Multilevel Emulation of Complex Computer Models, with application to EAGLE, a Galaxy Formation Simulation.

Ian Vernon

Department of Mathematical Sciences, Durham University

Bayesian Inference for High-Energy Physics Workshop, IPPP, May 2022

Joint work with: Richard G. Bower, Aaron Ludlow, Alejandro B Llambay, Dept. of Physics, EAGLE team.
With thanks to the VIRGO consortium

Overview

- Overview of Uncertainty Analysis of Complex Models (a.k.a. UQ)
  - Basic setup: expensive computer models
  - Bayesian Emulation.
  - Simple 1D example.
- Strategies for very expensive models
  - Multilevel Emulation
  - Application to EAGLE: a hydrodynamic galaxy formation simulation.

- The Bayesian uncertainty analysis of complex systems has led to the development of a large set of extremely general and powerful techniques.
- These have now been employed in a range of scientific disciplines:
  - Cosmology (galaxy formation simulations),
  - Climate science (climate models of global warming),
  - Environmental sciences (flood and rainfall runoff models),
  - Oil industry (oil reservoir models and geology models).
  - Energy Networks (energy/power system models, national grid).
  - Systems biology (genetic and metabolic network models),
  - Epidemiology (agent based stochastic HIV, TB, Covid-19 models).
  - Cardiac modelling (high resolution, multilevel cardiac models).
  - Nuclear Physics (ab initio, quantum many-body models of nuclei).
  - Many more...
- These techniques could be of substantial use to the High Energy Physics community.

Overview of Uncertainty Analysis of Complex Models

- The Bayesian uncertainty analysis of complex systems has led to the development of a large set of extremely general and powerful techniques.
- These have now been employed in a range of scientific disciplines:
  - Cosmology (galaxy formation simulations),
  - Climate science (climate models of global warming),
  - Environmental sciences (flood and rainfall runoff models),
  - Oil industry (oil reservoir models and geology models).
  - Energy Networks (energy/power system models, national grid).
  - Systems biology (genetic and metabolic network models),
  - Epidemiology (agent based stochastic HIV, TB, Covid-19 models).
  - Cardiac modelling (high resolution, multilevel cardiac models).
  - Nuclear Physics (ab initio, quantum many-body models of nuclei).
  - Many more...
- These techniques could be of substantial use to the High Energy Physics community.

Overview of Uncertainty Analysis of Complex Models

- The Bayesian uncertainty analysis of complex systems has led to the development of a large set of extremely general and powerful techniques.
- These have now been employed in a range of scientific disciplines:
  - Cosmology (galaxy formation simulations),
  - Climate science (climate models of global warming),
  - Environmental sciences (flood and rainfall runoff models),
  - Oil industry (oil reservoir models and geology models).
  - Energy Networks (energy/power system models, national grid).
  - Systems biology (genetic and metabolic network models),
  - Epidemiology (agent based stochastic HIV, TB, Covid-19 models).
  - Cardiac modelling (high resolution, multilevel cardiac models).
  - Nuclear Physics (ab initio, quantum many-body models of nuclei).
  - Many more...
- These techniques could be of substantial use to the High Energy Physics community.

Overview of Uncertainty Analysis of Complex Models

- The Bayesian uncertainty analysis of complex systems has led to the development of a large set of extremely general and powerful techniques.
- These have now been employed in a range of scientific disciplines:
  - Cosmology (galaxy formation simulations),
  - Climate science (climate models of global warming),
  - Environmental sciences (flood and rainfall runoff models),
  - Oil industry (oil reservoir models and geology models).
  - Energy Networks (energy/power system models, national grid).
  - Systems biology (genetic and metabolic network models),
  - Epidemiology (agent based stochastic HIV, TB, Covid-19 models).
  - Cardiac modelling (high resolution, multilevel cardiac models).
  - Nuclear Physics (ab initio, quantum many-body models of nuclei).
  - Many more...
- These techniques could be of substantial use to the High Energy Physics community.

- The Bayesian uncertainty analysis of complex physical systems, is now sometimes referred to as “Uncertainty Quantification".
- Focus on the following general scenario:
  - We have a physical model $f(x)$: a model based on theory, implemented on a computer, that may take a long time to evaluate.
  - The model takes a vector of input parameters x and returns a vector $f(x)$ of outputs.
  - We want to compare the vector of model outputs f, or a subset of them, with the vector of observed data z, observations of the real system y.
- Raises (at least) two major questions.

Overview of Uncertainty Analysis of Complex Models

- The Bayesian uncertainty analysis of complex physical systems, is now sometimes referred to as "Uncertainty Quantification".
- Focus on the following general scenario:
  - We have a physical model $f(x)$: a model based on theory, implemented on a computer, that may take a long time to evaluate.
  - The model takes a vector of input parameters x and returns a vector $f(x)$ of outputs.
  - We want to compare the vector of model outputs f, or a subset of them, with the vector of observed data z, observations of the real system y.
- Raises (at least) two major questions.

- The Bayesian uncertainty analysis of complex physical systems, is now sometimes referred to as “Uncertainty Quantification".
- Focus on the following general scenario:
    - We have a physical model $f(x)$: a model based on theory, implemented on a computer, that may take a long time to evaluate.
    - The model takes a vector of input parameters x and returns a vector $f(x)$ of outputs.
    - We want to compare the vector of model outputs f, or a subset of them, with the vector of observed data z, observations of the real system y.
- Raises (at least) two major questions.

- The Bayesian uncertainty analysis of complex physical systems, is now sometimes referred to as “Uncertainty Quantification".

- Focus on the following general scenario:
  - We have a physical model $f(x)$: a model based on theory, implemented on a computer, that may take a long time to evaluate.
  - The model takes a vector of input parameters x and returns a vector $f(x)$ of outputs.
  - We want to compare the vector of model outputs f, or a subset of them, with the vector of observed data z, observations of the real system y.

- Raises (at least) two major questions.

- The Bayesian uncertainty analysis of complex physical systems, is now sometimes referred to as "Uncertainty Quantification".
- Focus on the following general scenario:
  - We have a physical model $f(x)$: a model based on theory, implemented on a computer, that may take a long time to evaluate.
  - The model takes a vector of input parameters x and returns a vector $f(x)$ of outputs.
  - We want to compare the vector of model outputs f, or a subset of them, with the vector of observed data z, observations of the real system y.
- Raises (at least) two major questions.

Overview of Uncertainty Analysis of Complex Models

- The Bayesian uncertainty analysis of complex physical systems, is now sometimes referred to as “Uncertainty Quantification".
- Focus on the following general scenario:
  - We have a physical model $f(x)$: a model based on theory, implemented on a computer, that may take a long time to evaluate.
  - The model takes a vector of input parameters x and returns a vector $f(x)$ of outputs.
  - We want to compare the vector of model outputs f, or a subset of them, with the vector of observed data z, observations of the real system y.
- Raises (at least) two major questions.

- First major question: Is the model currently consistent with the observed measurements? To answer this we require:
  - Bayes Linear Emulation of the model (to combat speed of $f(x)$ problem: emulators mimic the model but are several orders of magnitude faster)
  - Implausibility Measures (using observed errors and model discrepancy)
  - A global parameter search known as iterative history matching.
- We will hence identify the set of all input parameters that produced model outputs consistent with known measurements.
- Not just searching for a single best match.
- Can then generate Bayesian posteriors if desired.

- First major question: Is the model currently consistent with the observed measurements? To answer this we require:
  - Bayes Linear Emulation of the model (to combat speed of $f(x)$ problem: emulators mimic the model but are several orders of magnitude faster)
  - Implausibility Measures (using observed errors and model discrepancy)
  - A global parameter search known as iterative history matching.
- We will hence identify the set of all input parameters that produced model outputs consistent with known measurements.
- Not just searching for a single best match.
- Can then generate Bayesian posteriors if desired.

- First major question: Is the model currently consistent with the observed measurements? To answer this we require:
  - Bayes Linear Emulation of the model (to combat speed of $f(x)$ problem: emulators mimic the model but are several orders of magnitude faster)
  - Implausibility Measures (using observed errors and model discrepancy)
  - A global parameter search known as iterative history matching.
- We will hence identify the set of all input parameters that produced model outputs consistent with known measurements.
- Not just searching for a single best match.
- Can then generate Bayesian posteriors if desired.

- First major question: Is the model currently consistent with the observed measurements? To answer this we require:
  - Bayes Linear Emulation of the model (to combat speed of $f(x)$ problem: emulators mimic the model but are several orders of magnitude faster)
  - Implausibility Measures (using observed errors and model discrepancy)
  - A global parameter search known as iterative history matching.
- We will hence identify the set of all input parameters that produced model outputs consistent with known measurements.
- Not just searching for a single best match.
- Can then generate Bayesian posteriors if desired.

- First major question: Is the model currently consistent with the observed measurements? To answer this we require:
  - Bayes Linear Emulation of the model (to combat speed of $f(x)$ problem: emulators mimic the model but are several orders of magnitude faster)
  - Implausibility Measures (using observed errors and model discrepancy)
  - A global parameter search known as iterative history matching.
- We will hence identify the set of all input parameters that produced model outputs consistent with known measurements.
- Not just searching for a single best match.
- Can then generate Bayesian posteriors if desired.

Overview of Uncertainty Analysis of Complex Models

- First major question: Is the model currently consistent with the observed measurements? To answer this we require:
  - Bayes Linear Emulation of the model (to combat speed of $f(x)$ problem: emulators mimic the model but are several orders of magnitude faster)
  - Implausibility Measures (using observed errors and model discrepancy)
  - A global parameter search known as iterative history matching.
- We will hence identify the set of all input parameters that produced model outputs consistent with known measurements.
- Not just searching for a single best match.
- Can then generate Bayesian posteriors if desired.

- Second major question: What is the most informative future experiment we can perform to learn more about the system?
- To answer this we need to:
  - Specify the class of possible experiments considered.
  - Use the results of the Global parameter search to obtain model predictions for all future experiments that are consistent with current observations.
  - Choose the most efficient experiment based on an Expected Space Reduction criteria and complementary robustness considerations.
- This will result in a design for a new experiment that is expected to be highly informative about the input parameters x of the system (or indeed of any scientific criteria that you care about). Or used to compare two models.

Simple 1D Exponential Growth Example

- Say we are interested in the concentration of a chemical which evolves in time. We will 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 will temporarily assume the initial conditions are $f_0 = f(x, t=0) = 1$.
- The system runs from $t = 0$ to $t = 5$ and we will measure $f(x,t)$ with error at $t = 3.5$.
- Model features an input parameter x which we want to learn about.
- Note that normally we would not have the analytic solution for $f(x,t)$.

Simple 1D Exponential Growth Example

- Say we are interested in the concentration of a chemical which evolves in time. We will 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 will temporarily assume the initial conditions are $f_0 = f(x, t=0) = 1$.
- The system runs from $t = 0$ to $t = 5$ and we will measure $f(x,t)$ with error at $t = 3.5$.
- Model features an input parameter x which we want to learn about.
- Note that normally we would not have the analytic solution for $f(x,t)$.

Simple 1D Exponential Growth Example

- Say we are interested in the concentration of a chemical which evolves in time. We will 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 will temporarily assume the initial conditions are $f_0 = f(x, t=0) = 1$.

- The system runs from $t=0$ to $t=5$ and we will measure $f(x,t)$ with error at $t=3.5$.

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

- Note that normally we would not have the analytic solution for $f(x,t)$.

Simple 1D Exponential Growth Example

- Say we are interested in the concentration of a chemical which evolves in time. We will 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:

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

- We will temporarily assume the initial conditions are $f_0 = f(x, t=0) = 1$.
- The system runs from $t = 0$ to $t = 5$ and we will measure $f(x,t)$ with error at $t = 3.5$.
- Model features an input parameter x which we want to learn about.
- Note that normally we would not have the analytic solution for $f(x,t)$.

Simple 1D Exponential Growth Example

- Say we are interested in the concentration of a chemical which evolves in time. We will 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:

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

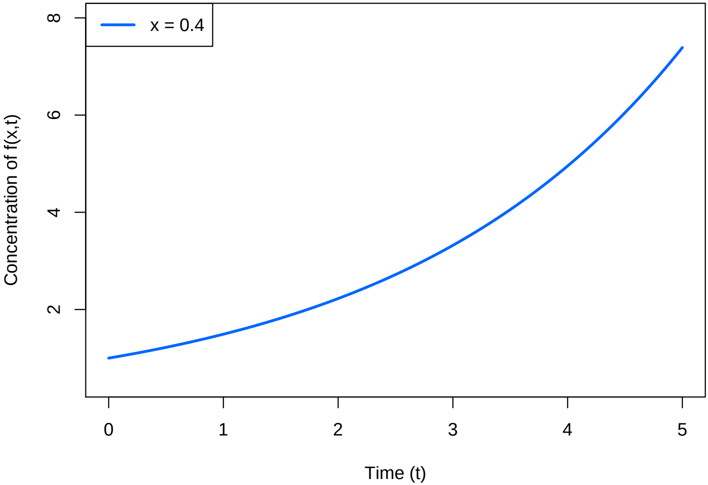
- We will temporarily assume the initial conditions are $f_0 = f(x, t=0) = 1$.
- The system runs from $t = 0$ to $t = 5$ and we will measure $f(x,t)$ with error at $t = 3.5$.
- Model features an input parameter x which we want to learn about.
- Note that normally we would not have the analytic solution for $f(x,t)$.

Simple 1D Exponential Growth Example

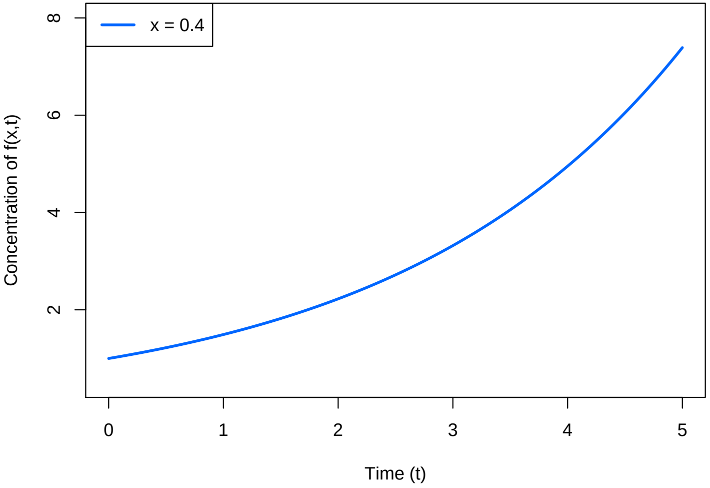
- Say we are interested in the concentration of a chemical which evolves in time. We will model this concentration as $f(x,t)$ where x is a rate parameter and t is time.
- We think $f(x,t)$ satisfies the differential equation or model:

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

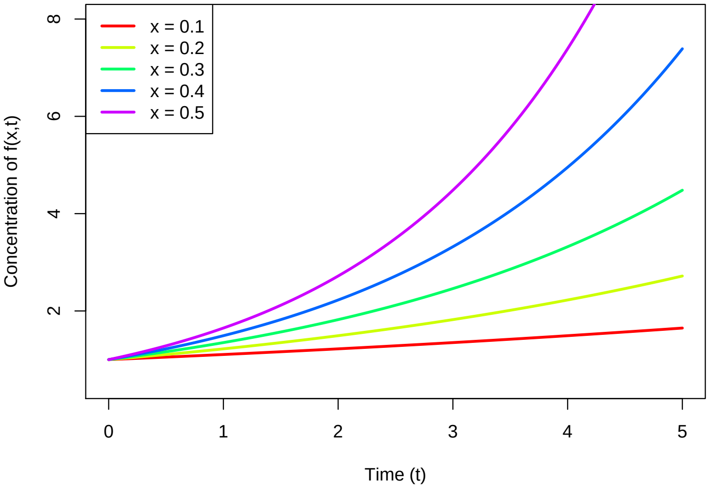
- We will temporarily assume the initial conditions are $f_0 = f(x, t = 0) = 1$.
- The system runs from $t = 0$ to $t = 5$ and we will measure $f(x,t)$ with error at $t = 3.5$.
- Model features an input parameter x which we want to learn about.
- Note that normally we would not have the analytic solution for $f(x,t)$.



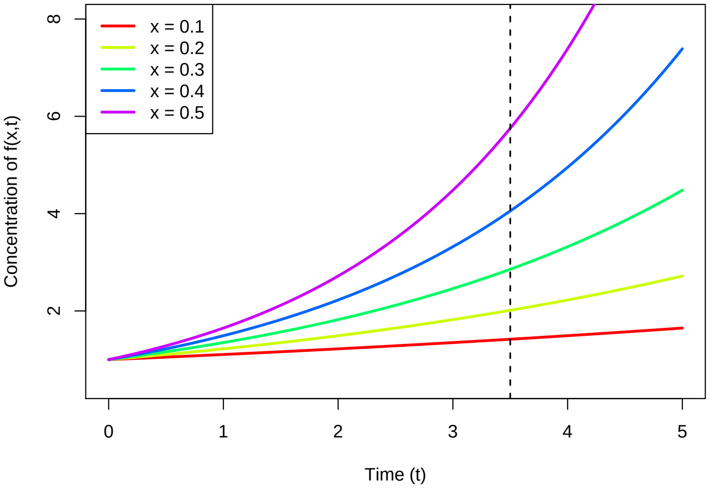
- One “model run" with the input parameter $x = 0.4$
- If we did not know the analytic solution for $f(x,t)$ this would be generated by numerically solving the differential equation.



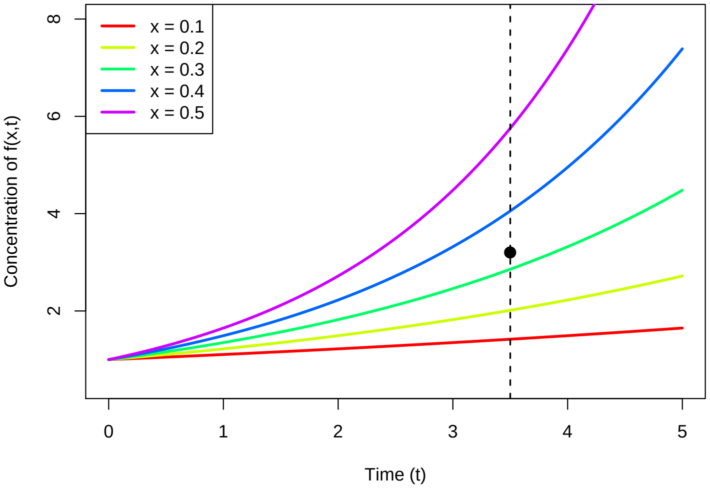
- One “model run" with the input parameter $x = 0.4$
- If we did not know the analytic solution for $f(x, t)$ this would be generated by numerically solving the differential equation.



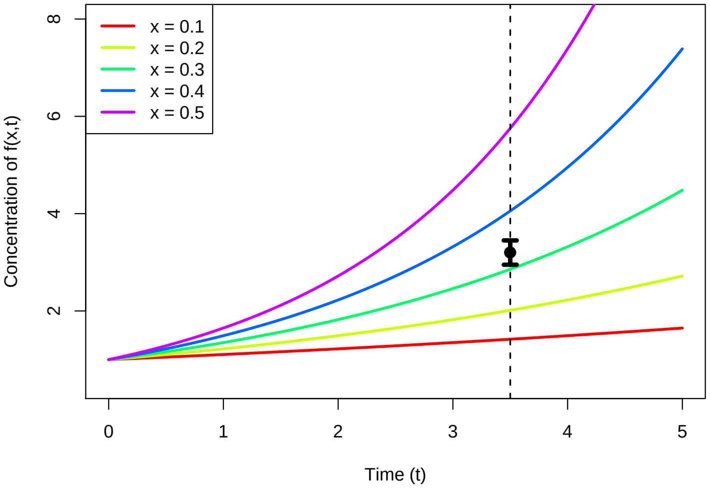
- Five model runs with the input parameter varying from $x = 0.1$ to $x = 0.5$
- We are going to measure $f(x,t)$ at $t = 3.5$
- The measurement is not a point but comes with measurement error.



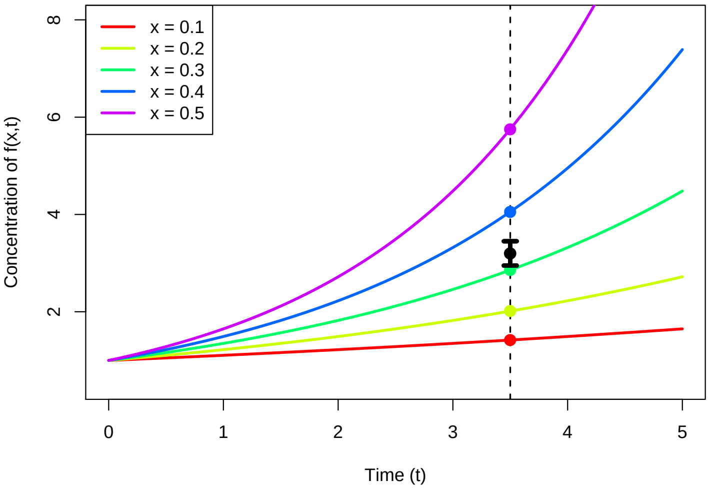
- Five model runs with the input parameter varying from $x = 0.1$ to $x = 0.5$
- We are going to measure $f(x,t)$ at $t = 3.5$
- The measurement is not a point but comes with measurement error.



- 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 $t = 3.5$, giving the value z
- The measurement is not a point but comes with measurement error.

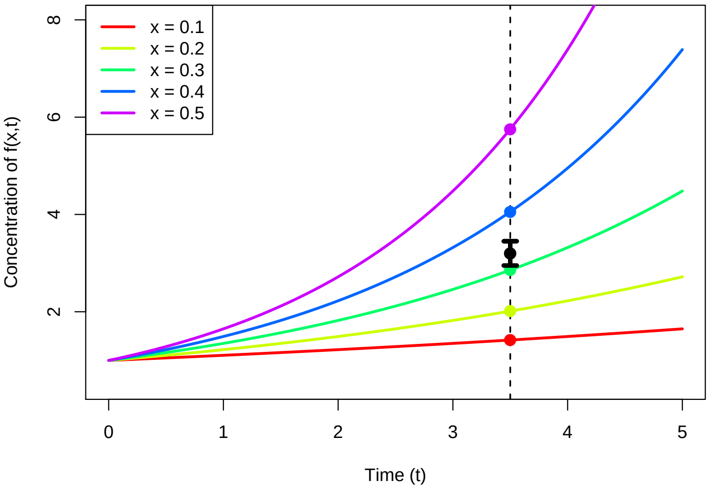


- Five model runs with the input parameter varying from $x = 0.1$ to $x = 0.5$
- We are going to measure $f(x, t)$ at $t = 3.5$, giving the value z
- The measurement is not a point but comes with measurement error.

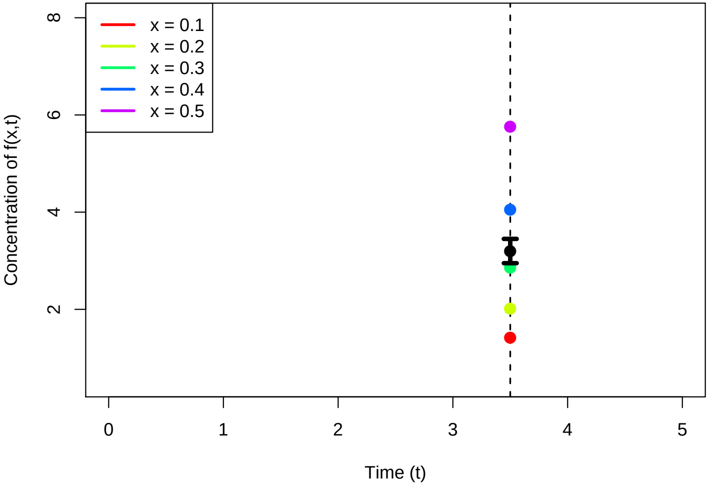


- Major question: which values of x ensure the output $f(x, t = 3.5)$ is consistent with the observations?

- It would seem that x has to be at least between 0.3 and 0.4.

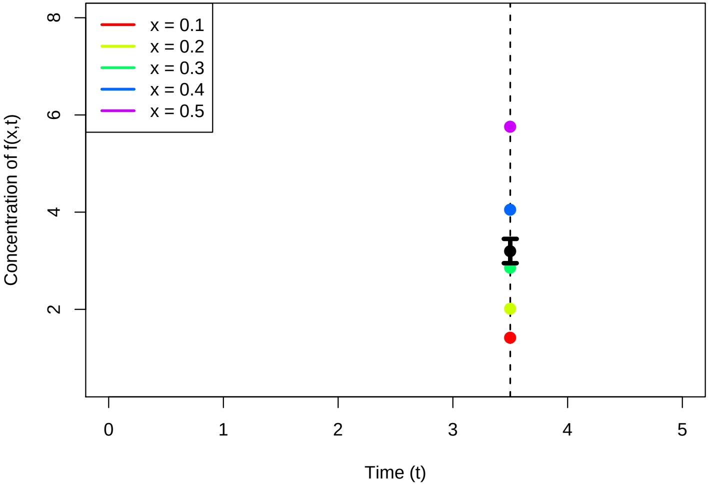


- Major question: which values of x ensure the output $f(x, t = 3.5)$ is consistent with the observations?
- It would seem that x has to be at least between 0.3 and 0.4.



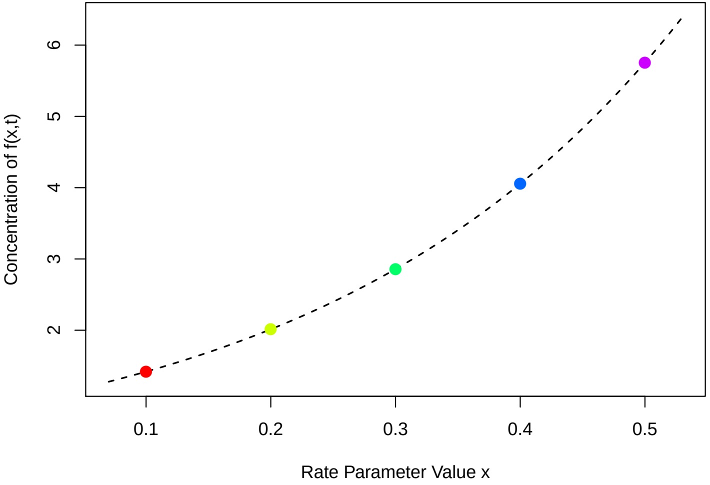
- To answer this, we can now discard other values of $f(x,t)$ and think of $f(x,t=3.5)$ as a function of x only.

- That is take $f(x) \equiv f(x,t=3.5)$

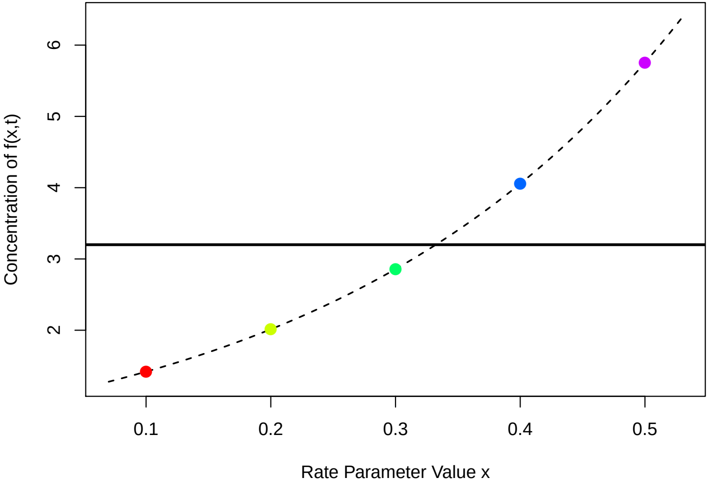


- To answer this, we can now discard other values of $f(x,t)$ and think of $f(x,t=3.5)$ as a function of x only.

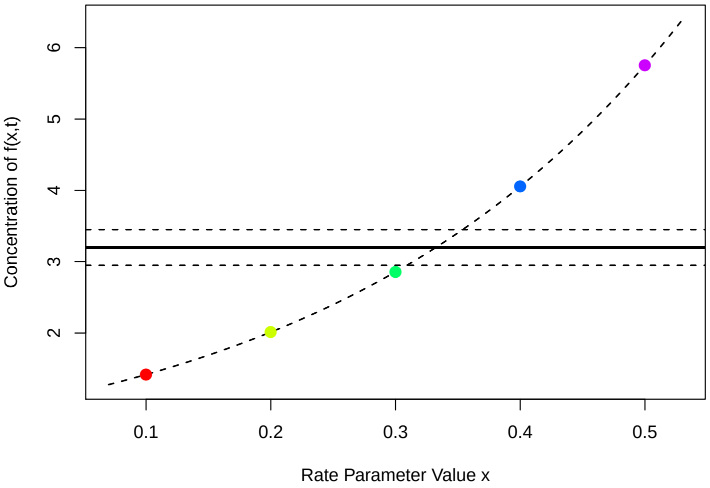
- That is take $f(x) \equiv f(x,t=3.5)$



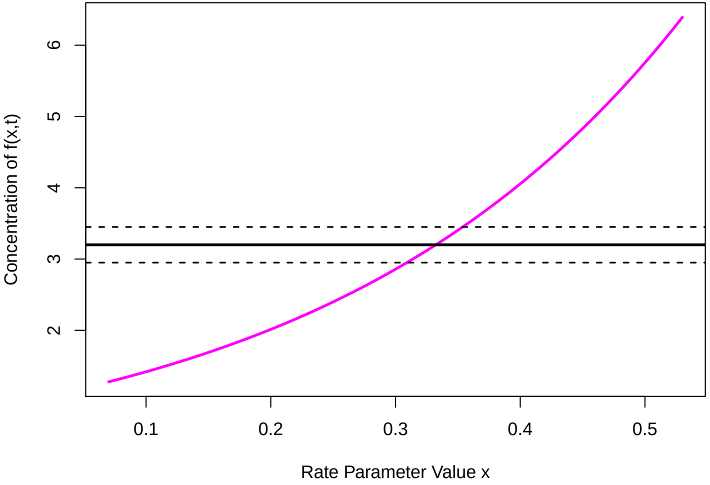
- We can now plot the concentration $f(x)$ as a function of the input parameter x.
- Black horizontal line: the observed measurement of f
- Dashed horizontal lines: the measurement errors



- We can now 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

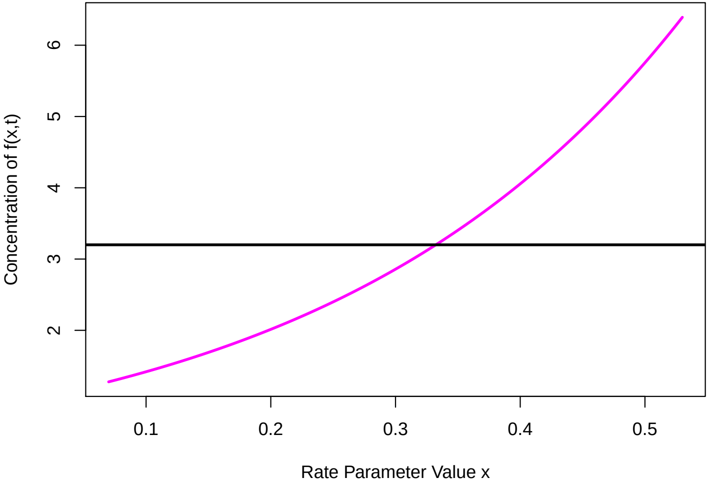


- We can now plot the concentration $f(x)$ as a function of the input parameter x.
- Black horizontal line: the observed measurement of f
- Dashed horizontal lines: the measurement errors



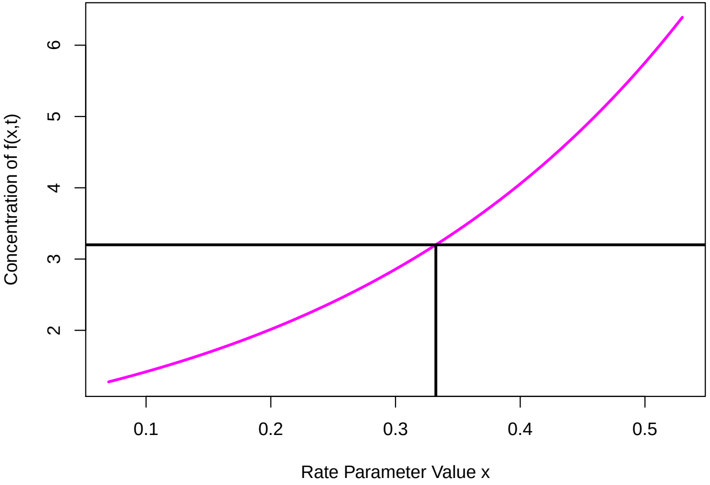
- If we know the analytical expression for $f(x) = \exp(3.5x)$, then we can identify the values of x of interest.

- Ignoring the measurement error would lead to a single value for x but this is incorrect: we have to include the errors.

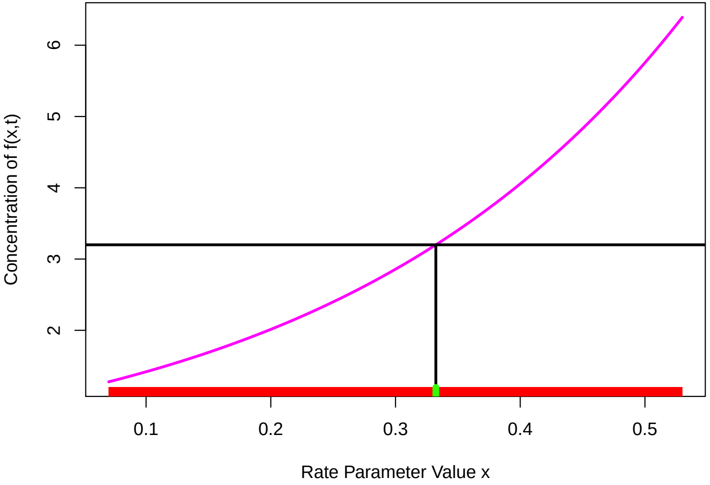


- If we know the analytical expression for $f(x) = \exp(3.5x)$, then we can identify the values of x of interest.

- Ignoring the measurement error would lead to a single value for x but this is incorrect: we have to include the errors.

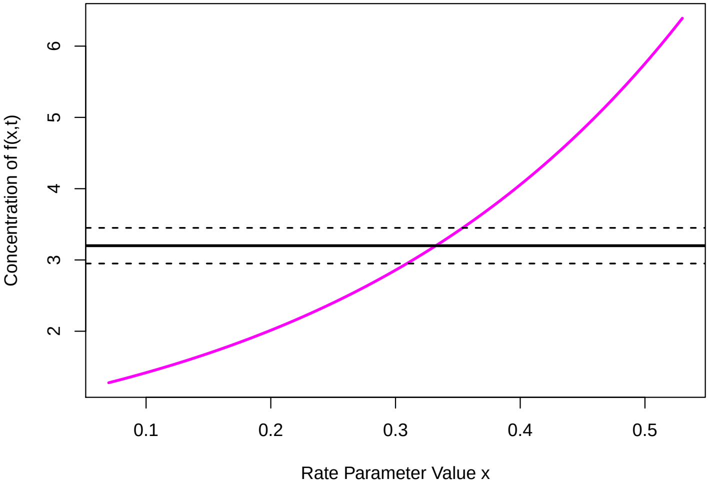


- If we know the analytical expression for $f(x) = \exp(3.5x)$, then we can identify the values of x of interest.
- Ignoring the measurement error would lead to a single value for x but this is incorrect: we have to include the errors.

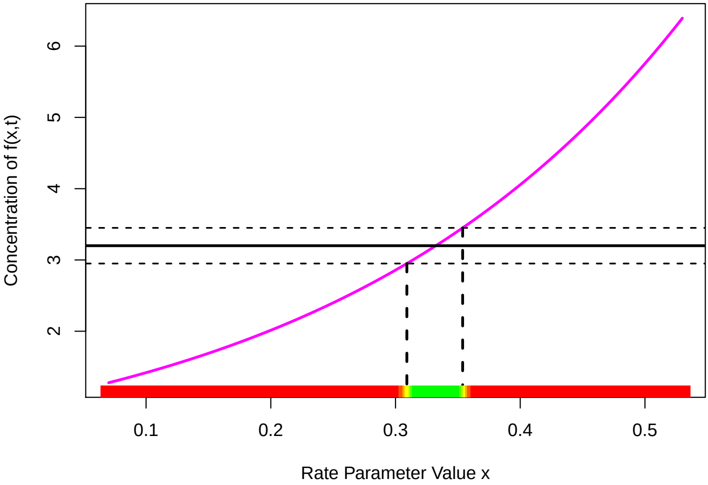


- If we know the analytical expression for $f(x) = \exp(3.5x)$, then we can identify the values of x of interest.

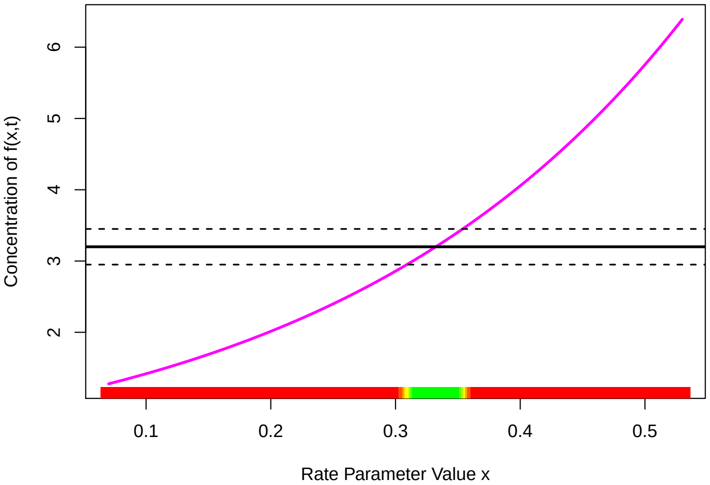
- Ignoring the measurement error would lead to a single value for x but this is incorrect: we have to include the errors.



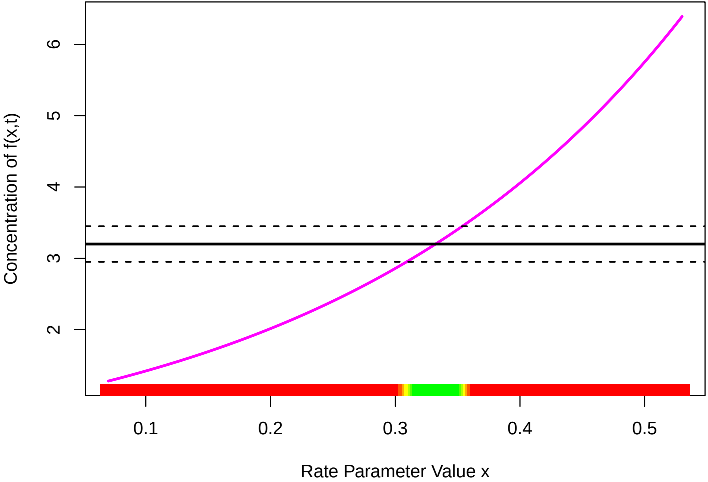
- Uncertainty in the measurement of $f(x, t = 3.5)$ leads to uncertainty in the inferred values of x.
- Hence we see a range (green/yellow) of possible values of x consistent with the measurements, with all the implausible values of x in red.



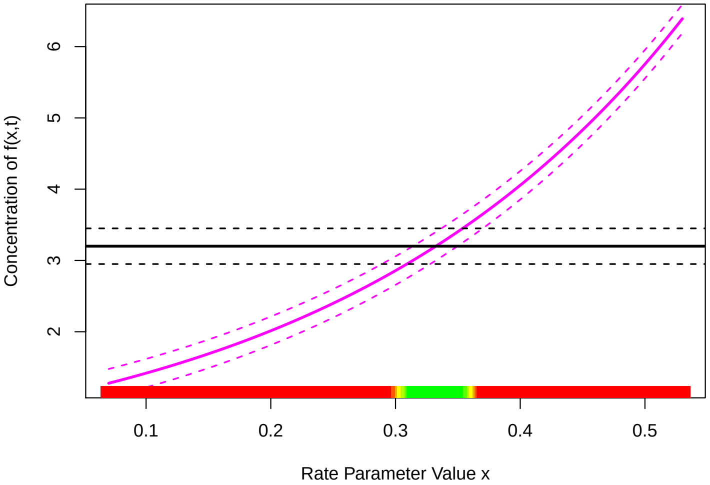
- Uncertainty in the measurement of $f(x, t = 3.5)$ leads to uncertainty in the inferred values of x.
- Hence we see a range (green/yellow) of possible values of x consistent with the measurements, with all the implausible values of x in red.



- Another important form of uncertainty is that of model discrepancy related to how accurate we believe the model to be.
- This uncertainty arises from many issues e.g. is the form of the model (the differential equation) appropriate, is the model a simplified description of a more complex system, is there uncertainty in the initial conditions etc?

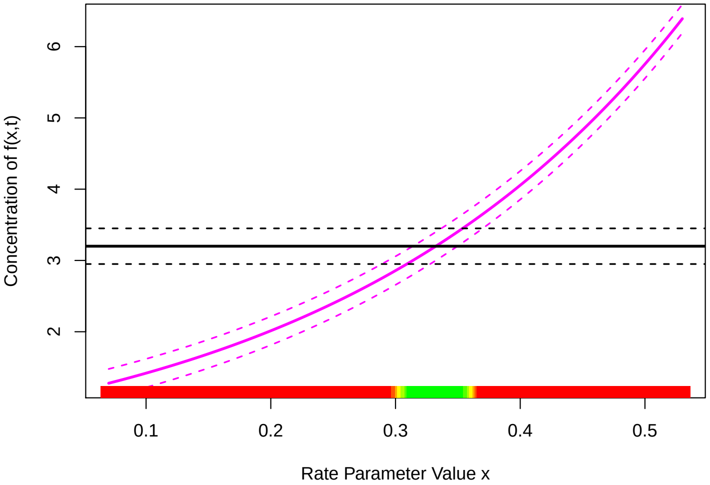


- Another important form of uncertainty is that of model discrepancy related to how accurate we believe the model to be.
- This uncertainty arises from many issues e.g. is the form of the model (the differential equation) appropriate, is the model a simplified description of a more complex system, is there uncertainty in the initial conditions etc?



- 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.



- 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.

Emulating the Model: Simple 1D Example

- For more realistic models we do not know the full analytic solution for $f(x)$.
- Instead for fixed values of x we would solve for $f(x)$ numerically: this can be too slow even for relatively fast models.
- For e.g. galaxy formation models, it takes between 1 day to 1.5 months to solve for $f(x)$ for a single x. Our techniques are designed to cope with this.
- If x was of moderate dimension e.g. there were 30 input parameters, then we need a vast number of evaluations of the model to fill this 30 dimensional space: e.g. corners only $2^{30} = 1$ billion evaluations.
- A Bayes Linear (or related GP) emulator is a statistical construct that mimics the model, but which is extremely fast to evaluate, often several orders of magnitude faster than the model: use the emulator to learn about x.
- In our galaxy formation model EAGLE the emulators are $10^9 - 10^{12}$ times faster than the model, depending which version of EAGLE we compare to.

Emulating the Model: Simple 1D Example

- For more realistic models we do not know the full analytic solution for $f(x)$.
- Instead for fixed values of x we would solve for $f(x)$ numerically: this can be too slow even for relatively fast models.
- For e.g. galaxy formation models, it takes between 1 day to 1.5 months to solve for $f(x)$ for a single x. Our techniques are designed to cope with this.
- If x was of moderate dimension e.g. there were 30 input parameters, then we need a vast number of evaluations of the model to fill this 30 dimensional space: e.g. corners only $2^{30} = 1$ billion evaluations.
- A Bayes Linear (or related GP) emulator is a statistical construct that mimics the model, but which is extremely fast to evaluate, often several orders of magnitude faster than the model: use the emulator to learn about x.
- In our galaxy formation model EAGLE the emulators are $10^9 - 10^{12}$ times faster than the model, depending which version of EAGLE we compare to.

Emulating the Model: Simple 1D Example

- For more realistic models we do not know the full analytic solution for $f(x)$.
- Instead for fixed values of x we would solve for $f(x)$ numerically: this can be too slow even for relatively fast models.
- For e.g. galaxy formation models, it takes between 1 day to 1.5 months to solve for $f(x)$ for a single x. Our techniques are designed to cope with this.
- If x was of moderate dimension e.g. there were 30 input parameters, then we need a vast number of evaluations of the model to fill this 30 dimensional space: e.g. corners only $2^{30} = 1$ billion evaluations.
- A Bayes Linear (or related GP) emulator is a statistical construct that mimics the model, but which is extremely fast to evaluate, often several orders of magnitude faster than the model: use the emulator to learn about x.
- In our galaxy formation model EAGLE the emulators are $10^9 - 10^{12}$ times faster than the model, depending which version of EAGLE we compare to.

Emulating the Model: Simple 1D Example

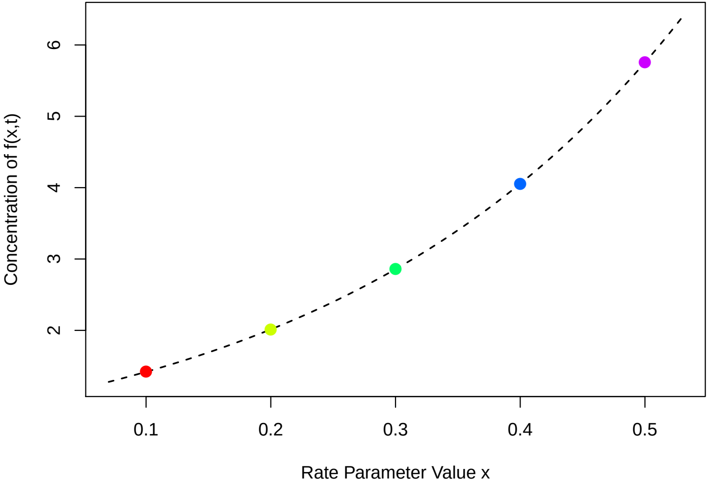
- For more realistic models we do not know the full analytic solution for $f(x)$.
- Instead for fixed values of x we would solve for $f(x)$ numerically: this can be too slow even for relatively fast models.
- For e.g. galaxy formation models, it takes between 1 day to 1.5 months to solve for $f(x)$ for a single x. Our techniques are designed to cope with this.
- If x was of moderate dimension e.g. there were 30 input parameters, then we need a vast number of evaluations of the model to fill this 30 dimensional space: e.g. corners only $2^{30} = 1$ billion evaluations.
- A Bayes Linear (or related GP) emulator is a statistical construct that mimics the model, but which is extremely fast to evaluate, often several orders of magnitude faster than the model: use the emulator to learn about x.
- In our galaxy formation model EAGLE the emulators are $10^9 - 10^{12}$ times faster than the model, depending which version of EAGLE we compare to.

Emulating the Model: Simple 1D Example

- For more realistic models we do not know the full analytic solution for $f(x)$.
- Instead for fixed values of x we would solve for $f(x)$ numerically: this can be too slow even for relatively fast models.
- For e.g. galaxy formation models, it takes between 1 day to 1.5 months to solve for $f(x)$ for a single x. Our techniques are designed to cope with this.
- If x was of moderate dimension e.g. there were 30 input parameters, then we need a vast number of evaluations of the model to fill this 30 dimensional space: e.g. corners only $2^{30} = 1$ billion evaluations.
- A Bayes Linear (or related GP) emulator is a statistical construct that mimics the model, but which is extremely fast to evaluate, often several orders of magnitude faster than the model: use the emulator to learn about x.
- In our galaxy formation model EAGLE the emulators are $10^9 - 10^{12}$ times faster than the model, depending which version of EAGLE we compare to.

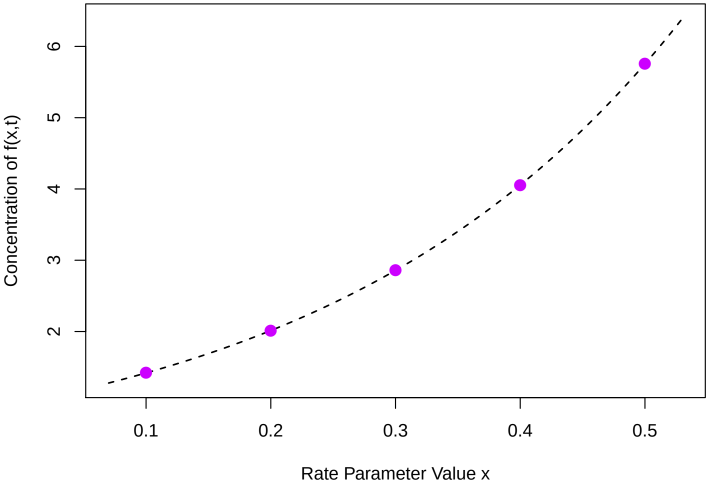
Emulating the Model: Simple 1D Example

- For more realistic models we do not know the full analytic solution for $f(x)$.
- Instead for fixed values of x we would solve for $f(x)$ numerically: this can be too slow even for relatively fast models.
- For e.g. galaxy formation models, it takes between 1 day to 1.5 months to solve for $f(x)$ for a single x. Our techniques are designed to cope with this.
- If x was of moderate dimension e.g. there were 30 input parameters, then we need a vast number of evaluations of the model to fill this 30 dimensional space: e.g. corners only $2^{30} = 1$ billion evaluations.
- A Bayes Linear (or related GP) emulator is a statistical construct that mimics the model, but which is extremely fast to evaluate, often several orders of magnitude faster than the model: use the emulator to learn about x.
- In our galaxy formation model EAGLE the emulators are $10^9 - 10^{12}$ times faster than the model, depending which version of EAGLE we compare to.

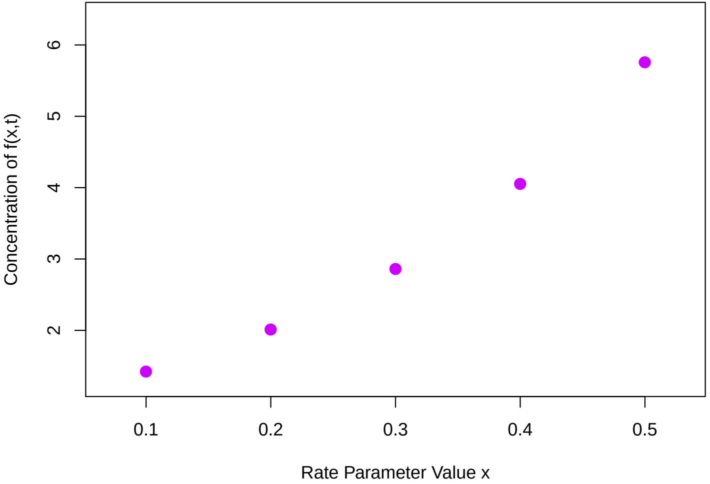


- Consider the graph of $f(x)$: in general we do not have the analytic solution of $f(x)$, here given by the dashed line.

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

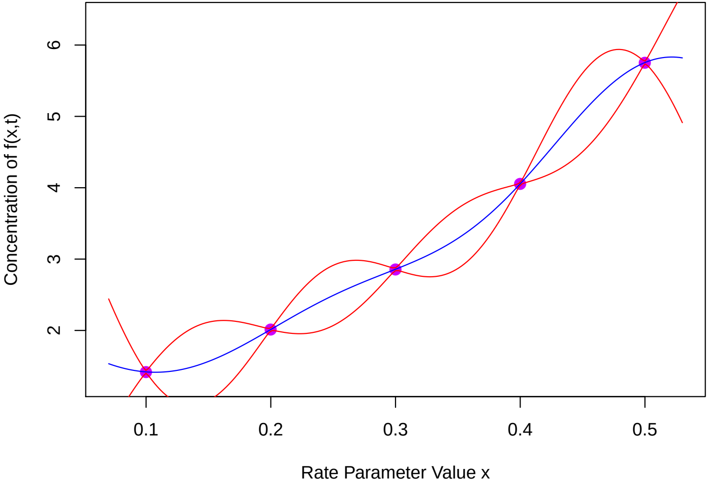


- Consider the graph of $f(x)$: in general we do not have the analytic solution of $f(x)$, here given by the dashed line.
- Instead we only have a finite number of runs of the model, in this case five.

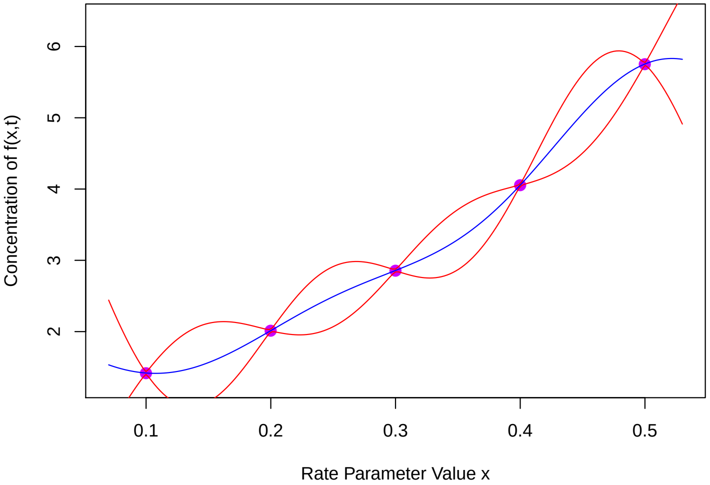


- Consider the graph of $f(x)$: in general we do not have the analytic solution of $f(x)$, here given by the dashed line.

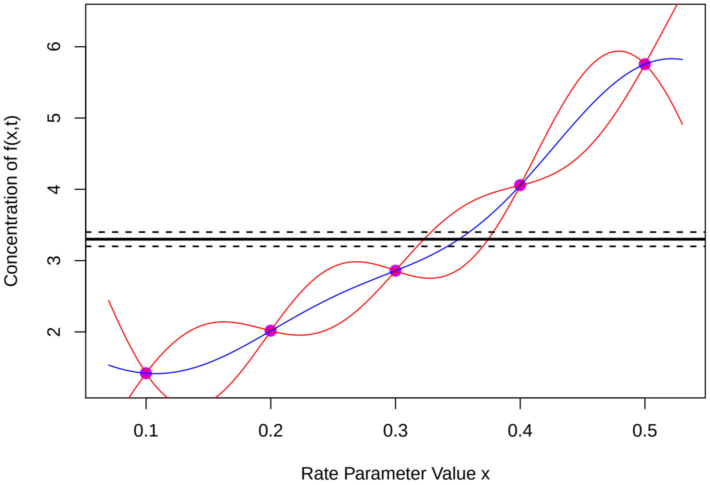
- Instead we only have a finite number of runs of the model, in this case five.



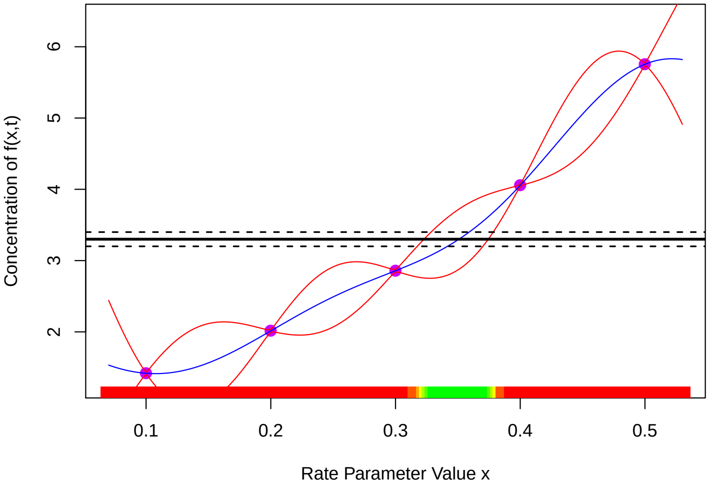
- The emulator can be used to represent our beliefs about the behaviour of the model at untested values of x, and is fast to evaluate.
- It gives both the expected value of $f(x)$ (the blue line) along with a credible interval for $f(x)$ (the red lines) representing the uncertainty about the model's behaviour.



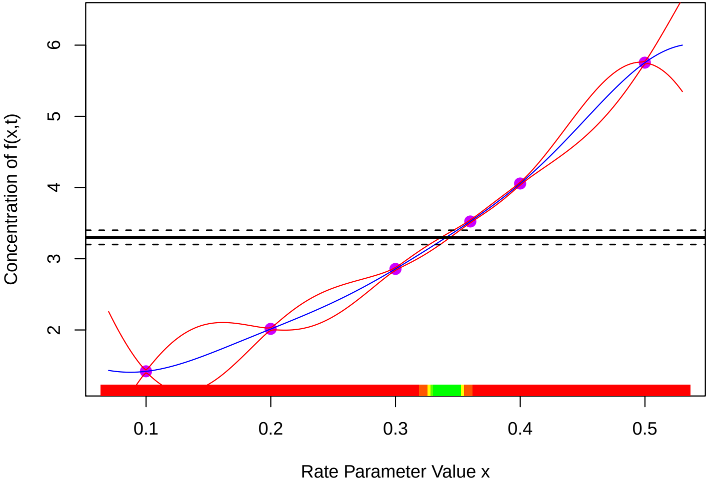
- The emulator can be used to represent our beliefs about the behaviour of the model at untested values of x, and is fast to evaluate.
- It gives both the expected value of $f(x)$ (the blue line) along with a credible interval for $f(x)$ (the red lines) representing the uncertainty about the model's behaviour.



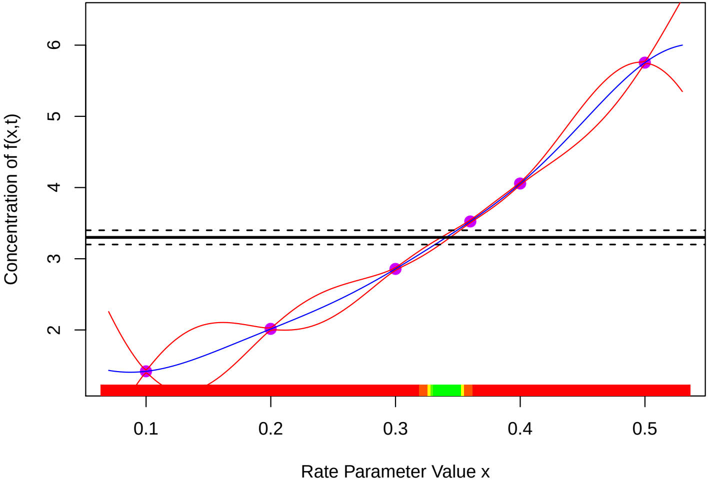
- Comparing the emulator to the observed measurement we again identify the set of x values currently consistent with this data (the observed errors here have been reduced for clarity).
- Note the uncertainty on x now includes uncertainty coming from the emulator.



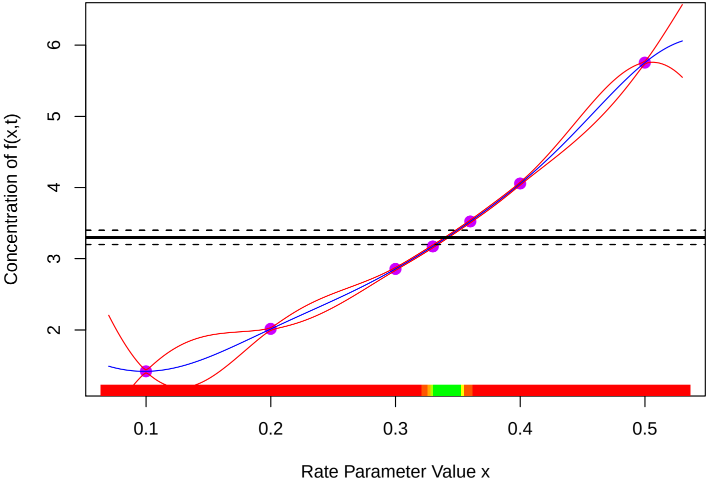
- Comparing the emulator to the observed measurement we again identify the set of x values currently consistent with this data (the observed errors here have been reduced for clarity).
- Note the uncertainty on x now includes uncertainty coming from the emulator.



- 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: the History Match is complete.

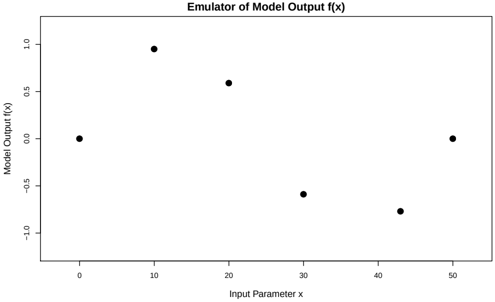


- We perform a 2nd iteration or wave of runs to improve emulator accuracy.
- The runs are located only at non-implausible (green/yellow) points.
- Now the emulator is more accurate than the observation, and we can identify the set of all x values of interest: the History Match is complete.

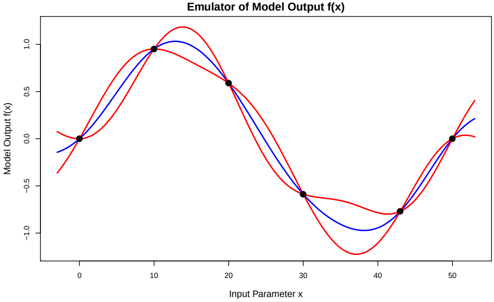


- We perform a 2nd iteration or wave of runs to improve emulator accuracy.
- The runs are located only at non-implausible (green/yellow) points.
- Now the emulator is more accurate than the observation, and we can identify the set of all x values of interest: the History Match is complete.

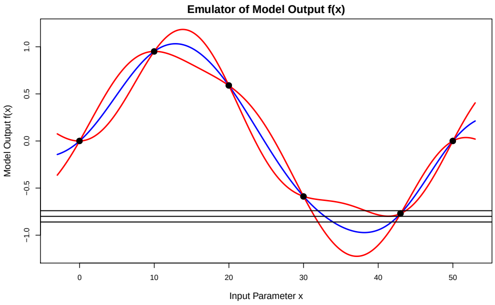
History Matching via Implausibility: a 1D Example



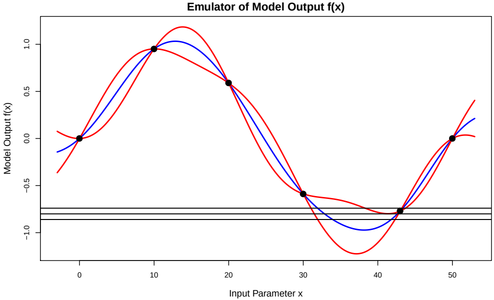
History Matching via Implausibility: a 1D Example

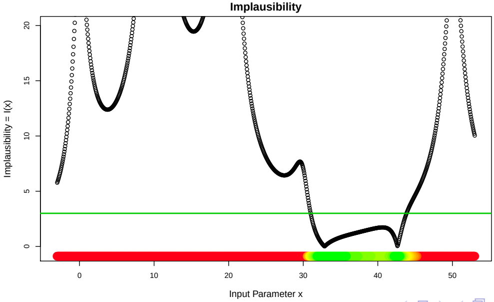


History Matching via Implausibility: a 1D Example

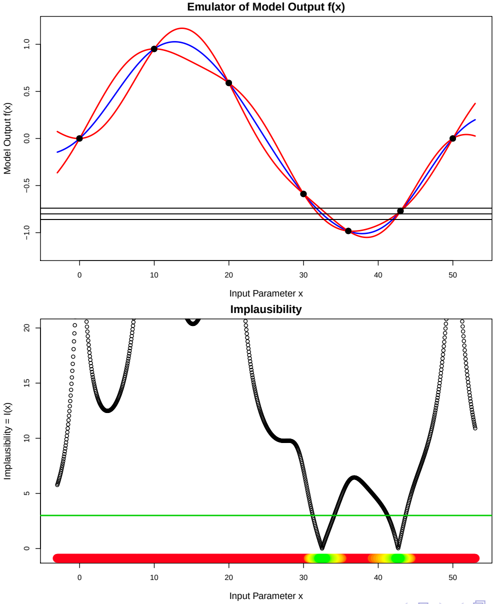


History Matching via Implausibility: a 1D Example

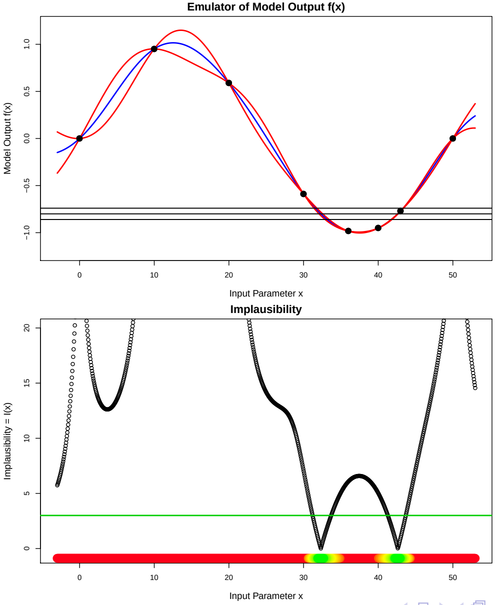




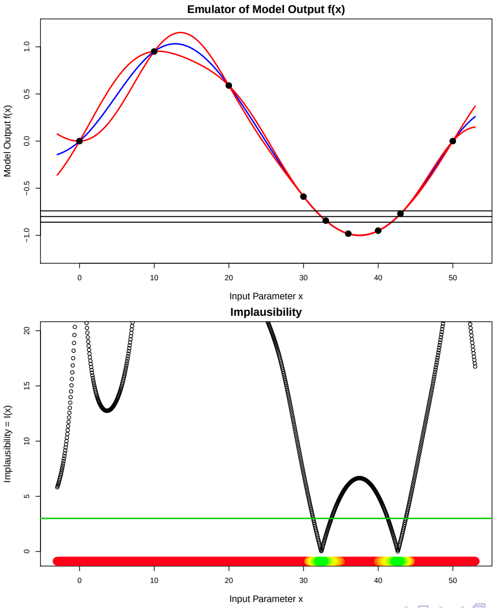
History Matching via Implausibility: a 1D Example



History Matching via Implausibility: a 1D Example



History Matching via Implausibility: a 1D Example



- Emulators mimic the model and are several orders of magnitude faster, but need a certain number of model runs to train them.
- Very approximately, about 10 times the dimension of the input space.
- But what do we do if the model is really slow?
- Answer: Multilevel emulation.
- We will demonstrate this on a model of galaxy formation called EAGLE.
- Note: multilevel emulation is gaining popularity now, but ideas date back to work by Goldstein in 1996.

Multilevel Emulation

- Emulators mimic the model and are several orders of magnitude faster, but need a certain number of model runs to train them.
- Very approximately, about 10 times the dimension of the input space.
- But what do we do if the model is really slow?
- Answer: Multilevel emulation.
- We will demonstrate this on a model of galaxy formation called EAGLE.
- Note: multilevel emulation is gaining popularity now, but ideas date back to work by Goldstein in 1996.

Multilevel Emulation

- Emulators mimic the model and are several orders of magnitude faster, but need a certain number of model runs to train them.
- Very approximately, about 10 times the dimension of the input space.
- But what do we do if the model is really slow?
- Answer: Multilevel emulation.
- We will demonstrate this on a model of galaxy formation called EAGLE.
- Note: multilevel emulation is gaining popularity now, but ideas date back to work by Goldstein in 1996.

- Emulators mimic the model and are several orders of magnitude faster, but need a certain number of model runs to train them.
- Very approximately, about 10 times the dimension of the input space.
- But what do we do if the model is really slow?
- Answer: Multilevel emulation.
- We will demonstrate this on a model of galaxy formation called EAGLE.
- Note: multilevel emulation is gaining popularity now, but ideas date back to work by Goldstein in 1996.

Multilevel Emulation

- Emulators mimic the model and are several orders of magnitude faster, but need a certain number of model runs to train them.
- Very approximately, about 10 times the dimension of the input space.
- But what do we do if the model is really slow?
- Answer: Multilevel emulation.
- We will demonstrate this on a model of galaxy formation called EAGLE.
- Note: multilevel emulation is gaining popularity now, but ideas date back to work by Goldstein in 1996.

- Emulators mimic the model and are several orders of magnitude faster, but need a certain number of model runs to train them.
- Very approximately, about 10 times the dimension of the input space.
- But what do we do if the model is really slow?
- Answer: Multilevel emulation.
- We will demonstrate this on a model of galaxy formation called EAGLE.
- Note: multilevel emulation is gaining popularity now, but ideas date back to work by Goldstein in 1996.

The EAGLE Model

- EAGLE (Evolution and Assembly of GaLaxies and their Environments: http://icc.dur.ac.uk/Eagle/) is a simulation aimed at understanding how galaxies form and evolve.
- It models the formation of structures in a cosmological volume of size (100 Megaparsecs)^3, approximately (326 million light-years)^3.
- This volume contains approximately 10,000 galaxies of the size of the Milky Way or larger, enabling a comparison with detailed galactic surveys.
- The EAGLE simulation is one of the largest cosmological hydrodynamical simulations ever, using nearly 7 billion particles to model the physics, and spanning over 5 orders of magnitude in resolution.
- It took more than one and a half months of computer time on 4064 cores of the DiRAC-2 supercomputer at Durham University, UK (about 5 million hours of CPU time), for a single evaluation.
- However, it still has 8 primary uncertain input parameters denoted x, that relate to the core collapse supernovae and supermassive black holes.
- EAGLE output $f(x)$ can be compared to a variety of observed galaxy data z: Stellar Mass Function, Galaxy Sizes, Mass Metalicity relation ...

The EAGLE Model

- EAGLE (Evolution and Assembly of GaLaxies and their Environments: http://icc.dur.ac.uk/Eagle/) is a simulation aimed at understanding how galaxies form and evolve.
- It models the formation of structures in a cosmological volume of size (100 Megaparsecs)^3, approximately (326 million light-years)^3.
- This volume contains approximately 10,000 galaxies of the size of the Milky Way or larger, enabling a comparison with detailed galactic surveys.
- The EAGLE simulation is one of the largest cosmological hydrodynamical simulations ever, using nearly 7 billion particles to model the physics, and spanning over 5 orders of magnitude in resolution.
- It took more than one and a half months of computer time on 4064 cores of the DiRAC-2 supercomputer at Durham University, UK (about 5 million hours of CPU time), for a single evaluation.
- However, it still has 8 primary uncertain input parameters denoted x, that relate to the core collapse supernovae and supermassive black holes.
- EAGLE output $f(x)$ can be compared to a variety of observed galaxy data z: Stellar Mass Function, Galaxy Sizes, Mass Metalicity relation ...

The EAGLE Model

- EAGLE (Evolution and Assembly of GaLaxies and their Environments: http://icc.dur.ac.uk/Eagle/) is a simulation aimed at understanding how galaxies form and evolve.
- It models the formation of structures in a cosmological volume of size (100 Megaparsecs)^3, approximately (326 million light-years)^3.
- This volume contains approximately 10,000 galaxies of the size of the Milky Way or larger, enabling a comparison with detailed galactic surveys.
- The EAGLE simulation is one of the largest cosmological hydrodynamical simulations ever, using nearly 7 billion particles to model the physics, and spanning over 5 orders of magnitude in resolution.
- It took more than one and a half months of computer time on 4064 cores of the DiRAC-2 supercomputer at Durham University, UK (about 5 million hours of CPU time), for a single evaluation.
- However, it still has 8 primary uncertain input parameters denoted x, that relate to the core collapse supernovae and supermassive black holes.
- EAGLE output $f(x)$ can be compared to a variety of observed galaxy data z: Stellar Mass Function, Galaxy Sizes, Mass Metalicity relation . . .

The EAGLE Model

- EAGLE (Evolution and Assembly of GaLaxies and their Environments: http://icc.dur.ac.uk/Eagle/) is a simulation aimed at understanding how galaxies form and evolve.
- It models the formation of structures in a cosmological volume of size (100 Megaparsecs)^3, approximately (326 million light-years)^3.
- This volume contains approximately 10,000 galaxies of the size of the Milky Way or larger, enabling a comparison with detailed galactic surveys.
- The EAGLE simulation is one of the largest cosmological hydrodynamical simulations ever, using nearly 7 billion particles to model the physics, and spanning over 5 orders of magnitude in resolution.
- It took more than one and a half months of computer time on 4064 cores of the DiRAC-2 supercomputer at Durham University, UK (about 5 million hours of CPU time), for a single evaluation.
- However, it still has 8 primary uncertain input parameters denoted x, that relate to the core collapse supernovae and supermassive black holes.
- EAGLE output $f(x)$ can be compared to a variety of observed galaxy data z: Stellar Mass Function, Galaxy Sizes, Mass Metalicity relation . . .

The EAGLE Model

- EAGLE (Evolution and Assembly of GaLaxies and their Environments: http://icc.dur.ac.uk/Eagle/) is a simulation aimed at understanding how galaxies form and evolve.
- It models the formation of structures in a cosmological volume of size (100 Megaparsecs)^3, approximately (326 million light-years)^3.
- This volume contains approximately 10,000 galaxies of the size of the Milky Way or larger, enabling a comparison with detailed galactic surveys.
- The EAGLE simulation is one of the largest cosmological hydrodynamical simulations ever, using nearly 7 billion particles to model the physics, and spanning over 5 orders of magnitude in resolution.
- It took more than one and a half months of computer time on 4064 cores of the DiRAC-2 supercomputer at Durham University, UK (about 5 million hours of CPU time), for a single evaluation.
- However, it still has 8 primary uncertain input parameters denoted x, that relate to the core collapse supernovae and supermassive black holes.
- EAGLE output $f(x)$ can be compared to a variety of observed galaxy data z: Stellar Mass Function, Galaxy Sizes, Mass Metalicity relation . . .

The EAGLE Model

- EAGLE (Evolution and Assembly of GaLaxies and their Environments: http://icc.dur.ac.uk/Eagle/) is a simulation aimed at understanding how galaxies form and evolve.
- It models the formation of structures in a cosmological volume of size (100 Megaparsecs)^3, approximately (326 million light-years)^3.
- This volume contains approximately 10,000 galaxies of the size of the Milky Way or larger, enabling a comparison with detailed galactic surveys.
- The EAGLE simulation is one of the largest cosmological hydrodynamical simulations ever, using nearly 7 billion particles to model the physics, and spanning over 5 orders of magnitude in resolution.
- It took more than one and a half months of computer time on 4064 cores of the DiRAC-2 supercomputer at Durham University, UK (about 5 million hours of CPU time), for a single evaluation.
- However, it still has 8 primary uncertain input parameters denoted x, that relate to the core collapse supernovae and supermassive black holes.
- EAGLE output $f(x)$ can be compared to a variety of observed galaxy data z: Stellar Mass Function, Galaxy Sizes, Mass Metalicity relation . . .

The EAGLE Model

- EAGLE (Evolution and Assembly of GaLaxies and their Environments: http://icc.dur.ac.uk/Eagle/) is a simulation aimed at understanding how galaxies form and evolve.
- It models the formation of structures in a cosmological volume of size (100 Megaparsecs)^3, approximately (326 million light-years)^3.
- This volume contains approximately 10,000 galaxies of the size of the Milky Way or larger, enabling a comparison with detailed galactic surveys.
- The EAGLE simulation is one of the largest cosmological hydrodynamical simulations ever, using nearly 7 billion particles to model the physics, and spanning over 5 orders of magnitude in resolution.
- It took more than one and a half months of computer time on 4064 cores of the DiRAC-2 supercomputer at Durham University, UK (about 5 million hours of CPU time), for a single evaluation.
- However, it still has 8 primary uncertain input parameters denoted x, that relate to the core collapse supernovae and supermassive black holes.
- EAGLE output $f(x)$ can be compared to a variety of observed galaxy data z: Stellar Mass Function, Galaxy Sizes, Mass Metalicity relation . . .

EAGLE Input Parameters (Details)

- To perform one run, we need to specify numbers for each of the following 8 inputs:

Input Parameter	min	max	Transform	Process
SNII_MinEnergyFraction	0.1	1.0	-	Supernova
SNII_MaxEnergyFraction	1.0	5.0	-	"
SNII_rhogas_power	0.1	3.0	-	"
SNII_rhogas_physdensnormfac	1	50	log_{10}	"
SNII_Width_logTvir_dex	0.1	3	log_{10}	"
BlackHoleViscousAlpha	10^3	10^8	log_{10}	Blackholes
BH_ConstantHeatTemp	10^8	10^9	log_{10}	"
BlackHoleFeedbackFactor	0.01	1	log_{10}	"

- What input values $x \in \mathcal{X}$ should we choose to get 'acceptable' outputs?

EAGLE Input Parameters (Details)

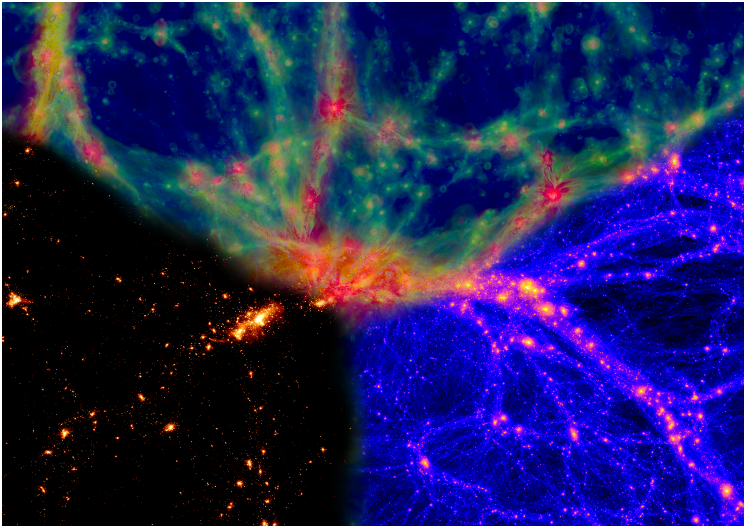
- To perform one run, we need to specify numbers for each of the following 8 inputs:

Input Parameter	min	max	Transform	Process
SNII_MinEnergyFraction	0.1	1.0	-	Supernova
SNII_MaxEnergyFraction	1.0	5.0	-	"
SNII_rhogas_power	0.1	3.0	-	"
SNII_rhogas_physdensnormfac	1	50	log_{10}	"
SNII_Width_logTvir_dex	0.1	3	log_{10}	"
BlackHoleViscousAlpha	10^3	10^8	log_{10}	Blackholes
BH_ConstantHeatTemp	10^8	10^9	log_{10}	"
BlackHoleFeedbackFactor	0.01	1	log_{10}	"

- What input values $x \in \mathcal{X}$ should we choose to get 'acceptable' outputs?

EAGLE Outputs

Gas Temperature



Visual spectrum

Dark Matter density

Multilevel Structure of EAGLE

- The standard EAGLE run (at 100Mpc) is far too expensive to repeat more than a couple of times, so direct emulation is impossible.
- However, thankfully EAGLE has been designed to run at 4 different levels of accuracy, with each level approximately 8 times faster than the previous one.
- These levels correspond to smaller volumes of the Universe:

Level	Volume$^{1/3}$	Approximate Evaluation Time
1	12.5 Mpc	1/512
2	25 Mpc	1/64
3	50 Mpc	1/8
4	100 Mpc	1

- EAGLE is stochastic: lower levels a) have much more noise and b) are structurally i.e. physically different from the higher levels due to limits on sizes of galaxies that can form (among other things).
- We therefore wish to build multilevel emulators that can incorporate expert judgements regarding the relationships between the levels.

Multilevel Structure of EAGLE

- The standard EAGLE run (at 100Mpc) is far too expensive to repeat more than a couple of times, so direct emulation is impossible.
- However, thankfully EAGLE has been designed to run at 4 different levels of accuracy, with each level approximately 8 times faster than the previous one.
- These levels correspond to smaller volumes of the Universe:

Level	Volume$^{1/3}$	Approximate Evaluation Time
1	12.5 Mpc	1/512
2	25 Mpc	1/64
3	50 Mpc	1/8
4	100 Mpc	1

- EAGLE is stochastic: lower levels a) have much more noise and b) are structurally i.e. physically different from the higher levels due to limits on sizes of galaxies that can form (among other things).
- We therefore wish to build multilevel emulators that can incorporate expert judgements regarding the relationships between the levels.

Multilevel Structure of EAGLE

- The standard EAGLE run (at 100Mpc) is far too expensive to repeat more than a couple of times, so direct emulation is impossible.
- However, thankfully EAGLE has been designed to run at 4 different levels of accuracy, with each level approximately 8 times faster than the previous one.
- These levels correspond to smaller volumes of the Universe:

Level	Volume$^{1/3}$	Approximate Evaluation Time
1	12.5 Mpc	1/512
2	25 Mpc	1/64
3	50 Mpc	1/8
4	100 Mpc	1

- EAGLE is stochastic: lower levels a) have much more noise and b) are structurally i.e. physically different from the higher levels due to limits on sizes of galaxies that can form (among other things).
- We therefore wish to build multilevel emulators that can incorporate expert judgements regarding the relationships between the levels.

Multilevel Structure of EAGLE

- The standard EAGLE run (at 100Mpc) is far too expensive to repeat more than a couple of times, so direct emulation is impossible.
- However, thankfully EAGLE has been designed to run at 4 different levels of accuracy, with each level approximately 8 times faster than the previous one.
- These levels correspond to smaller volumes of the Universe:

Level	Volume$^{1/3}$	Approximate Evaluation Time
1	12.5 Mpc	1/512
2	25 Mpc	1/64
3	50 Mpc	1/8
4	100 Mpc	1

- EAGLE is stochastic: lower levels a) have much more noise and b) are structurally i.e. physically different from the higher levels due to limits on sizes of galaxies that can form (among other things).
- We therefore wish to build multilevel emulators that can incorporate expert judgements regarding the relationships between the levels.

Multilevel Structure of EAGLE

- The standard EAGLE run (at 100Mpc) is far too expensive to repeat more than a couple of times, so direct emulation is impossible.

- However, thankfully EAGLE has been designed to run at 4 different levels of accuracy, with each level approximately 8 times faster than the previous one.

- These levels correspond to smaller volumes of the Universe:

Level	Volume$^{1/3}$	Approximate Evaluation Time
1	12.5 Mpc	1/512
2	25 Mpc	1/64
3	50 Mpc	1/8
4	100 Mpc	1

- EAGLE is stochastic: lower levels a) have much more noise and b) are structurally i.e. physically different from the higher levels due to limits on sizes of galaxies that can form (among other things).

- We therefore wish to build multilevel emulators that can incorporate expert judgements regarding the relationships between the levels.

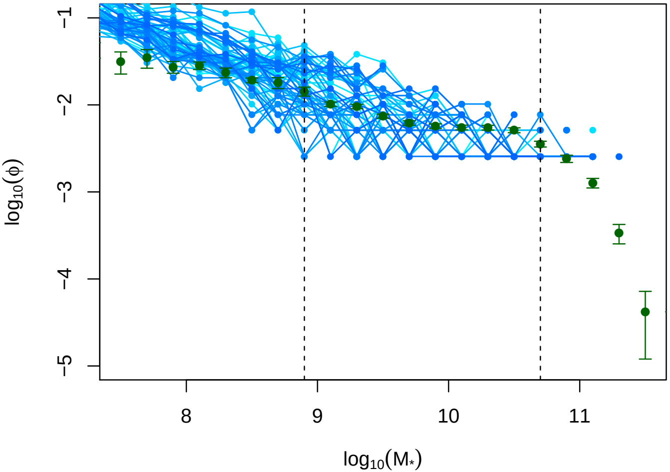
Multilevel Structure of EAGLE

- The standard EAGLE run (at 100Mpc) is far too expensive to repeat more than a couple of times, so direct emulation is impossible.
- However, thankfully EAGLE has been designed to run at 4 different levels of accuracy, with each level approximately 8 times faster than the previous one.
- These levels correspond to smaller volumes of the Universe:

Level	Volume$^{1/3}$	Approximate Evaluation Time
1	12.5 Mpc	1/512
2	25 Mpc	1/64
3	50 Mpc	1/8
4	100 Mpc	1

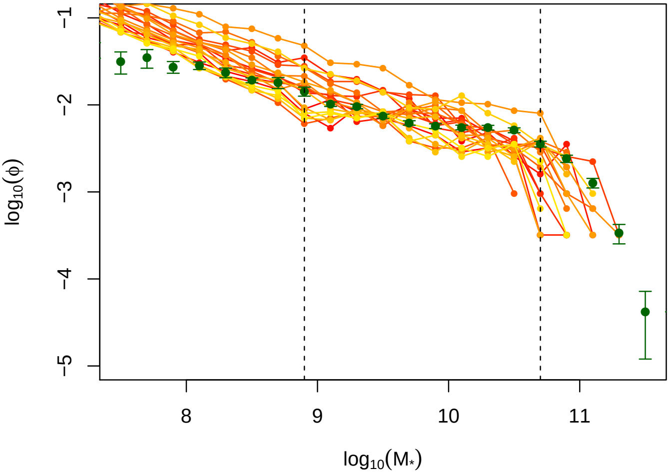
- EAGLE is stochastic: lower levels a) have much more noise and b) are structurally i.e. physically different from the higher levels due to limits on sizes of galaxies that can form (among other things).
- We therefore wish to build multilevel emulators that can incorporate expert judgements regarding the relationships between the levels.

The SMF: 60 runs at level 1 for 12.5Mpc

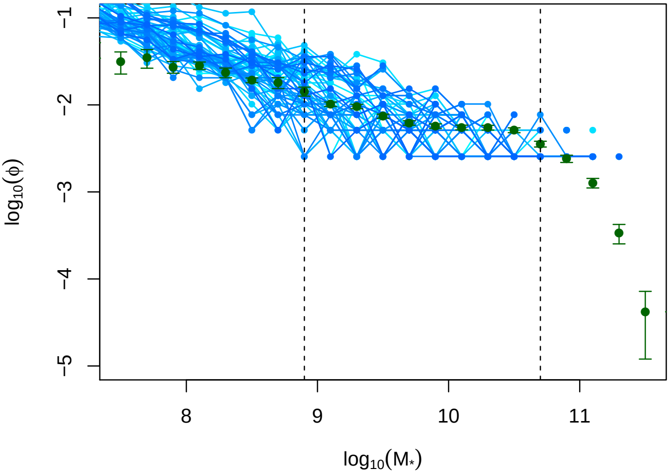


- Level 1: 60 runs of the 12.5 Mpc simulator (2 days on 32 processors per run).

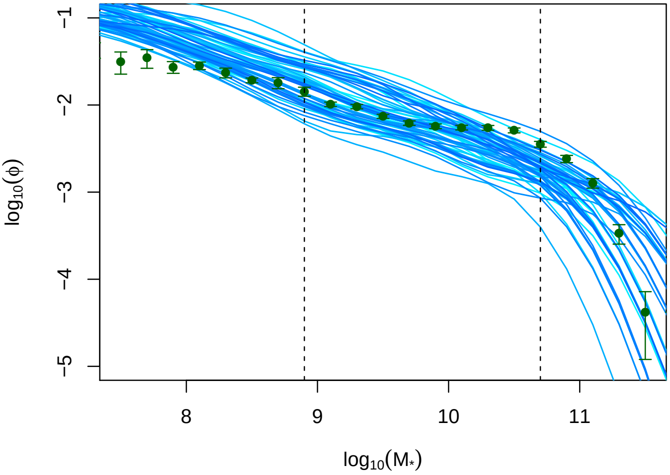
The SMF: 20 runs at level 2 for 25Mpc



- Level 2: 20 runs of the 25 Mpc simulator (8 days on 64 processors per run)

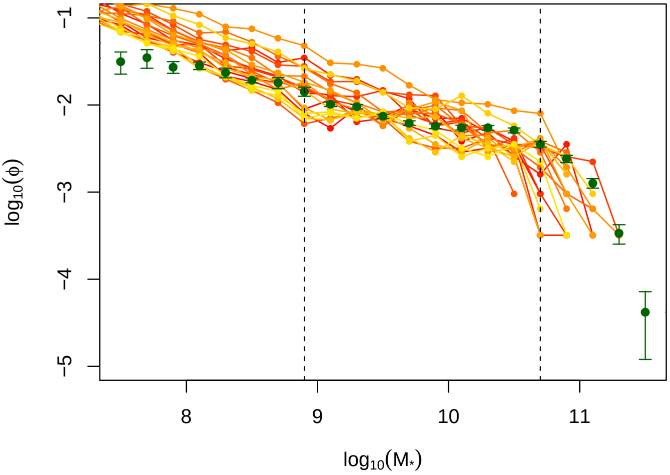


- Level 1: 60 runs of the 12.5 Mpc simulator.



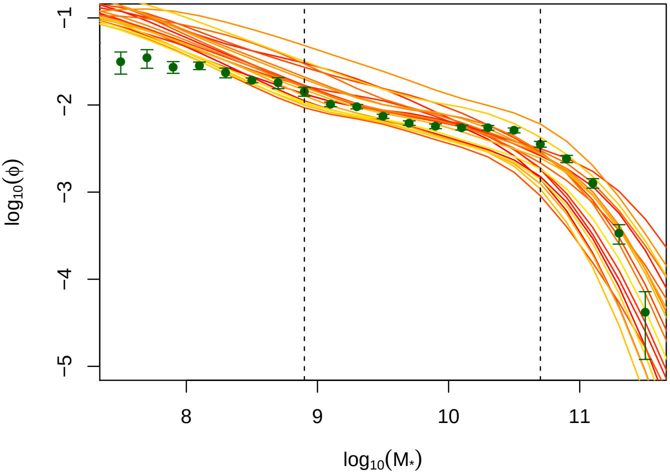
- Level 1: 60 runs of the 12.5 Mpc simulator, smoothed.

The SMF: 20 runs at level 2 for 25Mpc



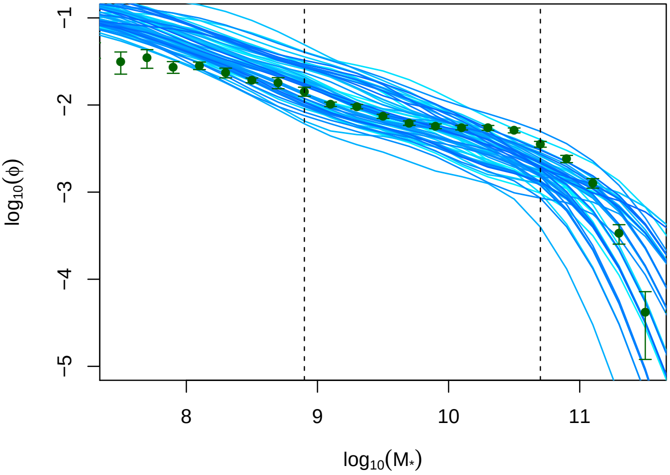
- Level 2: 20 runs of the 25 Mpc simulator.

The SMF: 20 smoothed runs at level 2 for 25Mpc



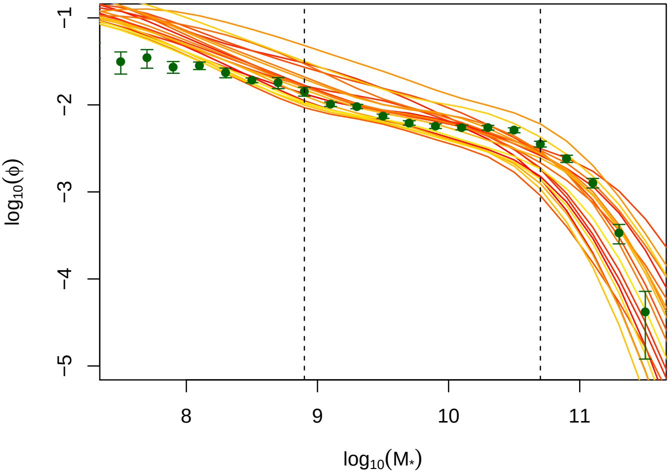
- Level 2: 20 runs of the 25 Mpc simulator, smoothed.

The SMF: 60 smoothed runs at level 1 for 12.5Mpc



- Level 1: 60 runs of the 12.5 Mpc simulator, smoothed.

The SMF: 20 smoothed runs at level 2 for 25Mpc



- Level 2: 20 runs of the 25 Mpc simulator, smoothed.

Linking EAGLE to the Real Universe

- A common major problem is caused by not acknowledging the difference between model $f(x)$ and the system or reality y, and failing to embed them and the observations z into an overarching statistical model.
- Our goal will be to link the real Universe y with EAGLE at the 4th level $f^{(4)}(x)$

$$y = f^{(4)}(x^*) + \epsilon^{(4)}$$

where we define $\epsilon^{(4)}$ to be the Structural Model Discrepancy, which represents the difference between $f^{(4)}(x)$ and the Universe y at some 'best input' x^*.
- (Actually, we may play the game of exploring linking at different levels using $y = f^{(k)}(x^*) + \epsilon^{(k)}$, with $k = 1, \ldots, 4$).
- We relate the true system y to the observed data z via observation errors e:

$$z = y + e$$

- If we assert probabilistic relations between the random vectors $f^{(4)}, \epsilon^{(4)}, e$ and x^* e.g. independence, we can proceed.
- Often, scientists may be able to specify say $\mathrm{E}[\epsilon^{(4)}]$, $\mathrm{E}[e]$ (often zero), and $\mathrm{Var}[\epsilon^{(4)}]$, $\mathrm{Var}[e]$, which, in the multiple output case, may have complex structure.

Linking EAGLE to the Real Universe

- A common major problem is caused by not acknowledging the difference between model $f(x)$ and the system or reality y, and failing to embed them and the observations z into an overarching statistical model.
- Our goal will be to link the real Universe y with EAGLE at the 4th level $f^{(4)}(x)$

$$y = f^{(4)}(x^*) + \epsilon^{(4)}$$

where we define $\epsilon^{(4)}$ to be the Structural Model Discrepancy, which represents the difference between $f^{(4)}(x)$ and the Universe y at some 'best input' x^*.

- (Actually, we may play the game of exploring linking at different levels using $y = f^{(k)}(x^*) + \epsilon^{(k)}$, with $k = 1, \ldots, 4$).
- We relate the true system y to the observed data z via observation errors e:

$$z = y + e$$

- If we assert probabilistic relations between the random vectors $f^{(4)}, \epsilon^{(4)}, e$ and x^* e.g. independence, we can proceed.
- Often, scientists may be able to specify say $\mathrm{E}[\epsilon^{(4)}]$, $\mathrm{E}[e]$ (often zero), and $\mathrm{Var}[\epsilon^{(4)}]$, $\mathrm{Var}[e]$, which, in the multiple output case, may have complex structure.

Linking EAGLE to the Real Universe

- A common major problem is caused by not acknowledging the difference between model $f(x)$ and the system or reality y, and failing to embed them and the observations z into an overarching statistical model.
- Our goal will be to link the real Universe y with EAGLE at the 4th level $f^{(4)}(x)$

$$y = f^{(4)}(x^*) + \epsilon^{(4)}$$

  where we define $\epsilon^{(4)}$ to be the Structural Model Discrepancy, which represents the difference between $f^{(4)}(x)$ and the Universe y at some 'best input' x^*.
- (Actually, we may play the game of exploring linking at different levels using $y = f^{(k)}(x^*) + \epsilon^{(k)}$, with $k = 1, \ldots, 4$).
- We relate the true system y to the observed data z via observation errors e:

$$z = y + e$$

- If we assert probabilistic relations between the random vectors $f^{(4)}, \epsilon^{(4)}, e$ and x^* e.g. independence, we can proceed.
- Often, scientists may be able to specify say $\mathrm{E}[\epsilon^{(4)}]$, $\mathrm{E}[e]$ (often zero), and $\mathrm{Var}[\epsilon^{(4)}]$, $\mathrm{Var}[e]$, which, in the multiple output case, may have complex structure.

Linking EAGLE to the Real Universe

- A common major problem is caused by not acknowledging the difference between model $f(x)$ and the system or reality y, and failing to embed them and the observations z into an overarching statistical model.
- Our goal will be to link the real Universe y with EAGLE at the 4th level $f^{(4)}(x)$

$$y = f^{(4)}(x^*) + \epsilon^{(4)}$$

  where we define $\epsilon^{(4)}$ to be the Structural Model Discrepancy, which represents the difference between $f^{(4)}(x)$ and the Universe y at some 'best input' x^*.
- (Actually, we may play the game of exploring linking at different levels using $y = f^{(k)}(x^*) + \epsilon^{(k)}$, with $k = 1, \ldots, 4$).
- We relate the true system y to the observed data z via observation errors e:

$$z = y + e$$

- If we assert probabilistic relations between the random vectors $f^{(4)}, \epsilon^{(4)}, e$ and x^* e.g. independence, we can proceed.
- Often, scientists may be able to specify say $\mathrm{E}[\epsilon^{(4)}]$, $\mathrm{E}[e]$ (often zero), and $\mathrm{Var}[\epsilon^{(4)}]$, $\mathrm{Var}[e]$, which, in the multiple output case, may have complex structure.

- A common major problem is caused by not acknowledging the difference between model $f(x)$ and the system or reality y, and failing to embed them and the observations z into an overarching statistical model.
- Our goal will be to link the real Universe y with EAGLE at the 4th level $f^{(4)}(x)$

$$y = f^{(4)}(x^*) + \epsilon^{(4)}$$

where we define $\epsilon^{(4)}$ to be the Structural Model Discrepancy, which represents the difference between $f^{(4)}(x)$ and the Universe y at some 'best input' x^*.

- (Actually, we may play the game of exploring linking at different levels using $y = f^{(k)}(x^*) + \epsilon^{(k)}$, with $k = 1, \ldots, 4$).
- We relate the true system y to the observed data z via observation errors e:

$$z = y + e$$

- If we assert probabilistic relations between the random vectors $f^{(4)}, \epsilon^{(4)}, e$ and x^* e.g. independence, we can proceed.
- Often, scientists may be able to specify say $\mathrm{E}[\epsilon^{(4)}]$, $\mathrm{E}[e]$ (often zero), and $\mathrm{Var}[\epsilon^{(4)}]$, $\mathrm{Var}[e]$, which, in the multiple output case, may have complex structure.

Linking EAGLE to the Real Universe

- A common major problem is caused by not acknowledging the difference between model $f(x)$ and the system or reality y, and failing to embed them and the observations z into an overarching statistical model.
- Our goal will be to link the real Universe y with EAGLE at the 4th level $f^{(4)}(x)$

$$y = f^{(4)}(x^*) + \epsilon^{(4)}$$

  where we define $\epsilon^{(4)}$ to be the Structural Model Discrepancy, which represents the difference between $f^{(4)}(x)$ and the Universe y at some 'best input' x^*.
- (Actually, we may play the game of exploring linking at different levels using $y = f^{(k)}(x^*) + \epsilon^{(k)}$, with $k = 1, \ldots, 4$).
- We relate the true system y to the observed data z via observation errors e:

$$z = y + e$$

- If we assert probabilistic relations between the random vectors $f^{(4)}, \epsilon^{(4)}, e$ and x^* e.g. independence, we can proceed.
- Often, scientists may be able to specify say $\mathrm{E}[\epsilon^{(4)}]$, $\mathrm{E}[e]$ (often zero), and $\mathrm{Var}[\epsilon^{(4)}]$, $\mathrm{Var}[e]$, which, in the multiple output case, may have complex structure.

Emulation Theory: Single Level

- To emulate at the lowest level, i.e. for $f^{(1)}(x)$ we proceed as follows.
- For each of the outputs of interest $f_i^{(1)}(x)$, we pick active variables x_{A_i} then emulate univariately (at first) using:

$$f_i^{(1)}(x) = \sum_j \beta_{ij}^{(1)}\, g_{ij}(x_{A_i}) + u_i^{(1)}(x_{A_i}) + v_i^{(1)}(x)$$

- The $\sum_j \beta_{ij}^{(1)}\, g_{ij}(x_{A_i})$ is a 2nd order polynomial in the active inputs, with $\beta_{ij}^{(1)}$ unknown constants: very important to include such global structure here.
- $u_i^{(1)}(x_{A_i})$ is a weakly stationary (GP) process representing local variation, with covariance:

$$\mathrm{Cov}[u_i^{(1)}(x_{A_i}), u_i^{(1)}(x'_{A_i})] \;=\; (\sigma_i^{(1)})^2 \exp[-|x_{A_i} - x'_{A_i}|^{p_i} / \theta_i^{(1)p_i}]$$

- The nugget $v_i^{(1)}(x)$ models the effects of inactive variables as random noise.
- The Emulators give the expectation $\mathrm{E}[f_i^{(1)}(x)]$ and variance $\mathrm{Var}[f_i^{(1)}(x)]$ at point x for each output of interest and are **fast** to evaluate.

Emulation Theory: Single Level

- To emulate at the lowest level, i.e. for $f^{(1)}(x)$ we proceed as follows.
- For each of the outputs of interest $f_i^{(1)}(x)$, we pick active variables x_{A_i} then emulate univariately (at first) using:

$$f_i^{(1)}(x) = \sum_j \beta_{ij}^{(1)}\, g_{ij}(x_{A_i}) + u_i^{(1)}(x_{A_i}) + v_i^{(1)}(x)$$

- The $\sum_j \beta_{ij}^{(1)}\, g_{ij}(x_{A_i})$ is a 2nd order polynomial in the active inputs, with $\beta_{ij}^{(1)}$ unknown constants: very important to include such global structure here.
- $u_i^{(1)}(x_{A_i})$ is a weakly stationary (GP) process representing local variation, with covariance:

$$\mathrm{Cov}[u_i^{(1)}(x_{A_i}), u_i^{(1)}(x'_{A_i})] \;=\; (\sigma_i^{(1)})^2 \exp[-|x_{A_i} - x'_{A_i}|^{p_i}/\theta_i^{(1)p_i}]$$

- The nugget $v_i^{(1)}(x)$ models the effects of inactive variables as random noise.
- The Emulators give the expectation $\mathrm{E}[f_i^{(1)}(x)]$ and variance Var$[f_i^{(1)}(x)]$ at point x for each output of interest and are **fast** to evaluate.

Emulation Theory: Single Level

- To emulate at the lowest level, i.e. for $f^{(1)}(x)$ we proceed as follows.
- For each of the outputs of interest $f_i^{(1)}(x)$, we pick active variables x_{A_i} then emulate univariately (at first) using:

$$f_i^{(1)}(x) = \sum_j \beta_{ij}^{(1)} g_{ij}(x_{A_i}) + u_i^{(1)}(x_{A_i}) + v_i^{(1)}(x)$$

- The $\sum_j \beta_{ij}^{(1)} g_{ij}(x_{A_i})$ is a 2nd order polynomial in the active inputs, with $\beta_{ij}^{(1)}$ unknown constants: very important to include such global structure here.
- $u_i^{(1)}(x_{A_i})$ is a weakly stationary (GP) process representing local variation, with covariance:

$$\mathrm{Cov}[u_i^{(1)}(x_{A_i}), u_i^{(1)}(x'_{A_i})] \quad = \quad (\sigma_i^{(1)})^2 \exp[-|x_{A_i} - x'_{A_i}|^{p_i} / \theta_i^{(1)p_i}]$$

- The nugget $v_i^{(1)}(x)$ models the effects of inactive variables as random noise.
- The Emulators give the expectation $\mathrm{E}[f_i^{(1)}(x)]$ and variance $\mathrm{Var}[f_i^{(1)}(x)]$ at point x for each output of interest and are **fast** to evaluate.

Emulation Theory: Single Level

- To emulate at the lowest level, i.e. for $f^{(1)}(x)$ we proceed as follows.
- For each of the outputs of interest $f_i^{(1)}(x)$, we pick active variables x_{A_i} then emulate univariately (at first) using:

$$f_i^{(1)}(x) = \sum_j \beta_{ij}^{(1)}\, g_{ij}(x_{A_i}) + u_i^{(1)}(x_{A_i}) + v_i^{(1)}(x)$$

- The $\sum_j \beta_{ij}^{(1)}\, g_{ij}(x_{A_i})$ is a 2nd order polynomial in the active inputs, with $\beta_{ij}^{(1)}$ unknown constants: very important to include such global structure here.
- $u_i^{(1)}(x_{A_i})$ is a weakly stationary (GP) process representing local variation, with covariance:

$$\mathrm{Cov}[u_i^{(1)}(x_{A_i}), u_i^{(1)}(x'_{A_i})] \;=\; (\sigma_i^{(1)})^2 \exp[-|x_{A_i} - x'_{A_i}|^{p_i} / \theta_i^{(1)p_i}]$$

- The nugget $v_i^{(1)}(x)$ models the effects of inactive variables as random noise.
- The Emulators give the expectation $\mathrm{E}[f_i^{(1)}(x)]$ and variance $\mathrm{Var}[f_i^{(1)}(x)]$ at point x for each output of interest and are **fast** to evaluate.

Emulation Theory: Single Level

- To emulate at the lowest level, i.e. for $f^{(1)}(x)$ we proceed as follows.
- For each of the outputs of interest $f_i^{(1)}(x)$, we pick active variables x_{A_i} then emulate univariately (at first) using:

$$f_i^{(1)}(x) = \sum_j \beta_{ij}^{(1)} \, g_{ij}(x_{A_i}) + u_i^{(1)}(x_{A_i}) + v_i^{(1)}(x)$$

- The $\sum_j \beta_{ij}^{(1)} \, g_{ij}(x_{A_i})$ is a 2nd order polynomial in the active inputs, with $\beta_{ij}^{(1)}$ unknown constants: very important to include such global structure here.
- $u_i^{(1)}(x_{A_i})$ is a weakly stationary (GP) process representing local variation, with covariance:

$$\mathrm{Cov}[u_i^{(1)}(x_{A_i}), u_i^{(1)}(x'_{A_i})] \;=\; (\sigma_i^{(1)})^2 \exp[-|x_{A_i} - x'_{A_i}|^{p_i} / \theta_i^{(1)p_i}]$$

- The nugget $v_i^{(1)}(x)$ models the effects of inactive variables as random noise.
- The Emulators give the expectation $\mathrm{E}[f_i^{(1)}(x)]$ and variance $\mathrm{Var}[f_i^{(1)}(x)]$ at point x for each output of interest and are **fast** to evaluate.

Emulation Theory: Single Level

- To emulate at the lowest level, i.e. for $f^{(1)}(x)$ we proceed as follows.
- For each of the outputs of interest $f_i^{(1)}(x)$, we pick active variables x_{A_i} then emulate univariately (at first) using:

$$f_i^{(1)}(x) = \sum_j \beta_{ij}^{(1)} g_{ij}(x_{A_i}) + u_i^{(1)}(x_{A_i}) + v_i^{(1)}(x)$$

- The $\sum_j \beta_{ij}^{(1)} g_{ij}(x_{A_i})$ is a 2nd order polynomial in the active inputs, with $\beta_{ij}^{(1)}$ unknown constants: very important to include such global structure here.
- $u_i^{(1)}(x_{A_i})$ is a weakly stationary (GP) process representing local variation, with covariance:

$$\mathrm{Cov}[u_i^{(1)}(x_{A_i}), u_i^{(1)}(x'_{A_i})] \quad = \quad (\sigma_i^{(1)})^2 \exp[-|x_{A_i} - x'_{A_i}|^{p_i} / \theta_i^{(1)p_i}]$$

- The nugget $v_i^{(1)}(x)$ models the effects of inactive variables as random noise.
- The Emulators give the expectation $\mathrm{E}[f_i^{(1)}(x)]$ and variance $\mathrm{Var}[f_i^{(1)}(x)]$ at point x for each output of interest and are **fast** to evaluate.

Emulation Theory: Bayes Linear Methods

- We perform an initial wave 1 set of n runs at input locations $x^{(1)}, x^{(2)}, \dots, x^{(n)}$ giving a column vector of model output values

$$D_i = (f_i(x^{(1)}), f_i(x^{(2)}), \dots, f_i(x^{(n)}))^T$$

- Perhaps we are only prepared to specify expectations, variances and covariances for our many uncertainties.

- To update our emulators there is an alternative version of Bayesian statistics that is easier to specify and far easier to calculate with.

- Instead of Bayes Theorem we use the Bayes linear update:

$$\begin{aligned} \mathrm{E}_{D_i}(f_i(x)) &= \mathrm{E}(f_i(x)) + \mathrm{Cov}(f_i(x), D_i)\mathrm{Var}(D_i)^{-1}(D_i - \mathrm{E}(D_i)) \\ \mathrm{Var}_{D_i}(f_i(x)) &= \mathrm{Var}(f_i(x)) - \mathrm{Cov}(f_i(x), D_i)\mathrm{Var}(D_i)^{-1}\mathrm{Cov}(D_i, f_i(x)) \end{aligned}$$

where $\mathrm{E}_{D_i}(f_i(x))$ and $\mathrm{Var}_{D_i}(f_i(x))$ are the Bayes Linear adjusted expectation and variance for $f_i(x)$ at new input point x, and are all that are needed for the subsequent implausibility measures and history match.

- We perform an initial wave 1 set of n runs at input locations $x^{(1)}, x^{(2)}, \ldots, x^{(n)}$ giving a column vector of model output values

$$D_i = (f_i(x^{(1)}), f_i(x^{(2)}), \ldots, f_i(x^{(n)}))^T$$

- Perhaps we are only prepared to specify expectations, variances and covariances for our many uncertainties.

- To update our emulators there is an alternative version of Bayesian statistics that is easier to specify and far easier to calculate with.

- Instead of Bayes Theorem we use the Bayes linear update:

$$\begin{aligned} \mathrm{E}_{D_i}(f_i(x)) &= \mathrm{E}(f_i(x)) + \mathrm{Cov}(f_i(x), D_i)\mathrm{Var}(D_i)^{-1}(D_i - \mathrm{E}(D_i)) \\ \mathrm{Var}_{D_i}(f_i(x)) &= \mathrm{Var}(f_i(x)) - \mathrm{Cov}(f_i(x), D_i)\mathrm{Var}(D_i)^{-1}\mathrm{Cov}(D_i, f_i(x)) \end{aligned}$$

  where $\mathrm{E}_{D_i}(f_i(x))$ and $\mathrm{Var}_{D_i}(f_i(x))$ are the Bayes Linear adjusted expectation and variance for $f_i(x)$ at new input point x, and are all that are needed for the subsequent implausibility measures and history match.

Emulation Theory: Bayes Linear Methods

- We perform an initial wave 1 set of n runs at input locations $x^{(1)}, x^{(2)}, \ldots, x^{(n)}$ giving a column vector of model output values

$$D_i = (f_i(x^{(1)}), f_i(x^{(2)}), \ldots, f_i(x^{(n)}))^T$$

- Perhaps we are only prepared to specify expectations, variances and covariances for our many uncertainties.
- To update our emulators there is an alternative version of Bayesian statistics that is easier to specify and far easier to calculate with.
- Instead of Bayes Theorem we use the Bayes linear update:

$$\begin{aligned} \mathrm{E}_{D_i}(f_i(x)) &= \mathrm{E}(f_i(x)) + \mathrm{Cov}(f_i(x), D_i)\mathrm{Var}(D_i)^{-1}(D_i - \mathrm{E}(D_i)) \\ \mathrm{Var}_{D_i}(f_i(x)) &= \mathrm{Var}(f_i(x)) - \mathrm{Cov}(f_i(x), D_i)\mathrm{Var}(D_i)^{-1}\mathrm{Cov}(D_i, f_i(x)) \end{aligned}$$

  where $\mathrm{E}_{D_i}(f_i(x))$ and $\mathrm{Var}_{D_i}(f_i(x))$ are the Bayes Linear adjusted expectation and variance for $f_i(x)$ at new input point x, and are all that are needed for the subsequent implausibility measures and history match.

- We perform an initial wave 1 set of n runs at input locations $x^{(1)}, x^{(2)}, \ldots, x^{(n)}$ giving a column vector of model output values

$$D_i = (f_i(x^{(1)}), f_i(x^{(2)}), \ldots, f_i(x^{(n)}))^T$$

- Perhaps we are only prepared to specify expectations, variances and covariances for our many uncertainties.

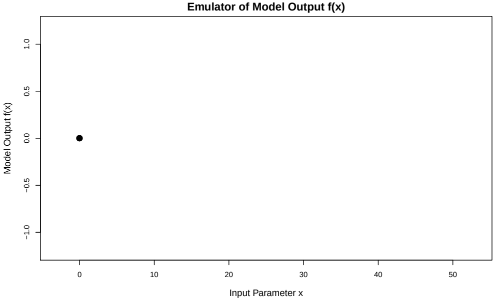
- To update our emulators there is an alternative version of Bayesian statistics that is easier to specify and far easier to calculate with.

- Instead of Bayes Theorem we use the Bayes linear update:

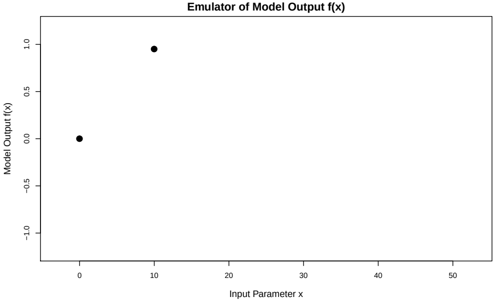
$$\begin{aligned} \mathrm{E}_{D_i}(f_i(x)) &= \mathrm{E}(f_i(x)) + \mathrm{Cov}(f_i(x), D_i)\mathrm{Var}(D_i)^{-1}(D_i - \mathrm{E}(D_i)) \\ \mathrm{Var}_{D_i}(f_i(x)) &= \mathrm{Var}(f_i(x)) - \mathrm{Cov}(f_i(x), D_i)\mathrm{Var}(D_i)^{-1}\mathrm{Cov}(D_i, f_i(x)) \end{aligned}$$

where $\mathrm{E}_{D_i}(f_i(x))$ and $\mathrm{Var}_{D_i}(f_i(x))$ are the Bayes Linear adjusted expectation and variance for $f_i(x)$ at new input point x, and are all that are needed for the subsequent implausibility measures and history match.

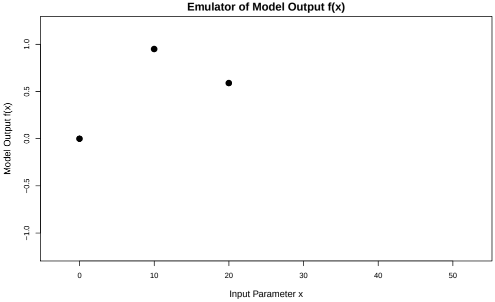
Emulation: a 1D Example



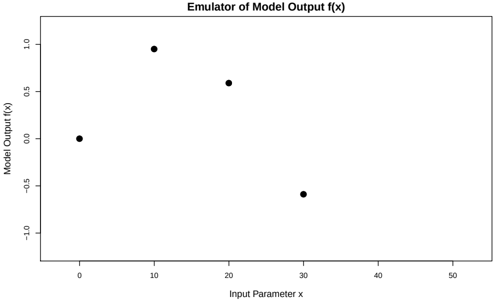
Emulation: a 1D Example



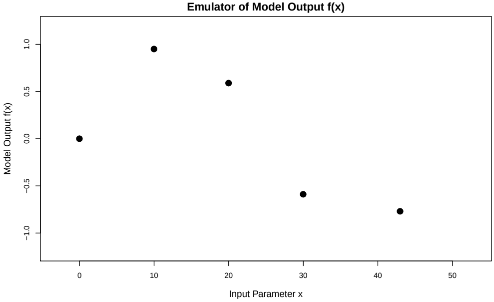
Emulation: a 1D Example



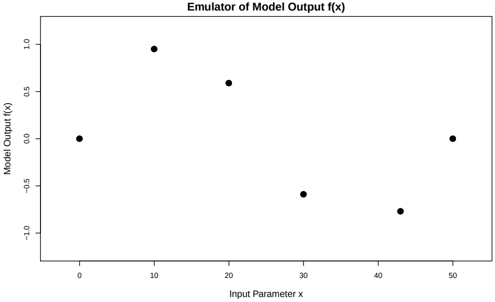
Emulation: a 1D Example



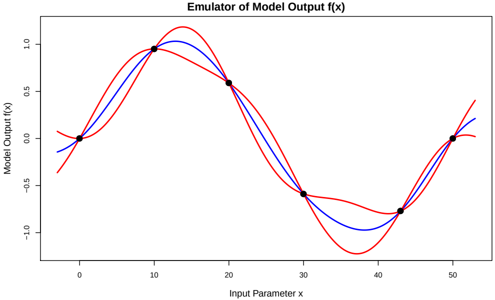
Emulation: a 1D Example



Emulation: a 1D Example



Emulation: a 1D Example



Multilevel Emulation

- Once we have constructed the emulator for level 1, we can use it to construct a highly informed prior for the level 2 emulator.
- We have for a univariate emulator at level 1, dropping the i index for simplicity so that $f_i^{(1)}(x) \to f^{(1)}(x)$:

$$f^{(1)}(x) = \sum_j \beta_j^{(1)} g_j(x_A) + u^{(1)}(x_A) + v^{(1)}(x)$$

- and similarly for level 2:

$$f^{(2)}(x) = \sum_j \beta_j^{(2)} g_j(x_A) + u^{(2)}(x_A) + v^{(2)}(x)$$

- We link $\beta_j^{(2)}$ to $\beta_j^{(1)}$ via:

$$\beta_j^{(2)} = a_j \beta_j^{(1)} + b_j$$

  with $a_j, b_j, \beta_j^{(1)}$ uncorrelated, and give a simple Bayes Linear specification:

$$\mathrm{E}[a_j] = 1, \quad \mathrm{Cov}[a_j, a_k] = \sigma_{a_j}^2 \delta_{jk}$$

$$\mathrm{E}[b_j] = 0, \quad \mathrm{Cov}[b_j, b_k] = \sigma_{b_j}^2 \delta_{jk}$$

- So the a_j describe a multiplicative uncertainty, and the b_j an uncertain offset.

Multilevel Emulation

- Once we have constructed the emulator for level 1, we can use it to construct a highly informed prior for the level 2 emulator.
- We have for a univariate emulator at level 1, dropping the i index for simplicity so that $f_i^{(1)}(x) \to f^{(1)}(x)$:

$$f^{(1)}(x) = \sum_j \beta_j^{(1)} g_j(x_A) + u^{(1)}(x_A) + v^{(1)}(x)$$

- and similarly for level 2:

$$f^{(2)}(x) = \sum_j \beta_j^{(2)} g_j(x_A) + u^{(2)}(x_A) + v^{(2)}(x)$$

- We link $\beta_j^{(2)}$ to $\beta_j^{(1)}$ via:

$$\beta_j^{(2)} = a_j \beta_j^{(1)} + b_j$$

with $a_j, b_j, \beta_j^{(1)}$ uncorrelated, and give a simple Bayes Linear specification:

$$\mathrm{E}[a_j] = 1, \quad \mathrm{Cov}[a_j, a_k] = \sigma_{a_j}^2 \delta_{jk}$$

$$\mathrm{E}[b_j] = 0, \quad \mathrm{Cov}[b_j, b_k] = \sigma_{b_j}^2 \delta_{jk}$$

- So the a_j describe a multiplicative uncertainty, and the b_j an uncertain offset.

- Once we have constructed the emulator for level 1, we can use it to construct a highly informed prior for the level 2 emulator.
- We have for a univariate emulator at level 1, dropping the i index for simplicity so that $f_i^{(1)}(x) \to f^{(1)}(x)$:

$$f^{(1)}(x) = \sum_j \beta_j^{(1)} g_j(x_A) + u^{(1)}(x_A) + v^{(1)}(x)$$

- and similarly for level 2:

$$f^{(2)}(x) = \sum_j \beta_j^{(2)} g_j(x_A) + u^{(2)}(x_A) + v^{(2)}(x)$$

- We link $\beta_j^{(2)}$ to $\beta_j^{(1)}$ via:

$$\beta_j^{(2)} = a_j \beta_j^{(1)} + b_j$$

with $a_j, b_j, \beta_j^{(1)}$ uncorrelated, and give a simple Bayes Linear specification:

$$\mathrm{E}[a_j] = 1, \quad \mathrm{Cov}[a_j, a_k] = \sigma_{a_j}^2 \delta_{jk}$$

$$\mathrm{E}[b_j] = 0, \quad \mathrm{Cov}[b_j, b_k] = \sigma_{b_j}^2 \delta_{jk}$$

- So the a_j describe a multiplicative uncertainty, and the b_j an uncertain offset.

Multilevel Emulation

- Once we have constructed the emulator for level 1, we can use it to construct a highly informed prior for the level 2 emulator.

- We have for a univariate emulator at level 1, dropping the i index for simplicity so that $f_i^{(1)}(x) \to f^{(1)}(x)$:

$$f^{(1)}(x) = \sum_j \beta_j^{(1)} \, g_j(x_A) + u^{(1)}(x_A) + v^{(1)}(x)$$

- and similarly for level 2:

$$f^{(2)}(x) = \sum_j \beta_j^{(2)} \, g_j(x_A) + u^{(2)}(x_A) + v^{(2)}(x)$$

- We link $\beta_j^{(2)}$ to $\beta_j^{(1)}$ via:

$$\beta_j^{(2)} = a_j \beta_j^{(1)} + b_j$$

  with $a_j, b_j, \beta_j^{(1)}$ uncorrelated, and give a simple Bayes Linear specification:

$$\mathrm{E}[a_j] = 1, \quad \mathrm{Cov}[a_j, a_k] = \sigma_{a_j}^2 \delta_{jk}$$

$$\mathrm{E}[b_j] = 0, \quad \mathrm{Cov}[b_j, b_k] = \sigma_{b_j}^2 \delta_{jk}$$

- So the a_j describe a multiplicative uncertainty, and the b_j an uncertain offset.

Multilevel Emulation

- Once we have constructed the emulator for level 1, we can use it to construct a highly informed prior for the level 2 emulator.
- We have for a univariate emulator at level 1, dropping the i index for simplicity so that $f_i^{(1)}(x) \to f^{(1)}(x)$:

$$f^{(1)}(x) = \sum_j \beta_j^{(1)} g_j(x_A) + u^{(1)}(x_A) + v^{(1)}(x)$$

- and similarly for level 2:

$$f^{(2)}(x) = \sum_j \beta_j^{(2)} g_j(x_A) + u^{(2)}(x_A) + v^{(2)}(x)$$

- We link $\beta_j^{(2)}$ to $\beta_j^{(1)}$ via:

$$\beta_j^{(2)} = a_j \beta_j^{(1)} + b_j$$

with $a_j, b_j, \beta_j^{(1)}$ uncorrelated, and give a simple Bayes Linear specification:

$$\mathrm{E}[a_j] = 1, \quad \mathrm{Cov}[a_j, a_k] = \sigma^2_{a_j} \delta_{jk}$$

$$\mathrm{E}[b_j] = 0, \quad \mathrm{Cov}[b_j, b_k] = \sigma^2_{b_j} \delta_{jk}$$

- So the a_j describe a multiplicative uncertainty, and the b_j an uncertain offset.

- Therefore the expectation and covariance of $\beta^{(2)}$ becomes

$$\begin{aligned} \mathrm{E}[\beta_j^{(2)}] &= \mathrm{E}[\beta_j^{(1)}] \\ \mathrm{Cov}[\beta_j^{(2)}, \beta_k^{(2)}] &= \mathrm{Cov}[\beta_j^{(1)}, \beta_k^{(1)}] + \sigma_{a_j}^2 \delta_{jk} (\mathrm{Var}[\beta_j^{(1)}] + \mathrm{E}[\beta_j^{(1)}]^2) + \sigma_{b_j}^2 \delta_{jk} \end{aligned}$$

- We also link $u^{(2)}(x_A)$ to $u^{(1)}(x_A)$ via

$$u^{(2)}(x_A) = u^{(1)}(x_A) + u^{(2/1)}(x_A)$$

where $u^{(1)}(x_A)$ and $u^{(2/1)}(x_A)$ are uncorrelated and $u^{(2/1)}(x_A)$ has zero mean and covariance structure

$$\mathrm{Cov}[u^{(2/1)}(x_A), u^{(2/1)}(x'_A)] = \sigma^2_{u^{(2/1)}} r^{(2)}_{\theta_2}(x_A - x'_A)$$

where $\sigma^2_{u^{(2/1)}} = \gamma \sigma^2_{u^{(1)}}$, and γ and θ_2 are to be specified.

- Therefore the expectation and covariance of $\beta^{(2)}$ becomes

$$\begin{aligned}
\mathrm{E}[\beta_j^{(2)}] &= \mathrm{E}[\beta_j^{(1)}] \\
\mathrm{Cov}[\beta_j^{(2)}, \beta_k^{(2)}] &= \mathrm{Cov}[\beta_j^{(1)}, \beta_k^{(1)}] + \sigma_{a_j}^2 \delta_{jk} (\mathrm{Var}[\beta_j^{(1)}] + \mathrm{E}[\beta_j^{(1)}]^2) + \sigma_{b_j}^2 \delta_{jk}
\end{aligned}$$

- We also link $u^{(2)}(x_A)$ to $u^{(1)}(x_A)$ via

$$u^{(2)}(x_A) = u^{(1)}(x_A) + u^{(2/1)}(x_A)$$

where $u^{(1)}(x_A)$ and $u^{(2/1)}(x_A)$ are uncorrelated and $u^{(2/1)}(x_A)$ has zero mean and covariance structure

$$\mathrm{Cov}[u^{(2/1)}(x_A), u^{(2/1)}(x_A')] = \sigma^2_{u^{(2/1)}} r^{(2)}_{\theta_2}(x_A - x_A')$$

where $\sigma^2_{u^{(2/1)}} = \gamma \sigma^2_{u^{(1)}}$, and γ and θ_2 are to be specified.

- Finally, we decompose the nugget $v_i^{(1)}(x)$ into two uncorrelated pieces:

$$v^{(1)}(x) = v_I^{(1)}(x) + v_S^{(1)}(x)$$

where $v_I^{(1)}(x)$ represents the inactive variables and $v_S^{(1)}(x)$ the stochasticity due to finite galaxy counts. We have that

$$\mathrm{Cov}[v^{(1)}(x), v^{(1)}(x')] = \sigma^2_{v^{(1)}}\delta_{xx'} = \left(\sigma^2_{v_I^{(1)}} + \sigma^2_{v_S^{(1)}}\right)\delta_{xx'}$$

- Similarly we have for the level 2 nugget:

$$v^{(2)}(x) = v_I^{(2)}(x) + v_S^{(2)}(x)$$

and make the judgement that $\sigma^2_{v_I^{(2)}} \simeq \sigma^2_{v_I^{(1)}}$ but that

$$\sigma^2_{v_S^{(2)}} = \frac{V_1}{V_2}\sigma^2_{v_S^{(1)}}$$

where V_i are the volumes of the EAGLE simulation at level i.

Multilevel Emulation

- Finally, we decompose the nugget $v_i^{(1)}(x)$ into two uncorrelated pieces:

$$v^{(1)}(x) = v_I^{(1)}(x) + v_S^{(1)}(x)$$

where $v_I^{(1)}(x)$ represents the inactive variables and $v_S^{(1)}(x)$ the stochasticity due to finite galaxy counts. We have that

$$\text{Cov}[v^{(1)}(x), v^{(1)}(x')] = \sigma^2_{v^{(1)}} \delta_{xx'} = \left(\sigma^2_{v_I^{(1)}} + \sigma^2_{v_S^{(1)}} \right) \delta_{xx'}$$

- Similarly we have for the level 2 nugget:

$$v^{(2)}(x) = v_I^{(2)}(x) + v_S^{(2)}(x)$$

and make the judgement that $\sigma^2_{v_I^{(2)}} \simeq \sigma^2_{v_I^{(1)}}$ but that

$$\sigma^2_{v_S^{(2)}} = \frac{V_1}{V_2} \sigma^2_{v_S^{(1)}}$$

where V_i are the volumes of the EAGLE simulation at level i.

Multilevel Emulation

- We now have all the pieces needed to construct the prior for the level 2 emulator.
- We can now update this emulator by the set of 20 level 2 runs.
- We can construct priors for and update the level 3 and 4 emulators similarly.
- The multilevel emulation structure now allows the incorporation of detailed judgements as to structural differences between the levels.
- Levels 2 and 3 have known and important differences which we have tried to capture.
- We can propose informative designs for the level 3 and 4 runs based on detailed priors.
- But now back to History Matching to the observed data.

Multilevel Emulation

- We now have all the pieces needed to construct the prior for the level 2 emulator.
- We can now update this emulator by the set of 20 level 2 runs.
- We can construct priors for and update the level 3 and 4 emulators similarly.
- The multilevel emulation structure now allows the incorporation of detailed judgements as to structural differences between the levels.
- Levels 2 and 3 have known and important differences which we have tried to capture.
- We can propose informative designs for the level 3 and 4 runs based on detailed priors.
- But now back to History Matching to the observed data.

Multilevel Emulation

- We now have all the pieces needed to construct the prior for the level 2 emulator.
- We can now update this emulator by the set of 20 level 2 runs.
- We can construct priors for and update the level 3 and 4 emulators similarly.
- The multilevel emulation structure now allows the incorporation of detailed judgements as to structural differences between the levels.
- Levels 2 and 3 have known and important differences which we have tried to capture.
- We can propose informative designs for the level 3 and 4 runs based on detailed priors.
- But now back to History Matching to the observed data.

- We now have all the pieces needed to construct the prior for the level 2 emulator.
- We can now update this emulator by the set of 20 level 2 runs.
- We can construct priors for and update the level 3 and 4 emulators similarly.
- The multilevel emulation structure now allows the incorporation of detailed judgements as to structural differences between the levels.
- Levels 2 and 3 have known and important differences which we have tried to capture.
- We can propose informative designs for the level 3 and 4 runs based on detailed priors.
- But now back to History Matching to the observed data.

Multilevel Emulation

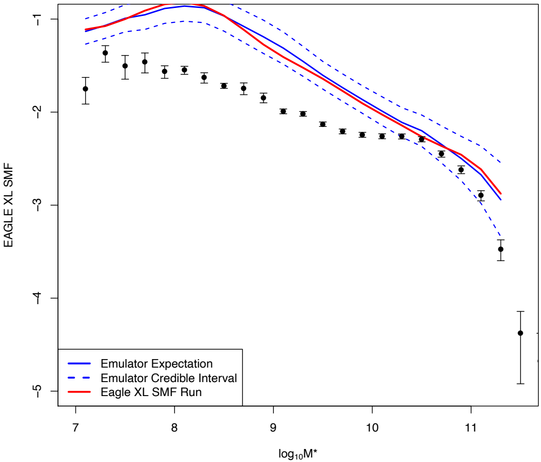
- We now have all the pieces needed to construct the prior for the level 2 emulator.
- We can now update this emulator by the set of 20 level 2 runs.
- We can construct priors for and update the level 3 and 4 emulators similarly.
- The multilevel emulation structure now allows the incorporation of detailed judgements as to structural differences between the levels.
- Levels 2 and 3 have known and important differences which we have tried to capture.
- We can propose informative designs for the level 3 and 4 runs based on detailed priors.
- But now back to History Matching to the observed data.

- We now have all the pieces needed to construct the prior for the level 2 emulator.
- We can now update this emulator by the set of 20 level 2 runs.
- We can construct priors for and update the level 3 and 4 emulators similarly.
- The multilevel emulation structure now allows the incorporation of detailed judgements as to structural differences between the levels.
- Levels 2 and 3 have known and important differences which we have tried to capture.
- We can propose informative designs for the level 3 and 4 runs based on detailed priors.
- But now back to History Matching to the observed data.

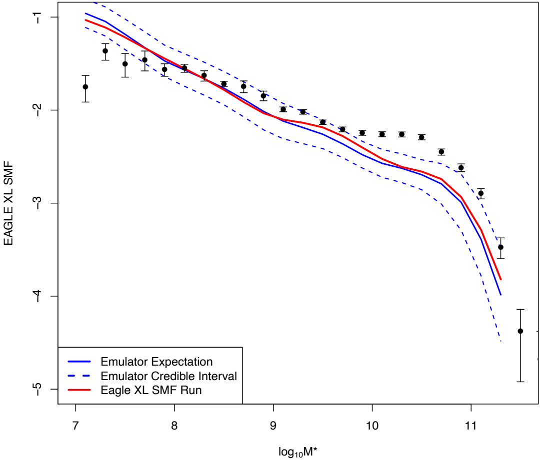
Multilevel Emulation

- We now have all the pieces needed to construct the prior for the level 2 emulator.
- We can now update this emulator by the set of 20 level 2 runs.
- We can construct priors for and update the level 3 and 4 emulators similarly.
- The multilevel emulation structure now allows the incorporation of detailed judgements as to structural differences between the levels.
- Levels 2 and 3 have known and important differences which we have tried to capture.
- We can propose informative designs for the level 3 and 4 runs based on detailed priors.
- But now back to History Matching to the observed data.

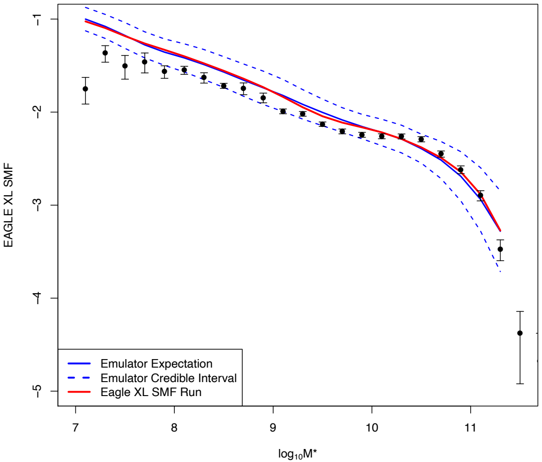
SMF Level 2 Diagnostics



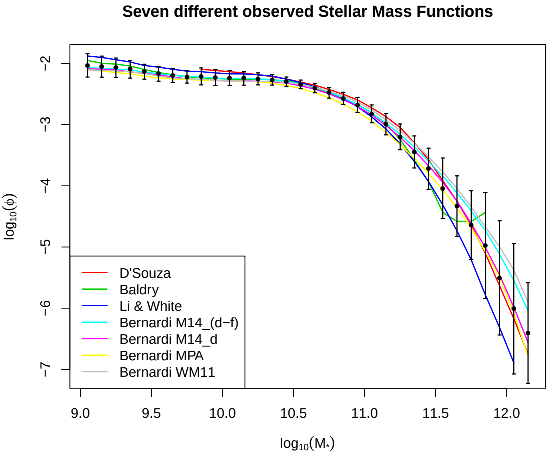
SMF Level 2 Diagnostics



SMF Level 2 Diagnostics



Observation Errors: Stellar Mass Function (Details)



- Often simulations are compared to the most recent SMF. But this is ‘theory laden’ data, which often under reports systematic errors.
- To counter this we gathered the 7 major data sets and amalgamated them.

Model Discrepancy (Details)

- Multiple components of the structural model discrepancy $\epsilon^{(4)}$ identified and assessed
- For the Stellar Mass Function:
  - Zero point discrepancy: correlation horizontal error over all outputs.
  - Normalisation mass discrepancy: overall correlated vertical error.
- For the Galaxy Sizes:
  - Light/mass size definition discrepancy
  - Projection reconstruction discrepancy
  - Different colour band sizes discrepancy
- In addition, the cosmologists have performed one good run, called the reference run, at all 4 levels on a previous EAGLE version.
- So a small external discrepancy term is added to $\mathrm{Var}[\epsilon^{(4)}]$ to ensure the reference run is always within 3 sigma of the observed data: ensures we will search for inputs comparable to or better than before.

- First identify set of outputs $i \in Q_j$ that are good to emulate.
- We can now calculate the **Implausibility** $I_{(i)}(x)$ at any input parameter point x for each of the $i \in Q_j$ good outputs. This is given by:

$$I^2_{(i)}(x) = \frac{|\mathrm{E}_{D_i}(f_i^{(4)}(x)) - z_i|^2}{(\mathrm{Var}_{D_i}(f_i^{(4)}(x)) + \mathrm{Var}[\epsilon_i^{(4)}] + \mathrm{Var}[e_i])}$$

- $\mathrm{E}_{D_i}(f_i^{(4)}(x))$ and $\mathrm{Var}_{D_i}(f_i^{(4)}(x))$ are the emulator expectation and variance at the 4th level (or at whatever level we wish to work with).
- z_i are the observed data and $\mathrm{Var}[\epsilon_i^{(4)}]$ and $\mathrm{Var}[e_i]$ are the (univariate) Model Discrepancy and Observational Error variances.
- **Large values** of $I_{(i)}(x)$ imply that we are **highly unlikely to obtain acceptable matches between model output and observed data at input** x**.**
- **Small values** of $I_{(i)}(x)$ **do not** imply that x is good!

Implausibility Measures (Univariate)

- First identify set of outputs $i \in Q_j$ that are good to emulate.
- We can now calculate the **Implausibility** $I_{(i)}(x)$ at any input parameter point x for each of the $i \in Q_j$ good outputs. This is given by:

$$I^2_{(i)}(x) = \frac{|\mathrm{E}_{D_i}(f_i^{(4)}(x)) - z_i|^2}{(\mathrm{Var}_{D_i}(f_i^{(4)}(x)) + \mathrm{Var}[\epsilon_i^{(4)}] + \mathrm{Var}[e_i])}$$

- $\mathrm{E}_{D_i}(f_i^{(4)}(x))$ and $\mathrm{Var}_{D_i}(f_i^{(4)}(x))$ are the emulator expectation and variance at the 4th level (or at whatever level we wish to work with).
- z_i are the observed data and $\mathrm{Var}[\epsilon_i^{(4)}]$ and $\mathrm{Var}[e_i]$ are the (univariate) Model Discrepancy and Observational Error variances.
- **Large values** of $I_{(i)}(x)$ imply that we are **highly unlikely to obtain acceptable matches between model output and observed data at input** x**.**
- **Small values** of $I_{(i)}(x)$ **do not** imply that x is good!

Implausibility Measures (Univariate)

- First identify set of outputs $i \in Q_j$ that are good to emulate.
- We can now calculate the **Implausibility** $I_{(i)}(x)$ at any input parameter point x for each of the $i \in Q_j$ good outputs. This is given by:

$$I^2_{(i)}(x) = \frac{|\mathrm{E}_{D_i}(f_i^{(4)}(x)) - z_i|^2}{(\mathrm{Var}_{D_i}(f_i^{(4)}(x)) + \mathrm{Var}[\epsilon_i^{(4)}] + \mathrm{Var}[e_i])}$$

- $\mathrm{E}_{D_i}(f_i^{(4)}(x))$ and $\mathrm{Var}_{D_i}(f_i^{(4)}(x))$ are the emulator expectation and variance at the 4th level (or at whatever level we wish to work with).
- z_i are the observed data and $\mathrm{Var}[\epsilon_i^{(4)}]$ and $\mathrm{Var}[e_i]$ are the (univariate) Model Discrepancy and Observational Error variances.
- **Large values** of $I_{(i)}(x)$ imply that we are **highly unlikely to obtain acceptable matches between model output and observed data at input** x**.**
- **Small values** of $I_{(i)}(x)$ **do not** imply that x is good!

- First identify set of outputs $i \in Q_j$ that are good to emulate.
- We can now calculate the **Implausibility** $I_{(i)}(x)$ at any input parameter point x for each of the $i \in Q_j$ good outputs. This is given by:

$$I^2_{(i)}(x) = \frac{|\mathrm{E}_{D_i}(f_i^{(4)}(x)) - z_i|^2}{(\mathrm{Var}_{D_i}(f_i^{(4)}(x)) + \mathrm{Var}[\epsilon_i^{(4)}] + \mathrm{Var}[e_i])}$$

- $\mathrm{E}_{D_i}(f_i^{(4)}(x))$ and $\mathrm{Var}_{D_i}(f_i^{(4)}(x))$ are the emulator expectation and variance at the 4th level (or at whatever level we wish to work with).
- z_i are the observed data and $\mathrm{Var}[\epsilon_i^{(4)}]$ and $\mathrm{Var}[e_i]$ are the (univariate) Model Discrepancy and Observational Error variances.
- **Large values** of $I_{(i)}(x)$ imply that we are **highly unlikely to obtain acceptable matches between model output and observed data at input** x.
- **Small values** of $I_{(i)}(x)$ **do not** imply that x is good!

Implausibility Measures (Univariate)

- First identify set of outputs $i \in Q_j$ that are good to emulate.
- We can now calculate the **Implausibility** $I_{(i)}(x)$ at any input parameter point x for each of the $i \in Q_j$ good outputs. This is given by:

$$I_{(i)}^2(x) = \frac{|\mathrm{E}_{D_i}(f_i^{(4)}(x)) - z_i|^2}{(\mathrm{Var}_{D_i}(f_i^{(4)}(x)) + \mathrm{Var}[\epsilon_i^{(4)}] + \mathrm{Var}[e_i])}$$

- $\mathrm{E}_{D_i}(f_i^{(4)}(x))$ and $\mathrm{Var}_{D_i}(f_i^{(4)}(x))$ are the emulator expectation and variance at the 4th level (or at whatever level we wish to work with).
- z_i are the observed data and $\mathrm{Var}[\epsilon_i^{(4)}]$ and $\mathrm{Var}[e_i]$ are the (univariate) Model Discrepancy and Observational Error variances.
- **Large values** of $I_{(i)}(x)$ imply that we are **highly unlikely to obtain acceptable matches between model output and observed data at input** x**.**
- **Small values** of $I_{(i)}(x)$ **do not** imply that x is good!

Implausibility Measures (Univariate)

- First identify set of outputs $i \in Q_j$ that are good to emulate.
- We can now calculate the **Implausibility** $I_{(i)}(x)$ at any input parameter point x for each of the $i \in Q_j$ good outputs. This is given by:

$$I^2_{(i)}(x) = \frac{|\mathrm{E}_{D_i}(f_i^{(4)}(x)) - z_i|^2}{(\mathrm{Var}_{D_i}(f_i^{(4)}(x)) + \mathrm{Var}[\epsilon_i^{(4)}] + \mathrm{Var}[e_i])}$$

- $\mathrm{E}_{D_i}(f_i^{(4)}(x))$ and $\mathrm{Var}_{D_i}(f_i^{(4)}(x))$ are the emulator expectation and variance at the 4th level (or at whatever level we wish to work with).
- z_i are the observed data and $\mathrm{Var}[\epsilon_i^{(4)}]$ and $\mathrm{Var}[e_i]$ are the (univariate) Model Discrepancy and Observational Error variances.
- **Large values** of $I_{(i)}(x)$ imply that we are **highly unlikely to obtain acceptable matches between model output and observed data at input** x**.**
- **Small values** of $I_{(i)}(x)$ **do not** imply that x is good!

Implausibility Measures (Univariate)

- We can combine the univariate implausibilities across the outputs in Q_i by maximizing over outputs:

$$I_M(x) = \max_{i \in Q_j} I_{(i)}(x) \qquad (1)$$

- We can then impose a cutoff

$$I_M(x) < c_M \qquad (2)$$

  in order to discard regions of input parameter space x that we now deem to be implausible.

- The choice of cutoff c_M is often motivated by Pukelsheim's 3-sigma rule, which does not require precise distributions.

- We may simultaneously employ other choices of combined implausibility measure: e.g. multivariate implausibility, second maximum etc.

- Note that we still emulate the raw physical model output $f_i^{(4)}(x)$ and NOT the value of the combined measure.

Implausibility Measures (Univariate)

- We can combine the univariate implausibilities across the outputs in Q_i by maximizing over outputs:
$$I_M(x) = \max_{i \in Q_j} I_{(i)}(x) \tag{1}$$
- We can then impose a cutoff
$$I_M(x) < c_M \tag{2}$$
in order to discard regions of input parameter space x that we now deem to be implausible.
- The choice of cutoff c_M is often motivated by Pukelsheim's 3-sigma rule, which does not require precise distributions.
- We may simultaneously employ other choices of combined implausibility measure: e.g. multivariate implausibility, second maximum etc.
- Note that we still emulate the raw physical model output $f_i^{(4)}(x)$ and NOT the value of the combined measure.

Common problems & mistakes: One shot analysis (Details)

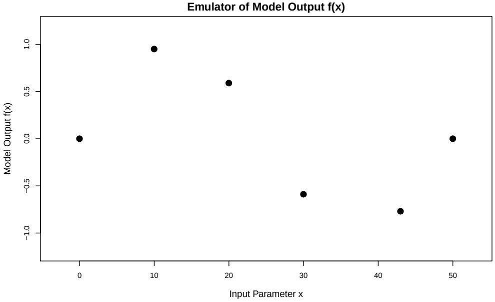
- Often the set of acceptable inputs $\mathcal{X}$ only occupies a tiny fraction of the original input space.
- Therefore we do not want to use a single one shot space filling design, as this would waste a lot of runs in implausible parts of the space.
- Instead we perform a series of iterations or waves, designing in ever smaller non-implausible regions of the input space (i.e. batch sequentially). Fairly obvious.
- However, we would also not want to use the same statistical form for the emulator across all waves, as the model will most likely behave very differently over the original input space $\mathcal{X}_1$ compared to $\mathcal{X}$ which may be a billion times smaller. Less obvious.
- Therefore we must fit emulators of possibly different structure and complexity at each iteration: to forget this is a mistake (it also has important implications for the full design calculation).
- This is even more important for the multilevel emulation case: we cannot hope to create accurate level 4 emulators over the whole input space.

Iterative History Matching for Reducing Input Space (Details)

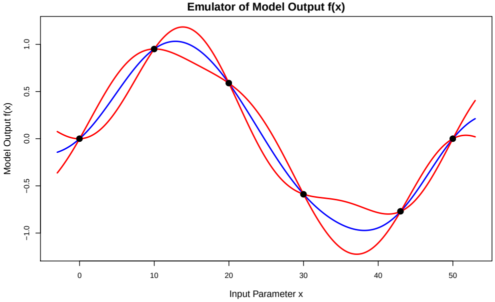
We use an iterative strategy to reduce the input parameter space. Denoting the current non-implausible volume by $\mathcal{X}_j$, at each stage or wave we:

1. Design and perform a set of runs over the non-implausible input region $\mathcal{X}_j$
2. Identify the set Q_{j+1} of informative outputs that we can emulate easily
3. Construct new emulators for $f_i(x)$, where $i \in Q_{j+1}$ defined only over $\mathcal{X}_j$
4. Evaluate the new implausibility functions $I_i(x), i \in Q_{j+1}$ only over $\mathcal{X}_j$
5. Define a new (reduced) non-implausible region $\mathcal{X}_{j+1}$, by $I_M(x) < c_M$, which should satisfy $\mathcal{X} \subset \mathcal{X}_{j+1} \subset \mathcal{X}_j$
6. Unless (a) the emulator variances are now small in comparison to the other sources of uncertainty (model discrepancy and observation errors) or (b) computational resources are exhausted or (c) all the input space is deemed implausible, return to step 1
7. If 6(a) true, generate a large number of acceptable runs from the final non-implausible volume $\mathcal{X}$

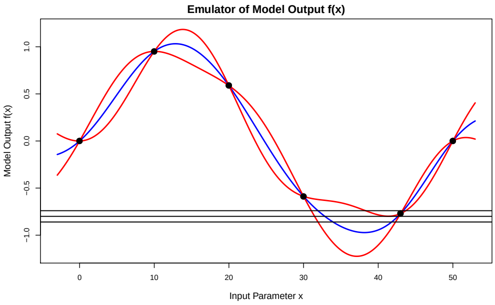
History Matching via Implausibility: a 1D Example



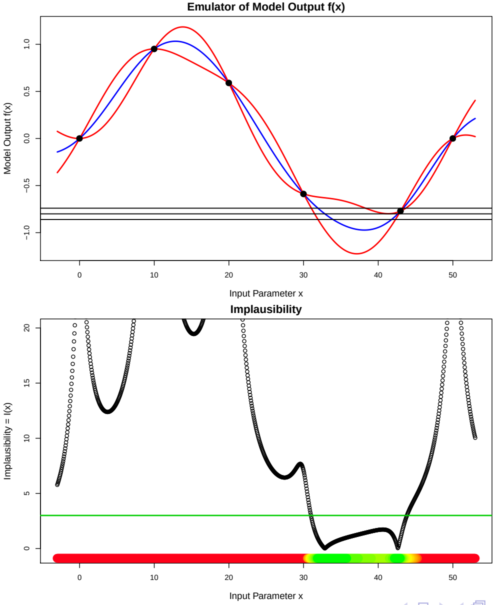
History Matching via Implausibility: a 1D Example



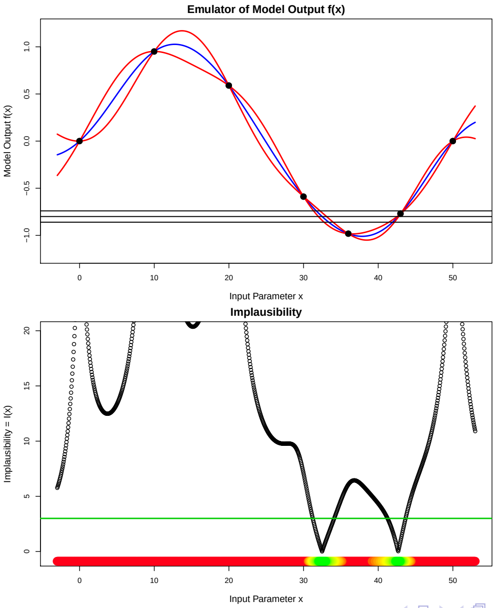
History Matching via Implausibility: a 1D Example



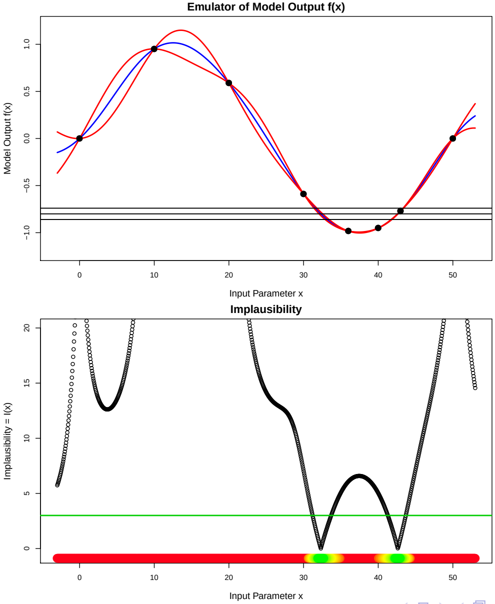
History Matching via Implausibility: a 1D Example



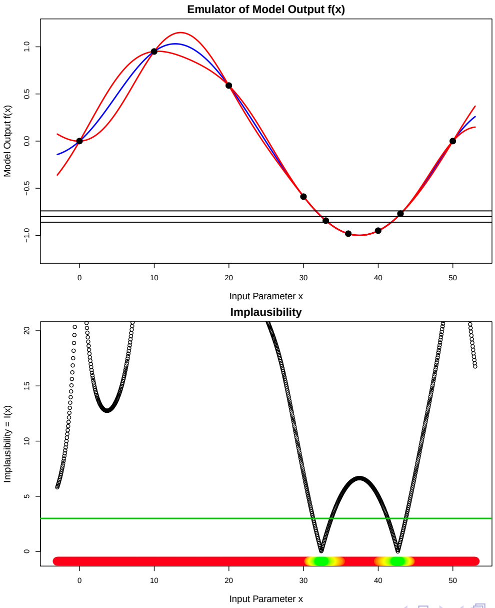
History Matching via Implausibility: a 1D Example



History Matching via Implausibility: a 1D Example



History Matching via Implausibility: a 1D Example



- Using the speed of the emulators, we can now blitz the input space by evaluating the implausibility

$$I_M(x) = \max_{i \in Q_j} I_{(i)}(x)$$

across a huge latin hypercube, where

$$I^2_{(i)}(x) = \frac{|\mathrm{E}_{D_i}(f_i(x)) - z_i|^2}{(\mathrm{Var}_{D_i}(f_i(x)) + \mathrm{Var}[\epsilon_i] + \mathrm{Var}[e_i])}$$

- To visualise this, we can project down into 2 dimensions, by minimising the implausibility.

$$I_P(x') \;=\; \min_{x''} I_M(x', x'')$$

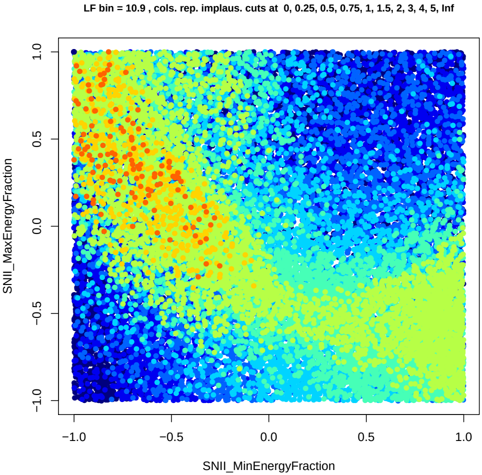
where x' is a 2 vector of the plotting variables, and x'' a 5 vector spanning the remaining inputs not in the plot.

Minimised Implausibility Plots

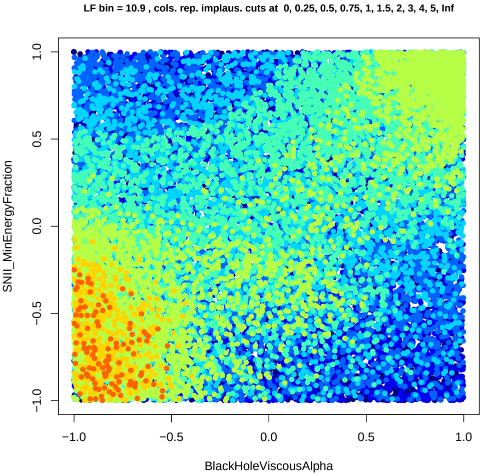
- Using the speed of the emulators, we can now blitz the input space by evaluating the implausibility
$$I_M(x) = \max_{i \in Q_j} I_{(i)}(x)$$
across a huge latin hypercube, where
$$I^2_{(i)}(x) = \frac{|\mathrm{E}_{D_i}(f_i(x)) - z_i|^2}{(\mathrm{Var}_{D_i}(f_i(x)) + \mathrm{Var}[\epsilon_i] + \mathrm{Var}[e_i])}$$

- To visualise this, we can project down into 2 dimensions, by minimising the implausibility.
$$I_P(x') \;=\; \min_{x''} I_M(x', x'')$$
where x' is a 2 vector of the plotting variables, and x'' a 5 vector spanning the remaining inputs not in the plot.

Minimised Implausibility Plots



Minimised Implausibility Plots



- Low implausibility at x can be due to the emulators predicting a good match at x, or just due to high emulator uncertainty there.
- We can examine which of these options is the case by plotting the zero emulator variance implausibility:

$$I_M(x) = \max_{i \in Q_j} I_{(i)}(x)$$

where now

$$I_{(i)}^2(x) = \frac{|\mathrm{E}_{D_i}(f_i(x)) - z_i|^2}{(\mathrm{Var}_{D_i}(f_i(x)) + \mathrm{Var}[\epsilon_i] + \mathrm{Var}[e_i])}$$

- We minimise the implausibility to obtain

$$I_P(x') \;=\; \min_{x''} I_M(x', x'')$$

as before.

Zero Emulator Variance Implausibility Plots

- Low implausibility at x can be due to the emulators predicting a good match at x, or just due to high emulator uncertainty there.

- We can examine which of these options is the case by plotting the zero emulator variance implausibility:

$$I_M(x) = \max_{i \in Q_j} I_{(i)}(x)$$

where now

$$I^2_{(i)}(x) = \frac{|\mathrm{E}_{D_i}(f_i(x)) - z_i|^2}{(\mathrm{Var}_{D_i}(f_i(x)) + \mathrm{Var}[\epsilon_i] + \mathrm{Var}[e_i])}$$

- We minimise the implausibility to obtain

$$I_P(x') \;=\; \min_{x''} I_M(x', x'')$$

as before.

Zero Emulator Variance Implausibility Plots

- Low implausibility at x can be due to the emulators predicting a good match at x, or just due to high emulator uncertainty there.

- We can examine which of these options is the case by plotting the zero emulator variance implausibility:

$$I_M(x) = \max_{i \in Q_j} I_{(i)}(x)$$

where now

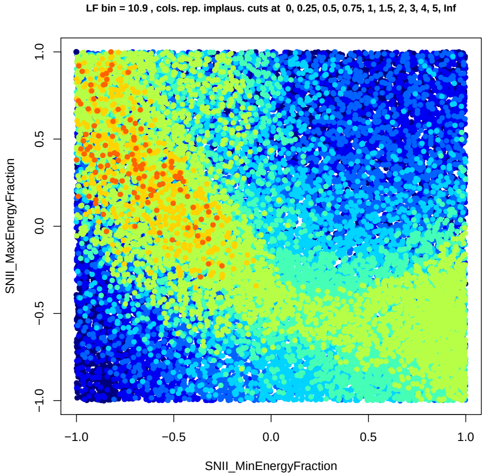
$$I_{(i)}^2(x) = \frac{|\mathrm{E}_{D_i}(f_i(x)) - z_i|^2}{(\mathrm{Var}[\epsilon_i] + \mathrm{Var}[e_i])}$$

- We minimise the implausibility to obtain

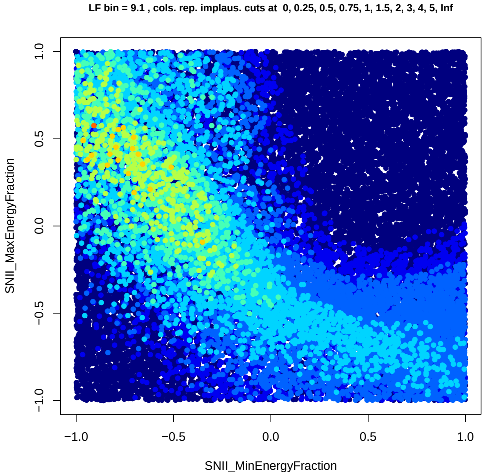
$$I_P(x') \;=\; \min_{x''} I_M(x', x'')$$

as before.

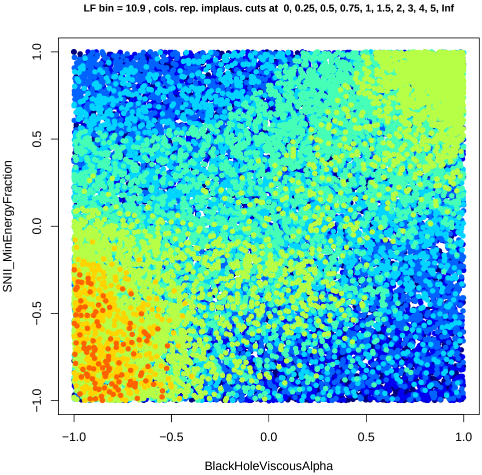
Minimised Implausibility Plots



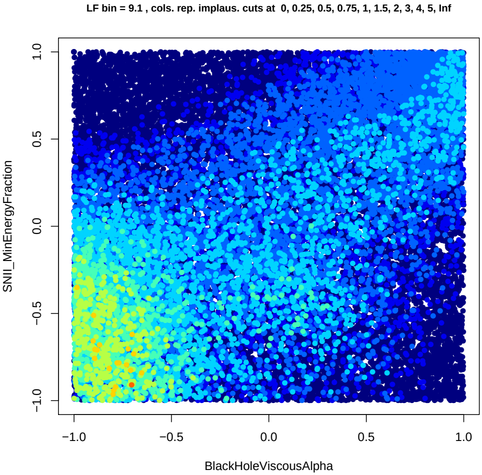
Zero Emulator Variance Implausibility Plots



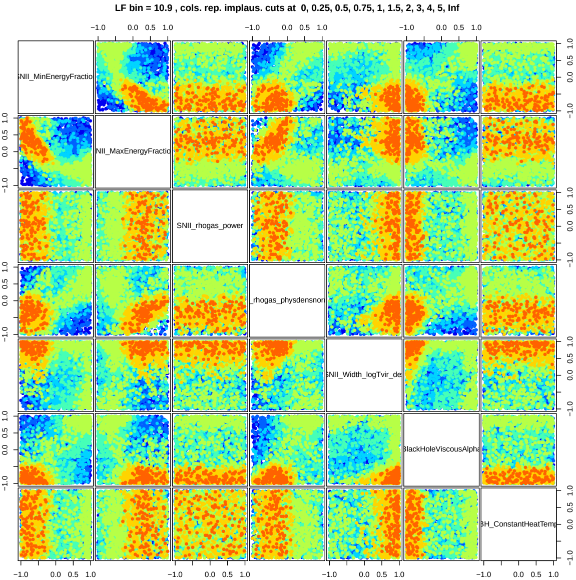
Minimised Implausibility Plots



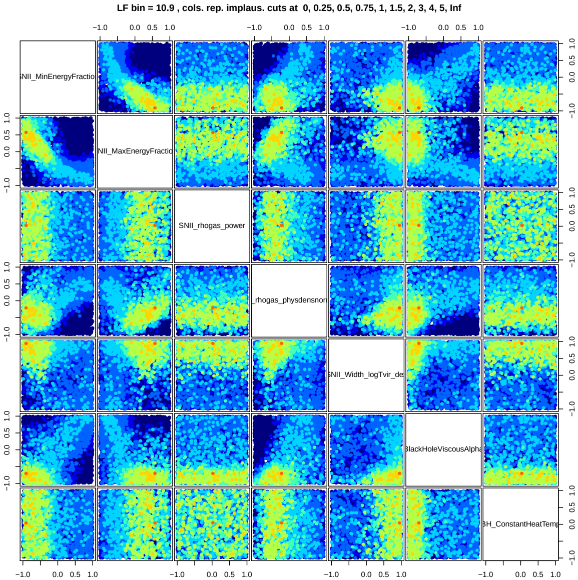
Zero Emulator Variance Implausibility Plots



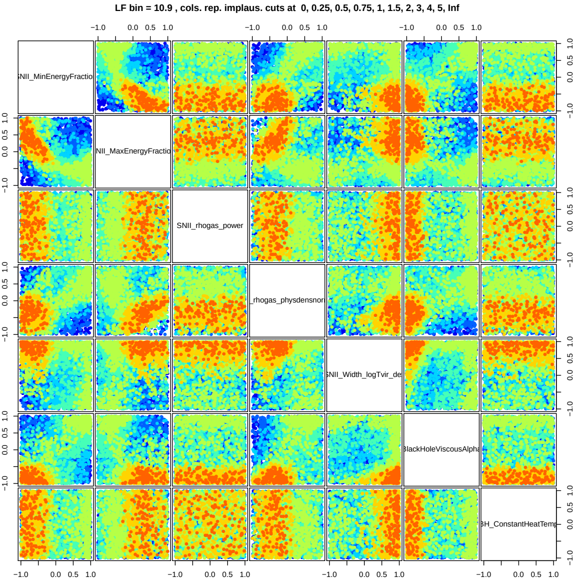
Results: Level 2, Minimised Implausibility



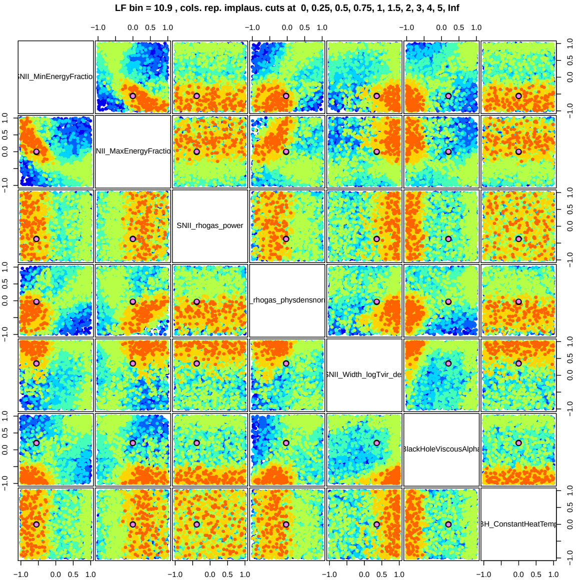
Results: Level 2, Zero Emulator Variance Implausibility



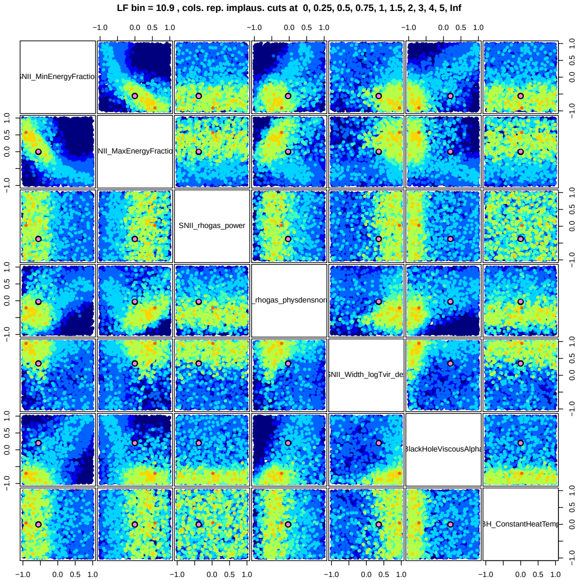
Results: Level 2, Minimised Implausibility



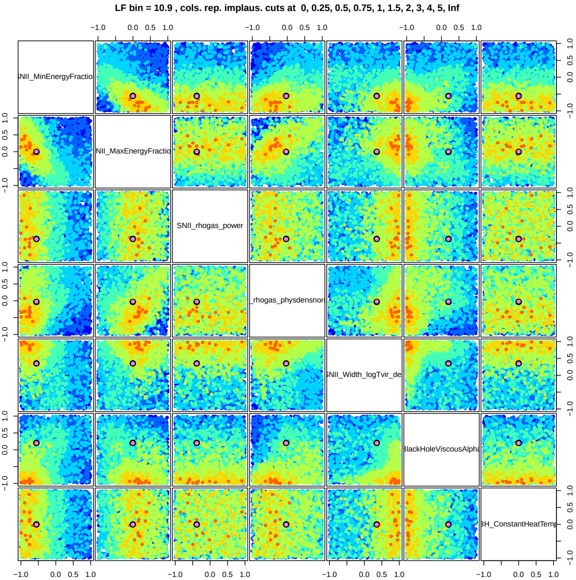
Results: Level 2, Minimised Implausibility, with Ref Run



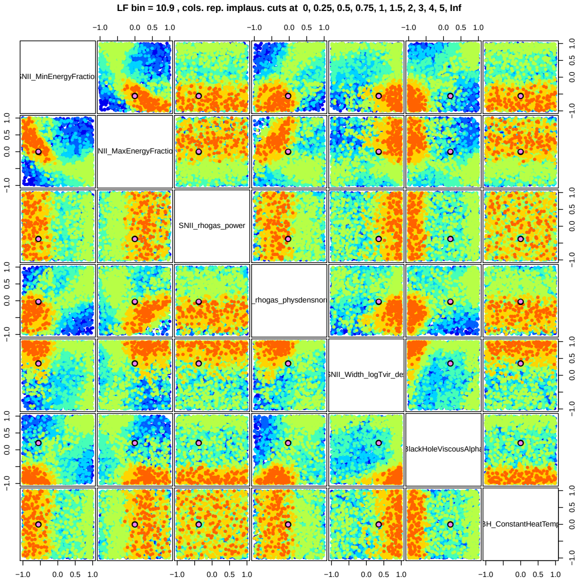
Results: Level 2, Zero Emulator Variance Implausibility, with Ref Run



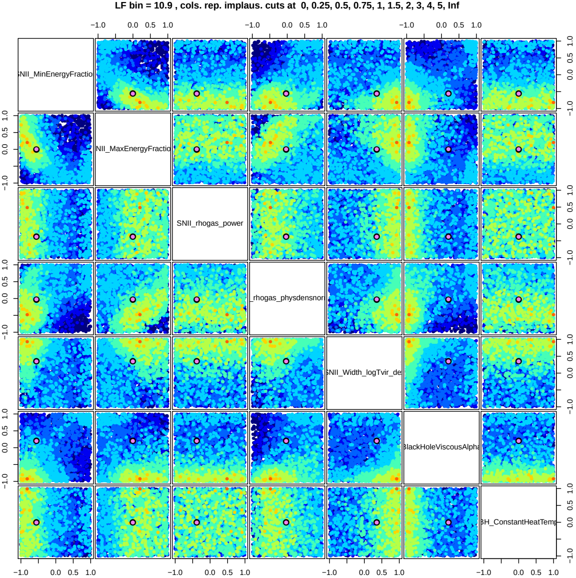
Results: Level 1, Minimised Implausibility, with Ref Run



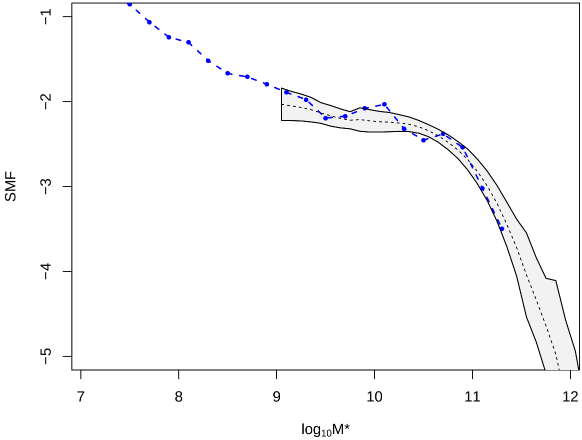
Results: Level 2, Minimised Implausibility, with Ref Run

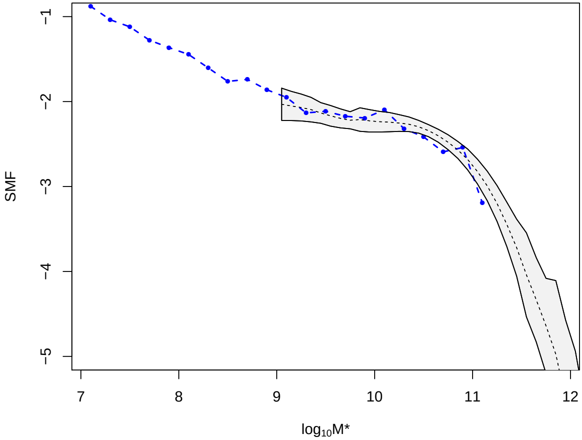


Results: Level 3, Minimised Implausibility, with Ref Run

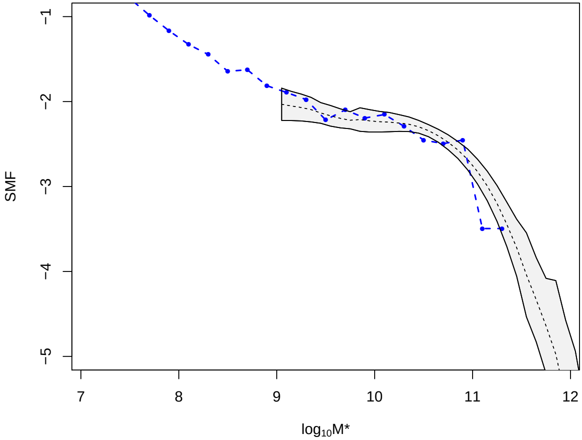


SMF: Wave 4 runs

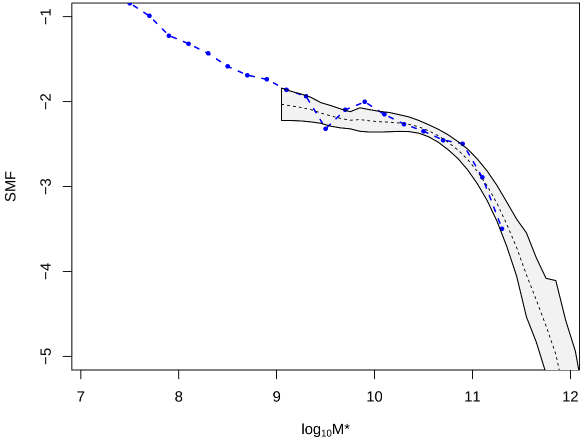



SMF
−1
−2
−3
−4
−5
7
8
9
10
11
12
$\log_{10}M^*$

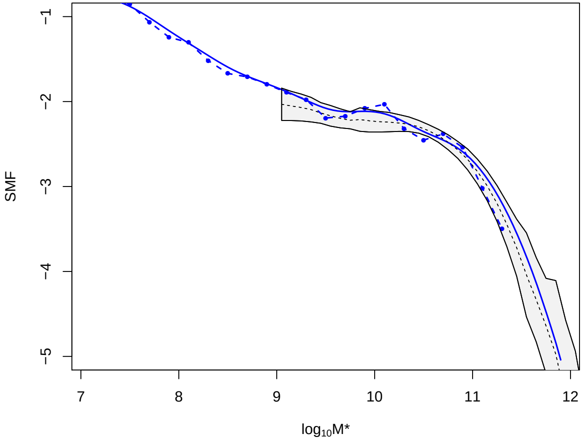
SMF: Wave 4 runs



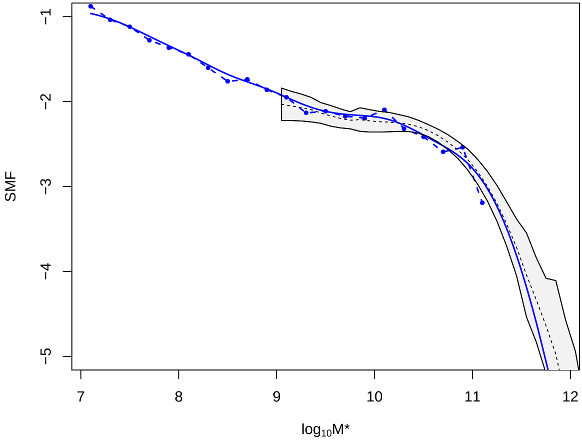
SMF: Wave 4 runs



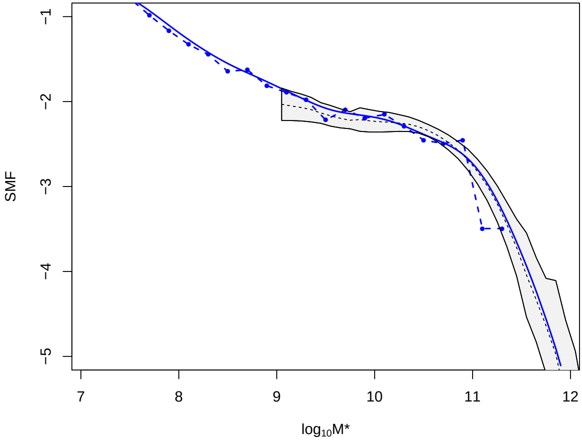
SMF: Wave 4 runs



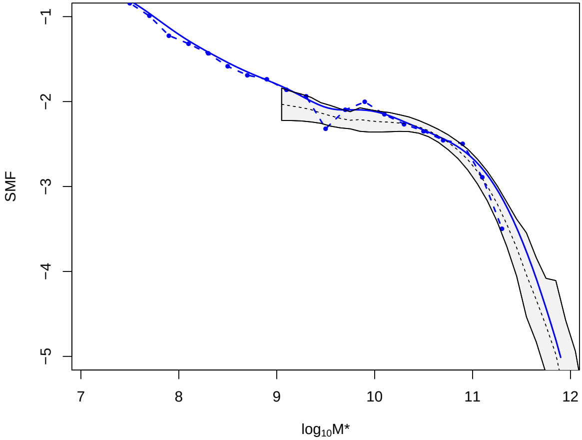
SMF: Wave 4 runs



SMF: Wave 4 runs



SMF: Wave 4 runs



Conclusion

- We have constructed a multilevel emulator for the EAGLE simulation.
- We have emulated at levels 1, 2 and 3 and history matched to rule out bad parts of the input space.
- Assessed multiple contributions to model discrepancy and observation errors.
- Have found the region containing acceptable fits to the SMF.
- Multilevel emulation allows us to tame such expensive models.

Conclusion

- We have constructed a multilevel emulator for the EAGLE simulation.
- We have emulated at levels 1, 2 and 3 and history matched to rule out bad parts of the input space.
- Assessed multiple contributions to model discrepancy and observation errors.
- Have found the region containing acceptable fits to the SMF.
- Multilevel emulation allows us to tame such expensive models.

Conclusion

- We have constructed a multilevel emulator for the EAGLE simulation.
- We have emulated at levels 1, 2 and 3 and history matched to rule out bad parts of the input space.
- Assessed multiple contributions to model discrepancy and observation errors.
- Have found the region containing acceptable fits to the SMF.
- Multilevel emulation allows us to tame such expensive models.

Conclusion

- We have constructed a multilevel emulator for the EAGLE simulation.
- We have emulated at levels 1, 2 and 3 and history matched to rule out bad parts of the input space.
- Assessed multiple contributions to model discrepancy and observation errors.
- Have found the region containing acceptable fits to the SMF.
- Multilevel emulation allows us to tame such expensive models.

Conclusion

- We have constructed a multilevel emulator for the EAGLE simulation.
- We have emulated at levels 1, 2 and 3 and history matched to rule out bad parts of the input space.
- Assessed multiple contributions to model discrepancy and observation errors.
- Have found the region containing acceptable fits to the SMF.
- Multilevel emulation allows us to tame such expensive models.

References

Rodrigues, L.F.S., Vernon, I., Bower, R.G.: *Constraints to galaxy formation models using the galaxy stellar mass function, stronger feedback during starbursts?*, MNRAS (2017) 466 (2): 2418-2435. arXiv:1609.06922v3

Vernon, I.; Goldstein, M.; Bower, R. G.; Galaxy Formation: “*Bayesian History Matching for the Observable Universe*". *Statistical Science* 29 (2014), no. 1, 81–90.

Vernon, I., Goldstein, M., and Bower, R. G. (2010), “*Galaxy Formation: a Bayesian Uncertainty Analysis*", *Bayesian Analysis*, 5(4): 619–670, with rejoinder. Invited discussion paper. Awarded Mitchell Prize.

Bower, R., Vernon, I., Goldstein, M., et al. (2010), “*The Parameter Space of Galaxy Formation*", *Mon.Not.Roy.Astron.Soc.*, 407: 2017–2045.

Andrianakis, I., Vernon, I., McCreesh, N., McKinley, T.J., Oakley, J.E., Nsubuga, R., Goldstein, M., White, R.G.: *Bayesian history matching of complex infectious disease models using emulation: A tutorial and a case study on HIV in Uganda*. PLoS Comput Biol. 11(1), 1003968 (2015)

Goldstein, M., Seheult, A., Vernon, I.: “*Assessing Model Adequacy*". In: Wainwright, J., Mulligan, M. (eds.) Environmental Modelling: Finding Simplicity in Complexity, 2nd edn. John Wiley & Sons, Ltd, Chichester, (2013)

Vernon, I, Goldstein, M, Rowe, J, Liu, J and Lindsey, K, "*Bayesian uncertainty analysis for complex systems biology models: emulation, global parameter searches and evaluation of gene functions.*", BMC Systems Biology 12: 1, arXiv:1607.06358v1 [q-bio.MN].

Andrianakis, I., McCreesh, N., Vernon, I, McKinley, T. J. Oakley, J. E. Nsubuga, R. Goldstein, M. & White, R. G. (2016). "*History matching of a high dimensional individual based HIV transmission model*". JUQ 5(1):694-719.

McCreesh, N., Andrianakis, I, Nsubuga, R., Strong, M., Vernon, I., McKinley, T.J. Oakley, J.E., Goldstein, M., Hayes, R. & White, R.G. "*Universal Test, Treat, and Keep: Improving ART Retention is Key in Cost-effective HIV Control in Uganda*". BMC Infectious Diseases (2017) 17:322

Trevelyan J. McKinley, Ian Vernon, Ioannis Andrianakis, Nicky McCreesh, Jeremy E. Oakley, Rebecca N. Nsubuga, Michael Goldstein, Richard G. White (2016). "*Approximate Bayesian Computation and simulation-based inference for large-scale stochastic epidemic models*". Statistical Science 33(1): 4–18.

Vernon, I. R., Jackson, S. E. & Cumming, J. A. (2019). “*Known Boundary Emulation of Complex Computer Models*". SIAM/ASA Journal on Uncertainty Quantification (to appear), arXiv:1801.03184v2 [stat.ME].

References

Cumming, J. A. & Goldstein, M. (2009). Small Sample Bayesian Designs for Complex High-Dimensional Models Based on Information Gained Using Fast Approximations. Technometrics 51(4): 377-388.

Cumming, J. A. & Goldstein, M. (2010). Bayes linear Uncertainty Analysis for Oil Reservoirs Based on Multiscale Computer Experiments. In The Oxford Handbook of Applied Bayesian Analysis. O'Hagan, A. & West, M. Oxford: Oxford University Press. 241-270.

Craig, P.S., Goldstein, M., Seheult, A.H. & Smith, J.A. (1997). Pressure matching for hydrocarbon reservoirs: a case in the use of Bayes linear strategies for large computer experiments (and discussion). In Case studies in Bayesian Statistics. Gatsonis et al New York: Springer-Verlag. III: 37-93.

Craig, P., Goldstein, M., Seheult, A. H. & Smith, J. A. (1996). Bayes linear strategies for matching hydrocarbon reservoir history. In Bayesian Statistics 5. Bernardo, J. M. et al. Clarendon Press. 69-95.