Mathematics for Engineers

Pál Burai

Numerical mathematics

This work was supported by the construction EFOP-3.4.3-16-2016-00021. The project was supported by the European Union, co-financed by the European Social Fund.

Integers

Determine the largest representable integer with the `intmax` command.

```
intmax
ans =
int32
2147483647
2147483647+1
ans =
2.1475e+09
```

Remark

The set of representable numbers has upper- and lower bound (maybe it is large, but finite). Not all the numbers are representable between the bounds!

Floating point numbers

The form of nonzero floating point numbers

$$\pm a^k \left(\frac{m_1}{a} + \frac{m_2}{a^2} + \cdots \frac{m_t}{a^t} \right),$$ where k is the **exponent** or **characteristic** and $(m_1, \ldots, m_t)$ is the **mantissa**.

Storage of floating point numbers:

$$[\pm, k, m_1, \ldots, m_t], \qquad 1 \leq m_1 \leq a-1, \quad 0 \leq m_i \leq a-1, \; i = 2, \ldots, t$$

If $k_{(-)} \leq k \leq k_{(+)}$, then the largest representable floating point number is

$$M_\infty = a^{k_{(+)}} \left(\frac{a-1}{a} + \frac{a-1}{a^2} + \cdots + \frac{a-1}{a^t} \right) = a^{k_{(+)}}(1 - a^{-t}).$$

The smallest representable floating point number is $-M_\infty$.
The floating point number nearest to zero: $\varepsilon_0 = a^{k_{(-)}-1}$
Machine epsilon: $\varepsilon_1 = a^{1-t}$, where $1 + \varepsilon_1$ is the subsequent representable floating point number after 1.

Floating Point Arithmetic, Rounding

Rounding of the input:
Let $|x| \leq M_\infty$ an let $\tilde{x}$ be the floating point number assigned to x, then

$$\tilde{x} = \begin{cases} 0, & \text{if } |x| < \varepsilon_0, \\ \text{the floating point number nearest to } x, & \text{if } \varepsilon_0 \leq |x| \leq M_\infty. \end{cases}$$

Then the rounding error is

$$|x - \tilde{x}| \leq \begin{cases} \varepsilon_0, & \text{if } |x| < \varepsilon_0, \\ \frac{1}{2}\varepsilon_1 |x|, & \text{if } \varepsilon_0 \leq |x| \leq M_\infty. \end{cases}$$

If truncation is used, that is to say, $\tilde{x}$ will be the nearest floating point number to x between x and zero, then the the rounding error will be ε_1 instead of $\frac{1}{2}\varepsilon_1$.

Floating Point Arithmetic, Rounding

Rounding errors occurring during arithmetic operations

Example

Let $a = 10,\ t = 3,\ k_- = -3,\ k_+ = 3$. Then $\varepsilon_1 = 0.01$.
Let us add $x = 3.45$ and $y = 0.567$. The exponents $(k_1 = 1,\ k_2 = 0)$ are different. It should have to be made them equal, so we shift the mantissa right by $k_1 - k_2 = 1$ digits:

	$+$	1	345
$+$	$+$	0	567
$=$	$+$	1	345
$+$	$+$	1	0567
$=$	$+$	1	4017

There is not enough digits to store the result because $t = 3$. After rounding we have the following result $\boxed{+ \mid 1 \mid 402}$ which is not exact.
We can derive the estimate for the rounding error:

$$|\text{exact} - \text{numerical}| = 0.003 = \varepsilon_1 * 0.3 = \frac{1}{2}\varepsilon_1 * 0.6 < \frac{1}{2}\varepsilon_1 * |\text{exact}|$$

Floating Point Arithmetic, Rounding

The previous estimate for the rounding error is true in general. It can be verified in a pretty similar way that the rounding error in the case of truncation is twice as much as in the case of rounding.
Let us denote with $\diamond$ one of the basic operations $+,\ -,\ *,\ /$ and with $\varepsilon_\diamond$ the **rounding error**, then

$$|\varepsilon_\diamond| \leq \varepsilon_1 * \begin{cases} 1 & \text{in the case of truncation} \\ \frac{1}{2} & \text{in the case of rounding.} \end{cases}$$

Important remark

The above estimate is not valid if the absolute value of the result is less than ε_0 (**underflow**) or its absolute value is greater than M_∞ (**overflow**).

The absolute value can be misleading without the magnitude of the input. This phenomena justifies the introduction of the **relative error**:

$$\left|\frac{\text{exact} - \text{numerical}}{\text{exact}}\right| = \left|\frac{\tilde{z} - z}{z}\right|$$

Accumulation of Errors

Addition

Suppose we have to add $n+1$ floating point numbers $(x_0, \dots, x_n)$, practically we add x_{k+1} to the kth partial sumS_k and then we apply truncation. The first partial sum is:

$$\tilde{S}_1 = \widetilde{x_0 + x_1} = (x_0 + x_1)(1 + \varepsilon_{01}) = S_1 + \varepsilon_{01} S_1, \text{ ahol } |\varepsilon_{01}| \le \varepsilon_1.$$

Repeating this process n times, and taking into account that the actual error is always less then or equal to the machine epsilon, we have the following estimate for the absolute error:

$$|\widetilde{S_n} - S_n| \le \varepsilon_1 \big(n|x_0| + n|x_1| + (n-1)|x_2| + \dots + 2|x_{n-1}| + |x_n|\big).$$

This shows that it is reasonable to start the addition with the smallest term (in absolute value).
For the relative error we get the estimate

$$\left| \frac{\tilde{S}_n - S_n}{S_n} \right| \le n\varepsilon_1 \text{ that is, if } n\varepsilon_1 \ge 1 \text{ the result is } \textbf{not acceptable}.$$

Accumulation of Errors

Multiplication

Quite similarly to the case of addition, the relative error of multiplication is

$$\left| \frac{\tilde{P}_n - P_n}{P_n} \right| \leq n\varepsilon_1.$$

This error is acceptable if

$$n\varepsilon_1 \leq \frac{1}{a + \frac{1}{2}},$$

where a is the base of the representation of numbers.

Linear system of equations, condition number

Norm of matrices

Let $A \in \mathscr{M}_{n\times n}$, then the quantities

$$\|A\|_1 := \max_j \sum_{i=1}^{n} |a_{ij}|, \qquad \|A\|_\infty := \max_i \sum_{j=1}^{n} |a_{ij}|$$

Linear system of equations, condition number

Norm of matrices

Let $A \in \mathscr{M}_{n\times n}$, then the quantities

$$\|A\|_1 := \max_j \sum_{i=1}^n |a_{ij}|, \qquad \|A\|_\infty := \max_i \sum_{j=1}^n |a_{ij}|$$

are called the **one norm** and **infinity norm** of the matrix A. Because of their calculation method, one norm is also called column sum norm, and infinity norm is called row sum norm.

Linear system of equations, condition number

Norm of matrices

Let $A \in \mathscr{M}_{n\times n}$, then the quantities

$$\|A\|_1 := \max_j \sum_{i=1}^{n} |a_{ij}|, \qquad \|A\|_\infty := \max_i \sum_{j=1}^{n} |a_{ij}|$$

are called the **one norm** and **infinity norm** of the matrix A. Because of their calculation method, one norm is also called column sum norm, and infinity norm is called row sum norm.

Let A be a regular matrix and b be a nonzero vector. Let's consider the linear system of equations $Ax = b$. Assume that the right-hand side vector b is perturbed with the error δb, and solve $Ax = b + \delta b$ instead of the original system. Then the solution will be$x + \delta x$ instead of x, and the relative error of the solution is

$$\frac{\|\delta x\|}{\|x\|} \leq \|A\| \cdot \|A^{-1}\| \cdot \frac{\|\delta b\|}{\|b\|}.$$

Linear system of equations, condition number

Norm of matrices

Let $A \in \mathscr{M}_{n\times n}$, then the quantities

$$\|A\|_1 := \max_j \sum_{i=1}^{n} |a_{ij}|, \qquad \|A\|_\infty := \max_i \sum_{j=1}^{n} |a_{ij}|$$

are called the **one norm** and **infinity norm** of the matrix A. Because of their calculation method, one norm is also called column sum norm, and infinity norm is called row sum norm.

Let A be a regular matrix and b be a nonzero vector. Let's consider the linear system of equations $Ax = b$. Assume that the right-hand side vector b is perturbed with the error δb, and solve $Ax = b + \delta b$ instead of the original system. Then the solution will be$x + \delta x$ instead of x, and the relative error of the solution is

$$\frac{\|\delta x\|}{\|x\|} \le \|A\| \cdot \|A^{-1}\| \cdot \frac{\|\delta b\|}{\|b\|}.$$

The number $\|A\| \cdot \|A^{-1}\|$ is called the **condition number** of the matrix$A$.

Linear system of equations, condition number

Properties of condition number

1. The condition number depends on the norm.

Linear system of equations, condition number

Properties of condition number

1. The condition number depends on the norm.
2. The condition number is greater then or equal to one.

Linear system of equations, condition number

Properties of condition number

1. The condition number depends on the norm.
2. The condition number is greater then or equal to one.
3. If the condition number is greater than or equal to $\frac{1}{\varepsilon_1}$, then the error can be unacceptable. In this case the matrix is called **ill-conditioned**.

Linear system of equations, condition number

Properties of condition number

1. The condition number depends on the norm.
2. The condition number is greater then or equal to one.
3. If the condition number is greater than or equal to $\frac{1}{\varepsilon_1}$, then the error can be unacceptable. In this case the matrix is called **ill-conditioned**.

If the matrix is perturbed, then the relative error of the solution can be estimated the following way:

$$\frac{\|\delta x\|}{\|x\|} \leq \frac{\kappa}{1-\kappa}, \qquad \text{ahol } \kappa = \leq \|A\| \cdot \|A^{-1}\| \cdot \frac{\|\delta A\|}{\|A\|}.$$

Least square approximation

Linear case

Given m points on the plane $(t_1, f_1), \dots (t_m, f_m)$. We are looking for the straight line which "fits" the best in some sense to these data.

Least square approximation

Linear case

Given m points on the plane $(t_1, f_1), \dots (t_m, f_m)$. We are looking for the straight line which "fits" the best in some sense to these data.More precisely, we would like to find a function $F(t) = a + bt$ such that its difference from the given points is minimal in the least square sense.

$$\sum_{i=1}^{m} (F(t_i) - f_i)^2 = \sum_{i=1}^{m} (a + bt_i - f_i)^2.$$

Least square approximation

Linear case

Given m points on the plane $(t_1, f_1), \dots (t_m, f_m)$. We are looking for the straight line which "fits" the best in some sense to these data.More precisely, we would like to find a function $F(t) = a + bt$ such that its difference from the given points is minimal in the least square sense.

$$\sum_{i=1}^{m}(F(t_i) - f_i)^2 = \sum_{i=1}^{m}(a + bt_i - f_i)^2.$$

This entails the solution of the following system:

$$\begin{bmatrix} m & \sum_{i=1}^m t_i \\ \sum_{i=1}^m t_i & \sum_{i=1}^m t_i^2 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^m f_i \\ \sum_{i=1}^m t_i f_i \end{bmatrix}$$

The augmented matrix is $(A^T A | A^T f)$, where

$$A^T = \begin{bmatrix} 1 & \cdots & 1 \\ t_1 & \cdots & t_m \end{bmatrix}, \quad f^T = \begin{bmatrix} f_1 & \cdots & f_m \end{bmatrix}.$$

Least square approximation

General case

Let $\varphi_1, \dots, \varphi_n$ be a given system of functions with n elements. We are looking for n parameters $x_1, \dots, x_n$ such that the difference $\sum_{i=1}^m (F(t_i) - f_i)^2$ is minimal, where $F(t) = x_1\varphi_1(t) + \dots + x_n\varphi_n(t)$. In detail, we should solve the following minimization problem:

$$\min_{x_1,\dots,x_n} \left\{ \sum_{i=1}^{m} \left(\sum_{j=1}^{n} x_j \varphi_j(t_i) - f_i \right)^2 \right\}.$$

Least square approximation

General case

Let $\varphi_1, \dots, \varphi_n$ be a given system of functions with n elements. We are looking for n parameters $x_1, \dots, x_n$ such that the difference $\sum_{i=1}^m (F(t_i) - f_i)^2$ is minimal, where $F(t) = x_1\varphi_1(t) + \cdots + x_n\varphi_n(t)$. In detail, we should solve the following minimization problem:

$$\min_{x_1,\dots,x_n} \left\{ \sum_{i=1}^m \left(\sum_{j=1}^n x_j \varphi_j(t_i) - f_i \right)^2 \right\}.$$

Like in the linear case, this requires the solution of the linear system below.

$$\underbrace{\begin{bmatrix} \varphi_1(t_1) & \cdots & \varphi_1(t_m) \\ \vdots & \dots & \vdots \\ \varphi_n(t_1) & \cdots & \varphi_n(t_m) \end{bmatrix}}_{A^T} \underbrace{\begin{bmatrix} \varphi_1(t_1) & \cdots & \varphi_n(t_1) \\ \vdots & \dots & \vdots \\ \varphi_1(t_m) & \cdots & \varphi_n(t_m) \end{bmatrix}}_{A} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \underbrace{\begin{bmatrix} \varphi_1(t_1) & \cdots & \varphi_1(t_m) \\ \vdots & \dots & \vdots \\ \varphi_n(t_1) & \cdots & \varphi_n(t_m) \end{bmatrix}}_{A^T} \underbrace{\begin{bmatrix} f_1 \\ \vdots \\ f_m \end{bmatrix}}_{f}$$

Least square approximation

Using the previous notations we get the equation

$$A^T A x = A^T f$$

which is called the **Gaussian normal equation**.

Remark

The normal equation always has a solution, which is unique if the columns of A are linearly independent. If they are dependent, then the model is complicated. We can increase the number of rows, or decrease the number of columns to rid of the dependency of the columns of A.

Lagrangian interpolation

Denote by P_n the space of polynomials with degree at most n, that is, $p \in P_n$ if and only if $p\colon \mathbb{R} \to \mathbb{R}$ and

$$p(x) = a_0 + a_1 x_1 + \cdots + a_n x^n,$$

where $a_i \in \mathbb{R},\ i = 1, \ldots, n$ are given.

The interpolation problem

Let $x_0, \ldots, x_n, f_0, \ldots, f_n$ be given real numbers such that $x_i \neq x_j$, if $i \neq j$. Find the polynomial $p \in P_n$ with the property

$$p(x_i) = f_i, \qquad i = 0, \ldots, n.$$

Lagrangian interpolation

The polynomial

$$\ell_j(x) := \prod_{\substack{i=0 \\ i\neq j}}^{n} \frac{x - x_i}{x_j - x_i}$$

is said to be the j**th Lagrangian basic polynomial**.

The solution of the Lagrangian interpolation problem

Using the previous notations the polynomial

$$\mathscr{L}_n(x) := \sum_{i=1}^{n} f_i \ell_i(x)$$

fulfils all the requirements of the Lagrangian interpolation problem.

Theorem

The solution of the Lagrangian interpolation problem $\mathscr{L}_n$ is unique.

Lagrangian interpolation

Newton's interpolation polynomials

Besides the previous notations let $\mathscr{N}_k$ be the polynomial with degree k $0 < k \leq n$ which fits to the data $\{x_i, f_i\}_{i=0}^{k}$. Then $\mathscr{L}_k = \mathscr{N}_k$. We use a different method like in the case of Lagrange polynomials. Let

$$\mathscr{N}_0(x) = \mathscr{N}_0(x_0) := b_0 = f_0.$$

Find now $\mathscr{N}_1(x) = \mathscr{N}_0(x) + b_1(x - x_0)$. We have

$$\mathscr{N}_1(x) = f_0 + \frac{f_1 - f_0}{x_1 - x_0}(x - x_0).$$

In the kth step we would like to find

$$\mathscr{N}_k(x) = \mathscr{N}_{k-1}(x) + b_k\omega_k(x), \qquad k = 1, \ldots, n,$$

where

$$\omega_0(x) \equiv 1, \quad \omega_k = \prod_{j=0}^{k-1}(x - x_j), \quad k = 1, \ldots, n.$$

Lagrangian interpolation

Newton's recursion

The required polynomial is

$$\mathscr{N}_n(x) = \sum_{i=0}^{n} b_i \omega_i(x).$$

Calculation of the coefficients of Newton's polynomial

$$\mathscr{N}_k(x) = \mathscr{L}_k(x) = \sum_{i=0}^{k} f_i \prod_{\substack{j=0\\ j\neq i}}^{k} \frac{x - x_j}{x_i - x_j} = x^k \sum_{i=0}^{k} f_i \prod_{\substack{j=0\\ j\neq i}}^{k} \frac{1}{x_i - x_j} + Q(x),$$

where the degree of $Q(x)$ is strictly less than k, so

$$b_k = \sum_{i=0}^{k} f_i \prod_{\substack{j=0\\ j\neq i}}^{k} \frac{1}{x_i - x_j} =: [x_0, \dots, x_k]f.$$

Lagrangian interpolation

The expression $[x_0, \dots, x_k]f$ is called the k**th order divided difference** of f with respect to the base points $x_0, \dots, x_k$. For $0 \le m < k \le n$ we get

$$[x_m, \dots, x_k]f = \frac{[x_{m+1}, \dots, x_k]f - [x_m, \dots, x_{k-1}]f}{x_k - x_m}.$$

Using the above mentioned formula we can calculate the coefficients with the scheme:

$$\begin{tikzcd}
x_0 & {[x_0]f} \arrow[dr] & & & & \\
 & & {[x_0, x_1]f} \arrow[dr] & & & \\
x_1 & {[x_1]f} \arrow[ur] \arrow[dr] & & {[x_0, x_1, x_2]f} \arrow[dr] & & \\
 & & {[x_1, x_2]f} \arrow[ur] & & \cdots \arrow[ddr] & \\
x_2 & {[x_2]f} \arrow[ur] & & & & \\
\vdots & \vdots & & & & {[x_0, \dots, x_n]f} \\
x_{n-1} & {[x_{n-1}]f} \arrow[dr] & & & & \\
 & & \cdots & & \cdots \arrow[uur] & \\
x_n & {[x_n]f} \arrow[ur] & & & &
\end{tikzcd}$$

Error of Lagrangian interpolation

Theorem

Assume that f is $n+1$ times continuously differentiable on the interval $[a,b]$, $a := x_0, b := x_n$. Then for all $x \in [a,b]$ there is a $\xi(x) \in [a,b]$ such that

$$f(x) - \mathscr{L}_n(x) = \frac{f^{(n+1)}(\xi(x))}{(n+1)!}\omega_n(x),$$

Corollary

With the assumption of the previous theorem, if $\max_{x\in[a,b]} |f^{(n+1)}(x)| \le M_{n+1}$, *then*

$$|f(x) - \mathscr{L}_n(x)| \le \frac{M_{n+1}}{(n+1)!}|\omega_n(x)|, \qquad x \in [a,b].$$

Hermitian interpolation

Find the polynomial $\mathscr{H}$ for which

$$\mathscr{H}^{(j)}(x_i) = f_{ij}, \qquad i = 0, \dots, n, \quad j = 0, \dots, m_i - 1,$$

where x_is and f_{ij}s are given real numbers a, m_is are given natural numbers, and $\mathscr{H}^{(j)}$ denotes the jth derivative of $\mathscr{H}$.

Theorem

Let $m := \sum_{i=0}^{n} m_i$, then the Hermitian interpolation problem has a unique solution in P_{m-1}.

Coefficients of the Hermite interpolation polynomial

We use a similar method as in the case of Lagrange interpolation. Namely, a modified version of Newton's iteration. In the scheme each node x_i appears m_i times as well as f_{i0}. Let us write f_{i1} in the third column of the scheme instead of $0/0$ and the first order divided differences otherwise. In a similar way, instead of $0/0$ in the fourth column we write $\frac{1}{2} f_{i2}$, and so on.

Piecewise polynomial interpolation

The problem

The interpolation polynomial can behave hectic near the endpoints in the previous models. To avoid this phenomena it is reasonable to approximate the data in every subintervals respectively. We can add smoothness constraints at the nodes.

Piecewise linear approximation

We join the points with a polygonal line. We require only continuity in this case. The exact form of the affine function on the *i*th interval is:

$$\varphi_i(x) = f_i + \frac{(f_{i+1} - f_i)}{(x_{i+1} - x_i)}(x - x_i), \quad x \in [x_i, x_{i+1}], \quad i = 1, \dots, n.$$

For higher order approximation it is needed to know the values of the derivatives up to the given order. From these we can construct the piecewise Hermite interpolation.
Another possibility is to produce the piecewise interpolation on the whole interval simultaneously.

Numerical integration, Elementary quadrature formulae

Let $[x_0, x_1]$ be a proper interval and $f\colon [x_0, x_1] \to \mathbb{R}$ be a Riemann integrable function.

The midpoint rule or rectangle rule

$$\int_{x_0}^{x_1} f(x)\,\mathrm{dx} \approx (x_1 - x_0) f\left(\frac{x_0 + x_1}{2}\right) = h_1 f(m_1).$$

Trapezoidal rule

$$\int_{x_0}^{x_1} f(x)\,\mathrm{dx} \approx \frac{(x_1 - x_0)}{2}\left(f(x_0) + f(x_1)\right) = \frac{h_1}{2}\left(f(x_0) + f(x_1)\right).$$

Simpson's rule

$$\int_{x_0}^{x_1} f(x)\,\mathrm{dx} \approx \frac{(x_1 - x_0)}{6}\left(f(x_0) + 4f\left(\frac{x_0 + x_1}{2}\right) + f(x_1)\right) =$$

$$\frac{h_1}{6}\left(f(x_0) + 4f\left(m_1\right) + f(x_1)\right).$$

Numerical integration

A general quadrature rule

Let $[a, b]$ be a proper interval and $f : [a, b] \to \mathbb{R}$ be a Riemann integrable function, then we look for the approximate value of the integral in the following form:

$$\int_a^b f(x)\,\mathrm{dx} \approx \sum_{i=0}^{n} a_i f(x_i),$$

where a_i are given weights.

We can get a quadrature rule like this if we integrate the Lagrangian interpolation polynomial.

$$\int_a^b f(x)\,\mathrm{dx} \approx \int_a^b \mathscr{L}_n(x)\,\mathrm{dx} = \sum_{i=0}^{n} f(x_i) \int_a^b \ell_i(x)\,\mathrm{dx}\,.$$

This type of quadrature rules (if the nodes are equidistant) are called **Newton-Cotes formulae**.

Numerical integration

Error of the Newton-Cotes formulae

$$\left|\int_a^b (f(x) - \mathscr{L}_n(x))\,\mathrm{dx}\right| = \left|\int_a^b \frac{f^{(n+1)}(\xi(x))}{(n+1)!}\omega_n(x)\,\mathrm{dx}\right| \le$$

$$\frac{M_{n+1}}{(n+1)!}\int_a^b |\omega_n(x)|\,\mathrm{dx} \le \frac{M_{n+1}}{(n+1)!}(b-a)^{n+1}.$$

Nonlinear equations, bisection method

Consider a continuous function $f\colon [a,b] \to \mathbb{R}$ and assume that $f(a)f(b) < 0$, that is, the signs of the values are different at the ends of the interval. According to Bolzano's theorem every continuous function has the Darboux property (intermediate value property), so, there is a $\bar{x} \in]a,b[$, such that

$$f(\bar{x}) = 0.$$

Nonlinear equations, bisection method

Consider a continuous function $f\colon [a,b] \to \mathbb{R}$ and assume that $f(a)f(b) < 0$, that is, the signs of the values are different at the ends of the interval. According to Bolzano's theorem every continuous function has the Darboux property (intermediate value property), so, there is a $\bar{x} \in]a,b[$, such that

$$f(\bar{x}) = 0.$$

Theoretical algorithm

Initialization: $x_0 := a,\ y_0 := a,\ m_0 := \frac{x_0+y_0}{2},\ k := 0.$

- If $f(m_k) = 0$, then STOP, otherwise go to the second step.
- If $f(x_k)f(m_k) < 0$, then let $x_{k+1} := x_k$ and $y_{k+1} := m_k$, otherwise $x_{k+1} := m_k$ and $y_{k+1} := y_k$, $k := k+1$. Go to the first step.

Nonlinear equations, bisection method

Consider a continuous function $f\colon [a,b] \to \mathbb{R}$ and assume that $f(a)f(b) < 0$, that is, the signs of the values are different at the ends of the interval. According to Bolzano's theorem every continuous function has the Darboux property (intermediate value property), so, there is a $\bar{x} \in]a,b[$, such that

$$f(\bar{x}) = 0.$$

Theoretical algorithm

Initialization: $x_0 := a,\ y_0 := a,\ m_0 := \frac{x_0+y_0}{2},\ k := 0$.

- If $f(m_k) = 0$, then STOP, otherwise go to the second step.
- If $f(x_k)f(m_k) < 0$, then let $x_{k+1} := x_k$ and $y_{k+1} := m_k$, otherwise $x_{k+1} := m_k$ and $y_{k+1} := y_k$, $k := k+1$. Go to the first step.

It is reasonable to give the maximum number of iterations in practice in order to avoid an infinite loop. Moreover, it is more practical to require $|x_k - y_k| \leq \varepsilon$ instead of $f(m_k) = 0$. Then one should change the stopping rule.

Nonlinear equations, Banach iteration

Let $T\colon [a,b] \to [a,b]$ be a function which fulfils the inequality

$$|T(x) - T(y)| \leq q|x - y|, \qquad x, y \in [a,b]$$

with some constant $0 < q < 1$. Then T is said to be a q-**contraction**.

Nonlinear equations, Banach iteration

Let $T\colon [a,b] \to [a,b]$ be a function which fulfils the inequality

$$|T(x) - T(y)| \le q|x-y|, \qquad x, y \in [a,b]$$

with some constant $0 < q < 1$. Then T is said to be a q-**contraction**.

Állítás (Banach)

Besides the above mentioned assumptions, there is a $\bar{x} \in [a,b]$ such that

$$T(\bar{x}) = \bar{x}.$$

Moreover, for an arbitrary $x \in [a,b]$ the iteration $x_0 = x$, $x_{k+1} = T(x_k)$ is convergent, and

$$\lim_{k\to\infty} x_k = \bar{x}.$$

Nonlinear equations, Banach iteration

Let $T\colon [a,b] \to [a,b]$ be a function which fulfils the inequality

$$|T(x) - T(y)| \le q|x-y|, \qquad x, y \in [a,b]$$

with some constant $0 < q < 1$. Then T is said to be a q-**contraction**.

Állítás (Banach)

Besides the above mentioned assumptions, there is a $\bar{x} \in [a,b]$ such that

$$T(\bar{x}) = \bar{x}.$$

Moreover, for an arbitrary $x \in [a,b]$ the iteration $x_0 = x$, $x_{k+1} = T(x_k)$ is convergent, and

$$\lim_{k\to\infty} x_k = \bar{x}.$$

The $\bar{x}$ with the previous property is called a **fixed point of** T.

Nonlinear equations, Banach iteration

Let $T\colon [a,b] \to [a,b]$ be a function which fulfils the inequality

$$|T(x) - T(y)| \le q|x-y|, \qquad x, y \in [a,b]$$

with some constant $0 < q < 1$. Then T is said to be a q-**contraction**.

Állítás (Banach)

Besides the above mentioned assumptions, there is a $\bar{x} \in [a,b]$ *such that*

$$T(\bar{x}) = \bar{x}.$$

Moreover, for an arbitrary $x \in [a,b]$ *the iteration* $x_0 = x$, $x_{k+1} = T(x_k)$ *is convergent, and*

$$\lim_{k\to\infty} x_k = \bar{x}.$$

The $\bar{x}$ with the previous property is called a **fixed point of** T. One can look for the fixed point of the function $T(x) = x - \omega f(x)$ instead of a root of the function $f\colon [a,b] \to \mathbb{R}$ if there is a parameter ω for which T will be a contraction.

Nonlinear equations

Newton iteration

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

Nonlinear equations

Newton iteration

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

- The bisection method and the Banach iteration are globally convergent.

Nonlinear equations

Newton iteration

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

- The bisection method and the Banach iteration are globally convergent.
- The bisection method and the Banach iteration can be applied for non-differentiable functions.

Nonlinear equations

Newton iteration

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

- The bisection method and the Banach iteration are globally convergent.
- The bisection method and the Banach iteration can be applied for non-differentiable functions.
- The speed of convergence of the bisection method and the Banach iteration is slower than the speed of convergence of Newton's method.

Nonlinear equations

Newton iteration

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

- The bisection method and the Banach iteration are globally convergent.
- The bisection method and the Banach iteration can be applied for non-differentiable functions.
- The speed of convergence of the bisection method and the Banach iteration is slower than the speed of convergence of Newton's method.
- The Newton iteration is convergent in general if the starting point x_0 is "close enough" to a root of the function. However, there are known global convergence theorems too for the Newton iteration.

Nonlinear equations

Newton iteration

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

- The bisection method and the Banach iteration are globally convergent.
- The bisection method and the Banach iteration can be applied for non-differentiable functions.
- The speed of convergence of the bisection method and the Banach iteration is slower than the speed of convergence of Newton's method.
- The Newton iteration is convergent in general if the starting point x_0 is "close enough" to a root of the function. However, there are known global convergence theorems too for the Newton iteration.
- For the Newton iteration differentiability is needed, to ensure the convergence two times differentiability is needed.

Exercises

- Find an approximate value of $\sqrt{2}$ using the introduced three methods!

Exercises

- Find an approximate value of $\sqrt{2}$ using the introduced three methods!
- Let us approximate the root of the equation $e^x = 3x$ on the interval $[0, 1]$ using the introduced three methods!

Exercises

- Find an approximate value of $\sqrt{2}$ using the introduced three methods!
- Let us approximate the root of the equation $e^x = 3x$ on the interval $[0, 1]$ using the introduced three methods!

Exercises

- Find an approximate value of $\sqrt{2}$ using the introduced three methods!
- Let us approximate the root of the equation $e^x = 3x$ on the interval $[0, 1]$ using the introduced three methods!

Looking for a stationary point

If a function is differentiable and it has a local extreme value at $\bar{x}$, then the derivative is zero at the same point. So, the introduced methods can be applied to solve optimization problems numerically.

Exercises

- Find an approximate value of $\sqrt{2}$ using the introduced three methods!
- Let us approximate the root of the equation $e^x = 3x$ on the interval $[0, 1]$ using the introduced three methods!

Looking for a stationary point

If a function is differentiable and it has a local extreme value at $\bar{x}$, then the derivative is zero at the same point. So, the introduced methods can be applied to solve optimization problems numerically.

Exercise

Let us approximate a stationary point of the function $f(x) = (x-1)^2(x+1)^2$.