

# 11MAG – 9. Cvičení

## Interpolace a aproximace funkcí

15. prosince 2014

### 1 Interpolace Lagrangeovým polynomem

V přednášce jsme si popsali princip konstrukce Lagrangeova interpolačního polynomu. Vytvořte podle tohoto popisu nejprve funkci `L=lagr_ctor(x,y)`, jež pro zadané interpolační uzly  $x_i$  a funkční hodnoty  $y_i = f(x_i)$ ,  $i = 0, 1, 2, \dots, n$ , shromáždí informaci nutnou pro budoucí vyčíslení celkem  $n + 1$  Lagrangeových bázových polynomů  $\ell_i(x)$  stupně  $n$  a jejich škáluje funkčními hodnotami  $f(x_i)$  interpolované funkce v uzlových bodech.

Výstupem funkce je matice  $\mathbf{L}_{(n+1) \times (n+1)}$  ve tvaru

$$\mathbf{L} = \begin{bmatrix} \frac{f(x_0)}{\prod_{k \neq 0} (x_0 - x_k)} & x_1 & x_2 & \cdots & x_n \\ \frac{f(x_1)}{\prod_{k \neq 1} (x_1 - x_k)} & x_0 & x_2 & \cdots & x_n \\ \vdots & & & \ddots & \\ \frac{f(x_n)}{\prod_{k \neq n} (x_n - x_k)} & x_0 & x_1 & \cdots & x_{n-1} \end{bmatrix}.$$

S maticí  $\mathbf{L}$  nebudeme provádět žádné maticové operace, používáme ji pouze jako kontejner pro uchování hodnot potřebných k výpočtu interpolačního polynomu. Stejně tak bychom mohli použít nějakou jinou vhodnou datovou strukturu.

Druhou funkcí, kterou je potřeba vytvořit, je funkce `y=lagr_eval(x,L)`, která pro maticové schéma datné obsahem matice  $\mathbf{L}$  vyčísleje Lagrangeův interpolační polynom  $L_n(x)$ , tedy součet

$$L_n(x) = \sum_{i=0}^n f(x_i) \ell_i(x) = \sum_{i=0}^n \frac{f(x_i)}{\prod_{k \neq i} (x_i - x_k)} \prod_{k \neq i} (x - x_k) = \sum_{i=1}^{n+1} \left( l_{i,1} \prod_{k=2}^{n+1} (x - l_{i,k}) \right).$$

Připomínám, že  $l_{i,j}$  označuje prvek v  $i$ -tém řádku a  $j$ -tém sloupci matice  $\mathbf{L}$ .

### 2 Intepolace Rungovy funkce

Pomocí funkcí vytvořených výše interpolujte Rungovu funkci

$$f(x) = \frac{1}{1 + 25x^2}$$

a ukažte, že pro rostoucí počet interpolačních uzlů vede jejich ekvidistantní rozmístění k velkým nepřesnostem na okrajích interpolačního intervalu. Obdobně jako v přednášce uvažujte interpolační interval  $\langle -1; 1 \rangle$ .

Rungovu funkci nejprve implementujte jako `y=rungefnc(x)`.

## 2.1 Ekvidistantní uzly

Zvolte ekvidistantní interpolační uzly pro interpolační polynom 5. a 10. stupně (uzlů bude tedy 6 respektive 11 a budou rozmístěny ve stejné vzdálenosti od sebe na intervalu  $\langle -1; 1 \rangle$ ) a uložte je do vektorů `xe5` a `xe10`. Vyhodnoťte Rungovu funkci pro tyto vektory a výsledek uložte do vektorů `ye5` a `ye10`.

Použijte funkci `lagr_ctor()` ke konstrukci matic naškálovaných báзовých polynomů `Le5` a `Le10` a funkci `lagr_eval()` k vyhodnocení  $L_5(x)$  a  $L_{10}(x)$  bodech

$$\mathbf{x} = -1,00, 0,98, 0,96, \dots, -0,02, 0,00, 0,02, \dots, 0,98, 1,00.$$

V Matlabu lze vektor  $\mathbf{x}$  vytvořit jednoduše jako `x=-1:0.02:1`.

Vykreslete do jednoho grafu průběh funkce  $f(x)$  a odpovídajících interpolačních polynomů  $L_5(x)$  a  $L_{10}(x)$ .

Vyhodnoťte diskretní normu chyby  $\|E\|_2^{41}$  a diskretní Čebyševovu normu  $\|E\|_\infty^{41}$  obou interpolačních polynomů pro výše zadaný vektor hodnot  $\mathbf{x}$ .

## 2.2 Čebyševovy uzly

Zvolte opět  $n + 1 = 6$  respektive 11 interpolačních uzlů rozmístěných tentokrát na intervalu  $\langle -1; 1 \rangle$  nerovnoměrně v bodech

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

a uložte je do vektorů `xc5` a `xc10`. Vyhodnoťte Rungovu funkci pro tyto vektory a výsledek uložte do vektorů `yc5` a `yc10`.

Použijte funkci `lagr_ctor()` ke konstrukci matic naškálovaných báзовých polynomů `Lc5` a `Lc10` a funkci `lagr_eval()` k vyhodnocení  $L_5(x)$  a  $L_{10}(x)$  bodech

$$\mathbf{x} = -1,00, 0,98, 0,96, \dots, -0,02, 0,00, 0,02, \dots, 0,98, 1,00.$$

Vykreslete do jednoho grafu průběh funkce  $f(x)$  a odpovídajících interpolačních polynomů  $L_5(x)$  a  $L_{10}(x)$ .

Vyhodnoťte diskretní normu chyby  $\|E\|_2^{41}$  a diskretní Čebyševovu normu  $\|E\|_\infty^{41}$  obou interpolačních polynomů pro výše zadaný vektor hodnot  $\mathbf{x}$ .

## 3 Interpolace kubickými splajny [bonus]

Jako *interpolační splajn* označujeme v oboru numerické matematiky jakoukoliv dostatečně hladkou a po částech polynomiální interpolační funkci určenou k interpolaci neznámé funkce  $f(x)$  zadané tabulkou  $n + 1$  funkčních hodnot  $y_0, y_1, \dots, y_n$ . Nejčastěji používáme kubické splajny, konstruované po částech z polynomů

$$\varphi_j(t) = a_j + b_j t + c_j t^2 + d_j t^3,$$

kde  $t \in \langle 0; 1 \rangle$  a  $j = 0, 1, \dots, n - 1$ . Polynom  $\varphi_j(t)$  interpoluje pro  $t = (x - x_j)/(x_{j+1} - x_j)$  původní funkci  $f(x)$  mezi uzlovými body  $x_j$  a  $x_{j+1}$ .

### 3.1 Výpočet koeficientů

Koeficienty  $a_j, b_j, c_j, d_j$  lze určit z podmínek pro „hladkou“ návaznost interpolační funkce (spojitost dílčích interpolačních polynomů v uzlech až do druhé derivace) následujícím postupem. Nejprve si připravíme

$$\begin{aligned}\varphi_j(0) &= y_j = a_j \\ \varphi_j(1) &= y_{j+1} = a_j + b_j + c_j + d_j \\ \varphi'_j(0) &= D_j = b_j \\ \varphi'_j(1) &= D_{j+1} = b_j + 2c_j + 3d_j.\end{aligned}$$

Za předpokladu, že známe hodnoty  $y_j$ ,  $y_{j+1}$ ,  $D_j$  a  $D_{j+1}$  lze z této soustavy určit  $a_j$ ,  $b_j$ ,  $c_j$ ,  $d_j$  jako

$$\begin{aligned} a_j &= y_j \\ b_j &= D_j \\ c_j &= 3(y_{j+1} - y_j) - 2D_j - D_{j+1} \\ d_j &= -2(y_{j+1} - y_j) + D_j + D_{j+1}. \end{aligned} \quad (1)$$

Zbývá určit hodnoty  $D_j$ . K tomu využijeme požadavku na spojitost až do druhé derivace v interpolačních uzlech,

$$\begin{aligned} \varphi_j(0) &= y_j \\ \varphi_{j-1}(1) &= y_j \\ \varphi'_{j-1}(1) &= \varphi'_j(0) = D_j \\ \varphi''_{j-1}(1) &= \varphi''_j(0), \end{aligned}$$

přičemž pro koncové uzly položíme druhé derivace rovné nule (je to volba stejně špatná, jako všechny ostatní volby, protože o funkci  $f(x)$  nevíme kromě funkčních hodnot v uzlech vůbec nic),

$$\begin{aligned} \varphi_0(0) &= y_0 \\ \varphi_{n-1}(1) &= y_n \\ \varphi''_0(0) &= 0 \\ \varphi''_{n-1}(1) &= 0. \end{aligned}$$

Celé to lze zapsat jako soustavu lineárních rovnic s třídiagonální maticí soustavy (pozor při implementaci na pravou stranu, první a poslední řádek vektoru  $\mathbf{b}$  má jinak sestavené hodnoty, než zbylé složky):

$$\mathbf{Ad} = \begin{pmatrix} 2 & 1 & & & \\ 1 & 4 & 1 & & \\ & 1 & 4 & 1 & \\ & & \ddots & \ddots & \ddots \\ & & & 1 & 4 & 1 \\ & & & & 1 & 2 \end{pmatrix} \begin{pmatrix} D_0 \\ D_1 \\ D_2 \\ \vdots \\ D_{n-1} \\ D_n \end{pmatrix} = \begin{pmatrix} 3(y_1 - y_0) \\ 3(y_2 - y_0) \\ 3(y_3 - y_1) \\ \vdots \\ 3(y_n - y_{n-2}) \\ 3(y_n - y_{n-1}) \end{pmatrix} = \mathbf{b}. \quad (2)$$

Tuto soustavu lze v Matlabu řešit jako  $\mathbf{d}=\mathbf{A} \backslash \mathbf{b}$ , případně můžete využít implementace GEM z minulého cvičení.

### 3.2 Implementace

Vytvořte podle tohoto popisu nejprve funkci  $\mathbf{S}=\text{spl\_ctor}(\mathbf{x},\mathbf{y})$ , jež pro zadané interpolační uzly  $x_i$  a funkční hodnoty  $y_i = f(x_i)$ ,  $i = 0, 1, 2, \dots, n$ , zkonstruuje kubický splajn  $\varphi(x)$  složený z celkem  $n$  kubických polynomů  $\varphi_j(x)$  stupně 3 tak, že nejprve určí vektor  $\mathbf{d}$  hodnot  $D_j$  ze soustavy (2) a poté jednotlivé koeficienty ze soustavy (1).

Výstupem funkce je matice  $\mathbf{S}_{(n+1) \times 5}$  ve tvaru

$$\mathbf{S} = \begin{bmatrix} x_0 & a_0 & b_0 & c_0 & d_0 \\ x_1 & a_1 & b_1 & c_1 & d_1 \\ & & \vdots & & \\ x_{n-1} & a_{n-1} & b_{n-1} & c_{n-1} & d_{n-1} \\ x_n & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Druhou funkcí, kterou je potřeba vytvořit, je funkce  $\mathbf{y}=\text{spl\_eval}(\mathbf{x},\mathbf{S})$ , která vyčísluje splajn  $\varphi(x)$  definovaný mezi uzly  $x_0$  a  $x_n$ . Funkce

1. nejprve podle polohy  $x$  určí odpovídající  $j = 0, 1, \dots, n - 1$ , potom
2. vypočte odpovídající hodnotu  $t = (x - x_j)/(x_{j+1} - x_j)$  a nakonec
3. na základě známých hodnot  $t$  a  $j$  vyhodnotí

$$\varphi_j(t) = a_j + b_j t + c_j t^2 + d_j t^3$$

například pomocí Hornerova schématu.

### 3.3 Vlastní úkol

Použijte Rungovu funkci `rungefnc(x)` definovanou v Odstavci 2. Zvolte ekvidistantní interpolační uzly obdobně, jako pro interpolační polynom 5. a 10. stupně v odstavci 2.1 (uzlů bude tedy 6 respektive 11 a budou rozmístěny ve stejné vzdálenosti od sebe na intervalu  $\langle -1; 1 \rangle$ ) a uložte je do vektorů `xe5` a `xe10`. Vyhodnoťte Rungovu funkci pro tyto vektory a výsledek uložte do vektorů `ye5` a `ye10`.

Použijte funkci `spl_ctor()` ke konstrukci maticových schémat `S5` a `S10` a funkci `spl_eval()` k vyhodnocení  $\varphi(x)$  v bodech

$$\mathbf{x} = -1,00, 0,98, 0,96, \dots, -0,02, 0,00, 0,02, \dots, 0,98, 1,00.$$

V Matlabu lze vektor  $\mathbf{x}$  vytvořit jednoduše jako `x=-1:0.02:1`.

Vykreslete do jednoho grafu průběh funkce  $f(x)$  a odpovídajících splajnů  $\varphi(x)$ .

Vyhodnoťte diskrétní normu chyby  $\|E\|_2^{41}$  a diskrétní Čebyševovu normu  $\|E\|_\infty^{41}$  obou splajnů pro výše zadaný vektor hodnot  $\mathbf{x}$ .