# Tomáš Oberhuber

Faculty of Nuclear Sciences and Physical Engineering
Czech Technical University in Prague

Video na Youtube

# Polynomiální aproximace

- v numerické matematice často pracujeme s funkcemi, které nelze vyjádřit analyticky
- nelze je tedy vyjádřit přesně, ale lze je alespoň aproximovat
- někdy zase známe některou závislost jen z experimentu tj. známe funkční hodnoty jen v několika bodech
- pro svou jednoduchost se nejčastěji volí polynomiální aproximace

# Lagrangeův polynom

- jednou možností by byl Taylorův polynom
- problém je v tom, že ten potřebuje znát derivace poměrně vysokých řádů
- měření derivací experimentálně je ale velmi složité až nemožné
- většinou naměříme pouze funkční hodnoty, ale zato v několika různých bodech
- místo Taylorova polynomu proto použijeme **Lagrangeův polynom**

# Lagrangeův polynom

## Matematická formulace problému

### Remark 1

*Buď* $f : \mathbb{R} \to \mathbb{R}$ *funkce jejíž hodnoty známe ve vzájemně různých bodech* $x_0 < x_1 < \ldots < x_n$. *Hledáme polynom* $L_n(x)$ *co nejnižšího stupně tak, aby platilo*

$$L_n(x_i) = f(x_i) \quad \text{pro} \quad i = 0, \ldots, n.$$

*Za* **vhodných podmínek** *by mohlo platit, že* $L_n(x)$ *bude blízko* $f(x)$ *i v ostatních bodech. Odhadujeme-li hodnotu f mezi body* $x_0, \ldots, x_n$, *jde o* **interpolaci** *jinak jde o* **extrapolaci**.

# Lagrangeův polynom

Je-li

$$L_n(x) = \sum_{i=0}^{n} a_i x^i,$$

pak lze podmínku $L_n(x_i) = f(x_i)$ pro $i = 0, \ldots, n$ přepsat jako

$$\begin{pmatrix} 1 & x_0 & x_0^2 & \ldots & x_0^n \\ 1 & x_1 & x_1^2 & \ldots & x_1^n \\ 1 & x_2 & x_2^2 & \ldots & x_2^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \ldots & x_n^n \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} f(x_0) \\ f(x_1) \\ f(x_2) \\ \vdots \\ f(x_n) \end{pmatrix}$$

resp.

$$\mathbb{V}^{(n)}_{x_0,\ldots,x_n} \vec{a} = \vec{f}.$$

# Lagrangeův polynom

Koeficienty $a_0, \ldots, a_n$ lze tedy získat vyřešením soustavy lineárních rovnic

$$\mathbb{V}^{(n)}_{x_0,\ldots,x_n} \vec{a} = \vec{f}.$$

## Definition 2

Nechť $x_0, \ldots, x_n \in \mathbb{R}$. Matice $\mathbb{V}^{(m)}_{x_0,\ldots,x_n} \in \mathbb{R}^{n+1,m+1}$ definovaná jako

$$\mathbb{V}^{(m)}_{x_0,\ldots,x_n} = \begin{pmatrix} 1 & x_0 & x_0^2 & \ldots & x_0^m \\ 1 & x_1 & x_1^2 & \ldots & x_1^m \\ 1 & x_2 & x_2^2 & \ldots & x_2^m \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \ldots & x_n^m \end{pmatrix}$$

se nazývá Vandermondova matice.

# Lagrangeův polynom

## Theorem 3

*Nechť $x_0, \ldots, x_n \in \mathbb{R}$ jsou navzájem různé. Pak příslušná Vandermondova matice $\mathbb{V}^{(n)}_{x_0,\ldots,x_n} \in \mathbb{R}^{n+1,n+1}$*

$$\mathbb{V}^{(n)}_{x_0,\ldots,x_n} = \begin{pmatrix} 1 & x_0 & x_0^2 & \ldots & x_0^n \\ 1 & x_1 & x_1^2 & \ldots & x_1^n \\ 1 & x_2 & x_2^2 & \ldots & x_2^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \ldots & x_n^n \end{pmatrix}$$

*je regulární.*

Důkaz.

Video na Youtube

# Lagrangeův polynom

## Theorem 4

*Buď $f : \mathbb{R} \to \mathbb{R}$ a body $x_0, \ldots, x_n \in D_f$. Pak existuje právě jeden polynom $P$ stupně $n$ splňující*

$$P(x_i) = f(x_i) \quad \text{pro} \quad \forall i = 0, \ldots, n.$$

## Důkaz.

Důkaz plyne z regularity matice $\mathbb{V}^{(n)}_{x_0,\ldots,x_n}$.
Alternativní důkaz: Video na Youtube

# Lagrangeův polynom

Co kdybychom zkoušeli interpolaci **polynomem nižšího stupně**?

- jednodušší polynom by byl praktičtější
- museli bychom řešit soustavu rovnic tvaru

$$\begin{pmatrix} 1 & x_0 & \dots & x_0^m \\ 1 & x_1 & \dots & x_1^m \\ 1 & x_2 & \dots & x_2^m \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & \dots & x_n^m \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_m \end{pmatrix} = \begin{pmatrix} f(x_0) \\ f(x_1) \\ f(x_2) \\ f(x_3) \\ \vdots \\ f(x_n) \end{pmatrix}$$

  pro $n > m$, která obecně nemá řešení.
- tj. polynom nižšího stupně obecně nemůže nabývat všech předepsaných hodnot

# Lagrangeův polynom

Co kdybychom zkoušeli interpolaci **polynomem vyššího stupně**?

- polynom vyššího stupně by mohl možná dát lepší interpolaci
- museli bychom řešit soustavu rovnic tvaru

$$\begin{pmatrix} 1 & x_0 & x_0^2 & x_0^3 & \dots & x_0^m \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & x_n^3 & \dots & x_n^m \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_m \end{pmatrix} = \begin{pmatrix} f(x_0) \\ \vdots \\ f(x_n) \end{pmatrix}$$

  pro $n < m$, která má nekonečně mnoho řešení
- tj. polynom vyššího stupně není určen jednoznačně

# Lagrangeův polynom

- matice $\mathbb{V}^{(n)}_{x_0,\ldots,x_n}$ může být špatně podmíněná
- konstrukce Lagrangeova polynomu řešením soustavy

$$\mathbb{V}^{(n)}_{x_0,\ldots,x_n}\vec{a} = \vec{f}$$

  může být tedy numericky nestabilní
- proto se raději používají jiné postupy pro konstrukci
  - **Lagrangeův tvar**
  - **Newtonova formule (Newtonův tvar)**

# Lagrangeův tvar

- pro $i = 0, \ldots, n$ definujeme polynomy

$$l_i = \prod_{j=0, j\neq i}^{n} \frac{x - x_j}{x_i - x_j},$$

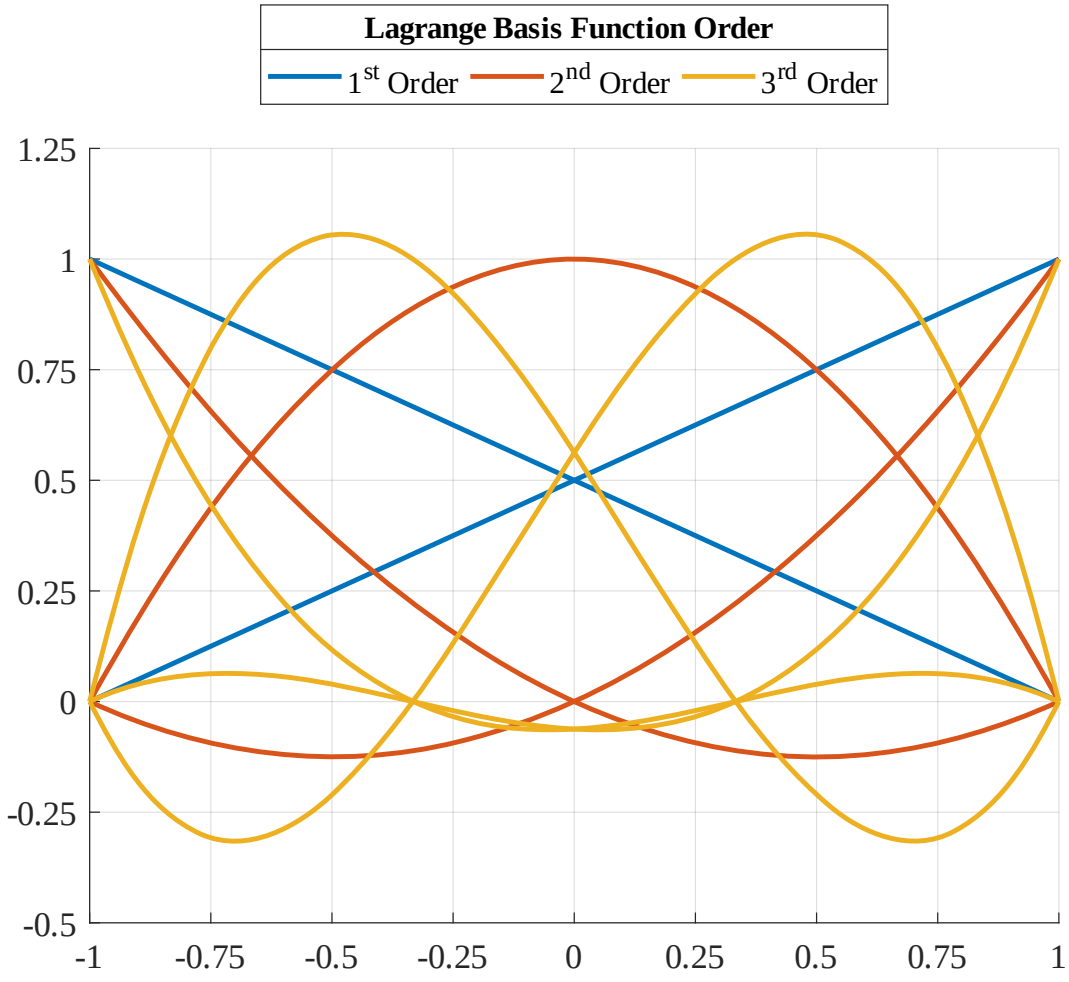
- pro ně platí

$$l_i(x_j) = \delta_{ij}.$$

- Lagrangeův interpolační polynom je pak definován jako

$$L_n(x) = \sum_{i=0}^{n} f(x_i) l_i(x).$$

# Lagrangeův tvar




Zdroj: Wikipedie

# Lagrangeův tvar

## Example 5

Napočítejte Lagrangeův interpolační polynom pro funkci $f$ se znalostí funkčních hodnot $f(-1) = 1, f(0) = 0, f(1) = 1$.

**Řešení:**

Je tedy $n = 2$ a $x_0 = -1, x_1 = 0, x_2 = 1$.

Dále platí

$$\begin{aligned} l_0(x) &= \frac{x-x_1}{x_0-x_1}\frac{x-x_2}{x_0-x_2} &&= \frac{1}{2}x(x-1), \\ l_1(x) &= \frac{x-x_0}{x_1-x_0}\frac{x-x_2}{x_1-x_2} &&= -(x+1)(x-1), \\ l_2(x) &= \frac{x-x_0}{x_2-x_0}\frac{x-x_1}{x_2-x_1} &&= \frac{1}{2}x(x+1). \end{aligned}$$

a

$$L_2(x) = \frac{1}{2}x(x-1) + 0[-(x+1)(x-1)] + \frac{1}{2}x(x+1) = x^2.$$

## Example 6

Napočítejte Lagrangeův interpolační polynom pro funkci $f$ se znalostí funkčních hodnot $f(-1) = 1, f(0) = 1, f(1) = 1$.

# Lagrangeův tvar

- definice Lagrangeova polynomu není pro výpočet příliš vhodná
- pro každé nové $x$ potřebujeme znovu napočítat $l_i(x)$ pro $i = 0, 1, \ldots n$
- jmenovatele na $x$ nezávisí, ale čitatele ano tj. musíme přepočítat celkem $n^2$ členů
- ukážeme si, jak tento polynom napočítat efektivněji
- použijeme **Newtonovu formuli**.

# Newtonova formule

- víme, že existuje právě jeden interpolační polynom $L_{k-1}(x)$ stupně nejvýše $k-1$, pro který platí

$$L_{k-1}(x_i) = f(x_i) \text{ pro } i = 0, 1, \ldots k-1$$

- pro polynom tvaru

$$L_k(x) = L_{k-1}(x) + c_k(x - x_0)(x - x_1)\ldots(x - x_{k-1})$$

tedy také platí

$$L_k(x_i) = f(x_i) \text{ pro } i = 0, 1, \ldots k-1$$

- vhodnou volbou $c_k$ dosáhneme i rovnosti $L_k(x_k) = f(x_k)$

# Newtonova formule

- musí platit

$$f(x_k) = L_k(x_k) = L_{k-1}(x_k) + c_k(x_k - x_0)(x_k - x_1)\dots(x_k - x_{k-1})$$

- odkud jen vyjádříme $c_k$

$$c_k = \frac{f(x_k) - L_{k-1}(x_k)}{(x_k - x_0)(x_k - x_1)\dots(x_k - x_{k-1})}$$

- koeficienty $c_0, \dots, c_n$ lze tedy napočítat rekurentně podle předchozího vztahu
- ukážeme si jednodušší způsob s pomocí **poměrných diferencí**

# Newtonova formule

Obecně lze psát

$$\begin{aligned}
L_n(x) =& L_{n-1}(x) + c_n(x-x_0)(x-x_1)\dots(x-x_{n-1}) \\
=& L_{n-2}(x) + c_{n-1}(x-x_0)(x-x_1)\dots(x-x_{n-2}) \\
&+ c_n(x-x_0)(x-x_1)\dots(x-x_{n-1}) \\
&\dots \\
=& c_0 + c_1(x-x_0) + c_2(x-x_0)(x-x_1) + \dots \\
&+ c_n(x-x_0)\dots(x-x_{n-1})
\end{aligned}$$

nebo také

$$L_n(x) = \sum_{i=0}^{n} c_i \prod_{j=0}^{i-1}(x-x_j).$$

# Newtonova formule

- definujeme

$$\mathbb{B}_{ij} = \begin{cases} 0 & \text{pro} \quad j > i \\ \prod_{j=0}^{i-1}(x - x_j) & \text{pro} \quad j \leq i \end{cases}$$

- pak lze vztahy

$$L_i(x_i) = f(x_i) \text{ pro } i = 0, \ldots, n$$

psát jako $\mathbb{B}\vec{c} = \vec{f}$ nebo také

$$\begin{pmatrix} 1 & 0 & 0 & \ldots & 0 \\ 1 & x_1 - x_0 & 0 & \ldots & 0 \\ 1 & x_2 - x_0 & (x_2 - x_0)(x_2 - x_1) & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 1 & x_n - x_0 & (x_n - x_0)(x_n - x_1) & \ldots & \prod_{k=0}^{n-1}(x_n - x_k) \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} f(x_0) \\ f(x_1) \\ f(x_2) \\ \vdots \\ f(x_n) \end{pmatrix}$$

# Newtonova formule

- z tvaru matice $\mathbb{B}$ vidíme, že
  - $c_0$ zavisí jen na $f(x_0)$ tj.

$$c_0 = f[x_0] = f(x_0)$$

  - $c_1$ zavisí jen na $f(x_0), f(x_1)$ tj.

$$c_1 = f[x_0, x_1]$$

  - obecně $c_k$ závisí na $f(x_0), \ldots, f(x_k)$ tj.

$$c_k = f[x_0, \ldots, x_k]$$

- lze pak tedy psát

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

# Newtonova formule

## Theorem 7

*Pro koeficienty v Newtonově formuli platí vztah*

$$f[x_i, \ldots, x_{i+k}] = \frac{f[x_{i+1}, \ldots, x_{i+k}] - f[x_i, \ldots, x_{i+k-1}]}{x_{i+k} - x_i}. \qquad (1)$$

## Důkaz.

Video na Youtube

□

## Definition 8

Výrazy tvaru $f[x_i, \ldots, x_i + k]$ splňující vztah (1) nazývámě **poměrná diference** (*divided differences*) $k$-tého řádu. Poměrnou diferenci nultého řádu definujeme jako $f[x_i] = f(x_i)$.

# Poměrné diference

Příklady konkrétních poměrných diferencí:

$$f[x_0, x_1] = \frac{f[x_1] - f[x_0]}{x_1 - x_0},$$

$$f[x_1, x_2] = \frac{f[x_2] - f[x_1]}{x_2 - x_1},$$

$$f[x_2, x_3] = \frac{f[x_3] - f[x_2]}{x_3 - x_2},$$

$$f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0},$$

$$f[x_1, x_2, x_3] = \frac{f[x_2, x_3] - f[x_1, x_2]}{x_3 - x_1},$$

$$f[x_0, x_1, x_2, x_3] = \frac{f[x_1, x_2, x_3] - f[x_0, x_1, x_2]}{x_3 - x_0}.$$

# Lagrangeův polynom - Newtonova formule

Poměrné diference napočítáváme po sloupcích v následující tabulce:

$$
\begin{array}{llllll}
x_0 & f(x_0) & & & & \\
x_1 & f(x_1) & f[x_0,x_1] & & & \\
x_2 & f(x_2) & f[x_1,x_2] & f[x_0,x_1,x_2] & & \\
x_3 & f(x_3) & f[x_2,x_3] & f[x_1,x_2,x_3] & \ddots & \\
x_4 & f(x_4) & f[x_3,x_4] & f[x_2,x_3,x_4] & & \\
\vdots & \vdots & \vdots & \vdots & & \ddots \\
x_{n-4} & f(x_{n-4}) & f[x_{n-5},x_{n-4}] & f[x_{n-6},x_{n-5},x_{n-4}] & & \\
x_{n-3} & f(x_{n-3}) & f[x_{n-4},x_{n-3}] & f[x_{n-5},x_{n-4},x_{n-3}] & & \\
x_{n-2} & f(x_{n-2}) & f[x_{n-3},x_{n-2}] & f[x_{n-4},x_{n-3},x_{n-2}] & & \\
x_{n-1} & f(x_{n-1}) & f[x_{n-2},x_{n-1}] & f[x_{n-3},x_{n-2},x_{n-1}] & & \\
x_n & f(x_n) & f[x_{n-1},x_n] & f[x_{n-2},x_{n-1},x_n] & \dots & \dots \quad f[x_0,x_1,\dots,x_n]
\end{array}
$$

# Lagrangeův polynom - Newtonova formule

## Example 9

Pomocí Newtonovy formule napočítejte Lagrangeův interpolační polynom pro funkci $f$ se znalostí funkčních hodnot $f(-1) = 1, f(0) = 0, f(1) = 1$ za předpokladu, že $f$ je dostatečně hladká.

**Řešení:**

$$\begin{aligned} f[x_0, x_1] &= \frac{f(x_1)-f(x_0)}{x_1-x_0} = -1, \\ f[x_1, x_2] &= \frac{f(x_2)-f(x_1)}{x_2-x_1} = 1, \\ f[x_0, x_1, x_2] &= \frac{f[x_1,x_2]-f[x_1,x_0]}{x_2-x_0} = 1. \end{aligned}$$

A tedy

$$\begin{aligned} L_2(x) &= f(x_0) + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) \\ &= 1 - 1(x - x_0) + 1(x - x_0)(x - x_1) \\ &= 1 - (x + 1) + 1(x + 1)x \\ &= x^2. \end{aligned}$$

# Lagrangeův polynom - Newtonova formule

```
1  void dividedDifferences( const double nodes[ n+1 ],
2                           const double fx[ n+1 ],
3                           double differences[ n+1][ n+1 ] )
4  {
5     for( int i = 0; i <= n; i++ )
6        differences[ i ][ 0 ] = fx[ i ];
7     for( int j = 1; j <= n; j++ )
8     {
9        for( int i = j; i <= n; i++ )
10          differences[ i ][ j ] =
11             ( differences[ i ][ j-1 ] -
12               differences[ i-1 ][ j-1 ] ) /
13             ( nodes[ i ] - nodes[ i-j ] );
14    }
15 }
16
17 double newtonFormula( const double nodes[ n+1 ],
18                       const double differences[ n+1 ][ n+1 ],
19                       double x )
20 {
21    double value = 0.0, product = 1.0;
22    for( int i = 0; i <= n; i++ )
23    {
24       value = value + differences[ i ][ i ] * product;
25       product = product * ( x - nodes[ i ] );
26    }
27 }
```