Computational Finance with C++
Numerical Methods for Optimization Models

Panos Parpas
Department of Computing
Imperial College London

www.doc.ic.ac.uk/∼pp500
p.parpas@imperial.ac.uk

Where are Optimization Models used in finance?

- Portfolio Selection (Markowitz Model)
- Parameter Estimation and Calibration e.g. smile modelling
- Machine Learning

Outline

1. **Linear Algebra**
   a) Basic operations on vectors and matrices
   b) Linear Equations
   c) Linear Independence
   d) Rank of a matrix, positive definite matrices
2. **Analysis**
   a) Gradient, Jacobian, Hessian
   b) Convex sets
   c) Convex functions
3. **Optimality Conditions**
4. **Optimization Algorithms**
   a) First order algorithms
   b) Second order algorithms

Additional material:

- Chapters 1-5, in *An Introduction to Optimization*, Chong & Zhak, Third Edition.
- Appendix A-C, in *Linear and Non-Linear Programming*, Luenberger & Ye, Third Edition.

Vectors

- We define a **column vector** in $\mathbb{R}^n$ as an array of n numbers,

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

- A **row vector** in $\mathbb{R}^n$ is

$$x = (x_1, x_2, \ldots, x_n)$$

- The **transpose** of a column vector is a row vector. If x is a column vector in $\mathbb{R}^n$ then,

$$x^\top = (x_1, x_2, \ldots, x_n)$$

- In this course all vectors are column vectors

Operations on Vectors

- Two vectors x and y are **equal**, $x = y$ if $x_i = y_i$, $i = 1, \ldots, n$
- The **sum of two vectors** $x + y$ is the vector

$$(x_1 + y_1, x_2 + y_2, \ldots, x_n + y_n)^\top$$

- **Multiplication** of a vector x by a scalar α is defined as,

$$\alpha x = (\alpha x_1, \alpha x_2, \ldots, \alpha x_n)^\top$$

Matrices

A **matrix** is a rectangular array of entries with n rows and m columns.
A matrix B with n rows and m columns belongs to $\mathbb{R}^{n\times m}$
A **symmetric matrix** is one that is equal to its transpose.
Multiplication: If $A \in \mathbb{R}^{n\times m}$ and $B \in \mathbb{R}^{m\times k}$, then $C = AB$ is a matrix in $\mathbb{R}^{n\times k}$. The $(i,j)^{th}$ entry of C is,

$$C_{ij} = \sum_{l=1}^{m} A_{il}B_{lj}$$

For two matrices A and B, the **transpose of their product** is $(AB)^\top = B^\top A^\top$.

Linear Equations I

Suppose that we are given m equations in n unknowns of the form,

$$
\begin{aligned}
a_{11}x_1 + \alpha_{12}x_2 + \ldots + a_{1n}x_n &= b_1 \\
a_{21}x_1 + \alpha_{22}x_2 + \ldots + a_{2n}x_n &= b_2 \\
&\vdots \\
a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n &= b_m.
\end{aligned}
$$

We can also represent the set of equations above in vector notation,

$$
a_1x_1 + a_2x_2 + \cdots + a_nx_n = b
$$

where

$$
a_j = \begin{pmatrix} a_{1j} \\ a_{2j} \\ \vdots \\ a_{mj} \end{pmatrix} \quad \text{and} \quad b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}
$$

Linear Equations II

We associate the matrix,

$$A = [a_1, a_2, \ldots, a_n]$$

with the system of equations, and represent the system as follows,

$$Ax = b$$

where

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

Matrices

A matrix $A \in \mathbb{R}^{n \times n}$ is called **positive semidefinite** if for all $d \in \mathbb{R}^n$,

$$d^\top A d \geq 0.$$

We use the notation $A \succeq 0$.
If the above inequality is satisfied strictly, i.e. if

$$d^\top A d > 0 \; \forall d \in \mathbb{R}^n \backslash 0,$$

then A is called **positive definite**.

Function notation

We write $f : X \to Y$ to mean that a function takes points from the set X (domain) to the set Y (range).

Example: The function,

$$f(x) = f(x_1, x_2) = \begin{bmatrix} x_1^2 + x_2 \\ \exp(x_1) + x_2 \\ x_2 \end{bmatrix},$$

evaluated at the point $(2, -1)$ is,

$$f(x) = f(2, -1) = \begin{bmatrix} 3 \\ \exp(2) - 1 \\ -1 \end{bmatrix}$$

Differentiation

The derivative of a function in one dimension is defined below.

$$\frac{df(x)}{dx} = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

When the above holds, in the sense that the limit exists, then we say that the **function is differentiable at the point** x.
When a function is differentiable and its derivative is also continuous we say that the function is **continuously differentiable**.

Differentiation n-dimensions

If a function is defined over $\mathbb{R}^n$, then its **partial derivative** with respect to dimension i is defined as,

$$\frac{\partial f(x)}{\partial x_i} = \lim_{h \to 0} \frac{f(x_1, \dots, x_i + h, \dots, x_n) - f(x_1, \dots, x_i, \dots, x_n)}{h}$$

The vector of the partial derivatives is called the **gradient**

$$\nabla f(x) = \begin{bmatrix} \frac{\partial f(x)}{\partial x_1} \\ \vdots \\ \frac{\partial f(x)}{\partial x_n} \end{bmatrix}$$

The Jacobian

If $f : \mathbb{R}^n \to \mathbb{R}^m$ we evaluate the gradient of each function, and call the matrix of derivatives given by,

$$\nabla f(x) = \begin{bmatrix} \frac{\partial f_1(x)}{\partial x_1} & \dots & \frac{\partial f_m(x)}{\partial x_1} \\ \vdots & & \vdots \\ \frac{\partial f_1(x)}{\partial x_n} & \dots & \frac{\partial f_m(x)}{\partial x_n} \end{bmatrix}$$

the gradient of f.
The transpose of the gradient is called the **Jacobian** matrix.

Example

Find the gradient matrix of,

$$f(x_1, x_2) = \begin{bmatrix} x_1 x_2 \\ \sin(x_1 + x_2) \\ x_1^2 - x_2^2 \end{bmatrix}$$

Then,

$$\nabla f_1(x) = \begin{bmatrix} x_2 \\ x_1 \end{bmatrix}, \ \nabla f_2(x) = \begin{bmatrix} \cos(x_1 + x_2) \\ \cos(x_1 + x_2) \end{bmatrix}, \ \nabla f_3(x) = \begin{bmatrix} 2x_1 \\ -2x_2 \end{bmatrix}$$

We therefore have the gradient matrix.

$$\nabla f(x) = \begin{bmatrix} \nabla f_1(x) & \nabla f_2(x) & \nabla f_3(x) \end{bmatrix} = \begin{bmatrix} x_2 & \cos(x_1 + x_2) & 2x_1 \\ x_1 & \cos(x_1 + x_2) & -2x_2 \end{bmatrix}$$

Example

Find the gradient matrix of,

$$f(x_1, x_2) = \begin{bmatrix} x_1 x_2 \\ \sin(x_1 + x_2) \\ x_1^2 - x_2^2 \end{bmatrix}$$

Then,

$$\nabla f_1(x) = \begin{bmatrix} x_2 \\ x_1 \end{bmatrix}, \ \nabla f_2(x) = \begin{bmatrix} \cos(x_1 + x_2) \\ \cos(x_1 + x_2) \end{bmatrix}, \ \nabla f_3(x) = \begin{bmatrix} 2x_1 \\ -2x_2 \end{bmatrix}$$

We therefore have the gradient matrix.

$$\nabla f(x) = \begin{bmatrix} \nabla f_1(x) & \nabla f_2(x) & \nabla f_3(x) \end{bmatrix} = \begin{bmatrix} x_2 & \cos(x_1 + x_2) & 2x_1 \\ x_1 & \cos(x_1 + x_2) & -2x_2 \end{bmatrix}$$

The Hessian

Given a function $f : \mathbb{R}^n \to \mathbb{R}$,

$$\frac{\partial^2 f}{\partial x_i \partial x_j},$$

to denote the i^{th} partial derivative of $\frac{\partial f}{\partial x_j}$.
We define the matrix $\nabla^2 f(x)$ to denote the matrix whose $(i,j)^{th}$ entry is given by $\frac{\partial^2 f}{\partial x_i \partial x_j}$ as the **Hessian** matrix of f, i.e.

$$Hf(x) = \nabla^2 f(x) = \begin{bmatrix} \frac{\partial^2 f(x)}{\partial x_1 \partial x_1} & \cdots & \frac{\partial^2 f(x)}{\partial x_1 \partial x_n} \\ \vdots & & \vdots \\ \frac{\partial^2 f(x)}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f(x)}{\partial x_n \partial x_n} \end{bmatrix}$$

Example

Find the Hessian matrix of the function $f : \mathbb{R}^3 \to \mathbb{R}$, given by
$f(x_1, x_2, x_3) = x_1^2 + x_1 x_3 + x_1 x_2 + x_3^2 + \exp(x_1 x_3)$

$$\nabla^2 f(x) = \begin{bmatrix} 2 + x_3^2 \exp(x_1 x_3) & 1 & 1 + (1 + x_1 x_3) \exp(x_1 x_3) \\ 1 & 0 & 0 \\ 1 + (1 + x_1 x_3) \exp(x_1 x_3) & 0 & 2 + x_1^2 \exp(x_1 x_3) \end{bmatrix}$$

Example

Find the Hessian matrix of the function $f : \mathbb{R}^3 \to \mathbb{R}$, given by $f(x_1, x_2, x_3) = x_1^2 + x_1 x_3 + x_1 x_2 + x_3^2 + \exp(x_1 x_3)$

$$\nabla^2 f(x) = \begin{bmatrix} 2 + x_3^2 \exp(x_1 x_3) & 1 & 1 + (1 + x_1 x_3)\exp(x_1 x_3) \\ 1 & 0 & 0 \\ 1 + (1 + x_1 x_3)\exp(x_1 x_3) & 0 & 2 + x_1^2 \exp(x_1 x_3) \end{bmatrix}$$

Line Segment & Convex Sets

Definition (Line Segment)

Given two points x and y both in $\mathbb{R}^n$, the set,

$$\left\{z \in \mathbb{R}^n \mid z = \lambda x + (1-\lambda)y,\ 0 \leq \lambda \leq 1\right\}.$$

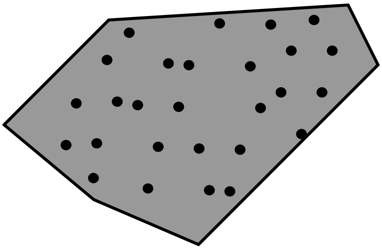
is called the line segment between x and y.

Definition (Convex Set)

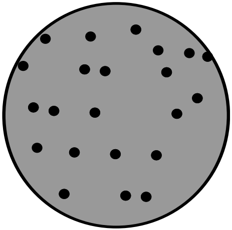
A subset S of $\mathbb{R}^n$ is called convex, if it contains the entire segment between any two of its points, i.e.

$$x \in S, y \in S, \text{then } \forall \lambda \in [0,1], \lambda x + (1-\lambda)y \in S$$

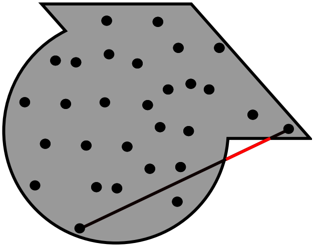
Examples



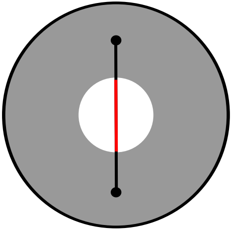
(a)



(b)

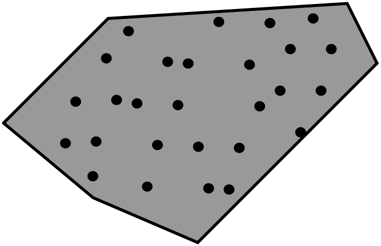


(c)

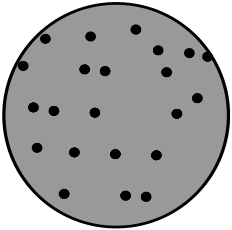


(d)

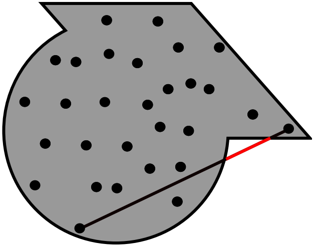
Examples



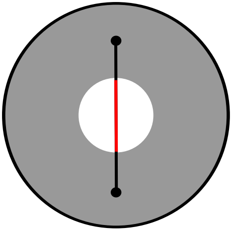
(a) Convex



(b) Convex



(c) Non-Convex



(d) Non-Convex

Convex Functions

Definition (Convex Function)

A function $f : X \to \mathbb{R}$ defined in a subset X of $\mathbb{R}^n$ and taking real values is called convex if:

- X is a convex set.
- For any $x, y \in X$, and every $0 \leq \lambda \leq 1$, the following holds:

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y)$$

If the inequality above is strict i.e.

$$f(\lambda x + (1-\lambda)y) < \lambda f(x) + (1-\lambda) f(y)$$

then f is called **strictly convex.**
Given a function f such that $-f$ is convex, is called **concave.**

Example

The function $f : \mathbb{R}^n \to \mathbb{R}$, defined as $f(x) = a^T x + b$ is convex. This follows from:

$$\begin{aligned} f(\lambda x + (1 - \lambda y)) &= a^T(\lambda x + (1 - \lambda)y) + b \\ &= \lambda a^T x + (1 - \lambda)a^T y + b \\ &= \lambda a^T x + (1 - \lambda)a^T y + \lambda b + (1 - \lambda)b \\ &= \lambda f(x) + (1 - \lambda)f(y). \end{aligned}$$

In fact, linear functions are both convex and concave (and they are the only functions with this property).

Test for convexity

This is a useful test to apply to see if a function is convex.

Suppose that C is a convex set, the function $f : C \to \mathbb{R}$ is twice continuously differentiable, then if the Hessian of f is positive semidefinite for all x in C then f is convex in C.

Optimality Conditions – Optimality Conditions

- Necessary conditions: a function f of a single variable x is said to have a minimum at a point x_0 if $f(x_0) > f(x)$ for all x. If x_0 is not a boundary point of an interval over which f is defined, then for x_0 to be a minimum it is necessary that:

$$\frac{df(x_0)}{dx} = 0$$

  This equation can be used to find a candidate for the minimum point of f

- Example: To find the minimum point of $x^2 + 12x$ we solve:

$$\frac{df}{dx} = 2x + 12 = 0$$

Optimality Conditions – Lagrange Multipliers

$$\min f(x)$$
$$\text{s.t } g_i(x) = 0, i = 1, \ldots, m$$

Suppose that f is convex and g_i is a linear function of x.

Lagrangian: $L(x, \lambda) = f(x) + \sum_{i=1}^{m} \lambda_i g_i(x)$ To solve the constrained problem we try to find a point that satisfies:

$$\frac{\partial f}{\partial x_i} + \sum_{j=1}^{m} \lambda_j \frac{\partial g_j}{\partial x_i} = 0 \qquad i = 1, \ldots, n$$

$$g_j(x) = 0 \qquad j = 1, \ldots, m$$

i.e. we need to solve $n + m$ equation with $n + m$ unknowns.

KKT Optimality Conditions - Linear Constraints

$$\begin{aligned} &\min f(x) \\ &Ax \leq b \end{aligned} \tag{0.1}$$

Assumptions:

- $f : \mathbb{R}^n \to (-\infty, +\infty]$ is differentiable
- $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^b$

Necessary optimality condition: If $x^\star$ is a local minimum of (0.1) then there exist $\lambda^\star \geq 0$ such that,

$$\nabla f(x^\star) + A^T \lambda^\star = 0$$

$$\lambda_i(a_i^T x - b_i) = 0 \ \text{ Complementarity Condition}$$

Equality constraints ($Ax = b$): then $\lambda \in \mathbb{R}^m$ (no need for multiplier to be positive) and Complementarity Condition is unnecessary.

Optimality Conditions - Convexity

Assumptions:

- $f : \mathbb{R}^n \to (-\infty, +\infty]$ is differentiable and **convex.**
- $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^b$

Necessary and sufficient optimality condition: $x^\star$, such that $Ax^\star \leq b$ is a global minimum of (0.1) if and only if then there exist $\lambda^\star \geq 0$ such that,

$$\nabla f(x^\star) + A^T \lambda^\star = 0$$

$$\lambda_i (a_i^T x - b_i) = 0 \text{ Complementarity Condition}$$

Equality constraints ($Ax = b$**)**: then $\lambda \in \mathbb{R}^m$ (no need for multiplier to be positive) and Complementarity Condition is unnecessary.

Example: Mean Variance Optimisation

The mean variance optimisation model attempts to capture the tradeoffs between risk and reward in investments.

- Proposed by H. Markowitz in 1952.
- Cornerstone of Modern Portfolio Theory
- Performance measured by expected returns
- Measures risk by variance of portfolio
- Shared with Miller & Sharpe the Nobel Memorial Prize in Economic Sciences (1990)

The Sveriges Riksbank Prize in Economic Sciences in Memory of Alfred Nobel 1990
Harry M. Markowitz, Merton H. Miller, William F. Sharpe

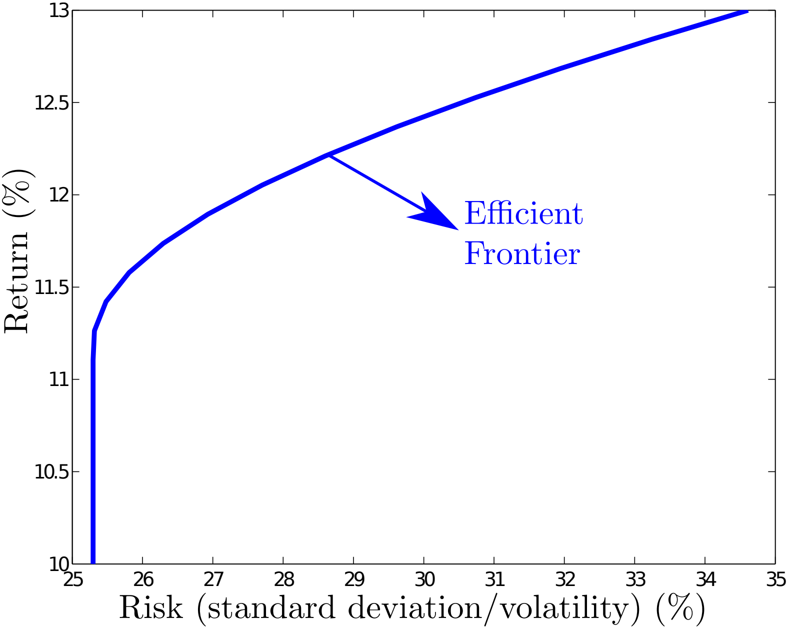
The Sveriges Riksbank Prize in Economic Sciences in Memory of Alfred Nobel 1990
Nobel Prize Award Ceremony
Harry M. Markowitz
Merton H. Miller
William F. Sharpe

Autobiography

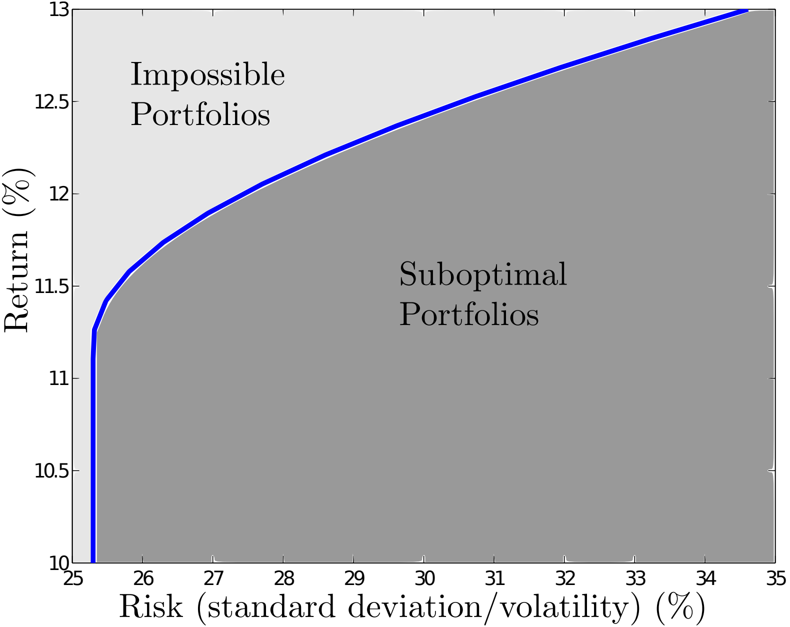
I was born in Chicago in 1927, the only child of Morris and Mildred Markowitz who owned a small grocery store. We lived in a nice apartment, always had enough to eat, and I had my own room. I never was aware of the Great Depression.

Growing up, I enjoyed baseball and tag football in the nearby empty lot or the park a few blocks away, and playing the violin in the high school orchestra. I also enjoyed reading. At first, my reading material consisted of comic books and adventure magazines, such as *The Shadow*, in addition to school assignments. In late grammar school and throughout high school I enjoyed popular accounts of physics and astronomy. In high school I also began to read original works of serious philosophers. I was particularly struck by David Hume's argument that, though we release a ball a thousand times, and each time, it falls to the floor, we do not have a necessary proof that it will fall the thousand-and-first time. I also read *The Origin of Species* and was moved by Darwin's marshalling of facts and careful consideration of possible objections.

Example: Mean Variance Efficient Frontier



Example: Mean Variance Efficient Frontier



The Markowitz Model I

- Markowitz formulated the problem to determine the efficient frontier as a mathematical optimization problem.
- Assume there are n risky assets with
  - mean returns $\bar{r}_1, \bar{r}_2, \ldots, \bar{r}_n$
  - covariances σ_{ij} for $i, j = 1, 2, \ldots, n$.
- The portfolio with weights $w_1, w_2, \ldots, w_n$ has
  - mean return $\bar{r}_{\mathrm{P}} = \sum_{i=1}^{n} w_i \bar{r}_i$
  - variance $\sigma_{\mathrm{P}}^2 = \sum_{i,j=1}^{n} w_i \sigma_{ij} w_j$.

The Markowitz Model II

$$
\begin{aligned}
\text{minimize} \quad & \frac{1}{2}\sum_{i,j=1}^{n} w_i \sigma_{ij} w_j && = \tfrac{1}{2}\sigma_{\mathrm{P}}^2 \\
\text{subject to} \quad & \sum_{i=1}^{n} w_i \bar{r}_i = \bar{r}_{\mathrm{P}} && = \text{ exp. return target} \\
& \sum_{i=1}^{n} w_i = 1 && = \text{ weights sum to } 1
\end{aligned}
$$

- In this formulation, short selling is allowed.
- The solution of the problem depends on the return target parameter $\bar{r}_{\mathrm{P}}$.
- The minimum-variance set is obtained by plotting the minimal σ_{P}^2 for different parameter values $\bar{r}_{\mathrm{P}}$.

Solution of the Markowitz Model I

$$
\begin{array}{llcc}
\text{minimize} & \displaystyle \frac{1}{2}\sum_{i,j=1}^{n} w_i \sigma_{ij} w_j & & \text{Lagrange multipliers:} \\
\text{subject to} & \displaystyle \sum_{i=1}^{n} w_i \bar{r}_i - \bar{r}_{\mathrm{P}} = 0 & \longleftarrow & \lambda \\
& \displaystyle \sum_{i=1}^{n} w_i - 1 = 0 & \longleftarrow & \mu
\end{array}
$$

The associated Lagrangian function L is given by

$$
L = \frac{1}{2}\sum_{i,j=1}^{n} w_i \sigma_{ij} w_j - \lambda\left(\sum_{i=1}^{n} w_i \bar{r}_i - \bar{r}_{\mathrm{P}}\right) - \mu\left(\sum_{i=1}^{n} w_i - 1\right) .
$$

Solution of the Markowitz Model II

Differentiate the Lagrangian w.r.t. $w_1, w_2, \ldots, w_n$, λ, and μ, and set all derivatives $= 0$:

$$
\begin{aligned}
w_i: &\qquad \sum_{j=1}^{n} \sigma_{ij} w_j - \lambda \bar{r}_i - \mu = 0 \qquad \text{for } i = 1, 2, \ldots, n \\
\lambda: &\qquad \sum_{i=1}^{n} w_i \bar{r}_i = \bar{r}_{\mathrm{P}} \\
\mu: &\qquad \sum_{i=1}^{n} w_i = 1
\end{aligned}
$$

$\Rightarrow n+2$ equations for $n+2$ unknowns $w_1, w_2, \ldots, w_n, \lambda, \mu$.

These equations characterize the efficient portfolios!

Vector Notation

Define

- $\boldsymbol{w} = (w_1, w_2, \ldots, w_n) \in \mathbb{R}^n$ vector of portfolio weights;
- $\bar{\boldsymbol{r}} = (\bar{r}_1, \bar{r}_2, \ldots, \bar{r}_n) \in \mathbb{R}^n$ vector of exp. asset returns;
- $\boldsymbol{e} = (1, 1, \ldots, 1) \in \mathbb{R}^n$ vector of 1's;
- $\mathbf{0} = (0, 0, \ldots, 0) \in \mathbb{R}^n$ vector of 0's;
- covariance matrix of asset returns

$$\Sigma = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1n} \\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{n1} & \sigma_{n2} & \cdots & \sigma_{nn} \end{pmatrix} \in \mathbb{R}^{n \times n}.$$

Markowitz Revisited

In vectorial notation, the Markowitz problem reads:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\boldsymbol{w}^\top \Sigma \boldsymbol{w} \\ \text{subject to} \quad & \boldsymbol{w}^\top \bar{\boldsymbol{r}} - \bar{r}_{\mathrm{P}} = 0 \\ & \boldsymbol{w}^\top \boldsymbol{e} - 1 = 0 \end{aligned}$$

The associated Lagrangian function can be rewritten as

$$L(\boldsymbol{w}, \lambda, \mu) = \frac{1}{2}\boldsymbol{w}^\top \Sigma \boldsymbol{w} - \lambda \left(\boldsymbol{w}^\top \bar{\boldsymbol{r}} - \bar{r}_{\mathrm{P}} \right) - \mu \left(\boldsymbol{w}^\top \boldsymbol{e} - 1 \right) ,$$

while the optimality conditions become

$$\Sigma \boldsymbol{w} - \lambda \bar{\boldsymbol{r}} - \mu \boldsymbol{e} = \boldsymbol{0}\,, \quad \bar{\boldsymbol{r}}^\top \boldsymbol{w} = \bar{r}_{\mathrm{P}} \quad \text{and} \quad \boldsymbol{e}^\top \boldsymbol{w} = 1\,.$$

Solution of Optimality Conditions

The optimality conditions

$$\Sigma \boldsymbol{w} - \lambda \bar{\boldsymbol{r}} - \mu \boldsymbol{e} = \boldsymbol{0}\,, \quad \bar{\boldsymbol{r}}^\top \boldsymbol{w} = \bar{r}_\mathrm{P} \quad \text{and} \quad \boldsymbol{e}^\top \boldsymbol{w} = 1$$

can be written as one vectorial equation

$$\begin{pmatrix} \Sigma & -\bar{\boldsymbol{r}} & -\boldsymbol{e} \\ -\bar{\boldsymbol{r}}^\top & 0 & 0 \\ -\boldsymbol{e}^\top & 0 & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{w} \\ \lambda \\ \mu \end{pmatrix} = \begin{pmatrix} \boldsymbol{0} \\ -\bar{r}_\mathrm{P} \\ -1 \end{pmatrix} .$$

This is solvable if Σ has full rank and $\bar{\boldsymbol{r}}$ is not a multiple of $\boldsymbol{e}$.

$$\Rightarrow \quad \begin{pmatrix} \boldsymbol{w} \\ \lambda \\ \mu \end{pmatrix} = \begin{pmatrix} \Sigma & -\bar{\boldsymbol{r}} & -\boldsymbol{e} \\ -\bar{\boldsymbol{r}}^\top & 0 & 0 \\ -\boldsymbol{e}^\top & 0 & 0 \end{pmatrix}^{-1} \begin{pmatrix} \boldsymbol{0} \\ -\bar{r}_\mathrm{P} \\ -1 \end{pmatrix} .$$

Markowitz Model w/o Short Selling

$$
\begin{aligned}
\text{minimize} \quad & \frac{1}{2}\sum_{i,j=1}^{n} w_i \sigma_{ij} w_j \\
\text{subject to} \quad & \sum_{i=1}^{n} w_i \bar{r}_i = \bar{r}_{\mathrm{P}} \\
& \sum_{i=1}^{n} w_i = 1 \\
& w_i \geq 0 \quad \text{for} \quad i = 1, 2, \ldots, n
\end{aligned}
$$

- This problem cannot be reduced to the solution of a set of linear equations. It is termed a quadratic program.
- Such problems can be solved via special algorithms (quadratic programming solvers, e.g. interior point methods)

One Dimensional Newton's Method

$$\min\ f(x)$$

1. Minimising a general non-linear function is difficult
2. Basic idea: minimise a quadratic approximation

$$\min q(x)$$

   where $q(x) = f(x_k) + f'(x_k)(x - x_k) + \frac{1}{2}f''(x_k)(x - x_k)^2$
3. Minimise quadratic approximation

$$0 = q'(x) = f'(x_k) + f''(x_k)(x - x_k)$$

4. Use approximate minimiser as new starting point

$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}$$

Additional material: Chapter 7, in *An Introduction to Optimization*, Chong & Zhak, Third Edition.

One Dimensional Newton's Method

$$\min\ f(x)$$

1. Minimising a general non-linear function is difficult
2. Basic idea: minimise a quadratic approximation

$$\min q(x)$$

where $q(x) = f(x_k) + f'(x_k)(x - x_k) + \frac{1}{2}f''(x_k)(x - x_k)^2$

3. Minimise quadratic approximation

$$0 = q'(x) = f'(x_k) + f''(x_k)(x - x_k)$$

4. Use approximate minimiser as new starting point

$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}$$

Additional material: Chapter 7, in *An Introduction to Optimization*, Chong & Zhak, Third Edition.

One Dimensional Newton's Method

$$\min\ f(x)$$

1. Minimising a general non-linear function is difficult
2. Basic idea: minimise a quadratic approximation

$$\min q(x)$$

   where $q(x) = f(x_k) + f'(x_k)(x - x_k) + \frac{1}{2}f''(x_k)(x - x_k)^2$
3. Minimise quadratic approximation

$$0 = q'(x) = f'(x_k) + f''(x_k)(x - x_k)$$

4. Use approximate minimiser as new starting point

$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}$$

Additional material: Chapter 7, in *An Introduction to Optimization*, Chong & Zhak, Third Edition.

One Dimensional Newton's Method

$$\min\ f(x)$$

1. Minimising a general non-linear function is difficult
2. Basic idea: minimise a quadratic approximation

$$\min q(x)$$

   where $q(x) = f(x_k) + f'(x_k)(x - x_k) + \frac{1}{2}f''(x_k)(x - x_k)^2$
3. Minimise quadratic approximation

$$0 = q'(x) = f'(x_k) + f''(x_k)(x - x_k)$$

4. Use approximate minimiser as new starting point

$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}$$

Additional material: Chapter 7, in *An Introduction to Optimization*, Chong & Zhak, Third Edition.

Example: One Dimensional Newton's Method

Example

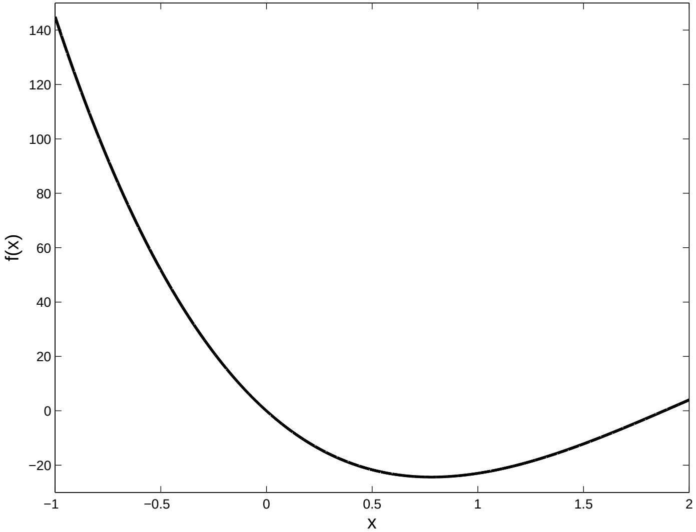
Use Newton's Method to find a minimiser of,

$$f(x) = x^4 - 14x^3 + 60x^2 - 70x.$$

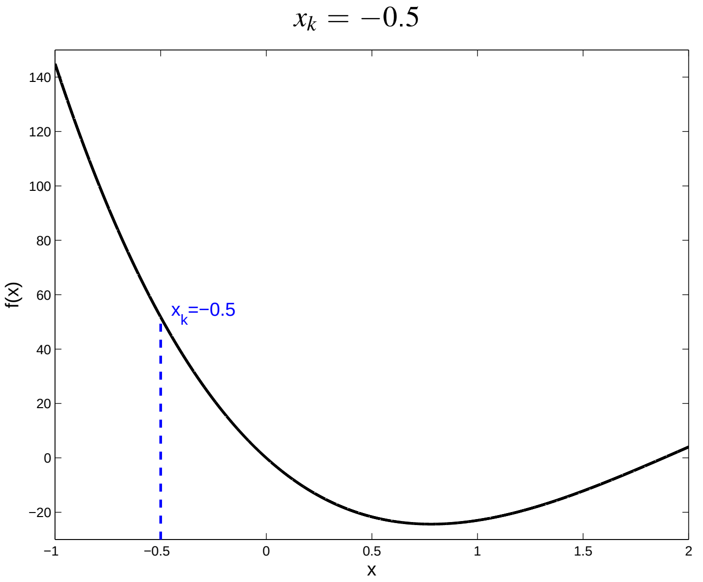
Start at $x(0) = -0.5$

Example: One Dimensional Newton's Method

$$\min f(x) = x^4 - 14x^3 + 60x^2 - 70x$$

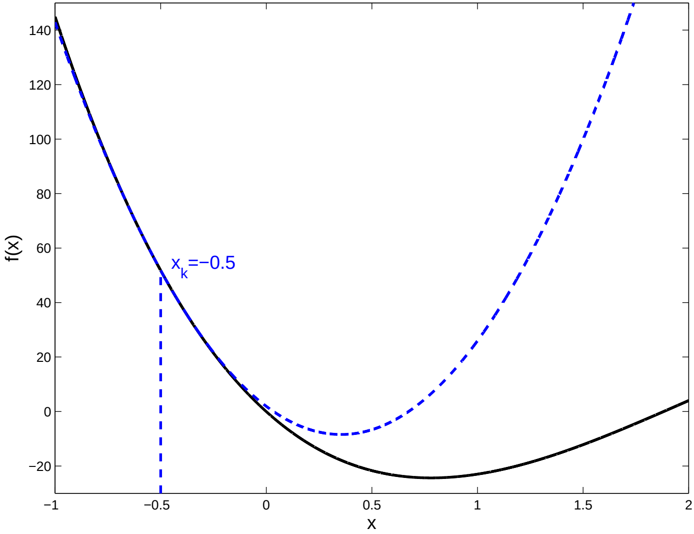


Example: One Dimensional Newton's Method



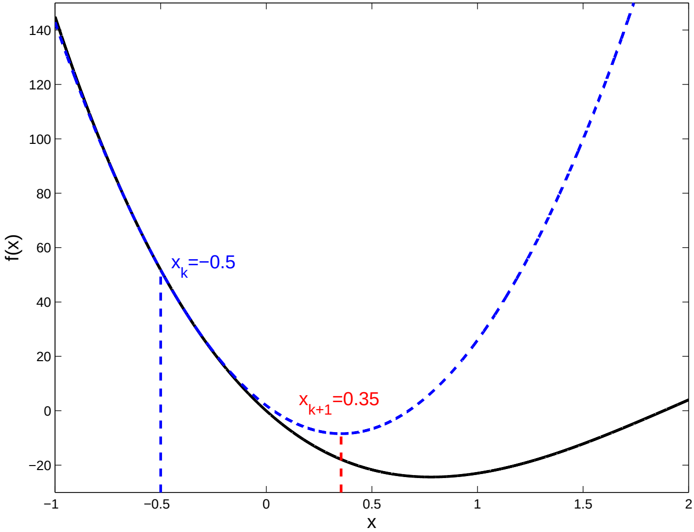
Example: One Dimensional Newton's Method

$$q(x) = f(-0.5) + f'(-0.5)(x+0.5) + \frac{1}{2}f''(-0.5)(x+0.5)^2$$



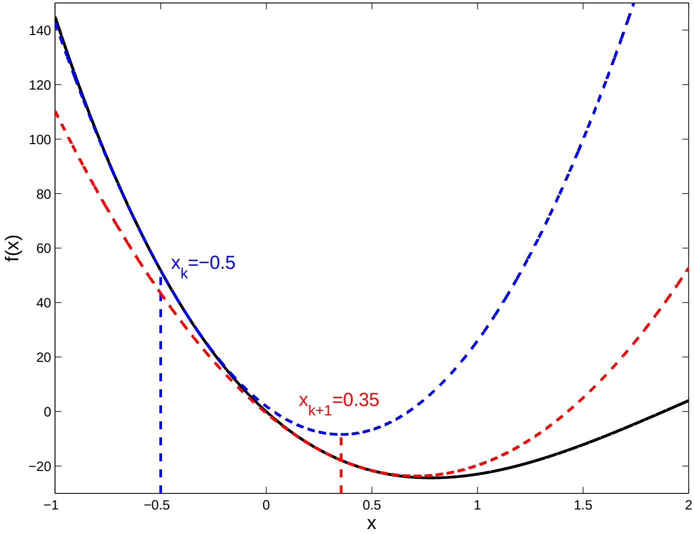
Example: One Dimensional Newton's Method

$$x_{k+1} = \arg\min\ f(-0.5) + f'(-0.5)(x+0.5) + \tfrac{1}{2} f''(-0.5)(x+0.5)^2$$



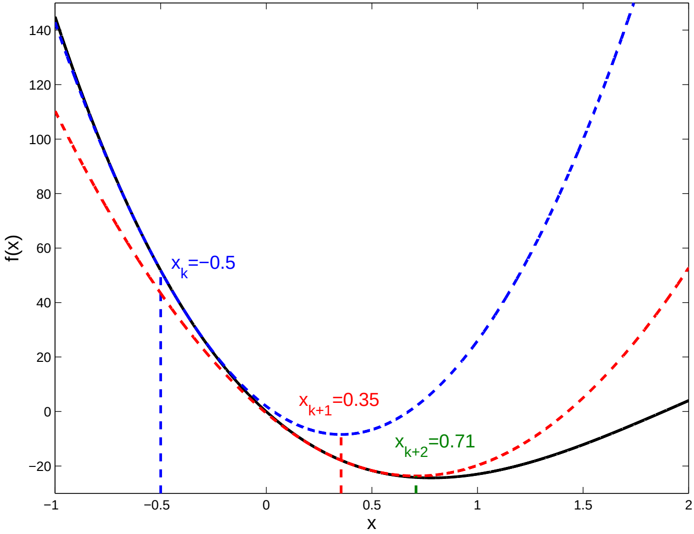
Example: One Dimensional Newton's Method

$$q(x) = f(0.35) + f'(0.35)(x - 0.35) + \frac{1}{2}f''(0.35)(x - 0.35)^2$$



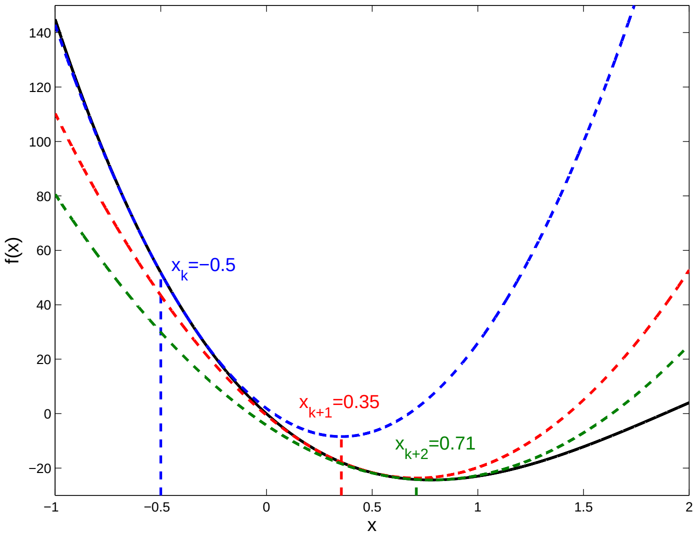
Example: One Dimensional Newton's Method

$$x_{k+2} = \arg\min\ f(0.35) + f'(0.35)(x - 0.35) + \frac{1}{2}f''(0.35)(x - 0.35)^2$$



Example: One Dimensional Newton's Method

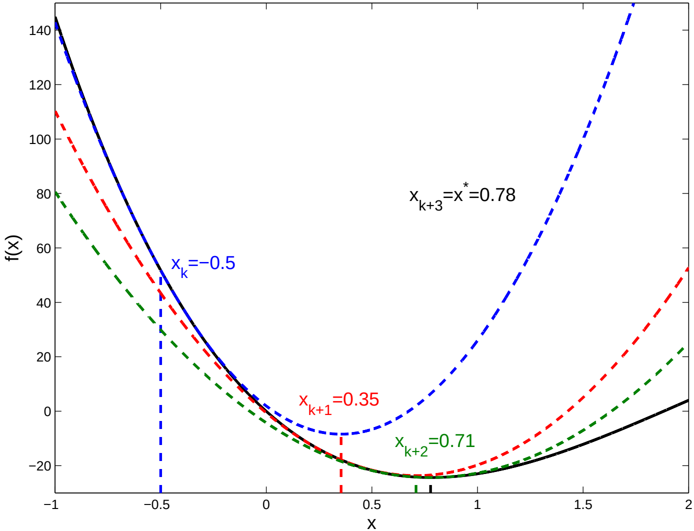
$$q(x) = f(0.71) + f'(0.71)(x - 0.71) + \frac{1}{2}f''(0.71)(x - 0.71)^2$$



Example: One Dimensional Newton's Method

$$x_{k+3} = \arg\min\ f(0.71) + f'(0.71)(x - 0.71) + \frac{1}{2}f''(0.71)(x - 0.71)^2$$

Convergence after 3 iterations!



Newton's Method for computing Roots of Equations

- Newton's method can also be seen as a way to solve for

$$f'(x) = 0$$

using the iterative procedure,

$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}$$

- But if we set $g(x) = f'(x)$ then we obtain an algorithm for solving for $g(x) = 0$

$$x_{k+1} = x_k - \frac{g(x_k)}{g'(x_k)}$$

Newton's Method for computing Roots of Equations

- Newton's method can also be seen as a way to solve for

$$f'(x) = 0$$

  using the iterative procedure,

$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}$$

- But if we set $g(x) = f'(x)$ then we obtain an algorithm for solving for $g(x) = 0$

$$x_{k+1} = x_k - \frac{g(x_k)}{g'(x_k)}$$

Example: Roots of Equations

Example

Use Newton's Method to find a root of,

$$f(x) = x^3 - 12.2x^2 + 7.45x + 42 = 0$$

Start at $x(0) = 12$. Perform two iterations.

Answer

$$x_1 = 12 - \frac{102.6}{146.65} = 11.33$$

$$x_2 = 11.33 - \frac{14.73}{116.11} = 11.21$$

Example: Roots of Equations

Example

Use Newton's Method to find a root of,

$$f(x) = x^3 - 12.2x^2 + 7.45x + 42 = 0$$

Start at $x(0) = 12$. Perform two iterations.

Answer

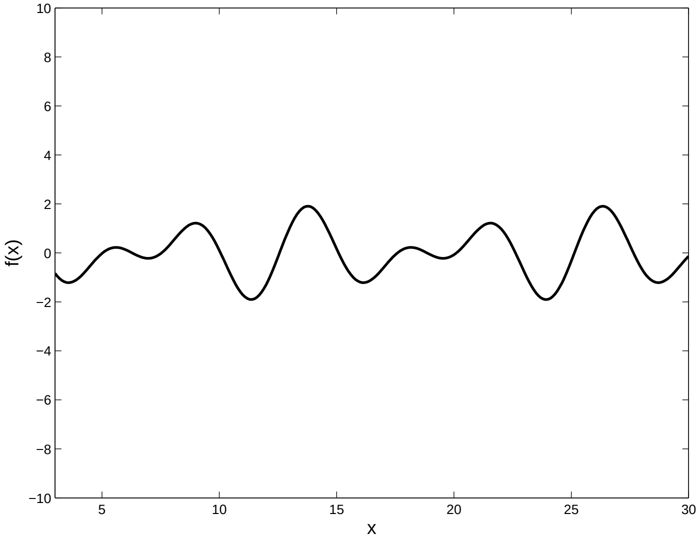
$$x_1 = 12 - \frac{102.6}{146.65} = 11.33$$
$$x_2 = 11.33 - \frac{14.73}{116.11} = 11.21$$

Failure to Converge

- The algorithm can fail to converge if $f''(x) < 0$
- Algorithm may find a point that satisfies the first order condition, not necessarily a minimiser
- Algorithm may cycle

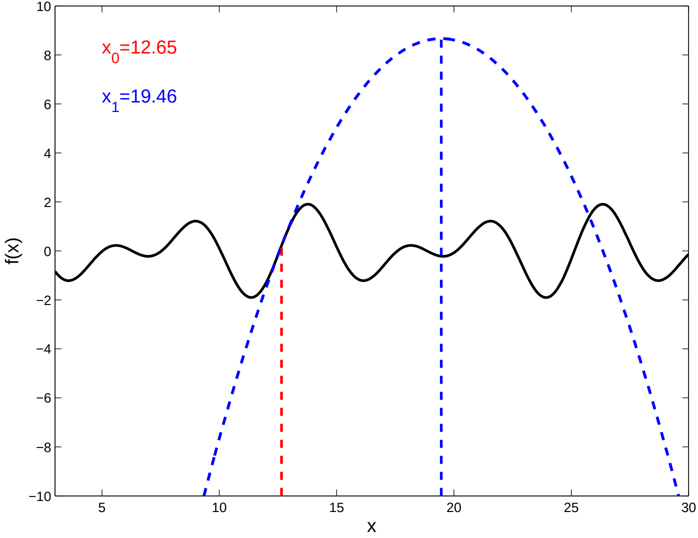
Example: Convergence Problems

$\min \; \sin(x) + \sin(3x/2)$ Initial Point $x_0 = 12.65$



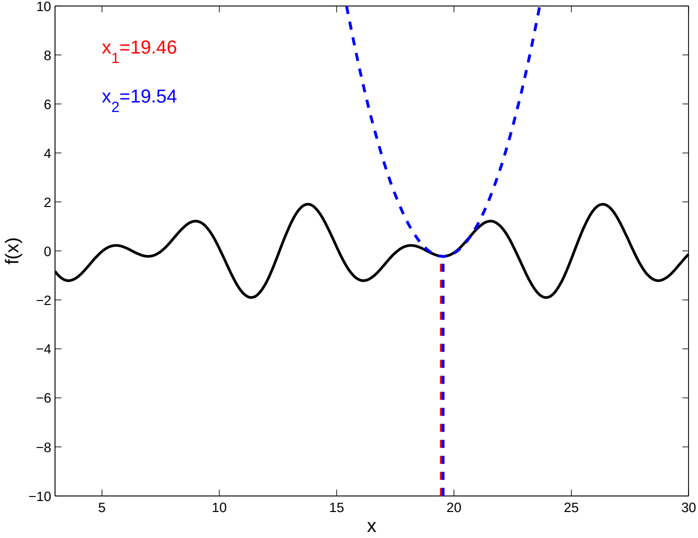
Example: Convergence Problems

$$x_0 = 12.65,\ f''(x_0) < 0,\ x_1 = 19.46$$



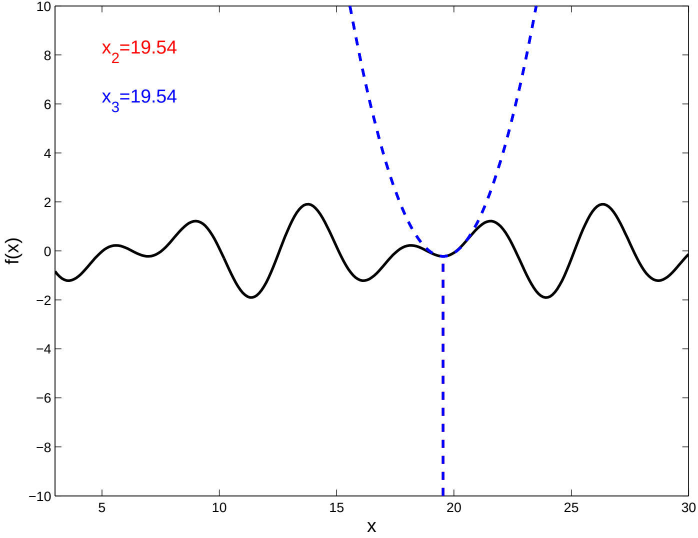
Example: Convergence Problems

$$x_1 = 19.46,\ f''(x_1) > 0,\ x_2 = 19.54$$



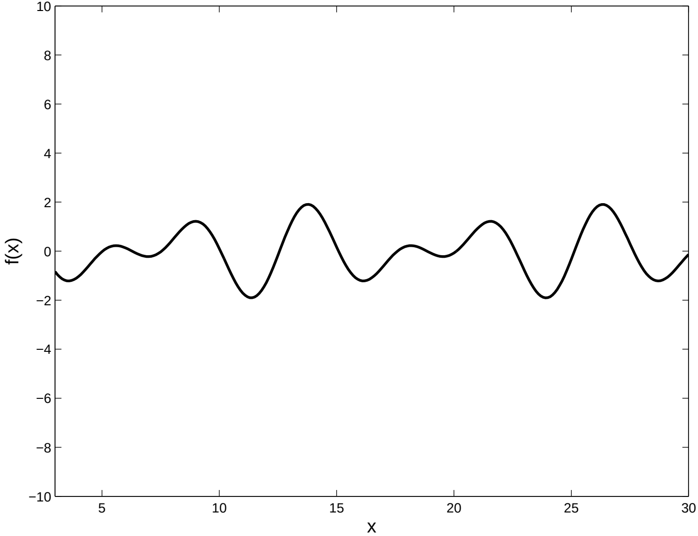
Example: Convergence Problems

$$x_2 = 19.54 \; x_3 = 19.54, \; f''(x_3) > 0, f'(x_3) \approx 0$$



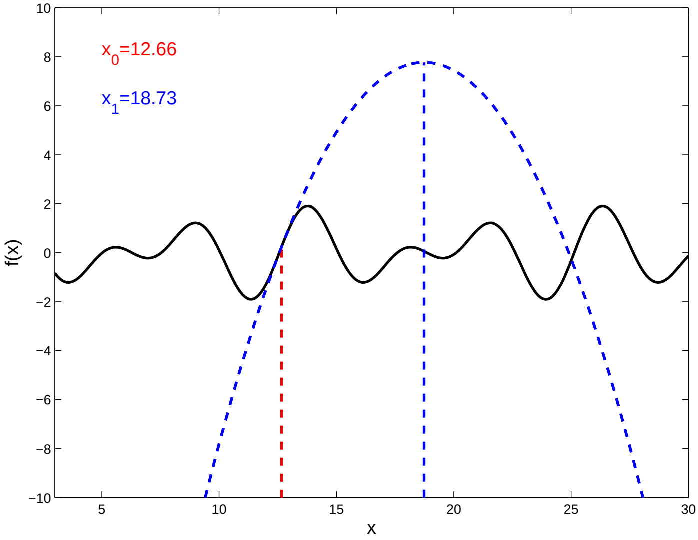
Example: Convergence Problems

$$\min \; \sin(x) + \sin(3x/2)$$ Initial Point $x_0 = \cancel{12.65}$ 12.66



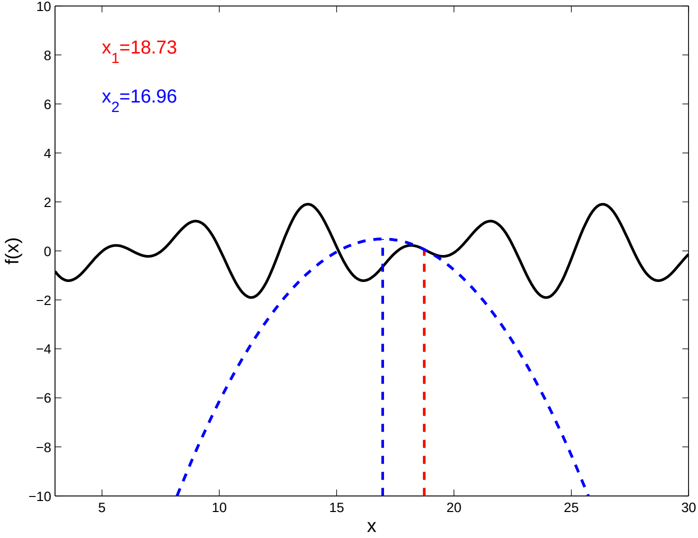
Example: Convergence Problems

$x_0 = 12.66$, $f''(x_0) < 0$, $x_1 = 18.73$



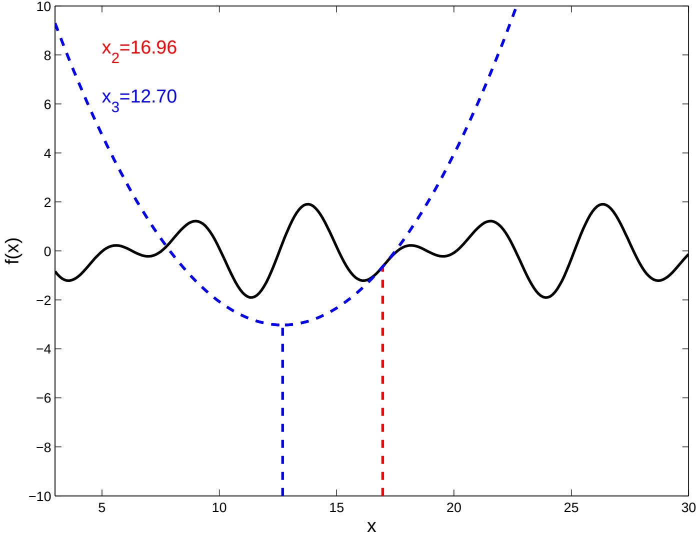
Example: Convergence Problems

$$x_1 = 18.73,\ f''(x_1) < 0,\ x_2 = 16.96$$



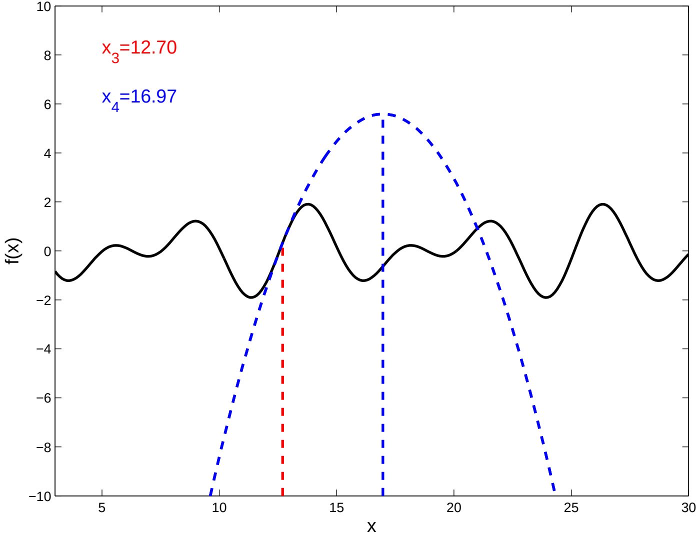
Example: Convergence Problems

$$x_2 = 16.96,\ f''(x_2) > 0,\ x_3 = 12.70$$



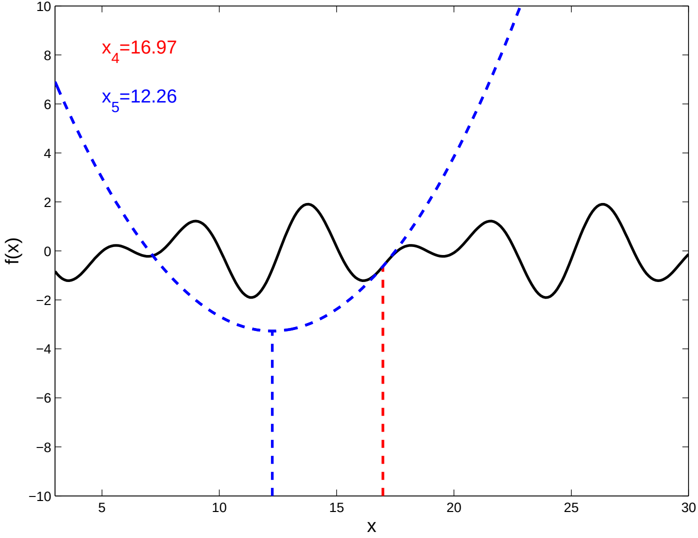
Example: Convergence Problems

$$x_3 = 12.70,\ f''(x_3) < 0,\ x_4 = 16.97$$



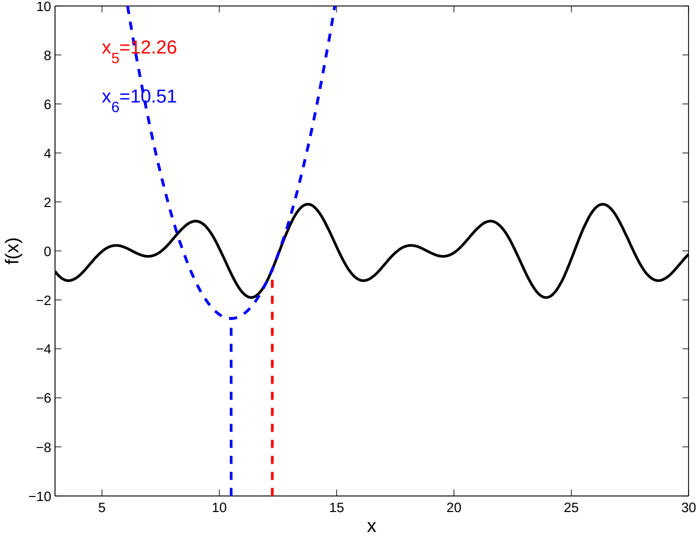
Example: Convergence Problems

$$x_4 = 16.97,\ f''(x_4) > 0,\ x_5 = 12.26$$



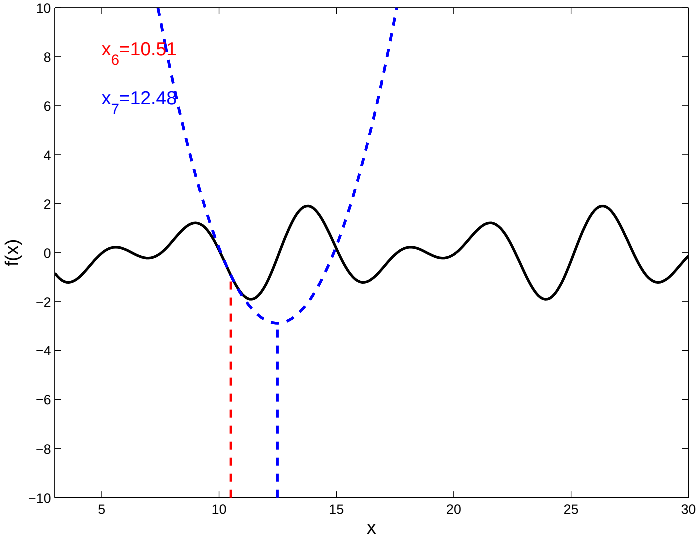
Example: Convergence Problems

$$x_5 = 12.26,\ f''(x_5) > 0,\ x_6 = 10.51$$



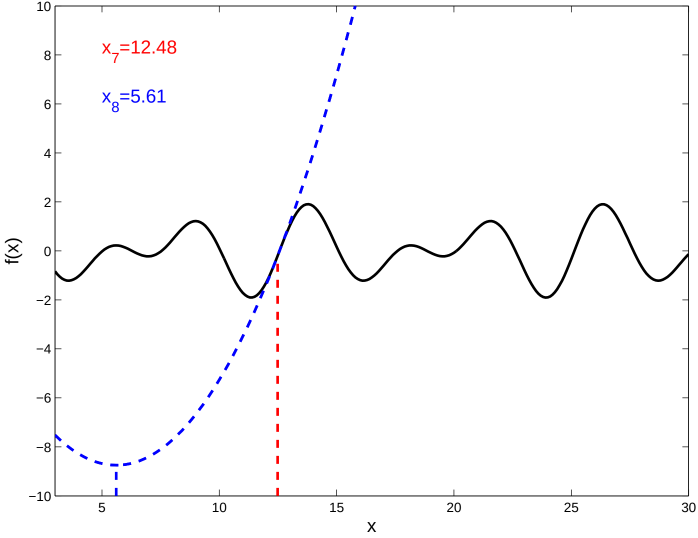
Example: Convergence Problems

$x_6 = 10.51$, $f''(x_6) > 0$, $x_7 = 12.48$



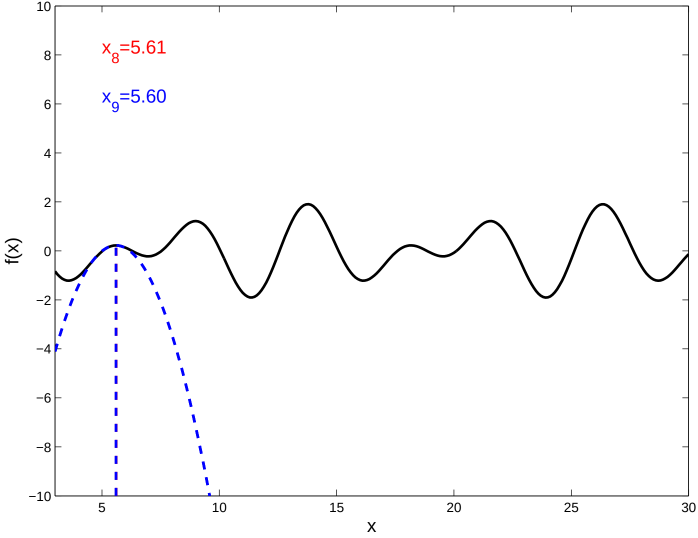
Example: Convergence Problems

$$x_7 = 12.48,\ f''(x_7) > 0,\ x_8 = 5.61$$



Example: Convergence Problems

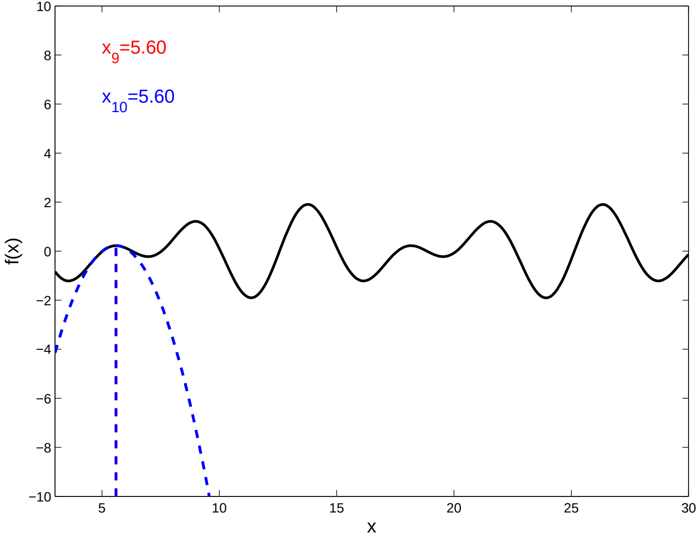
$$x_8 = 5.61,\ f''(x_8) < 0,\ x_9 = 5.60$$



Example: Convergence Problems

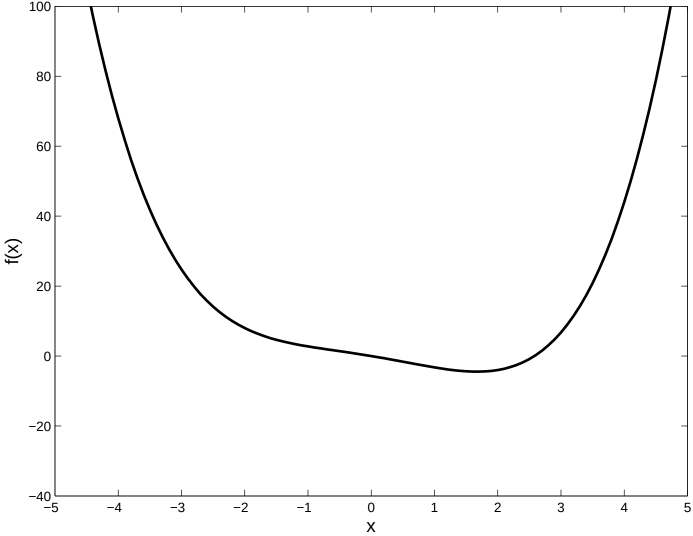
$$x_9 = 5.60, x_{10} = 5.60, f'(x_{10}) \approx 0, f''(x_{10}) < 0$$

Convergence to a local maximum!



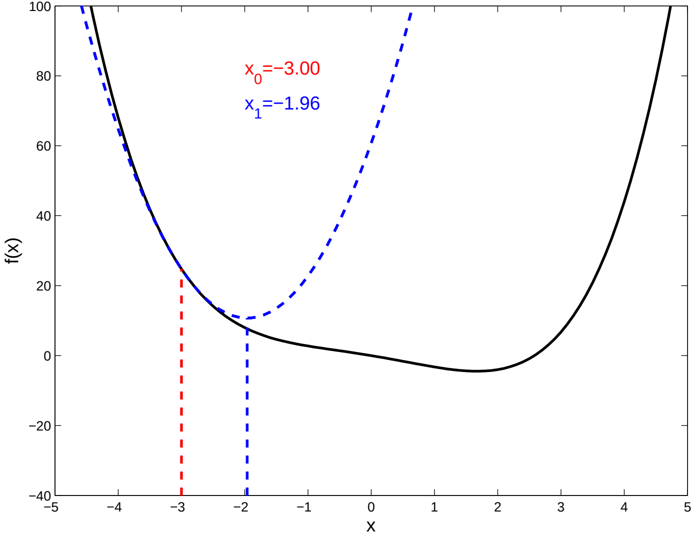
Example: Convergence Problems

$$\min \; \frac{1}{4}x^4 - \frac{1}{2}x^2 - 3x \; \text{Initial Point } x_0 = -3.0$$



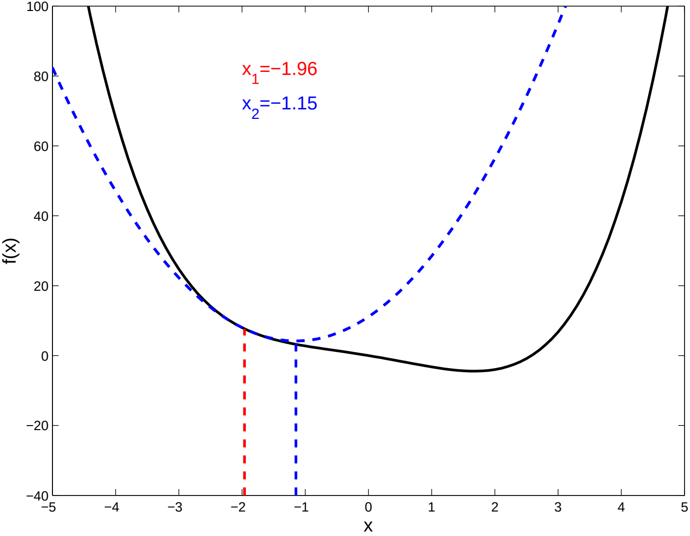
Example: Convergence Problems

$$x_0 = -3, \; x_1 = -1.96$$

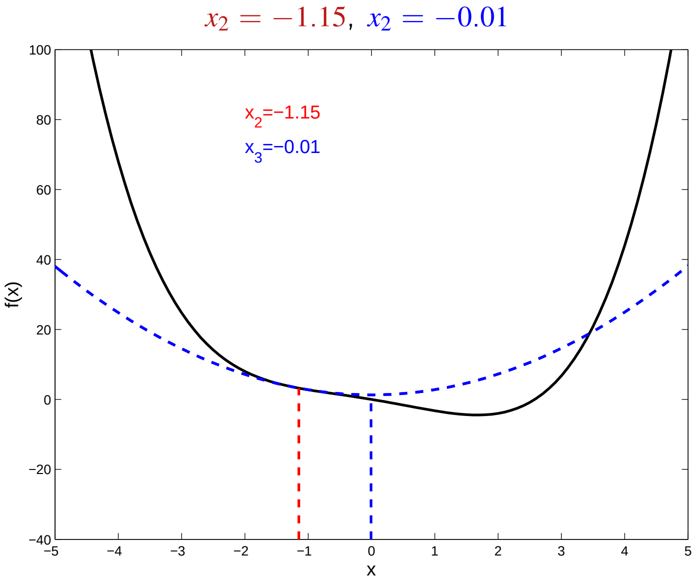


Example: Convergence Problems

$x_1 = -1.96$, $x_2 = -1.15$



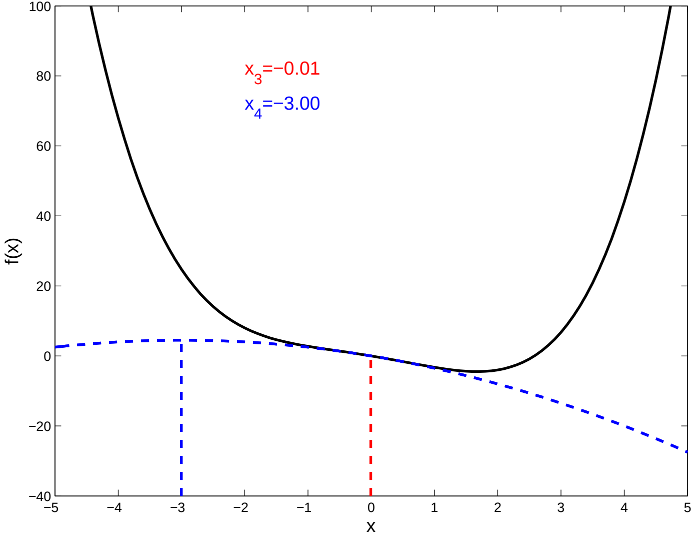
Example: Convergence Problems



Example: Convergence Problems

$x_3 = -.01, \; x_4 = -3.00 = x_0$

The algorithm returns to the initial point!



Towards a general Newton-Raphson Method

Issues with the Newton-Raphson Method we studied so far,

(a) Only applicable to single dimension

(b) The algorithm may cycle

(c) It may fail to find a descent direction

(d) It may converge to a saddle point or a local maximum

In this lecture:

(a) Multivariate extension

(b) Discuss conditions & modifications for guaranteed convergence

(c) Discuss convergence rates & practical implementation

Towards a general Newton-Raphson Method

Issues with the Newton-Raphson Method we studied so far,

(a) Only applicable to single dimension

(b) The algorithm may cycle

(c) It may fail to find a descent direction

(d) It may converge to a saddle point or a local maximum

In this lecture:

(a) Multivariate extension

(b) Discuss conditions & modifications for guaranteed convergence

(c) Discuss convergence rates & practical implementation

Multivariate Newton-Raphson Method

General problem,

$$\min_{\boldsymbol{x} \in \mathbb{R}^n} f(\boldsymbol{x})$$

1. As in 1-d case we construct a **quadratic** approximation around the current iterate x_k (second order Taylor series expansion)

$$\begin{aligned} f(\boldsymbol{x}) &\approx f(\boldsymbol{x}_k) + \nabla \boldsymbol{f}(\boldsymbol{x}_k)^T(\boldsymbol{x} - \boldsymbol{x}_k) + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}_k)^T \nabla^2 \boldsymbol{f}(\boldsymbol{x}_k)(\boldsymbol{x} - \boldsymbol{x}_k) \\ &\triangleq q(\boldsymbol{x}) \end{aligned}$$

2. Apply the FONC to $q(\boldsymbol{x})$,

$$0 = \nabla \boldsymbol{q}(\boldsymbol{x}) = \nabla \boldsymbol{f}(\boldsymbol{x}_k) + \nabla^2 \boldsymbol{f}(\boldsymbol{x}_k)(\boldsymbol{x} - \boldsymbol{x}_k)$$

3. Assume that $\nabla^2 f(\boldsymbol{x}_k) \succ 0$ (i.e. positive definite), then

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \nabla^2 \boldsymbol{f}(\boldsymbol{x}_k)^{-1} \nabla \boldsymbol{f}(\boldsymbol{x}_k)$$

Multivariate Newton-Raphson Method

General problem,

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x})$$

1. As in 1-d case we construct a **quadratic** approximation around the current iterate x_k (second order Taylor series expansion)

$$\begin{aligned} f(\boldsymbol{x}) &\approx f(\boldsymbol{x}_k) + \nabla \boldsymbol{f}(\boldsymbol{x}_k)^T(\boldsymbol{x}-\boldsymbol{x}_k) + \frac{1}{2}(\boldsymbol{x}-\boldsymbol{x}_k)^T \nabla^2 \boldsymbol{f}(\boldsymbol{x}_k)(\boldsymbol{x}-\boldsymbol{x}_k) \\ &\triangleq q(\boldsymbol{x}) \end{aligned}$$

2. Apply the FONC to $q(\boldsymbol{x})$,

$$0 = \nabla \boldsymbol{q}(\boldsymbol{x}) = \nabla \boldsymbol{f}(\boldsymbol{x}_k) + \nabla^2 \boldsymbol{f}(\boldsymbol{x}_k)(\boldsymbol{x}-\boldsymbol{x}_k)$$

3. Assume that $\nabla^2 f(\boldsymbol{x}_k) \succ 0$ (i.e. positive definite), then

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \nabla^2 \boldsymbol{f}(\boldsymbol{x}_k)^{-1} \nabla \boldsymbol{f}(\boldsymbol{x}_k)$$

Why is the assumption $\nabla^2 f(x_k) \succ 0$ needed?

Why is the assumption $\nabla^2 f(x_k) \succ 0$ needed?

If $\nabla^2 \boldsymbol{f}(\boldsymbol{x}_k)$ is positive definite then the Newton direction

$$\boldsymbol{d}_k = -\nabla^2 \boldsymbol{f}(\boldsymbol{x}_k)^{-1} \nabla \boldsymbol{f}(\boldsymbol{x}_k)$$

is a descent direction,

$$\nabla \boldsymbol{f}(\boldsymbol{x}_k)^T \boldsymbol{d}_k = -\nabla \boldsymbol{f}(\boldsymbol{x}_k)^T \nabla^2 \boldsymbol{f}(\boldsymbol{x}_k)^{-1} \nabla \boldsymbol{f}(\boldsymbol{x}_k) < 0.$$

Remark: Note that if a matrix is positive definite then so is its inverse, see for example any Linear algebra book e.g. *Matrix Analysis*, R.A. Horn, C.R. Johnson

Convergence Theory

Theorem

Suppose that f is three times continuously differentiable and that $\boldsymbol{x}^* \in \mathbb{R}^n$ satisfies,

$$\nabla \boldsymbol{f}(\boldsymbol{x}^*) = 0$$

and that $\nabla^2 \boldsymbol{f}(\boldsymbol{x}^*)$ is invertible. Then for all $\boldsymbol{x}_0$ (starting point) sufficiently close to $\boldsymbol{x}^*$ the following holds,

(1) *Newton's method is* **well defined** *for all k.*

(2) *The method* **converges to $\boldsymbol{x}^*$**.

(3) *The order of* **convergence is quadratic**.

Remarks:

(a) Conditions $\nabla \boldsymbol{f}(\boldsymbol{x}^*) = 0$ & $\nabla^2 \boldsymbol{f}(\boldsymbol{x}^*)$ invertible hold for *local maxima as well*. **The theorem does not say the method will converge to a minimum.**

(b) The starting point needs to be close to the solution

Convergence Theory

Theorem

Suppose that f is three times continuously differentiable and that $\boldsymbol{x}^* \in \mathbb{R}^n$ satisfies,

$$\nabla \boldsymbol{f}(\boldsymbol{x}^*) = 0$$

and that $\nabla^2 \boldsymbol{f}(\boldsymbol{x}^*)$ is invertible. Then for all $\boldsymbol{x}_0$ (starting point) sufficiently close to $\boldsymbol{x}^*$ the following holds,

(1) *Newton's method is* **well defined** *for all k.*

(2) *The method* **converges to $\boldsymbol{x}^*$**.

(3) *The order of* **convergence is quadratic**.

Remarks:

(a) Conditions $\nabla \boldsymbol{f}(\boldsymbol{x}^*) = 0$ & $\nabla^2 \boldsymbol{f}(\boldsymbol{x}^*)$ invertible hold for *local maxima as well*. **The theorem does not say the method will converge to a minimum.**

(b) The starting point needs to be close to the solution

Is the Newton algorithm a descent algorithm?

1. Given a point $\boldsymbol{x}_k$.
2. Derive a descent direction $\boldsymbol{d}_k \in \mathbb{R}^n$, i.e.

$$\nabla \boldsymbol{f}(\boldsymbol{x}_k)^T \boldsymbol{d}_k < 0.$$

3. Decide on a step-size α_k.
4. Transition to the next point,

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k$$

Is the Newton algorithm a descent algorithm?

Theorem

Suppose that $\{x_k\}$ *is a sequence generated by the algorithm. If the Hessian* $\nabla^2 f(x^k) \succ 0$ *and* $\nabla f(x^k) \neq 0$ *then the search direction*

$$d_k = -\nabla^2 f(x_k)^{-1} \nabla f(x_k) = x_{k+1} - x_k$$

is a descent direction for f *in the sense that there exists an* $\alpha \in (0, \bar{\alpha})$ *such that for all* $\alpha \in (0, \bar{\alpha})$,

$$f(x_k + \alpha d_k) < f(x_k).$$

The Newton algorithm is a descent algorithm with a descent direction given by

$$-\nabla^2 \mathbf{f}(\mathbf{x})^{-1} \nabla \mathbf{f}(\mathbf{x})$$

Is the Newton algorithm a descent algorithm?

Theorem

Suppose that $\{\boldsymbol{x}_k\}$ is a sequence generated by the algorithm. If the Hessian $\nabla^2 \boldsymbol{f}(\boldsymbol{x}^k) \succ 0$ and $\nabla f(\boldsymbol{x}^k) \neq 0$ then the search direction

$$\boldsymbol{d}_k = -\nabla^2 \boldsymbol{f}(\boldsymbol{x}_k)^{-1} \nabla \boldsymbol{f}(\boldsymbol{x}_k) = \boldsymbol{x}_{k+1} - \boldsymbol{x}_k$$

is a descent direction for f in the sense that there exists an $\alpha \in (0, \bar{\alpha})$ such that for all $\alpha \in (0, \bar{\alpha})$,

$$f(\boldsymbol{x}_k + \alpha \boldsymbol{d}_k) < f(\boldsymbol{x}_k).$$

The Newton algorithm is a descent algorithm with a descent direction given by

$$-\nabla^2 \mathbf{f}(\mathbf{x})^{-1} \nabla \mathbf{f}(\mathbf{x})$$

Line Search & Backtracking

Exact line search: The result in previous slide motivates the modification of the Newton method,

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k \nabla^2 \boldsymbol{f}(\boldsymbol{x}_k)^{-1} \nabla \boldsymbol{f}(\boldsymbol{x}_k)$$

where $\alpha_k = \arg\min_{\alpha \geq 0} f(\boldsymbol{x}_k - \alpha \nabla^2 \boldsymbol{f}(\boldsymbol{x}_k)^{-1} \nabla \boldsymbol{f}(\boldsymbol{x}_k))$ (exact line search)
Other types of line search algorithms are also used.
Backtracking:

```
while
```

Do not have sufficient decrease in objective function value

```
do
```

Reduce step size

Line Search & Backtracking

Exact line search: The result in previous slide motivates the modification of the Newton method,

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k \nabla^2 \boldsymbol{f}(\boldsymbol{x}_k)^{-1} \nabla \boldsymbol{f}(\boldsymbol{x}_k)$$

where $\alpha_k = \arg\min_{\alpha \geq 0} f(\boldsymbol{x}_k - \alpha \nabla^2 \boldsymbol{f}(\boldsymbol{x}_k)^{-1} \nabla \boldsymbol{f}(\boldsymbol{x}_k))$ (exact line search)
Other types of line search algorithms are also used.
Backtracking: Given two constants $0 < \beta < 0.5$, and $0 < \gamma < 1$ and a descent direction d then

```
while
```

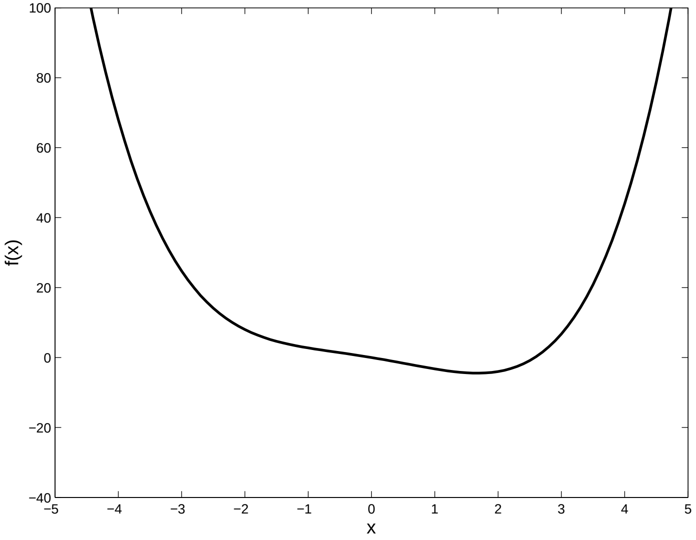
$$f(\boldsymbol{x} + \alpha \boldsymbol{d}) > f(\boldsymbol{x}) + \alpha \beta \nabla \boldsymbol{f}(\boldsymbol{x})^T \boldsymbol{d}$$

```
do
```

$$\alpha \leftarrow \gamma \alpha$$

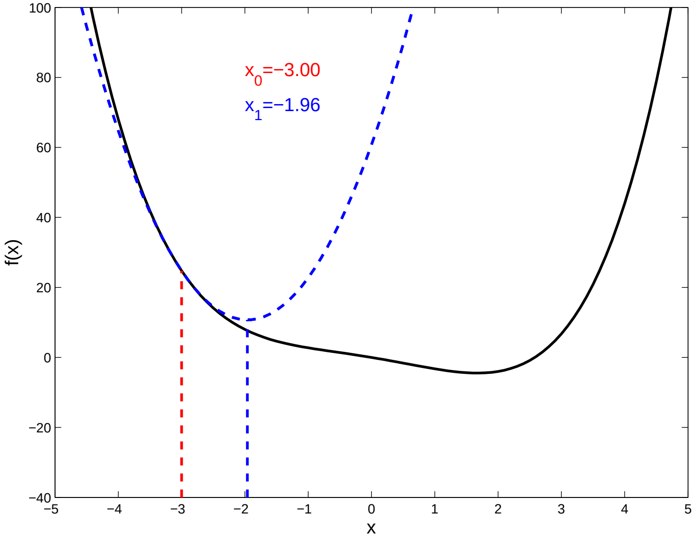
Example: Convergence Problems

$$\min \ \tfrac{1}{4}x^4 - \tfrac{1}{2}x^2 - 3x$$ Initial Point $x_0 = -3.0$

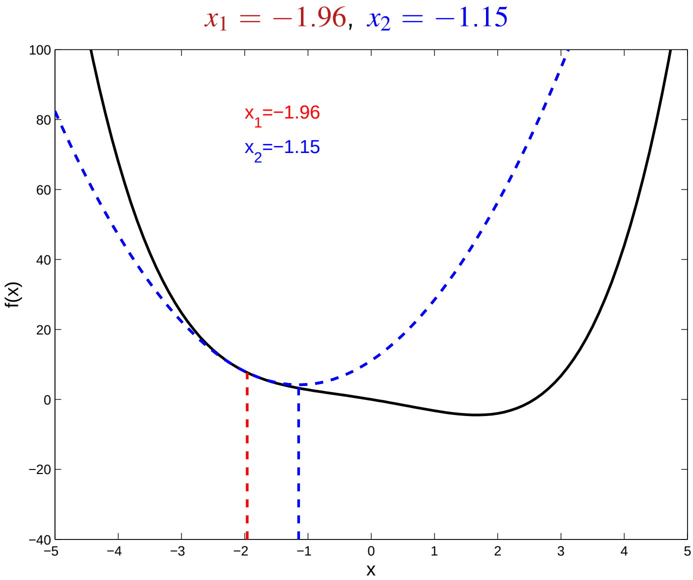


Example: Convergence Problems

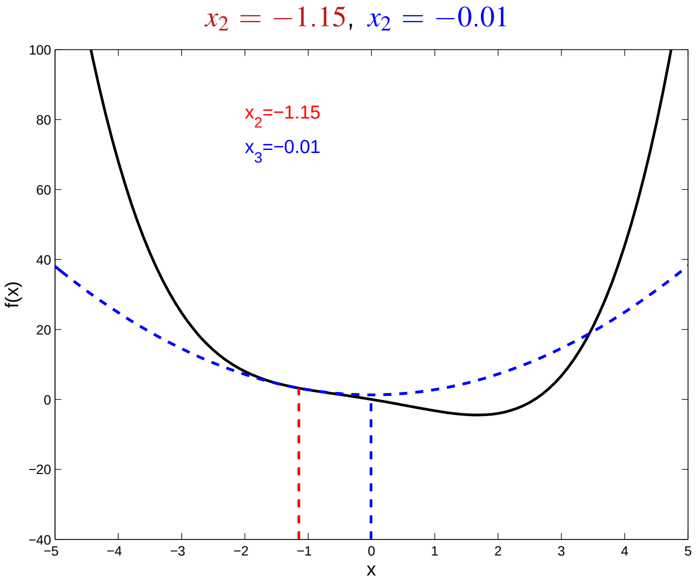
$x_0 = -3, \; x_1 = -1.96$



Example: Convergence Problems



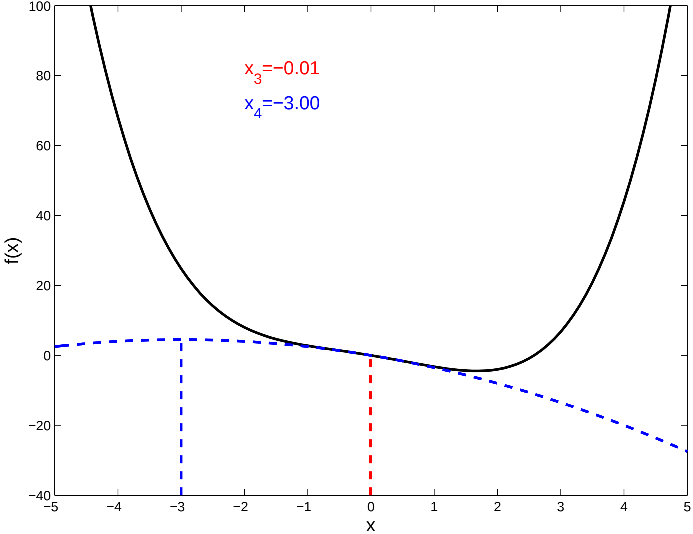
Example: Convergence Problems



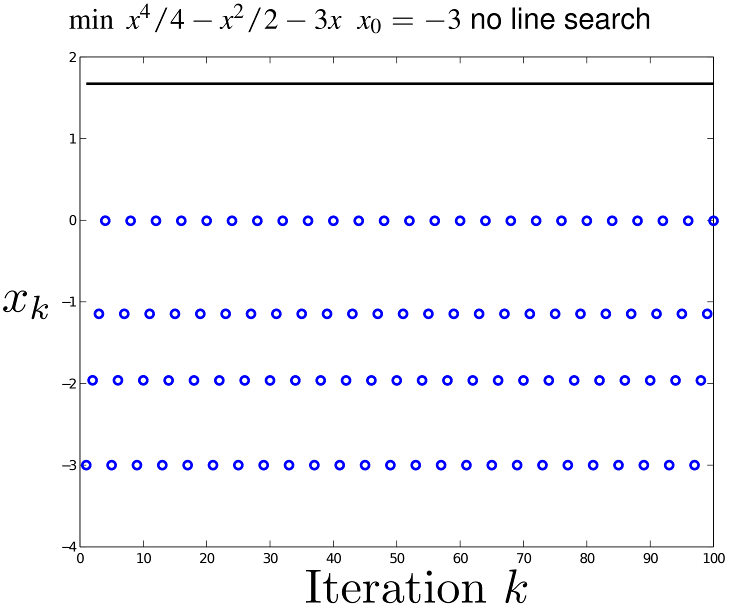
Example: Convergence Problems

$$x_3 = -.01, \; x_4 = -3.00 = x_0$$

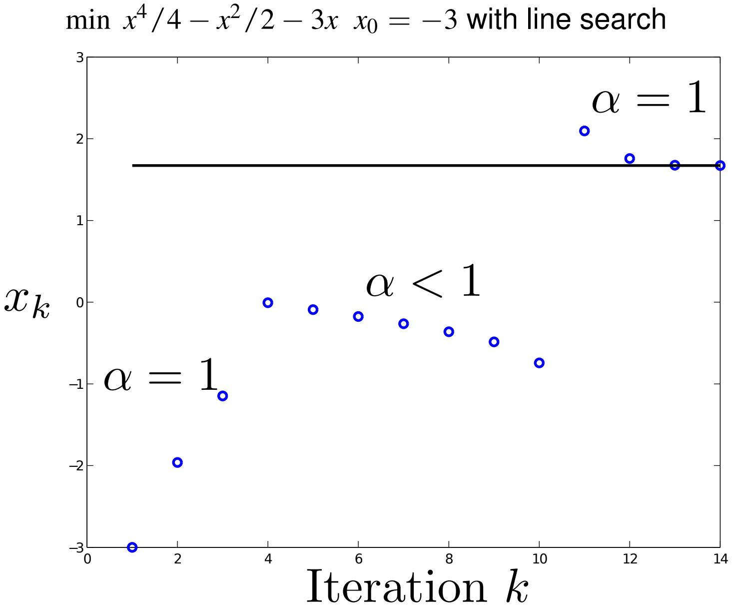
The algorithm returns to the initial point!



Example: `newtonExample0.m`



Example: `newtonExample0.m`



Convergence Theory: Positive Hessian

Key Assumption: The Hessian satisfies,

$$m\boldsymbol{I} \preceq \nabla^2 \boldsymbol{f}(\boldsymbol{x})$$

for some scalar $m > 0$ (this implies that the function is strongly convex and that it has a unique global minimum).
There exists a constants $\eta > 0$ and $\theta > 0$ such that

- If $\|\nabla \boldsymbol{f}(\boldsymbol{x}_k)\|_2 > \eta$ (far away from the solution) then

  $$f(\boldsymbol{x}_{k+1}) - f(\boldsymbol{x}_k) \leq -\theta$$

  i.e. the objective function is reduced at every iteration.
- If $\|\nabla \boldsymbol{f}(\boldsymbol{x}_k)\|_2 \leq \eta$ (close to a solution) then the algorithm converges to the minimum with a quadratic rate.

Illustration $(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c})$ randomly generated

$\min\ \boldsymbol{c}^T\boldsymbol{x} - \sum_{i=1}^{500} \ln(b_i - \boldsymbol{a}_i^T x)$ Backtracking $\beta = 0.01$, $\gamma = 0.5$

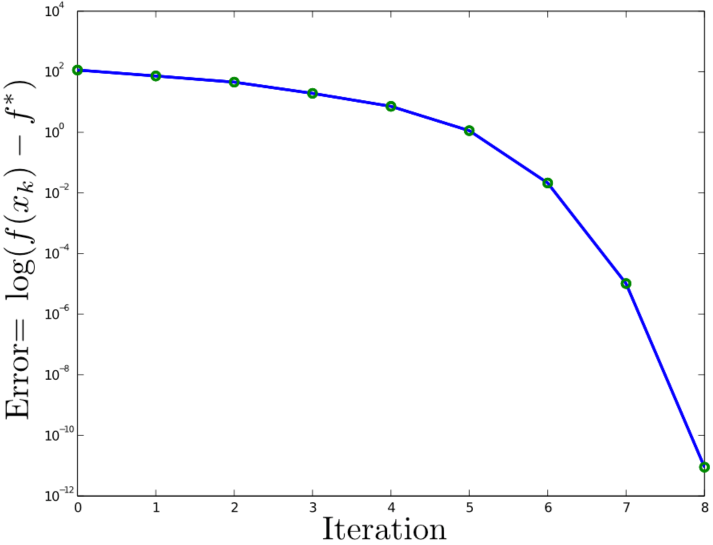
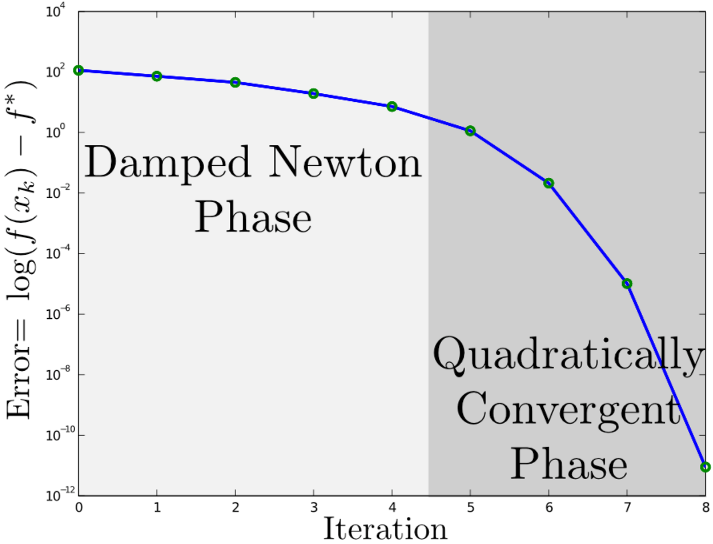


Illustration $(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c})$ randomly generated

$$\min \ \boldsymbol{c}^T\boldsymbol{x} - \sum_{i=1}^{500} \ln(b_i - \boldsymbol{a}_i^T x)$$ Backtracking $\beta = 0.01$, $\gamma = 0.5$



Levenberg–Marquardt Modification

If the Hessian $\nabla^2 \boldsymbol{f}(\boldsymbol{x}_k)$ is not positive definite then the search direction

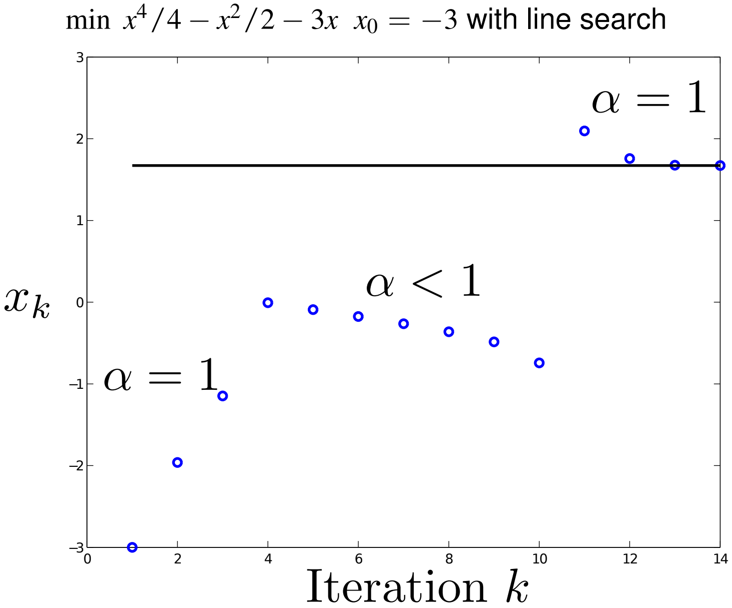
$$\boldsymbol{d}_k = \nabla^2 \boldsymbol{f}(\boldsymbol{x}_k)^{-1} \nabla \boldsymbol{f}(\boldsymbol{x}_k)$$

may not be a descent direction.
Levenberg–Marquardt Modification:

$$\boldsymbol{d}_k = \left(\nabla^2 \boldsymbol{f}(\boldsymbol{x}_k) + \mu_k \boldsymbol{I}\right)^{-1} \nabla \boldsymbol{f}(\boldsymbol{x}_k)$$

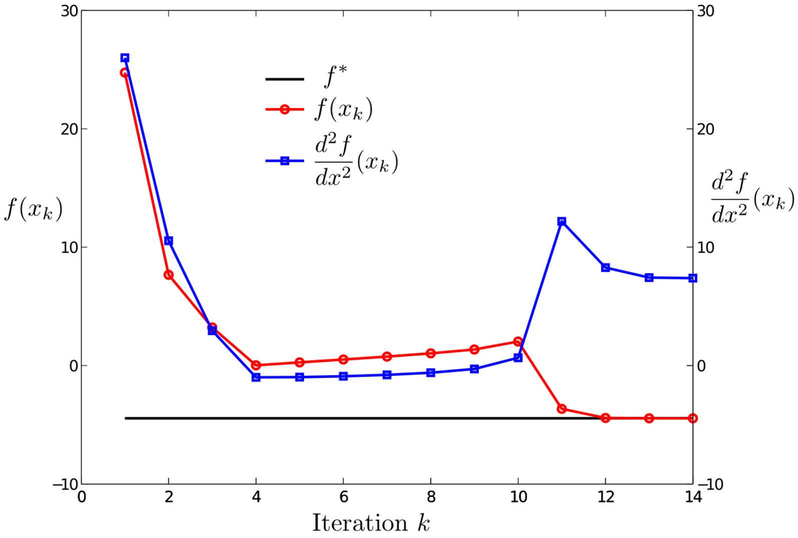
- As $\mu_k \to \infty$ method is like steepest descent with a small step size
- As $\mu_k \to 0$ method is like Newton Raphson
- In practice, start with a small μ and increase it until a descent condition is satisfied

Example: `newtonExample0.m`



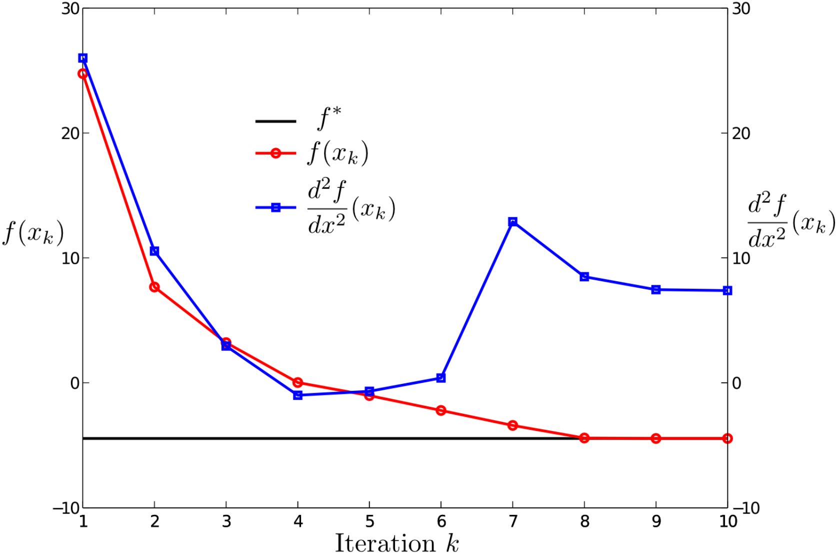
Example: `newtonExample0LV.m`

$\min\ x^4/4 - x^2/2 - 3x\ \ x_0 = -3$ with line search



Example: `newtonExample0LV.m`

With line search & Levenberg–Marquardt Modification ($\mu = 10$)



Quasi Newton Methods

- If the function is convex then Newton-Raphson with a line-search works well:
  1. Guaranteed to converge from any starting point (globally convergent)
  2. Quadratic rate of convergence
  3. Careful/Robust implementations available
- In general the method is not guaranteed to converge from any starting point (usually only locally convergence can be guaranteed)
- Computationally expensive if Hessian is large & dense

Quasi Newton Methods: (not covered in this course)

- Iteratively construct an approximation of $\nabla^2 \boldsymbol{f}(\boldsymbol{x}_k)^{-1}$.
- Most methods generate positive definite approximations
- Algorithms are globally convergent
- State-of-the-art in unconstrained optimisation