



Jak počítá kalkulačka sinus?

Bakalářská práce

Studijní program:

B1101 Matematika

Studijní obory:

Matematika se zaměřením na vzdělávání
Informatika se zaměřením na vzdělávání

Autor práce:

Štěpánka Najmannová

Vedoucí práce:

RNDr. Martina Šimůnková, Ph.D.
Katedra aplikované matematiky





Zadání bakalářské práce

Jak počítá kalkulačka sinus?

Jméno a příjmení: Štěpánka Najmannová
Osobní číslo: P17000189
Studijní program: B1101 Matematika
Studijní obory: Matematika se zaměřením na vzdělávání
Informatika se zaměřením na vzdělávání
Zadávací katedra: Katedra matematiky a didaktiky matematiky
Akademický rok: 2018/2019

Zásady pro vypracování:

Cílem bakalářské práce je hledat odpověď na přirozenou otázku, jak pracuje kalkulačka, která dříve či později napadne každého dobrého učitele informatiky. Studentka si zopakuje, co je Taylorův interpolační polynom včetně odhadu jeho odchylky od aproximované funkce a nastuduje totéž pro interpolační polynom. Dále navrhne způsob aproximace funkce sinus těmito polynomy tak, aby při výpočtu funkční hodnoty dosáhla stejné přesnosti jako běžná kalkulačka. V práci vyzkouší interpolační polynomy s různě rozloženými uzly, speciálně s uzly v kořenech Čebyševových polynomů. Dále provede rešerši způsobů výpočtu funkce sinus a porovná svoje metody s metodami nalezenými rešerší co do přesnosti a výpočtové náročnosti. Práce bude vysázena systémem LaTeX.

Rozsah grafických prací:
Rozsah pracovní zprávy:
Forma zpracování práce:
Jazyk práce:

tištěná/elektronická
Čeština



Seznam odborné literatury:

Veselý, J.: Základy matematické analýzy, Matfyzpress, 2004.

Stoer, J., Bulirsch, R.: Introduction to Numerical Analysis, Springer, 2002.

Rybička, J.: LaTeX pro začátečníky, Konvoj, 2003.

Dokumentace systému LaTeX: <https://www.latex-project.org/help/documentation>

Vedoucí práce:

RNDr. Martina Šimůnková, Ph.D.
Katedra aplikované matematiky

Datum zadání práce:

26. dubna 2019

Předpokládaný termín odevzdání:

30. dubna 2020

prof. RNDr. Jan Pícek, CSc.
děkan

L.S.

doc. RNDr. Jaroslav Mlýnek, CSc.
vedoucí katedry

V Liberci dne 2. května 2019

Prohlášení

Prohlašuji, že svou bakalářskou práci jsem vypracovala samostatně jako původní dílo s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářské práce a konzultantem.

Jsem si vědoma toho, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu Technické univerzity v Liberci.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědoma povinnosti informovat o této skutečnosti Technickou univerzitu v Liberci; v tomto případě má Technická univerzita v Liberci právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Současně čestně prohlašuji, že text elektronické podoby práce vložený do IS/STAG se shoduje s textem tištěné podoby práce.

Beru na vědomí, že má bakalářská práce bude zveřejněna Technickou univerzitou v Liberci v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů.

Jsem si vědoma následků, které podle zákona o vysokých školách mohou vyplývat z porušení tohoto prohlášení.

1. května 2021

Štěpánka Najmannová

Jak počítá kalkulačka sinus?

Abstrakt

Tato bakalářská práce se zabývá aproximací a interpolací funkce sinus. V teoretické části se naučíme sestavit Taylorův polynom. Lagrangeův a Newtonův polynom budeme sestavovat ve vlastních nebo ekvidistantních uzlech a dále v kořenech Čebyševova polynomu. Vysvětlíme, co je systém CORDIC. Ve druhé části spočítáme jednotlivé polynomy pro konkrétní úhly. Porovnáme jednotlivé polynomy co do přesnosti, tak i co do náročnosti výpočtů.

Klíčová slova: aproximace, Taylorův polynom, Lagrangeův polynom, Newtonův polynom, CORDIC

How does calculator calculate sine?

Abstract

This bachelor thesis deals with the approximation and interpolation of the sine function. In the theoretical part we will learn to construct a Taylor polynomial. We will construct the Lagrange and Newton polynomials in eigenvalues or equidistant nodes and also in the roots of the Chebyshev polynomial. We will explain what the CORDIC system is. In the second part we calculate individual polynomials for specific angles. We will compare individual polynomials in terms of accuracy as well as in terms of complexity of calculations.

Keywords: approximation, Taylor polynomial, Lagrange polynomial, Newton polynomial, CORDIC

Poděkování

Tímto bych chtěla poděkovat své vedoucí bakalářské práce, paní RNDr. Martině Šimůnkové, Ph.D. za odborné vedení práce, cenné rady, podnětné připomínky, trpělivost, ochotu a vstřícnost při konzultacích a během zpracovávání bakalářské práce.

Obsah

1 Taylorův polynom	12
1.1 Tvar Taylorova polynomu	12
1.2 Lagrangeův tvar zbytku	13
2 Lagrangeův polynom a Newtonův polynom	16
2.1 Tvar Lagrangeova polynomu	16
2.2 Newtonův interpolační polynom	17
2.2.1 Newtonův polynom v ekvidistantních uzlech	18
2.3 Chyba aproximace interpolačním polynomem	19
3 Čebyševův polynom	21
4 CORDIC	23
4.1 Princip algoritmu CORDIC	23
4.2 Ukázka výpočtu v systému CORDIC	25
5 Vlastní výpočty	28
5.1 Taylorův polynom	28
5.2 Lagrangeův a Newtonův interpolační polynom ve vlastních uzlech	29
5.3 Newtonův interpolační polynom v ekvidistantních bodech	33
5.4 Newtonův polynom v kořenech Čebyševových polynomů	35
5.5 MATLAB	37
6 Závěr	41

Seznam obrázků

1.1	Taylorův polynom	13
2.1	Newtonův polynom 3. stupně	20
2.2	Newtonův polynom 10. stupně	20
4.1	Rotace vektoru	23
5.1	Newtonův polynom s rovnoměrně rozmístěnými úhly	31
5.2	Newtonův polynom s nerovnoměrně rozmístěnými úhly	32
5.3	Newtonovy polynomy s ekvidistantními uzly	34
5.4	Newtonovy polynomy s uzly v kořenech Čebyševova polynomu	36
5.5	Odchytky pro N_3	40
5.6	Odchytky pro N_{10}	40
5.7	Odchytky pro N_{20}	40

Seznam tabulek

1.1	Odhad maximální chyby Taylorova polynomu	15
2.1	Odhad maximální chyby interpolačního polynomu	20
4.1	Záporné mocniny čísla 2 a odpovídající úhly	25
4.2	Hodnoty konstanty k pro $i \leq 10$	26
4.3	Rotace vektoru pro úhel 40° a průběžné výsledky	27
5.1	Hodnota a odchylka $T(n)$ pro úhel $\frac{2\pi}{9}$	29
5.2	Stupeň T_n v závislosti na vzdálenosti $ x - x_0 $	30

Úvod

Když při výpočtech na kalkulačce zmáčkne libovolné tlačítko, málokdo přemýšlí nad tím, co se vlastně stane. Jak kalkulačka dokáže spočítat jednoduché nebo i složité příklady, různé funkce? Kalkulačka pracuje ve dvojkové soustavě, umí jen sčítat a násobit a funkce aproximuje na polynom. [1]

Bakalářská práce je postavena na hledání vhodné aproximace a výpočet $\sin(x)$ pro různé úhly a porovnání přesnosti výpočtu s kalkulačkou. V první kapitole se seznámíme s Taylorovým polynomem, ve druhé kapitole se budeme zabývat Lagrangeovým a Newtonovým polynomem s vlastními i ekvidistantními body a pak i s body v uzlech Čebyševových polynomů. U všech těchto aproximací nás bude zajímat i chyba aproximace, tedy jaký stupeň polynomu musíme zvolit, abychom dosáhli požadované přesnosti. Protože dnešní kalkulačky využívají systému CORDIC, v poslední teoretické kapitole si povíme i o něm.[1] V praktické části si ukážeme, jak jednotlivé druhy polynomů více či méně vhodně aproximují danou funkci, jak moc záleží na stupni zvoleného polynomu i na výběru bodů, kterými funkce prochází.

1 Taylorův polynom

Při numerickém řešení úloh matematické analýzy často nahrazujeme danou funkci f jinou funkcí φ , která danou funkci vhodně napodobuje a snadněji se s ní počítá. Tuto funkci φ nazýváme aproximací (přiblížením) funkce f . Naším cílem je zadanou funkci f nahradit jinou, jednodušší funkcí, se kterou se jednodušeji a rychleji pracuje, přesto je dostatečně přesná. K tomu se nejlépe hodí polynomy. K jejich zpracování si vystačíme s operacemi sčítání a násobení. K aproximaci v okolí bodu používáme Taylorův polynom.

1.1 Tvar Taylorova polynomu

Předpokládejme, že daná funkce f má v daném bodě x_0 a jeho okolí spojité derivace až do řádu n . Dále máme funkci φ , která danou funkci f aproximuje. Hodnoty derivací funkcí f a φ v bodě x_0 jsou stejné až do řádu n . [2]

$$\varphi^j(x_0) = f^j(x_0), \quad j = 0, 1, \dots, n$$

Tuto podmínku splňuje Taylorův polynom.

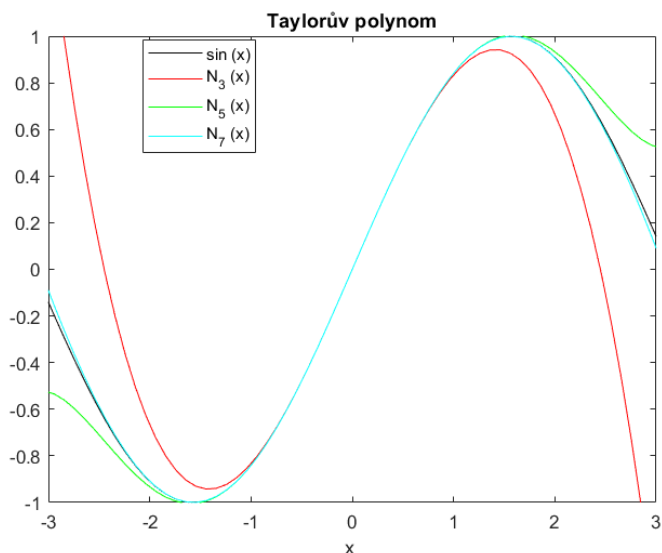
$$\begin{aligned} T_n(x) = f(x_0) + \frac{f'(x_0)}{1!}(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 + \\ \dots + \frac{f^n(x_0)}{n!}(x - x_0)^n = \sum_{k=0}^n \frac{f^k(x_0)}{k!}(x - x_0)^k \end{aligned} \quad (1.1)$$

Pokud za střed Taylorova polynomu určíme $x_0 = 0$, pak mluvíme o Maclaurinově polynomu.

$$T_n(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots = \sum_{n=0}^{\infty} (-1)^n \frac{x^{(2n+1)}}{(2n+1)!} \quad (1.2)$$

Na obrázku 1.1 je znázorněn průběh funkce $\sin(x)$ a Taylorovy polynomy různých stupňů. Na ose x jsou vyneseny úhly v intervalu $\langle -\pi, \pi \rangle$ a na ose y najdeme hodnoty funkcí pro dané x . Z obrázku je zřejmé, že s vyšším stupněm Taylorova polynomu

dochází k přiblížení k funkci $\sin(x)$ a tedy i ke zpřesnění výsledku. Dále je zřejmé, že v okolí bodu $x_0 = 0$ je aproximace Taylorovým polynomem nízkého stupně přesná. S přibývajícím vzdáleností od bodu x_0 je aproximace méně přesná a ke zpřesnění musíme použít polynom vyššího stupně. V kapitole 5.1 si ukážeme aproximaci pro konkrétní úhel a spočítáme, jaký stupeň Taylorova polynomu musíme použít pro danou vzdálenost bodu x od bodu x_0 .



Obrázek 1.1: Taylorův polynom

1.2 Lagrangeův tvar zbytku

Nechť v okolí bodu x_0 existují derivace funkce f až do stupně $n + 1$, pak

$$R_{n+1}(x) = f(x) - T_n(x) = f^{(n+1)}(\xi) \frac{(x - x_0)^{n+1}}{(n + 1)!} \quad (1.3)$$

kde ξ je číslo ležící mezi x_0 a x . Zbytek po n -tém členu nám vyjadřuje, jaké chyby jsme se pro daný stupeň aproximace polynomem dopustili. Umíme-li odhadnout $(n + 1)$ derivaci funkce f v daném okolí bodu x_0 , můžeme spočítat odhad chyby aproximace.

Platí-li pro každé x z okolí bodu x_0

$$|f^{(n+1)}(\xi)| \leq M$$

potom

$$|R_{n+1}(x)| \leq \frac{M}{(n+1)!} \cdot |x - x_0|^{n+1} \quad (1.4)$$

Protože většinou přesnou polohu bodu ξ neznáme, můžeme nepřesnost aproximace pomocí tvaru zbytku pouze odhadnout shora. M je maximum funkce $|f^{n+1}|$ na intervalu $\langle x_0, x \rangle$ respektive $\langle x, x_0 \rangle$. Protože funkce $\sin(x)$ má v absolutní hodnotě maximum v 1 a $x_0 = 0$ dostáváme pro $x \in \langle 0, \frac{\pi}{2} \rangle$

$$|R_{n+1}(x)| \leq \left| \frac{x^{n+1}}{(n+1)!} \right| \quad (1.5)$$

neboli

$$|R_{n+1}(x)| \leq \frac{(\frac{\pi}{2})^{n+1}}{(n+1)!} \quad (1.6)$$

n	$\frac{(\frac{\pi}{2})^{n+1}}{(n+1)!}$
0	1,57079632679490
1	1,23370055013617
2	0,64596409750625
3	0,25366950790105
4	0,07969262624617
5	0,02086348076335
6	0,00468175413532
7	0,00091926027484
8	0,00016044118479
9	0,00002520204237
10	0,00000359884324
11	0,00000047108748
12	0,00000005692173
13	0,00000000638660
14	0,00000000066880
15	0,00000000006566
16	0,00000000000607
17	0,00000000000053
18	0,00000000000004
19	0,00000000000000

Tabulka 1.1: Odhad maximální chyby Taylorova polynomu

2 Lagrangeův polynom a Newtonův polynom

U některých funkcí známe body, kterými daná funkce prochází a jejich funkční hodnoty. Požadujeme, aby aproximace interpolačním polynomem procházela právě těmito body.

2.1 Tvar Lagrangeova polynomu

Funkce f je dána svými hodnotami v $n+1$ bodech $x_i, i = 0, 1, \dots, n$ a hledáme funkci φ , která prochází zadanými body. Body x_i se nazývají uzly interpolace. Předpokládejme, že dané uzly jsou vzájemně různé. [2, 3, 4, 5]

$$\varphi(x_i) = f(x_i) \quad i = 0, 1, \dots, n \quad (2.1)$$

Tyto interpolační podmínky splňuje Lagrangeův interpolační polynom.

$$L_n(x) = \sum_{i=0}^n f(x_i) l_i(x) \quad (2.2)$$

l_i jsou polynomy n -tého stupně, pro které platí

$$l_i(x_k) = \begin{cases} 0 & \text{pro } k \neq i, \\ 1 & \text{pro } k = i \end{cases} \quad i, k = 0, 1, \dots, n \quad (2.3)$$

Víme, že $l_i(x)$ má kořeny $x_0, x_1, x_{i-1}, x_{i+1}, \dots, x_n$ a v bodě x_i nabývá hodnoty 1. Můžeme jej tedy zapsat ve tvaru

$$l_i(x) = \frac{(x - x_0)(x - x_1) \cdots (x - x_{i-1})(x - x_{i+1}) \cdots (x - x_n)}{(x_i - x_0)(x_i - x_1) \cdots (x_i - x_{i-1})(x_i - x_{i+1}) \cdots (x_i - x_n)} \quad (2.4)$$

2.2 Newtonův interpolační polynom

U Lagrangeova interpolačního polynomu neexistuje vztah mezi L_{n-1} a L_n . Když mezi známé uzly přidáme další, musíme všechny polynomy (2.4) spočítat znovu. Tuto nevýhodu odstraňuje Newtonův interpolační polynom.[2, 3, 4, 5]

Newtonův interpolační polynom má tvar

$$N_n(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \cdots + a_n(x - x_0)(x - x_1) \cdots (x - x_{n-1}) \quad (2.5)$$

Jsou-li uzly interpolace vzájemně různé, vstupní data nám jednoznačně určují koeficienty $a_0, a_1 \cdots a_n$, takové, že $N_n(x)$ splňuje interpolační podmínky:

$$N_n(x_i) = f(x_i), \quad i = 0, 1, \dots, n \quad (2.6)$$

Koeficienty $a_0, a_1 \cdots a_n$ lze spočítat ze soustavy rovnic

$$\begin{aligned} a_0 &= y_0 \\ a_0 + a_1(x_1 - x_0) &= y_1 \\ a_0 + a_1(x_2 - x_0) + a_2(x_2 - x_0)(x_2 - x_1) &= y_2 \\ &\vdots \\ a_0 + a_1(x_n - x_0) + \cdots + a_n(x_n - x_0) \cdots (x_n - x_{n-1}) &= y_n, \end{aligned}$$

ale jednodušší a přehlednější je vypočítat tyto koeficienty pomocí poměrných diferencí.

Poměrná diference 1. řádu:

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

Poměrná diference 2. řádu:

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

Poměrná diference n -tého řádu:

$$f[x_0, x_1, \dots, x_n] = \sum_{j=0}^n \frac{f(x_j)}{\prod_{i=0, i \neq j}^n (x_j - x_i)}, \quad i \neq j$$

Je výhodné uspořádat výpočty do tabulky poměrných diferencí

x_i	$f(x_i)$	$f[x_i, x_{i+1}]$	$f[x_i, x_{i+1}, x_{i+2}]$	\cdots
x_0	$f(x_0)$			
x_1	$f(x_1)$	$f[x_0, x_1]$		
x_2	$f(x_2)$	$f[x_1, x_2]$	$f[x_0, x_1, x_2]$	
\vdots	\vdots	\vdots	\vdots	\ddots
x_n	$f(x_n)$	$f[x_{n-1}, x_n]$	$f[x_{n-2}, x_{n-1}, x_n]$	$\cdots f[x_0, \dots, x_n]$

Podtržené poměrné difference se používají k sestrojení Newtonova interpolačního polynomu.

Výsledný sestrojený Newtonův polynom má stejný tvar jako Lagrangeův interpolační polynom stejného stupně. Důkaz najdeme v [5]

2.2.1 Newtonův polynom v ekvidistantních uzlech

V kapitole 5.2 si ukážeme, že volba vlastních úhlů pro sestrojení Newtonova nebo Lagrangeova polynomu není optimální. Použijeme pro sestrojení uzly x_0, \dots, x_n , jenž jsou ekvidistantní, tedy, že dané uzly mají od sebe konstantní vzdálenost.[5, 6]

$$x_i = x_0 + ih, \quad i = 0, 1, \dots, n \quad (2.7)$$

kde konstantu h nazýváme krok.

Pro výpočet Newtonova polynomu v ekvidistantních bodech používáme obyčejné difference. k -tá obyčejná difference bodu x_i na množině ekvidistantních uzlů má tvar

$$\Delta^k f_i = \Delta^{k-1} f_{i+1} - \Delta^{k-1} f_i, \quad i = 0, 1, \dots, n-1 \quad (2.8)$$

Tedy platí:

$$\begin{aligned} \Delta^0 f_i &= f_i \\ \Delta^1 f_i &= \Delta f_i = f_{i+1} - f_i \\ \Delta^2 f_i &= \Delta f_{i+1} - \Delta f_i = f_{i+2} - 2f_{i+1} + f_i \end{aligned}$$

Newtonův interpolační polynom v ekvidistantních uzlech má tvar

$$\begin{aligned} N_n(x) &= N_n(x_0 + sh) = f_0 + shf[x_0, x_1] + s(s-1)h^2f[x_0, x_1, x_2] + \cdots \\ &\quad \cdots + s(s-1)\cdots(s-n+1)h^n f[x_0, \dots, x_n], \end{aligned} \quad (2.9)$$

kde

$$x = x_0 + sh, \quad s \in [0, n] . \quad (2.10)$$

2.3 Chyba aproximace interpolačním polynomem

Mějme libovolný interval $\langle a, b \rangle$, který obsahuje všech $n + 1$ bodů. Nechť na tomto intervalu existuje konečná $(n + 1)$ -ní derivace funkce f . Poté pro chybu aproximace funkce f Lagrangeovým interpolačním polynomem na tomto intervalu platí [3, 5]

$$R(x) = f(x) - L_n(x) = \frac{f^{n+1}(\xi)}{(n+1)!} (x - x_0) \cdots (x - x_n) \quad (2.11)$$

kde x_0, x_1, \dots, x_n jsou uzly interpolace, $x \in \langle a, b \rangle$ a $\xi \in (a, b)$.

Platí-li

$$|f^{n+1}(\xi)| \leq M,$$

pak

$$|R(x)| \leq \frac{M}{(n+1)!} |(x - x_0) \cdots (x - x_n)| \quad (2.12)$$

Interval $\langle a, b \rangle$ proložíme ekvidistantními uzly, tj. existuje $h \neq 0$ tak, že [4]

$$x_k = x_0 + kh$$

neboli

$$x_k = a + k \frac{b - a}{n} \quad (2.13)$$

Pro funkci $\sin(x)$ platí, že má v absolutní hodnotě maximum v 1 a každý člen $|x - x_0|, |x - x_1|, \dots, |x - x_n|$ je menší nebo roven délce intervalu $\langle a, b \rangle$. Proto podle vět (2.12)(2.13) platí

$$|R(x)| \leq \frac{(b - a)^{n+1}}{(n + 1)!} \quad (2.14)$$

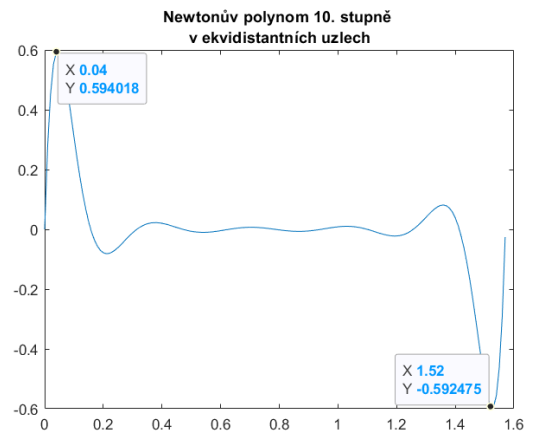
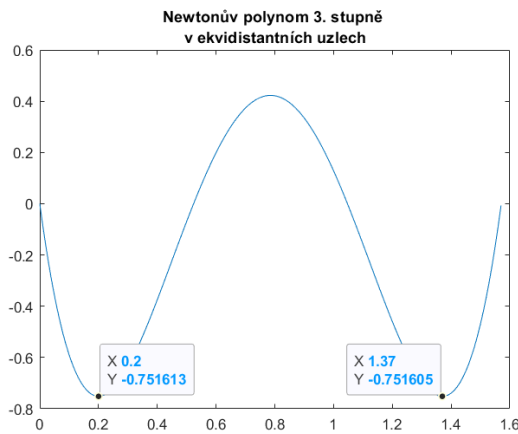
V tabulce 2.1 v prvním sloupci jsou uvedeny odchylky, pokud za maximum M zvolíme 1.

V MATLABu jsem si vykreslila graf pro Newtonův polynom 3. stupně. Na obrázku 2.1 je zvýrazněna maximální hodnota funkce $y = 10x \left(x - \frac{\pi}{2} \cdot \frac{1}{3}\right) \left(x - \frac{\pi}{2} \cdot \frac{2}{3}\right)$

$\left(x - \frac{\pi}{2} \cdot \frac{3}{3}\right)$. Tato funkce dosahuje svého maxima v absolutní hodnotě v 0,752. Pro lepší názornost je x zvětšeno desetkrát. Tuto novou hodnotu (zmenšenou desetkrát)

dosadíme do čitatele do vzorce 2.14 a v tabulce 2.1 ve druhém sloupci vidíme, že došlo ke zmenšení odhadu chyby. To samé uděláme ještě jednou pro Newtonův polynom 10. stupně. Na obrázku 2.2 je vidět průběh funkce $y = 1000x \left(x - \frac{\pi}{20}\right) \left(x - \frac{2\pi}{20}\right) \left(x - \frac{3\pi}{20}\right)$

$\left(x - \frac{4\pi}{20}\right) \left(x - \frac{5\pi}{20}\right) \left(x - \frac{6\pi}{20}\right) \left(x - \frac{7\pi}{20}\right) \left(x - \frac{8\pi}{20}\right) \left(x - \frac{9\pi}{20}\right) \left(x - \frac{\pi}{2}\right)$ a maximální hodnotu v absolutní hodnotě v 0,594 a x je opět pro lepší názornost zvětšeno, tentokrát tisíckrát. V tabulce 2.1 ve třetím sloupci jsou spočítané odhady chyby.



Obrázek 2.1: Newtonův polynom 3. stupně
 Obrázek 2.2: Newtonův polynom 10. stupně

n	$\frac{(b-a)^{n+1}}{(n+1)!}$	$\frac{0,07516}{(n+1)!}$	$\frac{0,0006}{(n+1)!}$
0	1,57079632679632	0,07516000000000	0,00060000000000
1	1,23370055013617	0,03758000000000	0,00030000000000
2	0,64596409750625	0,01252666666667	0,00010000000000
3	0,25366950790105	0,00313166666667	0,00002500000000
4	0,07969262624617	0,00062633333333	0,00000500000000
5	0,02086348076335	0,00010438888889	0,00000083333333
6	0,00468175413532	0,00001491269841	0,00000011904762
7	0,00091926027484	0,00000186408730	0,00000001488095
8	0,00016044118479	0,00000020712081	0,00000000165344
9	0,00002520204237	0,00000002071208	0,00000000016534
10	0,00000359884324	0,00000000188292	0,00000000001503
11	0,00000047108748	0,00000000015691	0,00000000000125
12	0,00000005692173	0,00000000001207	0,00000000000010
13	0,00000000638660	0,00000000000086	0,00000000000001
14	0,00000000066880	0,00000000000006	0,00000000000000
15	0,00000000006566	0,00000000000000	0,00000000000000
16	0,00000000000607	0,00000000000000	0,00000000000000

Tabulka 2.1: Odhad maximální chyby interpolačního polynomu

3 Čebyševův polynom

Při aproximaci Lagrangeovým a Newtonovým polynomem jsme dosud volili vlastní nebo ekvidistantní uzly. V kapitolách 5.2 a 5.3 ukáží, že tyto uzly nejsou zcela nejvhodnější. Jako lepší se jeví za uzly dosadit kořeny Čebyševových polynomů. [2, 7, 8]

Hledáme funkci $h(x)$, která v intervalu $\langle a, b \rangle$ minimalizuje maximální absolutní hodnotu rozdílu od funkce $f(x)$, tedy

$$\min_{h(x)}(\max_{x \in \langle a, b \rangle} |f(x) - h(x)|) \quad (3.1)$$

Pro interpolaci Čebyševovými polynomy se libovolný interval převede na interval $\langle -1, 1 \rangle$. Každému $t \in \langle a, b \rangle$ přiřadíme hodnotu $x \in \langle -1, 1 \rangle$ podle vztahu

$$x = \frac{t - \frac{1}{2}(a + b)}{b - a} \quad (3.2)$$

Čebyševovy polynomy zapisujeme ve tvaru

$$T_n(x) = \cos(n \arccos(x)) \quad (3.3)$$

Pro $n = 0$ dostáváme $T_0(x) = \cos(0 \cdot \arccos(x)) = 1$

Pro $n = 1$ dostáváme $T_1(x) = \cos(1 \cdot \arccos(x)) = x$

Pro sestrojení polynomů vyšších stupňů se využívá rekurentní vztah

$$T_{n+1}(x) = 2T_n(x)x - T_{n-1}(x) \quad n \in \mathbb{N} \quad (3.4)$$

Čebyševův polynom $T_n(x)$ má v intervalu $\langle -1, 1 \rangle$ n kořenů v bodech

$$x = \cos\left(\frac{\pi(k - \frac{1}{2})}{n}\right) \quad k = 1, 2, \dots, n \quad (3.5)$$

Pokud chceme funkci $f(x)$ aproximovat na intervalu $\langle a, b \rangle$, zvolíme za uzly body, které dostaneme pomocí předpisu

$$z = \frac{1}{2}(x(b - a) + a + b) \quad (3.6)$$

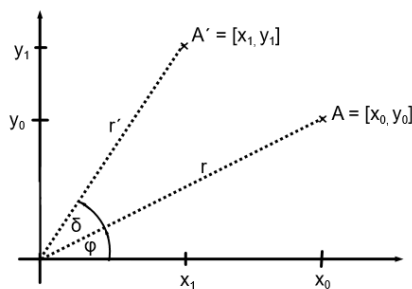
Tyto uzly pak dosadíme do Lagrangeova nebo Newtonova polynomu (2.4), (2.5)

4 CORDIC

CORDIC (z anglického COordinate Rotation DIgital Computer) je jednoduchý algoritmus především pro výpočet goniometrických funkcí. V roce 1959 ho navrhl a popsal Jack E. Volder. Je založen na sčítání, odčítání a posunu desetinné čárky ve dvojkové soustavě. [9]

4.1 Princip algoritmu CORDIC

Algoritmus CORDIC je založen na rotaci vektoru o určitý úhel δ a z následné úpravy vzorce pro rotaci tak, aby se eliminovaly složité a časově náročné multiplikační operace. Vektor \vec{r} , lze v kartézské soustavě souřadnic zapsat jako $\vec{r} = [x_0, y_0]$. Tento vektor můžeme zapsat i pomocí polárních souřadnic, ve kterých je vektor definován svou velikostí a orientovaným úhlem. Velikost vektoru \vec{r} budeme značit jako r . Na obrázku 4.1 vidíme rotaci bodu A respektive vektoru \vec{r} , vyjádřeného pomocí polárních i kartézských souřadnic.



Obrázek 4.1: Rotace vektoru

Vztah mezi polárními souřadnicemi a souřadnicemi v kartézské soustavě souřadnic vyjádříme takto

$$\begin{aligned}x_0 &= r \cdot \cos \varphi \\y_0 &= r \cdot \sin \varphi\end{aligned}\tag{4.1}$$

Rotaci o úhel δ pak lze zapsat jako

$$\begin{aligned}x_\delta &= r \cdot \cos(\varphi + \delta) \\y_\delta &= r \cdot \sin(\varphi + \delta)\end{aligned}\tag{4.2}$$

Po dosažení součtových vzorců pro $\sin \delta$ a $\cos \delta$ do (4.1) dostáváme

$$\begin{aligned}x_\delta &= r \cdot (\cos \varphi \cos \delta - \sin \varphi \sin \delta) = x_0 \cos \delta - y_0 \sin \delta \\y_\delta &= r \cdot (\sin \varphi \cos \delta + \cos \varphi \sin \delta) = x_0 \sin \delta + y_0 \cos \delta\end{aligned}\tag{4.3}$$

Vztah (4.3) dále upravíme

$$\begin{aligned}\frac{x_\delta}{\cos \delta} &= x_0 - y_0 \frac{\sin \delta}{\cos \delta} = x_0 - y_0 \tan \delta \\ \frac{y_\delta}{\cos \delta} &= y_0 + x_0 \frac{\sin \delta}{\cos \delta} = y_0 + x_0 \tan \delta\end{aligned}\tag{4.4}$$

neboli

$$\begin{aligned}x_\delta &= \cos \delta (x_0 - y_0 \tan \delta) \\y_\delta &= \cos \delta (y_0 + x_0 \tan \delta)\end{aligned}\tag{4.5}$$

Pokud volíme δ tak, že $\tan \delta$ nabývá hodnot 2^{-i} (pro $i \geq 0$), je možné $\tan \delta$ nahradit zvolenou hodnotou 2^{-i} a násobení touto hodnotou můžeme nahradit bitovým posunem. Tím jsme se zbavili násobení a algoritmus bude ještě rychlejší. Na druhou stranu máme omezený výběr úhlů, o které může vektor rotovat. To však není problém, neboť rotaci o zvolený úhel lze nahradit sérií rotací.

Tabulka 4.1 ukazuje hodnoty úhlů δ_i , po kterých se provádí rotace. Hodnoty jsou pro úsporu místa uvedeny pro $i = \{0, 1, \dots, 10\}$.

Parciální rotace můžeme provádět směrem doprava i doleva, můžeme místo hodnoty $\cos \delta$ nahradit konstantou k_i , protože platí $\cos \delta = \cos -\delta$. Místo $\tan \delta$ dosadíme mocninu dvojky 2^{-i} a směr rotace nahradíme parametrem d_i . Parametr d_i nabývá pouze hodnot $+1$ a -1 .

i	2^{-i}	úhel ve stupních	úhel v radiánech
0	1	45	0,785398
1	0,5	26,56	0,463647
2	0,25	14,03	0,244978
3	0,125	7,12	0,124354
4	0,0625	3,57	0,062418
5	0,03125	1,78	0,031239
6	0,015625	0,89	0,015623
7	0,0078125	0,44	0,007812
8	0,00390625	0,22	0,003906
9	0,001953125	0,11	0,001953
10	0,0009765625	0,05	0,000976

Tabulka 4.1: Záporné mocniny čísla 2 a odpovídající úhly

$$\begin{aligned}x_{\delta} &= k_i(x_0 - y_0 \cdot d_i \cdot 2^{-i}) \\y_{\delta} &= k_i(y_0 + x_0 \cdot d_i \cdot 2^{-i})\end{aligned}\tag{4.6}$$

Pro konstantu k_i platí

$$k_i = \cos(\arctan 2^{-i}) = \frac{1}{\sqrt{1+2^{-2i}}}$$

Během výpočtu můžeme konstantu k ignorovat a vynásobit jí výsledek až na konec. Součin konstant velmi rychle konverguje k 0,60725293501 (této hodnoty dosáhneme při $i = 18$), takže pro větší počet iterací můžeme do paměti uložit tuto jedinou hodnotu a tím zmenšit paměťové nároky.

V tabulce 4.2 jsou v druhém sloupci uvedeny hodnoty pro jednotlivé konstanty k_i a ve třetím sloupci hodnoty pro součin konstant od k do k_i .

4.2 Ukázka výpočtu v systému CORDIC

Tabulka 4.3 nám ukazuje, jak během jednotlivých rotací o úhel δ dochází k přibližování k námi zvolenému úhlu 40° . Ve vedlejším sloupci jsou hodnoty funkce sin pro jednotlivé úhly. Přesnosti jedenácti desetinných míst dosáhneme při 43. iteraci. Výpočty

i	k_i	součin k
0	0,70710678	0,70710678
1	0,89442719	0,63245553
2	0,97014250	0,61357199
3	0,99227787	0,60883391
4	0,99805257	0,60764825
5	0,99951207	0,60735177
6	0,99987795	0,60727764
7	0,99996948	0,60725911
8	0,99999237	0,60725447
9	0,99999809	0,60725332
10	0,99999952	0,60725303

Tabulka 4.2: Hodnoty konstanty k pro $i \leq 10$

byly prováděny v excelu.

Poznámka

Algoritmus je funkční pro úhly v prvním a čtvrtém kvadrantu. Pro úhly z druhého a třetího kvadrantu je nutné vstupní úhel δ posunout do prvního nebo čtvrtého kvadrantu a ve výsledku pak převrátit znaménko.

i	dílčí φ ve stupních	\sin (dílčí φ)
0	45	0.70710678119
1	18.43494882	0.31622776602
2	32.47119229	0.53687549219
3	39.59620863	0.63737300217
4	43.17254301	0.68419769479
5	41.38263240	0.66108446001
6	40.48745869	0.64928158976
7	40.03984452	0.64332017543
8	39.81603402	0.64032467590
9	39.92793970	0.64182365555
10	39.9838925	0.64257222831
11	40.01186904	0.64294628502
12	39.99788081	0.64275927581
13	40.00487493	0.64285278521
14	40.00137787	0.64280603170
15	39.99962934	0.64278265406
16	40.00050361	0.64279434295
17	40.00006648	0.64278849852
18	39.99984791	0.64278557629
19	39.99995719	0.64278703741
20	40.00001183	0.64278776797

Tabulka 4.3: Rotace vektoru pro úhel 40° a průběžné výsledky

5 Vlastní výpočty

Pro zobrazení výpočtů a porovnání výsledků výpočtů s kalkulačkou jsem zvolila úhel 40° ($\frac{2\pi}{9}$). Přesnost výpočtů je 11 desetinných míst. Hodnota funkce $\sin(\frac{2\pi}{9})$ na kalkulačce je 0,64278760968. Pro Taylorův polynom jsem zvolila střed $x_0 = 0$.

5.1 Taylorův polynom

Podle věty (1.2) sestrojíme Taylorův polynom stupně 1.

$$T_1(\frac{2\pi}{9}) = \frac{2\pi}{9} = 0,69813170079$$

Odchylka od výsledku na kalkulačce je 0,05534409111.¹

Podle věty (1.2) sestrojíme Taylorův polynom stupně 3.

$$T_3(\frac{2\pi}{9}) = \frac{2\pi}{9} - \frac{(\frac{2\pi}{9})^3}{3!} = 0,64142154683$$

Odchylka od výsledku na kalkulačce je 0,00136606285.

Vidíme, že dochází ke zpřesňování výsledku. Pokračujeme ve výpočtu zvyšováním stupně Taylorova polynomu. Další výsledky i odchylku od kalkulačky znázorňuje tabulka 5.1.

V tomto případě nám k požadované přesnosti stačí Taylorův polynom 11. stupně.

Jak to bude vypadat, pokud zvolíme jiný úhel?[7]

Na obrázku 1.1 vidíme, že kolem bodu $x_0 = 0$ dosahuje Taylorův polynom výborných výsledků pro jakýkoliv stupeň. A pokud požadujeme stále stejnou přesnost i pro vzdálenější body, musíme stupeň Taylorova polynomu zvyšovat. V tabulce 5.2 jsou znázorněné Taylorovy polynomy zvětšujícího stupně a pro jakou vzdálenost bodu x od bodu x_0 jsou vhodné pro námi zvolenou přesnost jedenácti desetinných míst. Z této tabulky také vyčteme, že pro celý interval $\langle 0, \frac{\pi}{2} \rangle$ nám stačí Taylorův polynom 17. stupně.

¹Nerozlišujeme kladnou nebo zápornou odchylku.

n	T_n	Odchylka
1	0,69813170079	0,0553440911
3	0,64142154683	0,00136606285
5	0,64280353889	0,00001592921
7	0,64278750161	0,00000010807
9	0,64278761017	0,00000000049
11	0,64278760969	0,00000000001
13	0,64278760969	0,00000000001

Tabulka 5.1: Hodnota a odchylka $T(n)$ pro úhel $\frac{2\pi}{9}$

5.2 Lagrangeův a Newtonův interpolační polynom ve vlastních uzlech

Nejprve sestojíme Lagrangeův polynom pro 4 uzly, za které jsem si zvolila úhly $\frac{\pi}{6}$, $\frac{\pi}{4}$, $\frac{\pi}{3}$ a $\frac{\pi}{2}$. Podle vzorce (2.4) sestojíme čtyři polynomy.

Pro úhel $\frac{\pi}{6}$:

$$L_0(x) = \frac{(x - \frac{\pi}{4})(x - \frac{\pi}{3})(x - \frac{\pi}{2})}{(\frac{\pi}{6} - \frac{\pi}{4})(\frac{\pi}{6} - \frac{\pi}{3})(\frac{\pi}{6} - \frac{\pi}{2})} = -\frac{9 \cdot (3x - \pi)(2x - \pi)(4x - \pi)}{\pi^3}$$

Pro úhel $\frac{\pi}{4}$:

$$L_1(x) = \frac{(x - \frac{\pi}{6})(x - \frac{\pi}{3})(x - \frac{\pi}{2})}{(\frac{\pi}{4} - \frac{\pi}{6})(\frac{\pi}{4} - \frac{\pi}{3})(\frac{\pi}{4} - \frac{\pi}{2})} = \frac{16 \cdot (3x - \pi)(2x - \pi)(6x - \pi)}{\pi^3}$$

Pro úhel $\frac{\pi}{3}$:

$$L_2(x) = \frac{(x - \frac{\pi}{4})(x - \frac{\pi}{6})(x - \frac{\pi}{2})}{(\frac{\pi}{3} - \frac{\pi}{4})(\frac{\pi}{3} - \frac{\pi}{6})(\frac{\pi}{3} - \frac{\pi}{2})} = -\frac{9 \cdot (6x - \pi)(2x - \pi)(4x - \pi)}{\pi^3}$$

Pro úhel $\frac{\pi}{2}$:

$$L_3(x) = \frac{(x - \frac{\pi}{4})(x - \frac{\pi}{3})(x - \frac{\pi}{6})}{(\frac{\pi}{2} - \frac{\pi}{4})(\frac{\pi}{2} - \frac{\pi}{3})(\frac{\pi}{2} - \frac{\pi}{6})} = \frac{(3x - \pi)(6x - \pi)(4x - \pi)}{\pi^3}$$

T_n	$ x - x_0 $	$R_{n+1}(x)$
T_3	$\frac{\pi}{180}$	$6,7 \cdot 10^{-11}$
T_5	$\frac{5\pi}{180}$	$5,4 \cdot 10^{-11}$
T_7	$\frac{13\pi}{180}$	$4,0 \cdot 10^{-11}$
T_9	$\frac{23\pi}{180}$	$1,2 \cdot 10^{-11}$
T_{11}	$\frac{38\pi}{180}$	$1,0 \cdot 10^{-11}$
T_{13}	$\frac{60\pi}{180}$	$2,3 \cdot 10^{-11}$
T_{15}	$\frac{80\pi}{180}$	$1,4 \cdot 10^{-11}$
T_{17}	$\frac{90\pi}{180}$	$1,0 \cdot 10^{-12}$

Tabulka 5.2: Stupeň T_n v závislosti na vzdálenosti $|x - x_0|$

Nyní můžeme sestavit Lagrangeův polynom 3. stupně podle vzorce (2.2).

$$\mathbf{L}(x) = -\frac{1}{2} \cdot \frac{9 \cdot (6x - \pi)(2x - \pi)(4x - \pi)}{\pi^3} + \frac{\sqrt{2}}{2} \cdot \frac{16 \cdot (3x - \pi)(2x - \pi)(6x - \pi)}{\pi^3} - \frac{\sqrt{3}}{2} \cdot \frac{9 \cdot (6x - \pi)(2x - \pi)(4x - \pi)}{\pi^3} + \frac{(3x - \pi)(6x - \pi)(4x - \pi)}{\pi^3}$$

Za x dosadíme úhel $\frac{2\pi}{9}$ a polynom spočítáme.

$$\mathbf{L}(x) = \frac{5}{54} + \frac{40\sqrt{2}}{81} - \frac{5\sqrt{3}}{54} + \frac{1}{81} = \frac{17 + 80\sqrt{2} - 15\sqrt{3}}{162} = 0,64294026466$$

Nyní pro stejné úhly sestavíme Newtonův polynom. Začneme poměrnými diferencemi, které uspořádáme do tabulky (2.2).

x_i	$f(x_i)$	$f[x_i, x_{i+1}]$	$f[x_i, x_{i+1}, x_{i+2}]$	$f[x_i, x_{i+1}, x_{i+2}, x_{i+3}]$
$\frac{\pi}{6}$	$\frac{1}{2}$			
$\frac{\pi}{4}$	$\frac{\sqrt{2}}{2}$	0,7910		
$\frac{\pi}{3}$	$\frac{\sqrt{3}}{2}$	0,6070	-0,3515	
$\frac{\pi}{2}$	1	0,2558	-0,4471	-0,0912

Poměrné diference dosadíme do vzorce (2.5)

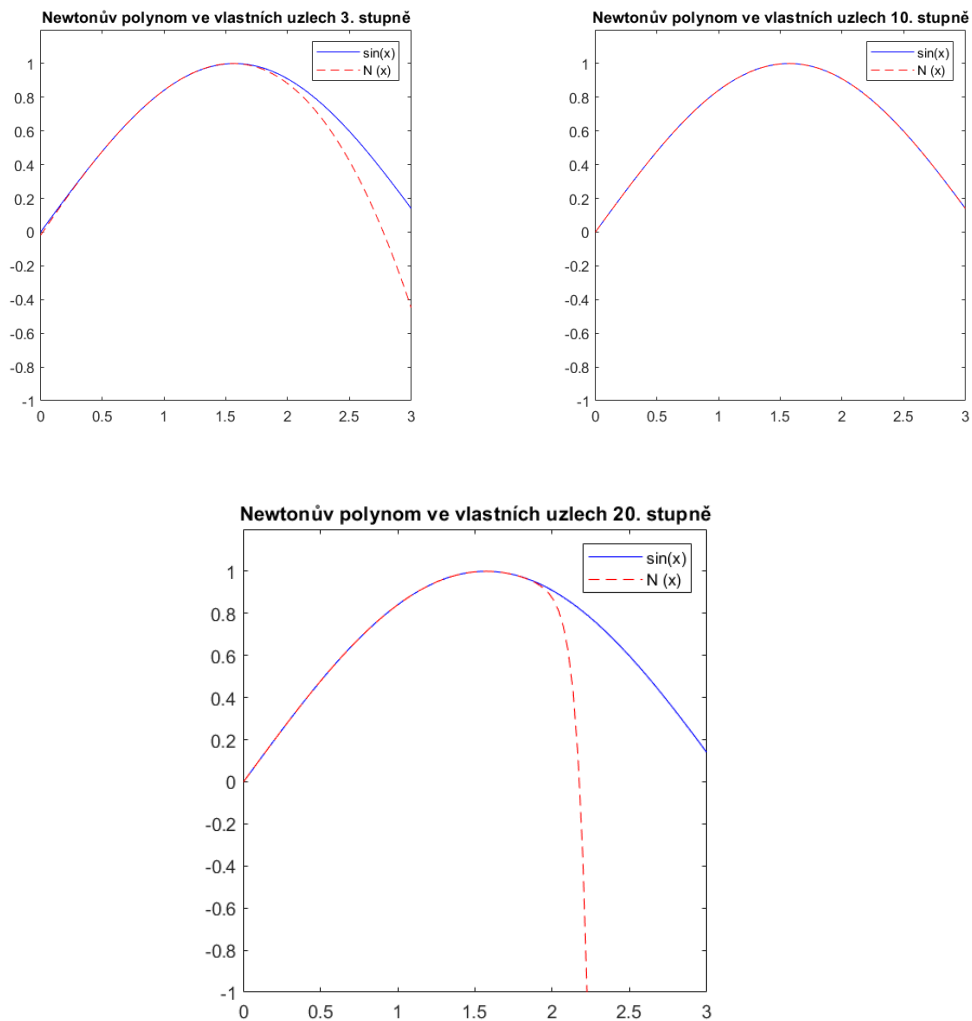
$$\begin{aligned} N(x) &= \frac{1}{2} + 0,7910(x - \frac{\pi}{6}) - 0,3515(x - \frac{\pi}{6})(x - \frac{\pi}{4}) - 0,0912(x - \frac{\pi}{6})(x - \frac{\pi}{4})(x - \frac{\pi}{3}) = \\ &= -0,912x^3 + 1,7973x^2 - 0,3740x + 0,3340 \end{aligned}$$

Za x dosadíme úhel $\frac{2\pi}{9}$ a polynom spočítáme.

$$N\left(\frac{2\pi}{9}\right) = -0,912\left(\frac{2\pi}{9}\right)^3 + 1,7973\left(\frac{2\pi}{9}\right)^2 - 0,3740\frac{2\pi}{9} + 0,3340 = 0,64294026466$$

Vidíme, že oba polynomy nám daly stejný výsledek a odchylka od hodnoty $\sin\left(\frac{2\pi}{9}\right)$ je

$$|e| = N_3\left(\frac{2\pi}{9}\right) - \sin\left(\frac{2\pi}{9}\right) = 0,00015265498$$

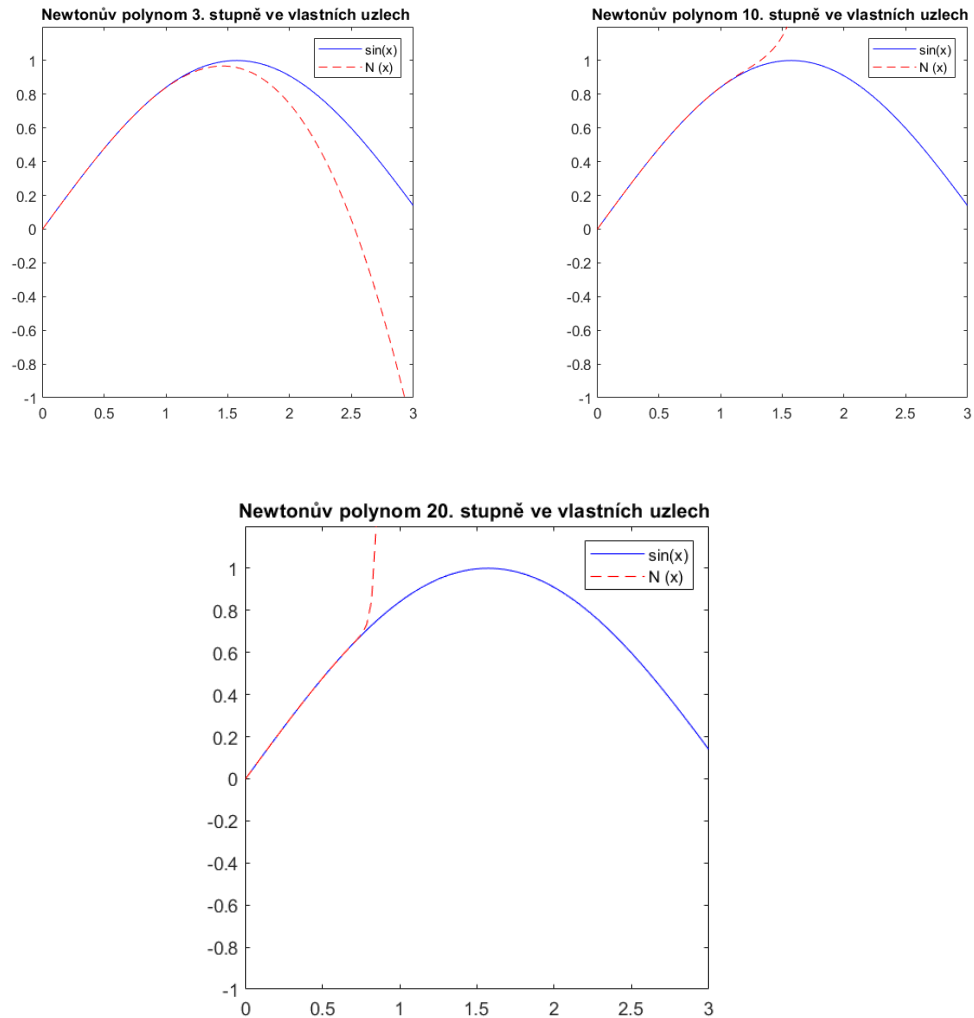


Obrázek 5.1: Newtonův polynom s rovnoměrně rozmístěnými úhly

Pro sestavení grafů na obrázku 5.1 jsem pro Newtonův polynom 3. stupně zvolila úhly 30° , 45° , 60° a 90° . Pro sestavení grafu Newtonova polynomu 10. stupně jsem přidala úhly 15° , 75° , 25° , 80° , 5° , 55° , 10° . I pro sestavení Newtonova polynomu

20. stupně jsem úhly přidávala rovnoměrně po celém intervalu $\langle 0, \frac{\pi}{2} \rangle$.

Jaká nastane situace, pokud uzly nerozmístíme po intervalu rovnoměrně, ale použijeme např. úhly jen z první poloviny intervalu? Jak se změní grafy pro jednotlivé stupně Newtonova polynomu, ukazují grafy na obrázku 5.2. Z grafů je možné vyčíst, že nyní aproximace funkce $\sin(x)$ Newtonovým polynomem není již tak přesná. I když zvyšujeme stupeň interpolačního polynomu, nedochází k lepší aproximaci, ba naopak.



Obrázek 5.2: Newtonův polynom s nerovnoměrně rozmístěnými úhly

5.3 Newtonův interpolační polynom v ekvidistantních bodech

Mějme čtyři uzly, které mají mezi sebou konstantní vzdálenost. Ze vzorce (2.2.1) spočítáme jednotlivé uzly. Za konstantu h zvolíme $\frac{\pi}{6}$.²

$$\begin{aligned}x_0 &= 0 \\x_1 &= 0 + h = \frac{\pi}{6} \\x_2 &= 0 + 2h = \frac{\pi}{3} \\x_3 &= 0 + 3h = \frac{\pi}{2}\end{aligned}$$

Spočítáme obyčejné difference, které uspořádáme do tabulky.

x_i	$f(x_i)$	$f[x_i, x_{i+1}]$	$f[x_i, x_{i+1}, x_{i+2}]$	$f[x_i, x_{i+1}, x_{i+2}, x_{i+3}]$
0	0			
$\frac{\pi}{6}$	$\frac{1}{2}$	0,9549		
$\frac{\pi}{3}$	$\frac{\sqrt{3}}{2}$	0,6990	-0,2443	
$\frac{\pi}{2}$	1	0,2558	-0,4232	-0,1138

Ze vztahu (2.10) spočítáme s .

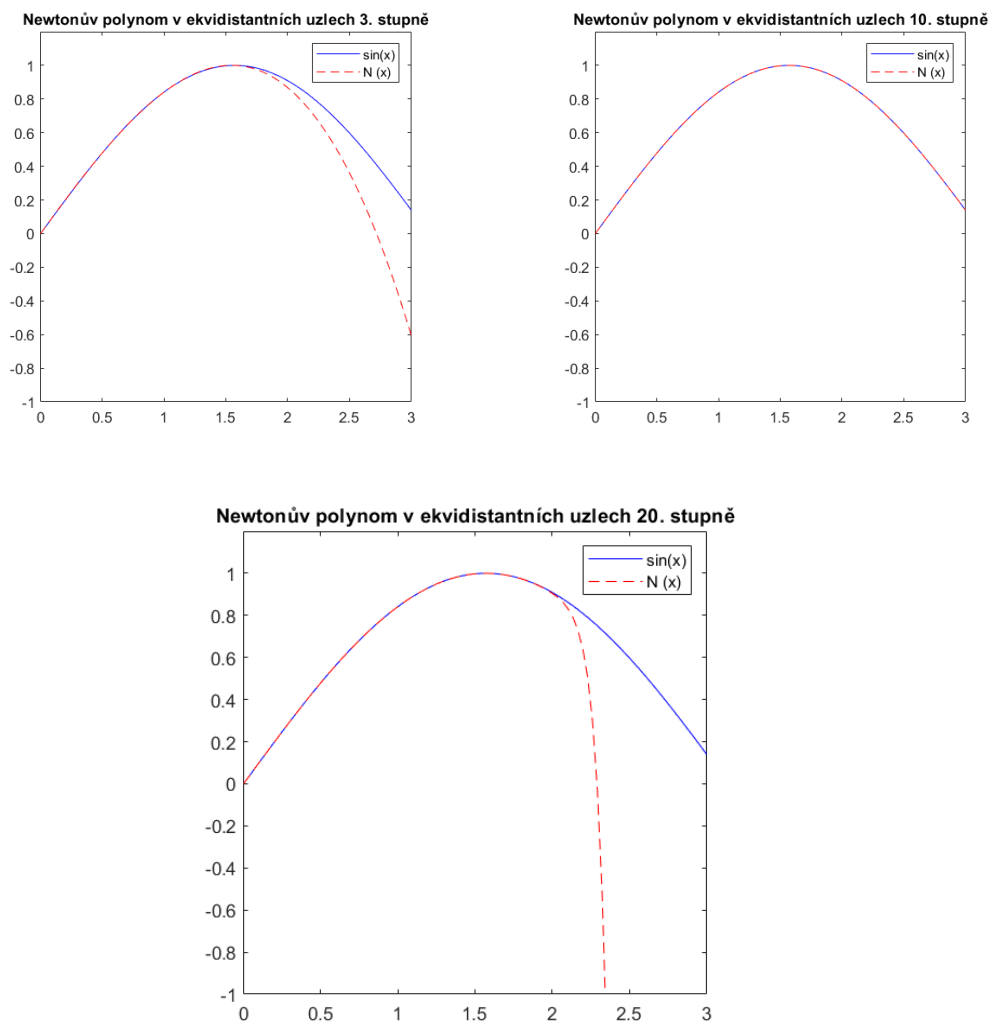
$$\begin{aligned}x &= x_0 + sh \\ \frac{2\pi}{9} &= 0 + s \cdot \frac{\pi}{6} \\ s &= \frac{4}{3}\end{aligned}$$

Obyčejné difference, h a s dosadíme do vzorce (2.9) a spočítáme.

$$\begin{aligned}N_3\left(\frac{2\pi}{9}\right) &= 0 + \frac{4}{3} \cdot \frac{\pi}{6} \cdot 0,9549 - \frac{4}{3} \cdot \left(\frac{4}{3} - 1\right) \cdot \left(\frac{\pi}{6}\right)^2 \cdot 0,2443 - \\ &\quad - \frac{4}{3} \cdot \left(\frac{4}{3} - 1\right) \cdot \left(\frac{4}{3} - 2\right) \cdot \left(\frac{\pi}{6}\right)^3 \cdot 0,1138 = 0,64173780387\end{aligned}$$

Absolutní odchylka od kalkulačkové hodnoty je

$$|e| = N_3\left(\frac{2\pi}{9}\right) - \sin\left(\frac{2\pi}{9}\right) = 0,00104980582$$



Obrázek 5.3: Newtonovy polynomy s ekvidistantními uzly

Vidíme, že v tomto případě interpolace Newtonovým polynomem v ekvidistantních uzlech vychází hůře, než interpolace ve vlastních uzlech.

Na obrázcích 5.3 máme srovnání průběhů Newtonových polynomů v ekvidistantních uzlech. V porovnání s grafy interpolačních polynomů ve vlastních uzlech došlo k lepší aproximaci.

²Interval $\langle 0, \frac{\pi}{2} \rangle$ chceme rozdělit na 3 intervaly se čtyřmi úhly

5.4 Newtonův polynom v kořenech Čebyševových polynomů

Pro výpočet Newtonova polynomu $N_3(x)$ potřebujeme 4 kořeny Čebyševových polynomů. Do vzorců (3.5) a (3.6) dosadíme za $k = 1, 2, 3, 4$, za interval $\langle a, b \rangle$ zvolíme interval $\langle 0, \frac{\pi}{2} \rangle$ a spočítáme.

$$\begin{aligned} x_1 &= \cos\left(\frac{\pi}{4}\right) = 0,9238 & z_1 &= \frac{1}{2}\left(0,9238\frac{\pi}{2} + \frac{\pi}{2}\right) = 1,5110 \\ x_2 &= \cos\left(\frac{3\pi}{4}\right) = 0,3826 & z_2 &= \frac{1}{2}\left(0,3826\frac{\pi}{2} + \frac{\pi}{2}\right) = 1,0859 \\ x_3 &= \cos\left(\frac{5\pi}{4}\right) = -0,3826 & z_3 &= \frac{1}{2}\left(-0,3826\frac{\pi}{2} + \frac{\pi}{2}\right) = 0,4848 \\ x_4 &= \cos\left(\frac{7\pi}{4}\right) = -0,9238 & z_4 &= \frac{1}{2}\left(-0,9238\frac{\pi}{2} + \frac{\pi}{2}\right) = 0,0597 \end{aligned}$$

Sestrojíme tabulku poměrných diferencí.

x_i	$f(x_i)$	$f[x_i, x_{i+1}]$	$f[x_i, x_{i+1}, x_{i+2}]$	$f[x_i, x_{i+1}, x_{i+2}, x_{i+3}]$
1,5110	0,9982			
1,0859	0,8847	0,2669		
0,4848	0,4660	0,6965	-0,4186	
0,0597	0,0597	0,9559	-0,2527	-0,1142

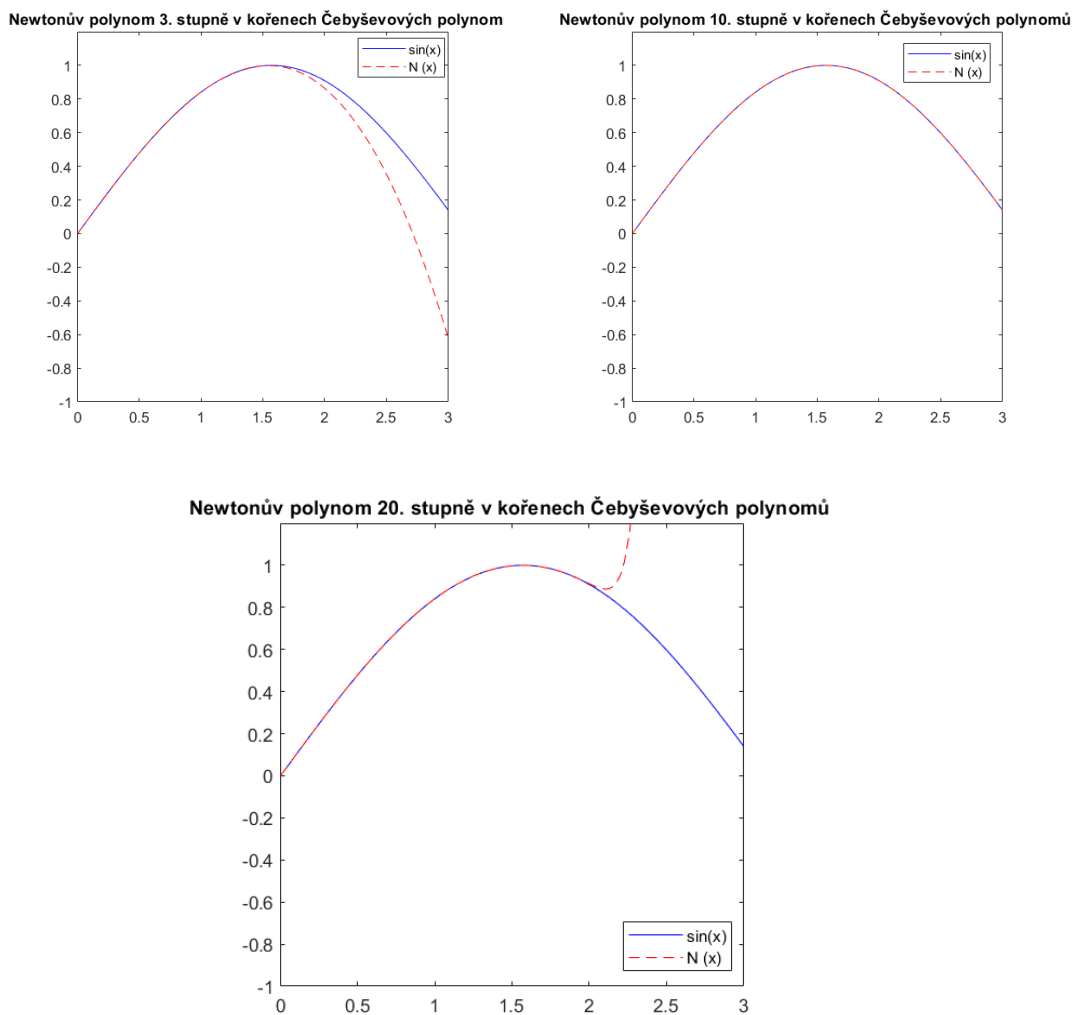
Po dosazení do vzorce pro Newtonův nebo Lagrangeův polynom (2.4), (2.5) získáváme

$$\begin{aligned} N(x) &= 0,9982 + 0,2669(x - 1,5110) - 0,4186(x - 1,5110)(x - 1,0859) - \\ &\quad - 0,1142(x - 1,5110)(x - 1,0859)(x - 0,4848) = \\ &= -0,1142x^3 - 0,0664x^2 + 1,0227x - 0,0011 = 0.64157078358 \end{aligned}$$

Absolutní odchylka od kalkulačkové hodnoty je

$$|e| = N_3\left(\frac{2\pi}{9}\right) - \sin\left(\frac{2\pi}{9}\right) = 0.00121682611$$

Na obrázcích 5.4 vidíme průběh Newtonova polynomu, pokud za uzly zvolíme kořeny Čebyševova polynomu.



Obrázek 5.4: Newtonovy polynomy s uzly v kořenech Čebyševova polynomu

Z grafů 5.3 a 5.4 se zdá, že mezi aproximací polynomy s ekvidistantními uzly nebo kořeny Čebyševových polynomů není žádný velký rozdíl. Pojd'me se podívat, jak to vypadá s odchylkami od kalkulačkové hodnoty.

Na obrázcích 5.5, 5.6 a 5.7 jsou znázorněné odchylky na intervalu $\langle 0, \frac{\pi}{2} \rangle$. Na ose x jsou vyneseny úhly z tohoto intervalu, u kterých vidíme, jaké odchylky dosahují od kalkulačkové hodnoty (na ose y). Úhly jsem volila tak, aby žádný úhel neležel v ekvidistantním uzlu či v kořenu Čebyševova polynomu. Pokud bych tento úhel zvolila, jeho odchylka by byla nulová a tento úhel(respektive jeho odchylka) by zmizel z grafu.

Vidíme, že uprostřed intervalu dosahují oba druhy polynomů podobných odchylek. Rozdíl mezi Newtonovým polynomem v ekvidistantních uzlech a Newtonovým polynomem v kořenech Čebyševova polynomu je na okrajích intervalu, kde polynom s ekvidistantními uzly neaproximuje funkci $\sin(x)$ tak přesně. Přestože v některých úhlech je aproximace polynomem s Čebyševovými kořeny horší, v rámci celého intervalu netrpí velkými výkyvy v přesnosti. V jeho případě se můžeme spolehnout na stejnou přesnost v celém námi zvoleném intervalu. Jeho rozptyl odchylek je u $N_3(x)$ mezi 10^{-3} a 10^{-4} , u $N_{10}(x)$ mezi 10^{-12} a 10^{-14} a u $N_{20}(x)$ je to mezi 10^{-15} a 10^{-16} .

5.5 MATLAB

Veškeré výpočty a grafy byly prováděny v Excelu a v MATLABu.

Pro vykreslení grafu Taylorova polynomu různých stupňů byl použit následující kód:

```
x=-pi:0.1:pi
y=sin(x)
z=x-x.^3/factorial(3)
a=x-x.^3/factorial(3)+x.^5/factorial(5)
b=x-x.^3/factorial(3)+x.^5/factorial(5)-x.^7/factorial(7)
axis([-3 3 -1 1])
plot(x,y,'b')
hold on
plot(x,z,'r')
plot(x,a,'g')
plot(x,b,'g')
hold off
```

Pro výpočet Newtonova polynomu různých stupňů a různě rozložených uzlů by kód pro výpočet a vykreslení grafu mohl vypadat takto:[\[10\]](#)

Máme funkci, která prochází body

$$[x_k, y_k] = [x_k, f(x_k)] \quad k = 0, 1, \dots, N$$

Tvar Newtonova polynomu je

$$N(x) = d_{0,0} + d_{1,1} \cdot (x - x_0) + \dots + d_{N,N} \cdot (x - x_0) \cdot \dots \cdot (x - x_{N-1}),$$

kde

$$d_{k,0} = y_k \text{ a } d_{k,j} = \frac{d_{k,j-1} - d_{k-1,j-1}}{x_k - x_{k-j+1}}$$

```

function [C,D]=Newton_e(~,~)

spojeni = ddeinit('excel', 'taylorr.xlsx')
X=ddereq(spojeni,'r3c4:r32c4')
Y=ddereq(spojeni,'r3c5:r32c5')

n=length(X);

D = zeros (n,n);
D (:,1) = Y

for j = 2: n
    for k = j: n
        D (k,j) = (D (k, j-1) -D (k-1, j-1)) / (X (k) -X (k-j + 1))
    end
end
C=D (n, n)
for k = (n-1) :-1: 1
    C=conv(C,poly (X (k)))
    m=length (C)
    C (m)=C (m) +D (k,k)
end

x = 0 : pi/100 : pi;
e = polyval(C, x);
format long
axis square, axis([0 3 -1 1.2])
x = 0 : pi/100 : pi; y =sin(x);
plot (x, y, 'b')
hold on
axis square, axis([0 3 -1 1.2])
x = 0 : pi/100 : pi
plot (x, e, 'r--')

hold off

```

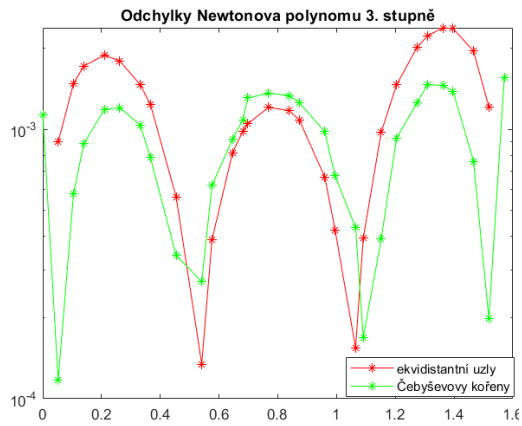
Pro znázornění rozdílů mezi interpolačním polynomem v ekvidistantních uzlech a v kořenech Čebyševova polynomu byl použit tento kód:

```

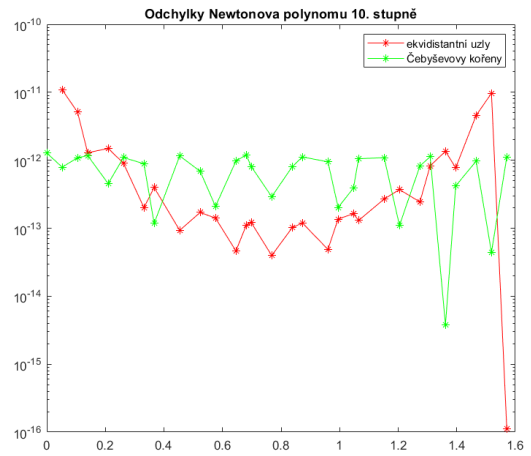
spojeni = ddeinit('excel', 'taylorr.xlsx')
x=ddereq(spojeni,'r35c6:r35c36')
a=ddereq(spojeni,'r38c6:r38c36')
b=ddereq(spojeni,'r45c6:r45c36')
axis([0 1.6 0.00000000000000000001 0.1])

```

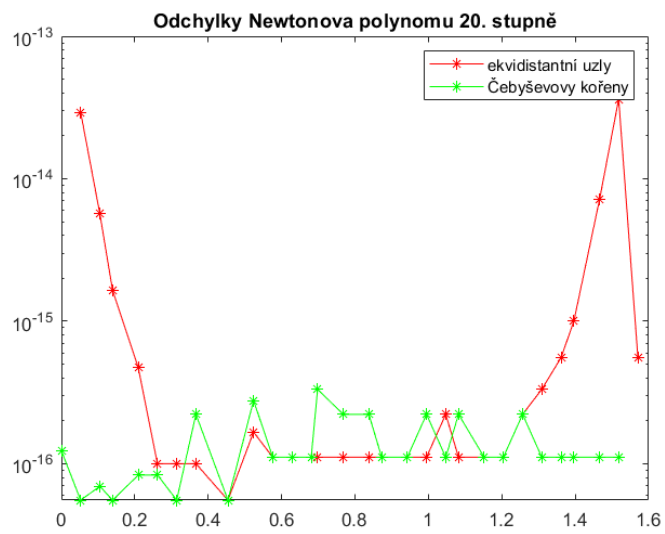
```
plot (x,a,'r*-')
hold on
plot(x,b,'g*-')
hold off
title('Odchylky Newtonova polynomu 3. stupně')
legend('ekvidistantní uzly','Čebyševovy kořeny')
```



Obrázek 5.5: Odchyly pro N_3



Obrázek 5.6: Odchyly pro N_{10}



Obrázek 5.7: Odchyly pro N_{20}

6 Závěr

V kapitolách 1.1 a 5.1 jsme dokázali, že Taylorův polynom je vhodný pro aproximaci v okolí určitého bodu. Pokud za střed Taylorova polynomu zvolíme bod $x_0 = 0$, je jeho výpočet poměrně jednoduchý a rychlý. Pro vzdálenější body se více hodí Lagrangeův a Newtonův polynom. Protože oba polynomy jsou totéž, je jednodušší zvolit Newtonův polynom. Pokud experimentujeme s přidáváním bodů, je Newtonův polynom vhodnější, protože nový bod lehce přidáme k ostatním a nemusíme vše celé přepočítávat jako u Lagrangeova polynomu. Tento problém by však určitě vyřešilo naprogramovat si kód pro výpočet těchto polynomů v nějakém programovacím jazyce. Dospěli jsme k závěru, že záleží nejen na volbě stupně polynomu, ale také na rozložení uzlů v intervalu. Jako nejlépe se jeví, vybrat si za uzly kořeny Čebyševových polynomů. S těmito uzly dosahuje interpolační polynom nejlepších výsledků a to na celém intervalu.

Literatura

- [1] Woodford, Chris. *Kalkulačky*. [online] 10. 11. 2020 [vid. 2021-03-06]
Dostupné z <https://www.explainthatstuff.com/calculators.html>.
- [2] DANĚK J. *Numerické metody*. ČZU [online]. Plzeň, 2017 [vid. 2021-03-01]
Dostupné z:
https://home.zcu.cz/~danek/DATA/WWW_STRANKY/soubory/prednasky/
- [3] VOTÍPKA, M. *Aproximace funkce polynomem*. Vysoká škola báňská -
Technická univerzita Ostrava. *Bakalářská práce*. [online]. Ostrava, 2013 [vid.
2021-03-16].
Dostupné z: <https://theses.cz/id/7hk51s>
- [4] ČERMÁK, L., HLAVIČKA, R. *Numerické metody* [online] 1. 11. 2006 [vid.
2021-03-28] dostupné z:
<https://mathonline.fme.vutbr.cz/UploadedFiles/240.pdf>
- [5] HASÍK, K. *Numerické metody*. [online]. Opava [vid. 2021-03-16]
Dostupné z:
<https://www.slu.cz/file/cul/3f2481de-0419-42ec-91b1-ab2cdde89f45>
- [6] PŘIKRYL, P. *Numerické metody matematické analýzy*. Státní nakladatelství
technické literatury, Praha, 1985
- [7] HOROVÁ, I., ZELINKA, J. *Numerické metody* 2. rozš.vyd. Brno: Masarykova
univerzita v Brně, 2004. ISBN 978-80-210-3317-7
- [8] LIMPOUCH, J., *Čebyševovy aproximace*. [online] [vid. 2021-04-05]
<http://www-troja.fjfi.cvut.cz/~limpouch/numet/aprox/node14.html>, 2000.
- [9] TIŠNOVSKÝ P. *Výpočet goniometrických funkcí algoritmem CORDIC* [online]
15. 6. 2006 [vid. 2021-04-02] dostupné z
<https://www.root.cz/clanky/vypocet-goniometrickych-funkci-algoritmem-cordic/>
- [10] KULIČKA, J. *Aproximace křivek v MATLABU – Newtonův interpolační
polynom*. [online] 18. 11. 2010 [vid. 2021-03-31] Dostupné z
<https://pernerscontacts.upce.cz/index.php/perner/article/view/1005>