Numerical methods

Jiří Zelinka

Autumn 2021, lecture 2

Solving of nonlinear equations

Equation

$$f(x) = 0,$$

$x \in I = [a, b]$, f is continuous real function

$\hat{x} \in I$ – solution, root of f.

$f(a) \cdot f(b) \leq 0 \Rightarrow$ there exists a solution $\hat{x} \in I$

Iterative process:
We create sequence $(x_k)_{k=0}^{\infty}$, $x_k \to \hat{x}$.
$(x_k)_{k=0}^{\infty}$: iterative sequence.

Fixed point iteration method

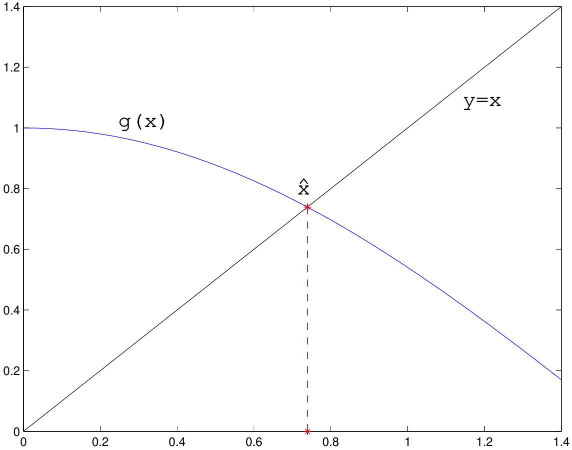
- Equation
$$x = g(x)$$
- g continuous on $I = [a, b]$
- Solution $\hat{x}$ is called the **fixed point** of the function g

iterative process

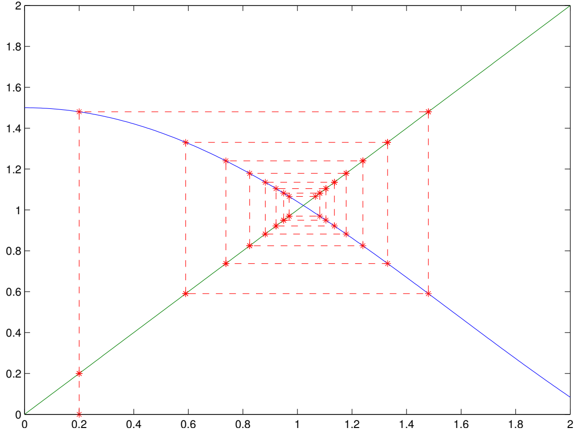
- Let us choose $x_0 \in I$ and $x_1 = g(x_0)$.
- Generally $x_{k+1} = g(x_k)$.
- Function g is called **iterative function**.

Geometric meaning

The fixed point $\hat{x}$ is the intersection of the function g and line $y = x$.



Graphical representation of the iterative process:



The existence and uniqueness of the fixed point

Theorem: If for the function g continuous on $I = [a, b]$ the following condition holds

$$\forall x \in I : g(x) \in I,$$

then there exists at least one fixed point $\hat{x} \in I$ of the function g.

Moreover, if there exists constant $0 \leq L < 1$ that $\forall x, y \in I$

$$|g(x) - g(y)| \leq L|x - y|,$$

then there exist one fixed point $\hat{x}$ and for any $x_0 \in I$ the iterative process given by formula

$$x_{k+1} = g(x_k)$$

converges to this fixed point.
Function g is called **contraction**.

Simpler condition: $|g'(x)| \leq L < 1, \forall x \in I$

Estimation of the error

$$|x_k - \hat{x}| \leq \frac{L^k}{1-L} |x_0 - x_1|$$

Creating of the iterative function

$$f(x) = 0 \qquad \to x = g(x)$$

Example

$$x^3 + 4x^2 - 10 = 0, \quad \hat{x} \in [1, 1.5]$$

Iterative functions: $g_1(x) = \sqrt{\frac{10}{x} - 4x}$,
$g_2(x) = \frac{1}{2}\sqrt{10 - x^3}$,
$g_3(x) = \sqrt{\frac{10}{4+x}}$

General procedure

$$g(x) = x - \frac{f(x)}{K}$$

More general procedure

$$g(x) = x - \frac{f(x)}{h(x)}$$

Example:

Computation of $\sqrt{a}$

$f(x) = x^2 - a \quad \Rightarrow \quad g(x) = x - \frac{x^2 - a}{K}$

Classification of the fixed points

The fixed point $\hat{x}$ of the function g is called

- **attracting** if $|g'(\hat{x})| < 1$, then the iterative process converges on some neighborhood of $\hat{x}$.
- **repelling** if $|g'(\hat{x})| > 1$, then the iterative process doesn't converge.

Demonstration

Newton's method (Newton–Raphson)

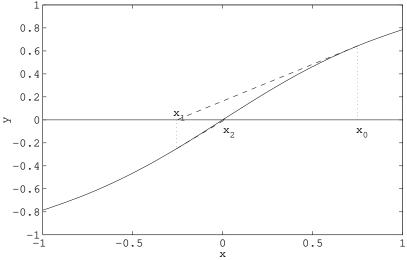
Let us return to the equation

$$f(x) = 0.$$

x_0 – initial iteration, x_1 – intersection of the tangent to f in x_0 the axis x.

x_{k+1} – intersection of the tangent to f in x_k the axis x

$\rightarrow$ **tangent method**



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

Iterative function:

$$g(x) = x - \frac{f(x)}{f'(x)}$$

Convergence

Theorem 1

If the second derivative of the function f is continuous in some neighborhood of $\hat{x}$, $f'(\hat{x}) \neq 0$ and the initial iteration x_0 is close enough to $\hat{x}$ then the Newton methods converges to the root $\hat{x}$.

Theorem 2

If the second derivative of the function f is continuous in some neighborhood of $\hat{x}$ and $f'(\hat{x}) \neq 0$ then $g'(\hat{x}) = 0$.
($g(x) = x - \frac{f(x)}{f'(x)}$ is the iterative function of Newton method.)

Example:

Computation of $\sqrt{a}$

$f(x) = x^2 - a$, $f'(x) = 2x$.

$$x_{k+1} = x_k - \frac{x_k^2 - a}{2x_k} = \frac{x_k^2 + a}{2x_k} = \frac{x_k}{2} + \frac{a}{2x_k}$$

Example to think about:

Computation of $\frac{1}{a}$ without division:

Numerical methods

Jiří Zelinka

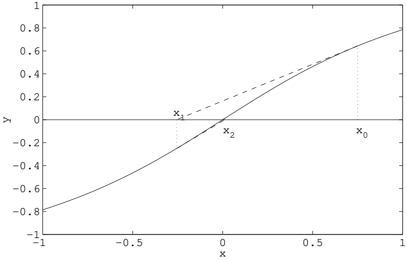
Autumn 2021, lecture 3

Newton's method

$$f(x) = 0.$$

x_0 – initial iteration, x_1 – intersection of the tangent to f in x_0 the axis x.

x_{k+1} – intersection of the tangent to f in x_k the axis x
$\rightarrow$ **tangent method**



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

Iterative function:

$$g(x) = x - \frac{f(x)}{f'(x)}$$

Convergence

Theorem 1

If the second derivative of the function f is continuous in some neighborhood of $\hat{x}$, $f'(\hat{x}) \neq 0$ and the initial iteration x_0 is close enough to $\hat{x}$ then the Newton's methods converges to the root $\hat{x}$.

Theorem 2

If the second derivative of the function f is continuous in some neighborhood of $\hat{x}$ and $f'(\hat{x}) \neq 0$ then $g'(\hat{x}) = 0$.
($g(x) = x - \frac{f(x)}{f'(x)}$ is the iterative function of Newton's method.)

Example 1:

Computation of $\sqrt{a}$

$f(x) = x^2 - a$, $f'(x) = 2x$.

$$x_{k+1} = x_k - \frac{x_k^2 - a}{2x_k} = \frac{x_k^2 + a}{2x_k} = \frac{x_k}{2} + \frac{a}{2x_k}$$

Example 2:

Computation of $\frac{1}{a}$ without division:

$f(x) = \frac{1}{x} - a \Rightarrow x_{k+1} = x_k(2 - ax_k)$

Fourier conditions

1. Let f has continuous the second derivative in $[a, b]$, $f(a) \cdot f(b) \leq 0$.
2. Let $\forall x \in [a, b] : f'(x) \neq 0$ and f'' doesn't change its sign in $[a, b]$

Let's choose $x_0 \in \{a, b\}$ such that $f(x_0) \cdot f'' \geq 0$. Then the sequence generated by Newton's method converges monotonously to $\hat{x}$.

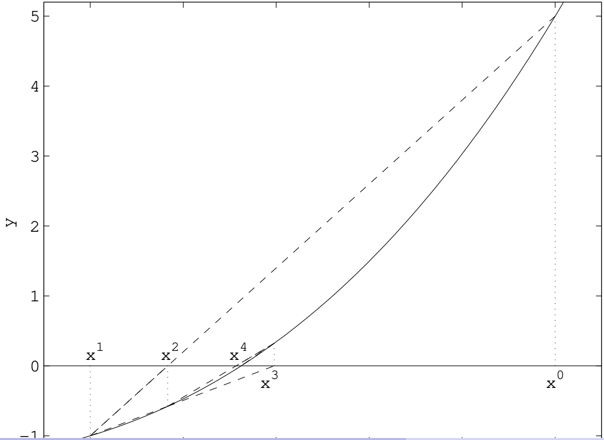
Examples:

Computation of $\sqrt[m]{a}$: $f()x) = x^m - a$ is convex increasing function $\Rightarrow\ f(x_0) > 0 \Rightarrow\ x_0 > \sqrt[m]{a}$.

Computation of $\frac{1}{a}$ without division: $f(x) = \frac{1}{x} - a$ is convex decreasing function $\Rightarrow f(x_0) > 0\ \Rightarrow\ 0 < x_0 < \frac{1}{a}$.

Secant methods

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



Two initial iterations x_0 and x_1 are required.

Example:

Computation of $\sqrt{a}$

$f(x) = x^2 - a$

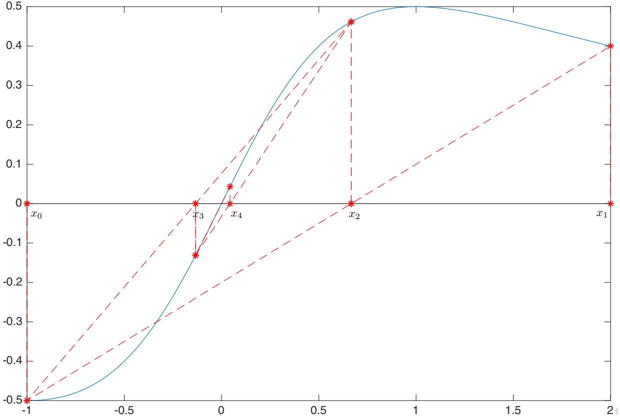
$$\begin{aligned}
x_{k+1} &= x_k - \frac{x_k - x_{k-1}}{(x_k^2 - a) - (x_{k-1}^2 - a)}(x_k^2 - a) \\
&= x_k - \frac{x_k - x_{k-1}}{x_k^2 - x_{k-1}^2}(x_k^2 - a) \\
&= x_k - \frac{x_k^2 - a}{x_k + x_{k-1}}
\end{aligned}$$

Method regula falsi (false position)

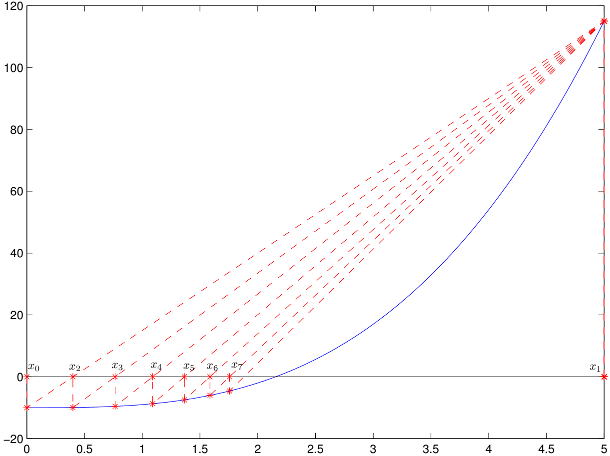
Idea: keep the opposite signs of the function in the border points of the subinterval (see also the bisection method).

$$x_{k+1} = x_k - \frac{x_k - x_s}{f(x_k) - f(x_s)} f(x_k), \qquad k = 1, 2, \ldots,$$

wher s is the largest index for which $f(x_k)f(x_s) \leq 0$.



Method regula falsi for convex or concave function:



- Convergence is monotone (maybe except for the beginning)
- $s \in \{0, 1\}$

Order of the convergence

Let $p \geq 1$, $x_k \to \hat{x}$, $e_k = x_k - \hat{x}$. If

$$\lim_{k\to\infty} \frac{|e_{k+1}|}{|e_k|^p} = C < \infty$$

then p is called the **order** of the convergence of the sequence $(x_k)_{k=0}^{\infty}$.

If the sequence $(x_k)_{k=0}^{\infty}$ is generated by the numerical methods, then p is the **order of the method**.

$p = 1 \ \to$ **linear method**

$p = 2 \ \to$ **quadratic method**

Theorem

Let the derivatives of the iteration function g be continuouns to order $q \geq p$. Then the order of the convergence of the sequence $(x_k)_{k=0}^{\infty}$ generated by the iteration process $x_{k+1} = g(x_k)$ is equal to p iff
$g(\hat{x}) = \hat{x}$, $g'(\hat{x}) = 0$, $g''(\hat{x}) = 0, \ldots$, $g^{(p-1)}(\hat{x}) = 0$, $g^{(p)}(\hat{x}) \neq 0$,

Orders of discussed methods:

Fixed point	1
Newton	2
Secant	$\frac{1+\sqrt{5}}{2} = \varphi \doteq 1.618$ (golden ratio)
Regula falsi	1

Example: determine the order of convergence of the geometric sequence

Multiple roots

Root of multiplicity M:
$f(\hat{x}) = 0,\ f'(\hat{x}) = 0, \ldots, f^{(M-1)}(\hat{x}) = 0,\ f^{(M)}(\hat{x}) \neq 0$

Modified Newton method: $x_{k+1} = x_k - M\frac{f(x_k)}{f'(x_k)}$
The order of the method is 2.

General method (unknown M):
Let $u(x) = \frac{f(x)}{f'(x)}$ and then we apply Newton method to the function u.

Numerical methods

Jiří Zelinka

Autumn 2021, lecture 4

Order of the convergence

Let $p \geq 1$, $x_k \to \hat{x}$, $e_k = x_k - \hat{x}$. If

$$\lim_{k\to\infty} \frac{|e_{k+1}|}{|e_k|^p} = C < \infty$$

then p is called the **order** of the convergence of the sequence $(x_k)_{k=0}^{\infty}$.

If the sequence $(x_k)_{k=0}^{\infty}$ is generated by the numerical methods, then p is the **order of the method**.

$p = 1 \ \to$ **linear method**

$p = 2 \ \to$ **quadratic method**

Theorem

Let the derivatives of the iteration function g be continuouns to order $q \geq p$. Then the order of the convergence of the sequence $(x_k)_{k=0}^{\infty}$ generated by the iteration process $x_{k+1} = g(x_k)$ is equal to p iff $g(\hat{x}) = \hat{x}$, $g'(\hat{x}) = 0$, $g''(\hat{x}) = 0, \dots, g^{(p-1)}(\hat{x}) = 0$, $g^{(p)}(\hat{x}) \neq 0$,

Orders of discussed methods:

Fixed point	1
Newton	2
Secant	$\frac{1+\sqrt{5}}{2} = \varphi \doteq 1.618$ (golden ratio)
Regula falsi	1

Acceleration of convergence – Aitken δ^2-method

Let $x_k \to \hat{x}$ linearly to $\hat{x}$, i.e. $e_{k+1} = Ce_k + o(1),\ |C| < 1$.
Let's mark $\varepsilon(x_k) = x_k - x_{k+1}$. Then

$$\varepsilon(x_k) = (x_k - \hat{x}) - (x_{k+1} - \hat{x}) = e_k - e_{k+1} = e_k(1 - C) + o(1)$$

$$\varepsilon(x_k) = x_k - x_{k+1}, \qquad \varepsilon(x_{k+1}) = x_{k+1} - x_{k+2}.$$

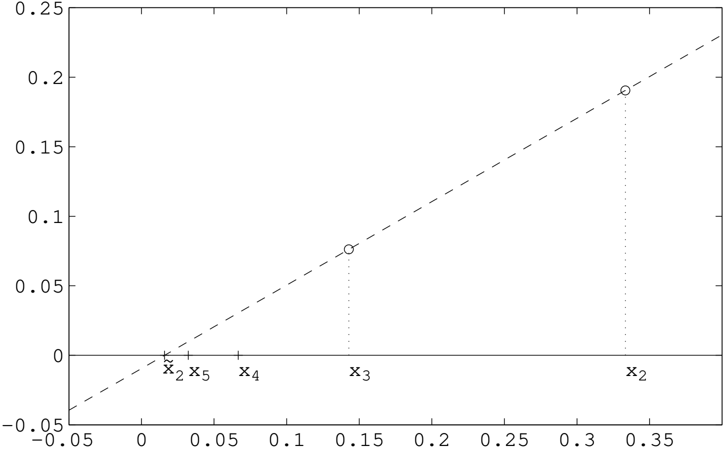
Points $[x_k, \varepsilon(x_k)]$, $[x_{k+1}, \varepsilon(x_{k+1})]$ are connected by the line. Its intersection with the axis x is the approximation of the limit of the sequence x_k.

The equation of the line:

$$y - \varepsilon(x_k) = \frac{\varepsilon(x_k) - \varepsilon(x_{k+1})}{x_k - x_{k+1}}(x - x_k)$$

The intersection with the axes x:

$$\tilde{x}_k = x_k - \frac{\varepsilon(x_k)(x_k - x_{k+1})}{\varepsilon(x_k) - \varepsilon(x_{k+1})} = x_k - \frac{(x_{k+1} - x_k)^2}{x_{k+2} - 2x_{k+1} + x_k}.$$


0.25
0.2
0.15
0.1
0.05
0
−0.05
−0.05
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
$\tilde{x}_2$
x_5
x_4
x_3
x_2

Theorem

Let $\{x_k\}_{k=0}^{\infty}$, $\lim\limits_{k\to\infty} x_k = \hat{x}$, $x_k \neq \hat{x}$, $k = 0, 1, 2, \ldots$, be a sequence and let

$$x_{k+1}-\hat{x} = (C+\gamma_k)(x_k-\hat{x}),\ k = 0, 1, 2, \ldots,\ |C| < 1,\ \lim_{k\to\infty} \gamma_k = 0.$$

Then

$$\tilde{x}_k = x_k - \frac{(x_{k+1} - x_k)^2}{x_{k+2} - 2x_{k+1} + x_k}$$

is defined for k enough large and

$$\lim_{k\to\infty} \frac{\tilde{x}_k - \hat{x}}{x_k - \hat{x}} = 0,$$

i.e., the sequence $\{\tilde{x}_k\}$ converges to $\hat{x}$ faster than $\{x_k\}$.

Alternative derivation:

If $x_k \to \hat{x}$ linearly and monotonically then

$$\frac{x_{k+1}-\hat{x}}{x_k-\hat{x}} \approx \frac{x_{k+2}-\hat{x}}{x_{k+1}-\hat{x}} \Rightarrow \hat{x} \approx x_k - \frac{(x_{k+1}-x_k)^2}{x_{k+2}-2x_{k+1}+x_k}$$

Ordinary differences:

$\Delta x_k = x_{k+1} - x_k$
$\Delta^2 x_k = \Delta x_{k+1} - \Delta x_k = x_{k+2} - 2x_{k+1} + x_k$
$\Delta^3 x_k = \Delta^2 x_{k+1} - \Delta^2 x_k$
$\vdots$

$$\tilde{x}_k = x_k - \frac{(\Delta x_k)^2}{\Delta^2 x_k}$$

Steffensen's method

Let g be iteration function for the equation $x = g(x)$. Let's put

$$y_k = g(x_k), \qquad z_k = g(y_k),$$

$$x_{k+1} = x_k - \frac{(y_k - x_k)^2}{z_k - 2y_k + x_k}.$$

This method id called **Steffensen's method** and it can be described by the iteration function φ:

$$x_{k+1} = \varphi(x_k),$$

for

$$\varphi(x) = x - \frac{(g(x) - x)^2}{g(g(x)) - 2g(x) + x} = \frac{xg(g(x)) - g^2(x)}{g(g(x)) - 2g(x) + x}.$$

Theorem 1

1. If $\varphi(\hat{x}) = \hat{x}$ then $g(\hat{x}) = \hat{x}$.
2. If $g(\hat{x}) = \hat{x}$, the derivative $g'(\hat{x})$ exits and $g'(\hat{x}) \neq 1$, then $\varphi(\hat{x}) = \hat{x}$.

Theorem 2

Let the derivatives of g be continuous up to order $p + 1$ in the neighborhood of the fixed point $\hat{x}$. Let the fixed point method defined by the process $x_{k+1} = g(x_k)$ is of order p.
The the Steffensen's method is of order $2p - 1$ for $p > 1$ and for $p = 1$ is its order at least 2 if $g'(\hat{x}) \neq 1$.

Examples

Π_n: space of polynomials of degree at most n swith real coefficients.
$P \in \Pi_n$:

$$P(x) = a_n x^n + a_{n-1} x^{n-1} \ldots + a_1 x + a_0.$$

Area containing all roots

Let

$$\begin{aligned} P(x) &= a_n x^n + a_{n-1} x^{n-1} \ldots + a_1 x + a_0, \\ A &= \max\left(|a_{n-1}|, \ldots, |a_0|\right), \\ B &= \max\left(|a_n|, \ldots, |a_1|\right), \end{aligned}$$

for $a_0 a_n \neq 0$. The following inequality is valid for all roots ξ of P:

$$\frac{1}{1 + \frac{B}{|a_0|}} \leq |\xi| \leq 1 + \frac{A}{|a_n|}.$$

Another estimates of upper bound for $|\xi|$

1. $|\xi_k| \leq \max\left\{1, \sum_{j=0}^{n-1}\left|\frac{a_j}{a_n}\right|\right\}$
2. $|\xi_k| \leq 2\max\left\{\left|\frac{a_{n-1}}{a_n}\right|, \sqrt{\left|\frac{a_{n-2}}{a_n}\right|}, \sqrt[3]{\left|\frac{a_{n-3}}{a_n}\right|}, \dots, \sqrt[n]{\left|\frac{a_0}{a_n}\right|}\right\}$
3. $|\xi_k| \leq \max\left\{\left|\frac{a_0}{a_n}\right|, 1+\left|\frac{a_1}{a_n}\right|, \dots, 1+\left|\frac{a_{n-1}}{a_n}\right|\right\}.$

Example:

$$\begin{aligned} P(x) &= (x-1)(x-2)(x-3)(x-4)(x-5) \\ &= x^5 - 15x^4 + 85x^3 - 225x^2 + 274x - 120 \end{aligned}$$

$$\begin{aligned} |\xi_k| &\le 1 + 274 = 275 \\ 1.\ |\xi_k| &\le \max\{1, 719\} = 719 \\ 2.\ |\xi_k| &\le 2\max\{30,\ 18.44,\ 12.16,\ 8.14,\ 5.21\} = 60 \\ 3.\ |\xi_k| &\le \max\{120,\ 275,\ 226,\ 86,\ 16\} = 275. \end{aligned}$$

The Double-step Newton's method for polynomials with all real roots

Newton's method – slow convergence for the largest root if the initial iteration is too large.

$$x_{k+1} = x_k - \frac{(x_k)^n + \dots}{n(x_k)^{n-1} + \dots} \approx x_k \left(1 - \frac{1}{n}\right)$$

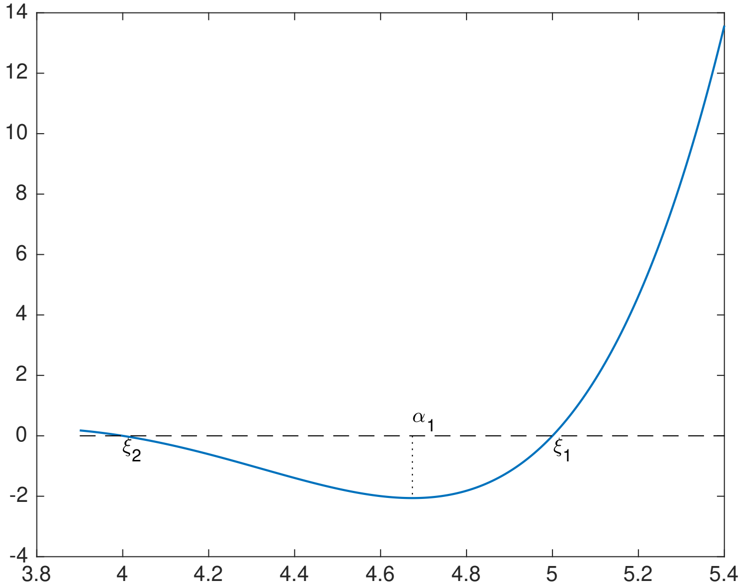
Example.

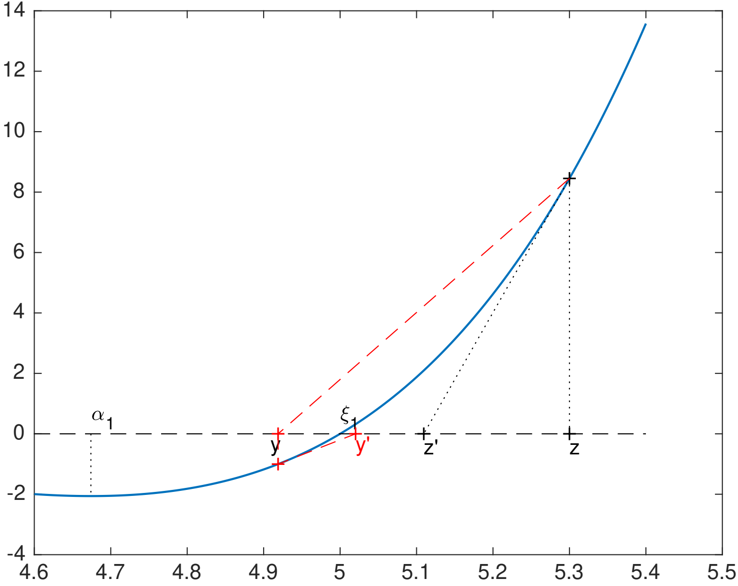
Overshooting theorem

Let P be a real polynomial of degree $n \geq 2$, all roots of which are real, $\xi_1 \geq \xi_2 \geq \cdots \geq \xi_n$. Let α_1 be the largest zero of P': $\xi_1 \geq \alpha_1 \geq \xi_2$. For $n = 2$, we require also that $\xi_1 > \xi_2$. Then for every $z > \xi_1$, the numbers

$$z' = z - \frac{P(z)}{P'(z)}, \qquad y = z - 2\frac{P(z)}{P'(z)}, \qquad y' = y - \frac{P(y)}{P'(y)}$$

are well defined and satisfy $\alpha_1 < y$ and $\xi_1 \leq y' \leq z'$ It is readily verified that $n = 2$ and $\xi_1 = \xi_2$ imply $y = \xi_1$ for any $z > \xi_1$.


14
12
10
8
6
4
2
0
-2
-4
3.8
4
4.2
4.4
4.6
4.8
5
5.2
5.4
α_1
ξ_2
ξ_1



14
12
10
8
6
4
2
0
-2
-4
4.6
4.7
4.8
4.9
5
5.1
5.2
5.3
5.4
5.5
α_1
ξ_1
y
y'
z'
z

Algorithm of the Double-step Newton's method:

1. Choose $x_0 > \xi_1$.
2. Evaluate $P_0 = P(x_0)$.
3. $x_1 = x_0 - 2\frac{P(x_0)}{P'(x_0)}$.
4. While $P(x_i) \cdot P_0 \geq 0$ (i.e. $x_i \geq \xi_1$) let $x_{i+1} = x_i - 2\frac{P(x_i)}{P'(x_i)}$, for $i = 1, 2, \ldots$ (double-step).
5. If $P(x_i) \cdot P_0 < 0$ (i.e. $x_i < \xi_1$) continue with the standard Newton's method $x_{j+1} = x_j - \frac{P(x_j)}{P'(x_j)}$, for $j = i, i+1, \ldots$.

Numerical methods

Jiří Zelinka

Autumn 2021, lecture 5

Vector and matrix norms

Vector norm $\|\cdot\| : \mathbb{R}^n \to \mathbb{R}$
Properties:

1. $\|\boldsymbol{x}\| \geq 0, \quad \forall \boldsymbol{x} \in \mathbb{R}^n$
2. $\|\boldsymbol{x}\| = 0 \Leftrightarrow \boldsymbol{x} = \boldsymbol{o}, \quad \boldsymbol{o} = (0, \ldots, 0)^T$
3. $\|\alpha \boldsymbol{x}\| = |\alpha| \, \|\boldsymbol{x}\|, \quad \forall \alpha \in \mathbb{R}, \quad \forall \boldsymbol{x} \in \mathbb{R}^n$
4. $\|\boldsymbol{x} + \boldsymbol{y}\| \leq \|\boldsymbol{x}\| + \|\boldsymbol{y}\|, \quad \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n.$

Examples:

1. $\|\mathbf{x}\|_2 = \left(\sum_{i=1}^{n} |x_i|^2\right)^{\frac{1}{2}}$ (Euclidean norm)
2. $\|\mathbf{x}\|_1 = \sum_{i=1}^{n} |x_i|$ (Manhattan norm)
3. $\|\mathbf{x}\|_\infty = \max_{1\leq i \leq n} |x_i|$ (maximum norm)
4. $\|\mathbf{x}\|_p = \left(\sum_{i=1}^{n} |x_i|^p\right)^{\frac{1}{p}}$ (p norm)

Metrics induced by the norm:
$\rho(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|$.

Convergence in norm: $\mathbf{x}_n \to \mathbf{x} \Leftrightarrow \|\mathbf{x}_n - \mathbf{x}\| \to 0$.

Matrix norm

$$A = \begin{pmatrix} a_{11} & \cdots & \cdots & a_{1n} \\ \vdots & & & \vdots \\ a_{n1} & \cdots & \cdots & a_{nn} \end{pmatrix}.$$

1. $\|A\| \geq 0, \quad \forall x \in \mathbb{R}^n$
2. $\|A\| = 0 \Leftrightarrow A = 0,$
3. $\|\alpha A\| = |\alpha| \, \|A\|, \quad \forall \alpha \in \mathbb{R}$
4. $\|A + B\| \leq \|A\| + \|B\|$
5. $\|A \cdot B\| \leq \|A\| \cdot \|B\|$ (submultiplicative norm)

A matrix norm $\|\cdot\|$ is called compatible with a vector norm $\|\cdot\|_a$ if

$$\|A\cdot \boldsymbol{x}\|_a \leq \|A\|\cdot\|\boldsymbol{x}\|_a.$$

A matrix norm $\|\cdot\|_a$ induced by a vector norm $\|\cdot\|_a$:

$$\|A\|_a = \sup_{\|\boldsymbol{x}\|_a=1} \|A\cdot \boldsymbol{x}\|_a.$$

Examples:

1. $\|A\|_1 = \max_{1\leq j\leq n} \sum_{i=1}^{n} |a_{ij}|,$
2. $\|A\|_\infty = \max_{1\leq i\ \leq n} \sum_{j=1}^{n} |a_{ij}|,$
3. $\|A\|_2 = \sqrt{\varrho(A^TA)}$, $\varrho(A^TA)$ is the maximal eigenvalue in absolute value (the spectral radius) of A^TA.

Systems of non-linear equations

$$\begin{aligned} f_1(x_1,\dots,x_n) &= 0\\ f_2(x_1,\dots,x_n) &= 0\\ &\vdots\\ f_n(x_1,\dots,x_n) &= 0 \end{aligned}$$

$$F(\boldsymbol{x}) = \boldsymbol{o}, \quad \hat{\boldsymbol{x}}\text{: solution}$$

Iterative form:

$$\begin{aligned} x_1 &= g_1(x_1,\dots,x_n)\\ x_2 &= g_2(x_1,\dots,x_n)\\ &\vdots\\ x_n &= g_n(x_1,\dots,x_n) \end{aligned}$$

$$\boldsymbol{x} = G(\boldsymbol{x}), \qquad \hat{\boldsymbol{x}}\text{: fixed point}$$

Iterative process: $\boldsymbol{x}^k$ – k-th iteration, $\boldsymbol{x}^{k+1} = G(\boldsymbol{x}^k)$.

Theorem:

Let $0 \leq q < 1$ and let $g_1, \ldots, g_n$ have continuous partial derivatives satisfying the inequality

$$\left\|\frac{\partial g_i(\boldsymbol{x})}{\partial x_j}\right\| \leq \frac{q}{n}, \quad i, j = 1, \ldots, n$$

in some neighborhood $O(\hat{x})$ of a fixed point $\hat{x}$. Then the iterative process given by $\boldsymbol{x}^{k+1} = G(\boldsymbol{x}^k)$ converges to the fixed point $\hat{x}$ for any $\boldsymbol{x}^0 \in O(\hat{x})$.

Newton's method

Let $F \in C^2(O(\hat{\boldsymbol{x}}))$

Taylor expansion for one function of n variables:

$$\begin{aligned} f(\boldsymbol{x}+\boldsymbol{h}) &= f(x_1+h_1,\dots,x_n+h_n) = \\ = f(x_1,\dots,x_n) &+ h_1\frac{\partial f}{\partial x_1}(\boldsymbol{x}) + \dots + h_n\frac{\partial f}{\partial x_n}(\boldsymbol{x}) + O\|\boldsymbol{h}\|^2 \end{aligned}$$

Taylor expansion for the system of functions:

$$F(\boldsymbol{x}+\boldsymbol{h}) = F(\boldsymbol{x}) + J_F(\boldsymbol{x})\boldsymbol{h} + O\|\boldsymbol{h}\|^2(1,\dots,1)^T$$

$$J_F(\boldsymbol{x}) = \begin{pmatrix} \dfrac{\partial f_1(\boldsymbol{x})}{\partial x_1} & \dots & \dfrac{\partial f_1(\boldsymbol{x})}{\partial x_m} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial f_m(\boldsymbol{x})}{\partial x_1} & \dots & \dfrac{\partial f_m(\boldsymbol{x})}{\partial x_m} \end{pmatrix}$$

Derivation of the method:

$$\boldsymbol{x} = \boldsymbol{x}^k, \quad \boldsymbol{x} + \boldsymbol{h} = \boldsymbol{x}^{k+1}$$

$$\boldsymbol{o} = F(\boldsymbol{x}^{k+1}) = F(\boldsymbol{x}^k) + J_F(\boldsymbol{x}^k)(\boldsymbol{x}^{k+1} - \boldsymbol{x}^k)$$

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k - J_F^{-1}(\boldsymbol{x}^k)F(\boldsymbol{x}^k)$$

Iteration function

$$G(\boldsymbol{x}) = \boldsymbol{x} - J_F^{-1}(\boldsymbol{x})F(\boldsymbol{x})$$

Example

$$\begin{aligned} x_2^2 - x_1 + 1 &= 0 \\ x_2^2 + x_1^2 - 2x_1 &= 0 \end{aligned}$$

System of linear equations – iterative methods

$$A\boldsymbol{x} = \boldsymbol{b} \quad \longrightarrow \quad \boldsymbol{x} = T\boldsymbol{x} + \boldsymbol{g}$$

Iteration process:

$$\boldsymbol{x}^{k+1} = T\boldsymbol{x}^{k} + \boldsymbol{g}, \qquad k = 0, 1, \dots$$

Solution:

$$\hat{\boldsymbol{x}} = (E - T)^{-1} g$$

Theorem

The sequence $\left\{\boldsymbol{x}^k\right\}_{k=0}^{\infty}$ determined by the iterative process $\boldsymbol{x} = T\boldsymbol{x} + \boldsymbol{g}$ converges for every initial iteration $\boldsymbol{x}^0 \in \mathbb{R}^n \iff \varrho(T) < 1$, i.e., $|\lambda| < 1$ for all eigenvalues λ of the matrix T.
In this case

$$\lim_{k\to\infty} \boldsymbol{x}^k = \hat{\boldsymbol{x}},\ \hat{\boldsymbol{x}} = T\hat{\boldsymbol{x}} + \boldsymbol{g}$$

Jacobi iterative method

System of linear equations:

$$A\boldsymbol{x} = \boldsymbol{b}$$

i-th equation:

$$a_{i1}x_1 + \cdots + a_{ii}x_i + \cdots + a_{in}x_n = b_i$$

The component x_i is expressed

$$x_i = -\sum_{\substack{j=1\\ j\neq i}}^{n} \frac{a_{ij}}{a_{ii}}x_j + \frac{b_i}{a_{ii}},$$

and it is used as the new $(k+1)$-th iteration

$$x_i^{k+1} = -\sum_{\substack{j=1\\ j\neq i}}^{n} \frac{a_{ij}}{a_{ii}}x_j^k + \frac{b_i}{a_{ii}},$$

Matrix notation

$$\begin{pmatrix} x_1^{k+1} \\ x_2^{k+1} \\ \vdots \\ x_n^{k+1} \end{pmatrix} = \begin{pmatrix} 0 & -\frac{a_{12}}{a_{11}} & \cdots & -\frac{a_{1n}}{a_{11}} \\ -\frac{a_{21}}{a_{22}} & 0 & & -\frac{a_{2n}}{a_{22}} \\ \vdots & & \ddots & \vdots \\ -\frac{a_{n1}}{a_{nn}} & -\frac{a_{n2}}{a_{nn}} & \cdots & 0 \end{pmatrix} \begin{pmatrix} x_1^{k} \\ x_2^{k} \\ \vdots \\ x_n^{k} \end{pmatrix} + \begin{pmatrix} \frac{b_1}{a_{11}} \\ \frac{b_2}{a_{22}} \\ \vdots \\ \frac{b_n}{a_{nn}} \end{pmatrix}.$$

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \qquad A = D + L + U,$$

$$A\boldsymbol{x} = (D + L + U)\boldsymbol{x} = \boldsymbol{b}$$

$$D = \begin{pmatrix} a_{11} & & 0 \\ & \ddots & \\ 0 & & a_{nn} \end{pmatrix},$$

$$L = \begin{pmatrix} 0 & & & 0 \\ a_{21} & \ddots & & \\ \vdots & \ddots & \ddots & \\ a_{n1} & \cdots & a_{n,n-1} & 0 \end{pmatrix},$$

$$U = \begin{pmatrix} 0 & a_{12} & \cdots & a_{1n} \\ & \ddots & \ddots & \vdots \\ & & \ddots & a_{n-1,n} \\ 0 & & & 0 \end{pmatrix}.$$

$$\boldsymbol{x} = -D^{-1}(L+U)\boldsymbol{x} + D^{-1}\boldsymbol{b}.$$

$$\boldsymbol{x}^{k+1} = -D^{-1}(L+U)\boldsymbol{x}^{k} + D^{-1}\boldsymbol{b}.$$

$$\mathbf{x}^{k+1} = T_J \mathbf{x}^k + D^{-1}\mathbf{b},$$

$$T_J = -D^{-1}(L+U),\ t_{ij} = -\frac{a_{ij}}{a_{ii}} \text{ for } i \neq j,\ t_{ii} = 0.$$

$$T_J = \begin{pmatrix} 0 & -\dfrac{a_{12}}{a_{11}} & \cdots & -\dfrac{a_{1n}}{a_{11}} \\ -\dfrac{a_{21}}{a_{22}} & 0 & & -\dfrac{a_{2n}}{a_{22}} \\ \vdots & & \ddots & \vdots \\ -\dfrac{a_{n1}}{a_{nn}} & -\dfrac{a_{n2}}{a_{nn}} & \cdots & 0 \end{pmatrix}, \quad D^{-1}\mathbf{b} = \begin{pmatrix} \dfrac{b_1}{a_{11}} \\ \dfrac{b_2}{a_{22}} \\ \vdots \\ \dfrac{b_n}{a_{nn}} \end{pmatrix}.$$

Gauss–Seidel iterative method

The component of the new iteration is used in the following step:

$$
\begin{aligned}
x_1^{k+1} &= \frac{1}{a_{11}}\left(b_1 - a_{12}x_2^k - a_{13}x_3^k - a_{14}x_4^k - \dots,\right)\\
x_2^{k+1} &= \frac{1}{a_{22}}\left(b_2 - a_{21}x_1^{k+1} - a_{23}x_3^k - a_{24}x_4^k - \dots,\right)\\
x_3^{k+1} &= \frac{1}{a_{33}}\left(b_3 - a_{31}x_1^{k+1} - a_{32}x_2^{k+1} - a_{34}x_4^k - \dots,\right)\\
&\vdots\\
x_i^{k+1} &= \frac{1}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{k+1} - \sum_{j=i+1}^{n} a_{ij}x_j^k\right)
\end{aligned}
$$

Matrix notation:

$$
\begin{aligned}
A\boldsymbol{x} &= \boldsymbol{b} \\
(D+L+U)\boldsymbol{x} &= \boldsymbol{b} \\
(D+L)\boldsymbol{x} &= -U\boldsymbol{x}+\boldsymbol{b} \\
\boldsymbol{x} &= -(D+L)^{-1}U\boldsymbol{x}+(D+L)^{-1}\boldsymbol{b}
\end{aligned}
$$

$$T_G = -(D+L)^{-1}U, \qquad \boldsymbol{x}^{k+1} = T_G\boldsymbol{x}^k + (D+L)^{-1}\boldsymbol{b}.$$

Theorem: If A is diagonally dominant matrix, i.e.

$$|a_{ii}| > \sum_{j\neq i} |a_{ij}| \quad \text{or} \quad |a_{ii}| > \sum_{j\neq i} |a_{ji}|$$

then Jacobi and Gauss–Seidel methods converge.

Relaxation (Succesive over–relaxation (SOR)) method

x^k – k-th iteration

x_{GS}^{k+1} – the following iteration aquired by the Gauss–Seidel metod

$\omega \in (0, 2)$ – relaxation parameter

$x^{k+1} = (1 - \omega)x^k + \omega x_{GS}^{k+1}$

Another iterative methods

- gradient descent method
- conjugate gradient method
- genetic algorithms
- parallel algorithms

Numerical methods

Jiří Zelinka

Autumn 2021, lecture 6

System of linear equations – direct methods

$$Ax = \boldsymbol{b}$$

- Inversion of A: $\hat{\boldsymbol{x}} = A^{-1}\boldsymbol{b}$
- Gaussian elimination: $A\boldsymbol{x} = \boldsymbol{b} \longrightarrow U\boldsymbol{x} = \tilde{\boldsymbol{b}}$, where U is upper triangular matrix, then we express $\boldsymbol{x}$ from the last component.
  Pivoting (partial) : swapping the rows to move the largest number in the column in the absolute value the to main diagonal - for numerical stability
- Matrix decomposition

LU decomposition

$$A = L \cdot U$$

L: lower triangular matrix, with 1 on the main diagonal
U: upper triangular matrix from Gaussian elimination

Computation of L: in the Gaussian elimination we add a c multiple of the k-th row to the l-th row $(l > k)$ to obtain zero at position l, k, $\longrightarrow$ into the position l, k in the matrix L we put the number $-c$.

Generally:

$$P \cdot A = L \cdot U$$

P: permutation matrix, exchanges the rows of the matrix A

LU decomposition is a modified form of Gaussian elimination.

Example:

$$
\begin{aligned}
2x_1 + 4x_2 - x_3 &= -5 \\
x_1 + x_2 - 3x_3 &= -9 \\
4x_1 + x_2 + 2x_3 &= 9
\end{aligned}
$$

Applications

Systems of linear equations

$$A \cdot x = b, \quad A = L \cdot U \Rightarrow A = L \cdot U \cdot x = b$$

$$y = U \cdot x \Rightarrow L \cdot y = b$$

We solve two system with triangular matrices.

Calculation of the inverse matrix

Matrix decompositions

QR decomposition

$$A = Q \cdot R$$

Q: orthogonal matrix, $Q^{-1} = Q^T$
R: upper triangular matrix

Application

Systems of linear equations

$$A \cdot x = b, \quad A = Q \cdot R \Rightarrow U \cdot x = Q^T b$$

QR decomposition has better numerical stability because of the orthogonal transformation.

QR algorithm: calculation of eigenvalues of the matrix

Matrix decompositions

Cholesky decomposition

Let A be a real symmetric positive definite matrix: $A^T = A$

$$A = R^T \cdot R$$

for upper triangular matrix R.

Theoretical background

$A \cdot x = b$: unsolvable system of linear equations

For given x let $r_x = b - A \cdot x$: residue for the vector x

$\hat{x}$ is called the *solution in sense of least squares* if $\|r_{\hat{x}}\| \leq \|r_x\|$ for any x.

$\mathcal{R}(A)$: the range space of the matrix A
$\mathcal{R}^{\perp}(A)$: the orthogonal complement of $\mathcal{R}(A)$
The vector b can be decomposed in the form $b = b_1 + b_2$, $b_1 \in \mathcal{R}(A)$, $b_2 \in \mathcal{R}^{\perp}(A)$
$A^T \cdot b_2 = o$, o is the zero vector

$\hat{x}$ is the solution of the system

$$A \cdot x = b_1$$

We have

$$A \cdot \hat{x} = b_1$$

$$A^T \cdot A \cdot \hat{x} = A^T \cdot b_1 + o = A^T \cdot b_1 + A^T \cdot b_2 = A^T \cdot b$$

So $\hat{x}$ is the solution of the *system of normal equations*:

$$A^T \cdot A \cdot x = A^T \cdot b$$

The solution is unique if columns of A are linearly independent. In this case

$$\hat{\boldsymbol{x}} = (AA^T)^{-1}A^T b.$$

Application for the function approximation

$x_0, \ldots, x_n$ – given points
$f_0, \ldots, f_n$ – given function values

$\Phi(x) = c_0\Phi_0(x) + \cdots + c_m\Phi_m(x)$ – given function depending on the parameters $c_0, \ldots, c_m$.

We want to find the parameters $\hat{c}_0, \ldots, \hat{c}_m$ to minimize

$$\sum_{k=0}^{n} [\Phi(x_k) - f_k]^2$$

We are looking for the solution in the sense of least squares of the system:

$$\begin{array}{ccccccccc}
c_0\Phi_0(x_0) & + & c_1\Phi_1(x_0) & + & \cdots & + & c_m\Phi_m(x_0) & = & f_0 \\
c_0\Phi_0(x_1) & + & c_1\Phi_1(x_1) & + & \cdots & + & c_m\Phi_m(x_1) & = & f_1 \\
c_0\Phi_0(x_2) & + & c_1\Phi_1(x_2) & + & \cdots & + & c_m\Phi_m(x_2) & = & f_2 \\
 & & & & & & & \vdots & \\
c_0\Phi_0(x_n) & + & c_1\Phi_1(x_n) & + & \cdots & + & c_m\Phi_m(x_n) & = & f_n
\end{array}$$

Let

$$A = \begin{pmatrix} \Phi_0(x_0) & \Phi_1(x_0) & \cdots & \Phi_m(x_0) \\ \Phi_0(x_1) & \Phi_1(x_1) & \cdots & \Phi_m(x_1) \\ \Phi_0(x_2) & \Phi_1(x_2) & \cdots & \Phi_m(x_2) \\ \vdots & & & \vdots \\ \Phi_0(x_n) & \Phi_1(x_n) & \cdots & \Phi_m(x_n) \end{pmatrix} \quad \text{and} \quad f = \begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ \vdots \\ f_n \end{pmatrix}$$

Then the parameters $\hat{c} = (\hat{c}_0, \dots, \hat{c}_m)^T$ are given by the normal equations

$$A^T \cdot A \cdot c = A^T \cdot f$$

i.e.

$$\hat{c} = \left(A^T \cdot A\right)^{-1} A^T \cdot f$$

Example:

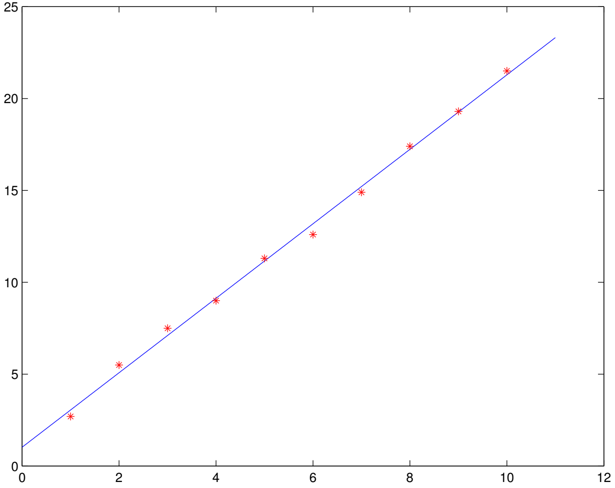
x_i	1	2	3	4	5	6	7	8	9	10
f_i	2.7	5.5	7.5	9.0	11.3	12.6	14.9	17.4	19.3	21.5

Find a linear function approximating data.

Solution: $\Phi_0(x) = 1$, $\Phi_1(x) = x$

$$A = \begin{pmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 1 & 4 \\ 1 & 5 \\ 1 & 6 \\ 1 & 7 \\ 1 & 8 \\ 1 & 9 \\ 1 & 10 \end{pmatrix}, \qquad f = \begin{pmatrix} 2.7 \\ 5.5 \\ 7.5 \\ 9.0 \\ 11.3 \\ 12.6 \\ 14.9 \\ 17.4 \\ 19.3 \\ 21.5 \end{pmatrix}.$$

$$\hat{c} = \left(A^T \cdot A\right)^{-1} A^T \cdot f \doteq \begin{pmatrix} 1.0267 \\ 2.0261 \end{pmatrix}, \quad \Phi(x) = 1.0267 + 2.0261x\,.$$



Numerical methods

Jiří Zelinka

Autumn 2021, lecture 7

Interpolation

$x_0, \ldots, x_n$ – given points (knots), $x_i \neq x_j$ for $i \neq j$
$f_0, \ldots, f_n$ – given function values (measurements), $f_i = f(x_i)$
$\Phi(x) = a_o\Phi_0(x) + \cdots + a_n\Phi_n(x)$ – given function depending on the parameters $a_0, \ldots, a_n$.

Examples:

$\Phi(x) = a_0 + a_1 x + \cdots + a_n x^n$: a polynomial,
$\Phi(x) = a_0 + a_1 e^{ix} + \cdots + a_n e^{inx}$: a trigonometric polynomial.

Problem of interpolation:
find the parameters $a_0, \ldots, a_n$ to fulfill conditions

$$\Phi(x_i) = f_i, \text{ for } i = 0, 1, \ldots, n.$$

Polynomial interpolation

Theorem

For given points $(x_i, f_i), i = 0, \dots, n,\ x_i \neq x_j$ for $i \neq j$ there exists the unique polynomial P of degree at most n with

$$P(x_i) = f_i, \quad i = 0, \dots, n.$$

Uniqueness:
If $P_1(x_i) = P_2(x_i) = f_i, \quad i = 0, \dots, n.$, then $Q = P_1 - P_2$ is a polynomial of degree at most n and $Q(x_i) = 0, \quad i = 0, \dots, n.$, i.e., Q has $n + 1$ roots so Q must be zero polynomial.

Existence:

Construction of P:

We construct the polynomials L_i:

- L_i is a polynomial of degree n,
- $L_i(x_j) = \begin{cases} 0 & \text{pro } i \neq j \\ 1 & \text{pro } i = j. \end{cases}$

Points $x_j, j \neq i$ are roots of L_i:

$$L_i(x) = A_i(x - x_0) \dots (x - x_{i-1})(x - x_{i+1}) \dots (x - x_n).$$

or

$$L_i(x) = A_i \pi_i(x), \text{ where } \pi_i(x) = \prod_{j \neq i} (x - x_j)$$

$$L_i(x_i) = 1 \ \Rightarrow \ A_i = \frac{1}{\pi_i(x_i)}.$$

$$L_i(x) = \frac{\pi_i(x)}{\pi_i(x_i)} = \frac{\prod\limits_{j \neq i}(x - x_j)}{\prod\limits_{j \neq i}(x_i - x_j)}$$

L_i – Lagrange base polynomials

Lagrange interpolation polynomial:

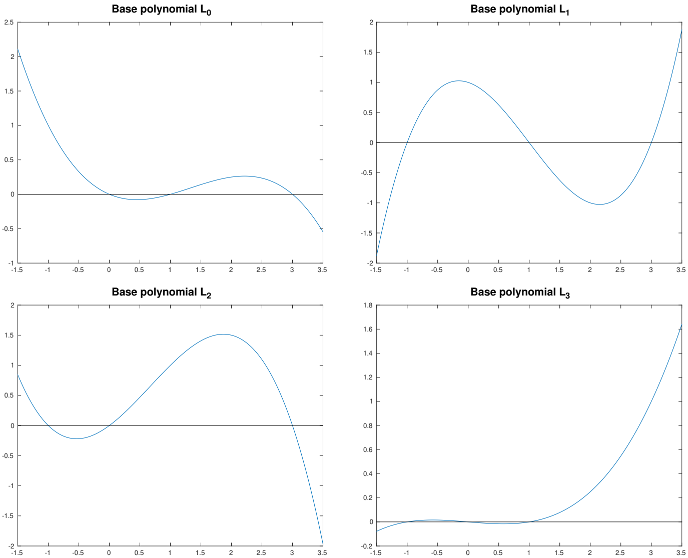
$$P(x) = \sum_{i=0}^{n} f_i L_i(x) = \sum_{i=0}^{n} f_i \frac{\prod\limits_{j \neq i}(x - x_j)}{\prod\limits_{j \neq i}(x_i - x_j)}$$

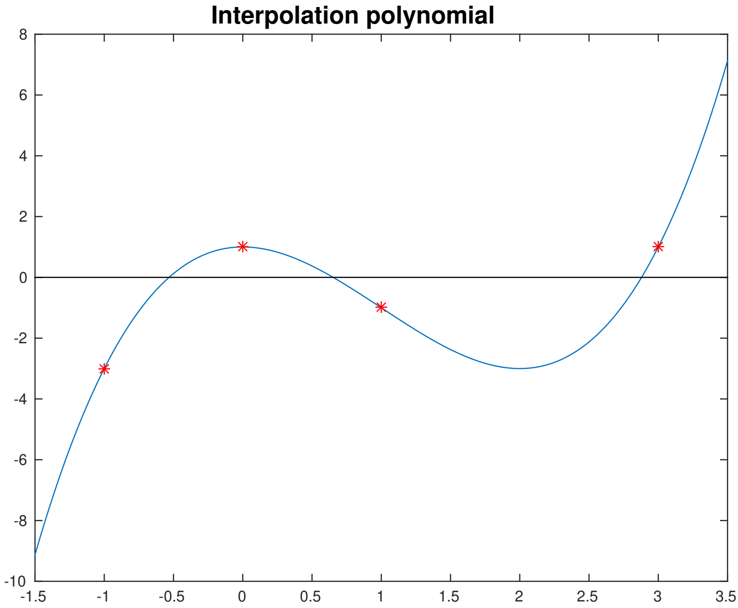
Example:

x_i	-1	0	1	3
f_i	-3	1	-1	1

$$\begin{aligned}
L_0(x) &= \frac{(x-0)(x-1)(x-3)}{(-1-0)(-1-1)(-1-3)} = -\frac{1}{8}x^3 + \frac{1}{2}x^2 - \frac{3}{8}x \\
L_1(x) &= \frac{(x+1)(x-1)(x-3)}{(0+1)(0-1)(0-3)} = \frac{1}{3}x^3 - x^2 - \frac{1}{3}x + 1 \\
L_2(x) &= \frac{(x+1)(x-0)(x-3)}{(1+1)(1-0)(1-3)} = -\frac{1}{4}x^3 + \frac{1}{2}x^2 + \frac{3}{4}x \\
L_3(x) &= \frac{(x+1)(x-0)(x-1)}{(3+1)(3-0)(3-1)} = \frac{1}{24}x^3 - \frac{1}{24}x \\
P(x) &= -3L_0(x) + L_1(x) - L_2(x) + L_3(x) = x^3 - 3x^2 + 1
\end{aligned}$$

Lagrange base polynomials




Interpolation polynomial
8
6
4
2
0
-2
-4
-6
-8
-10
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5
3
3.5

Effective calculation of L_i

Calculation of one base polynomial L_i is $O(n^2)$, i.e. direct calculation of the interpolation polynomial is $O(n^3)$.

Effective calculation:

$$
\begin{array}{ll}
\omega(x) = \prod_{j=0}^{n} (x - x_j) & O(n^2) \\
\pi_i(x) = \omega(x) : (x - x_i) & \text{Horner's scheme, } O(n) \\
\pi(x_i) & \text{Horner scheme's, } O(n) \\
P & O(n^2)
\end{array}
$$

Example:

x_i	-1	0	1	3

$$
\begin{aligned}
\omega(x) &= (x+1)(x-0)(x-1)(x-3) = x^4 - 3x^3 - x^2 + 3x \\
\pi_0(x) &= \omega(x) : (x+1) \\
\pi_1(x) &= \omega(x) : (x-0) \\
\pi_2(x) &= \omega(x) : (x-1) \\
\pi_3(x) &= \omega(x) : (x-3)
\end{aligned}
$$

Horner's scheme for division $\omega(x) : (x - x_0)$, i.e. $\omega(x) : (x + 1)$:

ω	1	-3	-1	3	0
-1	$\boxed{1}$	$\boxed{-4}$	$\boxed{3}$	$\boxed{0}$	0

$\pi_0(x) = x^3 - 4x^2 + 3x$

Horner scheme for $\pi_0(x_0) = \pi_0(-1)$:

$\pi(x)$	1	-4	3	0
-1	1	-5	8	$\boxed{-8}$

$\pi_0(-1) = -8$

$L_0(x) = \frac{\pi_0(x)}{\pi_0(x_0)} = -\frac{1}{8}(x^3 - 4x^2 + 3x)$

Similarly $L_1, L_2, \ldots$

Advantage of the Lagrange interpolation polynomial: easy computation of more polynomials on the same knots.

Disadvantage of the Lagrange interpolation polynomial: adding a point (x_{n+1}, f_{n+1}) will cause recalculation of all base polynomials L_i.

Newton interpolation polynomial

Base functions:

$\Phi_0(x) = 1,$
$\Phi_1(x) = (x - x_0),$
$\Phi_2(x) = (x - x_0)(x - x_1),$
$\vdots$
$\Phi_n(x) = (x - x_0) \cdots (x - x_{n-1}).$

Interpolation polynomial:

$$P_n(x) = a_o\Phi_0(x) + \cdots + a_n\Phi_n(x)$$

Adding a point (x_{n+1}, f_{n+1}):

$$P_{n+1}(x) = P_n(x) + a_{n+1}\Phi_{n+1}(x)$$

Calculation of parameters a_i:

$a_i = f[x_0, x_1, \ldots, x_i]$ – *divided difference*

$f[x_i] = f_i$

$f[x_i, x_j] = \frac{f_i - f_j}{x_i - x_j}$

$f[x_j, \ldots, x_{j+k}] = \frac{f[x_{j+1}, \ldots, x_{j+k}] - f[x_j, \ldots, x_{j+k-1}]}{x_{j+k} - x_j}$

i.e.

$f[x_0, \ldots, x_i] = \frac{f[x_1, \ldots, x_i] - f[x_0, \ldots, x_{i-1}]}{x_i - x_0}$

$$
\begin{aligned}
P(x) &= f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) + \\
&+ \cdots + f[x_0, \ldots, x_n](x - x_0)\cdots(x - x_{n-1})
\end{aligned}
$$

Table of divided differences

$$
\begin{array}{cc|c|c|c|c}
x_i & f_i & f[x_i, x_{i+1}] & f[x_i, x_{i+1}, x_{i+2}] & \cdots & \\
\hline
x_0 & \underline{f_0} & & & & \\
 & & > \underline{f[x_0, x_1]} & & & \\
x_1 & f_1 & & > \underline{f[x_0, x_1, x_2]} & & \\
 & & > f[x_1, x_2] & & > & \\
x_2 & f_2 & & \vdots & & \cdots > \underline{f[x_0, \ldots, x_n]} \\
 & & \vdots & & > & \\
\vdots & \vdots & & > f[x_{n-2}, x_{n-1}, x_n] & & \\
 & & > f[x_{n-1}, x_n] & & & \\
x_n & f_n & & & &
\end{array}
$$

Example:

$$
\begin{array}{cc|ccc}
x_i & f_i & f[x_i, x_{i+1}] & f[x_i, x_{i+1}, x_{i+2}] & f[x_0, x_1, x_2, x_3] \\
\hline
-1 & \boxed{-3} & & & \\
 & & > \frac{1+3}{0+1} = \boxed{4} & & \\
0 & 1 & & > \frac{-2-4}{1+1} = \boxed{-3} & \\
 & & > \frac{-1-1}{1-0} = -2 & & > \frac{1+3}{3+1} = \boxed{1} \\
1 & -1 & & > \frac{1+2}{3-0} = 1 & \\
 & & > \frac{1+1}{3-1} = 1 & & \\
3 & 1 & & &
\end{array}
$$

$$
\begin{aligned}
P(x) &= -3 + 4(x+1) - 3(x+1)x + 1(x+1)x(x-1) = \\
&= x^3 - 3x^2 + 1
\end{aligned}
$$

Error of the interpolation polynomial

$$f(x) - P_n(x) = \frac{\omega_{n+1}(x)}{(n+1)!} f^{(n+1)}(\xi), \qquad \xi \in [min\{x_i\}, max\{x_i\}].$$

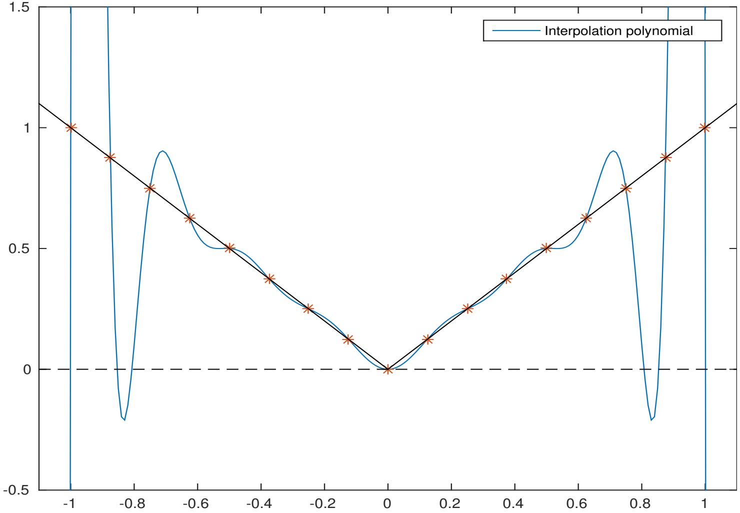
Knots selection can affect the interpolation error.

Knots minimizing absolute value of the error on $[-1, 1]$:

$x_i = \cos\left(\frac{2i+1}{n+1}\frac{\pi}{2}\right),\ i = 0, \ldots, n$

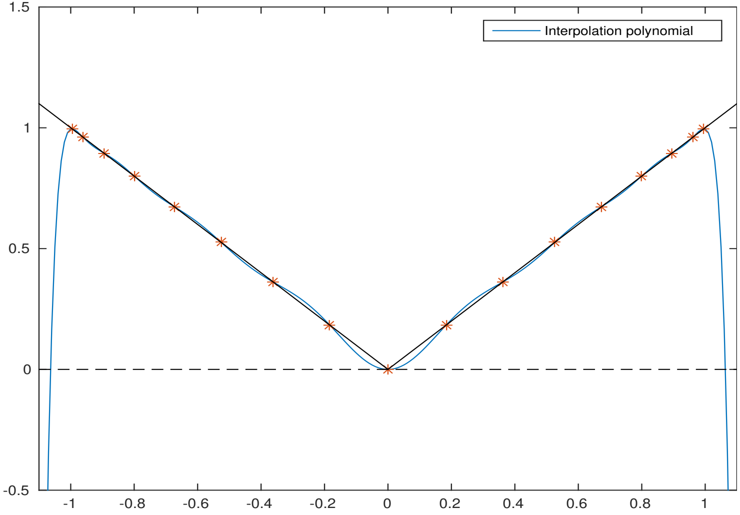
Optimal placement of knots

$f(x) = |x|$, $x_0, \ldots, x_n$ – equidistant on $[-1, 1]$



Optimal placement of knots

$$f(x) = |x|,\ x_i = \cos\left(\frac{2i+1}{n+1}\frac{\pi}{2}\right),\ i = 0, \dots, n$$



Numerical methods

Jiří Zelinka

Autumn 2021, lecture 8

Spline interpolation

$x_0, \dots, x_n$ – given points, $x_0 < x_1 < \cdots < x_n$
$f_0, \dots, f_n$ – given function values

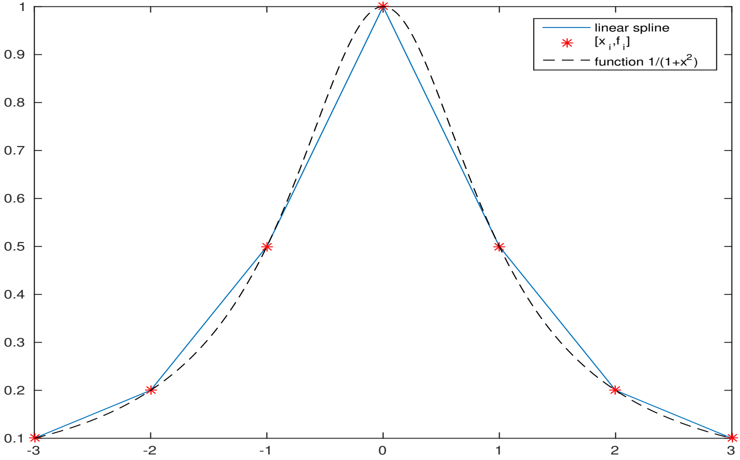
$r, d > 0$ natural numbers, r – degree, d – defect

S – *spline* – piecewise polynomials of degree r
S has continuous derivatives up to order $r - d$

$\mathcal{S}_{r,d}$ – space of splines of degree r with defect d

Example 1.

$\mathcal{S}_{1,1}$ – piecewise linear continuous functions



$S \in \mathcal{S}_{1,1}$ – linear spline, it is determined uniquely by the function values $f_0, \ldots, f_n$.

Example 2.

$\mathcal{S}_{3,1}$ – piecewise polynomials of degree 3 with continuous derivatives to order 2.

Number of parameters describing spline $S \in \mathcal{S}_{3,1}$:

We have n subintervals $I_k = [x_k, x_{k+1}],\ k = 0, \dots, n-1$, in every subinterval the spline is described by 4 parameters:

$$\text{For } x \in I_k \quad S(x) = S_k(x) = a_k + b_k(x - x_k) + c_k(x - x_k)^2 + d_k(x - x_k)^3$$

$\Rightarrow$ The spline S is described by 4n parameters.

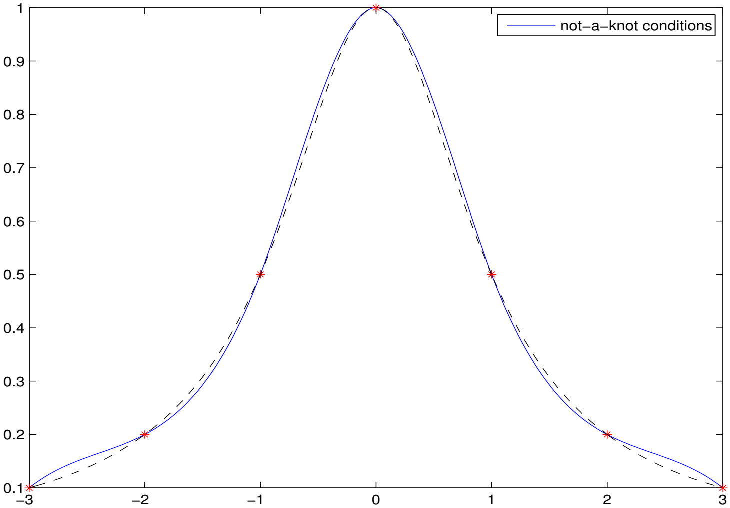
These parameters are bound by conditions:

S is continuous in $x_1, \dots, x_{n-1}$:	$n-1$ conditions
S' is continuous in $x_1, \dots, x_{n-1}$:	$n-1$ conditions
S'' is continuous in $x_1, \dots, x_{n-1}$:	$n-1$ conditions
$S(x_k) = f_k,\ k = 0, \dots, n$:	$n+1$ conditions
Together	$4n-2$ conditions

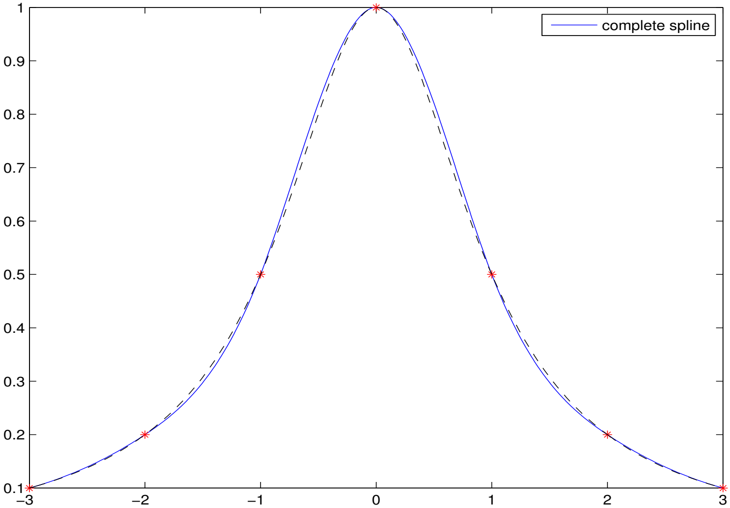
To obtain the unique cubic spline we need two additional *boundary conditions*:

1. $S'(x_0)$ and $S'(x_n)$ are given – complete cubic spline
2. $S''(x_0)$ and $S''(x_n)$ are given, especially $S''(x_0) = S''(x_n) = 0$: natural cubic spline
3. S''' is continuous in x_1 and x_{n-1}: not–a–knot conditions
4. $S(x_0) = S(x_n)$, $S'(x_0) = S'(x_n)$, $S''(x_0) = S''(x_n)$:periodic spline

Interpolation spline with not–a–knot conditions



Interpolation complete spline



Bernstein polynomials

Base polynomials:

$n \in \mathbb{N}$, $b_{k,n}(x) = \binom{n}{k} x^k (1-x)^{n-k},\ x \in [0,1],\ k = 0, \dots, n$

f is a real function defined on $[0,1]$, $f_k = f(\frac{k}{n})$
Bernstein polynomial of degree n for the function f:

$$B_{f,n}(x) = \sum_{k=0}^{n} f_k b_{k,n}(x)$$

Properties of Bernstein polynomials

- $\sum_{k=0}^{n} b_{k,n}(x) = 1$
- $b_{k,n}(x) \geq 0$ for $x \in [0,1]$
- $b_{k,n}(1-x) = b_{n-k,n}(x)$
- $b_{k,n}(0) = \delta_{k,0}$, $b_{k,n}(1) = \delta_{k,n}$, δ: Kronecker delta
- $b_{k,n}$ has roots 0 (of multiplicity k) and 1 of multiplicity $n-k$
- $b'_{k,n} = n(b_{k-1,n-1} - b_{k,n-1})$
- $\int b_{k,n} = \frac{1}{n+1} \sum_{j=k+1}^{n+1} b_{j,n}$

Theorem 1

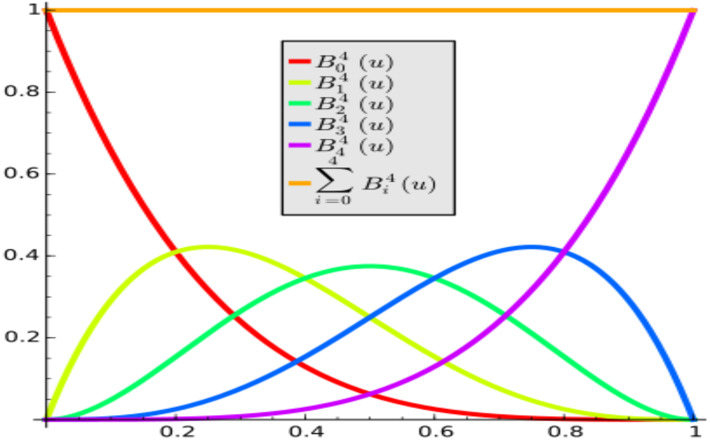
$B_{f,n}$ is a linear operator:

$$g = \sum a_j f_j \ \Rightarrow B_{g,n} = \sum a_j B_{f_j,n}$$

Theorem 2

If f is continuous on $[0,1]$, $f_k = f(\frac{k}{n})$, then $B_{f,n}$ converges uniformly on $[0,1]$ to the function f for $n \to \infty$.

Bernstein base polynomials, $n = 4$

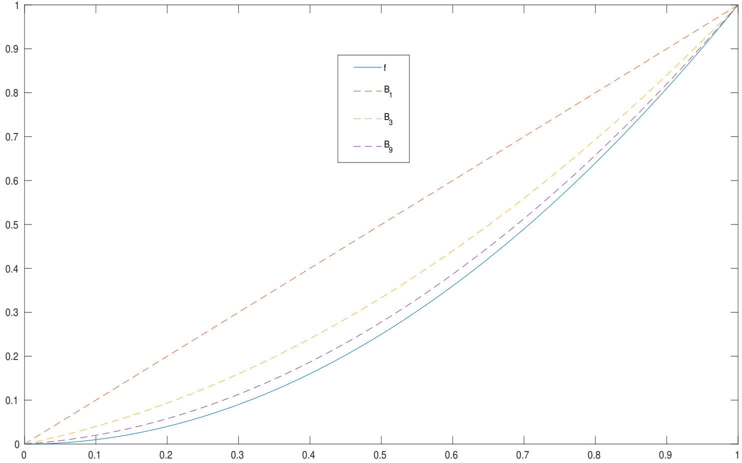


Examples

- $f(x) \equiv 1 \Rightarrow B_{f,n}(x) \equiv 1$
- $f(x) = x \Rightarrow B_{f,n}(x) = x$
- $f(x) = x^2 \Rightarrow B_{f,n}(x) = x^2 + \frac{x-x^2}{n} \neq x^2$

Example 2.

Bernstein polynomials of degree 1, 3 and 9 for $f(x) = x^2$.



Numerical methods

Jiří Zelinka

Autumn 2021, lecture 9

Bernstein polynomials

Base polynomials:

$n \in \mathbb{N}$, $b_{k,n}(x) = \binom{n}{k} x^k (1-x)^{n-k},\ x \in [0,1],\ k = 0, \dots, n$

f is a real function defined on $[0,1]$, $f_k = f(\frac{k}{n})$
Bernstein polynomial of degree n for the function f:

$$B_{f,n}(x) = \sum_{k=0}^{n} f_k b_{k,n}(x)$$

Bézier curves

The curves were discovered independently in 1959 and 1960 in Citroen and Renault car factories by Paul de Casteljau and Pierre Bézier.

$P_0, \cdots P_n$ given points,
$P_k = [x_k, y_k] \in \mathbb{R}^2$ or $P_k = [x_k, y_k, z_k] \in \mathbb{R}^3$

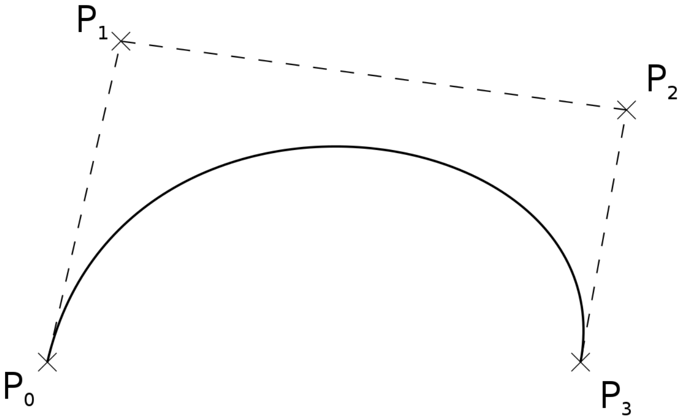
Explicit (parametric) definition:

$$B(t) = \sum_{k=0}^{n} P_k b_{k,n}(t), \quad t \in [0, 1]$$

$$B(t) = [x(t), y(t)], \text{ or } B(t) = [x(t), y(t), z(t)]$$

In coordinates:

$$\begin{aligned} x(t) &= \sum_{k=0}^{n} x_k b_{k,n}(t), \quad t \in [0,1] \\ y(t) &= \sum_{k=0}^{n} y_k b_{k,n}(t), \quad t \in [0,1] \\ z(t) &= \sum_{k=0}^{n} z_z b_{k,n}(t), \quad t \in [0,1] \end{aligned}$$


P_1
P_2
P_0
P_3

Derivative:

$$B'(t) = n \sum_{i-0}^{n-1} b_{i,n-1}(t)\,(P_{i+1} - P_i)$$

Consequence:

The tangent at the points P_0 and P_n is given by the difference $P_1 - P_0$ or $P_{n-1} - P_n$, respectively.

Proof:

$$\begin{aligned} B'(0) &= n \sum_{i-0}^{n-1} b_{i,n-1}(0)\,(P_{i+1} - P_i) = n b_{0,n-1}(0)\,(P_1 - P_0) \\ &= n\,(P_1 - P_0) \end{aligned}$$

$$\begin{aligned} B'(1) &= n \sum_{i-0}^{n-1} b_{i,n-1}(1)\,(P_{i+1} - P_i) = n b_{n-1,n-1}(1)\,(P_n - P_{n-1}) \\ &= n\,(P_n - P_{n-1}) \end{aligned}$$

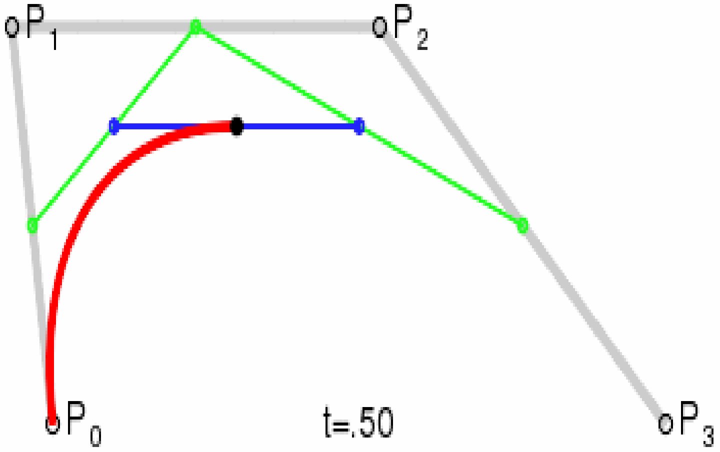
Recursive definition:

$B_{P_0P_1\dots P_n}$: the Bézier curve determined by given points

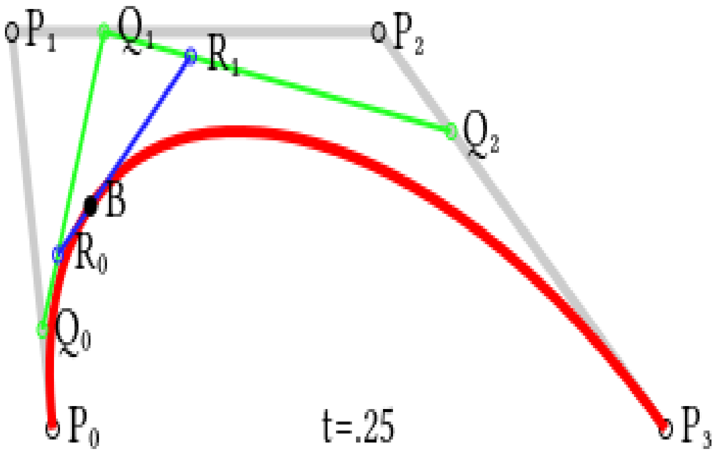
$$\begin{aligned} B_{P_k} &= P_k \\ B_{P_0P_1\dots P_n}(t) &= (1-t)B_{P_0P_1\dots P_{n-1}}(t) + tB_{P_1\dots P_n}(t) \end{aligned}$$

$t \in [0, 1]$

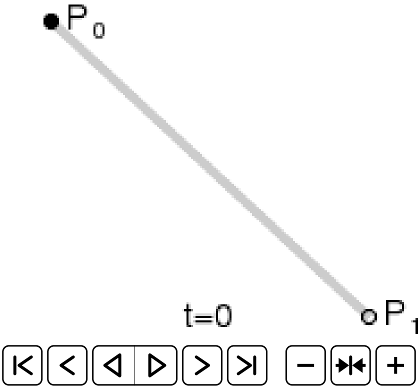
De Casteljau's algorithm



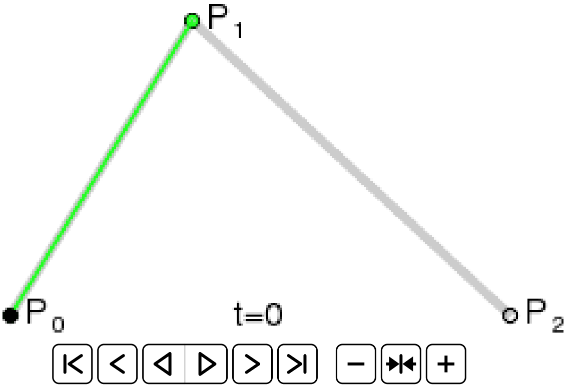
De Casteljau's algorithm



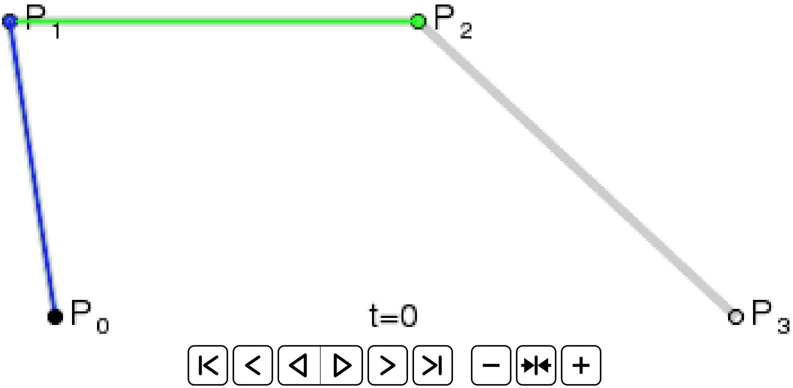
Linear Bézier curve



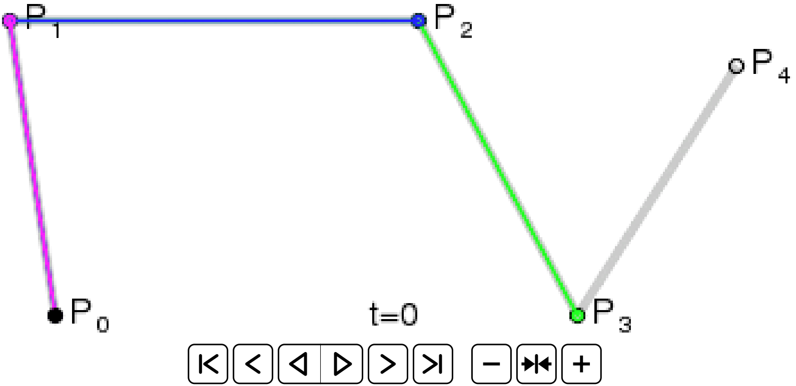
Quadratic Bézier curve



Cubic Bézier curve



Quartic Bézier curve



B-splines

Knot vector: $T = (t_0, t_1, \dots, t_m)$, t_i – knots, $t_i \leq t_{i+1}$

Degree of the B-spline: p

Base functions:

$$B_{i,0}(t) = \begin{cases} 1 & t_i \leq t < t_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

$$B_{i,k}(t) = \frac{t - t_i}{t_{i+k} - t_i} B_{i,k-1}(t) + \frac{t_{i+k+1} - t}{t_{i+k+1} - t_{i+1}} B_{i+1,k-1}(t),\ k = 1, \dots, p$$

Number of base functions: $n + 1$, $n = m - p - 1$

Properties

- $B_{i,p}$ is a piecwise polynomial of degree p.
- $B_{i,p}(t) = 0$ for $t < t_i$ and $t > t_{i+p}$.
- $\forall t \in [t_p, t_{m-p}] : \sum_{i=0}^{n} B_{i,p}(t) = 1$
- In the knot t_j of multiplicity r are the base functions $B_{i,p}$ continuous to order $p - r$.
- In (t_i, t_{i+1}) only the base functions $B_{i-p,p}, \dots, B_{i,p}$ are not equal to zero.

Internal knots: $t_{p+1}, \ldots, t_{m-p-1}$

Vector T is called *nonperiodic* (or *open*) if first $p+1$ knots are the same and last $p+1$ knots are the same.

Uniform B-spline: internal knots are equally spaced.

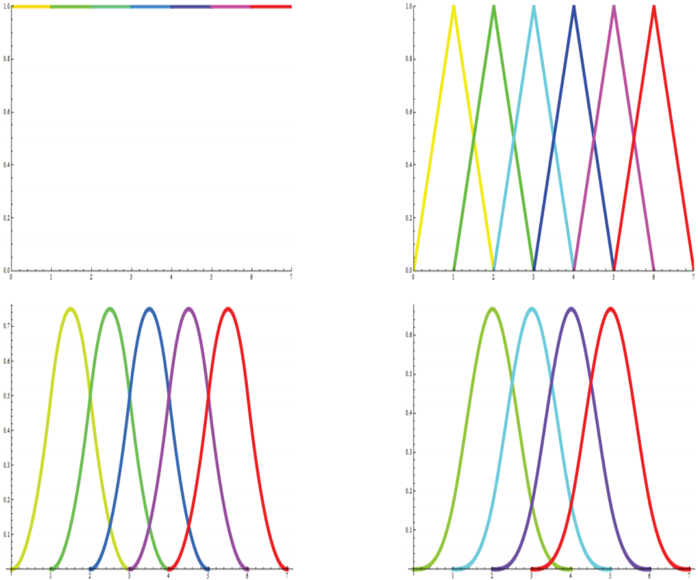
B-splines for knots $T = (\underbrace{0, \ldots, 0}_{p+1}, \underbrace{1, \ldots, 1}_{p+1})$ (no internal knots) are Bernstein base polynomials.

Examples:

- $T = (0, 1, \ldots, m)$
- $T = (\underbrace{0, \ldots, 0}_{p+1}, \underbrace{1, \ldots, 1}_{p+1})$

Example:

$$T = (0, 1, \ldots, m)$$



Numerical methods

Jiří Zelinka

Autumn 2021, lecture 10

B-splines

Knot vector: $T = (t_0, t_1, \dots, t_m)$, t_i – knots, $t_i \leq t_{i+1}$

Degree of the B-spline: p

Base functions:

$$B_{i,0}(t) = \begin{cases} 1 & t_i \leq t < t_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

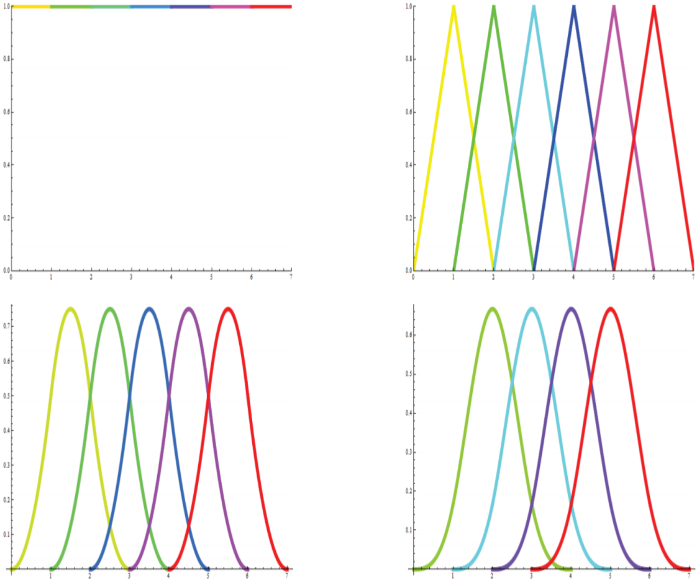
$$B_{i,k}(t) = \frac{t - t_i}{t_{i+k} - t_i} B_{i,k-1}(t) + \frac{t_{i+k+1} - t}{t_{i+k+1} - t_{i+1}} B_{i+1,k-1}(t),\ k = 1, \dots, p$$

For $t_i = t_{i+k}$ or $t_{i+1} = t_{i+k+1}$ is $B_{i,k}(t)$ defined as limit for $t_i \to t_{i+1}$ or $t_{i+k} \to t_{i+k+1}$.

Number of base functions: $n + 1$, $n = m - p - 1$

Example:

$$T = (0, 1, \ldots, m)$$



Properties

- $B_{i,p}$ is a piecwise polynomial of degree p.
- $B_{i,p}(t) = 0$ for $t < t_i$ and $t > t_{i+p+1}$.
- $\forall t \in [t_p, t_{m-p}] : \sum_{i=0}^{n} B_{i,p}(t) = 1$
- In the knot t_j of multiplicity r, the base functions $B_{i,p}$ are continuous to order $p - r$.
- In (t_i, t_{i+1}) only the base functions $B_{i-p,p}, \ldots, B_{i,p}$ are not equal to zero.

Internal knots: $t_{p+1}, \ldots, t_{m-p-1}$

Vector T is called *nonperiodic* (or *open*) if first $p+1$ knots are the same and last $p+1$ knots are the same.

Uniform B-spline: internal knots are equally spaced.

B-splines for knots $T = (\underbrace{0, \ldots, 0}_{p+1}, \underbrace{1, \ldots, 1}_{p+1})$ (no internal knots) are Bernstein base polynomials.

Examples:

- $T = (0, 1, \ldots, m)$
- $T = (\underbrace{0, \ldots, 0}_{p+1}, \underbrace{1, \ldots, 1}_{p+1})$

B-spline curves

Control points: $P_0, \ldots, P_n$ – control polygon

Base functions: $B_{0,p}, \ldots, B_{n,p}$
The relationship for n, m and p:

$$p = m - n - 1$$

B-spline curve:

$$C(t) = \sum_{i=0}^{n} P_i B_{i,p}$$

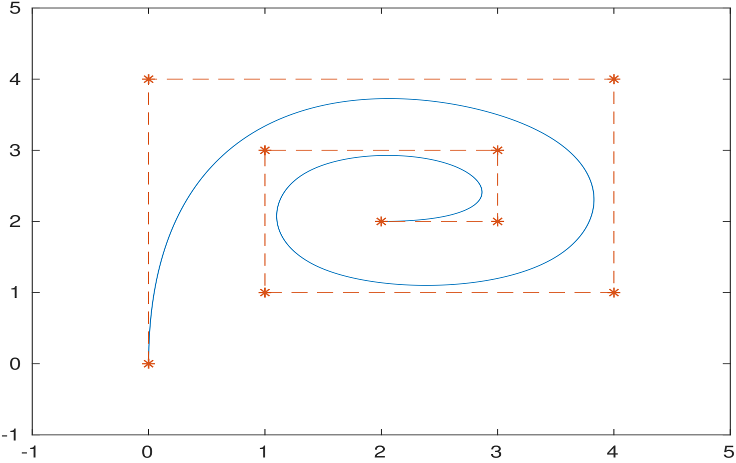
Properties

- Changing the P_i point affects the shape of the curve at an interval (t_i, t_{i+p+1}).
- Each part of the curve lies in a convex hull of $p+1$ points of control polygon.
- B-spline curve has continuous derivatives up to order $p-1$ if all the inner knots are of multiplicity 1 and the control points do not coincide.

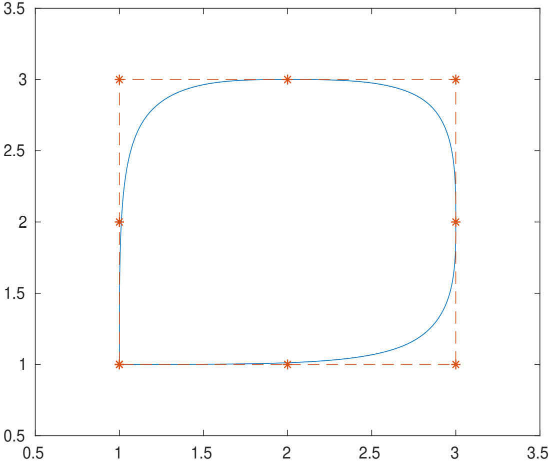
Computation of B-spline: de Boor's algorithm – generalization of de Casteljau's algorithm.

Examples:

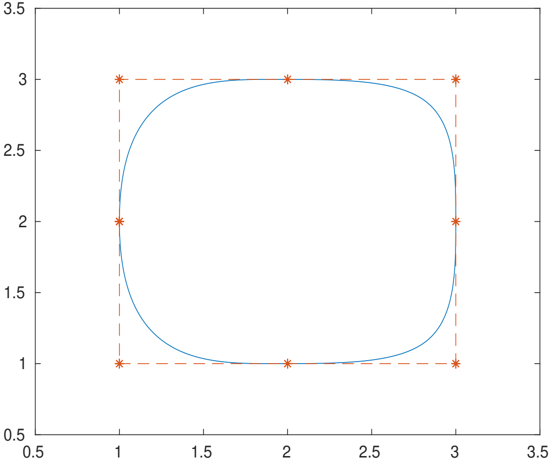
$T = (0, 0, 0, 0, 1, 2, 3, 4, 5, 6, 6, 6, 6)$, $m = 12$, $p = 3$, $n = 8$



Approximation of the square:



Approximation of the square:



Derivative of B-spline

$$\frac{d}{dt}B_{i,k}(t) = k\left[\frac{B_{i,k-1}(t)}{t_{i+k-1}-t_i} - \frac{B_{i+1,k-1}(t)}{t_{i+k}-t_{i+1}}\right]$$

The derivative of the spline $S(t) = \sum_{i=0}^{n} b_i B_{i,p}(t)$:

$$S'(t) = \sum_j c_j B_{j,p-1}(t)$$

for $c_j = \begin{cases} k\frac{b_j-b_{j-1}}{t_{j+k-1}-t_j} & \text{for} \quad t_{j+k-1} > t_j \\ 0 & \text{else} \end{cases}$

Interpolation using cubic B-splines

Points $x_0 < x_1 < \cdots < x_N$ with function values $f_0, \ldots, f_N$.

Knots $T = (x_0, x_0, x_0, x_0, x_1, \ldots, x_{N-1}, x_N, x_N, x_N, x_N)$
$m = N + 6,\ p = 3,\ n = N + 2.$

Boundary conditions: $S'(x_0)$ and $S'(x_N)$ are given.

Equations for the interpolation spline $S(t) = \sum_{i=0}^{n} b_i B_{i,p}(t)$:

$$
\begin{aligned}
b_0 B'_{0,3}(x_0) + b_1 B'_{1,3}(x_0) + b_2 B'_{2,3}(x_0) &= S'(x_0) \\
b_i B_{i,3}(x_i) + b_{i+1} B_{i+1,3}(x_i) + b_{i+2} B_{i+2,3}(x_i) &= f_i,\ i = 0, \ldots, N \\
b_N B'_{N,3}(x_N) + b_{N+1} B'_{n+1,3}(x_N) + b_{N+2} B'_{N+2,3}(x_N) &= S'(x_N)
\end{aligned}
$$

Non-uniform rational basis spline

$$C(t) = \frac{\sum_{i=0}^{n} w_i P_i B_{i,p}(t)}{\sum_{i=0}^{p} w_i B_{i,p}(t)}$$

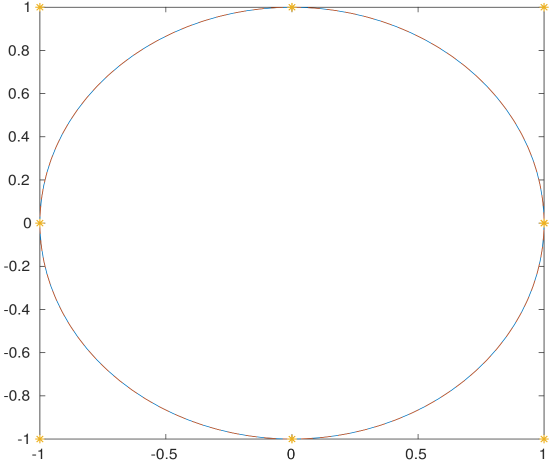
w_i – weights
P_i – points

For $w_i = w$ we obtain B-spline curve

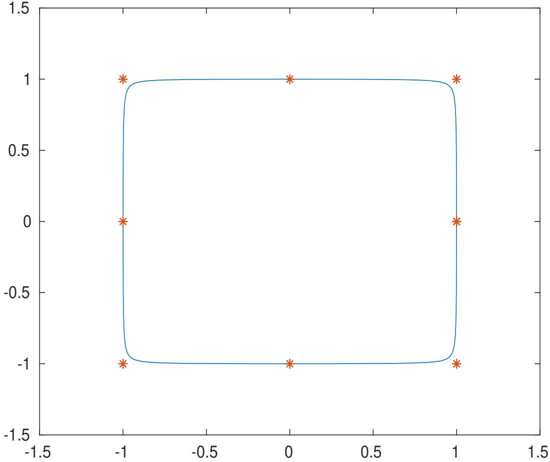
Examples:

Knots: $(0, 0, 0, 0.25, 0.25, 0.5, 0.5, 0.75, 0.75, 1, 1, 1)$
Points:
$[1; 0], [1; 1], [0; 1], [-1; 1], [-1; 0], [-1; -1], [0; -1], [1; -1], [1; 0]$

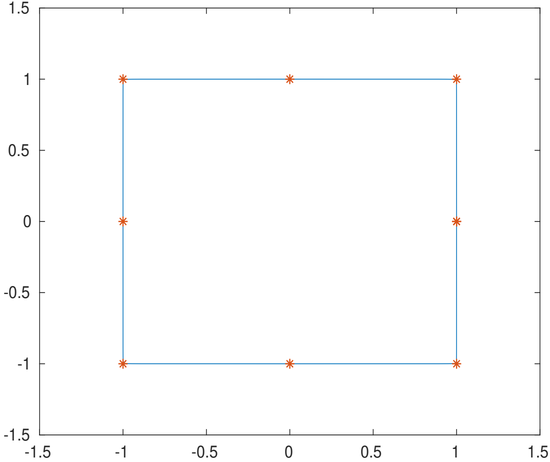
Weights: $1, \frac{\sqrt{2}}{2}, 1, \frac{\sqrt{2}}{2}, 1, \frac{\sqrt{2}}{2}, 1, \frac{\sqrt{2}}{2}, 1$



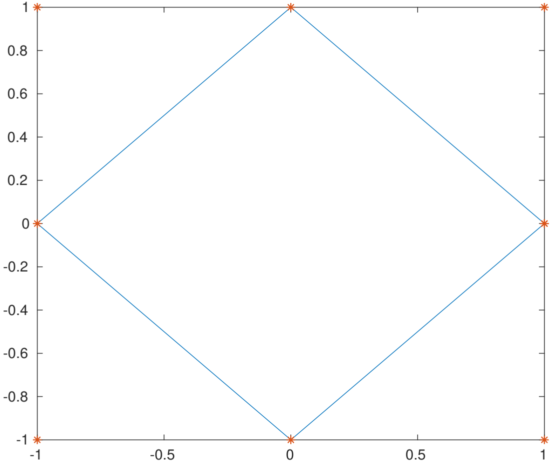
Weights: 1, 10, 1, 10, 1, 10, 1, 10, 1



Weights: $1, 100, 1, 100, 1, 100, 1, 100, 1$



Weights: $1, 0, 1, 0, 1, 0, 1, 0, 1$



Numerical methods

Jiří Zelinka

Autumn 2021, lecture 11

Numerical optimization

f – continuous real function defined on $I = [a, b]$, f takes the minimum value on I at the point $\hat{x} \in I$.

$\hat{x} \in I$ is called the *minimum point* of f.

Function f is called *unimodal* on I if it is decreasing on $[a, \hat{x}]$ and increasing on $[\hat{x}, b]$.

Numerical methods of searching $\hat{x}$:

- comparative methods
- gradient methods

Simple division method

- Let's define equally spaced points $a = x_0, x_1, \ldots, x_{n-1}, x_n = b$ on I.
- Let's find the minimal value $f(x_0), \ldots, f(x_n)$ in x_k.
- $\hat{x} \approx x_k$ with the error $h = \frac{b-a}{n}$ for the unimodal function f.

The method takes to much computations.

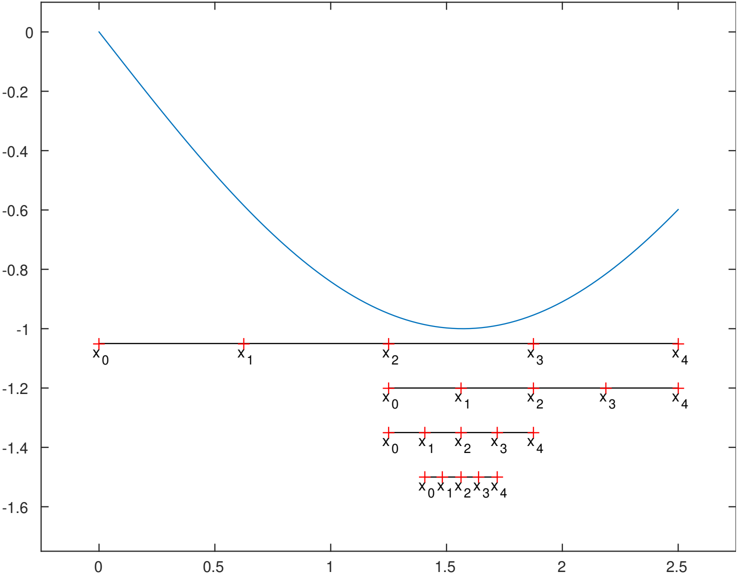
Bisection method

Algorithm

- Let's define equally spaced points $a = x_0, x_1, x_2, x_3, x_4 = b$ on I with the step $h = \frac{b-a}{4}$.
- Let's find the minimal value from $f(x_1), f(x_2), f(x_3)$ in x_k.
- Let's take new interval $[x_{k-1}, x_{k+1}]$ (half of the previous interval).
- Let's repeat (two new points in every step) until the final interval is short enough.

Computational complexity: in each step we calculate two new functional values and the interval is shorten into half length.

Bisection method



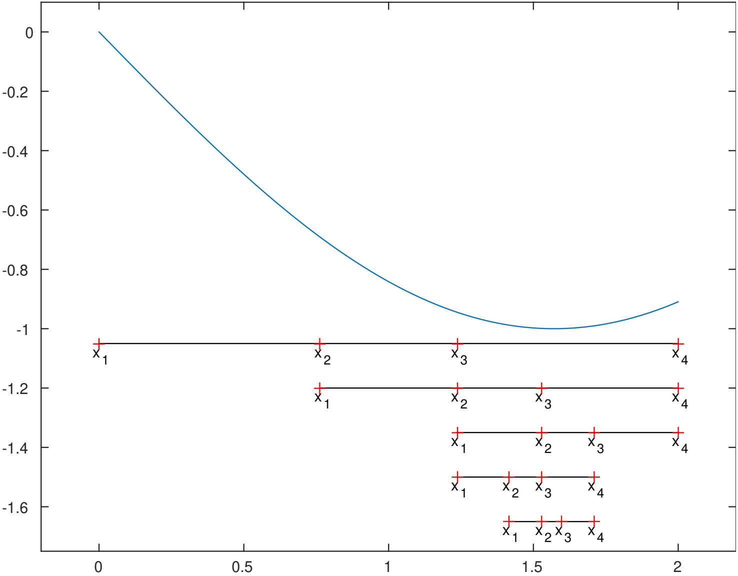
Golden ratio (section) method

Algorithm

- Let's define points $a = x_1, x_2, x_3, x_4 = b$ in I points divided by the golden ratio: $g = \frac{1+\sqrt{5}}{2}$:
$\frac{x_4 - x_1}{x_3 - x_1} = \frac{x_4 - x_1}{x_4 - x_2} = \frac{x_3 - x_1}{x_2 - x_1} = \frac{x_4 - x_2}{x_4 - x_3} = \frac{x_4 - x_3}{x_3 - x_2} = \frac{x_2 - x_1}{x_3 - x_2} = g.$
- Let's find the minimal value from $f(x_2), f(x_3)$ in x_k.
- Let's take new interval $[x_{k-1}, x_{k+1}]$. Its length is $1/g$ of the length the previous interval. Three points of the original four will remain the same
- We calculate the missing point and the functional value, then repeat from the beginning until the interval is small enough.

Computational complexity: in each step we calculate one new functional values and the interval is shorten into $1/g \doteq 0.618$ of the previous length.

Golden ration method



Fibonacci method (search)

This method is equivalent with golden ratio method, asymptotically. It is used if the number of steps $N > 2$ is given.

Fibonacci sequence: $F_0 = F_1 = 1$, $F_{k+1} = F_k + F_{k-1}$, $k = 1, 2, \ldots,$ $\frac{F_{k+1}}{F_k} \to g$.

Algorithm

- The interval $[a, b]$ is divided similarly as in golden ratio method but in ratio of Fibonacci sequence:
- $d_0 = b - a$, $d_1 = d_0 \frac{F_N}{F_{N+1}}$
- $d_k = d_{k-1} \frac{F_{N+1-k}}{F_{N+2-k}}$
- The points are chosen by the same way as in golden ratio method.

Golde ratio vs. Fibonacci

After N steps:

$$d_{G,N} = \frac{d_0}{g^N}, \quad d_{F,N} = \frac{d_0}{F_{N+1}}$$

$$F_N = \frac{g^{N+1} - (-g)^{-N-1}}{\sqrt{5}} \approx \frac{g^{N+1}}{\sqrt{5}} \text{ for large } N$$

$$\frac{d_{G,N}}{d_{F,N}} \approx \frac{g^2}{\sqrt{5}} \doteq 1.17$$

Quadratic interpolation method

A method based on finding the minimum of the interpolation polynomial.

Algorithm

- Let $c \in [a, b]$, c is the middle of the interval, usually.
- We construct an interpolation polynomial (parabola) in points a, b, c.
- We find the minimum at the point d – the zero point of the derivative: $d = \frac{1}{2}\left(a + b - \frac{f[a,b]}{f[a,b,c]}\right)$.
  (Is $f[a, b, c] = 0$ possible?)
- The construction will be repeated for points a, d, c, or c, d, b, respectively, depending on the subinterval containing d.
- If $c = d$, c must be chosen by the other way.

Newton method

This method is used if the analytical expression for the function f is known.

We look for the zero point of the derivative thus we can use the Newton method.

Algorithm

- Let $x_0 \in [a, b]$, x_0 is the middle of the interval, usually.
- $$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}$$
- We need to check if the iteration is within the interval.

Methods for functions of several variables

- Nelder-Mead method
- Gradient method
- Conjugate gradient method
- `https://www.benfrederickson.com/numerical-optimization/`

Numerical methods

Jiří Zelinka

Autumn 2021 – lecture 12

Numerical integration – quadrature formulae

$x_0, \dots, x_n$ – given points, $a \le x_0 < x_1 < \cdots < x_n \le b$
$f_0, \dots, f_n$ – given function values, $f_k = f(x_k)$

Let P be the interpolation polynomial for given data.

$$\int_a^b f(x)dx \approx \int_a^b P(x)dx$$

Equaly spaced points with step h: Newton–Cotes formulae.

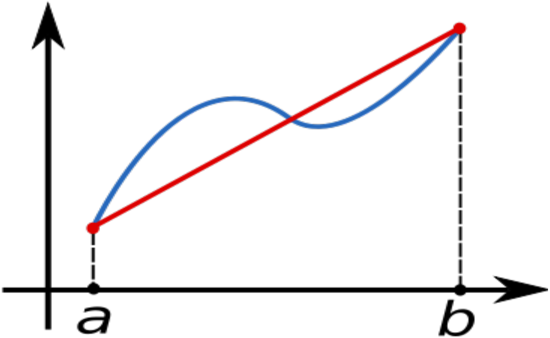
Example 1: trapezoidal rule

$n = 1,\ a = x_0, b = x_1,\ f(a), f(b)$

$P(x) = \frac{f(b)-f(a)}{b-a}(x-a) + f(a)$

$\int_a^b P(x)dx = \left[\frac{f(b)-f(a)}{b-a}\frac{(x-a)^2}{2} + f(a)x\right]_a^b = \frac{f(a)+f(b)}{2}(b-a)$

Trapezoidal rule



Example 2: Simpson's rule

$n = 2$:

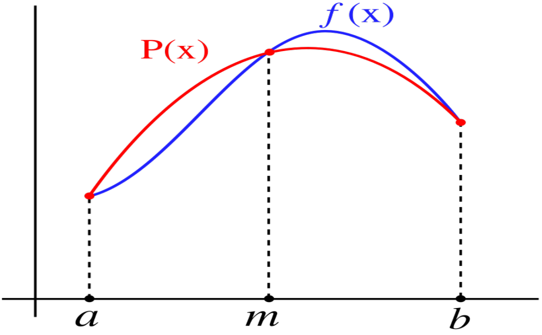
$x_0 = a, x_1 = a + h = \frac{a+b}{2}, x_2 = a + 2h = b,$

f_0, f_1, f_2 – function values, $f_i = f(x_i)$

$$P(x) = f_0 \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} + f_1 \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} + f_2 \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}$$

$$\begin{aligned} \int_a^b f(x)dx \approx \int_a^b P(x)dx &= \frac{b-a}{6}\left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right] \\ &= \frac{h}{3}\left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right] \\ &= \frac{h}{3}\left[f_0 + 4f_1 + f_2\right] \end{aligned}$$

Simpson's rule



Example 3: $3/8$**–rule**

$n = 3$:

$x_0 = a, x_1 = a+h = \frac{2a+b}{3}, x_2 = a+2h = \frac{a+2b}{3}, x_3 = a+3h = b,$
f_0, f_1, f_2, f_3 – function values, $f_i = f(x_i)$

$$\int_a^b f(x)dx \approx \int_a^b P(x)dx = \frac{3h}{8}\left[f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3)\right] = \frac{3h}{8}\left[f_0 + 3f_1 + 3f_2 + f_3\right]$$

Example 4: Milne's rule

$n = 4$:

$x_0 = a, x_1 = a + h, x_2 = a + 2h, x_3 = a + 3h, x_3 = a + 4h = b,$
f_0, f_1, f_2, f_3, f_4 – function values, $f_i = f(x_i)$

$$\int_a^b f(x)dx \approx \int_a^b P(x)dx = \frac{2h}{45}\left[7f_0 + 32f_1 + 12f_2 + 32f_3 + 7f_4\right]$$

General quadrature formula:

$$\int_a^b f(x)dx = Q(f) + E(f), \text{ where}$$

$$Q(f) = \sum_{i=0}^{n} A_i f_i, \quad E(f) \text{ is the error,}$$

A_i: coefficients of the quadrature formula.

Errors for Newton-Cotes formulae:

Trapezoidal rule	$\frac{1}{12}h^3 f^{(2)}(\xi)$
Simpson's rule	$\frac{1}{90}h^5 f^{(4)}(\xi)$
3/8–rule	$\frac{3}{80}h^5 f^{(4)}(\xi)$
Milne's rule	$\frac{8}{945}h^7 f^{(6)}(\xi)$

Definition: Degree of precision

The degree of precision of the quadrature formula $Q(f)$ is $m \in \mathbb{N}$ if $E(P_i) = 0$ for the polynomials P_i of degree $i,\ 0 \le i \le m$ and $E(P_{m+1}) \neq 0$.

Theorem

The quadrature formula obtained by the integration of the interpolation polynomial in points $x_0, \ldots, x_n$ has the degree of precision at least n.

Theorem

The quadrature formula $Q(f) = \sum\limits_{i=0}^{n} A_i f_i$ has the degree of precision at most $2n + 1$.

Gaussian quadrature formulae

Quadrature formulae of degree $2n+1$ (the highest degree). All parameters ($n+1$ points and $n+1$ coefficients) are freely selectable.

Example:

Gauss–Legendre integration

$$\int_{-1}^{1} f(x)dx = \sum_{i=0}^{n} A_i f_i$$

n	x_i	A_i
1	$\mp\sqrt{1/3}$	1
2	0	8/9
	$\mp\sqrt{3/5}$	5/9

Generalisation

$$\int_a^b w(x)f(x)dx = \sum_{i=0}^{n} A_i f_i + E(f), \text{ where}$$

w is so-called weight function including common parts or singularities.

Example: $\int\limits_{-1}^{1} \frac{1}{\sqrt{1-x^2}} f(x)dx$

Composite rules

Composite (chained) trapezoidal rule

Equidistant points:

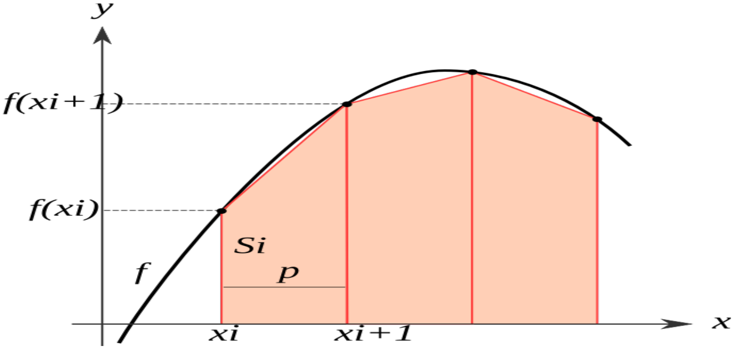
$a = x_0 < x_1 < \cdots < b = x_n,\ x_{i+1}` = x_i + h,\ f_i = f(x_i)$

We use the trapezoidal rule for every interval $[x_i, x_{i+1}]$:

$$\int_a^b f(x)dx \approx$$

$$\approx \frac{f_0 + f_1}{2}h + \frac{f_1 + f_2}{2}h + \frac{f_2 + f_3}{2}h + \cdots + \frac{f_{n-1} + f_n}{2}h =$$

$$= \frac{h}{2}[f_0 + 2f_1 + 2f_2 + \cdots + 2f_{n-1} + f_n]$$


y
f(xi+1)
f(xi)
f
Si
p
xi
xi+1
x

Composite Simpson's rule

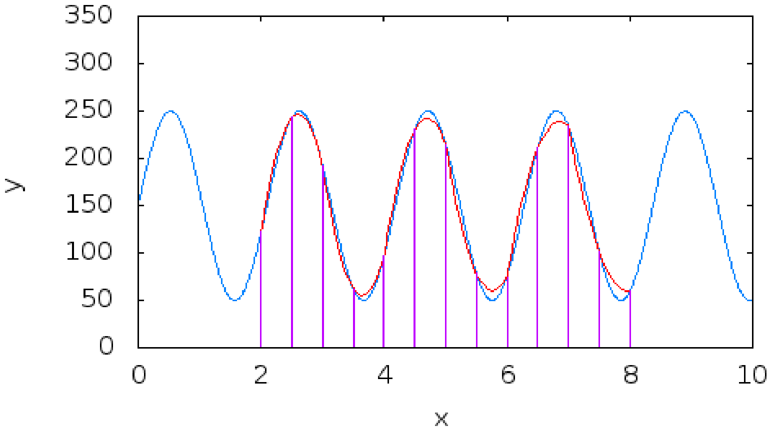
Equidistant points, n – even:
$a = x_0 < x_1 < \cdots < b = x_n,\ x_{i+1} = x_i + h,\ f_i = f(x_i)$

We use the Simpson's rule for every interval $[x_{2i}, x_{2i+2}]$:

$$\int_a^b f(x)dx \approx$$

$$\approx \frac{h}{3}[f_0 + 4f_1 + f_2] + \frac{h}{3}[f_2 + 4f_3 + f_4] + \cdots + + \frac{h}{3}[f_{n-2} + 4f_{n-1} + f_n]$$

$$= \frac{h}{3}[f_0 + 4f_1 + 2f_2 + 4f_3 + 2f_4 + \cdots + 2f_{n-2} + 4f_{n-1} + f_n]$$



350
300
250
200
150
100
50
0
y
0
2
4
6
8
10
x

Monte Carlo integration

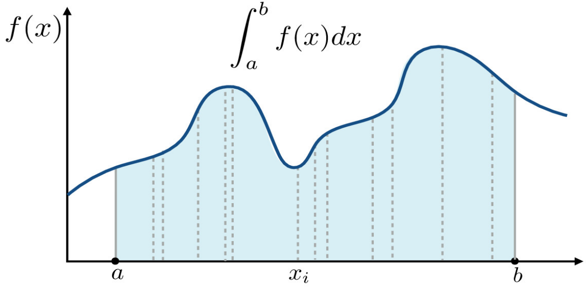
Method I

$X_1, \ldots, X_n$ – random numbers distributed uniformly on $[a, b]$

$$\int_a^b f(x)dx \approx \frac{b-a}{n}\sum_{i=1}^{n} f(X_i)$$

Monte Carlo Integration

Simple idea: estimate the integral of a function by averaging random samples of the function's value.



CS184/284A **Ren Ng**

Monte Carlo integration

Method II

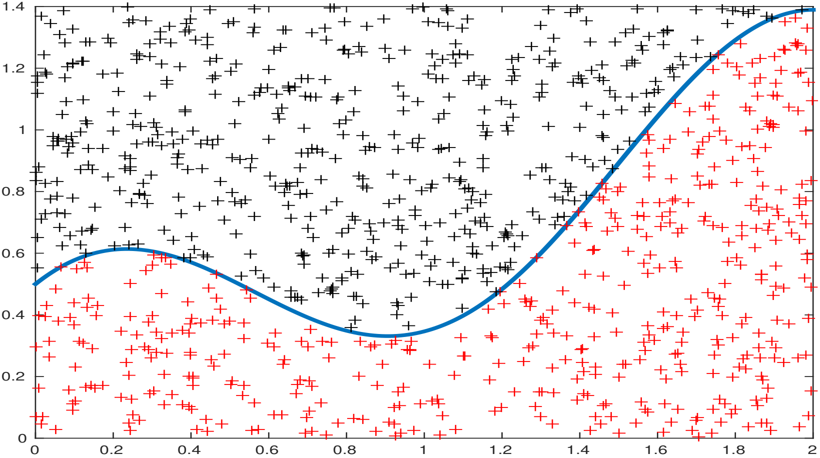
Let f be non-negative on $[a, b]$, $f(x) \leq M$ for every $x \in [a, b]$.

$[X_1, Y_1]$,…,$[X_n.Y_n]$ – observations of the random vector $[X, Y]$ distributed uniformly on $[a, b] \times [0, M]$

$$P(Y \leq f(X)) = \frac{\int\limits_a^b f(x)dx}{M(b-a)} \approx \frac{1}{n}\sum_{i=1}^{n} I_{Y_i \leq f(X_i)}$$

where I is the indicator function.

$$\int\limits_a^b f(x)dx \approx \frac{M(b-a)}{n}\sum_{i=1}^{n} I_{Y_i \leq f(X_i)}$$

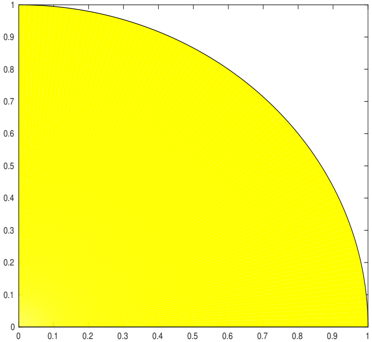


1.4
1.2
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2

Application:

Approximation of π:
$[X, Y]$ distributed uniformly on $[0,1] \times [0,1]$

$P(X^2 + Y^2 \leq 1) = \frac{\pi}{4}$



$[X_1, Y_1]$,. . . ,$[X_n.Y_n]$:
observations of $[X, Y]$

$$\pi \approx \frac{4}{n} \sum_{i=1}^{n} I_{Y_i^2 + X_i^2 \leq 1}$$

Numerical methods

Jiří Zelinka

Autumn 2021 – lecture 13

Numerical calculation of the derivative

$x_0, \dots, x_n$ – given points,
$f_0, \dots, f_n$ – given function values, $f_k = f(x_k)$

We want to calculate the approximation of $f'(x)$ from this data.

Let P be the interpolation polynomial for given data.

$$f'(x) \approx P'(x)$$

Example 1.

$n = 1$,
Data: $x_0, x_1,\ f_0, f_1$

$P(x) = \frac{f_1 - f_0}{x_1 - x_0}(x - x_0) + f_0$

$f'(x) \approx P'(x) = \frac{f_1 - f_0}{x_1 - x_0}$

Example 2.

$n = 2$, data: $x_0, x_1, x_2,\ f_0, f_1, f_2$

$$P(x) = f_0 \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} + f_1 \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} + f_2 \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}$$

$$P'(x) = f_0 \frac{2x-x_1-x_2}{(x_0-x_1)(x_0-x_2)} + f_1 \frac{2x-x_0-x_2}{(x_1-x_0)(x_1-x_2)} + f_2 \frac{2x-x_0-x_1}{(x_2-x_0)(x_2-x_1)}$$

Equally spaced points: $x_1 - x_0 = x_2 - x_1 = h$:

$$P'(x) = f_0 \frac{2x-x_1-x_2}{2h^2} - f_1 \frac{2x-x_0-x_2}{h^2} + f_2 \frac{2x-x_0-x_1}{2h^2}$$

$$P'(x_0) = \frac{1}{2h}(-3f_0 + 4f_1 - f_2)$$

$$P'(x_1) = \frac{1}{2h}(f_2 - f_0)$$

$$P'(x_2) = \frac{1}{2h}(f_0 - 4f_1 + 3f_2)$$

$$P''(x) = \frac{1}{h^2}(f_0 - 2f_1 + f_2)$$

Derivation from the Taylor series

$$
\begin{aligned}
I:\quad & f(x+h) = f(x) + f'(x)h + \tfrac{1}{2}f''(x)h^2 + \tfrac{1}{6}f'''(x)h^3 + O(h^4)\\
II:\quad & f(x-h) = f(x) - f'(x)h + \tfrac{1}{2}f''(x)h^2 - \tfrac{1}{6}f'''(x)h^3 + O(h^4)\\
I-II:\quad & f(x+h) - f(x-h) = 2f'(x)h + \tfrac{1}{3}f'''(x)h^3 + O(h^4)\\
& f'(x) = \tfrac{1}{2h}[f(x+h) - f(x-h)] + O(h^2)\\
I+II:\quad & f(x+h) + f(x-h) = 2f(x) + f''(x)h^2 + O(h^4)\\
& f''(x) = \tfrac{1}{h^2}[f(x+h) - 2f(x) + f(x-h)] + O(h^2)
\end{aligned}
$$

Application: Numerical solution of ordinary differential equations

Boundary problem for linear equation ot the 2nd order

$$p(x)y''(x) + q(x)y'(x) + r(x)y(x) = f(x), \quad x \in [a, b]$$

Boundary conditions: $y(a) = y_a$, $y(b) = y_b$.

Equally spaced knots: $h = (b - a)/N$, $x_0 = a$, $x_N = b$, $x_i = x_0 + i\,h$

Designation: $p_i = p(x_i)$, $q_i = q(x_i)$, $r_i = r(x_i)$, $f_i = f(x_i)$
Numerical solution: $y_i \approx y(x_i)$

Equation in the knot x_i:

$$p_i y''(x_i) + q_i y'(x_i) + r_i y(x_i) = f_i, \quad i = 1, \dots N-1.$$

Approximation of the eqution:

$$p_i \frac{1}{h^2}[y_{i-1} - 2y_i + y_{i+1}] + q_i \frac{1}{2h}[y_{i+1} - y_{i-1}] + r_i y_i = f_i, \quad i = 1, \dots N-1.$$

$$\left(\frac{p_i}{h^2} - \frac{q_i}{2h}\right) y_{i-1} + \left(r_i - 2\frac{p_i}{h^2}\right) y_i + \left(\frac{p_i}{h^2} + \frac{q_i}{2h}\right) y_{i+1} = f_i, \quad i = 1, \dots N-1.$$

The result is the system of linear equations with tridiagonal matrix.

Example:

$$y'' + y = \cos(x), \quad y(0) = 0, \; y(\frac{\pi}{2}) = 1$$

Analytical solution: $y(x) = \sin(x)(\frac{x}{2} + 1 - \frac{\pi}{4})$

Richardson extrapolation

$A(h)$: approximation of A depending on h:

$$A(h) = A + C\,h^n + O(h^{n+1})$$

$$t > 0: \quad A(h/t) = A + C\,\frac{h^n}{t^n} + O(h^{n+1})$$

$$t^n A(h/t) - A(h) = (t^n - 1)A + O(h^{n+1}) \Rightarrow$$

$$\Rightarrow \hat{A}(h,t) = \frac{t^n A(h/t) - A(h)}{t^n - 1} = A + O(h^{n+1})$$

Example: Taylor expansion:

$$f(x+h) = f(x) + c_1 h + c_2 h^2 + \cdots c_k h^k + \cdots, \quad c_k = \frac{f^{(k)}(x)}{k!}$$

$A = f(x)$, $A(h) = f(x+h)$

Multiple usage for $t = 2, 4, 8, \ldots$:

$$A_{0,0} = A(h) = f(x+h) = f(x) + c_1 h + c_2 h^2 + c_3 h^3 + \cdots$$

$$A_{0,1} = A(\frac{h}{2}) = f(x+h/2) = f(x) + c_1 \frac{h}{2} + c_2 \frac{h^2}{4} + c_3 \frac{h^3}{8} + \cdots$$

$$A_{0,2} = A(\frac{h}{4}) = f(x+h/4) = f(x) + c_1 \frac{h}{4} + c_2 \frac{h^2}{16} + c_3 \frac{h^3}{64} + \cdots$$

$$A_{0,3} = A(\frac{h}{8}) = f(x+h/8) = f(x) + c_1 \frac{h}{8} + c_2 \frac{h^2}{64} + c_3 \frac{h^3}{512} + \cdots$$

$$A_{0,k} = A(\frac{h}{2^k}) = f(x+h/2^k) = f(x) + c_1 \frac{h}{2^k} + c_2 \frac{h^2}{2^{2k}} + c_3 \frac{h^3}{2^{3k}} + \cdots$$

$$A_{1,0} = 2A_{0,1} - A_{0,0} = f(x) - \frac{1}{2}c_2h^2 - \frac{3}{4}c_3h^3 - \frac{7}{8}c_4h^4 - \cdots$$

$$A_{1,1} = 2A_{0,2} - A_{0,1} = f(x) - \frac{1}{8}c_2h^2 - \frac{3}{32}c_3h^3 - \frac{7}{128}c_4h^4 - \cdots$$

$$A_{1,2} = 2A_{0,3} - A_{0,2} = f(x) - \frac{1}{32}c_2h^2 - \frac{3}{256}c_3h^3 - \frac{7}{2048}c_4h^4 - \cdots$$

$$A_{2,0} = \frac{4A_{1,1} - A_{1,0}}{3} = f(x) + \frac{1}{8}c_3h^3 + \frac{7}{32}c_4h^4 + \cdots$$

$$A_{2,1} = \frac{4A_{1,2} - A_{1,1}}{3} = f(x) + \frac{1}{64}c_3h^3 + \frac{7}{256}c_4h^4 + \cdots$$

General formula:

$$A_{j,k} = \frac{2^j A_{j-1,k+1} - A_{j-1,k}}{2^j - 1} = f(x) + O(h^{j+1})$$

Taylor's expansion containing only even powers:

$$f(x+h) = f(x) + c_1 h^2 + c_2 h^4 + \cdots c_k h^{2k} + \cdots, \quad c_k = \frac{f^{(2k)}(x)}{(2k)!}$$

$$A_{j,k} = \frac{4^j A_{j-1,k+1} - A_{j-1,k}}{4^j - 1} = f(x) + O(h^{2(j+1)})$$

Example:

Calculation of π
Archimedes: $\frac{223}{71} < \pi < \frac{22}{7}$ by perimeters of regular n-gons for $n = 6, 12, 24, 48, 96$.

Romberg integration

$A(h)$: numerical integral using *Composite Trapezoidal Rule*
The error of $A(h)$ can be expressed using Taylor's expansion containing only even powers
$\Rightarrow$
we use Richardson extrapolation for $A_{0,0} = A(h)$, $A_{0,1} = A(h/2)$, $A_{0,2} = A(h/4)$. ...