C477: Computing for Optimal Decisions
The Newton-Raphson and Related Methods

Panos Parpas
Department of Computing
Imperial College London

www.doc.ic.ac.uk/$\sim$pp500
p.parpas@imperial.ac.uk

Review of 1-D Newton-Raphson 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. **Iterate**

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

Review of 1-D Newton-Raphson 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. **Iterate**

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

Review of 1-D Newton-Raphson 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. **Iterate**

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

Review of 1-D Newton-Raphson 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**

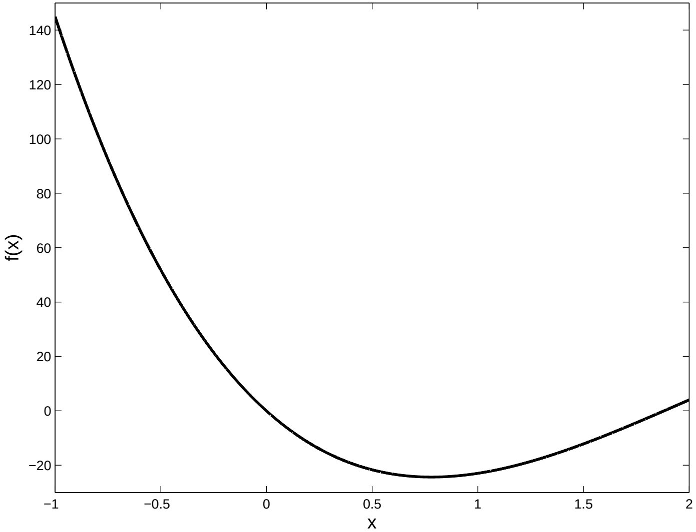
$$0 = q'(x) = f'(x_k) + f''(x_k)(x - x_k)$$

4. **Iterate**

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

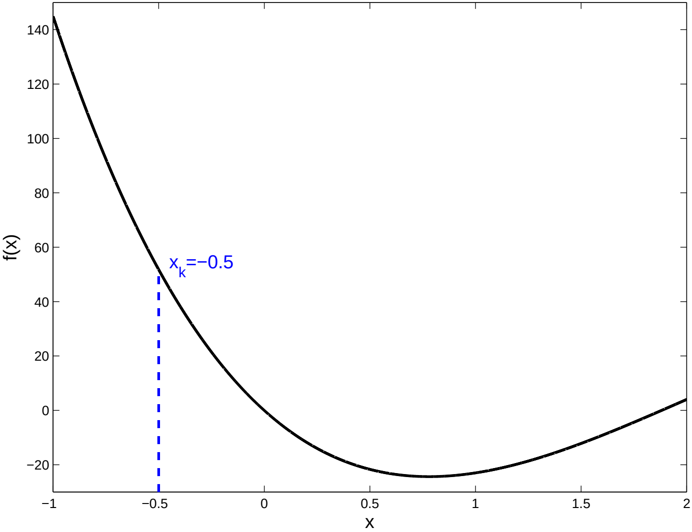
Example: One Dimensional Newton's Method

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



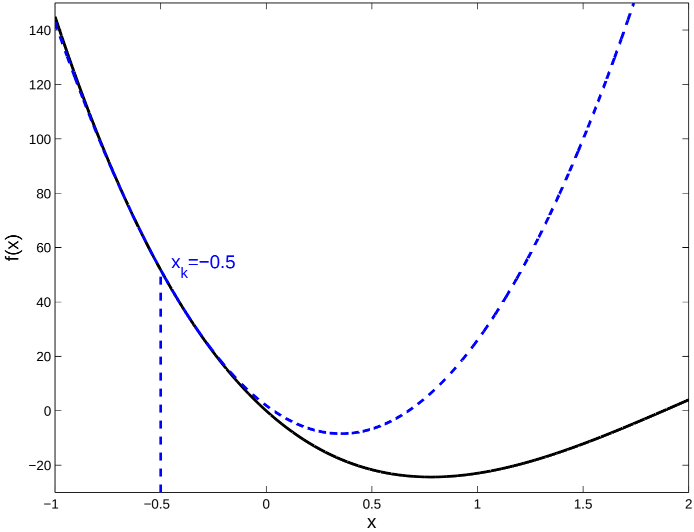
Example: One Dimensional Newton's Method

$$x_k = -0.5$$



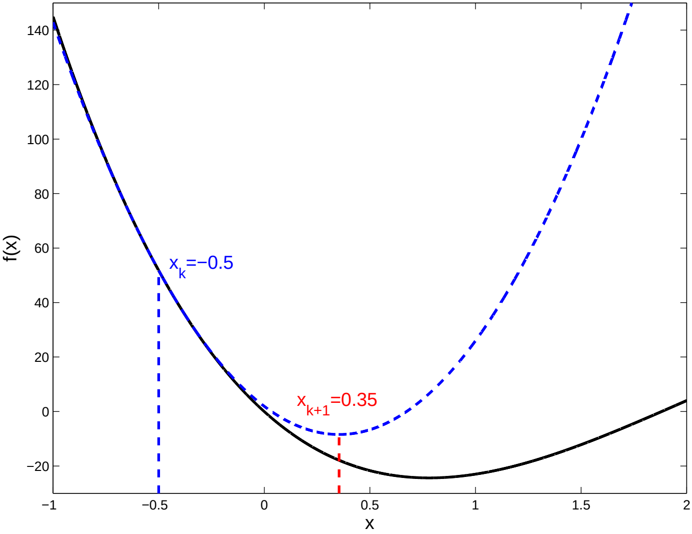
Example: One Dimensional Newton's Method

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



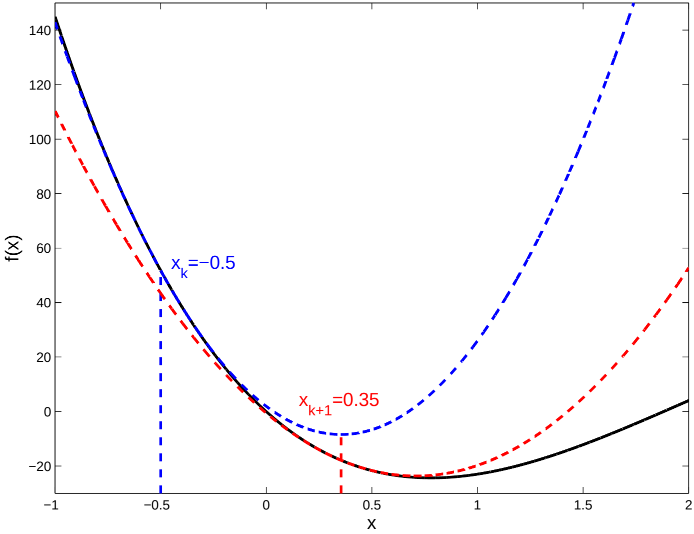
Example: One Dimensional Newton's Method

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



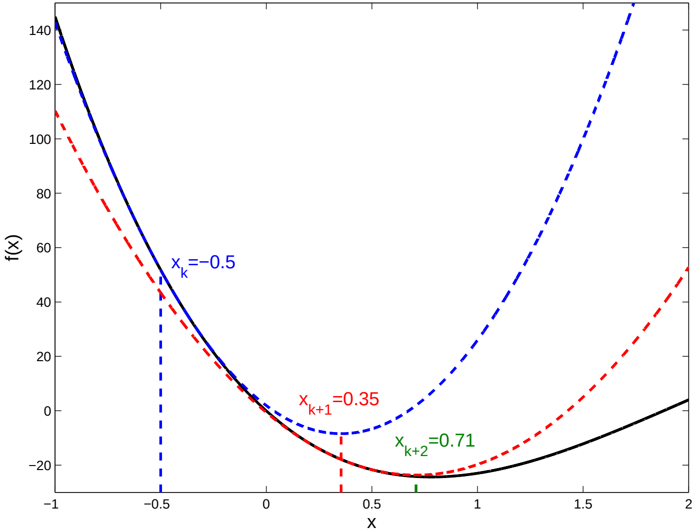
Example: One Dimensional Newton's Method

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



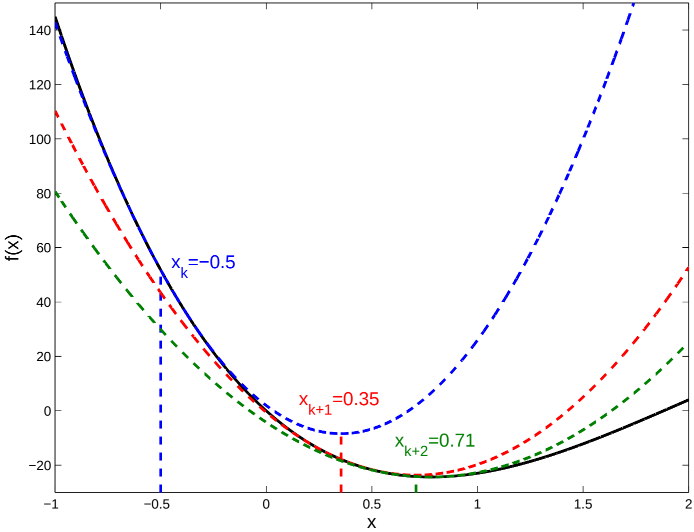
Example: One Dimensional Newton's Method

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



Example: One Dimensional Newton's Method

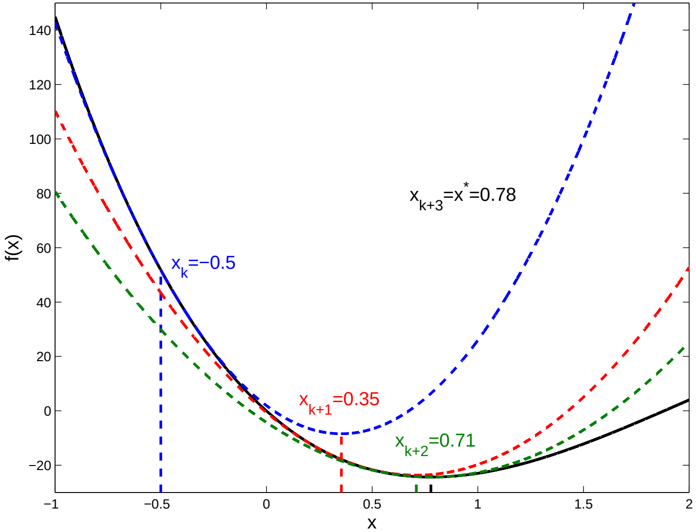
$$q(x) = f(0.71) + f'(0.71)(x - 0.71) + \tfrac{1}{2}f''(0.71)(x - 0.71)^2$$



Example: One Dimensional Newton's Method

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

Convergence after 3 iterations!



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

In the next lecture

(a) Constrained Optimality Conditions

(b) Constrained Optimisation Algorithms

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

In the next lecture

(a) Constrained Optimality Conditions

(b) Constrained Optimisation Algorithms

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

In the next lecture

(a) Constrained Optimality Conditions
(b) Constrained Optimisation Algorithms

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 $\nabla^2 \boldsymbol{f}(\boldsymbol{x}_k) \succ 0$ 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 $\{\boldsymbol{x}_k\}$ 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

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

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

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

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`

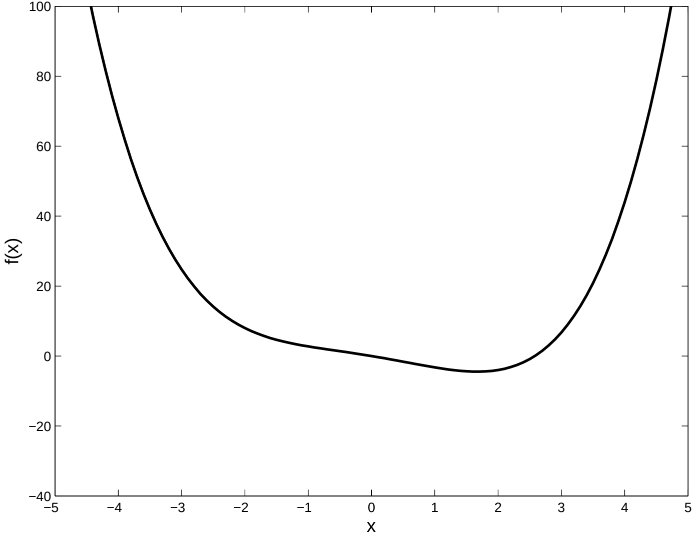
$$f(\boldsymbol{x} + \alpha \boldsymbol{d}) > f(\boldsymbol{x}) + \alpha \beta \nabla \boldsymbol{f}(\boldsymbol{x})^T \boldsymbol{d}$$

`do`

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

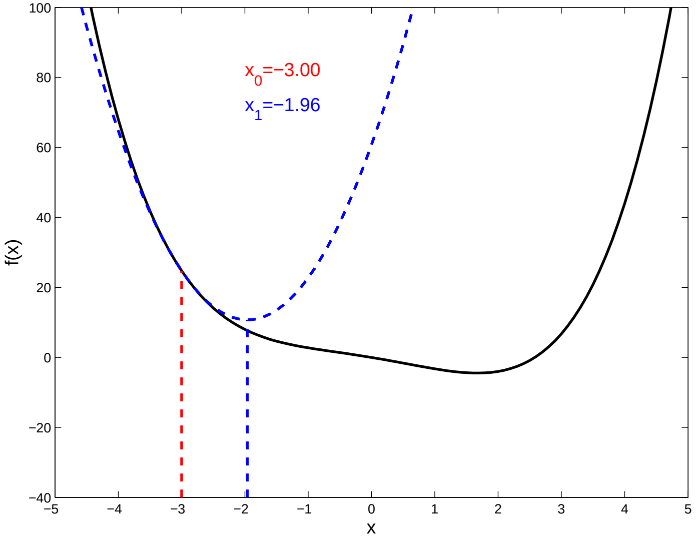
Example: Convergence Problems

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



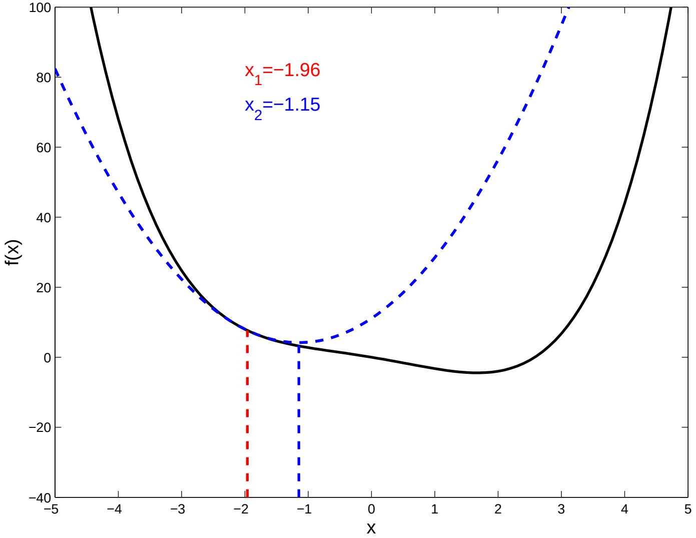
Example: Convergence Problems

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

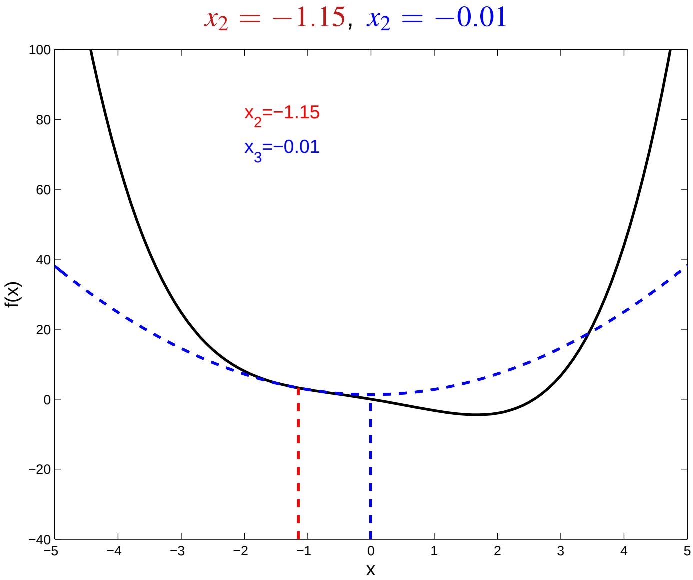


Example: Convergence Problems

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



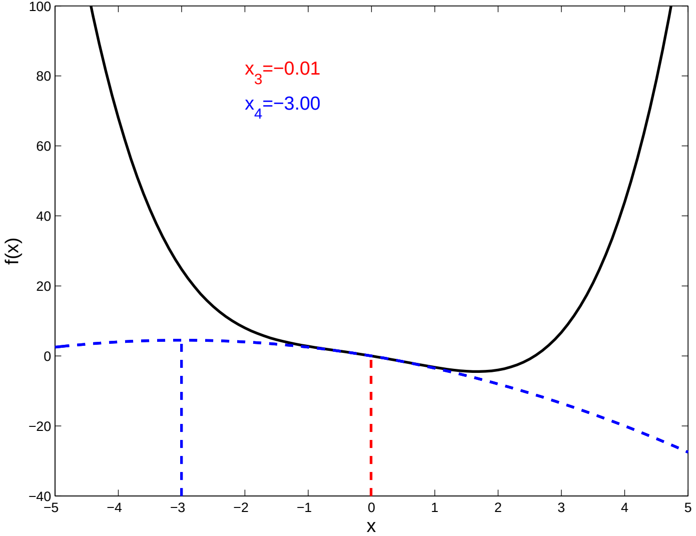
Example: Convergence Problems



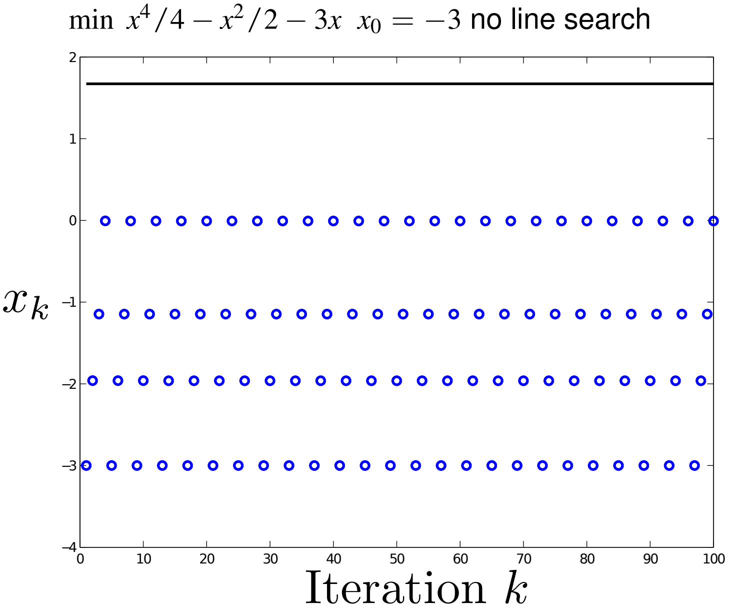
Example: Convergence Problems

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

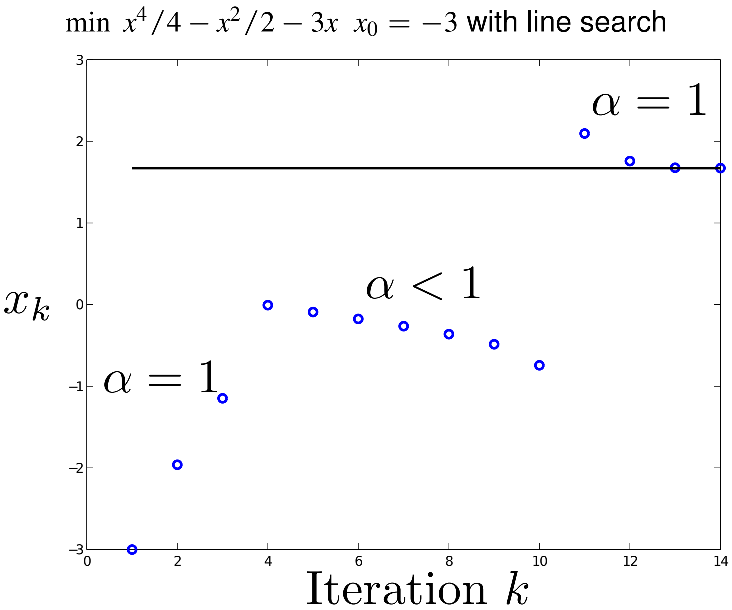
The algorithm returns to the initial point!



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 $\beta = 0.01$, $\gamma = 0.5$

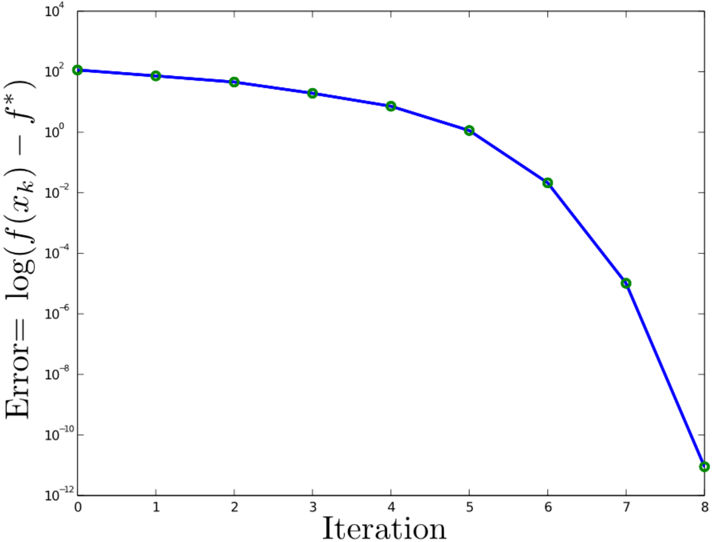
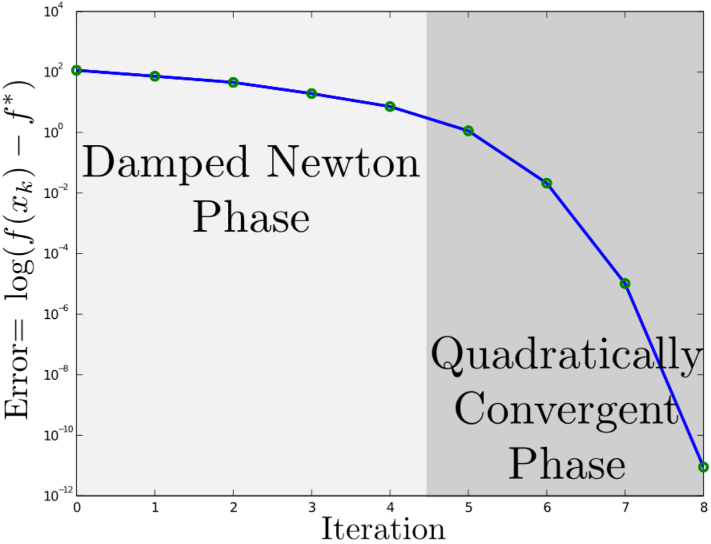


Illustration $(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c})$ randomly generated

$\min\ \boldsymbol{c}^T \boldsymbol{x} - \sum_{i=1}^{500} \ln(b_i - \boldsymbol{a}_i^T x)$ 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

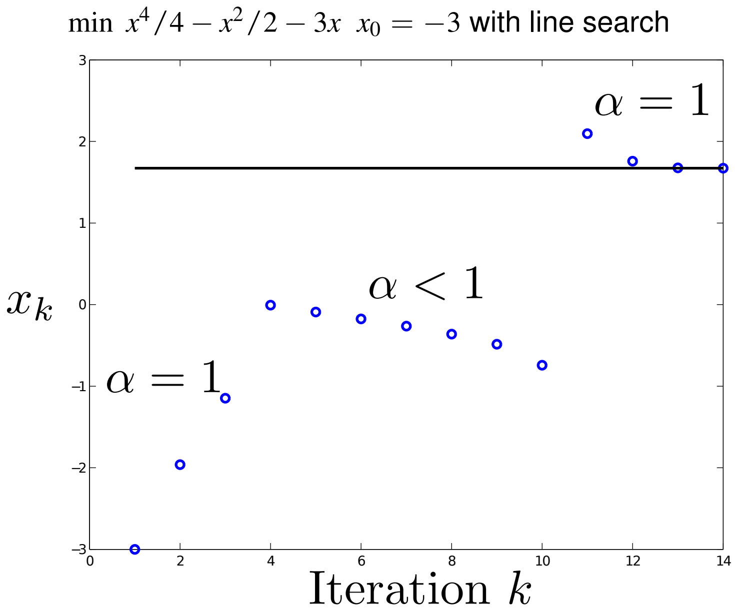
$$\boldsymbol{d}_k = \nabla^2 \boldsymbol{f}(\boldsymbol{x}_k)^{-1} \nabla \boldsymbol{f}(\boldsymbol{x}_k)$$

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

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

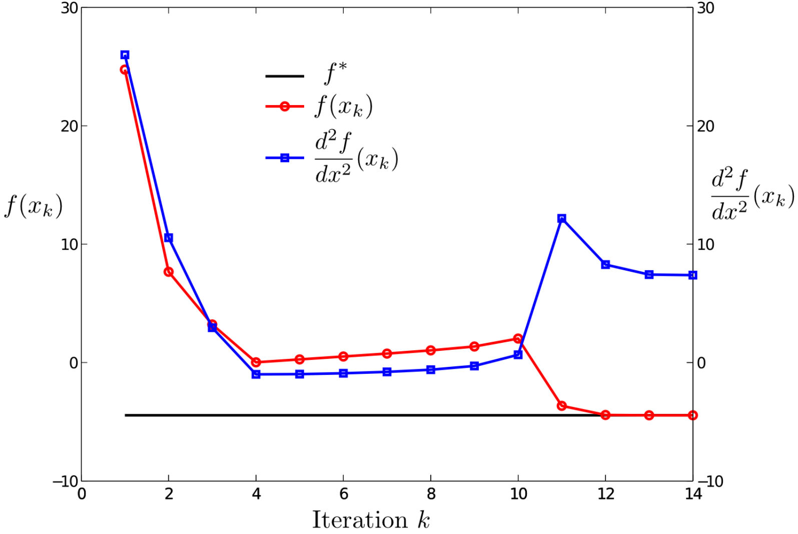
- As $\mu_k \to \infty$ method is like steepest descent with a small step size
- As $\mu_k \to 0$ method is like Newton Raphson
- In practice, start with a small μ and increase it until a descent condition is satisfied

Example: `newtonExample0.m`



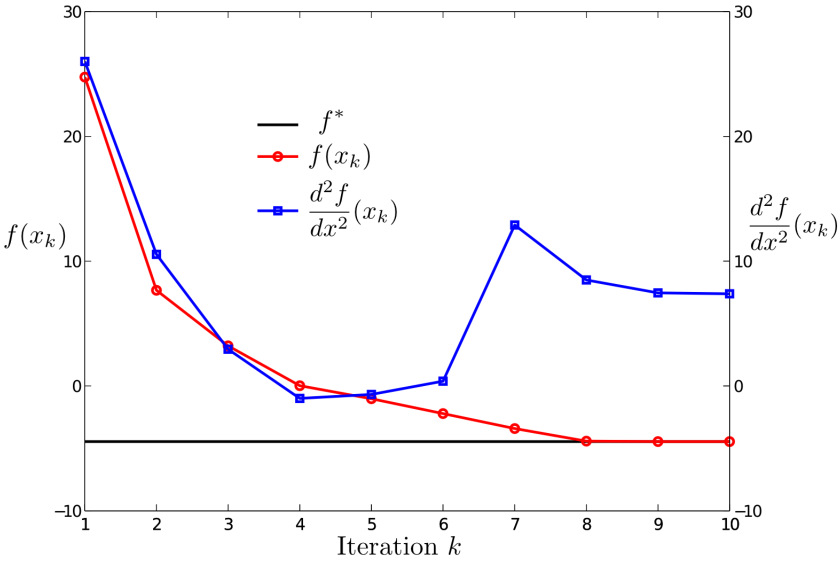
Example: `newtonExample0LV.m`

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