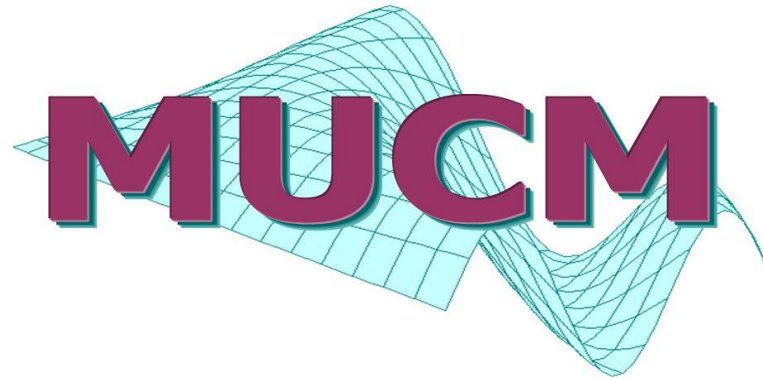


Using an emulator



Outline

- ▲ So we've built an emulator – what can we use it for?
- ▲ Prediction
 - ▲ What would the simulator output y be at an untried input x ?
- ▲ Uncertainty analysis
 - ▲ Given uncertainty in x , what is the implied uncertainty in y ?
- ▲ Sensitivity analysis
 - ▲ Which inputs influence the output most?
 - ▲ Which inputs are responsible for most output uncertainty?
- ▲ Calibration
 - ▲ Given observation of the real system, how can we use that to learn about the best input values?

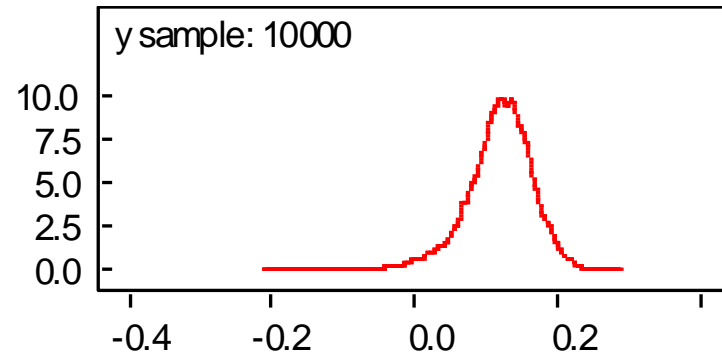
Prediction and UA

Prediction

- ▲ Prediction is simple because that's precisely what the emulator does
 - ▲ For any given x , the emulator mean $E(f(x))$ is an estimate
 - ▲ The emulator variance $\text{var}[f(x)]$ expresses uncertainty
 - ▲ Known as code uncertainty
- ▲ Similarly, given x and some threshold c we can evaluate $P[f(x) > c]$

Uncertainty analysis

- ▲ If X has distribution $g(x)$ then UA looks at the implied distribution of $Y = f(X)$
 - ▲ How do we evaluate that?
- ▲ In Session 1 we used Monte Carlo for a simple nonlinear simulator
 - ▲ Mean = 0.117
 - ▲ Median = 0.122
 - ▲ Std. dev. = 0.048
- ▲ But all these are estimates
 - ▲ Accuracy depends on the size of the Monte Carlo sample
 - ▲ 95% interval for the mean is (0.116, 0.118)



UA with an emulator

- ▲ Consider the expected output $M = E[Y]$
- ▲ It is uncertain because of code uncertainty
- ▲ The emulator mean value for M is
$$E[M] = \int E[f(x)] g(x) dx$$
- ▲ We can evaluate this by Monte Carlo
 - ▲ Sample many values of x , evaluate the emulator mean $E[f(x)]$ for each and average them
 - ▲ This is already much faster than making many simulator runs to evaluate $f(x)$
- ▲ But we can often do the integral exactly

Why emulation is more efficient

- ▲ Similarly we can evaluate $\text{var}[M]$
 - ▲ This is code uncertainty and depends on the number of simulator runs to build the emulator
 - ▲ We want to compute/estimate M sufficiently accurately, so we want $\text{var}[M]$ sufficiently small
- ▲ Emulation is more efficient because we can typically achieve the desired accuracy using far fewer simulator runs to build the emulator than using traditional methods
 - ▲ For the simple nonlinear model, using only 25 simulator runs to build the emulator, a 95% interval for M is (0.1173, 0.1179)
- ▲ Using the emulator we can also compute/estimate all those other quantities of interest, like $\text{var}[Y]$ or $P[Y > c]$

Sensitivity analysis

Sensitivity analysis

- ▲ Which inputs affect the output most?
- ▲ This is a common question
- ▲ Sensitivity analysis (SA) attempts to address it
- ▲ There are various forms of SA
- ▲ The methods most frequently used are not the most helpful!

Recap – the nonlinear model

- ▲ The simple nonlinear model of the first session
$$y = \sin(x_1) / \{1 + \exp(x_1 + x_2)\}$$
- ▲ Just two inputs
- ▲ Uncertainty analysis:
 - ▲ Normal distributions on inputs
 - ▲ Output mean = 0.117, median = 0.122
 - ▲ Std. dev. = 0.048
- ▲ Which of these two inputs influences output most?
 - ▲ And in what ways?

Local sensitivity analysis

- ▲ To measure the sensitivity of y to input x_i , compute the derivative of y with respect to x_i
 - ▲ Nonlinear model:
 - ▲ At $x_1 = x_2 = 0.5$, the derivatives are wrt x_1 , 0.142; wrt x_2 , -0.094
- ▲ How useful is this?
 - ▲ Derivatives evaluated only at the central estimate
 - ▲ Could be quite different at other points nearby
 - ▲ Doesn't capture interactions between inputs
 - ▲ E.g. sensitivity of y to increasing both x_1 and x_2 could be greater or less than the sum of their individual sensitivities
 - ▲ Not invariant to change of units

One-way SA

- ▲ Vary inputs one at a time from central estimate
 - ▲ Nonlinear model:
 - ▲ Vary x_1 to 0.25, 0.75, output is 0.079, 0.152
 - ▲ Vary x_2 to 0.25, 0.75, output is 0.154, 0.107
- ▲ Is this more useful?
 - ▲ Depends on how far we vary each input
 - ▲ Relative sensitivities of different inputs change if we change the ranges
 - ▲ But ranges are arbitrary
 - ▲ Also fails to capture interactions
 - ▲ Statisticians have known for decades that varying factors one at a time is bad experimental design!

Multi-way SA

- ▲ Vary factors two or more at a time
 - ▲ Maybe statistical factorial design
 - ▲ Full factorial designs require very many runs
- ▲ Can find interactions but hard to interpret
 - ▲ Often just look for the biggest change of output among all runs
- ▲ Still dependent on how far we vary each input

Probabilistic SA

- ▲ Inputs varied according to their probability distributions
 - ▲ As in Uncertainty Analysis (UA)
 - ▲ Sensitivities still depend on ranges of distributions (variances), but these are now not necessarily arbitrary
- ▲ Gives an overall picture and can identify interactions

Variance decomposition

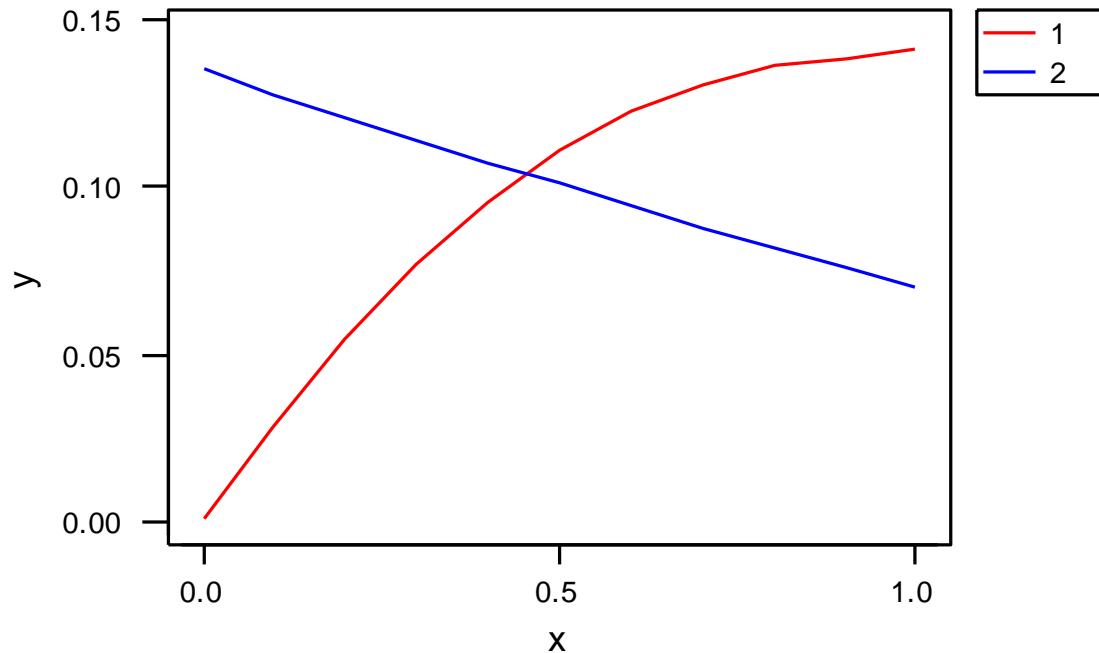
- ▲ One way to characterise the sensitivity of the output to individual inputs is to compute how much of the UA variance is due to each input
- ▲ For the simple non-linear model, we have

Input	Contribution
X1	80.30 %
X2	16.77 %
X1.X2 interaction	2.93 %

Main effects

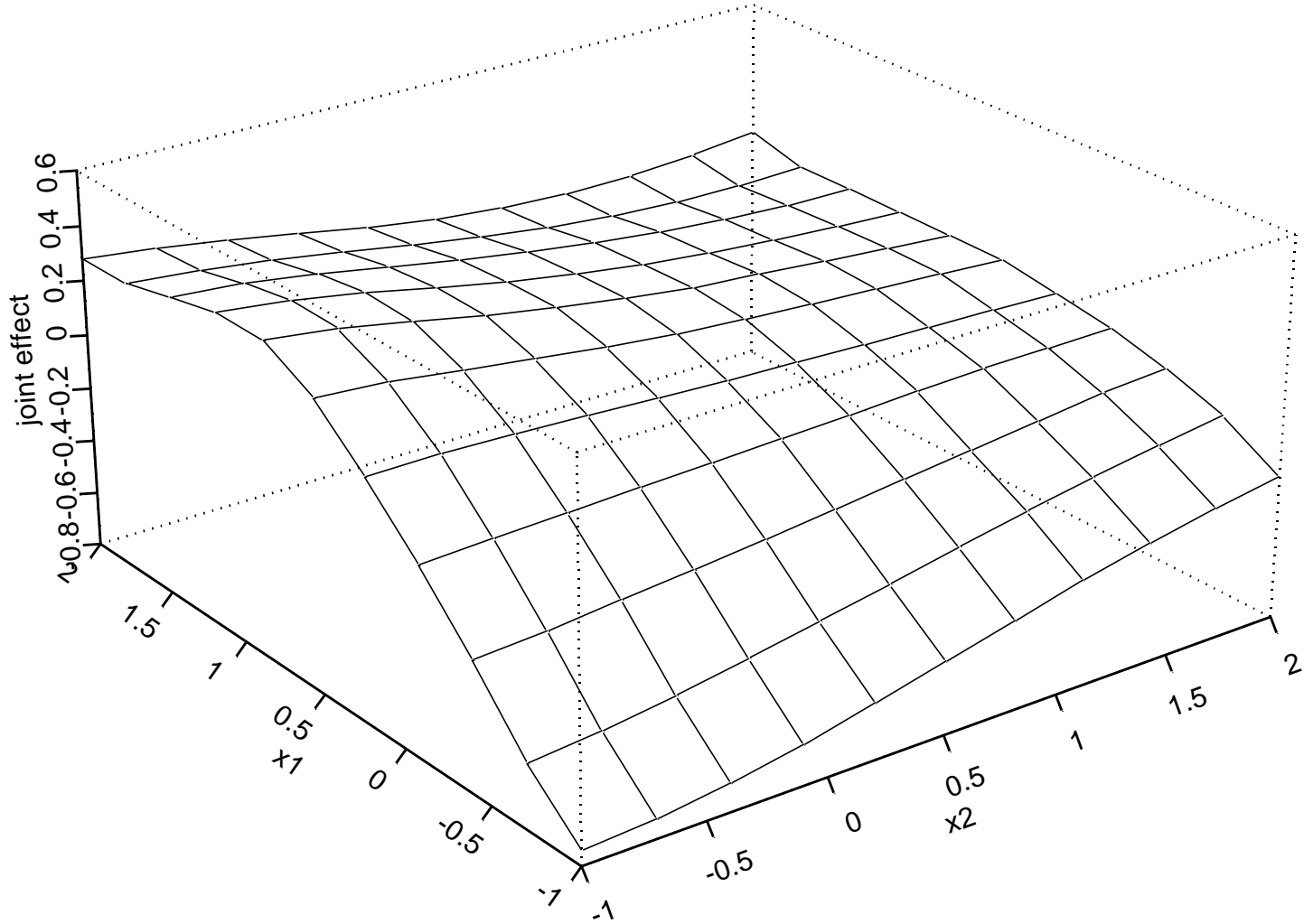
- ▲ We can also plot the effect of varying one input *averaged* over the others
- ▲ Nonlinear model
 - ▲ Averaging $y = \sin(x_1)/\{1+\exp(x_1+x_2)\}$ with respect to the uncertainty in x_2 , we can plot it as a function of x_1
 - ▲ Similarly, we can plot it as a function of x_2 averaged over uncertainty in x_1
- ▲ We can also plot interaction effects

Main effects in the simple nonlinear model

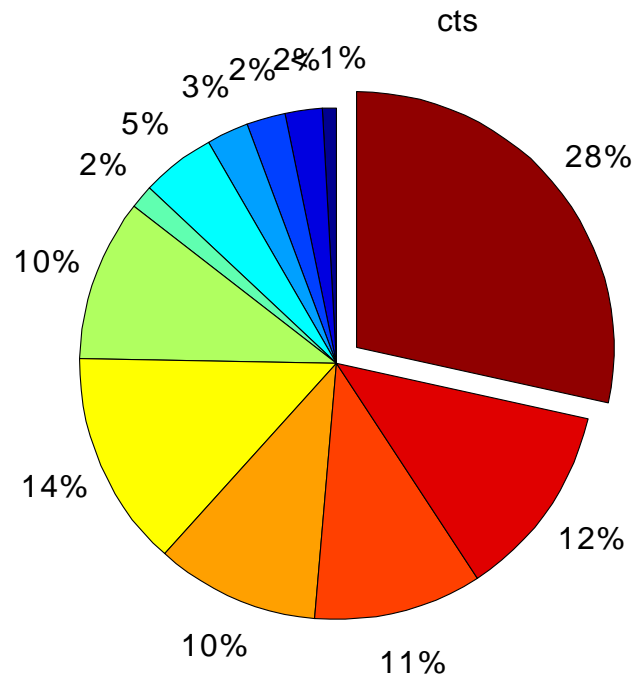


- ▲ Red is main effect of x_1 (averaged over x_2)
- ▲ Blue is main effect of x_2 (averaged over x_1)

Joint effect in the simple nonlinear model



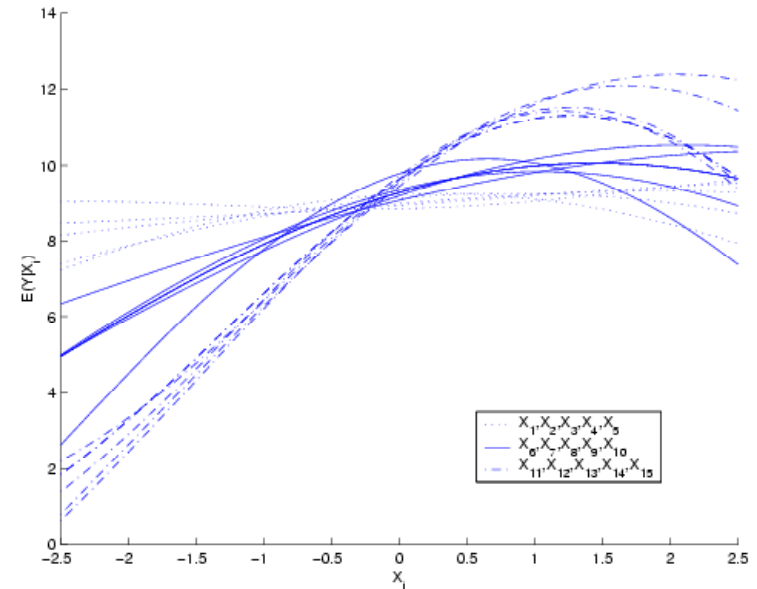
A more complex example



- ▲ 5 inputs have appreciable influence, and account for 57% of the total UA variance
- ▲ Interactions account for 28%

Amplifying on variances

- ▲ Main effect plots amplify on the information given in the variance decomposition
- ▲ The variance component associated with input x_i is equal to the amount by which its main effect varies over the range of uncertainty in x_i



The 5 inputs with most influence are dashed

Demonstration with GEM-SA

GEM-SA

- ▲ GEM-SA is a user-friendly piece of software that does many of the things we've been discussing
 - ▲ Can create some kinds of design
 - ▲ Fits an emulator to simulator output
 - ▲ Computes uncertainty and sensitivity analyses
- ▲ It's freely available
 - ▲ <http://tonyohagan.co.uk/academic/GEM>
- ▲ It's really useful for experimenting with relatively simple simulators
 - ▲ But not always reliable

The main GEM-SA window

menu

toolbar

Sensitivity Analysis | Main Effects | Output Summary








	Input name	Variance (%)	St.deviation	Total effect	St.deviation
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					

Sensitivity analysis output grid

Log window

Ready | No project | No inputs | No outputs

Toolbar icons

-  New project
-  Open project
-  Save project
-  Print output report
-  Edit project
-  Generate input design points
-  Rescale an input
-  Standardise design
-  Copy input design to clipboard
-  Convert input to integer
-  Run the analysis
-  Help

Output tabs

- ▲ When an emulator has been fitted, the contents of these tabs will provide the main results
- ▲ *Sensitivity Analysis*. This will report the SA variance decompositions
 - ▲ One line for each input parameter
 - ▲ One line for each pair of inputs, if joint effects are selected
- ▲ *Main effects*. This will plot the main effects of the various inputs
- ▲ *Results Summary*. This will present numerical summaries of emulator fit and uncertainty analysis

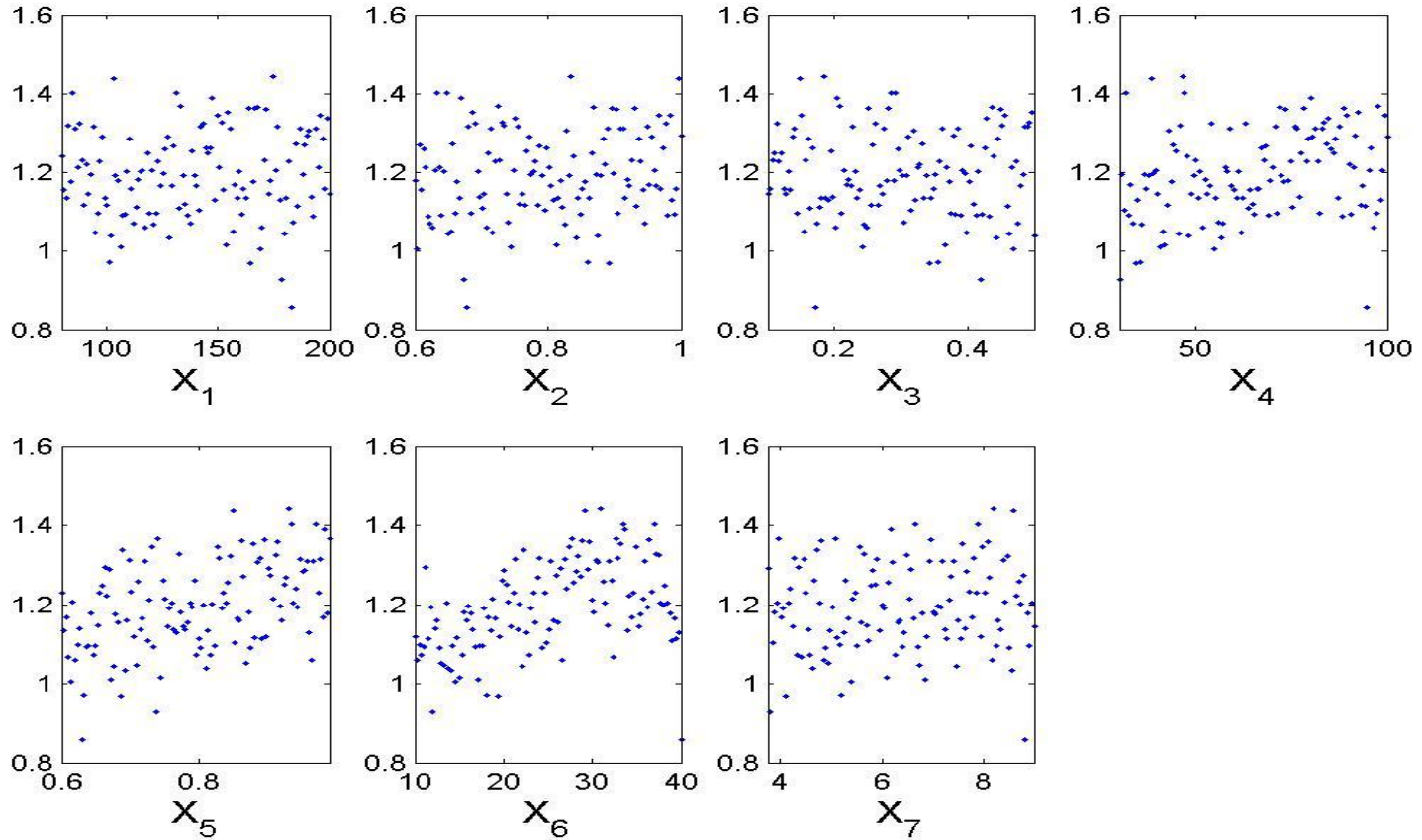
Log Window output

- ▲ Tells us
 - ▲ Which training data are being loaded/saved
 - ▲ Transformations applied to the data
 - ▲ Fitted Gaussian process parameters
 - ▲ Summary of cross-validation analysis
 - ▲ Summary of the uncertainty analysis

Example

- ▲ The GEM-SA package includes a number of datasets to play with
- ▲ One of these is from the ForestETP vegetation simulator
 - ▲ 7 input parameters
 - ▲ 120 simulator runs
- ▲ Objective: conduct a variance-based sensitivity analysis
 - ▲ To identify which uncertain inputs are driving the output uncertainty.

Exploratory scatter plots



First impressions

- ▲ Looks like X6 is most important, and probably also X5
- ▲ But these plots are hard to read because of the scatter
- ▲ GEM-SA fitted an emulator and carried out the uncertainty and sensitivity analyses

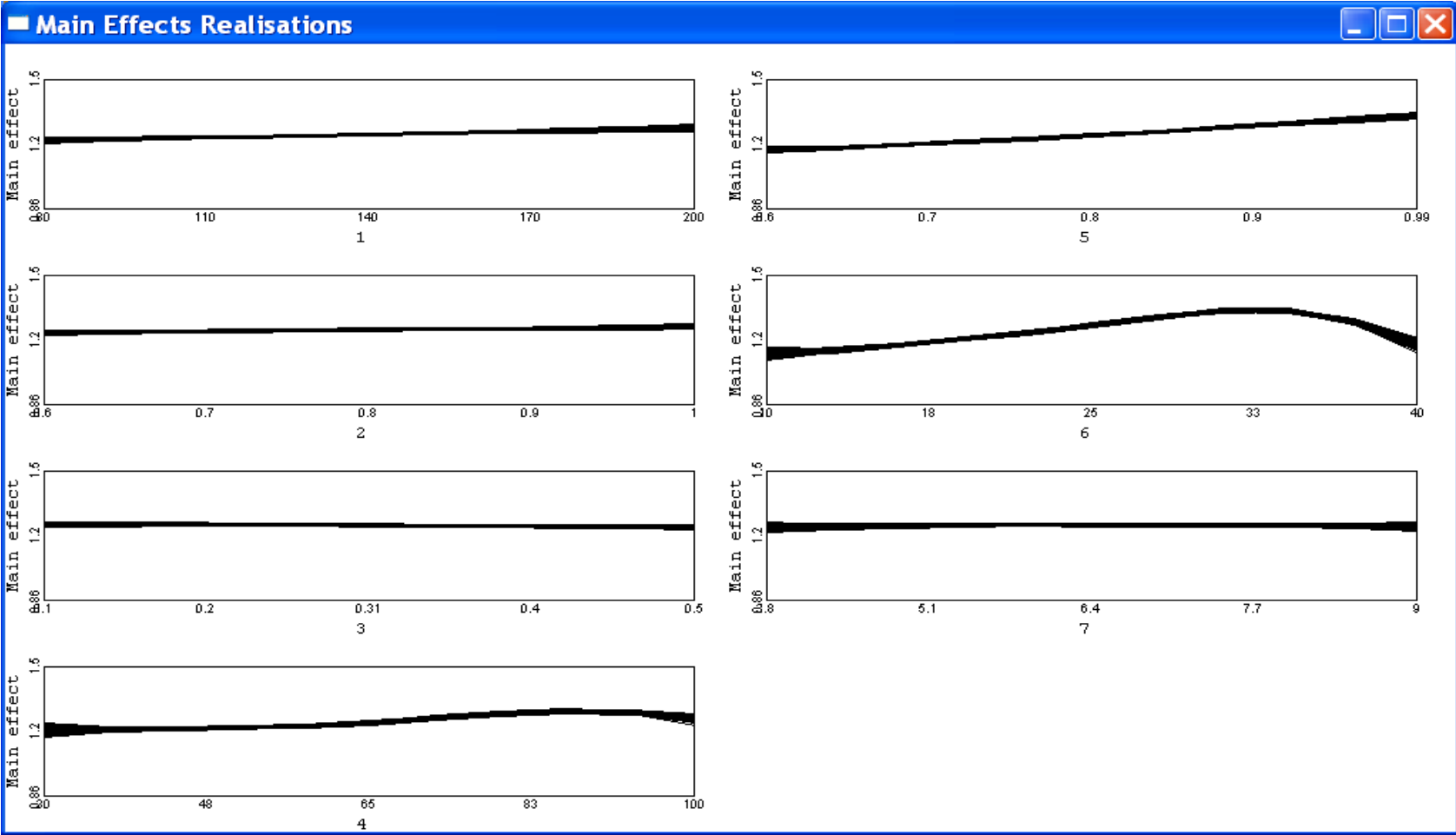
Variance of main effects

	Input name	Variance (%)	St. deviation	Total effect
1	X1	1.80		6.86
2	X2	0.55		1.43
3	X3	0.10		0.62
4	X4	9.31		39.14
5	X5	18.83		30.75
6	X6	36.04		43.26
7	X7	0.22		15.75
8		Total = 66.8311		

Main effects for each input. Input 6 has the greatest individual contribution to the variance

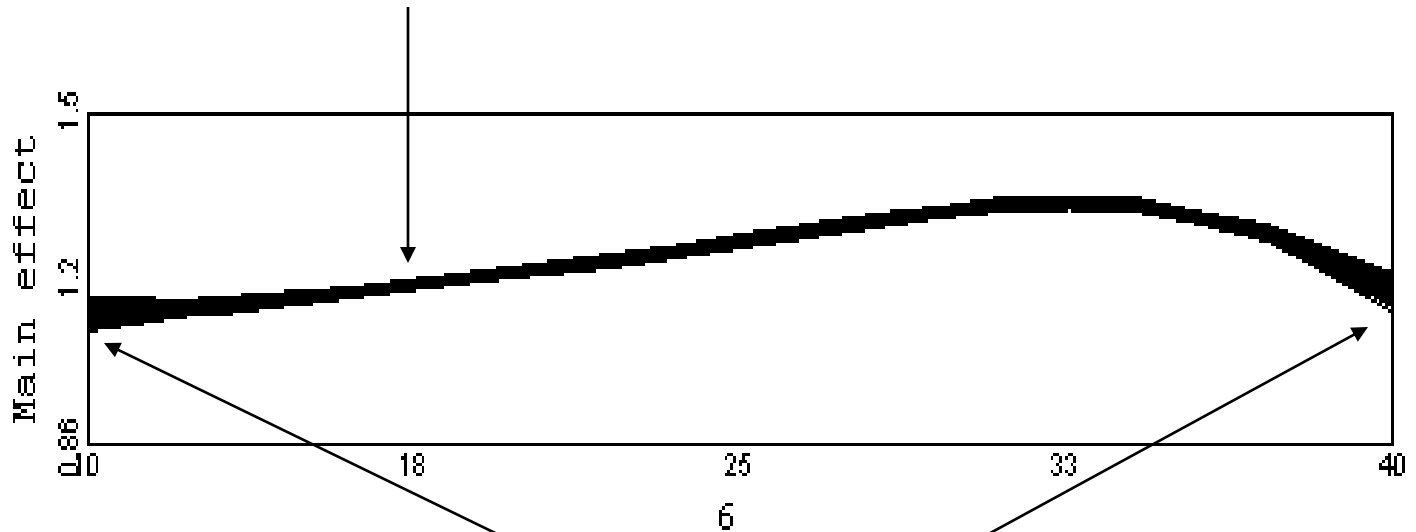
Main effects sum to 66% of the total variance

Main effect plots



Main effect plots

Fixing $X_6 = 18$, this point shows the *expected* value of the output (obtained by averaging over all other inputs).



Simply fixing all the other inputs at their central values and comparing $X_6=10$ with $X_6=40$ would underestimate the influence of this input

(The thickness of the band shows emulator uncertainty)

Interactions and total effects

- ▲ Main effects explain only 2/3 of the variance
 - ▲ Model must contain interactions
- ▲ Any input can have small main effect, but large interaction effect, so overall still an ‘important’ input
- ▲ We can ask GEM-SA to compute all pair-wise interaction effects
 - ▲ 435 in total for a 30 input model – can take some time!
- ▲ Useful to know what to look for

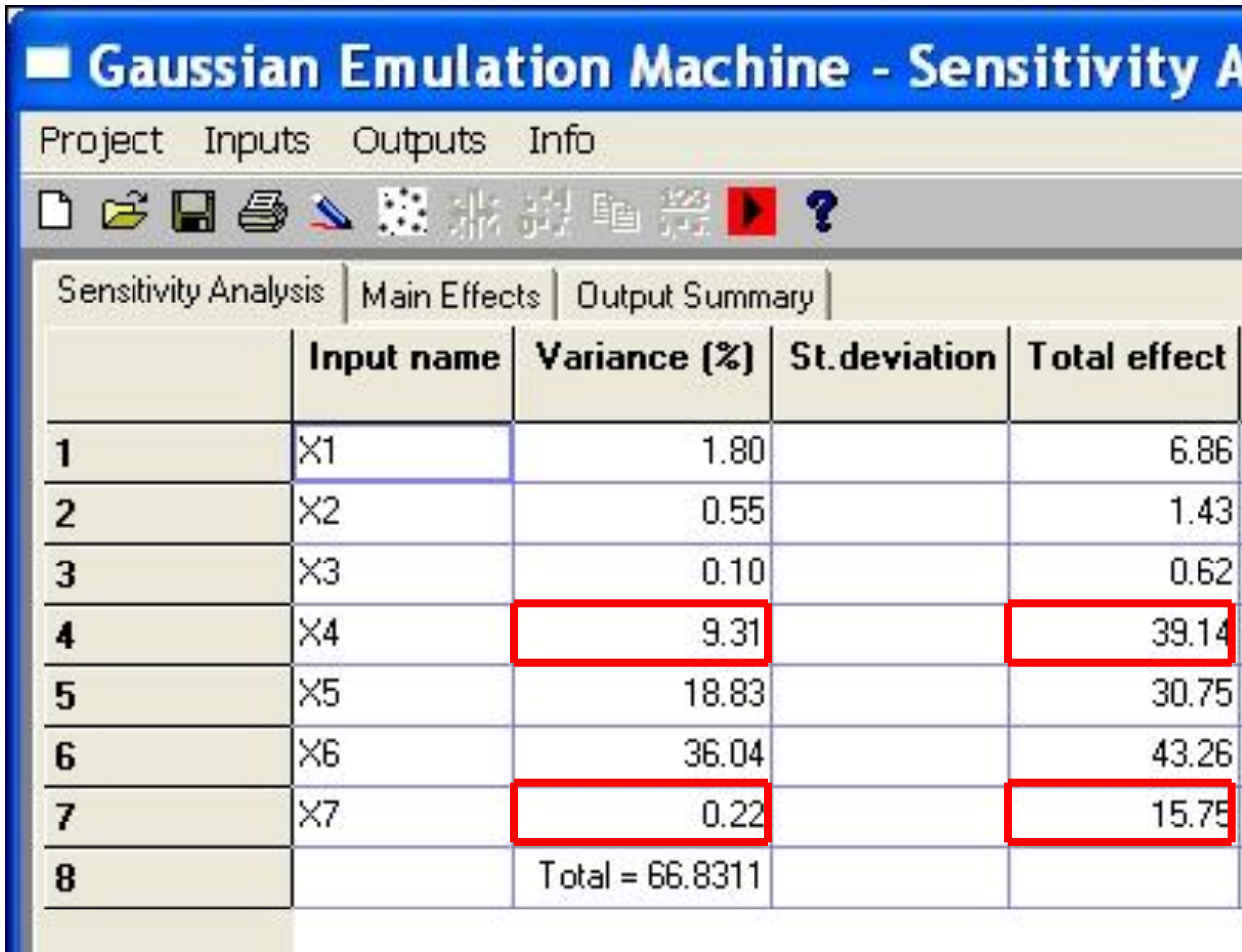
Interactions and total effects

- ▲ For each input X_i

Total effect = main effect for X_i + all interactions involving X_i

- ▲ Assumes independent inputs
- ▲ Main effects and total effects normalised by variance
- ▲ Total effect \gg main effect implies interactions in the model
- ▲ Look for inputs with large total effects relative to main effects
 - ▲ Investigate possible interactions involving those inputs

Interactions and total effects



The screenshot shows the 'Gaussian Emulation Machine - Sensitivity Analysis' software interface. The window title is 'Gaussian Emulation Machine - Sensitivity Analysis'. Below the title bar, there are tabs for 'Project', 'Inputs', 'Outputs', and 'Info'. A toolbar contains various icons for file operations and analysis. The main area has tabs for 'Sensitivity Analysis', 'Main Effects', and 'Output Summary'. A table displays the results of the sensitivity analysis, with columns for 'Input name', 'Variance (%)', 'St.deviation', and 'Total effect'. The table lists 8 rows, with the last row being a total. Red boxes highlight the 'Variance (%)' and 'Total effect' columns for inputs 4 and 7.

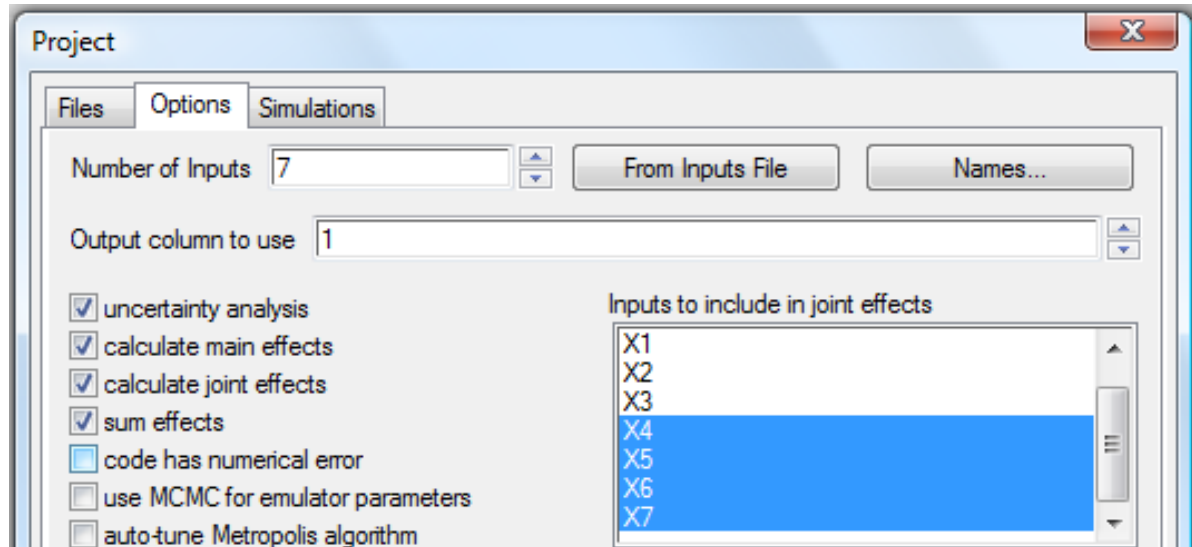
	Input name	Variance (%)	St.deviation	Total effect
1	X1	1.80		6.86
2	X2	0.55		1.43
3	X3	0.10		0.62
4	X4	9.31		39.14
5	X5	18.83		30.75
6	X6	36.04		43.26
7	X7	0.22		15.75
8		Total = 66.8311		

Total effects for inputs 4 and 7 much larger than their main effects.

Implies presence of interactions.

Interaction effects

- ▲ Ask GEM-SA to compute pair-wise joint effect variances
 - ▲ All interactions between X4, X5, X6, X7



Main and interaction effects

The screenshot shows the 'Gaussian Emulation Machine - Sensitivity Analysis' interface. The 'Main Effects' tab is selected, displaying a table with columns for 'Input name', 'Variance (%)', 'St.deviation', 'Total effect', and 'St.deviation'. The table lists 14 rows, including main effects for inputs X1 through X7 and interactions between X4, X5, X6, and X7. A red box highlights the interaction terms X4.X5, X4.X6, X4.X7, X5.X7, and X6.X7. Red arrows point from the text on the right to these highlighted rows and the total variance row.

	Input name	Variance (%)	St.deviation	Total effect	St.deviation
1	X1	1.80		6.86	
2	X2	0.55		1.43	
3	X3	0.10		0.62	
4	X4	9.31		39.14	
5	X5	18.83		30.75	
6	X6	36.04		43.26	
7	X7	0.22		15.75	
8	X4.X5	9.28			
9	X4.X6	2.06			
10	X4.X7	11.67			
11	X5.X6	0.72			
12	X5.X7	0.28			
13	X6.X7	0.91			
14		Total = 91.7539			

Note interactions involving inputs 4 and 7

Main effects and selected interactions now sum to almost 92% of the total variance

What have we learnt here?

- ▲ Most important inputs are X4, X5, X6, X7
 - ▲ We can more or less ignore X1, X2, X3
 - ▲ Together these 3 account for < 10% of overall variance
- ▲ X6 is most important single input
 - ▲ 36% of variance
 - ▲ Has only minor interactions with other inputs
- ▲ X4, X5, X7 interact in complex ways
 - ▲ But together account for over 50% of variance
- ▲ Main effect plots are useful
 - ▲ Particularly for X6
 - ▲ But less so for the others! Need to look at how they interact

SA summary

Why SA?

1. For the model user: SA identifies which inputs it would be most useful to reduce uncertainty about
2. For the model builder: main effect and interaction plots demonstrate how the simulator is behaving
 - ▲ Sometimes surprisingly!

Calibration

Calibration

- ▲ Simulator users often want to tune the simulator using observations of the real system
- ▲ Adjust the input parameters so that the simulator output matches observations as well as possible
- ▲ Two very important points
 1. Calibration will reduce uncertainty about x but will not eliminate it
 2. It is necessary to understand how the simulator relates to reality
 - ▲ Model discrepancy

Model discrepancy

- ▲ Simulator output $y = f(x)$ will not equal the real system value z
 - ▲ Even with best/correct inputs x
- ▲ Model discrepancy is the difference $z - f(x)$
- ▲ As discussed in Session 1, model discrepancy is due to
 - ▲ Wrong or incomplete science
 - ▲ Programming errors, rounding errors
 - ▲ Inaccuracy in numerically solving systems of equations
- ▲ Ignoring model discrepancy leads to poor calibration
 - ▲ Over-fitting of parameter estimates
 - ▲ Over-confidence in the fitted values

Resources for calibration

- ▲ Calibration is beyond the scope of this course
- ▲ But there is material in the toolkit on calibration and model discrepancy
- ▲ GEM-Cal is an extension of GEM-SA that does some simple calibration
 - ▲ <http://tonyohagan.co.uk/academic/GEM>