Surrogate models and Gaussian Process regression – lecture 1/5

Surrogate models in engineering

Mines St-Étienne – Majeure Data Science – 2016/2017

Nicolas Durrande (durrande@emse.fr)

Introduction

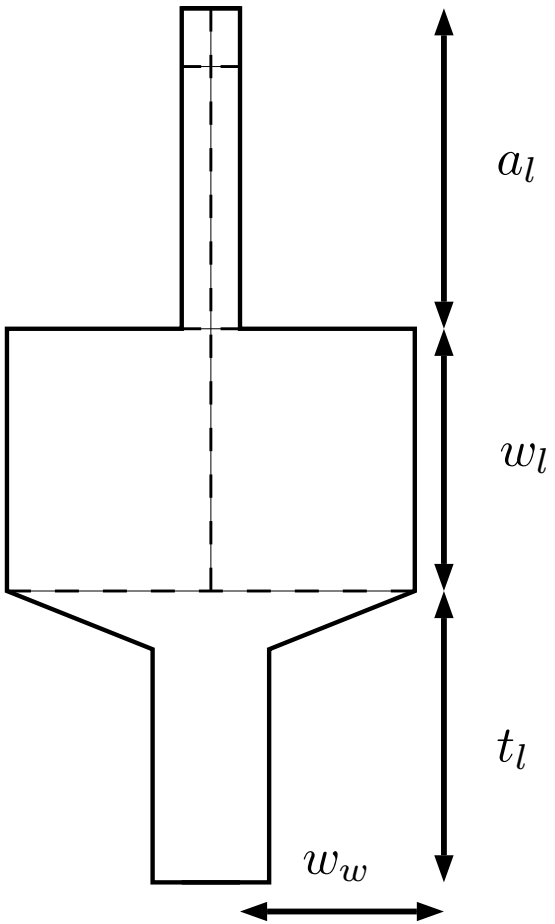
Context:

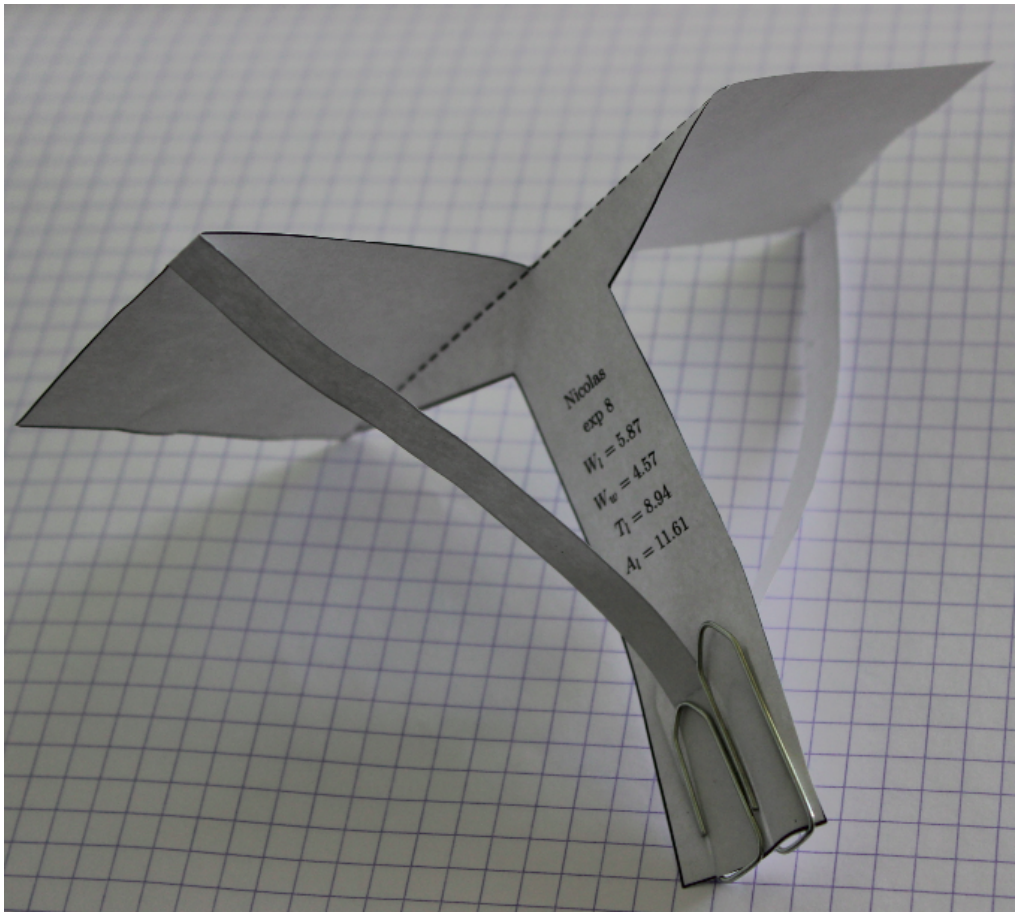
The aim of this module is to introduce a framework dedicated to the study of functions that are **costly to evaluate**.

Detail of the courses:

- Surrogate models and Gaussian process regression (N. Durrande, A. Lopez Lopera)
- Application to sensitivity analysis (E. Padonou)
- Optimization (Y. Richet, N. Garland)
- Helicopter project (N. Durrande, A. Lopez Lopera).

The project will be on the optimization of paper helicopters:





What value of $(a_l,\ w_l,\ t_l,\ w_w)$ gives the longest falling time?

	SEMAINES	49	50	51
		Déc	Déc	Déc
LUNDI	8H15-10H00	5	12 ND + AFLL: Uncertainty prop. (S2.23)	19 ND + AFLL : proj helico (S2.23)
	10H15-12H00	ND: surrogate models (S2.14)		
	13H30-15H		EP: Sensitivity analysis (S2.14)	ND + AFLL : proj helico (S2.23)
	15H15-16H45		EP: Sensitivity analysis (S2.23)	
	17H-18H30			
MARDI	8H15-10H00	6	13 ND: advanced Gps (S2.14)	20 ND + AFLL : proj helico
	10H15-12H00			
	13H30-15H		YR : EGO (?)	
	15H15-16H45		YR + NG : TP EGO (?)	
	17H-18H30			
MERCREDI	8H15-10H00	7 ND: Gauss Proc (F1)	14 YR: EGO (F1)	21 exam UP4 (2h - A1.04)
	10H15-12H00	ND + AFLL: Gauss Proc (E4.06)	YR + NG: EGO (E4.06)	
	13H30-15H		ND: DoE (S2.14)	
	15H15-16H45		ND + AFLL: (E4.06)	
	17H-18H30			
JEUDI	8H15-10H00	8 ND: kernels and likelihood (S2.14)	15 ND: DoE (S2.14)	22
	10H15-12H00			
	13H30-15H			

Course material:

- Handout
- Campus

- github: https://github.com/NicolasDurrande/EMSE$_{G}aussian_{p}rocess_{m}odels$

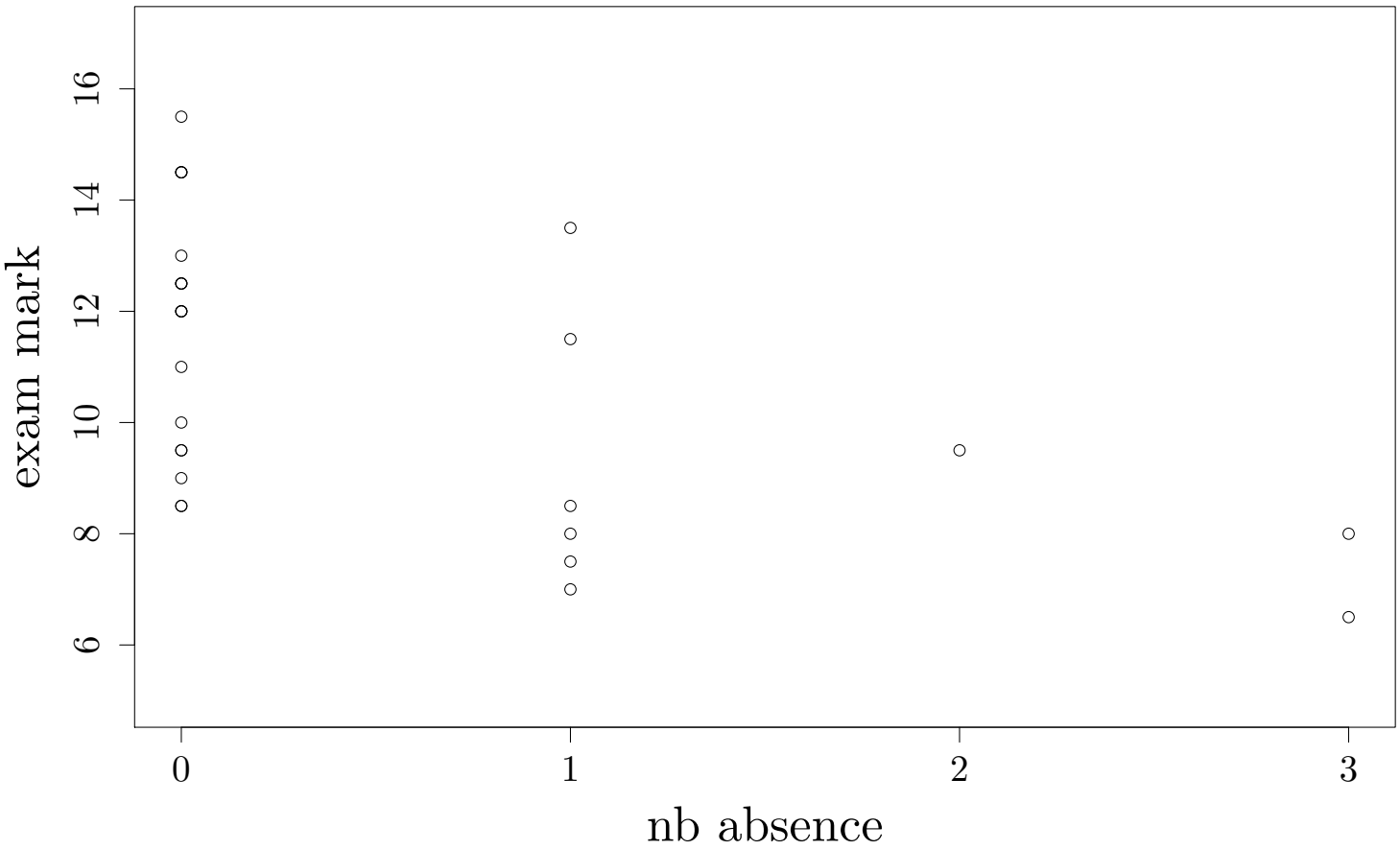
Marking:

- A 2h exam at the end of the course (60% of the mark).
- A report on the helicopter project (40% of the mark).

Presence:

- attending classes and lab session is mandatory.
- after one unjustified absence : -0.5pts on the mark per unjustified absence

Why is presence important:



Outline of today's lecture

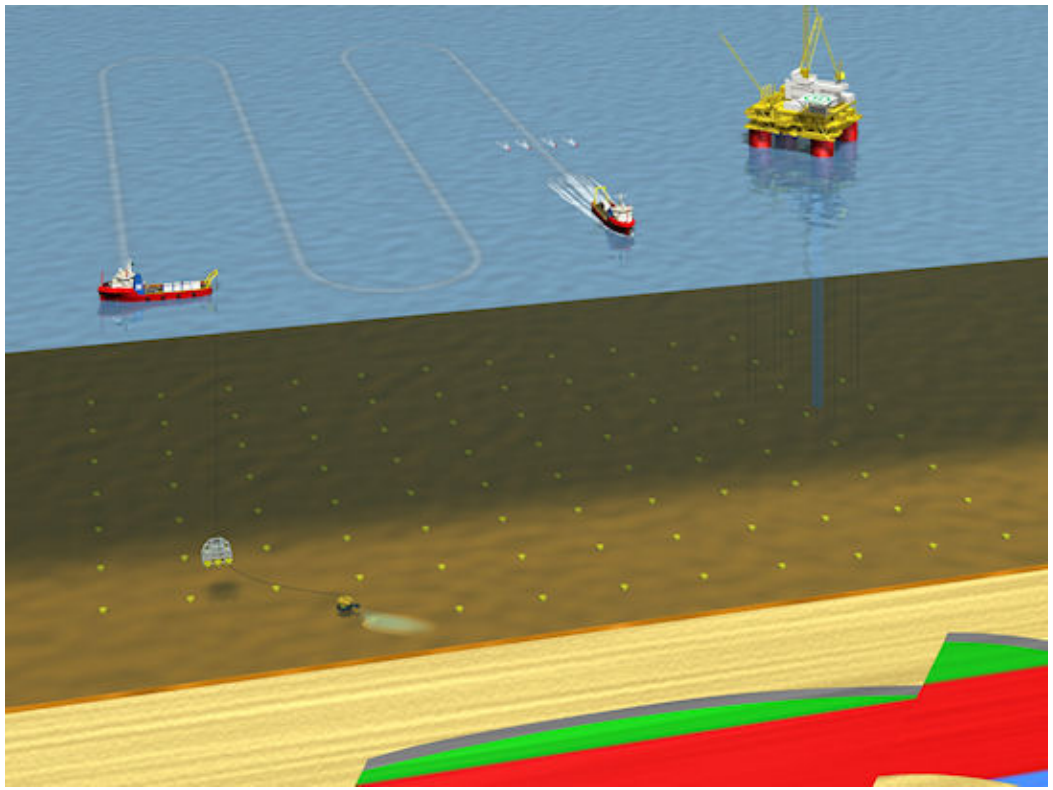
- Why (and when) statistical models can be useful in engineering?
- Some typical methods:
  - Linear regression
  - Polynomial Chaos
  - Neural networks
  - Gaussian process regression (Kriging)

Why are surrogate models relevant in engineering?

There is a wide variety of situations where getting data is extremely expensive.

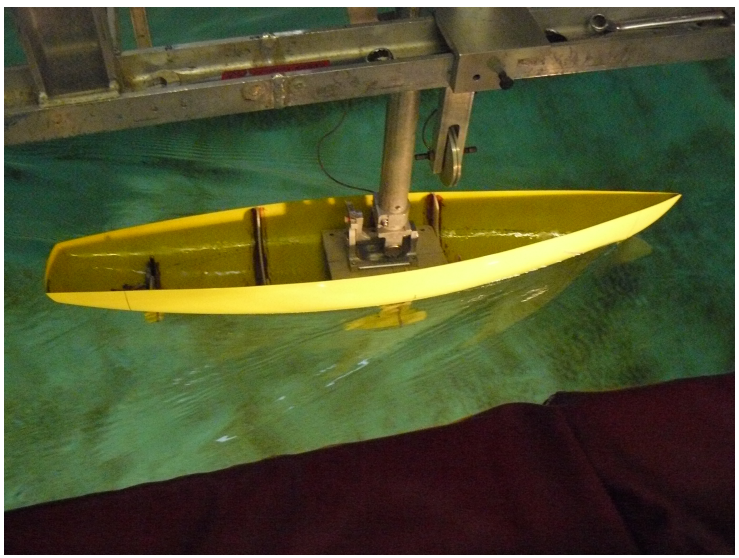
- real world experiments
- destructive tests
- prototyping
- numerical experiments

Example: real world experiments



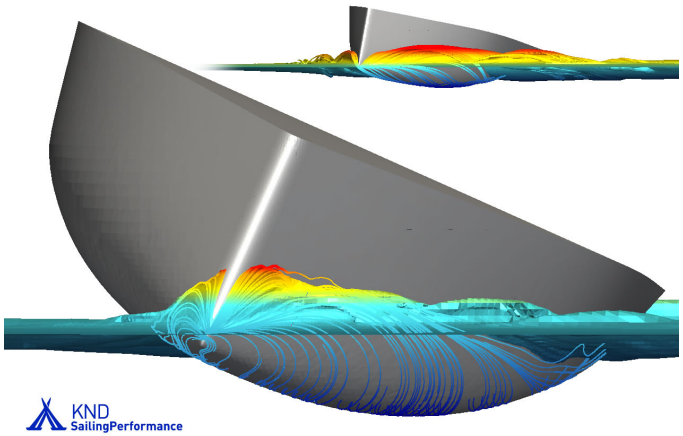
Example: Destructive tests

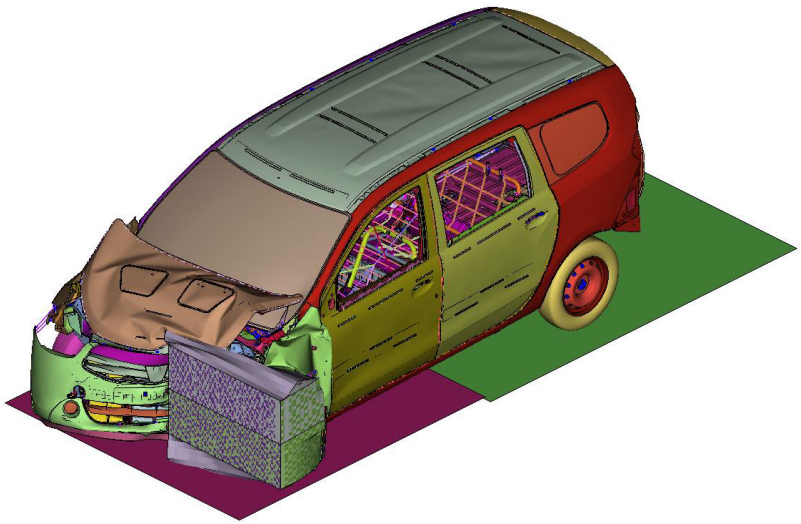
Example: Prototyping of a boat shape



Knowing the drag for a given design requires costly experiments

In practice: Numerical experiments are extremely common

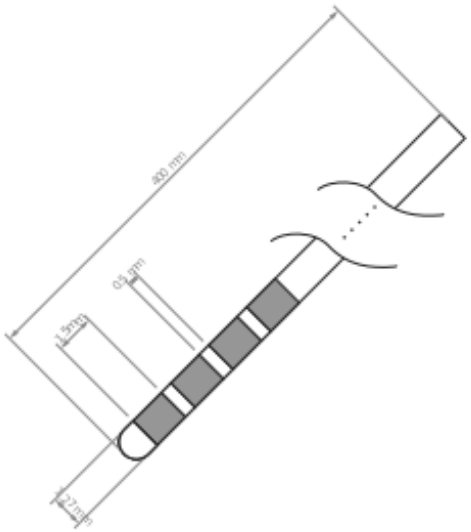




They are less expensive but can be very time consuming!

Example in medecine

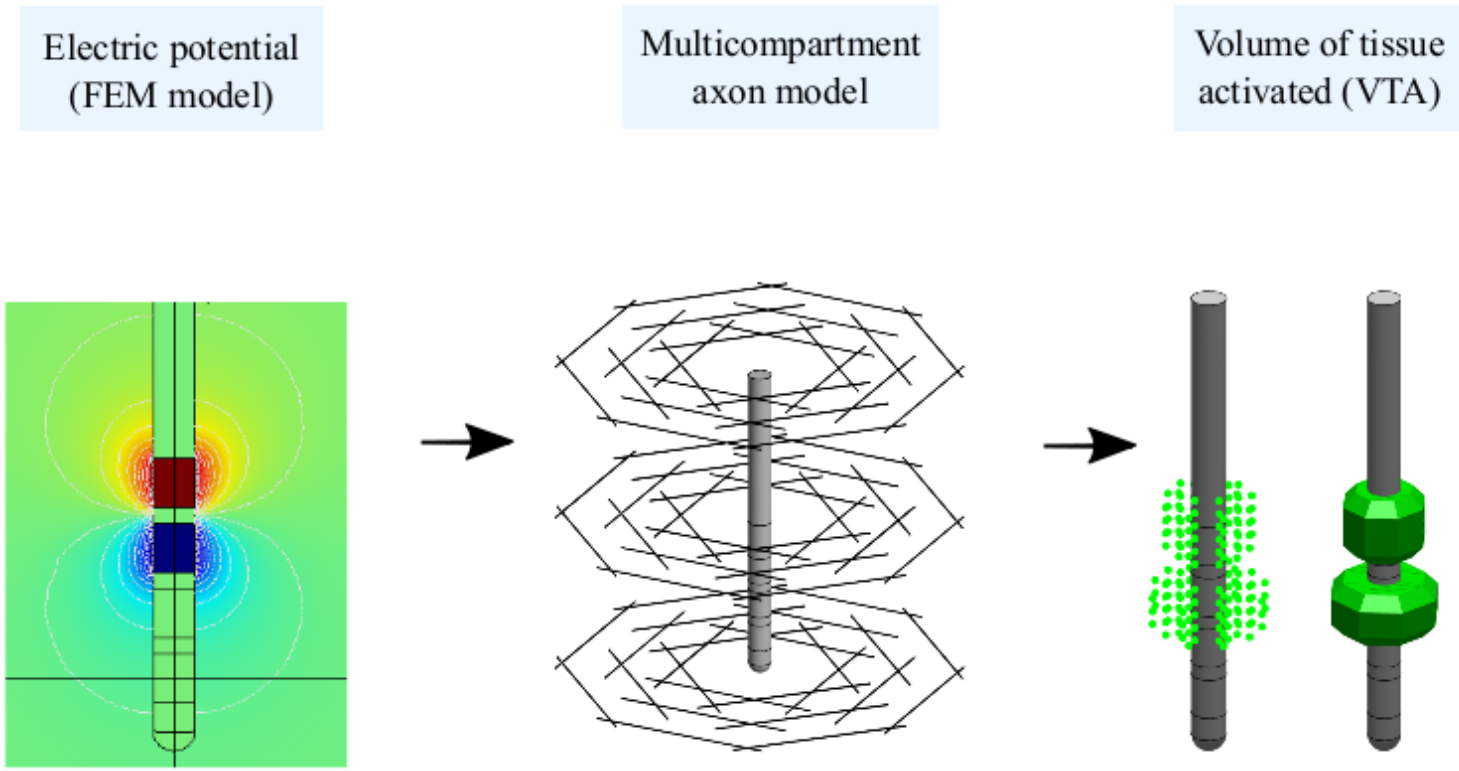
Deep brain simulation is an effective method to treat patients with Parkinson Desease



source: Seminar from M. Alvarez (Univ. Tech de Pereira, Colombia) at Mines St-Étienne, 2016.

Example in medecine

Computing the volume of tissus activated required two numerical simulator



source: Seminar from M. Alvarez (Univ. Tech de Pereira, Colombia) at Mines St-Étienne, 2016.

In all these cases, the variable of interest can be seen as a function of the input parameters

$$y = f(x).$$

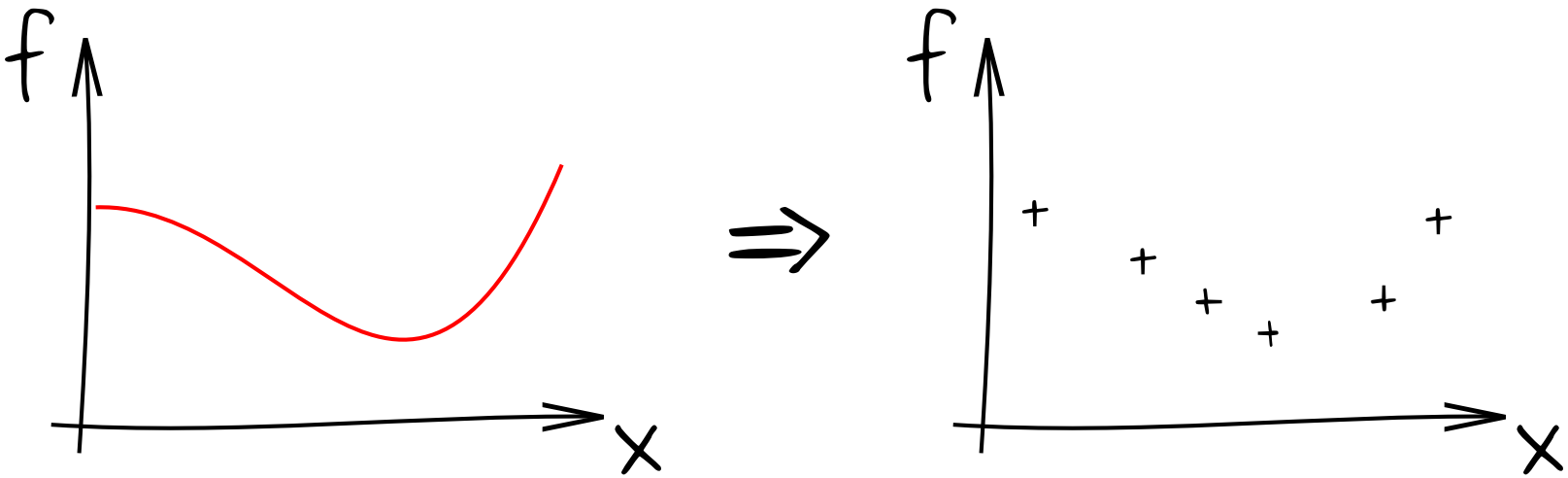
where f is a **costly to evaluate function**.

In the following, we will assume that

- $x \in \mathbb{R}^d$: There are many input parameters
- $y \in \mathbb{R}$: The output is a scalar.

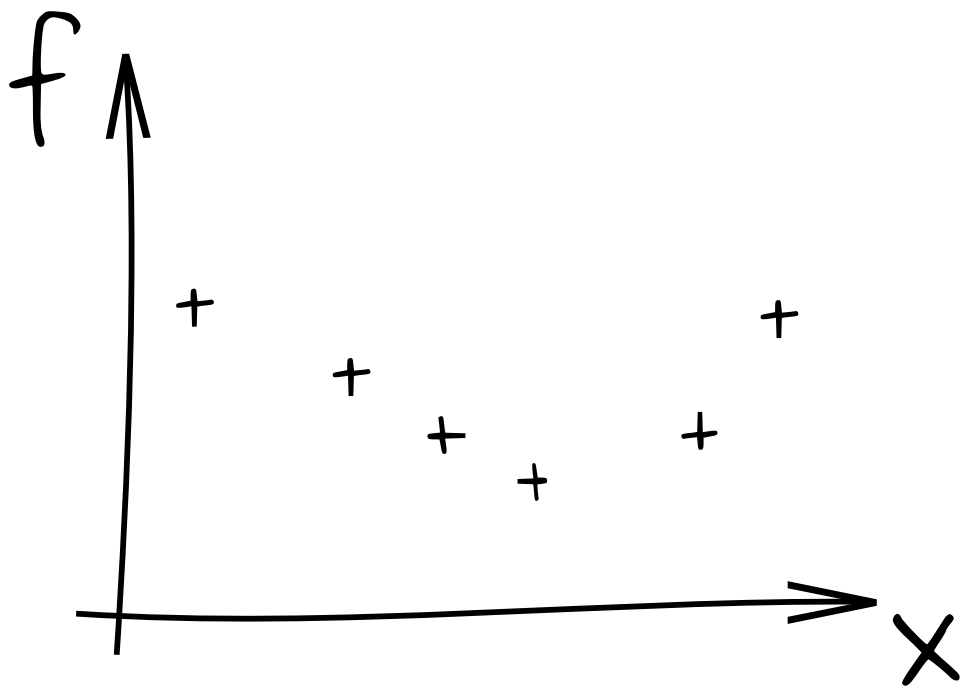
The fact that f is **costly to evaluate** changes a lot of things...

1. Representing the function is not possible...



The fact that f is **costly to evaluate** changes a lot of things...

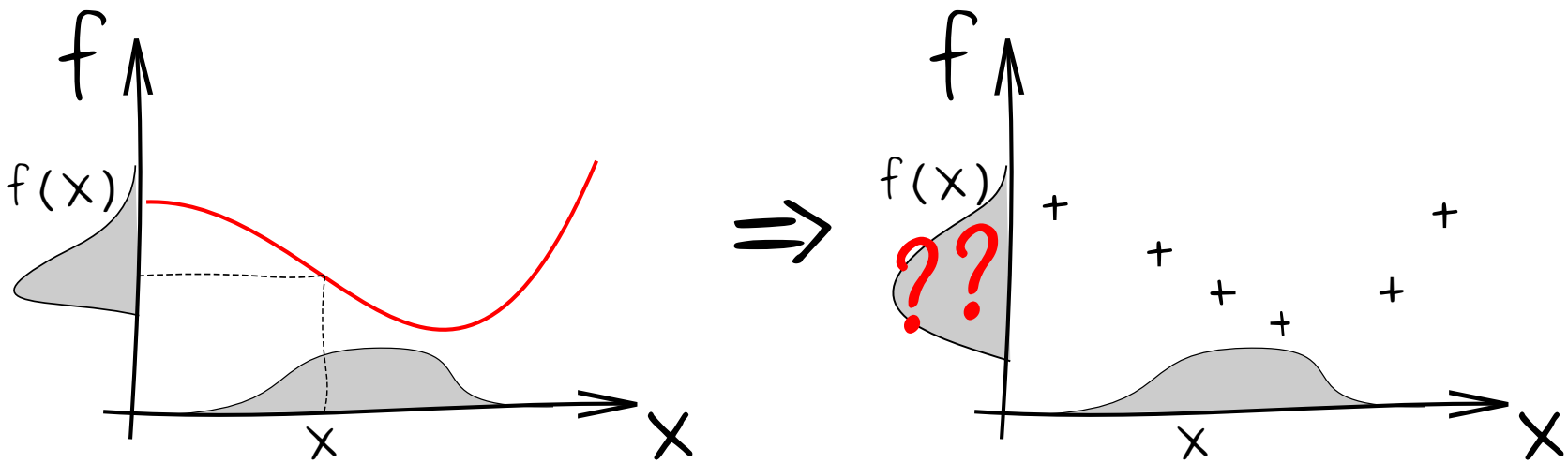
2. Computing integrals is not possible...



What is the mean value of f?

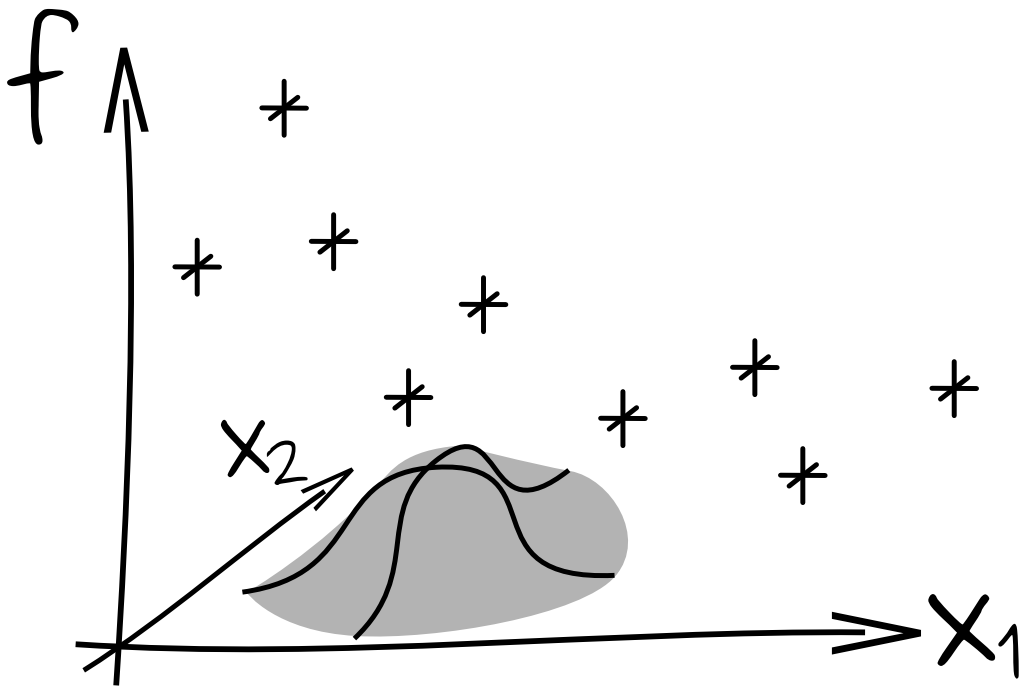
The fact that f is **costly to evaluate** changes a lot of things...

3. Uncertainty propagation is not possible...



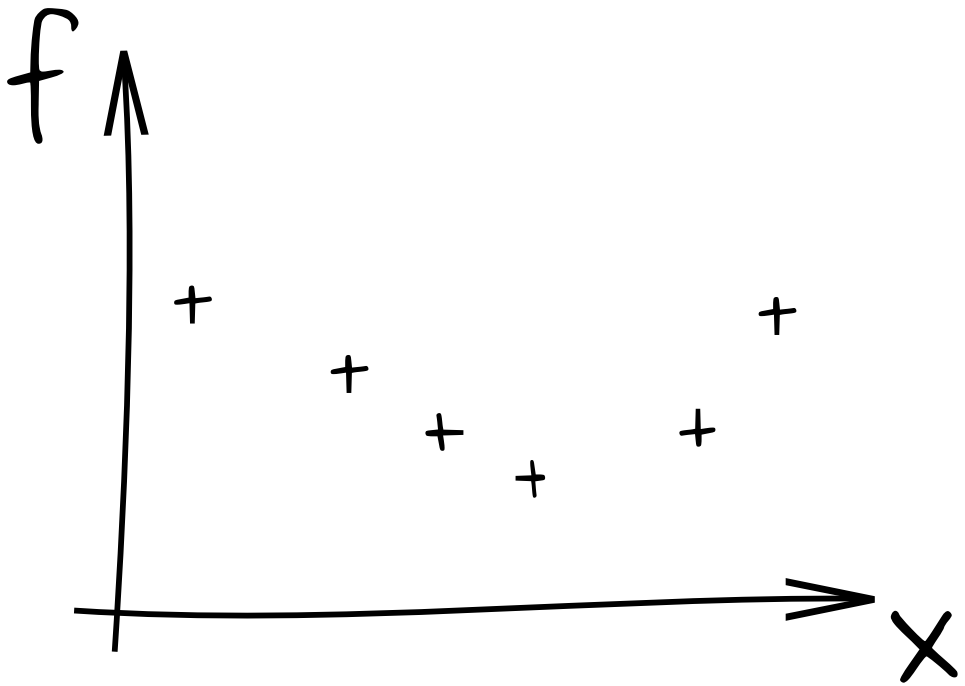
The fact that f is **costly to evaluate** changes a lot of things...

4. Sensitivity analysis is not possible...



The fact that f is **costly to evaluate** changes a lot of things...

5. Optimisation is also tricky...



Another example of scattered data is the coupling of solvers with different physics

Example

Fluid-structure interaction in aeroelasticity:

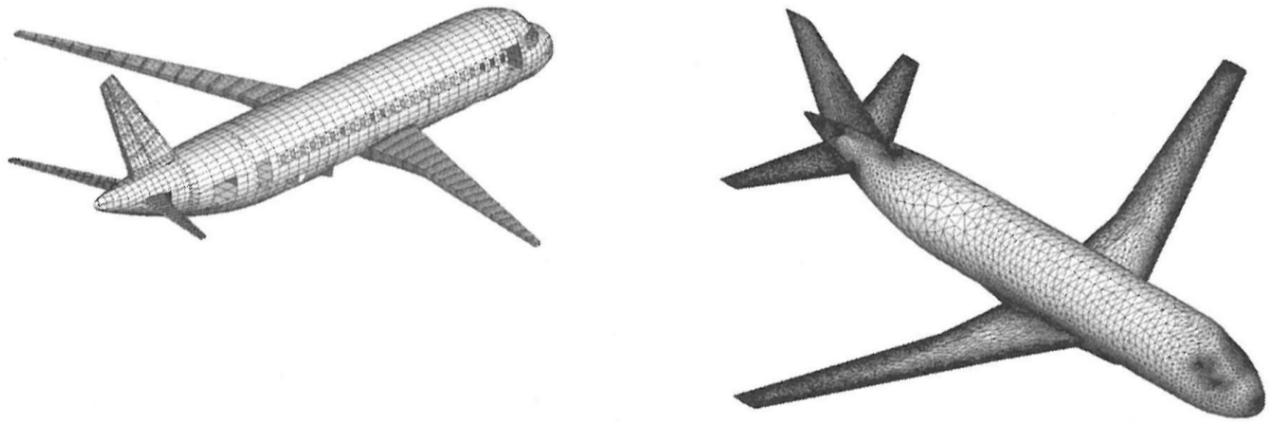
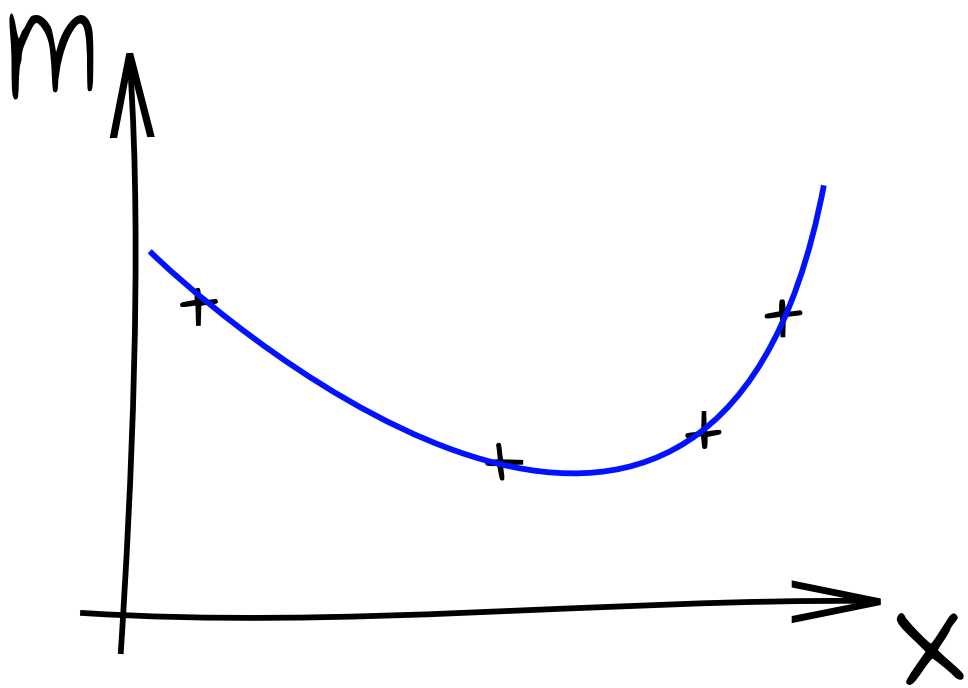


Fig. 1.3 The structural and aerodynamical model of a modern aircraft.

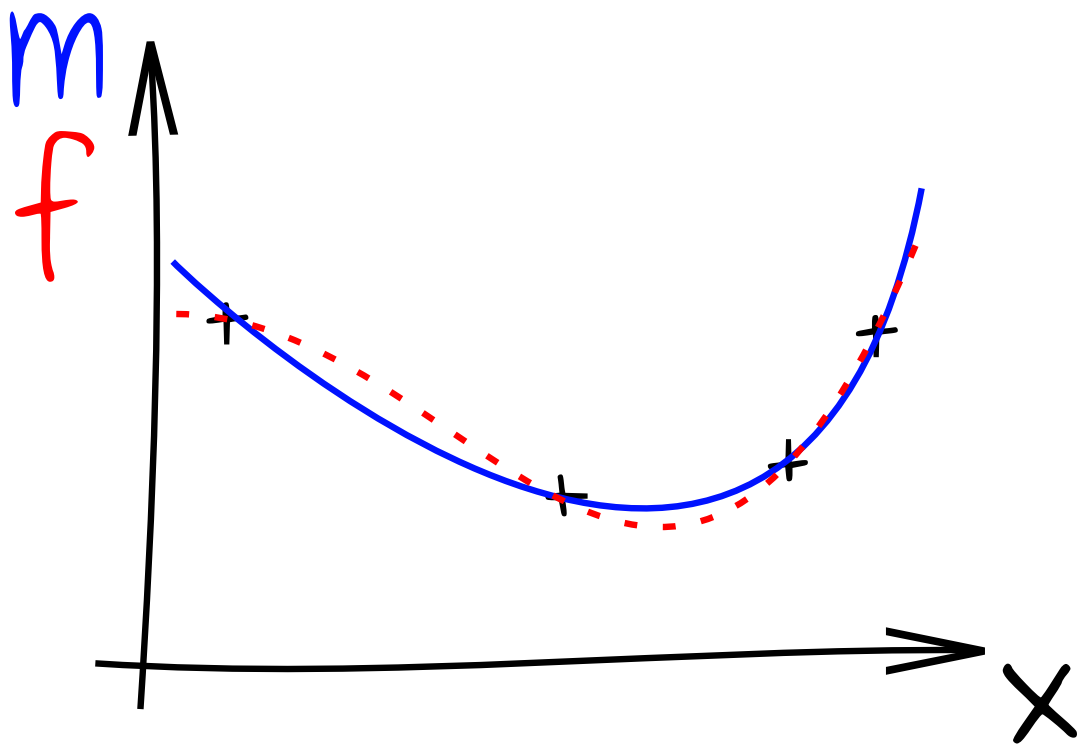
source: H. Wendland, *Scattered data approximation*. Vol. 17. Cambridge university press, 2004.

The principle of statistical modelling is to use the data to build a mathematical approximation of the function.



The model can then be used to answer all previous questions

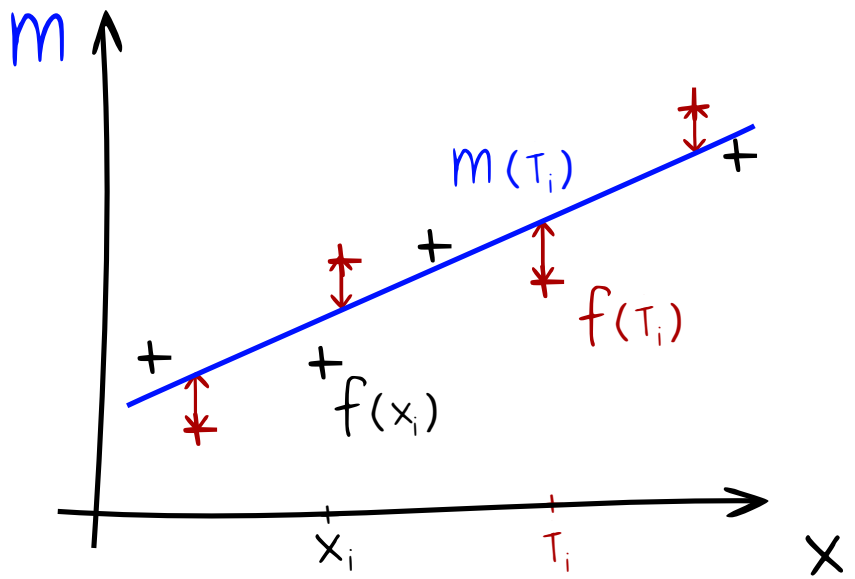
Of course, there is a difference between f and m...



Model validation is always of upper importance.

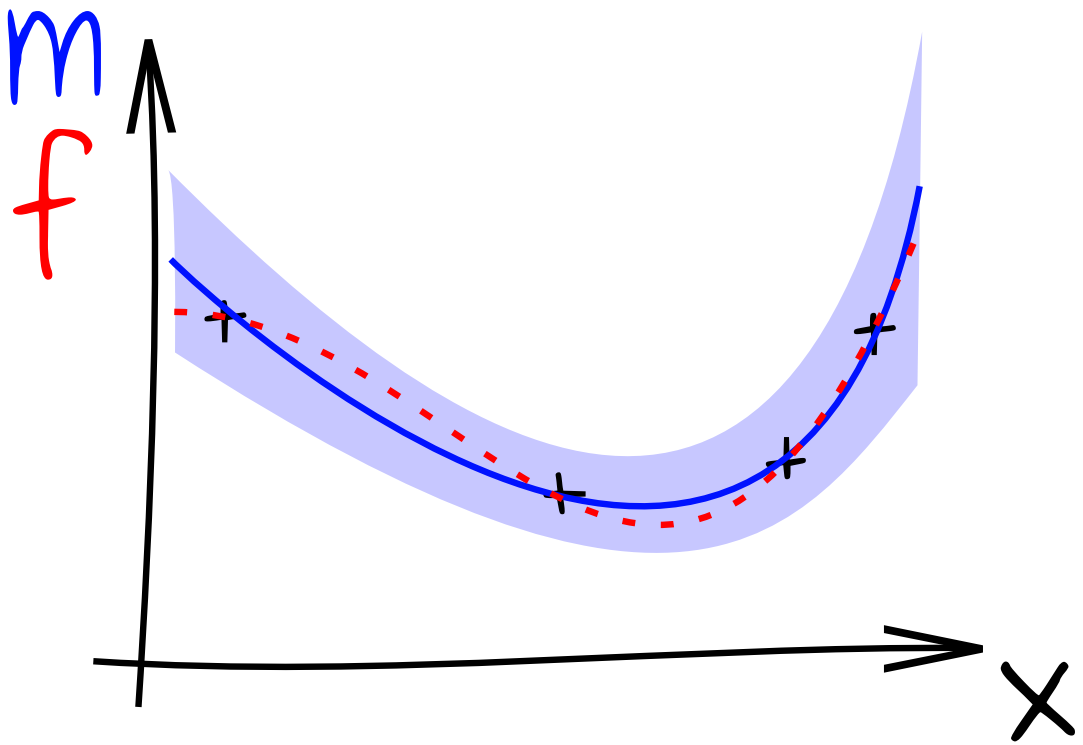
The goodness of fit can be measured by the **mean square error**:

$$MSE = \frac{1}{n}\sum_i \left(f(t_i) - m(t_i)\right)^2$$



where $T = (t_1, \ldots, t_n)$ is a set of test points. If no test set is available, one can use **cross validation** methods.

What about **statistical models**?
We want to be able to quantify the model error:



The confidence intervals can be used to obtain a **measure of uncertainty on the value of interest**.

In the sequel, we will use the following notations :

- The set of observation points will be represented by a $n \times d$ matrix $X = (X_1, ..., X_n)^t$
- The vector of observations will be denoted by F : $F_i = f(X_i)$ (or $F = f(X)$).

We will now introduce various types of surrogate models:

- linear regression
- polynomial chaos
- neural networks
- Gaussian process regression

Linear Regression

Linear regression is probably the most commonly used statistical model.

Given a set of basis functions $B = (b_0, \ldots, b_p)$, we assume that the observations come from the probabilistic model

$$F = B(X)\beta + \varepsilon \qquad \left(\text{i.e. } F_i = \sum_{k=1}^{p} \beta_k b_k(X_i) + \varepsilon_i \right)$$

where the vector β is unknown and the ε_i are independent and identically distributed.

If we consider a model of the form

$$m(x) = B(x)\hat{\beta}$$

the prediction error (Residual Sum of Square) is given by

$$RSS = (B(X)\hat{\beta} - F)^t(\hat{\beta}B(X) - F) \qquad \left(\text{i.e. } \sum_{k=1}^{n}(B(X_i)\hat{\beta} - F_i)^2\right)$$

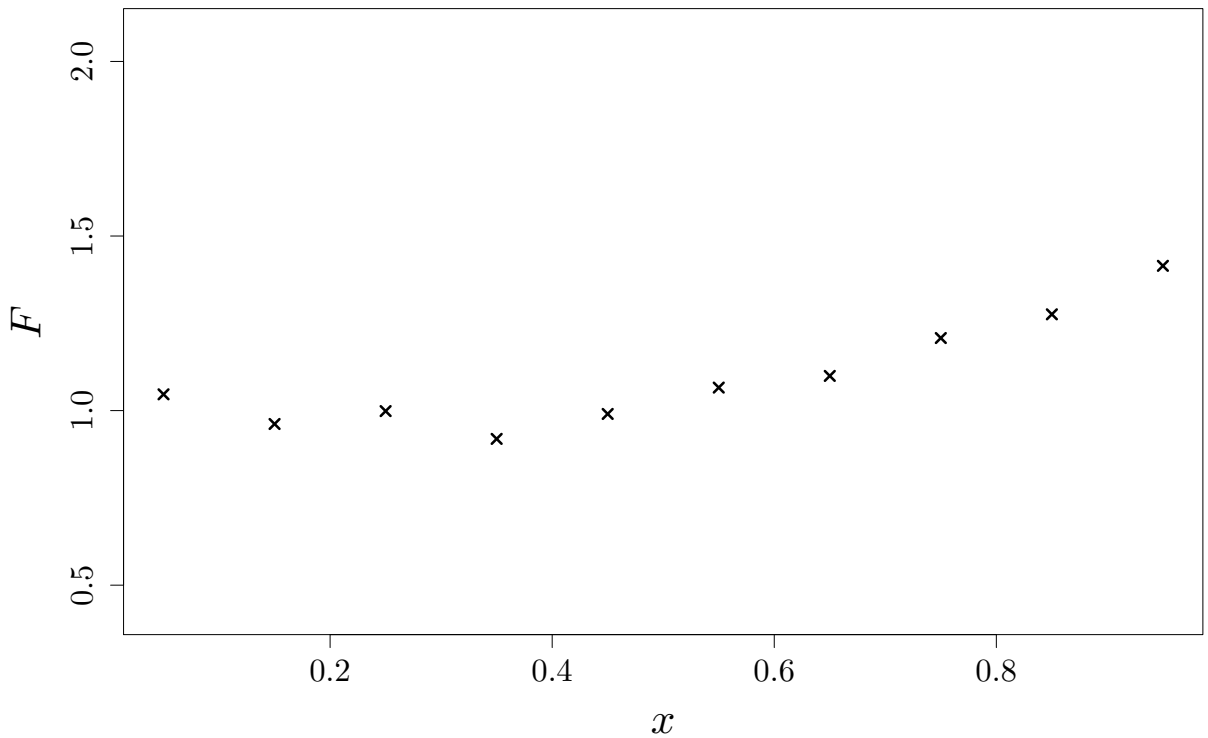
Finding the optimal value of $\hat{\beta}$ means minimizing a quadratic form. This can be done analytically and we obtain $\hat{\beta} = (B(X)^t B(X))^{-1} B(X)^t F$.

The associated linear regression model is thus

$$m(x) = B(x)(B(X)^t B(X))^{-1} B(X)^t F.$$

Example

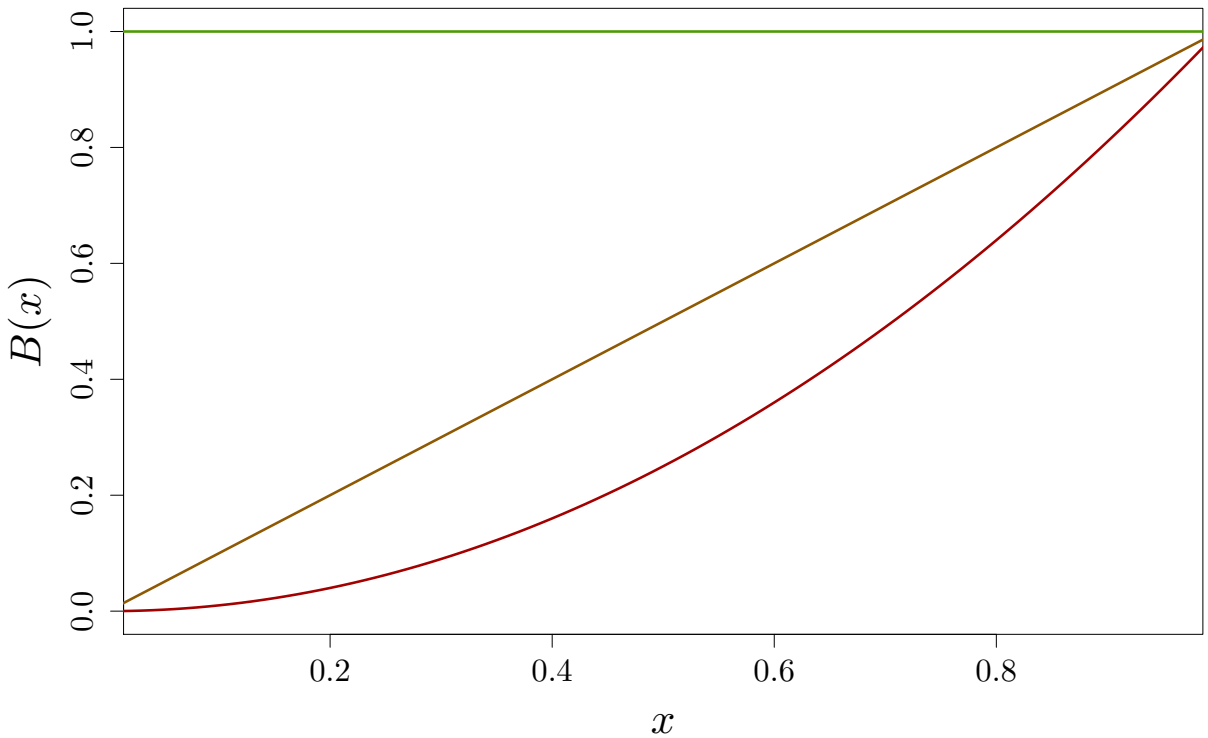
If we consider the following observations:



Example

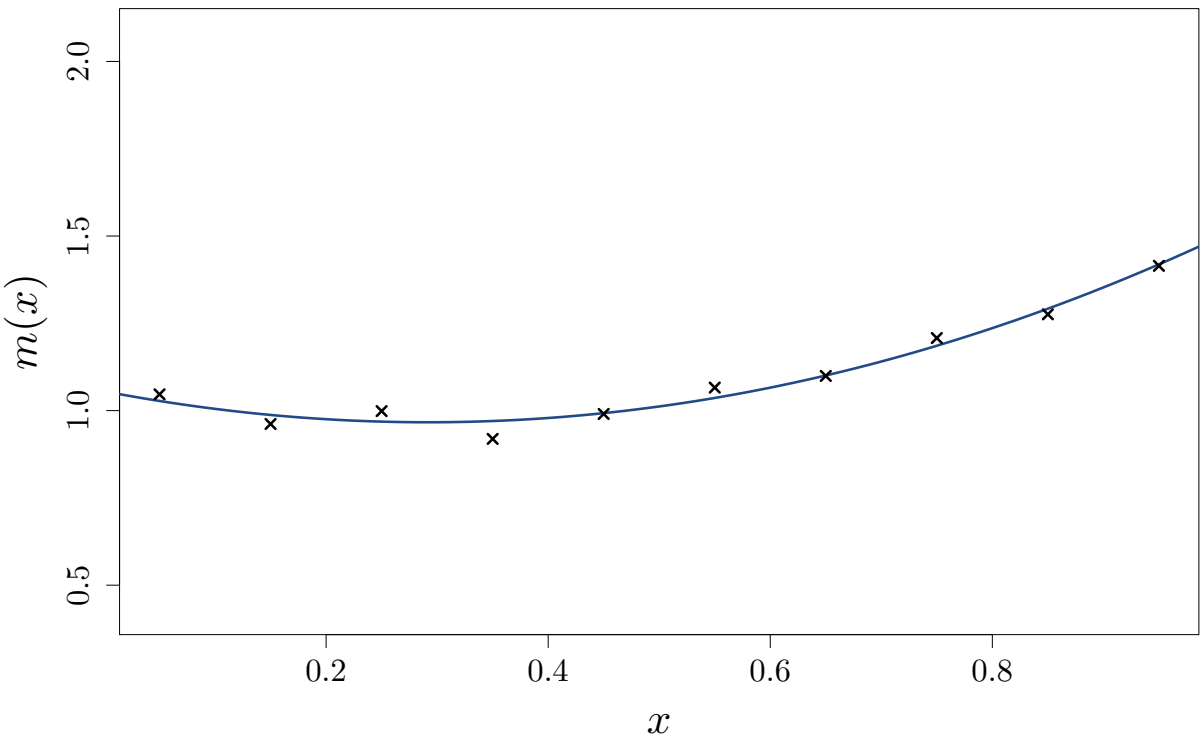
and a set of 3 basis functions:

$$b_0(x) = 1, \; b_1(x) = x, \; b_2(x) = x^2$$



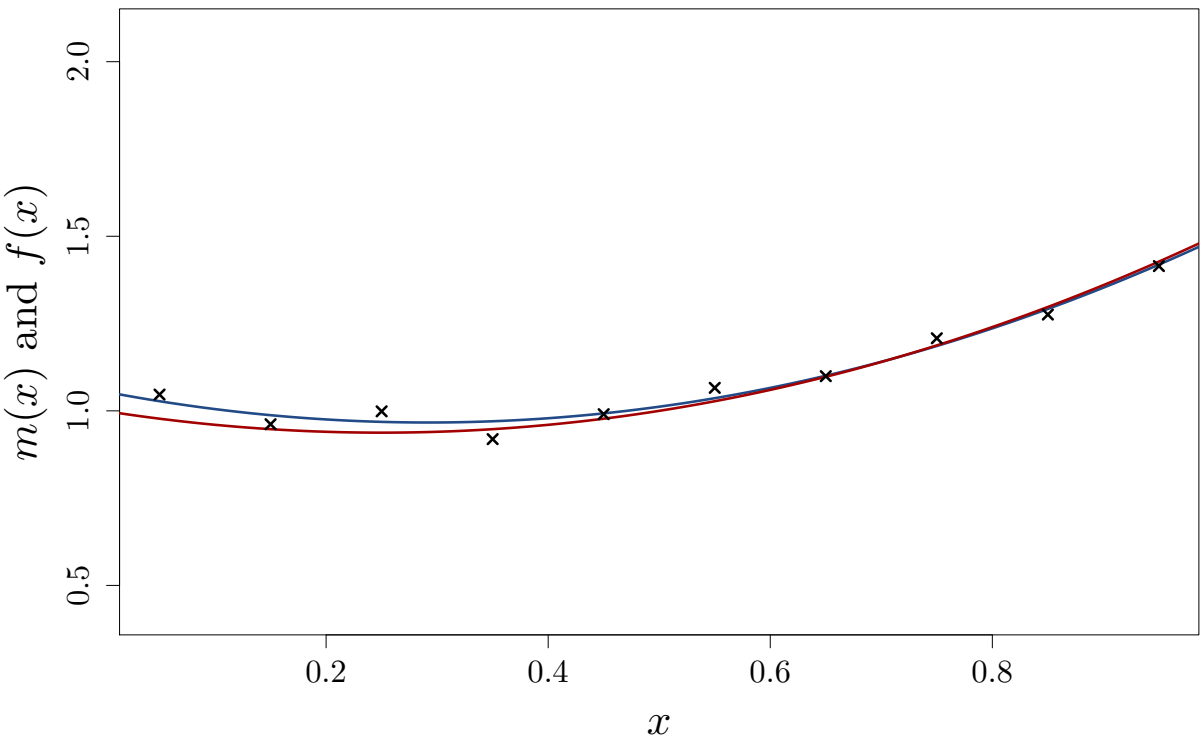
Example

We obtain $\hat{\beta} = (1.06, -0.61, 1.04)$ and the model is:



Example

There is of course an error between the true generative function and the model



Can this error be quantified?

The initial assumption is $F = B(X)\beta + \varepsilon$ and we have computed an estimator of β:

$$\hat{\beta} = (B(X)^t B(X))^{-1} B(X)^t F.$$

$\hat{\beta}$ can thus be seen as a sample from the random variable:

$$\hat{\beta} = (B(X)^t B(X))^{-1} B(X)^t (B(X)\beta + \varepsilon).$$

What about the distribution of $\hat{\beta}$?

The initial assumption is $F = B(X)\beta + \varepsilon$ and we have computed an estimator of β:

$$\hat{\beta} = (B(X)^t B(X))^{-1} B(X)^t F.$$

$\hat{\beta}$ can thus be seen as a sample from the random variable:

$$\hat{\beta} = (B(X)^t B(X))^{-1} B(X)^t (B(X)\beta + \varepsilon).$$

What about the distribution of $\hat{\beta}$?

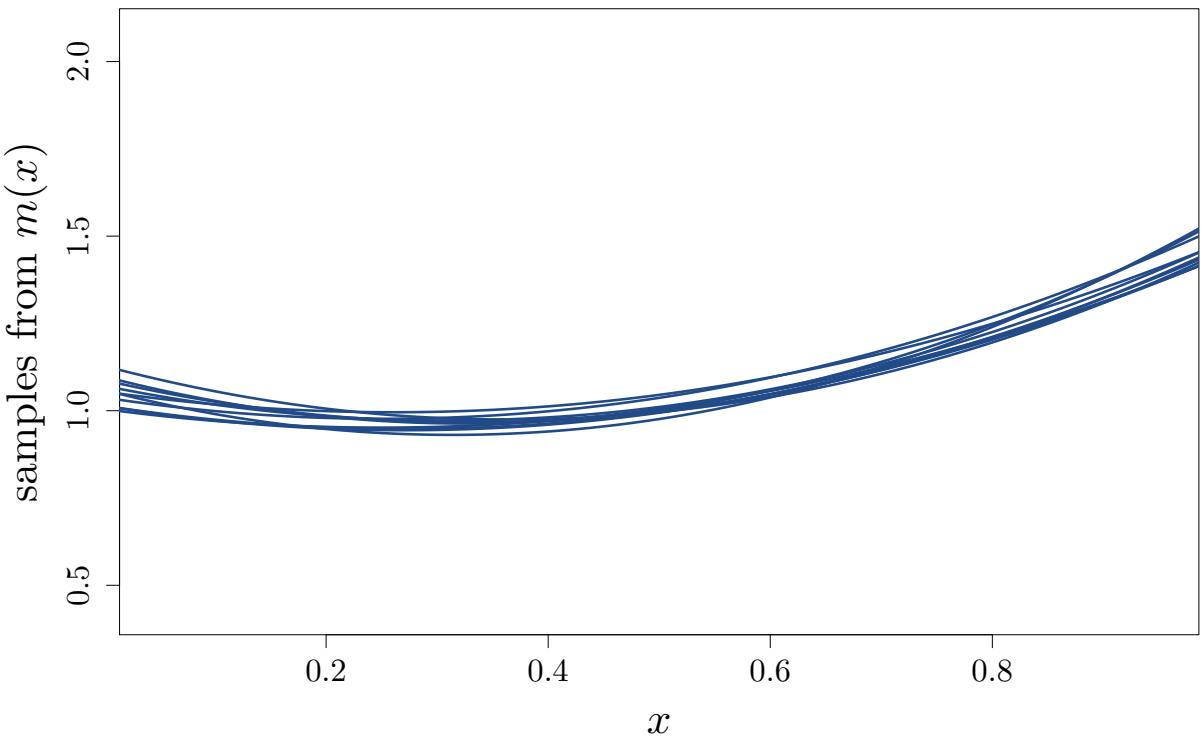
- Its expectation is $\beta \Rightarrow$ The estimator is unbiased
- Its covariance matrix is

$$(B(X)^t B(X))^{-1} B(X)^t \operatorname{cov}[\varepsilon, \varepsilon^t] B(X) (B(X)^t B(X))^{-1}$$

- If ε is multivariate normal, then $\hat{\beta}$ is also multivariate normal.

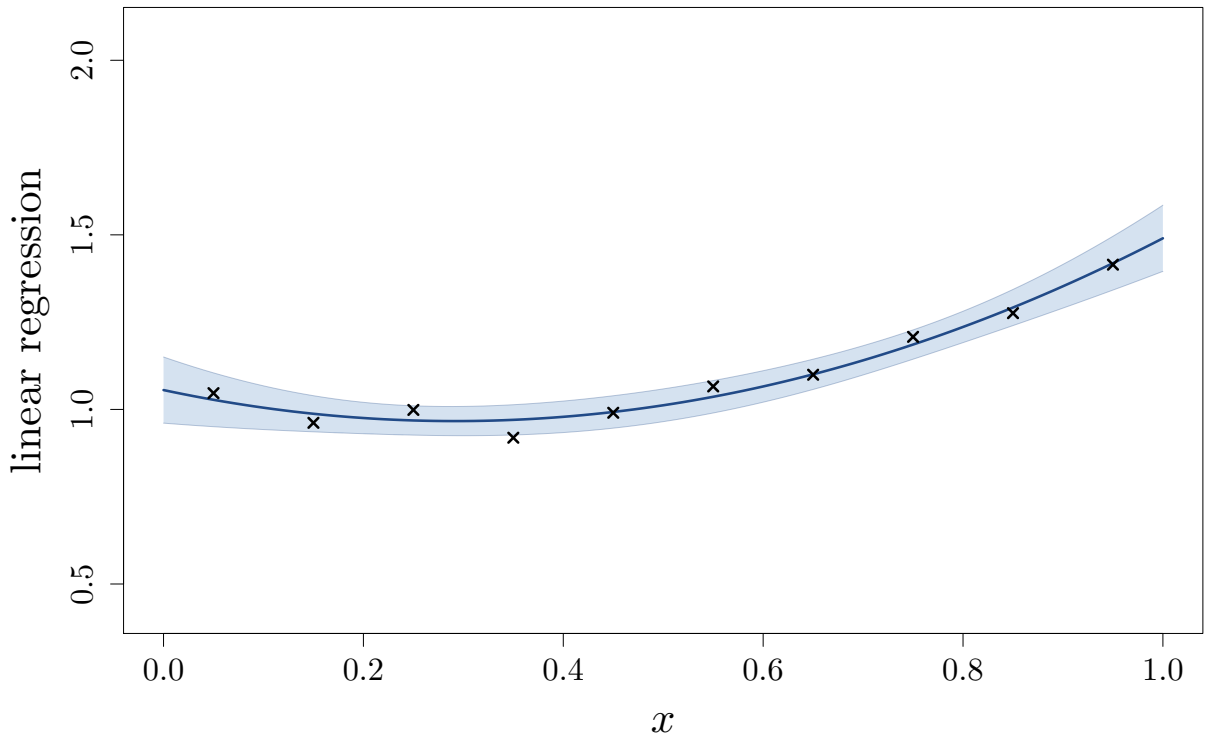
Sampling in the distribution of $\hat{\beta}$ gives us a large variety of models which represent the uncertainty about our estimation:

Back to the example



Back to the example

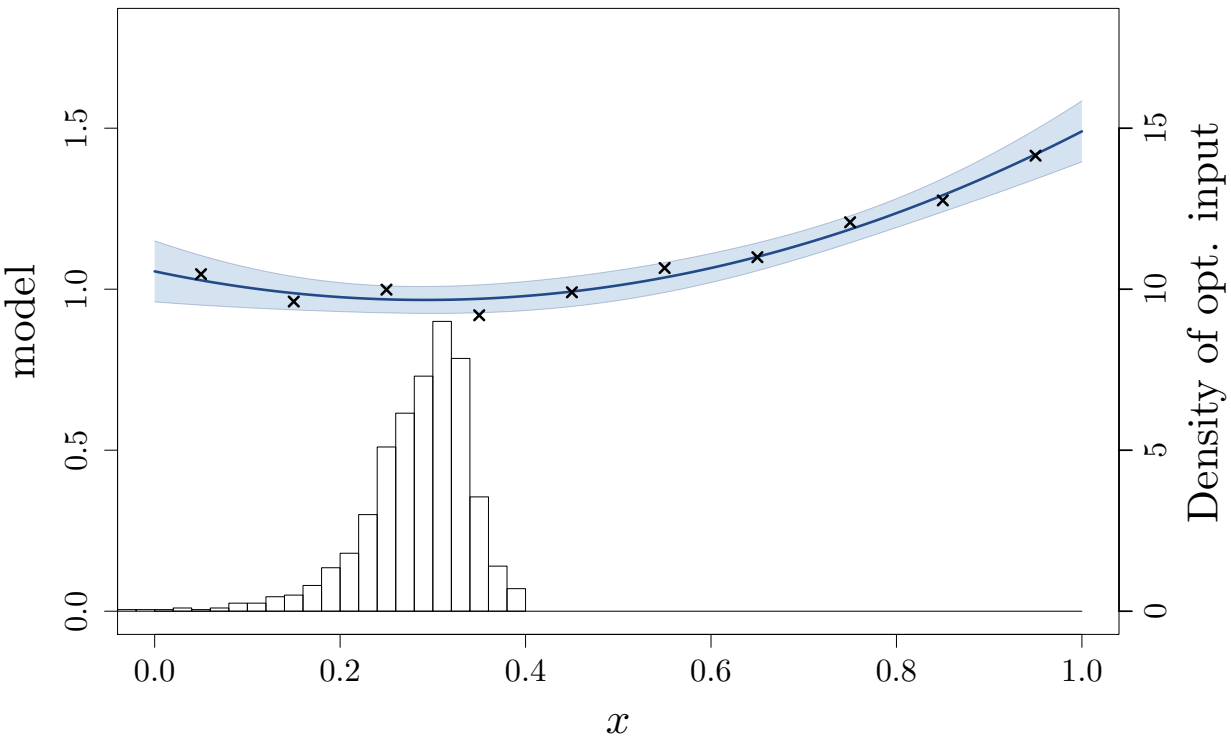
The previous picture can be summarized by showing the mean of m and 95% confidence intervals



Knowing the uncertainty on the model allows to compute an uncertainty on the quantity of interest.

Back to the example

For example, if we are interested in the value $x^\star$ minimizing $f(x)$:



The expectation of $x^\star$ **is not** the input minimizing $m(x)$.

We could dedicate the entire course to linear regression models...

- model validation
- choice of basis functions
- influence of input locations
- ...

We will just stress a few **pros and cons of these models**:

+ provide a good noise filtering
+ are easy to interpret
− are not flexible (need to choose the basis functions)
− do not interpolate
− may explode when using high order polynomials (overfitting)

Polynomial Chaos

The principle of polynomial chaos is to do linear regression with a basis of orthonormalised polynomials.

One dimension

For $x \in \mathbb{R}$, the h_i are of order i. Starting from the constant function $h_0 = 1$, the following ones can be obtain using Gram-Schmidt orthonormalisation.

d-dimension

In $\mathbb{R}^d$, the basis is obtained by a tensor product of one dimensional basis. For example, if $d = 2$:

$$
\begin{aligned}
h_{00}(x) &= 1 \times 1 \\
h_{10}(x) &= h_1(x_1) \times 1 \\
h_{01}(x) &= 1 \times h_1(x_2)
\end{aligned}
\qquad
\begin{aligned}
h_{11}(x) &= h_1(x_1) \times h_1(x_2) \\
h_{20}(x) &= h_2(x_1) \times 1 \\
\vdots \quad &= \qquad \vdots
\end{aligned}
$$

The orthonormal basis H depends on the measure over the input space D.

A uniform measure over $D = [-1, 1]$ gives the **Legendre basis**:

$$h_0(x) = 1/2 \qquad h_3(x) = 7/4\ (5x^3 - 3x)$$
$$h_1(x) = 3/2\ x \qquad h_4(x) = 9/16\ (35x^4 - 30x^2 + 3)$$
$$h_2(x) = 5/4\ (3x^2 - 1) \qquad \vdots \quad = \quad \vdots$$

A standard Gaussian measure over $\mathbb{R}$ gives the **Hermite basis**:

$$h_0(x) = 1/\sqrt{2\pi} \qquad h_3(x) = 1/(6\sqrt{2\pi})\ (x^3 - 3x)$$
$$h_1(x) = 1/\sqrt{2\pi}\ x \qquad h_4(x) = 1/(24\sqrt{2\pi})\ (x^4 - 6x^2 + 3)$$
$$h_2(x) = 1/(2\sqrt{2\pi})\ (x^2 - 1) \qquad \vdots \quad = \quad \vdots$$

Exercice

dimension 1:

Let $F = f(X)$ be a set of observations of $f : [-1, 1] \to \mathbb{R}$. What is the mean value of f according to a polynomial chaos model?

dimension 2:

Let m be a polynonial chaos model based on the basis $H = (h_{00}, h_{10}, h_{01}, h_{11}, h_{20}, h_{02})$. What can you say about $\int m(x_1, x_2)\, \mathrm{d}x_1$?

$\Rightarrow$ Using an appropriate basis makes the computations easy!

Exercice

Can you find a non-polynomial basis that would share the same interesting properties for a uniform measure on $D = [0, 1]$?

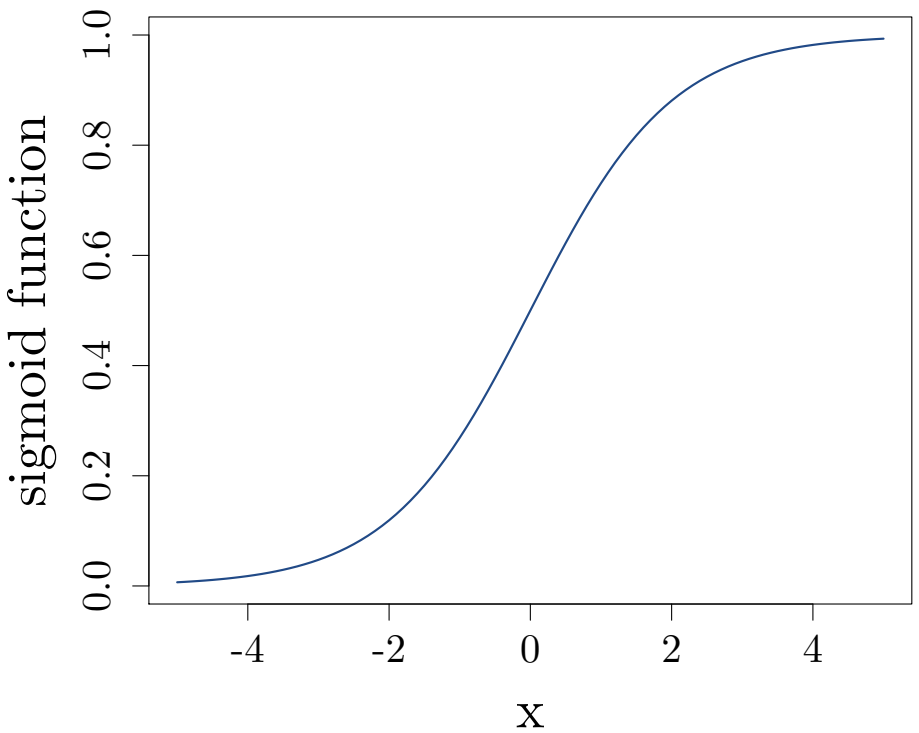
Neural networks

The principle of neural networks is to build a model based on nested functions s:

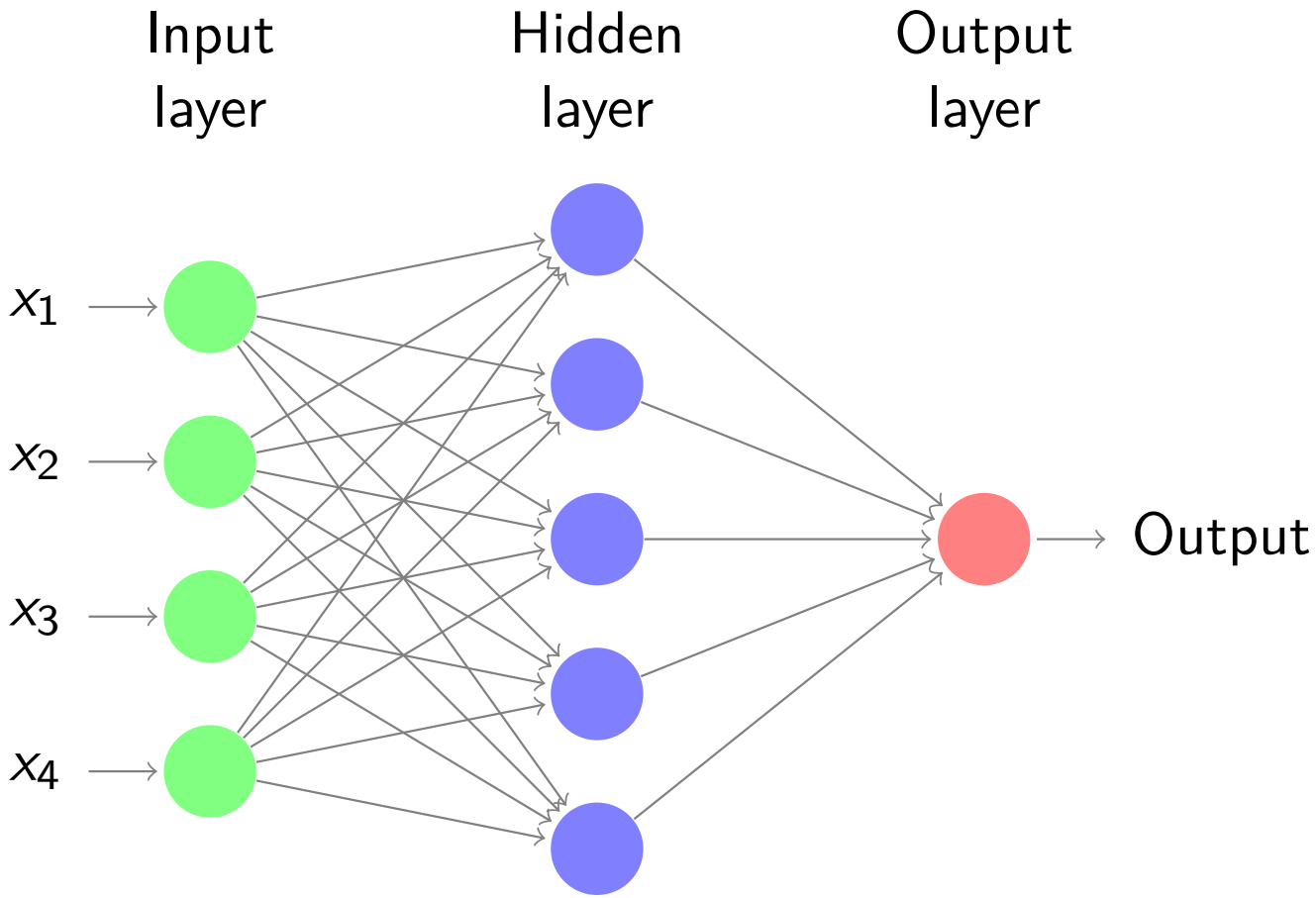
$$m(x) = w_0\ s\left(\sum_i w_{1,i} s\Big(\sum_j w_{2,i,j} x_j + b_2\Big) + b_1\right) + b_0$$

The function s is often chosen to be the sigmoid function:

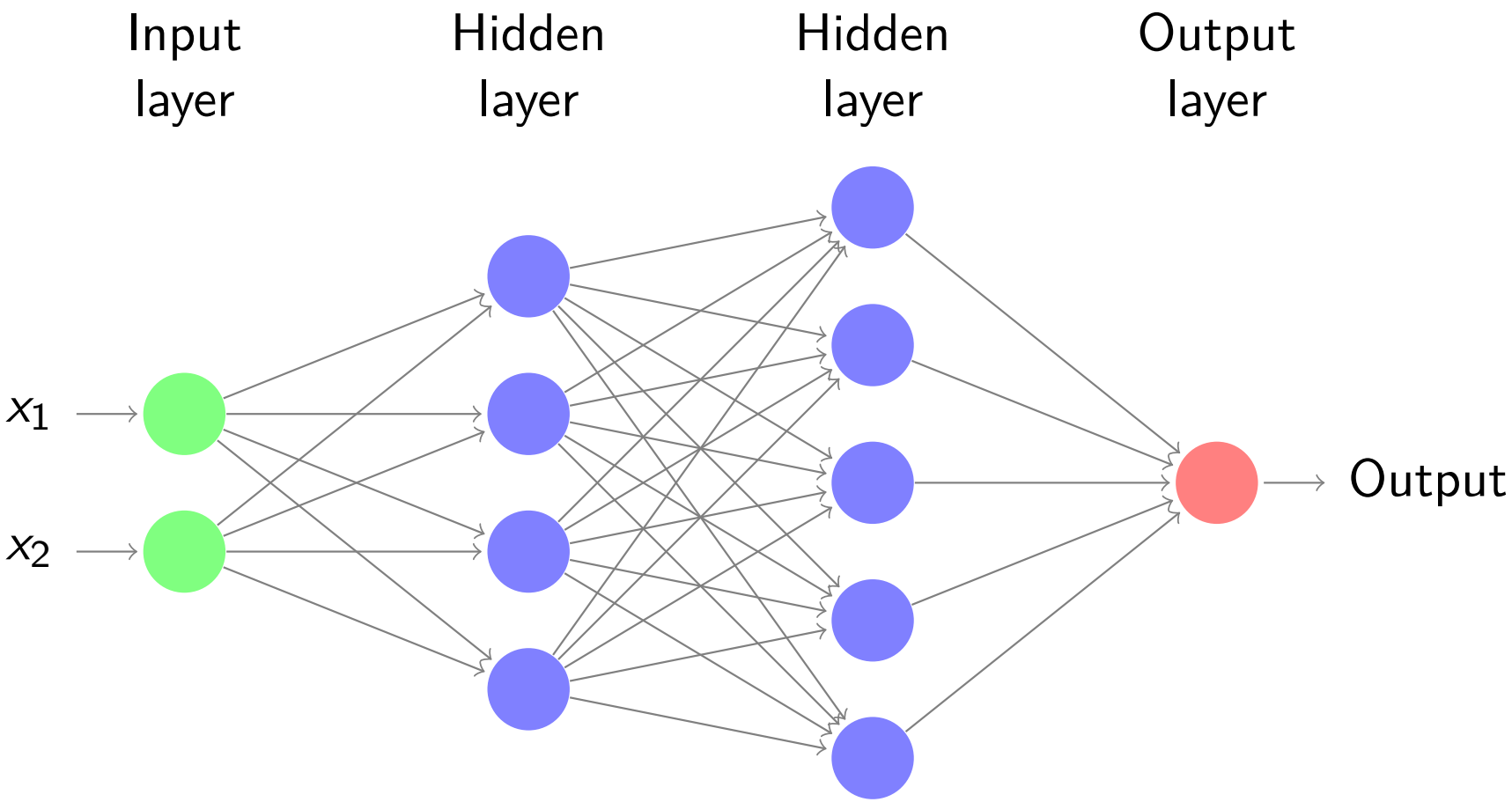
$$s(x) = \frac{1}{1+e^{-x}}$$



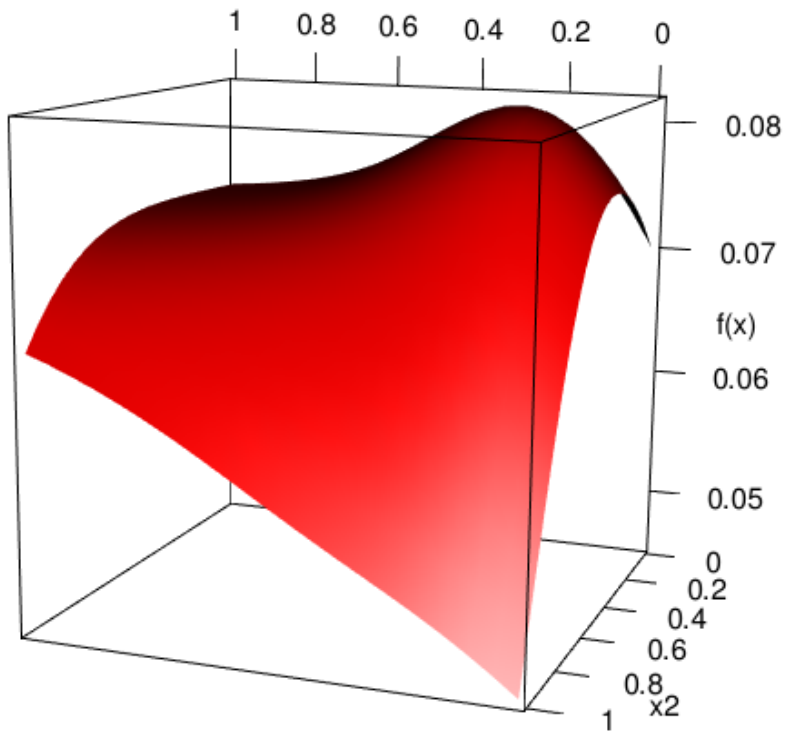
Neural networks can be better represented by graphs:



Of course, there can be many intermediate layers



Here is an example of a function generated with the previous net



Fitting a neural networks model means optimising the parameters w and b. There is typically a very large number of these !

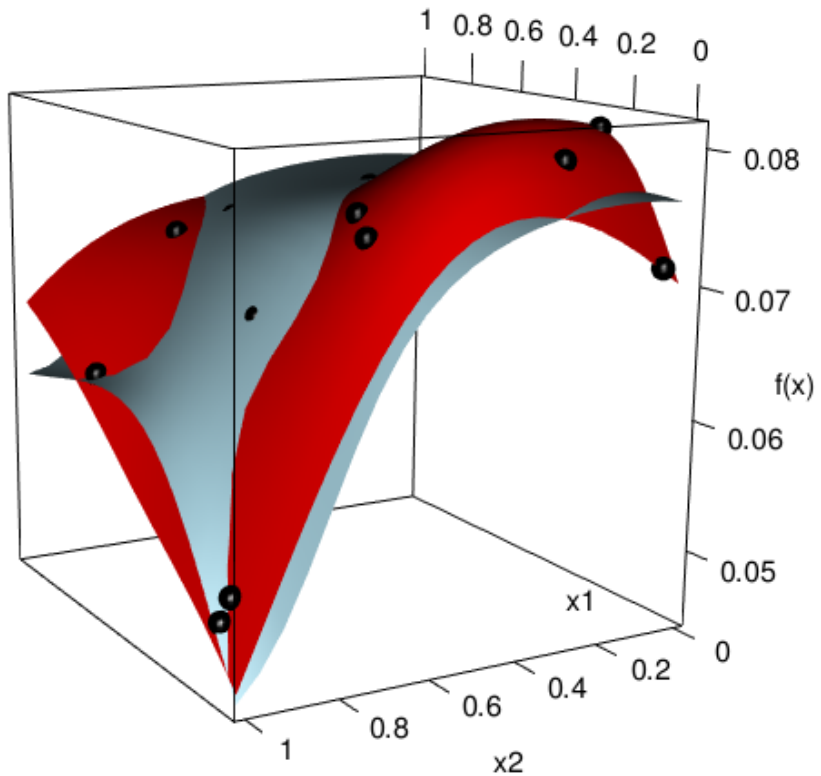
The function to optimise is the residual sum of square:

$$RSS(w, b) = \sum (F_i - m(X_i))^2$$

There is no analytical solution for this optimization problem...
$\Rightarrow$ We use a gradient descent algorithm based on backpropagation.

Example

With the previous net, the total number of parameters to optimize is 43. Given 15 observation points, we obtain the following model:



If there is a large number of observations, computing the gradient itself can be expensive.

Stochastic gradient is an alternative. The principle is to separate the data in smaller batches and to approximate the RSS gradient by the RSS gradient of one of the batches. The batches are treated one after each other, and we call an epoch the treatment of all batches.

Neural networks are famous for:

- Handling large datasets
- computer vision

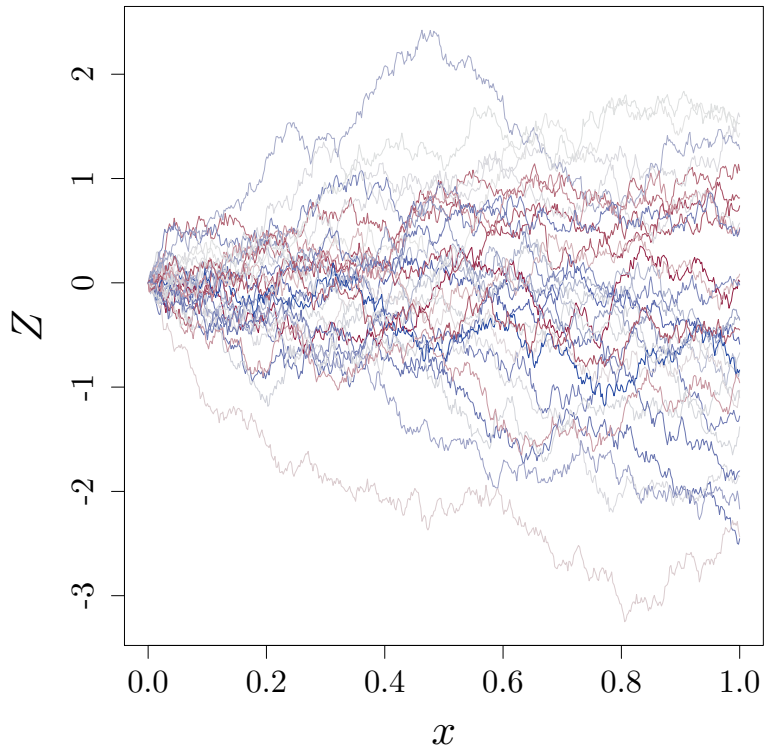
Their main drawbacks are:

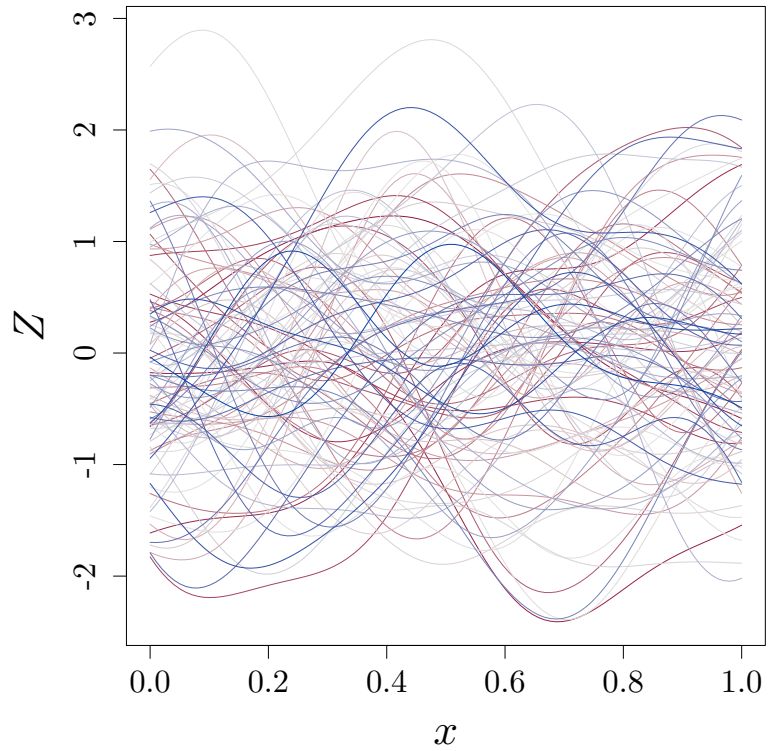
- Computationally expensive to train
- Choosing the structure (nb of hidden layers, ...) is difficult.

Gaussian Process Regression

A random process process Z is an object that returns a function for each draw:

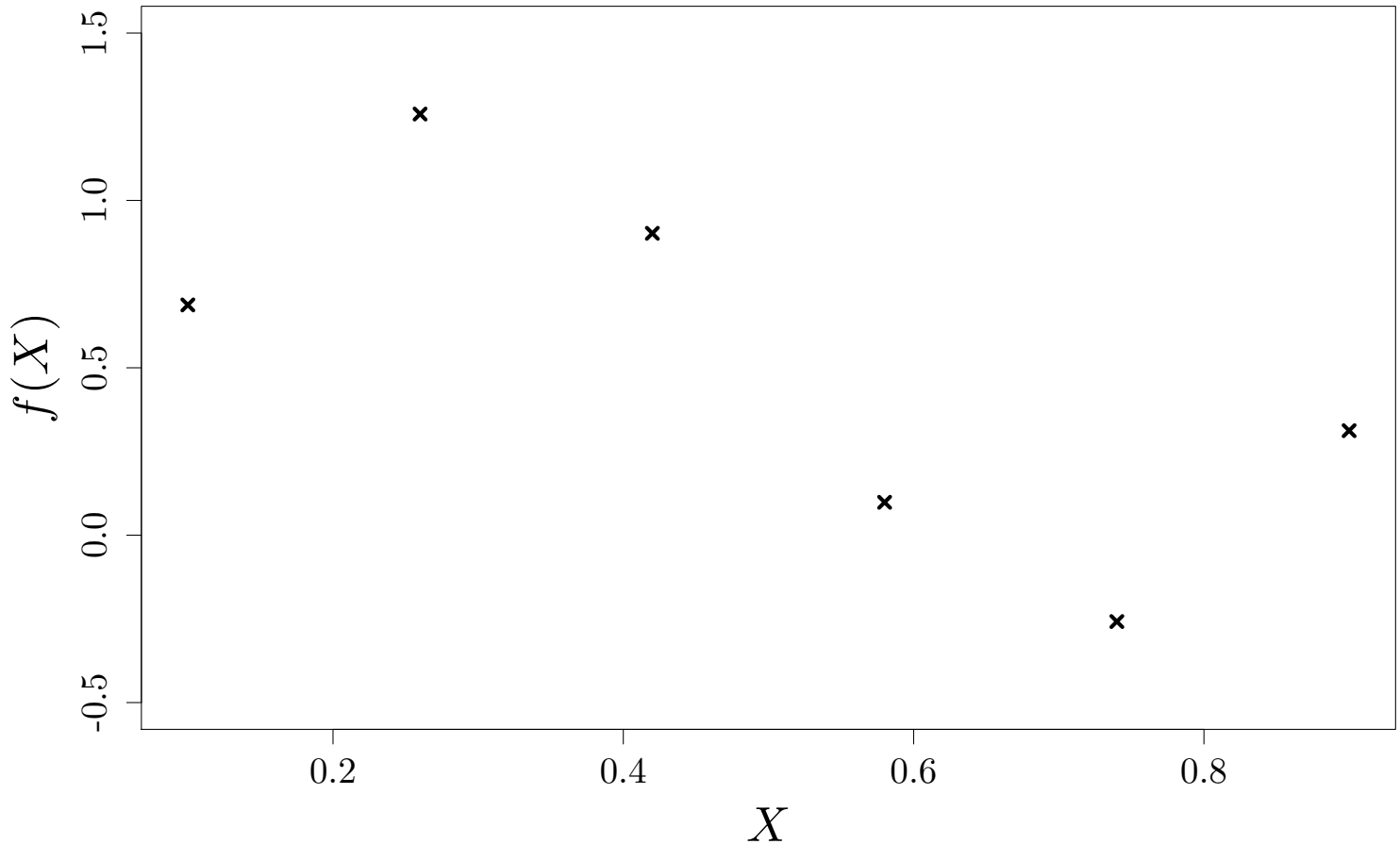
Two examples





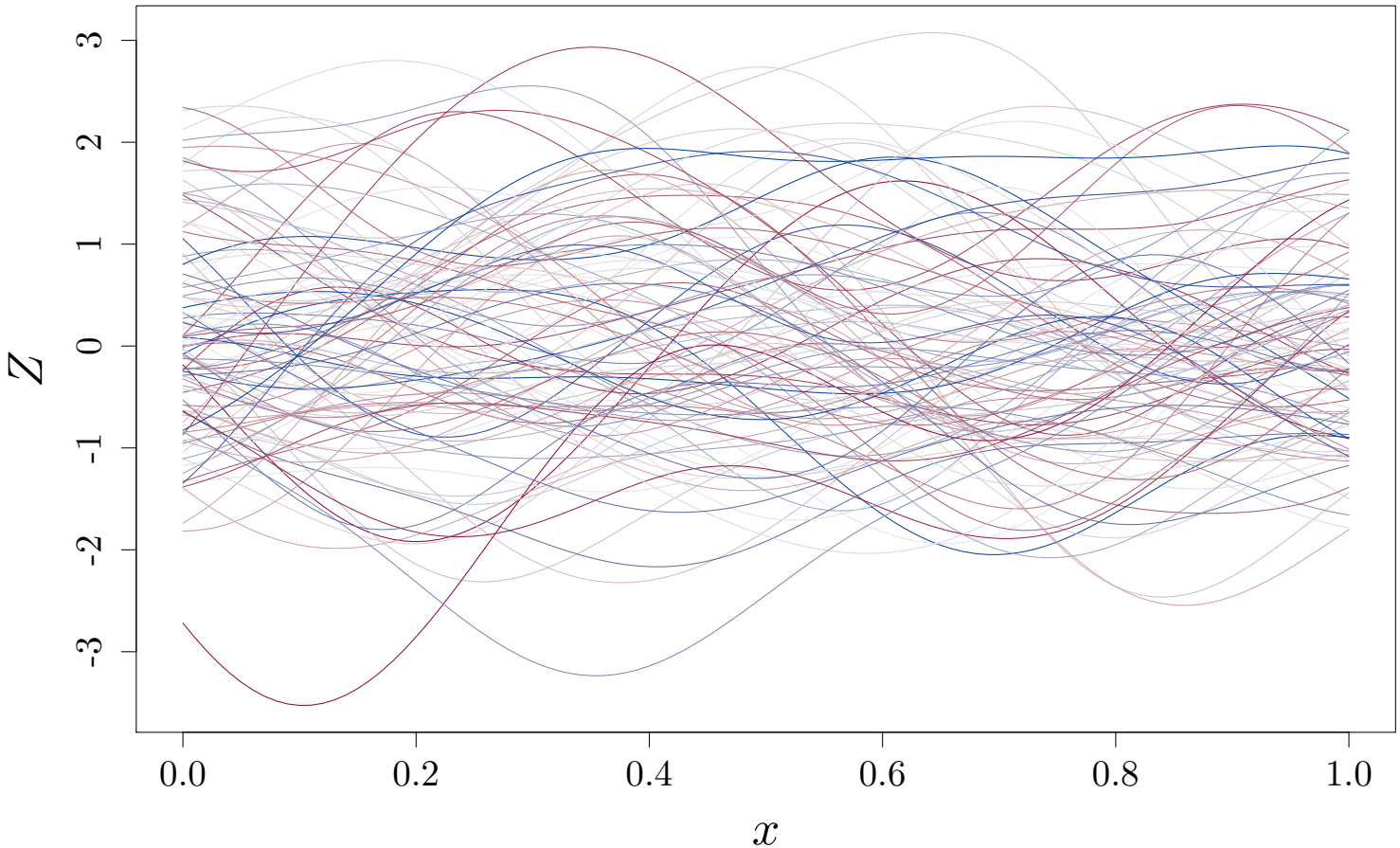
A random process process can be used to describe the **prior knowledge** we have about the function to approximate.

We have observed the function f for a set of points $X = (X_1, \ldots, X_n)$:



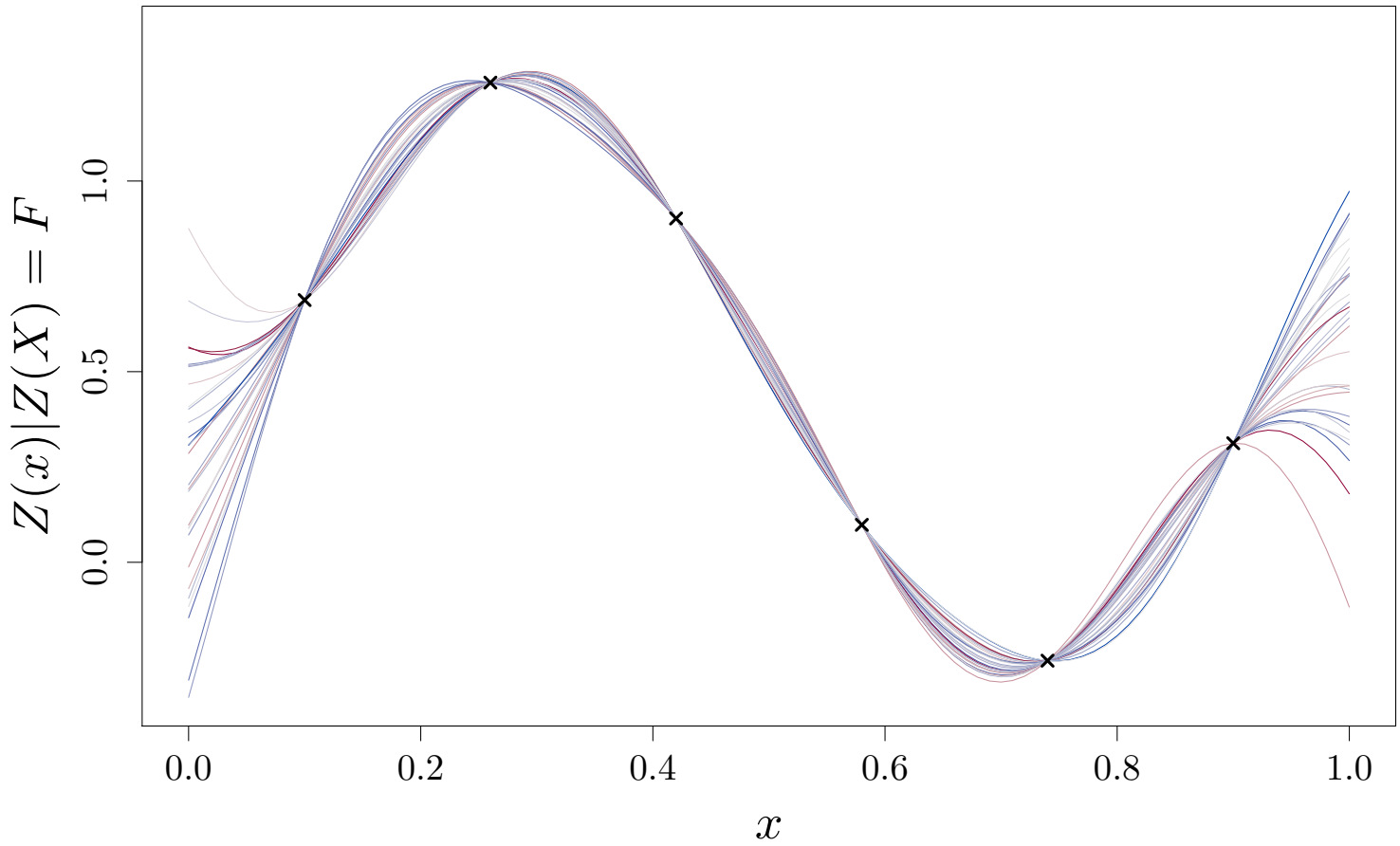
The vector of observations is $F = f(X)$ (ie $F_i = f(X_i)$).

We assume that f behaves as follow :

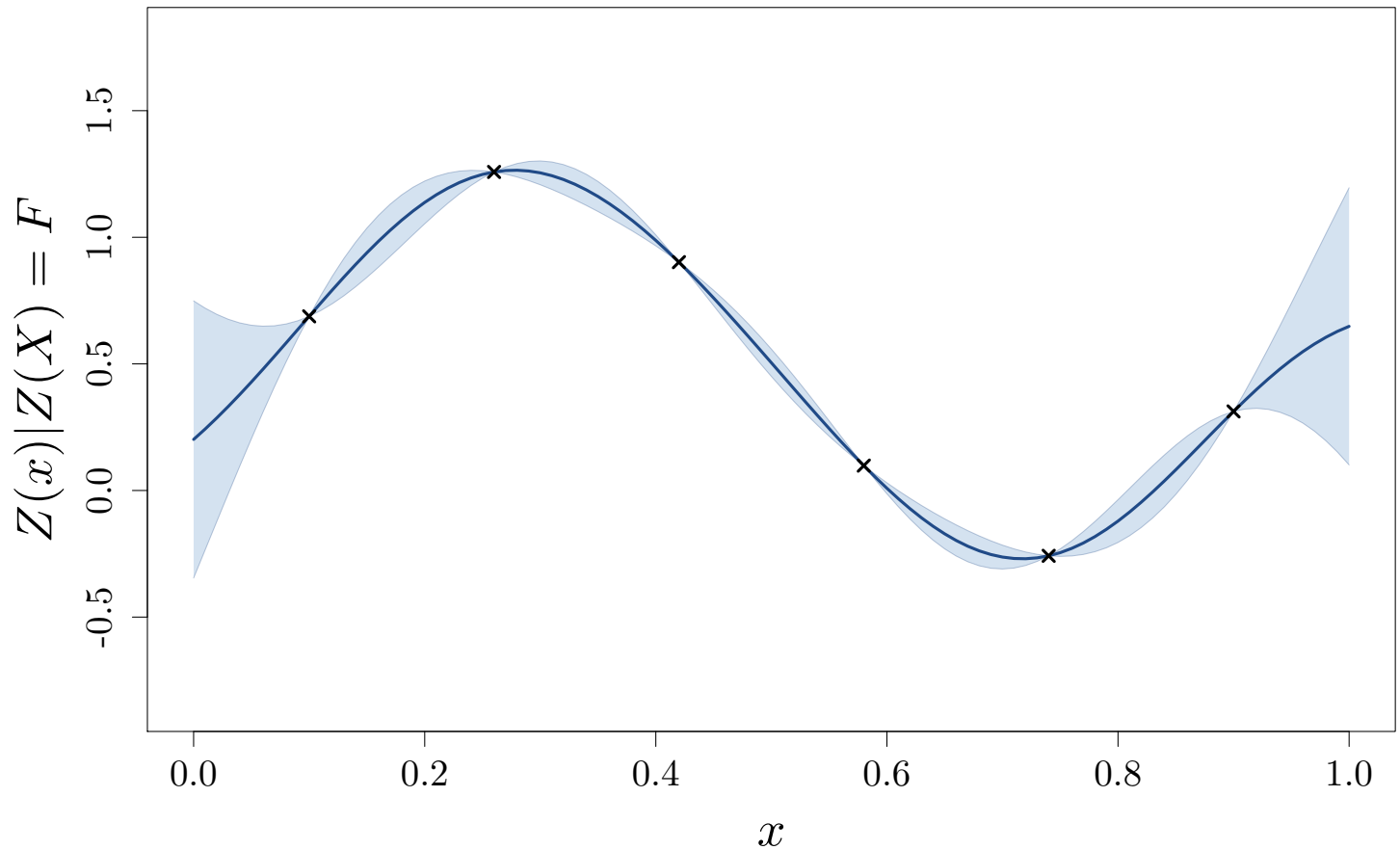


What can we say if we combine these two informations?

We obtain the following **conditional samples**:



Which can be summarized by a mean function and confidence intervals:



We will detail how to build such models in the next classes...