Using R for the Visualisation of Computer Experiments

Neil Diamond

Department of Econometrics and Business Statistics
MONASH University

Computer Experiments

- The design and analysis of computer experiments to explore the behavior of complex systems is becoming increasingly important in science and engineering.
- At least two books on the topic:
  - The Design and Analysis of Computer Experiments. T. J. Santner, B. J. Williamns, W.I Notz. (2003), Springer: New York.
  - Design and Modeling for Computer Experiments. K-T. Fang, R. Li, A. Sudjianto. (2006), Chapman & Hall/CRC: London.
- Some R packages-more on that later.

Nimrod

- Developed by Computer Scientists at Monash University's eScience and Grid Engineering Laboratory.
- Automates the formulation, running, and collation of the individual experiments.
- Includes a distributed scheduling component that can manage the scheduling of individual jobs.

Nimrod Set of Tools

Nimrod contains tools to

- perform a complete parameter sweep across all possible combinations (Nimrod/G),
- search using non-linear optimization algorithms (Nimrod/O),
- or use fractional factorial design techniques (Nimrod/E).

These can be run stand-alone or accessed via the Nimrod portal

Nimrod Applications

Nimrod has been used in an extensive range of applications

- Air Pollution Studies
- Laser Physics
- Ecology
- Quantum Chemistry
- CAD Digital Simulation
- Antenna Design
- Cardiac Modelling

Statistical Approach to Computer Experiments

- Unlike physical experiments, repeated experiments give the same results.
- Model the output as the realisation of a stochastic process with a correlation structure that depends on a distance to other points in the experiment.
- Allows estimates of untried experiments.
- Gives an estimate of the uncertainty.

Computer Experiments-Designs

- Simplest method-Latin Hypercubes
- Other more sophisticated methods include Orthogonal Arrays and Scrambled Nets.
- Various space filling designs.

Computer Experiments-Model

$$\text{Response} = \text{Linear Model} + \text{Departure}$$

$$y(x) = \beta + z(x)$$

$$E(z(x)) = 0$$

$$\text{Cov}(z(t), z(u)) = \sigma_z^2 \prod_{j=1}^{d} R_j(t_j, u_j)$$

$$R_j(t_j, u_j) = \exp\left[-\theta_j (t_j - u_j)^{p_j}\right]$$

MLE of θ, p, β, and σ^2

Reduces to numerically optimising

$$-\frac{1}{2}(n \ln \hat{\sigma}^2 + \ln \det R_D)$$

$R_D =$ Matrix of correlations for design points

$$\begin{aligned} \hat{\beta} &= (1^T R_D^{-1} 1^T)^{-1} 1^T R_D^{-1} y \\ \hat{\sigma}^2 &= \frac{1}{n}(y - 1\hat{\beta})^T R_D^{-1}(y - 1\hat{\beta}) \end{aligned}$$

Best Linear Unbiased Predictor for an untried x

$$\hat{y}_x = \hat{\beta} + r^T(x) R_D^{-1}(y - 1\hat{\beta})$$

where

$$r(x) = [R(x_1, x), R(x_2, x), \ldots, R(x_n, x)]^T$$

Design point : $[x_1, x_2, \ldots, x_n]$ Untried Input : x

Interpolates the data points.

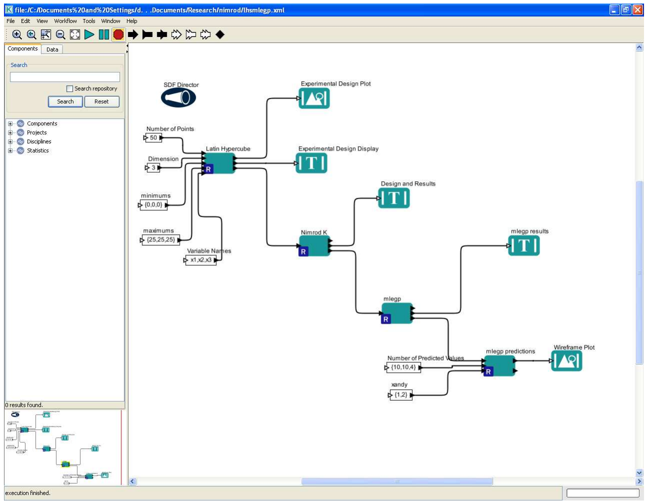
Implementations in R

- BACCO
  - Emulator
  - Approximator
  - Calibrator
- mlegp: an R package for Gaussian process modeling and sensitivity analysis
- Certainly others ...

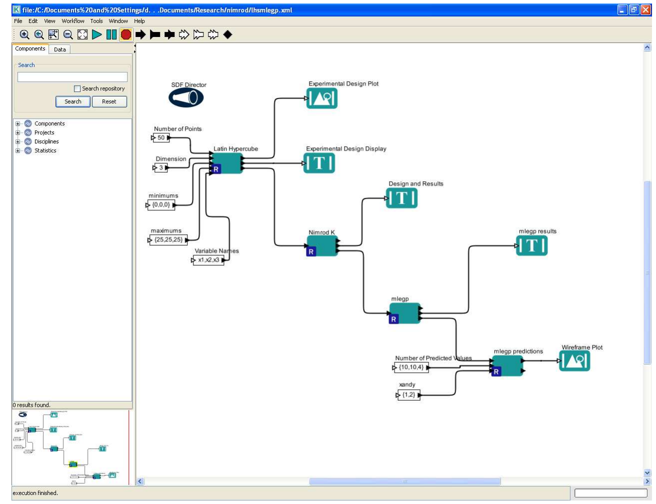
Nimrod/K

- A new tool (Nimrod/K) is being developed, based on the Kepler workflow engine (Kepler Core, 2009).
- It leverages a number of the techniques developed in the earlier Nimrod tools for distributing tasks to the Grid.
- Kepler allows the user to specify R expressions and access R objects as part of the scientific workflow.

Example Workflow



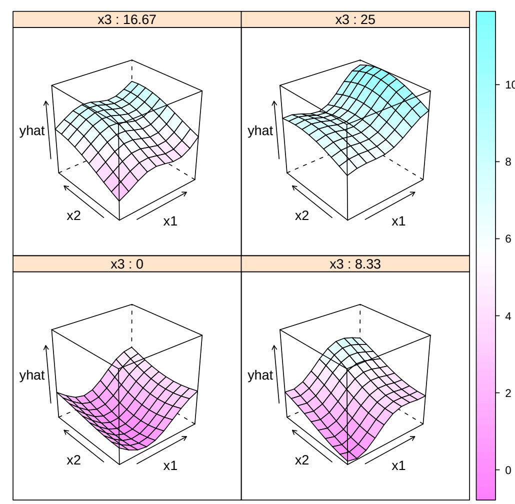
Example Workflow



Workflow

- The Latin Hypercube Actor creates the design
- Nimrod takes the experimental design and controls the running of the experiments and collation of results.
- Passes the results onto mlegp actor which fits the Gaussian model to the data.
- The Predictions Actor takes fitted model and predicts at a grid of untried inputs.
- Inputs are the granularity of the grid, and which are the primary and conditioning inputs.
- Uses Lattice graphics to produce a visualisation of the surface.

VizCompX



VizCompX

```
design <- LatinHypercube(50,3,maxs=rep(25,3))
response <- Nimrod0example(design)
mlegpfit <- mlegp(design,response)
wireframe(mlegpfit,c(5,5,4),c(1,2))
```


Extensions

The overall mean is

$$y_0 = \int_{[0,1]^d} y(x_1, \ldots, x_d) dx_1 \ldots dx_d$$

The main effect is

$$y_i(x_i) = \int_0^1 \ldots \int_0^1 y(x1, \ldots, x_d) dx_{-i} - y_0$$

The two factor interaction is

$$\begin{aligned} y_{i,j}(x_i, x_j) &= \int_0^1 \ldots \int_0^1 y(x1, \ldots, x_d) dx_{-ij} \\ &\quad - y_0 - y_i(x_i) - y_j(x_j) \end{aligned}$$

Extensions