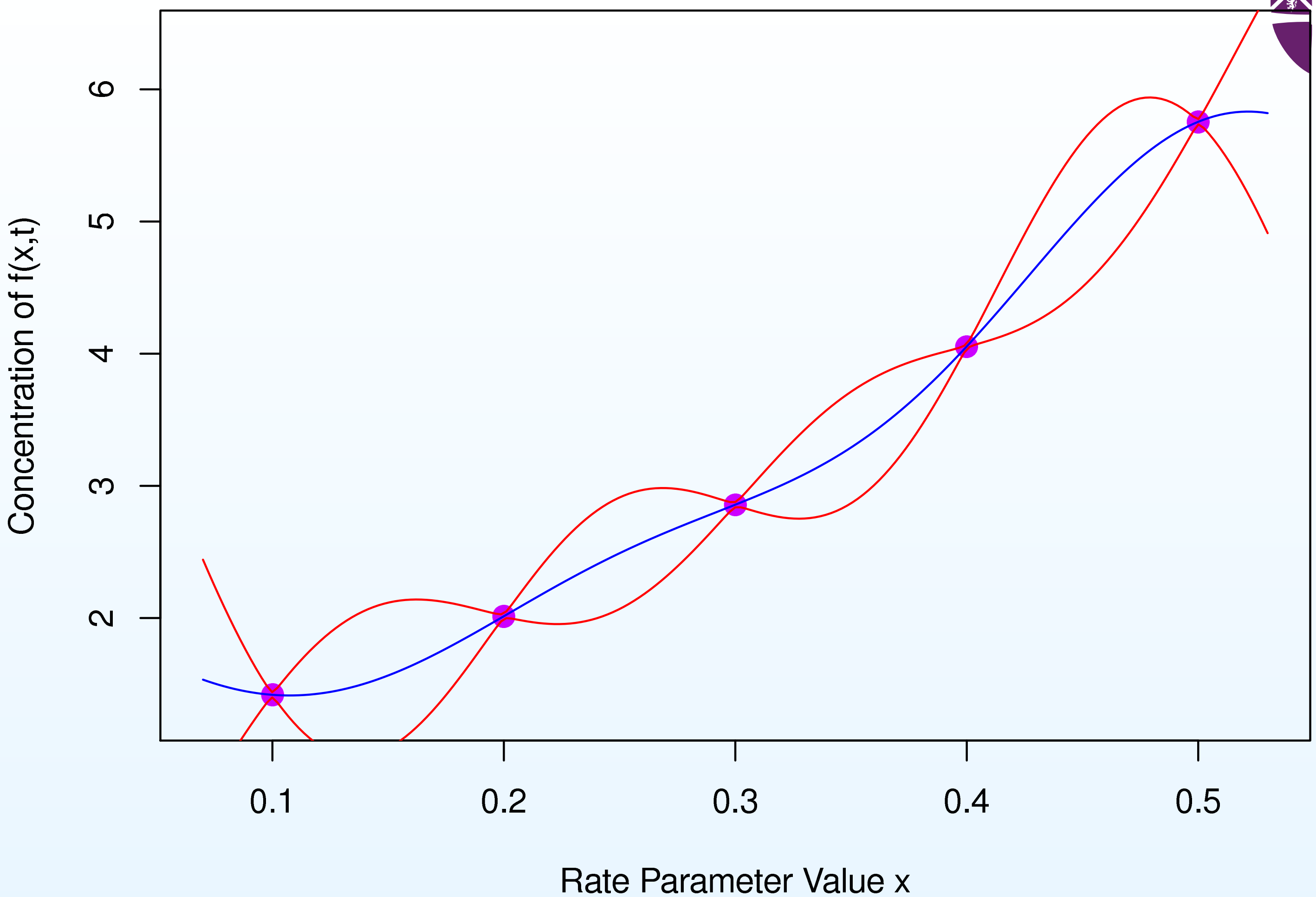


Consider the graph of $f(x)$: in general we do not have the analytic solution of $f(x)$, here given by the dashed line.

Instead we only have a finite number of runs of the model, in this case five.

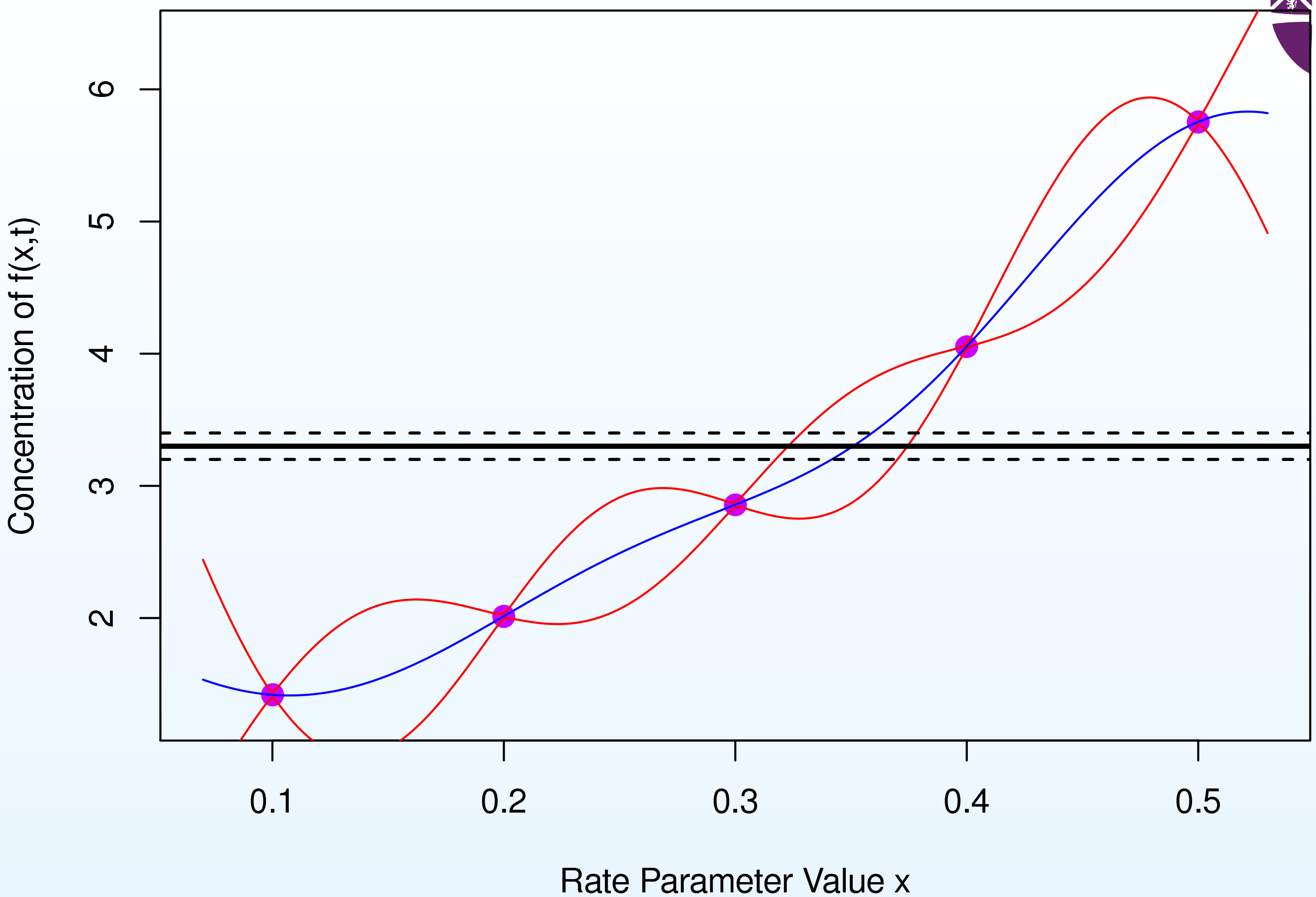


The emulator can be used to represent our beliefs about the behaviour of the model at untested values of x, and is fast to evaluate.

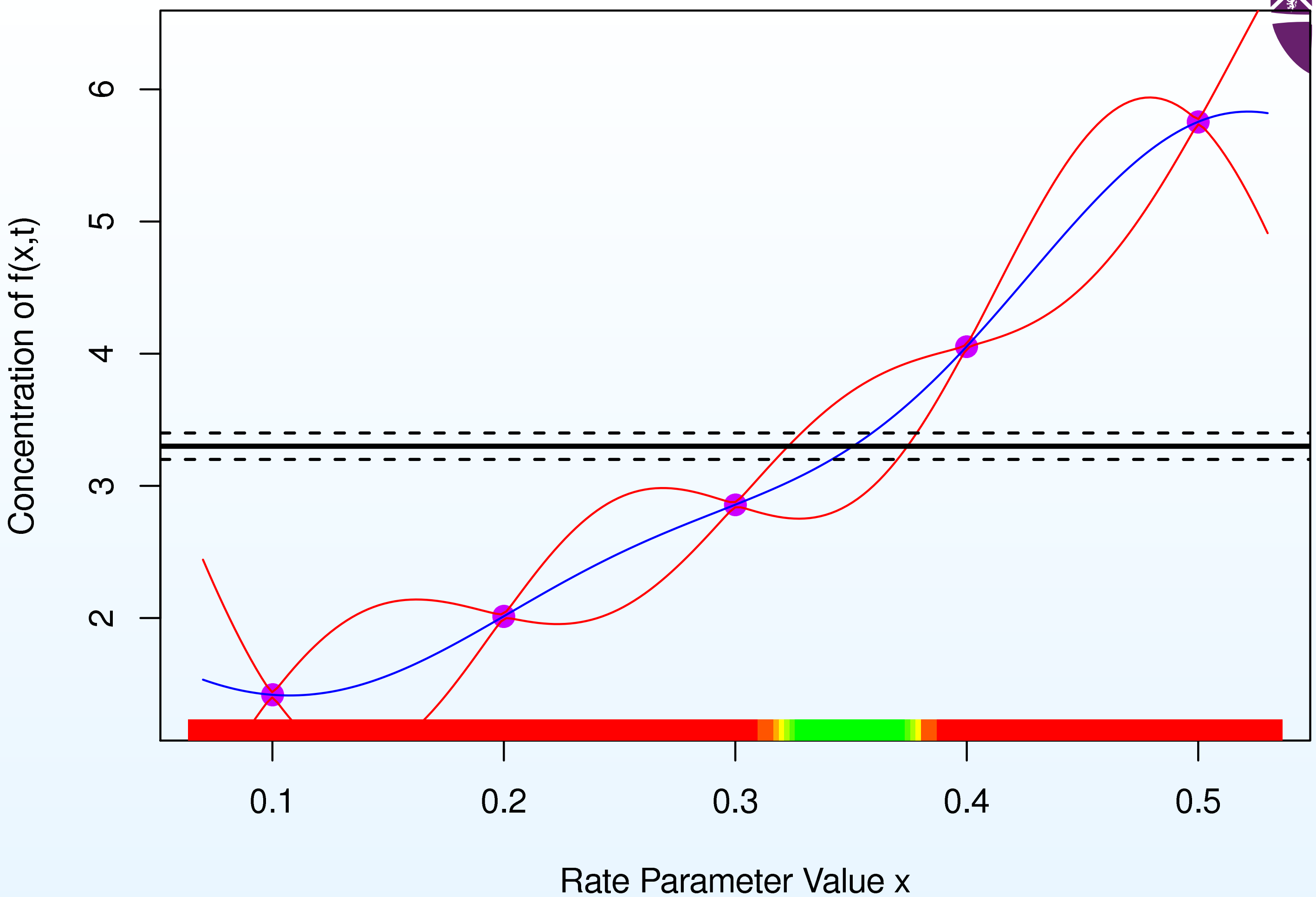


The emulator can be used to represent our beliefs about the behaviour of the model at untested values of x, and is fast to evaluate.

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.



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, given a collection of carefully chosen model evaluations, using our favourite statistical tools - generalised least squares, maximum likelihood, Bayes - with a generous helping of expert judgement.

Emulation methods

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

If the model is slow to evaluate, we typically create an informed prior assessment based on a fast approximation, then combine with a carefully designed set of runs of the full simulator to construct the emulator.

Emulation methods

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

If the model is slow to evaluate, we typically create an informed prior assessment based on a fast approximation, then combine with a carefully designed set of runs of the full simulator to construct the emulator.

We use efficient space filling (multi-level) designs to generate the set of simulator evaluations to carry out in order to fit the emulators.
(For example, maximin Latin Hypercubes.)

Emulation methods

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

If the model is slow to evaluate, we typically create an informed prior assessment based on a fast approximation, then combine with a carefully designed set of runs of the full simulator to construct the emulator.

We use efficient space filling (multi-level) designs to generate the set of simulator evaluations to carry out in order to fit the emulators.
(For example, maximin Latin Hypercubes.)

We 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).

Issues with calibration

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

Issues with calibration

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)

Issues with calibration

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)

Issues with calibration

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 match of the simulator outputs $f_h(x)$ to observed data, z, is good enough, 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 match of the simulator outputs $f_h(x)$ to observed data, z, is good enough, taking into account all of the uncertainties in the problem.

$C(z)$ might be empty - suggesting problems with the simulator.

History matching

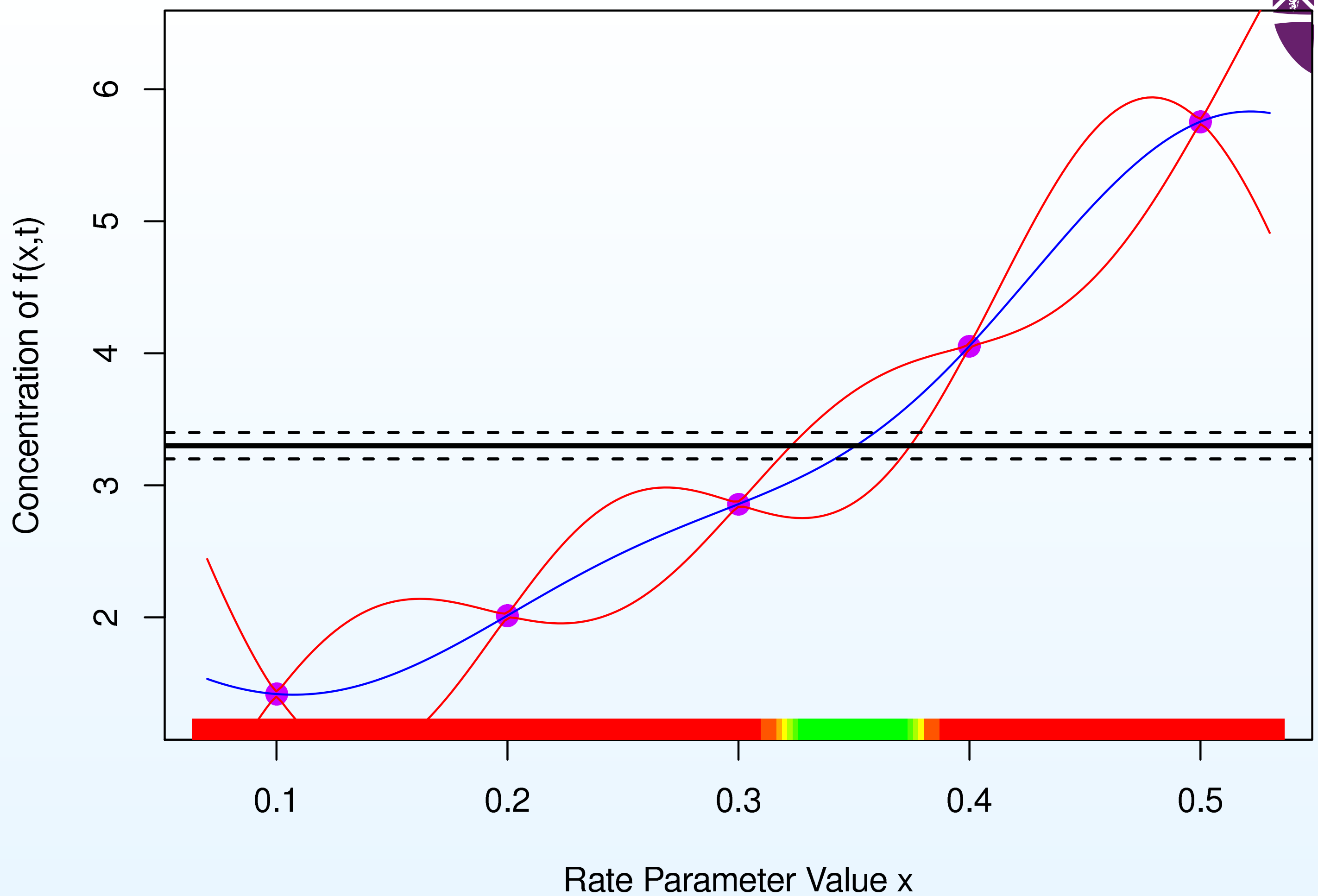
A conceptually simple procedure is “history matching”.

This means finding the collection, $C(z)$, of all input choices x for which the match of the simulator outputs $f_h(x)$ to observed data, z, is good enough, taking into account all of the uncertainties in the problem.

$C(z)$ might be empty - suggesting problems with the simulator.

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

Therefore, even if we do wish to calibrate the simulator, history matching is a useful preliminary step.



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).

History matching by implausibility

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 all of the individual variance terms e.g. measurement error, emulator error, discrepancy error and so forth.)

History matching by implausibility

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 all of the individual variance terms e.g. measurement error, emulator error, discrepancy error and so forth.)

The implausibility calculation can be performed univariately, or by multivariate calculation over sub-vectors.

The implausibilities are then combined to identify x with large $I(x)$ as implausible, i.e. unlikely to be appropriate choices for system inputs.

History matching in waves

Having identified a non-implausible region of the input space, we refocus our analysis on this region, by

History matching in waves

Having identified a non-implausible region of the input space, we refocus our analysis on this region, by

(i) making more simulator runs in the subregion

History matching in waves

Having identified a non-implausible region of the input space, we refocus our analysis on this region, by

(i) making more simulator runs in the subregion

(ii) refitting our emulators over the subregion,

History matching in waves

Having identified a non-implausible region of the input space, we refocus our analysis on this region, by

(i) making more simulator runs in the subregion

(ii) refitting our emulators over the subregion,

(iii) emulating additional outputs (which can be well emulated in the reduced space)

History matching in waves

Having identified a non-implausible region of the input space, we refocus our analysis on this region, by

(i) making more simulator runs in the subregion

(ii) refitting our emulators over the subregion,

(iii) emulating additional outputs (which can be well emulated in the reduced space)

and repeating the implausibility analysis.

History matching in waves

Having identified a non-implausible region of the input space, we refocus our analysis on this region, by

(i) making more simulator runs in the subregion

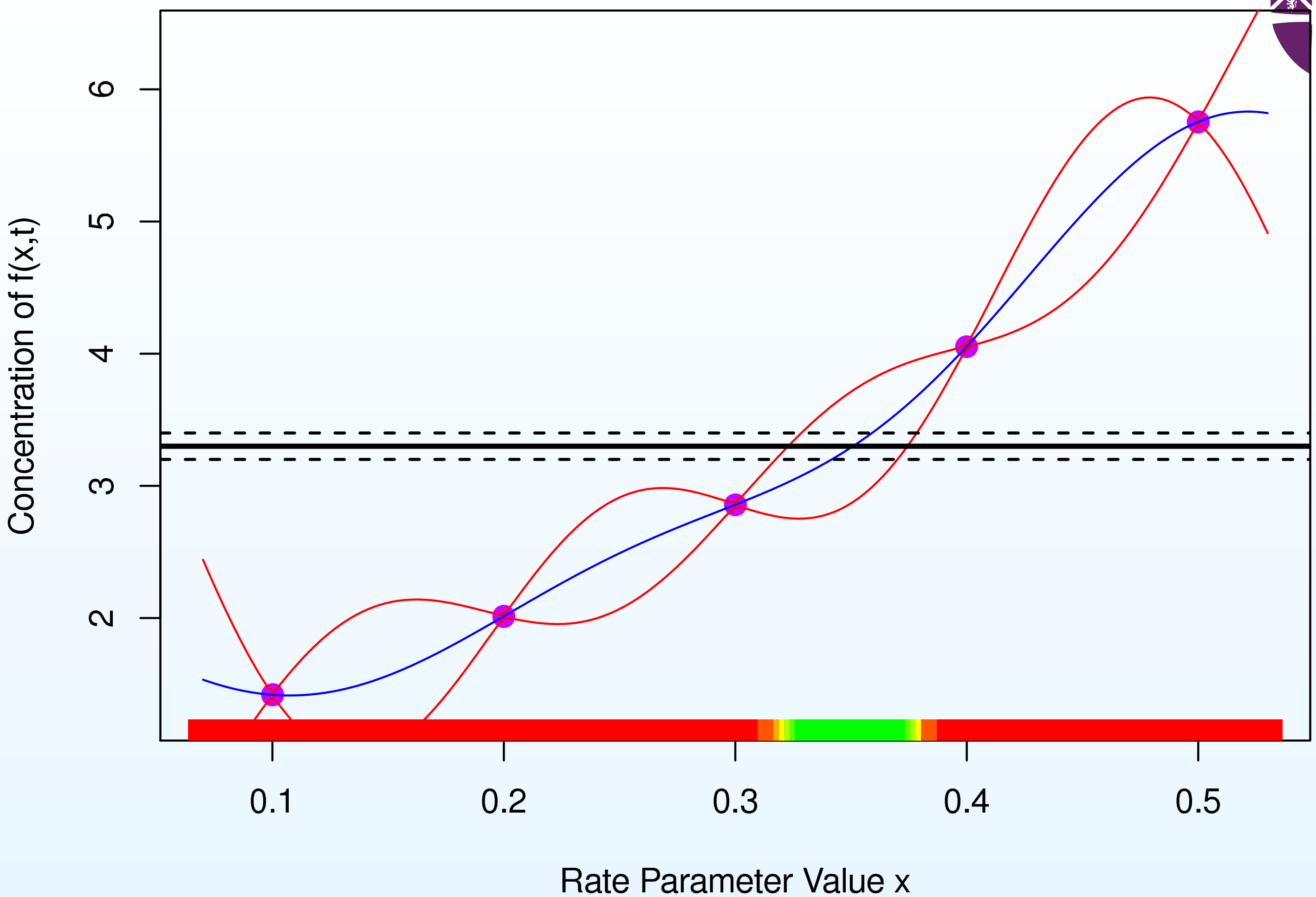
(ii) refitting our emulators over the subregion,

(iii) emulating additional outputs (which can be well emulated in the reduced space)

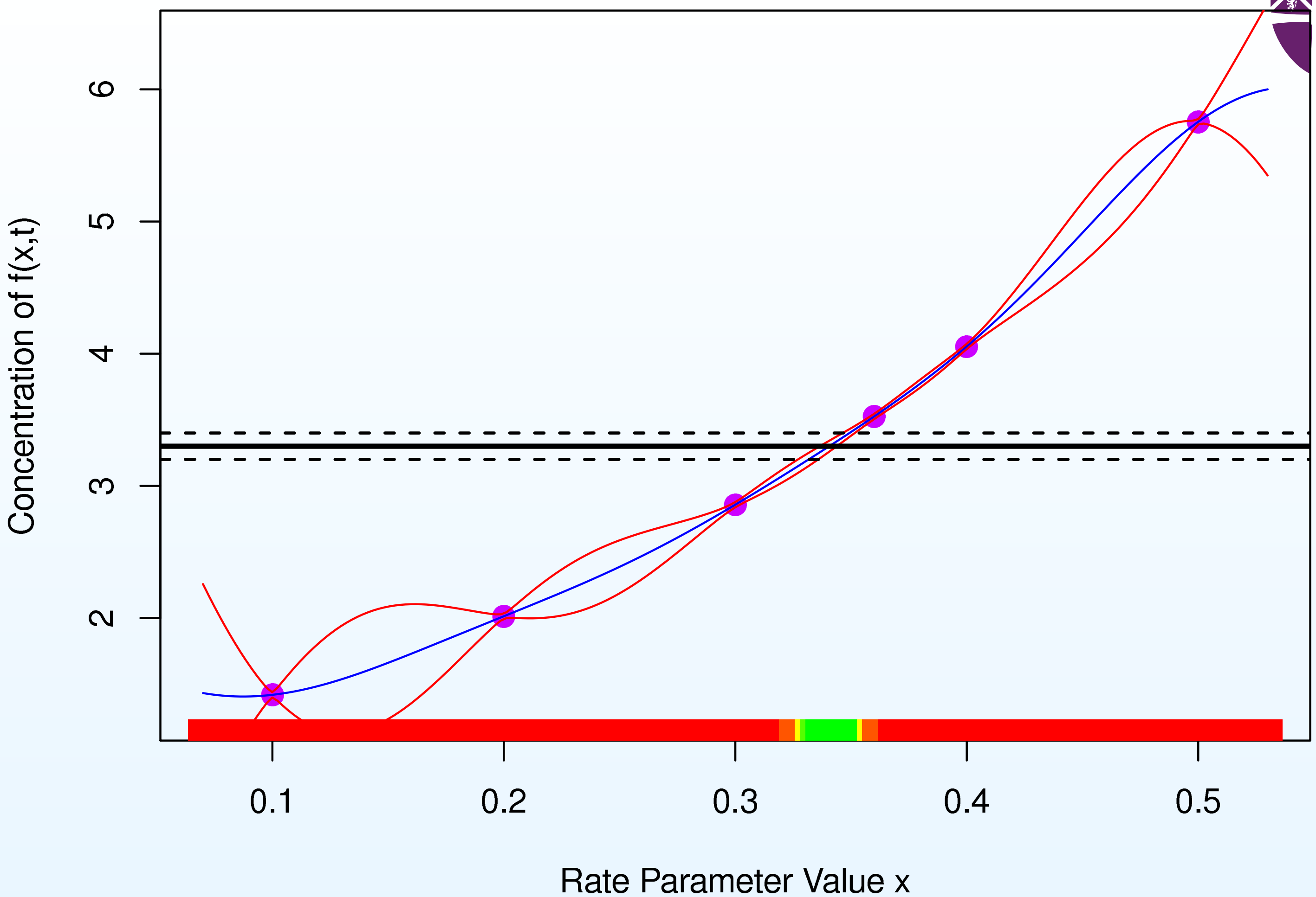
and repeating the implausibility analysis.

We continue until (hopefully) we identify the region of acceptable matches.

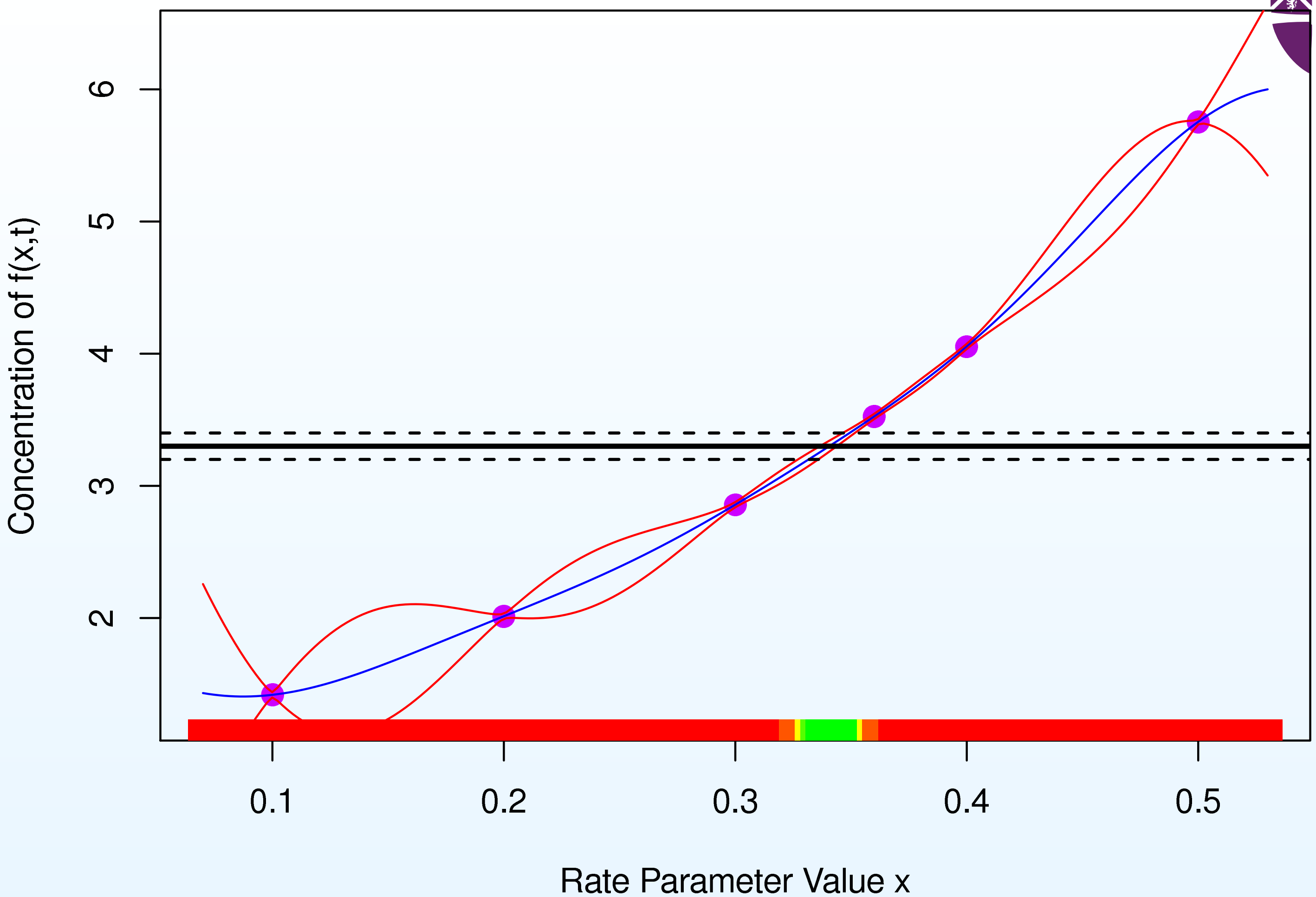
(This is a form of iterative global search.)



We now remove all of the implausible x values (the red values) and resample and re-emulate within the green region.

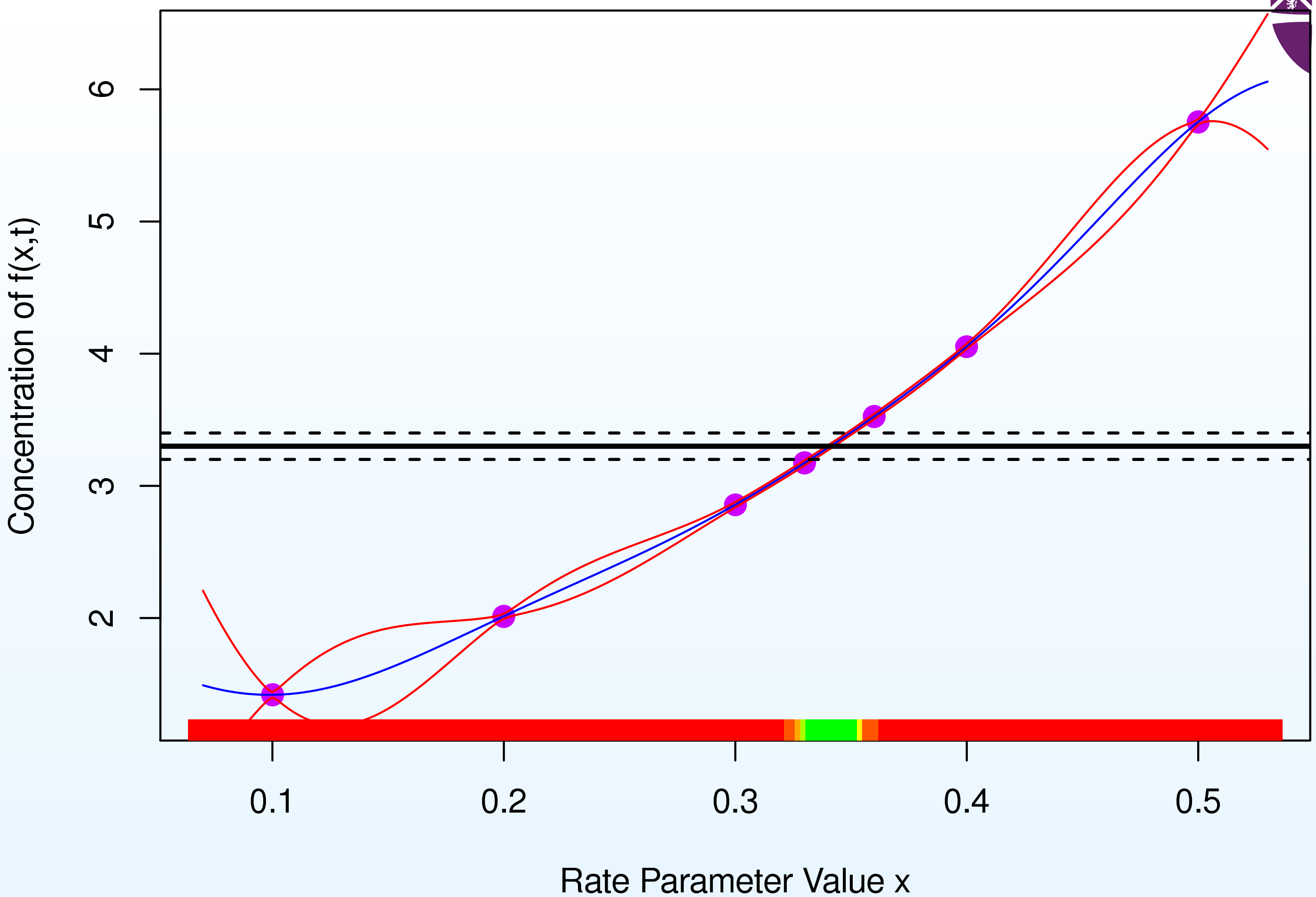


We perform a 2nd iteration or wave of runs to improve emulator accuracy.



We perform a 2nd iteration or wave of runs to improve emulator accuracy.

The runs are located only at non-implausible (green/yellow) points.

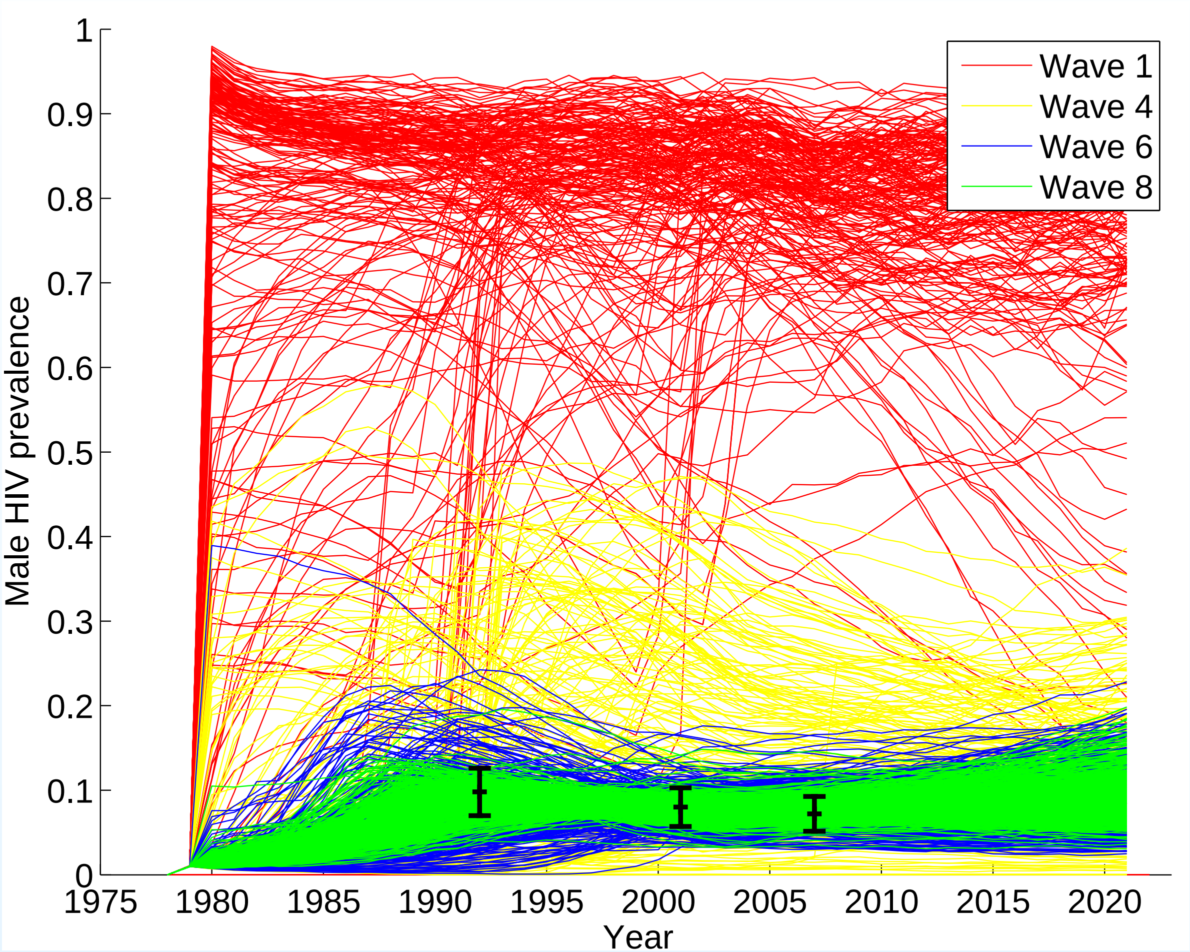


We perform a 2nd iteration or wave of runs to improve emulator accuracy.

The runs are located only at non-implausible (green/yellow) points.

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

History matching for the case study



In 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.

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.

Relating the model and the system

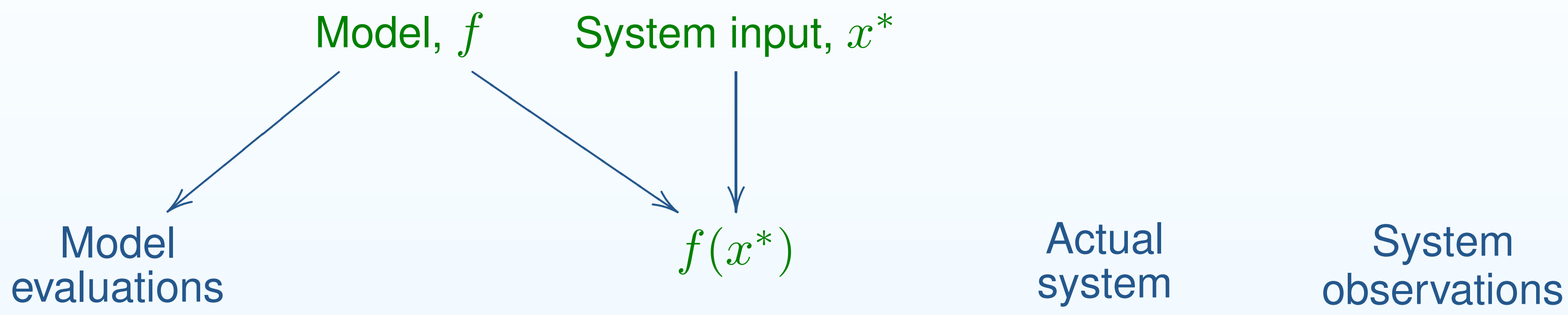
Model evaluations

Actual system

System observations

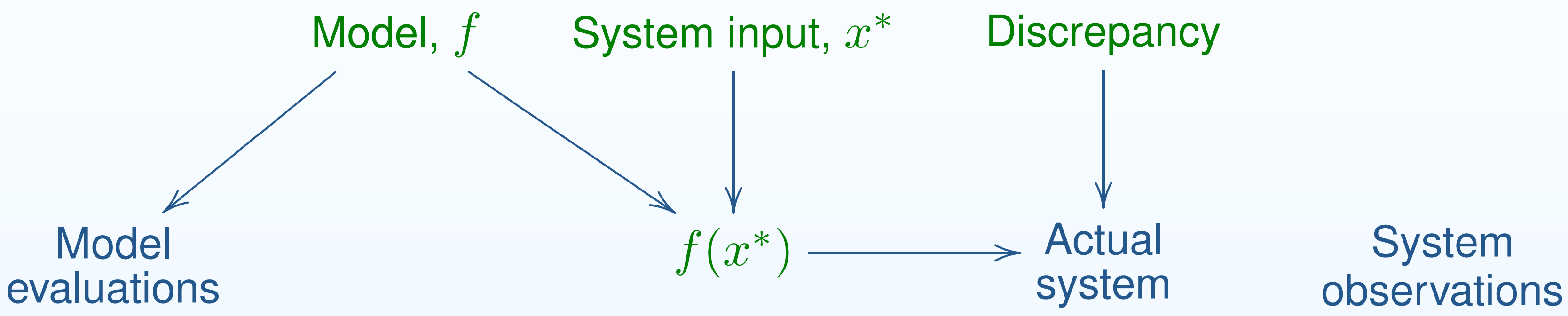
1. We start with a collection of model evaluations, and some observations on actual system
2. We link the model evaluations to the evaluation of the model at the (unknown) system values x^* for the inputs
3. We link the system evaluation to the actual system by adding model discrepancy
4. We incorporate measurement error into the observations

Relating the model and the system



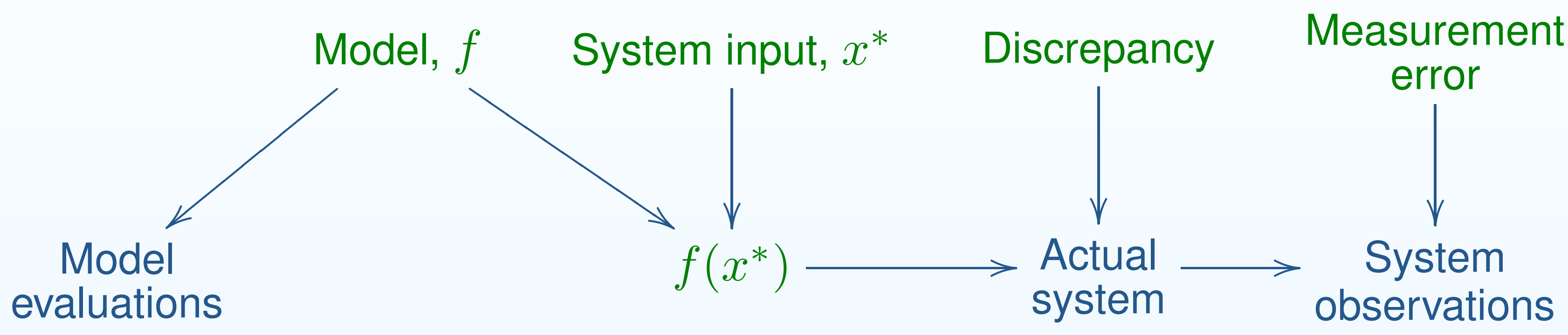
1. We start with a collection of model evaluations, and some observations on actual system
2. We link the model evaluations to the evaluation of the model at the (unknown) system values x^* for the inputs
3. We link the system evaluation to the actual system by adding model discrepancy
4. We incorporate measurement error into the observations

Relating the model and the system



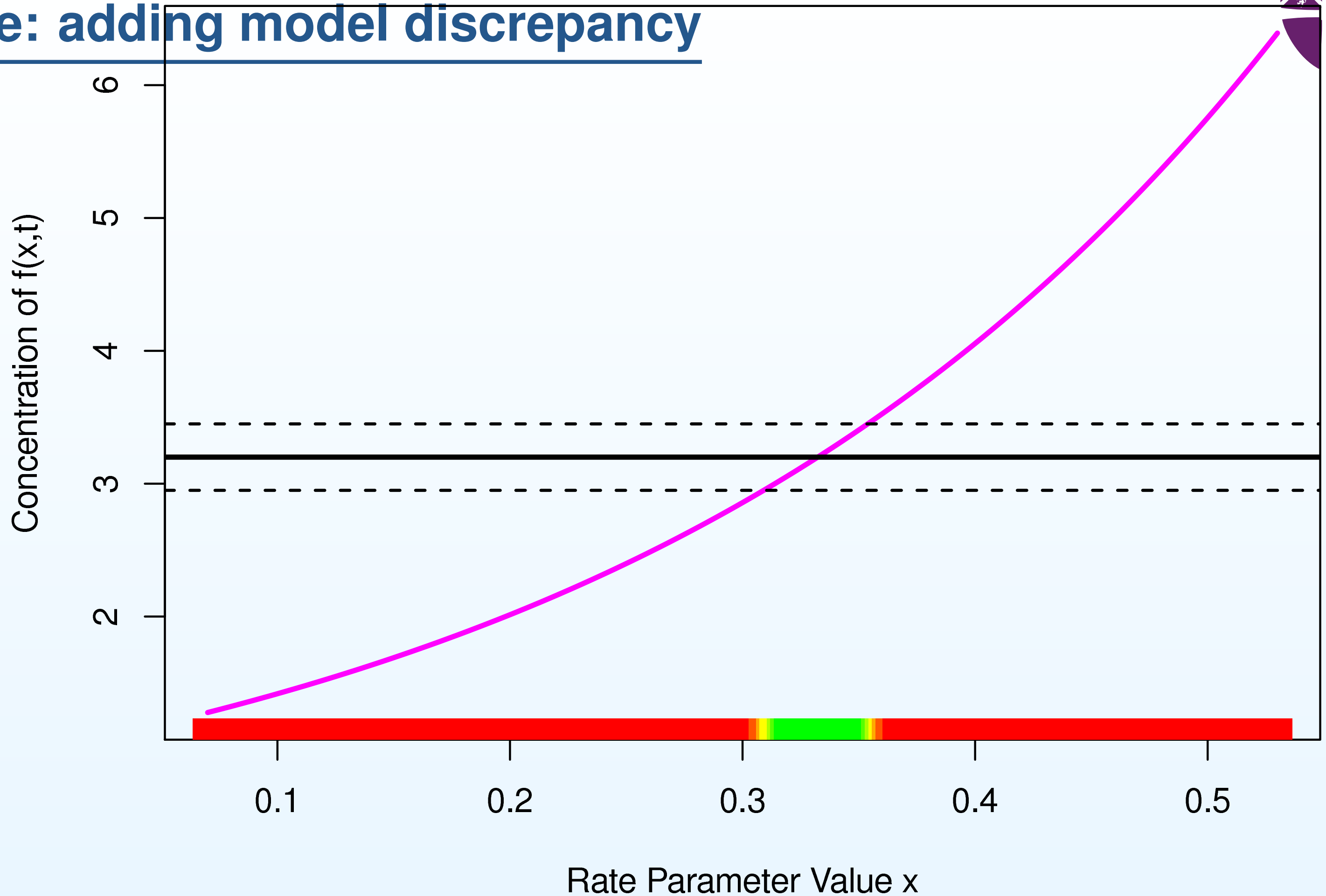
1. We start with a collection of model evaluations, and some observations on actual system
2. We link the model evaluations to the evaluation of the model at the (unknown) system values x^* for the inputs
3. We link the system evaluation to the actual system by adding model discrepancy
4. We incorporate measurement error into the observations

Relating the model and the system

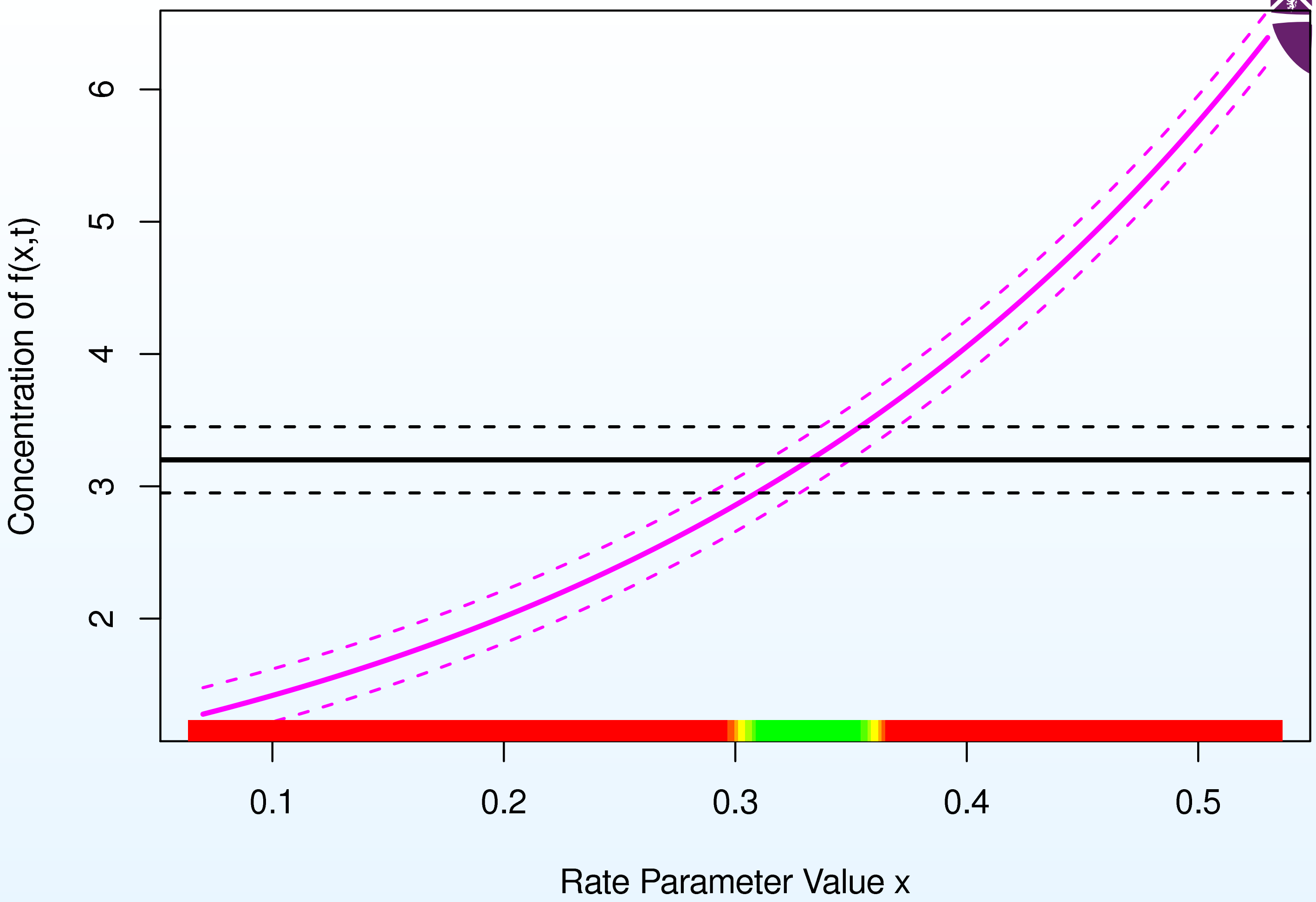


1. We start with a collection of model evaluations, and some observations on actual system
2. We link the model evaluations to the evaluation of the model at the (unknown) system values x^* for the inputs
3. We link the system evaluation to the actual system by adding model discrepancy
4. We incorporate measurement error into the observations

Example: adding model discrepancy



The notion of model discrepancy is related to how accurate we believe the model to be.



Model discrepancy is represented as uncertainty around the model output $f(x)$ itself: here the purple dashed lines.

This results in more uncertainty in x, and hence a larger range of x values.

Internal discrepancy

Structural uncertainty assessessment 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 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.

we may add noise to the forcing functions used to evaluate the simulator

Assessing internal discrepancy

We assess internal discrepancy by

Assessing internal discrepancy

We assess internal discrepancy by

(i) constructing a series of test experiments, for example an ensemble of perturbations to features that we are allowed to vary,

Assessing internal discrepancy

We assess internal discrepancy by

(i) constructing a series of test experiments, for example an ensemble of perturbations to features that we are allowed to vary,

(ii) carrying out detailed computer experiments where we vary the ensemble for some selected choices of input parameters to find the internal discrepancy variance for these input values

Assessing internal discrepancy

We assess internal discrepancy by

(i) constructing a series of test experiments, for example an ensemble of perturbations to features that we are allowed to vary,

(ii) carrying out detailed computer experiments where we vary the ensemble for some selected choices of input parameters to find the internal discrepancy variance for these input values

(iii) emulating internal discrepancy variance across all possible choices of inputs.

Assessing internal discrepancy

We assess internal discrepancy by

(i) constructing a series of test experiments, for example an ensemble of perturbations to features that we are allowed to vary,

(ii) carrying out detailed computer experiments where we vary the ensemble for some selected choices of input parameters to find the internal discrepancy variance for these input values

(iii) emulating internal discrepancy variance across all possible choices of inputs.

Note, in particular, that this method gives an order of magnitude assessment for the correlation between discrepancy values across time.

External discrepancy

(ii) **External discrepancy**

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

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 introduce, say, 10% additional error to account for structural discrepancy.

(This is simple, but much better than ignoring external discrepancy.)

External discrepancy and reification

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 abtract concept as if it was real).

External discrepancy and reification

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 abtract concept as if it was real).

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

External discrepancy and reification

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 abtract concept as if it was real).

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

For example, we can treat our actual simulator as a prior for the reified form.

This is similar to the way in which we use fast simulators to act as priors for slow simulators.

External discrepancy and reification

Durham University

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 abtract concept as if it was real).

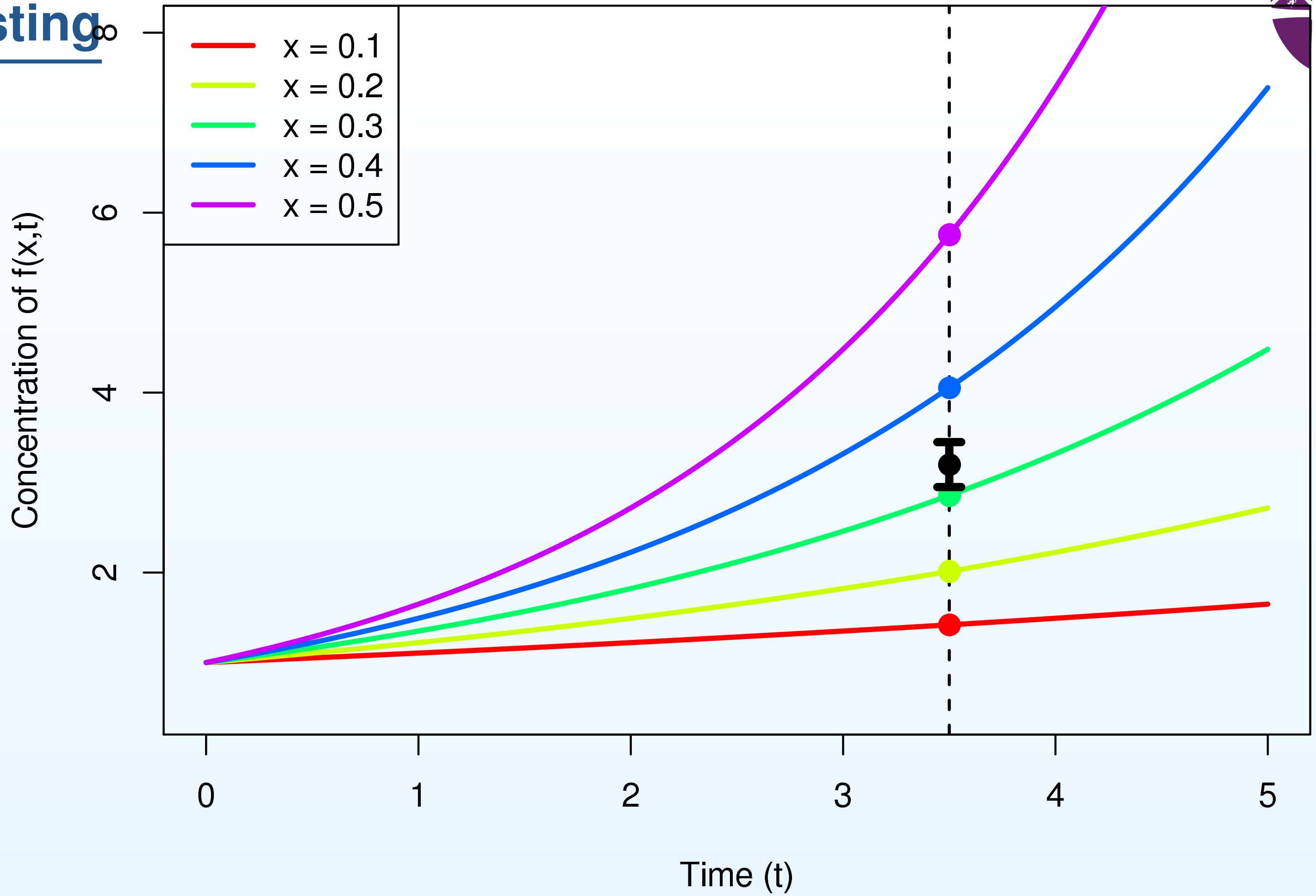
We cannot evaluate the reified simulator, but we can emulate it.

For example, we can treat our actual simulator as a prior for the reified form.

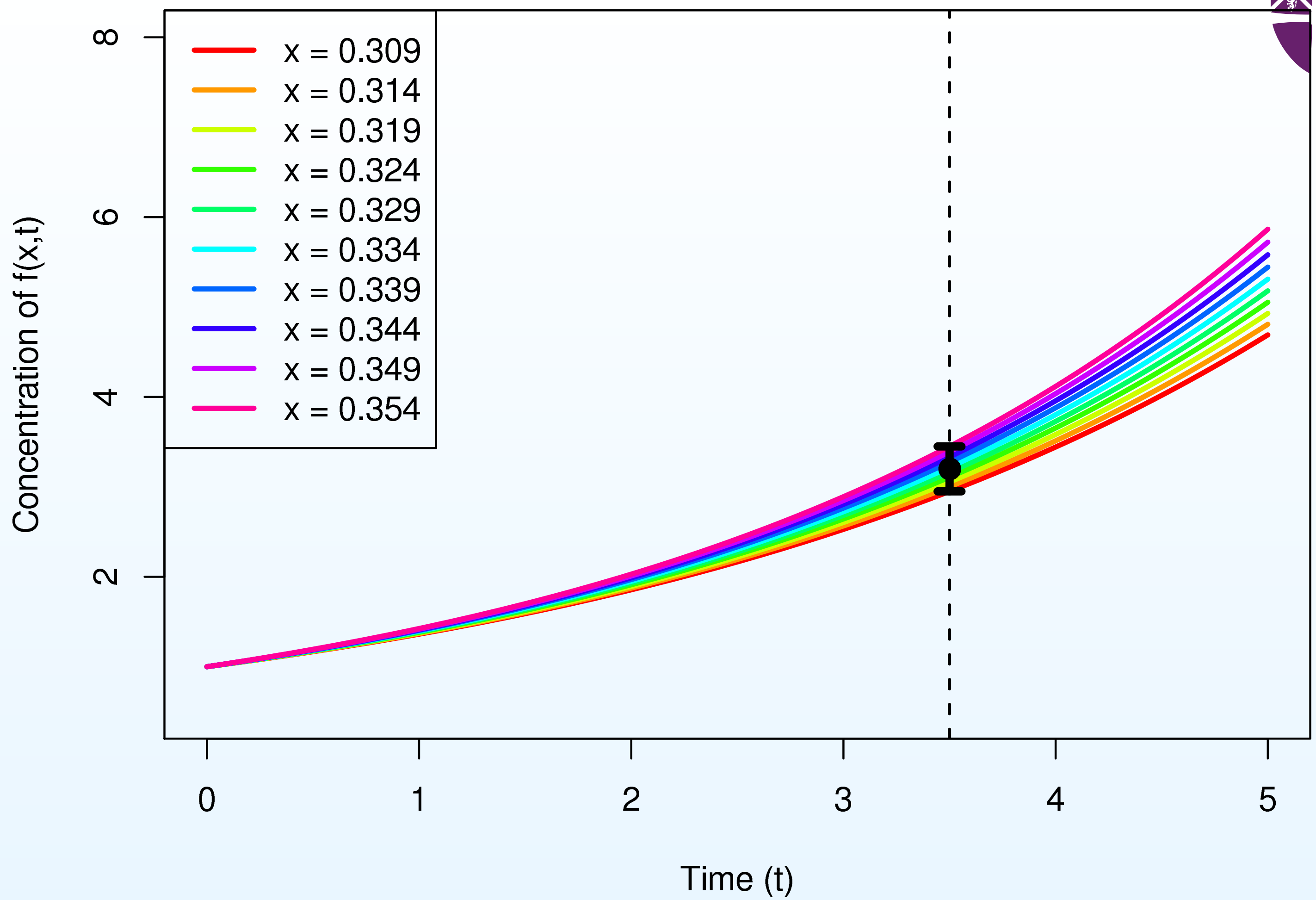
This is similar to the way in which we use fast simulators to act as priors for slow simulators.

So, the methods for history matching based on emulation will work in the same way using the reified emulator.

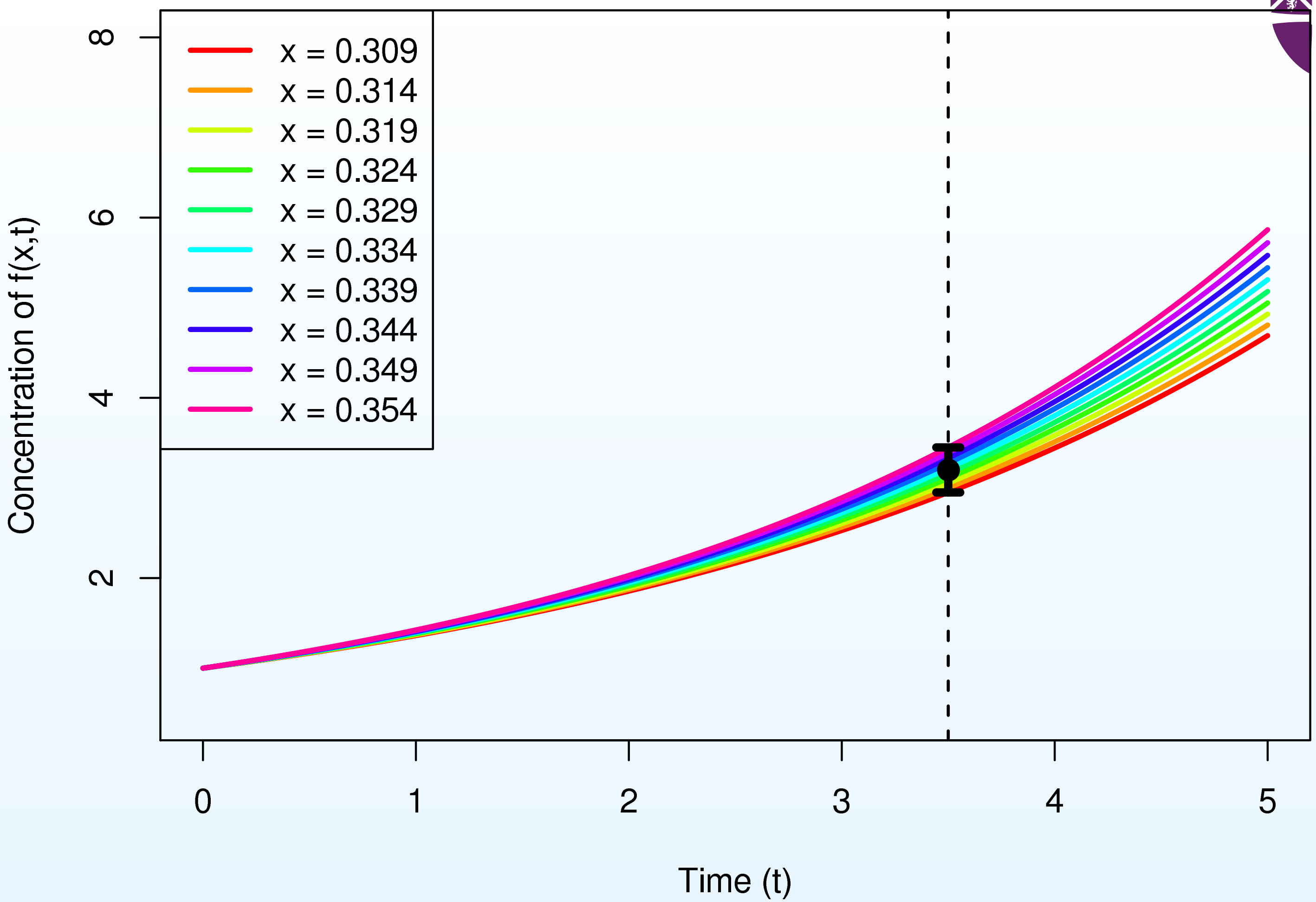
Forecasting



Constraints on x from observations impose constraints on $f(x,t)$ in the future.



We choose values of x consistent with the measurement of $f(x,t)$ at $t = 3.5$, and perform corresponding runs of the simulator, possibly at a variety of control choices. If the simulator is expensive, we may emulate these future outcomes.



These are future projections within the simulator. To transfer these to future projections for the world we need to add the effects of structural discrepancy.

Adding structural discrepancy to forecasts

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

Adding structural discrepancy to forecasts

We represent the physical system as the sum 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 Bayes linear ways to do this.

Adding structural discrepancy to forecasts

We represent the physical system as the sum 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 Bayes linear 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

Durham University

We represent the physical system as the sum 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 Bayes linear 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 sum 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 Bayes linear 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 Bayesian methodology for performing detailed uncertainty analyses for complex physical systems (such as disease transmission), which are modelled by computer simulators.

Concluding comments

There is a general Bayesian methodology for performing detailed uncertainty analyses for complex physical systems (such as disease transmission), which are modelled by computer simulators.

Key features of this methodology are

Concluding comments

Durham University

There is a general Bayesian methodology for performing detailed uncertainty analyses for complex physical systems (such as disease transmission), 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 Bayesian methodology for performing detailed uncertainty analyses for complex physical systems (such as disease transmission), 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) **history matching** to identify all input choices consistent with historical data, and thus all future outcomes consistent with these choices

Concluding comments

There is a general Bayesian methodology for performing detailed uncertainty analyses for complex physical systems (such as disease transmission), 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) **history matching** to identify all input choices consistent with historical data, and thus all future outcomes consistent with these choices

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

References

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

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

I. Vernon, M. Goldstein, R. Bower (2010), Galaxy Formation: a Bayesian Uncertainty Analysis (with discussion) , Bayesian Analysis, 5(4): 619–670.

A. Lawson, M.Goldstein, C.Dent (2017) Bayesian Framework for Power Network Planning Under Uncertainty, Sustainable Energy, Grids and Networks, to appear

M. Goldstein and J.C.Rougier (2009). Reified Bayesian modelling and inference for physical systems (with discussion), JSPI, 139, 1221-1239

M. Goldstein Subjective Bayesian analysis: principles and practice (2006) Bayesian Analysis, 1, 403-420 (and 'Rejoinder to discussion': 465-472)