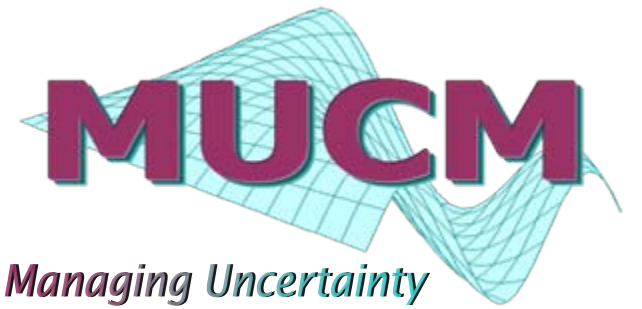
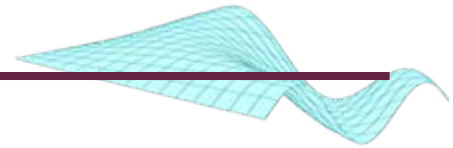
MUCM: An Overview

Tony O'Hagan, University of Sheffield



MUCM



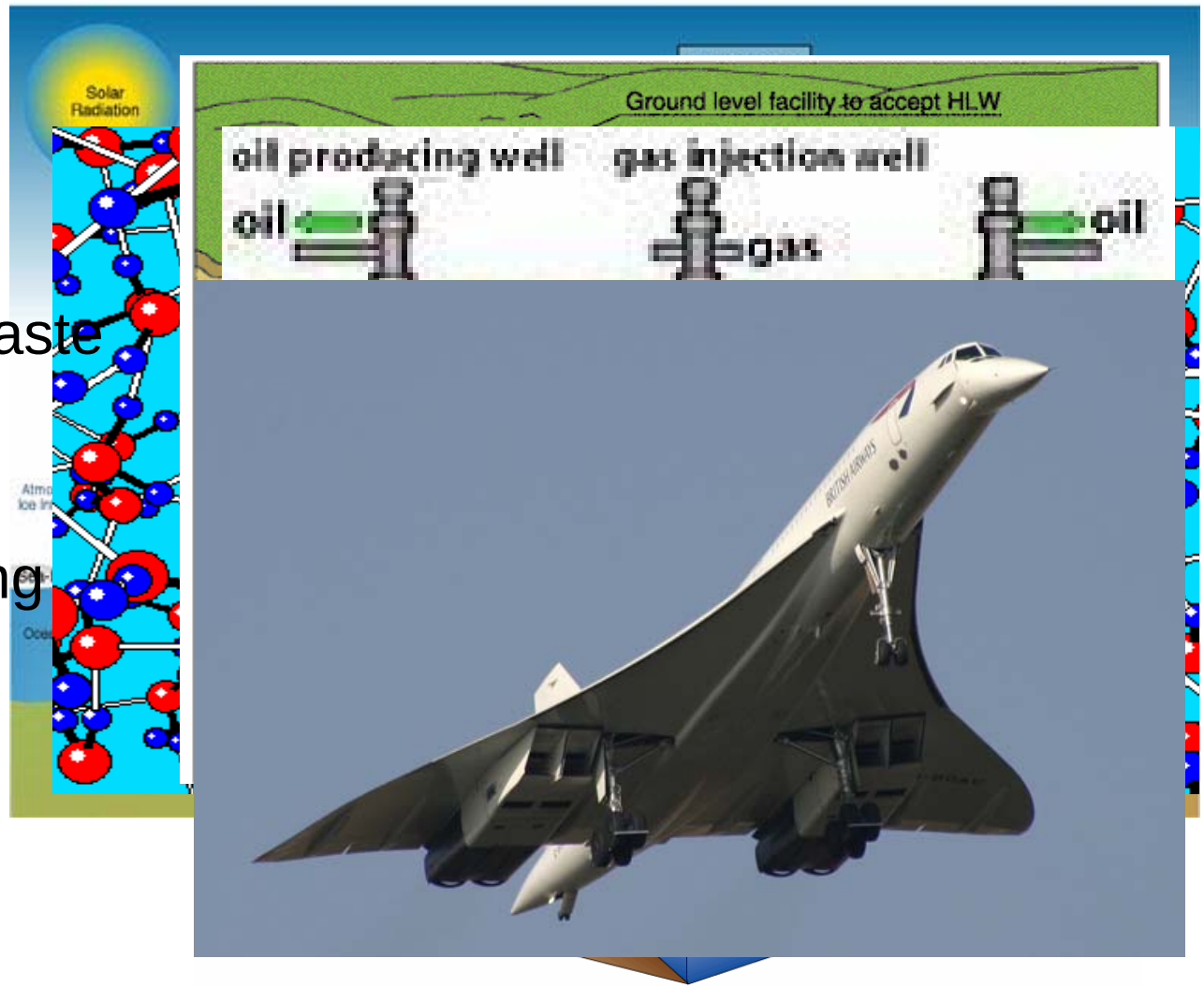
- Managing Uncertainty in Complex Models
  - Large 4-year research grant
  - June 2006 to September 2010
  - 7 postdoctoral research associates
  - 4 project PhD students
  - Based in Sheffield, Durham, Aston, Southampton, LSE
- What's it about?

Computer models

- In almost all fields of science, technology, industry and policy making, people use mechanistic models to describe complex real-world processes
  - For understanding, prediction, control
- There is a growing realisation of the importance of uncertainty in model predictions
  - Can we trust them?
  - Without any quantification of output uncertainty, it's easy to dismiss them

Examples

- Climate prediction
- Molecular dynamics
- Nuclear waste disposal
- Oil fields
- Engineering design
- Hydrology

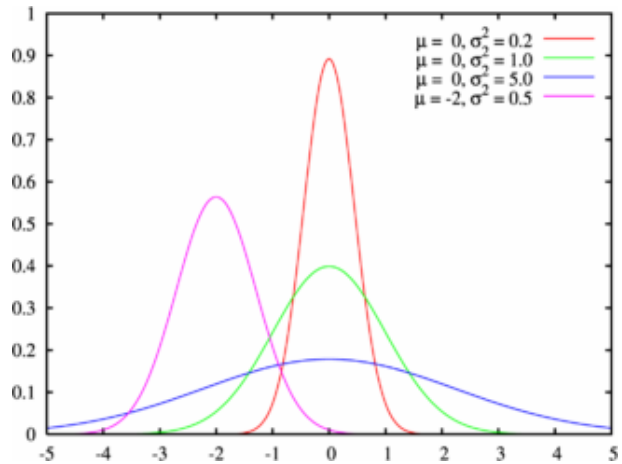


Sources of uncertainty

- A computer model takes inputs x and produces outputs y = f(x)
- How might y differ from the true real-world value z that the model is supposed to predict?
  - Error in inputs x
    - Initial values, forcing inputs, model parameters
  - Error in model structure or solution
    - Wrong, inaccurate or incomplete science
    - Bugs, solution errors

Quantifying uncertainty

- The ideal is to provide a probability distribution p(z) for the true real-world value
  - The centre of the distribution is a best estimate
  - Its spread shows how much uncertainty about z is induced by uncertainties on the last slide

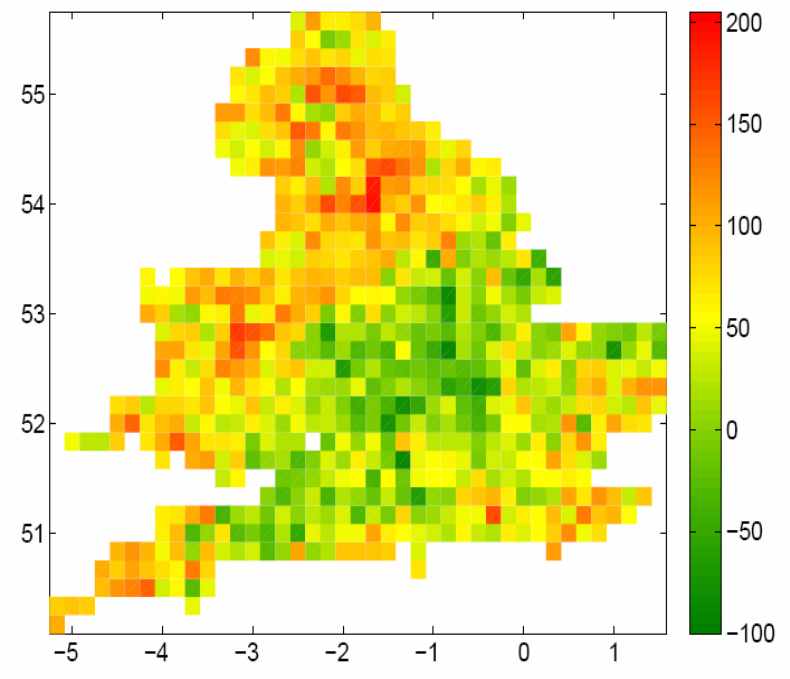


- How do we get this?
  - Input uncertainty: characterise p(x), propagate through to p(y)
  - Structural uncertainty: characterise p(z-y)

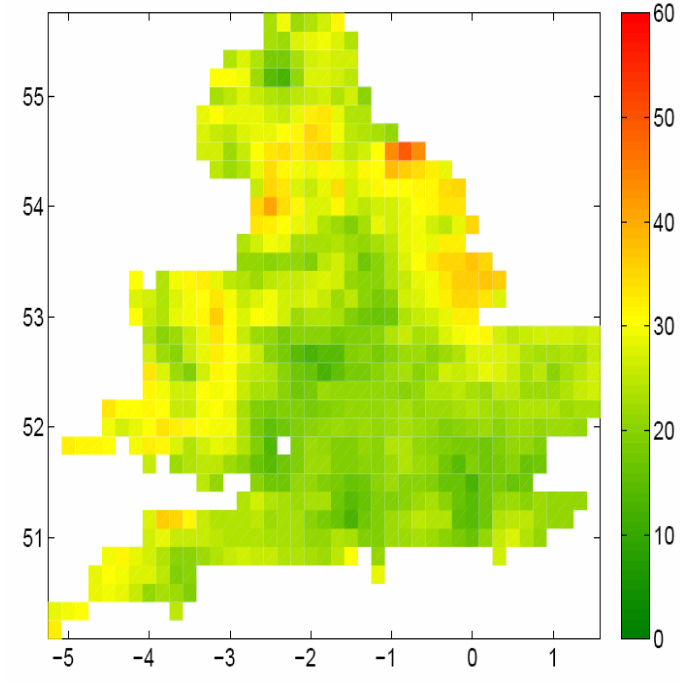
Example: UK carbon flux in 2000

- Vegetation model predicts carbon exchange from each of 700 pixels over England & Wales in 2000
  - Principal output is Net Biosphere Production
- Accounting for uncertainty in inputs
  - Soil properties
  - Properties of different types of vegetation
  - Land usage
  - (Not structural uncertainty)
- Aggregated to England & Wales total
  - Allowing for correlations
  - Estimate 7.46 Mt C
  - Std deviation 0.54 Mt C

Maps

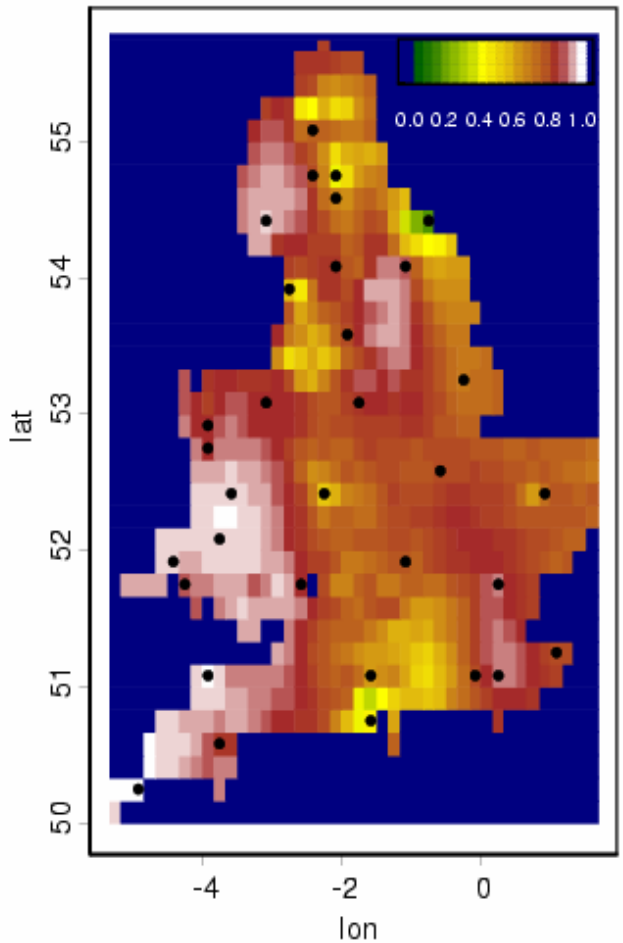


Mean NBP



Standard deviation

Sensitivity analysis



- Map shows proportion of overall uncertainty in each pixel that is due to uncertainty in the vegetation parameters
  - As opposed to soil parameters
- Contribution of vegetation uncertainty is largest in grasslands/moorlands

England & Wales aggregate

PFT	Plug-in estimate (Mt C)	Mean (Mt C)	Variance (Mt C^2)
Grass	5.28	4.37	0.2453
Crop	0.85	0.43	0.0327
Deciduous	2.13	1.80	0.0221
Evergreen	0.80	0.86	0.0048
Covariances			-0.0081
Total	9.06	7.46	0.2968

Reducing uncertainty

- To reduce uncertainty, get more information!
- Informal – more/better science
  - Tighten p(x) through improved understanding
  - Tighten p(z-y) through improved modelling or programming
- Formal – using real-world data
  - Calibration – learn about model parameters
  - Data assimilation – learn about the state variables
  - Learn about structural error z-y
  - Validation

So far, so good, but

- In principle, all this is straightforward
- In practice, there are many technical difficulties
  - Formulating uncertainty on inputs
    - Elicitation of expert judgements
  - Propagating input uncertainty
  - Modelling structural error
  - Anything involving observational data!
    - The last two are intricately linked
  - And *computation*

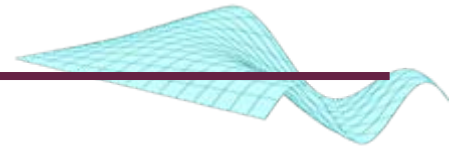
The problem of big models

- Tasks like uncertainty propagation and calibration require us to run the model many times
- Uncertainty propagation
  - Implicitly, we need to run f(x) at all possible x
  - Monte Carlo works by taking a sample of x from p(x)
  - Typically needs thousands of model runs
- Calibration
  - Traditionally this is done by searching the x space for good fits to the data
- Both become impractical if the model takes more than a few seconds to run
  - We need a more efficient technique

Gaussian process representation

- More efficient approach
  - First work in early 1980s (DACE)
- Represent the code as an unknown function
  - f(.) becomes a random process
  - We generally represent it as a Gaussian process (GP)
    - Or its second-order moment representation
- Training runs
  - Run model for sample of x values
  - Condition GP on observed data
  - Typically requires many fewer runs than MC
    - And x values don't need to be chosen randomly

Emulation

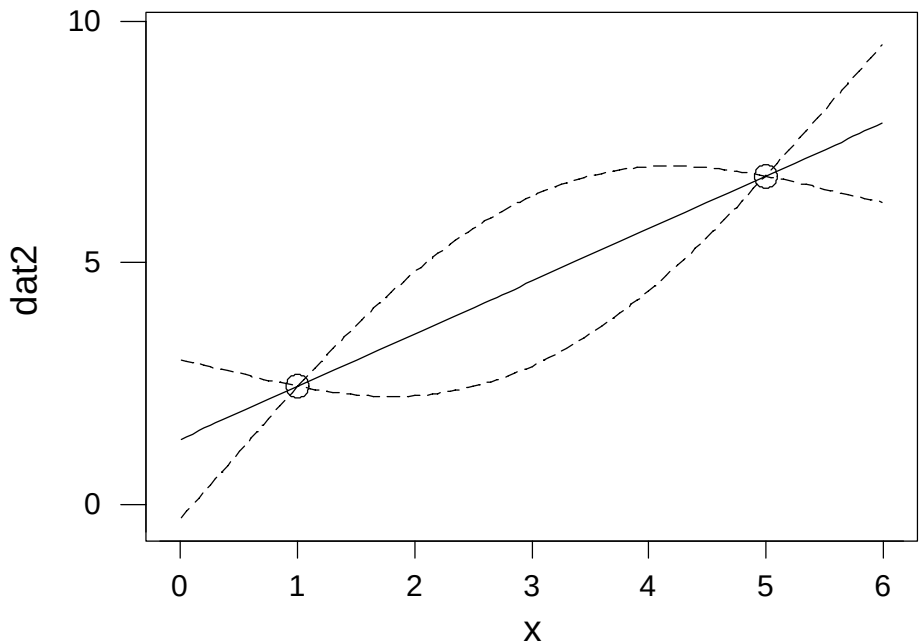


- Analysis is completed by prior distributions for, and posterior estimation of, hyperparameters

- The posterior distribution is known as an **emulator** of the computer code
  - Posterior mean estimates what the code would produce for any untried ***x*** (prediction)
  - With uncertainty about that prediction given by posterior variance
  - Correctly reproduces training data

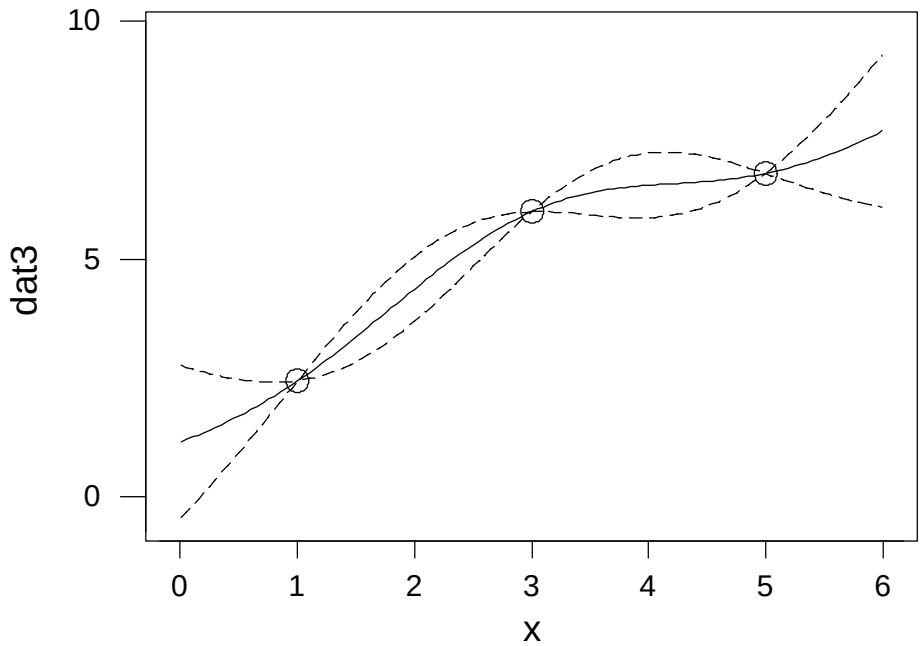
2 code runs

- Consider one input and one output
- Emulator estimate interpolates data
- Emulator uncertainty grows between data points



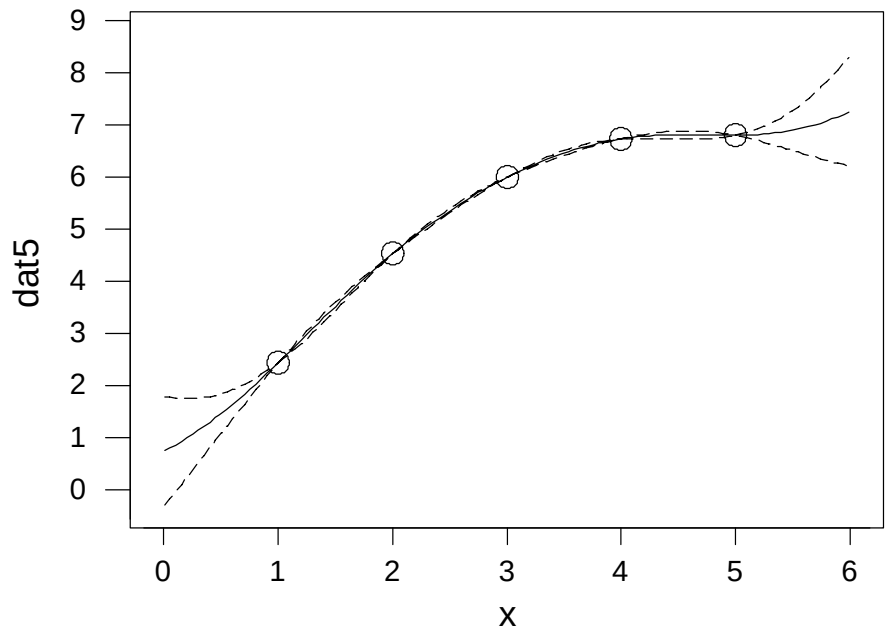
3 code runs

- Adding another point changes estimate and reduces uncertainty

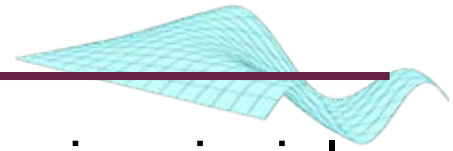


5 code runs

- And so on



Then what?



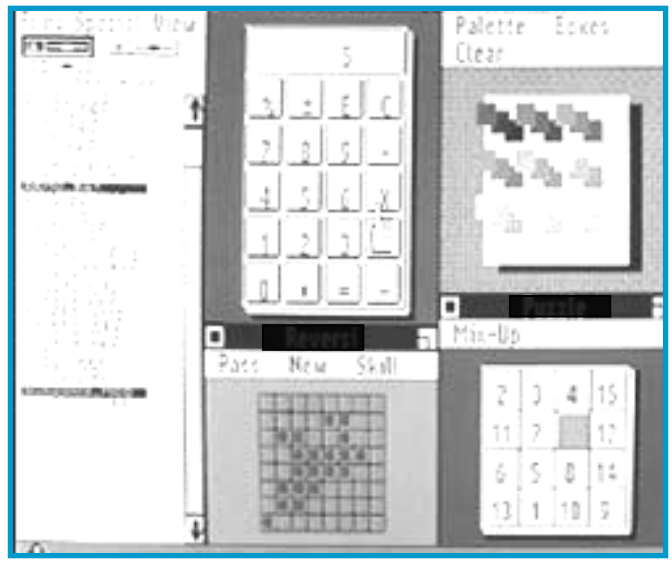
- Given enough training data points we can in principle emulate any model accurately
  - So that posterior variance is small “everywhere”
  - Typically, this can be done with orders of magnitude fewer model runs than traditional methods
    - At least in relatively low-dimensional problems
- Use the emulator to make inference about other things of interest
  - E.g. uncertainty analysis, calibration
- Conceptually very straightforward in the Bayesian framework
  - But of course can be computationally hard

BACCO

- This has led to a wide ranging body of tools for inference about all kinds of uncertainties in computer models
- All based on building the *emulator* of the model from a set of training runs
- This area is now known as BACCO
  - Bayesian Analysis of Computer Code Output
- MUCM's objective is to develop BACCO methods into a robust technology that is widely applicable across the spectrum of modelling applications

BACCO includes

- Uncertainty analysis
- Sensitivity analysis
- Calibration
- Data assimilation
- Model validation
- Optimisation
- Etc…
- All within a single coherent framework



MUCM workpackages

- Theme 1 – High Dimensionality
  - WP1.1: Screening
  - WP1.2: Sparsity and projection
  - WP1.3: Multiscale models
- Theme 2 – Using Observational Data
  - WP2.1: Linking models to reality
  - WP2.2: Diagnostics and validation
  - WP3.2: Calibration and data assimilation
- Theme 3 – Realising the Potential
  - WP3.1: Experimental design
  - WP3.2: Toolkit
  - WP3.3: Case studies

Primary deliverables

- Methodology and papers moving the technology forward
  - Particularly in Themes 1 and 2
  - Papers both in statistics and application area journals
- The toolkit
  - Wiki based
  - Documentation of the methods and how to use them
  - With emphasis on what is found to work reliably across a range of modelling areas
- Case studies
  - Three substantial and detailed case studies
  - Showcasing methods and best practice
  - Linked to toolkit
  - Published in book form as well as in a series of papers
- Workshops
  - Both conceptual and hands-on

Today

- Dissemination workshop
  - Concentrating on what we can do, rather than how to do it
  - Aimed at modellers, model users, funders of modelling, policy makers
  - Focussing on examples, primarily in the biological and health sciences
  - Remember, though, that the methods are generic and very widely applicable!
- Have a good day