Surrogate Modelling

A review with a focus on Kriging

In[61]:=
```
showCredentials[]
```

Out[61]=

Laurens Bogaardt

l.bogaardt@esciencecenter.nl

Netherlands eScience Center

University of Amsterdam

Computational Science Lab

2019-02-01

Content

- History
- Problem
- Applications
- Mathematics of Kriging
- Extensions of Kriging
- Other Surrogate Models
- Related Techniques
- Domains and Communities
- Theoretical Problems
- Implementation Problems
- Implementations

If you have any questions, just interrupt me.

History

“Gold values in a whole mine will be subject to a larger relative variation than those in a portion of the mine” [Krige, 1951, page 124].

“Two neighbouring samples are certainly not independent” [Matheron, 1963, page 1248].

“The variogram is an increasing function of distance” [Matheron, 1963, page 1250].

“Often, the codes are computationally expensive to run, and a common objective of an experiment is to fit a cheaper predictor of the output to the data” [Sacks et al., 1989, page 409].

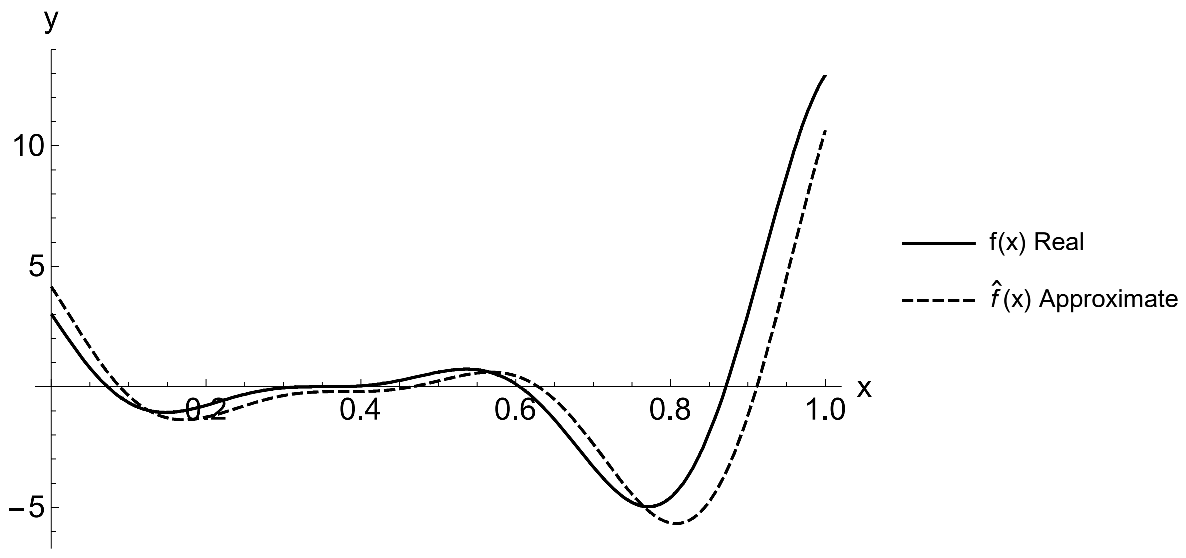
“Our statistical model, adopted from kriging in the spatial statistics literature, [...] treats the response as if it were a realization of a stochastic process” [Sacks et al., 1989, page 409].

“With this model, estimates of uncertainty of predictions are also available” [Sacks et al., 1989, page 409].

Problem

In[62]:=
```
showExample[]
```

Out[62]=



Timing-consuming *physical* experiment:

- What is the distribution of birds in this forest?
- What is the total amount of gold in this plot of land?

Timing-consuming *computer* experiment:

- Which shape-parameters of an aerofoil (wing) have little influence on the drag according to my Computational Fluid Dynamics software?
- Which configuration gives the global optimum of stability in this structure?

Surrogate model synonyms:

- Meta-models
- Response surface models
- Emulators

Applications

- Interpolation and Understanding
- Differentiation and Integration
- Sensitivity Analysis
- Optimisation
- Parameter Estimation
- Lazy Evaluation and Memoisation

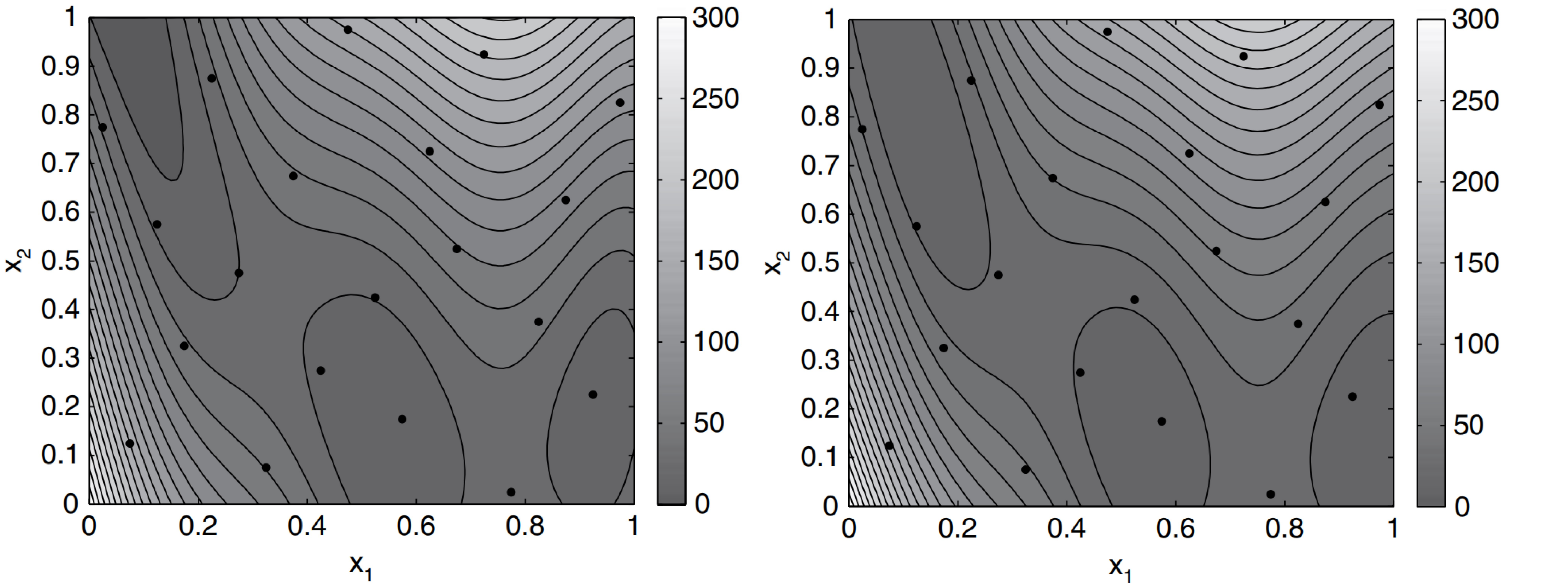
Applications

Interpolation and Understanding

- What is the distribution of birds in this forest?

In[63]:=
```
Show[Import[ToString[NotebookDirectory[]] <> "Figures\\BraninFunctionAndSurrogate.png"], ImageSize → 1300]
```

Out[63]=



The 2D Branin function and its surrogate, taken from [Forrester et al., 2008, page 63].

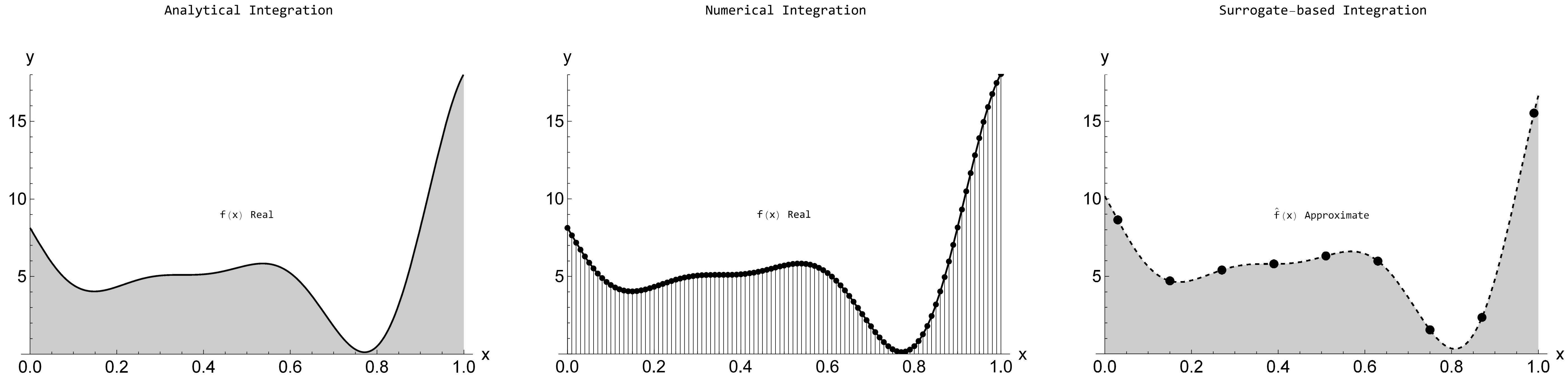
Applications

Differentiation and Integration

- What is the total amount of gold in this plot of land?

In[64]:=
```
generateIntegrationExample[]
```

Out[64]=



Text

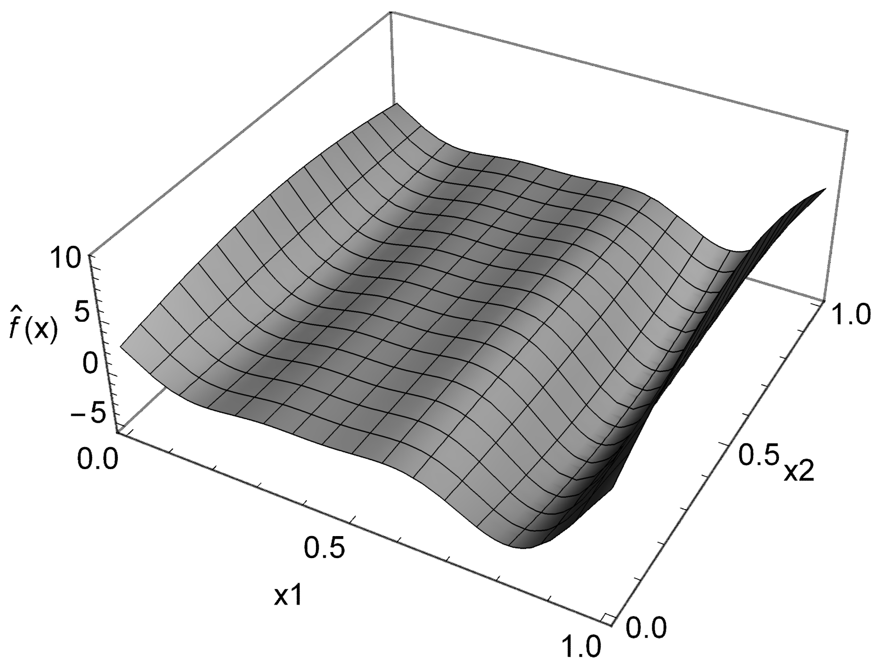
Applications

Sensitivity Analysis

- Which shape-parameters of an aerofoil (wing) have little influence on the drag according to my Computational Fluid Dynamics software?

In[65]:=
```
generateSensitivityExample[]
```

Out[65]=



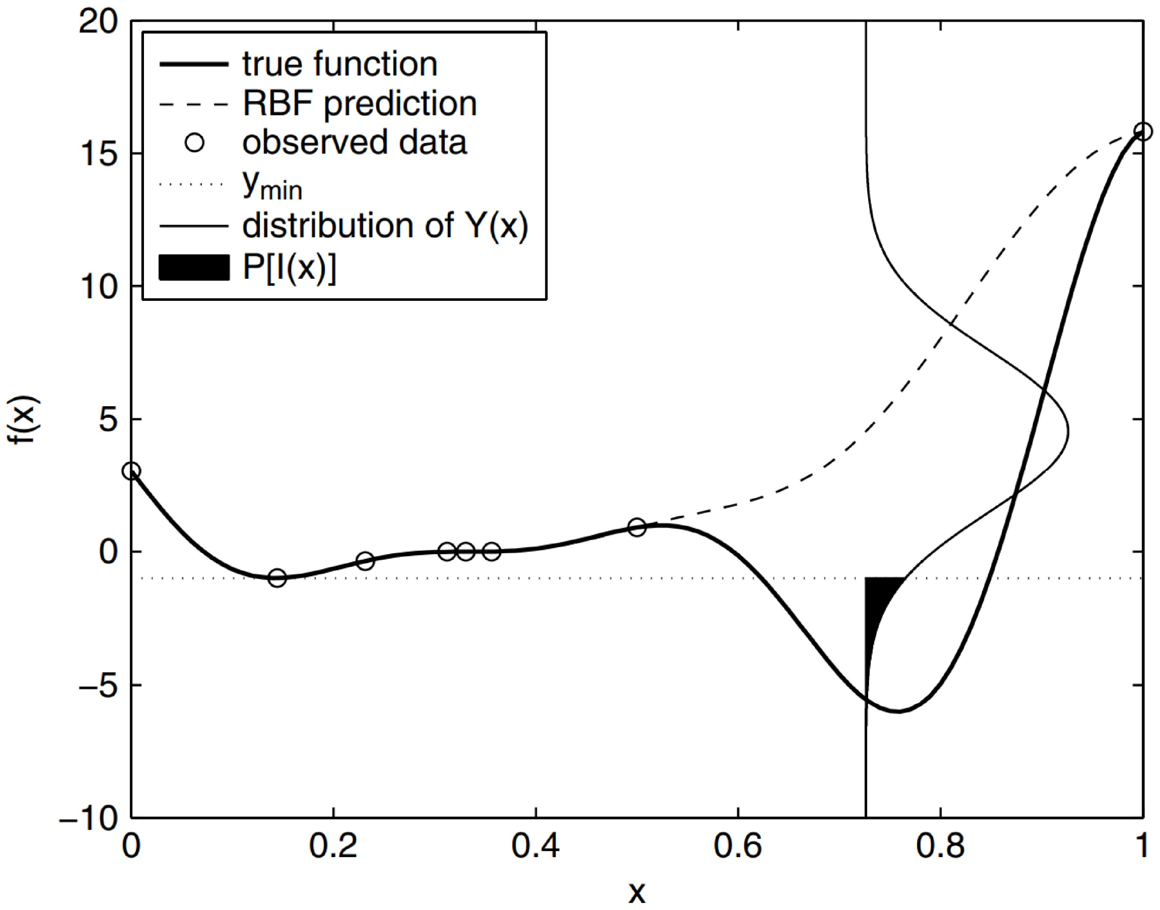
f(x) is more sensitive to x1 than to x2.

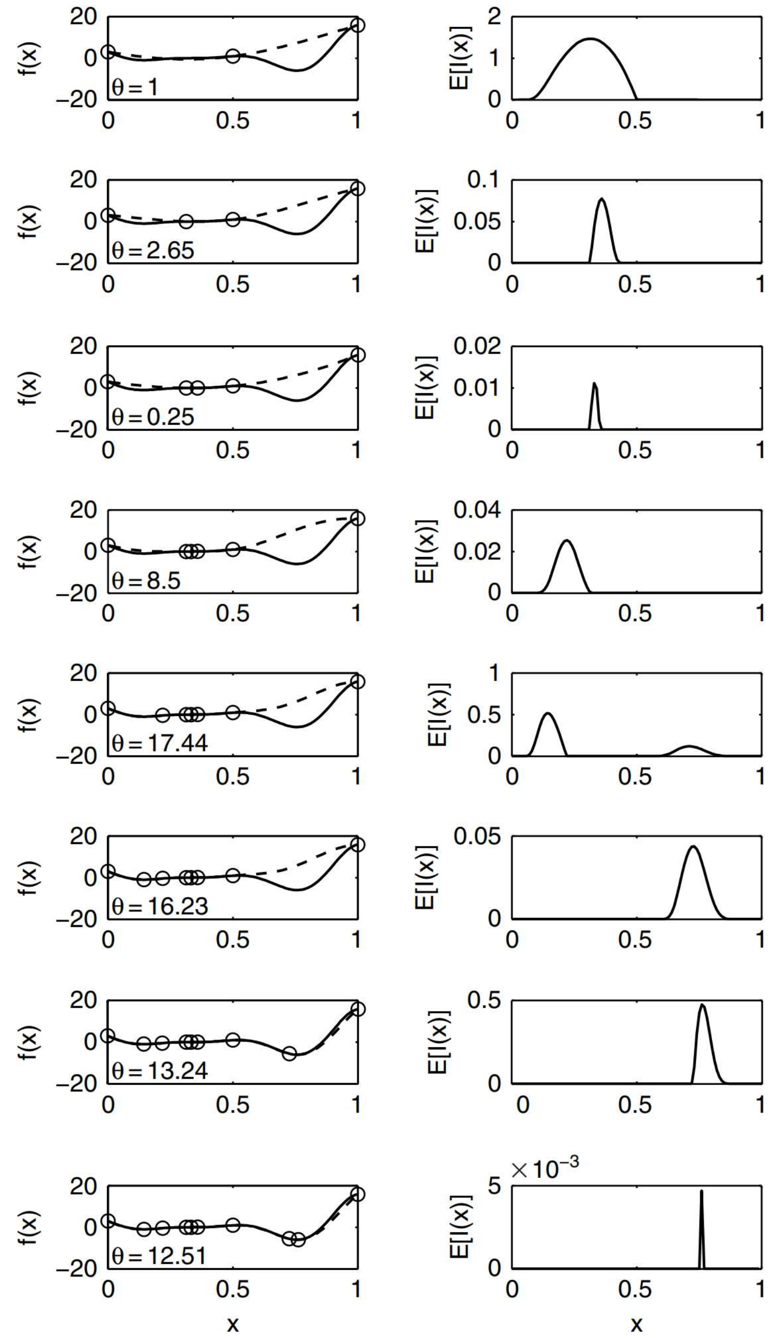
Optimisation

- Which configuration gives the global optimum of stability in this structure?

In[66]:=
```
Grid[{{Show[Import[ToString[NotebookDirectory[]] <> "Figures\\ProbabilityOfImprovement.png"], ImageSize → 600],
   Show[Import[ToString[NotebookDirectory[]] <> "Figures\\IterationsOfExpectedImprovement.png"], ImageSize → 390]}}, Spacings → {5, 2}]
```

Out[66]=





The surrogate provides a probability distribution at each x. This can be integrated to determine the probability of an improvement.

Applications

Parameter Estimation

- What parameters of my model of Black Holes and Gravitational Waves gives the best fit with observations?

Binary black hole system emits gravitational waves described by 7 parameters: two black hole spin vectors and the ratio of their masses

Simulating a binary black hole coalescence by solving Einstein’s equations is computationally expensive (days to months of supercomputing time)

[Blackman et al, 2015, 2017a, 2017b] create a surrogate which can be evaluated in 50 ms

It’s suitable for performing parameter estimation studies on gravitational wave detections

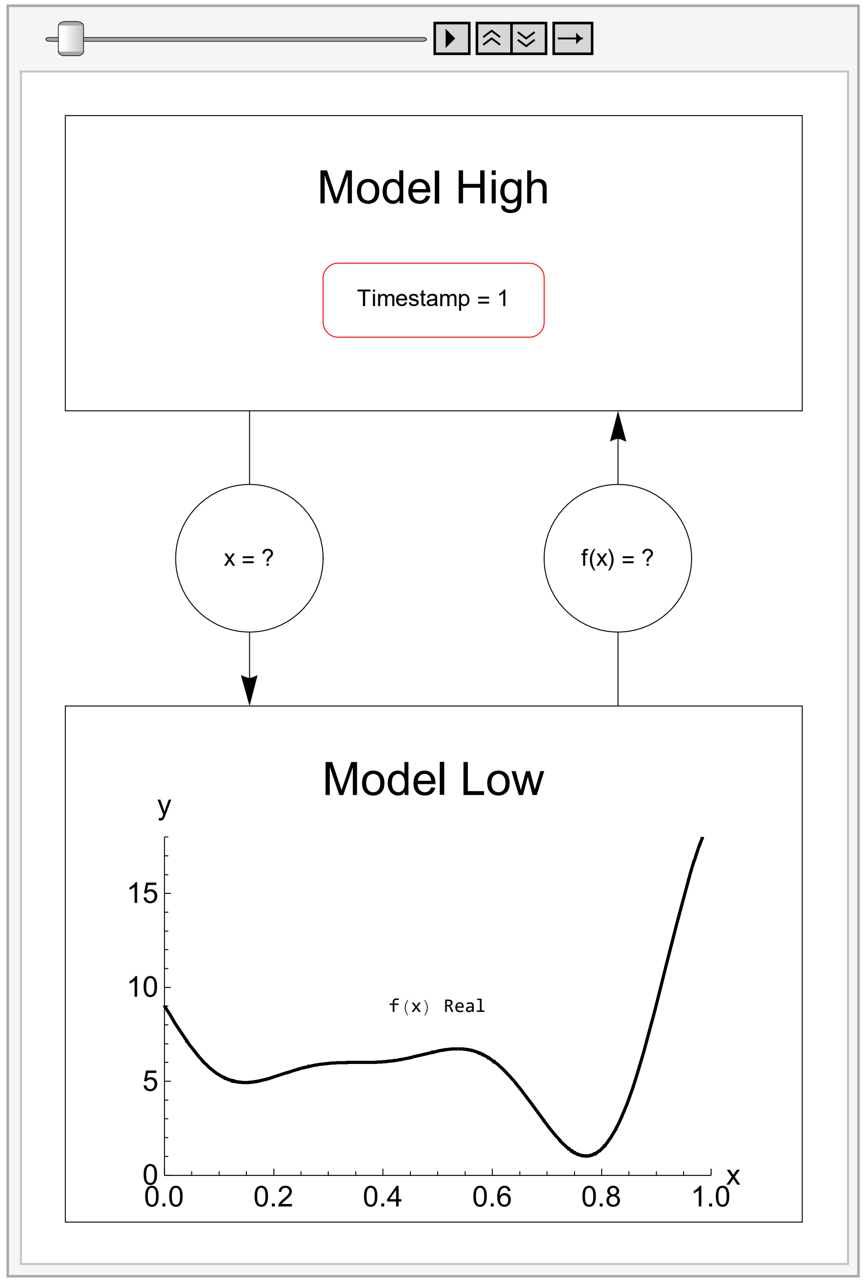
Applications

Lazy Evaluation and Memoisation

Useful application in Multi-Scale Modelling

In[67]:=
```
ListAnimate[animationOfLazyLoading, AnimationRunning → False, AnimationDirection → Forward, AnimationRate → 0.8]
```

Out[67]=



Mathematics of Ordinary Kriging

Take a function $f(x)$ which is slow in its evaluation. We wish to approximate this by a function $\hat{f}(x)$, which evaluates fast.

Surrogate models are educated guesses as to what $f(x)$ might look like, based on a few points in space where we can afford to measure the function values.

We sample the slow function a small number of times with different input values. We build up a statistical model of all functions and then incorporate what we know so far. We then infer the conditional distribution of the non-sampled inputs, given the sample of data.

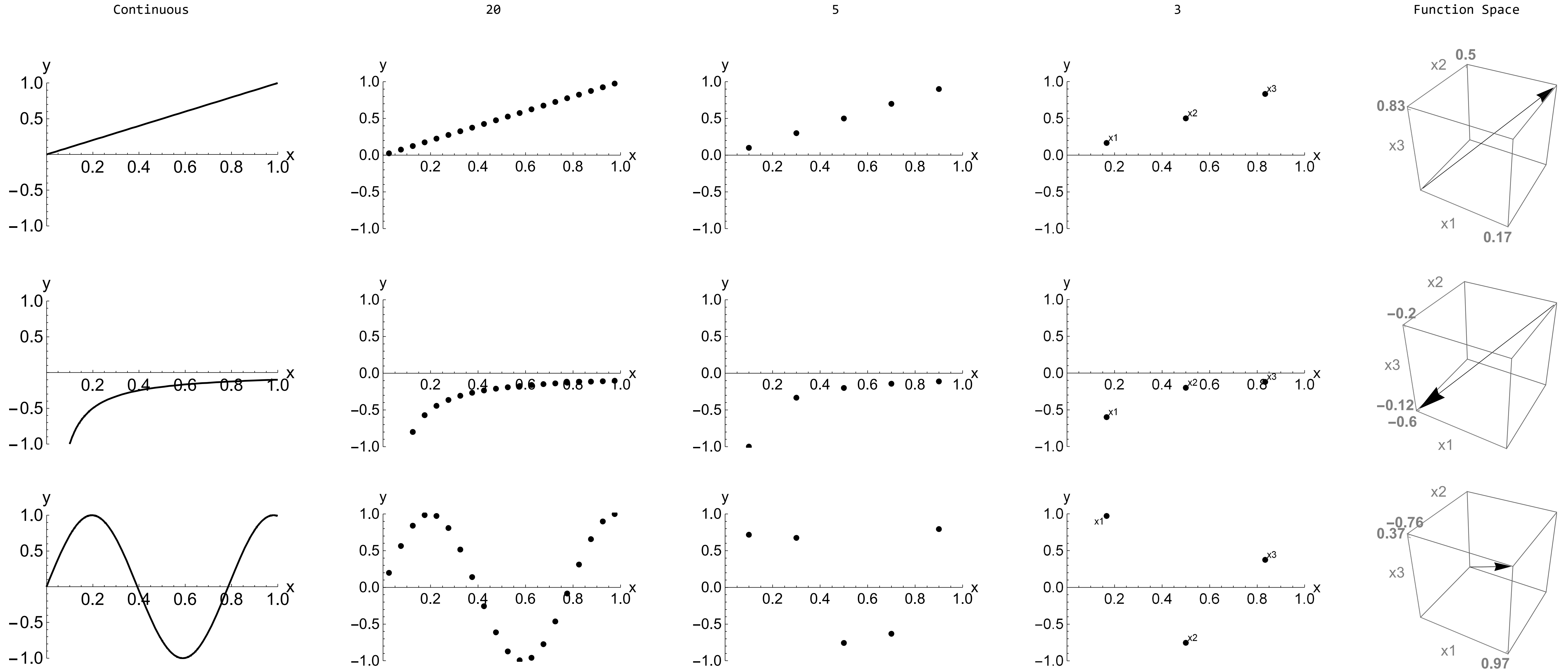
For now, we make the simplifying assumptions that:

- The output y is one dimensional
- The input x is one dimensional
- The input is discrete

Function Space

In[68]:=
```
generateFunctions[functions]
```

Out[68]=



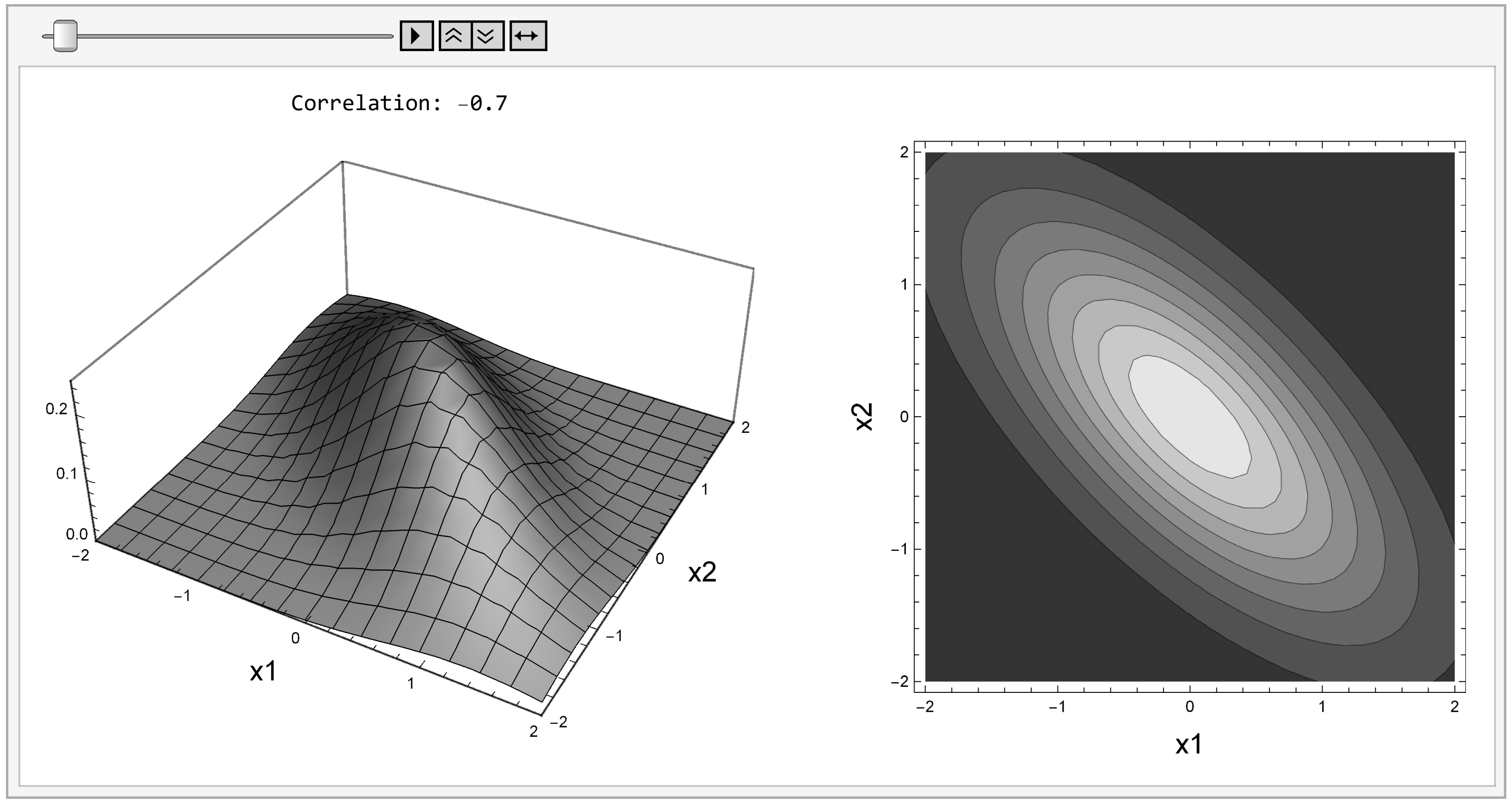
Gaussian Process

We add an additional assumption:

- The space of all functions has a Gaussian probability distribution

In[69]:=
```
ListAnimate[animationOfProbabilitySpace, AnimationRunning → False, AnimationDirection → ForwardBackward]
```

Out[69]=

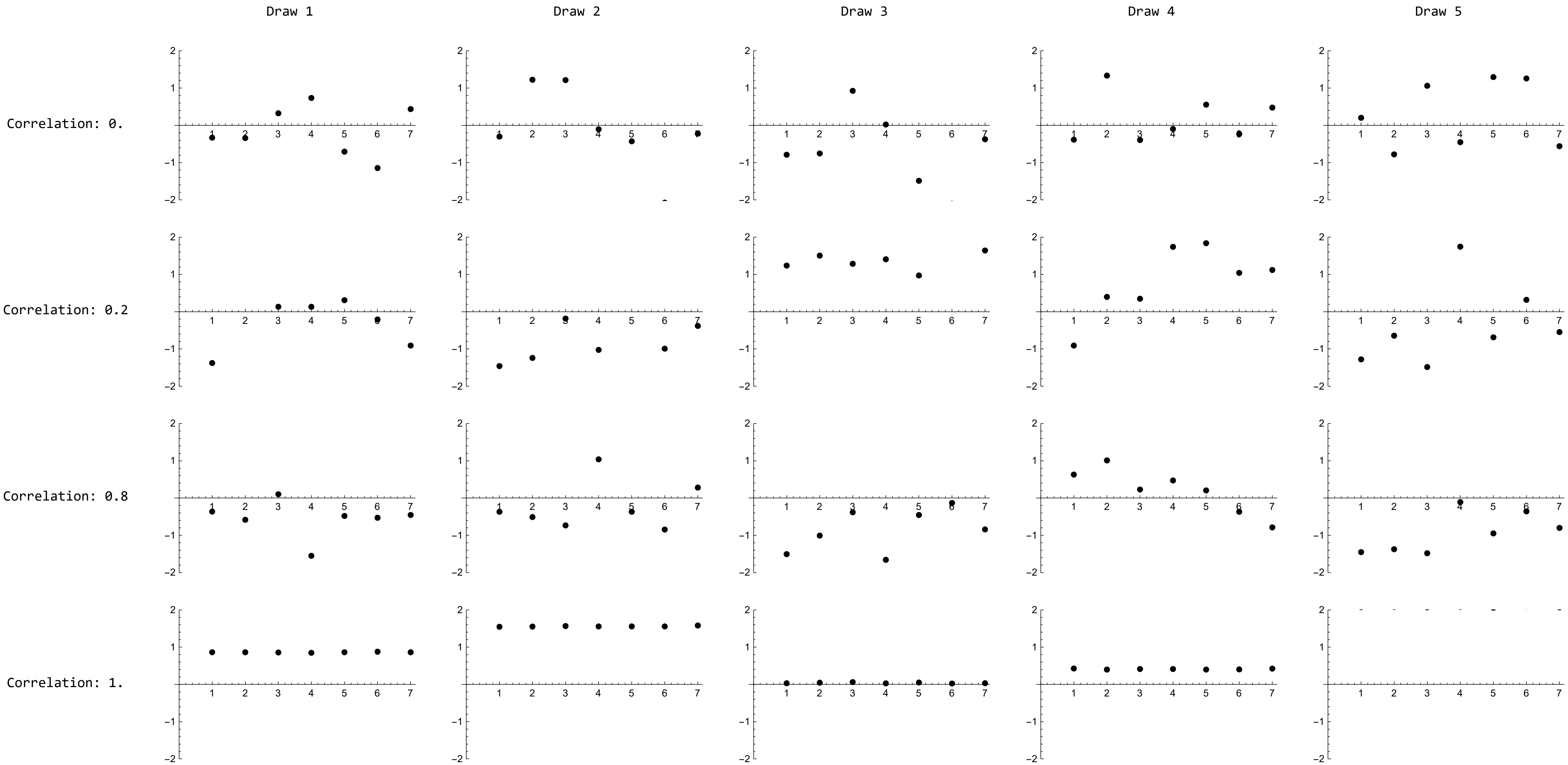


7D Gaussian Process

The probability space has 7 dimensions, our function still only 1 dimension.

In[70]:=
```
generateGrid[]
```

Out[70]=



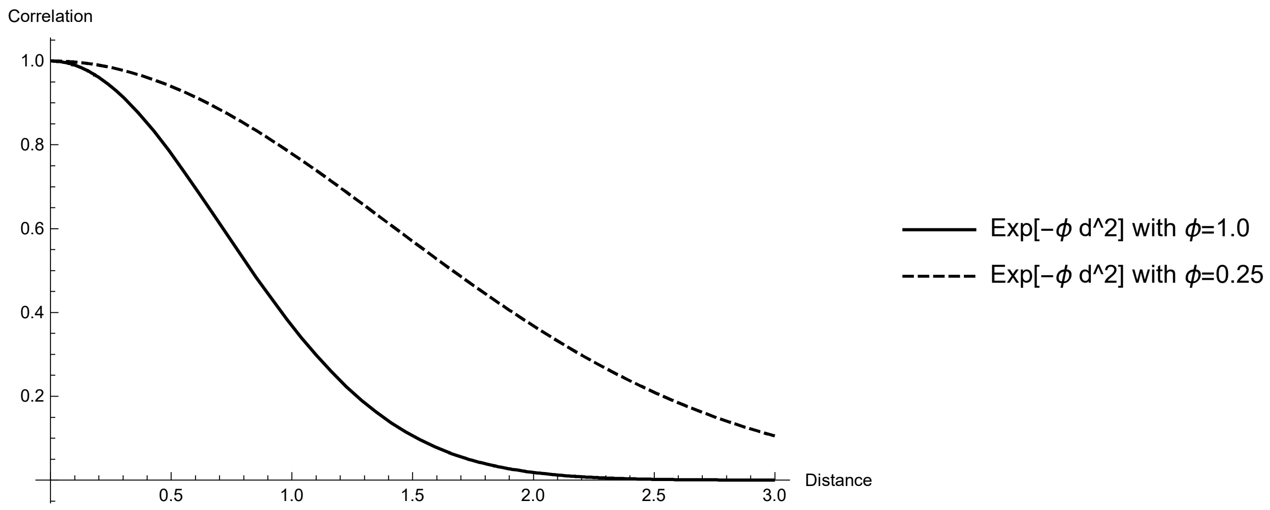
Correlation Functions

We add an additional assumption:

- Inputs close together lead to higher correlation outputs. This is equivalent to assuming the functions we are trying to approximate are somewhat continuous.

In[71]:=
```
generateCorrelationPlot[]
```

Out[71]=

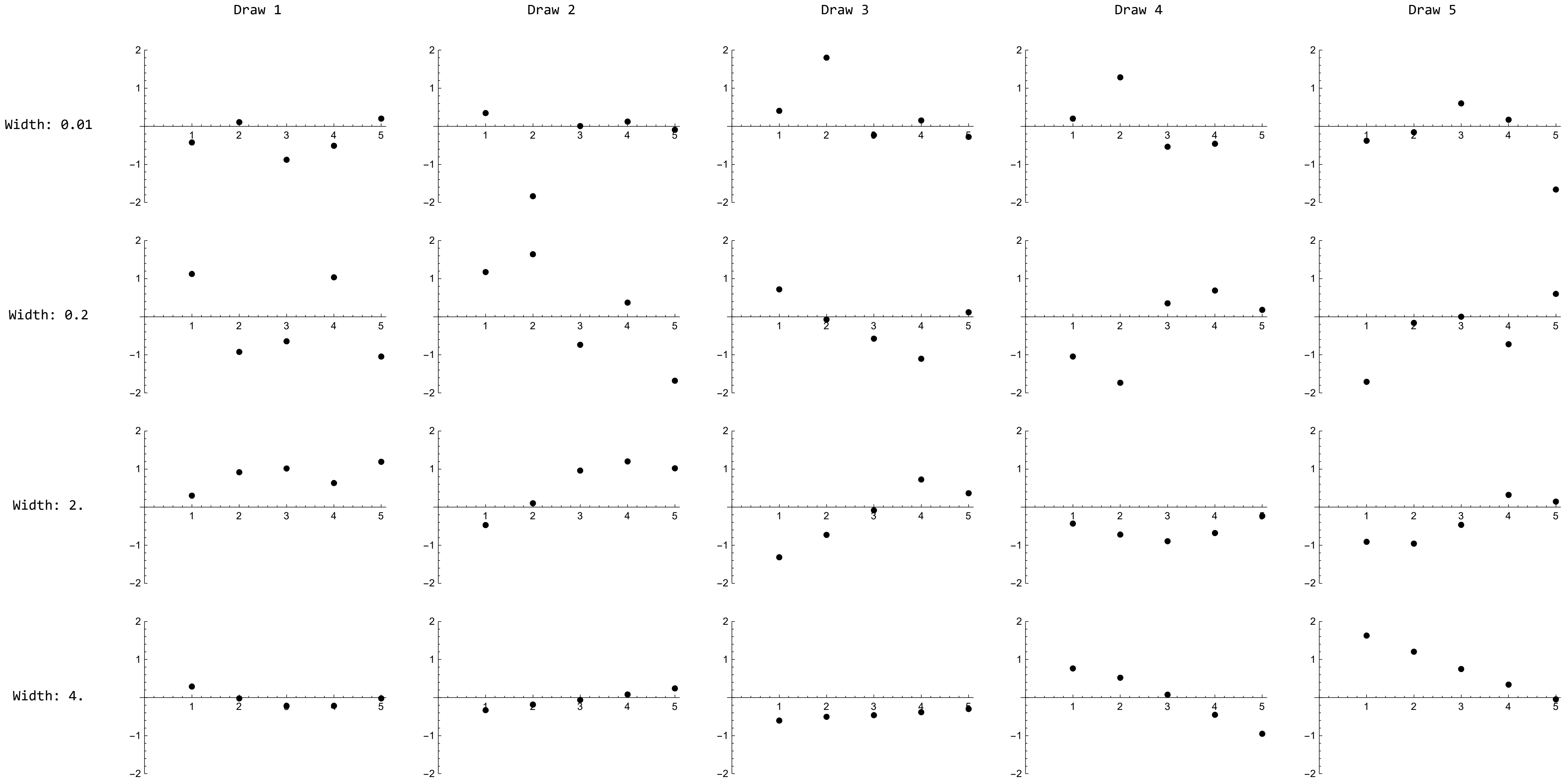


Gaussian Process with Correlation Function

We add an additional assumption:

In[72]:=
```
generateGrid2[]
```

Out[72]=

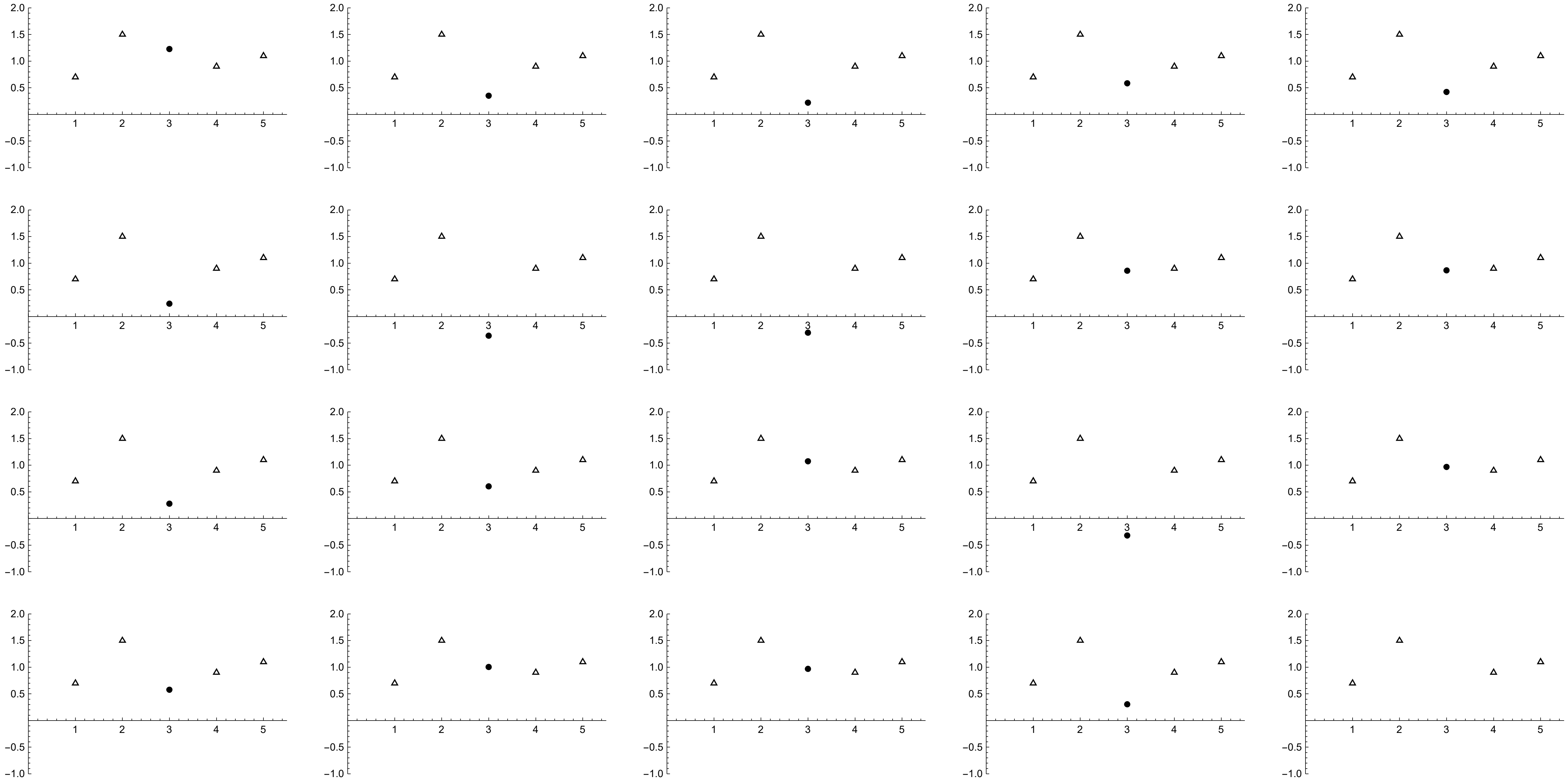


Gaussian Process given n-1 values

We keep 4 out of 5 values fixed. What will the 5th value be?

In[73]:=
```
generateGrid3[givenXs = {1, 2, 4, 5}, givenYs = {0.7, 1.5, 0.9, 1.1}, unknownX = 3]
```

Out[73]=



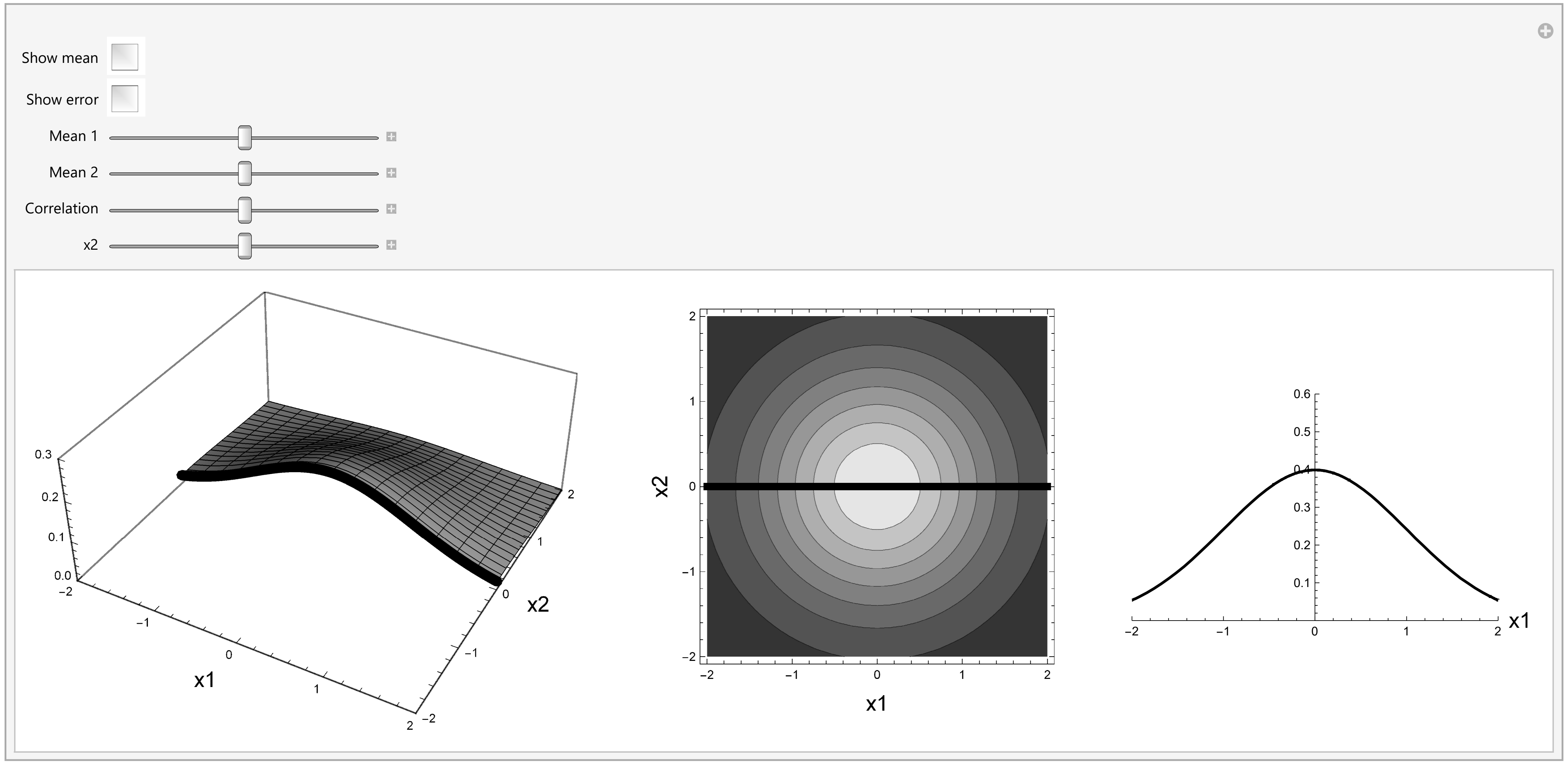
2D Gaussian Process

Visualisation

We assumed all functions were discrete because this allows us to visualise their probability space.

In[74]:=
```
generateManipulate[]
```

Out[74]=



2D Gaussian Process

Equations

A 2D binormal distribution for x1 and x2, conditional on a given value for x2, results in a 1D normal distribution with

- $\mu = \mu_1 + \rho(x_2 - \mu_2)$
- $\sigma = \sqrt{1 - \rho^2}$

Let's double check:

In[75]:=
```
Simplify[PDF[MultinormalDistribution[{μ1, μ2}, {{1, ρ}, {ρ, 1}}], {x1, x2}] / PDF[NormalDistribution[μ2, 1], x2]]
```

Out[75]=
$$\frac{e^{\frac{(x1-\mu1+(-x2+\mu2)\ \rho)^2}{2\left(-1+\rho^2\right)}}}{\sqrt{2\pi}\ \sqrt{1-\rho^2}}$$

In[76]:=
```
Simplify[PDF[NormalDistribution[μ1 + ρ (x2 - μ2), Sqrt[1 - ρ^2]], x1]]
```

Out[76]=
$$\frac{e^{\frac{(x1-\mu1+(-x2+\mu2)\ \rho)^2}{2\left(-1+\rho^2\right)}}}{\sqrt{2\pi}\ \sqrt{1-\rho^2}}$$

The general case:

- $y(x) = \mu + \psi(x)^T \Sigma^{-1}(x - \mu)$
- $s^2(x) = \sigma^2 \left(1 - \psi(x)^T \Sigma^{-1} \psi(x)\right)$

Kriging

Let's relax the assumption that all functions were discrete and return to the continuous case. The value of the unknown x doesn't need to be fixed nor an integer.

In[77]:=
```
generateManipulate2[]
```

Show mean

Show error

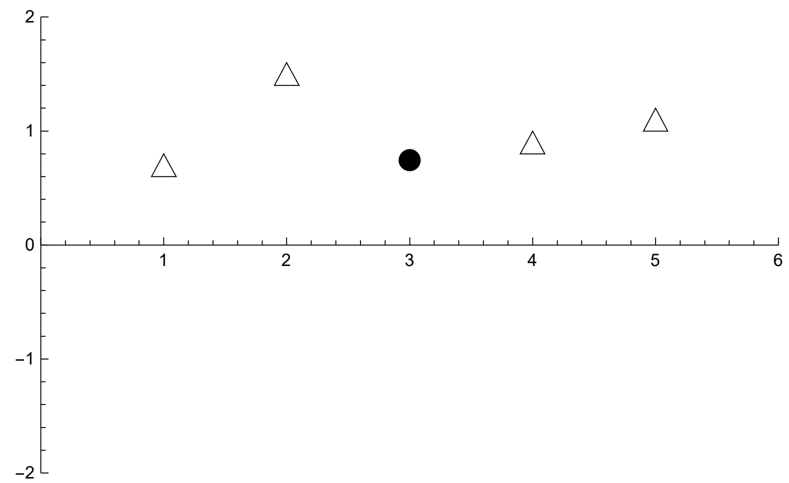
Mean

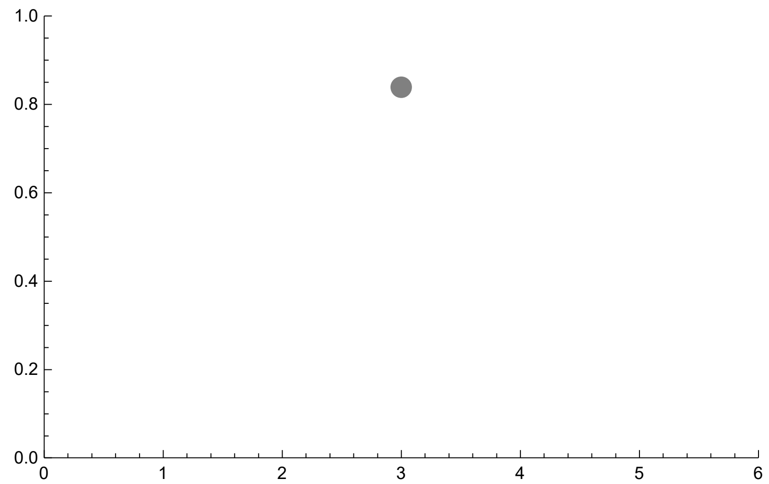
Correlation width

Unknown x

Out[77]=

	Distance	Correlation
x1	2.	0.018
x2	1.	0.37
x3	1.	0.37
x4	2.	0.018

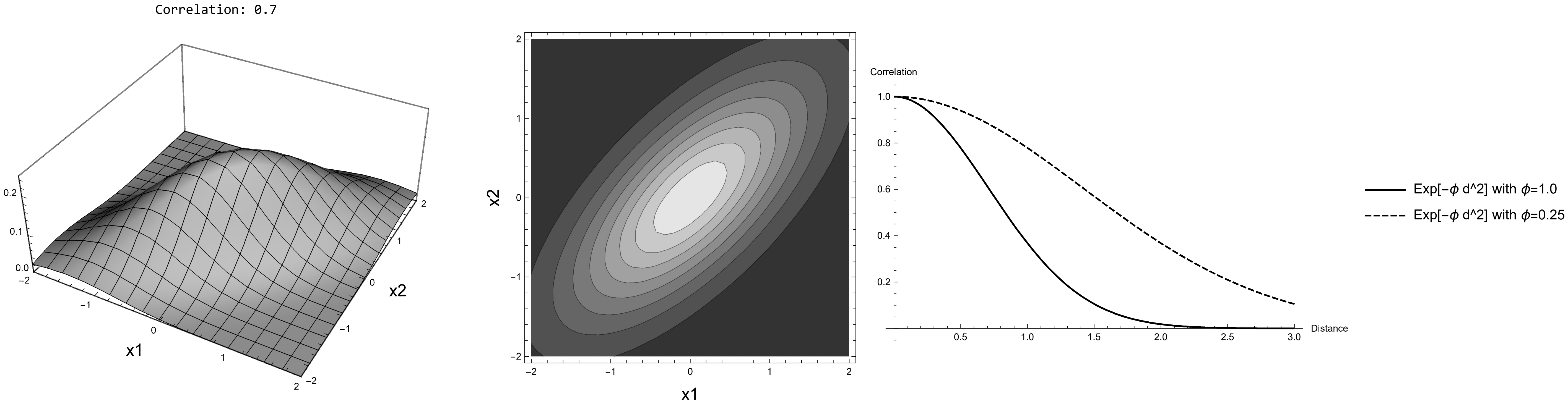




Flexibility of Kriging

In[78]:=
```
Grid[{{Last[animationOfProbabilitySpace], generateCorrelationPlot[]}}, Spacings → {1, 1}]
```

Out[78]=



The ‘unlikelihood’ of a function is divided out in the conditional probability.

The p’s in the correlation function $Exp(-\theta \mid x_i - x_j \mid^p)$ define ‘smoothness’:

- p = 2 gives a smooth correlation with a continuous gradient
- p < 2 increases the rate at which the correlation initially drops

θ is a width parameter that affects how far a sample point’s influence extends. θ is a measure of how ‘active’ the function is (see ‘sensitivity analysis’)

- low θ means that all points will have a high correlation
- high θ means that there is a significant difference between the y values

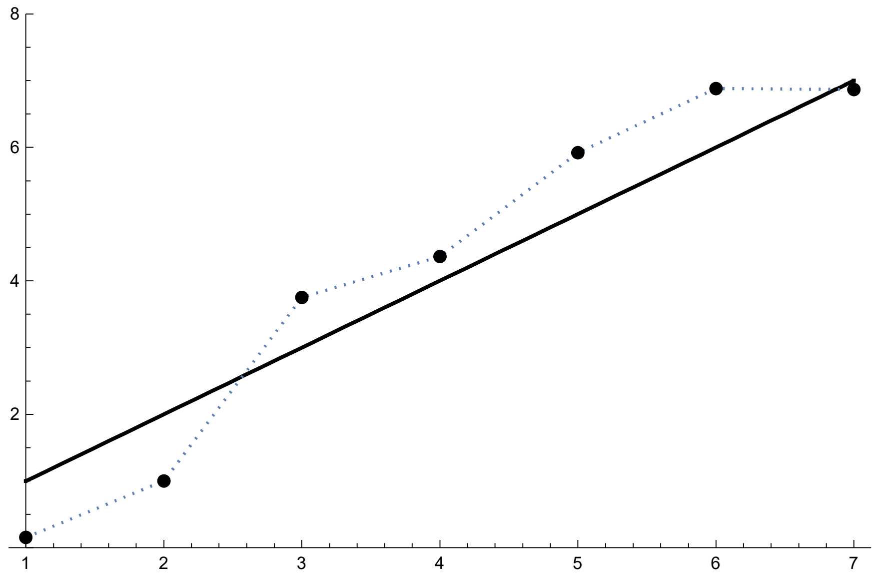
Possibly different θ and p in different dimensions. Choose θ and p to maximize the likelihood of y.

Extensions of Kriging

- Ordinary Kriging
- Universal Kriging
- Regression Kriging
- Gradient-Assisted Kriging
- Co-Kriging
- Non-Stationary Kriging

Universal Kriging

In[79]:=
```
generateUniversalKriging[]
```

Out[79]=


f (x) = $\sum_{j=0}^{k} \beta_j\, v_j(x) + Z(x)$

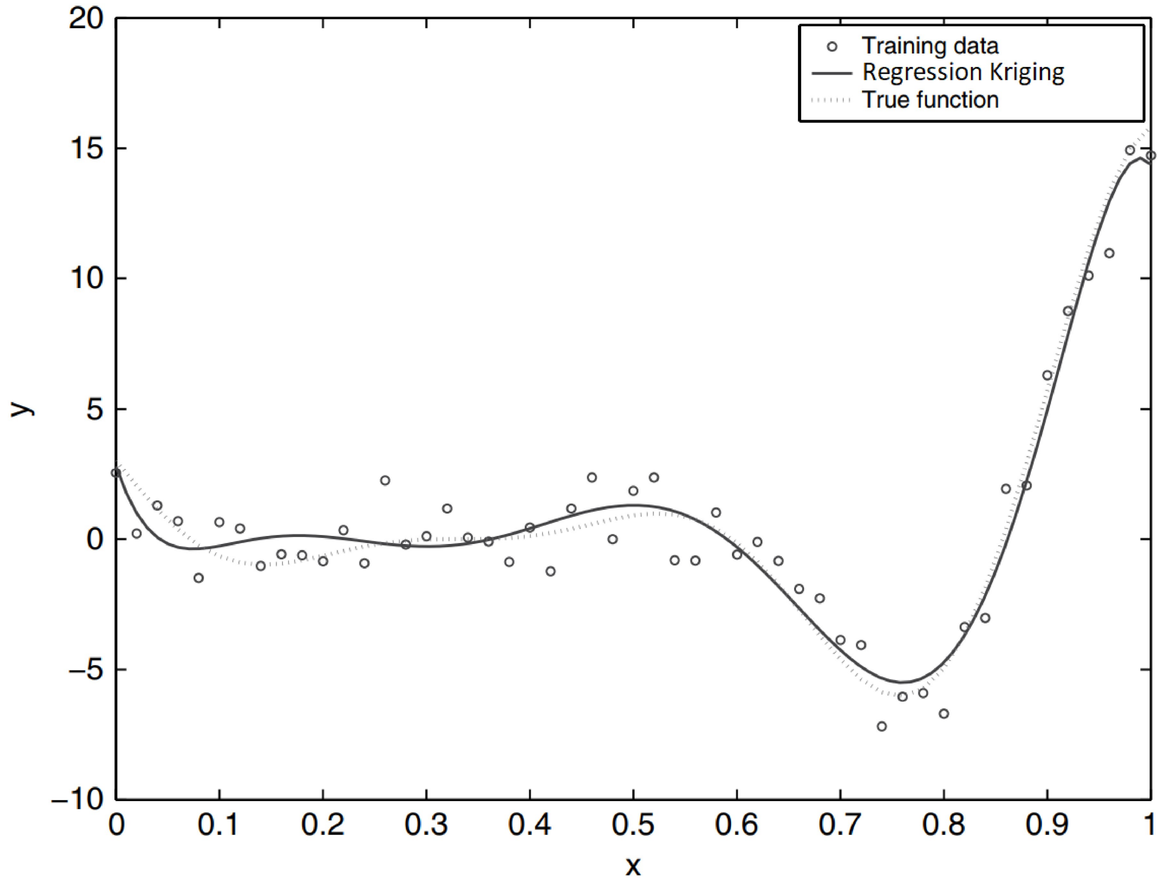
A random function (stochastic process) that includes a regression model. The *v(x)*'s can be any set of fixed functions. The *Z(x)* is like ordinary Kriging.

Regression Kriging

In[80]:=

```
Show[Import[ToString[NotebookDirectory[]] <> "Figures\\RegressionKriging.png"], ImageSize → 600]
```

Out[80]=



If the responses are corrupted by noise, the model may overfit the data.

Add a regression constant (often termed a regularization constant) λ to the correlation matrix ($R \Rightarrow R + \lambda I$)

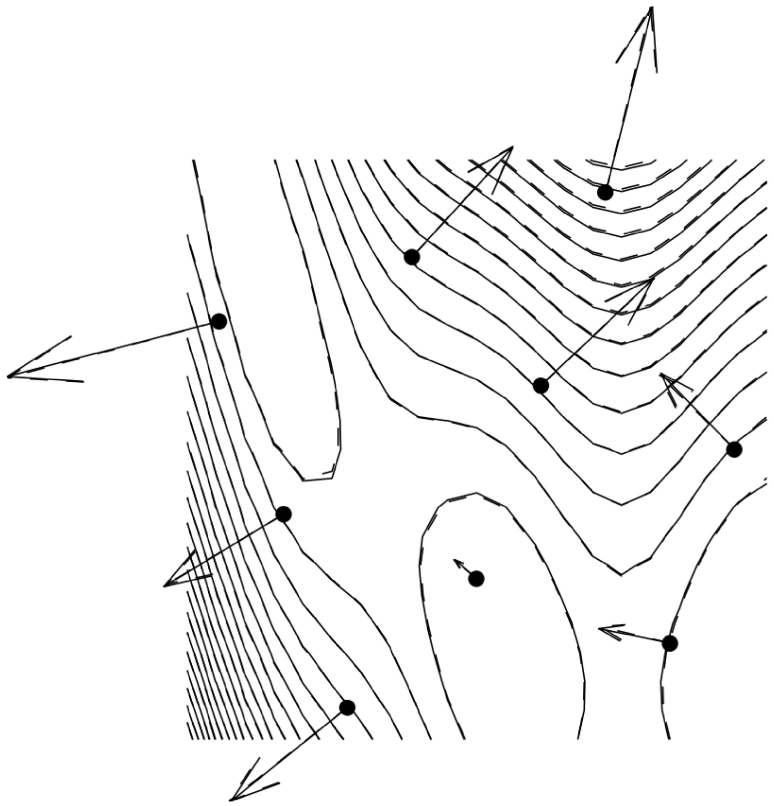
Each data point normally gets a basis function shape as the correlation function. The total Kriging function is a weighted average of these.

To regress, reduce the number of bases. Include only a subset of these functions but still use all the training data to solve for the coefficients in a least squares sense.

Gradient-Assisted Kriging

In[81]:=
```
Show[Import[ToString[NotebookDirectory[]] <> "Figures\\GradientAssistedKriging.png"], ImageSize → 400]
```

Out[81]=



Gradient information can be used to enhance the accuracy of a surrogate model of the design landscape.
It should only be used if it is available *cheaply*. Otherwise better to build the surrogate using a larger design sample.

Co-Kriging

A large quantity of cheap data may be coupled with a small amount of expensive data.

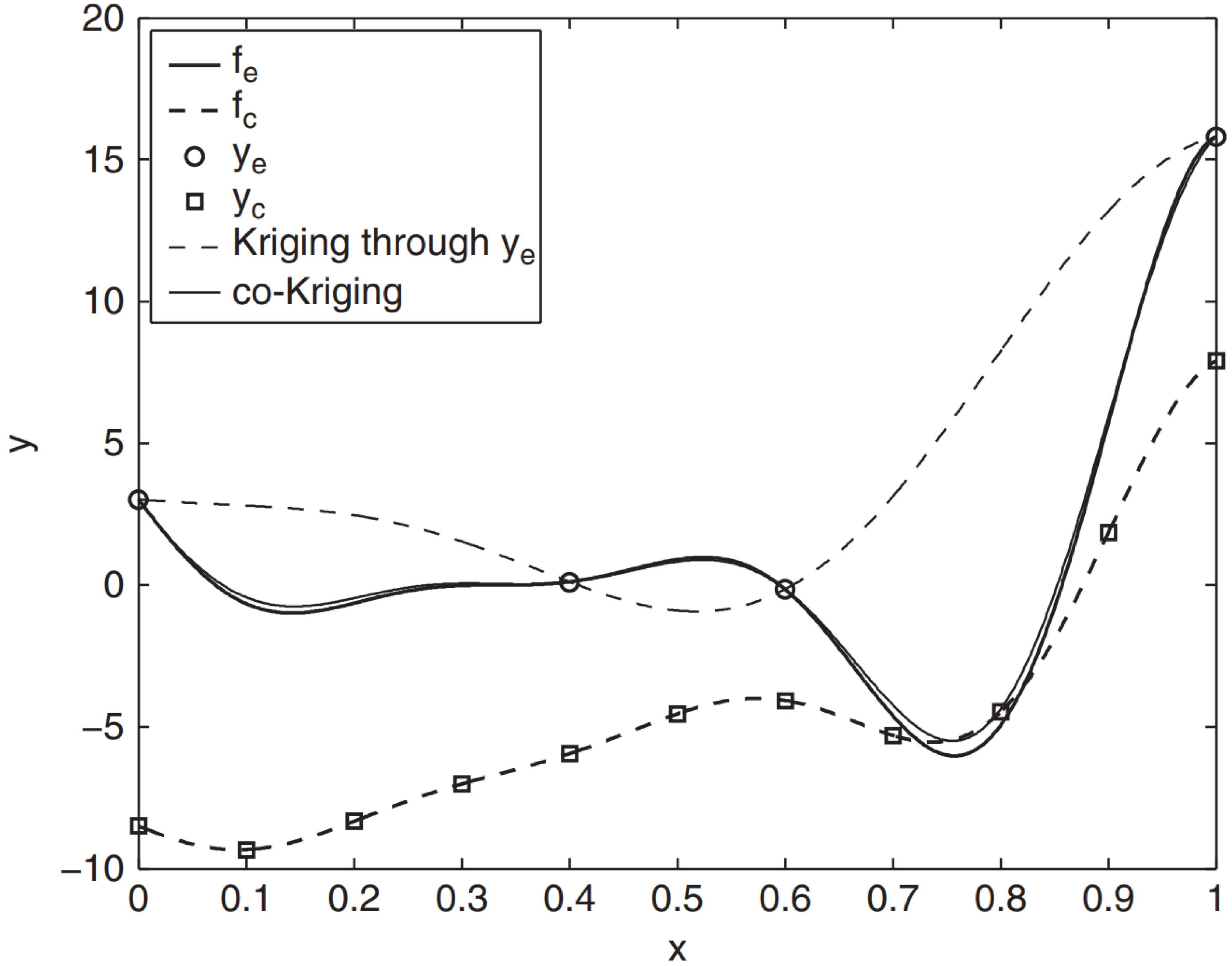
Formulate some form of correction process which models the differences between the cheap and expensive function(s).

$Z_e(x) = \rho Z_c(x) + Z_d(x)$

Used to correct $Z_c(x)$ when making predictions of the expensive function $Z_e(x)$

In[82]:=
```
Show[Import[ToString[NotebookDirectory[]] <> "Figures\\CoKriging.png"], ImageSize → 800]
```

Out[82]=

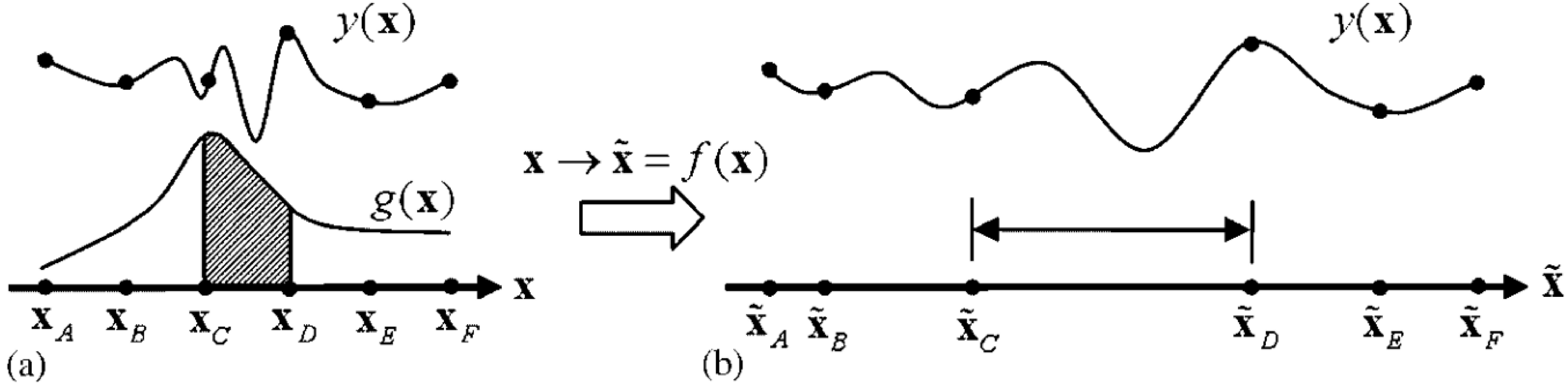


Non-Stationary Kriging

Use a non-linear map (a parameterized density function g (x)) to scale the original space to one in which the covariance becomes approximately stationary.

In[83]:=
```
Show[Import[ToString[NotebookDirectory[]] <> "Figures\\NonStationaryKriging.png"], ImageSize → 1200]
```

Out[83]=



Other Surrogate Methods

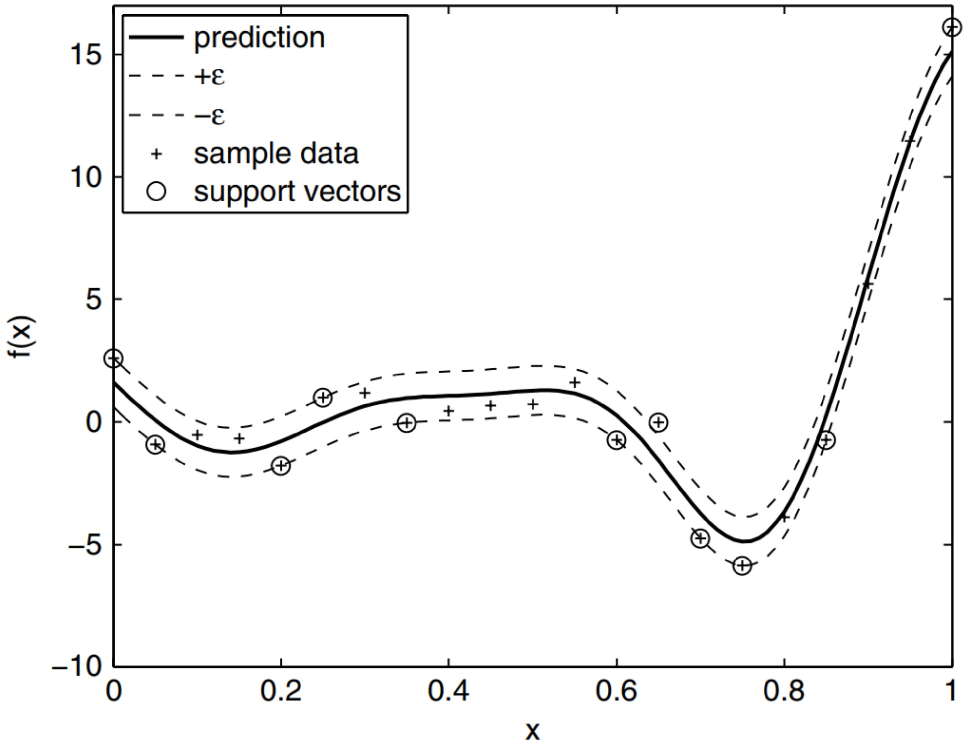
- Polynomials
- Radial Basis Functions
- Space Mapping (partially converged models)
- Polynomial Chaos
- Model Order Reduction (SVD)
- Support Vector Machines
- Neural Networks

Support Vector Machines and Neural Networks

Machine Learning techniques are mathematically quite similar to Radial Basis Functions.

In[84]:=
```
Show[Import[ToString[NotebookDirectory[]] <> "Figures\\SupportVectorMachines.png"], ImageSize → 500]
```

Out[84]=



Related Techniques

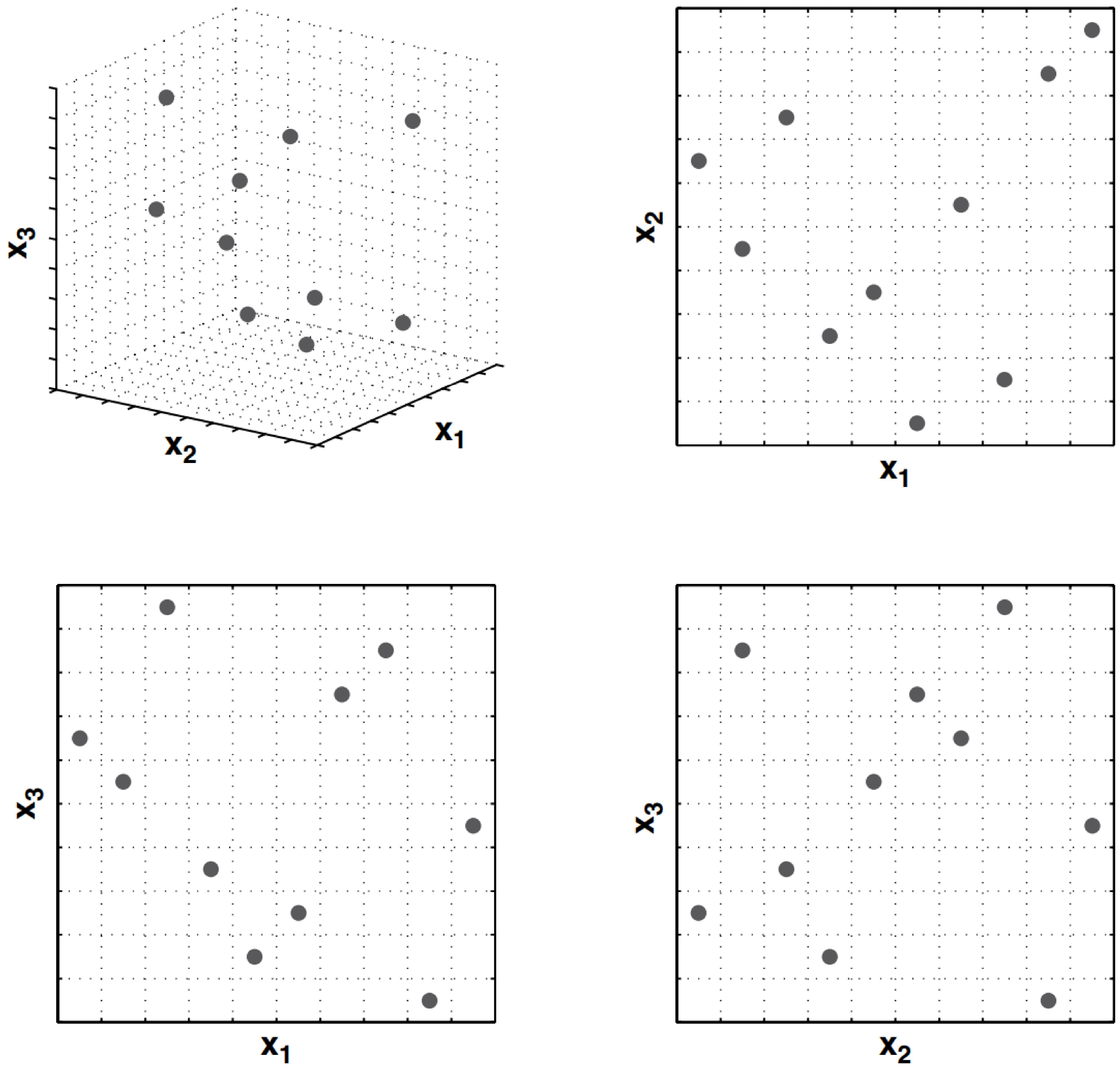
- Sampling
- Theory-Driven Methods
- Constraints
- Multiple Design Objectives

Sampling

Latin Hypercube

In[85]:=
```
Show[Import[ToString[NotebookDirectory[]] <> "Figures\\LatinHyperCube.png"], ImageSize → 800]
```

Out[85]=



Related Techniques

- Sampling
- Theory-Driven Methods (partially converged, knowledge in $p(x)$)
- Constraints
- Multiple Design Objectives (Pareto sets)

Domains and Communities

- Geography
- Engineering ("*Engineering Design via Surrogate Modelling: A Practical Guide*" by A. I. J. Forrester, A. Sobester and A. J. Keane, 2008)
- Physics
- Chemistry and Biology

Theoretical Problems

- Parameter Uncertainty
- Parameter Re-Estimation
- Validation, Regularisation and Variable Selection (blind Kriging)

Implementation Problems

- Maximum Likelihood Estimation
- Singular Correlation Matrix (Kriging guarantees positive semi-definite so Cholesky factorization, otherwise LU decomposition)

Implementations

- Python Implementations (pyKrige, pyKriging)
- R Implementations (DiceKriging)
- MatLab Implementations (DACE, ooDACE Toolbox)