

VŠB - Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra aplikované matematiky

Nejlepší polynomiální aproximace  
Best polynomial approximation

## Zadání bakalářské práce

Student: **Tomáš Staško**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 1103R031 Výpočetní matematika

Téma: **Nejlepší polynomiální aproximace**  
**Best polynomial approximation**

Jazyk vypracování: čeština

### Zásady pro vypracování:

Mnoho docela jednoduchých úloh např. výpočty určitých integrálů nebo řešení diferenciálních rovnic nelze řešit exaktně, ale pouze přibližně metodami numerické matematiky. Klíčový koncept je nahrazení (aproximace) např. integrandu polynomem. Nejraději používáme interpolaci nebo metodu nejmenších čtverců, neboť tyto metody vedou na soustavu lineárních rovnic. Chyba této aproximace je ale optimální jen v jistém průměru a v některých bodech může být nepříjemně vysoká.

Aproximaci, která minimalizuje maximum chyby v každém bodě intervalu, se říká *nejlepší aproximace*. Sice se ví, že úloha nejlepší polynomiální aproximace má právě jedno řešení, neví se však už, jak jej spočítat přesně. Je ovšem známo pár pěkných výsledků matematiky, která stojí v pozadí, díky nimž se lze přibližnému řešení dobat, např. pomocí Remezova algoritmu. Překvapivě dobré (near-best approximation) řešení dává také interpolace v uzlech Čebyševova polynomu.

Cílem této práce je:

1. Studium problému nejlepší polynomiální aproximace.
2. Studium Čebyševových polynomů.
3. Implementace Remezova algoritmu a porovnání jeho výsledků s Čebyševovou interpolací.

### Seznam doporučené odborné literatury:

- Lloyd N. Trefethen: Approximation Theory and Approximation Practice. SIAM 2013
- materiály ze stránky doc. Petra Tichého (KNM MFF UK), <http://numapprox.blogspot.com>

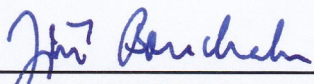
Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

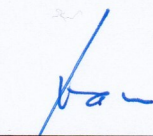
Vedoucí bakalářské práce: **Ing. Marie Sadowská, Ph.D.**

Datum zadání: 01.09.2019

Datum odevzdání: 30.04.2020



  
prof. RNDr. Jiří Bouchala, Ph.D.  
vedoucí katedry

  
prof. Ing. Pavel Brandštetter, CSc.  
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě, 14. 5. 2020

*Šťastko*

Na tomto místě bych rád poděkoval Ing. Marii Sadowské, Ph.D za její cenné rady, věcné připomínky a vstřícnost při konzultacích a hlavně za značnou dávku trpělivosti při zpracování této práce.

## Abstrakt

U některých matematických úloh nedokážeme nalézt řešení exaktně, ale pouze přibližně metodami numerické matematiky. Klíčovým konceptem je pak nahrazení neboli aproximace. Mezi nejčastěji používané metody patří například polynomiální interpolace, nicméně chyba této metody může být v některých bodech intervalu velmi vysoká. Aproximace, která v každém bodě daného intervalu minimalizuje maximum chyby, se nazývá nejlepší stejnoměrná aproximace (best approximation). Obecně existuje právě jedno řešení úlohy nalezení nejlepší polynomiální aproximace, ale nelze ji vypočítat přesně. K přibližnému řešení se lze dopracovat například použitím Remezova algoritmu. Překvapivě dobré (near-best approximation) řešení nám také nabízí interpolace v Čebyševových uzlech. Cílem naší práce je zejména nastudovat a naimplementovat Remezův algoritmus a otestovat jeho funkčnost. Remezův algoritmus nakonec porovnáme s Lagrangeovou interpolací v Čebyševových uzlech.

**Klíčová slova:** Lagrangeova interpolace, Čebyševovy polynomy, Čebyševova aproximace, chyba polynomiální interpolace, nejlepší polynomiální aproximace, polynom minimax, Remezův algoritmus

## Abstract

For some mathematical problems we are not able to find solutions exactly, but only approximately by using methods of numerical mathematics. The key concept is then replacement or in other words approximation. One of the most used methods is, for example, polynomial interpolation, however at some points of the interval the error of this method might be significant. The approximation which minimizes the maximum error at every point of the interval is called the best approximation. Generally, there is a unique solution to the problem of finding the best polynomial approximation, but it is not possible to calculate it exactly. The approximate solution can be obtained, for example, by using the Remez algorithm. The interpolation at Chebyshev nodes also yields surprisingly good results, generally being near-best approximations. The goal of our thesis is namely to study and implement the Remez algorithm and test its functionality. Finally, we compare the Remez algorithm with Lagrange interpolation at Chebyshev nodes.

**Keywords:** Lagrange interpolation, Chebyshev polynomials, Chebyshev approximation, error of polynomial interpolation, best polynomial approximation, minimax polynomial, Remez algorithm

## Seznam použitých zkratek a symbolů

■	konec důkazu
▲	konec příkladu
$\mathbb{N}$	množina všech přirozených čísel
$\mathbb{R}$	množina všech reálných čísel
$\delta_{ij}$	Kroneckerovo delta
$C^{n+1}(\langle a, b \rangle)$	prostor všech reálných funkcí se spojitými derivacemi na $\langle a, b \rangle$ až do řádu $(n + 1)$
$\mathcal{P}_n$	prostor všech reálných polynomů řádu nejvýše $n$
$\ g\ _\infty$	$\max_{x \in \langle a, b \rangle}  g(x) $
SLR	system lineárních rovnic
$\text{Lin} \{g_0, g_1, \dots, g_n\}$	lineární obal množiny funkcí $g_0, g_1, \dots, g_n$
$\text{sgn}(x)$	funkce signum (znaménková funkce)

# Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
<b>2</b>	<b>Úvod do polynomiální interpolace</b>	<b>4</b>
2.1	Vandermondova matice . . . . .	4
2.2	Lagrangeova interpolace . . . . .	5
2.3	Čebyševovy polynomy . . . . .	6
2.3.1	Vlastnosti Čebyševových polynomů . . . . .	7
2.3.2	Aproximace Čebyševovými polynomy . . . . .	10
<b>3</b>	<b>Chyba polynomiální interpolace</b>	<b>14</b>
3.1	Uniformní dělení . . . . .	15
3.2	Dělení Čebyševovými uzly . . . . .	17
3.3	Poznámka ke konvergenci polynomiální interpolace . . . . .	18
<b>4</b>	<b>Aproximace typu minimax</b>	<b>20</b>
4.1	Čebyševovy uzly pro optimální aproximaci . . . . .	24
4.2	Remezův algoritmus . . . . .	26
<b>5</b>	<b>Numerické experimenty</b>	<b>33</b>
5.1	Číslo podmíněnosti Vandermondovy matice . . . . .	33
5.2	Experimenty pro Lagrangeovu a Čebyševovu interpolaci . . . . .	33
5.3	Experimenty pro Remezův algoritmus . . . . .	37
5.3.1	Srovnání Remezova algoritmu s analytickými výsledky . . . . .	38
5.3.2	Chování Remezova algoritmu pro zvyšující se stupeň aproximačního polynomu . . . . .	39
5.3.3	Nejlepší polynomiální aproximace 3. stupně pro vybrané funkce . . . . .	40
5.3.4	Srovnání Remezova algoritmu s Lagrangeovou interpolací v Čebyševových uzlech . . . . .	42
<b>6</b>	<b>Závěr</b>	<b>44</b>
<b>A</b>	<b>Kód Remezova algoritmu v Matlabu</b>	<b>45</b>
	<b>Reference</b>	<b>49</b>

## Seznam obrázků

1	Interpolační polynom 2. stupně z příkladu . . . . .	6
2	Čebyševovy polynomy vybraných stupňů na intervalu $\langle -1, 1 \rangle$ . . . . .	10
3	Polynom 2. stupně z příkladu na aproximaci Čebyševovými polynomy . . . . .	12
4	Největší absolutní chyba Lagrangeovy interpolace v ekvidistantní síti uzlů a odhad maximální absolutní chyby pro funkci $f(x) = \sin(\pi x)$ na $\langle -1, 1 \rangle$ . . . . .	17
5	Největší absolutní chyba Lagrangeovy interpolace v Čebyševových uzlech a odhad chyby pro funkci $f(x) = \sin(\pi x)$ na $\langle -1, 1 \rangle$ . . . . .	18
6	Největší absolutní chyba Lagrangeovy interpolace u funkcí $f, g, h$ . . . . .	19
7	$\sqrt{x}$ na intervalu $\langle 0, 1 \rangle$ . . . . .	20
8	$\max\{c, 1 - c\}$ . . . . .	21
9	Nejlepší konstantní stejnoměrná aproximace $\sqrt{x}$ na $\langle 0, 1 \rangle$ . . . . .	21
10	Funkce $\sqrt{x} - p_*(x)$ na intervalu $\langle 0, 1 \rangle$ . . . . .	22
11	Nejlepší lineární stejnoměrná aproximace $\sqrt{x}$ na $\langle 0, 1 \rangle$ . . . . .	23
12	Funkce $\sqrt{x} - p_*(x)$ na intervalu $\langle 0, 1 \rangle$ . . . . .	24
13	Schéma zobrazující 1. případ a jeho 1. možnost . . . . .	30
14	Schéma zobrazující 1. případ a jeho 2. možnost . . . . .	30
15	Schéma zobrazující 2. případ a jeho 1. možnost . . . . .	31
16	Schéma zobrazující 2. případ a jeho 2. možnost . . . . .	31
17	Schéma zobrazující 3. případ a jeho 1. možnost . . . . .	31
18	Schéma zobrazující 3. případ a jeho 2. možnost . . . . .	32
19	Růst čísla podmíněnosti $\kappa(V)$ v závislosti na stupni $n$ aproximačního poly- nomu v logaritmickém měřítku . . . . .	33
20	Lagrangeova interpolace na uniformním dělení intervalu funkce $f$ . . . . .	34
21	Lagrangeova interpolace na uniformním dělení intervalu funkce $g$ . . . . .	34
22	Lagrangeova interpolace na Čebyševových uzlech funkce $f$ . . . . .	34
23	Lagrangeova interpolace na Čebyševových uzlech funkce $g$ . . . . .	35
24	Čebyševova aproximace funkce $f$ . . . . .	35
25	Čebyševova aproximace funkce $g$ . . . . .	35
26	Chyba pro Lagrangeovu interpolaci na uniformním dělení intervalu . . . . .	36
27	Chyba pro Lagrangeovu interpolaci v Čebyševových uzlech . . . . .	37
28	Chyba pro Čebyševovu aproximaci . . . . .	37
29	Funkce reziduí na intervalu $\langle 0, 1 \rangle$ . . . . .	39
30	Nejlepší polynomiální aproximace na funkci $f(x) = e^x \cos(4\pi x) \sin(\pi x)$ na intervalu $\langle 0, 1 \rangle$ . . . . .	40
31	Nejlepší polynomiální aproximace 3. stupně pro různě zadané funkce . . . . .	42
32	Srovnání chyby pro Lagrangeovu interpolaci v Čebyševových uzlech s nejlepší polynomiální aproximací . . . . .	43



# 1 Úvod

Úlohy zabývající se nalezením řešení diferenciálních rovnic nebo výpočtem určitého integrálu jsou v matematice velmi důležité. V praxi se však může stát, že tyto úlohy nemůžeme vyřešit přesně, ale pouze přibližně použitím numerických metod. Polynomiální aproximace je jednou z mnoha metod numerické matematiky, která slouží k nahrazení například integrandu polynomem. Jednou z možností nalezení polynomiální aproximace je úloha polynomiální interpolace. Metody, které řeší přímo tuto úlohu, obecně neminimalizují chybu v každém bodě intervalu, a proto se může stát, že v některých bodech bude chyba velmi vysoká. Aproximace, která v každém bodě intervalu minimalizuje maximum chyby, se nazývá tzv. nejlepší aproximace. Je známo, že nejlepší polynomiální aproximace existuje jednoznačně, ale její nalezení nelze vyřešit exaktně. V této práci si ukážeme, že k přibližnému řešení se můžeme dopracovat například pomocí tzv. Remezova algoritmu. Dobré řešení (near-best approximation) nabízí také interpolace v Čebyševových uzlech.

V rámci 2. kapitoly si stručně vysvětlíme pojem polynomiální interpolace a ukážeme si její tři metody: naivní přístup pomocí Vandermondovy matice, Lagrangeovu interpolaci a aproximaci Čebyševovými polynomy. Ve 3. kapitole se zaměříme na kvalitu polynomiální interpolace. Vysvětlíme si, co je to tzv. chyba polynomiální interpolace, a odvodíme z ní odhad pro největší absolutní chybu v intervalu. Provedeme také srovnání Lagrangeovy interpolace v ekvidistatní síti uzlů s Lagrangeovou interpolací v Čebyševových uzlech. V poslední části této kapitoly se vyjádříme ke konvergenci polynomiální interpolace a shrneme výsledky, ke kterým jsme se dopracovali. Ve 4. kapitole si ukážeme alternativní přístup, který přímo optimalizuje kvalitu aproximace. Nejdříve si definujeme úlohu nalezení tzv. aproximace typu minimax. Tuto úlohu pak přibližně řeší Remezův algoritmus, který si podrobně vysvětlíme a naimplementujeme v programu Matlab (kód je k nalezení v apendixu této práce). V poslední kapitole jsou prezentovány výsledky provedených numerických experimentů. V celé práci jsme se snažili zlepšit čitelnost podtržením nových pojmů.

## 2 Úvod do polynomiální interpolace

Uvažujme množinu dat

$$\{x_i, y_i\}, \quad i = 0, 1, \dots, n.$$

Tato data můžeme získat např. tak, že v čase  $x_i$  naměříme hodnotu  $y_i$  nějaké veličiny. Hledáme takový polynom  $p_n \in \mathcal{P}_n$ , aby platily tzv. interpolační podmínky:

$$p_n(x_i) = y_i, \quad i = 0, 1, \dots, n. \quad (1)$$

### 2.1 Vandermondova matice

Obecně můžeme hledaný polynom zapsat jako:

$$p_n(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n.$$

Interpolační podmínky (1) tedy mají tvar  $(n + 1)$  rovnic, které můžeme zapsat následovně:

$$\begin{aligned} p_n(x_0) = y_0 : & \quad a_0 + a_1x_0 + a_2x_0^2 + \dots + a_nx_0^n = y_0, \\ p_n(x_1) = y_1 : & \quad a_0 + a_1x_1 + a_2x_1^2 + \dots + a_nx_1^n = y_1, \\ & \quad \vdots \\ p_n(x_n) = y_n : & \quad a_0 + a_1x_n + a_2x_n^2 + \dots + a_nx_n^n = y_n. \end{aligned} \quad (2)$$

Neznámými hodnotami jsou pro nás koeficienty  $a_0, a_1, \dots, a_n$ . Definujme si vektor  $a$ , který bude obsahovat všechny dříve vypsané proměnné, tj.  $a = [a_0, a_1, \dots, a_n]^T$ , a vektor  $y$ , který obsahuje známé hodnoty  $y_0, y_1, \dots, y_n$ , tj.  $y = [y_0, y_1, \dots, y_n]^T$ . Řešíme tedy soustavu  $(n + 1)$  lineárních rovnic o  $(n + 1)$  neznámých. Tuto soustavu je možné zapsat pomocí tzv. Vandermondovy matice

$$V = \begin{bmatrix} 1 & x_0 & x_0^2 & \dots & x_0^n \\ 1 & x_1 & x_1^2 & \dots & x_1^n \\ 1 & x_2 & x_2^2 & \dots & x_2^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^n \end{bmatrix}.$$

Je zřejmé, že se jedná o  $(n + 1) \times (n + 1)$  matici. Prvky  $i$ -tého řádku jsou členy geometrické posloupnosti  $b_j = x_i^j$ ,  $j = 0, 1, \dots, n$ . Prvky matice lze zapsat:

$$[V]_{i,j} = x_i^{j-1}, \quad i, j = 1, 2, \dots, n + 1.$$

Naše soustava (2) je tedy ve tvaru

$$Va = y.$$

#### Výhody a nevýhody přístupu s Vandermondovou maticí

- (+) výpočetně a konstrukčně nenáročný,
- (-) číslo podmíněnosti matice  $V$  má vysokou hodnotu (viz obr. 19 na str. 33), proto se tento přístup nepoužívá, slouží pouze jako první (naivní nebo seznamovací) přístup při studiu polynomiální interpolace. Platí, že malá chyba zahrnutá v hodnotách vektoru  $y$  může vyvolat obrovskou chybu v řešení soustavy.

## 2.2 Lagrangeova interpolace

Ukážeme si nyní další metodu polynomiální interpolace. Definujeme kardinální funkce  $\ell_0, \ell_1, \dots, \ell_n$ , pro které platí:

$$\ell_i(x_j) = \delta_{ij} = \begin{cases} 1 & \text{pro } i = j, \\ 0 & \text{pro } i \neq j, \end{cases} \quad i, j = 0, 1, \dots, n, \quad (3)$$

pomocí:

$$\ell_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^n \frac{x - x_j}{x_i - x_j} = \frac{(x - x_0)(x - x_1) \cdots (x - x_{i-1})(x - x_{i+1}) \cdots (x - x_n)}{(x_i - x_0)(x_i - x_1) \cdots (x_i - x_{i-1})(x_i - x_{i+1}) \cdots (x_i - x_n)}. \quad (4)$$

Je zřejmé, že  $\ell_i(x_i) = 1$  a  $\ell_i(x_k) = 0$  pro  $k \neq i$ . Lagrangeův interpolační polynom je pak definován pomocí

$$p_n(x) = \sum_{i=0}^n \ell_i(x) y_i. \quad (5)$$

Díky (3) je zřejmé, že pro Lagrangeův interpolační polynom jsou splněny interpolační podmínky (1).

### Výhody a nevýhody Lagrangeovy interpolace

- (+) matematicky elegantní způsob,
- (+) oproti metodě, která využívá Vandermondovu matici, můžeme počítat s vyšším počtem dat,
- (-) konstrukce kardinálních funkcí je výpočetně náročná,
- (-) přidáním dalšího bodu (měření, pozorování) musíme vypočítat všechny kardinální funkce znovu.

**Příklad 2.1.** Sestavme Lagrangeův interpolační polynom 2. stupně pro následující zadaná data:

$$\begin{aligned} x_0 &= -1, & y_0 &= 0, \\ x_1 &= 0, & y_1 &= 0, \\ x_2 &= 1, & y_2 &= 0. \end{aligned}$$

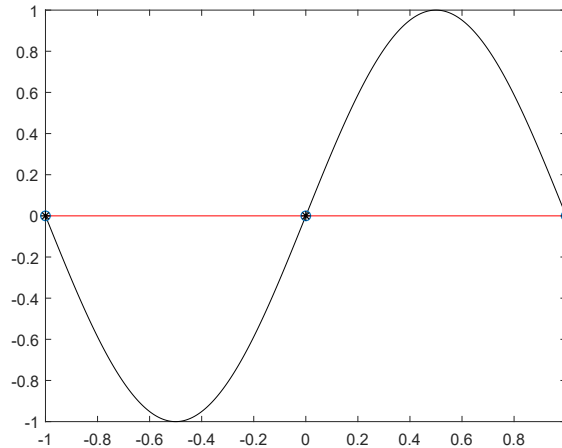
**Řešení.** Vyjdeme ze vztahu (4) a vypočítáme kardinální funkce  $\ell_i$ ,  $i = 0, 1, 2$ :

$$\begin{aligned} \ell_0(x) &= \frac{(x - 0)(x - 1)}{(-1 - 0)(-1 - 1)} = \frac{1}{2} (x^2 - x), \\ \ell_1(x) &= \frac{(x + 1)(x - 1)}{(0 + 1)(0 - 1)} = -(x^2 - 1), \\ \ell_2(x) &= \frac{(x + 1)(x - 0)}{(1 + 1)(1 - 0)} = \frac{1}{2} (x^2 + x). \end{aligned}$$

Pokračujeme dále ve výpočtu podle vztahu (5):

$$p_2(x) = \frac{1}{2} (x^2 - x) \cdot 0 - (x^2 - 1) \cdot 0 + \frac{1}{2} (x^2 + x) \cdot 0 = 0.$$

V tomto příkladě byly hodnoty  $y_i$  zvoleny jako  $\sin(\pi x_i)$ . Jedná se o umělý příklad, ve kterém ilustrujeme především vytváření kardinálních funkcí. I bez výpočtu kardinálních funkcí je jasné, že interpolační polynom bude nulový, protože funkční hodnoty funkce  $\sin(\pi x_i)$  vychází v našem příkladu ve všech uzlech 0. V následujícím grafu je interpolační polynom znázorněn červeně a funkce  $f(x) = \sin(\pi x_i)$  černě v intervalu  $\langle -1, 1 \rangle$ . Hvězdičky představují uzly zvolené pro interpolaci, modrá kolečka znázorňují průnik funkce  $f$  a interpolačního polynomu.



Obrázek 1: Interpolační polynom 2. stupně z příkladu

▲

Několik numerických experimentů pro Lagrangeovu interpolaci nalezneme v příkladu 3.2 na str. 16, v příkladu 3.3 na str. 18 a také v kapitole 5.2 na str. 33.

### 2.3 Čebyševovy polynomy

V této kapitole si ukážeme další možný přístup k polynomiální interpolaci, a to pomocí Čebyševových polynomů. Definujeme Čebyševovy polynomy  $T_n$ , pro které platí:

$$T_n(\cos \theta) = \cos(n\theta), \quad n \in \mathbb{N} \cup \{0\}. \quad (6)$$

Je zřejmé, že

$$\begin{aligned} T_0(\cos \theta) &= 1, \\ T_1(\cos \theta) &= \cos \theta. \end{aligned}$$

S využitím známých vzorců z trigonometrie nalezneme další Čebyševovy polynomy:

$$T_2(\cos \theta) = \cos(2\theta) = 2 \cos^2 \theta - 1.$$

Postup pro nalezení polynomu 3. řádu:

$$\begin{aligned} \cos(3\theta) &= \cos(2\theta) \cos \theta - \sin(2\theta) \sin \theta = (2 \cos^2 \theta - 1) \cos \theta - 2 \sin \theta \cos \theta \sin \theta = \\ &= 2 \cos^3 \theta - \cos \theta - 2 \cos \theta + 2 \cos^3 \theta = 4 \cos^3 \theta - 3 \cos \theta, \end{aligned}$$

tedy

$$T_3(\cos \theta) = 4 \cos^3 \theta - 3 \cos \theta.$$

A konečně:

$$\cos(4\theta) = \cos(2 \cdot 2\theta) = 2(2\cos^2\theta - 1)^2 - 1 = 8\cos^4\theta - 8\cos^2\theta + 1,$$

tedy

$$T_4(\cos\theta) = 8\cos^4\theta - 8\cos^2\theta + 1.$$

Nyní nalezneme zápis pro Čebyševův polynom  $(n+1)$ -ního stupně:

$$\begin{aligned} T_{n+1}(\cos\theta) &= \cos((n+1)\theta) = \cos(n\theta + \theta) = \cos(n\theta)\cos\theta - \sin(n\theta)\sin\theta = \\ &= 2\cos(n\theta)\cos\theta - \cos(n\theta)\cos\theta - \sin(n\theta)\sin\theta = \\ &= 2\cos(n\theta)\cos\theta - \cos((n-1)\theta) = 2T_n(\cos\theta)\cos\theta - T_{n-1}(\cos\theta). \end{aligned}$$

Odtud dostáváme rekurentní vzorec:

$$T_{n+1}(\cos\theta) = 2T_n(\cos\theta)\cos\theta - T_{n-1}(\cos\theta), \quad n \in \mathbb{N}.$$

Nechť  $n \geq 1$ , pak výraz  $\cos(n\theta)$  může být zapsán jako polynom tvořený mocninami  $\cos\theta$ . Nyní dosadíme  $x$  ( $x \in \langle -1, 1 \rangle$ ) za  $\cos\theta$  do předešlých vztahů a dostaneme

$$\begin{aligned} T_0(x) &= 1, \\ T_1(x) &= x, \\ T_2(x) &= 2x^2 - 1, \\ T_3(x) &= 4x^3 - 3x, \\ T_4(x) &= 8x^4 - 8x^2 + 1, \\ &\vdots \end{aligned}$$

Čebyševův polynom  $(n+1)$ -ního stupně je dán rekurentním vztahem:

$$T_{n+1}(x) = 2T_n(x)x - T_{n-1}(x), \quad n \in \mathbb{N}. \quad (7)$$

Pro  $x \in \langle -1, 1 \rangle$  a  $\theta \in \langle 0, \pi \rangle$  platí:

$$\cos(\theta) = x \Leftrightarrow \theta = \arccos(x).$$

Ze vztahu (6) pak vyplývá, že pro  $x \in \langle -1, 1 \rangle$  můžeme Čebyševovy polynomy zapisovat ve tvaru:

$$T_n(x) = \cos(n \arccos(x)). \quad (8)$$

### 2.3.1 Vlastnosti Čebyševových polynomů

Z rekurentního vztahu (7) vyplývá, že stupeň Čebyševova polynomu  $T_n$  je roven  $n$ . Navíc víme, že Čebyševovy polynomy jsou vždy ve tvaru:

$$T_n(x) = 2^{n-1}x^n + \text{polynom s mocninami } x \text{ nižšími než } n.$$

Vedoucí koeficient polynomu  $T_n$  je tedy vždy  $2^{n-1}$ .

**Tvrzení 2.1.** Čebyševův polynom  $T_n$  má právě  $n$  kořenů v intervalu  $\langle -1, 1 \rangle$ . Leží v těchto bodech:

$$\xi_k = \cos\left(\frac{2k-1}{2n}\pi\right), \quad k = 1, 2, \dots, n. \quad (9)$$

**Důkaz.** Vyjděme ze vztahu (8), pak:

$$T_n(\xi_k) = \cos\left(n \frac{2k-1}{2n} \pi\right) = \cos\left((2k-1) \frac{\pi}{2}\right).$$

Můžeme si povšimnout, že po úpravách bude v argumentu vždy lichý násobek  $\frac{\pi}{2}$ . To způsobí, že bude hodnota cosinu vždy 0. ■

Čebyševovy polynomy lze tedy zapsat pomocí součinu

$$T_n(x) = 2^{n-1} \prod_{k=1}^n (x - \xi_k).$$

**Tvrzení 2.2.** Polynom  $T_n$  má v intervalu  $\langle -1, 1 \rangle$  právě  $(n+1)$  extrémů. Nachází se v následujících bodech:

$$\eta_k = \cos\left(\frac{k\pi}{n}\right), \quad k = 0, 1, \dots, n.$$

Dále platí, že

$$T_n(\eta_k) = (-1)^k.$$

**Důkaz.** Vyjděme opět ze vztahu (8),  $x \in \langle -1, 1 \rangle$ , pak:

$$T'_n(x) = -\sin(n \arccos(x)) n \left(-\frac{1}{\sqrt{1-x^2}}\right).$$

Hledáme stacionární body polynomu  $T_n$  v  $(-1, 1)$ , tj. řešíme rovnici

$$\begin{aligned} \sin(n \arccos(x)) &= 0 \\ \Updownarrow \\ n \arccos(x) &= k\pi, \quad k = 1, 2, \dots, n-1. \end{aligned}$$

Stacionární body označme

$$x_k^s = \cos\left(\frac{k\pi}{n}\right), \quad k = 1, 2, \dots, n-1.$$

Potom

$$T_n(x_k^s) = \cos\left(n \arccos\left(\cos\left(\frac{k\pi}{n}\right)\right)\right) = \cos(k\pi), \quad k = 1, 2, \dots, n-1.$$

Je tedy zřejmé, že ve stacionárních bodech jsme našli  $(n-1)$  extrémů, které budou nabývat hodnot  $-1$ , když bude  $k$  liché, nebo hodnoty  $1$ , když bude  $k$  sudé. Nyní zbývá zkontrolovat krajní body intervalu  $\langle -1, 1 \rangle$ . Podle (8) platí:  $T_n(1) = \cos(0) = 1$ ,  $T_n(-1) = \cos(n\pi) = -1$ , když  $n$  nabývá liché hodnoty, nebo  $1$ , když  $n$  nabývá sudé hodnoty. ■

Lze taky ukázat, že Čebyševovy polynomy jsou ortogonální polynomy s váhou  $\frac{1}{\sqrt{1-x^2}}$  na intervalu  $\langle -1, 1 \rangle$ :

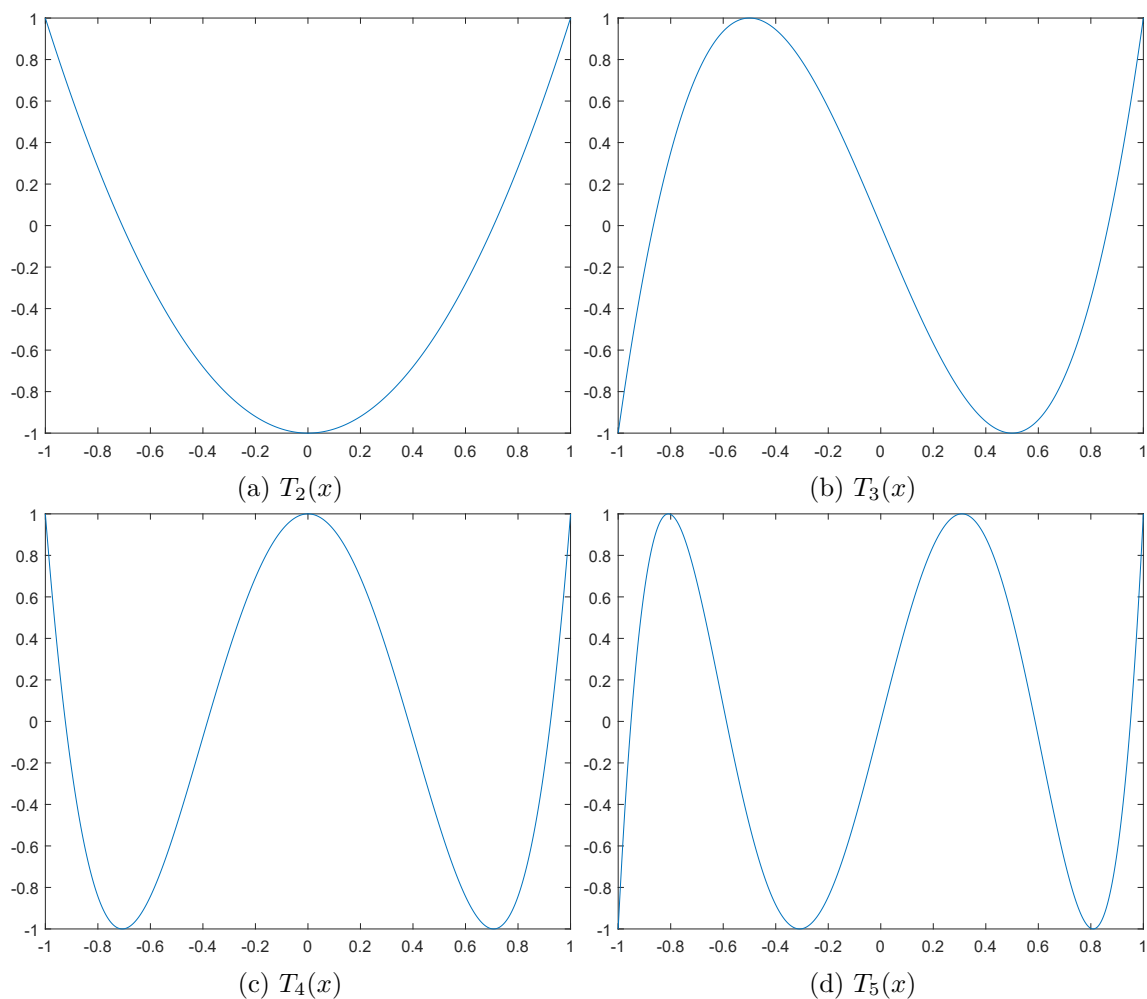
$$\int_{-1}^1 \frac{T_i(x)T_j(x)}{\sqrt{1-x^2}} dx = \begin{cases} 0 & \text{pro } i \neq j, \\ \frac{\pi}{2} & \text{pro } i = j \neq 0, \\ \pi & \text{pro } i = j = 0. \end{cases}$$

Diskrétní ortogonalita je pak ve tvaru:

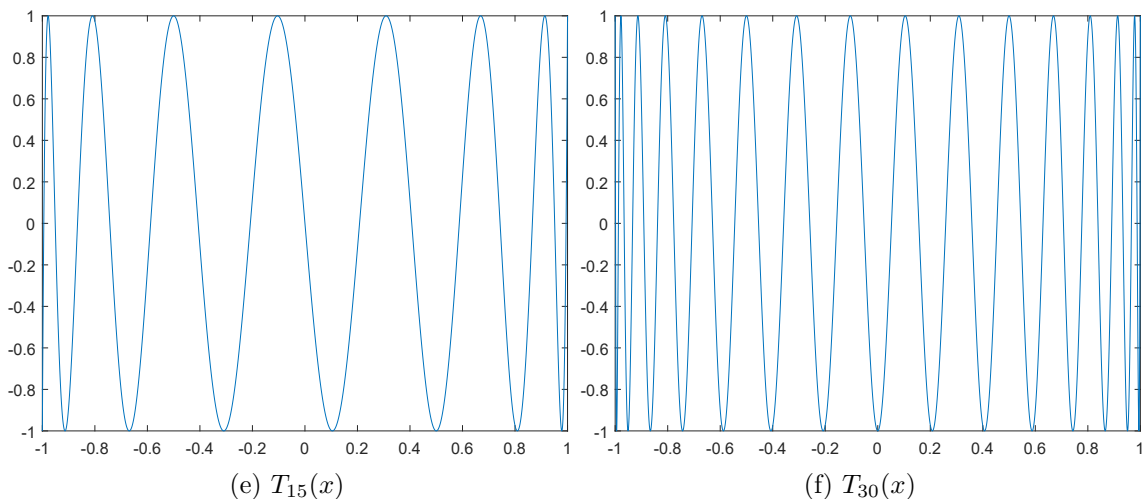
$$\sum_{k=1}^n T_i(x_k)T_j(x_k) = \begin{cases} 0 & \text{pro } i \neq j, \\ \frac{n}{2} & \text{pro } i = j \neq 0, \\ n & \text{pro } i = j = 0, \end{cases} \quad (10)$$

kde  $x_k$  ( $i, j < n$  a  $k = 1, \dots, n$ ) jsou kořeny Čebyševova polynomu  $T_n$ .

Čebyševovy polynomy vybraných stupňů na intervalu  $\langle -1, 1 \rangle$  jsou pak znázorněny na obr. 2.



Obrázek 2: Čebyševovy polynomy vybraných stupňů na intervalu  $\langle -1, 1 \rangle$



Obrázek 2: Čebyševovy polynomy vybraných stupňů na intervalu  $\langle -1, 1 \rangle$

### 2.3.2 Aproximace Čebyševovými polynomy

Při aproximaci funkce  $f$  na intervalu  $\langle -1, 1 \rangle$  pomocí Čebyševových polynomů hledáme polynom  $p_{n-1} \in \mathcal{P}_{n-1}$  ve tvaru

$$p_{n-1}(x) = \sum_{i=0}^{n-1} a_i T_i(x), \quad x \in \langle -1, 1 \rangle. \quad (11)$$

Interpoláční podmínky mají tvar  $n$  rovnic, které zapíšeme následujícím způsobem:

$$f(\xi_k) = \sum_{i=0}^{n-1} a_i T_i(\xi_k), \quad k = 1, 2, \dots, n, \quad (12)$$

kde  $\xi_k$  jsou kořeny polynomu  $T_n$ , které jsme si již odvodili v (9). Koeficienty  $a_0, a_1, \dots, a_{n-1}$  jsou pro nás neznámými hodnotami. Vztah (12) představuje  $n$  rovnic. Každou z těchto rovnic přenásobíme  $T_j(\xi_k)$  a následně všechny výsledné rovnice sečteme. Řešíme tedy soustavu  $n$  rovnic o  $n$  neznámých:

$$\sum_{k=1}^n f(\xi_k) T_j(\xi_k) = \sum_{k=1}^n \sum_{i=0}^{n-1} a_i T_i(\xi_k) T_j(\xi_k), \quad j = 0, 1, \dots, n-1. \quad (13)$$

Podívejme se nejdříve na levou stranu rovnic (13). Když proměnná  $j$  nabývá hodnoty 0, pak platí:

$$\sum_{k=1}^n f(\xi_k) T_0(\xi_k) = \sum_{k=1}^n f(\xi_k).$$

Když je  $j \neq 0$ , využijeme vztahu (8):

$$\sum_{k=1}^n f(\xi_k) T_j(\xi_k) = \sum_{k=1}^n f(\xi_k) \cos(j \arccos(\xi_k)).$$

Nyní se přesuneme na pravou stranu soustavy rovnic (13). Provedeme jednoduchou úpravu:

$$\sum_{k=1}^n \sum_{i=0}^{n-1} a_i T_i(\xi_k) T_j(\xi_k) = \sum_{i=0}^{n-1} a_i \sum_{k=1}^n T_i(\xi_k) T_j(\xi_k).$$



U druhé sumy využijeme diskrétní ortogonalitu Čebyševových polynomů (10). Je tedy zřejmé, že pravá strana soustavy rovnic (13) má pouze dva tvary:

$$\begin{aligned} a_0 n & \text{ pro } j = 0, \\ a_j \frac{n}{2} & \text{ pro } j \neq 0. \end{aligned}$$

Vrátíme se opět k soustavě (13) a vyjádříme si proměnné  $a_i$ ,  $i = 0, 1, \dots, n-1$ . Pro  $j = 0$ :

$$\begin{aligned} \sum_{k=1}^n f(\xi_k) &= a_0 n \\ \Downarrow \\ a_0 &= \frac{1}{n} \sum_{k=1}^n f(\xi_k) \\ \Downarrow \text{viz (9)} \\ a_0 &= \frac{1}{n} \sum_{k=1}^n f\left(\cos\left(\frac{2k-1}{2n}\pi\right)\right), \end{aligned} \tag{14}$$

pro  $j \neq 0$ :

$$\begin{aligned} \sum_{k=1}^n f(\xi_k) \cos(j \arccos(\xi_k)) &= a_j \frac{n}{2} \\ \Downarrow \\ a_j &= \frac{2}{n} \sum_{k=1}^n f(\xi_k) \cos(j \arccos(\xi_k)) \\ \Downarrow \text{viz (9)} \\ a_j &= \frac{2}{n} \sum_{k=1}^n f\left(\cos\left(\frac{2k-1}{2n}\pi\right)\right) \cos\left(j \frac{2k-1}{2n}\pi\right), \quad j = 1, 2, \dots, n-1. \end{aligned} \tag{15}$$

Vzorce (14) a (15) nám tedy poskytují návod, jak řešit soustavu (13) a tím tedy i získat aproximaci (11).

**Příklad 2.2.** Provedme aproximaci Čebyševovými polynomy na funkci  $f(x) = \sin(\pi x)$  na intervalu  $\langle -1, 1 \rangle$  pro 3 Čebyševovy uzly.

**Řešení.** Máme 3 uzly  $\xi_k$ ,  $k = 1, 2, 3$ . Využijeme vztah (9) k tomu, abychom vypočítali pozici uzlů  $\xi_k$ :

$$\begin{aligned} \xi_1 &= \cos\left(\frac{1}{6}\pi\right) = \frac{\sqrt{3}}{2}, \\ \xi_2 &= \cos\left(\frac{3}{6}\pi\right) = 0, \\ \xi_3 &= \cos\left(\frac{5}{6}\pi\right) = -\frac{\sqrt{3}}{2}. \end{aligned}$$

Pomocí vztahů (14) a (15) nalezneme hodnoty  $a_0, a_1$  a  $a_2$  :

$$a_0 = \frac{1}{3} \sum_{k=1}^3 f(\xi_k) = \frac{1}{3} \left( \sin \left( \frac{\sqrt{3}}{2} \pi \right) + \sin(0 \cdot \pi) + \sin \left( -\frac{\sqrt{3}}{2} \pi \right) \right) = 0,$$

$$a_1 = \frac{2}{3} \sum_{k=1}^3 f(\xi_k) \cdot \xi_k = \frac{2}{3} \left( \sin \left( \frac{\sqrt{3}}{2} \pi \right) \cdot \frac{\sqrt{3}}{2} + \sin(0 \cdot \pi) \cdot 0 + \right. \\ \left. + \sin \left( -\frac{\sqrt{3}}{2} \pi \right) \cdot \left( -\frac{\sqrt{3}}{2} \right) \right) = \frac{2}{3} \sqrt{3} \sin \left( \frac{\sqrt{3}}{2} \pi \right),$$

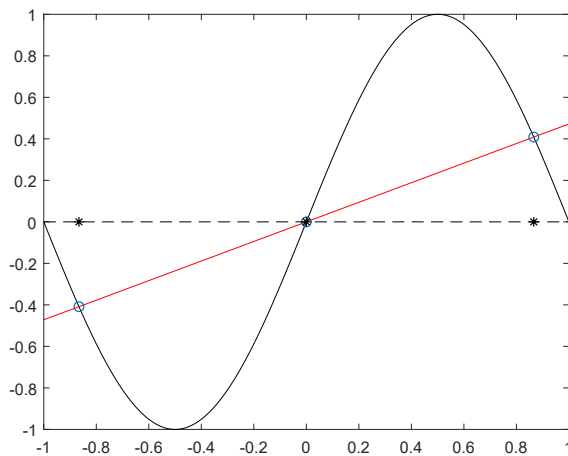
$$a_2 = \frac{2}{3} \sum_{k=1}^3 f(\xi_k) \cos \left( \frac{2k-1}{3} \pi \right) = \frac{2}{3} \left( \sin \left( \frac{\sqrt{3}}{2} \pi \right) \cdot \frac{1}{2} + \sin(0 \cdot \pi) \cdot (-1) + \right. \\ \left. + \sin \left( -\frac{\sqrt{3}}{2} \pi \right) \cdot \frac{1}{2} \right) = 0.$$

Díky vztahu (11) je už pak jednoduché vypočítat výsledný tvar aproximace:

$$p_2(x) = \sum_{i=0}^2 a_i T_i(x) = 0 \cdot 1 + \frac{2}{3} \sqrt{3} \sin \left( \frac{\sqrt{3}}{2} \pi \right) \cdot x + 0 \cdot (2x^2 - 1) = \\ = \frac{2\sqrt{3}}{3} \sin \left( \frac{\sqrt{3}}{2} \pi \right) x.$$

Srovnáme ještě tento příklad s příkladem 2.1. V tomto případě je aproximace Čebyševovými polynomy výhodnější než Lagrangeova aproximace na ekvidistantní síti 3 uzlů, protože největší absolutní chyba v intervalu  $\langle -1, 1 \rangle$  (více o této chybě v kapitole 3) je v příkladě 2.1 vyšší než v příkladě 2.2. Konkrétní hodnoty absolutních chyb pro různé stupně interpolačních polynomů jsou pak popsány v tabulkách v příkladech 3.2 a 3.3. Ukážeme si, že podstatnou roli hraje výběr interpolačních uzlů (nikoliv výběr polynomů).

Následující graf znázorňuje aproximaci Čebyševovými polynomy pro funkci  $f(x) = \sin(\pi x)$ . V intervalu  $\langle -1, 1 \rangle$  je aproximace Čebyševovými polynomy znázorněna červeně, funkce  $f(x) = \sin(\pi x)$  pak černě. Hvězdičky představují interpolační uzly, modrá kolečka znázorňují průsečíky funkce  $f$  s aproximací.



Obrázek 3: Polynom 2. stupně z příkladu na aproximaci Čebyševovými polynomy

▲

Numerické experimenty pro Čebyševovu aproximaci dále uvádíme v kapitole 5.2 na str. 33.

### 3 Chyba polynomiální interpolace

**Věta 3.1.** *Uvažujme množinu dat*

$$\{x_i, y_i\}, \quad i = 0, 1, \dots, n,$$

*kde všechna  $x_i$  jsou různá. Potom existuje právě jeden polynom  $p_n \in \mathcal{P}_n$ , pro který platí*

$$p_n(x_i) = y_i \quad i = 0, 1, \dots, n.$$

**Důkaz.** *Důkaz nalezneme např. v [1].* ■

**Příklad 3.1.** *Uvažujme Čebyševovy uzly z příkladu 2.2 a interpolujme funkci  $f(x) = \sin(\pi x)$  v těchto bodech. Použijme Lagrangeovu interpolaci.*

**Řešení.** *Vypočteme funkční hodnoty funkce  $f(x)$  v uzlech z příkladu 2.2.*

$$\begin{aligned} \xi_0 &= \frac{\sqrt{3}}{2}, & f(\xi_0) &= \sin\left(\frac{\sqrt{3}}{2}\pi\right), \\ \xi_1 &= 0, & f(\xi_1) &= \sin(0\pi) = 0, \\ \xi_2 &= -\frac{\sqrt{3}}{2}, & f(\xi_2) &= \sin\left(-\frac{\sqrt{3}}{2}\pi\right). \end{aligned}$$

*Vydeme ze vztahu (4) a vypočteme kardinální funkce  $\ell_0$  a  $\ell_2$  ( $\ell_1$  není třeba počítat, jelikož  $f(\xi_1) = 0$ ):*

$$\begin{aligned} \ell_0(x) &= \frac{(x-0)\left(x-\frac{\sqrt{3}}{2}\right)}{\left(\frac{\sqrt{3}}{2}-0\right)\left(\frac{\sqrt{3}}{2}-\frac{\sqrt{3}}{2}\right)} = \frac{2}{3}\left(x^2 + \frac{\sqrt{3}}{2}x\right), \\ \ell_2(x) &= \frac{\left(x-\frac{\sqrt{3}}{2}\right)(x-0)}{\left(-\frac{\sqrt{3}}{2}-\frac{\sqrt{3}}{2}\right)\left(-\frac{\sqrt{3}}{2}-0\right)} = \frac{2}{3}\left(x^2 - \frac{\sqrt{3}}{2}x\right). \end{aligned}$$

*Pokračujme dále ve výpočtu podle vztahu (5):*

$$\begin{aligned} p_2(x) &= \frac{2}{3}\left(x^2 + \frac{\sqrt{3}}{2}x\right) \cdot \sin\left(\frac{\sqrt{3}}{2}\pi\right) + \frac{2}{3}\left(x^2 - \frac{\sqrt{3}}{2}x\right) \cdot \sin\left(-\frac{\sqrt{3}}{2}\pi\right) = \\ &= \frac{2\sqrt{3}}{3} \sin\left(\frac{\sqrt{3}}{2}\pi\right) x. \end{aligned}$$

*Vidíme tedy, že interpolační polynom vyšel stejně jako v příkladu 2.2, kde jsme interpolovali pomocí Čebyševových polynomů.* ▲

Uvažujme funkci  $f \in C^{n+1}(\langle a, b \rangle)$  a množinu bodů

$$\{x_i\}, \quad x_i \in \langle a, b \rangle, \quad i = 0, 1, \dots, n.$$

Dále mějme polynom  $p_n \in \mathcal{P}_n$ , který splňuje interpolační podmínky

$$f(x_i) = p_n(x_i), \quad i = 0, 1, \dots, n.$$

Definujme funkci  $e$ , která určuje chybu polynomiální interpolace na  $\langle a, b \rangle$  a je dána předpisem

$$e(x) = f(x) - p_n(x), \quad x \in \langle a, b \rangle.$$

**Věta 3.2.** *Bud'  $f \in C^{n+1}(\langle a, b \rangle)$  a polynom  $p_n \in \mathcal{P}_n$  nechť interpoluje funkci  $f$  pro různé body  $x_i \in \langle a, b \rangle$ ,  $i = 0, 1, \dots, n$ . Potom platí:*

$$(\forall x \in \langle a, b \rangle)(\exists \xi \in \langle a, b \rangle) : f(x) - p_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{i=0}^n (x - x_i). \quad (16)$$

**Důkaz.** *Důkaz nalezneme například v [2] na str. 17 nebo v [3] na str. 184. ■*

Z věty 3.2 využijeme vzorec (16), abychom odvodili odhad pro největší absolutní chybu v intervalu  $\langle a, b \rangle$ :

$$\max_{x \in \langle a, b \rangle} |f(x) - p_n(x)| \leq \left( \max_{\xi \in \langle a, b \rangle} \frac{|f^{(n+1)}(\xi)|}{(n+1)!} \right) \left( \max_{x \in \langle a, b \rangle} \prod_{i=0}^n |x - x_i| \right),$$

tj.

$$\|e\|_\infty \leq \frac{\|f^{(n+1)}\|_\infty}{(n+1)!} \max_{x \in \langle a, b \rangle} \prod_{i=0}^n |x - x_i|. \quad (17)$$

Nyní se nabízí otázka, jak odhad (17) udělat co nejmenší. Je jasné, že můžeme zmenšit pouze část s maximem, tj.  $\max_{x \in \langle a, b \rangle} \prod_{i=0}^n |x - x_i|$ , a to vhodnou volbou uzlů. Pojdme si nyní v následujících kapitolách ukázat, jak to bude vypadat s odhadem pro interpolaci v ekvidistantní síti uzlů a pro interpolaci na Čebyševových uzlech.

### 3.1 Uniformní dělení

Mějme interval  $I = \langle a, b \rangle$ ,  $-\infty < a < b < \infty$ . Nyní rovnoměrně rozložíme interval  $I$  na  $n$  stejných podintervalů  $I_i$ ,  $i = 1, 2, \dots, n$ . Všechny podintervaly mají dva hraniční body a je zřejmé, že interval  $I$  obsahuje  $(n+1)$  těchto bodů. Nazveme je uzly intervalu  $I$  a označíme je  $x_i$ ,  $i = 0, 1, \dots, n$ .

Nechť  $h = \frac{b-a}{n}$ , pak  $h$  nazveme délkou mezi dvěma vedlejšími uzly intervalu  $I$ . Z toho vyplývá, že:

$$x_i = a + ih, \quad i = 0, 1, \dots, n.$$

**Tvrzení 3.1.** *Pro interpolaci v ekvidistantní síti uzlů platí, že*

$$\prod_{i=0}^n |x - x_i| \leq \frac{1}{4} h^{(n+1)} n!.$$

**Důkaz.** *Ze vztahu (16) je zřejmé, že chyba bude největší, pokud bude  $x$  ležet na krajích intervalu  $\langle a, b \rangle$ . Bez újmy na obecnosti můžeme předpokládat, že  $x \in (x_0, x_1)$ . Potom*

$$|(x - x_0)(x - x_1)| \leq \frac{(x_1 - x_0)^2}{4},$$

a tedy

$$|(x - x_0)(x - x_1)| \leq \frac{h^2}{4}.$$

Dále víme, že

$$\begin{aligned} |x - x_2| &\leq 2h, \\ |x - x_3| &\leq 3h, \\ |x - x_4| &\leq 4h, \\ &\vdots \end{aligned}$$

Všeobecně tedy můžeme říct, že pro  $i = 2, 3, \dots, n$  platí

$$|x - x_i| \leq ih.$$

Proto

$$\prod_{i=0}^n |x - x_i| \leq \frac{h^2}{4} (2h)(3h) \cdots (nh),$$

a tudíž

$$\prod_{i=0}^n |x - x_i| \leq \frac{1}{4} h^{n+1} n!.$$

■

Odhad maximální absolutní chyby u uniformního rozdělení intervalu je tedy:

$$\begin{aligned} \|e\|_\infty &\leq \frac{\|f^{(n+1)}\|_\infty}{(n+1)!} \frac{1}{4} h^{n+1} n! \\ &\Downarrow \\ \|e\|_\infty &\leq \frac{h^{n+1}}{4(n+1)} \|f^{(n+1)}\|_\infty. \end{aligned} \quad (18)$$

**Příklad 3.2.** Mějme funkci  $f(x) = \sin(\pi x)$ . Provedme na ní Lagrangeovu interpolaci v ekvidistantní síti uzlů na intervalu  $\langle -1, 1 \rangle$ . Tabulka níže zobrazuje odhad maximální absolutní chyby (18) a vypočtenou největší absolutní chybu mezi funkcí  $f$  a Lagrangeovým polynomem stupně  $n$  v intervalu  $\langle -1, 1 \rangle$ .

$n$	$\ e\ _\infty$	odhad max. chyby (18)
2	1	2,5839
4	0,1808	0,4782
8	$1,2055 \cdot 10^{-3}$	$3,1587 \cdot 10^{-3}$
16	$6,6540 \cdot 10^{-10}$	$1,8472 \cdot 10^{-9}$

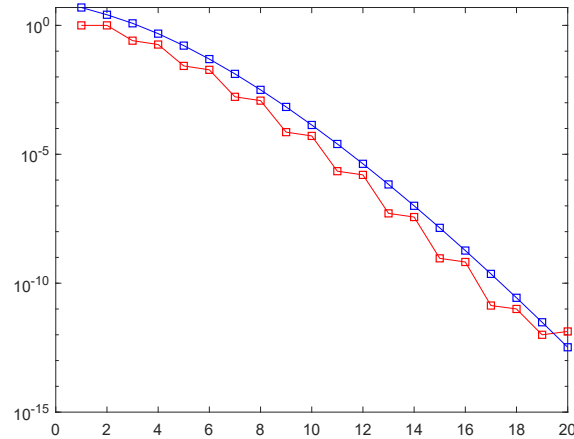
Nalezení odhadu maximální absolutní chyby (18) je často náročné, protože výpočet normy  $\|f^{(n+1)}\|_\infty$  může být velmi komplikovaný. V našem příkladě je však výpočet jednoduchý. Pojdme se na něj zaměřit. Nejdřív potřebujeme funkci několikrát zderivovat:

$$\begin{aligned} f(x) &= \sin(\pi x), \\ f'(x) &= \pi \cos(\pi x), \\ f''(x) &= -\pi^2 \sin(\pi x), \\ f'''(x) &= -\pi^3 \cos(\pi x), \\ f^{(4)}(x) &= \pi^4 \sin(\pi x), \\ &\vdots \end{aligned}$$

Následně hledáme  $\max_{x \in \langle -1, 1 \rangle} |f^{(n+1)}(x)|$ . Výpočet maxima je ovšem prostý, protože  $|\cos(\pi x)|$  nebo  $|\sin(\pi x)|$  v intervalu  $\langle -1, 1 \rangle$  může mít nejvyšší hodnotu 1. Tedy  $\|f^{(n+1)}\|_\infty = \pi^{n+1}$ .

V grafu na obr. 4 máme znázorněnou největší absolutní chybu Lagrangeovy interpolace na uzlech daných uniformním dělením a odhad maximální chyby v intervalu  $\langle -1, 1 \rangle$ . Vodorovná osa v grafu představuje stupeň interpolačního polynomu (hodnotu stupně volíme

$n = 1, 2, \dots, 20$ ). *Svislá osa pak představuje vypočtenou největší absolutní chybu na daném intervalu (označeno červeně) a odhad maximální absolutní chyby daný vztahem (18) (označeno modře).*



Obrázek 4: Největší absolutní chyba Lagrangeovy interpolace v ekvidistantní síti uzlů a odhad maximální absolutní chyby pro funkci  $f(x) = \sin(\pi x)$  na  $\langle -1, 1 \rangle$

▲

V uniformní síti se chyba kumuluje na krajích intervalu. Tento problém řeší Čebyševovy uzly, které chybu rovnoměrně distribuují do celého intervalu.

### 3.2 Dělení Čebyševovými uzly

Jak najít Čebyševovy uzly pro interval  $\langle -1, 1 \rangle$  již známe ze vztahu (9):

$$\xi_i = \cos\left(\frac{2i-1}{2n}\pi\right), \quad i = 1, 2, \dots, n.$$

Odtud lze odvodit, že pro obecný interval  $\langle a, b \rangle$  nalezneme Čebyševovy uzly takto:

$$\xi_i = \frac{1}{2}(a+b) + \frac{1}{2}(b-a)\cos\left(\frac{2i-1}{2n}\pi\right), \quad i = 1, 2, \dots, n. \quad (19)$$

**Tvrzení 3.2.** *Pro interpolaci v Čebyševových uzlech  $\xi_i \in \langle -1, 1 \rangle$  platí, že*

$$\max_{x \in \langle -1, 1 \rangle} \prod_{i=0}^n |x - \xi_i| = 2^{-n} \leq \max_{x \in \langle -1, 1 \rangle} \prod_{i=0}^n |x - x_i|,$$

kde  $\{x_i\}_{i=0}^n$  jsou libovolné jiné uzly v intervalu  $\langle -1, 1 \rangle$ .

**Důkaz.** *Důkaz tohoto tvrzení si ukážeme v následující kapitole, viz vztah (26) na str. 26.* ■

Z předchozího tvrzení je zřejmé, že odhad maximální chyby u intervalu  $\langle -1, 1 \rangle$  rozděleného pomocí Čebyševových uzlů je:

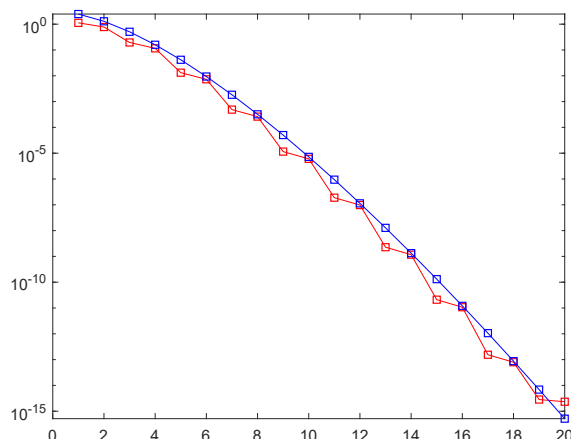
$$\|e\|_\infty \leq \frac{\|f^{(n+1)}\|_\infty}{2^n(n+1)!} \quad (20)$$

Lze ukázat, že pro obecný interval  $\langle a, b \rangle$  budou ve smyslu Tvrzení 3.2 optimální stejné uzly, jen vhodně posunuté a naškálované - viz (19).

**Příklad 3.3.** Mějme funkci  $f(x) = \sin(\pi x)$ . Provedme na ní Lagrangeovu interpolaci na Čebyševových uzlech na intervalu  $\langle -1, 1 \rangle$ . Tabulka níže zobrazuje odhad maximální absolutní chyby (20) a vypočtenou největší absolutní chybu mezi funkcí  $f$  a Lagrangeovým polynomem stupně  $n$  v intervalu  $\langle -1, 1 \rangle$ .

$n$	$\ e\ _\infty$	odhad max. chyby (20)
2	0,7754	1,2919
4	0,1156	0,1594
8	$2,6115 \cdot 10^{-4}$	$3,2088 \cdot 10^{-4}$
16	$1,0727 \cdot 10^{-11}$	$1,2134 \cdot 10^{-11}$

Graf na obr. 5 znázorňuje vypočtenou největší absolutní chybu Lagrangeovy interpolace na Čebyševových uzlech a odhad maximální chyby v intervalu  $\langle -1, 1 \rangle$ . Vodorovná osa v grafu představuje stupeň interpolačního polynomu (hodnotu stupně volíme  $n = 1, 2, \dots, 20$ ). Svislá osa pak představuje největší absolutní chybu na daném intervalu (označeno červeně) a odhad maximální chyby daný vztahem (20) (označeno modře).



Obrázek 5: Největší absolutní chyba Lagrangeovy interpolace v Čebyševových uzlech a odhad chyby pro funkci  $f(x) = \sin(\pi x)$  na  $\langle -1, 1 \rangle$

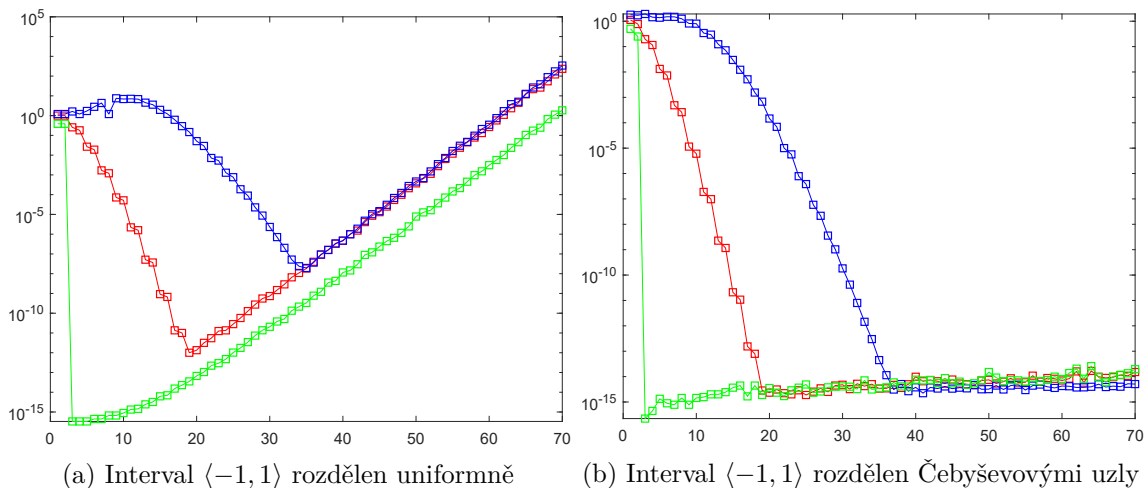
▲

### 3.3 Poznámka ke konvergenci polynomiální interpolace

V grafech na obr. 4 a obr. 5 pozorujeme, že pro  $n = 20$  je v obou příkladech vypočtená největší absolutní chyba vyšší než odhad maximální absolutní chyby. Nabízí se otázka, jak je to obecně s konvergencí chyby. Máme obecně zajištěno, že s přibývajícím stupněm  $n$  bude vypočtená největší absolutní chyba na intervalu klesat k nule?

**Příklad 3.4.** Mějme funkce  $f(x) = \sin(\pi x)$ ,  $g(x) = e^x \cos(2\pi x) \sin(2\pi x)$ ,  $h(x) = x^3$ . Na tyto funkce provedme na intervalu  $\langle -1, 1 \rangle$  Lagrangeovu interpolaci v ekvidistantní síti uzlů a Lagrangeovu interpolaci na Čebyševových uzlech. Hodnotu stupně polynomu volme  $n = 1, 2, \dots, 70$ . Do grafů opět vynesme největší absolutní chybu u jednotlivých funkcí.





Obrázek 6: Největší absolutní chyba Lagrangeovy interpolace u funkcí  $f, g, h$

Grafy na obr. 6 znázorňují největší absolutní chybu Lagrangeovy interpolace. Chyba je zobrazena pro  $f$  červeně, pro  $g$  modře a pro  $h$  zeleně. ▲

V předchozím příkladě můžeme pozorovat trend, kdy u interpolace v ekvidistantní síti uzlů má nejdříve největší absolutní chyba tendenci klesat k nule a od určitého stupně  $n$  se začne chyba významně zhoršovat. U Čebyševových uzlů zůstává největší absolutní chyba víceméně v počítačové přesnosti.

Obečně je obtížné pracovat s polynomy vysokých stupňů. Dále platí, že čím menší je původní interval, tím přesnější bude interpolace. Tyto myšlenky vedou na polynomiální interpolaci po částech s použitím tzv. splinů. Více např. v [1,2].

## 4 Aproximace typu minimax

V předchozích kapitolách jsme si vysvětlili způsob, jak lze nahrazovat (komplikovanou) funkci polynomem. Výběr interpolačních uzlů byl pro nás při tomto procesu klíčový. Nyní se podíváme na alternativní přístup, který přímo optimalizuje kvalitu aproximace. Tu budeme měřit pomocí maximální absolutní chyby na daném intervalu. Definujme nejprve úlohu nalezení nejlepší stejnoměrné aproximace (tzv. aproximace typu minimax).

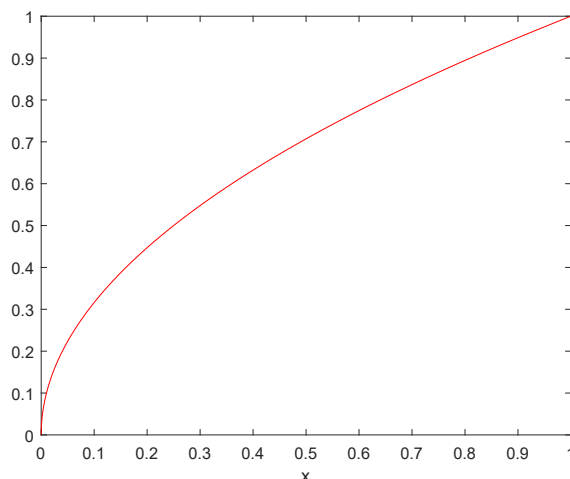
**Definice 4.1** (Aproximace typu minimax). *Bud'  $f \in C(\langle a, b \rangle)$ . Najděme polynom nejlepší stejnoměrné aproximace (v anglické literatuře polynom minimax)  $p_* \in \mathcal{P}_n$  funkce  $f$  na  $\langle a, b \rangle$  tak, že*

$$\max_{x \in \langle a, b \rangle} |f(x) - p_*(x)| = \min_{p \in \mathcal{P}_n} \max_{x \in \langle a, b \rangle} |f(x) - p(x)|. \quad (21)$$

Pro  $g \in C(\langle a, b \rangle)$  definujeme normu  $\|g\|_\infty = \max_{x \in \langle a, b \rangle} |g(x)|$ . Proto můžeme úlohu (21) zapsat takto: Najděme  $p_* \in \mathcal{P}_n$  tak, že

$$\|f - p_*\|_\infty = \min_{p \in \mathcal{P}_n} \|f - p\|_\infty. \quad (22)$$

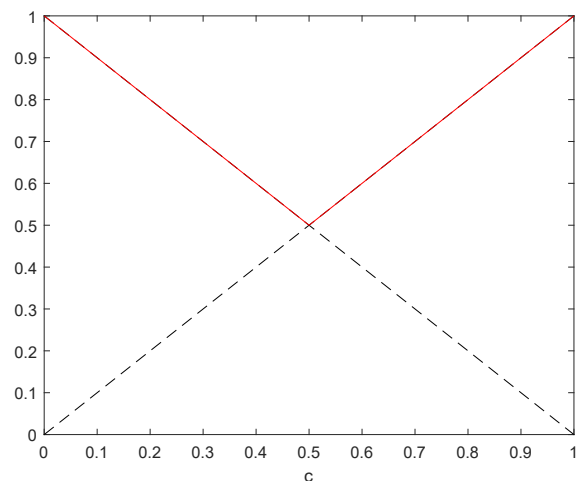
**Příklad 4.1.** *Hledejme konstantu, která co nejlépe aproximuje ve smyslu (22) funkci  $\sqrt{x}$  na intervalu  $\langle 0, 1 \rangle$ .*



Obrázek 7:  $\sqrt{x}$  na intervalu  $\langle 0, 1 \rangle$

**Řešení.** *Jelikož je  $\sqrt{x}$  monotónní funkce, tak musí hledaná konstanta  $p(x) = c$  ležet mezi  $\sqrt{0}$  a  $\sqrt{1}$ . Jinými slovy platí  $0 \leq c \leq 1$ . Protože je  $\sqrt{x}$  monotónní funkce, tak je i  $\sqrt{x} - c$  monotónní funkcí. Maximální chyba  $\max_{x \in \langle 0, 1 \rangle} |\sqrt{x} - c|$  je proto nabyta v jednom z krajních bodů intervalu  $\langle 0, 1 \rangle$ , tj.:*

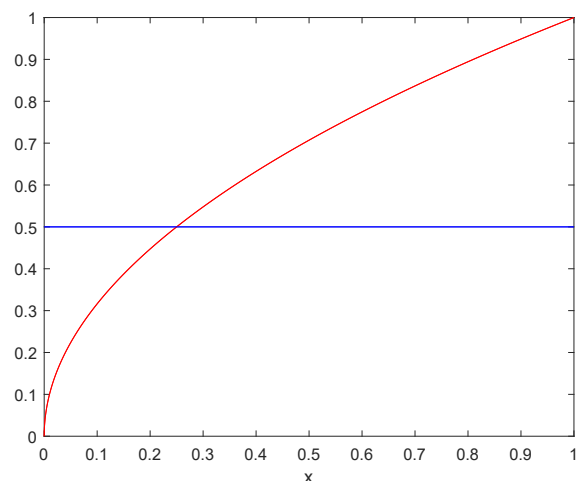
$$\begin{aligned} \|\sqrt{x} - c\|_\infty &= \max \left\{ \left| \sqrt{0} - c \right|, \left| \sqrt{1} - c \right| \right\} \\ &\Downarrow \\ \|\sqrt{x} - c\|_\infty &= \max \{c, 1 - c\} \\ &\Downarrow \text{viz obr. 8} \\ \min_{c \in \langle 0, 1 \rangle} \max \{c, 1 - c\} &= \frac{1}{2} \end{aligned}$$



Obrázek 8:  $\max\{c, 1 - c\}$

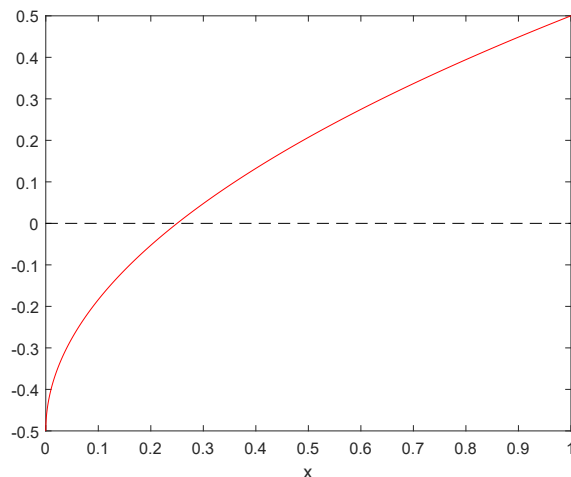
Můžeme tedy říct, že nejlepším konstantním polynomem aproximujícím  $\sqrt{x}$  na  $\langle 0, 1 \rangle$  je polynom

$$p_*(x) = \frac{1}{2}.$$



Obrázek 9: Nejlepší konstantní stejnoměrná aproximace  $\sqrt{x}$  na  $\langle 0, 1 \rangle$

Učiňme ještě následující pozorování: vynesme si do grafu chybu  $\sqrt{x} - p_*(x)$  pro  $x \in \langle 0, 1 \rangle$ .



Obrázek 10: Funkce  $\sqrt{x} - p_*(x)$  na intervalu  $\langle 0, 1 \rangle$

Vidíme, že chyba je největší na okrajích intervalu a v absolutní hodnotě nabývá stejné hodnoty. Tuto vlastnost budeme nazývat jako maximální chybu nabytou s alternujícím znaménkem. Pro nalezení nejlepší stejnoměrné aproximace je klíčová. ▲

**Věta 4.1** (Čebyševova věta o alternaci/Alternanční věta/Equioscillation theorem). Polynom  $p_* \in \mathcal{P}_n$  řeší úlohu (22) právě tehdy, když v intervalu  $\langle a, b \rangle$  existuje  $(n + 2)$  bodů  $x_0 < x_1 < \dots < x_{n+1}$  takových, že platí

$$|f(x_j) - p_*(x_j)| = \|f - p_*\|_\infty, \quad j = 0, 1, \dots, n + 1,$$

a znaménko chyby aproximace v těchto bodech alternuje:

$$f(x_j) - p_*(x_j) = -(f(x_{j+1}) - p_*(x_{j+1})), \quad j = 0, 1, \dots, n.$$

**Důkaz.** Důkaz této věty můžeme nalézt například v [2]. ■

Chyba  $f - p_*$  tedy nabývá v bodech  $x_0, x_1, \dots, x_{n+1}$  střídavě svého globálního minima a maxima na  $\langle a, b \rangle$ . Tato vlastnost se nazývá Čebyševova alternační vlastnost.

**Věta 4.2** (Jednoznačnost řešení problému (22)). Polynom nejlepší stejnoměrné aproximace je určen jednoznačně.

**Důkaz.** Důkaz nalezneme například v [3]. ■

Na alternační větě jsou založeny algoritmy hledající polynom nejlepší stejnoměrné aproximace. Iterativně upravují aproximující polynom, dokud nespĺňuje alternační podmínku. Nejznámějším takovým algoritmem je Remezův algoritmus. S ním se seznámíme v následující kapitole.

**Příklad 4.2.** Hledejme lineární funkci, která co nejlépe aproximuje ve smyslu (22) funkci  $\sqrt{x}$  na intervalu  $\langle 0, 1 \rangle$ .

**Řešení.** Hledáme  $p_* \in \mathcal{P}_1$ . Polynom  $p_*(x)$  je tedy ve tvaru  $\alpha + \beta x$ . Jelikož je  $\sqrt{x}$  konkávní, tak bude maximální chyba  $\delta$  nabyta v koncových bodech intervalu  $\langle 0, 1 \rangle$ . To znamená, že  $x_0 = 0, x_2 = 1$ . Nyní musíme nalézt třetí bod  $x_1 \in (0, 1)$ , ve kterém bude chyba nabývat hodnoty  $-\delta$ . Když nalezneme takový bod, tak potom podle alternační věty dostaneme, že příslušný polynom  $p_*$  je řešením (22).

Tedy:

$$\left. \begin{array}{l} \sqrt{x_0} - p_*(x_0) = \delta \\ \sqrt{x_1} - p_*(x_1) = -\delta \\ \sqrt{x_2} - p_*(x_2) = \delta \end{array} \right\} \Rightarrow \left. \begin{array}{l} 0 - \alpha - \beta \cdot 0 = \delta \\ \sqrt{x_1} - \alpha - \beta x_1 = -\delta \\ \sqrt{1} - \alpha - \beta = \delta \end{array} \right\} \Rightarrow \left. \begin{array}{l} -\alpha = \delta \\ \sqrt{x_1} - \alpha - \beta x_1 = -\delta \\ 1 - \alpha - \beta = \delta \end{array} \right\}$$

Nyní odečteme třetí rovnici od první:

$$-1 + \beta = 0 \Rightarrow \beta = 1$$

Všimněme si, že když známe hodnotu  $\beta$ , tak už můžeme zjistit hodnotu  $x_1$ . Hledáme totiž, kde se nachází extrém funkce  $\sqrt{x} - \alpha - \beta x$ . Jinými slovy hledáme stacionární bod této funkce:

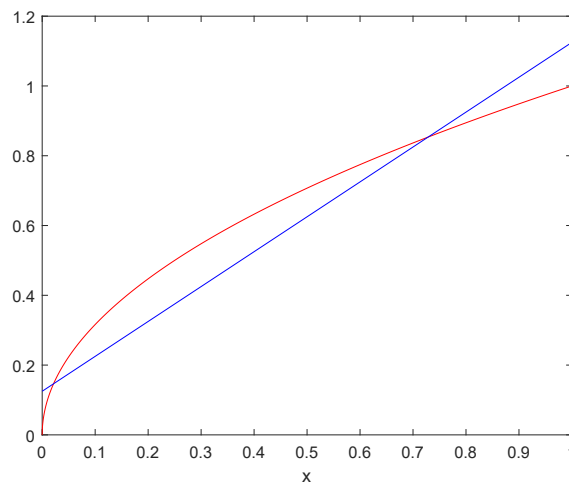
$$(\sqrt{x} - \alpha - \beta x)'(x_1) = 0, x_1 \in (0, 1) \Rightarrow \frac{1}{2\sqrt{x_1}} - \beta = 0 \Rightarrow \frac{1}{2\sqrt{x_1}} = 1 \Rightarrow \sqrt{x_1} = \frac{1}{2} \Rightarrow x_1 = \frac{1}{4}$$

Nyní sečteme první rovnici s druhou, abychom zjistili hodnotu  $\alpha$ .

$$\left. \begin{array}{l} -\alpha = \delta \\ \sqrt{x_1} - \alpha - \beta x_1 = -\delta \end{array} \right\} \Rightarrow \left. \begin{array}{l} -\alpha = \delta \\ \sqrt{\frac{1}{4}} - \alpha - 1 \cdot \frac{1}{4} = -\delta \end{array} \right\} \Rightarrow \frac{1}{2} - 2\alpha - \frac{1}{4} = 0 \Rightarrow \alpha = \frac{1}{8}$$

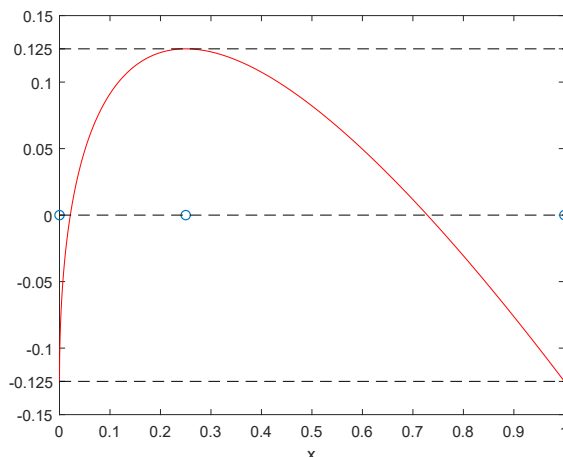
S nalezením hodnoty proměnné  $\alpha$  je zřejmé, že  $\delta = -\frac{1}{8}$ . Jelikož známe hodnoty  $\alpha$  a  $\beta$ , tak víme, že polynom  $p_*$  je ve tvaru:

$$p_*(x) = \frac{1}{8} + x.$$



Obrázek 11: Nejlepší lineární stejnoměrná aproximace  $\sqrt{x}$  na  $\langle 0, 1 \rangle$

Obdobně jako v předchozím příkladě si vykreslíme chybu  $\sqrt{x} - \frac{1}{8} - x$ ,  $x \in \langle 0, 1 \rangle$ .



Obrázek 12: Funkce  $\sqrt{x} - p_*(x)$  na intervalu  $\langle 0, 1 \rangle$

▲

#### 4.1 Čebyševovy uzly pro optimální aproximaci

Připomeňme si odhad chyby interpolace funkce  $f \in C^{n+1}(\langle a, b \rangle)$  polynomem  $p_n \in \mathcal{P}_n$  na intervalu  $\langle a, b \rangle$  (viz Větu 3.2 na str. 15):

$$(\forall x \in \langle a, b \rangle)(\exists \xi \in \langle a, b \rangle) : f(x) - p_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{j=0}^n (x - x_j),$$

kde  $\{x_j\}_{j=0}^n$  jsou uzly interpolace. Odtud dostáváme odhad maximální absolutní chyby interpolace na  $\langle a, b \rangle$ :

$$\|f - p_n\|_\infty \leq \frac{\|f^{(n+1)}\|_\infty}{(n+1)!} \max_{x \in \langle a, b \rangle} \prod_{j=0}^n |x - x_j|.$$

První část pravé strany nerovnosti  $\frac{\|f^{(n+1)}\|_\infty}{(n+1)!}$  je daná a nemůžeme ji nijak ovlivnit. Druhou část  $\max_{x \in \langle a, b \rangle} \prod_{j=0}^n |x - x_j|$  však ovlivnit lze. Tento člen chceme učinit co nejmenším a toho docílíme správnou volbou interpolačních uzlů  $x_j$ . Zabývejme se úlohou:

$$\min_{x_0, x_1, \dots, x_n} \max_{x \in \langle a, b \rangle} \prod_{j=0}^n |x - x_j| \quad (23)$$

Jinými slovy hledáme tzv. optimální uzly  $x_0, x_1, \dots, x_n$  minimalizující  $\max_{x \in \langle a, b \rangle} \prod_{j=0}^n |x - x_j|$ .

Platí:

$$\prod_{j=0}^n (x - x_j) = x^{n+1} - r_n(x),$$

kde  $r_n$  je polynom závislý na interpolačních uzlech  $x_0, x_1, \dots, x_n$ , a lze si rozmyslet, že je ve tvaru:

$$r_n(x) = x^n \sum_{j=0}^n x_j - x^{n-1} \sum_{j=0}^{n-1} \sum_{k=j+1}^n x_j x_k + \dots - (-1)^{n+1} \prod_{j=0}^n x_j, \quad r_n \in \mathcal{P}_n.$$

**Příklad 4.3.**

$$\begin{aligned}
 \bullet n = 1 : & \quad (x - x_0)(x - x_1) = x^2 - x(x_0 + x_1) + x_1x_0 = x^2 - \overbrace{(x(x_0 + x_1) - x_1x_0)}^{r_1(x)} \\
 \bullet n = 2 : & \quad (x - x_0)(x - x_1)(x - x_2) = (x^2 - x(x_0 + x_1) + x_1x_0)(x - x_2) = \\
 & = x^3 - x_2x^2 - x^2(x_0 + x_1) + x(x_0 + x_1)x_2 + x_1x_0x - x_0x_1x_2 = \\
 & = x^3 - \underbrace{(x^2(x_0 + x_1 + x_2) - x(x_0x_1 + x_0x_2 + x_1x_2) + x_0x_1x_2)}_{r_2(x)}
 \end{aligned}$$

▲

Bud

$$\widetilde{\mathcal{P}}_n = \{r_n(x) = x^{n+1} - \prod_{j=0}^n (x - x_j), \quad x_0, x_1, \dots, x_n \in \mathbb{R}\}.$$

Tedy:

$$\begin{aligned}
 & \text{úloha (23)} \\
 & \quad \Downarrow \\
 & \min_{r_n \in \widetilde{\mathcal{P}}_n} \max_{x \in \langle a, b \rangle} |x^{n+1} - r_n(x)| \\
 & \quad \Updownarrow \\
 & \min_{r_n \in \widetilde{\mathcal{P}}_n} \|x^{n+1} - r_n\|_\infty \tag{24}
 \end{aligned}$$

Nyní proto hledáme polynom  $r_{n,*} \in \widetilde{\mathcal{P}}_n$  takový, že

$$\|x^{n+1} - r_{n,*}\|_\infty = \min_{r_n \in \widetilde{\mathcal{P}}_n} \|x^{n+1} - r_n\|_\infty.$$

Pro jednoduchost se nyní omezíme na  $\langle a, b \rangle = \langle -1, 1 \rangle$ . Označme pomocí  $\hat{T}_{n+1}$  normalizovaný Čebyševův polynom stupně  $(n + 1)$  s vedoucím koeficientem, který je roven 1, tj.:

$$\hat{T}_{n+1} = 2^{-n}T_{n+1}.$$

Proto

$$\hat{T}_{n+1}(x) = x^{n+1} - r_n(x)$$

pro daný polynom  $\mathcal{P}_n \ni r_n(x) = x^{n+1} - \hat{T}_{n+1}(x)$ . Polynom  $\hat{T}_{n+1}$  osciluje mezi  $\pm 2^{-n}$  na  $\langle -1, 1 \rangle$ . Dále díky Tvzení 2.2 víme, že  $\hat{T}_{n+1}$  má  $(n + 2)$  extrémů na  $\langle -1, 1 \rangle$  v bodech  $\eta_j = \cos\left(\frac{j}{n+1}\pi\right)$ ,  $j = 0, 1, \dots, n + 1$ . Tedy

$$\hat{T}_{n+1}(\eta_j) = \eta_j^{n+1} - r_n(\eta_j) = (-1)^j 2^{-n}, \quad j = 0, 1, \dots, n + 1. \tag{25}$$

Našli jsme tudíž polynom  $r_n \in \mathcal{P}_n$  společně s  $(n + 2)$  uzly  $\eta_j \in \langle -1, 1 \rangle$  takový, že

$$\left| \eta_j^{n+1} - r_n(\eta_j) \right| = \max_{x \in \langle -1, 1 \rangle} |x^{n+1} - r_n(x)| = 2^{-n}, \quad j = 0, 1, \dots, n + 1.$$

Znaménko funkční hodnoty funkce  $x^{n+1} - r_n(x)$  v těchto bodech alternuje (viz (25)). Z ekviosilační věty tedy plyne, že  $r_n$  je polynomem nejlepší stejnoměrné aproximace  $x^{n+1}$  na  $\langle -1, 1 \rangle$ .

**Tvrzení 4.1.** Polynom nejlepší stejnoměrné aproximace funkce  $x^{n+1}$  na  $\langle -1, 1 \rangle$  má tvar

$$r_{n,*}(x) = x^{n+1} - 2^{-n}T_{n+1}(x) \in \mathcal{P}_n.$$

Jelikož  $r_{n,*}(x) = x^{n+1} - 2^{-n}T_{n+1}(x) = x^{n+1} - \prod_{j=0}^n (x - \xi_j)$ , tak platí, že  $r_{n,*} \in \widetilde{\mathcal{P}}_n \subset \mathcal{P}_n$ .

Proto kořeny  $\xi_j$  Čebyševova polynomu  $T_{n+1}$  jsou optimálními interpolačními uzly (viz ekvivalenci úloh (23) a (24)). Navíc

$$\max_{x \in \langle -1, 1 \rangle} \prod_{j=0}^n |x - \xi_j| = 2^{-n} \leq \max_{x \in \langle -1, 1 \rangle} \prod_{j=0}^n |x - x_j|, \quad (26)$$

kde  $\{x_j\}_{j=0}^n$  jsou libovolné jiné interpolační uzly (jiné než Čebyševovy) v intervalu  $\langle -1, 1 \rangle$ . Tímto jsme tedy dokázali Tvrzení 3.2 na str. 17. Tedy pro Čebyševovy uzly dostáváme pro  $f \in C^{n+1}(\langle -1, 1 \rangle)$  odhad chyby interpolace na  $\langle -1, 1 \rangle$ :

$$\|f - p_n\|_\infty \leq \frac{\|f^{(n+1)}\|_\infty}{2^n(n+1)!}.$$

## 4.2 Remezův algoritmus

V roce 1853 ruský matematik Pafnutij Lvovič Čebyšev formuloval pro  $f \in C(\langle a, b \rangle)$  problém:

$$\text{Najdi } p_* \in \mathcal{P}_n \text{ tak, aby } \|p_* - f\|_\infty = \min_{p \in \mathcal{P}_n} \|p - f\|_\infty. \quad (27)$$

- Víme, že (27) má jednoznačné řešení (viz Větu 4.2 na str. 22).
- Dále víme, že  $p_*$  je plně charakterizován Čebyševovou alternační vlastností (tj. že chyba  $f - p_*$  nabývá v bodech  $x_0, x_1, \dots, x_{n+1} \in \langle a, b \rangle$ ,  $x_0 < x_1 < \dots < x_{n+1}$ , střídavě svého globálního minima a globálního maxima na  $\langle a, b \rangle$ ) (viz Větu 4.1 na str. 22).

Tyto poznatky lze dále zobecnit tak, že  $\mathcal{P}_n$  nahradíme  $(n+1)$ -dimenzionálním podprostorem  $C(\langle a, b \rangle)$

$$Q = \text{Lin}\{g_0, g_1, \dots, g_n\},$$

kde  $g_0, g_1, \dots, g_n$  jsou lineárně nezávislé funkce spojité na  $\langle a, b \rangle$ , které splňují tzv. Haarovu podmínku:

$$\begin{vmatrix} g_0(x_0) & g_1(x_0) & \cdots & g_n(x_0) \\ g_0(x_1) & g_1(x_1) & \cdots & g_n(x_1) \\ \vdots & \vdots & \ddots & \vdots \\ g_0(x_n) & g_1(x_n) & \cdots & g_n(x_n) \end{vmatrix} \neq 0 \text{ pro libovolné různé } x_0, x_1, \dots, x_n \in \langle a, b \rangle. \quad (28)$$

Funkce  $\sum_{i=0}^n c_i g_i \in Q$ ,  $c_i \in \mathbb{R}$ , pak nazveme tzv. zobecněnými polynomy.

**Příklad 4.4.** Lze ukázat, že systém  $\{1, x, \dots, x^n\}$  splňuje Haarovu podmínku, jelikož

$$\begin{vmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ 1 & x_2 & x_2^2 & \cdots & x_2^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{vmatrix} = \prod_{0 \leq j < i \leq n} (x_i - x_j)$$

je dobře známý Vandermondův determinant, který je zřejmě nenulový pro různá  $x_0, x_1, \dots, x_n$ . ▲



**Věta 4.3** (Zobecnění Čebyševovy věty o alternaci). *Bud'*

- $\{g_j\}_{j=1}^n$  systém spojitých funkcí na  $\langle a, b \rangle$  splňujících Haarovu podmínku,
- $X$  uzavřená podmnožina  $\langle a, b \rangle$  obsahující alespoň  $(n + 1)$  bodů,
- $f$  spojitá na  $X$ ,
- $r(x) = f(x) - \sum_{j=1}^n c_j g_j(x)$ ,  $r$  nazýváme reziduem.

Pak koeficienty  $c_1, c_2, \dots, c_n$  minimalizují

$$\max_{x \in X} |r(x)|$$

právě tehdy, když

$$r(x_{i+1}) = -r(x_i) = \pm \max_{x \in X} |r(x)|, \quad i = 0, 1, \dots, n - 1$$

pro  $x_0 < x_1 < \dots < x_n$ ,  $x_0, x_1, \dots, x_n \in X$ .

**Důkaz.** Důkaz je popsán například v [4]. ■

Jak tedy nalezneme polynom  $p_*$ , abychom vyřešili úlohu (27)? To znamená, jak najdeme polynom nejlepší stejnoměrné aproximace funkce na intervalu  $\langle a, b \rangle$ ? V roce 1934 ruský matematik Jevgenij Jakovlevič Remez publikoval iterační proceduru, která konverguje k nejlepší stejnoměrné aproximaci dané funkce  $f \in C(\langle a, b \rangle)$  pomocí zobecněných polynomů  $\sum_{i=1}^n c_i g_i$ , kde systém  $\{g_1, g_2, \dots, g_n\}$  splňuje Haarovu podmínku. Tato procedura je založena na Větě 4.3.

Základní schéma Remezova algoritmu:

Uvažujme nějakou počáteční množinu  $(n + 1)$  uzlů v  $\langle a, b \rangle$  (tzv. referenci).

- 1) vypočteme  $q = \sum_{i=1}^n c_i g_i$  minimalizující maximum absolutní hodnoty rezidua na dané množině uzlů (vede na řešení SLR),
- 2) ověříme kvalitu nalezené aproximace  $q$ ,
- 3) upravíme danou referenci.

Z bodu 3) se opět vracíme do bodu 1), dokud  $q$  nebude „dost blízko“ nejlepší aproximaci.

Po výpočtu aproximace  $q$  v kroku 1) budeme muset vypočítat na  $\langle a, b \rangle$  lokální extrémy rezidua

$$r = f - q.$$

Tento krok může být náročný, jelikož těchto lokálních extrémů může být mnoho, případně funkce rezidua nemusí být na  $\langle a, b \rangle$  diferencovatelná.

Nyní se ptáme, jak budeme realizovat první krok Remezova algoritmu. Napoví nám následující důsledek Věty 4.3.

**Důsledek 4.1.** *Bud'*

- $\{g_j\}_{j=1}^n$  systém spojitých funkcí na  $\langle a, b \rangle$  splňující Haarovu podmínku,
  - $f \in C(\langle a, b \rangle)$  aproximovaná funkce,
  - $(n + 1)$  uzlů v  $\langle a, b \rangle$ , pro které platí  $a \leq x_0 < x_1 < \dots < x_n \leq b$  (tzv. reference).
- Pak koeficienty  $c_1, c_2, \dots, c_n$  minimalizující

$$\max_{i=0,1,\dots,n} |r(x_i)|$$

získáme jako řešení soustavy  $n$  lineárních rovnic o  $n$  neznámých:

$$\sum_{j=1}^n c_j (g_j(x_i) - (-1)^i g_j(x_0)) = f(x_i) - (-1)^i f(x_0), \quad i = 1, 2, \dots, n. \quad (29)$$

**Důkaz.** Označme  $q = \sum_{j=1}^n c_j g_j$ . Aplikujeme Větu 4.3 s  $X = \{x_0, x_1, \dots, x_n\}$ :

$$f(x_{i+1}) - q(x_{i+1}) = -(f(x_i) - q(x_i)), \quad i = 0, 1, \dots, n-1,$$

tedy

$$f(x_i) - q(x_i) = (-1)^i (f(x_0) - q(x_0)), \quad i = 1, 2, \dots, n.$$

Odtud

$$f(x_i) - \sum_{j=1}^n c_j g_j(x_i) = (-1)^i (f(x_0) - \sum_{j=1}^n c_j g_j(x_0)), \quad i = 1, 2, \dots, n,$$

a proto

$$\sum_{j=1}^n c_j (g_j(x_i) - (-1)^i g_j(x_0)) = f(x_i) - (-1)^i f(x_0), \quad i = 1, 2, \dots, n.$$

■

Před popisem Remezova algoritmu si ještě uvedme dvě jednoduchá pozorování.

**Pozorování 4.1.** Předpokládejme, že jsou splněny předpoklady Důsledku 4.1. Pomocí koeficientů  $c_1, c_2, \dots, c_n$ , které řeší danou SLR (29), definujeme reziduum

$$r(x) = f(x) - \sum_{j=1}^n c_j g_j(x).$$

Ze zobecněné Čebyševovy věty o alternaci víme (aplikujeme s  $X = \{x_0, x_1, \dots, x_n\}$ ), že

$$r(x_{i+1}) = -r(x_i) = \pm M, \quad i = 0, 1, \dots, n-1,$$

kdy  $M = |r(x_i)| = \text{konst.}$  pro  $i = 0, 1, \dots, n$ . Všimneme si tedy, že  $r$  alternuje ve znaménku na  $X$ . Z Darbouxovy vlastnosti spojitě funkce plyne, že funkce  $r$  má v každém intervalu  $(x_i, x_{i+1})$ ,  $i = 0, 1, \dots, n-1$ , kořen.

Hledáním kořenů rezidua v intervalu  $(x_i, x_{i+1})$  řešíme druhý krok Remezova algoritmu, který si blíže popíšeme později v této kapitole.

**Pozorování 4.2.** Necht' jsou opět splněny předpoklady Důsledku 4.1. Navíc označme:

- $q^*$  je nejlepší aproximace funkce  $f$  z  $Q = \text{Lin} \{g_1, g_2, \dots, g_n\}$ ,
- $q = \sum_{j=1}^n c_j g_j$ , kde  $c_1, c_2, \dots, c_n$  řeší SLR (29).

Zvolme nyní  $y \in \langle a, b \rangle$  takové, že  $|r(y)| = \max_{x \in \langle a, b \rangle} |r(x)|$ . Lze ukázat (viz například [5]), že

$$\|f - q\|_\infty \leq \|f - q^*\|_\infty + \delta, \quad \text{kde } \delta = |r(y)| - |r(x_0)|.$$

Tuto nerovnost použijeme jako ukončovací kritérium Remezova algoritmu. Jelikož tedy  $0 \leq \|f - q\|_\infty - \|f - q^*\|_\infty \leq \delta$ , tak ukončíme iterování algoritmu v momentě, kdy je  $\delta$  „dostatečně malé“ (tj. víme, že naše vypočtená aproximace  $q$  je „blízko“ nejlepší aproximace  $q^*$ ).

## Remezův algoritmus

### Vstup:

- interval  $\langle a, b \rangle$
- $f \in C(\langle a, b \rangle)$
- systém funkcí  $\{g_j\}_{j=1}^n$  spojitých na  $\langle a, b \rangle$  splňující Haarovu podmínku
- počáteční reference:  $a \leq x_0 < x_1 < \dots < x_n \leq b$
- konstanta  $\delta > 0$  pro ukončující podmínku
- proměnná  $k \in \mathbb{N}$ , která určuje počet iterací (vstupuje jako  $k = 1$ )

### Krok 1 (Nalezení $k$ -té aproximace)

- vyřešíme SLR (29) a získáme koeficienty  $c_1, c_2, \dots, c_n$  minimalizující výraz

$$\max_{i=0,1,\dots,n} \left| f(x_i) - \sum_{j=1}^n c_j g_j(x_i) \right|$$

- definujeme reziduum  $r = f - \sum_{j=1}^n c_j g_j$  s výše uvedenými koeficienty

$c_1, c_2, \dots, c_n$

### Ukončující podmínka

- najdeme  $y \in \langle a, b \rangle$  takové, že  $|r(y)| = \max_{x \in \langle a, b \rangle} |r(x)|$
- ukončíme iterační proces, pokud

$$|r(y)| - |r(x_0)| < \delta,$$

jinak pokračujeme krokem 2

### Krok 2 (Nalezení kořenů rezidua)

- $z_0 = a, z_{n+1} = b$
- najdeme kořen  $z_i$  rezidua  $r$  v intervalu  $(x_{i-1}, x_i)$ ,  $i = 1, 2, \dots, n$  (viz Pozorování 4.1)

### Krok 3 (Nalezení nové reference)

- $\sigma_i = \text{sgn}(r(x_i))$
- pro každé  $i = 0, 1, \dots, n$  najdeme  $y_i \in \langle z_i, z_{i+1} \rangle$  tak, že v  $y_i$  nabývá  $\sigma_i r(y)$  lokálního maxima a

$$\sigma_i r(y_i) \geq \sigma_i r(x_i)$$

- referenci  $\{x_0, x_1, \dots, x_n\}$  nahradíme referencí  $\{y_0, y_1, \dots, y_n\}$

### Krok 4 (Možné doplnění nové reference o bod $y$ )

Pro účely otestování ukončovací podmínky máme vypočteno

$$y \in \langle a, b \rangle \text{ takové, že } |r(y)| = \max_{x \in \langle a, b \rangle} |r(x)|.$$

- Pokud  $|r(y)| = \max_{i=0,1,\dots,n} |r(y_i)|$ , přejdeme na krok 1 s referenčními uzly definovanými v kroku 3.
- Pokud  $|r(y)| > \max_{i=0,1,\dots,n} |r(y_i)|$ , zařadíme  $y$  do referenční množiny  $\{y_0, y_1, \dots, y_n\}$  (na správné místo). Nyní musíme jeden z uzlů  $y_i$  odebrat tak, aby reziduum  $r$  stále alternovalo ve znaménku na výsledné referenční množině. Takto upravená množina je novou referenční množinou a pokračujeme krokem 1.
- $k = k + 1$

## Poznámky

1) V druhé odrážce kroku 1 funkce  $\sum_{j=1}^n c_j g_j$  představuje aproximaci vypočtenou v  $k$ -té iteraci, další kroky ovlivňují už pouze  $(k+1)$ -ní iteraci.

2) Krok 3 implementujeme pomocí `fminbnd()`, tj. funkce zabudované v Matlabu pro nalezení lokálního minima dané funkce na intervalu. Namísto  $\sigma_i r$  budeme na  $\langle z_i, z_{i+1} \rangle$  minimalizovat  $-\sigma_i r$ . Pokud `fminbnd()` nalezne takové lokální maximum  $y_i$ , že  $\sigma_i r(y_i) < \sigma_i r(x_i)$ , provedeme brute-force hledání maxima  $\sigma_i r(y)$  na  $\langle z_i, z_{i+1} \rangle$ . Pokud ani takto nalezený bod  $y_i$  nesplňuje  $\sigma_i r(y_i) \geq \sigma_i r(x_i)$ , pak  $y_i = x_i$ . Navíc, pokud jsme na koncových intervalech  $\langle z_0, z_1 \rangle$  a  $\langle z_n, z_{n+1} \rangle$  ( $z_0 = a$  a  $z_{n+1} = b$  nejsou obecně kořeny rezidua):

- $\sigma_1 r(y_1) < \sigma_1 r(a) \Rightarrow y_1 = a$ ,
- $\sigma_n r(y_n) < \sigma_n r(b) \Rightarrow y_n = b$ .

3) Strategie pro krok 4: Víme, že reziduum  $r$  alternuje ve znaménku na referenci  $y_0 < y_1 < \dots < y_n$ . Do této reference chceme začlenit bod  $y \in \langle a, b \rangle$  takový, že

$$|r(y)| = \max_{x \in \langle a, b \rangle} |r(x)|.$$

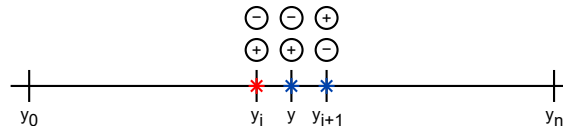
Začleňme tedy  $y$  na správné místo. Nyní musíme nějaké  $y_i$  odstranit, aby reziduum  $r$  stále alternovalo ve znaménku na výsledné referenci. Rozlišujeme 3 případy:

Případ 1:  $y_0 < y < y_n$

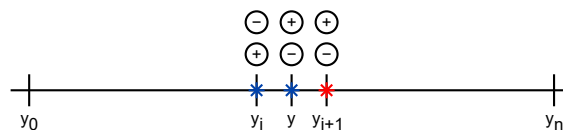
V prvním případě se bod  $y$  nachází někde mezi koncovými body reference. V tomto případě se rozhodujeme, který z vedlejších uzlů od  $y$  odstraníme, tj. zda odstraníme první bod nalevo nebo první bod napravo od  $y$ . Tyto body si pro naše účely označíme jako  $y_i$  a  $y_{i+1}$ . Platí  $y_i < y < y_{i+1}$  pro nějaké  $i \in \{0, 1, \dots, n-1\}$ . Nyní se zaměříme na znaménko  $r$  v těchto třech bodech. Mohou nastat pouze dvě možnosti.

Možnost 1: Reziduum  $r$  má v uzlech  $y_i$  a  $y$  stejné znaménko. Při této možnosti odstraňujeme z původní reference bod  $y_i$  a nahrazujeme ho bodem  $y$ . Tato situace je schématicky zobrazena na obr. 13. Plusy a mínusy v kolečkách symbolizují hodnotu znaménka  $r$  v uzlech  $y_i$ ,  $y_{i+1}$  a  $y$ , které jsou označeny hvězdičkami. Červená hvězdička označuje uzal, který bude nahrazen novým uzlem  $y$ .

Možnost 2: Reziduum  $r$  má v uzlech  $y_i$  a  $y$  různé znaménko. Když nastane tato možnost, tak v původní referenci nahrazujeme bod  $y_{i+1}$  novým uzlem  $y$ . Situace je opět schématicky zobrazena na obr. 14.



Obrázek 13: Schéma zobrazující 1. případ a jeho 1. možnost



Obrázek 14: Schéma zobrazující 1. případ a jeho 2. možnost

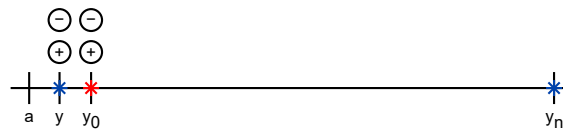
Případ 2:  $a \leq y < y_0$

V druhém případě se bod  $y$  nachází v polouzavřeném intervalu  $\langle a, y_0 \rangle$ . V tomto případě se rozhodujeme, zda odstraníme z původní reference uzel  $y_0$  nebo  $y_n$ . Zaměříme se na znaménka rezidua  $r$  v bodech  $y$  a  $y_0$ . Tak jako v prvním případě, i zde mohou nastat pouze dvě možnosti.

Možnost 1: Reziduum  $r$  má v uzlech  $y$  a  $y_0$  stejné znaménko. Pokud nastane tato možnost, odstraňujeme z původní reference bod  $y_0$  a nahrazujeme ho bodem  $y$ .

Možnost 2: Reziduum  $r$  má v uzlech  $y$  a  $y_0$  různé znaménko. Při této možnosti přidáváme do reference bod  $y$  a odebíráme bod  $y_n$ .

Schémata k jednotlivým možnostem druhého případu najdeme na obr. 15 a obr. 16. Hvězdičky značí body  $y$ ,  $y_0$  a  $y_n$ . Červená hvězdička symbolizuje uzel, který se po přidání uzlu  $y$  z reference vypustí.



Obrázek 15: Schéma zobrazující 2. případ a jeho 1. možnost



Obrázek 16: Schéma zobrazující 2. případ a jeho 2. možnost

Případ 3:  $y_n < y \leq b$

V třetím případě se bod  $y$  nachází v polouzavřeném intervalu  $(y_n, b)$ . V posledním případě odstraňujeme z původní reference bod  $y_0$  nebo  $y_n$ . Po odstranění uzlu opět přidáváme do reference uzel  $y$ . V tomto případě nás zajímají znaménka rezidua  $r$  u uzlů  $y$  a  $y_n$ . Tak jako v předešlých případech, i zde mohou vzniknout pouze dvě možnosti.

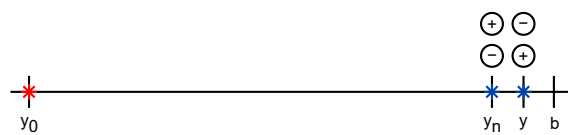
Možnost 1: Reziduum  $r$  má v uzlech  $y$  a  $y_n$  stejné znaménko. Potom odebíráme z reference bod  $y_n$  a následně přidáme bod  $y$ .

Možnost 2: Reziduum  $r$  má v uzlech  $y$  a  $y_n$  různé znaménko. V této situaci odstraníme z reference bod  $y_0$  a přidáváme bod  $y$ .

Na obr. 17 a obr. 18 pak můžeme vidět schémata k třetímu případu pro možnosti, které můžou nastat. Hvězdičky opět symbolizují uzly  $y$ ,  $y_0$  a  $y_n$ . Uzel, který je označen červenou hvězdičkou, bude v referenci vypuštěn.



Obrázek 17: Schéma zobrazující 3. případ a jeho 1. možnost



Obrázek 18: Schéma zobrazující 3. případ a jeho 2. možnost

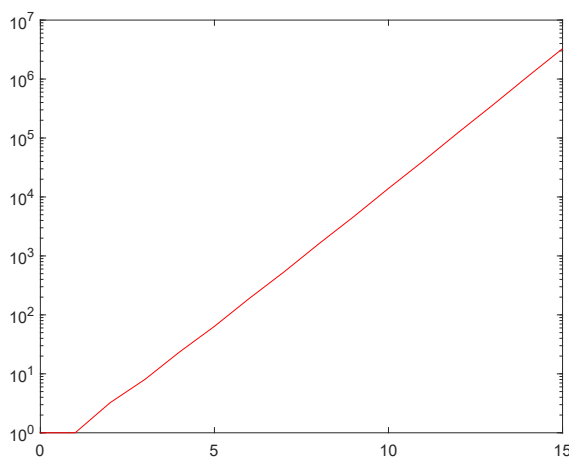
## 5 Numerické experimenty

Veškerá implementace byla provedena v programu MATLAB.

### 5.1 Číslo podmíněnosti Vandermondovy matice

V kapitole 2 jsme se seznámili s polynomiální interpolací. Jako první případ interpolace jsme si uvedli interpolaci pomocí Vandermondovy matice  $V$ . Tento naivní přístup se jeví jako dost neefektivní z důvodu vysokého čísla podmíněnosti matice  $V$ . S přibývajícím stupněm polynomu výrazně vzrůstá číslo podmíněnosti  $\kappa(V)$ . Růst tohoto čísla v závislosti na stupni polynomu můžeme sledovat v následující tabulce. Obr. 19 pak ukazuje zvyšování hodnoty čísla podmíněnosti Vandermondovy matice. Vodorovná osa v grafu představuje stupeň polynomu  $n = 0, 1, \dots, 15$ . Svislá osa pak zobrazuje hodnotu  $\kappa(V)$  pro polynomy daného stupně  $n$ .

Stupeň polynomu	1	2	3	6	10	15	21	28
$\kappa(V)$	1	3,2	8	189,8	$1,4 \cdot 10^4$	$3,3 \cdot 10^6$	$2,5 \cdot 10^9$	$6,1 \cdot 10^{12}$



Obrázek 19: Růst čísla podmíněnosti  $\kappa(V)$  v závislosti na stupni  $n$  aproximačního polynomu v logaritmickém měřítku

### 5.2 Experimenty pro Lagrangeovu a Čebyševovu interpolaci

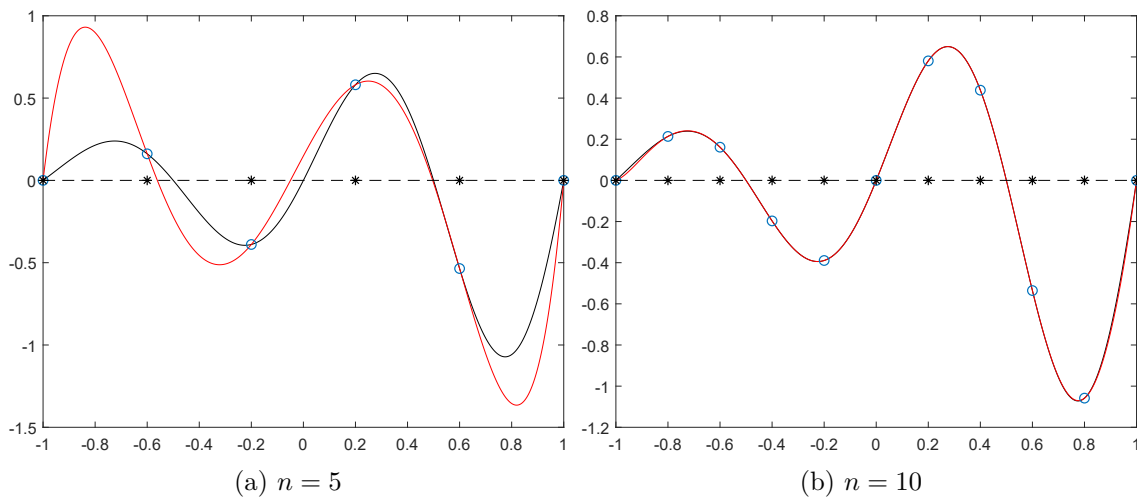
V této kapitole si vykreslíme dva způsoby interpolací na dvou níže zadaných funkcích. Nejdříve nalezneme pro daný počet interpolačních bodů Lagrangeův interpolační polynom příslušného stupně na ekvidistantí síti a na síti dané Čebyševovými uzly. Následně vykreslíme pro zadané funkce aproximaci Čebyševovými polynomy. Polynomy vytvořené touto metodou budou stejného stupně jako polynomy získané Lagrangeovou interpolací. Nakonec porovnáme grafy zobrazující největší absolutní chyby těchto interpolací.

Mějme funkce  $f$  a  $g$  dané předpisy

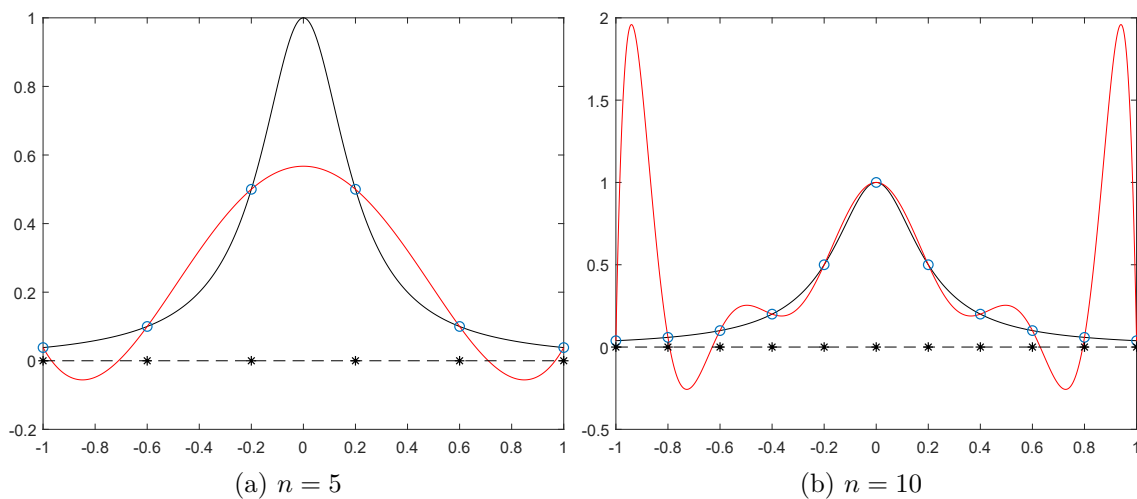
$$f(x) = e^x \sin(\pi x) \cos(\pi x), \quad x \in \langle -1, 1 \rangle,$$

$$g(x) = \frac{1}{1 + 25x^2}, \quad x \in \langle -1, 1 \rangle \text{ (škálovaná Rungeova funkce).}$$

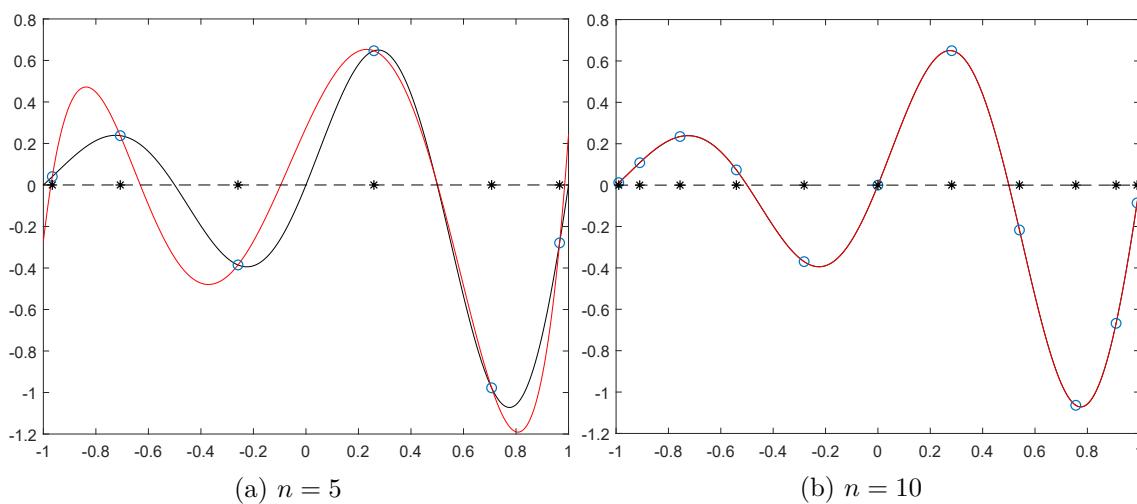
V následujících grafech jsou černě vykresleny původní funkce a interpolační polynomy pak červeně. Hvězdičky představují interpolační uzly, modrá kolečka znázorňují průsečíky původní funkce s interpolačním polynomem.



Obrázek 20: Lagrangeova interpolace na uniformním dělení intervalu funkce  $f$

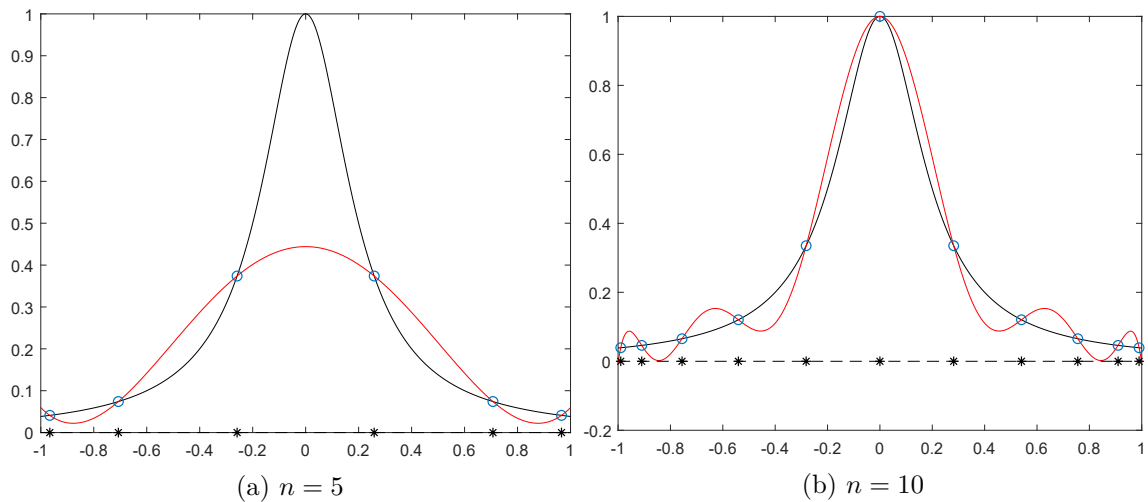


Obrázek 21: Lagrangeova interpolace na uniformním dělení intervalu funkce  $g$

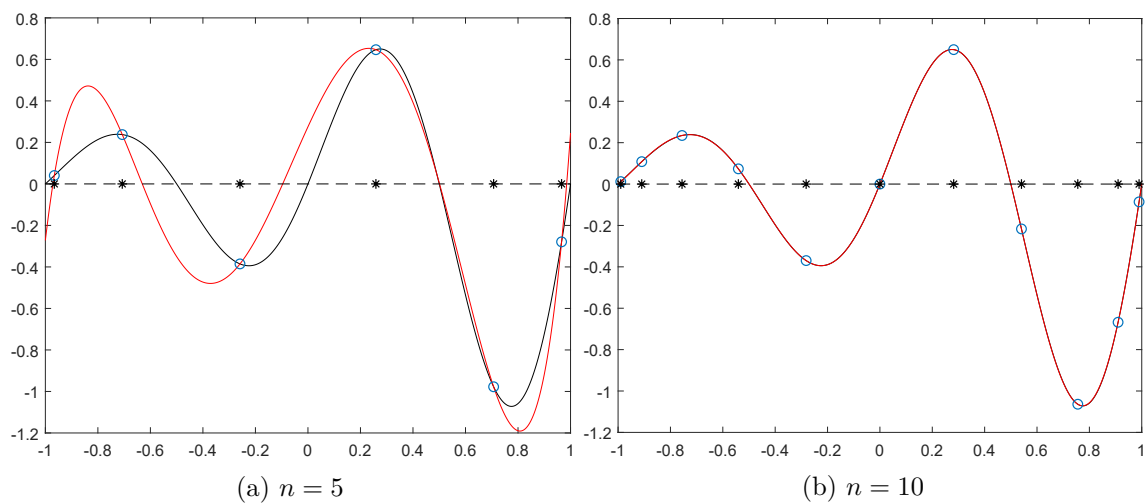


Obrázek 22: Lagrangeova interpolace na Čebyševových uzlech funkce  $f$

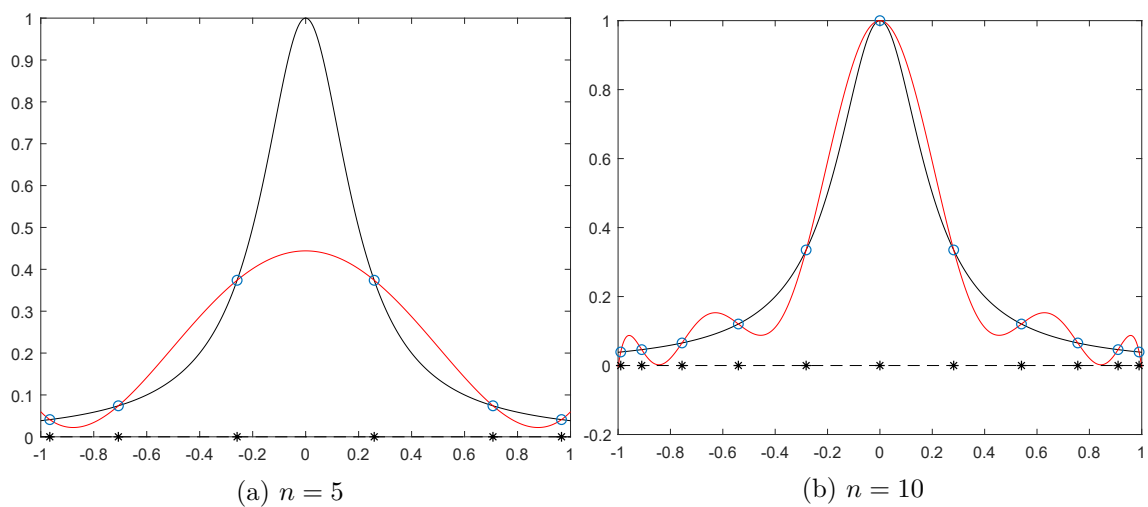




Obrázek 23: Lagrangeova interpolace na Čebyševových uzlech funkce  $g$



Obrázek 24: Čebyševova aproximace funkce  $f$



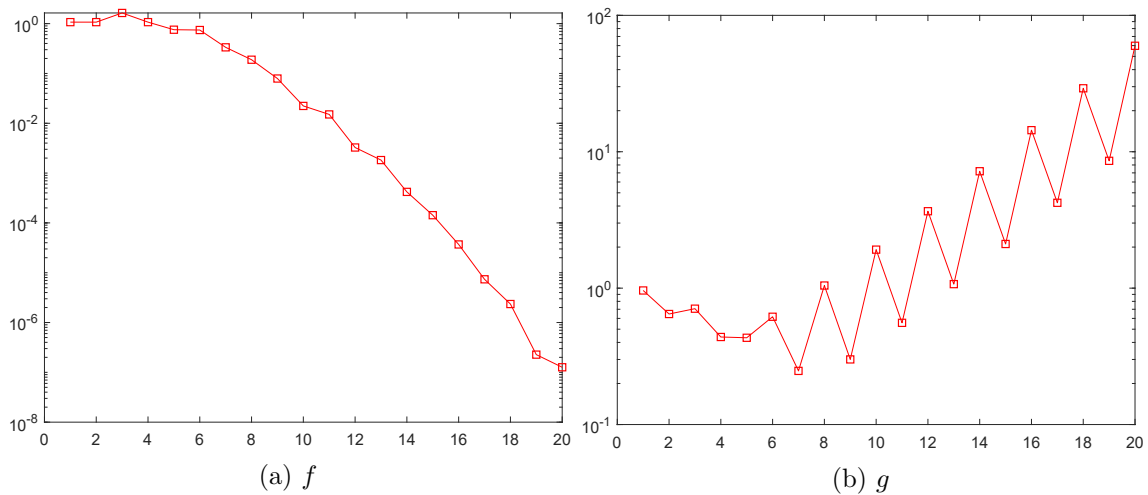
Obrázek 25: Čebyševova aproximace funkce  $g$

U funkce  $f$  vypadají všechny druhy interpolací dobře. Můžeme pozorovat, že kvalita interpolace se zlepšila, když jsme zvýšili stupeň polynomu z 5 na 10. U funkce  $g$  můžeme pozorovat, že Lagrangeova interpolace v ekvidistantní síti nedosahuje stejných kvalit jako zbývající druhy interpolací. U polynomu 10. stupně (viz obr. 21) pak můžeme sledovat Rungeův jev, který se projevuje na koncových oblastech intervalu. Rungeův jev vzniká při polynomiální interpolaci v ekvidistantní síti uzlů. Příčinou je, že hodnota derivací interpolované funkce  $n$ -tého řádu rychle roste se zvyšujícím se  $n$ . Od určitého  $n$  pak bude hodnota  $\|f^{(n+1)}\|_\infty$  vždy výrazně vyšší než hodnota  $(n+1)!$ , což způsobí, že největší absolutní chyba nebude konvergovat k nule, ale naopak se začne rychle zvyšovat. Vypočítané hodnoty derivací u Rungeovy funkce můžeme případně najít v [2] na str. 21.

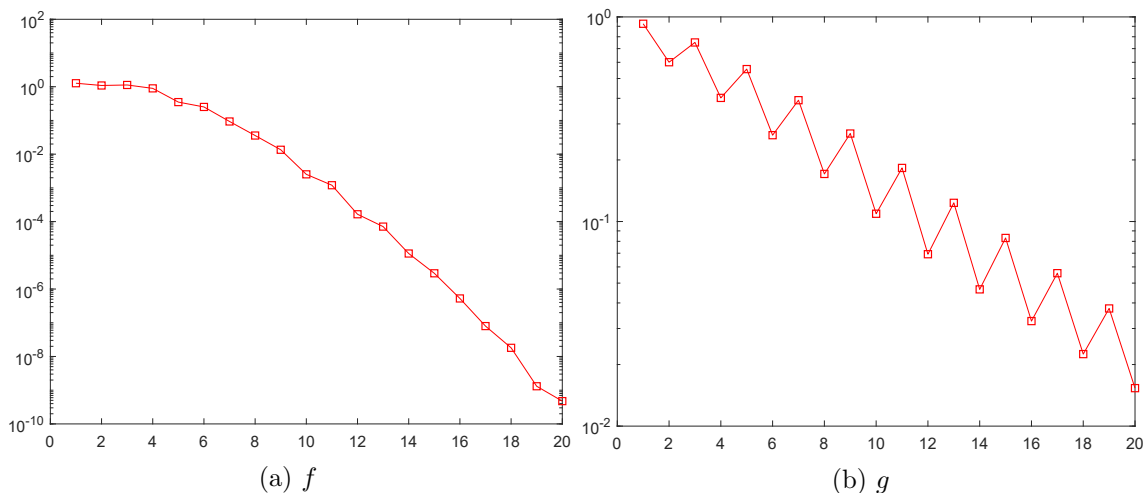
Nyní si vykreslíme grafy, které budou znázorňovat chybu jednotlivých interpolací funkcí  $f$  a  $g$ . Vodorovná osa v grafu představuje stupeň interpolačního polynomu. Hodnotu stupně polynomů volíme  $n = 1, 2, \dots, 20$ . Počet uzlů je tedy pro daný stupeň  $n$  polynomu roven  $n + 1$ . Svislá osa pak představuje největší absolutní chybu, která je daná pomocí

$$\max_{x \in (-1,1)} |p_n(x) - h(x)|,$$

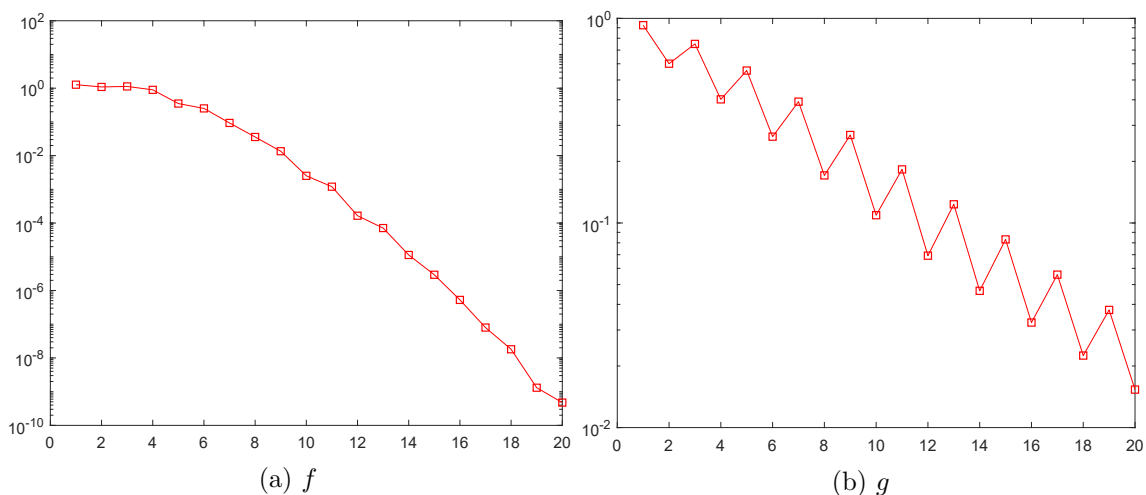
kde  $h$  je daná interpolovaná funkce (v našem případě zadaná funkce  $f$  nebo  $g$ ).



Obrázek 26: Chyba pro Lagrangeovu interpolaci na uniformním dělení intervalu



Obrázek 27: Chyba pro Lagrangeovu interpolaci v Čebyševových uzlech



Obrázek 28: Chyba pro Čebyševovu aproximaci

U grafů chyb si můžeme všimnout, že pro funkci  $f$  jsou všechny tři způsoby interpolací dobré, protože chyby konvergují k nule. Dále pozorujeme, že Lagrangeova interpolace na ekvidistantní síti bodů má téměř vždy vyšší chybu než zbývající způsoby. U Rungeovy funkce se Lagrangeova interpolace v ekvidistantní síti nevyplácí, protože chyba nekongruje k nule. U zbývajících případů nám chyba opět konverguje k nule, a proto můžeme říct, že jsou tyto interpolace dobré. Je zřejmé, že obr. 22 a 24 jsou totožné, stejně jako obr. 23 a 25 a obr. 27 a 28. Vyplyvá to z jednoznačnosti interpolačního polynomu (viz Větu 3.1).

### 5.3 Experimenty pro Remezův algoritmus

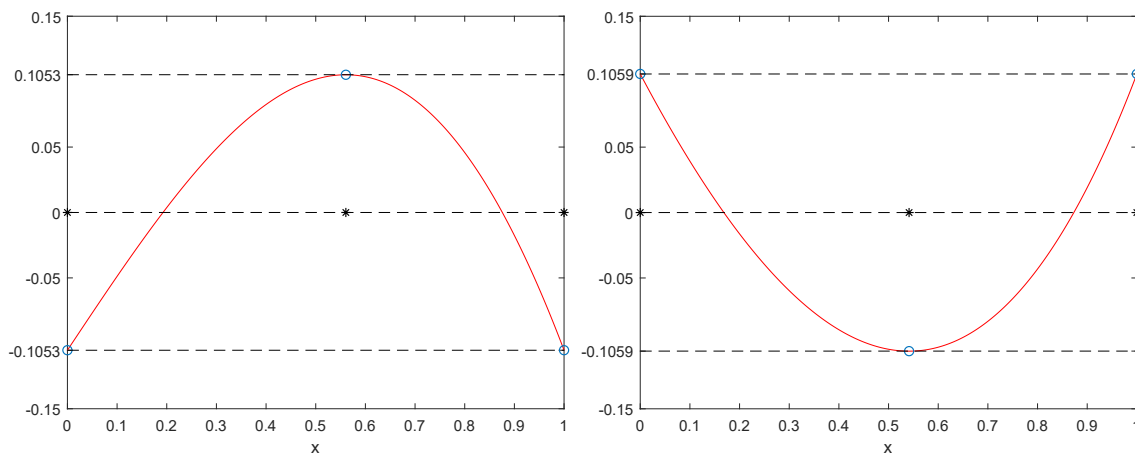
V této kapitole budeme vytvářet experimenty pro Remezův algoritmus, který jsme si naprogramovali v programu MATLAB (viz appendix A). V celé kapitole používáme systém  $\{1, x, x^2, \dots, x^n\}$  (proměnná  $n$  značí stupeň polynomiální aproximace), který splňuje Haarovu podmínku (28). Dále pro metodu brute-force volíme 10000 bodů. V následujících experimentech jsme se často inspirovali prací [4].

### 5.3.1 Srovnání Remezova algoritmu s analytickými výsledky

Nejdříve srovnáme nejlepší lineární aproximace získané Remezovým algoritmem s lineárními nejlepšími aproximacemi vypočtenými analyticky. Experiment provádíme pro tři funkce:  $\sin(\frac{1}{2}\pi x)$ ,  $e^x$  a  $\sqrt{x}$  na intervalu  $\langle 0, 1 \rangle$ , jak lze vidět v tabulce níže. Proměnná  $k$  označuje počet iterací Remezova algoritmu. Polynom  $p_*^r$  je nejlepší lineární aproximace získaná Remezovým algoritmem a polynom  $p_*^a$  je analyticky vypočítaná nejlepší lineární stejnoměrná aproximace. Zdroj pak uvádí místo, odkud čerpáme analytický výpočet. Iniciální referenci jsme volili pro všechny tři funkce uniformně.

V prvních dvou případech jsou hodnoty  $\|f - p_*^r\|_\infty$  a  $\|f - p_*^a\|_\infty$  zcela totožné pro dvanáct desetinných míst. U třetí funkce vznikl rozdíl u těchto hodnot, ale je velmi nepatrný. Můžeme tedy říct, že v těchto případech je kvalita naší implementace Remezova algoritmu ve srovnání s analyticky vypočtenou nejlepší stejnoměrnou aproximací vynikající. Grafy pod tabulkou (viz obr. 29) zobrazují funkce reziduí. Hvězdičky symbolizují body, kde absolutní hodnota rezidua nabývá maximální hodnoty. Modrá kolečka pak značí hodnotu rezidua v daných bodech, tj. vzdálenost mezi hvězdičkou a modrým kolečkem na stejné  $x$ -ové souřadnici představuje hodnotu  $\|f - p_*^r\|_\infty$ .

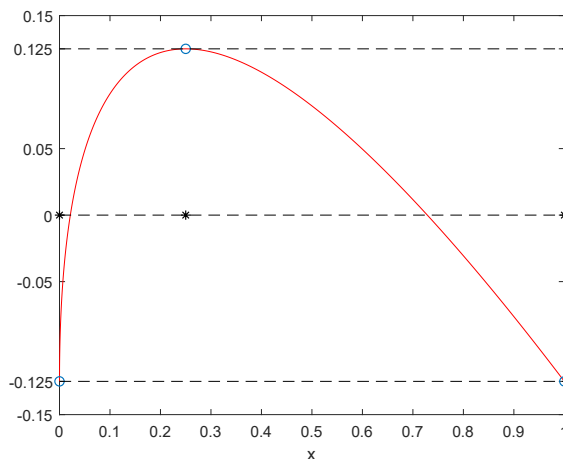
$f(x)$	$k$	$\ f - p_*^r\ _\infty$	$\ f - p_*^a\ _\infty$	zdroj
$\sin(\frac{1}{2}\pi x)$	2	0,105256831176	0,105256831176	[6], str. 76
$e^x$	2	0,105933416258	0,105933416258	[4], str. 16, př. 3
$\sqrt{x}$	2	0,124999999944	0,125	Příklad 4.2



(a) Reziduum  $\sin(\frac{1}{2}\pi x) - p_*^r(x)$

(b) Reziduum  $e^x - p_*^r(x)$

Obrázek 29: Funkce reziduí na intervalu  $\langle 0, 1 \rangle$



(c) Reziduum  $\sqrt{x} - p_*(x)$

Obrázek 29: Funkce reziduí na intervalu  $\langle 0, 1 \rangle$

### 5.3.2 Chování Remezova algoritmu pro zvyšující se stupeň aproximačního polynomu

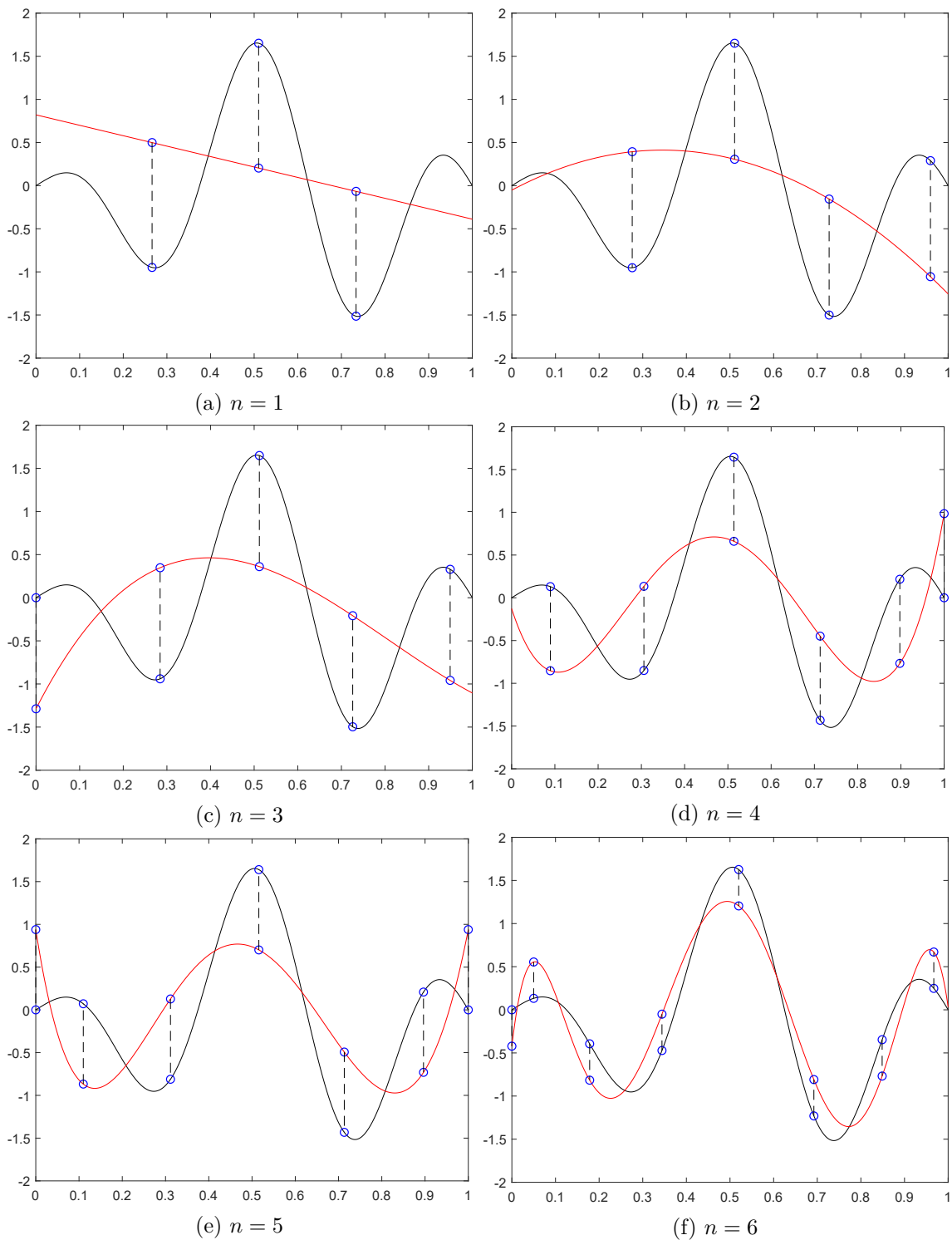
Pro následující numerický experiment jsme si vybrali funkci

$$f(x) = e^x \cos(4\pi x) \sin(\pi x), \quad x \in \langle 0, 1 \rangle.$$

Počáteční reference byla zvolena uniformně. S naší implementací Remezova algoritmu jsme získali aproximační polynomy  $p_*$  stupně  $n = 1, 2, \dots, 18$  v intervalu  $\langle 0, 1 \rangle$ . V tabulce níže máme opět zapsaný počet iterací Remezova algoritmu. Pro polynomiální aproximace stupňů  $n = 1, 2, \dots, 13$  byl algoritmus zastaven pro  $\delta = 10^{-5}$ . Pro stupeň  $n = 14$  jsme použili jako ukončovací podmínku  $\delta = 10^{-6}$ . U stupňů  $n = 15$  a  $n = 16$  byla ukončující podmínka zvolena jako  $\delta = 10^{-7}$ . Zbývající stupně pak používají podmínku zvolenou  $\delta = 10^{-8}$ .

$n$	$\ f - p_*\ _\infty$	$k$	$n$	$\ f - p_*\ _\infty$	$k$
1	1,447353590178816	3	10	$2,4120189680773 \cdot 10^{-2}$	5
2	1,344953656199910	4	11	$1,2157227816527 \cdot 10^{-2}$	5
3	1,287991029661326	3	12	$3,209394605699 \cdot 10^{-3}$	5
4	0,984090480830460	5	13	$1,101413521487 \cdot 10^{-3}$	5
5	0,939110805629159	3	14	$3,036285177537845 \cdot 10^{-4}$	5
6	0,421470766362252	4	15	$7,041831581400260 \cdot 10^{-5}$	6
7	0,393031382928451	4	16	$2,121731103332802 \cdot 10^{-5}$	5
8	0,124339481980745	5	17	$3,309174955258565 \cdot 10^{-6}$	6
9	$8,8971218124033 \cdot 10^{-2}$	5	18	$1,133916152820658 \cdot 10^{-6}$	6

Grafy polynomiálních aproximací stupně  $n = 1, 2, \dots, 6$  si pak můžeme prohlédnout v obr. 30. Černou barvou je vykreslená funkce  $f$ , červenou pak aproximace daného stupně vytvořená Remezovým algoritmem. Svislé přerušované čáry pak označují maximální velikost reziduí  $|r_{max}|$ . Modrá kolečka pak hodnoty aproximace či funkce ve výsledné referenci, a zároveň se jedná o krajní body úseček, které vyjadřují vzdálenost reziduí.



Obrázek 30: Nejlepší polynomiální aproximace na funkci  $f(x) = e^x \cos(4\pi x) \sin(\pi x)$  na intervalu  $(0, 1)$

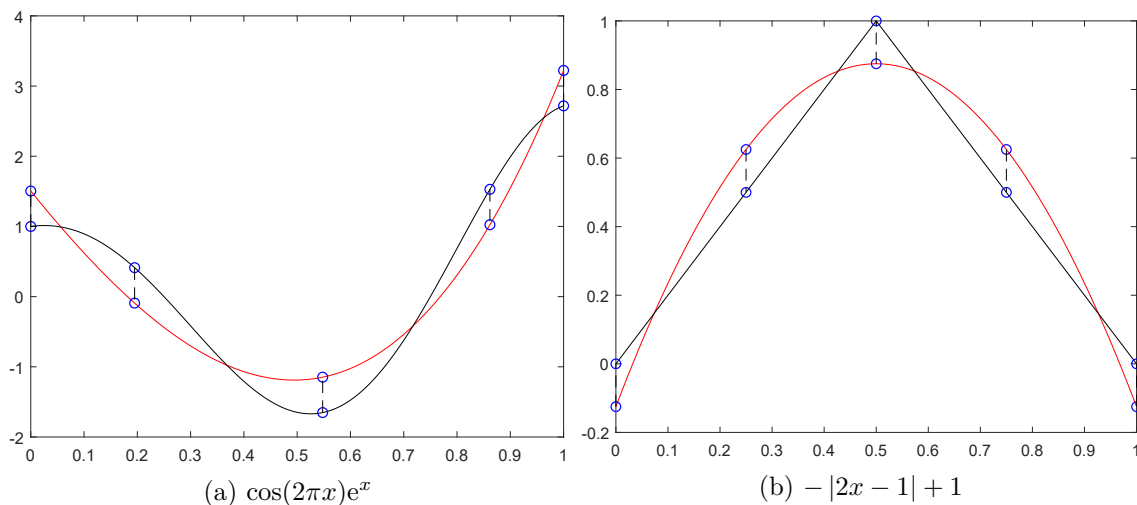
### 5.3.3 Nejlepší polynomiální aproximace 3. stupně pro vybrané funkce

V dalším experimentu aplikujeme Remezův algoritmus na šest vybraných funkcí, abychom získali aproximační polynom třetího stupně na intervalu  $(0, 1)$ . V tabulce níže pak srovnáváme aproximaci  $p_*^r$  získanou pomocí Remezova algoritmu (vstupní reference zvolena

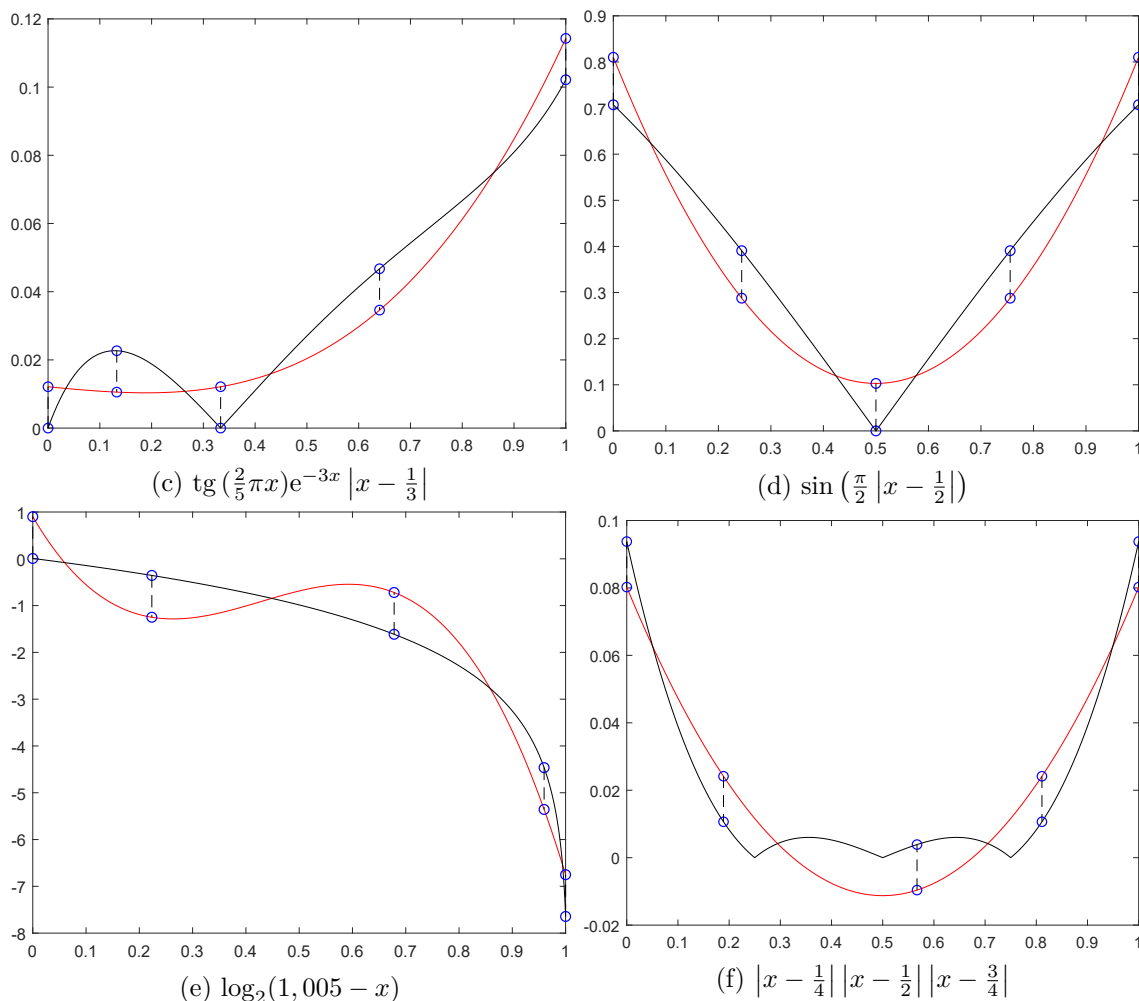
pomocí Čebyševových uzlů) s Lagrangeovou interpolací v Čebyševových uzlech  $p^\ell$  a aproximací  $p_0^r$  získanou pomocí Remezova algoritmu po 1. iteraci, kde vstupní reference je rovna Čebyševovým uzlům v  $(0, 1)$ . Pro ukončující podmínku volíme konstantu  $\delta = 10^{-4}$ .

$f(x)$	$\ f - p^\ell\ _\infty$	$\ f - p_0^r\ _\infty$	$\ f - p_*^r\ _\infty$
$\cos(2\pi x)e^x$	0,649081562484740	0,812537929854424	0,504551055557766
$- 2x - 1  + 1$	0,270498047727298	0,163719218383164	0,125000000000000
$\operatorname{tg}\left(\frac{2}{5}\pi x\right)e^{-3x} \left x - \frac{1}{3}\right $	0,019914308741658	0,019975761455811	0,012132081717964
$\sin\left(\frac{\pi}{2}\left x - \frac{1}{2}\right \right)$	0,219855433778437	0,136985493728641	0,103005200089895
$\log_2(1,005 - x)$	2,482016347671515	2,239172174714533	0,893358670386677
$\left x - \frac{1}{4}\right  \left x - \frac{1}{2}\right  \left x - \frac{3}{4}\right $	0,016326984706599	0,019078356970635	0,013539969092923

Můžeme pozorovat, že naše implementace Remezova algoritmu je pro získání polynomiálních aproximací u všech vybraných funkcí, co se týče kvality, výhodnější než Lagrangeova interpolace v Čebyševových uzlech. Pomocí třetího a čtvrtého sloupce tabulky srovnáme, jak Remezův algoritmus vylepší počáteční aproximaci. Počet iterací Remezova algoritmu byl pro první čtyři funkce roven 3, pro 5. funkci byly provedeny 4 iterace a pro 6. funkci byly provedeny 2 iterace. Na obr. 31 pak můžeme pozorovat jednotlivé funkce a polynomiální aproximace třetího stupně k nim vytvořené. Černou barvou je opět označena funkce, červenou pak polynomiální aproximace. Přerušované čáry znovu zobrazují maximální velikost reziduí  $|r_{max}|$ .



Obrázek 31: Nejlepší polynomiální aproximace 3. stupně pro různě zadané funkce



Obrázek 31: Nejlepší polynomiální aproximace 3. stupně pro různé zadané funkce

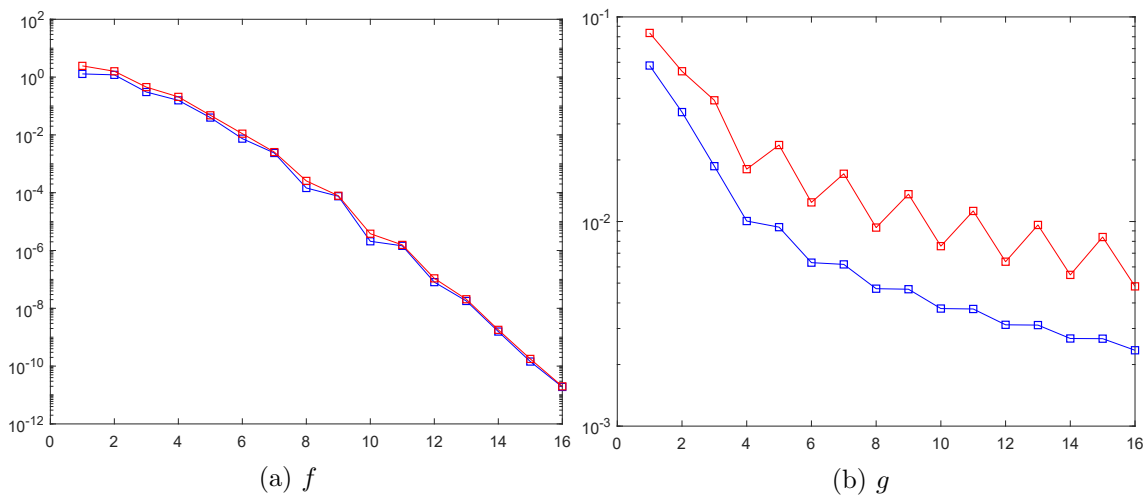
### 5.3.4 Srovnání Remezova algoritmu s Lagrangeovou interpolací v Čebyševových uzlech

Pro tento experiment uvažujme dvě funkce

$$f(x) = \sin(2\pi x)e^x \quad \text{a} \quad g(x) = \operatorname{tg}\left(\frac{2}{5}\pi x\right) e^{-2x} \left|x - \frac{1}{2}\right|, \quad x \in \langle 0, 1 \rangle.$$

Pomocí Remezova algoritmu získáme aproximační polynomy  $p_n^r$  stupně  $n = 1, 2, \dots, 15$  v intervalu  $\langle 0, 1 \rangle$  a dále vytvoříme Lagrangeovy polynomy  $p_n^\ell$  v Čebyševových uzlech pro stejné stupně na stejném intervalu. U Remezova algoritmu jsme volili vstupní referenci jako Čebyševovy uzly. Hodnotu  $\delta$  pro ukončující podmínku jsme volili podobně jako v kapitole 5.3.2. Z počátku je  $\delta = 10^{-4}$  a postupně hodnotu  $\delta$  zmenšujeme tak, aby byla o dva řády menší než hodnota chyby. V grafech na obr. 32 potom srovnáváme chyby Remezova algoritmu (modře) s Lagrangeovou interpolací v Čebyševových uzlech (červeně) pro dané polynomy stupně  $n$ .





Obrázek 32: Srovnání chyby pro Lagrangeovu interpolaci v Čebyševových uzlech s nejlepší polynomiální aproximací

Vidíme, že přestože Remezův algoritmus přibližně počítá nejlepší stejnoměrnou aproximaci dané funkce, poměrně dobře funguje i (Lagrangeova) interpolace v Čebyševových uzlech.

## 6 Závěr

Cílem této bakalářské práce bylo seznámit se s úlohou nalezení nejlepší polynomiální aproximace. Nejdříve jsme si uvedli způsob nalezení polynomiální aproximace pomocí polynomiální interpolace. Tento způsob však obecně neminimalizuje chybu ve všech bodech intervalu. Ve 3. kapitole jsme si dokázali, že výběr uzlů je pro kvalitu interpolace nejdůležitější, a že tzv. optimálními uzly jsou Čebyševovy uzly v intervalu  $\langle -1, 1 \rangle$  (pro obecný interval  $\langle a, b \rangle$  pak stejné uzly vhodně posunuté a naškálované). Důležitost výběru uzlů lze například sledovat u interpolování Rungeovy funkce (viz kapitolu 5.2). V případě ekvidistantní sítě uzlů výrazně roste chyba interpolace se zvyšujícím se stupněm polynomu. Obecně nelze přesně vyřešit úlohu nejlepší aproximace. Řešení najdeme pouze přibližně například pomocí Remezova algoritmu. Tento algoritmus se nám povedlo naimplementovat v programu MATLAB. Nejdříve jsme provedli experiment, ve kterém jsme srovnali analytické výsledky s Remezovým algoritmem. Ukázalo se, že máme algoritmus naimplementovaný správně, protože se nám výsledky shodují. V dalších experimentech jsme si ukázali, že se zvyšujícím se stupněm aproximačního polynomu se snižuje výsledná chyba. Nakonec jsme provedli experimenty, které porovnávaly chybu Remezova algoritmu s Lagrangeovou interpolací v Čebyševových uzlech. Na námi vybraných funkcích se ukázalo, že je aproximace pomocí Remezova algoritmu kvalitnější než Lagrangeova interpolace v Čebyševových uzlech. U některých funkcí stačil dokonce pouze jeden iterační krok algoritmu k tomu, aby byla výsledná aproximace lepší.

Kvůli rozsahu práce jsme se již nevěnovali některým tématům, které by mohly být také součástí této práce. Jelikož obecně je polynomiální interpolace polynomy vysokých stupňů neefektivní, nabízí se polynomiální interpolace po částech s využitím tzv. splinů (více například v [1,2]). Při použití Remezova algoritmu jsme používali systém funkcí spojitých na intervalu  $\langle a, b \rangle$ , který splňuje tzv. Haarovu podmínku. V celé práci jsme se zabývali systémem, který je vytvořen pomocí monomiálů. Příklady dalších systémů, které bychom mohli v práci zahrnout, nalezneme například v [4] na str. 30. Že se jedná o správně volené systémy je pak dále dokázáno v [7] na str. 297. Lloyd Nicholas Trefethen a jeho kolegové z Oxfordské univerzity vytvořili balíček funkcí Chebfun (k nalezení na stránce <https://www.chebfun.org/>) do programu Matlab. Chebfun obsahuje různé metody, které se zabývají řešením polynomiální aproximace. V práci bychom mohli porovnat chybu nejlepší polynomiální aproximace vytvořenou pomocí naší implementace Remezova algoritmu a metodou z balíčku Chebfun. Více o balíčku lze dále nalézt v knize [8].

## A Kód Remezova algoritmu v Matlabu

V této kapitole si popíšeme Remezův algoritmus vytvořený v programu Matlab. Algoritmus spustíme pomocí funkce **Remez.m**. Zadáme ji funkci a interval, ve kterém bude program zadanou funkci aproximovat, dále systém spojitých funkcí na intervalu splňující Haarovu podmínku, počáteční referenci a hodnotu delta, která bude sloužit jako ukončující podmínka. Následujících šest funkcí nám potom vytvoří všechno potřebné k tomu, aby mohla funkce **Remez.m** vrátit vytvořenou aproximaci, vypočítanou normu, počet celkových iterací a vektor nové reference, který obsahuje body, ve kterých je maximální odchylka mezi funkcí a aproximací. Následující kódy funkcí jsou seřazeny tak, jak jsou postupně spouštěny.

### Spouštěcí funkce Remez.m

```
1 function [Approx, Norm, Iterations, NewRef] = Remez(a, b, f,
    Haar, reference, delta)
2
3 Condition = delta + 1;
4 Iterations = 0;
5
6 %KROK 1 + ukončující podmínka:
7 while Condition > delta
8     [Condition, Residue, Approx, positionOfResidueMax, NewRef] =
        RemezAlgorithm(a, b, f, Haar, reference);
9     reference = NewRef;
10    Iterations = Iterations + 1;
11 end
12
13 Norm = abs(Residue(positionOfResidueMax));
14 end
```

### Spouštění algoritmu RemezAlgorithm.m

```
1 function [Condition, Residue, Approx, positionOfResidueMax,
    NewRef] = RemezAlgorithm(a, b, f, Haar, reference)
2 %KROK 1:
3 SLR = solutionOfSLR(f, Haar, reference);
4
5 Approx = @(x) Haar(x)*SLR;
6 Residue = @(x) f(x) - Approx(x);
7
8 [ResidueMax, positionOfResidueMax] = bruteForce(a, b, @(x) abs(
    Residue(x)));
9
10 Condition = abs(ResidueMax - abs(Residue(reference(1))));
11 degree = length(reference) - 1;
12 %KROK 2:
13 rRoots = residueRoots(Residue, reference, degree, a, b);
14 %KROK 3:
15 NewRef = newReference(a, b, degree, rRoots, reference, Residue,
    positionOfResidueMax);
16 end
```

### Vyřešení SLR `solutionOfSLR.m`

```
1 function SLR = solutionOfSLR(f, Haar, reference)
2
3 degree = length(reference) - 1;
4
5 vector2 = Haar(reference(1));
6 vector3 = f(reference(1));
7
8 c = zeros(1, degree);
9 A = zeros(degree);
10
11 for i = 1:degree
12     c(i) = f(reference(i + 1)) - power(-1,i)*vector3;
13     A(i, :) = Haar(reference(i + 1)) - power(-1,i)*vector2;
14 end
15
16 SLR = A\c';
17 end
```

### Hledání maxima funkce pomocí brute-force `bruteForce.m`

```
1 function [value, positionOfValue] = bruteForce(a, b, f)
2
3 xx = linspace(a, b, 10000)';
4 [value, position] = max(f(xx));
5 positionOfValue = xx(position);
6 end
```

### Nalezení kořenů rezidua `residueRoots.m`

```
1 function rRoots = residueRoots(residue, reference, degree, a, b)
2
3 rRoots = zeros(1, degree + 2);
4
5 for i = 2:(degree + 1)
6     rRoots(i) = fzero(residue, [reference(i - 1) reference(i)]);
7 end
8
9 rRoots(1) = a;
10 rRoots(degree + 2) = b;
11 end
```

### Vytvoření nové reference uzlů `newReference.m`

```
1 function NewRef = newReference(a, b, degree, rRoots, reference,
2     Residue, positionOfResidueMax)
3
4 NewRef = zeros(1, degree + 1);
5
6 for i = 1:(degree + 1)
7     sigma = sign(Residue(reference(i)));
8     NewRef(i) = fmaxsigmar(sigma, Residue, rRoots(i), rRoots(i +
9         1), reference(i), a, b);
```

```

8 end
9 %KROK 4:
10 maxR = max(abs(Residue(NewRef')));
11
12 if abs(Residue(positionOfResidueMax)) > maxR
13     if positionOfResidueMax < NewRef(1)
14         if sign(Residue(positionOfResidueMax)) ~= sign(Residue(
15             NewRef(1)))
16             NewRef(2:degree+1) = NewRef(1:degree);
17         end
18         NewRef(1) = positionOfResidueMax;
19
20     elseif NewRef(degree + 1) < positionOfResidueMax
21         if sign(Residue(positionOfResidueMax)) ~= sign(Residue(
22             NewRef(degree + 1)))
23             NewRef(1:degree) = NewRef(2:degree + 1);
24         end
25         NewRef(degree + 1) = positionOfResidueMax;
26
27     else
28         for i = 2:degree+1
29             if NewRef(i - 1) < positionOfResidueMax &&
30                 positionOfResidueMax < NewRef(i)
31                 if sign(Residue(positionOfResidueMax)) == sign(
32                     Residue(NewRef(i - 1)))
33                     NewRef(i - 1) = positionOfResidueMax;
34                 else
35                     NewRef(i) = positionOfResidueMax;
36                 end
37             end
38             break
39         end
40     end
41 end
42 end
43 end
44 end
45 end
46 end
47 end

```

#### Hledání lokálního maxima mezi kořeny rezidua fmaxsigmar.m

```

1 function positionOfMaxSigma = fmaxsigmar(sigma, Residue, lb, ub,
2     xi, a, b)
3 f = @(x) (-1)*sigma*Residue(x);
4 positionOfMaxSigma = fminbnd(f, lb, ub);
5
6 if sigma*Residue(xi) > sigma*Residue(positionOfMaxSigma)
7     [maxSigma, positionOfMaxSigma] = bruteForce(lb, ub, @(x)(
8         sigma*Residue(x)));
9 end

```

```
10 if sigma*Residue(xi) > sigma*Residue(positionOfMaxSigma)
11     positionOfMaxSigma = xi;
12 end
13
14 if lb == a && sigma*Residue(a) > sigma*Residue(
15     positionOfMaxSigma)
16     positionOfMaxSigma = a;
17 end
18 if ub == b && sigma*Residue(b) > sigma*Residue(
19     positionOfMaxSigma)
20     positionOfMaxSigma = b;
21 end
```

## Reference

Všechny on-line odkazy jsou referovány k datu odevzdání této práce.

- [1] W. Shen, *An Introduction To Numerical Computation*. World Scientific, 2019.
- [2] M. Embree, “Lecture notes on numerical analysis (for math 5466 at Virginia Tech).” <http://intranet.math.vt.edu/people/embree/math5466/nanotes.pdf>, 2016.
- [3] E. Süli and D. F. Mayers, *An introduction to numerical analysis*. Cambridge University Press, 2003.
- [4] E. de Groot, “Finding best minimax approximations with the Remez algorithm,” Bc project, Faculty of Science and Engineering, 2017.
- [5] M. J. D. Powell, *Approximation theory and methods*. Cambridge University Press, 1981.
- [6] E. W. Cheney, *Approximation Theory*. Chelsea Publishing Company, 1982. Druhé vydání.
- [7] C. B. Dunham, “Families satisfying the Haar condition,” *Journal of Approximation Theory*, vol. 12, no. 3, pp. 291–298, 1974.
- [8] L. N. Trefethen, *Approximation theory and approximation practice*. Siam, 2019.
- [9] K. Rektorys, *Přehled užití matematiky II*. Prometheus, 2003.
- [10] D. Lukáš, “Přednášky k předmětu numerické metody 1.” <https://home1.vsb.cz/~luk76/Teaching/NM1/Prednasky>, 2018.
- [11] S. Viswanath, “The Chebyshev polynomials,” 2017. <https://youtu.be/JpBb-muc2N0>.
- [12] J. Limpouch, “Čebyševovy aproximace.” <http://www-troja.fjfi.cvut.cz/~limpouch/numet/aprox/node14.html>, 2000.