Optimisation

Olaf Hauk

MRC Cognition and Brain Sciences Unit
olaf.hauk@mrc-cbu.cam.ac.uk

What is mathematical optimisation?

We are looking for an element x (number, vector, matrix...)
that minimises or maximises a "cost function", e.g.

$$x_m : f(x_m) < f(x) \; \forall \; x$$

The cost function could represent :
energy, money, residual variance, amount of waste, time etc. etc.

A maximisation problem for $f(x)$ can be turned into a minimisation problem, e.g.

$$f(x) \rightarrow 1\text{-}f(x)$$

$$f(x) \rightarrow \frac{1}{f(x)}$$

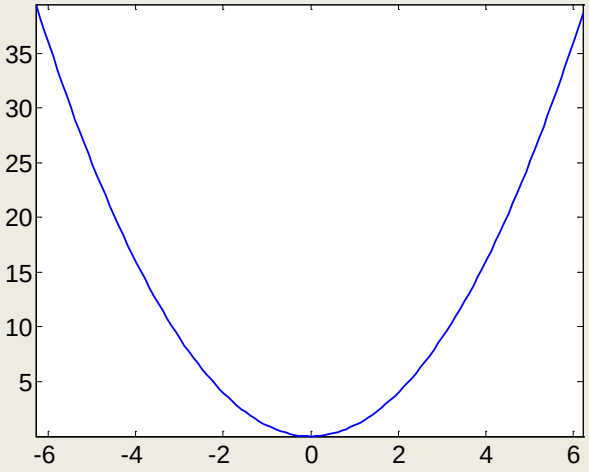
Certain transformations don't affect the location of maxima/minima,
but may make the problem easier to handle, e.g. :

$$\min e^{f(x)} \rightarrow \min \log(e^{f(x)}) \rightarrow \min f(x)$$

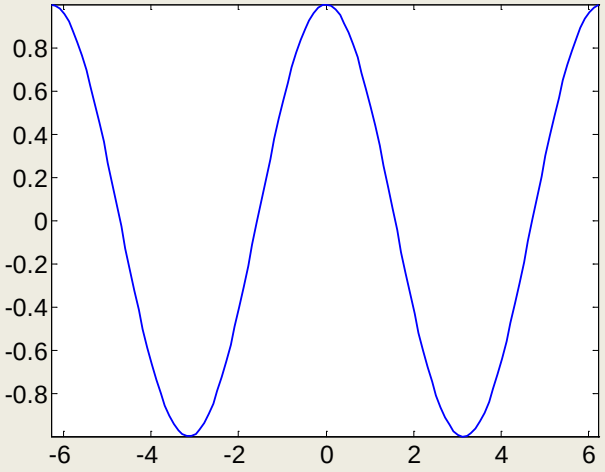
(Often works with probabilities/likelihoods)

Local and Global Minima

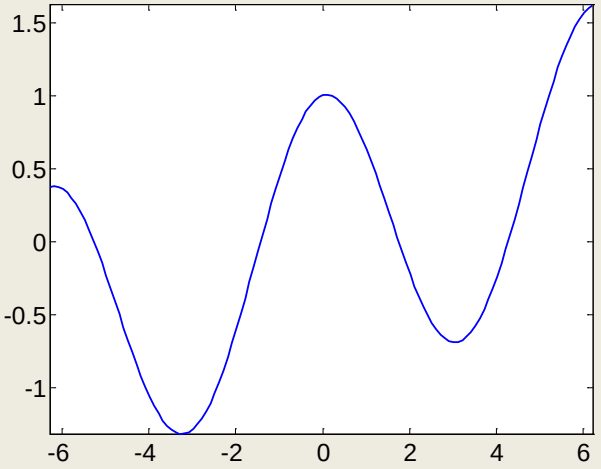
One local minimum
Unique global minimum



Two local minima
No unique global minimum



Two local minima
Unique global minimum



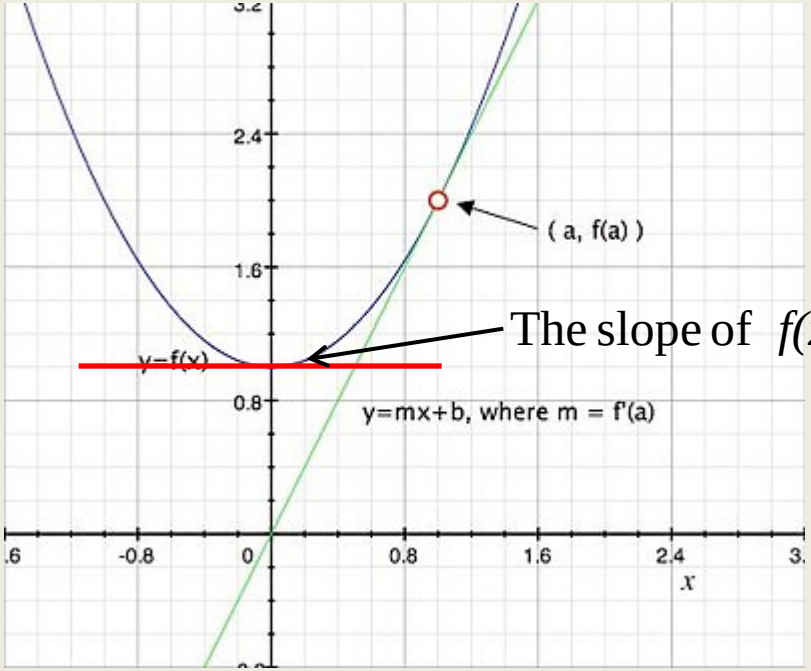
$$\text{Global minimum}:$$

$$x_m : f(x_m) < f(x) \; \forall \; x$$

$$\text{Local minimum}:$$

$$x_m : f(x_m) < f(x) \; \forall \; x \in [x_m - \varepsilon : x_m + \varepsilon], \varepsilon > 0$$

Finding an Analytic Solution



The slope of $f(x)$ at a global extremum is 0.

The first derivative of $f(x)$, i.e. $f'(t)$, is zero at a local extremum.

First steps to find extrema :

1) compute the first derivative of $f(x)$, $f'(x)$

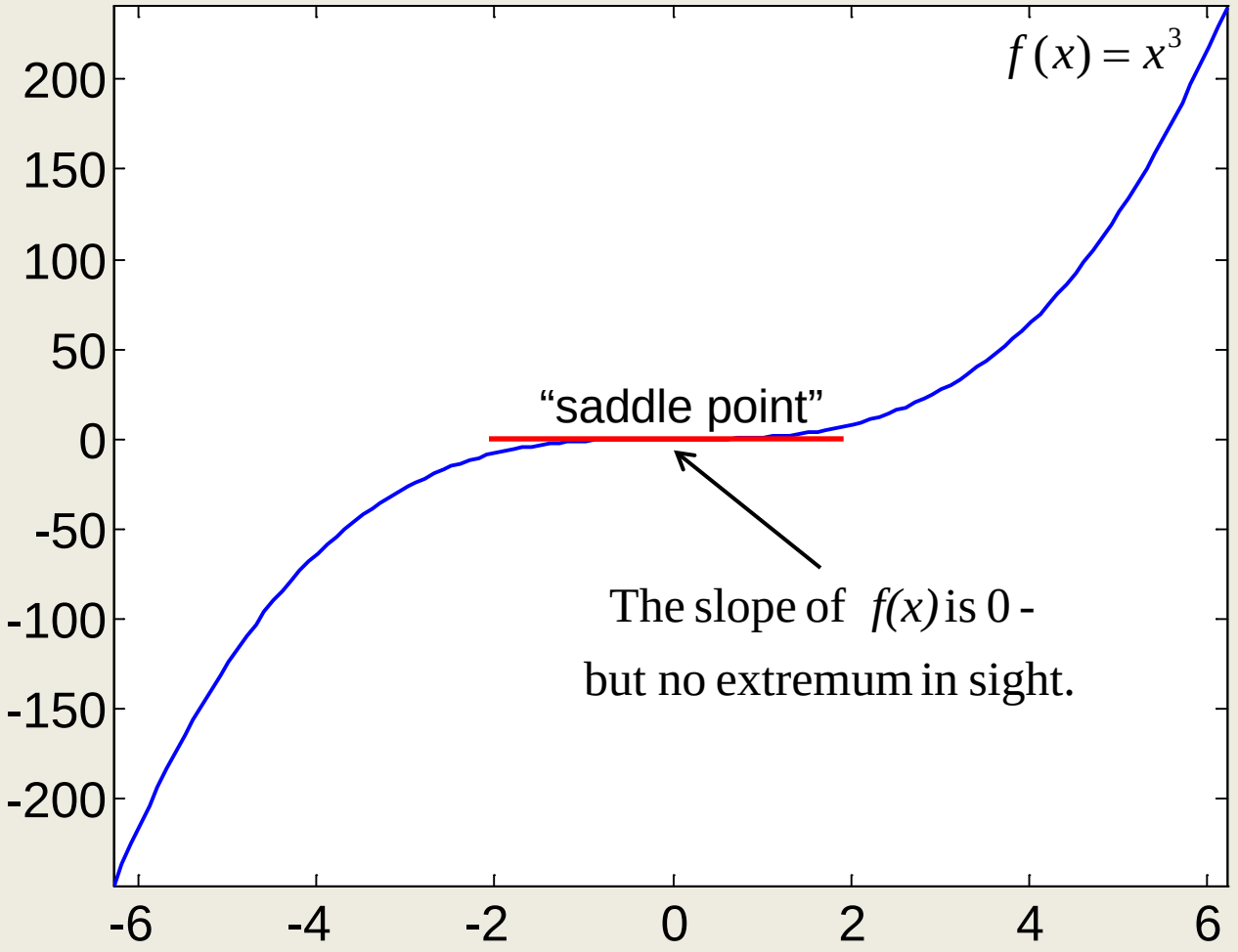
2) Find x for which $f'(x) = 0$

3) Check if those x are minima or maxima.

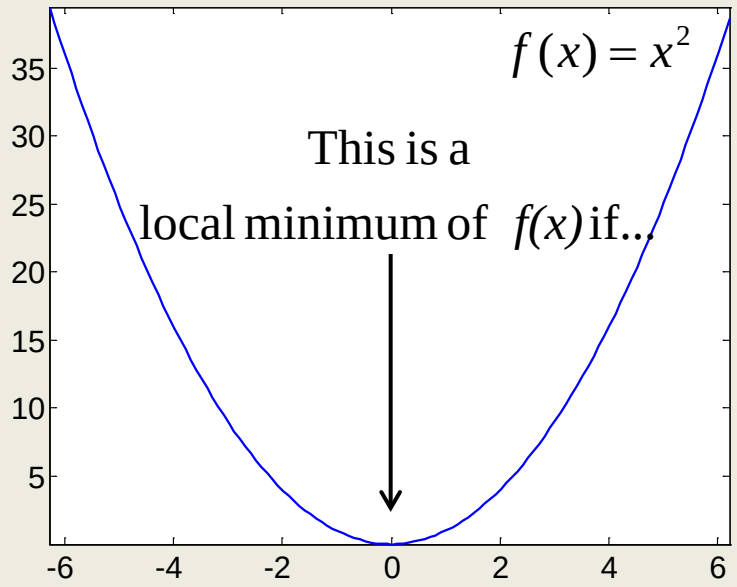
4) If there are more than one, check if there is a global extremum.

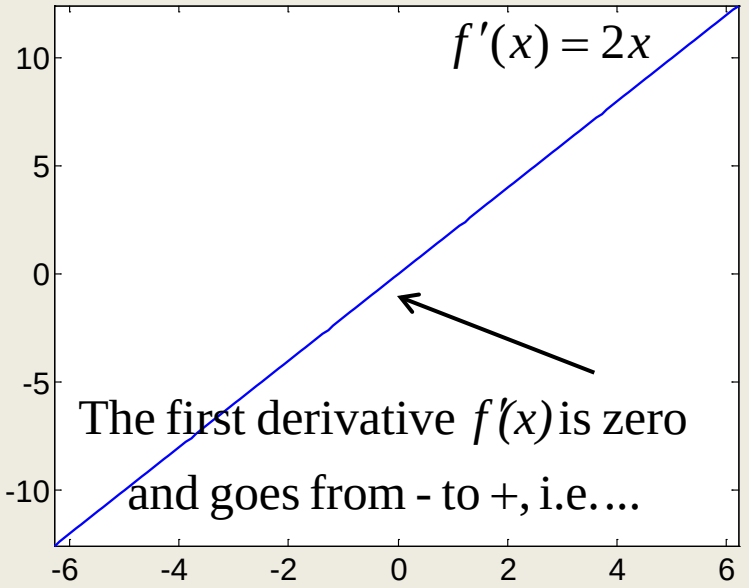
There is only one thing missing...

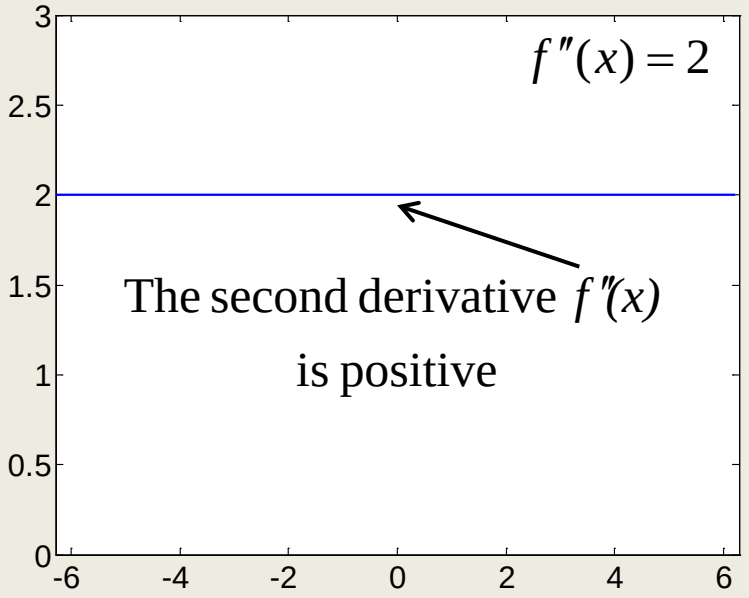
What about…?


$f(x) = x^3$
200
150
100
50
0
-50
-100
-150
-200
-6
-4
-2
0
2
4
6
“saddle point”
The slope of $f(x)$ is 0 -
but no extremum in sight.

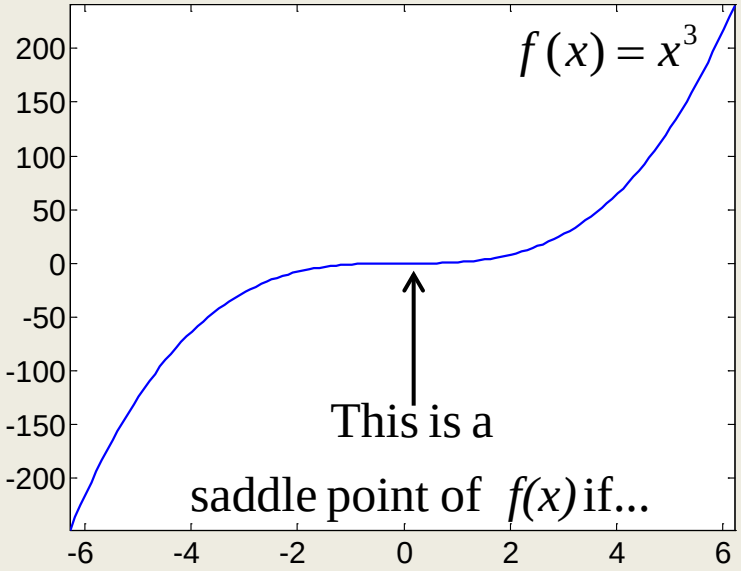
How To Distinguish Saddle Points from Extrema
The Second Derivative

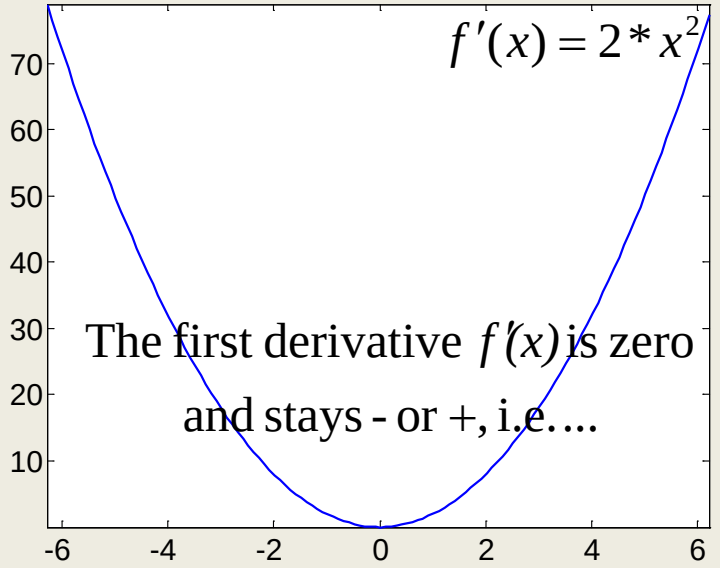


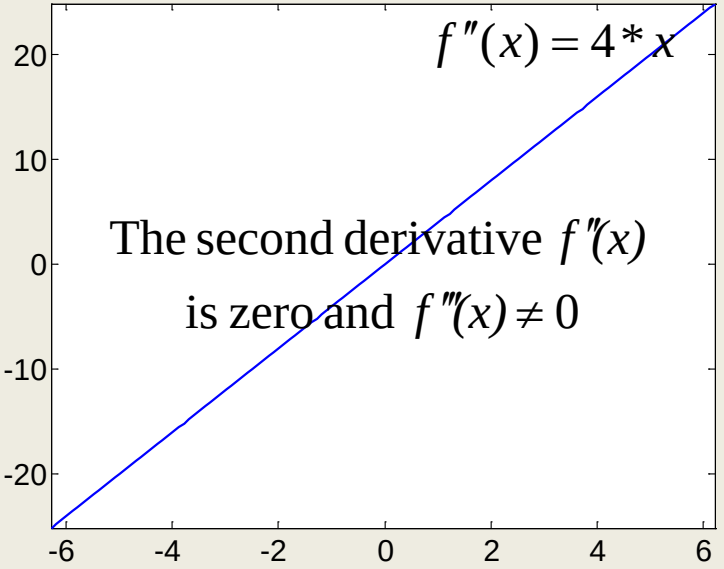




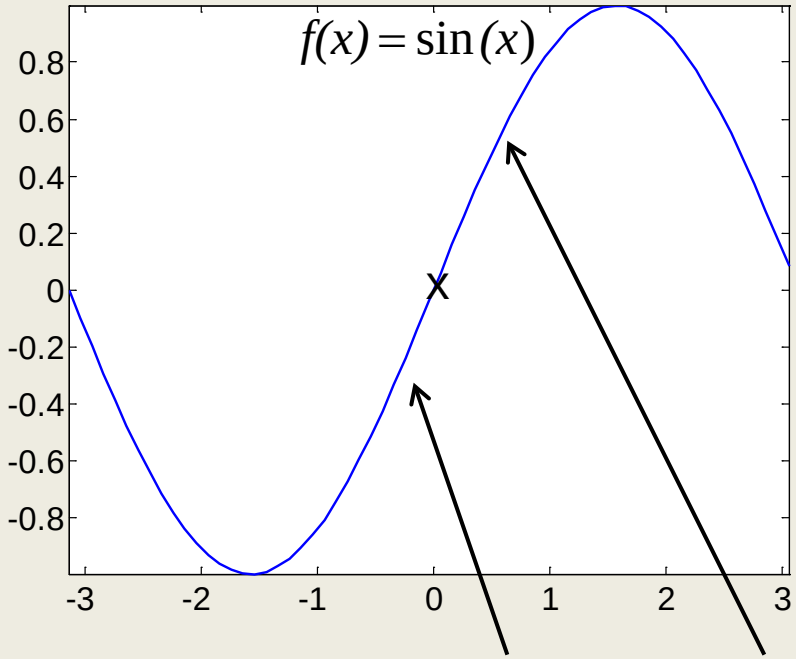
How To Distinguish Saddle Points from Extrema
The Second Derivative



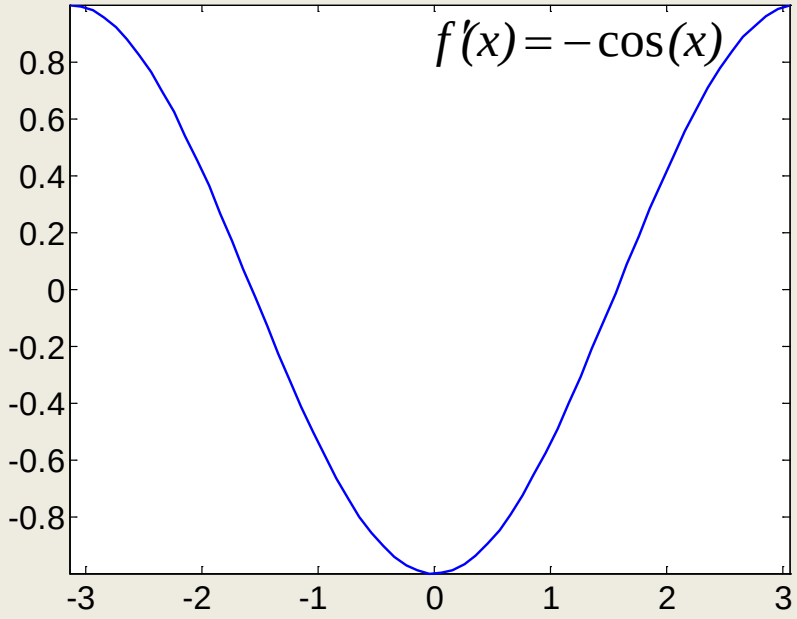




To Make It More Confusing - Inflection Points



The curve changes from "convex" to "concave"



i.e. f ′(x) has a local extremum

Least-Squares Estimation

Arguable the most common cost functions are "goodness - of - fit" measures,
i.e. you have some data d, some model parameters m,
and a forward model that predicts d from m, i.e.

$$\hat{d} = f(m)$$

You want to find m that minimises the difference between prediction and measurement :

$$\hat{m} \rightarrow \min(F(d - f(\hat{m})))$$

according to some cost function F()

The most common cost function is the least - squares difference
between prediction and measurement :

$$\hat{m} \rightarrow \min((d - f(\hat{m}))^2)$$

If **d** is a vector :

$$\hat{m} \rightarrow \min(\sum_i (d_i - f_i(\hat{m}))^2))$$

Overfitting

We already talked about linear estimation in the context of the General Linear Model and matrix (pseudo)inversion

Once a problem is of the form

$$\mathbf{y} = \mathbf{A}\mathbf{x}$$

Matlab can provide a solution with

$$\hat{\mathbf{x}} = pinv(\mathbf{A})*\mathbf{y}$$

It may not be obvious which basis function to choose for the columns of **A**, or how many.

Overfitting

Not enough parameters:
You may leave more data unexplained than necessary –
you don’t get as much as you deserve.

Too many parameters:
You may get a close fit of your data – but some of it is noise.
You may get more than you asked for - “overfitting”.

We need a criterion that weighs the amount of data explained against the number of parameters used.

Avoiding Overfitting

Akaike Information Criterion

$$\mathrm{AIC} = 2k - 2\ln(L)$$

Bayesian Information Criterion

$$BIC = k * \ln(n) - 2\ln(L)$$

In the case of equal known variances:

$$AIC \propto 2k + n * \ln(RSS)$$

$$BIC = k * \ln(n) + n * \ln(RSS / n)$$

k : number of model parameters

L : maximum likelihood of model

(related to model fit)

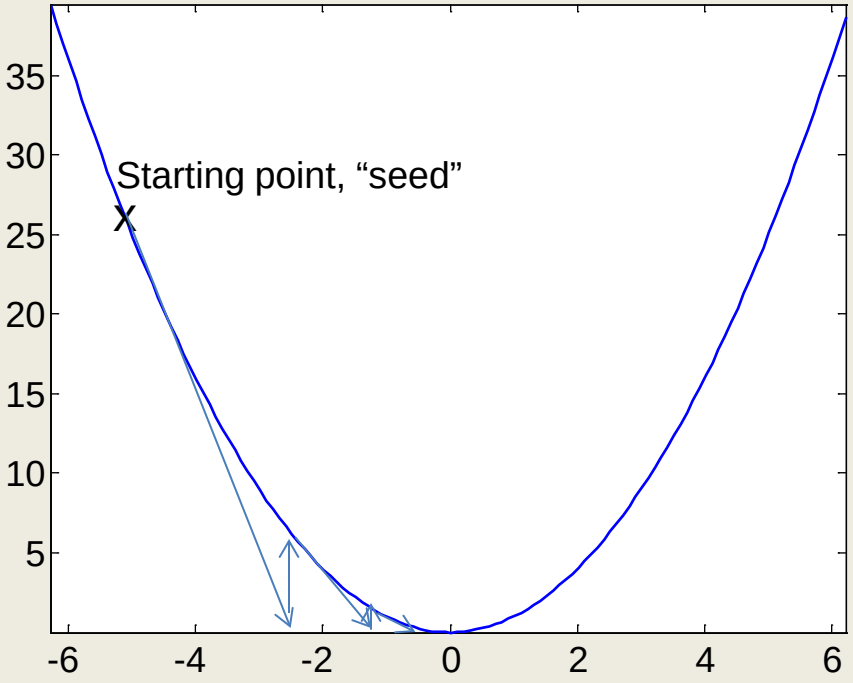
n : number of data points

RSS : Residual Sum of Squares

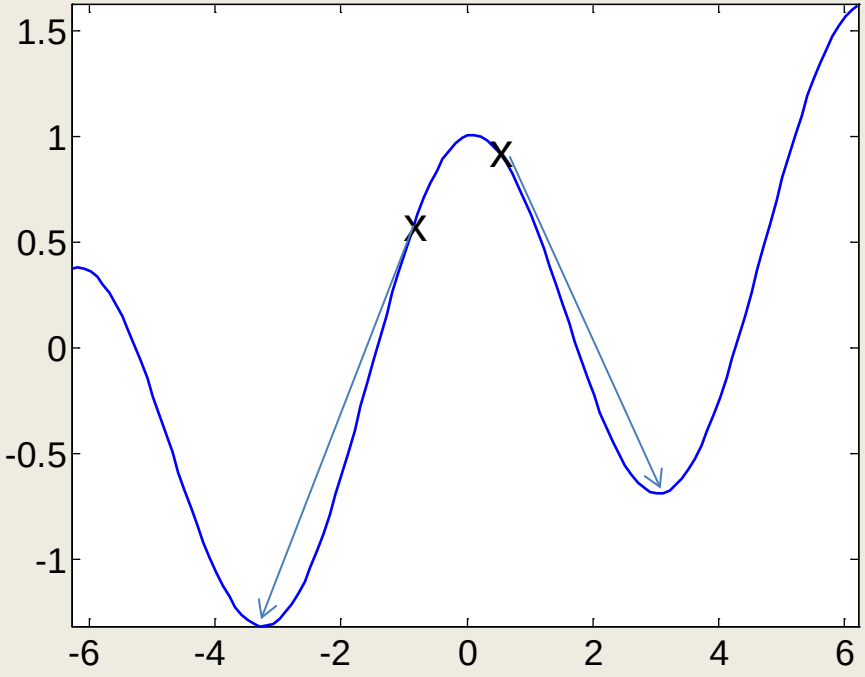
Non-linear Optimization

If your optimisation problem does not have an analytic solution (using derivatives etc.), or cannot be formulated as a GLM, then one can use iterative search procedures.

If your function is reasonably smooth, then the first derivate will guide the way "Gradient Descent"



Be aware of multiple local minima, you may have to use multiple seeds



These methods require some information about the structure of your data, e.g. smoothness and multiple local minima.

If Nothing Else Works…

If your optimisation problem is very complex, or you don't know much/anything about the structure of your cost function, then you can apply "clever random search" approaches:

- Monte Carlo Markov Chain (MCMC)
- Genetic algorithms
- Simulated annealing
- …