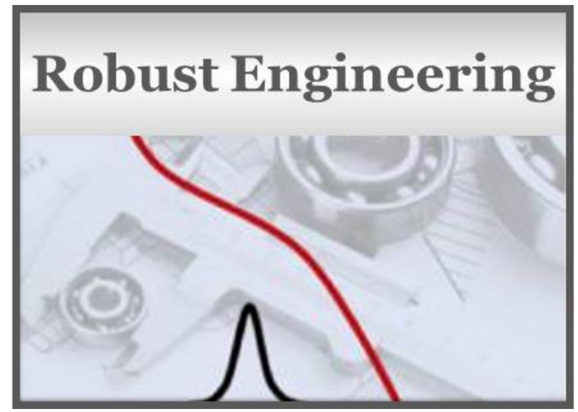


Session 11: Advanced Response Surface Methods

In This Session We Will…

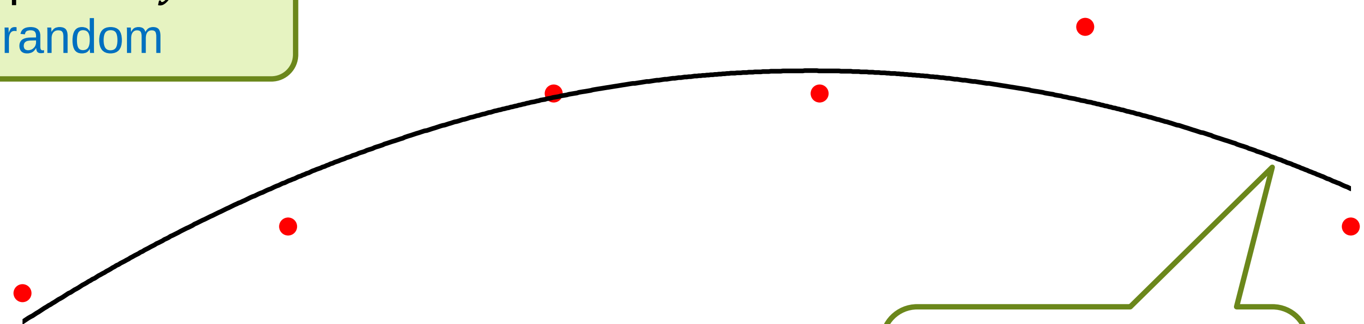
- Discuss the concept of over-fitting and potential problems with using higher-order polynomials.
- Introduce two new methods of generating a response surface (radial basis functions and kriging) which are effective for interpolation and also for smoothing noisy data.
- Explain what is meant by a space-filling design.
- Introduce Latin Hypercube (LH) designs.
- Compare strategies for hardware and computer experiments.
- Apply space-filling designs to engine testing.

An Example With One x-variable

Assume that this is ‘noisy’ data

i.e. part of y is random

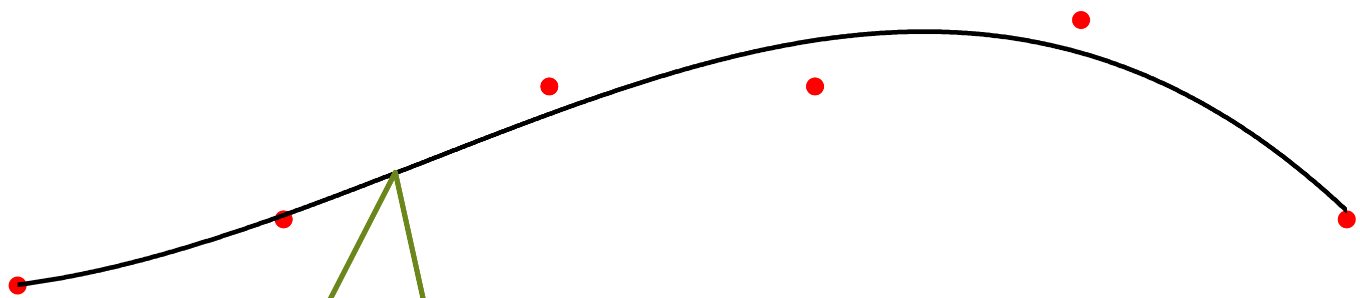
Let us use a simple example to illustrate over-fitting



We could fit a quadratic curve by least squares

$R^2 = 0.805$

3rd Order (Cubic) Polynomial



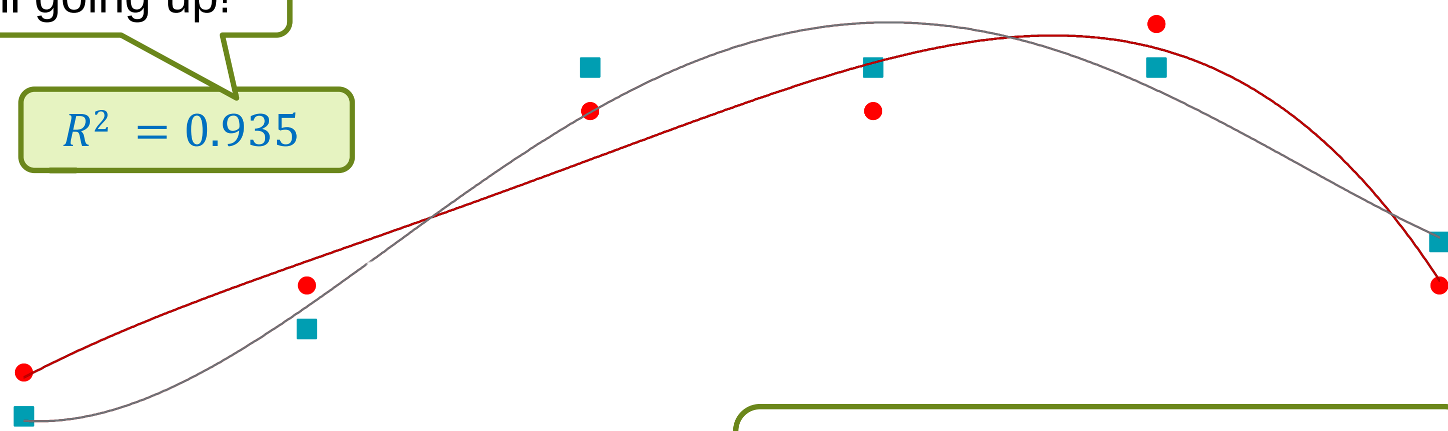
Is a cubic equation a better fit?

$R^2 = 0.867$

4^{th} Order Polynomial (Quartic)

Still going up!

$R^2 = 0.935$



But with slightly different data we get very different curve

This 4th order model is over-fitted

It is too close to the data rather than the physical phenomenon

Avoiding Over-fitting With Polynomials

- Cross-validation (PRESS) is helpful, provided we have enough data.
  - see the cautionary note about PRESS in TS10.
- Carefully selected 3rd order terms have been found to be useful (e.g. in engine mapping) but avoid 4th order or higher-order polynomials.
  - unless there are good physical reasons to use one.
  - or you have a very large data set.

Interpolation

- In some experiments the response y does not have a random component
  - e.g. if we run experiments on some CAE codes.

This doesn't mean we think the code is a perfect model – but CAE modelling errors are unlikely to be random

- In this situation we may want to fit a response surface that passes through (interpolates) our data points.
- Even in this situation, fitting polynomials is unlikely to be effective.

Return To Our Simple Data Set And Try To Interpolate

A 5^{th} order polynomial will pass through all the points

…. but it has a local minimum that may well be spurious

Better interpolation methods include radial basis function (RBF) modelling and kriging

Higher-order polynomials often have spurious oscillations

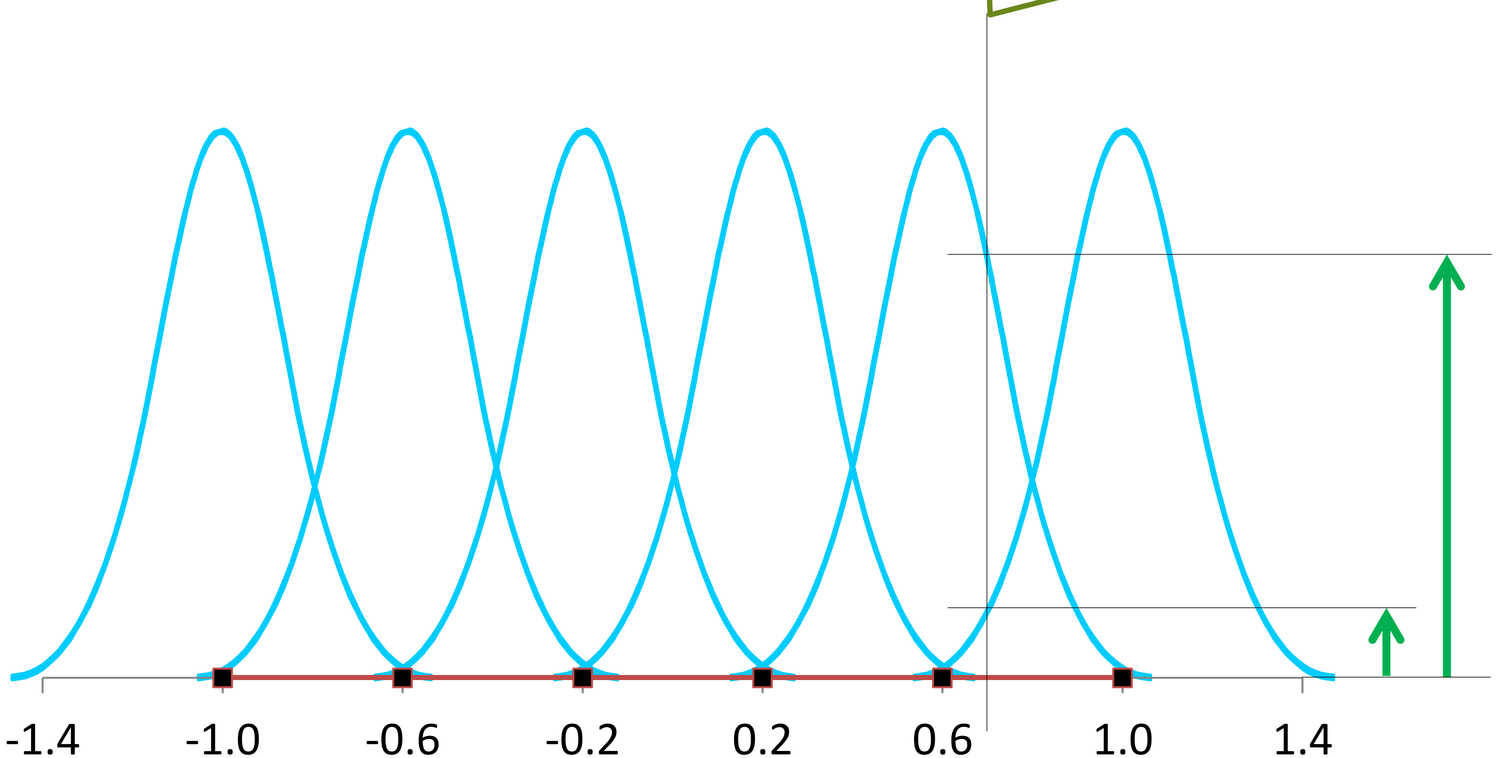
They do not behave like ‘French curves’

They can also be used to produce complicated surfaces from noisy data

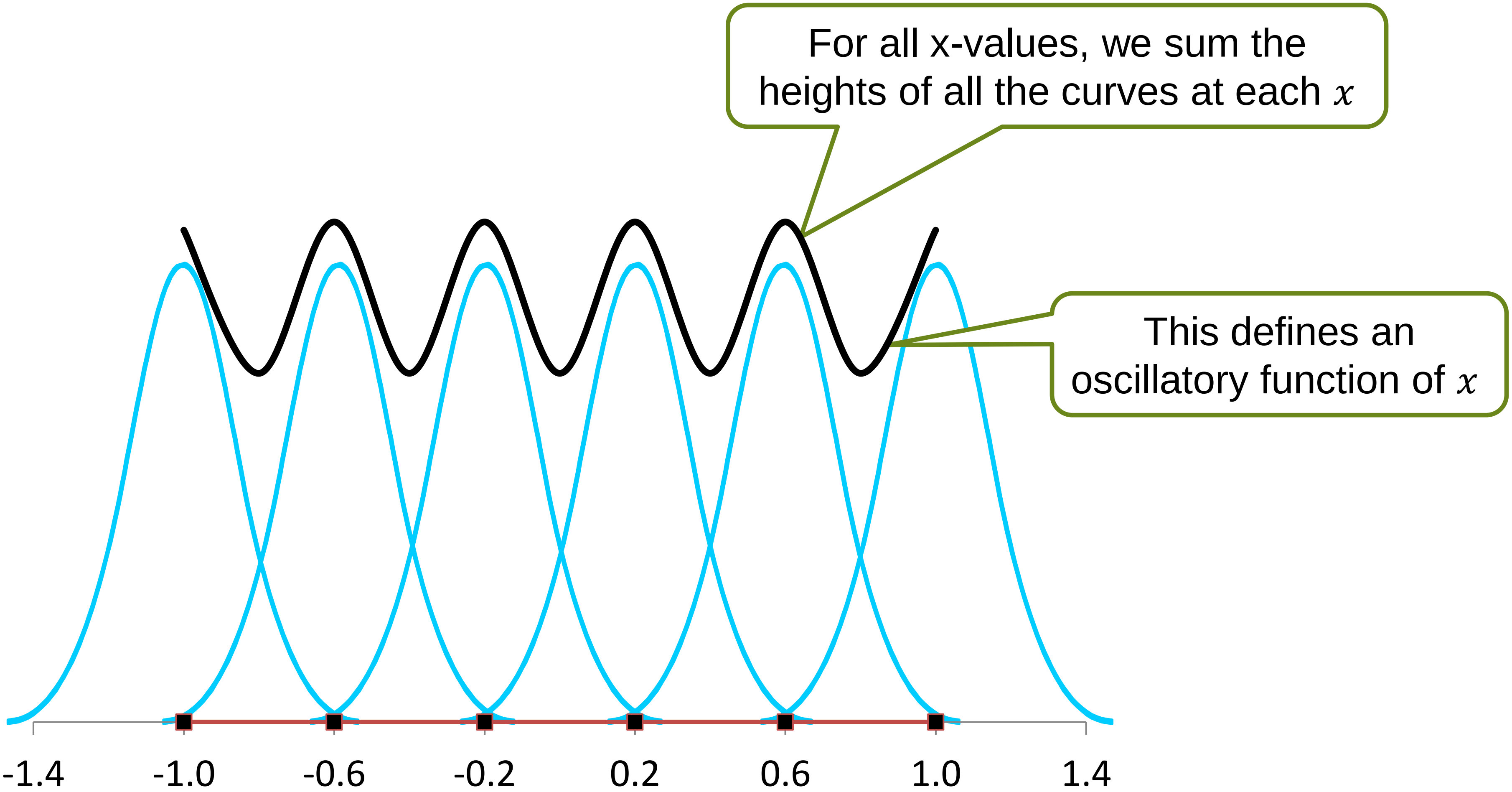
The Fundamentals of RBF Modelling

A Gaussian curve is centred on each x-value

For a given x-value, we sum the heights of all the curves at this x



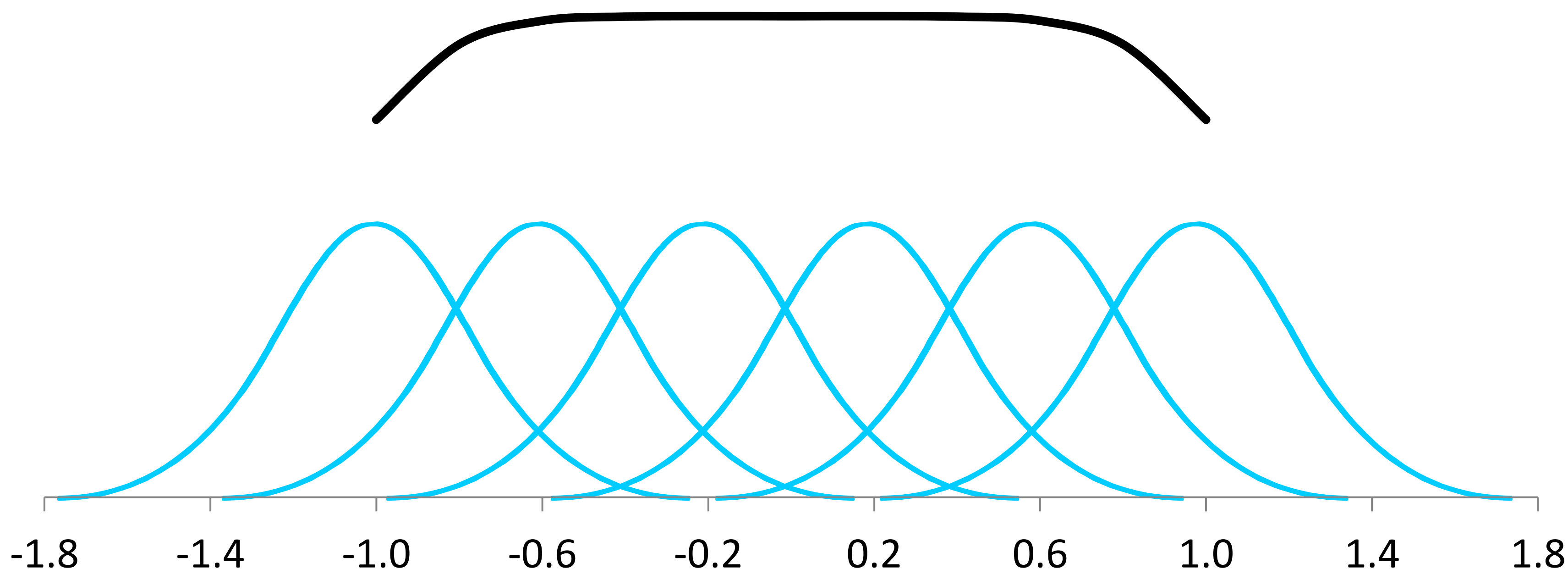
The Fundamentals of RBF Modelling (cont.)



Suppose We Use Wider Kernel Functions ….

The Gaussian curves are called kernel functions or basis functions

If we use wider kernels, we get a smoother result



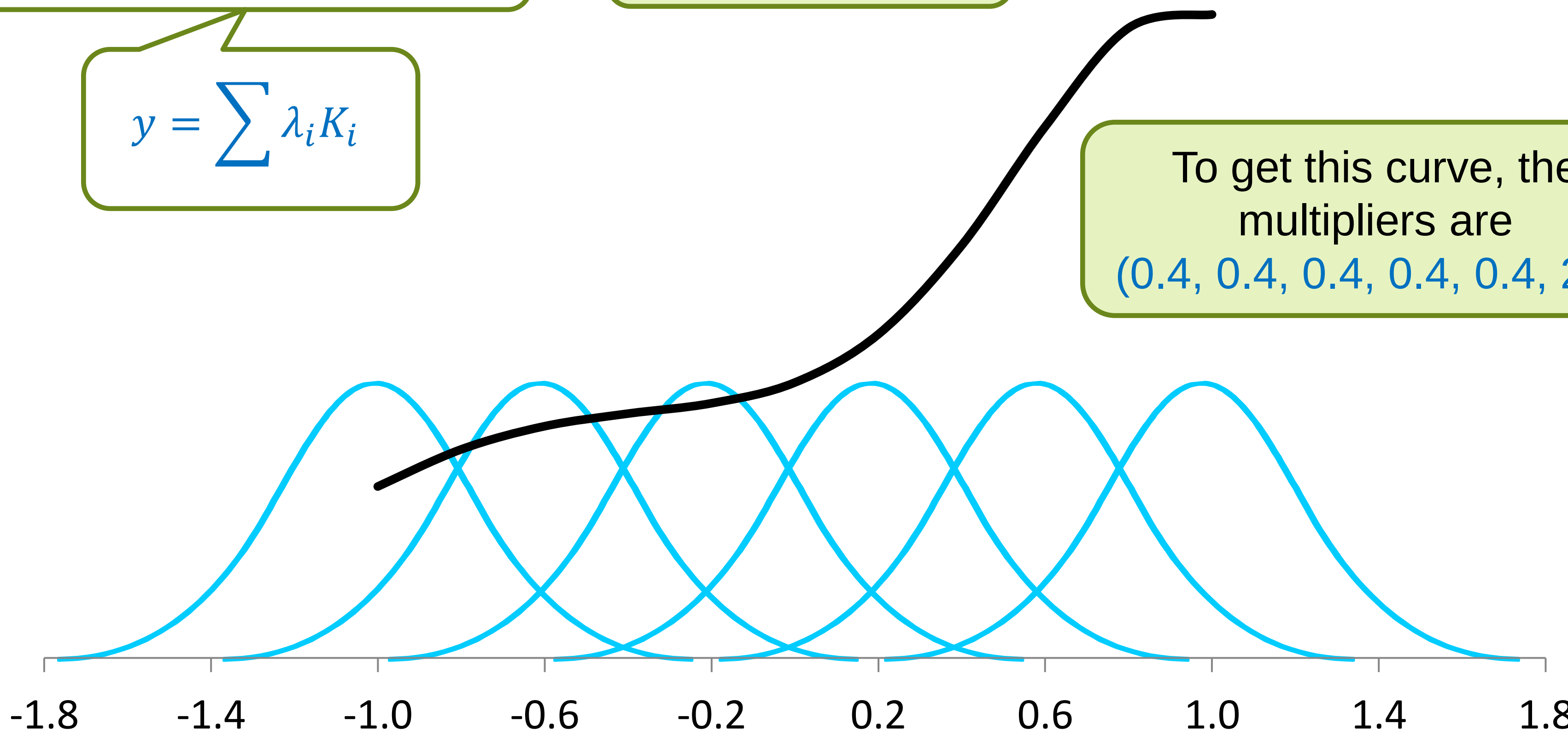
…. And Use Weighted Kernels

Multiply each kernel by a constant before adding up

$$y = \sum \lambda_i K_i$$

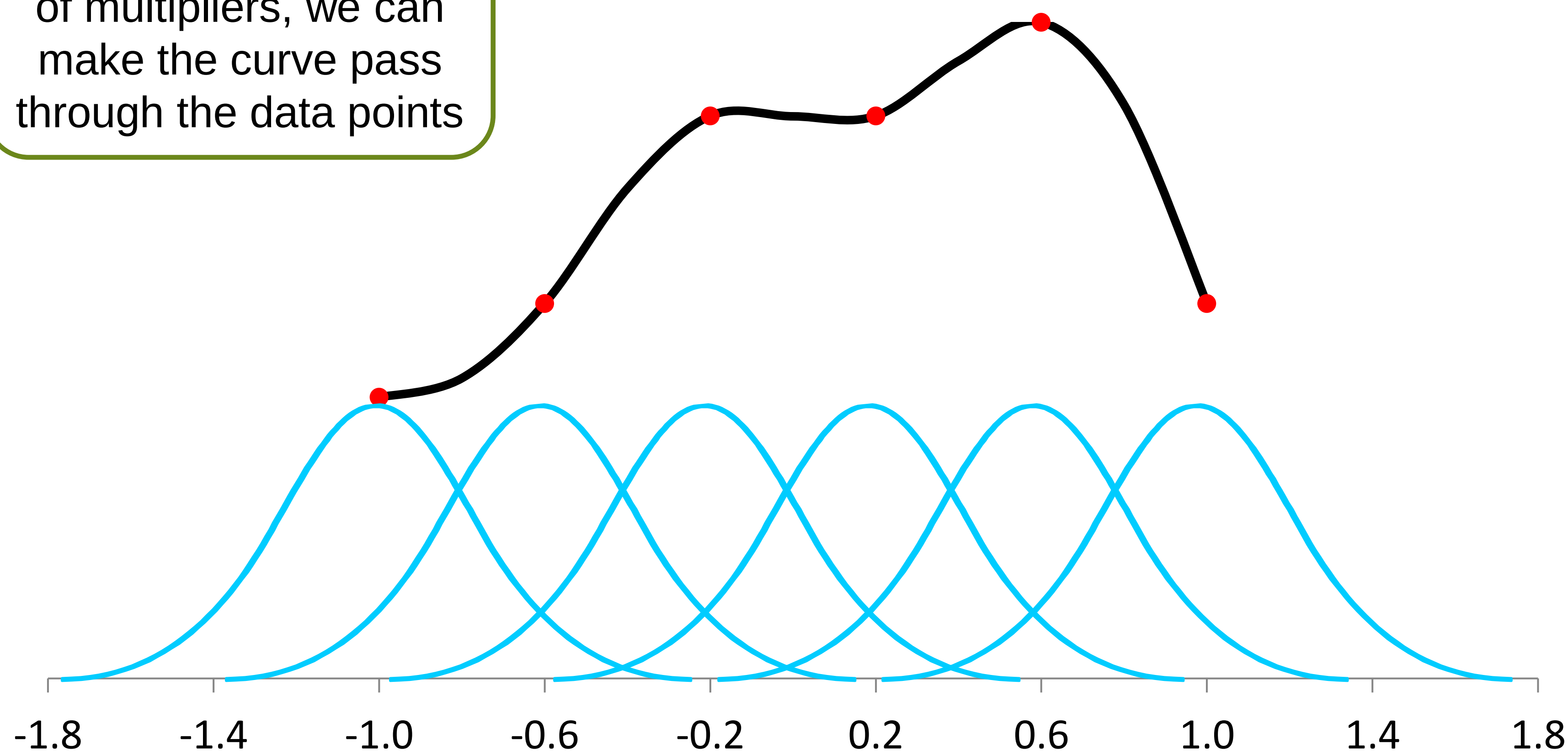
K_i is the i^{th} kernel function

To get this curve, the multipliers are (0.4, 0.4, 0.4, 0.4, 0.4, 2.4)



Using Kernels For Interpolation

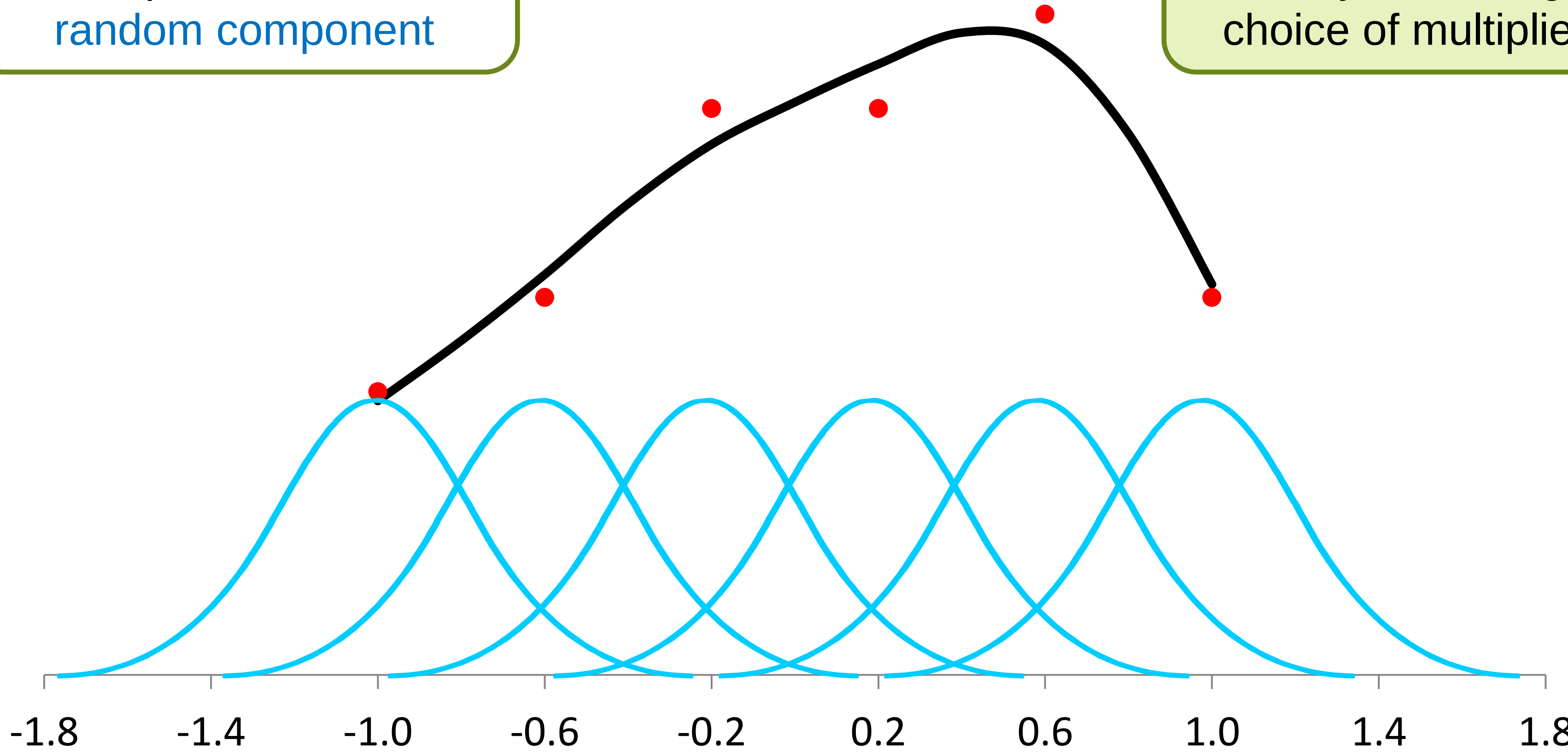
With a suitable choice of multipliers, we can make the curve pass through the data points



Using Kernels With Noisy Data

Suppose we know that the response contains a random component

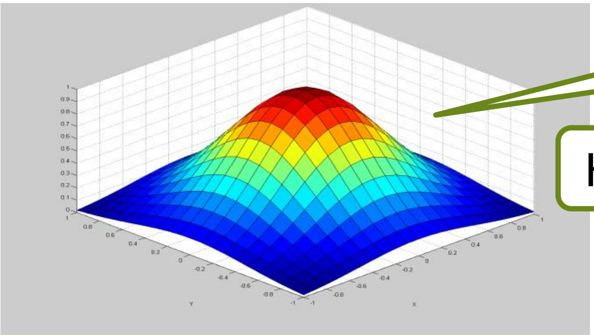
We can get a smoother curve by restricting the choice of multipliers



RBFs In Practice

- When we have more x-variables, we can use a multivariate Gaussian kernel which is symmetrical in the x's.

From MBC Help

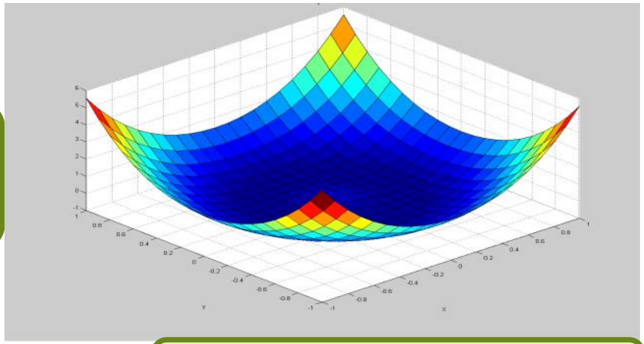


This function is radially symmetric

Hence the name radial basis function

RBF modelling is equivalent to a specific type of Neural Network

MBC has several other types of kernel, e.g.



'Multiquadric' RBF

Using Multivariate RBFs With Noisy Data

- We use fewer kernels than the number of data points.
- We can vary the widths of the kernels (not necessarily the same for all centres).
- We put a constraint on the multipliers to avoid over-fitting.

We try to optimize the choice of centres, widths and multipliers to give good prediction performance

e.g. by cross-validation

Kriging

- Kriging is named after D.G.Krige, a South African mining engineer who developed the method as a way of predicting the value of mineral deposits underground, based on sample results.
- The mathematical basis of kriging is quite complicated, but in practical terms it can be thought of as a generalisation of RBF modelling, in which the kernel functions are not radially symmetric.
  - this is useful because the 'bumpiness' of the surface may be quite different in different directions.
- Kriging models are usually fitted to the data by the method of Maximum Likelihood.

Kriging can be used for either interpolation or smoothing

As with RBF modelling, we can try different shapes of kernel function

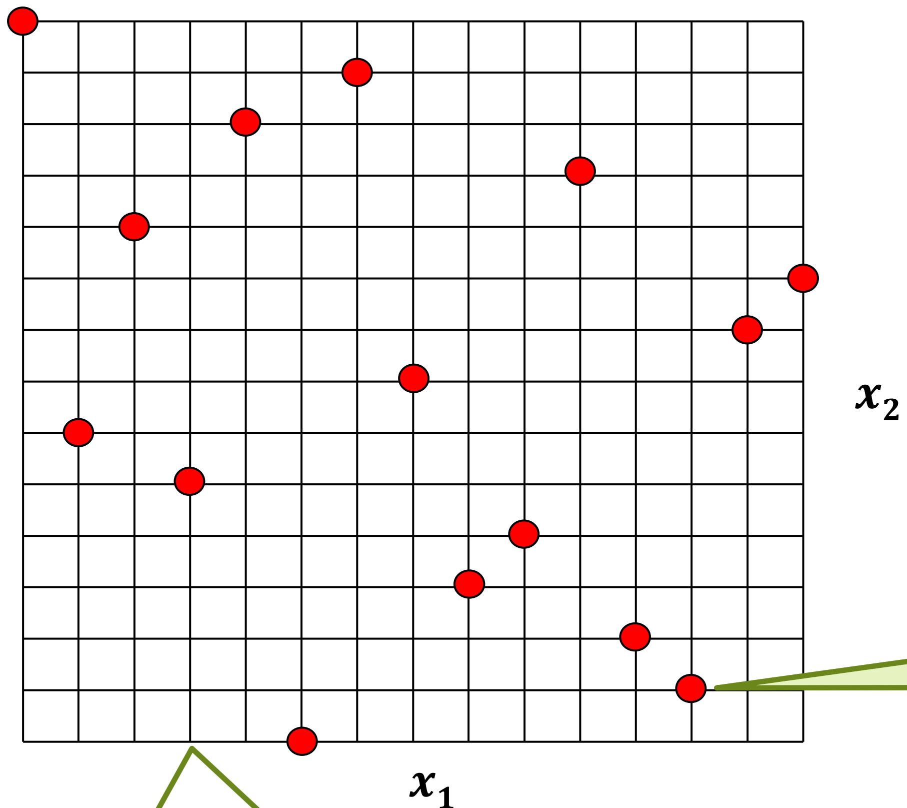
Experiments To Support Flexible Modelling

- Two-level and three-level designs are ideal for fitting low-order polynomials.
- For RBF modelling and kriging we need designs with:
  - more levels,
  - a uniform spread of points in the factor space.

Called space-filling designs

- **Latin Hypercubes** are the most commonly used type of space-filling design.

Example of A Latin Hypercube Design

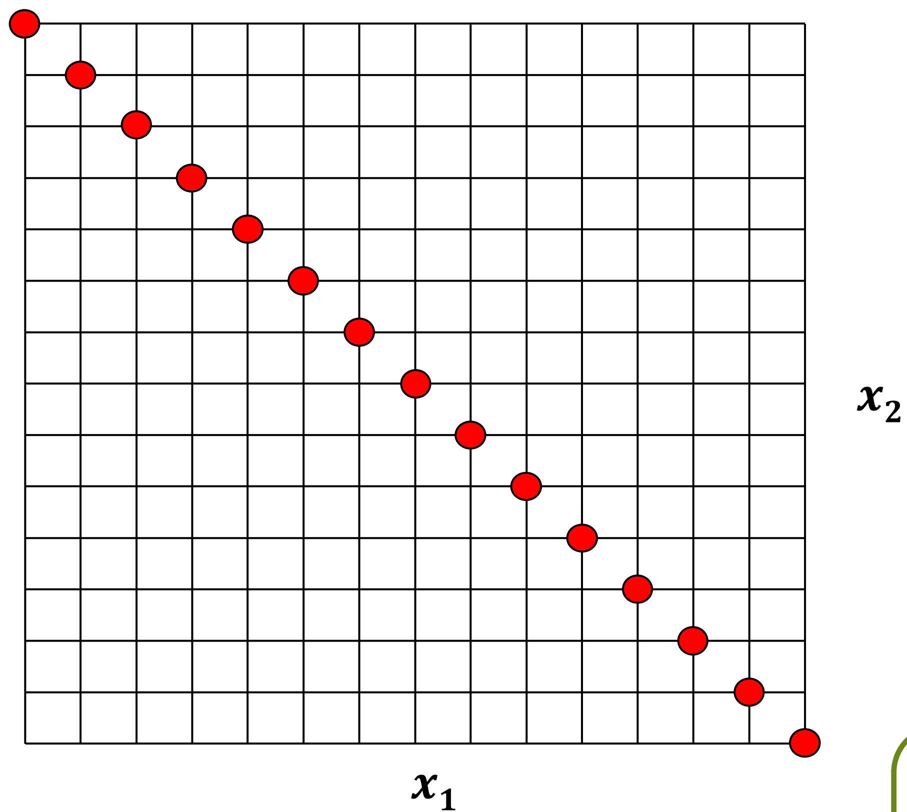


Two factors,
$n = 15$

One run in each row and in each column

15 equally-spaced levels for each factor

Another Latin Hypercube Design



Some LH designs are better than others!

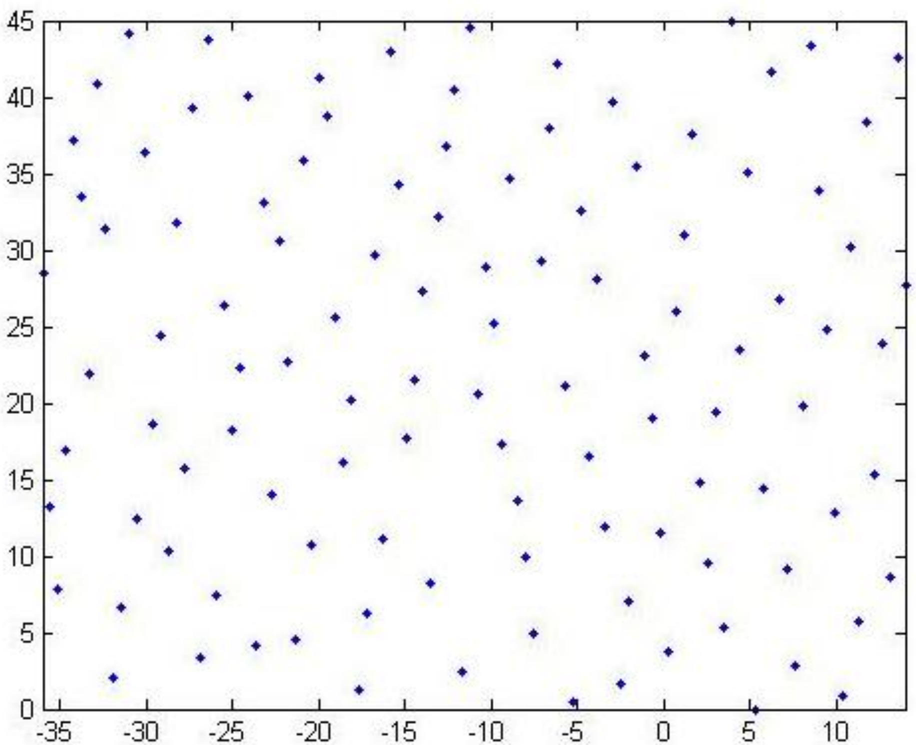
We need to choose a good one

We can use optimization algorithms to find the “best” space filling design according to a given criterion

e.g. maximize the minimum distance between any 2 test points, minimize the standard deviation of the minimum distance, etc

Optimal Latin Hypercube Design

- This OLH is optimized for “Potential Energy”, assuming that the points repel each other.

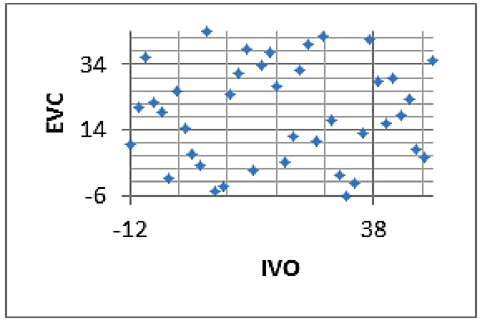


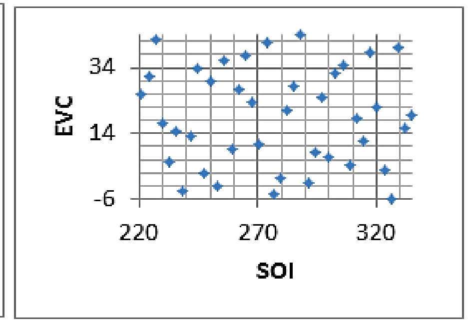
Most OLH algorithms use Genetic Algorithms to generate an optimal solution

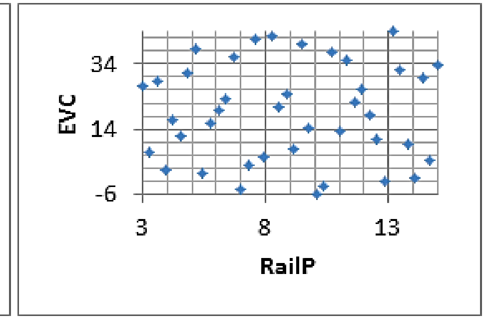
GAs are heavily reliant on “random numbers” and do not converge to an absolute optimum!

So you will get a different OLH every time you run the algorithm…

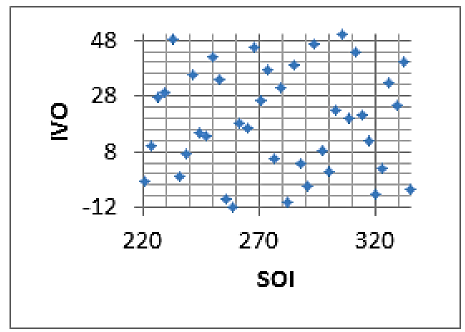
An optimal Latin Hypercube (OLH) In 4 Dimensions

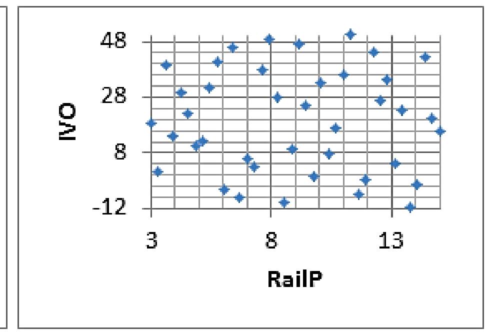




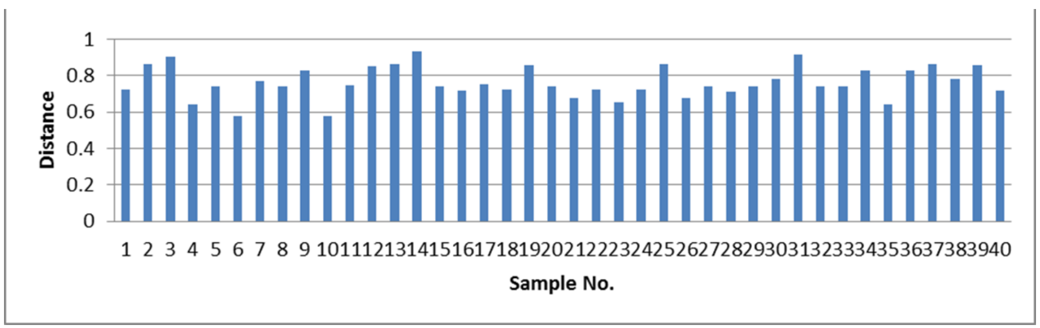


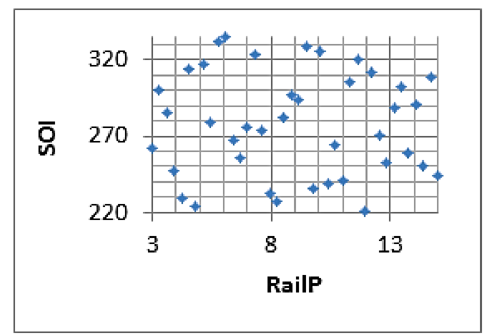
With multiple dimensions the distribution of points is hard to judge…





We can look at the distribution of the minimum distances





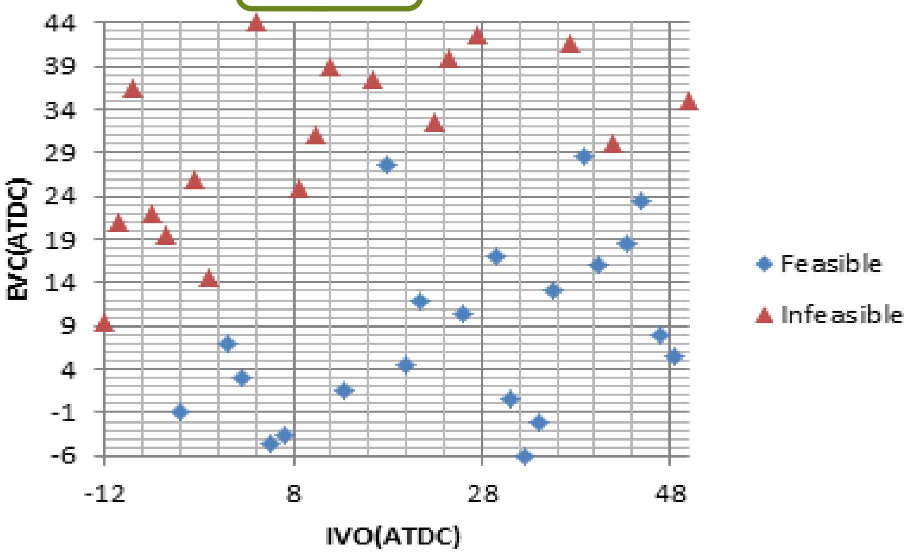
Strategy For Computer Experiments

- For hardware experiments we outlined a sequential strategy with just two stages – screening experiment and follow-up experiment.
- In a computer experiment, where we may get a very complicated response surface, a two-level screening design may not be effective.
- A different type of sequential strategy can be used:
    - run a small Latin Hypercube experiment and fit a surface (e.g. by kriging);
    - use the results to choose another set of points (e.g. in regions where we have most uncertainty about the surface);
    - analyse the combined set of points;
    - continue adding points until the prediction RMSE on a set of validation points is small enough.

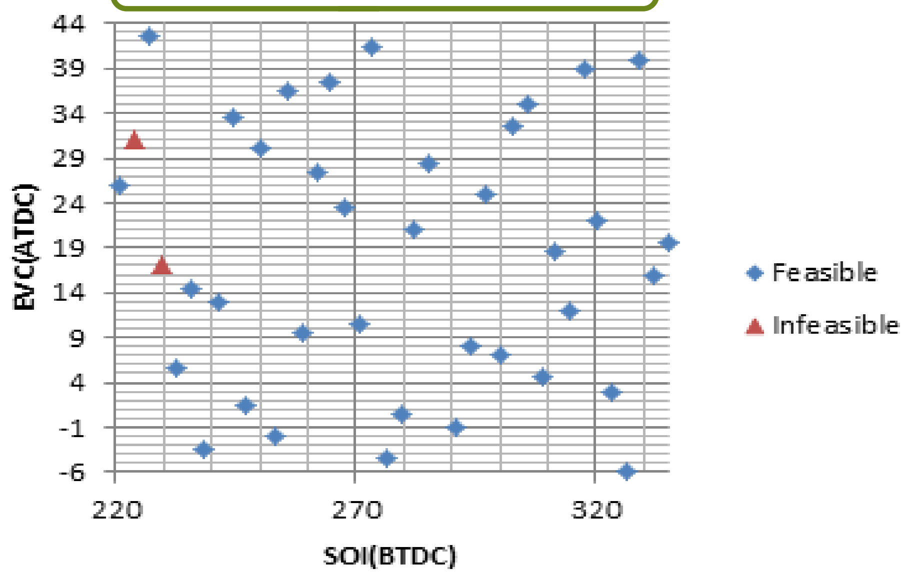
Screening Based On OLH Design

- Illustrated: GDI engine tests – 40 point screening design for 4 variables (EVC, IVO, SOI, RailP).

Idle



Part Load, 2000 RPM



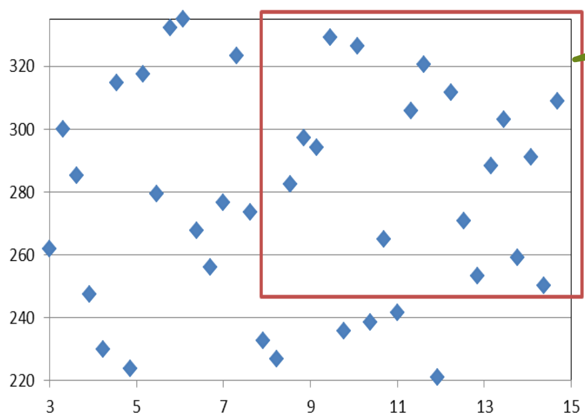
- Conclusion:
  - based on combustion stability we need $EVC < 30,\ IVO > EVC - 10$.

Kianifar, M., Campean, I.F., and Richardson, D., "Sequential DoE Framework for Steady State Model Based Calibration," SAE Int. J. Engines 6(2):843-855, 2013, doi:10.4271/2013-01-0972.

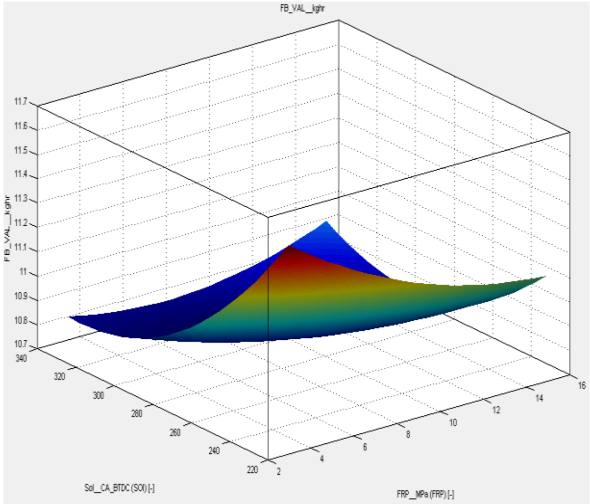
Screening Based On OLH Design (cont.)

- Constraints on SOI and RailP were chosen based on the fitted surfaces:
  - $SOI\ >\ 260\ degBTDC$
  - $RailP\ >\ 8\ bar$

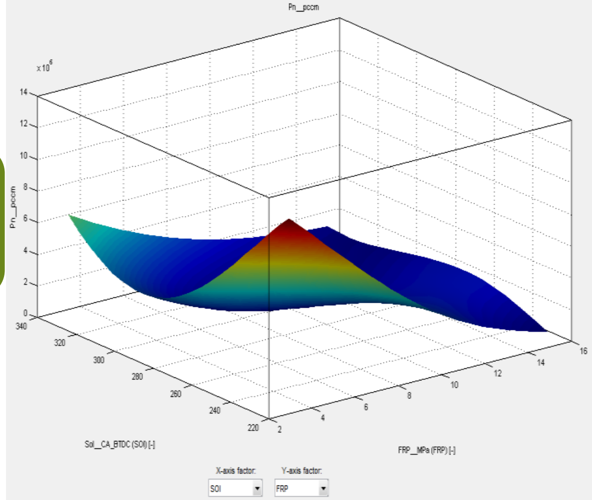
Area for model-building DoE



Fuel (kg/hr)

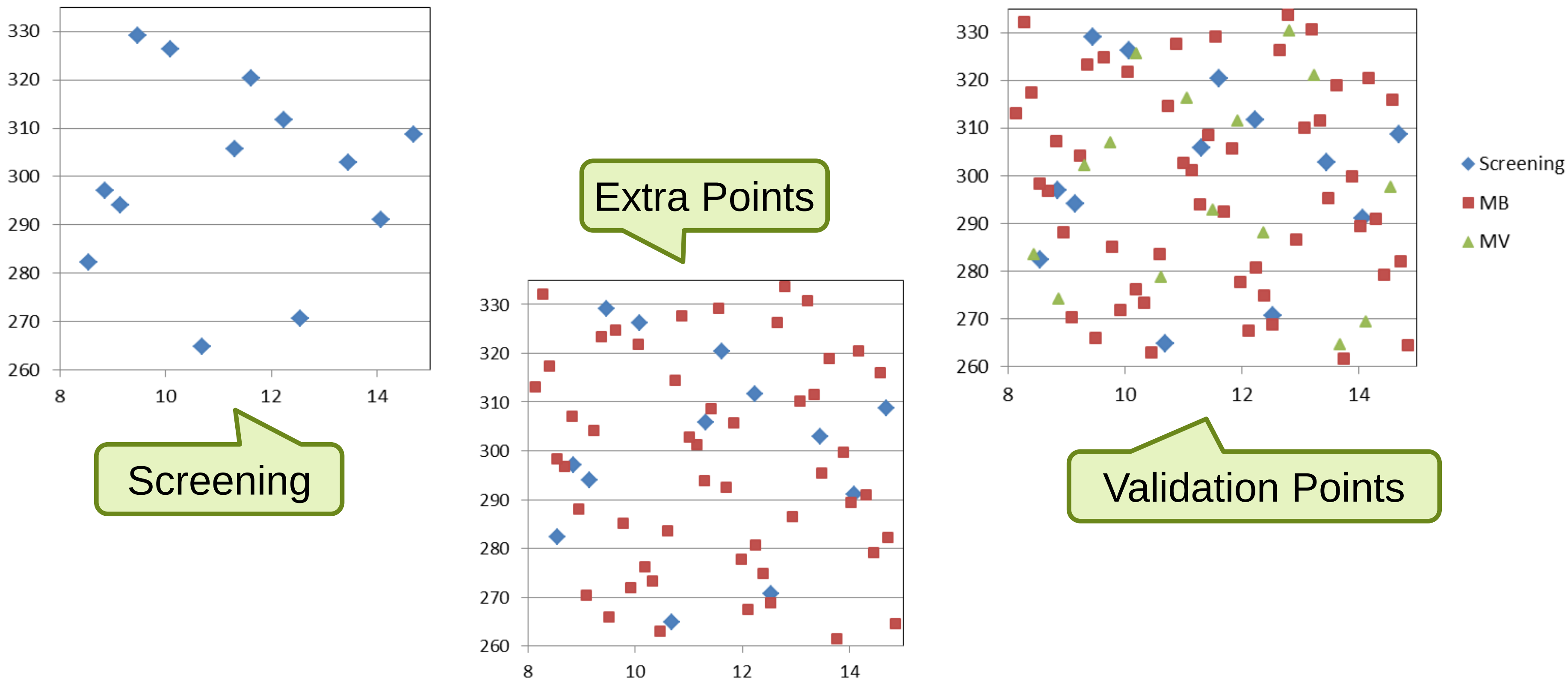


Pn (pccm)



Augmenting The Screening Design

- Augmentation based on Optimal Space Filling Designs.

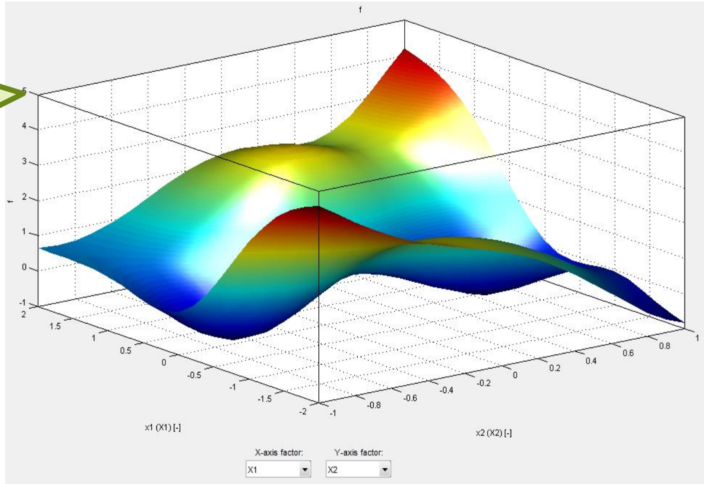


RSM DoEs – Another example

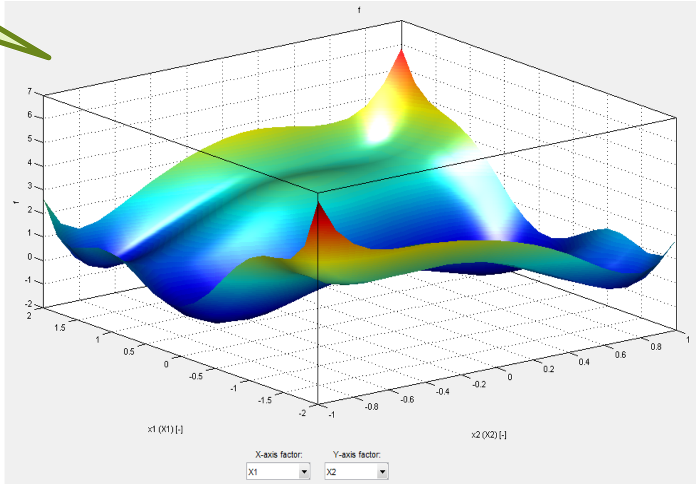
Model Built in four steps:

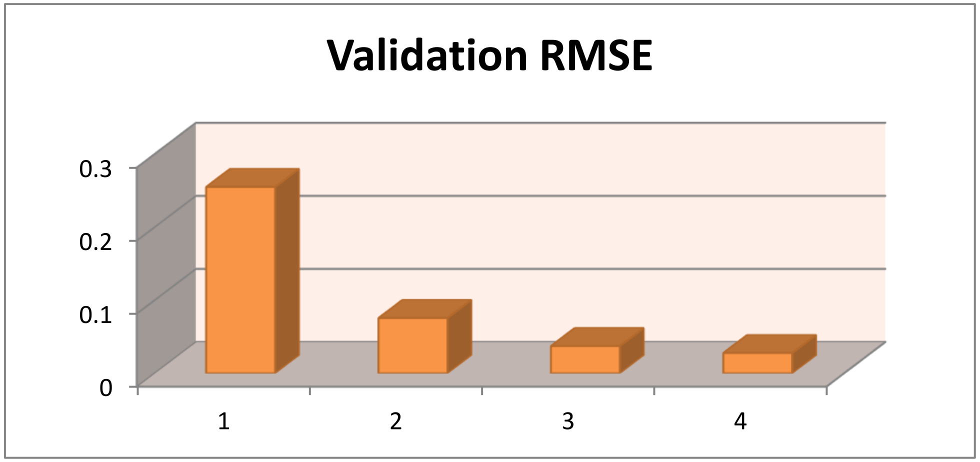
- Step1(60 MB- 40 MV)
- Step2(100 MB- 30 MV)
- Step3(130 MB- 20 MV)
- Step4(150 MB- 15 MV)

Test 1: MB60-MV40
RBF Model



Test 4: MB150-MV15
RBF Model





In This Session We Have…

- Discussed the concept of over-fitting and potential problems with using higher-order polynomials.
- Introduced two new methods (radial basis functions and kriging) which are effective for interpolation and also for smoothing noisy data.
- Introduced Latin Hypercube (LH) designs.
- Compared strategies for hardware and computer experiments.
- Applied space-filling designs to engine testing.