MASARYKOVA UNIVERZITA
Přírodovědecká fakulta
Ústav matematiky a statistiky

Bakalářská práce

Brno 2025 Šárka Svobodová

MASARYKOVA
UNIVERZITA
Přírodovědecká fakulta
Ústav matematiky a statistiky

Nepolynomiální interpolace

Bakalářská práce

Šárka Svobodová

Vedoucí práce: Mgr. Jiří Zelinka, Dr. Brno 2025

Bibliografický záznam

Autor:	Šárka Svobodová Přírodovědecká fakulta, Masarykova univerzita Ústav matematiky a statistiky
Název práce:	Nepolynomiální interpolace
Studijní program:	Matematika
Studijní obor:	Matematika - hlavní, Ekonomie - vedlejší
Vedoucí práce:	Mgr. Jiří Zelinka, Dr.
Akademický rok:	2024/2025
Počet stran:	xiii + 37
Klíčová slova:	nepolynomiální interpolace; trigonometrická interpolace; racionální interpolace; exponenciální interpolace; aproximace; MATLAB; metoda nejmenších čtverců; numerická matematika

Bibliographic Entry

Author:	Šárka Svobodová Faculty of Science, Masaryk University Department of Mathematics and Statistics
Title of Thesis:	Non-polynomial interpolation
Degree Programme:	Mathematics
Field of Study:	Mathematics - major, Economics - minor
Supervisor:	Mgr. Jiří Zelinka, Dr.
Academic Year:	2024/2025
Number of Pages:	xiii + 37
Keywords:	non-polynomial interpolation; trigonometric interpolation; rational interpolation; exponential interpolation; approximation; MATLAB; least squares method; numerical mathematics

Abstrakt

V této bakalářské práci se věnujeme nepolynomiální interpolaci, konkrétně interpolaci trigonometrické, racionální a exponenciální. Na závěr se podíváme na to, jak lze najít interpolační funkci s různými bázovými funkcemi. K interpolaci přistupujeme metodou nejmenších čtverců. Práce je doplněna implementací vybraných metod v prostředí MATLAB a také ukázkovými příklady.

Abstract

In this thesis we study non-polynomial interpolation, specifically trigonometric, rational, and exponential interpolation. In the final part, we explore how to construct an interpolation function using different basis functions. The interpolation is approached using the method of least squares. The thesis includes implementations of the selected methods in MATLAB, along with illustrative examples.

MASARYKOVA UNIVERZITA
PŘÍRODOVĚDECKÁ FAKULTA
KOTLÁŘSKÁ 2, 611 37 BRNO
IČ: 00216224
DIČ: CZ00216224

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Akademický rok: 2024/2025

Ústav:	Ústav matematiky a statistiky
Studentka:	Šárka Svobodová
Program:	Matematika
Specializace:	Ekonomie Matematika

Ředitel *ústavu* PřF MU Vám ve smyslu Studijního a zkušebního řádu MU určuje bakalářskou práci s názvem:

Název práce:	Nepolynomiální interpolace
Název práce anglicky:	Non-polynomial interpolation
Jazyk práce:	čeština

Oficiální zadání:
V práci se zaměřte na způsoby interpolace dat jinými funkcemi než polynomy, uveďte vlastnosti a aplikace takových aproximací a vhodné příklady. Práce může být doplněna programy, které prakticky realizují prezentované výsledky.

Literatura:
RALSTON, Anthony. *Základy numerické matematiky*. Translated by Milan Práger - Emil Vitásek. České vyd. 2. Praha: Academia, 1978, 635 s.

J. Barzilai and A. Ben-Tal, Nonpolynomial and Inverse Interpolation for Line Search: Synthesis and Convergence Rates, SIAM Journal on Numerical Analysis, vol. 19, no. 6, 1982, pp. 1263–77

Philip J. DAVIS, Interpolation and Approximation, Dover Publications, 1963

Ian H. Sloan, Nonpolynomial interpolation, Journal of Approximation Theory, Volume 39, Issue 2, 1983, Pages 97-117, ISSN 0021-9045

Vedoucí práce:	Mgr. Jiří Zelinka, Dr.
Datum zadání práce:	3. 9. 2024
V Brně dne:	5. 11. 2024

Zadání bylo schváleno prostřednictvím IS MU.

Šárka Svobodová, 25. 10. 2024
Mgr. Jiří Zelinka, Dr., 25. 10. 2024
RNDr. Jan Vondra, Ph.D., 25. 10. 2024

Poděkování

Na tomto místě bych chtěla poděkovat vedoucímu své bakalářské práce Mgr. Jiřímu Zelinkovi, Dr. za odborné vedení, vstřícný přístup a cenné připomínky, jež mi pomohly při vypracování této práce.

Prohlášení

Prohlašuji, že jsem svoji bakalářskou práci vypracovala samostatně pod vedením vedoucího práce s využitím informačních zdrojů, které jsou v práci citovány.

Brno 3. května 2025 . . . . . .

Šárka Svobodová

Obsah

Úvod **1**

Kapitola 1. Polynomiální interpolace **3**
1.1 Lagrangeova interpolace 3
1.2 Newtonova interpolace 4
1.3 Hermitova interpolace 6

Kapitola 2. Trigonometrická interpolace **7**
2.1 Interpolace na ekvidistantních uzlech 7
2.2 Interpolace na nerovnoměrně rozložených uzlech 15

Kapitola 3. Exponenciální interpolace **21**

Kapitola 4. Racionální interpolace **29**

Kapitola 5. Interpolace různými bázovými funkcemi **33**

Závěr **35**

Seznam použité literatury **37**

Úvod

Interpolace je metoda, která slouží k odhadu hodnot funkce mezi známými body. Můžeme ji využít například při zpracování experimentálních dat či k modelování, kdy je potřeba rekonstruovat funkci, jejíž hodnota je známa pouze v několika bodech. Nejpoužívanější metodou je polynomiální interpolace. Ta však nemusí vždy vhodně zachytit strukturu dat, navíc využití polynomů vysokých stupňů může vést k oscilacím. Z těchto důvodů je nutné hledat jiné funkce pro interpolaci, které lépe odpovídají specifickým charakteristikám dat. Ve své práci se tak chci zaměřit na to, jaké další funkce můžeme k interpolaci využívat.

Cílem práce je představit vybrané metody interpolace a ukázat možné postupy řešení na příkladech. V práci bude krátce ukázána polynomiální interpolace a dále již metody nepolynomiální interpolace, jako je trigonometrická, exponenciální a racionální interpolace. V závěru práce bude ukázáno, jak je možné tyto metody kombinovat. Tato práce si klade za cíl ukázat, že pro interpolaci je vždy nutné zvolit správnou funkci, případně kombinaci funkcí, pro dosažení optimálního výsledku, který bude odpovídat struktuře a charakteristice dat.

Interpolace je numerickým problémem, který je výhodnější řešit pomocí specializovaného softwaru. Součástí práce tak budou příslušné funkce v prostředí MATLAB, díky kterým lze proces zefektivnit a zautomatizovat. Software zároveň umožňuje snadné generování grafů pro vizualizaci a ověření správnosti výsledků.

Kapitola 1

Polynomiální interpolace

Polynomiální interpolací rozumíme metodu, při které se snažíme najít přibližné hodnoty mezi zadanými daty pomocí polynomu. Zadanou množinou uzlů $x_0, x_1, x_2, \ldots, x_n$ a funkčních hodnot v nich $f_{x_0}, f_{x_1}, f_{x_2}, \ldots, f_{x_n}$ se snažíme proložit polynomem stupně n. Pro každý index $i = 0, \ldots, n$ tedy musí platit: $f_{x_i} = f(x_i)$.

Přímočarým řešením tohoto problému je sestavení systému lineárních rovnic, ve kterém jsou neznámými koeficienty hledaného polynomu. Máme-li zadaných $n+1$ uzlů, hledaný polynom bude stupně n a můžeme jej zapsat ve tvaru:

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

Po dosazení zadaných hodnot získáváme soustavu rovnic, která má jednoznačné řešení za předpokladu, že uzly jsou různé. Tento způsob je však při vyšším počtu uzlů výpočetně náročný a zároveň nás nutí při jakékoliv změně v zadaných datech začít celý výpočet znovu.

1.1 Lagrangeova interpolace

Základem Lagrangeova interpolačního polynomu jsou fundamentální polynomy $l_i(x)$. Tyto polynomy získáme následovně:

$$l_i(x) = \prod_{j \neq i} \frac{(x - x_j)}{(x_i - x_j)}$$

Každý fundamentální polynom l_i má v bodě x_i funkční hodnotu 1 a v ostatních bodech x_j, kde $j \neq i$, hodnotu 0. Vynásobením daného polynomu $l_i(x)$ zadanou funkční hodnotou f_{x_i} tak zajistíme průchod požadovaným bodem. Výsledný Lagrangeův interpolační polynom má tedy tvar:

$$P(x) = \sum_{i=0}^{n} f_{x_i} l_i(x)$$

1.1.1 Příklad

Mějme body dané následující tabulkou, chceme najít Lagrangeův interpolační polynom procházející těmito body.

x_i	1	2	3	4
f_{x_i}	4	3	-2	1

Nejprve si spočteme fundamentální polynomy:

$$l_0(x) = \frac{(x-2)(x-3)(x-4)}{(1-2)(1-3)(1-4)} = -\frac{x^3-9x^2+26x-24}{6}$$
$$l_1(x) = \frac{(x-1)(x-3)(x-4)}{(2-1)(2-3)(2-4)} = \frac{x^3-8x^2+19x-12}{2}$$
$$l_2(x) = \frac{(x-1)(x-2)(x-4)}{(3-1)(3-2)(3-4)} = -\frac{x^3-7x^2+14x-8}{2}$$
$$l_3(x) = \frac{(x-1)(x-2)(x-3)}{(4-1)(4-2)(4-3)} = \frac{x^3-6x^2+11x-6}{6}$$

Získané fundamentální polynomy dále podle vzorce jednotlivě vynásobíme funkčními hodnotami a sečteme:

$$P(x) = -4\frac{x^3-9x^2+26x-24}{6} + 3\frac{x^3-8x^2+19x-12}{2} + 2\frac{x^3-7x^2+14x-8}{2} +$$
$$+\frac{x^3-6x^2+11x-6}{6} = 2x^3-14x^2+27x-11$$

1.2 Newtonova interpolace

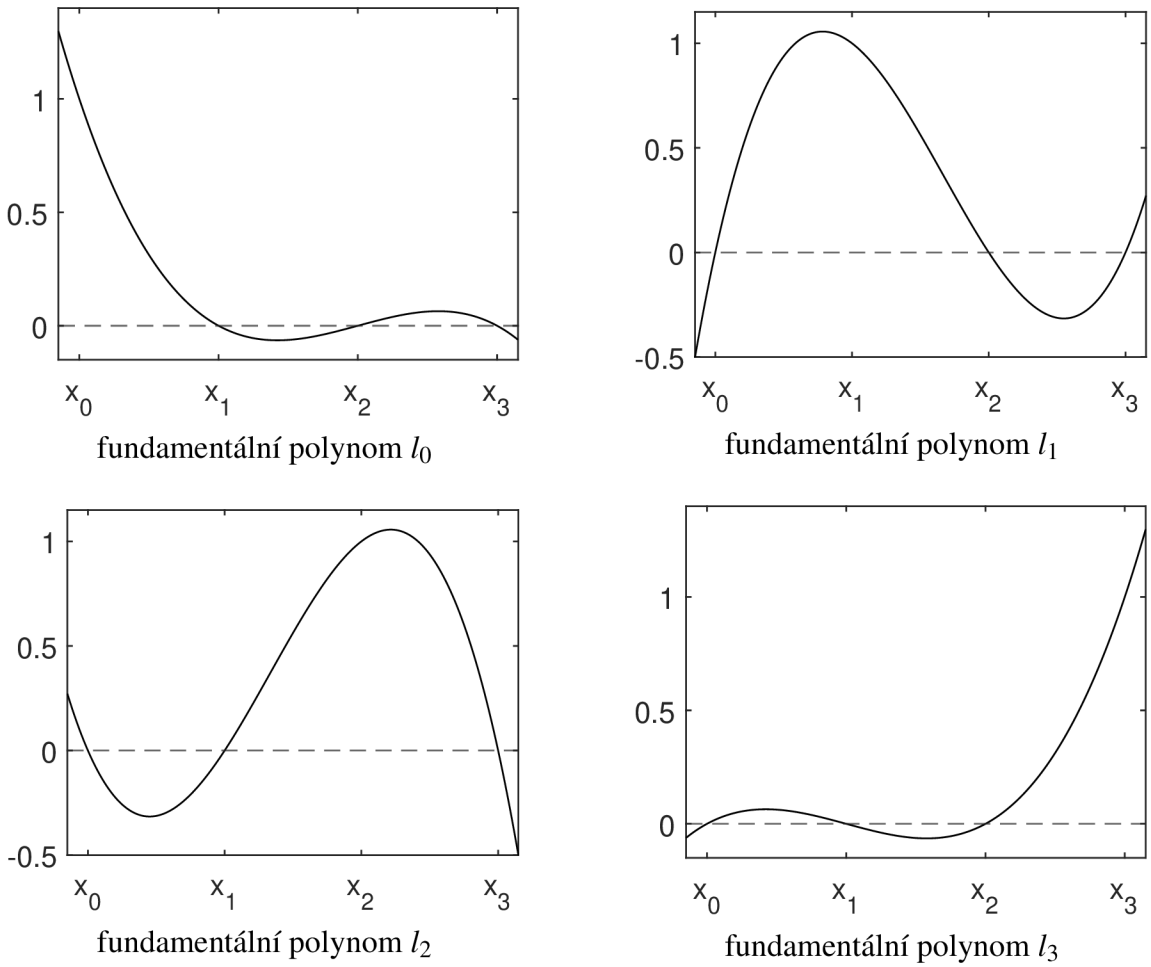
Pokud se nám změní funkční hodnoty, ale uzly zůstávají stejné, můžeme tuto změnu snadno provést při konstrukci Lagrangeova interpolačního polynomu. Fundamentální polynomy totiž zůstávají stejné, pouze je vynásobíme jinými funkčními hodnotami. V případě, že bychom chtěli přidat další uzel, už bychom museli začít celý výpočet znovu. V této situaci je vhodnější počítat Newtonův interpolační polynom.

Newtonův interpolační polynom pracuje s tzv. poměrnými diferencemi, které nám určují změnu funkční hodnoty mezi uzly. Poměrné diference se počítají rekurzivně, k výpočtu poměrných diferencí řádu $i+1$ využíváme poměrné diference i-tého řádu. To nám snadno umožňuje k interpolaci přidat další bod bez toho, abychom museli celý výpočet začít znovu, jako by tomu bylo u Lagrangeova interpolačního polynomu. Poměrné diference získáme následovně:

$$f[x_i] = f(x_i)$$
$$f[x_0,\dots,x_i] = \frac{f[x_{i+1},\dots,x_{i+k}] - f[x_i,\dots,x_{i+k-1}]}{x_{i+k}-x_i}$$

V samotném polynomu je každá poměrná diference násobena výrazem $\prod_{j=0}^{i-1}(x-x_j)$ a všechny jsou dohromady sečteny. Polynom tak můžeme zapsat jako:

$$P(x) = \sum_{i=0}^{n} f[x_0,\dots,x_i]\prod_{j=0}^{i-1}(x-x_j)$$
$$= f_0 + f[x_0,x_1](x-x_0) + \dots + f[x_0,\dots,x_n](x-x_0)\dots(x-x_{n-1})$$



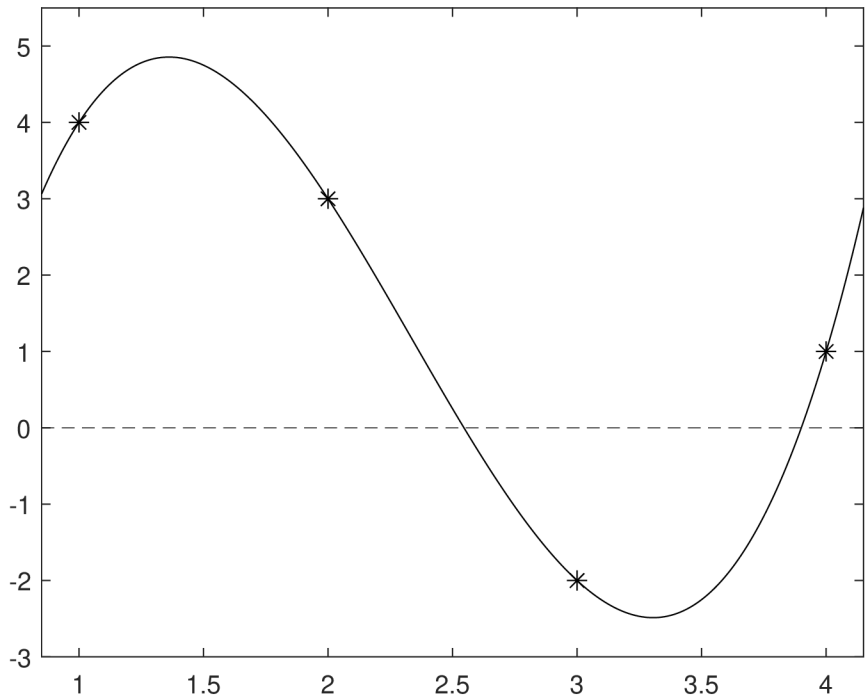
Obrázek 1.1: Fundamentální polynomy polynomu P(x)

1.2.1 Příklad

Využijeme stejné hodnoty jako v předchozím příkladu. Tentokrát však budeme hledat Newtonův interpolační polynom. Spočítáme si tabulku poměrných diferencí, které následně dosadíme do vzorce:

x_i	$f[x_i]$	$f[x_{i-1},x_i]$	$f[x_{i-2},x_{i-1},x_i]$	$f[x_{i-3},x_{i-2},x_{i-1},x_i]$
1	4			
2	3	$\frac{3-4}{2-1}=-1$		
3	-2	$\frac{-2-3}{3-2}=-5$	$\frac{-5-(-1)}{3-1}=-2$	
4	1	$\frac{1-(-2)}{4-3}=3$	$\frac{3-(-5)}{4-2}=2$	$\frac{4-(-2)}{4-1}=2$

$$P(x)=4-(x-1)-2(x-1)(x-2)+2(x-1)(x-2)(x-3)=2x^3-14x^2+27x-11$$



Obrázek 1.2: Polynom P(x)

Hledání polynomu procházejícího zadanou množinou uzlů $x_0, x_1, x_2, \ldots, x_n$ a jejich funkčních hodnot $f_{x_0}, f_{x_1}, f_{x_2}, \ldots, f_{x_n}$ je jednoznačný problém. Všechny tři výše uvedené postupy tak vedou ke stejnému výsledku. Každý způsob však k nalezení interpolačního polynomu využívá jiných bázových polynomů. První způsob, který vedl na soustavu rovnic, využívá standardní bázi prostoru polynomů: $x^n, x^{n-1}, \ldots, x, 1$. V Lagrangeově interpolačním polynomu tvoří bázi fundamentální polynomy. A Newtonův interpolační polynom využívá bázové funkce: $(x-x_0)\ldots(x-x_{n-1}), \ldots, (x-x_0)(x-x_1), (x-x_0), 1$.

1.3 Hermitova interpolace

Hermitova interpolace je rozšířená metoda polynomiální interpolace, která bere v úvahu i hodnoty derivací v zadaných bodech. Označíme-li k_i počet známých derivací v daném uzlu, můžeme počet podmínek, které musí polynom splnit označit následovně:

$$\sum_{i=0}^{n}(k_i+1)$$

Stupeň polynomu tak bude o 1 menší, než je celkový počet podmínek, tedy:

$$\sum_{i=0}^{n}(k_i+1)-1.$$

Podrobnější literaturu o polynomiální interpolaci nabízí například *Numerické metody* od Ivany Horové a Jiřího Zelinky (viz [3]) nebo *Interpolation and Approximation* od Phillipse J. Davise (viz [2]).

Kapitola 2

Trigonometrická interpolace

Trigonometrickou interpolaci provádíme pomocí trigonometrických polynomů, tedy lineární kombinací funkcí sinus a cosinus s různými frekvencemi. Bázi trigonometrické interpolace tvoří funkce: $1, \sin(x), \cos(x), \sin(2x), \cos(2x), \ldots, \sin(nx), \cos(nx)$. Trigonometrická interpolace je vhodná zejména pro periodické funkce nebo periodicky se opakující data.

2.1 Interpolace na ekvidistantních uzlech

Speciálním případem je situace, kdy jsou uzly $x_0, x_1, \ldots, x_{2n}$ rozmístěny rovnoměrně. Pro každý uzel x_i tedy musí platit $x_i = h + x_{i-1}$, kde $h = \frac{x_{2n}-x_0}{2n}$. Richard L. Burden a J. Douglas Faires ve své knize *Numerical Analysis* (viz [1]) zmiňují, že při řešení tohoto problému chceme metodou nejmenších čtverců aproximovat danou funkci s periodou 2π ve tvaru:

$$T_n(x) = \frac{1}{2}a_0 + \sum_{k=1}^{n} a_k \cos(kx) + \sum_{k=1}^{n} b_k \sin(kx)$$

O metodě nejmenších čtverců víme, že pokud máme stejný počet bázových funkcí a uzlů a vektory funčních hodnot jednotlivých bázových funkcí jsou lineárně nezávislé, výsledná aproximace bude procházet požadovanými body, a tedy se bude jednat o interpolaci. Jestliže máme zadáno $2n+1$ uzlů, budeme tedy potřebovat $2n+1$ bázových funkcí. Zvolení metody nejmenších čtverců pro interpolaci nejen trigonometrickými funkcemi je vhodné i díky tomu, že v případě, kdy není možné provést interpolaci, umožňuje najít nejlepší možnou aproximaci dat.

Tento vzorec lze přímo použít pokud máme lichý počet uzlů. Jetliže máme sudý počet uzlů, hodnotu n v situaci, jak ji máme popsanou, není možné použít. Označme tedy uzly $x_0, x_1, \ldots, x_{2n-1}$. Pro jednoznačnost polynomu na $2n$ uzlech však stále potřebujeme $2n$ funkcí. A proto je dle výše uvedené literatury nutné změnit zápis vzorce do následující podoby. Indexu n není možné přiřadit sinus i cosinus, volíme tak pouze cosinus, kterému dáváme poloviční váhu:

$$T_n(x) = \frac{1}{2}a_0 + \sum_{k=1}^{n-1} a_k \cos(kx) + \sum_{k=1}^{n-1} b_k \sin(kx) + \frac{1}{2}a_n cos(nx)$$

Stupeň trigonometrického polynomu n vždy značí největší index u koeficientu a. V literatuře se následně i dozvíme, jak získat koeficienty a_k a b_k:

$$a_k = \frac{2}{2n+1} \sum_{i=0}^{2n} f(x_i) \cos(kx_i)$$

$$b_k = \frac{2}{2n+1} \sum_{i=1}^{2n} f(x_i) \sin(kx_i)$$

V případě ekvidistantních uzlů je báze $1, \sin(x), \cos(x), \ldots, \sin(nx), \cos(nx)$ ortogonální a koeficienty a_k a b_k získáváme snadno pomocí diskrétní Fourierovy transformace.

2.1.1 Příklad

Mějme periodicky se opakující data s periodou 2π. Data jsou dána tabulkou:

x_i	0	$\frac{2\pi}{5}$	$\frac{4\pi}{5}$	$\frac{6\pi}{5}$	$\frac{8\pi}{5}$
f_{x_i}	1	$\frac{1}{2}$	2	-1	0

Z tabulky vidíme, že $2n = 4$, a tedy $n = 2$. Můžeme přejít k výpočtu koeficientů a_0, a_1, b_1, a_2 a b_2.

$$a_0 = \frac{2}{4+1}\left(1 + \frac{1}{2} + 2 - 1 + 0\right) = 1$$

$$a_1 = \frac{2}{4+1}\left[\cos(0) + \frac{1}{2}\cos\left(\frac{2\pi}{5}\right) + 2\cos\left(\frac{4\pi}{5}\right) - \cos\left(\frac{6\pi}{5}\right)\right] \approx 0.138197$$

$$b_1 = \frac{2}{4+1}\left[\sin(0) + \frac{1}{2}\sin\left(\frac{2\pi}{5}\right) + 2\sin\left(\frac{4\pi}{5}\right) - \sin\left(\frac{6\pi}{5}\right)\right] \approx 0.895554$$

$$a_2 = \frac{2}{4+1}\left[\cos(0) + \frac{1}{2}\cos\left(2\cdot\frac{2\pi}{5}\right) + 2\cos\left(2\cdot\frac{4\pi}{5}\right) - \cos\left(2\cdot\frac{6\pi}{5}\right)\right] \approx 0.361803$$

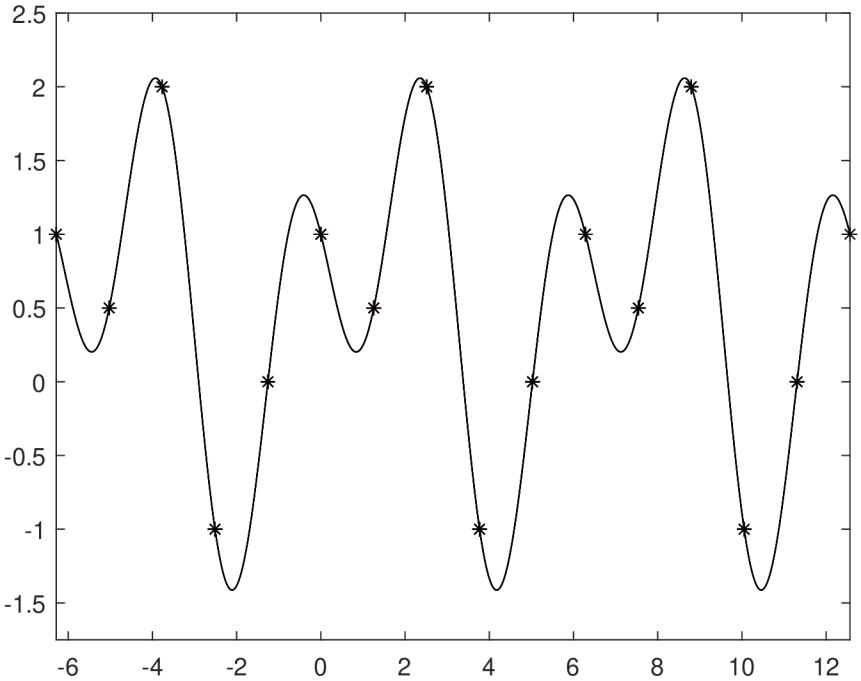
$$b_2 = \frac{2}{4+1}\left[\sin(0) + \frac{1}{2}\sin\left(2\cdot\frac{2\pi}{5}\right) + 2\sin\left(2\cdot\frac{4\pi}{5}\right) - \sin\left(2\cdot\frac{6\pi}{5}\right)\right] \approx -1.023711$$

Koeficienty dosadíme do vzorce a získáme výsledný trigonometrický polynom:

$$T_2(x) = \frac{1}{2} + 0.138197\cos(x) + 0.895554\sin(x) + 0.361803\cos(2x) - 1.023711\sin(2x)$$

Pokud se uzly neopakují s periodou 2π, musíme tento fakt zohlednit v argumentech funkcí sinus a cosinus. Označíme-li L délku periody, můžeme trigonometrický polynom zapsat obecně:

$$T_n(x) = \frac{1}{2}a_0 + \sum_{k=1}^{n} a_k \cos\left(\frac{2\pi kx}{L}\right) + \sum_{k=1}^{n} b_k \sin\left(\frac{2\pi kx}{L}\right)$$



Obrázek 2.1: Trigonometrický polynom $T_2(x)$ z příkladu 2.1.1 na intervalu $[-2\pi, 4\pi]$

$$a_k = \frac{2}{2n+1} \sum_{i=0}^{2n} f(x_i) \cos\left(\frac{2\pi k x_i}{L}\right)$$

$$b_k = \frac{2}{2n+1} \sum_{i=1}^{2n} f(x_i) \sin\left(\frac{2\pi k x_i}{L}\right)$$

Sumu počítáme přes všechny indexy i, v případě, že máme sudý počet bodů, bude horní hranice sumy u výpočtu koeficientů $2n-1$.

2.1.2 Příklad

Podívejme se na výpočet trigonometrického polynomu, když se data opakují s periodou π a máme sudý počet uzlů:

x_i	0	$\frac{\pi}{4}$	$\frac{\pi}{2}$	$\frac{3\pi}{4}$
f_{x_i}	$\frac{3}{4}$	$-\frac{1}{2}$	1	$\frac{5}{4}$

Potřebujeme tedy spočítat koeficienty a_0, a_1, b_1 a a_2:

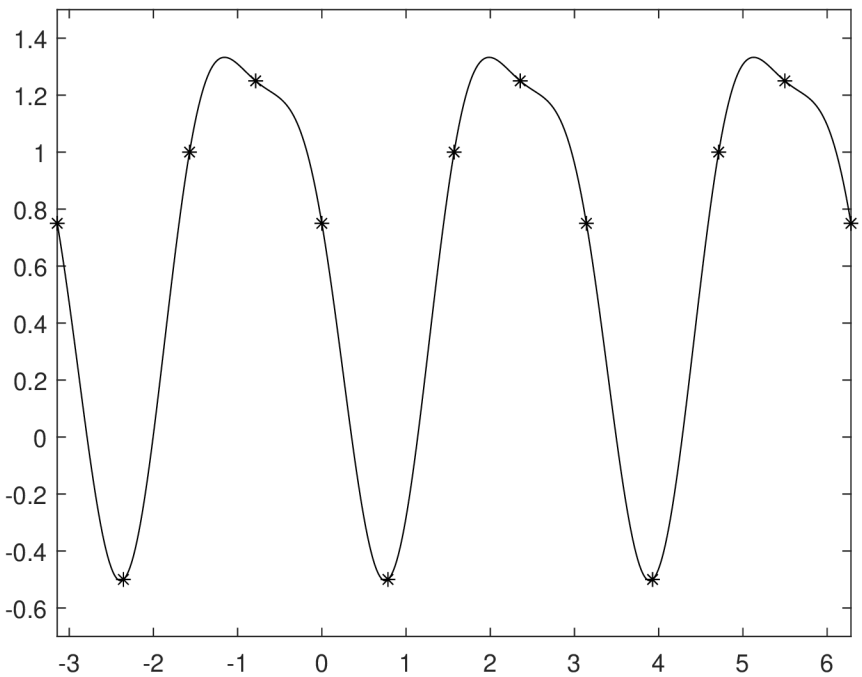
$$a_0 = \frac{2}{3+1}\left(\frac{3}{4} - \frac{1}{2} + 1 + \frac{5}{4}\right) = 1.25$$

$$a_1 = \frac{2}{3+1}\left[\frac{3}{4}\cos(0) - \frac{1}{2}\cos\left(\frac{2\pi\cdot\pi}{\pi\cdot 4}\right) + \cos\left(\frac{2\pi\cdot\pi}{\pi\cdot 2}\right) + \frac{5}{4}\cos\left(\frac{2\pi\cdot 3\pi}{\pi\cdot 4}\right)\right] = -0.125$$

$$b_1 = \frac{2}{3+1}\left[\frac{3}{4}\sin(0) - \frac{1}{2}\sin\left(\frac{2\pi\cdot\pi}{\pi\cdot 4}\right) + \sin\left(\frac{2\pi\cdot\pi}{\pi\cdot 2}\right) + \frac{5}{4}\sin\left(\frac{2\pi\cdot 3\pi}{\pi\cdot 4}\right)\right] = -0.875$$

$$a_2 = \frac{2}{3+1}\left[\frac{3}{4}\cos(0) - \frac{1}{2}\cos\left(\frac{4\pi\cdot\pi}{\pi\cdot 4}\right) + \cos\left(\frac{4\pi\cdot\pi}{\pi\cdot 2}\right) + \frac{5}{4}\cos\left(\frac{4\pi\cdot 3\pi}{\pi\cdot 4}\right)\right] = 0.5$$

$$T_2(x) = \frac{1}{2}a_0 + a_1\cos\left(\frac{2\pi}{\pi}x\right) + b_1\sin\left(\frac{2\pi}{\pi}x\right) + \frac{1}{2}a_2\cos\left(\frac{2\pi\cdot 2}{\pi}x\right) =$$
$$= 0.625 - 0.125\cos(2x) - 0.875\sin(2x) + 0.25\cos(4x)$$



Obrázek 2.2: Trigonometrický polynom $T_2(x)$ z příkladu 2.1.2 na intervalu $[-\pi, 2\pi]$

2.1.3 Funkce v MATLAB

Pro výpočet trigonometrického polynomu T na ekvidistantních uzlech s periodou L lze využít následující kód pro MATLAB. Funkce spočítá všechny koeficienty a_k a b_k, sestaví polynom a následně vykreslí graf. Uzly a jejich hodnoty je nutné zadávat pouze pro jednu periodu, tedy tak, aby se hodnoty na začátku a na konci vektorů neopakovaly. Nezáleží

na tom, zda je počet uzlů lichý nebo sudý, funkce automaticky přidá poloviční váhy odpovídajícím koeficientům.

```
function [T] = trig_int(uzly, hodnoty)
h = diff(uzly);
m = length(uzly);
n = floor(m/2);
if numel(unique(diff(uzly))) ~= 1
    error('Uzly nejsou ekvidistanti!')
end
if length(uzly) ~= length(hodnoty)
    error('Vektory uzlu a hodnot musi byt stejne delky!')
end
L = uzly(end) + h(1); % perioda
a = [];
b = [];
T = 0;
for k = 0:n % vypocet koeficientu
    for i = 1:m
        a = [a,(2/m)*hodnoty(i)*cos((2*pi/L)*k*uzly(i))];
        b = [b,(2/m)*hodnoty(i)*sin((2*pi/L)*k*uzly(i))];
    end
    if mod(m,2) == 0 % sudy pocet uzlu
        if k == 0 || k == n
            a = 0.5*a;
            b = 0;
        end
    else % lichy pocet uzlu
        if k == 0
            a = 0.5*a;
        end
    end
    syms x;
    T = T + sum(a)*cos((2*pi/L)*k*x) + sum(b)*sin((2*pi/L*k*x)
        );
    a = [];
    b = [];
end
fplot(T, [uzly(1), uzly(end)]);
hold on
plot(uzly, hodnoty, '*')
end
```

MATLAB však nabízí vestavěnou funkci `fft`, která počítá rychlou Fourierovu transformaci. Tato funkce nám tak může nabídnout o něco kratší řešení:

```
function [T] = trig_int2(uzly, hodnoty)
n = length(uzly);
h = diff(uzly);
L = uzly(end) + h(1); % perioda
```

```
if numel(unique(diff(uzly))) ~= 1
    error('Uzly nejsou ekvidistanti!')
end
if length(uzly) ~= length(hodnoty)
    error('Vektory uzlu a hodnot musi byt stejne delky!')
end
f = fft(hodnoty);
a = real(f(1)) / n;
T = a;
syms x
for i = 1:(floor(n/2))
    a = 2 * real(f(i+1)) / n;
    b = -2 * imag(f(i+1)) / n;
    T = T + a*cos((i)*x*2*pi/L) + b*sin((i)*x*2*pi/L);
end
if mod(n, 2) == 0 % pro sudy pocet uzlu
    a = real(f(i+2)) / N;
    T = T + a*cos(i*x*2*pi/L);
end
end
```

Výše uvedené vzorce lze díky Eulerově formuli přepsat i do následujícího tvaru:

$$T_n(x) = \sum_{k=0}^{2n} c_k e^{i\frac{2\pi k}{L}x}$$

$$c_k = \frac{1}{2n+1} \sum_{j=0}^{2n} c_k e^{-i\frac{2\pi k}{L}x_j}$$

Dalším možným způsobem řešení pro lichý počet uzlů a periodu 2π je Lagrangeův trigonometrický polynom, kterým se zabývají T. F. Xie a S. P. Zhou ve své knize *On approximation by trigonometric Lagrange interpolating polynomials* [6]. Upravený Lagrangeův polynom a fundamentální polynomy vypadají takto:

$$T_n(x) = \sum_{k=0}^{2n} f(x_k) l_k(x)$$

$$l_k(x) = \frac{1}{2n+1} \frac{\sin\left(\frac{2n+1}{2}(x-x_k)\right)}{\sin\left(\frac{x-x_k}{2}\right)}$$

Fundamentální polynomy v odpovídajících uzlech tvoří nedefinované výrazy. S využitím L'Hopitalova pravidla však můžeme spočítat jejich limitu:

$$\lim_{x\to x_k} \frac{1}{2n+1} \frac{\sin\left(\frac{2n+1}{2}(x-x_k)\right)}{\sin\left(\frac{x-x_k}{2}\right)} = \lim_{x\to x_k} \frac{\cos\left(\frac{2nx-2nx_k+x-x_k}{2}\right)}{\cos\left(\frac{x-x_k}{2}\right)} = \frac{\cos(0)}{\cos(0)} = 1$$

V uzlu x_k se hodnota fundamentálního polynomu l_k limitně blíží k 1 a v ostatních uzlech x_j, kde $j \neq k$, se rovná nule.

2.1.4 Příklad

Tento postup si můžeme vyzkoušet na hodnotách zadaných v příkladu 2.1.1:

x_i	0	$\frac{2\pi}{5}$	$\frac{4\pi}{5}$	$\frac{6\pi}{5}$	$\frac{8\pi}{5}$
f_{x_i}	1	$\frac{1}{2}$	2	-1	0

Výpočet fundamentálních polynomů:

$$l_0(x) = \frac{1}{5}\frac{\sin\left(\frac{5x}{2}\right)}{\sin\left(\frac{x}{2}\right)}$$

$$l_1(x) = \frac{1}{5}\frac{\sin\left(\frac{5}{2}\left(x-\frac{2\pi}{5}\right)\right)}{\sin\left(\frac{1}{2}\left(x-\frac{2\pi}{5}\right)\right)}$$

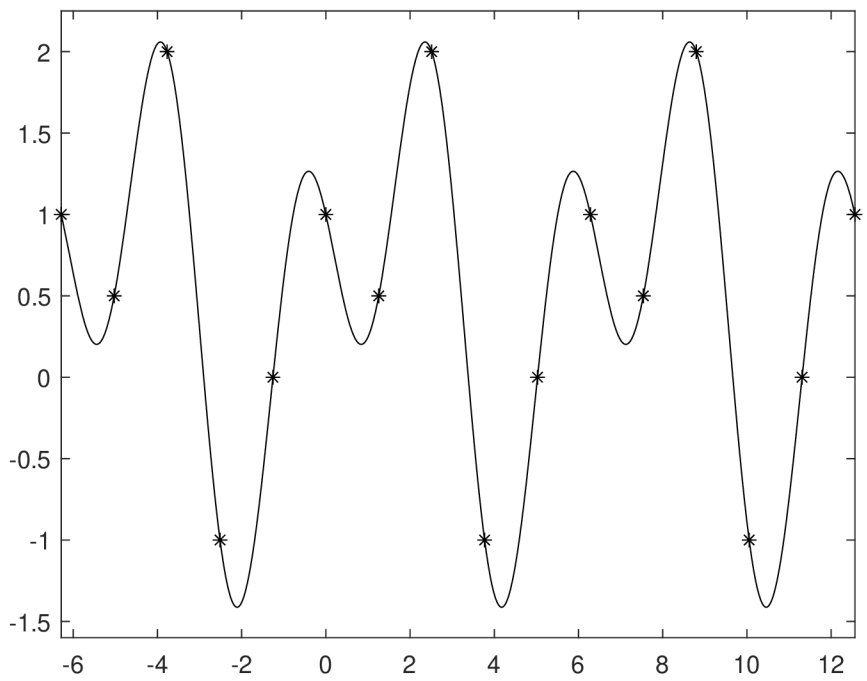
$$l_2(x) = \frac{1}{5}\frac{\sin\left(\frac{5}{2}\left(x-\frac{4\pi}{5}\right)\right)}{\sin\left(\frac{1}{2}\left(x-\frac{4\pi}{5}\right)\right)}$$

$$l_3(x) = \frac{1}{5}\frac{\sin\left(\frac{5}{2}\left(x-\frac{6\pi}{5}\right)\right)}{\sin\left(\frac{1}{2}\left(x-\frac{6\pi}{5}\right)\right)}$$

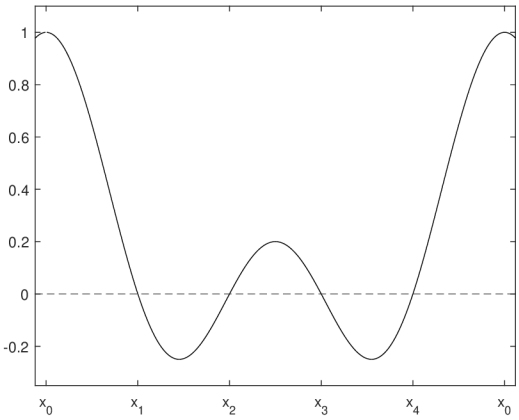
$$l_4(x) = \frac{1}{5}\frac{\sin\left(\frac{5}{2}\left(x-\frac{8\pi}{5}\right)\right)}{\sin\left(\frac{1}{2}\left(x-\frac{8\pi}{5}\right)\right)}$$

Výsledný trigonometrický polynom:

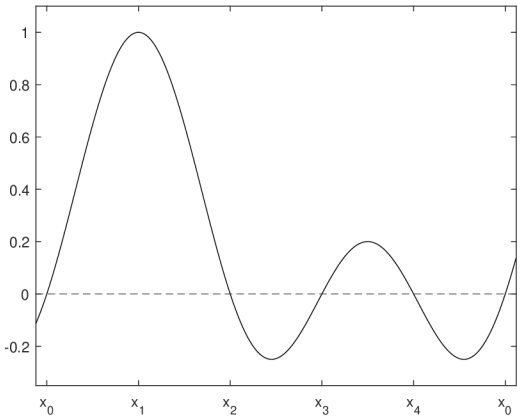
$$T_2(x) = \frac{\sin\left(\frac{5x}{2}\right)}{5\sin\left(\frac{x}{2}\right)} - \frac{\sin\left(\frac{5x}{2}\right)}{10\sin\left(\frac{x}{2}-\frac{\pi}{5}\right)} + \frac{2\sin\left(\frac{5x}{2}\right)}{5\sin\left(\frac{x}{2}-\frac{2\pi}{5}\right)} + \frac{\sin\left(\frac{5x}{2}\right)}{5\sin\left(\frac{x}{2}-\frac{3\pi}{5}\right)}$$



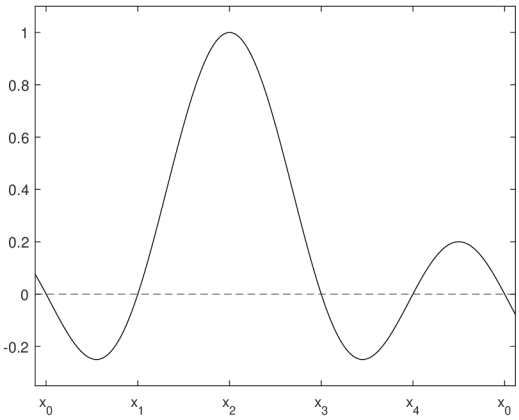
Obrázek 2.3: Trigonometrický polynom $T_2(x)$ na intervalu $[-2\pi, 4\pi]$



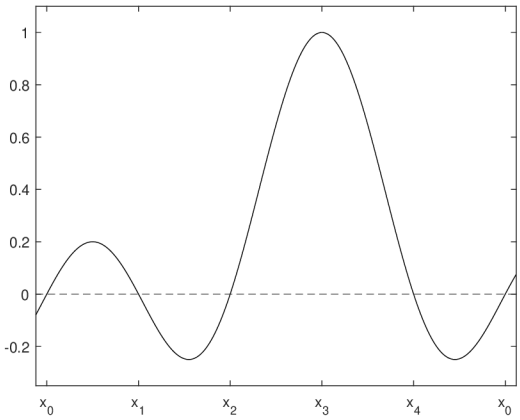
fundamentální polynom l_0



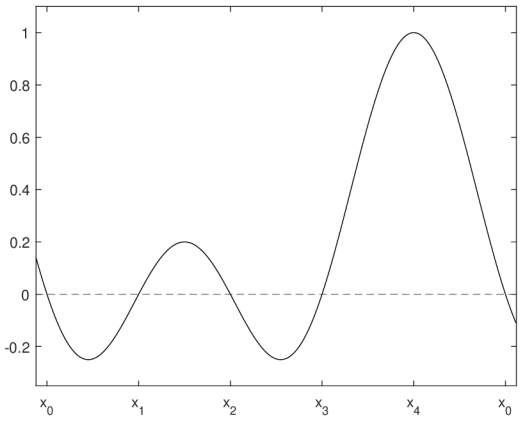
fundamentální polynom l_1



fundamentální polynom l_2



fundamentální polynom l_3



fundamentální polynom l_4

Obrázek 2.4: Fundamentální polynomy polynomu $T_2(x)$

2.2 Interpolace na nerovnoměrně rozložených uzlech

Jestliže máme nerovnoměrně rozložené uzly, pro výpočet trigonometrického polynomu nemůžeme použít dříve uvedené vzorce pro a_k a b_k, protože báze není ortogonální. Nicméně stále můžeme použít soustavu lineárních rovnic nebo metodu nejmenších čtverců.

2.2.1 Řešení v MATLAB

V prvním kroku musíme sestavit trigonometrický polynom T v požadovaném tvaru. Ten nám následně pomůže k sestavení matice A, kde řádky odpovídají trigonometrickému polynomu T po dosazení jednotlivých uzlů.

```
function[A, T] = mat_A(uzly, hodnoty, perioda)
if length(uzly) ~= length(hodnoty)
    error('Vektory uzlu a hodnot musi byt stejne delky!')
end
if nargin < 3 % nastaveni implicitni hodnoty periody
        perioda = 2*pi;
end
T = [1];
n = floor(length(uzly)/2);
syms x
for i = 1:n
    a = [cos(2*pi*i*x/perioda), sin(2*pi*i*x/perioda)];
    T = [T, a];
end
if mod(length(uzly),2) ~= 0 % pro sudy pocet uzlu
    T = [T, cos(2*pi*(n+1)*x/perioda)];
end
A = [];
for i = 1:length(uzly)
    A = [A;subs(T, x, uzly(i))];
end
end
```

Dále můžeme trigonometrický polynom získat řešením soustavy rovnic:

```
function[T] = soustava_rovnic(uzly, hodnoty, perioda)
[A, T] = mat_A(uzly, hodnoty, perioda)
y = double(A)\hodnoty';
T = T*y;
end
```

nebo použít metodu nejmenších čtverců:

```
function[T] = MNC(uzly, hodnoty, perioda)
[A, T] = mat_A(uzly, hodnoty, perioda)
y = inv(A'*A)*A'*hodnoty;
T = T*y;
end
```

Tento postup je samozřejmě možné použít i pro rovnoměrně rozložené uzly.

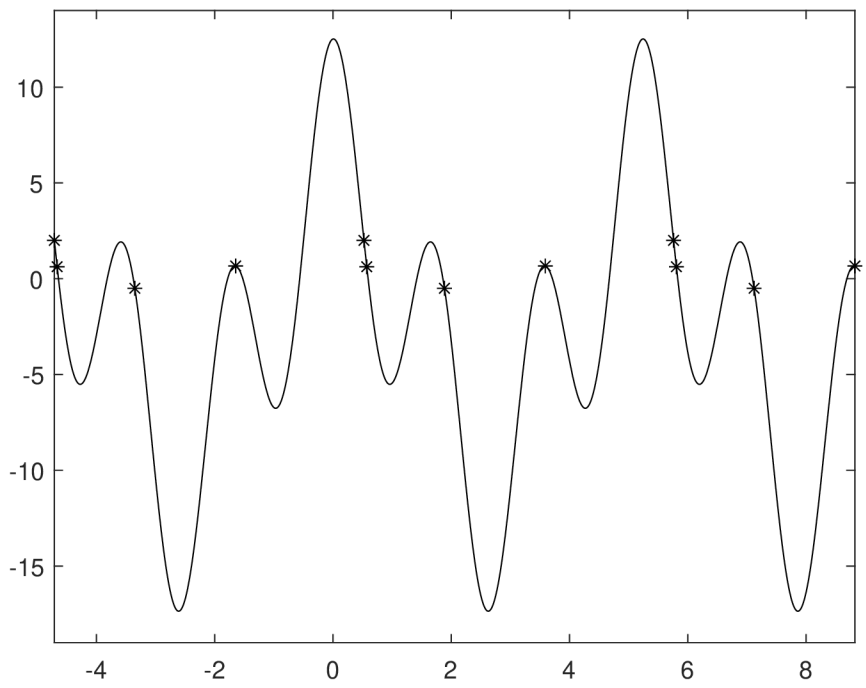
2.2.2 Příklad

Funkce si můžeme vyzkoušet na následujících hodnotách s periodou $\frac{5\pi}{3}$:

x_i	$\frac{\pi}{6}$	$\frac{2\pi}{11}$	$\frac{3\pi}{5}$	$\frac{8\pi}{7}$
f_{x_i}	2	$\frac{5}{8}$	$-\frac{1}{2}$	$\frac{2}{3}$

```
>> vpa(soustava_rovnic([pi/6, 2*pi/11, 3*pi/5, 8*pi/7], [2,
   5/8, -1/2, 2/3], 5*pi/3), 6)
ans =
7.72229*cos(1.2*x) + 0.682046*sin(1.2*x) + 7.20918*cos(3.6*x)
   - 2.4206

>> vpa(MNC([pi/6, 2*pi/11, 3*pi/5, 8*pi/7], [2, 5/8, -1/2,
   2/3], 5*pi/3), 6)
ans =
7.72229*cos(1.2*x) + 0.682046*sin(1.2*x) + 7.20918*cos(3.6*x)
   - 2.4206
```



Obrázek 2.5: Trigonometrický polynom $T_2(x)$ na intervalu $\left[-\frac{3\pi}{2}, \frac{59\pi}{21}\right]$

Phillip J. Davis ve své již zmíněné knize *Interpolation and Approximation* (viz [2]) popisuje další způsob trigonometrické interpolace, a tím je Gaussova formule trigonometrické interpolace:

$$T_n(x) = \sum_{k=0}^{2n} f(x_k) t_k(x)$$

$$t_k(x) = \frac{\prod\limits_{\substack{i=0 \\ i \neq k}}^{2n} \sin \frac{1}{2}(x - x_i)}{\prod\limits_{\substack{i=0 \\ i \neq k}}^{2n} \sin \frac{1}{2}(x_k - x_i)}, j = 0, 1, \ldots, 2n$$

Tato formule funguje pro lichý počet bodů. Výpočet je podobný jako u Lagrangových trigonometrických polynomů, avšak v tomto případě můžeme interpolovat i na nerovnoměrně rozložených uzlech. Davis zde také zmiňuje, že každá funkce t_k je lineární kombinací funkcí $1, cos(x), sin(x), \ldots, cos(nx), sin(nx)$.

2.2.3 Příklad

Gaussovu interpolační formuli si vyzkoušíme na hodnotách daných tabulkou:

x_i	0	$\frac{\pi}{4}$	$\frac{7\pi}{10}$	π	$\frac{8\pi}{5}$
f_{x_i}	$\frac{1}{4}$	4	1	-$\frac{1}{3}$	-3

Uzly dosadíme do vzorce pro $t_k(x)$:

$$t_0 = \frac{\sin\left(\frac{1}{2}\left(x - \frac{\pi}{4}\right)\right) \sin\left(\frac{1}{2}\left(x - \frac{7\pi}{10}\right)\right) \sin\left(\frac{1}{2}(x - \pi)\right) \sin\left(\frac{1}{2}\left(x - \frac{8\pi}{5}\right)\right)}{\sin\left(\frac{1}{2}\left(0 - \frac{\pi}{4}\right)\right) \sin\left(\frac{1}{2}\left(0 - \frac{7\pi}{10}\right)\right) \sin\left(\frac{1}{2}(0 - \pi)\right) \sin\left(\frac{1}{2}\left(0 - \frac{8\pi}{5}\right)\right)}$$

$$t_1 = \frac{\sin\left(\frac{1}{2}(x - 0)\right) \sin\left(\frac{1}{2}\left(x - \frac{7\pi}{10}\right)\right) \sin\left(\frac{1}{2}(x - \pi)\right) \sin\left(\frac{1}{2}\left(x - \frac{8\pi}{5}\right)\right)}{\sin\left(\frac{1}{2}\left(\frac{\pi}{4} - 0\right)\right) \sin\left(\frac{1}{2}\left(\frac{\pi}{4} - \frac{7\pi}{10}\right)\right) \sin\left(\frac{1}{2}\left(\frac{\pi}{4} - \pi\right)\right) \sin\left(\frac{1}{2}\left(\frac{\pi}{4} - \frac{8\pi}{5}\right)\right)}$$

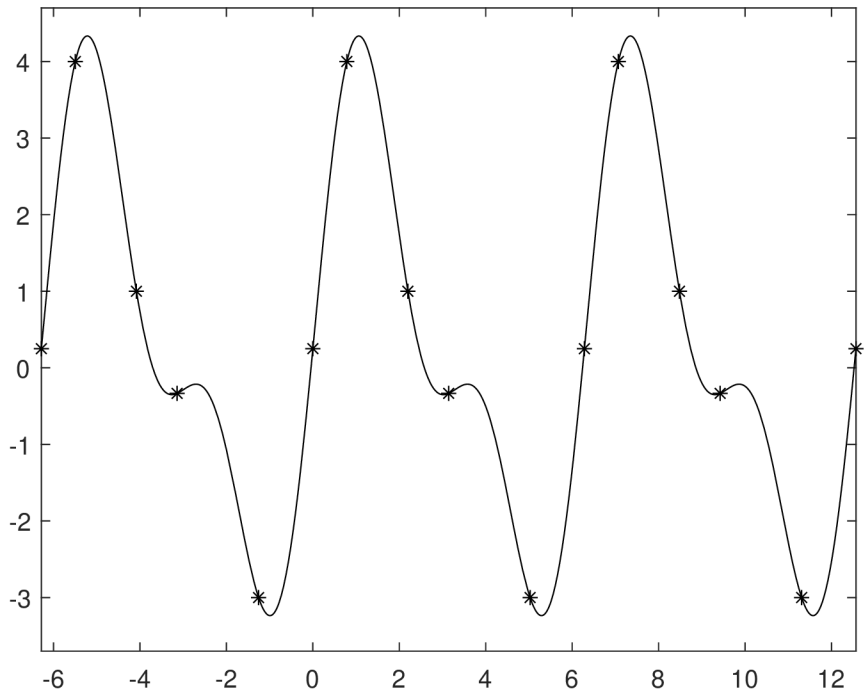
$$t_2 = \frac{\sin\left(\frac{1}{2}(x - 0)\right) \sin\left(\frac{1}{2}\left(x - \frac{\pi}{4}\right)\right) \sin\left(\frac{1}{2}(x - \pi)\right) \sin\left(\frac{1}{2}\left(x - \frac{8\pi}{5}\right)\right)}{\sin\left(\frac{1}{2}\left(\frac{7\pi}{10} - 0\right)\right) \sin\left(\frac{1}{2}\left(\frac{7\pi}{10} - \frac{\pi}{4}\right)\right) \sin\left(\frac{1}{2}\left(\frac{7\pi}{10} - \pi\right)\right) \sin\left(\frac{1}{2}\left(\frac{7\pi}{10} - \frac{8\pi}{5}\right)\right)}$$

$$t_3 = \frac{\sin\left(\frac{1}{2}(x - 0)\right) \sin\left(\frac{1}{2}\left(x - \frac{\pi}{4}\right)\right) \sin\left(\frac{1}{2}\left(x - \frac{7\pi}{10}\right)\right) \sin\left(\frac{1}{2}\left(x - \frac{8\pi}{5}\right)\right)}{\sin\left(\frac{1}{2}(\pi - 0)\right) \sin\left(\frac{1}{2}\left(\pi - \frac{\pi}{4}\right)\right) \sin\left(\frac{1}{2}\left(\pi - \frac{7\pi}{10}\right)\right) \sin\left(\frac{1}{2}\left(\pi - \frac{8\pi}{5}\right)\right)}$$

$$t_4 = \frac{\sin\left(\frac{1}{2}(x - 0)\right) \sin\left(\frac{1}{2}\left(x - \frac{\pi}{4}\right)\right) \sin\left(\frac{1}{2}\left(x - \frac{7\pi}{10}\right)\right) \sin\left(\frac{1}{2}(x - \pi)\right)}{\sin\left(\frac{1}{2}\left(\frac{8\pi}{5} - 0\right)\right) \sin\left(\frac{1}{2}\left(\frac{8\pi}{5} - \frac{\pi}{4}\right)\right) \sin\left(\frac{1}{2}\left(\frac{8\pi}{5} - \frac{7\pi}{10}\right)\right) \sin\left(\frac{1}{2}\left(\frac{8\pi}{5} - \pi\right)\right)}$$

A následně vynásobíme příslušnými funkčními hodnotami a sečteme:

$$T(x) = \frac{1}{4}t_0 + 4t_1 + t_2 - \frac{1}{3}t_3 - 3t_4$$



Obrázek 2.6: Trigonometrický polynom $T(x)$ na intervalu $[-2\pi, 4\pi]$

Dále ve své knize Phillip J. Davis popisuje kosinovou a sinovou interpolaci. Kosinovou interpolaci můžeme použít, pokud víme, že je interpolovaná funkce sudá. Pro $n+1$ různých uzlů $0 \leq x_0 < x_1 < \cdots < x_n < \pi$ vypadá kosinová interpolace následovně:

$$C(x) = \sum_{k=0}^{n} f(x_k) c_k(x)$$

$$c_k(x) = \frac{\prod\limits_{\substack{i=0 \\ i \neq k}}^{n} (\cos x - \cos x_i)}{\prod\limits_{\substack{i=0 \\ i \neq k}}^{n} (\cos x_k - \cos x_i)}$$

A obdobně je dána sinová interpolace na n různých uzlech $0 < x_1 < \cdots < x_n < \pi$ pro liché funkce:

$$S(x) = \sum_{k=0}^{n} f(x_k) s_k(x)$$

$$s_k(x) = \sin x \frac{\prod\limits_{\substack{i=1 \\ i \neq k}}^{n} (\cos x - \cos x_i)}{\prod\limits_{\substack{i=1 \\ i \neq k}}^{n} (\cos x_k - \cos x_i)}$$

Pro zobecnění této funkce pro interpolaci na nerovnoměrně rozložených uzlech musíme $\sin x$ vydělit funkční hodnotou funkce sinus v daném bodě, tedy $\sin x_k$:

$$s_k(x) = \frac{\sin x}{\sin x_k} \frac{\prod\limits_{\substack{i=1 \\ i \neq k}}^{n} (\cos x - \cos x_i)}{\prod\limits_{\substack{i=1 \\ i \neq k}}^{n} (\cos x_k - \cos x_i)}$$

U sinové a kosinové interpolace nezáleží na tom, zda máme sudý nebo lichý počet uzlů. Uvedené vzorce pro výpočet Gaussovy trigonometrické formule, sinové nebo kosinové interpolační funkce pracují s periodou 2π. Můžeme je však přizpůsobit pro obecnou periodu tím, že argument každé goniometrické funkce vynásobíme členem $\frac{2\pi}{L}$, kde L je požadovaná perioda.

2.2.4 Příklad

x_i	$-\frac{4\pi}{5}$	$-\frac{3\pi}{5}$	$-\frac{\pi}{6}$	$\frac{\pi}{6}$	$\frac{3\pi}{5}$	$\frac{4\pi}{5}$
f_{x_i}	-2	1	$-\frac{1}{5}$	$\frac{1}{5}$	-1	2

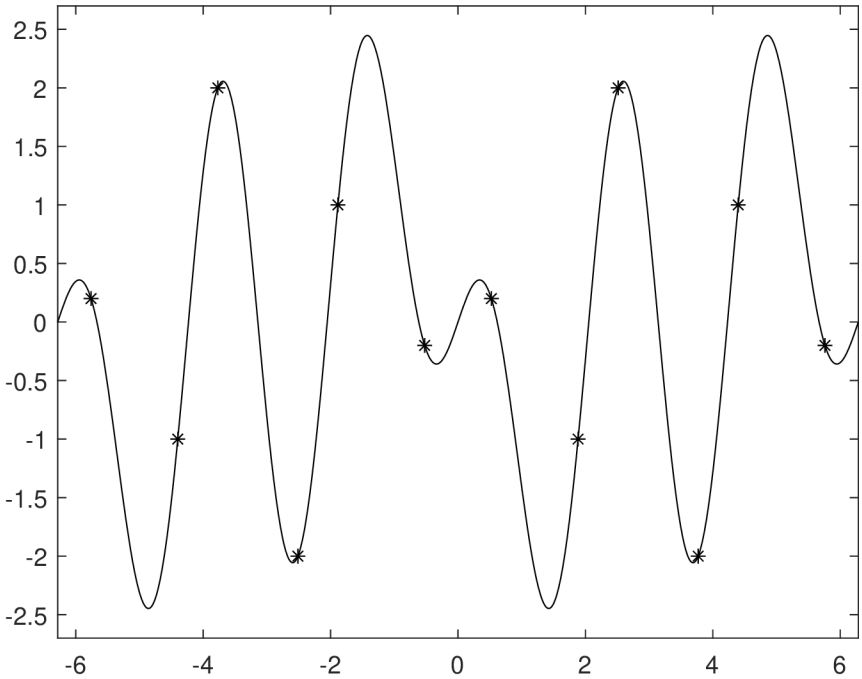
Z tabulky si můžeme všimnout symetrie bodů podle počátku, použijeme tedy sinovou interpolaci. K výpočtu nám díky symetrii stačí pouze kladné uzly. Dále je postup obdobný jako v předchozím příkladu:

$$s_1 = \frac{\sin(x)\left(\cos(x) - \cos\left(\frac{3\pi}{5}\right)\right)\left(\cos(x) - \cos\left(\frac{4\pi}{5}\right)\right)}{\sin\left(\frac{\pi}{6}\right)\left(\cos\left(\frac{\pi}{6}\right) - \cos\left(\frac{3\pi}{5}\right)\right)\left(\cos\left(\frac{\pi}{6}\right) - \cos\left(\frac{4\pi}{5}\right)\right)}$$

$$s_2 = \frac{\sin(x)\left(\cos(x) - \cos\left(\frac{\pi}{6}\right)\right)\left(\cos(x) - \cos\left(\frac{4\pi}{5}\right)\right)}{\sin\left(\frac{3\pi}{5}\right)\left(\cos\left(\frac{3\pi}{5}\right) - \cos\left(\frac{\pi}{6}\right)\right)\left(\cos\left(\frac{3\pi}{5}\right) - \cos\left(\frac{4\pi}{5}\right)\right)}$$

$$s_3 = \frac{\sin(x)\left(\cos(x) - \cos\left(\frac{\pi}{6}\right)\right)\left(\cos(x) - \cos\left(\frac{3\pi}{5}\right)\right)}{\sin\left(\frac{4\pi}{5}\right)\left(\cos\left(\frac{4\pi}{5}\right) - \cos\left(\frac{\pi}{6}\right)\right)\left(\cos\left(\frac{4\pi}{5}\right) - \cos\left(\frac{3\pi}{5}\right)\right)}$$

$$S(x) = \frac{1}{5}s_1 - s_2 + 2s_3$$



Obrázek 2.7: Trigonometrický polynom $S(x)$ na intervalu $[-2\pi, 2\pi]$

Kapitola 3

Exponenciální interpolace

Další funkcí, kterou můžeme využít jako bázovou funkci k interpolaci, je exponenciální funkce. Dle článku *Interpolation by Series of Exponential Functions Whose Exponents Are Condensed in a Certain Direction* od autorů S. Merzlyakov a S. Popenov (viz [4]) hledáme funkci následujícího tvaru, kterou autoři obecně řeší v komplexním oboru:

$$\phi(x) = a_1 e^{b_1 x} + a_2 e^{b_2 x} + \cdots + a_n e^{b_n x}$$

Je složité obecně určit, kdy dostaneme reálné řešení. V obecném případě řešení také nemusí vůbec existovat.

Hledáme $2n$ koeficientů, k čemuž potřebujeme $2n$ uzlů. V tomto vyjádření nemáme explicitně uveden konstantní člen, protože pokud by byl potřebný, metoda by k němu přirozeně dospěla. Pro jeden z členů součtu bychom dostali $b_k = 0$, což nám dá $a_k e^{0x} = a_k$. Konstantní člen je tak již zahrnut v obecném vyjádření. Pro uvedený tvar interpolační funkce je výhodné mít sudý počet uzlů, pokud však máme lichý počet uzlů, můžeme explicitně zahrnout konstantní člen, a tím dosáhnout požadovaného počtu neznámých.

Jednodušší situace nastává, pokud konkrétní bázové funkce máme zadané, nebo se je rozhodneme sami zvolit. Tím se zbavíme neznámých v exponentech a dostáváme opět lineární soustavu s reálným řešením. Pro jednoznačnost interpolace na 2n uzlech však v naší interpolační funkci stále potřebujeme $2n$ neznámých koeficientů:

$$\phi(x) = a_1 e^{b_1 x} + a_2 e^{b_2 x} + \cdots + a_{2n} e^{b_{2n} x}$$

Hledáme tak $a_1, ..., a_{2n}$ a $b_1, ..., b_{2n}$ máme zadané, nebo si je sami určíme.

3.1 Příklad

Podíváme se na to, jak se bude graf funkce měnit, volíme-li různé bázové exponenciální funkce. Uzly a funkční hodnoty v nich jsou dány tabulkou:

x_i	-3	-1	$\frac{1}{2}$	4	6
f_{x_i}	-2	0	5	$\frac{3}{7}$	4

Jako první můžeme zvolit bázové funkce: $e^{-2x}, e^{-x}, 1, e^{x}, e^{2x}$. Hledaná funkce je tak tvaru:

$$\phi(x) = a_1 e^{-2x} + a_2 e^{-x} + a_3 + a_4 e^{x} + a_5 e^{2x}$$

Dosadíme hodnoty a vyřešíme soustavu rovnic:

$$\begin{aligned}
-2 &= a_1 e^{-2(-3)} + a_2 e^{-(-3)} + a_3 + a_4 e^{-3} + a_5 e^{2(-3)} \\
0 &= a_1 e^{-2(-1)} + a_2 e^{-(-1)} + a_3 + a_4 e^{-1} + a_5 e^{2(-1)} \\
5 &= a_1 e^{-2\frac{1}{2}} + a_2 e^{-\frac{1}{2}} + a_3 + a_4 e^{\frac{1}{2}} + a_5 e^{2\frac{1}{2}} \\
\frac{3}{7} &= a_1 e^{-2\cdot 4} + a_2 e^{-4} + a_3 + a_4 e^{4} + a_5 e^{2\cdot 4} \\
4 &= a_1 e^{-2\cdot 6} + a_2 e^{-6} + a_3 + a_4 e^{6} + a_5 e^{2\cdot 6}
\end{aligned}$$

Výsledná interpolační funkce:

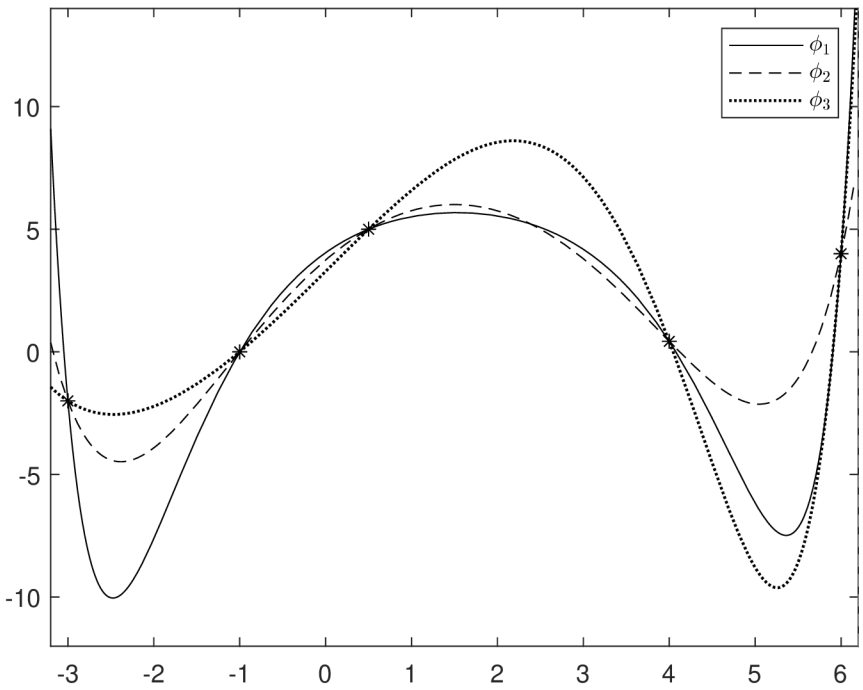
$$\phi_1(x) \approx 0.119695 e^{-2x} - 2.847128 e^{-x} + 6.904466 - 0.134947 e^{x} + 0.000317 e^{2x}$$

Obdobně můžeme spočítat koeficienty pro libovolné báze. Například pro bázové funkce $e^{-x}, e^{-\frac{x}{2}}, 1, e^{\frac{x}{2}}, e^{x}$ vychází interpolační funkce:

$$\phi_2(x) \approx 1.808930 e^{-x} - 12.162895 e^{-\frac{x}{2}} + 16.797606 - 2.805846 e^{\frac{x}{2}} + 0.109463 e^{x}$$

S bází $e^{-x}, 1, e^{\frac{x}{3}}, e^{\frac{2x}{3}}, e^{x}$ získáme:

$$\phi_3(x) \approx 0.234375 e^{-x} - 15.149309 + 25.980027 e^{\frac{x}{3}} - 8.510448 e^{\frac{2x}{3}} + 0.723388 e^{x}$$



Obrázek 3.1: Srovnání interpolačních funkcí $\phi_1(x), \phi_2(x)$ a $\phi_3(x)$

V situaci popsané na začátku této kapitoly, kde chceme najít vhodné koeficienty a_i i b_i, dostáváme systém $2n$ nelineárních rovnic o $2n$ neznámých, každá z rovnic má však pouze n členů. Nejjednodušší situace nastává, pokud máme zadané pouze 2 uzly a hodnoty v nich. Hledáme tak pouze jednu exponenciálu, na které zadané body leží. Abychom dostali reálné řešení, musí mít funkční hodnoty v uzlech stejné znaménko. Protože k funkci nepřičítáme žádný konstantní člen, jsou všechny její hodnoty kladné, záporné nebo nulové.

3.2 Příklad

Ukažme si to na jednoduchém příkladu:

x_i	-2	3
f_{x_i}	6	1

Hledáme funkci tvaru:

$$\phi(x) = a_1 e^{b_1 x}$$

Po dosazení získáme:

$$a_1 e^{-2b_1} = 6$$
$$a_1 e^{3b_1} = 1$$

V této situaci dostáváme pouze 2 rovnice a snadno můžeme z jedné rovnice dosadit do druhé. Vyjádříme si například z první rovnice a_1, dosadíme do druhé rovnice a dopočítáme:

$$a_1 = 6e^{2b_1}$$
$$6e^{2b_1}e^{3b_1} = 6e^{5b_1} = 1 \rightarrow b_1 = -\frac{1}{5}\ln(6)$$
$$a_1 = 6e^{2(-\frac{1}{5}\ln(6))} = \sqrt[5]{216}$$

Dostáváme tak výslednou funkci, jejíž graf je zobrazen na obrázku 3.2 na následující straně:

$$\phi(x) = \sqrt[5]{216}e^{-\frac{1}{5}\ln(6)x} = 6^{\frac{3}{5}} \cdot 6^{-\frac{1}{5}x}$$

To je však pouze reálná část řešení, pro tento příklad MATLAB našel 4 další komplexní řešení.

3.3 Příklad

Nyní si ukažme příklad, kde funkční hodnoty mají v uzech opačná znaménka.

x_i	0	1
f_{x_i}	1	-1

Hledaná funkce bude mít stejný tvar jako v předchozím příkladu, protože opět máme 2 uzly. Můžeme tedy rovnou dosadit:

$$a_1 e^{0b_1} = 1$$
$$a_1 e^{1b_1} = -1$$

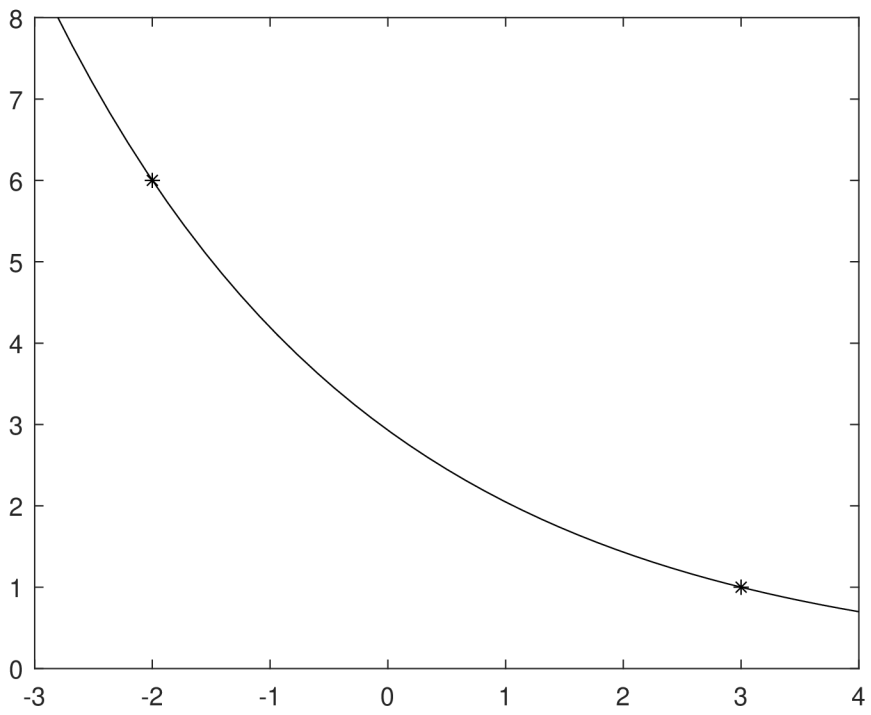
Z první rovnice přímo můžeme vidět, že $a_1 = 1$, a z druhé rovnice dostáváme $b_1 = i\pi$. Výsledná funkce je tedy:

$$\phi(x) = e^{i\pi x}$$

Můžeme ji také vyjádřit ve tvaru:

$$\phi(x) = \cos(\pi x) + i\sin(\pi x)$$

Jelikož je řešení komplexní, není snadné jej přímo zobrazit graficky. Nicméně dosazením jednotlivých bodů do interpolační funkce lze ověřit, že skutečně prochází zadanými body.



Obrázek 3.2: Interpolační funkce $\phi(x)$ z příkladu 3.2

3.4 Příklad

Následující tabulka obsahuje 3 uzly a jejich funkční hodnoty. Jelikož máme lichý počet uzlů, můžeme přidat konstantní člen.

x_i	-1	0	1
f_{x_i}	$\frac{7}{3}$	3	5

Hledáme interpolační funkci:

$$\phi(x) = a_0 + a_1 e^{b_1 x}$$

Po dosazení získáváme soustavu:

$$\begin{aligned} a_0 + a_1 e^{-b_1} &= \frac{7}{3} \\ a_0 + a_1 e^{0} &= 3 \\ a_0 + a_1 e^{b_1} &= 5 \end{aligned}$$

Pro usnadnění výpočtu soustavy můžeme využít MATLAB:

```
>> uzly = [-1, 0, 1];
>> hodnoty = [7/3, 3, 5];
>> syms a0 a1 b1 x
>> f(x) = a0 + a1*exp(b1*x);
>> S = solve(f(uzly(1))==hodnoty(1), f(uzly(2))==hodnoty(2),
   f(uzly(3))==hodnoty(3))
S =
  struct with fields:
    a0: 2
    a1: 1
    b1: log(3)
```

Po dosazení koeficientů získáme interpolační funkci:

$$\phi(x) = e^{ln(3)\cdot x} + 2 = 3^x + 2$$

Přidáme-li do tabulky bod, který leží na výsledné interpolační funkci, můžeme si vyzkoušet, zda funkce skutečně přirozeně dospěje ke konstantnímu členu.

x_i	-1	0	1	2
f_{x_i}	$\frac{7}{3}$	3	5	11

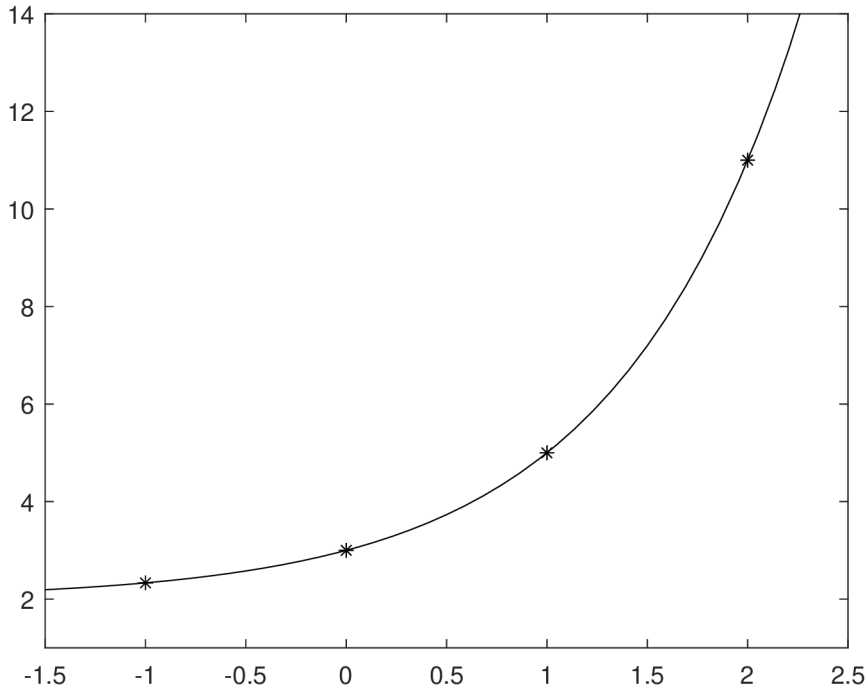
Interpolační funkce se nám tak změní na tvar:

$$\phi(x) = a_1 e^{b_1} + a_2 e^{b_2}$$

```
>> uzly = [-1, 0, 1, 2];
>> hodnoty = [7/3, 3, 5, 11];
>> syms a1 a2 b1 b2 x
>> f(x) = a1*exp(b1*x)+a2*exp(b2*x);
>> S = solve(f(uzly(1))==hodnoty(1), f(uzly(2))==hodnoty(2),
   f(uzly(3))==hodnoty(3), f(uzly(4))==hodnoty(4));
>> [S.a1,S.a2,S.b1,S.b2]
ans =
[2, 1,       0, log(3)]
[1, 2, log(3),       0]
```

Tentokrát dostáváme 2 výsledky, které jsou ve skutečnosti totožné, liší se pouze pořadím sčítanců. Výsledná interpolační funkce je tedy stejná, jako v předchozím případě. Dostali jsme tak konstantní člen, aniž bychom ho explicitně vyjádřili:

$$\phi(x) = e^{ln(3)\cdot x} + 2e^0 = 3^x + 2$$



Obrázek 3.3: Interpolační funkce $\phi(x)$ z příkladu 3.4

3.5 Příklad

Podívejme se na příklad exponenciální interpolace na 4 uzlech s reálným řešením:

x_i	1	2	3	4
f_{x_i}	5	2	-1	-7

Hledáme funkci:

$$\phi(x) = a_1 e^{b_1} + a_2 e^{b_2}$$

K nalezení řešení můžeme opět využít MATLAB:

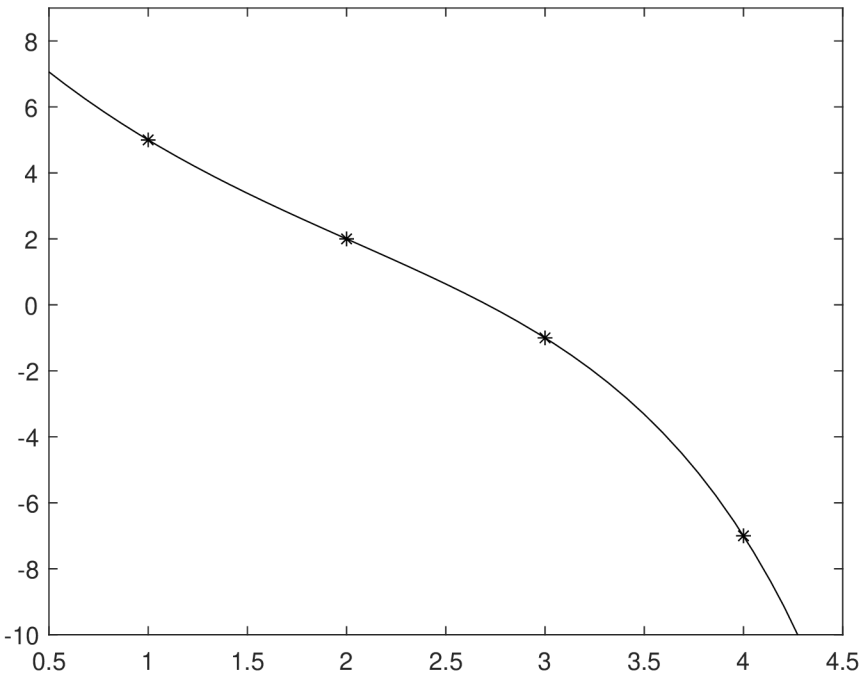
```
>> uzly = [1, 2, 3, 4];
>> hodnoty = [5, 2, -1, -7];
>> syms a1 a2 b1 b2 x
>> f(x) = a1*exp(b1*x)+a2*exp(b2*x);
>> S = solve(f(uzly(1))==hodnoty(1), f(uzly(2))==hodnoty(2),
   f(uzly(3))==hodnoty(3), f(uzly(4))==hodnoty(4));
```

```
>> [S.a1,S.a2,S.b1,S.b2]
ans =
[49/10 - (389*61^(1/2))/610, (389*61^(1/2))/610 + 49/10, log
   (61^(1/2)/6 + 11/6), log(11/6 - 61^(1/2)/6)]
[(389*61^(1/2))/610 + 49/10, 49/10 - (389*61^(1/2))/610, log
   (11/6 - 61^(1/2)/6), log(61^(1/2)/6 + 11/6)]
```

$$a_1 = \frac{49}{10} - \frac{389\sqrt{61}}{610} \qquad b_1 = \ln\left(\frac{\sqrt{61}}{6} + \frac{11}{6}\right)$$

$$a_2 = \frac{389\sqrt{61}}{610} + \frac{49}{10} \qquad b_2 = \ln\left(\frac{11}{6} - \frac{\sqrt{61}}{6}\right)$$

$$\phi(x) = \frac{\left(\frac{389\sqrt{61}}{610} + \frac{49}{10}\right)(11 - \sqrt{61})^x}{6^x} - \frac{\left(\frac{389\sqrt{61}}{610} - \frac{49}{10}\right)(\sqrt{61} + 11)^x}{6^x}$$



Obrázek 3.4: Interpolační funkce $\phi(x)$ z příkladu 3.5

Kapitola 4

Racionální interpolace

Krátce se zmíníme ještě o dalším typu interpolace, a tím je interpolace racionální lomenou funkcí, což umožňuje například lépe zachytit asymptotické chování dat. J. Stoer a R. Bulirsch v knize *Introduction to Numerical Analysis* (viz [5]) vymezují racionální interpolační funkci:

$$R_{m,n}(x) = \frac{P_m(x)}{Q_n(x)} = \frac{a_0 + a_1 x + \cdots + a_m x^m}{b_0 + b_1 x + \cdots + b_n x^n},$$

kde $P_m(x)$, $Q_n(x)$ jsou polynomy a m, n jejich stupně. Bez újmy na obecnosti však můžeme b_0 nahradit číslem 1:

$$R_{m,n}(x) = \frac{P_m(x)}{Q_n(x)} = \frac{a_0 + a_1 x + \cdots + a_m x^m}{1 + b_1 x + \cdots + b_n x^n}$$

Autoři uvádí, že problém můžeme řešit například pomocí vzájemných diferencí nebo algoritmem Nevillova typu. Nicméně k problému opět můžeme přistupovat metodou nejmenších čtverců. Počet podmínek je v tomto případě $m + n + 1$, a tedy pro dosažení interpolace opět potřebujeme stejný počet uzlů. Požadujeme tak, aby pro každé x_i, kde $i = 0, \ldots, m + n + 1$, platilo:

$$\frac{a_0 + a_1 x_i + \cdots + a_m {x_i}^m}{1 + b_1 x_i + \cdots + b_n {x_i}^n} = f_{x_i},$$

což lze jednoduchou úpravou, díky které se zbavíme zlomku, převést na následující tvar:

$$a_0 + a_1 x_i + \cdots + a_m {x_i}^m - f_{x_i} b_1 x_i - \ldots - f_{x_i} b_n {x_i}^n = f_{x_i}$$

Stupně m, n můžeme volit stejné nebo je přizpůsobit vlastnostem zadaných dat. Je však zřejmé, že pokud bychom zvolili $n = 0$, dostali bychom interpolační polynom. Racionální interpolace oproti polynomiální dosahuje nižšího stupně, což může být výhodou zejména při vyšším počtu uzlů.

4.1 Příklad

x_i	$-\frac{5}{2}$	-1	$\frac{1}{3}$	2	3
f_{x_i}	$\frac{1}{2}$	-2	$-\frac{7}{3}$	1	4

Máme zadaných 5 uzlů a potřebujeme dodržet podmínku $m+n+1=5$. Můžeme tak sestavit racionální interpolační funkci $R_{1,3}, R_{3,1}, R_{2,2}$ nebo $R_{0,4}$. Interpolační funkce $R_{4,0}$ by byla polynomiální. Ukažme si postup výpočtu interpolační funkce $R_{2,2}$:

$$R_{2,2}(x) = \frac{a_0 + a_1 x + a_2 x^2}{1 + b_1 x + b_2 x^2}$$

Zadané honoty dosadíme do upraveného tvaru funkce:

$$a_0 + a_1 \cdot \left(-\frac{5}{2}\right) + a_2 \cdot \left(-\frac{5}{2}\right)^2 - \frac{1}{2} b_1 \cdot \left(-\frac{5}{2}\right) - \frac{1}{2} b_2 \cdot \left(-\frac{5}{2}\right)^2 = \frac{1}{2}$$

$$a_0 + a_1 \cdot (-1) + a_2 \cdot (-1)^2 - (-2) b_1 \cdot (-1) - (-2) b_2 \cdot (-1)^2 = -2$$

$$a_0 + a_1 \cdot \frac{1}{3} + a_2 \cdot \left(\frac{1}{3}\right)^2 - \left(-\frac{7}{3}\right) b_1 \cdot \frac{1}{3} - \left(-\frac{7}{3}\right) b_2 \cdot \left(\frac{1}{3}\right)^2 = -\frac{7}{3}$$

$$a_0 + a_1 \cdot 2 + a_2 \cdot 2^2 - 1 b_1 \cdot 2 - 1 b_2 \cdot 2^2 = 1$$

$$a_0 + a_1 \cdot 3 + a_2 \cdot 3^2 - 4 b_1 \cdot 3 - 4 b_2 \cdot 3^2 = 4$$

Po vyřešení soustavy získáváme výslednou funkci:

$$R_{2,2}(x) = \frac{-1265 + 208x + 345x^2}{505 - 35x + 24x^2}$$

Pro porovnání můžeme uvést například interpolační funkci $R_{3,1}$, která má následující tvar:

$$R_{3,1}(x) = \frac{a_0 + a_1 x + a_2 x^2 + a_3 x^3}{1 + b_1 x}$$

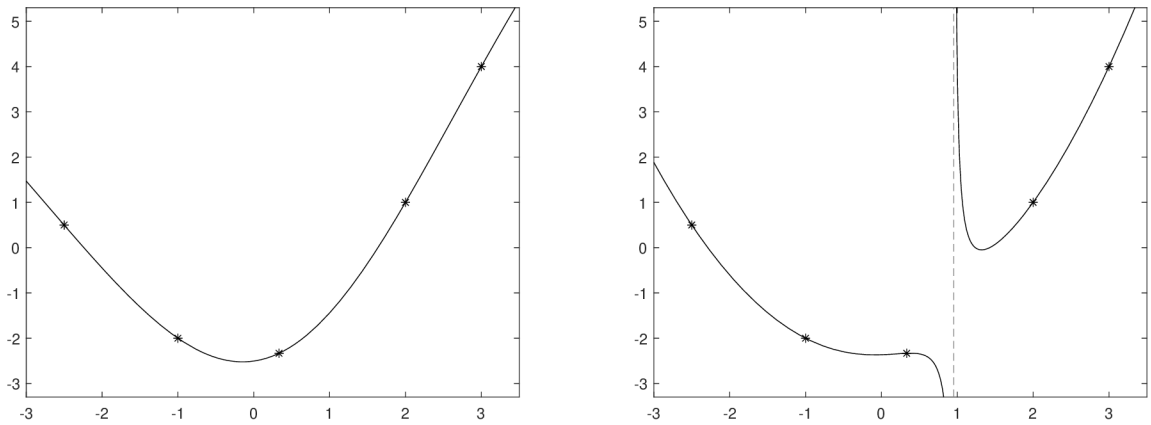
A po dopočítání získáváme:

$$R_{3,1}(x) = -\frac{-97 + 104x + 9x^2 - 24x^3}{-41 + 43x}$$

Srovnání funkcí $R_{2,2}(x)$ a $R_{3,1}(x)$ si můžete prohlédnout na obrázku 4.1.

4.2 Funkce v MATLAB

Výpočet si můžeme usnadnit funkcí v MATLAB. Tato funkce bere za své vstupní parametry vektory uzlů a hodnot, ale také požadované stupně polynomů m a n. Funkce funguje i pokud je součet $m+n+1$ menší než počet uzlů, upozorní nás však, že výsledná funkce nebude interpolací. K výpočtu využívá metodu nejmenších čtverců a jako výsledek dostáváme racionální funkci R.



$R_{2,2}(x)$ $R_{3,1}(x)$

Obrázek 4.1: Srovnání racionálních interpolačních funkcí

```
function [R] = rac_int(uzly,hodnoty, m, n)
if length(uzly) ~= length(hodnoty)
    error('Vektory uzlu a hodnot musi byt stejne delky!')
end
if m + n + 1 < length(uzly)
    disp('Nedodrzena podminka: m + n + 1 = pocet uzlu.
       Vysledna funkce nebude interpolaci!')
elseif m + n + 1 > length(uzly)
    error('Nedodrzena podminka: m + n + 1 = pocet uzlu. Soucet
        je prilis vysoky!')
end

syms x y
r = [1]; % sestaveni tvaru interpolacni funkce
for i = 1:m
    r = [r, x^i];
end
for j = 1:n
    r = [r, -y*x^j];
end

A = subs(r, {x, y}, {uzly', hodnoty'});
z = inv(A'*A)*A'*hodnoty'; % MNC
f = r*z-y;
R = solve(f, y);
end
```

Kapitola 5

Interpolace různými bázovými funkcemi

Pokud provádíme interpolaci, nemusíme jako bázové funkce volit pouze polynomy nebo trigonometrické či exponenciální funkce. Můžeme také zvolit libovolnou kombinaci těchto bází, díky čemu se můžeme přizpůsobit struktuře dat a zachovat požadované vlastnosti. Je však důležité, aby zvolené bázové funkce byly definovány ve všech uzlech.

5.1 Funkce v MATLAB

Pro tuto situaci můžeme vytvořit jednoduchou MATLAB funkci, která bere za vstupy vektory uzlů a hodnot, ale také vektor bázových funkcí, které chceme pro interpolaci použít. Tento vektor je nutné zadat v symbolické proměnné. Funkci je možné použít i v případě, kdy je počet bázových funkcí menší než počet hledaných koeficientů, dostaneme však pouze aproximaci zadaných hodnot. Funkce využívá metodu nejmenších čtverců.

```
function[F] = vlastni_bazove_funkce(vektor_funkci, uzly,
   hodnoty)
vektor_funkci = unique(vektor_funkci);
if length(uzly) ~= length(hodnoty)
    error('Vektory uzlu a hodnot musi byt stejne delky')
end

if length(uzly) > length(vektor_funkci)
    disp('Vysledna funkce bude pouze aproximaci hodnot')
elseif length(uzly) < length(vektor_funkci)
    error('Pocet uzlu musi byt stejny nebo vyssi nez pocet
       funkci')
end

A = subs(vektor_funkci, uzly'); % MNC
if ~isreal(A)
    error('Nevhodne zvolena baze')
end
y = inv(A'*A)*A'*hodnoty';
F = vektor_funkci*y;
end
```

5.2 Příklad

Funkci si můžeme vyzkoušet na následujícím příkladu:

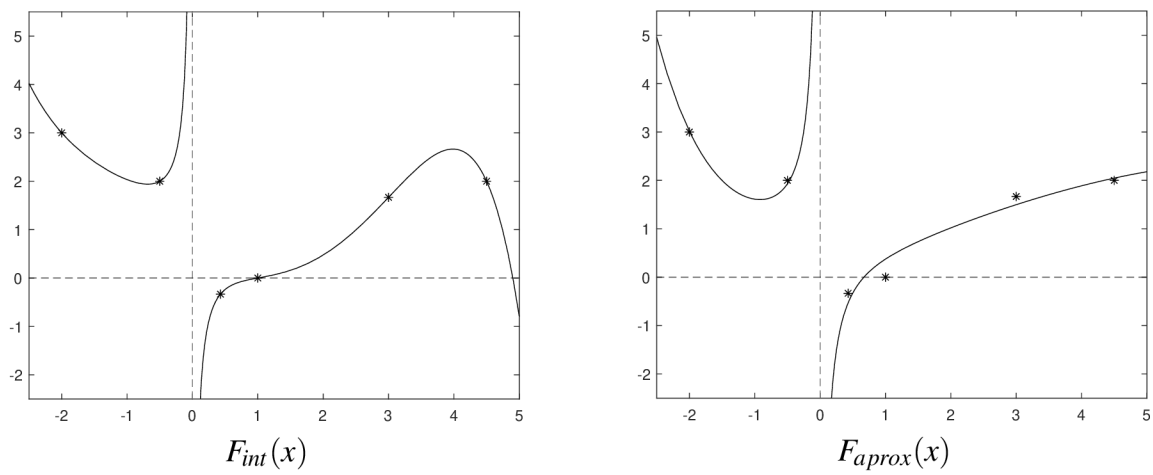
x_i	-2	$-\frac{1}{2}$	$\frac{3}{7}$	1	3	$\frac{9}{2}$
f_{x_i}	3	2	$-\frac{1}{3}$	0	$\frac{5}{3}$	2

Pro řešení si můžeme zvolit například bázové funkce: $1, x, x^3, \frac{1}{x}, e^x, e^{-x}$. Dostaneme tak interpolační funkci:

$$F_{int}(x) = 0.463706e^{-x} - 0.072538x - 0.139062e^x - 0.393836\frac{1}{x} + 0.158147x^3 + 0.51565$$

Odebereme-li z našeho vektoru bázových funkcí například funkci e^x dostaneme následnou funkci, která naše data však pouze aproximuje, a tedy neprochází přímo všemi body:

$$F_{aprox}(x) = 0.494511x + 0.461585e^{-x} - 0.567382\frac{1}{x} - 0.0037049x^3 + 0.280597$$



Obrázek 5.1: Srovnání interpolační a aproximační funkce

Obrázek názorně ukazuje, že při nedostatečném počtu bázových funkcí metoda nejmenších čtverců určuje nejlepší možné řešení, což podtrhuje výhodu tohoto přístupu.

Závěr

Cílem této bakalářské práce bylo představit vybrané metody nepolynomiální interpolace. V práci byla popsána trigonometrická, exponenciální a racionální interpolace. Pro sestavení interpolační funkce byly pro každou metodu uvedeny potřebné vzorce. K interpolaci jsme přistupovali metodou nejmenších čtverců, která při dostatečném množství parametrů zaručuje jednoznačné řešení. Tento postup lze použít i pro aproximaci nebo v případě, kdy chceme data proložit funkcí. Ve většině případů jsme dostali systém lineárních rovnic, výjimkou byla exponenciální interpolace, kde obecný postup vede na nelineární problém.

Práce zahrnuje několik konkrétních příkladů, které ilustrují praktickou aplikaci jednotlivých metod. Kromě samotného postupu výpočtů, práce upozorňuje i na potenciální úskalí, jako je například nutnost rozlišovat mezi vzorci platnými pro lichý či sudý počet uzlů, což je klíčové pro správné použití numerických metod. Pro usnadnění výpočtů byly pro vybrané postupy vytvořeny funkce v MATLAB.

Trigonometrická, exponenciální a racionální interpolace jsou pouze vybranými metodami nepolynomiální interpolace. Kromě nich existují například metody RBF interpolace, jež používají funkce založené na vzdálenosti mezi body. Dalším příkladem může být nepolynomiální spline interpolace, kterou by mohlo být vhodné použít v případě, že různé části dat vykazují různé struktury nebo vlastnosti. V oblasti polynomiální interpolace dále stojí za zmínku také polynomiální spline funkce. Mezi ně patří i lineární spline, jenž je rychlou a jednoduchou metodou interpolace, která k propojení jednotlivých bodů používá přímky.

V závěru práce jsme se podívali na to, jak můžeme různé typy interpolací kombinovat a používat tak různé bázové funkce. Tímto způsobem můžeme interpolační funkci přizpůsobit pro konkrétní data.

Seznam použité literatury

[1] BURDEN, Richard L. a J. Douglas FAIRES. *Numerical Analysis*. 9. vyd. Boston: Cengage Learning, 2010. 888 s. ISBN 978-0-538-73351-9.

[2] DAVIS, Philip J. *Interpolation and Approximation*. New York: Dover Publications, 1975. ISBN 0-486-62495-1.

[3] HOROVA, Ivana a Jiří ZELINKA. *Numerické metody*. 2. vyd. Brno: Masarykova univerzita v Brně, 2004. 294 s. 3871/Př-2/04-17/31. ISBN 80-210-3317-7.

[4] MERZLYAKOV, S. a S. POPENOV. *Interpolation by Series of Exponential Functions Whose Exponents Are Condensed in a Certain Direction*. *Journal of Mathematical Sciences*, 2021, roč. 257, s. 1–12. Dostupné z: https://doi.org/10.1007/s10958-021-05487-z. [cit. 2025-03-10].

[5] STOER, Josef a Roland BULIRSCH. *Introduction to Numerical Analysis*. 2. vyd. New York, Berlin, Heidelberg: Springer-Verlag, 1992. ISBN 3-387-97878-X.

[6] XIE, T.F. a S.P. ZHOU. On approximation by trigonometric Lagrange interpolating polynomials. *Bulletin of the Australian Mathematical Society*, 1989, roč. 40, č. 3, s. 425–428. Dostupné z: https://doi.org/10.1017/S0004972700017482. [cit. 2024-12-05].