Computational Numerical Statistics

PROJECT 2

Group G1 – TP2

Ana Mendes, Nº 57144
Emanuele Vivoli Nº 57284
Joao Camacho Nº 56861
Lia Schmid Nº 57629

Simao Goncalves Nº 54896

First Semester 2019–2020

Optimization

1. Introduction
2. Analytical solution
3. Bisection method
4. Newton-Raphson method
5. Fisher Scoring Method

Optimization

Let $X \sim Beta(\alpha, 1)$, which has p.d.f.

$$f(x;\alpha) = \frac{x^{\alpha-1}}{B(\alpha,1)}, \alpha > 0, x \in [0,1]$$

$\mathrm{B}(\alpha,1) = \frac{\Gamma(\alpha)\Gamma(1)}{\Gamma(\alpha+1)} = \frac{\cancel{\Gamma(\alpha)}\,\overbrace{\Gamma(1)}^{=1}}{\alpha\,\cancel{\Gamma(\alpha)}} = \frac{1}{\alpha}$

So the simplified p.d.f. is

$$f(x;\alpha) = \frac{x^{\alpha-1}}{\frac{1}{\alpha}} = \alpha x^{\alpha-1}$$

Let

```
0.5409477 0.8184872 0.7848854 0.9850439 0.8963032 0.6089008
0.9549606 0.6795304 0.8451902 0.5613979 0.4029634 0.2741569
0.3996693 0.6371445 0.7521881
```

be an observed sample from X.

Likelihood, log-likelihood and score functions

- Likelihood function:

$$L(\alpha) = \prod_{i=1}^{n} f(x_i|\alpha) = \alpha^n \prod_{i=1}^{n} x_i^{\alpha-1}$$

- Log-likelihood function:

$$l(\alpha) = logL(\alpha) = nlog(\alpha) + (\alpha - 1)\sum_{i=1}^{n} log(x_i)$$

- Score function:

$$s(\alpha) = l'(\alpha) = \frac{n}{\alpha} - \sum_{i=1}^{n} log(x_i)$$

Maximum likelihood estimation of α

The MLE of α is obtained by maximizing L(α)

1. Solve the equation $s(\alpha) = 0$

$$
\begin{aligned}
s(\alpha) = 0 &\Leftrightarrow \frac{n}{\alpha} + \sum_{i=1}^{n} log(x_i) = 0 \\
&\Leftrightarrow \frac{n}{\alpha} = -\sum_{i=1}^{n} log(x_i) \\
&\Leftrightarrow \alpha = -\frac{n}{\sum_{i=1}^{n} log(x_i)}
\end{aligned}
$$

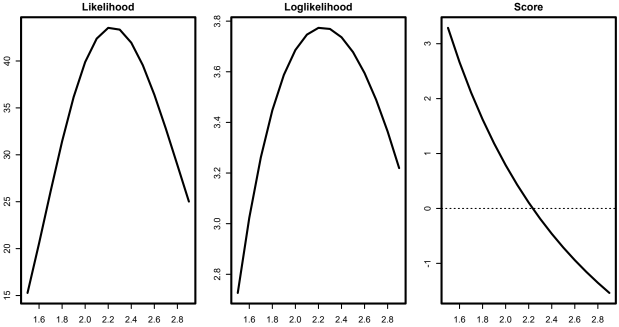
Maximum likelihood estimation of α

2. Confirm that $s'(\alpha) < 0$

$$s'(\alpha; x) = l''(\alpha) = \left(\frac{n}{\alpha} - \sum_{i=1}^{n} log(x_i)\right)' = -\frac{n}{\alpha^2} < 0$$

- The maximum likelihood estimator of α is $\hat{\alpha} = -\frac{n}{\sum_{i=1}^{n} log(x_i)}$
- Calculate MLE for sample from , $\hat{\alpha} = 2.235083$

Maximum likelihood estimation of α



Maximum likelihood estimation of α

- Approximation of the ML estimate α using the R function maxLik()

```
maxLik(logLik=loglik, start=2.2)
#Maximum Likelihood estimation
#Newton-Raphson maximisation, 3 iterations
#Return code 1: gradient close to zero
#Log-Likelihood: 3.775335 (1 free parameter)
#Estimate(s): 2.235083
```

First method

BISECTION

Bisection method

Bolzano Theorem

Theorem

Let f be a continuous function in the limited interval [a, b] $\in \mathbb{R}$ such that:

$$f(a)f(b) \leq 0$$

then f has at least one root $x^* \in \,]a, b[$.

Bisection method

Proof.

$$[a_{n+1}, b_{n+1}] = \begin{cases} [a_n, x^{(n)}] & \text{if } f(a_n)f(x^{(n)}) \leq 0 \\ [x^{(n)}, b_n] & \text{if } f(a_n)f(x^{(n)}) > 0 \end{cases}$$

using $x^{(t+1)} = \frac{a_{t+1}+b_{t+1}}{2}$
We had created two sequences which bound each other:

- $(a_n)_{n\geq 0}$ as a monotonic increasing sequence
- $(b_n)_{n\geq 0}$ as a monotonic decreasing sequence

$\square$

Bisection method

Proof.

- We proved the size of the interval (by Induction) follows the relation:

$$b_n - a_n = \frac{b_0 - a_0}{2^n}$$

- With $n \to \infty$ the $b^* - a^* \to 0$, so the two sequenced have same limit.
- Then $(a_n) \leq (x_n) \leq (b_n)$ using the Squeeze theorem

$$\lim_{n\to\infty} a_n = \lim_{n\to\infty} x_n = \lim_{n\to\infty} b_n = x^*$$

$\square$

Bisection method

What left?

Proof.

To show that $f(x^*) = 0$, and it is given from the hypothesis condition of the Bolzano Theorem that is always guaranteed: $f(a_n)f(b_n) \leq 0$.
It is also true in the limit point, so we we have:

$$\begin{aligned} f(x^*)f(x^*) \leq 0 &\Leftrightarrow [f(x^*)]^2 \leq 0 \\ &\Leftrightarrow f(x^*) = 0 \end{aligned}$$

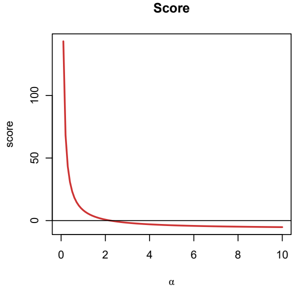
$\square$

Bisection method

We can observe that s function is strictly descending, because

$$s'(\alpha; x) = -\frac{n}{\alpha^2} < 0$$

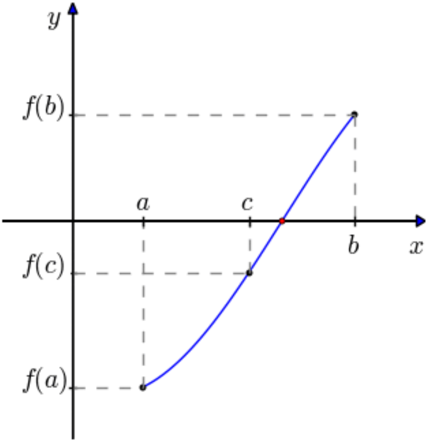
So, the function has one root in $\mathbb{R}^+$.



Therefore, we can conclude, whatever range that verifies the Bolzano theorem, this method will always converge.

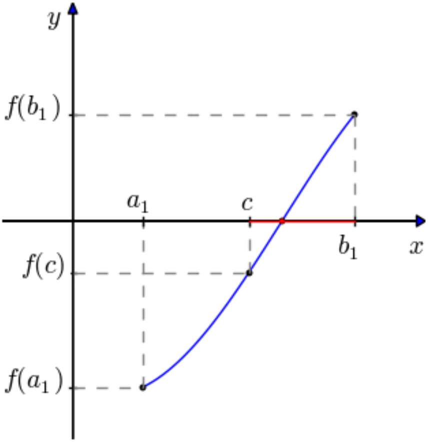
Bisection Method

Algorithm sketch



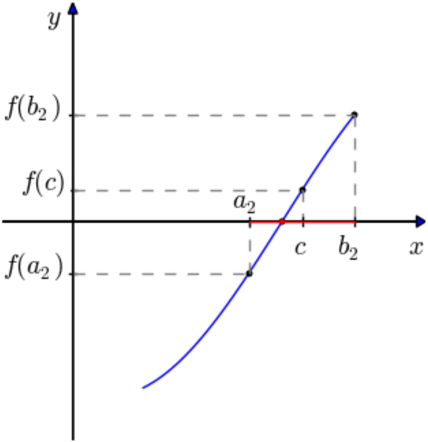
Bisection Method

Algorithm sketch



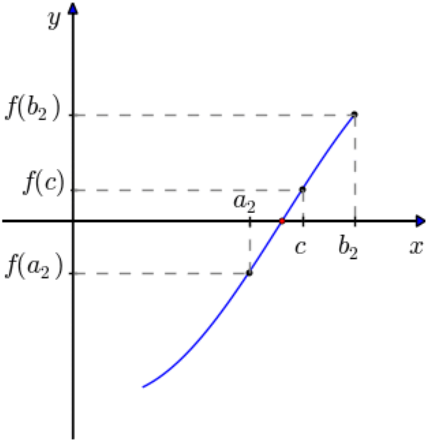
Bisection Method

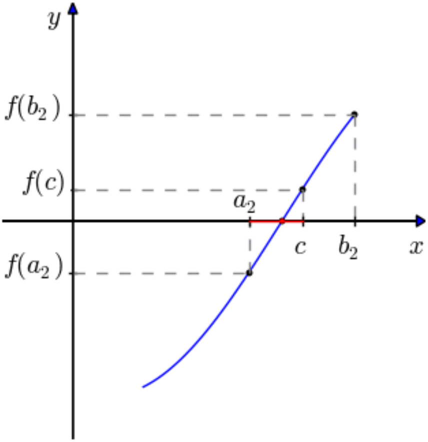
Algorithm sketch



Bisection Method

Algorithm sketch

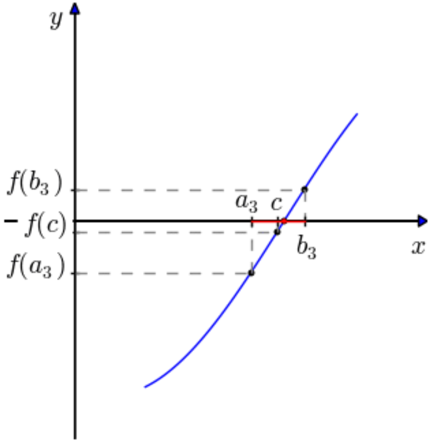




y
$f(b_2)$
$f(c)$
a_2
$f(a_2)$
c
b_2
x

Bisection Method

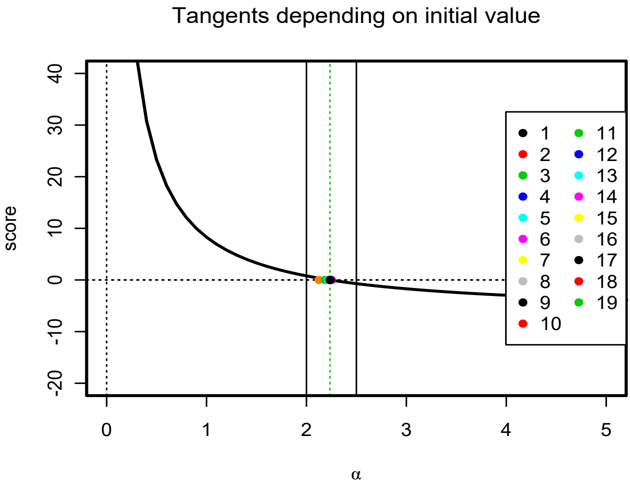
Algorithm sketch



Bisection method

$a_0 = 2, b_0 = 2.5$

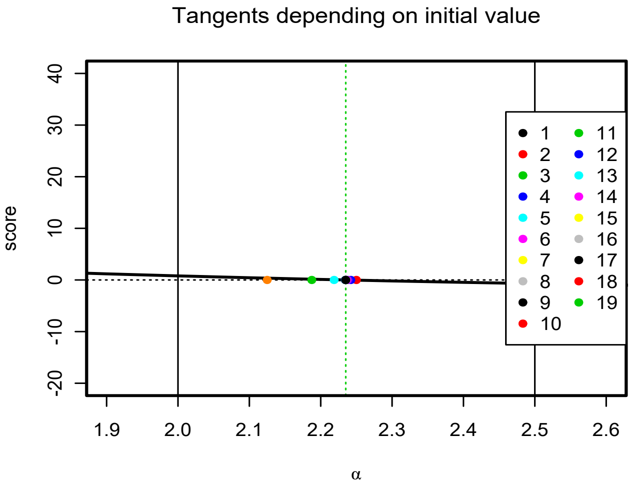
$\hat{\alpha}$ = 2.235084



Bisection method

$a_0 = 2, b_0 = 2.5$

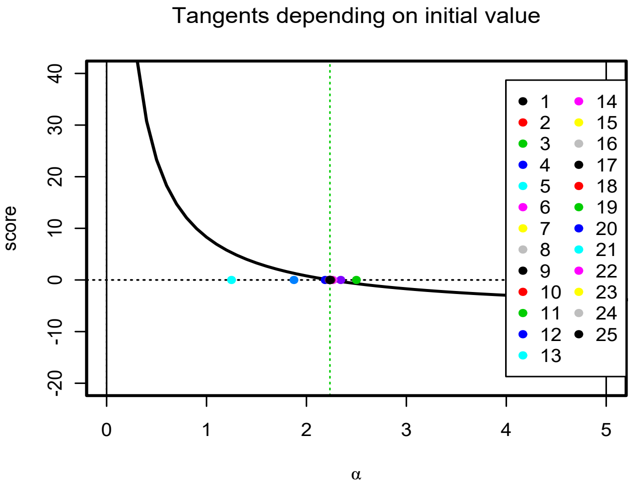
$\hat{\alpha}$ = 2.235084



Bisection method

$a_0 = \epsilon, b_0 = 5$

$\hat{\alpha}$ = 2.235082



NEWTON-RAPHSON

Newton-Raphson Method

Theorem

If

- $f \in \mathcal{C}^2$
- f'' *is continuous*
- x^* *is a simple root of* f

then

- $\exists$ *a neighborhood of* x^*

for which Newton-Raphson method ***converges to*** x^* *when started* ***from any*** x_0 ***in that neighborhood****.*

Newton-Raphson Method

TO NOTE: Only in the neighborhood.

Proof.

The main formulation of Newton-Raphson method is based on the Taylor approximation of the function, in a local point of the optimum value:

$$\begin{aligned} 0 &= f(x^*) \\ &= f(x_0) + (x^* - x_0) f'(x_0) + \frac{(x^* - x_0)^2}{2} f''(\xi_0) \end{aligned}$$

with $\xi_0 \in I(x_0, x^*)$.
Isolating the optimum point x^*, we can see that the update element (with the error of approximation value not used) is:

$$h = -\frac{f(x_0)}{f'(x_0)} \quad ; x_{n+1} = x_n - \frac{f(x_0)}{f'(x_0)}$$

The right part is the update rule of the method. □

Newton-Raphson Method

Proof.

For convergence criteria we need also the second order element:

$$x_n + h_n - x^* = (x^* - x_n)^2 \frac{f''(\xi_n)}{2f'(x_n)}$$

and we are able to define the error:

$$\epsilon_{n+1} \approx (\epsilon_n)^2 \frac{f''(\xi_n)}{2f'(x_n)}$$

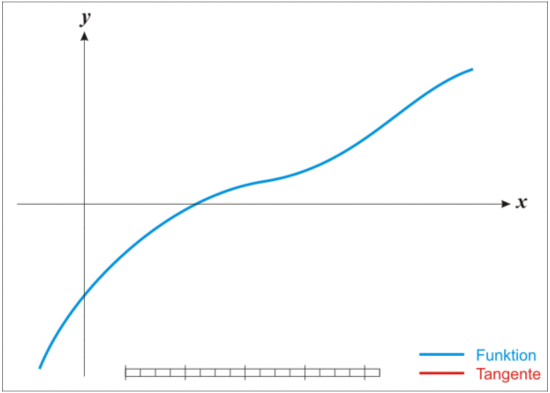
Moving the left parameter we obtain

$$\frac{\epsilon_{n+1}}{(\epsilon_n)^2} \approx \frac{f''(\xi_n)}{2f'(x_n)}$$

So, in case it converges, it has a quadratic (p= 2) speed of convergence. ($M = \max_{n \geq 0} \frac{f''(\xi_n)}{2f'(x_n)}$). $\square$

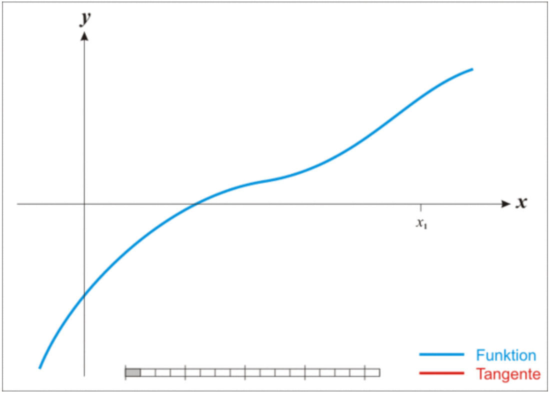
Newton Raphson Method

Algorithm sketch

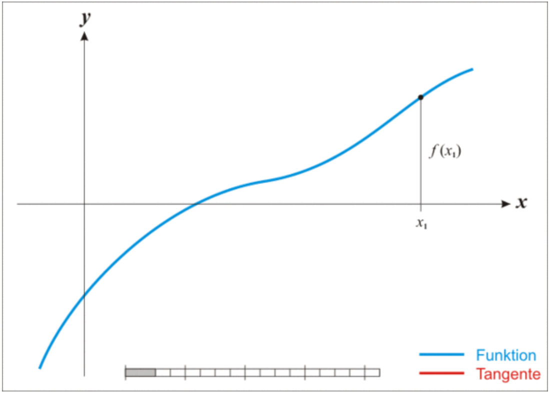


Newton Raphson Method

Algorithm sketch

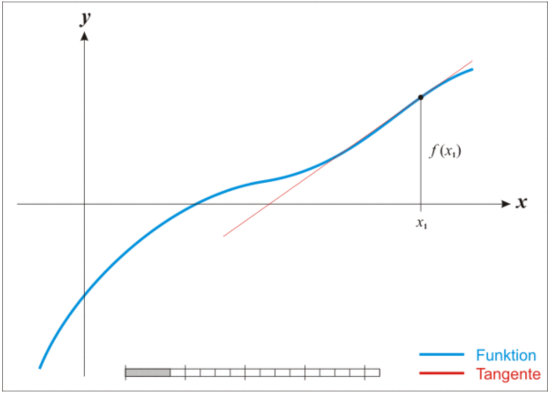


Newton Raphson Method
Algorithm sketch



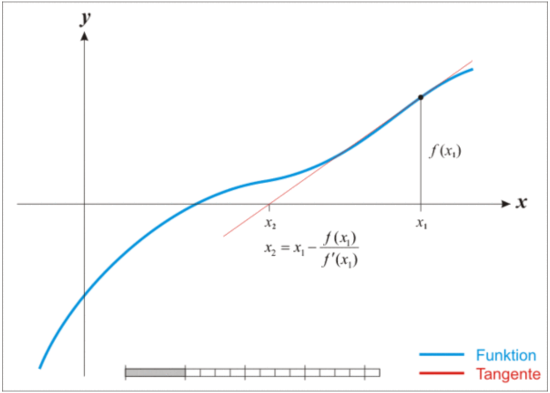
Newton Raphson Method

Algorithm sketch



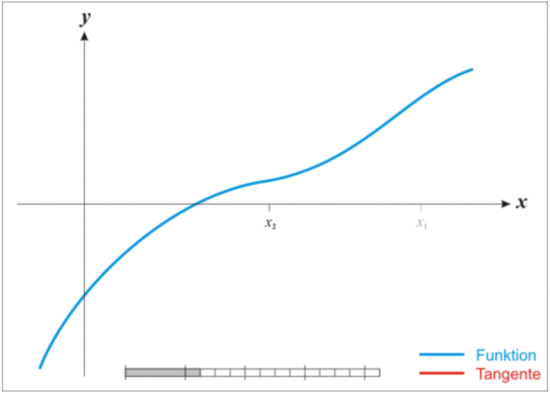
Newton Raphson Method

Algorithm sketch



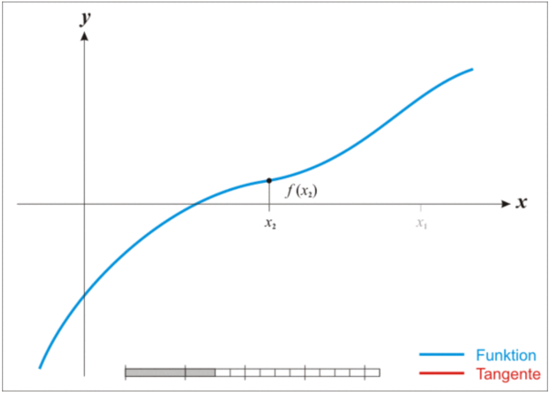
Newton Raphson Method

Algorithm sketch



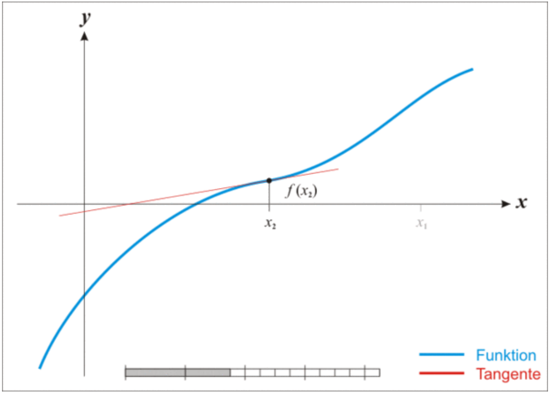
Newton Raphson Method

Algorithm sketch



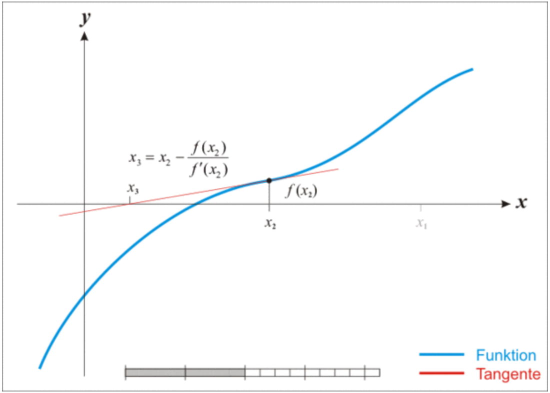
Newton Raphson Method

Algorithm sketch



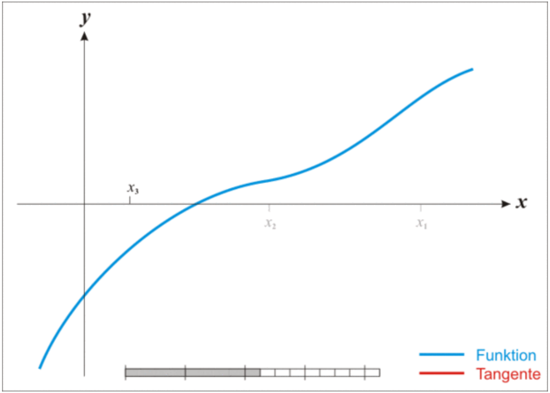
Newton Raphson Method

Algorithm sketch



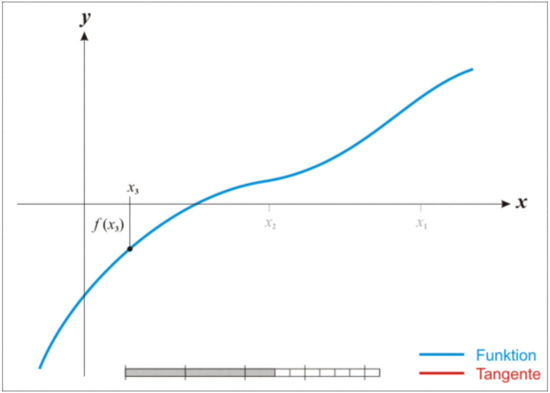
Newton Raphson Method

Algorithm sketch



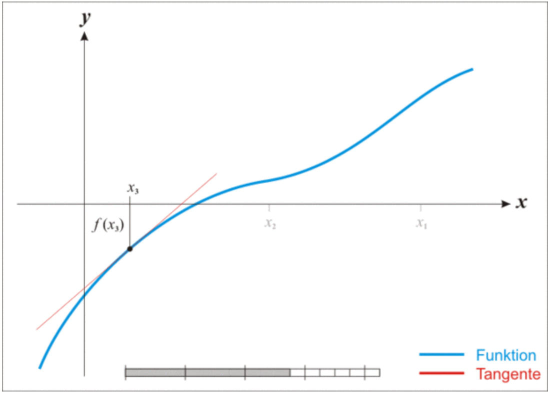
Newton Raphson Method

Algorithm sketch



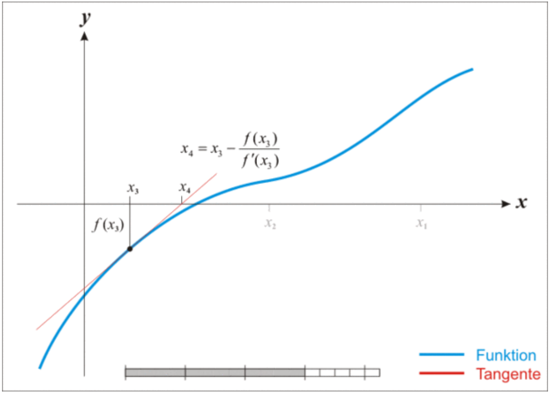
Newton Raphson Method

Algorithm sketch



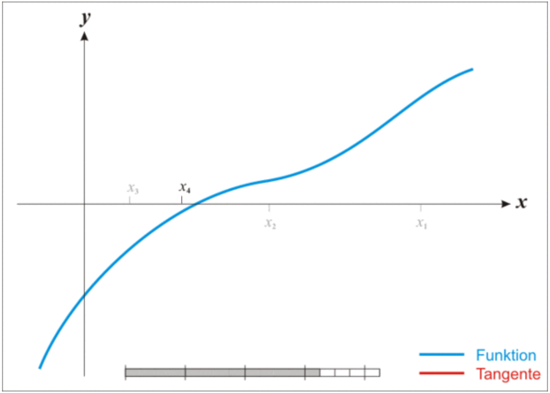
Newton Raphson Method

Algorithm sketch



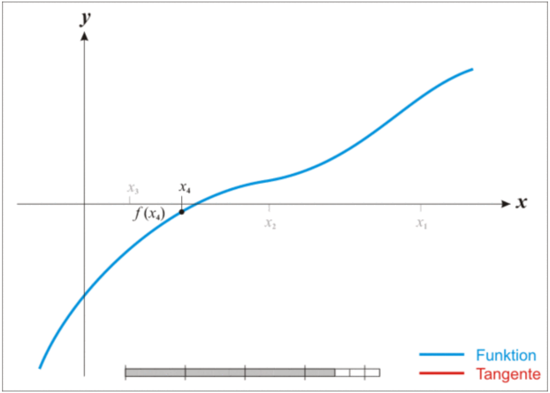
Newton Raphson Method

Algorithm sketch



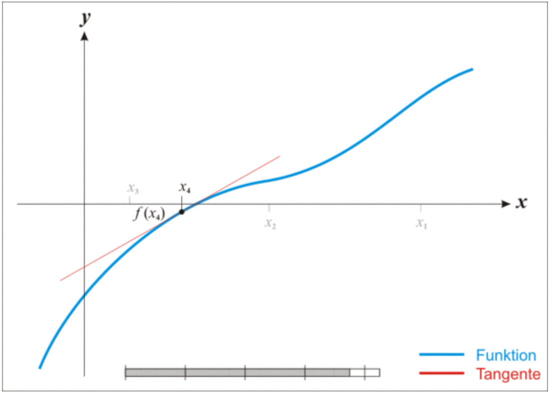
Newton Raphson Method

Algorithm sketch



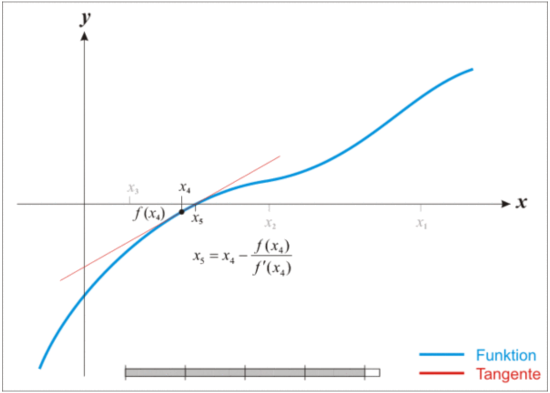
Newton Raphson Method

Algorithm sketch



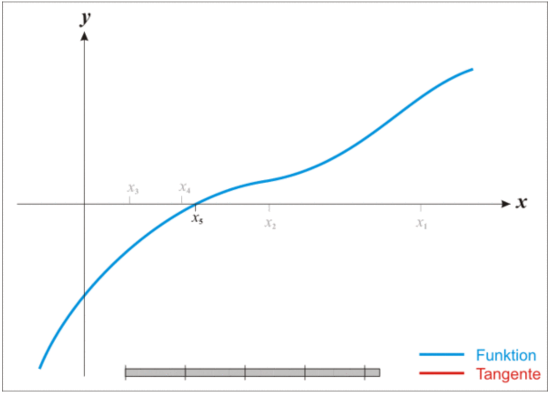
Newton Raphson Method

Algorithm sketch



Newton Raphson Method

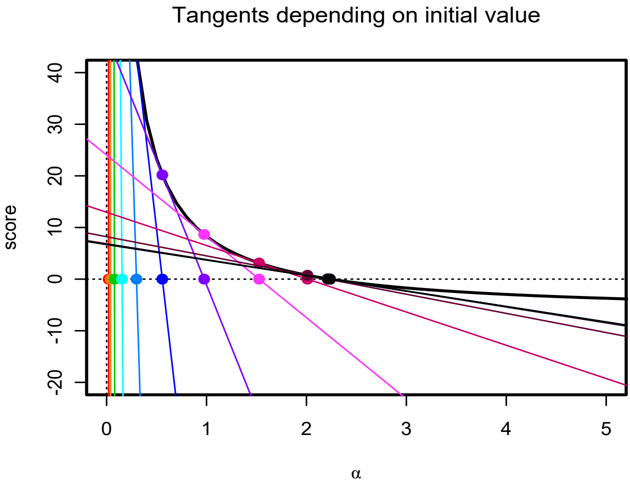
Algorithm sketch



Newton Raphson Method

$\alpha^{(0)} \approx \epsilon$

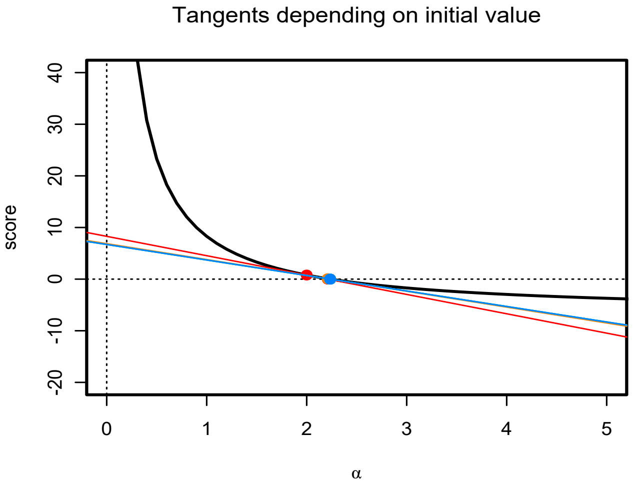
$\hat{\alpha} = 2.23508291$



Newton Raphson Method

$\alpha^{(0)} = 2$

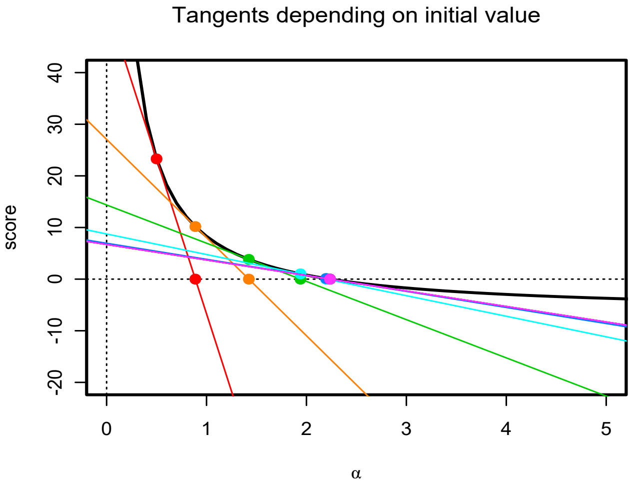
$\hat{\alpha} = 2.235083$



Newton Raphson Method

$\alpha^{(0)} = 0.5$

$\hat{\alpha} = 2.2350829$



Convergence Comprehension

https://www.youtube.com/watch?v=ImixIiH21HA

Third method

FISHER SCORING METHOD

Fisher Scoring Method

- Fisher Scoring Algorithm:

$$x_{n+1} = x_n + \frac{s(x_n)}{\mathcal{I}_n} = x_n + \frac{s(x_n)}{\frac{n}{\alpha^2}}$$

Where,

$$\mathcal{I}(\alpha) = -E\left[s'(\alpha; x)\right] = -E\left[-\frac{n}{\alpha^2}\right] = \frac{n}{\alpha^2}$$

- Newton Raphson Method:

$$x_{n+1} = x_n - \frac{s(x_n)}{s'(x_n)} = x_n - \frac{s(x_n)}{-\frac{n}{\alpha^2}} = x_n + \frac{s(x_n)}{\frac{n}{\alpha^2}}$$