Hochschule Karlsruhe
Technik und Wirtschaft
UNIVERSITY OF APPLIED SCIENCES

UNI FREIBURG

# Introduction to Unconstrained Newton-Type Optimization

Angelika Altmann-Dieses
Faculty of Management Science and Engineering
Karlsruhe University of Applied Sciences

Moritz Diehl
Department of Microsystems Engineering (IMTEK) and Department of Mathematics
University of Freiburg

*(some slide material was provided by W. Bangerth and K. Mombaur)*

# Aim of Newton type optimization algorithms

$$\min f(x) \quad (x \in R^n)$$

- Find a local minimizer $x^*$ of f(x), i.e. a point satisfying

$$\nabla f(x^*) = 0$$

# Derivative based algorithms

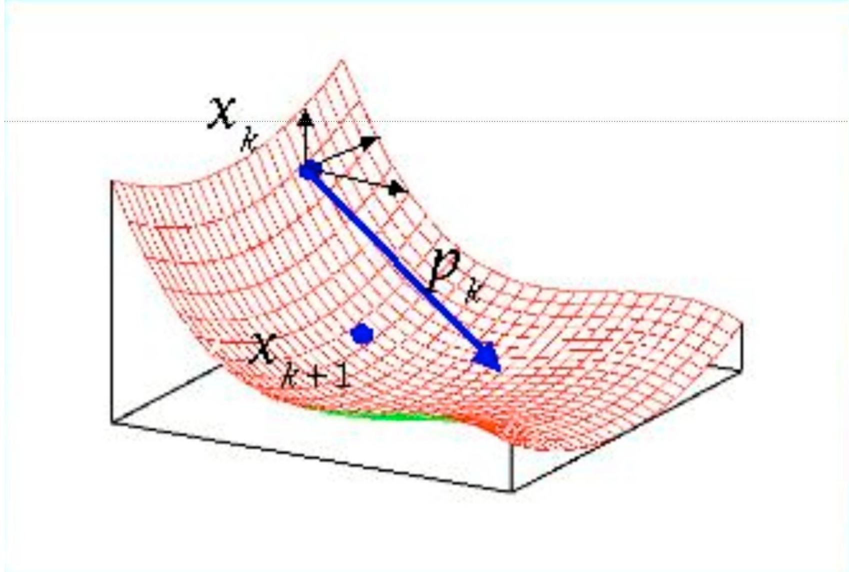
- **Fundamental underlying structure of most algorithms:**

  - choose start value $x_0$
  - for i=1, ......:
    - determine direction of search (descent) p
    - determine step length $\alpha$
    - new iterate $x_{i+1} = x_i + \alpha\, p$
    - check convergence

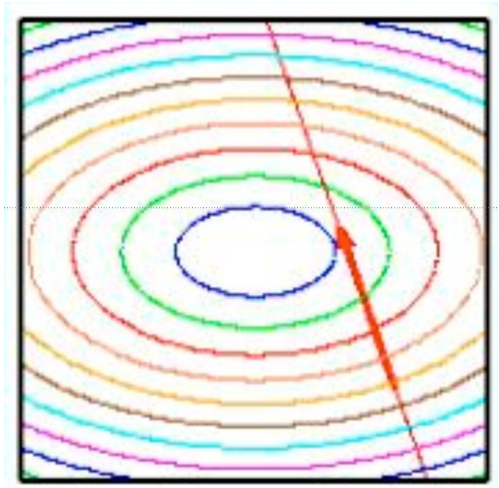
- Optimization algorithms differ in the choice of p und $\alpha$

# Basic algorithm:

**Search direction:**
choose descent direction
(f should be decreased)



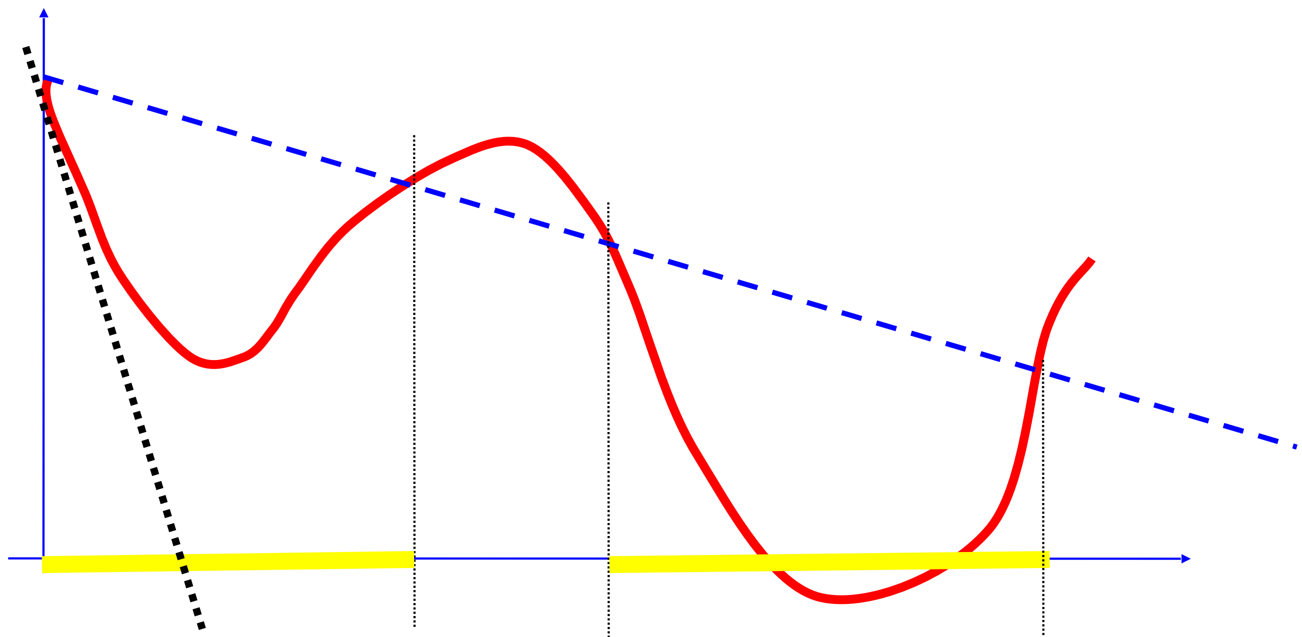

**Step length:**
solve1-d minimization approximately,
satisfy Armijo condition



# Computation of step length

- Dream:
  - exact line search: $\alpha^k = \arg\min_\alpha f(x^k + \alpha p^k)$
- In practice:
  - **inexact line search:** $\alpha^k \approx \arg\min_\alpha f(x^k + \alpha p^k)$
  - ensure sufficient decrease, e.g. Armijo condition

$$f(x_k + \alpha_k p_k) \leq f(x_k) + c_1 \alpha_k \nabla f_k^T p_k$$



# How to compute search direction?

- We discuss three algorithms:
  - Steepest descent method
  - Newton‘s method
  - general Newton-type methods

# Algorithm 1: Steepest descent method

- Based on first order Taylor series approximation of objective function

$$f(x_k + p_k) = f(x_k) + \underbrace{\nabla f(x_k)^T p_k}_{} + \ldots$$

  - maximum descent, if

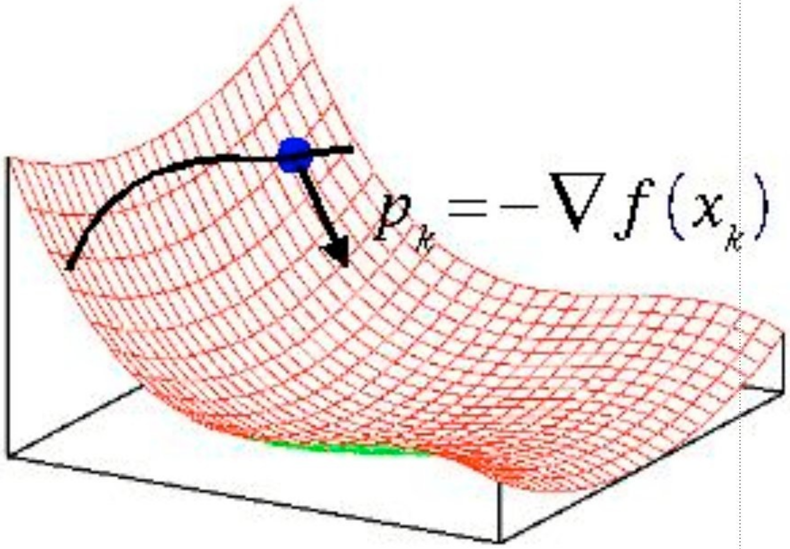
$$\begin{aligned} \frac{\nabla f(x_k)^T p_k}{||p_k||} &\rightarrow \min! \\ \Rightarrow p_k &= -\nabla f(x_k) \end{aligned}$$

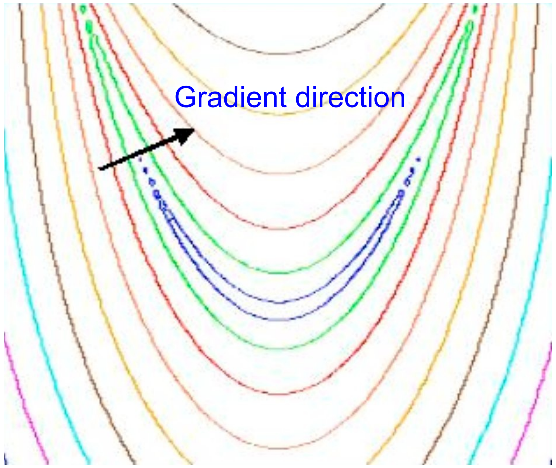
# Steepest descent method

Choose steepest descent search direction, perform (exact) line search:

$$p^k = -\nabla f(x^k) \qquad x^{k+1} = x^k - \alpha^k \nabla f(x^k)$$

search direction is perpendicular to level sets of *f(x)*



# Convergence of steepest descent method

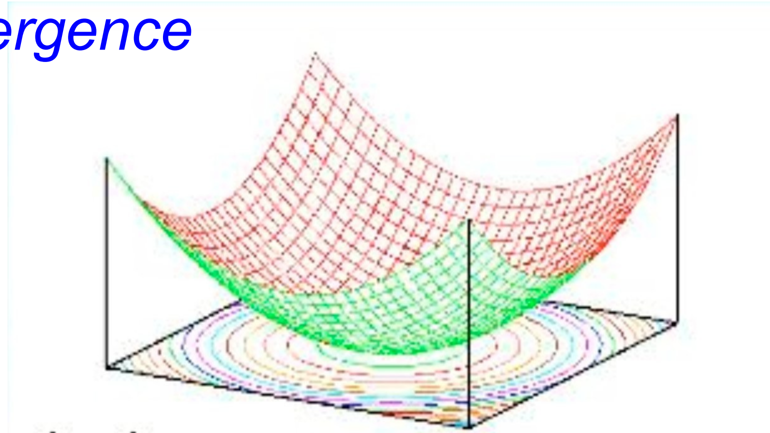
steepest descent method has *linear convergence*

i.e. $\left\| x^k - x^* \right\| \leq C \left\| x^{k-1} - x^* \right\|$

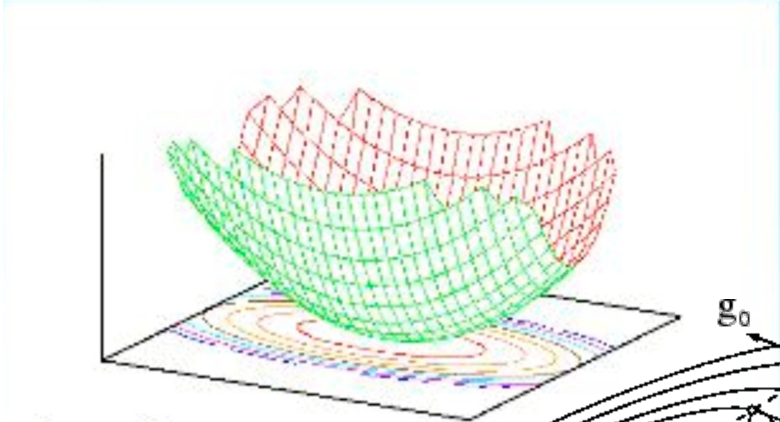
- gain is a fixed factor $C<1$
- convergence can be very slow if $C$ close to 1

*If* $f(x) = x^T A x$, $A$ positive definite, $\lambda$ denotes eigenvalues of $A$, one can show that

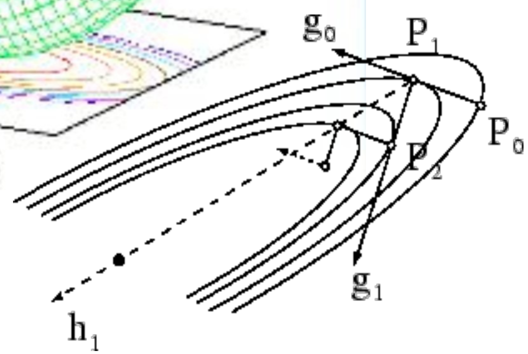
$$\Rightarrow \quad C \approx \frac{\lambda_{\max} - \lambda_{\min}}{\lambda_{\max} + \lambda_{\min}}$$



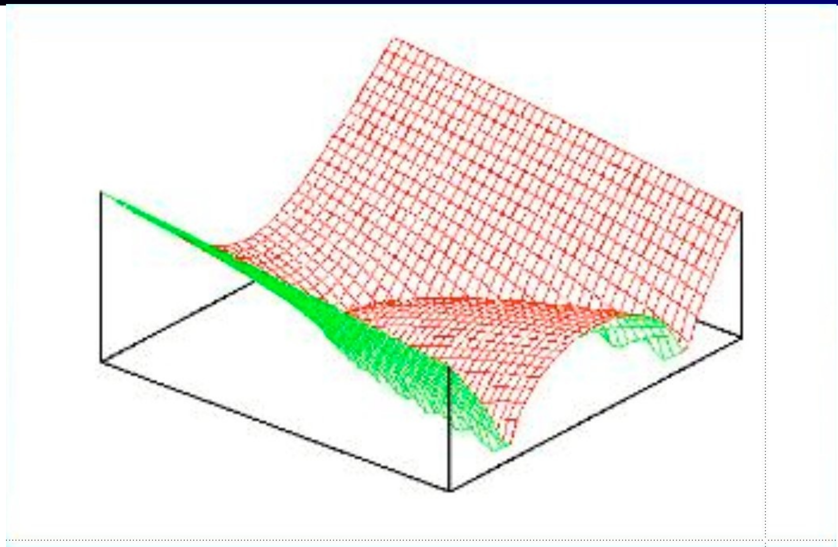
$x^2 + y^2 \rightarrow C=0$



$x^2 + 5y^2 \rightarrow C \approx 0.6$

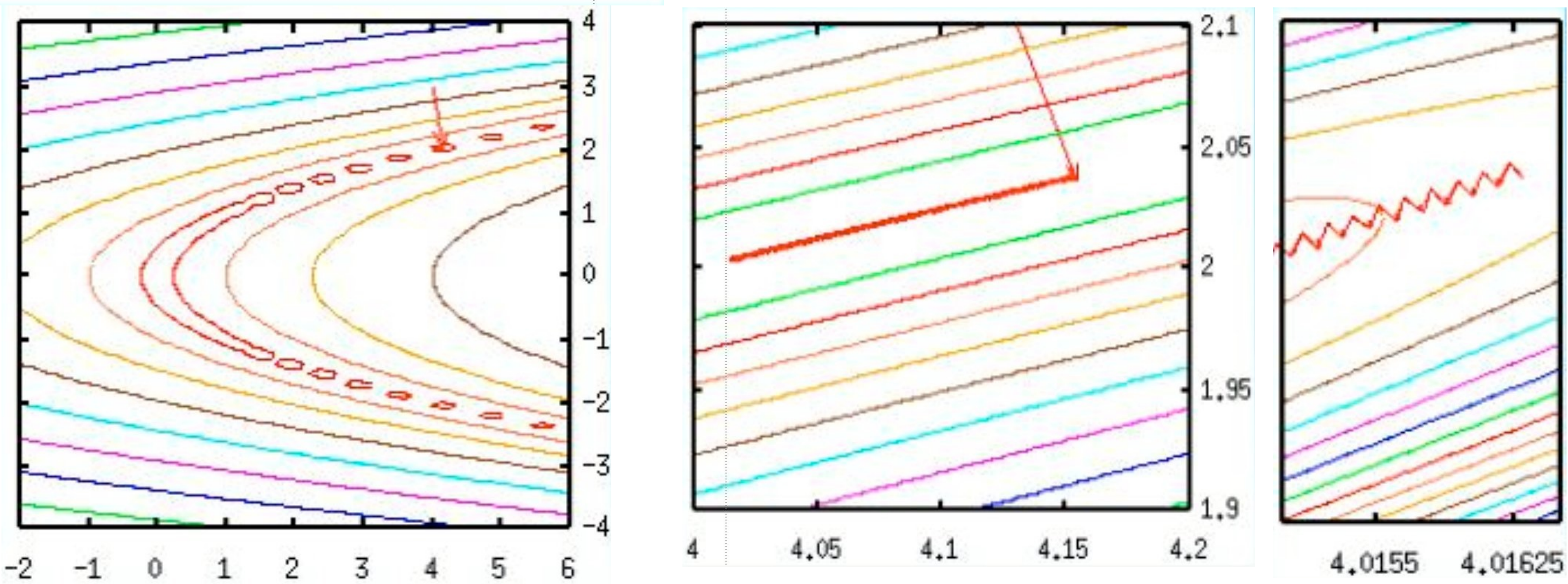


# Example - steepest descent method

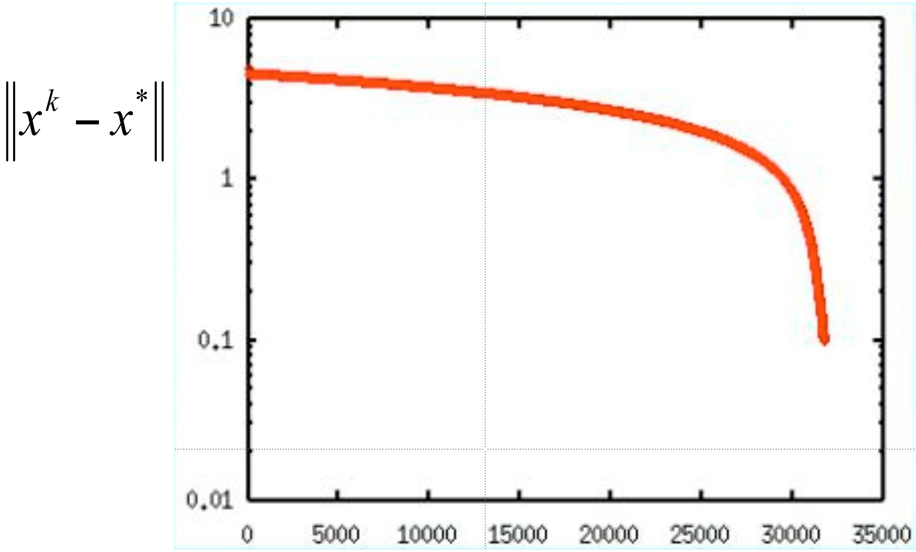


$$f(x,y)=\sqrt[4]{(x-y^2)^2+\frac{1}{100}}+\frac{1}{100}y^2$$

banana valley function,
global minimum at $x=y=0$



# Example - steepest descent method




*Convergence of steepest descent method:*

- needs almost 35.000 iterations to come closer than 0.1 to the solution
- mean value of convergence constant *C*: 0.99995
- at *(x=4,y=2)*, there holds

$$\lambda_1 = 0.1, \lambda_2 = 268 \quad \Rightarrow \quad C \approx \frac{268 - 0.1}{268 + 0.1} \approx 0.9993$$

# Algorithm 2: Newton's method

- Based on **second order** Taylor series approximation of f(x)

$$f(x_k + p_k) = f(x_k) + \underbrace{\nabla f(x_k)^T p_k + \frac{1}{2} p_k^T \nabla^2 f(x_k)\, p_k}_{} + \ldots$$

$$\nabla f(x_k)^T p_k + \frac{1}{2} p_k^T \nabla^2 f(x_k)\, p_k \to \min!$$

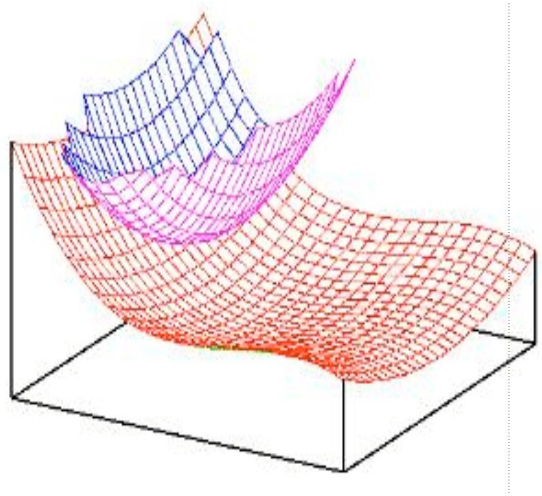
$$\Leftrightarrow \qquad \nabla^2 f(x_k)\, p_k = -\nabla f(x_k)$$

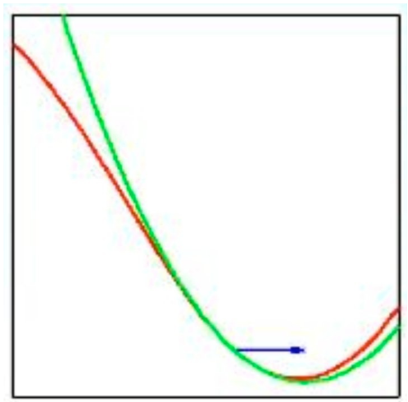
„Newton-Direction“ $p_k = -(\nabla^2 f(x_k))^{-1}\, \nabla f(x_k)$

# Visualization of Newton's method

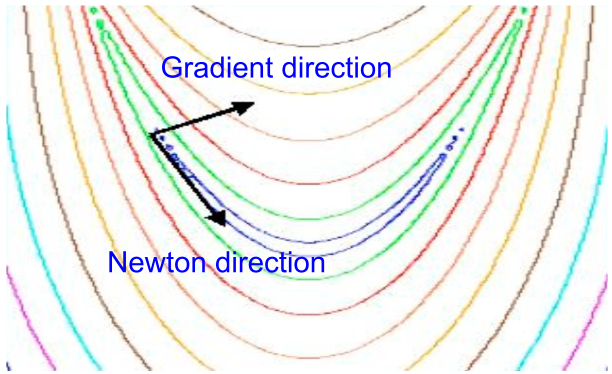
$p_k$ minimizes **quadratic approximation** of the objective

$$Q(p^k) = f(x^k) + \nabla f(x^k)p^k + \frac{1}{2}p^{k^T}\nabla^2 f(x^k)p^k$$





if quadratic model is good, then take full step with $\alpha^k = 1$



# Convergence of Newton's method

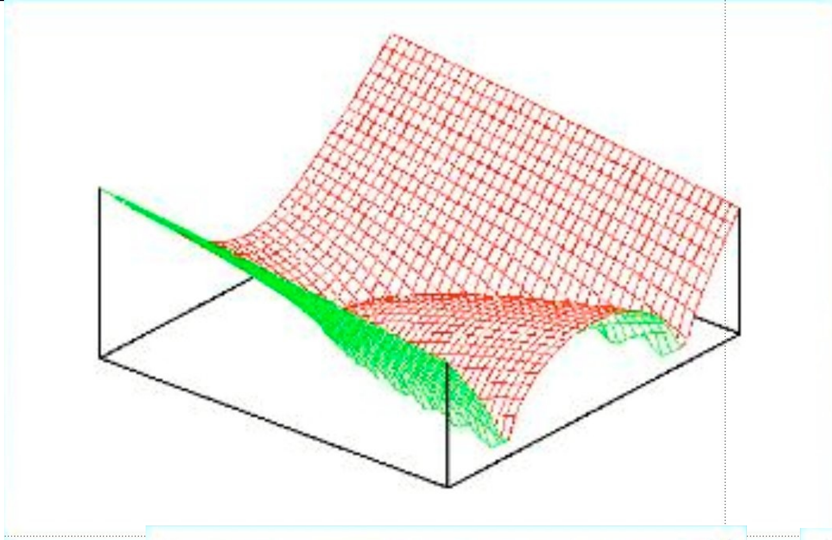
Newton's method has *quadratic convergence*

i.e. $$\left\| x^k - x^* \right\| \leq C \left\| x^{k-1} - x^* \right\|^2$$

This is ***very fast*** close to a solution:

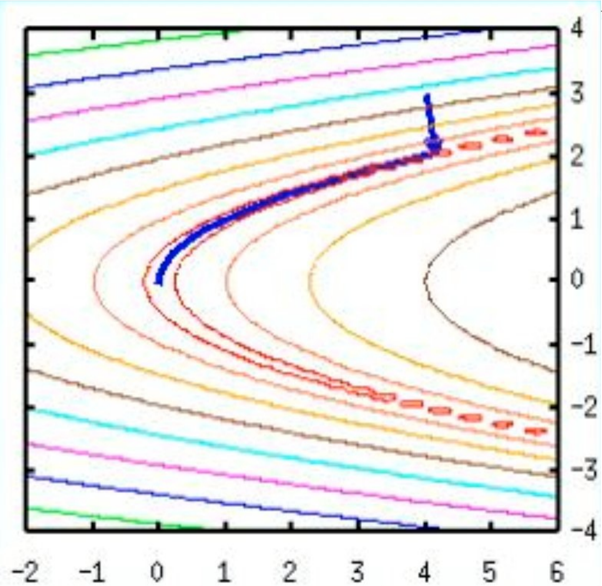
Correct digits double in each iteration!

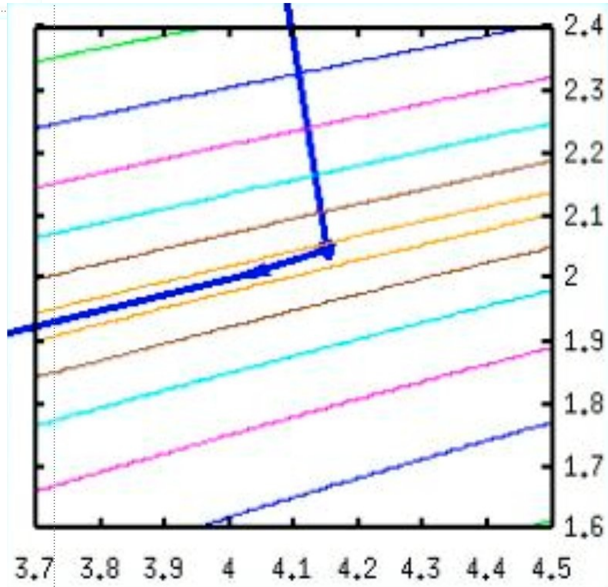
# Example - Newton‘s method

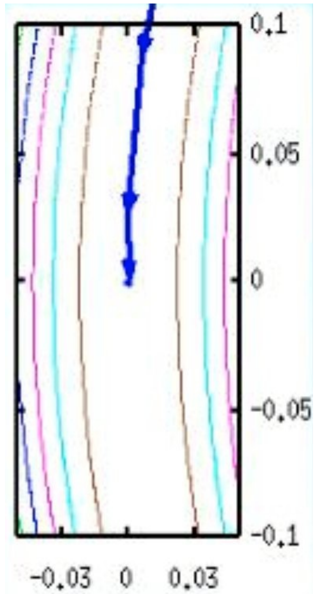


$$f(x,y)=\sqrt[4]{(x-y^2)^2+\frac{1}{100}}+\frac{1}{100}y^2$$

banana valley function,
global minimum at *x=y=0*

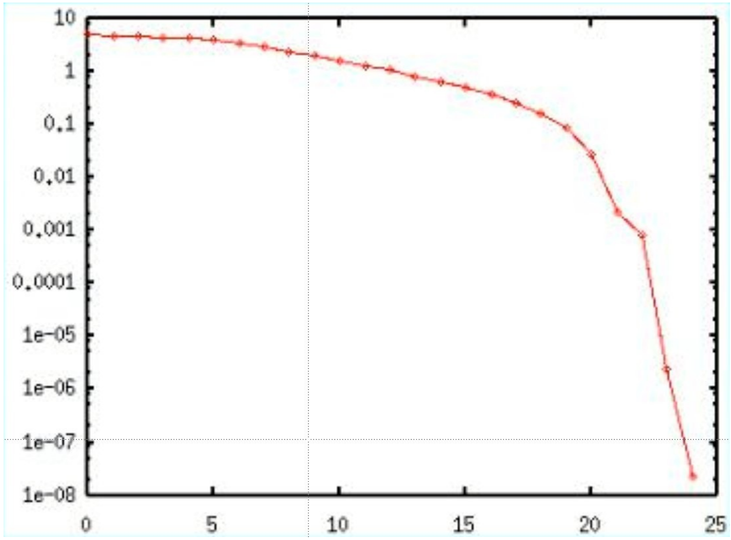


# Example - Newton‘s method

$\|x^k - x^*\|$

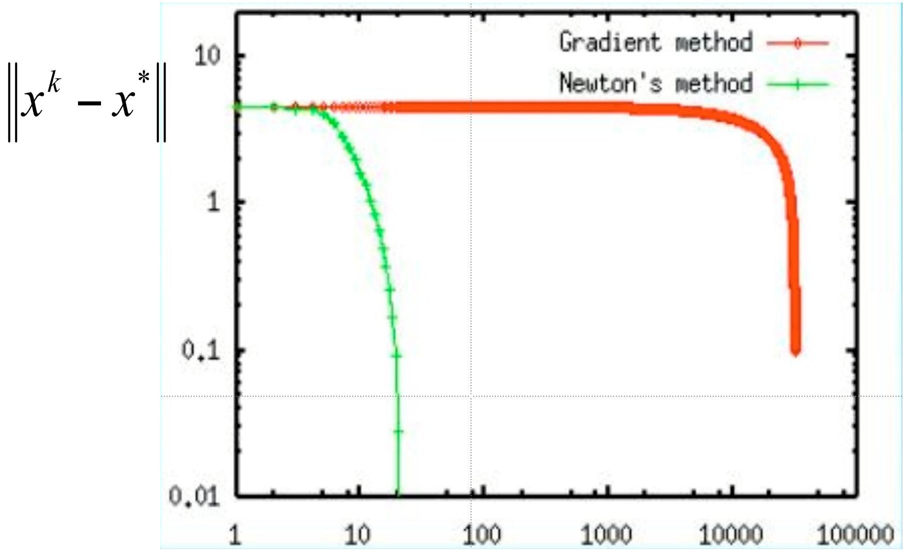



*Convergence of Newton‘s method:*

- less than 25 iterations for an accuracy of better than $10^{-7}$!
- convergence roughly *linear* for first 15-20 iterations since step length $\alpha_k \neq 1$
- convergence roughly *quadratic* for last iterations with step length

$\alpha_k = 1$

# Comparison of steepest descent and Newton




For banana valley example:

- Newton‘s method **much faster** than steepest descent method (factor 1000)
- Newton‘s method superior due to higher order of convergence
- steepest descent method converges too slowly for practical applications

# Generalization to Newton-type methods

In practice, evaluation of second derivatives
for the Hessian can be difficult

➔ approximate Hessian matrix $\nabla^2 f(x^k)$

➔ often methods ensure that the approximation $B_k$ is positive definite

$$x^{k+1} = x^k - B_k^{-1} \nabla f(x^k)$$

$$B_k \approx \nabla^2 f(x^k)$$

All these methods (including the previous ones) are collectively known as *Newton-type methods*

# Newton-type variants

- **Steepest Descent:**

$$B_k = I$$

Convergence rate: linear

- **Newton's Method:**

$$B_k = \nabla^2 f(x^k)$$

Convergence rate: quadratic

# Newton-type variants (continued)

- **BFGS** quasi-Newton update formula (Broyden, Fletcher, Goldfarb, Shanno)

$$B_{k+1} \quad = \quad B_k - \frac{B_k s s^T B_k}{s^T B_k s} + \frac{y y^T}{s^T y}$$

with $\quad s \;=\; x_{k+1} - x_k, \quad$ and $\quad y \;=\; \nabla f(x_{k+1}) - \nabla f(x_k)$

convergence rate: super-linear

- For Least-Squares Problems: **Gauss-Newton Method**

$$f(x) = \frac{1}{2}\|F(x)\|^2 \quad J(x) \;=\; \frac{\partial F(x)^T}{\partial x}$$

$$B_k \;=\; J(x^k)^T J(x^k)$$

convergence rate: linear

# Summary of Newton-type optimization (unconstrained)

- Aim: find **local minima** of smooth nonlinear problems: $\nabla f(x^*)=0$

- Derivative based methods iterate $x_{i+1} = x_i + \alpha_i p_i$ with
  - **search direction** $p_i$ and **step length** $\alpha_i$ .
  - start at **initial guess** $x_0$ ,

- Four examples of Newton-type methods:
  - steepest descent: intuitive, but slow linear convergence
  - exact Newton's method: very fast quadratic convergence
  - BFGS: fast superlinear convergence
  - Gauss-Newton (only for least-squares): fast linear convergence